Universitat de Girona
Escola Politècnica Superior

Treball final de grau

Estudi: Grau en Tecnologies Industrials

Títol: Programació d'un editor d'esquemes musicals per a mòbils

Document: Memòria

Alumne: David Ruscalleda Gómez

Tutor: Esteban Fermín del Acebo Peña
Departament: Informàtica, Matemàtica Aplicada i Estadística
Àrea: Llengües i sistemes informàtics

Convocatòria (mes/any) Setembre del 2016

1 ÍNDEX

2 ÍNDEX DE FIGURES

3 INTRODUCCIÓ

3.1 Antecedents

Entenem per harmonia allò que dóna context i sentit estructural a una peça musical: és a dir, els diferents grups de notes que són percebuts per l'oïda com un conjunt (els acords), així com les diferents relacions o enllaços que es formen entre ells quan els posem formant una successió.

Degut a la naturalesa estructural d'aquesta, canvis relativament subtils en l'harmonia poden tenir un impacte molt elevat en les sensacions que tindrà l'oient en escoltar una peça, així que no és d'estranyar que l'estudi de l'harmonia sigui un dels pilars fonamentals dels estudis de música a les escoles i conservatoris.

En aquests estudis, l'exercici d'harmonia per excel·lència consisteix a composar una seqüència d'acords a quatre veus que inclogui certs "ingredients" que es plantejaran com a requisits a l'enunciat, tot complint amb un seguit de normes teòriques conegudes, orientades a evitar la feblesa estructural de la música.

3.2 Objecte

L'objecte del projecte, doncs, seria programar una aplicació per a mòbils que serveixi de suport per a realitzar aquest tipus d'exercicis (un "editor d'esquemes musicals"), que permeti en primera instància entrar els diferents acords a utilitzar, i posteriorment fer-ne el desglossament per veus, detectant les diferents faltes o errors comesos en el procés (ja sigui a temps real o bé puntuant l'exercici al final a mode de correcció).

3.3 Abast

Degut que aquest tipus d'exercicis són comuns a tots els cursos però la dificultat s'incrementa gradualment (al llarg dels estudis s'afegeixen constantment "ingredients" nous amb les seves normes corresponents), l'aplicació es podria anar ampliant des d'una versió més bàsica fins a una versió que englobi els exercicis de dificultat més elevada.

Dins aquest projecte es plantejaran els requeriments que caldria tenir en compte a cada iteració de l'aplicació fins a la seva versió més complexa, però només es realitzarà el disseny i desenvolupament de la versió corresponent a les funcionalitats bàsiques, deixant per projectes futurs la incorporació de les funcionalitats necessàries per a permetre exercicis més complicats.

D'altra banda, degut que el conjunt de normes d'harmonia utilitzades varia significativament entre centres i professors, s'ha escollit per a aquesta primera versió limitar l'aplicació a la normativa ensenyada pel mestre Miquel Suñer al Conservatori de Música de Girona. En projectes futurs, si s'escau, es comprovaran les possibles diferències que hi pugui haver entre aquesta i l'aplicada per altres professors i conservatoris, i s'ajustarà l'aplicació en consonància.

4 DESCRIPCIÓ DEL PROBLEMA

Entenem per harmonia la disciplina que té per objecte l'estudi dels diversos acords i combinacions accidentals, la relació entre ells, i l'ús que se n'ha de fer en la composició musical.[1]

Per tal de comprendre les diferents normes i conceptes teòrics d'harmonia a què haurà de respondre el programa a l'hora de rebre, mostrar i corregir els esquemes, haurem de veure primer alguns dels conceptes bàsics de la teoria implicada en el llenguatge musical i la seva utilització.

4.1 Conceptes previs

4.1.1 Introducció

La música clàssica occidental consta de 12 sons bàsics, intrínsecament diferents entre ells, que es *repeteixen* en forma de sons *diferents però funcionalment equivalents* a diferents alçades, omplint d'aquesta manera tot l'espectre auditiu. Per tal de trobar un so equivalent a un so donat podem duplicar-ne la freqüència (per trobar el següent so equivalent per sobre), o dividir-ne la freqüència per dos (per trobar el següent so equivalent per sota).

Així doncs, si una freqüència i el doble d'aquesta són percebuts com un mateix so bàsic, els altres onze sons bàsics provindran de la divisió en 12 parts iguals (segons l'escala logarítmica) de l'espai entre una freqüència i el doble d'aquesta[2].

D'aquesta manera, si tenim un so bàsic i volem trobar el següent, haurem de multiplicar la freqüència del primer per $\sqrt[12]{2}$.

4.1.1.1 Notes

A cadascun d'aquests salts entre un so bàsic i el següent se l'anomena un *"semitò"*, o *"mig to"*, i a la successió ordenada dels 12 *semitons* se l'anomena *"escala cromàtica"*.

D'altra banda, degut a la manera com utilitzem aquests sons[3], només 7 dels 12 disposen de nom propi, i la resta s'anomenen aplicant diferents *"alteracions"* sobre aquests 7 sons principals[4].

Aquests 7 sons, o *"notes"* sense alteració representen les posicions 1, 3, 5, 6, 8, 10 i 12 de l'escala cromàtica, i porten els noms[5] (en aquest ordre) *Do, Re, Mi, Fa, Sol, La* i *Si*.

Com podem veure a la Taula 1, en la major part dels casos el salt entre una nota i la següent no és d'un semitò, sinó de dos semitons. Aquesta distància és el que anomenem un *"to"*.

1	2	3	4	5	6	7	8	9	10	11	12
Do		Re		Mi	Fa		Sol		La		Si

Taula 1: Posició de les set notes sense alteració dins l'escala cromàtica (color segons tecles del piano).

[1] NICOLÁS, RIMSKY-KORSAKOV. "Tratado práctico de armonía." *Editorial Ricordi Americana. Buenos Aires*. 1993.
[2] Divisió exacta només a partir de l'adopció del temperament igual.
[3] Veure l'apartat "Tonalitats i Modes"(pàg. 12)
[4] Les 7 notes amb nom propi són les que trobem representades amb tecles blanques al piano (Taula 1).
[5] Als països anglosaxons s'utilitzen, en lloc d'aquests noms, les primeres set lletres de l'abecedari, començant per la lletra A (la) i acabant a la G (sol).

4.1.1.2 Alteracions

D'altra banda, les alteracions que pot portar una nota són *"sostingut"* (♯), que equival a pujar mig to el so de la nota de referència, i *"bemoll"* (♭), que equival a baixar-lo mig to. D'aquesta manera, si volem referir-nos al segon so de l'escala cromàtica direm *"Do sostingut"* (Do♯) o bé *"Re bemoll"* (Re♭), depenent del context[6].

Podem veure alguns dels noms que poden rebre els 12 sons de l'escala cromàtica a la Taula 2.

Posició	1	2	3	4	5	6	7	8	9	10	11	12
Nom	Do	Do ♯	Re	Re ♯	Mi	Fa	Fa ♯	Sol	Sol ♯	La	La ♯	Si
	Si ♯	Re ♭	-	Mi ♭	Fa ♭	Mi ♯	Sol ♭	-	La ♭	-	Si ♭	Do ♭

Taula 2: Noms alternatius dels 12 semitons admetent alteracions de mig to.

Aquestes dues alteracions, a més, podran acumular-se, permetent per exemple la referència al so que es troba *un to* per sobre o un to per sota. Els símbols per als casos amb distància d'un to són 𝄪 (doble sostingut) i 𝄫 (doble bemoll)[7].

En cas que vulguem dir explícitament que una nota no està alterada, direm que és *"natural"* (♮), i si volem desfer l'alteració per defecte per posar-ne una de contrària, concatenarem els símbols que fan referència als dos procediments (p. ex. Si tinc per defecte ♭ i vull indicar que és ♯, escriuré ♮♯).

4.1.1.3 Afinació estàndard i Índex Acústic

Per tal de fixar una correspondència estandarditzada entre les diferents notes i l'espectre de freqüències, s'ha agafat com a referència internacional[8] el La amb freqüència de 440Hz (o, simplement, "*La 440*"), i la resta de notes es generen multiplicant o dividint aquesta freqüència per la proporció esmentada anteriorment (veure Taula 3).

$Nom_{i.a.}$	Do_3	$Do♯_3$	Re_3	$Re♯_3$	Mi_3	Fa_3
fr. (Hz)	261.63	277.18	293.66	311.13	329.63	349.23
$Fa♯_3$	Sol_3	$Sol♯_3$	La_3	$La♯_3$	Si_3	Do_4
369.99	392.00	415.30	440.00	466.16	493.88	523.25

Taula 3: Freqüències de les notes a 440 des de Do_3 fins a Do_4 (índex acústic franco-belga).

D'altra banda, per tal de distingir les diferents aparicions d'una mateixa nota al llarg de l'espectre auditiu disposem d'un número indicador de l'alçada a la qual ens trobem, anomenat *"índex acústic"*. Aquest número s'incrementa cada vegada que apareix un Do, i es manté al llarg de tota l'escala ascendent fins arribar al Do següent.

[6] Malgrat sonar igual, tindran sentits estructurals diferents depenent de la tonalitat de què s'estigui parlant, de manera que en general l'expressió correcta per a cada cas serà només una. Ho anirem veient més endavant.

[7] Per tal d'indicar alteracions de més d'un to no existeix un símbol específic, sinó que es combinarien els símbols esmentats fins aconseguir l'efecte desitjat (p. ex. 𝄪♯ equivaldria a tres semitons). Tot i això, en les èpoques que tractarem no és habitual utilitzar alteracions d'aquesta envergadura.

[8] Establert com a estàndard el 1939. Anteriorment la referència era més baixa, per la qual cosa actualment es tendeix a tocar en l'afinació original de l'època la música més antiga.

Existeixen principalment dues versions d'índex acústic segons quina freqüència considerem que ha de tenir el Do_0. A Espanya habitualment s'utilitza el sistema franco-belga, que agafa com a Do_3 el Do central del piano[9] (261.63 Hz), de manera que el La següent (La 440) serà el La_3.

4.1.2 Intervals

De la distància entre dos sons en direm un *"interval"*. Per tal de classificar els diferents *intervals* segons la seva naturalesa, ens haurem de fixar en dues dades: la distància en semitons existent entre els dos sons i la quantitat de notes amb nom que els separen[10].

D'aquesta manera, un interval vindrà determinat per un nom (provinent de la quantitat de notes entre els dos sons) i un qualificatiu (provinent de la distància en semitons), d'una manera similar a com un so ve determinat per un nom de nota i una alteració.

En el cas dels intervals, el nom serà en la majoria dels casos un ordinal que prendrà la forma regular pels intervals més petits (p. ex. *"segona", "tercera", "quarta"*) i la forma llatinitzada per als intervals més amples (p. ex. *"quinta", "sexta", "sèptima", "octava", ...*), i que es calcularà comptant les notes existents entre la nota inicial i la final, ambdues incloses (p. ex. Do i Re formen una *segona*, Do i Sol formen una *quinta*).

En el cas que es tracti d'una mateixa nota que apareix repetida en diferents veus (p. ex. Do_4 i Do_4), parlarem d'un *uníson*, i si és la mateixa nota però a dues alçades diferents (p. ex. Do_3 i Do_4), parlarem d'una *octava*.

D'altra banda, el qualificatiu de l'interval es dedueix de comparar la distància en semitons que hi ha entre els dos sons amb unes distàncies de referència preestablertes[11] per a cada nom d'interval.

Aquestes distàncies naturals dels diferents intervals, així com els qualificatius que reben, els podem veure a la Taula 4.

Com podem comprovar, els intervals que tenen dues possibles distàncies naturals reben els qualificatius *"Major"* o *"menor"* segons si responen a la distància llarga o a la curta, respectivament, mentre que els intervals que només tenen una distància natural, en utilitzar-la reben el qualificatiu de *"Justa"*.

En cas que un interval tingués mig to més que la seva distància natural més llarga, rebria el qualificatiu "augmentat" (A), mentre que un interval mig to per sota de la distància natural més curta rep el nom de "disminuït" (d) (p. ex. una tercera de 2 semitons és una *"tercera disminuïda"* (3d), mentre que una quinta de 8 semitons és una *"quinta augmentada"* (5A)).

[9] L'altra versió, l'índex acústic científic (SPN), agafa el Do central del piano com a Do_4.
[10] La quantitat de notes amb nom que separen els dos sons d'un interval és important perquè, de la mateixa manera que amb les alteracions, cadascuna de les maneres possibles d'anomenar un element porta implícita una interpretació diferent de la funció de l'element dins el conjunt, així que la utilització d'una nomenclatura coherent amb el context ens facilita la comprensió de la funció que fan els diferents elements dins la música.
[11] Provinents dels intervals que es generen entre la nota principal d'una tonalitat i les altres notes que hi pertanyen. Veure l'apartat "Tonalitats i Modes" (pàg. 12).

D'altra banda, qualsevol interval més gran d'una octava (o *"interval compost"*) farà una funció equivalent al que resultaria d'acostar les notes canviant-ne l'índex acústic (p. ex. Do_3 i Re_4 a efectes pràctics formen una segona Major). Tot i això, en segons quin cas ens pot interessar referir-nos explícitament a l'interval compost, que rebrà el mateix qualificatiu que el seu interval *simple* equivalent (p. ex. els mateixos Do_3 i Re_4 tècnicament formarien, en realitat, una novena Major).

Nom de l'interval	Qualificatiu	Abreviatura	Semitons
Segona	menor[12]	2m	1
	Major	2M	2
Tercera	menor	3m	3
	Major	3M	4
Quarta	Justa	4J	5
Quinta	Justa	5J	7
Sexta	menor	6m	8
	Major	6M	9
Sèptima	menor	7m	10
	Major	7M	11

Taula 4: Qualificatius principals dels intervals simples segons la distància entre els dos sons.

4.1.2.1 Classificació dels intervals segons el moviment

Degut que les aparicions dels intervals en la música no sempre són en forma de dos sons simultanis, podem fer una segona classificació d'aquests en funció del moviment que generen les dues notes que el formen.

En el cas anteriorment esmentat en què tenim dos sons simultanis, direm que l'interval és *"harmònic"*. En canvi, si els dos sons apareixen un després de l'altre, direm que l'interval és *"melòdic"*, i podrem distingir entre *"melòdic ascendent"* (quan la segona nota és més aguda[13] que la primera) i *"melòdic descendent"* (quan la segona nota és més greu).

D'altra banda, el cas concret de l'interval melòdic de segona sovint s'anomena també pel nom de *"graus conjunts"*.

4.1.2.2 Càlcul i propietats dels qualificatius dels intervals

Per tal de reduir la quantitat d'informació necessària per determinar els qualificatius dels intervals i facilitar-ne el càlcul, podem fixar-nos en les propietats següents:

- Si invertim[14] un interval menor, obtindrem un interval Major (i viceversa).
- Si invertim un interval just, obtindrem un interval just.
- Si invertim un interval Augmentat, obtindrem un interval disminuït (i viceversa).
- Totes les quintes sense alteracions són justes excepte Si-Fa, que és disminuïda.

[12] Habitualment s'utilitzen en música de manera deliberada Majúscula i minúscula fora les normes ortogràfiques per tal d'emfatitzar la distinció entre Major i menor, així com entre Augmentat i disminuït.
[13] Diem que una nota és més aguda quan la seva freqüència és més elevada, i més greu quan és més baixa.
[14] Un interval serà la inversió d'un altre si les notes que el formen són les mateixes, però la nota més greu d'un és la més aguda de l'altre (p. ex. podem invertir un interval simple pujant una octava la nota més greu).

Així doncs, només haurem de saber les distàncies en semitons dels intervals de segona i tercera (només una de les dues distàncies naturals, ja que l'altra ve separada per mig to), i podrem deduir la resta a partir de les propietats esmentades.

Podem veure un resum de les inversions d'aquests intervals que és útil conèixer a la Taula 5.

Interval conegut	2m	2M	3m	3M	5J	5d
Inversió	7M	7m	6M	6m	4J	4A

Taula 5: Resum d'inversions deduïdes dels intervals coneguts.

Si ens fixem en l'últim cas de la Taula 5 i deduïm a partir de la Taula 4 els semitons que separen les notes en cada cas, veurem que tant a l'interval de 5d com al de 4A els sons estan separats per 6 semitons (la inversió coincideix amb una de les maneres alternatives de descriure el mateix).

Aquesta distància de 6 semitons, degut a la dissonància[15] que crea, ha desenvolupat una rellevància especial en la música, i habitualment se l'anomena simplement *"tritò"* (tres tons).

4.1.3 Tonalitats i Modes

Degut a la manera com interactuen les freqüències, alguns dels intervals que podem generar entre els 12 sons bàsics seran més agradables a l'oïda (més *"consonants"*), mentre que uns altres seran més aviat desagradables (o *"dissonants"*) (p. ex. la proporció 3:2 entre les dues freqüències d'una quinta justa (Figura 1) resulta en un interval consonant).

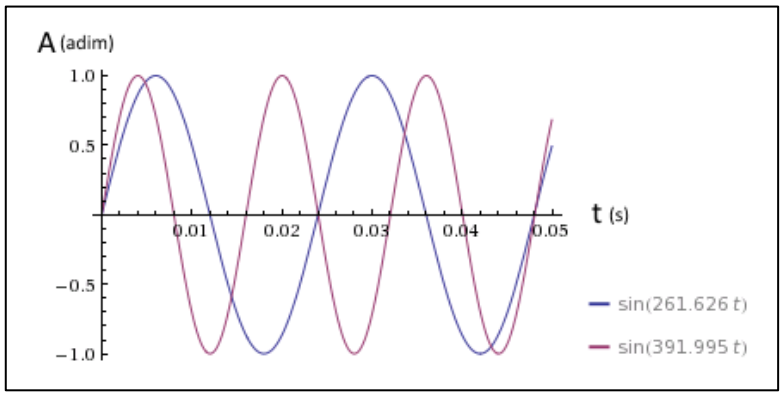

Figura 1: Representació d'una quinta justa a partir de les freqüències de Do_3 i Sol_3 (Wolfram|Alpha).

Per aquest motiu, si escollim un so de referència al voltant del qual generar la música (una *"tònica"*), el diferent grau de *consonància* d'aquest so central amb cadascun dels altres onze sons[16] determinarà la funció que fa cadascun d'ells dins el conjunt (i també quins d'ells ens interessarà excloure[17]).

[15] Combinació de sons que resulten en una sensació poc agradable per l'oient.
[16] Així com la relació entre els acords que es generen a partir de les dues notes implicades (veure 4.1.4 Acords).
[17] Dins el subconjunt generat per una tònica no hi haurà tots els sons possibles. Tot i això, dins una música basada en una tònica apareixeran també les notes que no pertanyen al subconjunt (en forma de préstecs agafats d'altres subconjunts que hi estiguin relacionats).

Aquesta organització jeràrquica dels sons al voltant d'un so central, en la forma en què es desenvolupa en la música clàssica occidental[18], és el que anomenem *"tonalitat"*.

D'altra banda, per extensió anomenarem també *"tonalitat"* a cadascun dels possibles subconjunts de sons i relacions que es generen a partir d'aquest sistema (p. ex. anomenarem *tonalitat de Do* al conjunt de sons i relacions generats amb el *Do* com a *tònica*).

4.1.3.1 Graus i funcions tonals

Dins de cada tonalitat distingirem 7 o 8 funcions estructurals bàsiques[19], anomenades *"funcions tonals",* que vindran representades pels 7 noms de nota esmentats anteriorment[20].

Tot i això, degut que les tonalitats es generen al voltant d'una tònica, les correspondències entre un nom i una *funció tonal* no seran sempre les mateixes, sinó que dependran de la tònica escollida (p. ex. en la *tonalitat de Do*, Do és la *tònica*, però en la *tonalitat de Sol* no ho és).

Així doncs, per tal de referir-nos a una funció tonal independentment de la tonalitat escollida farem servir el seu nom propi, o bé indicarem la seva posició dins la seqüència (o *"grau"*), en números romans (Taula 6).

Grau	Funció tonal	Nota corresponent (Do M)
I	Tònica	Do
II	Supertònica	Re
III	Mediant	Mi
IV	Subdominant	Fa
V	Dominant	Sol
VI	Superdominant	La
VII	Sensible / (Subtònica)	Si / (Si♭)

Taula 6: Funcions tonals de cada grau i nota que el representa en Do Major.

Podem veure que el setè grau pot fer dues funcions diferents. Això és degut que les dues distàncies naturals que pot prendre respecte a la tònica generen relacions molt diferents amb aquesta, de manera que cada distància equivaldrà a una funció (i. e. serà sensible si es troba a mig to de la tònica, i subtònica si es troba a un to).

D'altra banda, com que volem assignar un nom de nota diferent a cada grau, però hem de respectar les distàncies que creen intervals consonants (i. e. representar correctament els sons que formen part de la tonalitat), sovint haurem d'ajustar el nom de nota (posant alguna alteració) per tal que representi el so desitjat.

Així doncs, per a cada tonalitat cadascuna de les notes tindrà una alteració per defecte[21]. El conjunt d'alteracions que porta per defecte una tonalitat és el que anomenem *"armadura"*.

[18] Predominant des del segle XVII fins a l'abandonament de la tonalitat per part d'alguns autors al segle XX.
[19] A efectes pràctics, la subtònica pràcticament no s'utilitza en els tons majors, que tenen la sensible per defecte (veure "Escales i modes", pàg. 14).
[20] Veure l'apartat "Notes" (pàg. 8).
[21] En la majoria dels casos aquesta alteració serà "natural" (i. e. directament el so que representa el seu nom), però cada tonalitat tindrà un conjunt de notes que sí que necessitaran un canvi respecte al so per defecte.

A la Taula 7 podem veure alguns exemples de com es veurien afectats els noms de nota de la Taula 6 si canviéssim la tònica mantenint les distàncies entre els sons[22].

Tonalitat	I	II	III	IV	V	VI	VII
Do M	Do	Re	Mi	Fa	Sol	La	Si
La M	La	Si	Do♯	Re	Mi	Fa♯	Sol♯
Si ♭ M	Si♭	Do	Re	Mi♭	Fa	Sol	La

Taula 7: Exemples de les alteracions que s'obtenen en construir tonalitats.

Les distàncies que aquí hem traslladat, provinents d'una escala[23] sense alteracions (Do M) són representatives de la gran majoria de la música occidental, ja sigui en l'ordre en què apareixen a la taula o en les diferents configuracions que obtindríem si féssim el mateix trasllat però no ajustéssim les alteracions.

4.1.3.2 Escales i modes

Si ordenem les notes de qualsevol de les tonalitats esmentades a la Taula 7, i n'analitzem les distàncies, veurem que 5 dels intervals formen una distància d'un to, mentre que els altres 2 estan separats per mig to (Taula 8). A més, si ens fixem en la posició d'aquests intervals de mig to, veiem que la separació entre ells és la màxima que podem aconseguir amb les condicions anteriors, ja que les set notes tornaran a començar la seqüència tan bon punt arribem a la següent octava (i. e. no podem acostar el primer interval de mig to al principi, perquè l'estaríem acostant també al final).

Do	-	Re	-	Mi	Fa	-	Sol	-	La	-	Si	Do
1		1		½	1		1		1		½	...

Taula 8: Distàncies entre les notes de l'escala diatònica (exemple en Do M).

La seqüència que compleix aquestes condicions de distàncies i intervals (independentment de la nota i posició per la qual comenci) s'anomena *"escala diatònica"*, i prové de la realització de set intervals de quinta justa consecutius (p. ex. partint de Fa obtenim Fa-Do-Sol-Re-La-Mi-Si, que són el mateix subconjunt de sons que veiem a la Taula 8).

Les notes que es generen segons aquest procediment, però, no tenen cap ordre per defecte, de manera que podem obtenir 7 configuracions diferents depenent de quina nota considerem que és la primera. Cadascuna d'aquestes configuracions és el que anomenem un *"mode"*, i cadascun d'aquests *modes* rep un nom propi provinent de la cultura grega (Taula 9).

Num	Distàncies							Nom
1	1	1	½	1	1	1	½	Jònic
2	1	½	1	1	1	½	1	Dòric
3	½	1	1	1	½	1	1	Frigi
4	1	1	1	½	1	1	½	Lidi
5	1	1	½	1	1	½	1	Mixolidi
6	1	½	1	1	½	1	1	Eòlic
7	½	1	1	½	1	1	1	Locri

Taula 9: Distàncies entre les notes per als set modes gregorians.

[22] A aquesta pràctica se l'anomena "transportar".
[23] Successió ordenada de les notes d'un sistema.

D'aquestes 7 configuracions possibles, dues (els modes Jònic i Eòlic) han donat peu a les configuracions que s'han fet servir al llarg de tota la música clàssica occidental, i les altres 5 es van deixar de fer servir al voltant del segle XVII.

El Jònic (com podem comprovar comparant-lo amb la Taula 8) és el que dona peu a l'actual mode Major, mentre que l'Eòlic és el que dóna peu a l'actual mode menor.

Tot i això, a l'hora d'utilitzar l'escala menor s'introdueixen una sèrie de variants respecte l'escala del mode Eòlic (en endavant *"escala menor natural"*), que tenen en compte l'alteració de la setena nota per tal d'utilitzar-la amb la funció de sensible (Taula 10).

Per tal de facilitar-ne la comprensió s'ha utilitzat l'escala de *la menor*, que en la seva versió per defecte (*la menor natural*) no porta cap alteració.

m. natural	La	Si	Do	Re	Mi	Fa	Sol	La
m. harmònica	La	Si	Do	Re	Mi	Fa	Sol♯	La
m. melòdica	La	Si	Do	Re	Mi	Fa♯/(Fa)	Sol♯/(Sol)	La

Taula 10: Variants principals de l'escala menor.

Podem veure a la taula que l'escala *menor harmònica* puja mig to el setè grau per tal de convertir-lo en *sensible* (abans, degut que entre *Sol* i *La* hi ha un to sencer, la sèptima feia de subtònica).

Aquesta escala, però, genera un interval de segona augmentada (2A) entre la sexta i la sèptima (i. e. entre Fa i Sol♯), que donaria lloc a dissonàncies. Per arreglar-ho, l'escala melòdica introdueix una alteració equivalent també a la sexta quan es tracta d'un interval *ascendent*, mentre que fa servir la subtònica quan ens movem de manera *descendent* (p. ex. pujant faria Fa♯-Sol♯-La, però baixant faria La-Sol-Fa).

Pel que fa a l'escala Major, existeixen també certes variants que contemplen la utilització d'alteracions provinents de les diferents variants menors[24], però no les tindrem en compte degut que no són necessàries per al temari que tractarem.

4.1.3.3 Càlcul i propietats de les armadures

Degut que l'observació i repetició de les distàncies entre els graus és un mètode molt poc eficient per a calcular quines alteracions per defecte haurà de portar una tonalitat, a la pràctica s'utilitzen unes normes molt senzilles que ens permeten saber de seguida amb quin subconjunt de notes estem treballant.

Aquestes normes o tècniques es basen en la successió que es genera en fer intervals consecutius de quinta (de la mateixa manera que quan obteníem les notes de l'escala diatònica). Aquesta successió l'utilitzarem de manera ascendent o descendent segons estiguem parlant de sostinguts o bemolls. Podem veure cadascuna d'aquestes opcions a la Taula 11.

Sostinguts	Fa	Do	Sol	Re	La	Mi	Si
Bemolls	Si	Mi	La	Re	Sol	Do	Fa

Taula 11: Ordre dels sostinguts i dels bemolls segons el cercle de quintes.

[24] Escales Major mixta principal i Major mixta secundària.

Com podem veure, cadascuna d'aquestes seqüències és la inversa de l'altra.

Les notes que inicien cadascuna de les dues seqüències a la taula s'han escollit deliberadament per tal que la seqüència representi també la *prioritat* amb què alterarem cadascuna de les notes (p. ex. quan només tinguem un sostingut, serà sempre Fa, si en tenim dos seran sempre Fa i Do, etc).

D'altra banda, aquestes tècniques de càlcul ens serviran només per calcular les tonalitats Majors, però les armadures de les tonalitats menors les podrem extrapolar en cada cas de la tonalitat Major que comparteix armadura amb ella (del seu *"relatiu Major"*).

Així doncs, la manera de calcular les armadures d'una tonalitat Major a partir de les seqüències anteriors serà la següent[25]:

- Hi haurà dues tonalitats conegudes que no calcularem: Do M (que no porta res), i Fa M (que portarà el Si♭).
- Amb la resta de tonalitats, quan no portin bemoll al nom omplirem sostinguts fins arribar a un nom de nota per sota el nom de la tonalitat (p. ex. Si volem calcular l'armadura de La M afegirem sostinguts seguint la seqüència de la *Taula 11* fins arribar al Sol, de manera que tindrem Fa♯, Do♯, i finalment Sol♯).
- D'altra banda, quan les tonalitats sí que tinguin bemoll al nom, omplirem bemolls fins un després d'arribar al nom de la tonalitat (p. ex. Si volem calcular l'armadura de Mi♭ M, afegirem bemolls seguint la seqüència de la *Taula 11* fins arribar al Mi, i després n'afegirem un més, obtenint Si♭, Mi♭, i finalment La♭).

D'altra banda, per trobar la tonalitat Major que compartirà armadura amb una tonalitat menor, només haurem d'aplicar a la tonalitat menor una tercera menor ascendent (p. ex. Si volem saber l'armadura de Do m primer farem una 3m ascendent (Do-Re-Mi♭), i tot seguit utilitzarem la tècnica anterior sobre la nota trobada: Mi♭ M té tres bemolls, així que Do m tindrà el mateix).

Podem veure alguns exemples de tonalitats Majors habituals junt amb el seu relatiu menor i l'armadura que comparteixen a la Taula 12.

La M	fa ♯ m	3♯	Fa♯, Do♯, Sol♯
Re M	si m	2 ♯	Fa♯, Do♯
Sol M	mi m	1 ♯	Fa♯
Do M	la m	♮	-
Fa M	re m	1 ♭	Si♭
Si ♭ M	sol m	2 ♭	Si♭, Mi♭
Mi ♭ M	do m	3 ♭	Si♭, Mi♭, La♭

Taula 12: Exemples de tonalitats habituals amb la seva armadura corresponent.

D'altra banda, si ens fixem en la taula podem apreciar que cada vegada que afegim una quinta a la tonalitat estem afegint també un sostingut (o traient un dels bemolls) (p. ex. El salt de Sol M a Re M

[25] Això ens permet trobar les armadures més habituals. Hi haurà casos fora aquest rang que podrem calcular a partir de la seva versió dins el rang (p. ex. La norma no ens serviria per tonalitats com Sol♯ M, però podem afegir un sostingut a totes les notes en l'armadura de Sol M, de manera que traslladem totes les notes mig to enlaire i passem de tenir només Fa♯ a tenir Fa𝄪 i la resta de notes ♯).

és un interval de quinta, i passa d'un sostingut a dos sostinguts). Així doncs, ordenar les tonalitats segons la quantitat de bemolls i sostinguts que tenen resulta en la mateixa seqüència de quintes que hem utilitzat anteriorment per deduir-les.

Aquesta seqüència, degut que arriba un moment en què retorna al punt inicial, és el que anomenem habitualment "el cercle de quintes", i sovint es representa, seguint amb l'analogia, d'una manera circular (Figura 2).

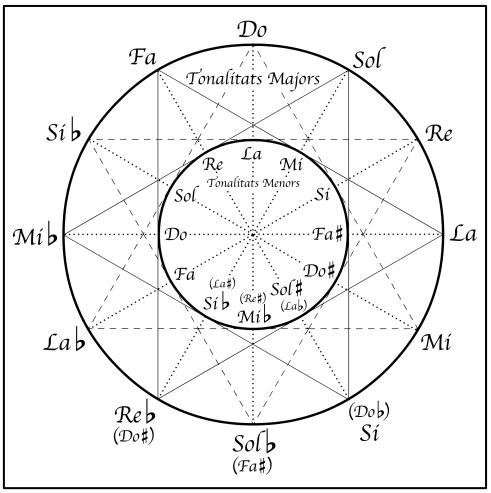

Figura 2: Representació gràfica del cercle de quintes.

4.1.4 Acords

Direm acord a la superposició de dos, tres, o quatre sons[26], separats entre sí per intervals de tercera[27] (p. ex. Mi, Do i Sol es poden ordenar superposant terceres seguint l'ordre ascendent Do-Mi-Sol, de manera que formen un acord).

Per tal de caracteritzar els acords, principalment haurem de mirar dues dades: la nota per la qual comença la successió de terceres (anomenada *"fonamental"*), i el qualificatiu de les diferents terceres que es generen a partir d'aquesta (i. e. les distàncies entre els diferents sons).

D'altra banda, haurem de veure també la quantitat de notes que formen part de l'acord, i si la nota més greu és la *fonamental* o en canvi la nota més greu és una de les altres (i. e. si l'acord està "*invertit"*).

Per tal de referir-nos a les altres notes ho farem a partir del nom de l'interval (sense el qualificatiu), de manera que la primera nota a partir de la fonamental serà *"la tercera de l'acord"*, i la nota següent serà *"la quinta de l'acord"*. De la mateixa manera, si l'acord està format per quatre sons, la última nota de la successió serà *"la sèptima"*, i si són cinc, tindrem a més *"la novena"*.

D'altra banda, quan un acord consta de tres notes, direm que és un acord *"tríada"* o *"perfecte"*.

[26]Habitualment es parla també d'acord encara que la superposició sigui de cinc sons (si compleix la condició dels intervals de tercera).
[27]NICOLÁS, RIMSKY-KORSAKOV. "Tratado práctico de armonía." *Editorial Ricordi Americana. Buenos Aires*. 1993.

Si variem el qualificatiu de les terceres que formen un acord *tríada* (generem totes les combinacions de 3M i 3m possibles) obtindrem quatre acords bàsics (Taula 13), dels quals la major part del temps només farem servir dos (el *Perfecte Major* i el *Perfecte menor*).

Nom de l'acord	Perfecte Augmentat			Perfecte Major			Perfecte menor			Perfecte disminuït		
Quinta	Sol♯	3M	5A	Sol	3m	5J	Sol	3M	5J	Sol♭	3m	5d
Tercera	Mi	3M		Mi	3M		Mi♭	3m		Mi♭	3m	
Fonamental	Do	-		Do	-		Do	-		Do	-	

Taula 13: Intervals que formen els diferents acords perfectes (exemple en Do M).

Podem veure que l'acord Perfecte Major respon a les distàncies naturals que trobaríem entre els graus 1, 3 i 5 de l'escala Major, i que de la mateixa manera, l'acord Perfecte menor tindrà les mateixes distàncies que trobaríem entre els mateixos graus en l'escala menor (en qualsevol de les tres variants).

Dins una tonalitat, però, tindrem set graus diferents, de manera que podrem generar set acords diferents segons quina nota escollim com a fonamental. Podem veure els acords que es generarien en una tonalitat Major a la Taula 14.

Grau	I	II	III	IV	V	VI	VII
Qualificatiu	Major	menor	menor	Major	Major	menor	disminuït
Quinta	Sol	La	Si	Do	Re	Mi	Fa
Tercera	Mi	Fa	Sol	La	Si	Do	Re
Fonamental	Do	Re	Mi	Fa	Sol	La	Si

Taula 14: Qualificatius dels acords generats a partir dels graus d'una tonalitat Major (exemple en Do M).

En el cas de les tonalitats menors, degut a les alteracions opcionals que ens aporten les variants de l'escala, el conjunt d'acords possibles que es generarien seria més ampli, però en tot cas la idea bàsica serà sempre la mateixa.

D'altra banda, sobre aquests acords podrem afegir una quarta nota a dalt per tal de construir els acords *"de sèptima"* corresponents, o dues notes per construir els acords *"de novena"*. La incorporació d'aquestes notes, però, no afectarà al qualificatiu de l'acord, sinó que s'incorporarà a la descripció de l'acord la nomenclatura corresponent a la nota afegida (p. ex. si Do-Mi-Sol és un acord de *Do Major*, llavors Do-Mi-Sol-Si serà un acord de *Do Major* amb *Sèptima Major*[28]).

4.1.4.1 Inversions dels acords

Per tal de comprendre millor els acords hem escollit ordenar-ne els sons per intervals de tercera ascendent a partir d'una fonamental. Dins la música, però, poques vegades els trobarem en aquest ordre, i si bé l'ordre de les notes més agudes té un efecte petit en la sonoritat que tindrà l'acord, la diferència que provoca canviar la nota més greu és considerable.

Degut a això, classificarem també els acords en funció de quina de les notes es trobi en la posició més greu. De la nota en aquesta posició (independentment de quina sigui) en direm el *"baix"*, i de cadascuna de les possibilitats que sorgeixen d'aquesta classificació en direm una *"inversió"*.

[28] Habitualment obviarem el qualificatiu de la Sèptima a l'hora d'anomenar l'acord, ja que aquest acostuma a venir determinat pel context (p. ex. en el cas anterior sovint direm només "Do Major amb Sèptima", i serà Major o menor segons ho provoquin les alteracions de la tonalitat en què estiguem treballant).

Per tal de referir-nos a les diferents inversions, disposarem d'un nom, i d'un *"xifrat"*. El nom, quan la *fonamental* no sigui al *baix*, serà un ordinal representatiu de les vegades que hem pujat el baix una tercera (p. ex. un acord amb la *tercera* al baix estarà en "*primera inversió"*, mentre que un acord amb la *quinta* al baix estarà en "*segona inversió"*. El cas per defecte en què la fonamental sí que es troba al baix l'anomenarem simplement *"estat fonamental"*.

D'altra banda, el *xifrat* serà la manera en què abreviarem el nom de la inversió de cara a escriure'l juntament amb el *grau* de l'acord, i consistirà en un conjunt de números representatius dels nous intervals que es generen entre el baix i les altres notes dins la inversió en qüestió (p. ex. en primera inversió entre el baix i la fonamental hi haurà una sexta, així que la primera inversió es *xifra* "6").

Podem veure les tres possibles inversions d'un acord tríada, amb el seu nom i xifrat, a la Taula 15.

Nom de la inversió	Estat Fonamental			Primera Inversió			Segona Inversió		
Grau i Xifrat	I (o també I_5)			I_6			I_4^6		
Ordre de les notes	Sol	-	5	Do	-	6	Mi	-	6
	Mi	3		Sol	3		Do	4	
	Do	-		Mi	-		Sol	-	

Taula 15: Inversions de l'acord tríada i intervals que es formen (exemple en Do Major).

Pel que fa als acords amb sèptima, les inversions funcionaran de la mateixa manera, però n'hi haurà una més, i canviarà una mica el xifrat (els números passen a representar els intervals generats fins a la fonamental i fins a la sèptima). Ho podem veure a la Taula 16.

Nom de la inversió	Estat Fonamental				Primera Inversió				Segona Inversió				Tercera Inversió			
Grau i Xifrat	I_7				I_5^6				I_3^4				I_2			
Ordre de les notes	Si	-		7	Do	-		6	Mi	-		6	Sol	-		6
	Sol	-	5		Si	-	5		Do	-	4		Mi	-	4	
	Mi	3			Sol	3			Si	3			Do	2		
	Do	-			Mi	-			Sol	-			Si	-		

Taula 16: Inversions de l'acord amb sèptima i intervals que es formen (exemple en Do Major).

D'altra banda, de la mateixa manera que a les taules s'ha fet servir com a exemple l'acord de *Do Major* dins la tonalitat de *Do Major* (i. e. el *primer grau* de Do Major), podríem escriure de la mateixa manera el xifrat per qualsevol altre grau (p. ex. dins de Do Major, un acord amb sèptima en segona inversió sobre el cinquè grau (Re-Fa-Sol-Si) s'escriuria V_3^4).

4.1.5 Llenguatge musical

Evidentment, tots els conceptes que hem vist de manera escrita hauran d'acabar sent escrits en una partitura per tal de representar-los d'una manera més compacta que en simplifiqui la comprensió.

El camp del llenguatge musical en la seva totalitat tindria una amplitud considerable, però els exercicis d'harmonia que treballarem només requereixen d'una part molt reduïda de les eines de llenguatge musical de què disposem (p. ex. no tenen en compte res que tingui a veure amb el ritme, els silencis o el volum), així que ens limitarem a veure'n només la part necessària.

4.1.5.1 Pentagrama i alteracions

El pentagrama, junt amb les claus, serà l'eina bàsica que ens permetrà representar les notes d'una manera senzilla i compacta.

Un pentagrama serà bàsicament un conjunt de cinc línies horitzontals equidistants sobre els quals posicionarem les notes en funció de la seva alçada (i. e. de com d'*agudes* siguin). Les notes es representaran mitjançant el·lipses buides, i podran ubicar-se sobre qualsevol de les línies del pentagrama o dins de qualsevol dels espais que les separen (Figura 3).

Figura 3: Notes sobre les línies i els espais d'un pentagrama (Noteflight).

Cada vegada que passem d'una línia a l'espai immediatament superior, o d'un espai a la línia immediatament superior, el nom de nota s'incrementarà una posició (p. ex. si una línia representa un Mi, el següent espai representarà un Fa).

Degut que l'espai dins el pentagrama només ens permetria representar un rang d'alçades molt limitat (conté només 9 posicions possibles), podrem representar també qualsevol nota que es trobi per sobre i per sota d'aquest rang ubicant-la de forma tangent a l'última línia o bé ampliant localment el pentagrama mitjançant la utilització del que anomenem "*línies addicionals*" (Figura 4).

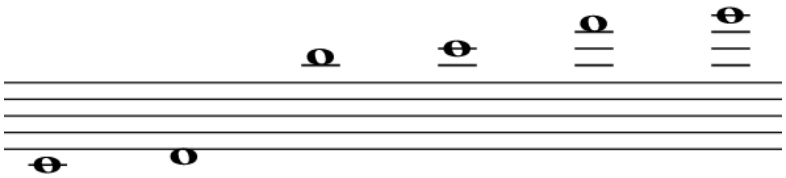

Figura 4: Línies addicionals i notes més enllà del rang delimitat pel pentagrama (Noteflight).

D'altra banda, de la mateixa manera que passava amb els noms de nota, la notació bàsica en aquest cas tampoc és suficient per fer referència a tots els sons de què disposem (i. e. només disposem de posicions al pentagrama per a les set notes que tenen nom), així que haurem de fer servir també aquí les alteracions[29].

En aquest cas, a diferència de com ho fèiem amb la notació escrita, l'alteració l'escriurem abans de la nota que volem alterar (Figura 5).

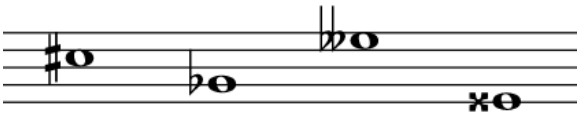

Figura 5: Alteracions aplicades sobre les notes en un pentagrama (Noteflight).

[29] Veure apartat 4.1.1.2 Alteracions (pàg. 9).

D'altra banda, per tal de simplificar la lectura i escriptura de les alteracions que portarà per defecte la tonalitat en què estiguem escrivint, les agruparem totes al principi de cada línia, i les donarem per suposades a partir de llavors (i. e. haurem d'indicar expressament els casos en què volem que la nota sigui natural).

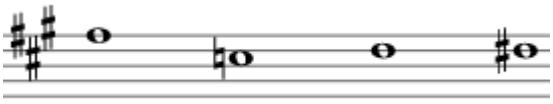

Figura 6: Armadura sobre el pentagrama i efecte en les notes (Noteflight)

Observant la Figura 6 podem veure que s'han posat tres sostinguts d'armadura, de manera que les notes en els espais marcats per aquests (així com els seus equivalents en altres octaves) es veuran afectats per defecte per aquestes alteracions. Així, si l'alteració per defecte ens sembla bé no haurem d'escriure-la (com a la primera nota), mentre que si la volem anul·lar ho haurem d'indicar expressament amb un natural (com a la segona). La resta de notes que no haguem alterat des de l'armadura mantindran el comportament habitual.

4.1.5.2 Claus i Índex Acústic

Per tal de fixar una correspondència estandarditzada entre els diferents espais del pentagrama i les diferents notes (incloent l'índex acústic d'aquestes) disposem de les claus.

Una clau és un símbol ubicat al principi del pentagrama que ens marca la posició dins aquest d'una de les set notes (així com l'índex acústic d'aquesta), de manera que totes les altres notes queden definides a partir d'aquesta primera. Així, sempre podrem triar la clau que més s'ajusti al rang d'alçades en què treballarem.

Les claus més habituals (i les que utilitzarem per als exercicis d'harmonia) són les claus de Sol i Fa. La clau de Sol ubica el Sol_3 a la segona línia del pentagrama començant per baix, mentre que la clau de Fa ubica el Fa_2 a la quarta. Així, la clau de Sol respon al registre[30] típic de les veus femenines, mentre que la clau de Fa respon al de les veus masculines.

A la Figura 7 podem veure la clau de Sol (esquerra), junt amb un Sol_3, així com la clau de Fa (centre) amb un Fa_2. A la dreta de la nota representativa de cada clau podem veure la posició en què quedarà el Do central del piano (Do_3) en cadascun dels casos. Com podem veure, la clau de Sol ens permet representar fàcilment els registres que quedin majoritàriament per sobre del Do central, mentre que la clau de Fa serà més adequada per representar els registres que quedin majoritàriament per sota.

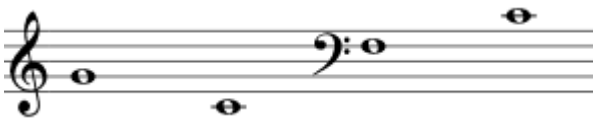

Figura 7: Claus de Sol i Fa junt amb la seva nota representativa i el Do_3

En el cas que haurem de veure dels exercicis d'harmonia, el que representarem serà una composició per a cor mixt a quatre veus, de manera que escriurem les dues veus femenines sobre una clau de

[30] Rang de notes a què pot arribar un instrument (en aquest cas, la veu).

sol, i paral·lelament les dues veus masculines en una clau de fa que ubicarem en un pentagrama diferent a sota (Figura 8).

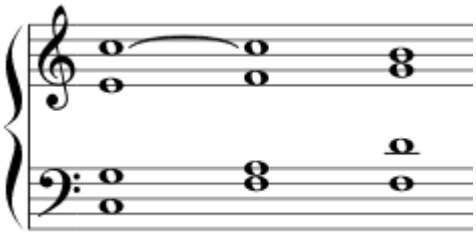

Figura 8: Exemple de la utilització de les claus per representar les quatre veus d'un cor mixt (Do M).

D'altra banda, en els nostres exercicis d'harmonia voldrem representar també el grau i xifrat dels diferents acords que es formen entre les veus. Així doncs, ubicarem aquests a la part inferior, sota la clau de fa (Figura 9).

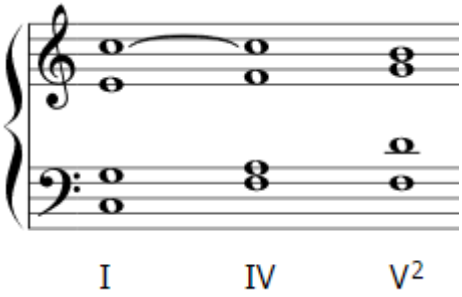

Figura 9: Exemple de la Figura 8 complementat amb el grau i xifrat.

Com podem observar (i tal com comentàvem a l'apartat 4.1.4.1 sobre inversions dels acords) el xifrat en cada cas no es veu afectat per l'ordre de les notes superiors, sinó que depèn únicament de quina de les notes de l'acord es troba al baix (p. ex. l'acord de V a la dreta de tot (de baix a dalt Fa-Re-Sol-Si) porta un 2, malgrat la posició que trobaríem per defecte per aquesta inversió seria Fa-Sol-Si-Re).

D'altra banda, podem veure a les figures anteriors que quan hi hagi dues notes iguals seguides podem marcar aquesta continuïtat amb un arc que les ajunti[31], anomenat *"lligadura"*.

[31] Quan les dues notes iguals seguides siguin al baix habitualment no ho marcarem, però a la resta de veus sí.

4.2 Normes i conceptes teòrics d'harmonia

Un cop hem vist els conceptes previs necessaris per a comprendre els conceptes d'harmonia amb què haurà de treballar l'aplicació, podem passar a veure pròpiament les diferents normes a què haurà de respondre, així com la teoria que les envolta.

Aquestes normes són un seguit de directrius que s'han anat extraient de l'anàlisi de les obres dels grans compositors de música clàssica, de manera que responen al que s'ha anat veient que funcionava al llarg dels anys. Tot i això, si bé en la qüestió acadèmica s'apliquen d'una manera relativament estricta (degut que està bé conèixer-les), a la pràctica estan plenes d'excepcions, i el compositor se les saltarà sovint en pro de la creativitat.

En tot cas, però, el caire acadèmic de l'aplicació ens portarà a voler que s'hi tinguin en compte tal i com estan descrites, i posteriorment ja serà en tot cas l'usuari qui decideixi saltar-se les normes que cregui pertinents en cada cas.

4.2.1 Introducció

Com s'ha suggerit ja a la introducció, l'estructura dels exercicis que es treballaran consisteix en una primera tria dels acords que s'utilitzaran a l'esquema i un desglossament posterior d'aquests acords en forma de composició per a quatre veus mixtes.

Així doncs, el primer que farem serà escollir la successió d'acords que utilitzarem per a la nostra solució, així com escriure'ls de la manera més simple i genèrica possible (p. ex. sense inversions ni notes duplicades) en un sol pentagrama en clau de sol. A partir d'aquí, desplegarem les notes dels acords escollint alçades adequades per cadascuna d'elles, fent les inversions d'acords que siguin necessàries i duplicant o eliminant les notes que calgui per tal de respectar al màxim les normes que veurem a continuació.

4.2.2 Normes bàsiques

Les primeres normes que veurem, doncs, seran les normes que hauríem d'utilitzar fins i tot si no haguéssim d'escollir nosaltres mateixos els acords a utilitzar (que alhora són una versió relaxada[32] de les normes que es veuen al temari dels cursos de música anteriors als treballats). Aquestes normes tenen a veure amb les tessitures[33] habituals dels diferents cantants, els moviments que permetrem fer a les veus, i les notes que duplicarem en els diferents tipus d'acord.

4.2.2.1 Tessitures estàndard dels cantants

Pel que fa als cantants, l'estructura de cor mixt contempla la utilització de quatre veus diferents, de les quals dues són de dona i dues d'home. Les dues d'home seran un Baix a la veu més greu i un Tenor a la veu més aguda, i les de dona seran una Contralt a la veu més greu i una Soprano a la veu més aguda. Degut a la nomenclatura anglesa, sovint aquesta estructura la trobarem representada per les sigles SATB. Podem veure els rangs estàndard de les diferents veus a la Figura 10.

[32] Habitualment als cursos anteriors a 5è de G.P. es treballa amb unes normes bastant més estrictes per tal de restringir les opcions que farien fàcil cometre faltes a la realització. Nosaltres veurem ja la versió d'aquestes normes que s'utilitza als exercicis de 5è i 6è.

[33] Rangs de notes als quals es considera que pot arribar cada tipus de cantant.

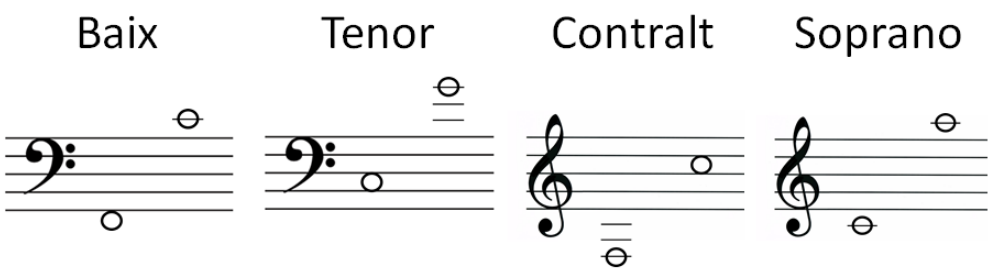

Figura 10: Tessitures estàndard de les quatre veus d'un cor mixt.

Com podem observar, el baix podrà anar des d'un Fa_1 fins a un Do_3, el tenor des d'un Do_2 fins a un Sol_3, la contralt des d'un Fa_2 fins a un Do_4 i la soprano des d'un Do_3 fins a un La_4. Aquests límits, doncs, ens marcaran els diferents rangs de notes que podrem utilitzar als exercicis per a cadascuna de les veus. Tot i això, degut que la tessitura d'un cantant depèn bastant de cada persona en concret, aquests límits no seran massa estrictes, i sovint permetrem escollir alçades una nota més enllà del límit proposat si tenim bons motius per fer-ho (p. ex. fer arribar la contralt a un Re_4 en algun moment donat).

N1: Durant la realització[34] de l'esquema, haurem de respectar les tessitures estàndard de cadascuna de les diferents veus:
N1.1: Les notes assignades al baix hauran d'estar entre el Fa_1 i el Do_3.
***N1.2*:** Les notes assignades al tenor hauran d'estar entre el Do_2 i el Sol_3.
***N1.3*:** Les notes assignades a la contralt hauran d'estar entre el Fa_2 i el Do_4.
***N1.4*:** Les notes assignades a la soprano hauran d'estar entre el Do_3 i el La_4.

4.2.2.2 Moviments de les veus

A més d'aquestes tessitures, entre una veu i la següent (excepte en el cas del baix) no hi podrà haver mai més d'una octava de diferència (Figura 11).

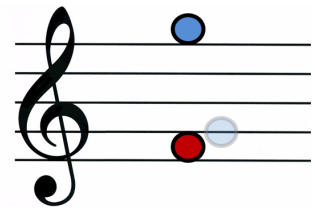

Figura 11: Exemple de distància superior a la octava entre dues veus adjacents.

N2: Entre dues veus adjacents no hi podrà haver mai més d'una octava, excepte si és entre el Tenor i el Baix.

A més, les veus no podran tampoc creuar-se:

Considerarem que dues veus es creuen si en un moment donat la veu més aguda de les dues està fent la nota més greu de les dues[35] (Figura 12a). A més, considerarem també que dues veus es

[34] El desplegament dels acords a quatre veus.
[35] Contemplar aquest cas no tindrà gaire sentit entre les dues veus d'un pentagrama (ja que no hi ha res que indiqui quina nota pertany a quina veu més que les respectives alçades), però sí que tindrà sentit a l'espai

creuen si passa el mateix en l'enllaç entre un acord i el següent (p. ex. si la veu més greu de les dues fa en el segon acord una nota més aguda que la veu més aguda en el primer acord) (Figura 12b).

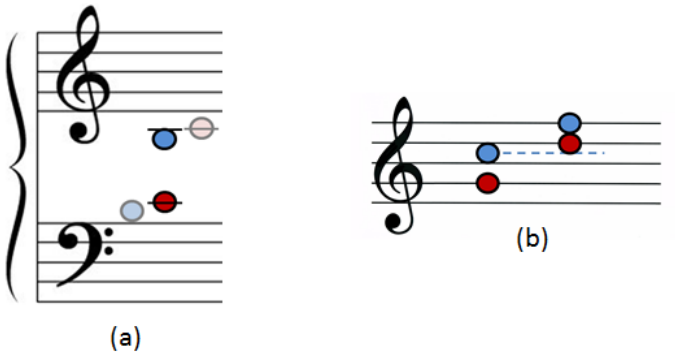

Figura 12: Exemples de creuaments entre veus.
(a) En un mateix acord (N3.1). (b) Entre un acord i el següent (N3.2).

N3: No hi podrà haver mai creuaments entre dues veus:

***N3.1*:** Per tot acord, ordenar les veus segons l'alçada de les notes que han de cantar sempre ha de resultar en l'ordenació estàndard del cor mixt (d'agut a greu, SATB).

***N3.2*:** A cap veu se li podrà assignar en un acord una nota que estigui fora els límits marcats per les notes que cantaven les veus adjacents a l'acord anterior.

D'altra banda, una altra cosa que evitarem en el moviment de les veus serà l'aparició de quintes o octaves seguides (i. e. intervals de quinta o octava generats entre les dues mateixes veus però a alçades diferents[36] durant dos o més acords consecutius) (Figura 13).

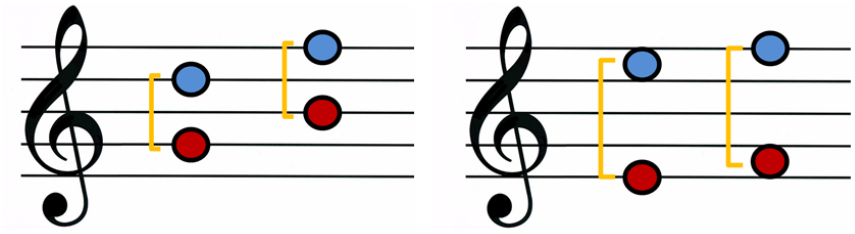

Figura 13: Exemples de quintes (esquerra) i octaves (dreta) seguides.

A més, aquesta restricció s'aplicarà també a l'interval de sèptima quan es proposi la seva utilització a les normes posteriors.

N4: Durant la realització haurem de vigilar l'aparició consecutiva de certs intervals:

***N4.1*:** No podrem fer octaves seguides.

***N4.2*:** No podrem fer sèptimes seguides.

***N4.3*:** No podrem fer quintes seguides, excepte en els casos següents:

***N4.3.1*:** Podrem fer quintes seguides si aquestes comparteixen un dels sons.

***N4.3.2*:** Podrem fer quintes seguides si aquestes tenen qualificatius diferents.

***N4.3.3*:** Podrem fer quintes seguides si l'interval entre aquestes és d'un semitò.

---

entre pentagrames, ja les línies addicionals poden acabar provocant que la nota del pentagrama de dalt sigui més greu que la del pentagrama de baix (Figura 12a).

[36] No es consideren quintes o octaves seguides si es tracta de la repetició de les mateixes notes a la mateixa octava.

Com podem veure a les normes esmentades, la realització de quintes seguides es permet en alguns casos concrets en què la configuració n'atenua o emmascara l'efecte.

D'altra banda, un altre tipus de moviment que haurem de tenir en compte serà el provinent d'intervals augmentats i disminuïts.

Si bé aquests intervals no és habitual que es generin en utilitzar les versions més senzilles de les normes, en totes les tonalitats (tant majors com menors) tindrem al menys per defecte un tritò possible[37]. D'altra banda, els intervals augmentats i disminuïts apareixeran amb molta més facilitat en versions més avançades de les normes, a mesura que es vagin incorporant ingredients nous amb les seves pertinents alteracions.

Així doncs, pel que fa als intervals disminuïts, sempre que el moviment d'una veu en generi un, l'haurem de resoldre fent anar posteriorment la veu en la direcció contrària per graus conjunts (Figura 14). El cas dels intervals augmentats, en canvi, simplement l'evitarem.

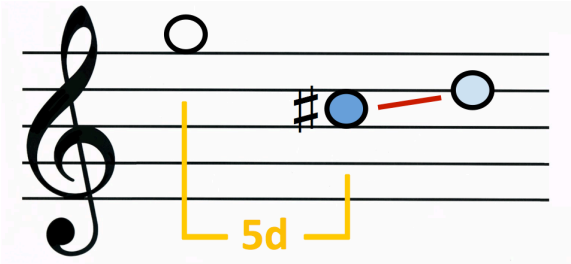

Figura 14: Exemple de resolució correcta d'un interval disminuït.

N5: Haurem d'anar amb compte amb els intervals augmentats i disminuïts:

***N5.1*:** El pas d'una veu d'un acord al següent no podrà ser mai en forma d'interval augmentat.

***N5.2*:** El pas d'una veu d'un acord al següent podrà ser en forma d'interval disminuït sempre i quan es resolgui posteriorment en direcció contrària per graus conjunts.

***N5.2.1*:** En cas que sigui necessari[38] podrem mantenir lligada la segona nota de l'interval disminuït i resoldre el moviment en acabar la lligadura.

4.2.2.3 Duplicació i supressió de notes

Degut que per defecte els acords tenen només tres notes (forma tríada), en una realització a quatre veus sovint n'haurem de duplicar alguna. La nota duplicada serà per defecte la fonamental (Figura 15). En cas que escollim no utilitzar la opció per defecte, la nota duplicada no podrà ser mai la sensible, i no podrà ser tampoc la sèptima de l'acord en qüestió[39]. A més, en cas que la nota duplicada sigui el baix d'un acord en primera inversió tríada, la duplicació només serà correcte si aquest baix és un dels graus tonals (i. e. la primera, quarta o cinquena notes de l'escala de la tonalitat).

[37] i. e. A les tonalitats majors sempre es genera un tritò entre la quarta i la sensible i a les tonalitats menors entre la segona i la sexta quan aquesta última no està alterada.

[38] Habitualment no ens hi trobarem, però un exemple on s'aplicaria seria el cas d'un interval disminuït que va a parar a una sèptima. En un cas així, pot ser que l'única manera de resoldre tant la sèptima com l'interval disminuït sigui mantenir la sèptima aguantada de manera que es posposi la resolució de l'interval.

[39] No serà habitual duplicar una nota en un acord de sèptima (ja en té quatre), però pot passar si l'acord es deixa incomplet suprimint la quinta (ho veiem tot seguit).

N6: Sovint haurem de duplicar alguna nota per poder omplir les quatre veus. Es preferirà que aquesta nota sigui la fonamental de l'acord.

***N6.1*:** En cas que no es dupliqui la fonamental, la nota duplicada no podrà ser mai la sensible de la tonalitat ni la sèptima de l'acord.

***N6.2*:** Si l'acord està en primera inversió tríada, només podrem duplicar el baix en cas que aquest sigui un grau tonal (I, IV o V).

D'altra banda, algunes de les normes que veurem ens demanaran expressament que dupliquem una nota diferent de la que duplicaríem per defecte[40].

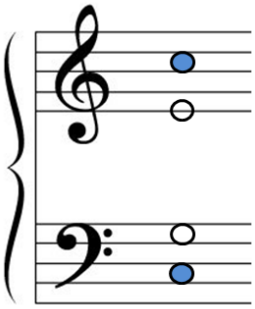

Figura 15: Exemple del desplegament d'un acord tríada a quatre veus (acord de Do M).

De la mateixa manera que sovint duplicarem una de les notes, en alguns casos (especialment al final del nostre esquema) ens interessarà suprimir una de les notes d'un acord per tal de respectar alguna de les altres normes que veurem.

N7: En cas que sigui necessari podrem suprimir la quinta de l'acord (deixar l'acord "incomplet"), però la fonamental i la tercera hi hauran d'aparèixer.

En cas que haguem de suprimir una nota d'un acord tríada o amb sèptima, la nota suprimida serà la quinta[41] (Figura 16), i dels acords sense aquesta nota en direm acords *"incomplets"*. Ho veurem amb més detall quan treballem la Cadència Perfecta amb sèptima de dominant (pàg. 35).

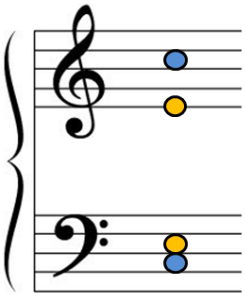

Figura 16: Exemple del desplegament incomplet d'un acord tríada (acord de Do M).

[40] Veure Cadència Trencada (pàg. 31).

[41] Degut a la relació que té amb les altres notes de l'acord, la quinta és una nota més lliure i que se sobreentén amb més facilitat, de manera que podem treure-la sense que això afecti massa a la comprensió de la música per part de l'oient. Tot i això, quan la quinta sigui una característica rellevant de l'acord (p. ex. acord de quinta augmentada) no la podrem obviar.

4.2.2.4 Utilització bàsica dels graus

A més de les normes bàsiques sobre el moviment de les veus, també haurem de tenir en compte certes restriccions respecte a l'ús dels diferents graus:

N8: Els esquemes hauran de començar en acord de tònica tríada en E.F.
N9: No utilitzarem mai el setè (VII) si porta sensible:
***N9.1*:** En menor podrem utilitzar el setè VII sempre i quan porti subtònica.

Com es pot veure a les normes anteriors, els nostres esquemes els començarem per l'acord de tònica, en forma tríada i estat fonamental, i els acabarem amb una Cadència Perfecta[42]. A més, no podrem utilitzar el setè grau en la seva forma amb sensible[43].

4.2.3 Enllaços

Les primeres normes que veurem i que haurem de tenir en compte dins l'aplicació són les que fan referència a la relació entre un acord i el següent, ja que la tria de la successió d'acords serà el primer que farem durant els exercicis d'harmonia (el desglossament a quatre veus el realitzarem després).

Com podem observar a la Figura 17, podrem tenir tres tipus d'enllaç segons quin sigui l'interval que separa les fonamentals[44] dels dos acords implicats. Aquests tipus seran *"Fort"*, *"Dèbil"*, i *"Molt fort"*, i els podrem diferenciar fàcilment si observem quines notes diferents té el segon acord respecte les que tenia el primer.

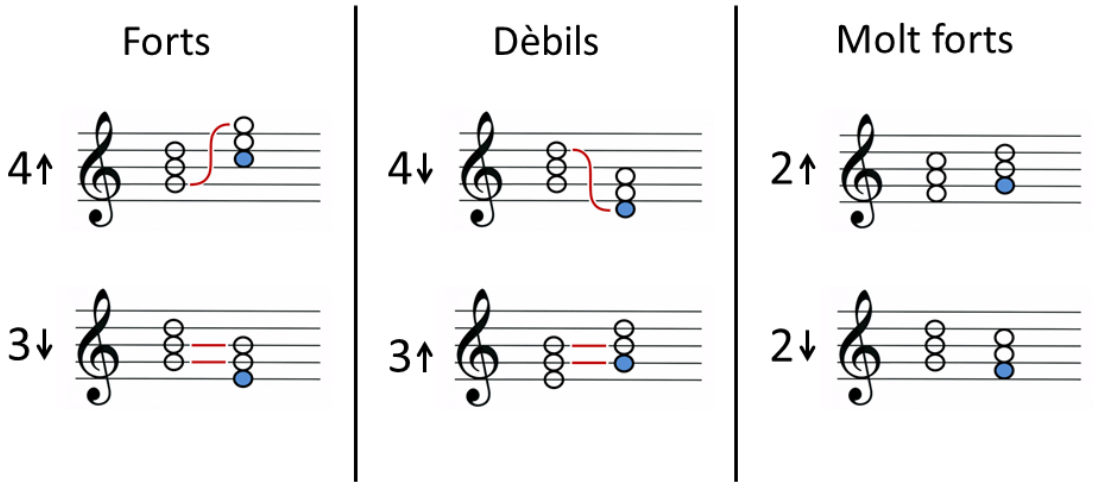

Figura 17: Classificació dels intervals segons el tipus d'enllaç i exemples.

Així doncs, si ens fixem en els exemples de la figura anterior veurem que en el cas dels enllaços forts la fonamental del segon acord és nova (no apareixia a l'acord anterior, i aquesta *"novetat"* li dóna força a l'enllaç), mentre que en el cas dels enllaços dèbils, el segon acord té una fonamental que ja havia aparegut abans (el primer acord ja la tenia, de manera que el nou acord perd força i haurem

[42] Veure l'apartat sobre la Cadència Perfecta (pàg. 29).
[43] Un VII amb sensible té en realitat la funció d'un V, i l'haurem de tractar com a tal. Tot i això, la versió de l'aplicació implementada actualment no contempla aquest cas. D'altra banda, sí que podrem fer servir el VII si porta subtònica (i. e. quan estiguem en menor si no li alterem la fonamental).
[44] Nota més greu quan l'acord no està invertit (veure "Acords", pàg. 17).

de fer alguna cosa per compensar aquesta debilitat). D'altra banda, en el cas dels enllaços molt forts totes les notes del segon acord seran noves, de manera que l'acord tindrà molta força, però per contra tindrà poca relació amb el que hagi passat abans. Així, malgrat no siguin dèbils haurem de vigilar de no abusar de la seva utilització.

Podem veure-ho amb més detall a les normes corresponents:

N10: Els enllaços dèbils els haurem de resoldre. Tenim dues opcions:
***Opció 1*:** Tornem al mateix acord del qual havíem partit (*"harmonia de brodadura"*).
***Opció 2*:** Fem un enllaç fort entre el primer acord i el tercer (*"harmonia de pas"*).
N11: No posarem mai dos enllaços molt forts consecutius.

Com es pot veure, els enllaços forts no tenen restriccions de cap tipus.

A la Figura 18 podem veure un exemple de les possibilitats que tenim a l'hora de resoldre un enllaç dèbil segons les opcions esmentades anteriorment.

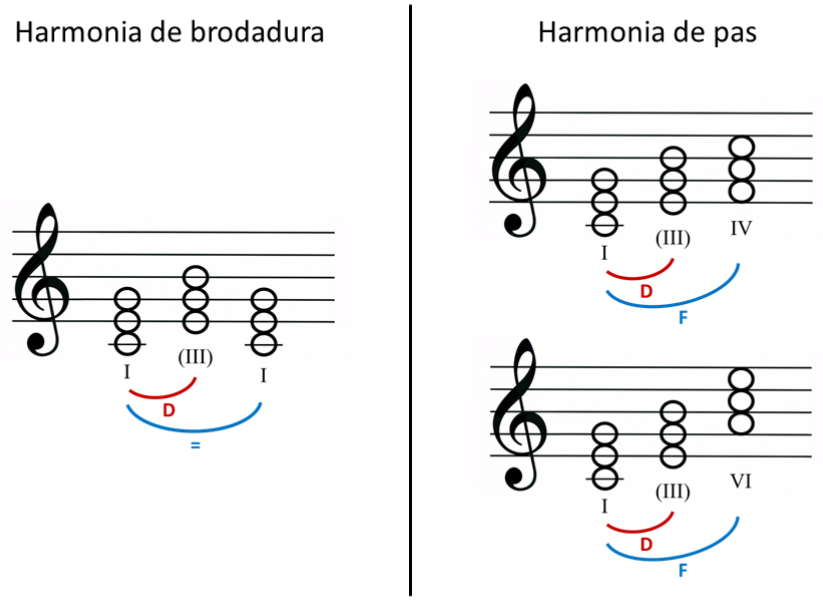

Figura 18: Resolucions possibles de l'enllaç dèbil I-III (N10, exemple en Do M).

4.2.4 Puntuació: la Cadència Perfecta

Com en el cas dels textos, a la música necessitarem una manera d'acabar les frases que doni a entendre al receptor que hi ha hagut un final. Això s'aconsegueix mitjançant la Cadència Perfecta.

N12: Els esquemes sempre hauran d'acabar amb una Cadència Perfecta[45].

L'estructura genèrica d'una cadència perfecta serà la que podem veure descrita a la Figura 19 i resolta a la Figura 20.

[45] En aquest apartat veurem la configuració estàndard de la cadència. Tot i això, la Cadència Perfecta es pot realitzar de diferents maneres (n'anirem veient d'altres més endavant), i qualsevol de les variants possibles serà vàlida com a final de l'esquema.

$$\ldots \text{SD} \underbrace{\text{I}_4^6 \ \text{V}_{\text{E.F.}}}_{{}^6_4\,cadencial} \text{I}^*_{\text{E.F.}}$$

Figura 19: Estructura habitual de la Cadència Perfecta.
*acabant sempre la melodia en tònica.

On SD significa qualsevol acord que pugui fer la funció de Subdominant (II o bé IV), i els acords amb E.F. hauran d'estar en *estat fonamental*[46]. A més, per tal que sigui una Cadència Perfecta, la melodia haurà d'acabar sempre en tònica (la nota més aguda de l'últim acord haurà de ser la tònica de la tonalitat).

D'altra banda, la combinació $\text{I}_4^6\ \text{V}_{\text{E.F.}}$ s'anomena habitualment ${}^6_4\,cadencial$, i és una de les tres fórmules preestablertes que existeixen a l'hora d'utilitzar els acords en segona inversió. Tot i això, com veurem més endavant, nosaltres resoldrem l'acord de ${}^6_4$ d'una manera més lliure (pàg. 36).

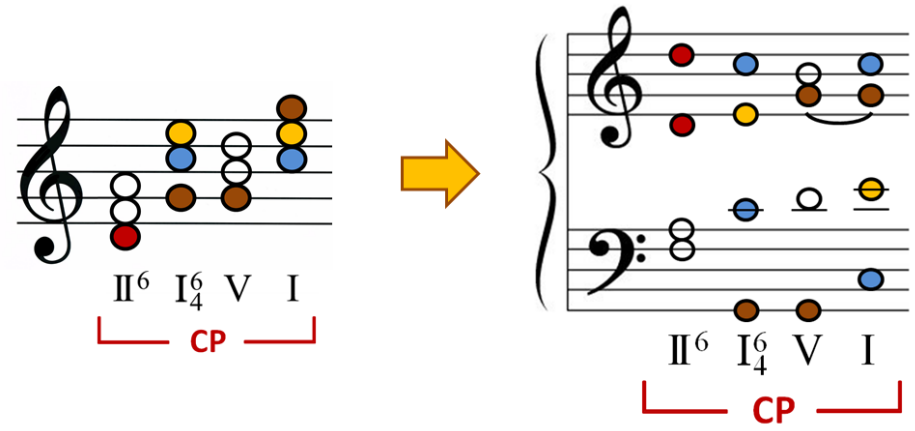

Figura 20: Exemple de realització d'una Cadència Perfecta (Do M).

Podem veure un exemple de realització d'una Cadència Perfecta (amb les notes més rellevants marcades en colors diferents) a la Figura 20. Podem observar també que l'últim acord està complet. Això serà fàcil d'aconseguir a la versió tríada de la cadència, però en canvi quan s'utilitzi el cinquè amb sèptima hi haurà casos en què la resolució completa no serà trivial[47].

4.2.5 Acord de Dominant

L'acord de la dominant (V) tindrà una rellevància especial a les composicions musicals, ja que és un acord que tendeix a tònica (i. e. per la manera com sona, l'oient s'espera que vagi seguit d'un primer grau).

Així doncs, la tònica serà sempre la nostra opció per defecte quan haguem posat un cinquè.

N13: La dominant per defecte sempre anirà a tònica $(\text{V} \rightarrow \text{I})$.

[46] Sense inversió (veure "Inversions dels acords", pàg. 18)
[47] Quan el cinquè porti sèptima i no estigui invertit per defecte un dels dos acords haurà de quedar incomplet. Veurem les solucions a aquest inconvenient a l'apartat 4.2.7.2 (pàg. 35).

***N13.1*:** Si la dominant no va a tònica, haurà d'anar a algun dels seus substituts.

Com podem deduir de la N13.1, podrem saltar-nos el comportament natural de l'acord de dominant si ho fem de la manera correcta. Això implicarà trencar amb el que s'espera l'oient (sorprendre'l), i per tant s'haurà de fer d'una manera controlada si volem evitar desmuntar massa la coherència de la música que estem escrivint.

Així doncs, la manera més habitual que tenim de saltar-nos el moviment per defecte de l'acord de dominant és el que anomenem *"Cadència Trencada"*.

4.2.5.1 Cadència Trencada

La Cadència Trencada és una *estafa*[48] que consisteix a canviar l'acord de primer grau que vindria després d'un V per un acord de IV o VI. Això ho podem fer perquè els acords de IV i VI són els altres dos acords de la tonalitat que, en la seva forma tríada, tindran en algun lloc la tònica[49], de manera que aquesta aparició de la tònica com a nota dins l'acord ens serveix per compensar una part d'aquest trencament que suposa que l'acord de tònica en sí no hi sigui.

D'altra banda, per reforçar aquest efecte esmorteïdor de la tònica, el que farem serà duplicar-la en el moment de realitzar els acords a quatre veus (en lloc de duplicar la fonamental de l'acord com es faria habitualment). A més, durant la realització intentarem també fer els mínims moviments possibles (buscarem que les veus facin intervals tan petits com sigui possible) per tal de suavitzar així el procés.

***N13.2*:** La dominant podrà fer una Cadència Trencada $(\mathrm{V} \rightarrow \mathrm{IV})$ o $(\mathrm{V} \rightarrow \mathrm{VI})$, sempre i quan a l'hora de de realitzar-la se segueixin els criteris següents:

***N13.2.1*:** Haurem de duplicar la tònica (al segon acord).

***N13.2.2*:** Haurem de fer els mínims moviments possibles dins de cada veu(en passar d'un acord a l'altre).

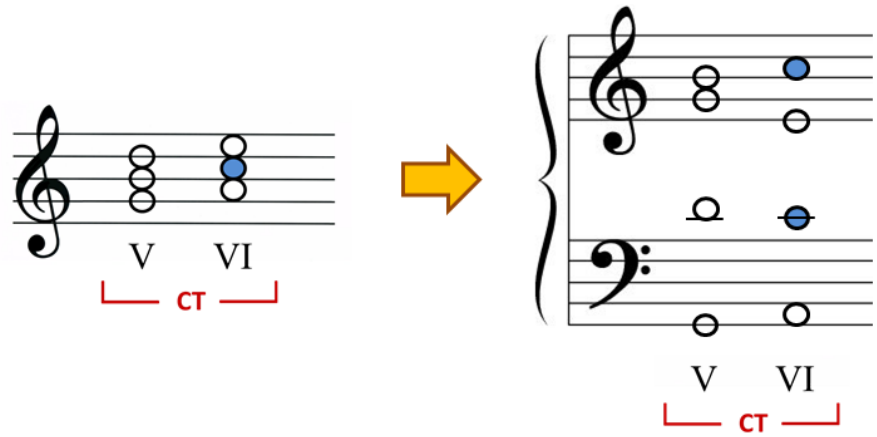

Figura 21: Exemple de realització de la Cadència Trencada (V-VI, Do M).

Podem veure a la Figura 21 un exemple de realització d'una Cadència Trencada. Com es pot observar, al segon acord s'ha duplicat la tònica de la tonalitat (i. e. el Do), i s'han resolt les veus fent

[48] En el sentit que estem prometent a l'oient una cosa (un acord de primer grau) que no li donarem. (Miquel Suñer, 2012).

[49] A l'acord de IV tenim la tònica a la quinta, mentre que a l'acord de VI tenim la tònica a la tercera.

intervals tan petits com ha estat possible (en la major part dels casos per graus conjunts, i quan no s'ha pogut per terceres).

4.2.6 Conducció melòdica

Una de les coses que més importància tindran a l'hora de *realitzar* un esquema serà la tria d'una bona conducció melòdica (i. e. una melodia que tingui sentit) (Figura 22). El fet de forçar això, de fet, serà una de les coses que ens generaran la major part dels errors que ens apareixeran durant la realització, ja que tindrem la veu de dalt[50] més o menys fixada per la melodia escollida, i majoritàriament haurem de jugar només amb les tres de baix per tal de solucionar els problemes que es vagin generant.

En general, el que és interessant en una melodia és que es mogui d'una manera suau (i. e. que predominin les lligadures i els graus conjunts, que no generi una gràfica dentada), i que les vegades que salti[51] s'equilibri posteriorment el salt tornant per graus conjunts en la direcció contrària. D'altra banda, evitarem fer salts seguits en la mateixa direcció, així com oscil·lar al voltant d'una sola nota o repetir la nota més aguda[52].

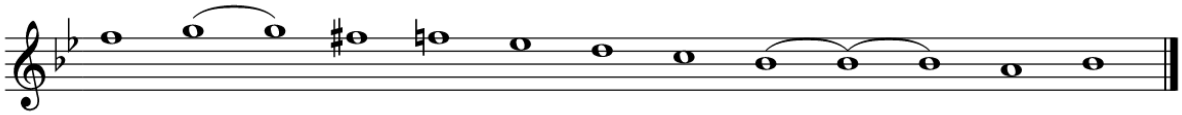

Figura 22: Exemple d'una melodia adequada a les directrius N14 (Si♭ M).

A més, com hem vist a l'apartat sobre la Cadència Perfecta (4.2.4, pàg. 29), la melodia haurà d'acabar amb la tònica de la tonalitat.

N14: Per defecte construirem la melodia mitjançant la utilització de lligadures i graus conjunts.
- ***N14.1***: Podrem fer salts, però els considerarem desequilibris i els haurem de gestionar bé:
  - ***N14.1.1*:** Per equilibrar un salt, tornarem al punt d'origen per graus conjunts.
  - ***N14.1.2*:** Mai farem salts seguits en la mateixa direcció.
  - ***N14.1.3***: La "gràfica" de la melodia haurà de ser corba (no dentada).
- ***N14.2*:** El "pinyol" (la nota més alta) no es repetirà mai.
- ***N14.3***: Procurarem no oscil·lar tota l'estona al voltant d'un mateix punt.
- ***N14.4***: Podrem començar amb la tònica, però preferirem començar amb una nota diferent.
- ***N14.5***: La melodia acabarà sempre en tònica (C.P. – N12).

4.2.7 Sèptima de Dominant

En una segona visita a l'acord de dominant veurem que podem donar a aquest una sonoritat especial si n'aprofitem la versió amb sèptima (V_7). Aquesta sonoritat especial es deu a l'interval dissonant que es crea entre la sèptima de l'acord i la seva fonamental.

[50] La veu més aguda.
[51] Direm que la melodia salta quan l'interval entre una nota i la següent sigui més gran que una segona.
[52] La nota més aguda d'una frase té una rellevància especial, i perdrà força si es repeteix en dos moments diferents (serà correcte, però, repetir-la si és en diversos acords consecutius, ja que en aquest cas quedarà lligada).

Degut a la naturalesa poc agradable de les dissonàncies, una correcta utilització d'aquestes dins la música implica suavitzar-ne l'entrada i la sortida[53] de manera que no suposin un xoc brusc per l'oient. Tot i això, com anirem veient, tota dissonància ben utilitzada suposa afegir color i caràcter a les composicions.

En el cas que ens ocupa de la sèptima de dominant, per tal de preparar l'interval de sèptima el que farem és forçar que una de les dues notes que formen la dissonància entri lligada o per graus conjunts (connectant-la així d'una manera més forta a l'acord anterior). Així doncs, si una de les dues entra lligada o per graus conjunts l'altra tindrà la possibilitat de saltar (Figura 23). Tot i això, si la que salta és la sèptima, no la farem saltar mai cap a baix[54].

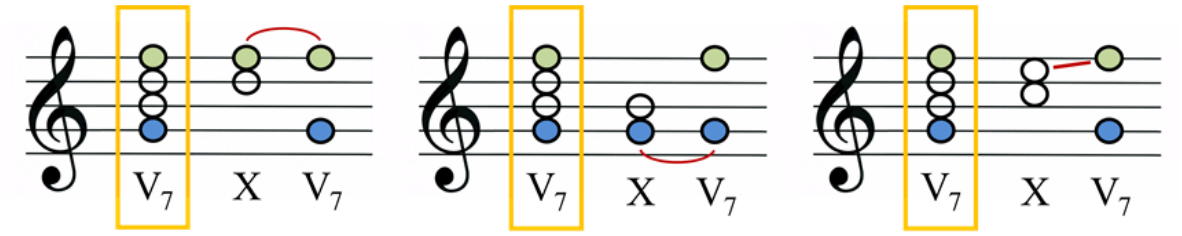

Figura 23: Exemples d'una correcta preparació de l'interval de sèptima segons N15.1 (Do M).

D'altra banda, per tal de resoldre la tensió que genera l'interval de sèptima tindrem tres opcions: baixar la sèptima per graus conjunts, mantenir la sèptima aguantada (lligada) o bé pujar la sèptima cromàticament[55].

Com podem veure a la Figura 24, el darrer dels casos no el podem trobar en cap de les possibilitats[56] que hem vist fins ara per a la resolució de l'acord de dominant. Tot i això, degut que anirem afegint noves variants i tipus d'acords al conjunt d'ingredients disponibles, eventualment ens trobarem casos en què aquest tercer cas sí que es pot aplicar.

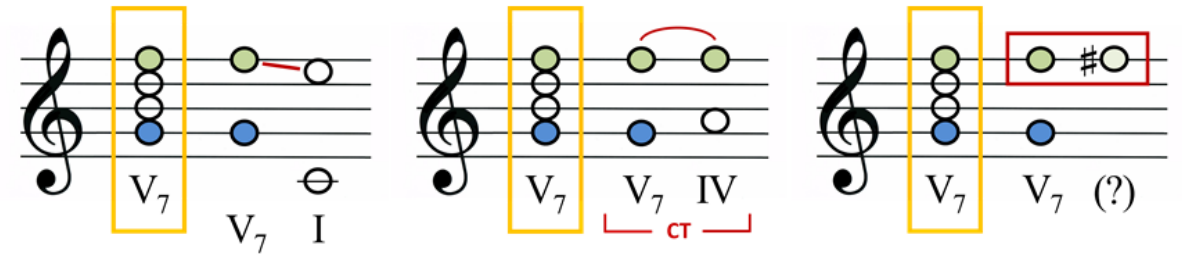

Figura 24: Exemples d'una correcta resolució de la sèptima segons N15.2 (Do M).

Pel que fa a les altres veus implicades en l'enllaç, habitualment farem anar la sensible a tònica, mantindrem la fonamental i farem baixar la quinta. Tot i això, en aquests casos la directriu és molt

[53] Anticipar d'alguna manera l'interval dissonant, així com resoldre'n posteriorment la dissonància en la direcció cap a on l'interval tendeix.
[54] No farem mai un salt descendent cap a la sèptima. Això no treu, però, que una sèptima pugui entrar per graus conjunts descendents (ja que només és un salt si l'interval és al menys una tercera).
[55] Augmentar-la en un semitò sense canviar-ne el nom de nota (i. e. fer un uníson augmentat ascendent).
[56] Veure normes N13 (pàg. 30).

laxa i molt sovint trencarem aquests comportaments en pro de solucionar altres errors més greus relacionats amb els moviments de les veus.

N15: Podrem posar sèptima sobre l'acord de dominant sempre i quan es prepari i resolgui:

***N15.1*:** Per preparar la sèptima una de les dues notes haurà d'entrar lligada o per graus conjunts.

***N15.1.1*:** Si arribem a la sèptima a través d'un salt, aquest no podrà ser mai descendent.

N15.2:Haurem de resoldre la sèptima segons una d'aquestes tres opcions:

***Opció 1*:**Fem baixar la sèptima per graus conjunts.

***Opció 2*:** Mantenim la sèptima aguantada.

***Opció 3*:** Pugem la sèptima cromàticament.

***N15.3*:** De moment per les altres notes preferirem per defecte els moviments següents:

***N15.3.1*:** Mantenir la fonamental (lligada).

N15.3.2: Fer anar la sensible a tònica.

***N15.3.3*:** Fer baixar la quinta.

En tot cas, una de les coses que veurem si intentem seguir les normes N15 és que esdevé impossible posar la seqüència V-I amb els dos acords complerts quan ambdós acords estan en estat fonamental (i. e. a l'hora de realitzar, els moviments obligatoris de les veus només van a parar a les notes fonamental i tercera del segon acord, i cap dels moviments resulta en la quinta).

Això habitualment no serà un problema perquè l'acord de sèptima de dominant es pot invertir amb facilitat[57]. Tot i això, sí que ens ho trobarem molt a la cadència perfecta, degut que allà l'acord de dominant haurà d'estar sempre obligatòriament en estat fonamental.

Així doncs, si volem posar sèptima de dominant a la Cadència Perfecta (i serà quelcom preferible) haurem de treure la quinta d'un dels dos acords (Figura 25) o bé utilitzar una de les tres tècniques que es proposen més endavant per a posar els dos acords complets[58].

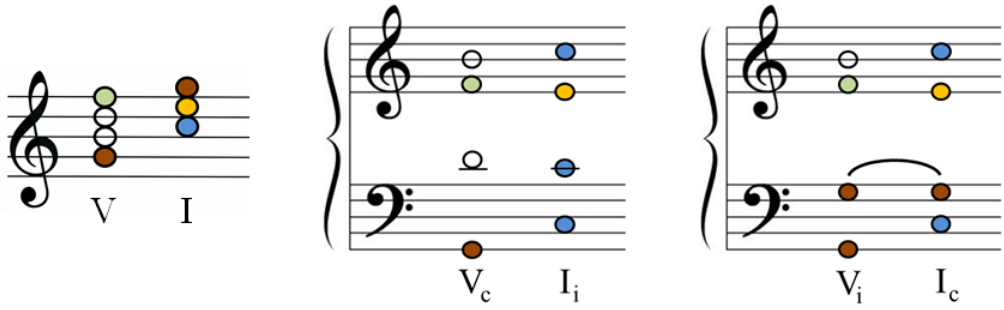

Figura 25: Exemples de resolució per defecte de la seqüència V_{EF} I_{EF} (Do M).

4.2.7.1 Resolució irregular de la sèptima

Degut al caire estricte de les normes referents a la resolució de la sèptima, sovint a l'hora de realitzar el desplegament de l'esquema a quatre veus tindrem ja predeterminat el comportament d'una de les quatre notes. Això fa que en segons quins casos no tinguem la llibertat que ens

[57] Quan l'acord de dominant està invertit, la seva fonamental mantinguda acaba essent la quinta que ens faltava a l'acord de tònica.

[58] Veure apartat 4.2.7.2 (pàg. 35)

agradaria per tal de solucionar altres errors, o per tal d'escollir una bona conducció melòdica per al nostre esquema.

En aquests casos, sovint ens interessarà sacrificar la resolució habitual de la sèptima per tal de facilitar-nos trobar una configuració de les veus que ens sigui satisfactòria (Figura 26).

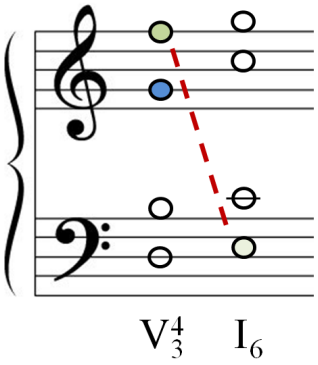

Figura 26: Exemple de resolució irregular de la sèptima (Do M).

La resolució irregular de la sèptima (RI_7) consistirà, doncs, en portar la resolució de la sèptima a una altra veu (és a dir, haurà d'existir dins el segon acord la nota que faria possible la resolució habitual de la sèptima, però no caldrà que aquesta nota es trobi a la mateixa veu en què estava la sèptima al primer acord). D'altra banda, un cop trencada la resolució habitual, la resta de veus les arreglarem per sentit comú[59].

***N15.4*:** Si tenim una bona raó, podrem saltar-nos la resolució habitual de la sèptima.
N15.4.1: La sèptima s'haurà de poder resoldre igualment, malgrat resolgui en una altra veu.
***N15.4.2*:** Farem resolucions irregulars un màxim de dues vegades per exercici.

Per tal de conservar la dificultat de resoldre sèptimes de la manera habitual, als nostres exercicis d'harmonia se'ns permetrà un màxim de dues resolucions irregulars. A més, per tal de donar a entendre que la resolució irregular ha estat quelcom intencionat, acostumarem a marcar-la amb una línia de punts (Figura 26).

4.2.7.2 Cadència perfecta amb els dos acords complets.

Degut que l'acord de dominant sempre apareix a la Cadència Perfecta, serà habitual també substituir en aquest cas l'acord per la seva versió amb sèptima. Com hem vist anteriorment, però, aquesta substitució provoca que per defecte un dels dos acords hagi de sortir incomplet.

Per tal d'evitar aquest inconvenient i poder realitzar la Cadència Perfecta amb els dos acords complets, disposarem de tres tècniques o possibilitats diferents. Aquestes tres possibilitats seran les següents: resoldre de manera irregular la sèptima (RI_7), resoldre de manera irregular la sensible (RI_S) i utilitzar una de les variants permeses del moviment per quintes paral·leles (anomenada habitualment les *"quintes de Rimsky").*

Com podem veure a la Figura 27, en el primer cas hem fet servir la possibilitat que tenim de fer quintes seguides si una de les notes implicades és coincident. D'altra banda, els segon cas respon a

[59] i. e. respectant les normes bàsiques sobre el moviment de les veus.

la norma N15.4, i el tercer prové de saltar-nos les directrius N15.3 respecte al moviment de les notes en la resolució de l'acord V_7.

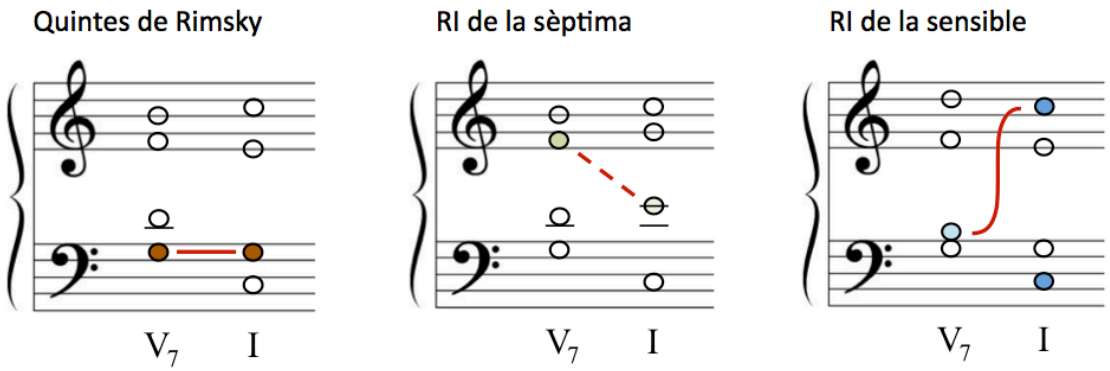

Figura 27: Exemples de realització de la Cadència Perfecta amb els dos acords complets (Do M).

4.2.8 Acord de Segona Inversió

En els primers cursos d'harmonia la resolució de l'acord de segona inversió $(\frac{6}{4})$ es planteja d'una manera mecànica que respon a un seguit de fórmules preestablertes (cadencial, de brodadura i de pas). La utilització d'aquestes, però, malgrat simplifica la quantitat d'informació a tenir en compte per l'estudiant, limita en gran mesura les possibilitats de què aquest disposa en el moment de fer la realització a quatre veus.

Així doncs, a partir dels cursos més avançats es planteja la versió general de les normes que marquen la resolució de la segona inversió, deixant llibertat a l'estudiant per realitzar-la de maneres que no responguin a cap de les fórmules esmentades anteriorment. Aquestes normes seran similars a les que hem vist per a l'interval de sèptima, i aniran dirigides a suavitzar l'entrada i la sortida de l'interval de quarta generat entre el baix i la fonamental.

N16: Si posem acords tríades en segona inversió, haurem de seguir les pautes següents:

***N16.1*:** Tant el baix com la quarta hauran d'entrar i sortir lligats o per graus conjunts.

***N16.1.1*:** Si la quarta està duplicada, només caldrà preparar i resoldre una de les dues aparicions de la nota (i podrem preparar una de les dues i resoldre l'altra).

***N16.2*:** No posarem mai dos acords de $\frac{6}{4}$ consecutius.

D'altra banda, degut a la sonoritat d'aquesta configuració (i com en el cas dels enllaços molt forts) no ens interessarà posar-ne mai dos de seguits.

4.2.9 Mode menor

Degut a les característiques de l'escala menor esmentades a l'apartat sobre tonalitats (Taula 10), la realització d'esquemes en aquest mode implicarà tenir en compte algunes normes extra que gestionin les variants d'aquesta escala.

Una de les primeres coses que haurem de tenir en compte és que la flexibilitat que ens permeten les variants de l'escala menor farà més fàcil l'aparició d'intervals augmentats i disminuïts, de manera que haurem de posar especial èmfasi en les normes que els tracten (N5.2, pàg. 26).

D'altra banda, haurem de gestionar les alteracions de les notes sexta i sèptima de l'escala d'una manera que respongui a l'escala melòdica[60]. És a dir, l'interval ascendent el realitzarem utilitzant la sensible i pujant la sexta, mentre que l'interval descendent anirà amb la sexta natural i la subtònica (Figura 28).

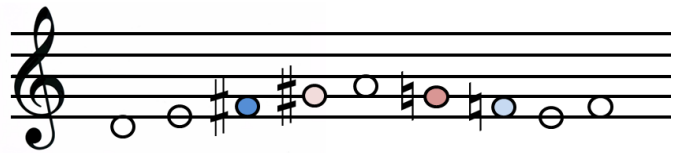

Figura 28: Exemple d'una gestió correcta de l'interval entre les notes dels graus sisè i setè (Do M).

En cas que no es vegin implicades les dues notes, considerarem que la versió per defecte del sisè és la que es correspon amb l'armadura i que la versió per defecte del setè és fent de sensible.

N17: Haurem de vigilar amb l'interval de segona entre la sexta i la sèptima:

***N17.1*:** Si apareixen seguides i el moviment és ascendent, pujarem la 6a (i posarem sensible).

***N17.2*:** Si apareixen seguides el moviment és descendent, posarem subtònica (i la 6a natural).

A més, en el cas dels modes menors tindrem dues possibles configuracions per a l'acord de V amb dos comportaments intrínsecament diferents, ja que l'acord de V només serà un acord de dominant si porta sensible (i per tant només haurà de respondre a les normes N13 en aquest cas).

N18: L'acord de dominant sempre portarà sensible (en cas contrari no serà de dominant).

Així doncs, la utilització de l'acord de V sense sensible serà possible, però aquest no tindrà cap de les obligacions que imposem als acords de dominant (el tractarem com un acord qualsevol).

4.2.10 Sèptima sobre els altres acords

De la mateixa manera que hem posat la sèptima sobre l'acord de dominant, podem posar sèptima també sobre qualsevol dels altres acords, sempre i quan es respectin les normes pertinents.

Si bé en un acord de dominant la sèptima sempre serà menor, hi haurà casos entre la resta dels acords en què la sèptima resultarà ser major. Aquests casos els haurem de tractar amb una cura especial, degut que la dissonància de l'interval de sèptima en aquest cas serà més forta.

Els casos de sèptima menor, en canvi, els tractarem igual com tractàvem la sèptima de l'acord de dominant.

N19: Podrem posar sèptimes sobre qualsevol acord, sempre i quan tinguem en compte el següent:

***N19.1*:** Si la sèptima de l'acord és menor, l'haurem de preparar i resoldre com a l'acord de sèptima de dominant (normes N15.1, N15.2 i N15.4).

***N19.2*:** Si la sèptima de l'acord és Major, l'haurem de preparar fent entrar una de les dues notes lligada, i l'haurem de resoldre com en el cas menor (N19.1).

***N19.3*:** Si posem sèptima als dos acords de la cadència trencada, només tindrem en compte el comportament de les sèptimes (deixarem de banda les normes N13.2.1 i N13.2.2).

[60] Veure "Escales i modes" (pàg. 14).

***N19.4*:** Si posem sèptima Major sobre l'acord de VII del menor (VII amb el sisè alterat), haurem d'aguantar-la i posteriorment fer anar el sisè a sensible.

Com podem veure a les normes, el fet que la dissonància de la sèptima Major sigui més pronunciada ens porta a voler forçar que entri lligada (fer-la entrar per graus conjunts en aquest cas no es considera suficient per suavitzar-la). A més, el cas específic del VII amb el sisè alterat ens implicarà una contradicció, ja que la sèptima voldrà resoldre cap a baix però tindrem un sisè alterat que implicarà un moviment ascendent cap a sensible. L'única manera de resoldre-ho, doncs, serà resoldre la sèptima aguantant-la i posteriorment fer pujar el sisè alterat a la sensible que demana.

D'altra banda, a partir d'aquest punt les directrius N15.3 passaran a un segon pla i les considerarem poc rellevants.

5 EINES ACTUALS

Actualment existeixen algunes eines bastant equivalents a l'aplicació proposada, però són només per ordinador, no són massa modernes i no són conegudes a la majoria d'escoles de música. Tot i això, sí que existeixen i s'utilitzen molt sovint alguns programes similars que realitzen funcions significativament diferents però que formarien part del mateix àmbit.

En aquest apartat, doncs, farem un cop d'ull a algunes d'aquestes eines, començant per les més diferents però més utilitzades i acabant per les més desconegudes però que són més properes al projecte que ens ocupa.

5.1 Edició de partitures

En un primer lloc podríem ubicar les aplicacions que s'encarreguen d'ajudar en l'edició de partitures. Aquestes eines bàsicament actuen com un equivalent musical dels processadors de textos, en tant que permeten entrar els diferents símbols i anotacions que podem voler incloure en una partitura, i posteriorment preparar-ne la configuració per a imprimir-les.

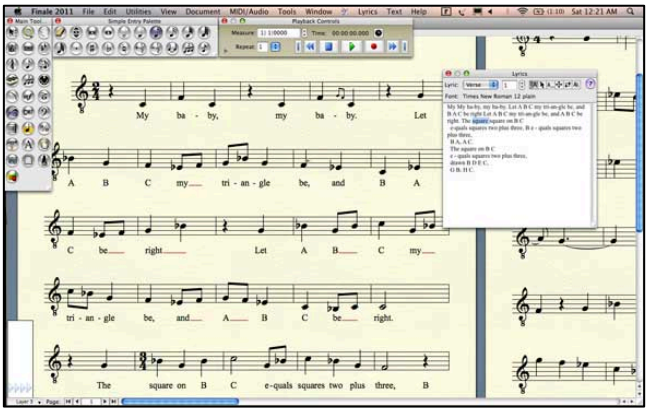

Figura 29: Interfície de Finale 2011 (MacWorld).

Els processadors de partitures, però, a diferència dels processadors de textos i de l'aplicació que ens ocupa, habitualment no tenen cap eina similar a un corrector ortogràfic (si bé alguns disposen de connectors que realitzen parts senzilles de la correcció, com la detecció de 5es i 8ves paral·leles).

L'eina de què sí que disposen habitualment és un reproductor d'àudio que interpreta la partitura escrita per permetre'ns escoltar-la.

En tot cas, aquest tipus de programes treballaran amb un conjunt de símbols i elements de la notació musical molt extens que no necessitem en realitat per als exercicis d'harmonia que tractarem, i d'altra banda no ofereixen una manera completa i senzilla de corregir les composicions, que és el que ens interessarà per a l'aplicació.

D'altra banda, pel que fa a exemples de programes concrets, en versió d'escriptori tindríem com a més importants **Finale** i **Sibelius**, mentre que en versió online existeix una alternativa gratuïta anomenada **Noteflight**[61].

[61] Gran part de les imatges de l'apartat sobre el llenguatge musical (pàg. 19) s'han generat en aquest lloc web.

5.2 Reproducció de seqüències d'acords

En un segon lloc dins aquesta llista d'eines similars podríem trobar els programes que s'encarreguen de reproduir una base musical a partir d'una sèrie d'acords donada. Aquests, s'acostumen a utilitzar en àmbits de música més moderna per a poder practicar peces o successions d'acords de manera individual sense haver de disposar de la resta de la banda.

Així doncs, podem donar al programa una seqüència d'acords, una velocitat i un estil, i el programa ens reproduirà la seqüència simulant uns quants instrumentistes que toquen en l'estil escollit, de manera que podem practicar les nostres veus dins una peça o la nostra habilitat per a fer solos tenint com a referència la base que el programa ens proposa.

Aquests programes seran similars a l'aplicació proposada en dos sentits: en primer lloc, permetran introduir una seqüència d'acords (com a la primera fase de la resolució dels nostres exercicis), i en segon lloc, permetran reproduir la seqüència escollida (de manera similar a la reproducció que oferirem a la segona fase dins l'aplicació).

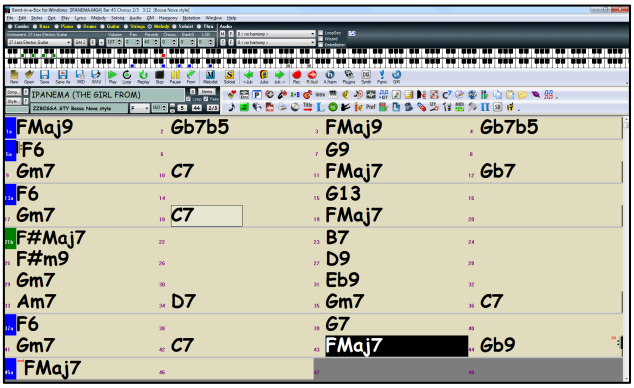

Figura 30: Interfície de Band-in-a-box (Wikipedia).

La diferència, doncs, serà la part de separar els acords en quatre veus, ja que amb aquest tipus de programes l'únic que ens interessa són els acords, i per tant no podem ni editar la distribució de les notes en cada acord, ni corregir aquesta distribució mitjançant les normes d'harmonia. A més, pel mateix motiu, no disposaran d'una vista que ens mostri les notes en un pentagrama com en el cas dels editors de partitures (i. e. mostraran només la notació que representa els acords escollits).

Alguns exemples de programari dins aquest bloc podrien ser **Band-in-a-box**, que es presenta en versió d'escriptori, i l'aplicació per a mòbils **iReal Pro** (ambdues de pagament).

5.3 Eines de composició assistida

En un punt mig entre l'edició de partitures i la correcció d'harmonia trobarem unes aplicacions que mitjançant tècniques d'intel·ligència artificial ajuden a completar o generar les obres musicals que componem. En aquest cas, parlem d'eines relativament antigues i certament poc conegudes, però seran algunes de les eines que veurem que més a prop es trobaran del que volem fer en aquest projecte.

En un primer lloc i donant una importància més gran a la generació automàtica d'obres musicals, tindríem **Tonica Fugata**. Aquest programa (en versió d'escriptori) permet entrar una melodia, i a

partir d'aquesta generar (principalment mitjançant xarxes neuronals) una obra completa en base a un estil escollit.

D'altra banda, permet també aprofitar la programació que incorpora en matèria d'intel·ligència artificial per tal de trobar alguns dels tipus d'errors en composicions creades per l'usuari.

En tot cas, però, el programa està sobre tot centrat en la generació de música (partitures), deixant com a prestació secundària l'anàlisi i correcció. A més, a diferència del cas que ens ocupa, el programa està preparat per gestionar el ritme de les peces, cosa que afegeix una complexitat en l'edició que a nosaltres en realitat no ens interessa.

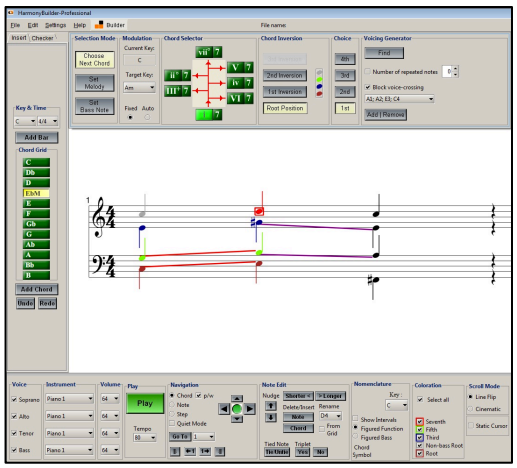

Figura 31: Interfície del programa Harmony Builder (saxonline.it).

D'altra banda, en segon lloc dins aquest bloc tindríem **Harmony Builder**, també en versió d'escriptori. En aquest cas, el programa sí que està pensat per a la composició a quatre veus, i sí que té un fort èmfasi a la composició que no té en compte el ritme (com la que gestionarà la nostra aplicació) i en el fet de tenir-hi en compte les normes d'harmonia. Tot i això, el programa està principalment orientat a facilitar una composició correcta des del principi (i. e. assisteix en la tria dels acords i mostra quines configuracions de les quatre veus són possibles respectant les normes i paràmetres escollits).

Aquestes característiques, si bé poden ser útils per a composar sense necessitat de tenir grans coneixements d'harmonia, no resulten massa interessants en termes d'aprenentatge, ja que es dona tot fet a l'usuari i no es deixa gaire llibertat a aquest per explorar la creativitat i poder-se equivocar i aprendre de la correcció.

D'altra banda, probablement la interfície d'aquest programa és excessivament complexa per al propòsit que ens ocupa (especialment si tenim en compte que ha de poder funcionar en un mòbil).

5.4 Correctors d'exercicis d'harmonia

Finalment, tindrem dins aquesta secció un programa fonamentalment orientat al mateix objectiu que l'aplicació escollida per al nostre projecte: permetre la realització d'exercicis d'harmonia i corregir-los.

El programa en qüestió s'anomena **Harmony Practice 3**, i amb una interfície prou senzilla permet introduir esquemes musicals a quatre veus, per posteriorment buscar-ne els errors i mostrar-ne la normativa associada a l'usuari. De nou, el programa existeix únicament en versió d'escriptori.

Un petit inconvenient que podríem trobar al programa és que la correcció no es realitza de manera automàtica (cal llançar-la expressament), però degut que es pot llançar prement una sola tecla no ens importarà massa. A més, si bé en les primeres versions la normativa i els errors s'havien de consultar en finestres a part, actualment la correcció es mostra directament sobre l'exercici, i la normativa és molt més accessible, també.

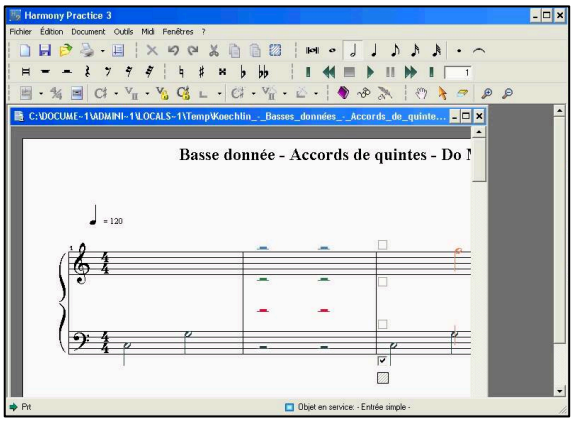

Figura 32: Interfície de l'Harmony Practice 3 (YouTube).

D'altra banda, una altra diferència que tindríem respecte a la manera de realitzar els exercicis que ens interessa és que el programa **HE3** va directament a la realització, mentre que nosaltres acostumarem a escollir primer la seqüència d'acords a utilitzar.

En tot cas, però, pot ser una bona referència a tenir en compte durant el desenvolupament de l'aplicació.

5.5 Conclusions

Com hem vist en aquest apartat, si bé existeixen moltes eines similars a l'aplicació que es proposa crear, només una de les que hem trobat en la recerca ha resultat oferir un servei pràcticament equivalent. A més, d'entre les eines que s'han esmentat, les més properes a l'aplicació proposada comparteixen el fet d'estar desenvolupades només en versió d'escriptori, de manera que no tenim constància de cap aplicació per a mòbils que realitzi la mateixa funció que l'aplicació que volem programar.

D'altra banda, totes aquestes aplicacions comparteixen també un disseny antic i carregat que no s'ajusta massa al disseny que esperen els usuaris actualment, de manera que haurem d'intentar que la nostra aplicació tingui una interfície més senzilla i intuïtiva que els programes que hem vist si volem que encaixi en el mercat de les aplicacions per a mòbils.

6 EINES I ENTORNS DE PROGRAMACIÓ

Per al desenvolupament de l'aplicació s'ha escollit la utilització de l'entorn de software lliure Eclipse (concretament la versió per a desenvolupadors de Java), així com les diferents eines de Google necessàries per treballar amb Android dins el programa (l'SDK i alguns connectors). D'altra banda, s'ha escollit treballar sobre el framework LibGDX.

Aquesta tria, en primera instància respon a una qüestió pragmàtica: l'estiu anterior havia après a utilitzar-los per desenvolupar un joc matemàtic[62] per Android, de manera que ja coneixia les eines i m'agradava la manera com funcionaven.

D'altra banda, però, els motius que em van portar a escollir aquestes eines quan havia de decidir quines eines aprendria a utilitzar per fer el joc, també segueixen essent bons motius per escollir-les en el context d'aquesta nova aplicació: permeten molta llibertat de disseny (en lloc d'haver de gestionar els elements del sistema Android, bàsicament tens un full en blanc on posar tu mateix el que vulguis), i permeten el desenvolupament multiplataforma (de manera que un mateix codi serveix per Android, per la versió d'escriptori, i si es volgués per iOS, fent alhora molt fàcil executar el programa a l'ordinador sense haver de connectar necessàriament el mòbil per provar els canvis).

D'altra banda, per tal d'editar i redimensionar les imatges necessàries per a dibuixar els elements gràfics de l'aplicació, sovint s'ha utilitzat el software d'edició fotogràfica Photoshop, en la creació del logotip s'ha utilitzat Flash i en l'edició dels àudios s'ha utilitzat Audacity.

6.1 Eclipse

Eclipse és, segons es defineixen al seu web, una comunitat d'eines, projectes i grups de treball col·laboratiu de codi obert.

En el nostre cas, la part d'aquesta comunitat que ens interessa són els entorns integrats de desenvolupament; en concret la versió pensada per al desenvolupament en Java[63].

Els entorns de desenvolupament Eclipse són finalment una aplicació desenvolupada en Java que ens permet gestionar els projectes i recursos, escriure codi que serà comprovat en busca d'errors, i fer córrer l'aplicació a l'ordinador o en simuladors de mòbil segons convingui. La versió bàsica d'aquests entorns és genèrica, i Eclipse permet a través del que anomenen *Marketplace* afegir connectors (*plugins*) segons convingui per tal d'adaptar-la a qualsevol tipus de feina que es pugui voler realitzar (dins les que ofereixen, si més no).

Així doncs, la versió per a Java que hem escollit no és res més que aquesta versió bàsica ja preparada amb els connectors necessaris per desenvolupar aquest llenguatge.

Tot i això, degut que voldrem desenvolupar per Android, haurem d'afegir manualment a l'aplicació els connectors corresponents a aquest àmbit (en el nostre cas el connector anomenat ADT Bundle,

[62] "Primes!" (de "Rusca8") a Google Play Store.
[63] Eclipse IDE for Java developers.

tot i que actualment s'està tendint a substituir per un connector equivalent més modern degut que el desenvolupament de l'original ja no tindrà continuïtat[64]).

6.2 Android

El terme *Android* engloba tot un conjunt de programari de Google en relació als dispositius mòbils. Tot i això, la part que ens interessarà a nosaltres serà purament el sistema operatiu que porta el mateix nom, en tant que és el lloc on haurà de córrer la nostra aplicació.

En concret, ens interessarà saber quines eines hem d'utilitzar durant el desenvolupament de la nostra aplicació per tal que aquesta es pugui fer servir a la majoria dels mòbils que utilitzin aquest sistema.

El conjunt d'eines que ofereix Google als desenvolupadors de la seva plataforma és el que anomenem *Android SDK*, i són aquestes eines, en la versió que ens interessi, el que haurem d'enllaçar amb el nostre entorn *Eclipse* per tal que aquest sàpiga com exportar les aplicacions a la plataforma mòbil, fent-les compatibles.

6.3 LibGDX

6.3.1 Introducció

LibGDX és un marc de treball (*framework*) pensat per al desenvolupament de jocs que funcionin a totes les grans plataformes amb els mínims canvis possibles entre el codi d'una plataforma i l'altra.

Bàsicament, LibGDX consisteix a treballar amb un seguit de projectes enllaçats a un projecte central (un per cada plataforma que es vol tenir en compte).

Dins aquest projecte central (**core**) es desenvolupa tot el codi comú (en els casos que ens ocupa serà pràcticament tot el codi), i dins els projectes específics (**desktop**, **android**, **ios**, ...) bàsicament hi tindrem el codi que ens permet executar des de cada plataforma concreta el codi que tindrem al projecte central.

D'altra banda, en cas que quelcom s'hagi d'abordar de maneres diferents a les diferents plataformes, podrem aconseguir-ho creant un mètode genèric al codi central, i sobreescrivint-lo amb mètodes específics diferents per a cada plataforma (ubicats als projectes de cadascuna d'aquestes plataformes). Tot i això, la major part d'aquestes distincions entre plataformes ja les gestionen les pròpies biblioteques del LibGDX, de manera que en la majoria dels casos podrem treballar directament sobre el codi central mitjançant les funcions ambivalents que se'ns ofereixen.

6.3.2 Pantalles

La unitat bàsica del nostre codi en LibGDX seran les pantalles (**Screen**). Cadascuna d'aquestes pantalles disposarà d'un seguit de mètodes que s'executaran en moments concrets: quan passem a ensenyar-la (**show**), quan passem a amagar-la per ensenyar-ne una altra (**hide**), quan es canvien les mides del dispositiu o finestra (**resize**), i en bucle continu mentre s'està mostrant (**render**), entre d'altres.

[64] El desenvolupava Google i ha deixat de fer-ho degut que ara se centren en el seu propi entorn de desenvolupament, anomenat *"Android Studio"*.

La utilització de pantalles, doncs, ens permetrà separar el codi en apartats d'una manera clara, i gestionar fàcilment els canvis entre aquests apartats (tant pel que fa als elements gràfics com pel que fa a la gestió de clics i qüestions per l'estil).

A efectes pràctics, es definirà cada pantalla en forma de classe i posteriorment se'n crearà una instància dins la classe principal de l'aplicació[65] per tal de poder-la utilitzar.

6.3.2.1 SpriteBatch, Camera i Viewport.

Per tal de treballar amb els elements gràfics de l'aplicació disposem d'un seguit de classes que s'encarreguen de gestionar les diferents parts del procés de dibuix.

Degut que tindrem molta variació en el rang de mides de pantalla en què es farà córrer l'aplicació (especialment si tenim en compte que podrà córrer també en versió d'escriptori), necessitarem una manera senzilla de traduir les mides del món de la nostra aplicació a les mides del dispositiu que la utilitzi. L'element que s'encarrega de fer això és la càmera (**OrthographicCamera**).

Tot i això, la càmera no entén de proporcions, de manera que unes proporcions de dispositiu diferents a les de la càmera resultarien en una imatge distorsionada. Per evitar això, disposem del segon element, la finestra (**Viewport**). Aquesta, s'encarrega de fer una gestió intel·ligent de la diferència de proporcions que hi pugui haver entre la càmera i el dispositiu, en base a uns paràmetres que podem escollir. Així, podem escollir que distorsioni la imatge, però també podem escollir deixar barres negres als costats o ampliar el camp de visió més enllà del que veu la càmera per tal d'omplir-les. Alhora, podem centrar la càmera al punt que vulguem del món de l'aplicació, permetent-nos gestionar a quin costat de la pantalla volem deixar les barres negres o el camp de visió extra. A l'apartat de disseny veurem com s'ha gestionat aquesta diferència de mides en el cas que ens ocupa.

Un cop configurada la càmera i la finestra de la nostra pantalla, disposem d'un últim element que serà l'encarregat de dibuixar-hi les imatges, formes o fonts que vulguem utilitzar: el lot (**SpriteBatch**).

El lot l'utilitzarem dins la funció **resize**, i el que farà és calcular totes les ordres gràfiques que se li donin, i posteriorment mostrar-ne el resultat responent a la càmera i finestra escollida.

Com hem comentat a la introducció, aquestes funcions són genèriques, de manera que al final donaran un mateix resultat per a totes les plataformes.

6.3.3 Gestió d'entrades

Per tal de gestionar d'una manera genèrica les diferents entrades que pugui tenir l'aplicació (en el nostre cas els diferents esdeveniments que puguin originar els clics), disposem d'una altra eina de LibGDX anomenada **InputAdapter**. Aquesta s'afegeix a cadascuna de les pantalles per tal que puguin rebre aquestes entrades i respondre-hi a partir de les funcions que hi afegeix.

[65] Es podria crear més d'una pantalla del mateix tipus si fos necessari, però habitualment, degut que la definició d'una pantalla serà bastant específica, acostumarem a crear només una instància a partir de cada definició.

En el nostre cas, les funcions utilitzades seran **touchDown** (quan es premi en algun punt de la pantalla), **touchDragged** (quan s'arrossegui després de prémer), i **touchUp** (quan es deixi anar després de prémer). Totes aquestes esperen els esdeveniments que les representen i s'activen quan succeeixen, donant-nos informació sobre el lloc del dispositiu en què s'ha premut, així com altres dades que poden ser útils per a plataformes més específiques (quin botó del ratolí ha estat o quin és el número del dit a la pantalla si s'ha premut a més d'un lloc a la vegada).

En tot cas, aquestes dades vindran referides a les posicions del dispositiu, de manera que les haurem de traduir a posicions del món de l'aplicació a través de la mateixa finestra que utilitzem per dibuixar.

6.3.4 Recursos

Amb LibGDX els recursos seran compartits entre els diferents projectes (habitualment es troben dins la carpeta del projecte Android i la resta de plataformes funcionen amb un accés directe a aquesta, evitant duplicats). A efectes pràctic, però, accedirem als recursos directament des del codi central.

Pel que fa a la nostra aplicació, bàsicament tindrem tres tipus de recursos: imatges (**Sprite**), fonts (**BitmapFont**), i talls curts d'àudio (**Sound**).

En el cas de les imatges, primer haurem d'importar-les en forma de textura (**Texture**), i després d'aplicar-ne els filtres pertinents les assignarem a un **Sprite**, que és la classe amb què treballarem a l'hora de dibuixar. Els filtres, al seu torn, són la manera com es processarà la imatge quan s'hagi d'augmentar o encongir per respondre a la mida d'aquesta dins el món de l'aplicació, i al seu torn a la mida real del dispositiu on l'aplicació estigui funcionant.

Veurem la tria que se n'ha fet en el cas que ens ocupa a l'apartat de disseny.

D'altra banda, pel que fa a les fonts, la millor manera que té LibGDX de gestionar-les és convertir-les en imatges abans d'utilitzar-les. Així doncs, haurem de carregar un objecte de la classe **BitmapFont** per a cadascuna de les mides[66] que es vulguin utilitzar dins l'aplicació. Aquesta càrrega de les diferents fonts vindrà a càrrec d'un generador de fonts anomenat **FreeTypeFontGenerator**, que alhora agafa els paràmetres escollits d'un objecte que els recull, anomenat **FreeTypeFontParameter**.

Per tal d'escollir els colors de les fonts i les imatges, el que farem és importar-les en blanc, i pintar-les en el moment de dibuixar escollint el color en què dibuixa el lot o bé el color assignat a la font, respectivament.

Pel que fa a les imatges que tinguin més d'un color, les podem importar ja en els colors originals, i posteriorment dibuixar-les amb el lot en blanc (el color del lot multiplica el color de la imatge, de manera que les imatges blanques agafen el color del lot, però les imatges pintades queden igual si el lot les pinta en blanc).

Veurem la gestió que s'ha fet de fonts, imatges i color dins l'aplicació que ens ocupa a l'apartat de disseny.

[66] En proporció al món de l'aplicació, només, ja que l'ampliació o reducció d'aquestes mides per respondre a la mida real del dispositiu també corre a càrrec de càmera i finestra.

D'altra banda, els sons els carregarem directament als objectes de tipus **Sound**, i els podrem reproduir directament des de l'objecte demanant-li **play**.

6.3.5 Vectors

Com podem veure al codi, el tipus de vector (**Array**) que s'ha utilitzat en la majoria dels casos no és **l'Array** estàndard de Java, sinó la versió pròpia que inclou LibGDX. Aquesta versió pròpia d'aquestes llistes està especialment pensada per respondre amb velocitat i eficiència, i inclou una sèrie de mètodes que ens seran molt útils a l'hora d'afegir, eliminar i editar elements.

6.3.6 Gradle i GDX Setup

Per tal de facilitar la construcció de les aplicacions a partir del codi, així com la gestió dels múltiples projectes que estarem construint en realitat (un per cada plataforma), LibGDX utilitza un sistema anomenat *Gradle*.

Gradle és un sistema que realitza dues funcions diferenciades: en primer lloc construcció, i en segon lloc gestió de dependències.

La primera d'aquestes funcions bàsicament el que fa és facilitar la feina de convertir tots els fitxers del nostre projecte en una aplicació que pugui córrer a la plataforma desitjada (i alhora facilitar que aquesta conversió es pugui fer des de qualsevol IDE[67] sense haver de canviar res).

La segona, d'altra banda, s'encarrega de permetre que puguem utilitzar biblioteques de terceres parts sense necessitat d'haver-les de tenir entre els recursos del nostre projecte. És a dir, tindrem dins el nostre projecte només un fitxer que indica quines biblioteques es necessiten, i Gradle s'encarregarà de descarregar aquestes biblioteques d'una font central de programari (en cas que no les tinguem) i desar-les en un fitxer fora el nostre projecte (fent per tant que puguem compartir els recursos entre les aplicacions).

En el cas que ens ocupa, la utilització d'aquest sistema és tan fàcil com utilitzar un petit programa que ofereix LibGDX per a preparar el nostre projecte, anomenat *GDX setup*. Aquest bàsicament consisteix en una finestra on escollirem variables relacionades amb el nom del projecte i la versió d'Android que volem utilitzar, així com les biblioteques que voldrem incloure a la nostra aplicació (p. ex. motors de física, gestió d'entrades i similars). Una vegada escollit tot això, el programa generarà tots els fitxers que ha de tenir l'aplicació inicialment, i ens deixarà l'estructura de fitxers ja preparada per poder començar a treballar en el codi tan bon punt haguem importat el projecte a Eclipse.

[67] Entorn Integrat de Desenvolupament.

7 ESPECIFICACIONS

L'aplicació proposada s'ha plantejat seguint el temari fins a 6è de Grau Professional de Música. Tot i això, pràcticament tot el temari complex que es té en compte forma part només del curs de 6è, així que amb una versió molt més senzilla que la que es planteja es podria cobrir del tot l'harmonia que s'estudia fins a 5è.

En tot cas, per tal de gestionar aquest augment progressiu de la complexitat dels exercicis s'ha plantejat el desenvolupament de l'aplicació d'una manera incremental (començant per les funcions que responen a les normes més bàsiques i afegint poc a poc complexitat i funcionalitats més avançades).

Es planteja a continuació, doncs, el desglossament de les especificacions distribuïdes en un seguit d'iteracions, de manera que a cada iteració s'esmenten les funcionalitats que caldria afegir (obviant les especificacions ja tractades en iteracions anteriors).

Així doncs, el resultat d'implementar les especificacions de totes les iteracions aquí descrites seria una aplicació amb totes les funcionalitats que seria interessant tenir en un programa capaç de respondre a qualsevol exercici plantejat dins el temari de 6è. Tot i això, com s'ha comentat anteriorment, pel que fa a l'abast d'aquest projecte només tindrem en compte la primera iteració.

7.1 Primera Iteració (Tríades i sèptimes)

Aquesta primera iteració inclourà el temari fins a les primeres classes de 5è. Degut que és la iteració que s'ha tingut com a referència durant el desenvolupament de l'aplicació, s'han deixat indicades a la llista amb un color més clar les especificacions que havien estat previstes però finalment no s'han implementat.

Així doncs, les especificacions previstes per a l'aplicació dins el marc d'aquest projecte seran les indicades a continuació:

- Triar un to:
  - Triar la nota
  - Triar el mode (M/m)
  - Generar aleatòriament
- Triar enunciat:
  - Triar entre els diferents acords i estructures
    - Triar un grau:
      - Permetre graus del primer al setè (I-VII)
      - Permetre deixar a elecció de qui el resol (X)
    - Triar tipus d'acord:
      - Tríada (5)
      - Amb sèptima (7ª)
      - Permetre no especificar
    - Triar inversions i altres casos especials:
      - Per acords tríades:
        - Estat Fonamental (EF)

- Primera inversió (6)
- Segona inversió (6/4)
- Permetre no especificar

- Per acords amb sèptima:
  - Estat Fonamental (EF)
  - Primera inversió (6/5)
  - Segona inversió (4/3)
  - Tercera inversió (2)
  - Permetre no especificar

- Triar altres estructures:
  - Cadència Trencada (C.T.)
  - Resolució irregular de la sèptima (RI_7)
- Triar quantitats i relacions entre ells:
  - Triar vegades que ha d'aparèixer un ingredient
    - Triar si és obligatori que cada vegada sigui diferent
      - Diferents acords
      - Diferents inversions
    - Demanar que se'n posin *"tants com es pugui"* (<ingr>!)

- Gestionar els ingredients
  - Editar ingredients
  - Esborrar ingredients
- Generar aleatòriament

- Entrar un esbós
  - Mostrar l'esbós
    - Mostrar el to
    - Dibuixar clau, armadura i pentagrama
    - Dibuixar els acords
      - Dibuixar les notes al pentagrama
      - Dibuixar les alteracions
        - Marcar alteracions opcionals
    - Escriure els graus
      - Escriure els graus en el to original
    - Escriure els enllaços
    - Escriure les cadències
      - Escriure la Cadència Trencada (C.T.)
      - Escriure la Cadència Perfecta (C.P.)
    - Marcar referències i punts perillosos
      - Indicar els acords amb Sèptima Major (7_M)
    - Permetre desplaçament quan l'esquema és més llarg que la pantalla
  - Entrar i esborrar acords
    - Triar grau
    - Triar tipus d'acord
    - Esborrar un acord
    - Inserir un acord enmig

        - Editar un acord
        - Marcar que hem acabat l'esbós
    - Mostrar referències
        - Mostrar els ingredients de l'enunciat
- Corregir un esbós
    - Indicar errors en els enllaços
        - Indicar enllaços dèbils mal resolts (N10)
        - Indicar enllaços molt forts seguits (N11)
        - Indicar dominants mal resoltes (N13)
    - Comprovar que hi ha Cadència Perfecta (N12) (Forma estàndard)
    - Indicar errors en la utilització dels graus (N9)
    - Indicar errors en les inversions
        - Indicar 6/4 seguits (N16.2)
    - Mostrar les normes que justifiquen els errors
- Entrar realització a quatre veus
    - Mostrar la realització
        - Mostrar el to
        - Dibuixar les claus, l'armadura i els pentagrames
        - Dibuixar les notes als pentagrames
            - Dibuixar les alteracions accidentals
            - Dibuixar alteracions de cortesia
            - Dibuixar les lligadures
        - Escriure els graus i xifrats
        - Escriure les cadències
            - Escriure la Cadència Trencada (C.T.)
            - Escriure la Cadència Perfecta (C.P.)
        - Marcar les resolucions irregulars
        - Permetre desplaçament quan l'esquema és més llarg que la pantalla
    - Permetre reproducció de l'esquema
        - Reproduir la realització a quatre veus
    - Entrar i esborrar notes
        - Marcar acord actual
        - Triar veu (S,A,T,B)
        - Triar nota de l'acord (F,3,5,7,9)
        - Triar l'octava de la nota (canviar índex acústic)
        - Triar les alteracions opcionals
            - Mostrar quan hi ha alteracions alternatives
        - Canviar d'acord (navegar per l'esquema)
    - Mostrar referències
        - Mostrar els ingredients de l'enunciat
        - Mostrar restriccions de quantitat
            - Mostrar vegades que s'ha fet servir la Resolució Irregular (RI_7)
        - Mostrar les tessitures de les veus
        - Mostrar l'esbós previ

                - Marcar acord actual a l'esbós
                - Diferenciar notes ja utilitzades
- Corregir la realització a quatre veus
    - Mostrar les diferents faltes comeses
        - Tessitures dels cantants (N1)
        - Distància excessiva entre dues veus (N2)
        - Creuaments entre dues veus (N3)
        - Moviments paral·lels de 8va, 7a i 5a (N4)
        - Intervals augmentats i disminuïts (N5)
        - Notes duplicades i suprimides (N6 i N7)
        - Inversions dels acords (N8 i N16.2)
        - Cadències Trencades (N13.2)
        - Conducció melòdica (N14)
        - Preparació i resolució de sèptimes (N15 i N19)
        - Preparació i resolució dels 6/4 (N16)
        - Alteracions opcionals del menor (N17 i N18)
    - Mostrar les normes que justifiquen les diferents faltes
    - Mostrar explicacions que justifiquen les diferents alertes

7.2 Segona iteració (Dominants Secundàries i Napolitana)

En una segona iteració ens interessarà ampliar algunes funcionalitats ja implementades a la iteració anterior, així com ampliar el temari fins admetre pràcticament tot el que es veu a cinquè. Això implica tot un seguit d'ingredients i normes que no s'han explicat a l'apartat sobre harmonia, però que d'una manera resumida implicarien els requeriments següents:

- Triar un to:
    - Generar aleatòriament
        - Triar nivell de dificultat
- Triar enunciat:
    - Triar entre els diferents acords i estructures
        - Triar tipus d'acord:
            - Dominant Secundària (D.S.)
            - *Combinacions dels tipus que tenim fins ara*
            - Casos especials:
                - Sexta napolitana ($\flat\flat b_{nap}$)
        - Triar inversions i altres casos especials:
            - Per acords de la família de la dominant:
                - Sèptima disminuïda de sensible (7_{ds})
        - Triar quantitats i relacions entre ells:
            - Enllaçar dos requeriments (demanar que es posin un rere l'altre)
    - Generar aleatòriament
        - Triar nivell de dificultat:
            - Activar i desactivar ingredients
            - Demanar ingredients *obligatoris* (coses que voldré practicar segur)

- Entrar un esbós
    - Mostrar l'esbós
        - Marcar referències i punts perillosos
            - Indicar els moviments obligatoris
            - Indicar els moviments preferibles
            - Indicar el to d'on prové una D.S.
    - Entrar i esborrar acords
        - Inserir un acord enmig
            - Inserir l'acord de D.S. que va al següent
    - Mostrar referències
        - Mostrar el mapa dels tons veïns
- Entrar realització a quatre veus
    - Mostrar la realització
        - Dibuixar les notes als pentagrames
            - Dibuixar les fonamentals virtuals
    - Entrar i esborrar notes
        - Canviar d'acord (navegar per l'esquema)
            - Automatitzar el desplaçament per facilitar entrar tota la veu seguida
        - Bloquejar notes impossibles
            - Impedir que la fonamental i la novena es posin a la vegada
    - Mostrar referències
        - Mostrar mapa dels tons veïns
- Corregir la realització a quatre veus
    - Donar una qualificació aproximada en funció de les faltes fetes
- Mostrar esquema acabat a pantalla complerta
    - Exportar imatge de l'esquema acabat
- Gestionar els esquemes
    - Desar esquemes
    - Obrir esquemes
    - Esborrar esquemes

7.3 Tercera iteració (Primeres Modulacions)

En tercer lloc, voldrem aconseguir que l'aplicació pugui gestionar les modulacions, és a dir, el fet de començar l'exercici en una tonalitat i acabar-lo en una altra sense canvis bruscos dins la seqüència d'acords. Això portarà més feina, i implicarà més o menys els requeriments següents:

- Triar enunciat:
    - Triar entre els diferents acords i estructures:
        - Triar modulacions:
            - Triar tonalitat de destinació
        - Triar quantitats i relacions entre ells:
            - Triar vegades que ha d'aparèixer un ingredient
                - Triar si les vegades han de ser a cada part d'una modulació
- Entrar un esbós

    - Mostrar l'esbós
        - Mostrar el to
            - Mostrar el to d'arribada (quan modula)
        - Escriure els graus
            - Escriure els graus en el to d'arribada (quan modula)
            - Marcar la Zona Híbrida
    - Mostrar referències
        - Mostrar el mapa dels tons veïns
            - Mostrar el mapa ampliat quan modula
    - Planificar les modulacions
        - Triar to d'arribada
        - Gestionar la Zona Híbrida (Z.H.)
            - Començar la Zona Híbrida
            - Tancar la Zona Híbrida
- Entrar realització a quatre veus
    - Mostrar la realització
        - Mostrar el to
            - Mostrar el to d'arribada (quan modula)
        - Escriure els graus i les inversions (to d'origen i to d'arribada)
        - Marcar la Zona Híbrida
    - Mostrar referències
        - Mostrar mapa dels tons veïns
    - Mostrar mapa ampliat quan modula

7.4 Última iteració (Segona ampliació i modulacions avançades)

Finalment, voldrem incorporar a l'aplicació la resta dels acords i possibilitats ofertes dins el temari de Grau Professional, que inclouran noves ampliacions de la tonalitat i altres maneres de modular a tonalitats més llunyanes. Els requeriments que ens quedaran per donar l'aplicació per acabada seran els següents:

- Triar enunciat:
    - Triar entre els diferents acords i estructures
        - Triar tipus d'acord:
            - De l'homònim menor (m)
            - *Combinacions dels tipus que tenim fins ara*
            - De sexta augmentada clàssic ($6A_{class}$)
            - Casos especials:
                - Dominant de la napolitana ($\cancel{V}Ib_{dom.nap.}$)
                - Tríada de quinta augmentada (V_{5+})
        - Triar modulacions:
            - Triar tècniques de modulació
                - Zona Híbrida
                - Pedal
                - Enharmonia

- Entrar un esbós
    - Planificar les modulacions
        - Planificar un pedal
            - Mostrar graus dels acords sobre el pedal entre parèntesi
            - Indicar la nota del pedal en els acords
        - Entrar una enharmonia
- Corregir un esbós
    - Detectar combinacions d'acords que seran incompatibles en el moment de realitzar
- Entrar realització a quatre veus
    - Mostrar la realització
        - Escriure els graus i les inversions (to d'origen i to d'arribada)
            - Marcar els graus del pedal
        - Marcar els acords de les enharmonies
- Corregir la realització a quatre veus
    - Permetre mode més restrictiu seguint les normes de cursos baixos

8 DISSENY

En aquest apartat veurem les decisions més rellevants que s'han anat prenent a nivell de disseny, tant pel que fa a l'aspecte gràfic com pel que fa a la tria de variables i altres elements relacionats amb el codi. Aquestes decisions que anirem veient es poden veure reflectides a l'aplicació resultant o bé al codi adjunt als annexes.

La versió de l'aplicació que veurem aquí respon a la implementació de la primera iteració presentada a l'apartat d'especificacions. Les següents iteracions que s'hi descriuen no s'abordaran dins aquest TFG, però s'hi han deixat indicades com a referència del que s'haurà de fer posteriorment en cas que es vulgui continuar el desenvolupament de l'aplicació.

8.1 Estructura de l'aplicació

L'aplicació consta d'un conjunt d'unitats de compilació (arxius de codi) que podem separar en un seguit de grups diferenciats.

En primer lloc tenim la unitat central, que representa l'aplicació, i és la unitat a partir de la qual accedirem a les altres.

En segon lloc tenim les pantalles[68], que recullen el codi dels diferents apartats que tindrà l'aplicació. En el nostre cas, s'han definit principalment com a pantalles diferents la tria dels ingredients de l'enunciat (**PIng**), la tria dels acords de l'esbós (**Pant**), i la realització a quatre veus (**Pant2**). A més, s'han definit un seguit de pantalles secundàries per a funcions més concretes: una per quan l'aplicació s'està carregant (**PInici**), una pel menú principal de l'aplicació (**PMenu**), una per a la secció "Quant a..." (**PAbout**), i una per permetre triar el mode de l'esquema abans de començar amb els ingredients (**PTon**[69]).

A la Figura 33 podem veure la manera com han estat organitzades aquestes pantalles: cadascuna de les set tindrà la seva secció principal ressaltada amb un contorn més gruixut, mentre que els requadres adjacents sense ressaltar ens indicaran la resta de seccions a què es pot accedir dins de cadascuna.

Pel que fa a les dades, les variables relacionades amb la mida de pantalla, així com altres elements compartits entre pantalles (especialment entre les pantalles d'esbós i realització) s'han desat sovint únicament a la pantalla de l'esbós (la primera que es va programar), de manera que les altres pantalles accedeixen a aquesta quan necessiten obtenir-ne les dades comuns.

D'altra banda, s'ha muntat un tercer conjunt d'unitats de compilació dedicat a recollir diferents funcions i variables d'utilitat (un conjunt de biblioteques pròpies), per tal de mantenir aquestes funcions i variables separades dels fils principals i poder-les utilitzar d'una manera més invisible. Aquestes unitats portaran per nom lletres majúscules, i el contingut de cadascuna d'elles respondrà a un àmbit diferent. Així, tindrem les funcions relacionades amb l'animació a **A**, les funcions relacionades amb la música a **H**[70], les funcions relacionades amb la normativa d'harmonia a **N**, les

[68] Veure la descripció que se n'ha fet anteriorment a l'apartat sobre LibGDX (pàg. 44).
[69] "Pantalla de la Tonalitat" (es preveu incorporar una tria completa del to directament en aquesta pantalla).
[70] Inicial d'Hemiolia, que és el nom que vaig escollir per a l'aplicació: "hemiolia" és un concepte de música que en termes generals significa una cosa així com "trencar amb el ritme establert".

que tenen a veure amb el color a **S**[71], i les altres funcions genèriques que puguin ser d'utilitat a **F** (majoritàriament relacionades amb el tractament del text).

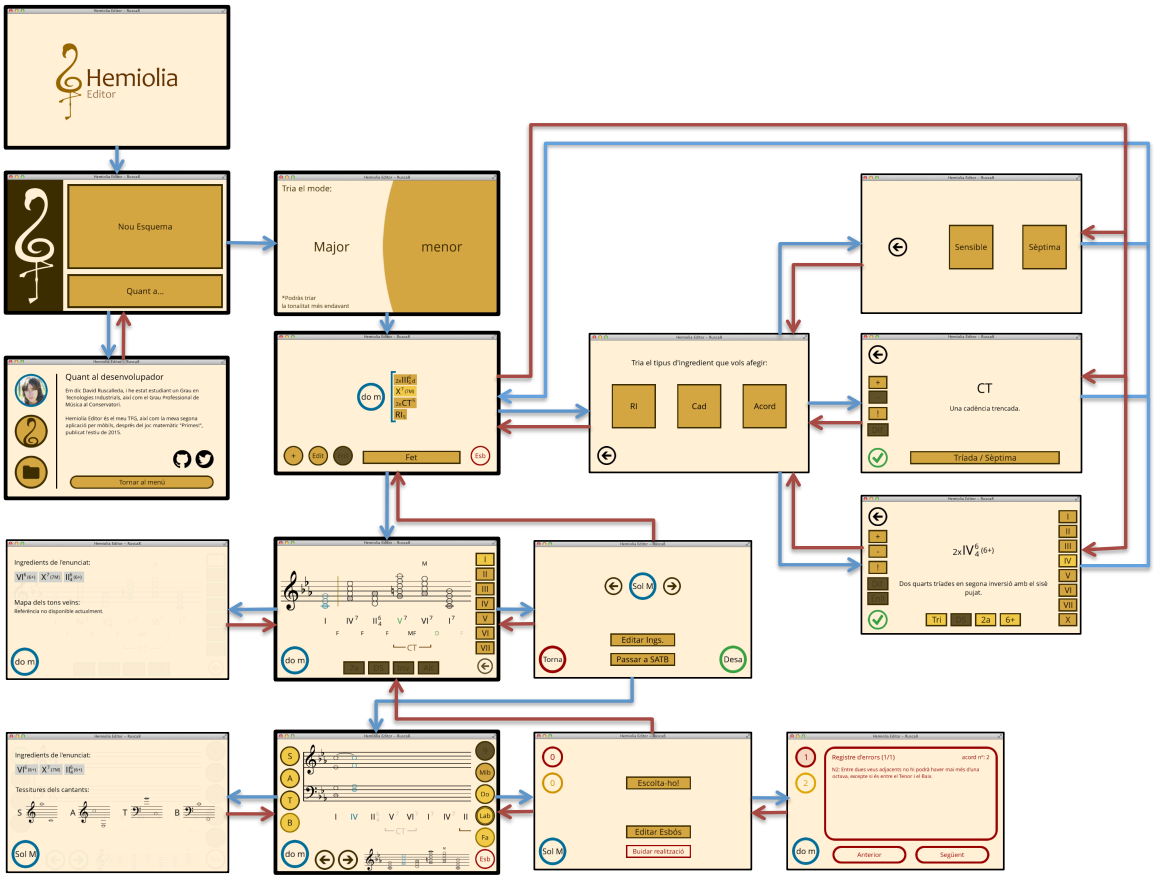

Figura 33: Diagrama de flux de l'aplicació.

A efectes pràctics, les biblioteques més rellevants dins el codi seran la **H** (que realitzarà totes les operacions que tinguin a veure amb les notes, acords i intervals), les variables de **S** (que apareixeran pertot a l'apartat dels gràfics i a l'execució de les correccions), i les funcions d'alinear de **F** (que serviran per omplir cert buit que deixa LibGDX en el terreny de l'alineament de textos), així com un seguit de funcions i variables també de **F** que ens serviran per mantenir càlculs previs i evitar redundàncies.

Finalment, tancant les unitats de compilació més importants, tindrem la unitat de **Recursos**, que és el lloc on s'ha desat tot el codi que té a veure amb elements multimèdia (i. e. imatges, fonts i sons), i per tant el lloc on els anirem a buscar tant per carregar-los a l'aplicació com per utilitzar-los des de les diferents pantalles.

I arribats a aquest punt, la resta d'unitats de compilació són petites definicions de classes o objectes que han anat essent necessaris a mesura que s'ha anat desenvolupant el codi. Hi podem trobar des de les definicions dels objectes "nota" (**Note**) i "acord" (**Chord**), fins els diferents tipus de botó que s'han utilitzat dins l'aplicació, passant pels diferents tipus d'ingredients que podem demanar a l'enunciat i per elements relacionats amb els registres d'errors i alertes.

[71] Inicial de Sinestèsia (associació de diversos sentits no necessàriament relacionats), en al·lusió a la relació que s'hi fa entre colors i coses que no necessàriament impliquen el color escollit.

Com es pot veure al llarg del codi (Annex A), per tal de separar els diferents apartats d'una manera clara, he utilitzat sovint els comentaris tipus **/** */** a mode de títol. D'altra banda, trec per pantalla moltes coses que en realitat a l'hora de fer servir l'aplicació no es veuen, però em resulta una manera molt senzilla de veure ràpid a quina part del codi em trobo (tant per la part de solucionar errors mirant la consola com pel que fa a navegar a través del codi quan l'estic escrivint).

8.2 Definicions dels elements musicals

Abans de començar a veure l'estructura de les principals unitats de compilació veurem com s'ha abordat el problema de desar informació sobre elements del món de la musica, per als quals no tindrem tipus de dada bàsic dins el llenguatge de programació.

En un principi, ens interessarà desar notes i acords, així com intervals. A més, haurem de decidir de quina manera volem desar les dades de l'esbós i la realització per tal que sigui pràctic posteriorment utilitzar-les, manipular-les i corregir-les.

8.2.1 Objecte de tipus nota (Note)

Com hem vist a l'apartat sobre música, per tal de definir completament una nota necessitem tres dades: el nom de nota, l'alteració, i l'índex acústic. Degut que no tots els salts entre un nom de nota i el següent tenen la mateixa distància en semitons, un enfocament que agafi el nom i l'alteració com a dades bàsiques acabarà esdevenint en un sistema innecessàriament complicat (p. ex. en alguns casos posar un sostingut equival a saltar al següent nom de nota, però en els altres casos no).

Per tal de solucionar aquest inconvenient que es presenta en utilitzar les alteracions com a dada, s'ha optat per substituir la dada de l'alteració per una referència al so de l'escala cromàtica que resulta en aplicar-la.

Així, una nota vindrà definida per un nom (Taula 17), una posició dins l'escala cromàtica (Taula 18), i si s'escau un índex acústic[72].

0	1	2	3	4	5	6
Do	Re	Mi	Fa	Sol	La	Si

Taula 17: Codi numèric que s'ha assignat a cadascun dels noms de nota.

D'aquesta manera, com que sabem quina posició ocupen els diferents noms de nota (quan no porten alteració) dins l'escala cromàtica (Taula 18, blau cel), podem deduir l'alteració d'una nota qualsevol comparant aquesta posició natural del seu nom amb la posició cromàtica real desada a la nota. Vegem-ho amb un exemple:

Si vull desar un Fa#, desaré el codi del nom de nota (3) i el codi de la posició que ocupa dins els 12 semitons (6), quedant com a resultat $[3,6]$.

0	1	2	3	4	5	6	7	8	9	10	11
Do	Do#	Re	Re#	Mi	Fa	Fa#	Sol	Sol#	La	La#	Si

Taula 18: Codi numèric assignat a cada posició de l'escala cromàtica i nota d'exemple per a cada posició.

[72] Al codi s'ha utilitzat en tots els casos l'índex acústic franco-belga (veure "Afinació estàndard i Índex Acústic", (pàg. 9).

Si després vull saber a quina nota correspon aquest codi, només he de mirar quin nom de nota tinc $(3 = Fa)$, i quina posició té aquest Fa dins l'escala cromàtica $(crom(Fa) = 5)$. Com que $6 - 5 = +1$, sé que l'alteració l'ha de pujar mig to, així que el Fa serà sostingut.

Finalment, seguint amb l'exemple, si volgués especificar que el Fa# és un Fa#4, l'únic que hauria de fer és afegir l'índex acústic com a dada extra, quedant-me [3,6,4].

Tot això, a efectes pràctics ho desaré en un objecte que he anomenat **Note**, que consta de tres variables tipus **byte** representatives d'aquestes tres dades: **nom**, **so** i **ia**, respectivament.

8.2.1.1 Constructors i mètodes

Dins l'objecte de tipus nota s'han implementat dos constructors diferents per tal de facilitar la creació de notes amb i sense especificar l'índex acústic (si no s'especifica se li assigna -1).

A més, s'ha definit el mètode **equals**, que és el mètode per defecte que s'encarrega de realitzar comparacions entre objectes (en aquest cas retorna "true" si els tres paràmetres són iguals en les dues notes), i s'ha creat un segon mètode similar que no té en compte l'índex acústic, anomenat **mateixaNota**, que ens serà útil quan l'octava de la nota no sigui important.

8.2.1.2 Representació dels intervals

Degut a la naturalesa dels intervals (al fet que són una diferència entre dues notes) no ens caldrà definir un objecte específic per a representar-los, sinó que podrem representar-los dins un objecte de tipus nota, sempre i quan n'entenguem els paràmetres de la manera adequada.

En aquest cas, el paràmetre **nom** representarà el salt numèric entre els noms de les dues notes (serà en valor absolut una unitat menys que el qualificatiu de l'interval[73]), el paràmetre **so** representarà la distància en semitons entre les dues notes de l'interval simple, i el paràmetre **ia** representarà quantes octaves extra té en realitat l'interval compost.

Així doncs, l'interval quedarà de la següent manera:

$$[(qualificatiu - 1), \quad distància\ simple, \quad octaves\ extra]$$

Per tal de treballar amb aquestes entitats d'una manera còmoda s'han creat un seguit de funcions d'utilitat que hi operen de manera transparent per al programador. Les veurem quan parlem de la biblioteca **H** (pàg. 65).

8.2.2 Objecte de tipus acord (Chord)

De la mateixa manera que amb les notes, s'ha definit un objecte específic per desar el conjunt de dades que representa un acord. En aquest cas, per tal de facilitar posteriors ampliacions de l'aplicació s'ha definit l'acord ja tenint en compte gran part dels diferents acords que podran anar apareixent en successives iteracions.

[73] No és directament el qualificatiu perquè segons la manera que tenim en música d'anomenar aquest tipus de salts, s'inclou en el recompte tant la nota inicial com la final (p. ex. de la nota 4 a la nota 5 hi ha un salt d'1, però en música ho anomenem una "segona"). Explicació més detallada a l'apartat 4.1.2. (pàg. 10).

D'altra banda, degut que l'aplicació sempre treballarà sobre una tonalitat en concret, els acords s'han definit sempre en relatiu a la tonalitat de l'aplicació, per la qual cosa no necessitarem desar la nota en concret a partir de la qual es generen (i. e. desarem només el parentiu que té l'acord amb la tonalitat en qüestió).

Tenint en compte aquestes dades, els primers paràmetres que voldrem desar seran el grau al qual pertany l'acord (un número representatiu dels números I-VII) i la inversió en la qual aquest es troba (si més no en cas que la vulguem deixar escollida en fer l'esbós). Aquests paràmetres s'han desat en variables tipus **byte** anomenades **grau** i **inv**.

D'altra banda, voldrem saber també si l'acord porta sèptima (variable **sept**), i en cas que l'acord sigui d'una tonalitat menor, voldrem saber si se n'ha alterat alguna nota[74] (variables **alt6** i **alt7**).

Chord									
grau	byte	sept	boolean	alt6	boolean	alt9	boolean	hom	boolean
inv	byte	dom	boolean	alt7	boolean	alt5	byte	nap	boolean

Taula 19: Resum de les variables escollides per representar un acord de l'esbós.

Aquestes variables, doncs, seran bàsicament les que necessitarem per gestionar la iteració actual de l'aplicació. Tot i això, tenint en compte el temari més avançat sabem que voldrem marcar també si l'acord és una dominant secundària (**dom**), si aquesta dominant secundària porta rebaixada la novena (**alt9**), si és de l'homònim menor (**hom**), si és una napolitana (**nap**) o si és un acord amb la quinta alterada (**alt5**), de manera que ja s'han afegit les variables corresponents a la definició de la classe.

Podem veure el resum d'aquestes variables escollides a la Taula 19.

8.2.2.1 Constructors i mètodes

Pel que fa als constructors, s'han definit tres maneres d'inicialitzar l'acord: indicant el grau, indicant el grau i si porta sèptima, i donant directament com argument un altre acord. D'aquestes, bàsicament la que s'ha utilitzat a l'interior del codi ha estat la que tan sols indica el grau (ja que l'usuari primer indica el grau i després si convé canvia els altres paràmetres). El constructor que permet indicar la sèptima, en canvi, ha estat definit més per una qüestió pragmàtica (era més còmode a l'hora d'entrar un esbós escollit ja a priori per provar altres coses), i finalment, el que permet entrar un acord sencer serveix en algunes parts del codi per copiar acords sencers (en lloc d'assignar una referència, que és el que passa amb els objectes si assignes i prou).

D'altra banda, s'ha creat un mètode (**addInv**) que s'encarrega de canviar la inversió de l'acord (puja una vegada la inversió, i en cas que hagi superat el màxim torna a l'estat fonamental). D'aquesta manera, el canvi d'inversió el gestiona el mateix acord, i evitem tenir codi innecessari dins les funcions de la pantalla.

8.2.3 Objecte de tipus cadència

Per tal de desar les cadències d'una manera pràctica s'ha creat també una classe específica, anomenada **Cadencia**.

[74] A les tonalitats menors tindrem alteracions opcionals al sisè grau i la sensible, provinents de l'escala melòdica (veure "Escales i modes", pàg. 14).

D'una cadència ens interessarà saber primerament en quina posició comença (el seu acord inicial), i en segon lloc quants acords dura (en general seran sempre 2, però podria expandir-se de diverses maneres, així que necessitem també la dada com a paràmetre per poder-ho tenir en compte).

A més, hi ha una manera especial de fer la cadència trencada que implica posar una dominant secundària[75] enmig, i ens pot interessar saber si s'ha fet servir aquesta tècnica especial, així que deixarem un booleà expressament per poder marcar-ho.

Finalment, voldrem poder marcar si es tracta d'una cadència trencada o si es tracta de la cadència perfecta del final, de manera que necessitarem també variables per diferenciar-ho.

Així doncs, les variables que s'han utilitzat per a l'objecte de tipus cadència són: acord on comença (**n**), quantitat d'acords que dura (**w**), marca per si és amb dominant secundària (**ds**), i marca per si és la cadència perfecta (**cp**).

8.2.4 Unitat bàsica de la realització (Desplegat)

De la mateixa manera que s'han definit les unitats de l'esbós (**Chord**), també ens caldrà emmagatzemar d'alguna manera la matriu de notes de la realització.

Degut que ja tindrem els acords definits a l'esbós, i que per tant sabem quines notes pertanyen a cada acord, no ens serà necessari emmagatzemar una matriu de notes, sinó que podrem desar simplement quina de les notes de l'acord canta cadascuna de les veus, i a quina alçada la canta.

L'objecte que s'encarrega de desar aquests valors s'ha anomenat **Desplegat**, i consta de dos vectors de mida 4, encarregats de recollir quina nota de l'acord (**1**, **3**, **5** o **7**) canta cadascuna de les quatre veus, i quin en quin índex acústic es troben aquestes notes. Aquests vectors s'han anomenat, respectivament, **satb**, i **ia**.

8.2.4.1 Constructors i mètodes

La classe **Desplegat** tindrà un sol constructor que simplement posarà a zero tots els valors dels dos vectors. Trobar un zero, doncs, ens indicarà a l'hora de representar i corregir la realització que encara no disposem de nota a la posició observada.

D'altra banda, per facilitar l'assignació de notes a les diferents posicions d'un **Desplegat**, s'ha creat un mètode anomenat **set**, que rep com a paràmetres la **veu** a qui volem assignar la nota, la **nota** que hi volem assignar i l'**índex** que hi volem assignar, i que realitza l'assignació de valors pertinent en resposta a aquests paràmetres.

8.3 Definicions dels ingredients de l'enunciat

Per tal de fer fàcil la gestió dels ingredients de l'enunciat, s'ha creat un seguit de classes que s'encarregaran de representar-los d'una manera compacta.

Tindrem bàsicament tres tipus d'ingredient (acord, cadència i resolució irregular), amb dues classes per a cada objecte: una que recollirà la informació de l'ingredient en sí, i una que ens ajudarà amb la

[75] Les dominants secundàries són els acords de dominant dels tons veïns (propers en nombre d'alteracions). A la iteració actual el programa podria gestionar-les, però no sabria corregir-les, així que s'han mantingut desactivades.

representació d'aquest i que farà també la funció de botó a l'hora de permetre a l'usuari manipular-lo dins la llista d'ingredients.

8.3.1 Característiques generals dels tipus d'ingredient

Totes les instàncies que es creïn de qualsevol dels ingredients tindran un número identificatiu únic[76], que desarem dins una variable anomenada **id**. Aquesta variable, a més, serà l'únic que aportarem al constructor en el moment de crear una instància de qualsevol dels tipus d'ingredient.

D'altra banda, s'ha redefinit per a tots els tipus d'ingredient el mètode **equals**, escollint que només tingui en compte aquest número identificatiu a l'hora de decidir si dos elements són el mateix, cosa que ens serà molt útil en el moment de cercar, editar i esborrar ingredients.

8.3.2 Definició dels ingredients tipus Acord (IngAcord)

En el moment d'escollir un acord com a ingredient per un esquema, voldrem poder escollir més paràmetres a part dels paràmetres que ja tenen els acords de l'esbós. Així doncs, un ingredient de tipus acord (**IngAcord**) constarà d'un objecte acord (**Chord**), i d'un seguit de variables afegides representatives de les diferents coses extra que puc voler demanar (a més de l'identificador de què hem parlat a l'apartat anterior).

La primera i més evident d'aquestes variables extra que podem voler escollir serà la quantitat de vegades que volem que es posi l'acord triat dins l'esquema. Aquest número el desarem en una variable anomenada **n**.

A més, voldrem poder escollir que l'acord sigui obligatòriament tríada, que porti obligatòriament l'alteració escollida, o que porti la sèptima major. Aquests paràmetres s'han desat a les variables **triada**, **alt**, i **septMajor**.

Cal esmentar en aquest punt que no tots els acords poden portar activats tots els paràmetres, i que sovint voldrem demanar un paràmetre en concret independentment de l'acord en què s'utilitzi. Per tal de permetre deixar el grau de l'acord a elecció de l'alumne (X), s'ha incorporat a la llista de graus possibles el valor 7 com a valor neutre[77]. Ho veurem amb més detall a l'explicació sobre la pantalla dels ingredients (Apartat 8.10).

IngAcord									
acord					Chord				
id	int	triada	boolean	septMajor	boolean	difInv	boolean	tcp	boolean
n	byte	alt	boolean	difAc	boolean	enll	boolean	eAmb	int

Taula 20: Resum de la definició de variables i objectes dels ingredients tipus acord.

D'altra banda, en el moment en què decidim posar més d'una vegada el mateix ingredient apareixen un seguit de noves indicacions que podríem voler demanar a qui resoldrà l'exercici: podem voler que les diferents aparicions de l'ingredient siguin sobre acords diferents (p. ex. dos acords amb sèptima diferents), podem voler que siguin en inversions diferents (p. ex. dos acords de IV grau en inversions diferents), podem voler que es posi l'ingredient tantes vegades com es pugui, i podem voler que les dues aparicions d'un ingredient que apareix dos cops estiguin una darrere l'altre (p. ex. dos acords

[76] Únic dins el conjunt de tots els tipus d'ingredient, no només dins el tipus en qüestió.
[77] Els graus I-VII vindran representats per números del 0 al 6, de manera que el 7 és el primer número lliure.

de sèptima enllaçats). Aquests paràmetres s'han representat a les variables **difAc** (diferent acord), **difInv** (diferent inversió), **enll** (enllaçats), i **tcp** (tants com puguis).

A més, preveient que en algun moment voldrem poder enllaçar ingredients de diferents tipus[78] (p. ex. una dominant secundària que vagi a cadència trencada), s'ha deixat ja definida una variable que recolliria l'identificador de l'ingredient amb què l'enllaçaríem, anomenada **eAmb** (enllaça amb).

Finalment, pel que fa al constructor, s'ha escollit assignar el valor 7 al grau de tots els ingredients tipus acord en el moment de crear-los, per tal que la opció per defecte sigui "qualsevol".

8.3.3 Definició dels ingredients tipus cadència (IngCadencia)

En aquest cas, la definició és molt més senzilla que en el cas que acabem de veure. Les cadències tindran també una quantitat (**n**), la possibilitat de demanar "tants com puguis" (**tcp**), i l'enllaç amb un altre ingredient (**eAmb**), però la resta d'atributs seran diferents (malgrat alguns conceptes siguin similars).

Així doncs, tindrem una variable encarregada de demanar que el segon acord de la cadència sigui amb sèptima (**sept**) i un encarregat de demanar que sigui obligatòriament tríada (**triada**). A més, tindrem l'equivalent a les variables que a l'objecte anterior ens indicaven acords diferents: **dif**. En aquest cas, la única diferència possible serà en el fet de ser tríada o amb sèptima, així que amb una variable per marcar-ho ja fem prou.

8.3.4 Definició dels ingredients tipus resolució irregular (IngRes)

Finalment, tenim la definició de les resolucions irregulars. En aquest cas, l'objecte és encara més senzill que els anteriors, ja que només tindrà l'identificador, i una marca que s'activarà si és una resolució irregular de la sensible (en oposició a la resolució irregular de la sèptima, que és el més habitual).

A més, com que aquests ingredients pràcticament no es demanaran mai (i seria molt estrany demanar-ne més d'un d'igual) no s'ha inclòs res que tingui a veure amb la quantitat a la definició de l'ingredient.

8.3.5 Definició dels botons associats als ingredients (BotIA, BotIC i BotIR)

Els botons associats als diferents ingredients seran tots similars, i bàsicament tindran un seguit de paràmetres relacionats amb les mides, i un seguit de funcions que calculen algunes d'aquestes mides en funció del contingut de l'ingredient. A més tindran algunes funcions i variables relacionades amb el text que el botó mostrarà. Vegem-ho en detall:

Les primeres variables que trobem en tots aquests botons seran les posicions **x** i **y** del botó, així com la seva amplada (**w**) i alçada (**h**), i un vector **d** encarregat d'emmagatzemar amplades parcials relacionades amb les diferents seccions que pugui tenir el text de l'ingredient.

D'altra banda, també tindrem un objecte del tipus d'ingredient que el botó representi, que servirà per recollir la referència a l'objecte associat a cada botó que es creï[79].

[78] També ingredients del mateix tipus però diferents (i que per tant no es poden gestionar amb l'atribut "enllaçats").

Una vegada acabades les declaracions d'objectes i variables tindrem un parell de constructors: un per quan realment volem crear un botó (assigna **x**, **y**, i un ingredient), i un per quan només volem crear quelcom amb què comparar altres botons per realitzar cerques, edicions i supressions (i que per tant només s'ocupa d'assignar l'**id** amb què ens interessa operar).

A continuació, trobarem la funció que calcula amplada i alçada (**updateWH**), que bàsicament mira quant ocupa el text que mostrarà el botó, i desa els càlculs parcials a **d** per a futures referències. En alguns dels botons, a més, desa parts d'aquest text per evitar repetir-ne els càlculs o el codi en altres punts de l'aplicació.

I per tancar aquesta secció sobre mides, trobem la funció **shrink**, que s'encarrega de reduir les mides originals de tot a unes mides de botó més petites, per tal de passar el botó de les mides que té quan l'usuari està escollint l'ingredient i l'ha de veure gran a les mides de quan mostro la llista amb tots els ingredients en petit.

Finalment, després de les funcions relacionades amb les mides, tenim una funció que calcula una versió plana del text del botó per tal que es pugui entendre llegida per la consola (per facilitar el *debugging*) i una segona funció que treu aquest text per pantalla (per poder demanar al botó que s'escrigui a sí mateix per consola). A més, trobarem l'habitual redefinició de la funció **equals**.

Cal esmentar, però, que bona part d'aquestes funcions no han estat implementades al botó de les resolucions irregulars degut que no han estat necessàries.

8.4 Definicions d'altres objectes d'utilitat

Juntament amb les definicions dels elements musicals i dels objectes relacionats amb els ingredients de l'enunciat, tindrem algunes definicions més, pertanyents a un àmbit més genèric. En aquest apartat veurem la definició del botó estàndard, i tres classes relacionades amb la correcció.

8.4.1 Definició del botó estàndard (BotQ)

De la mateixa manera com hem vist amb els botons dels ingredients, per tal de facilitar la gestió dels esdeveniments d'entrada, s'ha creat un botó genèric (**BotQ**) que tindrà les variables necessàries per emmagatzemar la posició i les mides d'un quadrilàter, així com un parell de variables relacionades amb el text o funció del botó.

Així doncs, en primer lloc tindrem les variables **x**, **y**, **w**, i **h**, que respondran als mateixos paràmetres que en el cas dels botons d'ingredients.

A més, però, tindrem una cadena de text (anomenada simplement "**text**"), que ens servirà per desar el text que s'haurà de mostrar dins el botó en cas que el botó ho requereixi.

D'altra banda, degut que alguns botons tindran una funció fortament relacionada amb un número, s'ha afegit una variable extra per poder desar aquest valor, anomenada "**grau**".

[79] Sempre tindrem un únic botó per a cada ingredient.

8.4.1.1 Constructors

Degut a l'àmplia utilització de la classe al llarg de l'aplicació, ha anat essent interessant crear diferents constructors per respondre a diferents necessitats. En general, els botons s'inicialitzaran donant una posició, unes dimensions i un text, però s'ha obert la possibilitat de deixar unes dimensions estàndard, de no especificar el text, o d'especificar un número en lloc del text (el número que es desarà a **grau**), entre d'altres.

8.4.2 Element de correcció puntual (CorPuntual)

Com veurem als apartats sobre les diferents pantalles, una part de les correccions es duu a terme sobre cadascun dels acords (hi ha correcció estigui com estigui), però en gran part dels casos preferirem desar només la correcció quan aquesta sigui rellevant (estalviar-nos desar tota l'estona que està bé).

Per tal de recollir aquest tipus de correcció s'ha creat l'element de correcció puntual (**CorPuntual**).

Un element de correcció puntual constarà d'un seguit de valors que ens indiquen on s'ha comès l'error (o alerta) i d'un caràcter que ens indica la correcció[80].

Pel que fa als valors sobre la ubicació, tindrem un primer valor que ens indicarà a quin acord correspon la correcció (**ac**), un segon valor que ens indicarà en quina veu s'ha comès (**veu**), i un valor auxiliar (**aux**) que habitualment servirà per marcar una segona veu (per quan l'error no és degut a una sola veu, sinó a la relació entre dues veus).

D'altra banda, el caràcter de correcció es desarà a la variable **cor**.

A l'hora de realitzar la correcció, doncs, tindrem un seguit de llistes (**Array**) formades per elements **CorPuntual**, cadascuna dedicada a un tipus de correcció. Ho veurem amb més detall a la pantalla de la realització (pàg. 86).

8.4.2.1 Constructors

Com en el cas dels botons, l'àmplia utilització dels elements **CorPuntual** fa que hagi estat útil crear constructors per a totes les circumstàncies. Així doncs, tindrem constructors que només necessiten un acord i una veu, constructors que no necessiten caràcter de correcció, i constructors que no utilitzen el valor auxiliar, i cadascun servirà per un seguit de tipus d'error concrets.

8.4.2.2 Gestió de lligadures i solapaments

Les característiques dels elements **CorPuntual** els converteix també en una molt bona manera de gestionar tant les lligadures com les notes solapades, ja que aquestes només succeiran en moments molt concrets, i en ubicacions que responen al mateix concepte de les variables esmentades anteriorment.

Així doncs, utilitzarem un **Array** d'elements **CorPuntual** també per desar les dades corresponents a aquestes situacions.

[80] Per més informació sobre els caràcters de la correcció veure apartat 8.8.3 (pàg. 82).

8.4.3 Definició dels objectes tipus registre (Log)

A més de desar els errors que es vagin trobant, ens interessarà també desar una informació que ens permeti mostrar a l'usuari la norma que justifica l'error que s'ha trobat. Per poder fer això necessitarem dues coses: una manera de desar la referència a les normes, i un lloc on es reculli la normativa. La primera d'aquestes accions la gestionarem a través dels objectes tipus registre, anomenats **Log**.

Cadascun d'aquests objectes disposarà simplement de dos números: un que ens indiqui a quin acord fa referència (com en el cas de les correccions), i un que ens indiqui de forma unívoca quina de les normes caldrà mostrar a l'usuari.

Aquestes variables s'han anomenat, respectivament, **nAc**, i **ref**.

En aquest cas, el constructor és senzill: rep els dos nombres i els assigna a les dues variables.

8.4.4 Definició dels objectes tipus norma (Norma)

Les referències que s'hauran desat a les llistes d'objectes tipus **Log** les farem servir per accedir a la llista de normes. Degut que les normes pot ser que variïn amb el temps (ja sigui perquè se n'afegeixin, perquè se'n suprimeixin o perquè es reordenin), s'ha escollit no referenciar les normes a partir de la seva posició dins la llista, sinó amb un número identificatiu únic.

Així doncs, cada objecte **Norma** tindrà associat un d'aquests números (**ref**) i una cadena que respondrà al text de la norma en qüestió (**text**).

El conjunt de totes les normes, com veurem més endavant, s'han desat a la biblioteca **N**.

8.4.4.1 Constructors i mètodes

Com en els casos dels ingredients, per facilitar certes funcions de cerca, s'ha afegit un constructor a més del constructor normal en què només cal indicar la referència (el constructor principal demana referència i text).

A més, pel mateix motiu i com en moltes de les altres classes, s'ha redefinit el mètode de comparació **equals**.

8.5 Biblioteques pròpies de l'aplicació

Un cop vistes les diferents definicions de les classes que necessitarem al llarg de l'aplicació, un segon apartat que ens interessarà veure són les diferents biblioteques que s'han anat programant per respondre a les diferents funcions que s'han anat necessitant al llarg del procés de programar.

Per tal de seguir un ordre una mica lògic, veurem les diferents biblioteques començant per les de més extensió i acabant per les més senzilles. Dins de cadascuna, comentarem en termes generals les funcions i variables més importants que s'hi inclouen.

8.5.1 Funcions relacionades amb la música (biblioteca H)

La més important de les biblioteques que s'han creat per a l'aplicació és la que recull les funcions encarregades de gestionar tot el que tingui a veure amb la música. Aquestes, apareixeran contínuament al llarg de tot el codi i són en essència el nucli que permet que funcioni tot.

En aquesta biblioteca, doncs, tindrem des de les variables que ens indiquen la tonalitat actual fins les funcions que calculen l'armadura, els intervals entre les diferents notes, o les notes que pertanyen als diferents acords. A continuació veurem una llista més detallada d'aquestes funcions i variables, ordenades segons els àmbits a què pertanyen.

8.5.1.1 Variables genèriques sobre conceptes musicals

El primer que trobarem dins la biblioteca **H** és un seguit de variables que tenen a veure amb els conceptes musicals que haurem d'utilitzar.

Entre aquestes, tindrem les paraules corresponents a les tonalitats (p. ex. "Major"), el text corresponent a tots els graus (p. ex. "I", "VII"), i els noms de totes les notes (p. ex. "Do", "Sol", "Si"), tant en majúscules com en minúscules (per facilitar la distinció entre Major i menor), així com resumides en una sola lletra (p. ex. "D", "S", "T"), entre d'altres.

A més, tindrem els noms de les diferents veus, així com un seguit de variables **Note** que recolliran els límits de les tessitures de cadascuna (p. ex. **tMinS**, **tMaxB**).

D'altra banda, també a la zona de les variables genèriques, tindrem la tonalitat del nostre esquema representada mitjançant un objecte tipus **Note** i un **byte**, que emmagatzemaran, respectivament, la nota corresponent a la tonalitat, i el mode en què ens trobem. A més, tindrem un segon grup com aquest pensat per gestionar certs canvis en l'armadura que es puguin voler fer (i desfer).

Finalment, trobarem una variable numèrica encarregada de desar el codi de l'armadura (un valor absolut que indica quina quantitat de sostinguts o bemolls té, i un signe que distingeix entre les dues) (p. ex. cinc sostinguts tindrà per codi 5, mentre que quatre bemolls tindrà per codi -4).

Aquesta, s'ha anomenat simplement **armadura**.

8.5.1.2 Enllaços i xifrat

Una vegada deixem l'apartat de les variables genèriques, trobem un parell de funcions dedicades a calcular els enllaços (una que retorna un codi numèric identificatiu del tipus d'enllaç, i una que retorna un text que el representa).

Tipus d'enllaç	Fort	Dèbil	Molt fort	Inexistent
Codi numèric	1	2	3	0
Text associat	F	D	MF	

Taula 21: Codificació dels diferents tipus d'enllaç i text associat a cadascun d'ells.

La funció que retorna el codi numèric s'ha anomenat **linkNum**, i necessita com a paràmetres dos acords. D'altra banda, la funció que retorna el text (**link**), simplement s'ocupa de realitzar la traducció d'aquests números per retornar el text associat. Podem veure aquests codis i textos recollits a la Taula 21.

Cal esmentar en aquest punt que, malgrat no s'ha comentat a l'apartat sobre música, podem trobar-nos també que no es faci cap tipus d'enllaç (que es repeteixi el mateix grau en una configuració diferent). Aquesta situació, doncs, també s'ha hagut de tenir en compte (es desarà amb el valor 0 i retornarà una cadena de text buida).

D'altra banda, dins aquest apartat tindrem també la funció que calcula els xifrats corresponents a un acord donat (en funció de si porta sèptima o no i de quina inversió se li ha donat). Per tal de facilitar l'accés individual a cadascun dels dos números que formen el xifrat, la funció (anomenada **xifrat**) s'ha dividit en dues seccions a les quals accedirem mitjançant la utilització d'un argument booleà anomenat **baix**: si activem **baix**, la funció ens retornarà el nombre inferior del xifrat, mentre que mantenir-lo desactivat ens retornarà el nombre superior.

Finalment, tancant aquest apartat tenim una funció que s'encarrega de retornar el text extra que cal escriure en alguns acords al costat del grau (p. ex. quan el cinquè d'un mode menor no porta sensible s'escriu $\mathrm{V_m}$ en lloc de V). Aquesta funció s'ha anomenat **textGrau**.

D'altra banda, cal comentar que al llarg del codi sovint s'ha aprofitat el polimorfisme que permet *Java* per tal de dotar a les funcions de jocs d'arguments alternatius. En el cas de les funcions que acabem de veure, per exemple, podem veure que existeixen dues construccions alternatives de la funció **link**: una que demana només el codi numèric, i una que demana els dos acords que formen l'enllaç (i que per tant s'encarrega ella mateixa d'obtenir el codi que necessita). Aquest tipus d'enfocament serà quelcom recurrent dins aquesta biblioteca, ja que moltes de les funcions seran necessàries en contextos diferents, i ens facilitarà la feina poder treballar directament amb els arguments que tinguem disponibles.

8.5.1.3 Gestió de les alteracions

Pel que fa a la gestió d'alteracions trobarem dins la biblioteca bàsicament tres funcions diferenciades: una que calcula l'armadura d'una tonalitat, una que calcula l'alteració que porta una nota, i una que ens indica si cal indicar expressament aquesta alteració o si per contra ja la tenim indicada a l'armadura.

La primera d'aquestes funcions, doncs (anomenada **armadura**) s'encarregarà de rebre una tonalitat (nota + mode), i de retornar un codi numèric representatiu de les alteracions que porta el to[81].

A més, juntament amb aquesta funció trobarem una funció extra que ens retornaria l'equivalent en text d'aquest codi (anomenada **textArm**) (p. ex. si armadura retorna el valor 5, *textArm* retornarà 5#).

Un cop acaben aquestes funcions, trobem a la biblioteca la funció **getAltNum**, que rebrà una nota, i s'encarregarà de comparar el seu valor **so** amb el valor cromàtic equivalent al seu valor **nom**, de la manera com s'ha descrit anteriorment a l'explicació sobre l'objecte de tipus nota[82].

Com en el cas anterior, una segona funció ens traduirà aquest valor a una cadena de text per als casos en què vulguem escriure l'alteració. Aquesta funció s'ha anomenat **getAlt**.

I finalment, tancant la secció sobre alteracions trobarem la funció **accidental**. Aquesta funció compararà l'alteració que porta una nota amb l'alteració que correspon a aquest nom de nota segons l'armadura, i en cas que siguin diferents retornarà **true**. D'aquesta manera, a l'hora de renderitzar les notes podrem saber amb facilitat quines d'elles necessiten que se'ls dibuixi l'alteració que porten, i quines no ho necessiten perquè l'alteració que porten ja és a l'armadura.

[81] Codificació equivalent a l'esmentada dins la secció 8.5.1.1 en parlar de la variable **armadura**.
[82] Apartat 8.2.1, pàg. 57.

8.5.1.4 Gestió d'intervals

Pel que fa a les funcions que treballen amb els intervals, tindrem principalment tres grups: un que s'encarrega dels càlculs relacionats amb els intervals, un que extreu informació d'intervals ja existents, i un que s'encarrega de modificar aquests intervals ja existents per llegir-los de formes diferents (formes que facin més fàcil l'aplicació de certes correccions).

Abans de tot, però, tindrem una parella de vectors (**qMaj** i **qMin**) que emmagatzemen respectivament les distàncies naturals més llargues i més curtes per a cadascun dels noms d'interval (i.e. de primera a sèptima, set valors representatius de les distàncies en semitons). A més, degut que les distàncies fins a les notes d'una tonalitat menor no responen a cap d'aquestes dues llistes, tindrem un tercer vector per recollir-les (**dMin**).

Podem veure el contingut d'aquests vectors a la Taula 22.

	Primera	Segona	Tercera	Quarta	Quinta	Sexta	Sèptima
qMaj	0	2	4	5	7	9	11
qMin	0	1	3	5	7	8	10
dMin	0	2	3	5	7	8	10

Taula 22: Contingut dels vectors relacionats amb les distàncies de cada nom d'interval.

Aquests tres vectors són la manera com calcularem pràcticament tot el que tingui a veure amb les notes que té una tonalitat, les notes que té un acord, o l'alteració que porta per defecte una nota dins la tonalitat en qüestió (p. ex. la funció *accidental* esmentada a l'apartat anterior en el fons treballa comparant les notes que li arriben amb aquestes llistes).

Una vegada deixem enrere aquestes llistes, el següent que trobarem serà la funció que ens permet calcular quin és l'interval que hi ha entre dues notes (**getInterval**).

Si bé en un principi semblaria que aquesta funció hauria de ser quelcom trivial, un primer intent d'implementar-la ens farà veure que qualsevol situació en què els noms de nota o de so travessin els llindars de la llista implicarà haver d'ajustar el resultat. Així doncs, situacions com el Do bemoll (representat com [0,11] en lloc del més intuïtiu [0,-1]), o com les que resulten de calcular intervals envoltant un salt d'índex acústic, s'han hagut de tenir en compte.

En tot cas, una vegada arribem al resultat de la funció, el nostre interval ha estat ajustat per tal de respondre a la definició d'interval esmentada dins la descripció dels objectes tipus nota[83].

Seguint amb les funcions disponibles dins aquest apartat, trobarem una funció que ens servirà per obtenir una nota resultat d'afegir un interval concret a una nota d'origen (p. ex. donat un Do, afegim un interval de 4J, i obtenim un Sol). Aquesta funció s'ha anomenat **addInterval**, i haurà d'ajustar també el seu resultat en resposta als problemes esmentats anteriorment.

I, finalment, tindrem les funcions encarregades d'analitzar i modificar els intervals.

Pel que fa a les modificacions, tindrem tres funcions encarregades respectivament de passar un interval donat al seu equivalent ascendent (p. ex. donada una 2M descendent, retorna una 7m ascendent), passar un interval donat al seu equivalent descendent (cas invers de l'anterior), i

[83] Dins el subapartat 8.2.1.2 sobre la representació dels intervals (pàg. 58).

retornar el nom de l'interval compost equivalent a un interval donat (p. ex. donada una segona amb una octava extra, retorna que és una novena). Aquestes funcions s'han anomenat, respectivament **ascendent**, **descendent**, i **compost**.

D'altra banda, pel que fa a l'anàlisi, tindrem una funció que comprova si l'interval és "menor, Major o just" (anomenada **mMj**), una que comprova si és augmentat (anomenada **augmentat**), i una que comprova si és disminuït (anomenada **disminuit**), totes amb resposta booleana.

8.5.1.5 Obtenció de notes i gestió d'acords

El següent subapartat que veurem dins la biblioteca de funcions musicals és el que va dedicat a calcular les notes que tenen els acords, així com a treballar amb les alteracions que aquests acords poden portar.

En una primera instància, trobarem la funció **getNote**, que és la que s'encarrega de dir-nos quina nota correspon a un grau en concret d'una tonalitat en concret (gran part de les funcions que veurem a continuació utilitzen aquesta funció com a part dels seus càlculs interns).

En segon lloc, tindrem una funció que fa un càlcul semblant, però que ja treballa amb les notes que tenen els acords. Aquesta funció s'ha anomenat **getChordNote**, i demana un acord (**Chord**), una posició de nota dins l'acord (**1**, **3**, **5**, **7** o **9**), i una tonalitat (p. ex. volem saber quina nota és la quinta de l'acord de IV grau de Do m). Una vegada se li proporcionen aquests paràmetres, la funció retorna una nota corresponent a la nota que tindrà aquell acord, en aquella tonalitat, en aquella posició.

Aquesta funció, doncs, serà el lloc on anirem a buscar la informació respecte quines notes hem de dibuixar en cadascun dels acords del nostre esquema.

Cal apuntar, a més, que aquesta funció està ja preparada per gestionar acords en dominant secundària, així com acords de l'homònim menor, de manera que la implementació futura d'aquests nous tipus d'acord pràcticament només implicarà programar-ne les normes de correcció.

D'altra banda, com haurem vist al codi de les funcions que hem anat detallant fins ara, sovint s'ofereix mitjançant el polimorfisme anteriorment esmentat una construcció alternativa que no requereix indicar la tonalitat. En aquests casos funció utilitzarà per al càlcul la tonalitat per defecte indicada a les variables sobre la tonalitat que s'han comentat al principi d'aquesta biblioteca.

A continuació després de la funció **getChordNote**, trobarem una funció que ens permet introduir un acord i rebre'n una versió modificada en funció de l'alteració que ens interessi (i. e. alt6, alt7 o alt9). Aquesta funció s'ha anomenat **altChord**, i ens serà molt útil en el moment de mostrar a l'usuari quin és el resultat d'aplicar l'alteració opcional dels acords que en permeten (i. e. quines notes té disponibles per triar donat un acord de l'esquema).

I finalment, tancant aquest apartat sobre acords i alteracions, tindrem una funció que ens calcula, donat un acord, quina de les seves notes es pot alterar (si és que se'n pot alterar alguna), i a quin dels tipus d'alteració disponible correspon aquesta acció (p. ex. la tercera de l'acord es pot alterar segons la variable alt6).

Aquesta funció s'ha anomenat **opcionals**, i retorna com a resposta un **byte** de dues xifres en què les desenes representen la nota de l'acord que es pot alterar (**1**, **3**, **5**, **7** o **9**), i les unitats el tipus

d'alteració que s'hi pot fer (**6**, **7** o **9**). Així doncs, si per exemple la quinta es pot alterar segons **alt7**, la funció retornarà 57, mentre que si es pot alterar la sèptima segons **alt6**, la resposta serà 76.

En cas que no es pugui alterar cap nota, la funció retornarà un 0. D'altra banda, a efectes pràctics no s'ha inclòs l'especificació respecte a quina nota s'altera amb **alt9** (es retorna simplement 9), ja per definició la nota alterada sempre serà la novena, de manera que el codi sempre seria 99.

Com a afegit en l'àmbit d'aquesta funció, a més, tindrem una funció més petita que el que farà serà retornar-nos un text curt identificatiu de l'estat de l'alteració en l'acord en qüestió (p. ex. si l'acord pot alterar la 6a (**alt6**) i la té alterada, retornarà "6+", mentre que si pot però no la hi té, retornarà "6=").

Aquesta última funció s'ha anomenat **altOpText**, i ens serà útil per a mostrar a l'usuari indicacions sobre l'alteració escollida en alguns dels botons de la interfície.

8.5.1.6 Gestió de les tessitures

En aquest apartat trobarem un seguit de funcions dedicades a tractar amb les tessitures dels cantants, tant pel que fa a la correcció de la realització com pel que fa a escollir l'índex acústic de les notes dins l'esquema.

La primera funció que trobarem (**defaultIA**), serà doncs la que s'encarrega d'escollir un índex acústic per defecte per a les noves notes que s'incorporen a la realització. Aquesta, comprova el nom de la nota que ha entrat l'usuari (previ càlcul en funció de la tonalitat actual i el grau de l'acord), i la veu a la qual l'ha afegit, i col·loca aquesta nota en una octava que correspongui a un rang intermedi dins la tessitura del cantant.

En segon lloc, degut que voldrem permetre a l'usuari canviar aquesta octava dins els marges permesos per la tessitura estàndard, tindrem una funció que s'encarrega de gestionar aquest canvi, anomenada **canviaOctava**.

El comportament de la funció en aquest cas és senzill: s'incrementa l'octava de la nota en una unitat, i si la nota queda massa fora de la tessitura estàndard, es redueix l'octava en tres. Finalment, degut que dins de cada veu algunes notes només apareixen en dues octaves diferents, es comprova que no hagi quedat la nota massa per sota el límit inferior de la tessitura, i en cas que hagi passat això, es corregeix incrementant l'octava fins que la nota queda dins.

Finalment, per tal de fer fàcil comprovar si una nota és dins d'una tessitura o no (i com de lluny de la tessitura es troba en cas que surti dels límits), s'ha creat una última funció anomenada **dinsTessitura**. Aquesta, retorna 0 en cas que la nota donada es trobi dins la tessitura de la veu escollida, i en cas contrari retorna la quantitat de noms de nota que hi ha des del límit fins a la nota en qüestió.

A més, per tal de simplificar la funció **dinsTessitura** (per treure les veus com a factor comú, per dir-ho d'alguna manera) s'ha creat una petita funció que retorna el límit mínim o màxim (segons es vulgui) de la veu que se li demana. D'aquesta manera, el càlcul de **dinsTessitura** es realitza de la mateixa manera sigui quina sigui la veu, i és aquesta funció extra (anomenada **tLimit**) la que s'encarrega d'accedir a les dades específiques de cada veu en cada cas.

8.5.1.7 Posicionament dels elements musicals

Finalment, en un últim apartat dins la biblioteca **H**, trobem les funcions encarregades de posicionar els elements musicals a la pantalla: des de les posicions de cada nota dins el pentagrama fins la posició de cadascun dels símbols d'alteració de l'armadura, entre d'altres.

La primera funció que trobarem dins l'apartat és l'encarregada d'escollir la posició de les notes que formen els acords de l'esbós. Aquesta, s'ha anomenat **nStackPos**, i bàsicament comprova que cada nova nota que es posa a la torre de notes que forma l'acord, es posi a la part superior d'aquesta (i. e. en gestiona l'índex acústic i tradueix la nota a una distància respecte al pentagrama[84]).

Degut que per a la primera nota no existirà comprovació possible, disposarem d'una funció més petita dins **nStackPos** que escull la posició per defecte d'una nota dins l'esbós (i. e. l'índex acústic que tindrà si **nStackPos** no l'hi canvia, en forma de distància respecte al pentagrama). Aquesta funció s'ha anomenat **nPos**, i bàsicament ubica els dibuixos de les notes respecte al pentagrama de manera ascendent començant pel Do_3.

D'altra banda, com que no sempre voldrem col·locar les notes en forma de torres (a la realització les voldrem posar on l'usuari hagi escollit que les vol), necessitarem una funció equivalent a aquestes però que directament pugui ubicar qualsevol nota (**Note**) que se li entri en relatiu a qualsevol dels dos pentagrames del sistema[85]. Aquesta funció és la que s'ha anomenat **notePos**.

A continuació, després de les funcions per ubicar les notes en vertical dins el pentagrama, tindrem una altra funció (**ajustSolapades**) que s'encarregarà d'ajustar el posicionament horitzontal quan convingui: s'activarà en el moment de dibuixar cada nota i comprovarà si hi ha algun solapament registrat a nom de la nota en qüestió. En cas que hi sigui, retornarà un nombre[86] que incrementarà el valor **x** del dibuix en la mesura necessària per evitar el solapament, i en cas que no hi sigui retornarà zero.

Seguidament, trobarem una funció (**ySolFa**) que senzillament retorna l'alçada de la posició vertical zero de les notes de la veu que s'introdueixi com a paràmetre (i. e. l'alçada del Fa_3 en clau de sol per a les veus de noia, i l'alçada del La_1 en clau de fa per a les veus de noi). Aquesta funció ens serà útil també a mode de "factor comú", per poder simplificar el codi de les funcions de dibuix.

D'altra banda, tancant la biblioteca tindrem dues funcions més, anomenades **sobreLinia** i **armPos**, que s'ocuparan respectivament de comprovar si la nota introduïda com a paràmetre és en una línia o en un espai, i d'ubicar els símbols de l'armadura a les alçades adequades.

La primera, **sobreLinia**, retornarà 1 si la nota és sobre una línia, i 0 si és en un espai. Això ens servirà per ajustar les posicions de certs elements gràfics de la correcció que en cas contrari quedarien solapats amb les línies del pentagrama i per tant poc visibles.

[84] A l'aplicació, tots els dibuixos de notes estan ubicats en relatiu a la posició de la primera línia del pentagrama (la de més a baix). Degut que les notes les dibuixaré des del vèrtex (no des del centre) i que LibGDX té el vèrtex de les imatges a la cantonada inferior esquerra, aquesta ubicació 0 serà la de la nota del primer espai.
[85] A la realització tindrem un pentagrama en clau de sol i un pentagrama en clau de fa.
[86] Un tant per u que a l'hora de realitzar el desplaçament anirà multiplicat per l'amplada de la nota.

D'altra banda, la segona funcionarà d'una manera molt similar a com treballen les primeres funcions que hem comentat en aquest apartat de la biblioteca, amb la diferència que haurà de distingir entre les armadures que porten sostinguts i les armadures que porten bemolls (degut que per a algunes de les notes cadascun dels símbols s'ubica en una octava diferent).

8.5.2 Funcions genèriques i relacionades amb el text (biblioteca F)

Si bé la biblioteca **H** és molt àmplia i conté moltes funcions diferents, la resta de les biblioteques que veurem a partir d'ara seran molt més senzilles, i sovint estaran enfocades a contextos molt més concrets. En aquest cas, les funcions que veurem tindran a veure amb la persistència de certes variables, amb l'alineament de les línies de text, i amb el tractament del llenguatge.

8.5.2.1 Funcions de persistència

En aquest apartat podrem veure en primer lloc un seguit de variables que serviran per emmagatzemar certs valors que preferirem no haver de calcular diverses vegades en cas que els necessitem de manera successiva.

La majoria d'aquests valors es desaran a través d'un seguit de funcions anomenades **keep**, que l'únic que faran serà desar el valor d'entrada a la variable corresponent i a continuació retornar-lo. D'aquesta manera, si vull desar el resultat de certa funció **func** per poder-lo aprofitar al llarg d'un futur proper, l'únic que he de fer és envoltar aquesta funció de la funció **keep**, quedant-me **keep**(**func**).

Aquesta funció ens serà molt útil especialment en combinació amb les funcions d'alineament, ja que en el renderitzat de text haurem de proveir amb el text desitjat tant la funció de dibuixar com les funcions d'alinear, i aquestes tres aparicions del mateix text tindran lloc en una mateixa línia de codi. Aprofitant el **keep**, doncs, podrem escriure la cadena de text només la primera vegada, i accedir a la variable que s'encarrega de la seva persistència les altres dues vegades.

D'una manera similar, un altre moment en què ens seran útils aquestes funcions és a l'hora de dibuixar les notes, ja que ens permetran estalviar-nos calcular la nota que hem de dibuixar cada vegada que necessitem accedir a algun dels seus valors (la calcularem només la primera vegada, i la resta de les vegades accedirem al valor desat).

8.5.2.2 Funcions d'alineament del text

El segon apartat dins aquesta biblioteca serà el dedicat a l'alineament del text.

LibGDX disposa de certes funcions per alinear, però en general són més complexes del que ens cal, i demanen una gran quantitat de paràmetres que la major part de les vegades no ens interessen ni volem haver de calcular. A més, alinear les coses nosaltres mateixos ens permetrà conèixer amb facilitat les mides de les cadenes de text i per tant ubicar fàcilment unes cadenes en relació a unes altres (p. ex. el xifrat d'un acord en funció de l'amplada del text del grau).

Així doncs, per tal de gestionar d'una manera senzilla l'alineament de textos s'han creat dues funcions diferents (una per l'alineament vertical i una per l'horitzontal), que són capaces d'agafar directament el text desat a les variables de persistència, fent que sovint l'únic que calgui enviar a les funcions com a paràmetre sigui quina de les fonts estem fent servir.

Aquestes funcions (**align** pel cas horitzontal i **vAlign** pel vertical) es basen en un objecte de LibGDX anomenat **GliphLayout**, que bàsicament representa una caixa de la mida d'una cadena de text donada (tenint en compte els possibles salts de línia). Així, les funcions l'únic que fan és omplir aquesta caixa amb el text especificat tenint en compte que és de la mida de font especificada, i posteriorment mesurar-ne l'amplada o l'alçada, retornant-la completa si es vol ajustar al límit contrari, o dividida a la meitat en cas que es vulgui centrar el text.

Per tal de poder escollir quina de les dues opcions ens retorna, s'ha implementat un argument de tipus caràcter (**char**) que serà **r** si volem el text a la dreta, **c** si el volem centrat, i **d** si el volem alineat a baix.

Cal destacar que, al contrari del que passa amb les imatges, a LibGDX el text té el punt de referència a la cantonada superior esquerra (en lloc d'inferior esquerra), de manera que l'alineament per defecte serà amunt-esquerra.

Així doncs, la manera com utilitzarem aquestes funcions serà sumant-ne el resultat a les posicions **x** i **y** que es demanin dins la funció de dibuix, quedant aquestes finalment com (**x** + **align**) i (**y** + **vAlign**), respectivament.

D'altra banda, com a funció extra dins aquest apartat trobarem **gAlign**, que tindrà com a únic propòsit alinear el conjunt del grau amb el seu text extra (p. ex. dins el text V_m, cada lletra és d'una mida diferent, així que no podrem alinear el conjunt amb les funcions habituals).

8.5.2.3 Funcions per al tractament del llenguatge

Finalment, tancant la biblioteca tindrem un seguit de funcions que ens simplificaran el tractament del llenguatge.

Abans, però, disposarem d'un vector amb els noms dels números ("Un"..."Deu"), i una funció que s'assegura de retornar el número en xifres en cas que el nom del número no estigui disponible (i. e. a partir d'onze).

Aquest conjunt de funció i vector serà el que utilitzarem a la pantalla dels ingredients per generar la part de la quantitat dins la descripció escrita de cada ingredient[87].

Seguidament, i per motius similars, tindrem dues funcions anomenades **plurals** i **fem**, encarregades, respectivament, de retornar el plural de la paraula entrada en cas que la quantitat sigui més gran que 1, i retornar la versió femenina de la paraula entrada.

Evidentment, aquestes funcions només estaran implementades per les paraules que s'han anat utilitzant al llarg del programa, de manera que qualsevol paraula nova que es volgués utilitzar necessitaria prèviament ser afegida a aquestes. Tot i això, s'ha deixat la norma d'afegir una "s" a la paraula com a opció per defecte de la funció **plurals**, fent que la majoria de paraules estiguin en realitat contemplades.

[87] Veure pantalla dels ingredients (Figura 54, pàg. 98).

En tot cas, aquestes funcions ens serviran bàsicament per canviar a plural la descripció dels ingredients en cas que se'n demani més d'un, i per traduir els números que tenen traducció (i. e. "Un" i "Dos") en cas que l'ingredient sigui una cadència en lloc d'un acord, respectivament.

Després de **plurals** i **fem** trobarem una funció que bàsicament s'encarrega de traduir els valors booleans **true** i **false** a les paraules **sí** i **no**. Aquesta funció, però, de moment només s'ha utilitzat sobre text extret per consola, de manera que ara per ara no té implicació real pel que fa a l'usuari.

Finalment, tancant la llibreria trobarem una funció inspirada en el tipus de funció **IF** que utilitza *mySQL*, utilitzada a l'aplicació per poder aplicar condicionals directament sobre les cadenes de text.

Com en el cas de *mySQL*, la funció rep tres paràmetres: una condició, un valor a retornar en cas que sigui certa, i un valor a retornar en cas que no ho sigui. D'aquesta manera podrem inserir fàcilment dins les cadenes de text fragments que només apareixeran si certes condicions són satisfetes.

Degut que sovint no voldrem escriure res en el cas que la condició sigui falsa, s'ha construït també una versió alternativa de la funció en què aquest paràmetre no s'ha d'especificar (i. e. serà per defecte una cadena de text buida).

8.5.3 Normativa d'harmonia i funcions associades (biblioteca N)

La classe **N** bàsicament s'encarregarà de gestionar la informació referent a la normativa d'harmonia que hem anat veient a les primeres parts d'aquest document[88].

Bàsicament, tindrem un **Array** d'elements tipus **Norma**, que recollirà dividit en trossos tot el text de la normativa que hem vist que sigui rellevant per a l'aplicació. Degut a la naturalesa de les llistes LibGDX, els elements que les formen s'han d'afegir a posteriori, de manera que aquesta acció es durà a terme en un mètode que s'ha anomenat **carrega** i que cridarem en el moment de carregar la resta de recursos[89] (p. ex. les imatges i les fonts).

D'altra banda, s'ha inclòs a la classe un parell de mètodes dedicats respectivament a mostrar el text de les altres normes que pugui necessitar com a context la norma que volem veure, i accedir d'una manera senzilla al text corresponent a un número de referència.

Aquests mètodes s'han anomenat respectivament **ambContext** i **textDe**, i ambdós demanen un número de referència com a argument.

8.5.4 Funcions relacionades amb el tractament del color (biblioteca S)

El cas de la biblioteca **S** és similar al de **N** degut que bona part del seu contingut són variables, i només hi ha un parell de funcions dedicades a la gestió d'aquestes.

En aquest cas, tindrem bàsicament variables de tipus **Color**, que venen representades per quatre números en tant per u indicatius dels paràmetres **r**, **g**, **b**, i **alpha**.

[88] Veure 4.2 Normes i conceptes teòrics d'harmonia (pàg. 23).
[89] Podem veure com s'ha fet el tractament de la resta de recursos a l'apartat sobre la unitat de compilació que els gestiona (8.6, pàg. 75).

Per tal de deixar els colors de l'aplicació parametritzats de manera que sigui fàcil canviar-los d'una manera global, s'han inclòs dins aquesta llibreria variables amb el color del fons, de la lletra, dels errors, de les alertes, i de totes les parts i estats possibles dels botons, entre d'altres.

A més, s'ha afegit a continuació una funció (**cColor**) que rep un caràcter de correcció i en retorna el color associat, cosa que ens serà molt útil a l'hora de renderitzar els elements gràfics corresponents a les correccions.

D'altra banda, s'ha afegit també una funció que ens permetrà de manera fàcil obtenir un color nou a partir de canviar la transparència d'un color donat, de manera que ens resulti senzill pintar amb els mateixos colors que tenim a les variables, però d'una manera que quedi semitransparent.

Aquesta última funció s'ha anomenat **newAlpha**.

8.5.5 Funcions relacionades amb la gestió de les animacions (biblioteca A)

Finalment, tancant les biblioteques pròpies de l'aplicació trobarem la biblioteca que s'encarregarà de gestionar les animacions que hi pugui haver a l'aplicació.

Aquesta, bàsicament consta d'un seguit de variables representatives dels diferents paràmetres que poden prendre les diferents animacions de què es farà càrrec, així com d'un mètode **update** que el que farà és actualitzar l'estat d'aquestes variables en funció del valor **delta** que se li faci arribar.

Com veurem en parlar de la funció **render** de les pantalles LibGDX, aquest valor **delta** en general serà en segons la quantitat de temps succeïda des de l'anterior fotograma. Gràcies a aquest valor, podrem actualitzar més o menys les animacions en funció de si ha avançat més o menys el temps, mantenint les velocitats d'aquestes a un ritme coherent amb la realitat.

D'altra banda, cal destacar d'entre les variables contingudes en aquesta biblioteca la que porta per nom **tHold**, ja que portarà a dins el valor numèric que decidirà a partir de quin temps considerem que s'ha mantingut premut un botó (i per tant com de llarg podem prémer sense que s'accioni la resposta alternativa en els botons que admeten els dos tipus d'interacció).

8.6 Gestió dels recursos

Com hem comentat en fer el resum de l'estructura (Apartat 8.1), per tal de mantenir separat de la resta tot el que tingui a veure amb els recursos, s'ha creat una unitat de compilació a part exclusivament dedicada a gestionar-los. Aquesta unitat s'ha anomenat, simplement, **Recursos**.

Dins la unitat **Recursos** tindrem en primer lloc una extensa declaració de variables recollint tot el que puguem necessitar per tal de desar i utilitzar els diferents recursos que ens puguin anar fent falta durant l'execució de l'aplicació. Aquesta part de la biblioteca vindrà separada en tres clares seccions: imatges, fonts, i finalment àudio.

A continuació, trobarem un seguit de funcions que seran les encarregades de carregar les fonts i imatges i deixar-les preparades per a la seva posterior utilització en l'aplicació: **carrega**, per les imatges que es fan servir al llarg de tota l'aplicació, **preCarrega**, per la imatge que necessitem només a la pantalla inicial, i **carregaAbout**, per a les imatges que només utilitzarem a la secció "Quant a...".

El fet de mantenir aquests recursos en funcions separades ens permetrà carregar individualment les imatges que només es fan servir en moments puntuals, de manera que un cop es tanqui la finestra que les necessita podrem alliberar-ne els recursos mitjançant la funció **dispose**.

Finalment, després de les funcions per a la càrrega d'imatges, trobarem un parell de funcions relacionades amb els sons: una per carregar els sons que es necessitin per a reproduir un esquema donat i una per reproduir els sons corresponents a un vector de quatre notes (i. e. un acord de la realització a quatre veus). Les veurem amb més detall a l'apartat sobre els sons d'aquesta secció.

8.6.1 Gestió de les imatges

Tal com hem comentat a la secció sobre els recursos dins LibGDX (pàg. 46), les imatges vindran donades per dos objectes: un de tipus **Texture** i un de tipus **Sprite**, essent el segon el que utilitzarem per dibuixar (el primer (**Texture**) serà només necessari en tant que formarà part del segon).

Així doncs, una vegada entrem a la inicialització dels recursos (**carrega**) tindrem per a cada imatge tres línies de codi: una primera en què s'importa el fitxer i s'assigna a l'objecte **Texture**, una que escull quins filtres s'utilitzaran per a escalar la imatge, i una última que assigna l'objecte **Texture** amb els filtres ja escollits a l'objecte **Sprite** de manera que el puguem utilitzar dins la funció **render** de les pantalles.

Pel que fa als filtres escollits, s'ha decidit demanar al programa la utilització de *MipMaps*, que són un seguit de còpies successives de la imatge cadascuna reduïda a la meitat[90]. D'aquesta manera, en cas que l'aparició en pantalla de la imatge sigui més petita que el fitxer original, en lloc d'haver de calcular sobre la imatge grossa el programa pot agafar la imatge del conjunt que més s'apropi a la mida que la imatge haurà de tenir en última instància, estalviant gran part del càlcul.

Tot i això, per tal que el *MipMap* funcioni correctament, en principi LibGDX demana que les dimensions de les imatges que en portin siguin totes potències de dos, així que a l'hora de preparar els fitxers d'aquestes n'haurem de ser conscients.

D'altra banda, pel que fa al tipus de filtre, a efectes pràctics s'ha escollit un filtre **Linear** (fa una mitjana dels píxels originals més propers al punt que dibuixarà) tant per quan fem la imatge petita com per quan la fem gran. L'altra opció possible seria el filtre **Nearest**, que agafa directament el píxel més proper, però amb aquest els resultats són pitjors. D'altra banda, els MipMaps només s'han aplicat al filtre de reduir, ja que en el d'augmentar no tenen massa sentit.

En tot cas, la utilització correcta d'aquests filtres és el que ens permet obtenir uns gràfics suavitzats en lloc d'obtenir imatges amb *aliasing* (i. e. vores pixelades).

8.6.2 Gestió de les fonts

Pel que fa a les fonts, s'ha escollit generar vuit mides diferents, de les quals pràcticament sempre utilitzarem dues (les anomenades **font**, i **xifrat**). La resta de fonts, potser amb l'excepció de **bMenu** i **bNotes**, les utilitzarem pràcticament només per a situacions molt concretes.

[90] Aquestes còpies les genera automàticament el programa (si li demanem) en el moment d'assignar un fitxer d'imatge a un objecte de tipus Texture.

En tot cas, per a LibGDX cada tipus de font en una mida en concret és un recurs diferent, de manera que hem hagut de generar un objecte de tipus **BitmapFont** per a cadascuna de les mides d'aquesta (els podem veure a la declaració d'objectes i variables de la unitat de compilació).

Pel que fa a la inicialització d'aquestes, el procediment serà similar al que s'ha dut a terme amb els objectes **Texture** i **Sprite**: un primer objecte (**FreeTypeFontGenerator**) recollirà el fitxer original de la font, a continuació assignarem els paràmetres que creguem oportuns i finalment generarem la font assignant-la al nostre objecte **BitmapFont**, que és el que dibuixarem.

En aquest cas, però, degut que l'assignació de paràmetres és més complexa (hi ha més paràmetres que ens pot interessar escollir), així que els gestionarem a part demanant-los a l'objecte que els recull (**FreeTypeFontParameter**). Aquest paràmetre, doncs, és el que recollirà de quin color volem la font (blanc en cas que la vulguem pintar després) i de quina mida voldrem la font, així com la nostra tria pel que fa als MipMaps.

Degut que la tria dels MipMaps i del color seran compartides per a totes les mides, podrem aprofitar el mateix objecte dels paràmetres per a totes les fonts: assignarem la font amb una mida al primer objecte **BitmapFont**, i posteriorment canviarem la mida del paràmetre per continuar assignant la nova mida a l'objecte següent.

Finalment, després de cada assignació haurem de triar els filtres de l'objecte en qüestió (a diferència del cas de les imatges, aquí el filtre el rep directament l'objecte que dibuixarem). Per a les fonts, la tria de filtres ha estat la mateixa que en el cas de les imatges, per les mateixes raons.

8.6.3 Gestió dels sons

Si bé en un principi es volia realitzar la reproducció de l'esquema mitjançant MIDI, s'ha vist finalment que en Java i degut a la naturalesa de la música amb què haurem de tractar, era molt més senzill implementar la reproducció mitjançant àudios curts provinents d'enregistrar notes individuals a partir d'un instrument real (en anglès, *"samples"*).

Aquesta opció, a més, va esdevenir especialment senzilla en descobrir que la *Philarmonia Orchestra* té sota llicència *Creative Commons* una gran col·lecció de notes gravades pels seus integrants.

Per tal de poder disposar d'àudios amb timbres similars per a totes les tessitures, es va escollir utilitzar les gravacions dels instrumentistes de corda fregada. En concret, s'han utilitzat gravacions de violí i de contrabaix (cobrint aproximadament cadascun la meitat del registre), amb durada d'1.5 segons, vibrat normal i dinàmica *forte*.

Degut que dins aquest subapartat de la col·lecció no totes les notes es trobaven disponibles, s'han generat les notes que faltaven transportant els sons més propers mig to mitjançant *Audacity*. A més, s'ha hagut de retocar mitjançant el mateix programa el moment d'atac de la nota dins de cada fitxer, degut que la majoria de fitxers tenien l'inici de la nota en un punt diferent (i. e. l'espai de silenci fins al principi de la nota durava diferent d'un fitxer a l'altre).

En tot cas, per tal de facilitar la gestió dels fitxers d'una manera paramètrica dins l'aplicació, s'ha assignat a cada nota un número de tres xifres en què les centenes corresponen a l'índex acústic i les desenes i unitats al so cromàtic (p. ex. 402 representaria el segon so de l'escala cromàtica d'índex

acústic 4, és a dir, un Re_4), i s'ha anomenat tots els àudios de la mateixa manera, canviant únicament aquests números.

L'enfocament, doncs, dins l'aplicació ha estat en un primer lloc carregar (d'entre els 48 àudios que tindrem disponibles[91]) els sons corresponents a totes les notes que s'han utilitzat dins l'esquema (funció **carregaSons**) i a continuació reproduir-les acord a acord (funció **playChord**), esperant uns 1.5 segons entre el llançament d'un acord i el llançament del següent. La gestió d'aquests temps, així com la crida d'aquestes funcions, la veurem en parlar de la pantalla de la realització (pàg. 86).

Per tal d'aconseguir amplitud en l'àudio reproduït i facilitar així a l'usuari la distinció i seguiment de les notes individuals dins el conjunt, s'ha panoramitzat cadascuna de les veus per tal que prenguin posicions diferents en l'espai auditiu (i. e. la soprano s'ha ubicat a l'altaveu esquerra, la contralt a un terç, el tenor a dos terços, i el baix a la dreta del tot).

8.6.3.1 Asset Manager

Degut que la gestió dels duplicats i la càrrega de recursos de forma asíncrona feien complexa la gestió dels sons, s'ha decidit utilitzar una eina que ofereix LibGDX anomenada **AssetManager**. Aquesta bàsicament s'encarrega de rebre una llista de recursos que volem carregar, descartar els repetits i els que ja estiguin carregats, i posteriorment carregar una mica de la llista cada vegada que se li demana des de la funció **render** d'una pantalla[92] (i. e. una mica a cada fotograma, en lloc de bloquejar-ho tot fins que està de carregar).

Posteriorment, quan vulguem utilitzar els sons que **AssetManager** hagi carregat, només haurem d'assignar-los a un objecte de tipus **Sound** i reproduir-los. De la mateixa manera, quan vulguem alliberar els recursos corresponents als sons utilitzats, només li haurem de demanar a l'**AssetManager** que es buidi amb el mètode **clear**.

8.7 Unitat central (hEditor)

Una vegada vistes les diferents unitats de compilació "extra" que s'han utilitzat com a complement de les unitats principals, podem passar a veure les unitats principals en sí. La primera unitat que veurem dins aquest grup (la unitat central, anomenada **hEditor**) serà també la primera unitat a què accedirà la nostra aplicació una vegada es posi en marxa, així com el marc general dins el qual succeirà tota la resta. Aquí, doncs, és on s'ubicaran les instàncies de les diferents pantalles LibGDX[93].

D'altra banda, la unitat disposa d'un mètode activat al moment de crear-la (**create**), que ens servirà per inicialitzar i marcar com a pantalla activa la pantalla inicial[94].

Aquesta activació de la pantalla es realitzarà mitjançant la funció de LibGDX **setScreen**.

[91] Dotze sons per a cada índex acústic de l'1 fins al 4, rang que engloba una mica més que les tessitures estàndard dels quatre cantants.
[92] Parlarem de la funció render a les seccions sobre les diferents pantalles de l'aplicació.
[93] Si més no, les pantalles que vulguem conservar obertes tota l'estona. Algunes pantalles a què s'accedeixi menys sovint ens pot interessar crear-les al moment d'accedir-hi, i destruir-les en acabat, de manera que en aquests casos no se'n crearia una instància a priori.
[94] Veure "Pantalla inicial (PInici)" (pàg. 102).

8.8 Pantalla de l'Esbós (Pant)

La pantalla central de l'aplicació (si més no en termes de variables compartides) és la pantalla de l'esbós. Aquesta és on s'ubiquen les variables derivades d'observar les mides del dispositiu que fa córrer l'aplicació, així com altres valors relatius a les mides del món de l'aplicació. A més, la seva funció serà permetre a l'usuari escollir una seqüència d'acords que posteriorment haurà de realitzar en forma de composició a quatre veus.

8.8.1 Disseny Gràfic

La pantalla de l'esbós consta d'un pentagrama en clau de sol amb la seqüència d'acords escollida, i d'un seguit de botons a baix i a la dreta que permeten editar aquesta seqüència, així com d'un botó amb el nom del to actual que fa la funció de botó del menú (Figura 34).

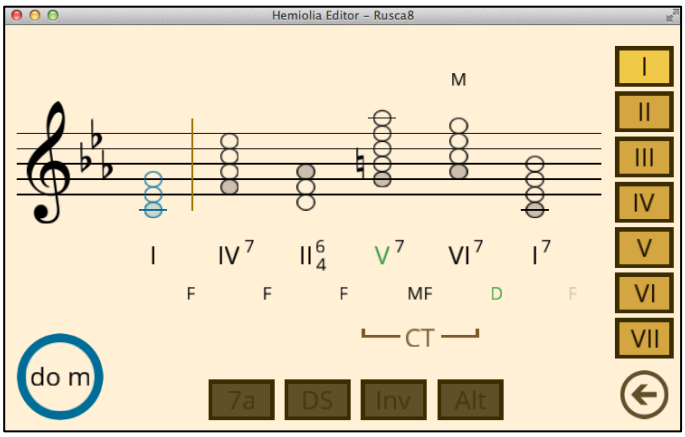

Figura 34: Exemple de seqüència d'acords escollida dins la pantalla de l'esbós.

Sota cadascun dels acords escollits podem veure el seu grau en el to actual[95], i un xifrat representatiu de la configuració escollida. A més, sota cada parella d'acords podem veure una lletra que representa el tipus d'enllaç[96] que formen (F si és fort, D si és dèbil i MF si és molt fort), i sobre alguns dels acords podem veure una M que ens avisa que la sèptima és Major (i per tant s'haurà de realitzar diferent[97]).

D'altra banda, en una tercera línia podem veure marcades també les cadències trencades.

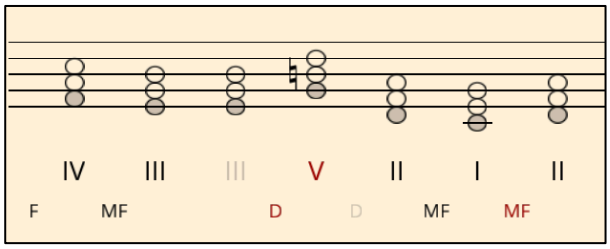

Figura 35: Detall de la correcció dels enllaços i els graus.

[95] Veure "Graus i funcions tonals" (pàg. 13).
[96] Veure "Enllaços" (pàg. 28).
[97] Veure norma N19.2 (pàg. 37).

Els colors del text referent a enllaços, grau i xifrat (Figura 35) responen a la correcció realitzada a partir de la normativa associada a aquests. Veurem amb més detall el funcionament d'aquesta correcció a l'apartat posterior sobre el codi de la pantalla.

Pel que fa als botons, a la dreta tenim els diferents graus disponibles (I-VII), així com un botó per esborrar, i a baix tenim les diferents opcions que podem escollir per cadascun dels graus. Degut que no tots els acords admeten totes les opcions (i algunes opcions s'afecten entre elles), les opcions que no es puguin escollir es veuran desactivades (veurem el funcionament d'aquesta desactivació a les explicacions sobre el codi).

Les opcions que contempla el codi actualment[98] són afegir sèptima (botó 7a), passar a dominant secundària (botó DS/DD/V9[99]), canviar la inversió (botó Inv), i activar l'alteració opcional (botó Alt).

Per tal de marcar les inversions, s'ha omplert la nota que anirà al baix amb un color més fosc.

Degut que els darrers acords de les seqüències de certa longitud quedarien amagats sota el marge dret, s'ha implementat una funció que permet arrossegar el pentagrama cap als dos costats per tal de veure'ls (Figura 36). A més, s'ha implementat una funció complementària a aquesta que realitzarà el desplaçament de manera automàtica si el programa detecta que s'està editant un acord fora el camp de visió (per evitar que l'usuari editi acords que no veu i no se n'adoni).

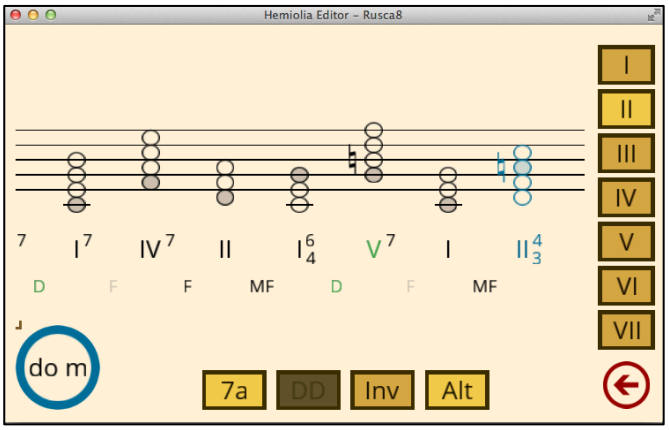

Figura 36: Pantalla de l'esbós amb un valor diferent de desplaçament i un acord actual diferent que provoca canvis en els botons de les opcions.

D'altra banda, s'ha com es pot veure a les dues imatges, els acords seleccionats quedaran destacats en color, juntament amb tots els elements gràfics representatius de paràmetres que es puguin editar amb la selecció actual (i. e. segons com seleccionem l'acord[100], els botons de grau inseriran un nou acord a continuació o bé editaran el grau de l'acord seleccionat. En cas que puguem editar el grau de l'acord actual, el text d'aquest també apareixerà ressaltat).

[98] Responent als requeriments de la primera iteració definida a l'apartat 7.1 (pàg. 48)
[99] De moment, l'aplicació sap gestionar les dominants secundàries, però no les contempla a la correcció, de manera que el botó s'ha mantingut de moment desactivat.
[100] Prémer sobre un acord activarà l'edició completa, mentre que prémer entre dos acords activarà el mode d'inserció.

8.8.1.1 Plantejament de la interfície d'usuari

La tria de la posició dels botons dins l'aplicació s'ha dut a terme intentant que aquesta sigui còmoda per l'usuari: la major part de les vegades s'anirà intercalant un clic als botons de grau amb un o dos clics als botons d'opcions, i posar aquests dos grups de botons a la dreta i a baix permet una posició còmoda de les mans agafant el telèfon, així com realitzar els clics de cadascun d'aquests dos apartats amb cadascun dels dos polzes i sense que el fet de prémer un botó impliqui ocultar amb els dits una part rellevant de l'aplicació. A més, que sigui a baix en lloc de a l'esquerra evita que les columnes de botons tapin una part massa gran de la longitud horitzontal visible del pentagrama[101].

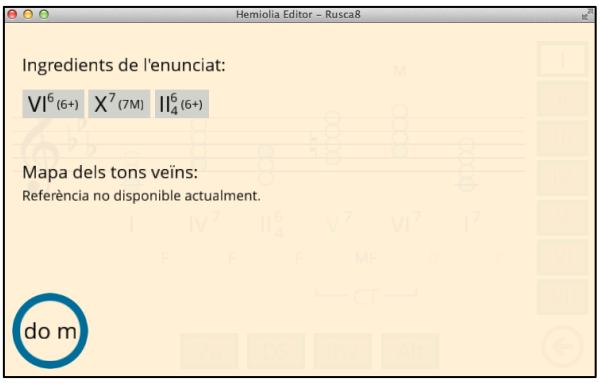

Figura 37: Mantenir premut el botó del to mostra les referències (de moment els ingredients).

D'altra banda, s'han implementat dues funcions diferenciades pel que fa a prémer el botó del menú. La primera és mostrar les referències que l'usuari pugui necessitar consultar durant la tria de l'esbós[102]. Aquesta es mostra en mantenir premut el botó un temps una mica més llarg que el que dura un clic normal (Figura 37), i desapareix tan bon punt es deixa de prémer. D'aquesta manera, l'usuari podrà fer una ullada ràpida a les referències sense sortir de la pantalla en què es troba.

Figura 38: Prémer el botó del to (sense aguantar) ens porta al menú de la pantalla.

[101] L'altra opció similar seria posar columnes a esquerra i dreta, però el tros vertical que guanyaríem no l'utilitzaríem per res, i en canvi ens interessa veure com més tros horitzontal millor.

[102] En aquesta iteració de l'aplicació només ens caldrà mostrar els ingredients de l'enunciat, però en futures iteracions serà útil per a l'usuari poder tenir a mà quines tonalitats són properes pel que fa a l'armadura.

En cas que el botó es deixi anar abans del temps establert (clic normal), s'activarà el menú pròpiament dit (Figura 38), en què podem canviar la tonalitat escollida, així com demanar el salt a la pantalla de la realització o el retorn a la pantalla dels ingredients.

Per tal de fer aquest canvi de tonalitat d'una manera coherent amb les armadures, les fletxes esquerra i dreta afegiran i restaran, respectivament, una quinta. Així, la fletxa dreta afegirà un sostingut (o traurà un bemoll), i l'esquerra afegirà un bemoll (o traurà un sostingut).

8.8.2 Objectes i variables genèrics

Com hem vist anteriorment, la pantalla haurà de disposar de càmera, finestra i lot, i a més haurà de desar una referència a la unitat central de manera que hi puguem accedir posteriorment per utilitzar-ne les altres pantalles. Aquests quatre elements els podem veure al principi del codi de la pantalla.

Per tal d'aprofitar les mides reals del dispositiu sense fer massa complicat el tractament de les posicions, s'ha decidit configurar la finestra (**Viewport**) per tal que escali de manera proporcional les mides de la càmera fins tocar amb l'alçada del dispositiu, i posteriorment amplïi el camp de visió cap a la dreta. Així doncs, la nostra alçada serà fixa, i l'amplada variarà segons l'amplada del dispositiu en què corri l'aplicació, ampliant o escurçant la distància fins al marge dret.

Després de les instàncies de la càmera, la finestra i el lot, tindrem les variables de les mides, que respondran a l'explicació del paràgraf anterior. Tindrem una alçada de pantalla fixa (fictícia), i calcularem a partir de l'alçada i amplada real una proporció (**ratio**) que ens servirà per descobrir amb quina amplada fictícia hem de treballar si volem aprofitar tota l'amplada del dispositiu.

Així doncs, tots els botons i elements gràfics es posicionaran en relació al marge esquerre i aquesta amplada calculada.

D'altra banda, per tal de gestionar els clics necessitarem un parell d'objectes **Vector3**, que representen punts en l'espai (la z no la utilitzarem, però és necessària). Això és degut que els mètodes que s'activen amb els clics ens donaran la posició en relatiu a les mides reals del dispositiu, i el **Viewport** necessita que aquestes posicions estiguin guardades en un objecte **Vector3** per poder-hi aplicar la transformació que les tradueix a posicions dins el món de l'aplicació.

A més, també en relació als esdeveniments tàctils, tindrem un seguit de variables relacionades amb la funció de desplaçament (bàsicament els límits esquerra i dreta del camp de visió i el valor de desplaçament que s'ha assignat en un moment concret).

D'altra banda, en aquest apartat també tindrem un seguit de variables relacionades amb les mides i posicions dels elements gràfics, que ens interessarà tenir parametritzats per si en algun moment ens interessa canviar-ne la ubicació.

8.8.3 Objectes i variables relacionats amb la correcció

El següent apartat del codi dins la definició d'objectes i variables està dedicat als objectes i variables que tenen a veure amb la correcció de l'esbós segons les normes d'harmonia. Tot i això, la majoria de les normes que hem vist a l'apartat 4.2 tenen a veure amb la realització, de manera que en aquesta pantalla la part de la correcció serà relativament petita.

Dins aquest apartat tindrem la seqüència d'acords (un Array d'objectes Chord anomenat **esbos**), una variable per marcar l'acord actual (**aAct**), i un booleà que ens indicarà si estem editant l'acord actual o bé inserint un acord nou a continuació d'aquest (**insert**).

A més, tindrem un seguit de llistes que treballen pròpiament amb les dades que es generen en corregir: una que desa els tipus d'enllaç calculats, una que desa les cadències trencades i perfecta, i tres que desen els resultats de la correcció pel que fa a graus, enllaços i inversions.

En aquesta pantalla, la correcció ve donada per un Array de caràcters (**Character**), que tindrà un valor per a cada acord (o per a cada enllaç, segons el cas). Aquests valors, doncs, ens marcaran si l'entitat corregida està correcta (**c**), amb error (**e**) o en alerta (**a**), entre d'altres.

8.8.4 Objectes i variables relacionats amb els botons

Finalment, l'últim apartat dins la declaració d'objectes i variables serà tot el que tingui a veure amb els botons de l'aplicació. Els botons pròpiament dits seran un objecte (**BotQ**) que desa les mides i posicions horitzontal i vertical, així com alguna altra dada d'utilitat, com pot ser el text que voldré que es mostri sobre el botó[103].

Si fem una ullada a la Figura 34, els botons que veurem a la dreta (botons dels graus) són un conjunt de botons que tenen funcions pràcticament idèntiques (canvia un paràmetre). Degut a això, ens interessarà poder-los tractar com un conjunt. Per aquest motiu, s'han desat en un Array que els engloba tots 7.

En canvi, els botons d'opcions de la barra inferior, degut que fan funcions molt diferents, s'ha preferit desar-los de manera individual.

D'altra banda, juntament amb els botons d'opcions, s'han creat una sèrie de booleans que serviran per indicar si el botó es pot fer servir en un context donat (cosa que es calcularà en una funció a part ubicada cap al final del codi, anomenada **permisosBotons**).

Finalment, s'han definit alguns booleans més per respondre a qüestions més pragmàtiques (p. ex. estalviar càlcul a la funció de clic quan ja s'ha activat algun botó (**clickBot**), o marcar que hi ha hagut canvis prou grans com per haver de corregir de nou (**calCorregir**)). A més, s'han definit els botons necessaris per implementar el menú de la pantalla, així com la variable que ens marcarà si el menú està activat o no.

8.8.5 Constructor de la pantalla

Pel que fa al constructor de la pantalla, bàsicament s'inicialitzaran les variables i els botons, i es farà un seguit de tests de les funcions H en forma de text enviat a la consola.

8.8.6 Render

Després del constructor tindrem la funció de dibuix de les pantalles LibGDX. Aquesta sempre incorpora una variable que ens permet saber quants segons han passat des de l'anterior iteració (de

[103] Per facilitar la utilització i el seguiment del codi s'ha començat per "b" el nom de tots els objectes de tipus botó.

manera que puguem posar en marxa qualsevol animació de manera fluida sense que hi afecti el fet que el telèfon trigui més per uns fotogrames que per uns altres). Aquesta variable (**delta**), malgrat útil, té el perill que si el mòbil queda una mica penjat durant un fotograma, al següent les animacions faran un salt molt brusc. Per evitar això, el primer que farem dins aquesta funció és dir-li que si el temps entre fotogrames passa d'un quart de segon, preferirem considerar que no ha estat més que aquest quart de segon.

Posteriorment, es fa una neteja de la pantalla amb el color de fons escollit, i s'actualitza la càmera.

En aquest punt, executaré la correcció de l'esbós en cas que hagi marcat que calia, a partir d'una funció a part (**corregeix**) que podem trobar al final del codi de la pantalla, i un cop això estigui fet, encetaré el lot de dibuixos pròpiament dit. Aquest estarà dividit en dues seccions segons si estic mostrant el menú o no, i dins de cadascun d'aquests primer pintaré els elements gràfics normals, i posteriorment afegiré els botons.

Finalment, abans de tancar el fotograma, actualitzo les animacions enviant el número **delta** a la biblioteca que se n'encarrega (**A**).

8.8.7 Gestió d'entrades

El següent que podrem veure després de **render** són les funcions encarregades de respondre als esdeveniments d'entrada. Dins d'aquestes, el primer que veurem és la finestra traduint la posició real del clic a la posició equivalent dins el nostre món.

A partir d'aquí, la dinàmica és senzilla: es van comprovant els botons (sovint per àrees per tal de descartar ràpidament els botons que quedin lluny del clic), i si se n'ha polsat algun se n'activa l'efecte corresponent i es marca que hi ha hagut un clic activant **clickBot**. En general, si aquesta variable ha estat activada, ja no es comprovaran els botons que hi pugui haver a continuació, de manera que ens estalviarem els càlculs de col·lisió pertinents. A més, tenir aquesta dada com una variable a part ens permet no haver de tenir tots els botons en una llarga tira de **if** – **else** i per tant poder-los separar per seccions de la pantalla com s'ha comentat.

La major part dels botons s'accionen en el moment de prémer (**touchDown**), però hi haurà casos en què ens interessarà esperar que es deixi anar abans de fer res (p. ex. accions a la zona de desplaçament que passen només si després de prémer no s'ha arrossegat). En aquests casos s'ha utilitzat **touchUp**, i per gestionar el desplaçament pròpiament dit s'ha utilitzat **touchDragged**.

Per la part de **touchDragged**, l'enfocament ha estat calcular el canvi de posició entre el clic anterior i la posició després d'arrossegar, i afegir aquest increment al valor de desplaçament (**scroll**). Un cop aquest càlcul ha estat fet, es comprovarà el context i els marges esquerre i dret, i s'ajustarà el desplaçament en conseqüència (impedint moure l'esbós si ja es veu tot o si ja s'ha arribat a una de les dues puntes).

A part, s'ha implementat una funció extra anomenada **fixScroll** que s'encarrega de fer saltar l'esbós fins a un punt en què es pugui veure l'acord actual. Aquesta funció es cridarà des de **touchDown** quan alguna de les accions dels botons canviï l'acord actual, per evitar que l'usuari estigui fent canvis que no veu.

8.8.8 Resize i show

Un cop acabades les funcions de resposta a les entrades, tindrem les funcions que gestionen canvis en la pantalla: **resize** quan es canvien les mides, i **show** quan es canvia de pantalla dins l'aplicació.

A **resize** hi trobarem una nova realització de tots els càlculs que tinguin a veure amb la mida de la pantalla: s'actualitzarà la finestra, s'actualitzarà la **ratio**, i s'actualitzaran els valors de **x** (posició horitzontal) de tots els botons que no estiguin ancorats al marge esquerre, així com qualsevol amplada de botó que depengui de l'amplada de pantalla.

D'altra banda, degut que en aquesta pantalla alguns botons canvien de posició i mida en resposta a l'estat d'algunes variables, s'ha creat una funció apart anomenada **posaElsBotonsPer** que s'encarrega de gestionar-ho.

A més, s'actualitzarà el límit dret del desplaçament, així com la posició de la càmera dins la finestra, i es calcularà la ubicació dels ingredients de l'enunciat a la pantalla de les referències.

Pel que fa a la funció **show**, es posaran a zero alguns valors de posicions i variables que puguin haver quedat moguts en sortir de la pantalla (i que voldrem a la posició original quan hi tornem), i s'assignarà de nou a la pantalla la responsabilitat de gestionar les entrades (cada vegada que canviem de pantalla hem de dir a l'aplicació que ara les entrades s'hauran de gestionar a través de la pantalla nova).

A més, actualitzarem els càlculs de l'armadura i realitzarem una primera correcció en el primer moment que es mostra la pantalla per assegurar que existeixen certs valors de correcció i variables que podríem necessitar a la funció de dibuix (**render**).

8.8.9 Pause, resume, hide i dispose.

Si ens fixem en el codi, darrere **show** hi haurà un seguit de mètodes que no s'han fet servir. Aquests han d'estar implementats obligatòriament degut a la naturalesa de la classe **Screen**, però pel que hem de fer nosaltres no ens caldrà que portin res a dins (en aquesta pantalla, al menys).

8.8.10 Funcions de càlcul i correcció

L'últim apartat d'aquesta pantalla (així com l'últim apartat de la pantalla de la realització), seran les funcions relacionades amb el posicionament d'elements i amb la correcció. En aquest cas, primer tindrem la funció **posicionaIngs**, que és la que s'encarrega d'ubicar els ingredients de l'enunciat per tal que es puguin representar en forma de línia a la pantalla de les referències, i tot seguit la definició de la funció **posaElsBotonsPer**, esmentada anteriorment. Finalment, abans de la correcció pròpiament dita tindrem la funció **permisosBotons** de què hem parlat anteriorment, que comprovarà l'acord actual i decidirà quins botons hi podem aplicar.

A continuació, doncs, la funció **corregeix** realitzarà tots els càlculs necessaris per a poder corregir, així com els càlculs de la correcció pròpiament dita: calcularà el tipus dels enllaços entre acords[104] i desarà un caràcter de correcció per cadascun d'ells, buscarà cadències trencades[105], desarà caràcters

[104] Veure "Enllaços" (pàg. 28).
[105] Veure "Cadència Trencada" (pàg. 31).

de correcció per cadascun dels graus i per cadascuna de les inversions[106], i comprovarà l'existència de la cadència perfecta[107].

Un cop realitzada la correcció, la mateixa funció realitzarà un càlcul del nombre d'errors que tenim de cada tipus i farà un recompte general. A més, tornarà a calcular els permisos dels botons (es calculen quan es corregeix o bé quan s'escull un acord diferent prement dins l'àrea de desplaçament sense arrossegar).

8.9 Pantalla de la realització (Pant2)

La pantalla de la realització serà la pantalla més complexa, degut a la gran quantitat d'elements gràfics que haurà de mostrar, i a la gran quantitat de codi que formarà part de la correcció.

L'estructura del codi, però, en termes generals serà molt similar a la de la pantalla de l'esbós.

8.9.1 Disseny Gràfic

La pantalla de la realització consta d'un seguit de botons relacionats amb les diferents veus del cor mixt i amb les diferents notes de l'acord actual, així com d'un sistema de pentagrames (un en clau de sol i un en clau de fa), que seran els encarregats de recollir la solució de l'usuari pel que fa a la realització de l'esquema a quatre veus (les dues veus de noia aniran al pentagrama superior i les dues de noi al pentagrama inferior). A més, a la part inferior tindrem un pentagrama petit amb l'esbós realitzat a la pantalla anterior (a mode de referència), així com un parell de fletxes que ens permetran navegar a través de l'esquema (canviar d'acord actual).

Com es pot veure a la Figura 39, al costat esquerre podrem seleccionar la veu a qui voldrem assignar la nota escollida, mentre que al costat dret podrem escollir quina nota de l'acord li assignem, així com esborrar la nota seleccionada. Les notes mostrades a la columna dreta responen a les notes que té l'acord seleccionat en cada moment, i s'aniran actualitzant cada vegada que l'usuari canviï d'acord navegant per l'esquema. L'acord actual, com es pot veure, quedarà marcat en blau tant a la realització com a l'esbós.

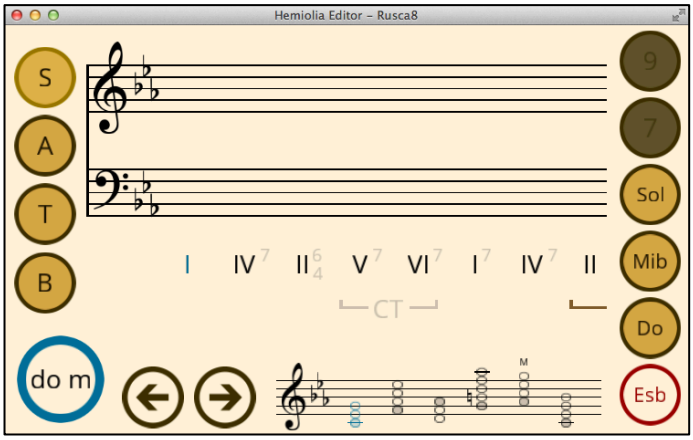

Figura 39: Pantalla de la realització (buida).

[106] Veure norma N16.2 (pàg. 36).
[107] Veure apartat 4.2.4 (pàg. 29).

D'altra banda, podem observar dues línies sota el sistema amb els graus, xifrats i cadències[108]. Els graus i cadències ja sabem del cert quins seran degut que s'han escollit a l'esbós, de manera que ja són els definitius. El xifrat, en canvi, pot acabar essent diferent segons com es col·loquin les notes a l'hora de realitzar, així que els números mostrats aquí abans que l'acord tingui totes les notes (els xifrats en semitransparent) seran només una al·lusió a la inversió que s'havia escollit en el moment de fer l'esbós, i canviaran una vegada l'acord estigui ple en cas que la resolució difereixi de l'opció planejada.

Si afegim algunes notes a la realització buida de la Figura 39, començarem a veure amb més detall el comportament de l'aplicació (Figura 40). Com es pot observar, els botons de les veus que ja tenen una nota assignada s'il·luminen en groc, així com les notes de l'acord que ja han estat ubicades.

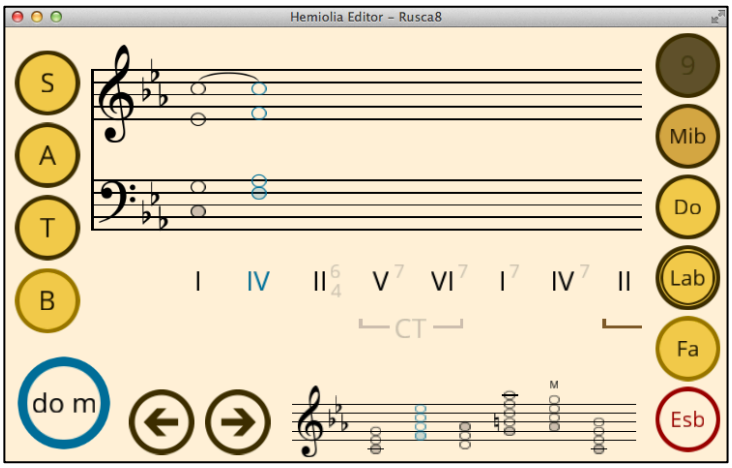

Figura 40: Pantalla anterior després d'introduir algunes notes a l'esquema.

D'altra banda, podem veure que ha desaparegut el xifrat provisional del segon acord, degut que aquest ja té notes assignades a totes les veus i ha acabat essent tríada en estat fonamental[109].

Podem veure també que ara que ens trobem en el segon acord, les notes dels botons del marge dret han canviat en consonància, i que ara tenim actiu un botó més que abans (havíem triat acord amb sèptima[110] a l'esbós), malgrat hem decidit no utilitzar-lo i deixar finalment l'acord sense sèptima. A més, un cercle extra sobre el botó de la tercera ens indica que aquesta nota disposa d'alteracions opcionals, a les quals podem accedir si el mantenim premut (Figura 41).

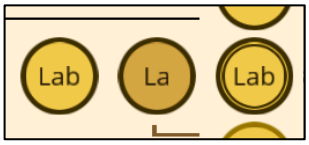

Figura 41: Detall del desplegament de botons corresponent a la tria d'alteracions opcionals.

A més, si ens fixem en les notes, veurem que el programa ha detectat dues notes iguals consecutives dins la mateixa veu i hi ha aplicat una lligadura[111].

[108] A partir d'aquest punt ja no ens interessarà veure els tipus d'enllaç, així que ja no s'hi mostren.
[109] Veure xifrats de la Taula 15 (pàg. 19).
[110] Veure "Acords" (pàg. 17).
[111] La detecció de lligadures es realitza simultàniament a la correcció.

Podem veure altres exemples de la manera com s'executa gràficament la correcció a Figura 42, Figura 43, Figura 44 i Figura 45.

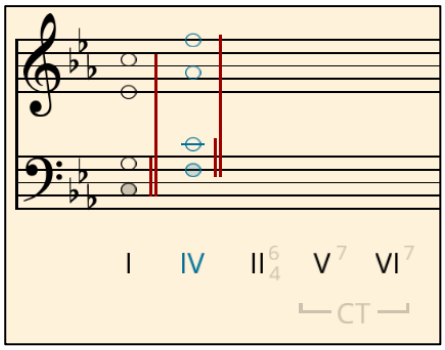

Figura 42: Exemple de correcció de moviments paral·lels.

La Figura 42 ens mostra el resultat de no respectar les normes sobre moviments paral·lels. En aquest cas, dues barres vermelles ens mostren cadascuna de les faltes: les dues barres curtes de la part inferior ens marquen que s'han fet quintes seguides entre els dos acords que les tenen, mentre que les dues barres llargues ens marquen que s'hi ha fet octaves[112].

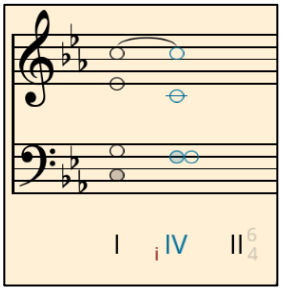

Figura 43: Exemple de correcció de notes suprimides i gestió de solapaments.

D'altra banda, a la Figura 43 podem veure com el programa ens marca un error en les notes suprimides (s'ha suprimit la tercera de l'acord, que és una nota important, de manera que ens ho marca amb una i[113] vermella). A més, podem veure al pentagrama inferior que el programa ha detectat un solapament entre dues notes (dues veus es troben tan a prop que les notes quedarien solapades), així que ha mogut la nota de la veu superior cap a la dreta per tal que es vegin totes correctament.

El cas de la Figura 44 ens mostra ja una quantitat important d'errors en un mateix acord. Com es pot veure, tenim una creu vermella entre les dues notes del mig, dos punts a la nota del tenor, i dues indicacions al costat del grau. La creu vermella indica que les dues veus interiors s'han creuat (el tenor canta més agut que la contralt), i els dos punts indiquen que la nota que els té s'ha de preparar i resoldre i que aquestes preparació i resolució estan pendents. A més, els dos punts en groc al costat del grau indiquen que hi ha hagut una duplicació estranya, i la "i" vermella indica de nou que s'ha comès un error en la supressió de notes (en aquest cas s'ha suprimit la fonamental).

[112] La longitud de la barra no és per l'interval trobat, sinó per la distància entre les dues notes que el formen.
[113] Significant "incomplet".

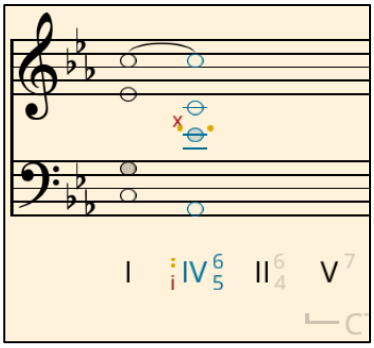

Figura 44: Cúmul d'errors i alertes sobre un mateix acord.

Finalment, tancant aquest seguit d'exemples al voltant del disseny de la manera com s'indiquen errors i alertes, tenim la Figura 45. En aquest cas es marca amb una A vermella que hi ha un interval augmentat entre les notes col·locades als dos primers acords[114]. A més, les notes del tercer acord han estat pintades de vermell i groc per marcar que se surten de la tessitura estàndard dels cantants[115].

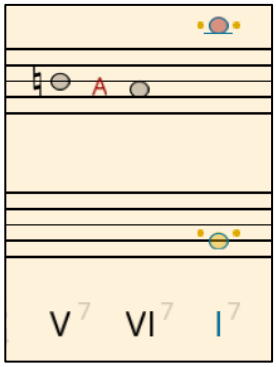

Figura 45: Correcció d'intervals augmentats i de tessitures.

A més, veiem els mateixos punts grocs de l'exemple anterior, marcant que les notes s'han de preparar i resoldre (en aquest cas degut que són sèptimes, però també passarà amb els acords tríades en segona inversió).

En aquests casos, com en el cas de l'esbós, la correcció també funciona en una funció a part[116] que s'executa quan hi ha canvis i que consta de tots els càlculs i comprovacions necessaris per trobar les faltes comeses. Aquests càlculs desen els errors i alertes, i posteriorment aquests s'utilitzen per pintar la correcció, ja sigui afegint elements gràfics o canviant els colors dels elements que es mostren sempre.

8.9.1.1 Plantejament de la interfície d'usuari

En aquest cas, el plantejament ha estat similar al de la pantalla de l'Esbós: S'han muntat dues zones formades per línies de botons pensant que l'usuari les utilitzarà cadascuna amb un polze. En aquest cas, però, s'ha preferit moure els botons de la mà esquerra al marge esquerre (en lloc de deixar-los a

[114] Aquest tipus d'interval melòdic està prohibit per la normativa (N5.1, pàg. 26).
[115] Veure normes N1 (pàg. 24). La tria de groc o vermell respon a com d'estricte s'ha considerat el límit (i. e. a fins a quin punt les diferents fonts consultades han coincidit en la ubicació d'aquest).
[116] Veure apartat 8.9.10 sobre les funcions de càlcul i correcció (pàg. 95).

baix), perquè ens interessa mantenir certa amplitud vertical per poder mostrar l'esbós en petit com a referència per l'usuari.

Així doncs, s'ha muntat una columna a l'esquerra pel polze esquerre i una a la dreta pel polze dret. Alhora, s'han posat els botons de navegació (les dues fletxes) a prop del marge esquerre, ja que l'habitual serà alternar la inserció d'una nota (marge dret) amb un canvi d'acord o de veu (marge esquerre), segons l'usuari ompli l'esquema en horitzontal o en vertical, respectivament.

En una primera instància es va plantejar la ubicació de les fletxes a banda i banda de l'esbós, però degut al comportament d'usuari descrit al paràgraf anterior, aquesta configuració era incòmoda d'utilitzar, així que es va canviar en pro de la comoditat i l'eficiència.

D'altra banda, en aquesta pantalla també s'ha implementat la pantalla de les referències de la mateixa manera que a la pantalla de l'esbós, i també s'hi ha fet un menú.

La pantalla de les referències en aquest cas inclourà a més de l'enunciat quatre pentagrames curts indicant els rangs als quals arriben els cantants (Figura 46).

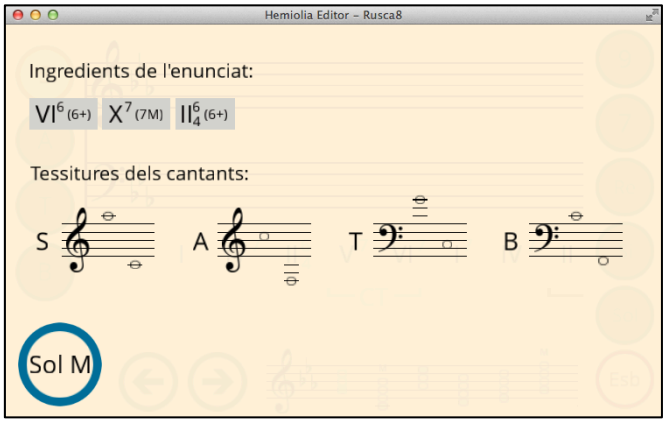

Figura 46: Pantalla de les referències dins la realització.

En aquest cas, el menú ens permetrà tornar a l'esbós per editar-lo, així com esborrar tota la realització mantenint l'esbós (netejar i començar de zero, per no haver d'esborrar-ho tot nota per nota). A més, un tercer botó ens permetrà carregar els sons necessaris per a reproduir l'esquema, i posteriorment escoltar-lo. Podem veure la pantalla del menú a la Figura 47.

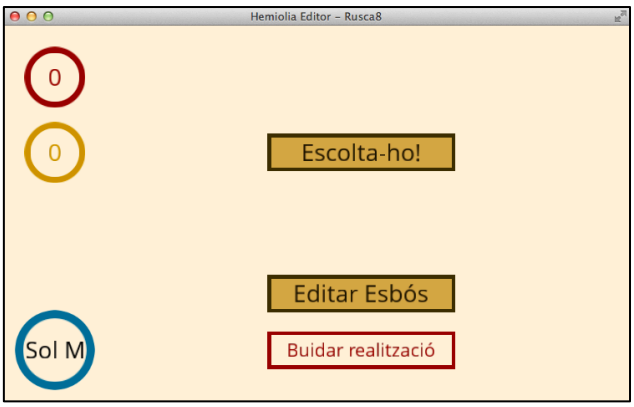

Figura 47: Menú de la pantalla de la realització.

La reproducció d'àudio aturarà les funcions de tots els botons, i mostrarà la pantalla amb els pentagrames mentre l'esquema està sonant. A més, durant la reproducció l'aplicació anirà ressaltant l'acord que sona en cada moment, per tal de facilitar-ne el seguiment a l'usuari (Figura 48).

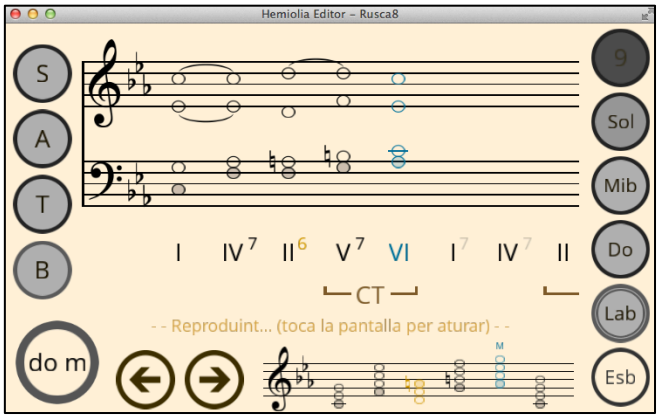

Figura 48: Estat de la pantalla durant la reproducció de l'esquema.

D'altra banda, el menú d'aquesta pantalla ens permetrà consultar els motius de les diferents alertes i errors que el programa hagi anat trobant al nostre esquema, mostrant-nos la posició de l'acord que conté l'error i la norma que el justifica (Figura 49). De la mateixa manera, ens donarà també informació sobre les alertes generades.

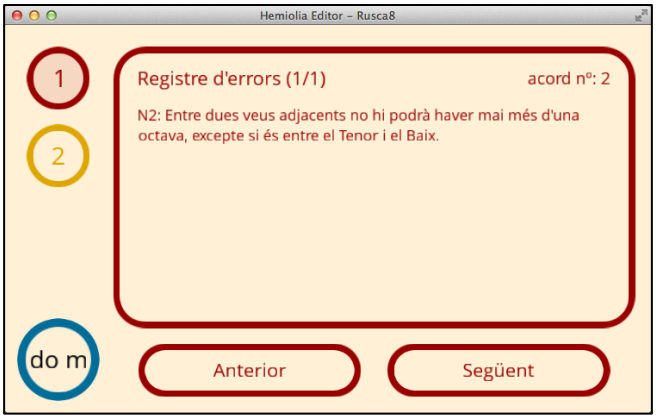

Figura 49: Menú de la pantalla de la realització mostrant el registre d'errors.

8.9.2 Objectes i variables genèrics

La declaració d'objectes i variables d'aquesta pantalla serà molt similar a la de la pantalla anterior. Dins la part genèrica tindrem com en el cas anterior la finestra, la càmera i el lot, els dos objectes **Vector3** que gestionen els clics, i un bon grapat de mides i posicions que ens serviran per gestionar fàcilment els canvis en la interfície gràfica. En aquest cas, gran part de les mides són el resultat de multiplicar les mides de la pantalla de l'esbós per una ràtio (d'aquesta manera, aprofito les proporcions entre elements gràfics que han funcionat a la pantalla de l'esbós, i les repeteixo més petites a la pantalla de la realització).

En aquest apartat, com en el cas anterior, també tindrem les variables relacionades amb la funció que s'encarrega del desplaçament.

8.9.3 Objectes i variables relacionats amb la correcció

Seguint en la mateixa línia de la primera pantalla, tindrem a continuació les variables que s'encarreguen de fer possible la correcció. En aquest cas, tindrem dues variables que s'encarreguen de marcar quina és la veu actual (**vAct**), i quin és l'acord actual (**aAct**), de manera que la intersecció d'aquestes serà la posició de la matriu a la qual estarem entrant notes en un moment concret.

La informació sobre l'esquema, doncs, la desarem en una matriu (d'una manera genèrica que no depèn de la tonalitat, de manera que ens estalviarem desar informació redundant quan l'aplicació permeti desar els esquemes). Aquesta matriu s'anomena **esquema**, i està formada dels elements **Desplegat** descrits a l'apartat 8.2.4 (pàg. 60).

Un cop passades aquestes variables, trobarem una llarga llista de vectors de diferents tipus d'objectes i variables, dels quals alguns recolliran informació que ens cal per poder corregir, mentre que altres desaran la correcció pròpiament dita.

Com en el cas de l'altra pantalla, cada tipus d'error es desa en un vector (**Array**) diferent, per facilitar-ne la posterior consulta. En el cas d'aquesta pantalla, la majoria dels vectors de correcció venen formats per elements **corPuntual**[117].

En aquest apartat, a més, també desarem els punts en què cal lligadura, i els punts en què cal evitar un solapament.

D'altra banda, tindrem una segona matriu encarregada de recollir la traducció de la matriu **esquema** a notes reals (per la correcció sí que ho necessitem), i tindrem també les variables que ens diuen si podem fer servir els botons de les notes 7a i 9a, que alguns acords no els tindran (el càlcul, com a l'altra pantalla, es realitza amb les funcions de correcció).

Finalment, tancant aquest apartat tindrem un parell de llistes més que recullen les justificacions dels diferents errors i alertes que haguem anat trobant al llarg de l'esquema.

8.9.4 Objectes i variables relacionats amb els botons

Com en el cas de l'altra pantalla, tindrem a continuació les diferents variables i objectes que ens calen per gestionar els botons.

En aquest cas, les dues columnes de botons a esquerra i dreta ens interessarà tractar-les com un conjunt, així que s'ha creat un **Array** de botons per cadascuna (**bVeus**, i **bNotes**). D'altra banda, la resta de botons s'han creat com botons individuals.

A més, tornarem a tenir els booleans **clickBot** i **calCorregir**, i els botons corresponents al menú, així com la variable que marca si l'estic mostrant o no.

D'altra banda, com que en aquest cas el menú pot mostrar coses (i. e. podem demanar que ens ensenyi els registres d'errors i alertes) tindrem un parell de variables booleanes que ens indiquen si aquestes finestres dels registres estan tancades o bé obertes, així com les variables i botons que necessitem per navegar dins aquestes.

[117] Veure apartat sobre els "elements de correcció puntual" (pàg. 64).

Finalment, tindrem les variables i objectes necessaris per gestionar la comanda d'esborrar-ho tot, així com un petit apartat amb tot el que necessitem per gestionar la reproducció d'àudio.

8.9.5 Constructor de la pantalla

Com en el cas anterior, al constructor hi tindrem una inicialització de les diferents variables i botons, així com algunes proves de funcions **H** que no s'havien provat a la pantalla de l'esbós.

8.9.6 Render

La gestió del dibuix en aquest cas torna a ser similar a la de la pantalla de l'esbós. En aquest cas, però, haurem de tenir en compte que s'ha de dibuixar tant l'esquema com l'esbós de la pantalla anterior.

En tot cas, començarem amb la comprovació que tenim una **delta** acceptable, i posteriorment netejarem el color de fons i ajustarem la càmera i el lot.

Un cop aquí, executarem la correcció en cas que sigui necessari, i a continuació començarem a dibuixar.

La funció començarà dibuixant els pentagrames, claus i armadures, i a continuació anirà acord per acord dibuixant les diferents notes de l'esquema i les diferents correccions que siguin necessàries, així com les notes de l'acord equivalent sobre l'esbós, i el xifrat dels acords de l'esquema.

Un cop acaba el dibuix de la seqüència d'acords, afegeix els dibuixos de les cadències, i després de posar algunes cortines[118] comença amb els dibuixos de botons.

Tota aquesta primera part dels dibuixos només es tindrà en compte si no estem mostrant el menú, i en cas que aquest s'estigui mostrant es dibuixarà en canvi una segona secció. En aquesta, tindrem els diferents botons del menú, així com la part sobre els registres d'errors i alertes de què hem parlat anteriorment.

D'altra banda, en un tercer tall de la funció **render** trobarem el dibuix dels ingredients i altres elements de la pantalla de les referències, que només es dibuixaran si portem prou estona mantenint premut el botó del menú.

Finalment, l'últim tall de la funció s'encarregarà en primer lloc de passar la **delta** a la biblioteca **A** per tal d'actualitzar les animacions, i en segon lloc d'actualitzar l'estat de la càrrega o reproducció d'àudio en cas que aquestes s'estiguin produint. Aquest últim apartat relacionat amb l'àudio funcionarà de la manera següent:

Si estem carregant els sons, es demanarà a l'**AssetManager** que carregui una mica del que li queda pendent, i si aquest ha acabat marcarem que ja no estem carregant sons i autoritzarem la reproducció.

D'altra banda, si l'esquema s'està reproduint, es comprovarà quants segons han passat des de l'últim acord que s'ha reproduït, i si ha passat prou estona es llançarà el següent (sempre que en

[118] Panells rectangulars del color del fons que em serveixen per tapar la part dels pentagrames que queda sota la zona on es troben els botons.

quedi algun més amb notes introduïdes). En cas que ja no en quedin més, es marcarà que ja no estem escoltant, i es retornarà a l'estat anterior a la reproducció (i. e. es recuperarà l'acord actual que teníem abans de reproduir, i es tornarà al menú).

8.9.7 Gestió d'entrades

La gestió de les entrades en el cas de la pantalla de la realització pràcticament és la mateixa que en el cas de la pantalla de l'esbós, així que ens centrarem només en els canvis més significatius.

En aquest cas, a diferència de l'esbós, tenim una part més important dels botons gestionada des de **touchUp**: per gestionar les alteracions opcionals, l'usuari aguanta premut el botó de la nota que les admet i les diferents opcions apareixen a l'esquerra, deixant que l'usuari arrossegui el dit i el deixi anar a l'opció escollida.

D'altra banda, tindrem un canvi important també en la gestió del desplaçament, ja que voldrem moure a la vegada la realització i l'esbós. En aquest cas, el que s'ha fet ha estat donar un valor de desplaçament a l'esbós segons el valor de desplaçament que té la realització. Així doncs, quan s'arrossega la realització l'esbós es mou de manera proporcional. De la mateixa manera, quan la realització s'ajusta per mostrar un acord fora el camp de visió, l'esbós fa el mateix de manera independent.

8.9.8 Resize i show

Com en el cas de l'altra pantalla, la funció **resize** recollirà els canvis en la mida de la pantalla (posant-los a les variables de la pantalla de l'esbós), i arreglarà les posicions **x** i les amplades (**w**) dels botons que ho necessitin per tal de respondre a la nova amplada fictícia de la pantalla.

A més, com en la pantalla de l'esbós, tindrem la funció **posaElsBotonsPer**, encarregant-se de les posicions de botons que depenen de l'estat de la pantalla.

Aquesta funció la podrem veure també funcionant al principi de la funció **show**, encarregant-se de posar els botons mòbils a la seva posició i amplada per defecte. A més, com amb l'altra pantalla, es posaran a zero les variables que puguin haver quedat alterades en sortir de la pantalla per anar a veure'n una de diferent.

En aquest cas, a més, tindrem una part de la funció **show** que s'executarà només si estic creant un nou esquema (si és la primera vegada que entro a la pantalla), i que s'encarregarà d'inicialitzar-lo.

D'altra banda, com en el cas de la pantalla anterior, realitzarem una primera correcció per assegurar-nos que tenim totes les dades que puguem venir a buscar, i ajustarem també les variables de desplaçament que puguin haver estat alterades.

8.9.9 Pause, resume, hide i dispose

Com en el cas anterior, la pantalla necessita de certs mètodes que en realitat nosaltres de moment no farem servir. Tot i això, en aquesta pantalla sí que utilitzarem el mètode **hide**, ja que ens interessarà alliberar els recursos dels sons que haguem carregat en cas que marxem de la pantalla.

8.9.10 Funcions de càlcul i correcció

Finalment, en un últim apartat dins la pantalla tindrem les diferents funcions que s'encarreguen de dur a terme càlculs útils, així com la correcció de l'esquema.

La primera que trobarem serà **posaElsBotonsPer**[119], i tot seguit trobarem la que s'encarrega d'inicialitzar l'esquema dins el mètode **show**.

Després d'aquestes, tindrem la funció **permisosBotons**, que calcula si puc utilitzar els botons de 7a i 9a, i si hi ha alguna alteració opcional a l'esquema (i en cas que n'hi hagi quina és).

A continuació, tindrem la funció que s'encarrega d'agafar la informació de l'esbós (quins graus he posat), la informació de la tonalitat i la informació de l'esquema (quina nota de l'acord he posat a cada veu[120]), i traduir-la a una matriu d'objectes tipus nota (**Note**) per tal que la correcció pugui comprovar intervals i altres dades que amb les variables originals serien enrevessades de calcular. Aquesta funció s'anomena **tradueixNotes**.

I després d'aquesta traducció podem trobar la funció de correcció (**corregeix**). Aquesta s'encarregarà de recórrer l'esquema en horitzontal i en vertical buscant errors, així com de comprovar les cadències, duplicacions, supressions i inversions, i l'adequació a les tessitures, entre d'altres. A més, en acabar la correcció es realitzarà un recompte d'errors i alertes per tal de poder informar a l'usuari de quantes de cada s'han trobat, a la pantalla del menú (juntament amb els registres).

Finalment, abans d'acabar el codi de la pantalla podrem trobar una funció que comprova si tinc correcció **corPuntual** per a un paràmetre, acord i veu concrets (p. ex. tenor del segon acord, preparació de sèptimes), i que alhora desa la correcció trobada en un objecte dins la biblioteca **F** per facilitar-ne l'accés a continuació. Aquesta funció serà de gran utilitat a l'hora de dibuixar els resultats de les diferents correccions dins la funció **render**.

8.10 Pantalla dels ingredients (PIng)

La *"pantalla dels ingredients"* (**PIng**), s'ha plantejat en realitat més aviat com un conjunt de pantalles íntimament relacionades. Si bé a efectes de codi el programa la tractarà com una única pantalla, tindrem contingudes dins aquesta les pantalles que gestionaran la tria d'opcions per a cada tipus d'ingredient, la tria de quin tipus d'ingredient volem afegir, i la llista sencera dels ingredients que s'han triat per a l'enunciat.

Per tal de mantenir la unitat de compilació organitzada, s'han anat dividint les diferents parts del codi d'una manera clara, i s'ha mirat de mantenir el mateix ordre en les divisions per a tots els mètodes implicats (p. ex. s'ha dividit i ordenat de la mateixa manera la definició de variables, la funció **render** i la funció **touchDown**). A més, com s'ha comentat anteriorment, s'ha aprofitat l'existència de comentaris tipus **/** */**, per a marcar amb claredat les diferents seccions.

8.10.1 Disseny gràfic i interfície d'usuari

El primer que veurem en arribar a la pantalla dels ingredients (Figura 50) serà un cercle amb la tonalitat actual i un espai buit preparat per a ser omplert amb els requeriments que s'escullin.

[119] Veure "Resize i show" a la pantalla de l'esbós (pàg. 85).
[120] Veure la classe "Desplegat" a les definicions d'objectes musicals (apartat 8.2.4, pàg. 60).

A més, a l'espai inferior trobarem un seguit de botons rodons que ens serviran per afegir ingredients, editar-los, enllaçar-los i esborrar-los, així com un botó allargat que s'encarregarà de donar per acabada la tria de l'enunciat i saltar a la pantalla de l'esbós per tal de continuar l'exercici.

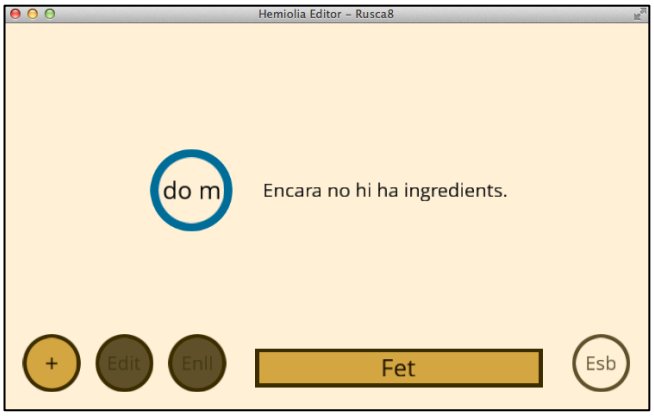

Figura 50: Estat inicial de la pantalla dels ingredients.

Si afegim alguns ingredients a la pantalla, podrem veure com una clau ens engloba una llista de botons representatius dels diferents ingredients que s'hagin afegit fins ara (Figura 51). Com es pot veure, cadascun d'aquests botons podrà ésser seleccionat per tal d'activar a la part inferior els botons que permeten realitzar accions sobre un ingredient ja afegit.

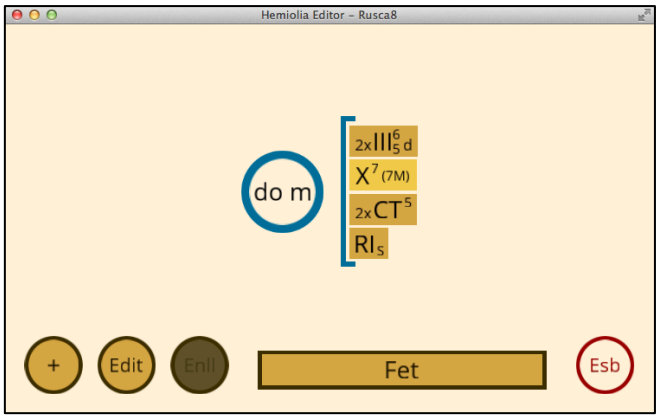

Figura 51: Pantalla dels anterior després d'haver afegit alguns ingredients.

8.10.1.1 Pantalla de la tria d'ingredients

Pel que fa a la tria d'ingredients en sí, tan bon punt es premi el botó "+" s'obrirà una primera pantalla que permetrà escollir quin dels tres tipus d'ingredient que permet l'aplicació voldrem afegir. Aquests tres tipus, com es pot veure a la (Figura 52) seran Acord, Cadència i Resolució Irregular.

En termes d'interfície cal destacar que s'ha ubicat més a la dreta l'ingredient que apareixerà més sovint, per tal de facilitar la seva tria amb el polze dret després d'haver activat la pantalla amb el polze esquerre.

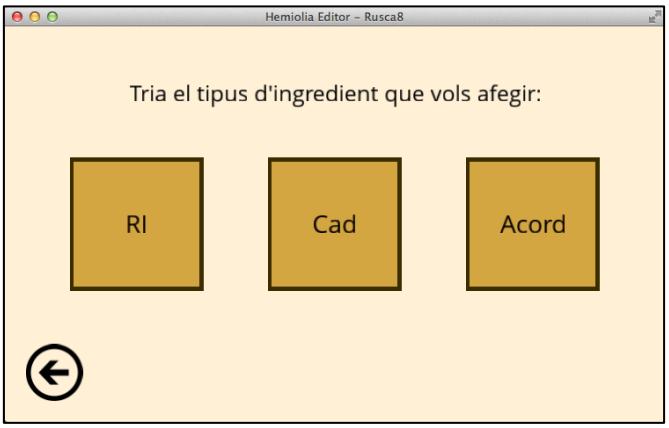

Figura 52: Pantalla on es realitzarà la tria del tipus d'ingredient.

8.10.1.2 Pantalla dels ingredients de tipus acord

A partir d'aquest punt tindrem tres possibilitats pel que fa als ingredients, i per tant tres pantalles diferents que recolliran les opcions que podrem demanar per a cadascun d'ells.

La pantalla més important d'entre aquestes serà la que respon als ingredients de tipus acord (Figura 53). Com podem veure, gran part de les opcions d'aquesta pantalla coincidiran amb les que podíem escollir a la pantalla de l'esbós (opcions que ja té per defecte la classe **Chord**), mentre que la resta dels botons que hi trobarem responen a algunes[121] de les variables que es van plantejar en el moment de definir aquest tipus d'ingredient[122].

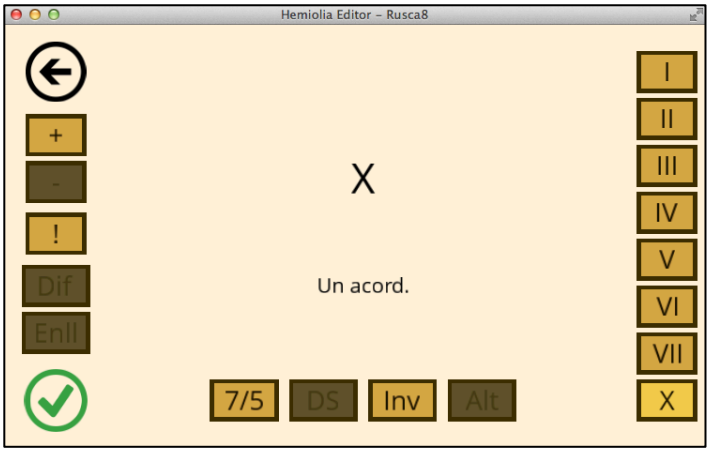

Figura 53: Pantalla de tria d'opcions per als ingredients de tipus acord (buida).

D'altra banda, a més de les opcions hi podrem veure els botons corresponents a la funció de tornar enrere i a la funció de desar l'ingredient, així com un codi similar al xifrat que ens descriurà de forma compacta l'ingredient, que vindrà complementat per una descripció textual d'aquest. Podem veure un exemple d'ingredient (en aquest cas bastant enrevessat) a la Figura 54.

Com s'ha comentat a l'apartat sobre els ingredients tipus acord (**IngAcord**), el valor **X** s'utilitzarà dins els ingredients per poder indicar que l'ingredient no implica cap restricció pel que fa al grau.

[121] La resta d'aquestes funcions extra simplement s'han afegit dins els botons que ja existien.
[122] Veure "Definició dels ingredients tipus Acord (IngAcord)" (pàg. 61).

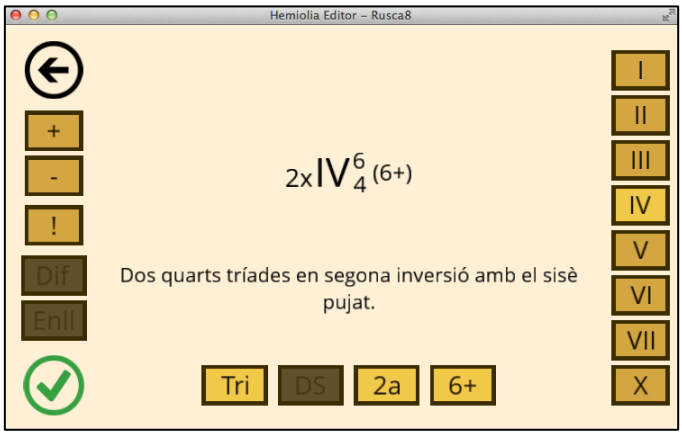

Figura 54: Pantalla anterior després d'escollir opcions per un ingredient de tipus acord.

De nou, a nivell d'interfície d'usuari s'ha apropat els botons a les vores del telèfon més accessibles per als polzes. D'altra banda, en aquest cas cal destacar que s'ha ubicat el botó de descartar l'ingredient (i. e. la fletxa de tornar enrere) en un punt el més allunyat possible de la majoria de botons per tal d'evitar que es premi sense voler. A més, sovint el primer que indicarem en un ingredient serà la quantitat, de manera que posar els botons de quantitat a prop de la fletxa en general no suposarà un perill (no es perdria pràcticament res en cas que la fletxa es premés per accident intentant prémer el "+").

8.10.1.3 Pantalla dels ingredients de tipus cadència

En segon lloc pel que fa a la importància i complexitat de l'ingredient, tindrem la pantalla per a triar les cadències. En aquest cas, deixant de banda els botons de la barra esquerra (que seran equivalents als dels ingredients de tipus acord), tindrem només un botó d'opcions: el que decidirà si volem la cadència trencada específicament tríade, específicament amb sèptima, o si ens serveix de qualsevol de les dues maneres.

Podem veure com quedarà el disseny d'aquesta pantalla a la Figura 55.

Com amb el cas dels acords, trobarem al centre de la pantalla una descripció codificada de l'ingredient, així com una descripció completa en forma de text.

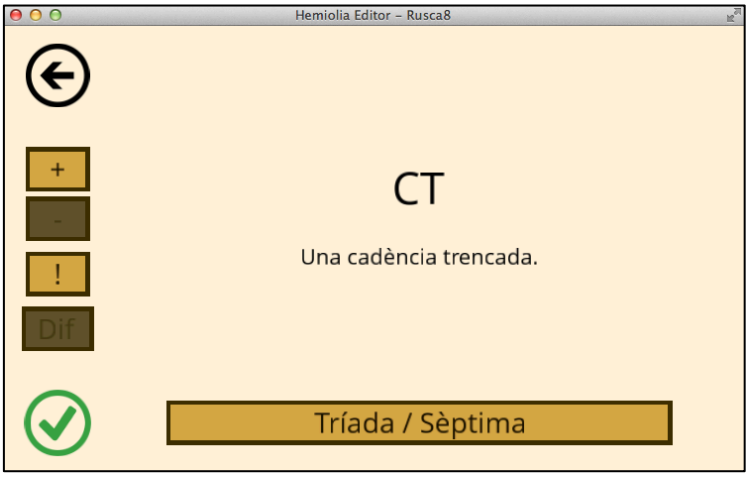

Figura 55: Pantalla de tria d'opcions per als ingredients de tipus cadència.

8.10.1.4 Pantalla dels ingredients de tipus resolució irregular

Finalment, en tercer lloc tindrem la pantalla que ens servirà per a escollir les resolucions irregulars.

Degut que en aquest cas s'ha considerat que no valia la pena indicar quantitats i que aquest tipus d'ingredient només consta d'un paràmetre rellevant a nivell d'opcions, la pantalla serà la més senzilla de totes: simplement tindrà un botó gran per a cadascun dels dos valors del paràmetre, i un tercer botó amb la fletxa que hem anat veient per tornar enrere sense desar res (Figura 56).

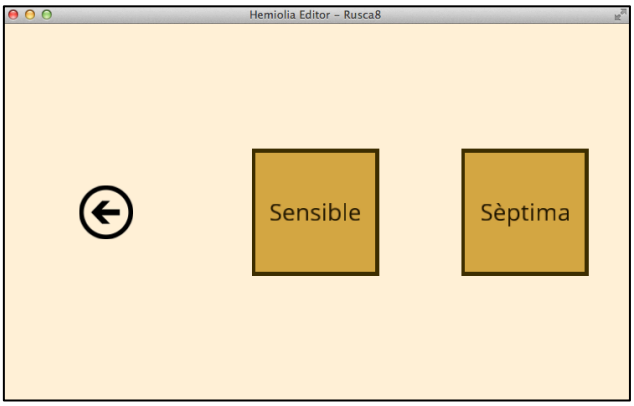

Figura 56: Pantalla per a la tria del tipus de resolució irregular que volem demanar.

Així doncs, en aquest cas no ens caldrà cap mena de descripció de l'ingredient, i podrem desar l'elecció directament en el moment de fer-la (tan bon punt s'esculli una de les dues opcions, es considerarà que s'ha confirmat la tria de l'ingredient i es tornarà a la pantalla de la llista).

8.10.2 Estructura interna de la pantalla

En aquest cas, abans de veure els diferents mètodes de que consta la pantalla, veurem primer quina ha estat la ordenació de les coses dins d'aquests pel que fa a la divisió de la pantalla en pantalles més petites diferents.

En totes les funcions que han necessitat participar d'aquesta divisió primer s'ha posat la part relacionada amb la tria del tipus d'ingredient (Figura 52), a continuació les parts relacionades amb cadascun dels ingredients, i finalment la part que té a veure amb la llista d'ingredients (Figura 50).

Pel que fa a l'ordre dels diferents ingredients, primer trobarem el codi relacionat amb la pantalla de les resolucions irregulars (Figura 56), en segon lloc el de la pantalla de les cadències (Figura 55), i finalment el dels ingredients tipus acord (Figura 53).

A més, en cas que calgui hi haurà un apartat final amb la part del codi que pugui ser compartida entre pantalles.

8.10.3 Objectes i variables

Deixant de banda el lot, la finestra i la càmera, el primer que trobarem en aquesta pantalla seran tres parelles de llistes, cadascuna dedicada a desar un dels tipus d'ingredient. Dins de cada parella, doncs, tindrem una llista d'ingredients del tipus que pertoqui, així com una llista del tipus de botó que s'associa amb l'ingredient en qüestió (p. ex. a la parella sobre tipus acord tindrem un vector (**Array**) d'**IngAcord**, i un segon vector de **BotIA**).

Aquestes dues llistes seran el que ens permetrà, per a cada tipus d'ingredient, desar les diferents tries que faci l'usuari per a l'enunciat del seu exercici. Finalment, abans de començar amb les seccions esmentades a l'apartat anterior, trobarem una cadena de text encarregada de recollir les descripcions dels diferents ingredients, anomenada **desc**.

A continuació, dins la part de la tria del tipus d'ingredient, senzillament trobarem tres botons, un per a cadascun dels tipus, que seran els que ens portaran a cadascuna de les pantalles d'aquests. S'han anomenat, segons l'ordre ja comentat, **bTipusR**, **bTipusC** i **bTipusA**.

Pel que fa als apartats dels ingredients en concret, bàsicament tindrem un seguit de botons i valors encarregats de gestionar les diferents opcions que aquests permetin, així com una parella d'objectes auxiliars dels mateixos tipus que els continguts a les parelles de llistes ja esmentades (i.e. per al tipus acord tindrem un **IngAcord** auxiliar i un **BotIA** auxiliar). Aquestes parelles d'objectes auxiliars ens serviran per desar la referència als elements de la llista amb què estiguem treballant per tal d'evitar haver de realitzar una cerca dins la llista cada vegada que en vulguem canviar un paràmetre.

A més, per a cadascun dels ingredients trobarem també alguns valors relacionats amb la ubicació dels elements dins la pantalla, així com alguns booleans relacionats amb l'activació i desactivació dels botons segons el context (com en el cas de les altres pantalles ja treballades).

Pel que fa a la pantalla de la llista, el concepte serà similar, així com ho seran les variables i objectes compartits entre pantalles.

Finalment, trobarem dins la declaració d'objectes i variables els diferents booleans que ens indicaran quina pantalla s'està mostrant en cada moment.

8.10.4 Constructor

En aquest cas, a diferència dels anteriors, no s'ha comprovat cap funció de les biblioteques, però sí que de nou tindrem al constructor les inicialitzacions.

8.10.5 Render

En aquest cas, degut que no s'haurà de realitzar cap mena de correcció ni s'ha aplicat cap actualització de les animacions, el mètode **render** simplement consistirà en un seguit de seccions (marcades per l'ordre escollit a l'estructura de la pantalla) cadascuna dedicada a dibuixar els diferents elements de la pantalla que representa.

8.10.6 Resposta als esdeveniments d'entrada

Pel que fa als esdeveniments d'entrada, en aquesta pantalla només ha calgut implementar la funció **touchDown**. Com en els altres casos, aquesta funció estarà subdividida segons l'estructura escollida, i el codi que contindrà serà similar al que podem trobar explicat a la resta de les pantalles ja comentades.

En aquest cas, malgrat no haurem de marcar que cal corregir, sí que caldrà en segons quins casos demanar que es generi una nova descripció de l'ingredient actual (així com que s'actualitzin el text i la mida del botó associat).

8.10.7 Resize i show

Com en els altres casos, el mètode **resize** contindrà un seguit d'ajustos de les posicions i mides dels botons per tal d'adaptar-los a les noves mides de pantalla. En aquest cas, però, tornarem a tenir els ajustos dividits segons les seccions escollides. D'altra banda, degut que tindrem diversos canvis de posició d'alguns botons compartits (principalment el de tornar enrere), s'ha utilitzat també per a gestionar-los una funció com la comentada a les altres pantalles (**posaElsBotonsPer**).

Pel que fa al mètode **show**, tornarem a tenir la línia que indica a l'aplicació que els esdeveniments d'entrada s'han de gestionar des d'aquí, i un parell de funcions ubicant les coses al seu lloc per si venim d'una altra pantalla.

8.10.8 Pause, resume, hide i dispose

En aquest cas, com a la pantalla de l'esbós, les funcions extra que demana la pantalla de LibGDX no s'han utilitzat.

8.10.9 Funcions de càlcul i posicionament

En un últim apartat dins el codi de la pantalla trobarem les diferents funcions que s'encarreguen de posicionar els elements (algunes s'han comentat ja als apartats anteriors), així com les que s'encarreguen de generar la descripció de l'element actual o determinar quins dels botons d'opcions han d'estar disponibles en cada moment.

Així doncs, el primer que trobarem serà l'equivalent per aquesta pantalla de la funció **permisosBotons** que ja havíem vist a les altres pantalles. En aquest cas, però, tindrà un argument que ens permetrà dir-li amb quin tipus d'ingredient estem treballant, de manera que faci els càlculs per als botons que ens interessen.

En segon lloc, trobarem una funció que s'encarrega de posicionar el cercle de la tonalitat, així com els diferents botons dels ingredients escollits a la pantalla que en mostra la llista (i.e. centra la llista d'ingredients a l'eix vertical i centra el conjunt "ingredients + cercle" a l'eix horitzontal). Aquesta funció s'ha anomenat simplement **posiciona**.

A continuació, trobarem la funció **posaElsBotonsPer**, que ja hem comentat amb anterioritat, i tot seguit la funció que genera les descripcions dels ingredients a partir dels paràmetres que s'hagin escollit per a aquests (anomenada **descriu**). En aquesta funció és on aprofitarem la majoria dels mètodes relacionats amb la gestió del llenguatge[123] desenvolupats a la biblioteca **F**.

Finalment, tancant la unitat de compilació tindrem un mètode encarregat de retornar de quin tipus és un element en funció del seu número identificador únic[124], anomenat **tipusDe**. Aquest mètode ens serà útil en el moment d'esborrar i editar ingredients, ja que la variable que ens marca quin ingredient s'ha seleccionat (**ingAct**) només recull l'identificador d'aquest. Així doncs, degut que els ingredients poden ser de diferents tipus (i que això implica diferents objectes), abans de fer res ens interessarà esbrinar amb quin tipus d'objecte haurem de tractar.

[123] Veure apartat 8.5.2.3 (pàg. 73).
[124] Variable id descrita a la definició dels ingredients de l'enunciat (pàg. 60).

Podem veure la utilització que s'ha fet d'aquesta funció a la part d'editar ingredients del codi de la funció **touchDown**.

8.11 Pantalles menors

Envoltant aquestes tres pantalles principals que hem vist tindrem un seguit de pantalles molt més senzilles implementades amb propòsits molt més específics. Aquestes seran la pantalla inicial (**PInici**), la pantalla del menú principal (**PMenu**), la pantalla de la secció "Quant a..." (**PAbout**) i la pantalla de la tria del mode[125] (**PTon**).

Degut que gran part del contingut d'aquestes pantalles respon a les mateixes estructures i filosofies emprades en les pantalles que ja hem vist, ens limitarem a veure'n el disseny gràfic escollit i només les parts rellevants que hi pugui haver dins el codi, sense passar per tots els apartats que ja s'han comentat extensament a les pantalles anteriors.

8.11.1 Pantalla inicial (PInici)

La pantalla inicial (**PInici**) serà la primera pantalla que s'obrirà, i per tant la primera pantalla que veurà l'usuari (la unitat central la crea i l'activa en el moment d'obrir-se l'aplicació).

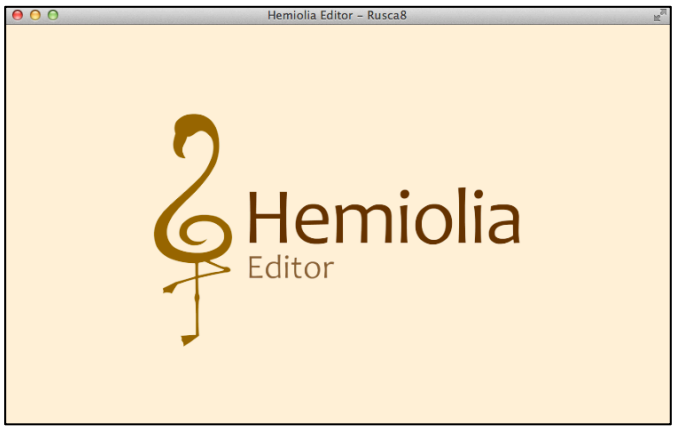

Figura 57: Pantalla inicial mostrant el logotip dissenyat per a l'aplicació.

L'objectiu d'aquesta pantalla serà, doncs, mostrar el més aviat possible alguna cosa que indiqui a l'usuari que l'aplicació s'està carregant (evitant així deixar la pantalla negra els segons que pugui durar el procés).

La imatge escollida per a tal propòsit ha estat simplement un logotip sobre un fons pla (Figura 57), i ha estat dissenyada expressament per a l'aplicació.

8.11.1.1 Aspectes a destacar del codi

Pel que fa al codi, en aquest cas s'ha substituït la finestra de tipus **ExtendViewport** per una finestra de tipus **FitViewport**. Aquesta, bàsicament el que fa és engrandir el món fins que toca amb el límit de la pantalla, i a continuació centrar-lo i deixar del color del fons les barres que sobrin als laterals.

[125] Anomenada ja amb la idea d'acabar-hi traslladant en un futur la tria completa de la tonalitat.

Així doncs, el fet d'escollir per a la pantalla una imatge amb el fons pla[126] ens permet dibuixar-la exactament de la mateixa mida que té el món i deixar que la finestra s'encarregui de centrar-la i ampliar-ne el color del fons fins a la mida de la pantalla, fent que sembli una imatge feta a mida amb el logotip centrat.

D'altra banda, la pantalla funcionarà de la següent manera: el constructor carregarà la imatge del logotip, i el primer fotograma la mostrarà per pantalla. A partir d'aquí, el segon fotograma demanarà la càrrega de la resta de recursos (les imatges, les fonts i les normes d'harmonia), així com la inicialització de les tres pantalles principals que havíem declarat a la unitat central. Un cop aquests s'hagin carregat[127] mirarà que s'hagi mostrat la imatge al menys mig segon, i si està tot en ordre saltarà a la pantalla següent.

En aquest procés, primer alliberarà els recursos que hem utilitzat en aquesta pantalla (els de la imatge i el lot), i a continuació saltarà a una pantalla del menú (**PMenu**) que crearà en el mateix moment del salt.

8.11.2 Pantalla del menú principal (PMenu)

La pantalla del menú (**PMenu**) bàsicament consta de dos grans botons que ens permetran, respectivament, començar l'exercici i consultar la secció "Quant a...". A l'esquerra d'aquests s'ha mantingut una imatge amb el logotip sense lletres a mode d'ambientació (Figura 58).

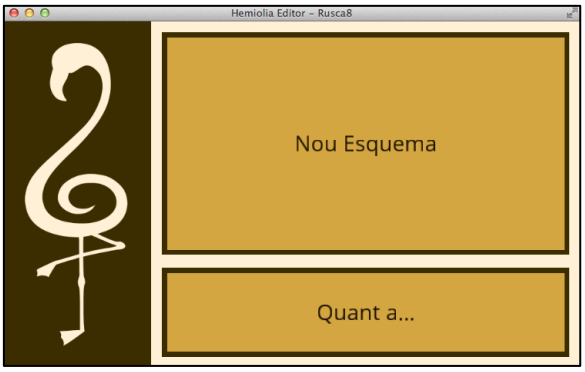

Figura 58: Pantalla del menú principal.

8.11.2.1 Aspectes a destacar del codi

Com a aspecte a destacar del codi val a dir que s'ha creat una funció anomenada **posiciona**, que s'encarrega d'ubicar ambdós botons en funció d'uns pesos que se'ls ha assignat.

D'altra banda, com en el cas de la pantalla inicial, el salt produït en prémer un dels botons implicarà l'alliberament dels recursos d'aquesta pantalla i la creació de la pantalla a la qual saltarem en el mateix moment de saltar-hi.

[126] De fet, transparent, en aquest cas, per deixar que s'ompli amb el color escollit per al fons.
[127] Aquest tipus de càrrega succeeix de manera síncrona (aturant l'aplicació fins que la càrrega ha acabat). És per això que ens interessa renderitzar abans el primer fotograma, de manera que no deixi congelada una pantalla negra sinó la imatge que volem mostrar.

8.11.3 Pantalla de la tria del mode (PTon)

Degut que els ingredients disponibles per a un enunciat són diferents segons es tracti d'un exercici en mode Major o d'un en mode menor, s'ha implementat una pantalla prèvia a la tria d'ingredients a la qual podrem escollir el mode de l'exercici que farem (Figura 59). Aquesta bàsicament tindrà una gran divisió enmig, i saltarà a la pantalla dels ingredients tan bon punt s'esculli una de les dues opcions (després de deixar marcada l'opció escollida).

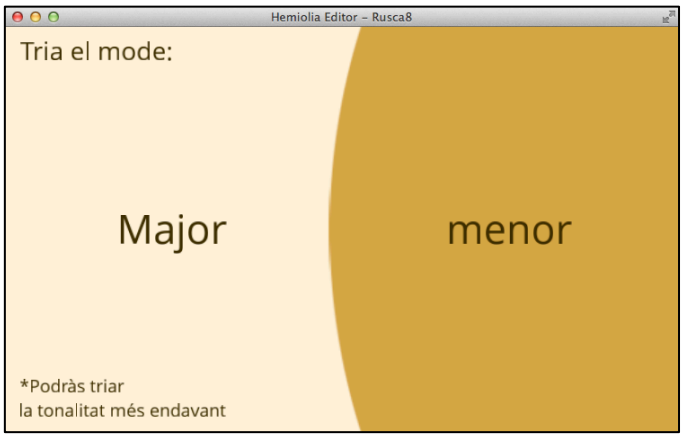

Figura 59: Pantalla de la tria del mode.

Pel que fa al disseny, s'ha escollit separar ambdues opcions per un arc de cercle en referència a la divisió entre les tonalitats majors i les tonalitats menors en el cercle de quintes (Figura 2, pàg. 17).

D'altra banda, en futures iteracions en el desenvolupament de l'aplicació, es preveu escollir també en aquesta pantalla la tonalitat (a més del mode), a partir d'una interfície gràfica que recordi també al cercle de quintes, de manera que preparar la pantalla ja en aquesta direcció servirà per estalviar feina posterior.

8.11.3.1 Aspectes a destacar del codi

En aquesta pantalla, degut que el salt cap a la pantalla següent ja va a parar a una de les pantalles que tenim inicialitzades des del principi, no ens caldrà crear la pantalla a la qual saltarem. Tot i això, sí que ens interessarà alliberar els recursos de la pantalla en què ens trobem, ja que no hi tornarem en cap moment (i en tot cas hauríem de crear-la de nou perquè no en tindríem cap referència).

8.11.4 Pantalla de la secció "Quant a..." (PAbout)

Aquesta serà l'última i també la més complexa de les pantalles menors. En aquest cas, tindrem una pantalla amb tres apartats diferenciats, accessibles mitjançant tres botons circulars laterals, a mode de pestanyes. Aquestes tres seccions separaran la informació que s'oferirà a la pantalla segons si té a veure amb el desenvolupador, amb l'aplicació o amb els recursos.

Podem veure l'aparença de la pantalla a la Figura 60.

Bàsicament, tindrem una mica de context respecte al desenvolupador dins la primera, una mica de context respecte a l'aplicació a la segona, i informació respecte a l'origen dels diferents recursos utilitzats a la tercera.

A més, dins la part del desenvolupador s'han incorporat un parell de botons circulars que enllaçaran amb els comptes de GitHub i Twitter.

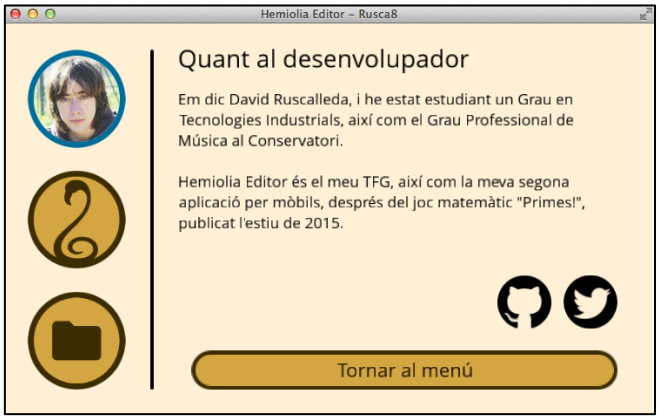

Figura 60: Pantalla de la secció "Quant a..." oberta per la primera pestanya.

D'altra banda, compartit a les tres pestanyes tindrem un botó inferior que ens permetrà tornar al menú per poder accedir de nou a la realització d'exercicis.

8.11.4.1 Aspectes a destacar del codi

Una primera diferència que trobarem en el codi respecte a la resta de pantalles és que el constructor d'aquesta requereix d'un argument extra: la referència a la pantalla del menú. Això es deu a que la pantalla del menú no la tenim ubicada dins la unitat central (ja que ens en desfarem tan bon punt haguem entrat a la realització de l'exercici).

Així doncs, necessitem tenir la referència a la pantalla del menú en algun lloc per tal de poder-la tornar a fer activa quan hi vulguem tornar, i la manera de fer-ho ha estat agafar aquesta referència a través del constructor.

I d'altra banda, una segona diferència respecte a tot el que havíem vist és que en aquesta pantalla s'utilitza l'eina que ofereix LibGDX per obrir enllaços en un navegador extern (funció **openURI**).

Aquesta, si l'aplicació està funcionant en un ordinador obrirà el navegador predeterminat, i si és en un mòbil farà aparèixer la pantalla que pregunta a l'usuari amb quin navegador vol realitzar l'acció.

9 RESULTATS

L'aplicació que hem obtingut com a resultat d'aquest procés, en principi hauria d'estar preparada per poder gestionar qualsevol de les normes que s'han comentat a l'apartat de teoria, amb un parell d'excepcions: les normes sobre la melodia (que són bastant més subjectives i requeriran una implementació posterior més complexa), i les normes sobre la segona entre el 6è i el 7è graus del menor (el programa detectarà l'interval augmentat en cas que no s'hi faci res, però no distingirà entre les solucions ascendent i descendent que distingeixen les normes).

En tot cas, deixant de banda les normes que s'hauran d'implementar en futures versions, ens interessarà saber com de bé respon l'aplicació en l'àmbit de les normes per les quals sí que ha estat preparada. Això ho podrem comprovar, per exemple, comparant algunes correccions realitzades per un professor real amb la correcció que ens ofereix l'aplicació per als mateixos esquemes.

9.1 Resposta de l'aplicació a exemples d'exercicis reals.

Així doncs, s'han seleccionat alguns esquemes corregits[128] provinents dels deures de les primeres setmanes del curs de 5è de Grau Professional del Conservatori de Música de Girona per tal d'introduir-ne la resolució a l'aplicació i comparar-ne la resposta amb les indicacions dels professors.

Per tal de comprovar tants errors com sigui possible i amb l'objectiu de veure errors bàsics que no es donaran pràcticament mai un cop entrat el curs, s'han seleccionat alguns dels primers exercicis realitzats per l'alumna, reflectint un moment en què la classe encara no havia tingut temps d'assimilar les primeres normes.

9.1.1 Exemple 1: Moviments paral·lels i cadència perfecta.

El primer exemple que veurem correspon a l'esquema realitzat a la Figura 61. En aquest cas podem veure de seguida que s'hi han trobat una gran quantitat de moviments paral·lels (quintes i octaves). D'altra banda, s'ha trobat un error en les notes utilitzades al tercer acord (el tenor tenia una nota que no formava part de l'acord), i s'havia utilitzat la sensible al tercer (la nostra versió de les normes això no ho considera un error si la realització és correcta).

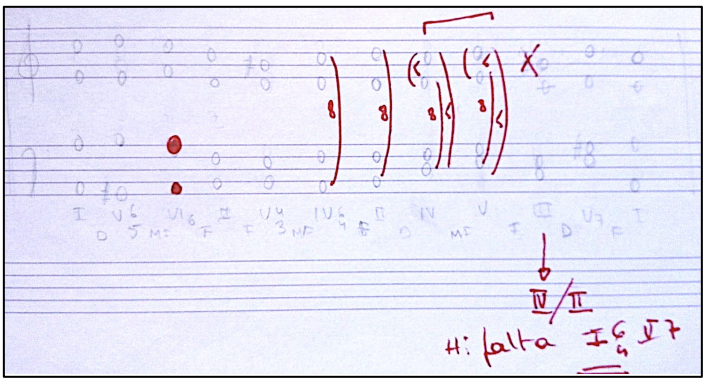

Figura 61: Realització de l'alumna i correcció de la professora per a l'exemple 1.

[128] Els esquemes dels exemples 1 i 2 han estat cedits per l'alumna Maria Pérez, i van ser corregits per la professora Imma Ponsatí (CMG, 5è, curs 2015-2016).

D'altra banda, una gran indicació a la part inferior ens mostra que no s'ha realitzat correctament la cadència perfecta del final (té el $\mathrm{V_{EF}}$ i el $\mathrm{I^5_{EF}}$, però li falta la subdominant i el $\mathrm{I^6_4}$ previs).

Si entrem la mateixa seqüència d'acords a la nostra aplicació comprovarem que aquesta també considera correcta la successió d'enllaços (Figura 62). Tot i això, el programa discrepa respecte a un dels xifrats escrits per l'alumna: el quart acord començant pel final no porta sensible, així que en realitat no és un V sinó un $\mathrm{V_m}$. Tot i això, l'ús que s'ha fet de l'acord ha estat el corresponent al $\mathrm{V_m}$ (no s'ha tractat com un acord de dominant), així que l'error és únicament degut al desconeixement de la notació corresponent a aquest cas especial. D'altra banda, pot ser que aquest error de notació se li hagi passat per alt a la professora, o que simplement no li hagi donat importància.

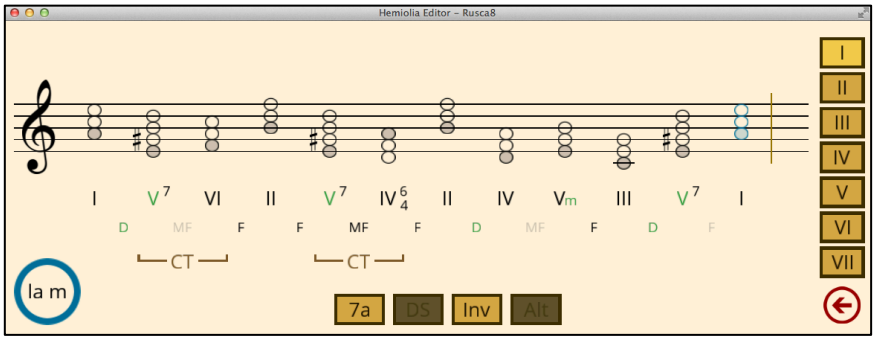

Figura 62: Esbós de l'esquema anterior introduït a l'aplicació (finestra allargada per encabir-ho tot).

D'altra banda, l'aplicació també ha trobat a faltar la cadència perfecta del final: no ha quedat marcada, i si intentem passar a la realització ens avisarà que no en tenim (Figura 63).

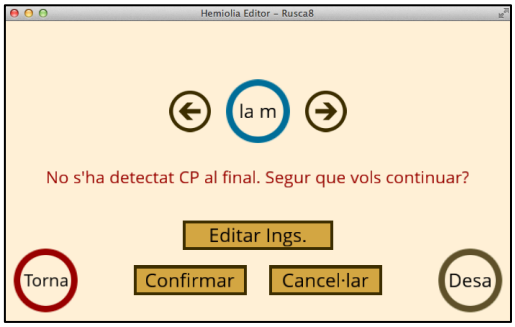

Figura 63: Pantalla del menú mostrant l'avís de manca de CP en intentar saltar a la realització.

Un cop passat a fer la realització (Figura 64) començarem a veure com l'aplicació els errors de moviments paral·lels que ja havia trobat la professora. A més, l'aplicació ha vist que el segon cas de moviments paral·lels (acords 8-9) en realitat durava tres acords, cosa que a jutjar per la correcció havia passat desapercebuda a la professora.

D'altra banda, l'aplicació ens ha marcat amb una alerta l'últim incomplet, quan en realitat és bastant habitual que ho sigui i per tant no hauria de quedar marcat.

Finalment, la nota que havíem introduït a la versió escrita que en realitat no formava part de l'acord, dins l'aplicació ha estat impossible d'escollir, de manera que ha quedat ben posada des del principi.

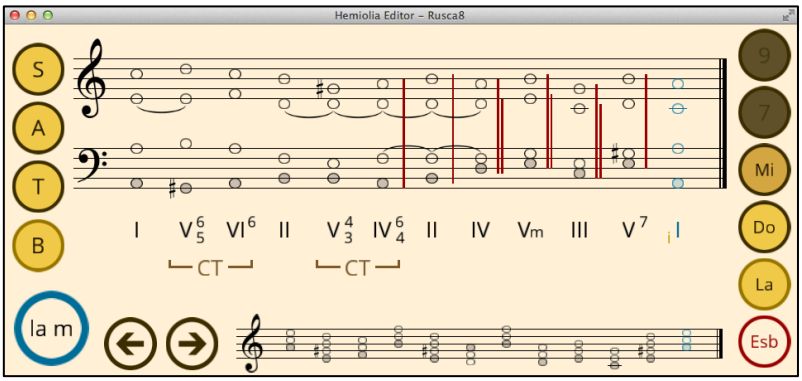

Figura 64: Mateixa realització de l'exemple 1 introduïda a l'aplicació (finestra allargada per encabir-ho tot).

D'altra banda, no tenim dades respecte al temari explicat a classe en el punt en què es va fer aquest exercici, però la melodia escollida en aquesta resolució seria bastant millorable en tant que oscil·la molta estona al voltant d'una mateixa nota (el Si_3), de manera que habitualment s'hauria afegit alguna indicació al respecte per a l'alumne. En tot cas, degut que l'aplicació encara no contempla aquestes directrius, no s'ha indicat en cap de les dues correccions.

9.1.2 Exemple 2: Sèptimes i comportament de la dominant.

En un segon cas de resolució menys dramàtica tindríem l'exemple de la Figura 65. En aquest cas, veiem que s'han marcat amb errors dues sèptimes (una degut a la preparació i una degut a la resolució), i que s'ha indicat també un error en el càlcul dels enllaços i una resolució errònia de l'acord de dominant. A més, en aquest cas sí que s'ha deixat una indicació respecte a la qualitat de la melodia: no té direcció i oscil·la tota l'estona al voltant d'una sola nota que a més resulta ser la tònica de la tonalitat (cosa que fa aquesta oscil·lació especialment redundant per a l'oient).

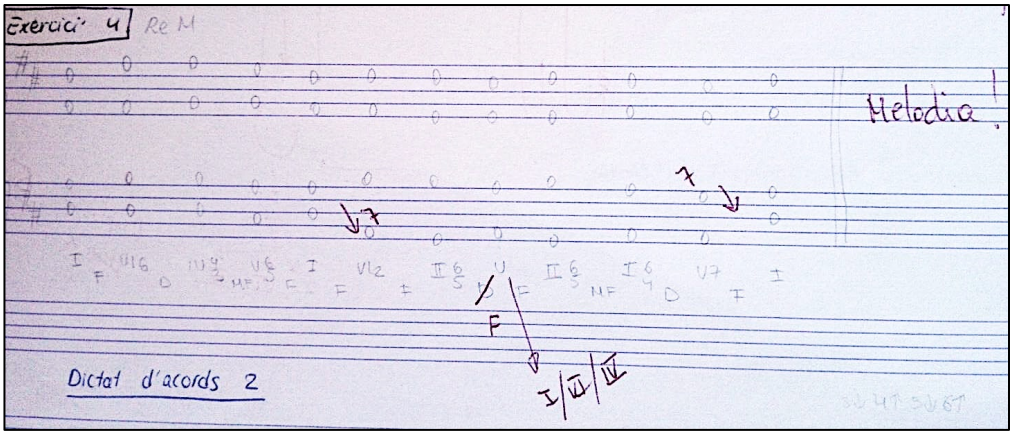

Figura 65: Realització de l'alumna i correcció de la professora per a l'exemple 2.

Si introduïm l'esbós d'aquest nou exemple a la nostra aplicació (Figura 66), el primer que veurem destacat és el penúltim acord de dominant (V). Aquest, com a la correcció de la professora, ha estat marcat amb un error degut que no va a tònica ni a un dels seus substituts. Tot i això, com a curiositat val a dir que en realitat la seqüència V-II seria una opció vàlida en cas que tot seguit retornés al V (i que malgrat no ha estat descrita a les normes, aquesta successió V-II-V és una excepció que el codi de l'aplicació de fet contempla).

En tot cas, la resolució d'aquest acord de dominant no ha estat l'esperada, i ambdues correccions ho han fet notar. D'altra banda, el càlcul dels enllaços que realitza l'aplicació ha discrepat respecte al de l'alumna en dos punts: abans i després d'aquesta dominant mal resolta. La primera d'aquestes discrepàncies ja havia estat recollida per la professora a la correcció original, però la segona havia quedat emmascarada sota l'altre error (probablement com que igualment no era l'acord que hi havia d'anar, la professora ja no es va fixar en l'enllaç calculat).

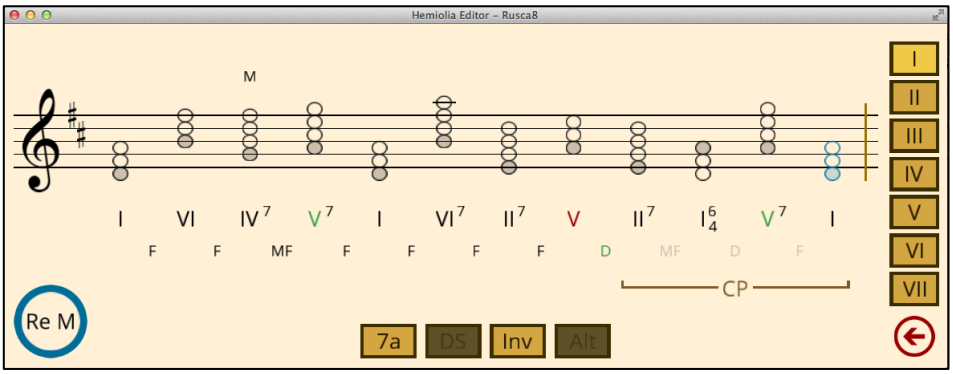

Figura 66: Esbós de l'exemple 2 introduït a l'aplicació (finestra allargada per encabir-ho tot).

Pel que fa a la realització (Figura 67), el primer que veurem serà que l'aplicació sorprenentment ha trobat dues quintes paral·leles. Després de descartar un possible error en l'aplicació veiem que, efectivament, són dues quintes justes i que per tant l'aplicació ha complert bé amb les normes que se li han donat. Tot i això, podria ser que en aquest cas la professora consideri correctes les quintes paral·leles que es mouen per graus conjunts (de la mateixa manera que considerem correctes nosaltres les que es mouen només mig to), així que no està clar si se li havia passat per alt o si per contra les ha deixat com a correctes expressament.

En tot cas, el moviment es correspon amb un dels errors que havíem definit a les nostres normes, i el programa l'ha indicat conseqüentment.

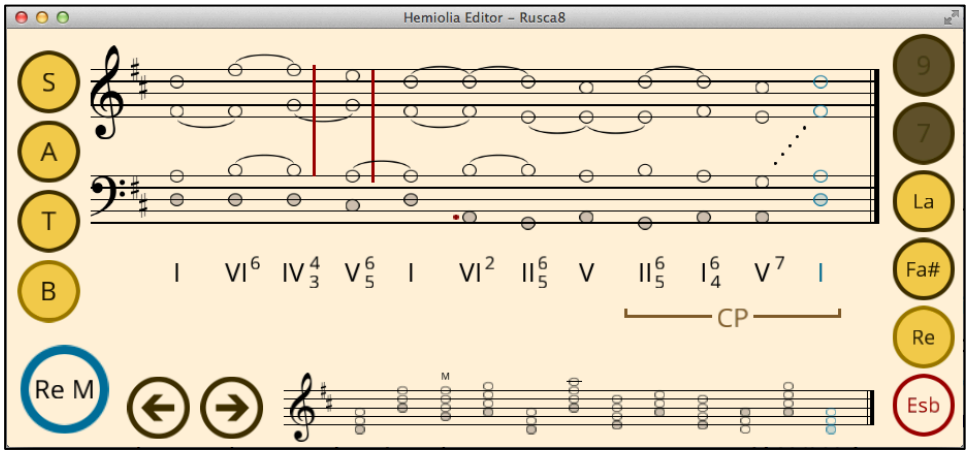

Figura 67: Realització de l'exemple 2 introduïda a l'aplicació (finestra allargada per encabir-ho tot).

Seguint amb els errors que ens havia indicat la professora, trobarem a la nostra realització dues indicacions relacionades amb les sèptimes. En el primer cas, la sèptima s'ha atacat per salt descendent, de manera que s'hi ha indicat l'error. En el segon, per contra, degut que la sèptima no estava resolta de la manera estàndard, s'hi ha indicat amb una línia de punts la resolució irregular

(no és necessàriament un error, però a les escoles se'ns demana indicar-la per tal de fer saber al professor que s'ha realitzat així expressament).

Finalment, com en el cas anterior, la melodia no seria correcta, però el programa de moment no ha estat preparat per analitzar-les.

10 CONCLUSIONS

Com hem anat veient amb els exemples de l'apartat anterior, en la major part dels casos el programa detecta els mateixos errors que apareixen a la correcció original, i sovint detecta a més algun error extra més petit que pugui haver passat per alt al professor. D'altra banda, hi ha alguns errors (principalment els de la melodia) que passaran desapercebuts a l'aplicació, i alguns altres (especialment alertes) que degut a certs contextos estranys el programa els indicarà malgrat en realitat seran correctes.

A més, es nota també que sovint hi ha petites diferències entre la versió de les normes que aplica cada professor, cosa que fa especialment útil el fet de registrar una explicació per cada error trobat (de manera que l'alumne pugui jutjar si l'error que se li marca es correspon amb la teoria que se li ha explicat o si per contra és quelcom que ells no tenen en compte). Tot i això, pel mateix motiu hi haurà normes que altres professors aplicaran (més restrictives) i que nosaltres no estarem tenint en compte, de manera que l'aplicació no els mostrarà els errors que se'n derivin.

En aquest sentit, serà interessant en un futur implementar una secció d'ajustos o opcions a la qual es pugui escollir quin joc de normes es vol aplicar, i d'aquesta manera anar incrementant el ventall de professors i escoles als quals l'aplicació respon amb precisió.

D'altra banda, els exemples que s'han escollit per a provar l'aplicació formen part només del principi del curs de cinquè (temari que s'ha inclòs en l'aplicació), de manera que qualsevol exercici una mica més avançat no es podria introduir en l'aplicació ni es podria corregir. Tot i això, gran part de l'aplicació està ja preparada per a gestionar una part d'aquests temaris més avançats, de manera que hauria de ser fàcil ampliar l'aplicació al menys fins al temari complet de 5è en una iteració posterior.

En tot cas, dissenyar i desenvolupar l'aplicació ha estat un repte molt interessant, i espero poder anar trobant estones al llarg del curs per a seguir-la millorant i completant. D'altra banda, serà interessant veure quina rebuda té entre els estudiants de conservatoris i escoles de música i en quina mesura s'utilitza i és útil una vegada estigui publicada.

11 BIBLIOGRAFIA

ANDROID DEVELOPER. Android SDK. (https://developer.android.com/studio/index.html, 3 de setembre de 2016)

CAPELLA SOFTWARE. Tonica Fugata. (http://www.capella-software.com/us/index.cfm/products/tonica-fugata/info-tonica-fugata/, 3 de setembre de 2016)

ECLIPSE FOUNDATION. About us. (https://eclipse.org/org/, 3 de setembre de 2016)

GITHUB. LibGDX Wiki. (https://github.com/libgdx/libgdx/wiki, 3 de setembre de 2016)

GOOGLE PLAY STORE. iReal Pro MusicBook-PlayAlong. (https://play.google.com/store/apps/details?id=com.massimobiolcati.irealb&hl=ca, 3 de setembre de 2016)

HARMONY BUILDER. Pàgina principal. (http://www.harmonybuilder.com/, 3 de setembre de 2016)

HARMONY PRACTICE PROJECT. Harmony practice 3 (http://harmonypractice.altervista.org/, 3 de setembre de 2016)

LIBGDX. Get Started. (https://libgdx.badlogicgames.com/, 3 de setembre de 2016)

MACWORLD. Finale 2011. (http://www.macworld.com/article/1153774/finale11.html, 3 de setembre de 2016)

MAKEMUSIC. What is finale. (http://www.finalemusic.com/products/finale/, 3 de setembre de 2016)

NICOLÁS, RIMSKY-KORSAKOV. "Tratado práctico de armonía." *Editorial Ricordi Americana. Buenos Aires*. 1993.

NOTEFLIGHT. Pàgina principal. (https://www.noteflight.com/ 3 de setembre de 2016)

PG MUSIC. Pàgina principal. (http://www.pgmusic.com/, 3 de setembre de 2016)

PHILHARMONIA ORCHESTRA. Sound Samples. (http://www.philharmonia.co.uk/explore/make_music, 3 de setembre de 2016)

SAXONLINE. Harmony Builder (http://www.saxonline.it/harmony-builder-2/, 3 de setembre de 2016)

STACK OVERFLOW. Respostes diverses. (http://stackoverflow.com/, 3 de setembre de 2016)

WOLFRAM ALPHA. Plot sin(261.626t) and sin(391.995t) from 0 to 0.05. (https://www.wolframalpha.com/, 3 de setembre de 2016)

ZAMACOIS, JOAQUÍN. *Tratado de armonía*. 2002.

ANNEX A

CODI INFORMÀTIC DE L'APLICACIÓ

A. ANNEX: CODI INFORMÀTIC

En aquest annex trobarem les diferents unitats de compilació de què s'ha anat parlant al llarg de la secció de Disseny, així com el codi que fa possible el llançament de l'aplicació a les dues plataformes principals que s'han tingut en compte (i.e. escriptori i mòbil Android).

Tot i això, s'ha de tenir en compte que a part hi haurà una sèrie de fitxers i llibreries que queden referenciats o inclosos dins el projecte de manera indirecta des del moment en què el creem amb l'eina de LibGDX. Així doncs, en aquest annex només veurem els fitxers que han estat modificats des de la creació del projecte amb l'eina *GDX Setup* (esmentada a l'apartat 6.3.6), ja que són en sí els que conformen l'aplicació.

Per tal d'intentar facilitar la lectura del codi i de fer-lo més manejable s'ha escollit donar-li una mida de lletra significativament més petita que la present a la resta del document. D'aquesta manera s'evitarà en molts casos que el sagnat acumulat en certes parts del codi provoqui salts de línia inexistents a l'entorn de programació, i alhora s'evitarà que les funcions s'allarguin durant moltes pàgines, cosa que en faria dificultós el seguiment.

A.1. Codi del projecte central (HemioliaEditor-core)

En primer lloc veurem el codi central, que és la part que s'ha anat comentant al llarg dels apartats de disseny del projecte. El tindrem organitzat per ordre alfabètic.

A.1.1. [A.java] – Biblioteca de les animacions

```
package com.rusca8.hEditor;

public class A { //gestió de les animacions

    //genèriques
    static float tHold=0.3f;
    //static final double TAU = 2*Math.PI;
    //botó del menú
    static float tMenu=-1; //temps que porta el menú premut (-1 quan no està actiu)
    static float rMenu=0.8f; //tant per u del forat del cercle del menú respecte al cercle gros
    //animació cursor
    static float tCursor = 0; //valor de temps de l'animació del cursor
    static float aCursor = 0; //traducció d'aquest a alpha
    //animació opcionals
    static float tAltOp =-1; //temps que porto pressionant una nota amb alteracions opcionals.
    static byte bAltOp; //número del botó que estic esperant a veure si es manté polsat

    public static void update(float delta){

        /**- - Animació pressionant el menú -*/
        if(tMenu>=0){
            tMenu+=delta; //afegeixo el temps de fotograma al temps que porta el menú pressionat.
        }

        if(tMenu<0){
            rMenu=0.8f;
        }else if(tMenu>tHold){
            rMenu=0.8f;
        }else{
            rMenu=0.8f*(1f-tMenu/tHold);
        }

        /**- - Animació del cursor - -*/
        tCursor+=delta;
        if(tCursor>1){
            tCursor=0;
        }else{
            if(tCursor<0.5f){
                aCursor=1;
            }else{
                aCursor=0;
```

```
            }
        }

        if(tAltOp>=0){
            tAltOp+=delta;
        }

    }
}
```

A.1.2.[BotIA.java] – Botó dels ingredients tipus acord

```
package com.rusca8.hEditor;

import com.rusca8.hEditor.BotIA;
import com.rusca8.hEditor.F;
import com.rusca8.hEditor.H;
import com.rusca8.hEditor.IngAcord;
import com.rusca8.hEditor.Recursos;

public class BotIA {
    IngAcord iAc; //ingredient associat al botó

    int x;
    int y;
    int w=0; //ja es calculen a baix. Deixo un número per si quelcom hi accedeix abans que estigui calculat.
    int h=0;

    int[] d = new int[4]; //per desar les distàncies acumulades des del principi fins cada zona de l'escrit
    String carro="";
    String carro2="";

    public BotIA(float x, float y, IngAcord iAc){
        this.x = (int) x;
        this.y = (int) y;

        this.iAc = iAc;

        updateWH();
    }

    public BotIA(int id){ //constructor que utilitzaré per esborrar ítems de la llista
        this.iAc=new IngAcord(id);
    }

    void updateWH(){
        h = (int) (3.2f*F.vAlign("bMenu", 'd', "6")); //alçada del botó en proporció al que ocupen els xifrats

        d[0] = (int) (-F.align("font",'r',F.ifText(iAc.n>1 && !iAc.tcp, iAc.n+"x."))); //amplada de la quantitat
(el punt no el posaré, és per deixar espai).
        d[1] = d[0]+ (int) (-F.align("bigFont", 'r', H.roman[iAc.acord.grau])-
F.align("font",'r',F.ifText((iAc.acord.sept || iAc.triada || iAc.acord.inv!=0), "."))); //amplada del grau (si tinc
xifrat poso espai darrere)
        d[2] = d[1]+ (int) (Math.max(-F.align("bMenu", 'r', H.xifrat(iAc.acord, true)),-
F.align("bMenu",'r',F.ifText(iAc.triada && iAc.acord.inv<1, "5", H.xifrat(iAc.acord, false))))); //amplada del
xifrat (màxim de les dues parts)
        d[3] = d[2]+ (int) (-F.align("bMenu",'r',F.keep(F.ifText(iAc.alt || iAc.septMajor, " (")
                                                + F.ifText(iAc.alt, H.altOpText(iAc.acord))
                                                + F.ifText(iAc.alt && iAc.septMajor, "|") //de fet mai no
apareixeran les dues juntes, així que la barra no surt mai. Potser més endavant és útil, però. La deixo de moment.
                                                + F.ifText(iAc.septMajor, "7M")
                                                + F.ifText(iAc.alt || iAc.septMajor, ")"))));
        carro=F.lastText; //deso tota la parafernàlia aquesta per no haver-la de calcular cada vegada que la pinti,
que deixaria moltes línies atapeides dins el codi
        w = d[3]+ (int) (-F.align("font",'r',F.keep(F.ifText(iAc.difAc,
                                                        F.ifText(iAc.difInv, " dai", " d"),
                                                //else (!iAc.difAc)
                                                        F.ifText(iAc.difInv, " di"))
                                                + F.ifText(iAc.enll, " e")
                                                + F.ifText(iAc.tcp, " !"))));
        carro2=F.lastText; //i deso aquesta segona part
    }

    void shrink(){ //empetiteix les mesures calculades responent a la nova mida que tindrà el botó un cop acceptat
l'ingredient (utilitzaré mides de font diferents en conseqüència)
        //faré:
        // bigFont -> font
        // font -> iQuant
```

```
        // bMenu -> iXifrat

        float ratio = Recursos.font.getCapHeight()/Recursos.bigFont.getCapHeight();

        h *= ratio; //sembla que ja fa el cast ell sol, fet així
        for(int i=0;i<d.length;i++){
            d[i] *= ratio;
        }
        w *= ratio;
    }

    public String getText(){
        return ("ID="+iAc.id+" | "
                  + F.ifText(iAc.n>1 && !iAc.tcp, iAc.n+"x ")
                  + H.roman[iAc.acord.grau]
                  + " [" + H.xifrat(iAc.acord, false) + "/" + H.xifrat(iAc.acord, true)
                  + "] " + carro);
    }

    public void printLn(){
        System.out.println(getText());
    }

    @Override
    public boolean equals(Object obj){ //redefineixo la funció de comparar (amb els objectes és necessari per tal
que funcioni bé).
        if(!(obj instanceof BotIA)){
            System.out.println("L'objecte no és un botó d'ingredient tipus acord.");
            return false;
        }
        BotIA bot2 = (BotIA) obj;

        return (this.iAc.id==bot2.iAc.id); //només associaré un botó a cada ingredient, així que puc comparar els
botons també per l'id.
    }
}
```

A.1.3.[BotIC.java] – Botó dels ingredients tipus cadència

```
package com.rusca8.hEditor;

import com.rusca8.hEditor.BotIC;
import com.rusca8.hEditor.F;
import com.rusca8.hEditor.IngCadencia;
import com.rusca8.hEditor.Recursos;

public class BotIC {
    IngCadencia iCad; //explicacions a BotonIA

    int x;
    int y;
    int w=0;
    int h=0;

    int[] d = new int[3];

    public BotIC(float x, float y, IngCadencia iCad){
        this.x = (int) x;
        this.y = (int) y;

        this.iCad = iCad;

        updateWH();
    }

    public BotIC(int id){
        this.iCad = new IngCadencia(id);
    }

    void updateWH(){
        h = (int) (3.2f*F.vAlign("bMenu", 'd', "6")); //mantinc la mateixa alçada que els tipus acord

        d[0] = (int) (-F.align("font",'r',F.ifText(iCad.n>1 && !iCad.tcp, iCad.n+"x."))); //amplada de la quantitat
        d[1] = d[0]+ (int) (-F.align("bigFont",'r',"CT")-F.align("font",'r',F.ifText(iCad.sept ||
iCad.triada,"."))); //amplada de CT
        d[2] = d[1]+ (int) (-F.align("bMenu",'r',F.ifText(iCad.sept, "7",F.ifText(iCad.triada, "5")))); //amplada
del "xifrat"

        w = d[2]+ (int) (-F.align("font",'r',F.ifText(iCad.dif, " d")+F.ifText(iCad.tcp, " !")));
```

```
    }

    void shrink(){
        //faré
        //bigFont -> font
        //font -> iQuant
        //bMenu -> iXifrat

        float ratio = Recursos.font.getCapHeight()/Recursos.bigFont.getCapHeight();

        h *= ratio;
        for(int i=0;i<d.length;i++){
            d[i] *= ratio;
        }
        w *= ratio;
    }

    public String getText(){
        return("ID="+iCad.id+" | "
                + F.ifText(iCad.n>1 && !iCad.tcp, iCad.n+"x ")
                + "CT"
                + F.ifText(iCad.sept, " [7] ",F.ifText(iCad.triada, " [5] "))
                + F.ifText(iCad.dif, " d")+F.ifText(iCad.tcp, " !"));
    }

    public void printLn(){
        System.out.println(getText());
    }

    @Override
    public boolean equals(Object obj){ //redefineixo la funció de comparar (amb els objectes és necessari per tal
que funcioni bé).
        if(!(obj instanceof BotIC)){
            System.out.println("L'objecte no és un botó d'ingredient tipus cadència.");
            return false;
        }
        BotIC bot2 = (BotIC) obj;

        return (this.iCad.id==bot2.iCad.id); //només associaré un botó a cada ingredient, així que puc comparar
els botons també per l'id.
    }
}
```

A.1.4.[BotIR.java] – Botó dels ingredients tipus resolució irregular

```
package com.rusca8.hEditor;

import com.rusca8.hEditor.BotIR;
import com.rusca8.hEditor.F;
import com.rusca8.hEditor.IngRes;

public class BotIR { //botó associat als ingredients tipus resolució irregular
    IngRes iRI; //explicacions a BotonIA

    int x;
    int y;
    int w=0;
    int h=0;

    int d;

    public BotIR(float x, float y, IngRes iRI){
        this.x = (int) x;
        this.y = (int) y;

        this.iRI = iRI;

        updateWH();
    }

    public BotIR(int id){
        this.iRI = new IngRes(id);
    }

    void updateWH(){
        h = (int) (3.2f*F.vAlign("iXifrat",'d',"6"));

        d = (int) (-F.align("font",'r',"RI")-F.align("iQuant",'r',".")); //amplada del text "RI"
        w = d + (int) (-F.align("iXifrat",'r',F.ifText(iRI.sens, "S","7"))); //amplada del "xifrat"
```

```
    }

    public String getText(){
        if(iRI.sens){
            return ("ID="+iRI.id+"RIs");
        }else{
            return ("ID="+iRI.id+"RI7");
        }
    }

    public void printLn(){
        System.out.println(getText());
    }

    @Override
    public boolean equals(Object obj){ //redefineixo la funció de comparar (amb els objectes és necessari per tal
que funcioni bé).
        if(!(obj instanceof BotIR)){
            System.out.println("L'objecte no és un botó d'ingredient tipus RI.");
            return false;
        }
        BotIR bot2 = (BotIR) obj;

        return (this.iRI.id==bot2.iRI.id); //només associaré un botó a cada ingredient, així que puc comparar els
botons també per l'id.
    }
}
```

A.1.5.[BotQ.java] – Botó genèric

```
package com.rusca8.hEditor;

public class BotQ {
    int x;
    int y;
    int w=115;
    int h=80;

    int grau;
    String text;

    public BotQ(float x, float y, int grau){
        this.x = (int) x;
        this.y = (int) y;
        this.grau = grau;
    }

    public BotQ(float x, float y, String text){
        this.x = (int) x;
        this.y = (int) y;
        this.text = text;
    }

    public BotQ(float x, float y){
        this.x = (int) x;
        this.y = (int) y;
    }

    public BotQ(float x, float y, float w, float h, String text){
        this.x = (int) x;
        this.y = (int) y;
        this.w = (int) w;
        this.h = (int) h;
        this.text = text;
    }

    public BotQ(float x, float y, float w, float h, int grau){ //aquest l'utilitzo pels botons de notes a la
segona fase (no són graus, però com que és un número aprofito i el guardo a grau).
        this.x = (int) x;
        this.y = (int) y;
        this.w = (int) w;
        this.h = (int) h;
        this.grau = grau;
    }

    public BotQ(float x, float y, float w, float h){
        this.x = (int) x;
        this.y = (int) y;
        this.w = (int) w;
```

```
        this.h = (int) h;
    }
}
```

A.1.6.[Cadencia.java] – Element tipus cadència

```
package com.rusca8.hEditor;

public class Cadencia {

    byte n; //posició inicial
    byte w; //acords que dura (no inclou el primer, així que en realitat són salts d'acord que dura)
    boolean ds = false; //porta dominant secundària? (cert en cas que sigui amb III-)
    boolean cp; //marca per si és la cadència perfecta.

    public Cadencia(int n, int w){
        this.n = (byte) n;
        this.w = (byte) w;
    }
    public Cadencia(int n, boolean ds, int w){
        this.n = (byte) n;
        this.w = (byte) w;
        this.ds = ds;
    }
    public Cadencia(int n, int w, boolean cp){
        this.n = (byte) n;
        this.w = (byte) w;
        this.cp = cp;
    }
}
```

A.1.7.[Chord.java] – Element tipus acord

```
package com.rusca8.hEditor;

import com.rusca8.hEditor.Chord;

public class Chord { //classe que servirà per escriure els acords de l'esbós (poso variables que caldran molt més
endavant, però per tenir ja ben definit l'acord des del principi.
    byte grau; //grau de l'acord
    byte inv=0; //inversió (si volem indicar-la). 0=EF, 1-3 inversions 1a a 3a

    boolean sept=false; //porta sèptima?
    boolean dom=false; //és de la família de la dominant? (D.S., D.D., V9)
    boolean hom=false; //és de l'homònim menor? (o estic en un esquema menor, que ve a ser el mateix).

    boolean nap=false; //és una napolitana?
    boolean alt6=false, alt7=false; //porta alterada la sexta? i la sensible?
    boolean alt9=false; //porta alterada la novena? (7ds)
    byte alt5=0; //alteració de la quinta en les dominants amb la 5a alterada (-1 baixada, 0 res, 1 augmentada, 6
sexta augmentada clàssic).

    public Chord(int grau){
        this.grau = (byte) grau;
    }

    public Chord(int grau, boolean septima){
        this.grau = (byte) grau;
        this.sept = septima;
    }

    public Chord(Chord ac){ //per copiar un acord en lloc de referenciar-lo
        this.grau=ac.grau;
        this.inv=ac.inv;

        this.sept=ac.sept;
        this.dom=ac.dom;
        this.hom=ac.hom;

        this.nap=ac.nap;
        this.alt6=ac.alt6;
        this.alt7=ac.alt7;
        this.alt9=ac.alt9;
        this.alt5=ac.alt5;
    }

    public void addInv(){
        if(dom || sept){
            inv = (byte) ((inv+1)%4);
```

```
        }else{
            inv = (byte) ((inv+1)%3);
        }
    }
}
```

A.1.8.[CorPuntual] – Element de correcció puntual

```
package com.rusca8.hEditor;

public class CorPuntual { //correccions que no val la pena guardar per tots els acords (només guardo en els llocs
puntuals en què hi ha errors).
    byte ac; //acord on hi ha l'error
    byte veu; //veu on hi ha l'error
    byte aux; //valor auxiliar (per quan són errors entre dues veus o coses així)
    char cor; //valor de la correcció.

    public CorPuntual(int ac, int veu, int aux, char cor){
        this.ac = (byte) ac;
        this.veu = (byte) veu;
        this.aux = (byte) aux;
        this.cor = cor;
    }
    public CorPuntual(int ac, int veu, int aux){
        this.ac = (byte) ac;
        this.veu = (byte) veu;
        this.aux = (byte) aux;
    }
    public CorPuntual(int ac, int veu, char cor){
        this.ac = (byte) ac;
        this.veu = (byte) veu;
        this.cor = cor;
    }

    public CorPuntual(int ac, int veu){
        this.ac = (byte) ac;
        this.veu = (byte) veu;
    }
}
```

A.1.9.[Desplegat] – Unitat bàsica de la realització

```
package com.rusca8.hEditor;

public class Desplegat {
    byte[] satb;
    byte[] ia;

    public Desplegat(){
        satb = new byte[] {0,0,0,0};
        ia = new byte[] {0,0,0,0};
    }

    public void set(int veu, int nota, int index){
        satb[veu] = (byte) nota;
        ia[veu] = (byte) index;
    }
}
```

A.1.10. [F.java] – Biblioteca de funcions genèriques

```
package com.rusca8.hEditor;

import com.badlogic.gdx.graphics.g2d.GlyphLayout;
import com.rusca8.hEditor.CorPuntual;
import com.rusca8.hEditor.Note;
import com.rusca8.hEditor.Recursos;

public class F { //altres funcions d'utilitat

    static String lastText;
    static Note lastNote;
    static int lastInt;
    static CorPuntual lastCor;

    /************* FUNCIONS DE PERSISTÈNCIA **************/

    public static String keep(String text){ //per guardar un text alhora que el faig servir (de manera que tot
```

```
seguit el pugui alinear sense escriure'l dues vegades).
        lastText = text;
        return text;
    }
    public static Note keep(Note nota){
        lastNote = new Note(nota.nom,nota.so,nota.ia);
        return nota;
    }
    public static int keep(int num){
        lastInt = num;
        return num;
    }

    /************ FUNCIONS D'ALINEAMENT *************/

    static GlyphLayout caixaDeText = new GlyphLayout();

    /**- - - - - Alineament horitzontal - - - - - */

    public static float align(){ //Totes aquestes retornen la distància que s'ha de moure el text per què quedi
alineat on tries
        return align("font",'c',lastText); //per fer textos llargs amb límits a esquerra i dreta hi ha una versió
més senzilla directament a font.draw
    }
    public static float align(String font){
        return align(font,'c',lastText);
    }
    public static float align(char alignment){
        return align("font",alignment,lastText);
    }
    public static float align(char alignment, String text){
        return align("font",alignment,text);
    }
    public static float align(String font, char alignment){
        return align(font,alignment,lastText);
    }
    public static float align(String font, char alignment, String text){
        if(font=="bigFont"){
            caixaDeText.setText(Recursos.bigFont, text);
        }else if(font=="font"){
            caixaDeText.setText(Recursos.font, text);
        }else if(font=="bMenu"){
            caixaDeText.setText(Recursos.bMenu, text);
        }else if(font=="bNotes"){
            caixaDeText.setText(Recursos.bNotes, text);
        }else if(font=="xifrat"){
            caixaDeText.setText(Recursos.xifrat, text);
        }else if(font=="iQuant"){
            caixaDeText.setText(Recursos.iQuant, text);
        }else if(font=="iXifrat"){
            caixaDeText.setText(Recursos.iXifrat, text);
        }else if(font=="eXifrat"){
            caixaDeText.setText(Recursos.eXifrat, text);
        }else{
            System.out.println("No has implementat aquesta lletra a la funció d'alinear..");
            return 42;
        }
        if(alignment=='c'){
            return (-caixaDeText.width/2f);
        }else if(alignment=='r'){
            return (-caixaDeText.width);
        }else{
            System.out.println("mira't bé les opcions d'alineament");
            return 42;
        }
    }

    /**- - - - Alineament vertical - - - -*/

    public static float vAlign(){ //equivalent vertical. El pin del text és a dalt a l'esquerra
        return vAlign("font",'c',lastText);
    }
    public static float vAlign(String font){
        return vAlign(font,'c',lastText);
    }
    public static float vAlign(char alignment){
        return vAlign("font",alignment,lastText);
    }
    public static float vAlign(char alignment, String text){
        return vAlign("font",alignment,text);
```

```
    }
    public static float vAlign(String font, char alignment){
        return vAlign(font,alignment,lastText);
    }
    public static float vAlign(String font, char alignment, String text){
        if(font=="bigFont"){
            caixaDeText.setText(Recursos.bigFont, text);
        }else if(font=="font"){
            caixaDeText.setText(Recursos.font, text);
        }else if(font=="bMenu"){
            caixaDeText.setText(Recursos.bMenu,text);
        }else if(font=="bNotes"){
            caixaDeText.setText(Recursos.bNotes,text);
        }else if(font=="xifrat"){
            caixaDeText.setText(Recursos.xifrat,text);
        }else if(font=="iQuant"){
            caixaDeText.setText(Recursos.iQuant,text);
        }else if(font=="iXifrat"){
            caixaDeText.setText(Recursos.iXifrat,text);
        }else if(font=="eXifrat"){
            caixaDeText.setText(Recursos.eXifrat,text);
        }else{
            System.out.println("No has implementat aquesta lletra a la funció d'alinear..");
            return 42;
        }
        if(alignment=='c'){
            return (caixaDeText.height/2f);
        }else if(alignment=='d'){
            return (caixaDeText.height);
        }else{
            System.out.println("mira't bé les opcions d'alineament.");
            return 42;
        }
    }

    /**- - Alineament dels graus - - */
    public static float gAlign(Chord ac){ //retorna l'alineament dels graus (centrat) tenint en compte el text
extra dels que en porten
        return (align("font",'c',H.roman[ac.grau])+align("xifrat",'c',H.textGrau(ac)));
    }

    /******** GESTIÓ DEL LLENGUATGE *******/

    /**- - - Noms dels números - - -*/

    static final String[] nomNums = {"Zero", "Un", "Dos", "Tres", "Quatre", "Cinc", "Sis", "Set", "Vuit", "Nou",
"Deu"};

    public static String nomNums(int n){
        if(n<nomNums.length){
            return nomNums[n];
        }else{
            return Integer.toString(n);
        }
    }

    /**- - - - Plurals - - - -*/

    public static String plurals(String singular, int quantitat){ //retorna el plural donat el singular.
        if(quantitat==1){
            return singular;
        }else{
            if(singular=="cinquè"){
                return "cinquens";
            }else if(singular=="sisè"){
                return "sisens";
            }else if(singular=="setè"){
                return "setens";
            }else if(singular=="tríada"){
                return ("tríades");
            }else{
                return (singular+"s");
            }
        }
    }

    /** - - - Femenins - - - */

    public static String fem(String masc){ //alguns femenins que aniré necessitant (de moment pels números 1-10);
        if(masc=="Un"){
```

```
            return "Una";
        }else if(masc=="Dos"){
            return "Dues";
        }else{
            return masc;
        }
    }

    /**- - - booleans a Sí/No - - -*/

    public static String siNo(boolean b){
        if(b){
            return "sí";
        }else{
            return "no";
        }
    }

    /**- - - IF de mySQL - - -*/

    public static String ifText(boolean b, String textSi){ //la majoria de les vegades no voldré res si és fals
            return ifText(b,textSi,"");
    }
    public static String ifText(boolean b, String textSi, String textNo){ //com a mySQL i tota la pesca
        if(b){
            return textSi;
        }else{
            return textNo;
        }
    }
}
```

A.1.11. [H.java] – Biblioteca de funcions relacionades amb la música

```
package com.rusca8.hEditor;

import com.rusca8.hEditor.Chord;
import com.rusca8.hEditor.F;
import com.rusca8.hEditor.H;
import com.rusca8.hEditor.Note;
import com.rusca8.hEditor.Pant;
import com.rusca8.hEditor.Pant2;

public class H { // funcions i variables relacionades amb la música

                              //  0     1     2     3     4     5     6
    static final String[] notes  = { "Do", "Re", "Mi", "Fa", "Sol", "La", "Si"};
    static final String[] mNotes = { "do", "re", "mi", "fa", "sol", "la", "si"};
    static final String[] rNotes = { "D",  "R",  "M",  "F",  "S",   "L",  "T" };

    /**- - - Majúscules segons mode - - -*/
    public static String modeNotes(int nota, int mode) {
        if (mode == 0) {
            return notes[nota]; // majúscula
        } else {
            return mNotes[nota]; // minúscula
        }
    }
                              //  0       1
    static final String[] modes  = { "Major", "menor"};
    static final String[] rModes = { "M",     "m"    };

    static final String[] roman  = { "I",      "II",    "III",    "IV",    "V",      "VI",    "VII",  "X"
};
    static final String[] graus  = { "primer", "segon", "tercer", "quart", "cinquè", "sisè", "setè", "qualsevol"
};

    static final String[] inversions = { "estat fonamental", "primera inversió", "segona inversió", "tercera
inversió" };

    static final String[] veus = { "Soprano", "Contralt", "Tenor", "Baix" };
    static final String[] rVeus = { "S",      "A",        "T",     "B" };

    public static Note tMinS = new Note(0, 0, 3); // límits de les tessitures de les veus
    public static Note tMaxS = new Note(5, 9, 4);
    public static Note tMinA = new Note(3, 5, 2);
    public static Note tMaxA = new Note(0, 0, 4);
    public static Note tMinT = new Note(0, 0, 2);
    public static Note tMaxT = new Note(4, 7, 3);
```

```
    public static Note tMinB = new Note(3, 5, 1);
    public static Note tMaxB = new Note(0, 0, 3);

    static Note tonalitat = new Note(0,0); // (nom, so)
    static byte mode = 0; // 0=Major, 1=menor

    static Note tonalitatPrevia; // per quan canvio el to però després descarto els canvis (menú dins pantalla de
l'esbós).
    static byte modePrevi;

    static byte armadura; // calculat (demano el càlcul al constructor de pantalla).

    /******** CÀLCUL D'ENLLAÇOS ********/
    public static String link(int linkNum) {
        switch (linkNum) {
        case 1:
            return "F";
        case 2:
            return "D";
        case 3:
            return "MF";
        default:
            return "";
        }
    }

    public static String link(int gAnt, int gAct) {
        return link(linkNum(gAnt, gAct));
    }

    /**- - - Versió en codi numèric - - -*/
    public static byte linkNum(int gAnt, int gAct) {
        int interval = gAct - gAnt + 1; // pujant positiu. TODO passar tot el que no calgui gros a byte
        while (interval <= 0) { // sumo de 7 fins positiu, i miro l'enllaç sempre pujant, per no complicar.
            interval += 7;
        }
        if (interval == 4 || interval == 6) {
            return 1;
        } else if (interval == 3 || interval == 5) {
            return 2;
        } else if (interval == 2 || interval == 7) {
            return 3;
        } else {
            return 0;
        }
    }

    /****** XIFRAT ******/
    public static String xifrat(int inv, boolean baix) { // per si vull demanar la inversió a partir de les
inversions normalitzades
                                                         // de la correcció (durant SATB)
        Chord ac = new Chord(0); // acord qualsevol. És per passar-li a la funció original.
        if (inv < 5) { // si és tríade
            ac.sept = false;
            ac.dom = false;
            ac.inv = (byte) inv;
        } else { // si té sèptima
            ac.sept = true;
            ac.inv = (byte) (inv % 10);
        }
        return xifrat(ac, baix);
    }

    public static String xifrat(Chord ac, boolean baix) {
        if (baix) { // part del subíndex
            if (ac.sept || ac.dom) { // quan té sèptima
                switch (ac.inv) {
                case -1:
                    return "EF"; // -1 serà per quan vull 7 EF específicament
                case 0:
                    return "";
                case 1:
                    return "5";
                case 2:
                    return "3";
                case 3:
                    return "";
                default:
                    return "?";
                }
```

```
        } else {
            switch (ac.inv) {
            case -1: // -1 serà per quan vull 5 EF específicament
                return "EF";
            case 0:
                return "";
            case 1:
                return "";
            case 2:
                return "4";
            default:
                return "?";
            }
        }
    } else { // part del superíndex
        if (ac.sept || ac.dom) {
            switch (ac.inv) {
            case -1:
                return "7"; // a la sèptima -1 i 0 són el mateix xifrat,
                    // però a ingredients -1 implicarà EF obligatori
            case 0:
                return "7";
            case 1:
                return "6";
            case 2:
                return "4";
            case 3:
                return "2";
            default:
                return "?";
            }
        } else {
            switch (ac.inv) {
            case -1:
                return ""; // EF específic
            case 0:
                return "";
            case 1:
                return "6";
            case 2:
                return "6";
            default:
                return "?";
            }
        }
    }
}

public static String textGrau(Chord ac){ //text extra que porten alguns acords a més del grau
    if(ac.grau==4 && (mode==1 || ac.hom) && !ac.dom && !ac.alt7){ // V del menor sense sensible (Vm)
        return ("m");
    }else{
        return "";
    }
}

/************** GESTIÓ DE LES ALTERACIONS ***********/

/**- - - ARMADURA - - -*/

public static int armadura(int nom, int so, int mode) { // polimorfisme per comoditat.
    return armadura(new Note(nom, so), mode);
}

public static int armadura(Note to, int mode) { // Càlcul de l'armadura d'una tonalitat
    int n;
    int i;
    if (mode == 0) { // Tonalitats Majors
        if (to.nom == 0 && to.so == 0) { // Do M
            n = 0;
        } else if (to.nom == 3 && to.so == 5) { // Fa M
            n = -1;
        } else if (getAltNum(to) == 0) { // Si el nom de la tonalitat no té alteració...
            i = 4; // començant per Sol
            n = 1; // que té un sostingut
            while (i != to.nom) {
                i = (i + 4) % 7;
                n++; // afegint un sostingut a cada iteració
            }
        } else if (getAltNum(to) == -1) { // Si el nom de la tonalitat té un bemoll...
```

```
                i = 6; // començant a Si
                n = -2; // que té dos bemolls
                while (i != to.nom) {
                    i = (i + 3) % 7; // recorro el cercle descendent de quintes.
                    n--; // afegint un bemoll a cada iteració
                }
            } else {
                n = armadura(to.nom, to.so - getAltNum(to.nom, to.so), 0) + 7 * getAltNum(to.nom, to.so); // miro
l'equivalent en natural i afegeixo 7
                                                                                                    // per cada mig
to d'alteració al nom de tonalitat
            }
        } else { // si és menor
            n = armadura(addInterval(to, 2, qMin[2]), 0); // busca l'armadura del relatiu major
        }
        return n;
    }

    /**- - - Versió en text - - -*/
    public static String textArm(int armadura) {
        if (armadura > 0) {
            return ((armadura) + "#");
        } else if (armadura < 0) {
            return ((-armadura) + "b");
        } else {
            return "";
        }
    }

    /**- - - GET ALT - - -*/

    public static int getAltNum(int nom, int so) {
        return getAltNum(new Note(nom, so));
    }

    public static int getAltNum(Note nota) { // càlcul de la diferència entre un nom i un so
        int alterat = nota.so - cromatic(nota.nom); // busco el semitò corresponent al nominal de la nota
                                                    //i el comparo
                        // amb el semitò corresponent al so real
        if (alterat > 6) { // corregeixo per agafar el camí més curt (si no,
                // comparant la nota 0 amb el so 11 dona 11 en lloc de -1).
            alterat -= 12;
        } else if (alterat < -6) {
            alterat += 12;
        }
        return alterat;
    }

    /**- - - Versió en text - - -*/
    public static String getAlt(int nom, int so) {
        return getAlt(new Note(nom, so));
    }

    public static String getAlt(Note nota) {
        switch (getAltNum(nota)) {
        case 0:
            return "";
        case 1:
            return "#";
        case 2:
            return "x";
        case -1:
            return "b";
        case -2:
            return "bb";
        default:
            return "?";
        }
    }

    /**- - - ACCIDENTAL - - -*/

    public static boolean accidental(int nom, int so) { // polimorfisme per comoditat
        return accidental(new Note(nom, so), tonalitat, mode);
    }

    public static boolean accidental(Note nota) { // assumeixo que si no demano res és de la tonalitat actual
        return accidental(nota, tonalitat, mode);
    }
```

```
    public static boolean accidental(int nom, int so, int nomTo, int soTo, int mode) {
        return accidental(new Note(nom, so), new Note(nomTo, soTo), mode);
    }

    public static boolean accidental(int nom, int so, Note to, int mode) {
        return accidental(new Note(nom, so), to, mode);
    }

    public static boolean accidental(Note nota, Note to, int mode) { // diu si cal escriure l'alteració d'una
nota
                            // o si ja és a l'armadura
        byte dNatural;
        Note itv = getInterval(to, nota); // calculo l'interval corresponent a la nota dins el to
        if (itv.nom < 0 || itv.so < 0) {
            itv.nom += 7;
            itv.so += 12; // (just in case. No fos cas que hi hagi índex acústic i em surti descendent l'interval).
        }
        if (mode == 0) { // calculo la distància natural corresponent al nom d'interval trobat
            dNatural = qMaj[itv.nom];
        } else {
            dNatural = dMin[itv.nom];
        }
        if (dNatural == itv.so) { // si ambdós qualificatius són iguals, l'alteració ja ve de l'armadura
            return false;
        } else { // si són diferents, s'haurà d'alterar
            return true;
        }
    }


    /************* GESTIÓ D'INTERVALS **********/

    public static byte qMaj[] = { 0, 2, 4, 5, 7, 9, 11 }; // distàncies dels intervals amb qualificatius majors
    public static byte qMin[] = { 0, 1, 3, 5, 7, 8, 10 }; // " " " " " menors
    public static byte dMin[] = { 0, 2, 3, 5, 7, 8, 10 }; // distàncies de les notes respecte a la fonamental en
les tonalitats menors.

    /**- - - CROMATIC - - -*/
    public static byte cromatic(int nota) { // passa la nota al seu semitò equivalent: chromatic[nota] =  semitò
(és el mateix que llegir
            // qMaj, però com que conceptualment és diferent millor ho deixo separat).
        return qMaj[nota];
    }

    /**- - - GET INTERVAL - - -*/

    public static Note getInterval(int nom1, int so1, int ia1, int nom2, int so2, int ia2) { // versions
polimòrfiques per proves i altres càlculs
        return getInterval(new Note(nom1, so1, ia1), new Note(nom2, so2, ia2));
    }

    public static Note getInterval(int nom1, int so1, int nom2, int so2) {
        return getInterval(new Note(nom1, so1), new Note(nom2, so2));
    }

    public static Note getInterval(Note nota1, Note nota2) { // obté un interval a partir de dues notes
        Note itv = new Note(nota2.nom - nota1.nom, nota2.so - nota1.so, nota2.ia - nota1.ia);
        if (itv.nom > 0 && itv.so < 0) { // Aquesta ha estat estranya: quan faig un interval amb alguna nota tipus
                                         // Dob (0,11) em queda l'interval tipus (+,-) o (-,+). Vull signe
                                         // del qualificatiu coherent amb la direcció, així que arreglo.
            itv.so += 12;
        } else if (itv.nom < 0 && itv.so > 0) {
            itv.so -= 12;
        }
        if (itv.ia > 0 && (itv.nom < 0 || itv.so < 0)) { // ajusto quan enllaça fent la volta als noms de nota (com
                                                         // La-Re, que canvia d'índex acústic) (ascendents)
            itv.nom += 7;
            itv.so += 12;
            itv.ia--;
        } else if (itv.ia < 0 && (itv.nom > 0 || itv.so > 0)) { // igual per descendents
            itv.nom -= 7;
            itv.so -= 12;
            itv.ia++;
        }
        return itv; // interval = Note on (nom,so,ia) = (nom_interval-1, semitons, octaves extra).
    }

    /**- - - ADD INTERVAL - - -*/

    static Note addInterval(Note nota, int nom, int qualif) { // afegeix un interval a una nota
```

```
        nota = new Note(nota.nom, nota.so); // IMPORTANT: NO COPIA ELS OBJECTES.
                    // Sense aquesta línia estaria
                    // modificant directament a la nota
                    // que li entri (habitualment la
                    // tonalitat :____D)
        nota.nom += nom;
        nota.so += qualif;
        while (nota.nom > 6) { // passo els canvis d'octava a increments de
                            // l'índex acústic (vull la nota entre 0 i 6, i
                            // el so entre 0 i 12)
            nota.nom -= 7;
            nota.so -= 12;
            nota.ia += 1;
        }
        while (nota.nom < 0) { // i el mateix per al cas descendent
            nota.nom += 7;
            nota.so += 12;
            nota.ia -= 1;
        }
        return nota;
    }

    /**- - - MODIFICACIONS DELS INTERVALS - - -*/

    /**- - - Ascendent - - -*/
    public static Note ascendent(Note interval) { // passa un interval a l'equivalent simple ascendent (no tinc en
                                                  // compte ia). Si ja és ascendent el deixa com està.
        interval = new Note(interval.nom, interval.so, interval.ia); // em desfaig de la igualtat per referència
        if (interval.nom > 0) {
            return interval;
        } else {
            interval.nom += 7;
            interval.so += 12;
            return interval;
        }
    }

    /**- - - Descendent - - -*/
    public static Note descendent(Note interval) { // equivalent descendent a la funció anterior
        interval = new Note(interval.nom, interval.so, interval.ia); // em desfaig de la igualtat per referència
        if (interval.nom < 0) {
            return interval;
        } else {
            interval.nom -= 7;
            interval.so -= 12;
            return interval;
        }
    }

    /**- - - Compost - - -*/
    public static int compost(Note itv) { // retorna el nom de l'interval compost
        return itv.nom + 7 * itv.ia;
    }

    /**- - - COMPROVACIONS SOBRE INTERVALS - - -*/

    /**- - - Major, menor o Just - - -*/
    public static boolean mMJ(Note itv) { // retorna true si l'acord és major, menor o just, false si és fora els
límits.
        return (Math.abs(itv.so) <= qMaj[Math.abs(itv.nom)] && Math.abs(itv.so) >= qMin[Math.abs(itv.nom)]);
    }

    /**- - - Augmentat - - -*/
    public static boolean augmentat(Note itv) { // retorna true si l'interval és augmentat, false si no ho és.
        return (Math.abs(itv.so) == qMaj[Math.abs(itv.nom)] + 1); // miro que la distància de l'interval
                                                                  // sigui més gran que la seva distància
                                                                  // natural més gran.
    }

    /**- - - Disminuit - - -*/
    public static boolean disminuit(Note itv) { // retorna true si l'interval és disminuït, false si no ho és.
        return (Math.abs(itv.so) == qMin[Math.abs(itv.nom)] - 1); // miro que la distància de l'interval
                                                                 //sigui més petita que la seva distància natural més
petita.
    }


    /******** OBTENCIÓ DE NOTES I GESTIÓ D'ACORDS *******/

    /**- - - GET NOTE - - -*/
```

```
    public static Note getNote(int i) { // si no especifico assumeixo que vull
                                        // de la tonalitat actual
        return getNote(i, tonalitat, mode);
    }

    public static Note getNote(int grau, int nom, int so, int mode) { // munto també la versió sense objecte
nota, que serà útil en segons quins casos
        return getNote(grau, new Note(nom, so), mode);
    }

    public static Note getNote(int grau, Note to, int mode) { // funció per calcular una nota en funció del grau
i la tonalitat
        to = new Note(to.nom, to.so);
        if (mode == 0) { // major
            return addInterval(to, grau, qMaj[grau]);
        } else { // menor
            return addInterval(to, grau, dMin[grau]);
        }
    }

    /**- - - GET CHORD NOTE - - -*/

    public static Note getChordNote(Chord acord, int nota) { // com abans, si no dic res considero la tonalitat
de l'exercici
        return getChordNote(acord, nota, tonalitat, mode);
    }

    public static Note getChordNote(Chord acord, int nota, Note to, int mode) {
        to = new Note(to.nom, to.so);
        Note chordNote;
        // TODO afegir una bona pila de ifs i coses quan calgui calcular acords
        // més estranys
        switch (nota) {
        case 1:
            if (acord.nap) { // Si és un IIb nap.
                chordNote = getNote((acord.grau + 1) % 7, to, mode); // segon grau de l'escala
                chordNote.so -= 1; // rebaixat
            } else { // per la resta d'acords
                if (mode == 1 || acord.hom) { // si són d'una tonalitat menor
                    chordNote = getNote(acord.grau, to, 1); // nota del menor...
                    if (acord.grau == 5 && acord.alt6) { // i ajust de la nota segons alteracions opcionals
                        chordNote.so += 1; // era 6a i estava alterada
                    }
                } else { // si són d'una tonalitat major
                    chordNote = getNote(acord.grau, to, 0); // nota del major (directa, no hi ha opcionals)
                }
            }
            break;
        case 3:
            if (acord.dom) { // si és de la família de la dominant (sempre F+3M)
                chordNote = addInterval(getChordNote(acord, 1, to, mode), 2, qMaj[2]);
            } else { // per la resta d'acords
                if (mode == 1 || acord.hom) { // acords del menor
                    chordNote = getNote((acord.grau + 2) % 7, to, 1); // nota del menor..
                    if (acord.grau == 3 && acord.alt6 // ajust de la nota segons alteracions opcionals
                            || acord.grau == 4 && acord.alt7) {
                        chordNote.so += 1;
                    }
                } else { // acords del major
                    chordNote = getNote((acord.grau + 2) % 7, to, 0); // nota del major (directa, no hi ha opcionals)
                }
            }
            break;
        case 5:
            if (acord.nap || acord.dom) { // si és IIb nap. o és dominant (sempre F+5J)
                chordNote = addInterval(getChordNote(acord, 1, to, mode), 4, qMaj[4]);
            } else {
                if (mode == 1 || acord.hom) { // acords del menor
                    chordNote = getNote((acord.grau + 4) % 7, to, 1); // nota del menor...
                    if (acord.grau == 1 && acord.alt6 // ajust de la nota segons alteracions opcionals
                            || acord.grau == 2 && acord.alt7) {
                        chordNote.so += 1;
                    }
                } else { // acords del major
                    chordNote = getNote((acord.grau + 4) % 7, to, 0); // nota del major (directa, no hi ha opcionals)
                }

            }
            break;
```

```
        case 7:
            if (acord.nap || acord.dom) { // si és IIb nap. o és dominant (sempre F+7m)
                chordNote = addInterval(getChordNote(acord, 1, to, mode), 6, qMin[6]);
            } else { // per la resta d'acords...
                if (mode == 1 || acord.hom) { // acords del menor
                    chordNote = getNote((acord.grau + 6) % 7, to, 1); // nota del menor
                    if (acord.grau == 6 && acord.alt6 // ajust de la nota segons alteracions opcionals
                            || acord.grau == 0 && acord.alt7) {
                        chordNote.so += 1;
                    }
                } else { // acords del major
                    chordNote = getNote((acord.grau + 6) % 7, to, 0); // nota del major (directa, no hi ha opcionals)
                }
            }
            break;
        case 9:
            if (acord.dom) { // comprovo que és de dominant (que té novena)
                if (mode == 1 || acord.hom) { // dominant i ds del menor
                    if (acord.grau == 0 || acord.grau == 1 || acord.grau == 4 || acord.alt9) { // sèptima disminuïda
segons procedència de la ds (i segons alteració escollida).
                        chordNote = addInterval(getChordNote(acord, 3, to, mode), 6, qMin[6] - 1); // sèptima
disminuïdasobre la sensible
                    } else {
                        chordNote = addInterval(getChordNote(acord, 3, to, mode), 6, qMin[6]); // sèptima menor sobre
la sensible
                    }
                } else { // dominant i ds del major
                    if (acord.grau == 2 || acord.grau == 5 || acord.grau == 6 || acord.alt9) {
                        chordNote = addInterval(getChordNote(acord, 3, to, mode), 6, qMin[6] - 1); // sèptima
disminuïda sobre la sensible
                    } else {
                        chordNote = addInterval(getChordNote(acord, 3, to, mode), 6, qMin[6]); // sèptima menor sobre
la sensible
                    }
                }
            } else {
                System.out.println("L'acord no té novena");
                chordNote = null;
            }
            break;
        default:
            System.out.println("La nota demanada no és a l'acord");
            chordNote = null;
            break;
        }
        return chordNote;
    }

    /**- - - ALT CHORD - - -*/

    public static Chord altChord(Chord ac, int alt, boolean b) { // retorna un acord com el del paràmetre, però
amb el canvi d'alteració demanat
        ac = new Chord(ac);
        switch (alt) {
        case 9:
            if (b) {
                ac.alt9 = true;
            } else {
                ac.alt9 = false;
            }
        case 6:
            if (b) {
                ac.alt6 = true;
            } else {
                ac.alt6 = false;
            }
        case 7:
            if (b) {
                ac.alt7 = true;
            } else {
                ac.alt7 = false;
            }
        }
        return ac;
    }

    /**- - - OPCIONALS - - -*/

    public static byte opcionals(Chord ac) { // retorna un codi de dues xifres de la forma [nota][altN]
        if (ac.dom) { // acord de dominant (alt9)
```

```
        if (mode == 1 || ac.hom) { // dominant del menor
            if (ac.grau == 0 || ac.grau == 1 || ac.grau == 4) {
                return 0; // no puc alterar res
            } else {
                return 9; // puc alterar la novena (com ja sé de quina manera no ho envio)
            }
        } else { // dominant del Major
            if (ac.grau == 2 || ac.grau == 5 || ac.grau == 6) {
                return 0;
            } else {
                return 9;
            }
        }
    } else if (mode == 1 || ac.hom) { // és del menor (alt6 i alt7)
        switch (ac.grau) {
        case 0: // I
            if (ac.sept) {
                return 77; // puc alterar la sèptima mitjançant alt7
            } else {
                return 0;
            }
        case 1: // II
            return 56; // la quinta amb alt6
        case 2: // III
            return 57; // la quinta amb alt7
        case 3: // IV
            return 36; // la tercera amb alt6
        case 4: // V
            return 37; // la tercera amb alt7
        case 5: // VI
            return 16; // la primera amb alt6
        case 6: // VII
            if (ac.sept) {
                return 76; // la sèptima amb alt6
            } else {
                return 0; // aquí tècnicament pots fer alt7, però llavors és V9, així que no la poso
            }
        }
    } else { //si no és cap de les dues no tindrà opcionals
        return 0;
    }
    if (ac.grau == 7) { // petita clàusula per gestionar els ingredients de grau X
        return 0;
    }
    System.out.println("No hauries de ser aquí.");
    return 42;
}

/**- - - Text de l'alteració opcional - - -*/
public static String altOpText(Chord ac) { // retorna un text representatiu de l'estat de l'alteració opcional
    switch (opcionals(ac) % 10) {
    case 9:
        if (ac.alt9) {
            return "9-";
        } else {
            return "9=";
        }
    case 6:
        if (ac.alt6) {
            return "6+";
        } else {
            return "6=";
        }
    case 7:
        if (ac.alt7) {
            return "7+";
        } else {
            return "7=";
        }
    default:
        return "";
    }
}


/********* GESTIÓ DE LES TESSITURES ********/

/**- - - DEFAULT IA - - -*/

public static byte defaultIA(Chord acord, int nota, int veu) {
```

```
        return defaultIA(acord, nota, veu, tonalitat);
    }

    public static byte defaultIA(Chord acord, int nota, int veu, Note to) { // índex acústic per defecte que
donaré a una nota segons a quina veu la vulgui
        int nom = (to.nom + (nota - 1) + acord.grau) % 7; // calculo el nom de nota que estic intentant posar
        if (nom < 0) {
            nom += 7; // penso que en alguns casos estranys passa (posar-ho ha arreglat un comportament estrany).
        }
        switch (veu) {
        case 0: // soprano
            if (nom > 3) {
                return 3;
            } else {
                return 4;
            }
        case 1: // contralt
            if (nom > 5) {
                return 2;
            } else {
                return 3;
            }
        case 2: // tenor
            if (nom > 2) {
                return 2;
            } else {
                return 3;
            }
        case 3: // baix
            if (nom > 3) {
                return 1;
            } else {
                return 2;
            }
        default:
            System.out.println("La veu " + veu + " no existeix. Boig");
            return 42;
        }
    }

    /**- - - CANVIA OCTAVA - - -*/

    public static int canviaOctava(int nota, int ia, int veu) {
        ia++; // incremento la octava.
        if (dinsTessitura(nota, ia, veu) > 1) { // si surto més de d'una nota de la tessitura permesa (permeto una
mica de joc aquí perquè
                                                // l'usuari pugui posar notes no permeses i rebre així la correcció).
            ia -= 3; // faig la volta (poso la nota tres octaves més avall, perquè les notes que apareixen més
vegades apareixen en tres ia diferents).
        }
        while (dinsTessitura(nota, ia, veu) < -1) { // com que algunes notes només surten a dues alçades, comprovo
que no
                                                    // m'he passat de greu i en cas que sí ho corregeixo pujant una
octava.
            ia++;
        }
        return ia;
    }

    /**- - - DINS TESSITURA - - -*/ // TODO potser adaptar per tenir en compte les alteracions, però vaja, no
sembla molt rellevant.

    public static int dinsTessitura(Note nota, int veu) {
        return dinsTessitura(nota.nom, nota.ia, veu);
    }

    public static int dinsTessitura(int nota, int ia, int veu) { // retorna 0 si la nota és dins la tessitura de
la
                                                                 // veu, o quants noms de nota fora he quedat en cas
contrari.
        Note itv;
        itv = H.getInterval(tLimit(veu, true), new Note(nota, cromatic(nota), ia));
        if (itv.nom <= 0 && itv.ia <= 0) { // estic per sota el límit superior (o just al límit)
            itv = H.getInterval(tLimit(veu, false), new Note(nota, cromatic(nota), ia));
            if (itv.nom >= 0 && itv.ia >= 0) { // i estic per sobre el límit inferior (o just a l límit)
                return 0;
            } else { // es passa de greu
                return compost(itv); // retorno el nom de l'interval compost, que és la distància real en notes.
            }
        } else { // es passa d'agut
```

```
            return compost(itv);
        }
    }

    /** - - - tLimit - - - */
    public static Note tLimit(int veu, boolean max) { // retorna el límit màxim de la veu si max=true, i el
límit
                                                      // mínim en cas contrari.
        switch (veu) {
        case 0:
            if (max) {
                return tMaxS;
            } else {
                return tMinS;
            }
        case 1:
            if (max) {
                return tMaxA;
            } else {
                return tMinA;
            }
        case 2:
            if (max) {
                return tMaxT;
            } else {
                return tMinT;
            }
        case 3:
            if (max) {
                return tMaxB;
            } else {
                return tMinB;
            }
        default:
            System.out.println("No hauries de demanar veus que no existeixen. Gamberro. " + veu);
            return null;
        }
    }


    /********** POSICIONAMENT D'ELEMENTS MUSICALS **********/

    /**- - - STACK POS - - -*/

    static float stackH;
    static boolean resetStack;

    public static float nStackPos(Note nota) {
        return nStackPos(nota, Pant.dEspai); // si no dic res és el de la pantalla de l'esbós.
    }

    public static float nStackPos(Note nota, float dEspai) { // càlcul de la posició d'una nota dins l'esbós,
tenint en
                                                             // compte les altres que he posat ja de l'acord.
        if (resetStack) {
            stackH = nPos(nota, dEspai);
            resetStack = false;
        } else {
            if (nPos(nota, dEspai) > stackH) {
                stackH = nPos(nota, dEspai);
            } else { // si l'índex ha fet que quedi per sota, afegeixo una
                     // octava
                if (nPos(nota, dEspai) + dEspai * 3.5f > stackH) {
                    stackH = nPos(nota, dEspai) + dEspai * 3.5f;
                } else { // si amb una octava no n'hi ha prou, n'afegeixo dues (no m'estic a muntar un while perquè
no caldrà mai afegir més de dues)
                    stackH = nPos(nota, dEspai) + dEspai * 3.5f * 2f;
                }
            }
        }
        return stackH;
    }

    public static float nStackPos(Note nota, boolean reset) {
        return nStackPos(nota, Pant.dEspai, reset);
    }

    public static float nStackPos(Note nota, float dEspai, boolean reset) { // per si vull triar el reset
directament quan demani la funció.
        if (reset) {
```

```
            resetStack = true;
        } else {
            resetStack = false;
        }
        return nStackPos(nota, dEspai);
    }

    public static float nPos(Note nota) { // posició més greu d'una nota dins l'esbós (només ho utilitzo per
nStackPos)
        return nPos(nota.nom, Pant.dEspai); // si no especifico, vull l'espaiat de la pantalla de l'esbós.
    }

    public static float nPos(Note nota, float dEspai) {
        return nPos(nota.nom, dEspai);
    }

    public static float nPos(int nom) {
        return nPos(nom, Pant.dEspai);
    }

    public static float nPos(int nom, float dEspai) {
        return (dEspai * (nom * 0.5f - 1.5f)); // començo al Do, i el baixo tres notes respecte la referència, que
és el Fa.
    }

    /**- - - NOTE POS - - -*/

    static float lastNotePos;

    public static float notePos(Note nota, int veu) { // per poder entrar nota directa
        return notePos(nota.nom, nota.ia, veu);
    }

    public static float notePos(Note nota, int veu, boolean savePos) { // per poder demanar que no es guardi la
última posició.
        return notePos(nota.nom, nota.ia, veu, savePos);
    }

    public static float notePos(int nom, int ia, int veu) { // polimorfisme per poder posar allà mateix clau de
sol i de fa (així dient la veu ja m'ho separa)
        return notePos(nom, ia, veu, true);
    }

    public static float notePos(int nom, int ia, int veu, boolean savePos) {
        if (veu < 2) {
            return notePos(nom, ia, false, savePos);
        } else {
            return notePos(nom, ia, true, savePos);
        }
    }

    public static float notePos(int nom, int ia, boolean fa) {
        return notePos(nom, ia, fa, Pant2.dEspai);
    }

    public static float notePos(int nom, int ia, boolean fa, boolean savePos) {
        return notePos(nom, ia, fa, Pant2.dEspai, savePos);
    }

    public static float notePos(int nom, int ia, boolean fa, float dEspai) { // posició d'una nota segons índex
acústic
                                                                            // (clau de Sol i clau de Fa)
        return notePos(nom, ia, fa, dEspai, true);
    }

    public static float notePos(int nom, int ia, boolean fa, float dEspai, boolean savePos) {
        float pos;
        if (!fa) { // clau de sol
            pos = dEspai * (nom * 0.5f - 1.5f + (ia - 3) * 3.5f); // començo al Do 3, que és tres notes per sota la
referència
                                                                  // (el fa 3), i ajusto segons l'índex acústic.
        } else { // clau de fa
            pos = dEspai * (nom * 0.5f + 1f + (ia - 2) * 3.5f); // començo al Do 2, que és dues notes per sobre la
referència
                                                                // (La 1), i ajusto segons l'índex acústic.
        }
        if (savePos) {
            lastNotePos = pos;
        }
        return pos;
```

```
    }

    /**- - - AJUST SOLAPADES - - -*/

    static float lastSolapades; // deso l'últim càlcul buscat a partir de l'ajust de solapades

    public static float ajustSolapades(int ac, int veu, boolean save) {
        float ajust = 0;
        if (Pant2.hiHaCor(Pant2.solapades, ac, veu)) {
            ajust = ajustSolapades(F.lastCor.aux);
        }
        if (save) {
            lastSolapades = ajust;
        }
        return ajust;
    }

    public static float ajustSolapades(int ac, int veu) { // per defecte deso el càlcul (em serveix per les
línies addicionals i tal
        return ajustSolapades(ac, veu, true);
    }

    public static float ajustSolapades(int itv) { // retorna quantes vegades he d'afegir el diàmetre de
                                                  // la nota a la posició x per evitar que se solapin.
        if (itv == 1) { // separades una segona
            return 0.87f;
        } else {
            return 1f;
        }
    }

    public static float ySolFa(int veu) { // per poder treure factor comú dels dibuixos de línies addicionals i
coses per l'estil
        if (veu < 2) {
            return Pant2.ySol;
        } else {
            return Pant2.yFa;
        }
    }

    /**- - - SOBRE LINIA - - -*/
    public static byte sobreLinia(Note nota) {
        return sobreLinia(nota.nom, nota.ia);
    }

    public static byte sobreLinia(int nom, int ia) { // retorna 1 si la nota és sobre una línia i 0 si és sobre
un espai.
        if ((nom + ia) % 2 == 0) { // A cada nota canvia. Com que do4 és un espai, (0+4)%2=0 => 0 = espai
            return 0; // A més, el Do3 és en una línia a les dues claus, així que la fórmula es manté per als dos
casos.
        } else {
            return 1;
        }
    }

    /**- - - ARM POS - - -*/

    public static float armPos(int nom, boolean sostinguts) {
        return armPos(nom, sostinguts, Pant.dEspai);
    }

    public static float armPos(int nom, boolean sostinguts, float dEspai) { // armadures en clau de sol.
        if (sostinguts) {
            return (dEspai * (((nom + 2) % 7) * 0.5f + 1f)); // començo al La, i el pujo dues notes respecte la
referència, que és el Fa.
        } else {
            return (dEspai * (((nom + 4) % 7) * 0.5f)); // començo al Fa, i com és la referència ja no el pujo.
        }
    }

    public static float armPos(int nom, boolean sostinguts, boolean fa) { // per especificar quina de les dues
claus vull. En aquest cas, com que especifico clau, entenc que
                                                                    // em refereixo a l'esquema a quatre veus i agafo
per tant la distància d'allà.
        if (fa) {
            return (armPos(nom, sostinguts, Pant2.dEspai) - Pant2.dEspai); // si estic en clau de fa, baixo tot
dues notes (un espai o línia).
        } else {
            return armPos(nom, sostinguts, Pant2.dEspai);
        }
```

```
    }
}
```

A.1.12. [hEditor.java] – Unitat central de l'aplicació

```
package com.rusca8.hEditor;

import com.badlogic.gdx.Game;
import com.rusca8.hEditor.PIng;
import com.rusca8.hEditor.Pant;
import com.rusca8.hEditor.Pant2;

public class hEditor extends Game {
    PIng pIng; //deixo aquí les pantalles que no voldré crear cada vegada que les necessiti (han de guardar info).
    Pant pant;
    Pant2 pant2;

    @Override
    public void create () {
        setScreen(new PInici(this)); //escullo la pantalla inicial (aquesta no em cal tenir-la desada a dalt)
    }
}
```

A.1.13. [IngAcord.java] – Ingredient de tipus acord

```
package com.rusca8.hEditor;

import com.rusca8.hEditor.Chord;
import com.rusca8.hEditor.IngAcord;

public class IngAcord { //ingredient de tipus acord
    int id; //número de referència de l'ingredient (començarà a 1, perquè necessito el zero lliure per eAmb)

    byte n=1; //quantitat de vegades que haurà d'aparèixer l'ingredient

    Chord acord; //acord base

    boolean triada = false; //si el vull específicament tríade (si no dic triada ni acord.sept, llavors em serveix
qualsevol)
    boolean alt = false; //si vull específicament l'alteració que hi ha escollida
    boolean septMajor = false; //si vull sèptima major (vàlid només per X, perquè els altres ja la tenen fixada)

    boolean difAc = false; //diferents (diferent acord)
    boolean difInv = false; //diferent inversió
    boolean enll = false; //enllaçades (per ingredients tipus "2xDS enllaçades")
    boolean tcp = false; //"tants com puguis" (marca per quan no hi ha quantitat concreta, però se'n vol densitat)

    int eAmb = 0; //enllaça amb... (per generar ingredients compostos tipus "7ds a CT tríade").

    public IngAcord(int id){ //constructor
        this.id = id;

        acord = new Chord(7); //poso per defecte grau "qualsevol" (0-6 són I-VII, i 7 és "qualsevol" (X))
    }

    @Override
    public boolean equals(Object obj){ //redefineixo la funció de comparar (amb els objectes és necessari per tal
que funcioni bé).
        if(!(obj instanceof IngAcord)){
            System.out.println("L'objecte no és un ingredient tipus acord.");
            return false;
        }
        IngAcord iAc2 = (IngAcord) obj;

        return (this.id==iAc2.id); //com que porten l'identificador únic, seran iguals si els id coincideixen.
    }
}
```

A.1.14. [IngCadencia.java] – Ingredient de tipus cadència

```
package com.rusca8.hEditor;

import com.rusca8.hEditor.IngCadencia;

public class IngCadencia { //ingredient de tipus cadència (explicacions a IngAcord)
    int id;

    byte n=1;
```

```
    boolean triada = false;
    boolean sept = false;

    boolean dif = false; //diferents (5/7)... TODO més endavant V/DS (potser afegir nova variable com amb
difAc/difInv)
    boolean tcp;

    int eAmb = 0;

    public IngCadencia(int id){
        this.id = id;
    }

    @Override
    public boolean equals(Object obj){
        if(!(obj instanceof IngCadencia)){
            System.out.println("L'objecte no és un ingredient tipus cadència.");
            return false;
        }
        IngCadencia iCad2 = (IngCadencia) obj;

        return (this.id==iCad2.id);
    }
}
```

A.1.15. [IngRes.java] – Ingredient de tipus resolució irregular

```
package com.rusca8.hEditor;

import com.rusca8.hEditor.IngRes;

public class IngRes { //ingredient tipus resolució irregular
    int id;

    boolean sens; //per indicar resolució irregular de la sensible

    public IngRes(int id){
        this.id = id;
    }

    @Override
    public boolean equals(Object obj){
        if(!(obj instanceof IngRes)){
            System.out.println("L'objecte no és un ingredient tipus RI.");
            return false;
        }
        IngRes iRI2 = (IngRes) obj;

        return (this.id==iRI2.id);
    }
}
```

A.1.16. [Log.java] – Unitat bàsica del registre

```
package com.rusca8.hEditor;

public class Log {
    byte nAc;
    int ref;

    public Log(int nAc, int ref){
        this.nAc = (byte) nAc;
        this.ref = ref;
    }
}
```

A.1.17. [N.java] – Recull de la normativa i funcions associades

```
package com.rusca8.hEditor;

import com.badlogic.gdx.utils.Array;

public class N { //normativa d'harmonia

    /********* LLISTA REF <--> NORMA **********/
    static Array<Norma> normes = new Array<Norma>();
```

```
    public static void carrega(){
        normes.add(new Norma(0,"Encara no hi ha redactada la norma corresponent."));
        //Tessitures
        normes.add(new Norma(1,"N1.1: Les notes assignades al baix hauran d'estar entre Fa1 i Do3."));
        normes.add(new Norma(2,"N1.2: Les notes assignades al tenor hauran d'estar entre Do2 i Sol3."));
        normes.add(new Norma(3,"N1.3: Les notes assignades a la contralt hauran d'estar entre Fa2 i Do4."));
        normes.add(new Norma(4,"N1.4: Les notes assignades a la soprano hauran d'estar entre Do3 i La4."));

        //Distància i creuaments
        normes.add(new Norma(5,"N2: Entre dues veus adjacents no hi podrà haver mai més d'una octava, excepte si és
entre el Tenor i el Baix."));
        normes.add(new Norma(6,"N3: No hi podrà haver mai creuaments entre dues veus:"
                            + "\n\nN3.1: Per tot acord, ordenar les veus segons l'alçada de les notes que han de
cantar sempre ha de resultar en l'ordenació estàndard del cor mixt (d'agut a greu, SATB)."));
        normes.add(new Norma(7,"N3: No hi podrà haver mai creuaments entre dues veus:"
                            + "\n\nN3.2: A cap veu se li podrà assignar en un acord una nota que estigui fora els
límits marcats per les notes que cantaven les veus adjacents a l'acord anterior."));

        //Moviment paral·lel
        normes.add(new Norma(8,"N4.1: No podrem fer octaves seguides."));
        normes.add(new Norma(9,"N4.2: No podrem fer sèptimes seguides."));
        normes.add(new Norma(10,"N4.3: No podrem fer quintes seguides, excepte en els casos següents:"
                            + "\n\nN4.3.1: Podrem fer quintes seguides si aquestes comparteixen un dels sons."
                            + "\n N4.3.2: Podrem fer quintes seguides si aquestes tenen qualificatius diferents."
                            + "\n N4.3.3: Podrem fer quintes seguides si l'interval entre aquestes és d'un
semitò."));

        //Augmentats i disminuïts
        normes.add(new Norma(11,"N5.1: El pas d'una veu d'un acord al següent no podrà ser mai en forma d'interval
augmentat."));
        normes.add(new Norma(12,"N5.2: El pas d'una veu d'un acord al següent podrà ser en forma d'interval
disminuït sempre i quan es resolgui posteriorment en direcció contrària per graus conjunts."));

        //Duplicacions i supressions
        normes.add(new Norma(102,"N6.1: No podrem duplicar la sensible de la tonalitat ni la sèptima de
l'acord."));
        normes.add(new Norma(103,"N6.2: Només podrem duplicar el baix d'un acord en primera inversió tríada si
aquest baix és un dels graus tonals (I, IV o V)"));
        normes.add(new Norma(104,"N7: Si és necessari podrem suprimir la quinta de l'acord, però mai la fonamental
o la tercera."));

        //Utilització bàsica dels graus
        normes.add(new Norma(105,"N8: Els esquemes hauran de començar en acord de tònica tríada en estat
fonamental."));
        normes.add(new Norma(101,"N9: No posarem mai el VII si porta sensible (l'acord que representa serà en
realitat un V de dominant)."
                            + "\n N9.1: En menor podrem posar la versió del VII que porta subtònica."));

        //Enllaços
        normes.add(new Norma(13,"N10: Els enllaços dèbils els haurem de resoldre. Tenim dues opcions:"
                            + "\n Opció 1: Tornem al mateix acord del qual havíem partit (\"harmonia de
brodadura\")."
                            + "\n Opció 2: Fem un enllaç fort entre el primer acord i el tercer (\"harmonia de
pas\")."));
        normes.add(new Norma(14,"N11: No posarem mai dos enllaços molt forts consecutius."));

        //CP
        normes.add(new Norma(15,"N12: Els esquemes sempre hauran d'acabar amb una Cadència Perfecta (SD, I_6/4,
V_EF, I(5)_EF)."));

        //Comportament de la dominant
        normes.add(new Norma(16,"N13: La dominant per defecte sempre anirà a tònica (V -> I)."
                            + "\n N9.1: Si la dominant no va a tònica, haurà d'anar a algun dels seus
substituts."));
        normes.add(new Norma(17,"N13.2: La dominant podrà fer una Cadència Trencada (V -> IV) o (V -> VI), sempre i
quan se segueixin les normes següents:"
                            + "\n N13.2.1: Haurem de duplicar la tònica al segon acord."
                            + "\n N13.2.2: Haurem de fer els mínims moviments possibles dins de cada veu."));

        normes.add(new Norma(501,"NX1: Podem fer anar el V a II, però després haurà de tornar a V (amplificació de
dominant).")); //TODO posar això completat del tot on toqui quan toqui posar-ho (de moment ho deixo aquí mig a
mitges perquè ho necessito en algun lloc donat que permeto aplicar-ho a l'esbós).
        normes.add(new Norma(502,"NX2: Podem fer anar el V a III- (DS), sempre i quan després posem un VI (CT amb
DS al mig)."));

        //Melodia
        normes.add(new Norma(18,"N14.1.1: Els salts de la melodia els considerarem desequilibris. Per equilibrar-
los, tornarem al punt d'origen per graus conjunts."));
        normes.add(new Norma(19,"N14.1.2: A la melodia mai farem salts seguits en la mateixa direcció."));
        normes.add(new Norma(20,"N14.1.3: La \"gràfica\" de la melodia haurà de ser corba (no dentada)"));
```

```
        normes.add(new Norma(21,"N14.2: El \"pinyol\" (la nota més alta) no es repetirà mai"));
        normes.add(new Norma(22,"N14.5: La melodia sempre acabarà en tònica (C.P. - N8)."));

        //Sèptima de dominant
        normes.add(new Norma(23,"N15.1: Per preparar la sèptima una de les dues notes (sèptima o fonamental) haurà
d'entrar lligada o per graus conjunts."));
        normes.add(new Norma(231, "N15.1.1: Si arribem a la sèptima a través d'un salt, aquest no podrà ser mai
descendent.")); //ref d'aquestes normes a baix. Solucionat amb la funció de context.
        normes.add(new Norma(24,"N15.2: Haurem de resoldre la sèptima segons una d'aquestes tres opcions:"
                            + "\n Opció 1: Fer baixar la sèptima per graus conjunts."
                            + "\n Opció 2: Mantenir la sèptima aguantada."
                            + "\n Opció 3: Pujar la sèptima cromàticament."));
        normes.add(new Norma(25,"N15.4: Si tenim una bona raó, podrem saltar-nos la resolució habitual de la
sèptima. Tot i això:"
                            + "\n N15.4.1: La sèptima s'haurà de poder resoldre igualment, malgrat resolgui en una
altra veu (RI7)."));
        normes.add(new Norma(251,"N15.4.2: Farem resolucions irregulars de la sèptima un màxim de dues vegades per
exercici."));

        //Acords de 6/4
        normes.add(new Norma(26,"N16.1: En els acords tríades en segona inversió (6/4), tant el baix com la quarta
hauran d'entrar i sortir lligats o per graus conjunts."));
        normes.add(new Norma(261,"N16.1.1: Si la quarta està duplicada, només caldrà preparar i resoldre una de les
dues aparicions de la nota (i podem preparar-ne una i resoldre l'altra)."));
        normes.add(new Norma(27,"N16.2: No posarem mai dos acords de 6/4 consecutius."));

        //Mode menor
        normes.add(new Norma(28,"N17: Haurem de vigilar amb l'interval entre la segona i la sèptima als modes
menors:"
                            + "\n N17.1: Si apareixen seguides i el moviment és ascendent, pujarem la 6a (i posarem
sensible)."
                            + "\n N17.2: Si apareixen seguides i el moviment és descendent, posarem subtònica (i la
6a natural)."));
        normes.add(new Norma(29,"N18: L'acord de dominant sempre portarà sensible (en cas contrari no serà de
domninant)."));

        //Sèptimes en general
        normes.add(new Norma(31,"N19.2: Si la sèptima de l'acord és Major, l'haurem de preparar fent entrar una de
les dues notes lligada."));
        normes.add(new Norma(32,"N19.3: Si posem sèptima als dos acords de la cadència trencada, només tindrem en
compte el comportament de les sèptimes (deixarem de banda N9.2.1 i N9.2.2)."));

        /** - - Indicacions per a les alertes - - */
        normes.add(new Norma(1001,"Compte! La inversió que has realitzat no és la que havies planejat a
l'esbós!"));
        normes.add(new Norma(1006,"* L'aplicació no assigna la nota més aguda a la veu més aguda: assegura't que
les assignes a la veu adient."));
        normes.add(new Norma(1017,"Compte! La teva resolució de la C.T. mou més les veus del que s'esperaria.\nSi
el context no és estrany, pot ser que estigui malament."));
        normes.add(new Norma(1102,"Duplicació poc habitual (tot i que no necessàriament incorrecta)."));
        normes.add(new Norma(1104,"Acord incomplet en ubicació poc habitual. Si no és per un bon motiu no hauries
de suprimir la quinta."));

    }

    public static String ambContext(int num){ //per les normes que va bé que tinguin alguna altra norma com a
context
        switch(num){
        case 261:
            return (textDe(26)+"\n"+textDe(261));
        case 1006:
            return (textDe(6)+"\n"+textDe(1006));
        default:
            return textDe(num);
        }
    }

    public static String textDe(int num){
        return normes.get(normes.indexOf(new Norma(num), false)).text;
    }
}
```

A.1.18. [Norma.java] – Unitat bàsica de la normativa

```
package com.rusca8.hEditor;

import com.rusca8.hEditor.Norma;

public class Norma {
```

```
    int ref; //número de referència únic de cada norma per referència
    String text; //text de la norma (explicació)

    public Norma(int ref, String text){
        this.ref = ref;
        this.text = text;
    }

    public Norma(int ref){
        this.ref = ref;
    }

    @Override
    public boolean equals(Object obj){
        if(!(obj instanceof Norma)){
            System.out.println("L'objecte no és una norma.");
            return false;
        }
        Norma norma2 = (Norma) obj;

        return (this.ref==norma2.ref); //només m'interessa si són la mateixa norma (si tenen la mateixa
referència)
    }
}
```

A.1.19. [Note.java] – Element tipus nota

```
package com.rusca8.hEditor;

import com.rusca8.hEditor.Norma;

public class Norma {
    int ref; //número de referència únic de cada norma per referència
    String text; //text de la norma (explicació)

    public Norma(int ref, String text){
        this.ref = ref;
        this.text = text;
    }

    public Norma(int ref){
        this.ref = ref;
    }

    @Override
    public boolean equals(Object obj){
        if(!(obj instanceof Norma)){
            System.out.println("L'objecte no és una norma.");
            return false;
        }
        Norma norma2 = (Norma) obj;

        return (this.ref==norma2.ref); //només m'interessa si són la mateixa norma (si tenen la mateixa
referència)
    }
}
```

A.1.20. [PAbout.java] – Pantalla "Quant a..."

```
package com.rusca8.hEditor;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.InputAdapter;
import com.badlogic.gdx.Screen;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.OrthographicCamera;
import com.badlogic.gdx.graphics.g2d.SpriteBatch;
import com.badlogic.gdx.math.Vector3;
import com.badlogic.gdx.utils.Align;
import com.badlogic.gdx.utils.viewport.ExtendViewport;
import com.badlogic.gdx.utils.viewport.Viewport;

public class PAbout extends InputAdapter implements Screen {
    hEditor app;
    PMenu pMenu;

    OrthographicCamera camera;
    Viewport encuadre;
    SpriteBatch lote;
```

```
    Vector3 click = new Vector3(0,0,0);

    byte veient=0; // 0 = dev / 1 = app / 2 = Rec

    float wBarra = 8f;

    BotQ bJo; //botó sobre el desenvolupador
    BotQ bHE; //botó sobre l'aplicació
    BotQ bRs; //botó sobre els recursos

    BotQ bTwit; //botons d'enllaços per la secció Jo
    BotQ bGit;

    BotQ bMenu; //botó tornar al menú
    float mM=8f; //marge del botó del menú

    public PAbout(hEditor app, PMenu pMenu){
        this.app = app;
        this.pMenu = pMenu; //deso referència a la pantalla d'on vinc per quan vulgui tornar

        lote = new SpriteBatch();

        camera = new OrthographicCamera();
        encuadre = new ExtendViewport(0,Pant.screenH,camera);
        encuadre.apply(false); //explicacions de tot això a "Pant"

        bJo = new BotQ(150,Pant.screenH*4.2f/5.2f,200,200); //fer meitat i cinquens divideix guai
        bHE = new BotQ(150,Pant.screenH*0.5f,200,200);
        bRs = new BotQ(150,Pant.screenH*1/5.2f,200,200);

        //compte, coses a resize
        bMenu = new BotQ((Pant.realW*Pant.ratio+bJo.x*2f+wBarra)/2f, bRs.y-bRs.h/2f+40, (Pant.realW*Pant.ratio-
(bJo.x*2f+wBarra))*0.85f, 80, "Tornar al menú");

        //específic secció Jo
        bTwit = new BotQ(bMenu.x+bMenu.w/2f-55f,2.5f*bMenu.y,110f,110f);
        bGit = new BotQ(bMenu.x+bMenu.w/2f-190f,2.5f*bMenu.y,110f,110f);
    }

    @Override
    public void render(float delta) {
        delta = Math.min(delta, 0.25f);

        Gdx.gl.glClearColor(S.fons.r,S.fons.g,S.fons.b,S.fons.a);
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);

        camera.update();
        lote.setProjectionMatrix(camera.combined);

        lote.begin();

            /**- - elements comuns - -*/
            lote.setColor(S.graus);
            lote.draw(Recursos.rect, 2*bJo.x, bRs.y-bRs.w/2f+wBarra/2f, wBarra, bJo.y-bRs.y+bJo.h-wBarra);
            lote.draw(Recursos.cir, 2*bJo.x, bJo.y+bJo.w/2f-wBarra, wBarra, wBarra);
            lote.draw(Recursos.cir, 2*bJo.x, bRs.y-bRs.w/2f, wBarra, wBarra);

            /**- - Text dels apartats - -*/
            Recursos.font.setColor(S.graus);
            Recursos.iQuant.setColor(S.graus);
            if(veient==0){
                Recursos.font.draw(lote,"Quant al desenvolupador", 2.4f*bJo.x,bJo.y+bJo.h/2f);
                Recursos.iQuant.draw(lote,"Em dic David Ruscalleda, i he estat estudiant un Grau en Tecnologies
Industrials, així com el Grau Professional de Música al Conservatori."
                        + "\n\nHemiolia Editor és el meu TFG, així com la meva segona aplicació per mòbils, després
del joc matemàtic \"Primes!\", publicat l'estiu de 2015.",
                        2.4f*bJo.x,Pant.screenH*0.82f,bMenu.w,Align.left,true);
            }else if(veient==1){
                Recursos.font.draw(lote,"Quant a l'aplicació", 2.4f*bJo.x,bJo.y+bJo.h/2f);
                Recursos.iQuant.draw(lote,"Hemiolia Editor neix de les ganes d'oferir una manera ràpida i senzilla
d'ampliar i de posar a prova els teus coneixements d'harmonia sense haver de necessitar "
                        + "l'experiència d'algú extern que et pugui corregir."
                        + "\n\nSi bé de moment l'aplicació funciona només amb els acords de la tonalitat sense
ampliar, es preveu en un futur la incorporació de D.S., acords de l'homònim i modulacions, entre d'altres.",
                        2.4f*bJo.x,Pant.screenH*0.82f,bMenu.w,Align.left,true);
            }else if(veient==2){
                Recursos.font.draw(lote,"Quant als recursos", 2.4f*bJo.x,bJo.y+bJo.h/2f);
                Recursos.iXifrat.setColor(S.graus);
                Recursos.iXifrat.draw(lote,"El logotip de l'aplicació (La clau flamenc) l'he dissenyat jo mateix, "
```

```
                    + "i les imatges d'elements musicals (claus i alteracions) provenen de Wikimedia Commons."
                    + "\n\nEls fitxers d'àudio provenen de la Philarmonia Orchestra (CC BY-SA 3.0), tot i que els
he modificat per ajustar-ne el moment d'inici i per obtenir algunes notes que faltaven."
                    + "\n\nLa font utilitzada és \"Open Sans\", de Steve Matteson (Apache 2.0), "
                    + "i algunes altres icones venen de icons8.com (CC BY-ND 3.0).",
                    2.4f*bJo.x,Pant.screenH*0.82f,bMenu.w,Align.left,true);
        }

        /****** BOTONS ******/
        /**- - Botons de l'esquerra - -*/

        //avatar
        if(veient==0){
            lote.setColor(S.actual);
        }else{
            lote.setColor(S.graus);
        }
        lote.draw(Recursos.cir, bJo.x-bJo.w/2f, bJo.y-bJo.h/2f, bJo.w, bJo.h);
        if(veient==0){
            lote.setColor(S.blanc);
        }else{
            lote.setColor(0.7f,0.7f,0.7f,1f);
        }
        lote.draw(Recursos.avatar, bJo.x-bJo.w/2f+12, bJo.y-bJo.h/2f+12, bJo.w-24, bJo.h-24);

        //logo HE
        if(veient==1){
            lote.setColor(S.b_marc_ressaltat);
        }else{
            lote.setColor(S.b_marc);
        }
        lote.draw(Recursos.cir, bHE.x-bHE.w/2f, bHE.y-bHE.h/2f, bHE.w, bHE.h);
        if(veient==1){
            lote.setColor(S.b_premut);
        }else{
            lote.setColor(S.b_normal);
        }
        //part interior de la icona de la app
        lote.draw(Recursos.launcher, bHE.x-bHE.w/2f, bHE.y-bHE.h/2f, bHE.w, bHE.h);

        //carpeta (recursos)
        if(veient==2){
            lote.setColor(S.b_marc_ressaltat);
        }else{
            lote.setColor(S.b_marc);
        }
        lote.draw(Recursos.cir, bRs.x-bRs.w/2f, bRs.y-bRs.h/2f, bRs.w, bRs.h);
        if(veient==2){
            lote.setColor(S.b_premut);
        }else{
            lote.setColor(S.b_normal);
        }
        lote.draw(Recursos.cir, bRs.x-bRs.w/2f+12, bRs.y-bRs.h/2f+12, bRs.w-24, bRs.h-24);
        if(veient==2){
            lote.setColor(S.b_marc_ressaltat);
        }else{
            lote.setColor(S.b_marc);
        }
        lote.draw(Recursos.folder, bRs.x-bRs.w/2f+40, bRs.y-bRs.h/2f+40, bRs.w-80, bRs.h-80);

        //Tornar al menú
        lote.setColor(S.b_marc);
        lote.draw(Recursos.rect, bMenu.x-bMenu.w/2f+bMenu.h/2f, bMenu.y-bMenu.h/2f, bMenu.w-bMenu.h, bMenu.h);
        lote.draw(Recursos.cir, bMenu.x-bMenu.w/2f, bMenu.y-bMenu.h/2f, bMenu.h, bMenu.h);
        lote.draw(Recursos.cir, bMenu.x+bMenu.w/2f-bMenu.h, bMenu.y-bMenu.h/2f, bMenu.h, bMenu.h);
        lote.setColor(S.b_normal);
        lote.draw(Recursos.rect, bMenu.x-bMenu.w/2f+bMenu.h/2f, bMenu.y-bMenu.h/2f+mM, bMenu.w-bMenu.h, bMenu.h-
2*mM);
        lote.draw(Recursos.cir, bMenu.x-bMenu.w/2f+mM, bMenu.y-bMenu.h/2f+mM, bMenu.h-2*mM, bMenu.h-2*mM);
        lote.draw(Recursos.cir, bMenu.x+bMenu.w/2f-bMenu.h+mM, bMenu.y-bMenu.h/2f+mM, bMenu.h-2*mM, bMenu.h-
2*mM);
        Recursos.bNotes.setColor(S.b_text);
        Recursos.bNotes.draw(lote,F.keep(bMenu.text),bMenu.x+F.align("bNotes"),bMenu.y+F.vAlign("bNotes"));

        //específic secció Jo
        if(veient==0){
            lote.setColor(S.graus);
            lote.draw(Recursos.cir, bTwit.x-bTwit.w/2f, bTwit.y-bTwit.h/2f, bTwit.w, bTwit.h);
            lote.setColor(S.fons);
            lote.draw(Recursos.twitter, bTwit.x-bTwit.w/2f, bTwit.y-bTwit.h/2f-6, bTwit.w+5, bTwit.h+5);
```

```
            lote.setColor(S.graus);
            lote.draw(Recursos.github, bGit.x-bGit.w/2f, bGit.y-bGit.h/2f, bGit.w, bGit.h);
        }
    lote.end();
}

@Override
public boolean touchDown(int screenX, int screenY, int pointer, int button) {
    click.x = screenX;
    click.y = screenY;
    encuadre.unproject(click);

    if(click.x<bJo.x+bJo.w/2){ //estalvio el càlcul si no és al marge
        if(Math.pow(click.x-bJo.x, 2)+Math.pow(click.y-bJo.y, 2)<Math.pow(bJo.w/2f,2)){ //dins el cercle
            System.out.println("Veure desenvolupador");
            veient=0;
        }else if(Math.pow(click.x-bHE.x, 2)+Math.pow(click.y-bHE.y, 2)<Math.pow(bHE.w/2f,2)){ //dins el cercle
            System.out.println("Veure aplicació");
            veient=1;
        }else if(Math.pow(click.x-bRs.x, 2)+Math.pow(click.y-bRs.y, 2)<Math.pow(bRs.w/2f,2)){ //dins el cercle
            System.out.println("Veure recursos");
            veient=2;
        }
    }else{
        if(click.y<bMenu.y+bMenu.h/2f && click.x>bMenu.x-bMenu.w/2f && click.x<bMenu.x+bMenu.w/2f){
            System.out.println("Tornar al menú");
            this.dispose(); //em desfaig dels recursos que no necessitaré a fora
            app.setScreen(pMenu);
        }else{
            if(veient==0){
                if(Math.pow(click.x-bTwit.x, 2)+Math.pow(click.y-bTwit.y, 2)<Math.pow(bTwit.w/2f,2)){
                    System.out.println("Obrint Twitter en finestra web externa.");
                    Gdx.net.openURI("https://twitter.com/Rusca8");
                }else if(Math.pow(click.x-bGit.x, 2)+Math.pow(click.y-bGit.y, 2)<Math.pow(bGit.w/2f,2)){
                    System.out.println("Obrint Github en finestra web externa.");
                    Gdx.net.openURI("https://github.com/Rusca8");
                }
            }
        }
    }
    return true;
};

@Override
public void resize(int width, int height) {
    encuadre.update(width, height);

    Pant.realW = Gdx.graphics.getWidth();
    Pant.realH = Gdx.graphics.getHeight();
    Pant.ratio = (float) Pant.screenH/Pant.realH;

    bMenu.x = (int) ((Pant.realW*Pant.ratio+bJo.x*2f+wBarra)/2f);
    bMenu.w = (int) ((Pant.realW*Pant.ratio-(bJo.x*2f+wBarra))*0.85f);

    bTwit.x = (int) (bMenu.x+bMenu.w/2f-55);
    bGit.x = (int) (bMenu.x+bMenu.w/2f-190);

    camera.position.set(Pant.realW/2f*Pant.ratio,Pant.screenH/2f,0);
}
@Override
public void show() {
    Gdx.input.setInputProcessor(this);
}

@Override
public void pause() {
}
@Override
public void resume() {
}
@Override
public void hide() {
}

@Override
public void dispose() {
    Recursos.tex_avatar.dispose();
    Recursos.tex_launcher.dispose();
    Recursos.tex_folder.dispose();
```

```
        Recursos.tex_twitter.dispose();
        Recursos.tex_github.dispose();
        lote.dispose();
    }

}
```

A.1.21. [Pant.java] – Pantalla de l'esbós

```
package com.rusca8.hEditor;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.InputAdapter;
import com.badlogic.gdx.Screen;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.OrthographicCamera;
import com.badlogic.gdx.graphics.g2d.SpriteBatch;
import com.badlogic.gdx.math.Vector3;
import com.badlogic.gdx.utils.Align;
import com.badlogic.gdx.utils.Array;
import com.badlogic.gdx.utils.viewport.ExtendViewport;
import com.badlogic.gdx.utils.viewport.Viewport;
import com.rusca8.hEditor.A;
import com.rusca8.hEditor.BotQ;
import com.rusca8.hEditor.Cadencia;
import com.rusca8.hEditor.Chord;
import com.rusca8.hEditor.Desplegat;
import com.rusca8.hEditor.F;
import com.rusca8.hEditor.H;
import com.rusca8.hEditor.Log;
import com.rusca8.hEditor.Pant2;
import com.rusca8.hEditor.Recursos;
import com.rusca8.hEditor.S;
import com.rusca8.hEditor.hEditor;

public class Pant extends InputAdapter implements Screen {
    hEditor app;

    OrthographicCamera camera; //càmera per grabar coses i traduïr-ne les mides a les mides del mòbil
    Viewport encuadre; //per no perdre les proporcions de les coses que grabo al moment d'ensenyar-les pel mòbil
    SpriteBatch lote; //gestor dels lots d'imatges i text que renderitzaré

    static int screenH = 800; //mides del meu món (la horitzontal no em cal degut al tipus de viewport)
    static int realW = Gdx.graphics.getWidth(); //mides del mòbil
    static int realH = Gdx.graphics.getHeight();
    static float ratio = (float) screenH/realH; //proporció que utilitzo per ajustar les mides del meu món a les
mides del mòbil

    Vector3 click = new Vector3(0,0,0); //posició dels clicks.
    Vector3 click2 = new Vector3(0,0,0); //posició del cursor després d'arrossegar un click.

    int yEsb = screenH/3+200; //alçada de referència marcada per la línia més baixa del pentagrama
    static int dEAc = 150; //distància entre acords (per defecte 100)
    static float dEspai = 30f; //distància entre línies del pentagrama
    float dNota = dEspai*1.22f;
    float dAlt = dNota*1.4f; //distància entre l'alteració i la nota. TODO s'haurà d'ajustar d'alguna manera quan
toquin entre elles... (més o menys fet, però en segons quin cas igual no acaba de quadrar).
    float dArm = dEspai*0.7f; //distància entre alteracions de l'armadura

    int rLim = (int) (realW*ratio-135); //límits de l'scroll (no inclou clau ni armadura) (Compte que són al
resize).
    int lLim = 250;
    int scroll = lLim; //posició del primer acord de l'esquema.
    int vScroll = 0; //velocitat instantània de scroll. (de moment no ho he fet servir)
    int aScroll = 3; //acceleració de frenada de l'scroll.

    int ii = 0; //iterador global
    byte nAc = 0; //quantitat de notes a pintar en un acord de l'esbós

    static int fila=0; //posicionament d'ingredients (trec la variable fora per accedir des de la realitz.)

    /** - - - - Esbós i correcció - - - - */
    static Array<Chord> esbos = new Array<Chord>(); //Aquest tipus d'Array són els propis de LibGDX.
    static byte aAct; //acord actual
    boolean insert=true; //per marcar si es canviarà l'acord actual o se n'afegirà un al darrere

    Array<Byte> links = new Array<Byte>(); //llista d'enllaços entre els acords
    static Array<Cadencia> cadencies = new Array<Cadencia>(); //llista de cadències
    static boolean tincCP=false;
```

```
    Array<Character> cLinks = new Array<Character>(); //correcció dels enllaços
    Array<Character> cGraus = new Array<Character>(); //correcció dels graus (resolució de dominants i coses així).
    Array<Character> cInv = new Array<Character>(); //correcció de les inversions.

    static Array<Log> logE = new Array<Log>(); //per deixar constància de la normativa que causa l'error comès
    static Array<Log> logA = new Array<Log>(); //mateix per alertes TODO implementar a la correcció

    int neLinks = 0; //quantitat d'errors d'enllaç
    int neGraus = 0; //quantitat d'errors de grau
    int neInv = 0; //quantitat d'errors d'inversió
    int nErrors = 0; //quantitat total d'errors (accediré des de la realització)

    /** - - - Coses de botons - - - - */
    Array<BotQ> bGraus = new Array<BotQ>();
    BotQ bBorrar;
    BotQ b7a,bDom,bInv, bAlt;
    boolean puc7a,pucDom,pucInv; //booleà que marca si puc polsar el botó (si el puc fer servir o si està
desactivat).
    boolean pucAlt6, pucAlt7, pucAlt9; //per marcar quines alteracions opcionals puc posar en un acord donat
(desactiva el botó de Alt, també).  TODO en algun moment mirar de canviar-ho per H.opcionals()
    BotQ bMenu;

    boolean clickBot = false; //per marcar a la funció touchDown que ja he executat un click.
    boolean heArrossegat = false;
    boolean calCorregir = false;

    //botons del menú
    boolean mostraMenu = false;
    BotQ bMesTo, bMenysTo, bMode; //botons que faré servir de moment per permetre canviar el to de l'esquema des
del menú.
    BotQ bDesa, bTorna; //botons que faré servir per confirmar o descartar els canvis que es facin en el menú.

    int dBM = 120; //distància entre botons del menú
    int yAvisos = (int) (screenH*0.5f); //alçada del text que avisa coses quan premo botons.
    BotQ bIngs; //botó per tornar a editar l'enunciat.
    BotQ bAcaba; //botó per passar a fer la realització a quatre veus.
    BotQ bCancel; //botó per cancel·lar el salt a la realització a quatre veus o el d'anar als ingredients
    boolean vullAcabar; //per poder demanar confirmació abans de saltar a la realització.
    boolean vullIngs; //per demanar confirmació

    /****** CONSTRUCTOR ******/
    public Pant(hEditor app){
        this.app = app;

        lote = new SpriteBatch();

        camera = new OrthographicCamera();
        encuadre = new ExtendViewport(0, screenH, camera); //Extend escala proporcionalment fins tocar les vores de
la pantalla i després amplia el camp de visió omplint l'espai que sobra.
        //tal com el tinc fet (W=0 i posició de la càmera) el que fa és escalar la horitzontal a la mida del telèfon
i ampliar després cap a la dreta.
        encuadre.apply(false); //true centra la càmera (al 0,0, crec. però vaja, ja la centro jo bé a "resize")

        //botons
        for(int i=0;i<7;i++){
            bGraus.add(new BotQ(realW*ratio-75f, screenH*((8f-i)/9+0.01f),i)); //Compte, la coordenada x
s'actualitza al resize. Si vols canviar canvia allà.
        }

        b7a  = new BotQ(realW*ratio/2f-200, screenH*0.08f, 130, 80, "7a" ); //x a resize
        bDom = new BotQ(realW*ratio/2f-50,  screenH*0.08f, 130, 80, "DS" );
        bInv = new BotQ(realW*ratio/2f+100, screenH*0.08f, 130, 80, "Inv");
        bAlt = new BotQ(realW*ratio/2f+250, screenH*0.08f, 130, 80, "Alt");

        bBorrar = new BotQ(realW*ratio-75f, 75f, 95f, 95f);

        //botons del menú
        bMenu = new BotQ(110f,110f,170f,170f); //aquest no es veu afectat pel resize, perquè està respecte el marge
esquerra.

        bTorna = new BotQ(bMenu.x,bMenu.y,bMenu.w,bMenu.h);
        bDesa = new BotQ(realW*ratio-bMenu.x,bMenu.y,170,170); //TODO afegir al resize els botons que no hi siguin.
        bIngs = new BotQ(realW*ratio/2f, bMenu.y+dBM, 400,80,"Confirmar");
        bAcaba = new BotQ(realW*ratio/2f,bMenu.y,400,80,"Confirmar");
        bCancel = new BotQ(realW*ratio/2f+180,bMenu.y,300,80,"Cancel·lar");

        bMesTo = new BotQ(realW*ratio/2f+190,screenH*0.7f,110,110);
        bMenysTo = new BotQ(realW*ratio/2f-190,screenH*0.7f,110,110);

        System.out.println("Inicialitzant esquema en "+H.notes[H.tonalitat.nom]+H.getAlt(H.tonalitat)+"
```

```
"+H.modes[H.mode]+"...");
        esbos.add(new Chord(0));
        /*//de moment per desenvolupar la segona part afegeixo un esquema ja des del principi.
        esbos.add(new Chord(3,true));
        esbos.add(new Chord(1));
        esbos.get(esbos.size-1).inv=2;
        esbos.add(new Chord(4, true));
        esbos.get(esbos.size-1).alt7=true;
        esbos.get(esbos.size-1).alt9=true;
        esbos.add(new Chord(5,true));
        esbos.add(new Chord(0,true));
        esbos.add(new Chord(3,true));
        esbos.add(new Chord(1));
        esbos.add(new Chord(0));
        esbos.get(esbos.size-1).inv=2;
        esbos.add(new Chord(4,true));
        esbos.get(esbos.size-1).alt7=true;
        esbos.add(new Chord(0));
        aAct = (byte) (esbos.size-1);*/
        corregeix();

        //proves del correcte funcionament de les funcions H
        System.out.println("\nComprovant la implementació de les funcions musicals...\n");
        System.out.println("Un " + H.notes[1] + H.getAlt(1,3) + " és un " + H.notes[2] + H.getAlt(2, 3));
        System.out.println("Un " + H.notes[6] + H.getAlt(6,11) + " és un " + H.notes[0] + H.getAlt(0, 11));
        System.out.println("Un " + H.notes[3] + H.getAlt(3,7) + " és un " + H.notes[4] + H.getAlt(4, 7));
        System.out.println("\nUna 4J (3,5) sobre Do (0,0) és Fa (" + H.addInterval(H.tonalitat, 3, 5).nom +","+
H.addInterval(H.tonalitat, 3, 5).so+ ") = (3,5)");
        System.out.println("\nEl " + H.roman[4] + " grau de " + H.notes[H.tonalitat.nom] + H.getAlt(H.tonalitat) +
" " + H.modes[H.mode] + " és " + H.notes[H.getNote(4).nom] + H.getAlt(H.getNote(4)));
        System.out.println("El " + H.roman[2] + " grau de " + H.notes[0] + H.getAlt(0, 0) + " " + H.modes[1] + "
és " + H.notes[H.getNote(2,0,0,1).nom] + H.getAlt(H.getNote(2,0,0,1)));
        System.out.println("El " + H.roman[3] + " grau de " + H.notes[6] + H.getAlt(6, 10) + " " + H.modes[0] + "
és " + H.notes[H.getNote(3,6,10,0).nom] + H.getAlt(H.getNote(3,6,10,0)));
        System.out.println("\nL'acord del " + H.roman[esbos.get(0).grau] + " grau de " + H.notes[H.tonalitat.nom]
+ H.getAlt(H.tonalitat) + " " + H.rModes[H.mode] + " és " + H.notes[F.keep(H.getChordNote(esbos.get(0),1)).nom] +
H.getAlt(F.lastNote) + "-" + H.notes[F.keep(H.getChordNote(esbos.get(0),3)).nom] + H.getAlt(F.lastNote) + "-" +
H.notes[F.keep(H.getChordNote(esbos.get(0),5)).nom] + H.getAlt(F.lastNote));
        System.out.println("\nL'ajust de les notes en la posició més greu serà:\n(Do...Si)" + H.nPos(0) + ",
"+H.nPos(1) + ", "+H.nPos(2) + ", "+H.nPos(3) + ", "+H.nPos(4) + ", "+H.nPos(5) + ", "+H.nPos(6));
        System.out.println("\nEntre Do (0,0) i Fa (3,5) hi ha ("+H.getInterval(0,0,3,5).nom+","+H.getInterval(0,
0,3,5).so+")");
        System.out.println("Entre Do4 (0,0,4) i Sol3 (4,7,3) hi ha
("+H.getInterval(0,0,4,4,7,3).nom+","+H.getInterval(0,0,4,4,7,3).so+","+H.getInterval(0,0,4,4,7,3).ia+")");
        System.out.println("Entre Sol3 (4,7,3) i Do4 (0,0,4) hi ha
("+H.getInterval(4,7,3,0,0,4).nom+","+H.getInterval(4,7,3,0,0,4).so+","+H.getInterval(4,7,3,0,0,4).ia+")");
        System.out.println("Entre Do3 (0,0,3) i Re4 (1,2,4) hi ha
("+H.getInterval(0,0,3,1,2,4).nom+","+H.getInterval(0,0,3,1,2,4).so+","+H.getInterval(0,0,3,1,2,4).ia+")");
        System.out.println("Entre Sib2 (6,10,2) i Do4 (0,0,4) hi ha
("+H.getInterval(6,10,2,0,0,4).nom+","+H.getInterval(6,10,2,0,0,4).so+","+H.getInterval(6,10,2,0,0,4).ia+")");
        System.out.println("\nA "+H.notes[H.tonalitat.nom]+H.getAlt(H.tonalitat)+" "+H.modes[H.mode]+" la nota
"+H.notes[2]+H.getAlt(2,3)+" porta accidental? "+H.accidental(2,3));
        System.out.println("A "+H.notes[1]+H.getAlt(1,2)+" "+H.rModes[0]+" la nota "+H.notes[0]+H.getAlt(0,1)+"
porta accidental? "+H.accidental(0, 1, 1, 2, 0));
        System.out.println("A "+H.notes[0]+H.getAlt(0,0)+" "+H.rModes[1]+" la nota "+H.notes[1]+H.getAlt(1,1)+"
porta accidental? "+H.accidental(1, 1, 0, 0, 1)+"\n");
        System.out.println("La tonalitat de "+H.notes[H.tonalitat.nom]+H.getAlt(H.tonalitat)+" "+H.modes[H.mode]+"
porta "+H.textArm(H.armadura(H.tonalitat,H.mode)));
        System.out.println("La tonalitat de "+H.notes[3]+H.getAlt(3,5)+" "+H.modes[0]+" porta
"+H.textArm(H.armadura(3,5,0)));
        System.out.println("La tonalitat de "+H.notes[2]+H.getAlt(2,4)+" "+H.modes[0]+" porta
"+H.textArm(H.armadura(2,4,0)));
        System.out.println("La tonalitat de "+H.notes[1]+H.getAlt(1,3)+" "+H.modes[0]+" porta
"+H.textArm(H.armadura(1,3,0)));
        System.out.println("La tonalitat de "+H.notes[3]+H.getAlt(3,3)+" "+H.modes[0]+" porta
"+H.textArm(H.armadura(3,3,0))); //això és de penjats, però mola veure que funcionaria igual. xD
        System.out.println("La tonalitat de "+H.notes[0]+H.getAlt(0,2)+" "+H.modes[0]+" porta
"+H.textArm(H.armadura(0,2,0)));
    }


    /*********** RENDER ***********/
    @Override
    public void render(float delta) { //delta és variació de temps entre fotogrames (1/fps).
        delta=Math.min(delta, 0.25f); //Asseguro que no hagi passat massa temps (que no s'hagi empanat el trasto xD)
per evitar salts bruscos en les animacions i altres comportaments estranys.

        Gdx.gl.glClearColor(S.fons.r,S.fons.g,S.fons.b,S.fons.a); //color de fons
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT); //coses seves
```

```
        camera.update(); //actualitza la càmera
        lote.setProjectionMatrix(camera.combined);

        /**- Execució de la correcció -*/
        if(calCorregir){
            corregeix();
            calCorregir = false;
            System.out.println("Delta: "+delta);
        }

        lote.begin();
        if(!mostraMenu){ //engloba tot el que no voldré renderitzar quan estigui mostrant el menú

            /** - - Pentagrames i armadura - - */
            lote.setColor(S.pentagrama);
            for(int i=0;i<5;i++){ //dibuixo el pentagrama TODO canviar per una línia original més prima, que ho
estàs fent amb un quadrat i el tornes boig. (de fet, al mòbil crec que es veu bé)
                lote.draw(Recursos.rect, 25, yEsb+dEspai*i, rLim-50, 3f);
            }
            lote.draw(Recursos.sol, 45f+(scroll-lLim), yEsb-56f, dEspai*4f, dEspai*8f); //dibuixo la clau de sol
            if(H.armadura>0){ //dibuixo l'armadura. TODO ajustar per gestionar alteracions dobles.
                ii=0;
                while(ii<H.armadura){
                    lote.draw(Recursos.sharp, 45f+(scroll-lLim)+dEspai*3.5f+ii*dArm, yEsb+H.armPos((3+4*ii)%7,true)-
dNota*0.38f, dNota*0.5f*1.7f, dNota*1.7f);
                    ii++;
                }
            }else if(H.armadura<0){
                ii=0;
                while(ii>H.armadura){
                    lote.draw(Recursos.bemoll, 45f+(scroll-lLim)+dEspai*3.5f+(-ii)*dArm, yEsb+H.armPos((6+3*(-
ii))%7,false)-dNota*0.05f, dNota*0.5f*1.7f, dNota*1.7f);
                    ii--;
                }
            }

            /**- - - - - DIBUIX DELS ACORDS - - - -*/
            ii=0;
            for(Chord ac: esbos){ //per cada acord a la seqüència...
                H.resetStack=true; //pinto les notes de l'acord (reset li diu que pot començar per baix amb la
octava quan calcula la posició de les notes (en oposició a apilonar com dins l'acord)).
                if(ac.dom){ //calculo primer quantes notes hauré de pintar, de cara a definir el "for"
                    nAc = 5;
                }else if(ac.sept){
                    nAc = 4;
                }else{
                    nAc = 3;
                }
                if(ii==aAct){ //pinto diferent l'acord actual
                    lote.setColor(S.actual);
                }else{
                    lote.setColor(S.pentagrama);
                }
                for(byte j=0;j<nAc;j++){ //per cada nota que té l'acord...
                    if(j==ac.inv){ //omplo la nota que anirà al baix (per les inversions) TODO en algun moment
passar a dibuixar l'acord invertit (al menys quan és tríade o quan és 6/4)
                        lote.setColor(lote.getColor().r,lote.getColor().g,lote.getColor().b,0.2f); //color atenuat
(transparència) (r,g,b,alpha)
                        lote.draw(Recursos.cir, scroll+dEAc*ii-dNota/2,
yEsb+H.nStackPos(F.keep(H.getChordNote(ac,1+2*j))), dNota, (dNota+dEspai)/2f);
                        lote.setColor(lote.getColor().r,lote.getColor().g,lote.getColor().b,1f); //el torno a posar
opac
                        lote.draw(Recursos.nota, scroll+dEAc*ii-dNota/2, yEsb+H.stackH, dNota, dNota); //i la pinto
                    }else{
                        lote.draw(Recursos.nota, scroll+dEAc*ii-dNota/2,
yEsb+H.nStackPos(F.keep(H.getChordNote(ac,1+2*j))), dNota, dNota); //pinto la nota sense omplir-la primer.
                    }

                    if(H.stackH<(-dEspai)){ //pinto les línies addicionals (inferior) -- La variable stackH la guarda
nStackPos, i és la última posició calculada per a una nota.
                        lote.draw(Recursos.rect, scroll+dEAc*ii-dNota*0.75f,yEsb-dEspai,dNota*1.5f,3f);
                    }else if(H.stackH>dEspai*5f){ //segona superior (do)
                        lote.draw(Recursos.rect, scroll+dEAc*ii-dNota*0.75f, yEsb+dEspai*6f, dNota*1.5f, 3f);
                    }else if(H.stackH>dEspai*4f){ //primera superior (la)
                        lote.draw(Recursos.rect, scroll+dEAc*ii-dNota*0.75f, yEsb+dEspai*5f, dNota*1.5f, 3f);
                    }

                    /**- - - Alteracions accidentals - - -*/
                    if(H.accidental(F.lastNote)){ //pinto l'alteració de la nota en cas que calgui l'accidental
                        switch(H.getAltNum(F.lastNote)){
```

```
                    case 1:
                        lote.draw(Recursos.sharp, scroll+dEAc*ii-dAlt, yEsb+H.stackH-dNota*0.38f, dNota*0.5f*1.7f,
dNota*1.7f);
                        break;
                    case 0:
                        lote.draw(Recursos.natural, scroll+dEAc*ii-dAlt, yEsb+H.stackH-dNota*0.38f,
dNota*0.5f*1.7f, dNota*1.7f);
                        break;
                    case -1:
                        lote.draw(Recursos.bemoll, scroll+dEAc*ii-dAlt-15f*(j%2)+5f, yEsb+H.stackH-dNota*0.05f,
dNota*0.5f*1.5f, dNota*1.5f); //alterno amb el %2 les alteracions que es molesten entre elles
                        break;
                    default:
                        System.out.println("Accidental sense dibuix!!!
"+H.notes[F.lastNote.nom]+H.getAlt(F.lastNote));
                    }
                }
            }

            /** - - - - TEXT GRAUS XIFRAT I ENLLAÇOS - - - -*/
            if(ii<cGraus.size){ //comprovo que hi hagi totes les dades de correcció que pugui voler anar a
buscar (perquè si renderitza a mitja correcció s'ho pot menjar amb patates).
                if(ii==aAct && cGraus.get(ii)=='n' && !insert){ //marco també el grau de l'acord actual (si no hi
ha errors) (desactivo si estic insertant, perquè vol dir que els botons no el canviaran)
                    Recursos.font.setColor(S.actual);
                }else{
                    Recursos.font.setColor(S.cColor(cGraus.get(ii))); //demano el color corresponent a la
correcció.
                }
            }
            Recursos.font.draw(lote,F.keep(H.roman[ac.grau]),scroll+dEAc*ii+F.gAlign(ac),yEsb-100f); //escric el
grau
            if(ac.dom && ac.grau!=4 || ac.hom || ac.nap){ //tatxo els alterats
                lote.setColor(lote.getColor().r,lote.getColor().g,lote.getColor().b,0.5f);
                lote.draw(Recursos.rect, scroll+dEAc*ii-25, yEsb-118f-2, 25, 2, 50, 4, 1, 1, 10); //TODO
arreglar el disseny. Potser canviar per un gràfic d'una tatxada fet amb la tauleta, ja directament inclinat, fins i
tot. Ajustar amplada a amplada de les lletres.
            }
            Recursos.xifrat.draw(lote,F.keep(H.textGrau(ac)), scroll+dEAc*ii+F.gAlign(ac)-
F.align('r',H.roman[ac.grau]),yEsb-100f+F.vAlign("xifrat",'d')-F.vAlign('d',H.roman[ac.grau]));

            /** - - Xifrat - - */
            if(ii<cInv.size){ //comprovo que hi ha totes les dades de correcció que pugui voler anar a buscar
                if(ii==aAct && cGraus.get(ii)=='n'){ //i pinto també el xifrat de l'actual quan no té errors.
                    Recursos.xifrat.setColor(S.actual);
                }else{
                    Recursos.xifrat.setColor(S.cColor(cInv.get(ii)));
                }
            }
            Recursos.xifrat.draw(lote,F.keep(H.xifrat(ac, false)),scroll+dEAc*ii-F.gAlign(ac)+10f,yEsb-
100f+F.vAlign("xifrat")); //part del superíndex
            Recursos.xifrat.draw(lote,F.keep(H.xifrat(ac, true)),scroll+dEAc*ii-F.gAlign(ac)+10f,yEsb-
100f+F.vAlign("xifrat")-F.vAlign('d', H.roman[ac.grau])); //part del subíndex
            if(ac.sept){
                if(F.keep(H.ascendent(H.getInterval(H.getChordNote(ac,1),H.getChordNote(ac,7)))).nom==6 &&
F.lastNote.so==11){ //si la sèptima és major...
                    Recursos.xifrat.setColor(S.graus);
                    Recursos.xifrat.draw(lote,F.keep("M"),scroll+dEAc*ii+F.align("xifrat"),yEsb+H.stackH+120f);
                }
            }

            /** - - Enllaços - - */
            if(ii>0){ //a partir del segon acord pinto l'enllaç
                if(ii-1<cLinks.size){ //asseguro que no vagi a buscar informació que no hi és
                    Recursos.xifrat.setColor(S.cColor(cLinks.get(ii-1))); //demano el color de la correcció
                    Recursos.xifrat.draw(lote,F.keep(H.link(links.get(ii-1))),scroll+dEAc*(ii-
0.5f)+F.align("xifrat", 'c'),yEsb-180f); //escric enllaç
                }
            }
            ii++;
        }

        /** - - - Cursors - - -*/
        if(insert){
            lote.setColor(S.cursor.r,S.cursor.g,S.cursor.b,A.aCursor);
            lote.draw(Recursos.rect, scroll+dEAc*(aAct+0.5f), yEsb-dEspai, 4f, dEspai*6f);
        }

        /** - - - Cadencies - - - */
        lote.setColor(S.cadencies);
```

```
            Recursos.font.setColor(S.cadencies);
            for(Cadencia cad: cadencies){
                lote.draw(Recursos.rect, scroll+dEAc*(cad.n)-40f, yEsb-280f, dEAc*(cad.w/2f), 6f);
                lote.draw(Recursos.rect, scroll+dEAc*(cad.n+cad.w/2f)+40f, yEsb-280f, dEAc*(cad.w/2f), 6f);
                lote.draw(Recursos.rect, scroll+dEAc*(cad.n)-40f, yEsb-280f, 6f, 18f);
                lote.draw(Recursos.rect, scroll+dEAc*(cad.n+cad.w)+40f, yEsb-280f, -6f, 18f);
                if(cad.cp){
                    Recursos.font.draw(lote,F.keep("CP"),scroll+dEAc*(cad.n+cad.w/2f)+F.align(),yEsb-
280f+F.vAlign());
                }else{
                    Recursos.font.draw(lote,F.keep("CT"),scroll+dEAc*(cad.n+cad.w/2f)+F.align(),yEsb-
280f+F.vAlign());
                }
            }

            lote.setColor(S.fons); //Pinto la cortina que tapa l'esbos quan faig scroll
            lote.draw(Recursos.rect, 0, 0, 25, screenH); //barra esquerra
            lote.draw(Recursos.rect, rLim-25, 0, realW*ratio-rLim+25, screenH); //barra dreta


            /** - - - - BOTONS - - - - **/

            //pinto botons dels graus
            ii=0;
            for(BotQ b: bGraus){
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, b.x-b.w/2f, b.y-b.h/2f, b.w, b.h);
                if(b.grau == esbos.get(aAct).grau){ //marco el grau de l'acord actual
                    lote.setColor(S.b_premut);
                }else{
                    if(aAct==0 && !insert){
                        lote.setColor(S.b_inactiu); //desactivo els altres graus si sóc sobre el primer acord (sense
insertar darrere)
                    }else{
                        lote.setColor(S.b_normal);
                    }
                }
                lote.draw(Recursos.rect, b.x-b.w/2f+8f, b.y-b.h/2f+8f, b.w-16f, b.h-16f);
            }
            Recursos.font.setColor(S.b_text);
            ii=0;
            for(BotQ b: bGraus){ //Escric text dels botons dels graus
                Recursos.font.draw(lote,F.keep(H.roman[b.grau]), b.x+F.align(), b.y+F.vAlign());
                ii++;
            }

            lote.setColor(S.b_marc); //botó de posar sèptimes
            lote.draw(Recursos.rect, b7a.x-b7a.w/2f, b7a.y-b7a.h/2f, b7a.w, b7a.h);
            Recursos.font.setColor(S.b_text);
            if(!puc7a){ //si no puc posar sèptimes desactivat
                lote.setColor(S.b_inactiu);
                Recursos.font.setColor(S.b_text_inactiu);
            }else if(esbos.get(aAct).sept){ //si puc posar sèptima i en té
                lote.setColor(S.b_premut);
            }else{ //si puc i no en té
                lote.setColor(S.b_normal);
            }
            lote.draw(Recursos.rect, b7a.x-b7a.w/2f+8f, b7a.y-b7a.h/2f+8f, b7a.w-16f, b7a.h-16f);
            Recursos.font.draw(lote, F.keep(b7a.text), b7a.x+F.align(), b7a.y+F.vAlign());

            lote.setColor(S.b_marc); //botó de dominants secundàries
            lote.draw(Recursos.rect, bDom.x-bDom.w/2f, bDom.y-bDom.h/2f, bDom.w, bDom.h);
            Recursos.font.setColor(S.b_text);
            if(!pucDom){ //si no puc posar DS desactiva
                lote.setColor(S.b_inactiu);
                Recursos.font.setColor(S.b_text_inactiu);
            }else if(esbos.get(aAct).dom){ //si puc i en té
                lote.setColor(S.b_premut);
            }else{ //si puc i no en té
                lote.setColor(S.b_normal);
            }
            lote.draw(Recursos.rect, bDom.x-bDom.w/2f+8f, bDom.y-bDom.h/2f+8f, bDom.w-16f, bDom.h-16f);
            if(esbos.get(aAct).grau==4){ //canvio el text si és només la dominant normal amb 9a
                Recursos.font.draw(lote, F.keep("V9"), bDom.x+F.align(), bDom.y+F.vAlign());
            }else if(esbos.get(aAct).grau==1){ //i si és un segon, marco D.D.
                Recursos.font.draw(lote, F.keep("DD"), bDom.x+F.align(), bDom.y+F.vAlign());
            }else{
                Recursos.font.draw(lote, F.keep(bDom.text), bDom.x+F.align(), bDom.y+F.vAlign());
            }
```

```
            lote.setColor(S.b_marc); //botó de les inversions
            lote.draw(Recursos.rect, bInv.x-bInv.w/2f, bInv.y-bInv.h/2f, bInv.w, bInv.h);
            Recursos.font.setColor(S.b_text);
            if(!pucInv){
                lote.setColor(S.b_inactiu);
                Recursos.font.setColor(S.b_text_inactiu);
            }else{
                lote.setColor(S.b_normal);
            }
            lote.draw(Recursos.rect, bInv.x-bInv.w/2f+8f, bInv.y-bInv.h/2f+8f, bInv.w-16f, bInv.h-16f);
            Recursos.font.draw(lote, F.keep(bInv.text), bInv.x+F.align(), bInv.y+F.vAlign());

            lote.setColor(S.b_marc); //botó de les alteracions opcionals
            lote.draw(Recursos.rect, bAlt.x-bAlt.w/2f, bAlt.y-bAlt.h/2f, bAlt.w, bAlt.h);
            Recursos.font.setColor(S.b_text);
            if(!pucAlt6 && !pucAlt7 && !pucAlt9){
                lote.setColor(S.b_inactiu);
                Recursos.font.setColor(S.b_text_inactiu);
            }else{
                if(esbos.get(aAct).alt6
                || esbos.get(aAct).alt7
                || esbos.get(aAct).alt9){
                    lote.setColor(S.b_premut);
                }else{
                    lote.setColor(S.b_normal);
                }
            }
            lote.draw(Recursos.rect, bAlt.x-bAlt.w/2f+8f, bAlt.y-bAlt.h/2f+8f, bAlt.w-16f, bAlt.h-16f);
            Recursos.font.draw(lote, F.keep(bAlt.text), bAlt.x+F.align(), bAlt.y+F.vAlign());

            if(aAct>0){
                lote.setColor(S.error); //botó d'esborrar
            }else{
                lote.setColor(S.b_inactiu);
            }
            lote.draw(Recursos.arrow, bBorrar.x-bBorrar.w/2f, bBorrar.y-bBorrar.h/2f, bBorrar.h, bBorrar.w);

        }else{ //si mostraMenu

            //botons de desar i de tornar
            if(!H.tonalitat.equals(H.tonalitatPrevia) || H.mode!=H.modePrevi){ //activo el botó de desar només si ha
canviat alguna cosa
                if(!Pant2.nouEsquema){ //si estic remenant el to d'una realització ja començada aviso que ballaran
les tessitures
                    Recursos.xifrat.setColor(S.alerta);
                    Recursos.xifrat.draw(lote,"S'actualitzaran les notes de la realització per respondre al nou
to.\nEs mantindrà l'índex acústic de les notes.",15f,screenH*0.5f,realW*ratio-30f, Align.center, true);
                }
                lote.setColor(S.correcte);
            }else{
                lote.setColor(S.b_inactiu);
            }
            lote.draw(Recursos.cir, bDesa.x-bDesa.w/2f, bDesa.y-bDesa.h/2f, bDesa.w, bDesa.h);
            lote.setColor(S.error);
            lote.draw(Recursos.cir, bTorna.x-bTorna.w/2f, bTorna.y-bTorna.h/2f, bTorna.w, bTorna.h);
            lote.setColor(S.fons);
            lote.draw(Recursos.cir, bDesa.x-bDesa.w*A.rMenu/2f, bDesa.y-bDesa.h*A.rMenu/2f, bDesa.w*A.rMenu,
bDesa.h*A.rMenu);
            lote.draw(Recursos.cir, bTorna.x-bTorna.w*A.rMenu/2f, bTorna.y-bTorna.h*A.rMenu/2f, bTorna.w*A.rMenu,
bTorna.h*A.rMenu);
            Recursos.font.setColor(0f,0f,0f,1f);
            Recursos.font.draw(lote, F.keep("Desa"), bDesa.x+F.align(), bDesa.y+F.vAlign());
            Recursos.bMenu.setColor(0f,0f,0f,1f);
            Recursos.bMenu.draw(lote, F.keep("Torna"), bTorna.x+F.align("bMenu"), bTorna.y+F.vAlign("bMenu"));

            //botons de canviar el to
            lote.setColor(S.b_marc);
            lote.draw(Recursos.arrow, bMesTo.x+bMesTo.w/2f, bMesTo.y-bMesTo.h/2f, -bMesTo.w, bMesTo.h);
            lote.draw(Recursos.arrow, bMenysTo.x-bMenysTo.w/2f, bMenysTo.y-bMenysTo.h/2f, bMenysTo.w, bMenysTo.h);

            //botó d'editar ingredients
            lote.setColor(S.b_marc);
            lote.draw(Recursos.rect, bIngs.x-bIngs.w/2f, bIngs.y-bIngs.h/2f, bIngs.w, bIngs.h);
            lote.setColor(S.b_normal);
            lote.draw(Recursos.rect, bIngs.x-bIngs.w/2f+8f, bIngs.y-bIngs.h/2f+8f, bIngs.w-16f, bIngs.h-16f);
            Recursos.font.setColor(S.graus);
            if(vullIngs){
                Recursos.bMenu.setColor(S.alerta);
                Recursos.bMenu.draw(lote,"Tornar a la pantalla de l'enunciat?", 15f, yAvisos, realW*ratio-30f,
Align.center,true);
```

```
            Recursos.font.draw(lote, F.keep(bIngs.text), bIngs.x+F.align(),bIngs.y+F.vAlign());
        }else{
            Recursos.font.draw(lote, F.keep("Editar Ings."), bIngs.x+F.align(),bIngs.y+F.vAlign());
        }

        //botó de passar a SATB
        lote.setColor(S.b_marc);
        lote.draw(Recursos.rect, bAcaba.x-bAcaba.w/2f, bAcaba.y-bAcaba.h/2f, bAcaba.w, bAcaba.h);
        lote.setColor(S.b_normal);
        lote.draw(Recursos.rect, bAcaba.x-bAcaba.w/2f+8f, bAcaba.y-bAcaba.h/2f+8f, bAcaba.w-16f, bAcaba.h-16f);
        Recursos.font.setColor(S.graus);
        if(vullAcabar){ //s'ha polsat "Passar a SATB" i estic demanant confirmació
            if(tincCP){
                if(cLinks.contains('e', true) || cGraus.contains('e', true) || cInv.contains('e', true)){
                    Recursos.bMenu.setColor(S.error);
                    Recursos.bMenu.draw(lote, "Hi ha "+F.keep(neLinks+neGraus+neInv)+"
"+F.plurals("error",F.lastInt)+" a l'esbós. Segur que vols continuar?", 15f, yAvisos, realW*ratio-30f,
Align.center, true);
                }else{
                    Recursos.bMenu.setColor(S.correcte);
                    Recursos.bMenu.draw(lote,"L'esquema acaba en CP. Es procedirà amb la
realització.",15f,yAvisos,realW*ratio-30f, Align.center, true);
                }
            }else{
                Recursos.bMenu.setColor(S.error);
                Recursos.bMenu.draw(lote,"No s'ha detectat CP al final. Segur que vols continuar?", 15f, yAvisos,
realW*ratio-30f, Align.center,true);
            }
            Recursos.font.draw(lote, F.keep(bAcaba.text), bAcaba.x+F.align(),bAcaba.y+F.vAlign()); //bAcaba passa
a ser per confirmar.
        }else{
            if(Pant2.nouEsquema){
                Recursos.font.draw(lote, F.keep("Passar a SATB"), bAcaba.x+F.align(),bAcaba.y+F.vAlign());
            }else{

                Recursos.font.draw(lote, F.keep("Tornar a SATB"), bAcaba.x+F.align(),bAcaba.y+F.vAlign());
            }
        }

        //botó de cancelar
        if(vullAcabar || vullIngs){
            lote.setColor(S.b_marc);
            lote.draw(Recursos.rect, bCancel.x-bCancel.w/2f, bCancel.y-bCancel.h/2f, bCancel.w, bCancel.h);
            lote.setColor(S.b_normal);
            lote.draw(Recursos.rect, bCancel.x-bCancel.w/2f+8f, bCancel.y-bCancel.h/2f+8f, bCancel.w-16f,
bCancel.h-16f);
            Recursos.font.draw(lote, F.keep(bCancel.text), bCancel.x+F.align(),bCancel.y+F.vAlign());
        }
    }

        if(A.tMenu>A.tHold){ //pantalla de les referències

            lote.setColor(S.newAlpha(S.fons,0.97f)); //requadre que tapa mig transparent
            lote.draw(Recursos.rect, 0, 0, realW*ratio, screenH);

            Recursos.bNotes.setColor(S.graus);
            Recursos.bNotes.draw(lote,"Ingredients de l'enunciat:",50,screenH*0.9f);

            if(PIng.bTipusAcord.size+PIng.bTipusCadencia.size+PIng.bTipusRes.size>0){
                Recursos.font.setColor(S.graus);
                Recursos.iQuant.setColor(S.graus);
                Recursos.iXifrat.setColor(S.graus);

                for(BotIA b: PIng.bTipusAcord){
                    lote.setColor(S.ingredients);

                    lote.draw(Recursos.rect, b.x-b.h*0.2f, b.y-b.h*0.2f, b.w+b.h*0.4f, b.h);
                    //quantitat (p. ex. 2x)
                    if(b.iAc.n>1 && !b.iAc.tcp){
                        Recursos.iQuant.draw(lote, b.iAc.n+"x", b.x, b.y+F.vAlign("iQuant", 'd'));
                    }
                    //grau
                    Recursos.font.draw(lote, F.keep(H.roman[b.iAc.acord.grau]), b.x+b.d[0], b.y+F.vAlign("font",
'd'));
                    //xifrat
                    Recursos.iXifrat.draw(lote, F.keep(F.ifText(b.iAc.triada && b.iAc.acord.inv<1, "5",
H.xifrat(b.iAc.acord, false))), b.x+b.d[1],
b.y+F.vAlign("font",'d',H.roman[b.iAc.acord.grau])+F.vAlign("iXifrat")*0.7f);
                    Recursos.iXifrat.draw(lote, F.keep(H.xifrat(b.iAc.acord, true)), b.x+b.d[1],
b.y+F.vAlign("iXifrat")*1.3f);
```

```
                    //carro
                    Recursos.iXifrat.draw(lote, F.keep(b.carro), b.x+b.d[2],
b.y+F.vAlign("font",'c',H.roman[b.iAc.acord.grau])+F.vAlign("iXifrat",'c'));
                    Recursos.iQuant.draw(lote, F.keep(b.carro2), b.x+b.d[3], b.y+F.vAlign("iQuant",'d'));
                }

                for(BotIC b: PIng.bTipusCadencia){
                    lote.setColor(S.ingredients);

                    lote.draw(Recursos.rect, b.x-b.h*0.2f, b.y-b.h*0.2f, b.w+b.h*0.4f, b.h);
                    //quantitat (p. ex. 2x)
                    if(b.iCad.n>1 && !b.iCad.tcp){
                        Recursos.iQuant.draw(lote, b.iCad.n+"x", b.x, b.y+F.vAlign("iQuant", 'd'));
                    }
                    //text "CT"
                    Recursos.font.draw(lote, F.keep("CT"), b.x+b.d[0], b.y+F.vAlign("font",'d'));
                    //"xifrat"
                    Recursos.iXifrat.draw(lote, F.keep(F.ifText(b.iCad.sept, "7",F.ifText(b.iCad.triada,
"5"))),b.x+b.d[1],b.y+F.vAlign("font",'d',"CT")+F.vAlign("iXifrat")*0.7f);
                    //diferent i tcp
                    Recursos.iQuant.draw(lote, F.keep(F.ifText(b.iCad.dif, " d")+F.ifText(b.iCad.tcp, " !")),
b.x+b.d[2], b.y+F.vAlign("iQuant",'d'));
                }

                for(BotIR b: PIng.bTipusRes){
                    lote.setColor(S.ingredients);

                    lote.draw(Recursos.rect, b.x-b.h*0.2f, b.y-b.h*0.2f, b.w+b.h*0.4f, b.h);
                    //text "RI"
                    Recursos.font.draw(lote, F.keep("RI"), b.x, b.y+F.vAlign("font",'d'));
                    //"xifrat"
                    Recursos.iXifrat.draw(lote, F.keep(F.ifText(b.iRI.sens,
"S","7")),b.x+b.d*1.1f,b.y+F.vAlign("iXifrat")*1.4f);
                }
            }else{ //si no hi ha ingredients
                Recursos.iQuant.setColor(S.graus);
                Recursos.iQuant.draw(lote,"No s'ha definit cap ingredient (enunciat lliure).",50,screenH*0.83f);
            }

            Recursos.bNotes.draw(lote,"Mapa dels tons veïns:",50,screenH*0.6f);
            Recursos.iQuant.setColor(S.graus);
            Recursos.iQuant.draw(lote,"Referència no disponible actualment.",50,screenH*0.53f);

        } //fi pantalla de les referències

        lote.setColor(S.fons); //cercle del to (botó del menú)
        lote.draw(Recursos.cir, bMenu.x-bMenu.w*1.2f/2f, bMenu.y-bMenu.h*1.2f/2f, bMenu.w*1.2f, bMenu.h*1.2f);
        lote.setColor(S.tonalitat);
        lote.draw(Recursos.cir, bMenu.x-bMenu.w/2f, bMenu.y-bMenu.h/2f, bMenu.w, bMenu.h);
        lote.setColor(S.fons);
        lote.draw(Recursos.cir, bMenu.x-bMenu.w*A.rMenu/2f, bMenu.y-bMenu.h*A.rMenu/2f, bMenu.w*A.rMenu,
bMenu.h*A.rMenu);
        if(H.getAlt(H.tonalitat)==""){
            Recursos.font.setColor(0f,0f,0f,1f);
            Recursos.font.draw(lote, F.keep(H.modeNotes(H.tonalitat.nom,H.mode)+H.getAlt(H.tonalitat)+"
"+H.rModes[H.mode]), bMenu.x+F.align(), bMenu.y+F.vAlign());
        }else{ //si el nom de la tonalitat és massa llarg, faig més petita la lletra.
            Recursos.bNotes.setColor(0f,0f,0f,1f);
            Recursos.bNotes.draw(lote, F.keep(H.modeNotes(H.tonalitat.nom,H.mode)+H.getAlt(H.tonalitat)+"
"+H.rModes[H.mode]), bMenu.x+F.align("bNotes"), bMenu.y+F.vAlign("bNotes"));
        }

    lote.end();

    /** - Actualització de les animacions - */
    A.update(delta);

} //Render end


/******** CLICK *********/
@Override
public boolean touchDown(int screenX, int screenY, int pointer, int button) {
    clickBot = false;
    click.x = screenX;
    click.y = screenY;
    encuadre.unproject(click); //IMPORTANT!!!! Si desfas des de la càmera hi haurà un embolic gros. Sempre des
del viewport.

    if(!mostraMenu){
```

```
        if(click.x>bGraus.get(0).x-bGraus.get(0).w/2f){ //si pico al marge dret
            for(BotQ b: bGraus){
                if(click.y>b.y-b.h/2f){ //he clicat un botó de grau
                    if(insert){ //inserint acords
                        System.out.println("Afegeixo: " + H.roman[b.grau] +" a "+(aAct+1));
                        esbos.insert(aAct+1, new Chord(b.grau)); //afegeixo després de l'actual
                        if(!Pant2.nouEsquema){ //nouEsquema fa que la realització es posi a zero. Si no ho tinc vol
dir que voldré conservar la part de la realització que no hagi tocat, així que l'edito tb.
                            Pant2.esquema.insert(aAct+1, new Desplegat());
                        }
                        aAct+=1; //ara l'actual és el nou que he posat
                        if(esbos.get(aAct).grau==4){
                            esbos.get(aAct).alt7=true; //si poso V, per quan sigui en menor, per defecte poso
sensible (el faig dominant). (Si és major no afecta, així que tant li fa posar-la a tot arreu).
                        }
                        System.out.println("Hi ha " + esbos.size + " acords a l'esbós.");
                        calCorregir = true;
                    }else{ //canviant l'acord actual
                        if(aAct!=0){
                            esbos.get(aAct).grau = (byte) b.grau;
                            if(!Pant2.nouEsquema){ //poso a zero la realització de l'acord canviat
                                Pant2.esquema.removeIndex(aAct);
                                Pant2.esquema.insert(aAct, new Desplegat());
                            }
                            calCorregir = true;
                        }else{
                            System.out.println("No pots canviar el primer");
                        }
                    }

                    A.tCursor=0; //poso animació del cursor a 0
                    fixScroll();
                    clickBot = true;
                    break;
                }
            }
            if(!clickBot && click.y<bBorrar.y+30 && click.y>bBorrar.y-70){
                if(aAct>0){
                    System.out.println("(esborrar)");
                    esbos.removeIndex(aAct); //esborro l'actual
                    if(!Pant2.nouEsquema){
                        Pant2.esquema.removeIndex(aAct);
                    }
                    aAct-=1; //ara l'actual és l'anterior
                    A.tCursor=0; //poso animació del cursor a 0
                    System.out.println("Hi ha " + esbos.size + " acords a l'esbós.");
                    fixScroll();
                    calCorregir = true;
                }else{
                    System.out.println("No pots esborrar el primer");
                }
                clickBot = true;
            }
        }else if(click.y<b7a.y+b7a.h/2f && click.x>250f){ //si pico al marge inferior (més a la dreta del botó
d'inici) TODO això de l'inici es pot treure quan tregui el missatge de no definit. De moment per debugging pot ser
útil
            if(click.x>b7a.x-b7a.w/2f && click.x<b7a.x+b7a.w/2f){ //botó de 7a
                if(puc7a){ //si és D.S. no deixo remenar la sèptima. Si és el primer de l'esquema tampoc. (canvis
dels permisos a la correcció)
                    if(esbos.get(aAct).sept){
                        System.out.println("Sense 7a");
                        if(esbos.get(aAct).inv>2){ //comprovo que treure la sèptima no deixi l'acord orfe d'inversió
                            esbos.get(aAct).inv=0;
                        }
                        if(!Pant2.nouEsquema){
                            for(int veu=0;veu<4;veu++){ //si mantinc la realització, trec les sèptimes que hi havia
posat
                                if(Pant2.esquema.get(aAct).satb[veu]==7){
                                    Pant2.esquema.get(aAct).satb[veu]=0;
                                }
                            }
                        }
                        esbos.get(aAct).sept=false;
                    }else{
                        System.out.println("Amb 7a");
                        esbos.get(aAct).sept=true;
                    }
                    fixScroll(); //ajusto l'scroll per si està editant un acord que no veu.
                    calCorregir = true;
                }else{
```

```
                System.out.println("No pots posar sèptima en aquest acord");
            }
            clickBot = true;
        }else if(click.x>bDom.x-bDom.w/2f && click.x<bDom.x+bDom.w/2f){ //si pico al botó de família de la
dominant...
            if(pucDom){ //la comprovació del permís per posar dominant és a la correcció.
                if(esbos.get(aAct).dom){
                    System.out.println("Desfer dominant secundària");
                    if(!Pant2.nouEsquema){ //si mantinc la realització, trec les 9es (si té sèptimes marco sept,
per si les sèptimes sí que les vol conservar)
                        for(int veu=0;veu<4;veu++){
                            if(Pant2.esquema.get(aAct).satb[veu]==9){
                                Pant2.esquema.get(aAct).satb[veu]=0;
                            }else if(Pant2.esquema.get(aAct).satb[veu]==7){
                                esbos.get(aAct).sept=true;
                            }
                        }
                    }
                    esbos.get(aAct).dom = false;
                }else{
                    System.out.println("Utilitzar dominant secundària");
                    esbos.get(aAct).dom = true;
                }
                fixScroll();
                calCorregir = true;
            }else{
                System.out.println("No pots fer servir la família de la dominant en aquest acord.");
            }
            clickBot = true;
        }else if(click.x>bInv.x-bInv.w/2f && click.x<bInv.x+bInv.w/2f){ //si pico al botó de les inversions
            if(pucInv){
                esbos.get(aAct).addInv(); //deixo que el propi acord gestioni el canvi d'inversió, que queda més
simple.
                fixScroll();
                calCorregir = true;
            }else{
                System.out.println("No pots invertir aquest acord");
            }
            clickBot = true;
        }else if(click.x>bAlt.x-bAlt.w/2f && click.x<bAlt.x+bAlt.w/2f){ //si pico al botó de les alteracions
            if(pucAlt6 || pucAlt7 || pucAlt9){
                if(pucAlt9){ //per dominants
                    if(esbos.get(aAct).alt9){
                        esbos.get(aAct).alt9=false;
                    }else{
                        esbos.get(aAct).alt9=true;
                    }
                }else if(pucAlt7){ //puc canviar la sensible
                    if(esbos.get(aAct).alt7){
                        esbos.get(aAct).alt7=false;
                    }else{
                        esbos.get(aAct).alt7=true;
                    }
                }else{ //puc canviar el sisè (no passa res perquè estigui en else, perquè tal com ho he muntat
cap acord pot variar les dues (podria el VII del menor quan és V en 7ds, però no el permeto precisament perquè és
V))
                    if(esbos.get(aAct).alt6){
                        esbos.get(aAct).alt6=false;
                    }else{
                        esbos.get(aAct).alt6=true;
                    }
                }
                fixScroll();
                calCorregir = true;
            }else{
                System.out.println("No hi ha alteracions opcionals disponibles.");
            }
            clickBot = true;
        }else{
            System.out.println("no definit dins marge inferior");
        }
    }else{ //ni al marge dret ni al marge inferior
        heArrossegat=false;
    }
    }else{ //si mostraMenu (botons del menú)
        if(click.x>bTorna.x-bTorna.w/2f && click.x<bTorna.x+bTorna.w/2f && click.y>bTorna.y-bTorna.h/2f &&
click.y<bTorna.y+bTorna.h/2f){ //botó de desfer
            if(H.tonalitatPrevia!=H.tonalitat || H.modePrevi!=H.mode){ //si he canviat alguna cosa de la
tonalitat, quan torno la desfaig
                H.tonalitat=H.tonalitatPrevia;
```

```
                H.mode=H.modePrevi;
            }

            mostraMenu = false; //trec el menú i recoloco el botó del to
            bMenu.x= (int) 110f;
            bMenu.y= (int) 110f;

            vullAcabar = false; //recoloco també el botó de demanar SATB i el poso a zero.
            bAcaba.x=(int)(realW*ratio/2f);
            bAcaba.w=400;

            clickBot = true;
        }else if(click.x>bDesa.x-bDesa.w/2f && click.x<bDesa.x+bDesa.w/2f && click.y>bDesa.y-bDesa.h/2f &&
click.y<bDesa.y+bDesa.h/2f){
            if(H.tonalitatPrevia!=H.tonalitat || H.modePrevi!=H.mode){ //només actua el botó si ha canviat
alguna cosa de la tonalitat
                H.tonalitatPrevia = H.tonalitat; //si deso els canvis, marco als valors previs que els he
canviat.
                H.modePrevi = H.mode;

                H.armadura = (byte) H.armadura(H.tonalitat,H.mode); //càlcul de la nova armadura
                lLim=(int) (230 + dArm*Math.abs(H.armadura)); //ajust del límit esquerra de l'scroll en funció de
la longitud de l'armadura
                scroll=lLim; //posada a zero de l'scroll

                calCorregir = true;
            }

            clickBot = true;
        }else if(click.x>bMesTo.x-bMesTo.w/2f && click.x<bMesTo.x+bMesTo.w/2f && click.y>bMesTo.y-bMesTo.h/2f
&& click.y<bMesTo.y+bMesTo.h/2f){
            H.tonalitat = H.addInterval(H.tonalitat, 4, 7);

            clickBot = true;
        }else if(click.x>bMenysTo.x-bMenysTo.w/2f && click.x<bMenysTo.x+bMenysTo.w/2f && click.y>bMenysTo.y-
bMenysTo.h/2f && click.y<bMenysTo.y+bMenysTo.h/2f){
            H.tonalitat = H.addInterval(H.tonalitat, -4, -7);

            clickBot = true;
        }else if(click.x>bAcaba.x-bAcaba.w/2f && click.x<bAcaba.x+bAcaba.w/2f && click.y>bAcaba.y-bAcaba.h/2f
&& click.y<bAcaba.y+bAcaba.h/2f){
            if(vullAcabar){
                System.out.println("Esbós finalitzat. Passant a realització SATB...");
                app.setScreen(app.pant2);
            }else{
                System.out.println("Es demana permís per passar a SATB. Comprovant CP i demanant confirmació.");
                posaElsBotonsPer("vullAcabar");
                vullIngs=false;
                vullAcabar=true;
            }
            clickBot=true;
        }else if(click.x>bIngs.x-bIngs.w/2f && click.x<bIngs.x+bIngs.w/2f && click.y>bIngs.y-bIngs.h/2f &&
click.y<bIngs.y+bIngs.h/2f){
            if(vullIngs){
                System.out.println("Tornant a la pantalla dels ingredients...");
                app.setScreen(app.pIng);
            }else{
                System.out.println("Es demana editar ingredients. Confirmar la tria.");
                posaElsBotonsPer("vullIngs");
                vullAcabar=false;
                vullIngs=true;
            }
            clickBot=true;
        }else if((vullAcabar || vullIngs ) && click.x>bCancel.x-bCancel.w/2f && click.x<bCancel.x+bCancel.w/2f
&& click.y>bCancel.y-bCancel.h/2f && click.y<bCancel.y+bCancel.h/2f){
            posaElsBotonsPer("menu");
            if(vullAcabar){
                System.out.println("Pas a SATB cancel·lat");
                vullAcabar=false;
            }else{ //vull ings
                System.out.println("Pas a Ingredients cancel·lat");
                vullIngs=false;
            }

            clickBot=true;
        }
    }

    if(!clickBot && click.x>bMenu.x-bMenu.w/2f && click.x<bMenu.x+bMenu.w/2f && click.y>bMenu.y-bMenu.h/2f &&
click.y<bMenu.y+bMenu.h/2f){ //si pico al botó rodó del menú
```

```
            if(mostraMenu){
                H.mode=(byte) ((H.mode+1)%2); //faig que el botó del to canvii de mode quan sóc dins el menú.
            }else{
                A.tMenu = 0;
            }
        }

        return true;
    };

    @Override
    public boolean touchUp(int screenX, int screenY, int pointer, int button) {
        click2.x = screenX;
        click2.y = screenY;
        encuadre.unproject(click2);

        //accions dins l'àrea d'scroll (selecció d'acords i espais entre acords).
        if(!mostraMenu){
            if(!heArrossegat){ //si ja s'ha gestionat l'scroll a "touchDragged" no clico res
                if(click.y > screenH*0.15 && click.y < screenH*0.95 && click.x > 25 && click.x < rLim //havia clicat
sobre l'àrea d'scroll
                && click2.y> screenH*0.15 && click2.y< screenH*0.95 && click2.x> 25 && click2.x< rLim){ //i he
aixecat el dit també sobre l'àrea
                    for(int i=0; i<esbos.size; i++){ //per cada acord dins l'esbós
                        if(click2.x < scroll+dEAc*(i+0.75f)){ //trobo l'acord clicat
                            aAct = (byte) i;
                            if(click2.x < scroll+dEAc*(i+0.25)){ //a més, he clicat sobre l'acord (i no a l'espai que
hi ha fins al següent)
                                insert=false;
                            }else{ //si he clicat a l'espai, sí que inserto
                                insert=true;
                                A.tCursor=0;
                            }
                            System.out.println("\nNou acord actual: "+aAct+". Insert="+insert);
                            permisosBotons(); //calculo els nous permisos per l'acord actual
                            break; //si ja l'he trobat surto del bucle
                        }
                    }
                }
            }
        } //fi if(!mostramenu)

        //part del click al botó del menú
        if(A.tMenu>=0f && A.tMenu<A.tHold){ //marge de temps dins el qual considero que no s'ha mantingut aguantat
el botó del menú
            System.out.println("\nMostra el menú");
            A.rMenu = 0.8f;
            mostraMenu = true;
            bMenu.x=(int) (realW*ratio/2f); //moc el botó a la seva posició dins el menú
            bMenu.y=(int) (screenH*0.7f);
        }
        A.tMenu = -1;

        return true;
    };


    @Override
    public boolean touchDragged(int x, int y, int pointer) { //es dispara quan el click es mou (després que
s'hagi fet un click i abans que es retiri el dit) i diu on ha anat
        click2.x=x;
        click2.y=y;
        encuadre.unproject(click2);

        if(!mostraMenu){
            if(click.y > screenH*0.15f && click.y < screenH*0.95f && click.x > 25 && click.x < rLim){ //si els
clicks inicial i final...
                if(click2.y > screenH*0.15f && click2.y < screenH*0.95 && click2.x > 25 && click2.x < rLim){ //són
dins l'àrea mòbil...
                    if(Math.abs(click2.x-click.x)>5 || Math.abs(click2.y-click.y)>5){ //restringeixo a les vegades
que realment s'ha intentat moure expressament.
                        heArrossegat=true; //marco pel touchUp que he arrossegat
                        scroll+=click2.x-click.x; //TODO donar inèrcia a l'scroll. Ja hi ha fetes les variables, però
s'ha d'implementar.
                        System.out.println("\nArrossegat " + (click2.x-click.x));
                        System.out.println("Nou valor proposat: "+scroll);
                        System.out.println("Hi ha "+esbos.size+" acords i en caben "+ (rLim-lLim)/dEAc);
                        System.out.println("Hi ha ocupat "+(dEAc*esbos.size)+" i es veu "+(rLim-lLim));
                        if(esbos.size*dEAc<(rLim-lLim)){
                            System.out.println("Moviment impedit. Tots els acords són visibles.");
```

```
                        scroll=lLim;
                    }else if(scroll>lLim){
                        System.out.println("Ajustat al límit esquerra: "+scroll);
                        scroll=lLim;
                    }else if(scroll<((rLim)-dEAc*esbos.size)){ //TODO afegir alguna mena d'histèresi petita per
evitar que es torni boig. Investigar el comportament i això..
                        scroll=(int) ((rLim)-dEAc*esbos.size);
                        System.out.println("Ajustat al límit dret: "+scroll);
                    }else{
                        System.out.println("Moviment permès");
                    }
                    click.x=click2.x; //actualitzo la última posició clicada per poder fer una arrossegada
contínua fiable.
                    click.y=click2.y;
                }else{
                    //aquí hi va el que vulgui fer si s'arrossega una mica però no prou dins la zona arrossegable
(probablement res, perquè ja ho recull touchUp)
                }
            }
        }
    } //fi if(!mostraMenu)

        return true;
    };

    public void fixScroll(){
        if(scroll+(aAct+1)*dEAc > rLim){ //si l'actual és massa a la dreta
            scroll = (int) (rLim-dEAc*(aAct+1)); //poso l'actual a la dreta de tot de la zona visible
        }else if(scroll+aAct*dEAc < lLim){ //si l'actual és massa a l'esquerra
            scroll = (int) (lLim-aAct*dEAc); //poso l'actual a l'esquerra de tot de la zona visible
        }else{ //l'actual està bé. Comprovo que no estic deixant espai en blanc innecessari
            if(scroll<lLim && scroll<((rLim)-dEAc*esbos.size)){ //si hi ha tant scroll que deixo espai en blanc
                scroll = (int) ((rLim)-dEAc*esbos.size);
            }
            if(scroll>lLim){
                scroll=lLim;
            }
        }
    }

    @Override
    public void resize(int width, int height) {
        encuadre.update(width, height);

        realW = Gdx.graphics.getWidth();
        realH = Gdx.graphics.getHeight();
        ratio = (float) screenH/realH;

        for(BotQ b: bGraus){
            b.x=(int) (realW*ratio-75f);
        }
        bBorrar.x = (int) (realW*ratio-75f);

        b7a.x = (int) (realW*ratio/2f-200);
        bDom.x = (int) (realW*ratio/2f-50);
        bInv.x = (int) (realW*ratio/2f+100);
        bAlt.x = (int) (realW*ratio/2f+250);

        bDesa.x = (int) (realW*ratio-bTorna.x);
        bCancel.x = (int) (realW*ratio/2f+180);
        posaElsBotonsPer();

        if(bMenu.x!=110f){
            bMenu.x = (int) (realW*ratio/2f);
        }

        bMesTo.x = (int) (realW*ratio/2f+180f);
        bMenysTo.x = (int) (realW*ratio/2f-180f);

        rLim = (int) (realW*ratio-135);

        camera.position.set(realW/2f*ratio,screenH/2f,0);

        posicionaIngs();
    };

    @Override
    public void show() {
        posaElsBotonsPer("menu");
        bMenu.x= (int) 110f;
```

```
        bMenu.y= (int) 110f;
        vullAcabar=false;
        vullIngs=false;
        mostraMenu=false;
        Gdx.input.setInputProcessor(this); //Important! Sense això no respon als clicks.

        aAct=Pant2.aAct; //agafo l'actual de l'altra pantalla, per ajudar a ubicar l'usuari
        corregeix(); //per si hi ha hagut canvis derivats de la realització

        H.tonalitatPrevia = H.tonalitat;
        H.modePrevi = H.mode;

        H.armadura = (byte) H.armadura(H.tonalitat,H.mode); //càlcul de l'armadura (dependrà de si he escollit
major o menor abans a PTon)
        lLim=(int) (230 + dArm*Math.abs(H.armadura)); //ajust del límit esquerra de l'scroll en funció de la
longitud de l'armadura
        scroll=lLim; //posada a zero de l'scroll

        posicionaIngs(); //poso els botons dels ingredients a la posició d'aquesta pantalla
    };

    @Override
    public void pause() {
    }
    @Override
    public void resume() {
    }
    @Override
    public void hide() {
    }
    @Override
    public void dispose() {
    }

    /**** POSICIONAMENT DELS ELEMENTS ****/

    /**- - Posició dels ingredients de l'enunciat - -*/

    public static void posicionaIngs(){ //els moc per posar-los en fila com escrits seguits
        fila = 0;
        int xPos0 = 60;
        int xPos = xPos0;
        if(PIng.bTipusAcord.size+PIng.bTipusCadencia.size+PIng.bTipusRes.size>0){ //si hi ha algun ingredient
            for(BotIA b: PIng.bTipusAcord){ //tipus acord
                if(xPos+b.w>Pant.realW*Pant.ratio-xPos0){ //salt de línia
                    xPos=xPos0;
                    fila++;
                }
                b.y = (int) (screenH*0.75f-80*fila); //assignació
                b.x = xPos;
                xPos += (b.w + 35); //increment
            }
            for(BotIC b: PIng.bTipusCadencia){ //tipus cadencia
                if(xPos+b.w>Pant.realW*Pant.ratio-xPos0){ //salt de línia
                    xPos=xPos0;
                    fila++;
                }
                b.y = (int) (screenH*0.75f-80*fila); //assignació
                b.x = xPos;
                xPos += (b.w + 35); //increment
            }
            for(BotIR b: PIng.bTipusRes){ //tipus RI
                if(xPos+b.w>Pant.realW*Pant.ratio-xPos0){ //salt de línia
                    xPos=xPos0;
                    fila++;
                }
                b.y = (int) (screenH*0.75f-80*fila); //assignació
                b.x = xPos;
                xPos += (b.w + 35); //increment
            }
        }
    }

    /**- - Posició dels botons del menú - -*/

    public void posaElsBotonsPer(){
        if(vullAcabar){
            posaElsBotonsPer("vullAcabar");
        }else if(vullIngs){
            posaElsBotonsPer("vullIngs");
```

```
        }else{
            posaElsBotonsPer("menu");
        }
    }
    public void posaElsBotonsPer(String config){
        if(config=="vullAcabar"){
            bAcaba.x= (int) (realW*ratio/2f-180);
            bAcaba.w=300;
            bCancel.y=bAcaba.y;

            bIngs.x=(int)(realW*ratio/2f);
            bIngs.w=400;
        }else if(config=="vullIngs"){
            bIngs.x= (int) (realW*ratio/2f-180);
            bIngs.w=300;
            bCancel.y=bIngs.y;

            bAcaba.x=(int)(realW*ratio/2f);
            bAcaba.w=400;
        }else{
            bIngs.x=(int)(realW*ratio/2f);
            bIngs.w=400;
            bAcaba.x=(int)(realW*ratio/2f);
            bAcaba.w=400;
        }
    }

    /************ FUNCIONS DE CORRECCIÓ *************/

    /**- - - Desactivació de botons - - */
    public void permisosBotons(){
        puc7a=true; //el desactivaré quan no es pugui, que en general es pot.
        pucDom=false; //el faré true quan es pot, que no és massa sovint, de moment.
        pucInv=true; //el desactivaré quan no es pugui, que en general es pot.

        pucAlt6=false; //permisos d'alteracions
        pucAlt7=false;
        pucAlt9=false;

        if(aAct==0){ //si estic al primer acord no puc fer res
            puc7a=false;
            pucDom=false;
            pucInv=false;
        }else{ //després del primer
            /*if(esbos.get(aAct).grau==4){
                pucDom=true; //si és un V sempre puc posar V9.
            }else if(esbos.get(aAct).grau==2){ //si estic en un tercer, comprovo si es pot fer servir per CT amb
III-
                if(esbos.get(aAct-1).grau==4 || esbos.get(aAct-1).grau==2 && esbos.get(aAct-1).dom){ //l'anterior és
V o III-
                    pucDom=true;
                }
            }*/ //De moment no permeto dominants amb 9a, perquè no he implementat la nota negra ni res

            //TODO passar a utilitzar H.opcionals()
            if(esbos.get(aAct).dom){ //acord de dominant (permisos de 9a, i desactivo la 7a)
                pucDom=true; //marco que puc desfer-ho si l'acord és de dominant
                puc7a=false; //(ja ve de normal, no es pot posar i treure)
                if(H.mode==1 || esbos.get(aAct).hom){ //acord del menor
                    if(esbos.get(aAct).grau==0
                    || esbos.get(aAct).grau==1
                    || esbos.get(aAct).grau==4){
                        pucAlt9=false;
                    }else{
                        pucAlt9=true;
                    }
                }else{
                    if(esbos.get(aAct).grau==2
                    || esbos.get(aAct).grau==5
                    || esbos.get(aAct).grau==6){
                        pucAlt9=false;
                    }else{
                        pucAlt9=true;
                    }
                }
            }else{ //no és dominant
                pucAlt9=false; //no tindré 9a que remenar
                if(H.mode==1 || esbos.get(aAct).hom){ //acords del menor (permís de 6a i sensible)
                    if(esbos.get(aAct).grau==1 //permisos de 6a
                    || esbos.get(aAct).grau==3
```

```
                || esbos.get(aAct).grau==5
                || esbos.get(aAct).grau==6 && esbos.get(aAct).sept){ //al VII s'altera la sèptima. Si no en té
trec la opció.
                    pucAlt6=true;
                }else{
                    pucAlt6=false;
                }
                if(esbos.get(aAct).grau==0 && esbos.get(aAct).sept //permisos de sensible (com abans, al I
s'altera la sèptima).
                || esbos.get(aAct).grau==2
                || esbos.get(aAct).grau==4){
                    pucAlt7=true;
                }else{
                    pucAlt7=false;
                }
            }
        }
        System.out.println("7DI: "+puc7a+"/"+pucDom+"/"+pucInv);
        System.out.println("679: "+pucAlt6+"/"+pucAlt7+"/"+pucAlt9);
    }

    /**- - - - Correcció de l'esbós - - -*/
    public void corregeix(){
        System.out.println("\nCorregint esbós...");

        System.out.println("Posant a zero el registre...");
        logE.clear();
        logA.clear();

        System.out.println("Renovant llista d'enllaços...");
        links.clear();
        int i=0;
        for(Chord ac: esbos){
            if(i<esbos.size-1){ //comprovo que no surto dels límits
                links.add(H.linkNum(ac.grau, esbos.get(i+1).grau)); //apunto l'enllaç entre l'actual i el següent a
la llista d'enllaços
            }
            i++;
        }
        System.out.println(links);

        /** - - - - - Correcció dels enllaços - - - - - -*/

        System.out.println("Calculant correcció d'enllaços...");
        cLinks.clear();
        ii=0;
        while(ii<links.size){
            //System.out.println(ii+": "+links.get(ii));
            switch(links.get(ii)){
            case 0:
                cLinks.add('o'); //si no hi ha enllaç: ocult.
                break;
            case 1:
                cLinks.add('n'); //si és fort: normal.
                break;
            case 2:
                System.out.println("Trobat enllaç dèbil després del "+ii);
                i=1;
                while(ii+i<links.size){ //asseguro que no surto de la llista
                    if(links.get(ii+i)==0){
                        i++;
                    }else{
                        if(H.linkNum(esbos.get(ii).grau,esbos.get(ii+i+1).grau)<2){ //si l'enllaç des del primer fins
al trobat és fort o són el mateix acord
                            System.out.println("- Resolt correctament.");
                            cLinks.add('c'); //dèbil ben resolt: correcte.
                        }else{
                            System.out.println("- Dèbil mal resolt.");
                            cLinks.add('e'); //dèbil mal resolt: error.
                            logE.add(new Log(ii,13));
                        }
                        for(int j=1;j<i;j++){ //com que en realitat ja he comprovat tots els del mig, els aplico
també a la llista.
                            cLinks.add('o'); //tants ocults com enllaços iguals després del dèbil
                        }
                        cLinks.add('i'); //un irrellevant al final on passa del segon al tercer del dèbil (no ens
importa perquè l'enllaç rellevant és el gros).
                        ii+=i; //salto endavant amb l'iterador per no repetir els ja comprovats
                        i=0;
```

```
                    break; //si trobat trenco el while
                }
            }
            if(i!=0){
                cLinks.add('a'); //si he sortit del while sense resoldre la correcció (sóc al final de l'esbós i encara no sé com ho resoldrà) marco alerta.
            }
            break;
        case 3: //TODO aquestes comprovacions me les podria estalviar si guardés l'últim enllaç no-zero. Mirar si val la pena i fer el que creguis oportú.
            System.out.println("Trobat enllaç molt fort després del "+ii);
            i=1;
            while(ii-i>=0){ //comprovo que no surti de la llista (per davant)
                if(links.get(ii-i)==0){
                    i++;
                }else{
                    if(links.get(ii-i)==3){ //si el primer no zero és molt fort, poso error.
                        System.out.println("- Dos enllaços MF seguits.");
                        cLinks.add('e');
                        logE.add(new Log(ii, 14));
                    }else{
                        System.out.println("- Enllaç correcte.");
                        cLinks.add('n'); //si no, poso normal.
                    }
                    i=0;
                    break;
                }
            }
            if(i!=0){ //si he sortit del while sense confirmar res és que sóc al principi i està bé:
                cLinks.add('n');
            }
            break;
        default:
            System.out.println("Demanat corregir un enllaç fora el conjunt conegut.");
            break;
        }
        ii++;
    }
    System.out.println(cLinks);

    /**- - - - Quartes no justes - - - - */
    //TODO quan permeti acords hom, comprovar els enllaços de quarta.

    /**- - - - Cadències i correcció dels graus - - - - - */

    System.out.println("Calculant cadències i correcció dels graus...");
    cGraus.clear();
    cadencies.clear();
    ii=0;
    while(ii<esbos.size){
        int j=1;
        //System.out.println("ii: "+ii);
        if(esbos.get(ii).grau==4 && !(H.mode==1 && !esbos.get(ii).alt7)){ //tinc un V i no és "estant en mode menor i sense sensible" TODO adaptar a dominants secundàries o repetir amb l'equivalent si hi ha diferències.
            System.out.println("Dominant trobada a la posició "+ii);
            i=1;
            while(ii+i<esbos.size){ //asseguro que la cerca no surti dels límits
                //System.out.println("i: "+i);
                if(esbos.get(ii+i).grau==4 && !(H.mode==1 && !esbos.get(ii+i).alt7)){ //si el següent acord segueix sent un V de dominant
                    i++;
                }else{
                    if(esbos.get(ii+i).grau==0){ //si vaig a I, correcte.
                        System.out.println("Resolta a I");
                        cGraus.add('c');
                        for(int k=1;k<i;k++){ //en aquest punt "i" és una més que les dominants extres
                            cGraus.add('i');
                        }
                        ii+=i-1; //ajusto l'iterador global per tenir en compte els extres que he calculat durant el procés
                    }else if(esbos.get(ii+i).grau==3 || esbos.get(ii+i).grau==5){ //si vaig a IV o a VI, correcte i CT.
                        System.out.println("Resolta en CT");
                        cGraus.add('c');
                        cadencies.add(new Cadencia(ii,i));
                        for(int k=1;k<i;k++){ //mateixa filosofia
                            cGraus.add('i');
                        }
                        ii+=i-1;
                        //TODO marcar cadència trencada en una llista o alguna cosa.
```

```
                    }else if(esbos.get(ii+i).grau==1){ //si vaig a II, comprovo ampliació de la dominant.
                        System.out.println("Va a II, ampliació de dominant?");
                        j=1;
                        while(ii+i+j<esbos.size){ //asseguro els límits de la subcerca
                            //System.out.println("j: "+j);
                            if(esbos.get(ii+i+j).grau==1){ //si segueixen essent II...
                                j++;
                            }else{
                                if(esbos.get(ii+i+j).grau==4 && !(H.mode==1 && !esbos.get(ii+i+j).alt7)){ //si he
tornat a V dom, correcte (ampliació de dominant).
                                    System.out.println("Ampliació de la dominant correcta.");
                                    cGraus.add('c');
                                }else{ //si no, no cola. Error.
                                    System.out.println("Resolució incorrecta.");
                                    cGraus.add('e');
                                    logE.add(new Log(ii,501));
                                }
                                j=-j; //per no perdre la quantitat, marco canviant el signe.
                                break;
                            }
                        } //end j while
                        if(j>0){ //si he sortit del while sense definir res...
                            cGraus.add('a'); //alerta al primer V (aquest primer tros és exactament el que hi ha a
fora el while i).
                        }else{ //si havia trobat coses, reverteixo la marca per poder fer factor comú amb els for
que venen ara...
                            j=-j;
                        }
                        for(int k=1;k<i;k++){ //Marco irrellevants les repeticions del V fins al II
                            cGraus.add('i');
                        }
                        ii+=i-1;
                        cGraus.add('n'); //Marco normal el primer II
                        for(int k=1;k<j;k++){
                            cGraus.add('i'); //i irrellevants les següents repeticions
                        }
                        ii+=j;

                    }else if(esbos.get(ii+i).grau==2 && esbos.get(ii+i).dom){ //si vaig a III-, comprovo CT amb
DS
                        System.out.println("CT amb DS?");
                        j=1;
                        while(ii+i+j<esbos.size){ //asseguro els límits de la subcerca
                            System.out.println("j: "+j);
                            if(esbos.get(ii+i+j).grau==2 && esbos.get(ii+i+j).dom){ //si segueixen essent III-...
                                j++;
                            }else{
                                if(esbos.get(ii+i+j).grau==5){ //si he arribat a un VI, correcte (CT amb DS)
                                    System.out.println("Correcte. Cadència trencada amb DS");
                                    cGraus.add('c');
                                    cadencies.add(new Cadencia(ii,true,i+j)); //marco que hi ha ds amb el "true"
del mig.
                                }else{ //si no, no cola. Error.
                                    System.out.println("Resolució incorrecta.");
                                    cGraus.add('e');
                                    logE.add(new Log(ii,502));
                                }
                                j=-j; //per no perdre la quantitat, marco trobat canviant el signe.
                                break;
                            }
                        } //end j while
                        if(j>0){ //si he sortit del while sense definir res...
                            cGraus.add('a'); //alerta al primer V (aquest primer tros és exactament el que hi ha a
fora el while i).
                        }else{ //si havia trobat coses, reverteixo la marca per poder fer factor comú amb els for
que venen ara...
                            j=-j;
                        }
                        for(int k=1;k<i;k++){ //Marco irrellevants les repeticions del V fins al III-
                            cGraus.add('i');
                        }
                        ii+=i-1;
                        cGraus.add('n'); //Marco normal el primer III-
                        for(int k=1;k<j;k++){
                            cGraus.add('i'); //i irrellevants les següents repeticions
                        }
                        ii+=j;
                    }else{
                        cGraus.add('e');
                        logE.add(new Log(ii,16));
```

```
                    }
                    i=-i;
                    break;
                }
            } //end i while
            if(i>0){ //si he quedat penjat buscant que acabés la dominant, poso una alerta i tants irrellevants
com dominants extres hi hagi hagut
                cGraus.add('a');
                for(int k=1;k<i;k++){ //en aquest punt "i" és una més que les dominants extres
                    cGraus.add('i');
                }
                ii+=i-1; //ajusto l'iterador global per tenir en compte els extres que he calculat durant el
procés
            }
        }else if(esbos.get(ii).grau==6 && !(H.mode==1 || esbos.get(ii).hom)){ //si és un setè (en major) marco
error. TODO quan passi a posar dominants secundàries parlem de què fem amb això.
            cGraus.add('e');
            logE.add(new Log(ii,101));
        }else{ //si no era V ni VII, marcar normal...
            if(ii>0){
                if(esbos.get(ii-1).grau==esbos.get(ii).grau){ //...a menys que sigui repetit, que llavors
irrellevant.
                    cGraus.add('i');
                }else{
                    cGraus.add('n');
                }
            }else{
                cGraus.add('n');
            }
        }
        ii++;
    } //end ii while
    System.out.println(cGraus);

    /**- - - - Correcció de les inversions- - - -*/
    System.out.println("Calculant correcció de les inversions...");
    cInv.clear();
    ii=0;
    while(ii<esbos.size){
        if(esbos.get(ii).inv==2 && !(esbos.get(ii).sept || esbos.get(ii).dom)){ //si trobo un 6/4... TODO
preveure impossibilitat en la realització correcta del 6/4 i marcar.
            System.out.println("Trobat 6/4 a la posició "+ii);
            if(esbos.get(ii-1).inv==2 && !(esbos.get(ii-1).sept || esbos.get(ii-1).dom)){ //i el d'abans també
és un 6/4...
                System.out.println("Incorrecte. Dos 6/4 seguits.");
                cInv.add('e');
                logE.add(new Log(ii,27));
            }else{
                cInv.add('n');
            }
        }else{
            cInv.add('n');
        }
        ii++;
    }
    System.out.println(cInv);

    /**- - - - Detecció de la cadència perfecta - - - */
    tincCP=false;
    System.out.println("Comprovant l'acabament de l'esbós..."); //TODO comprovar les diferents versions que
tenim de CP i ajustar en conseqüència.
    if(esbos.size>3 && esbos.get(esbos.size-1).grau==0 && esbos.get(esbos.size-1).inv==0 &&
!esbos.get(esbos.size-1).sept && !esbos.get(esbos.size-3).dom){ //al menys l'esquema tindrà el I del principi, i
SD, V, I.
        System.out.println("- Trobat un I candidat al final de l'esquema");
        if(esbos.get(esbos.size-2).grau==4 && !esbos.get(esbos.size-2).dom && !esbos.get(esbos.size-2).hom &&
!(H.mode==1 && !esbos.get(esbos.size-2).alt7)){
            System.out.println("- Confirmat el V anterior");
            if(esbos.get(esbos.size-3).grau==0 && esbos.get(esbos.size-3).inv==2 && !esbos.get(esbos.size-
3).sept && !esbos.get(esbos.size-3).dom){ //hi ha un I_6/4
                System.out.println("- Confirmat el 6/4 cadencial");
                if(esbos.get(esbos.size-4).grau==1 || esbos.get(esbos.size-4).grau==3){ //subdominant (II o IV)
                    System.out.println("Cadència Autèntica confirmada.");
                    cadencies.add(new Cadencia(esbos.size-4,3,true));
                    tincCP=true;
                    cLinks.removeRange(cLinks.size-2, cLinks.size-1);
                    if(links.get(cLinks.size-1)!=3){ //si aquest enllaç fos molt fort voldria dir que tinc un II
subdominant que fa dos MF seguits (i no m'interessa esborrar-ne la correcció).
                        cLinks.removeIndex(cLinks.size-1); //però si no és molt fort passa a ser irrellevant.
                        cLinks.add('i');
```

```
                    }
                    for(int k=0;k<2;k++){ //converteixo en irrellevants els altres dos enllaços a partir de la
cadència.
                        cLinks.add('i');
                    }
                }
            }
        }
        if(!tincCP){
            System.out.println("No s'ha trobat CP al final");
        }

        /** - - - Càlcul del nombre d'errors - - - */
        neLinks = 0;
        for(Character c: cLinks){
            if(c=='e'){
                neLinks++;
            }
        }
        neGraus = 0;
        for(Character c: cGraus){
            if(c=='e'){
                neGraus++;
            }
        }
        neInv = 0;
        for(Character c: cInv){
            if(c=='e'){
                neInv++;
            }
        }
        nErrors = neGraus+neLinks+neInv;
        System.out.println("\nS'han trobat:\n"+neGraus+" errors de grau\n"+neLinks+" errors d'enllaç\n"+neInv+"
errors d'inversió.\n");

        /**- - Calculo permisos dels botons- - */
        permisosBotons();
    }
}
```

A.1.22. [Pant2.java] – Pantalla de la realització

```
package com.rusca8.hEditor;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.InputAdapter;
import com.badlogic.gdx.Screen;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.OrthographicCamera;
import com.badlogic.gdx.graphics.g2d.SpriteBatch;
import com.badlogic.gdx.math.Vector3;
import com.badlogic.gdx.utils.Align;
import com.badlogic.gdx.utils.Array;
import com.badlogic.gdx.utils.viewport.ExtendViewport;
import com.badlogic.gdx.utils.viewport.Viewport;
import com.rusca8.hEditor.A;
import com.rusca8.hEditor.BotQ;
import com.rusca8.hEditor.Cadencia;
import com.rusca8.hEditor.CorPuntual;
import com.rusca8.hEditor.Desplegat;
import com.rusca8.hEditor.F;
import com.rusca8.hEditor.H;
import com.rusca8.hEditor.Log;
import com.rusca8.hEditor.N;
import com.rusca8.hEditor.Note;
import com.rusca8.hEditor.Pant;
import com.rusca8.hEditor.Recursos;
import com.rusca8.hEditor.S;
import com.rusca8.hEditor.hEditor;

public class Pant2 extends InputAdapter implements Screen {
    hEditor app;

    OrthographicCamera camera;
    Viewport encuadre;
    SpriteBatch lote;

    Vector3 click = new Vector3(0,0,0);
```

```
    Vector3 click2 = new Vector3(0,0,0);

    static float r12 = 0.75f; //ratio entre el pentagrama de la pantalla 1 el d'aquesta
    static float r1e = 0.55f; //ratio entre el pentagrama de la pantalla 1 i el de l'esbós d'aquí.
    static float dEspai = Pant.dEspai*r12; //distància entre línies del pentagrama
    float dEspaiEsb = Pant.dEspai*r1e; //mateix per a l'esbós
    static int ySol = Pant.screenH-170; //alçada del pentagrama en clau de sol.
    static int yFa = (int) (ySol-9*dEspai); //alçada del pentagrama en clau de fa.
    int xSolFa = 160; //posició x on comença l'esquema
    int rxSolFa = 160; //distància des d'on acaba l'esquema fins la paret dreta
    int yEsb = (int) (2.5f*dEspaiEsb);
    int xEsb = 530; //posició x on comença l'esbós (punt fins on tapa la cortina).
    int rxEsb = 170; //distància des d'on acaba l'esbós fins la paret dreta.
    int dEAc = (int) (Pant.dEAc*r12); //distància entre acords
    int dEAcEsb = (int) (dEAc*r1e/r12);
    float dNota = dEspai*1.22f; //diàmetre de la nota (de l'sprite, que és quadrat i té un espai buit).
    float dNotaEsb = dEspaiEsb*1.22f;
    float dAlt = dNota*1.4f; //distància entre l'alteració i la nota
    float dAltEsb = dNotaEsb*1.4f;
    float dArm = dEspai*0.7f; //distància entre alteracions de l'armadura
    float dArmEsb = dEspaiEsb*0.7f;
    int dCad = 180; //distància entre yFa i la senyalització de les cadències.

    int rLim = (int) (Pant.realW*Pant.ratio-rxSolFa);
    int lLim = (int) (310+dArm*Math.abs(H.armadura)); //ajusto el límit esquerra de l'scroll en funció de
l'armadura
    int rLimEsb = (int) (Pant.realW*Pant.ratio-rxEsb); //compte que rLim surt a resize.
    int lLimEsb = (int) (xEsb+120f+dArmEsb*Math.abs(H.armadura)); //límit (valor 0) de l'scroll de l'esbós. Compte
que ambdós lLim surten a show per si s'ha canviat l'armadura.
    int scroll = lLim;
    int scrollEsb = lLimEsb;

    int ii = 0; //iterador global
    byte nAc = 0; //quantitat de notes a pintar en un acord de l'esbós

    //pantalla de referències
    int yTess = (int) (Pant.screenH*0.42f); //compte, canvi a resize
    int wTess = 190;
    int dTess = (int) (Pant.realW*Pant.ratio/4.2f); //compte, canvia a resize
    int mTess = (int) ((Pant.realW*Pant.ratio-(3*dTess+wTess))/2);

    /**- - - Esquema i correcció - - - */
    byte vAct = 0; //veu actual
    static byte aAct = 0; //acord actual
    static Array<Desplegat> esquema = new Array<Desplegat>(); //per guardar quina nota (num de la nota dins
l'acord) va a cada veu. (la nota real la deduïré quan la necessiti a partir d'aquest num i altres dades com l'acord
i la tonalitat, així que no em caldrà guardar-la quan vulgui permetre desar els esquemes).
    Desplegat dAux; //auxiliar per no haver de fer .get tota l'estona. TODO si veig que no val la pena, treure'l.

    Array<boolean[]> tincNotes = new Array<boolean[]>(); //per guardar quines notes he posat i quines no durant
la correcció (13579).
    Array<Boolean> tincPle = new Array<Boolean>(); //per marcar si hi ha notes a totes les veus dins l'acord
    Array<Boolean> complet = new Array<Boolean>(); //per marcar si l'acord està complet (si hi ha totes les notes
que hi hauria d'haver).
    Array<Character> cComplet = new Array<Character>(); //correcció dels incomplets.
    Array<Byte> duplicades = new Array<Byte>(); //per desar quina veu s'ha duplicat en cada acord
    Array<Character> cDuplicades = new Array<Character>(); //correcció de les duplicacions
    Array<Byte> inversions = new Array<Byte>(); //per guardar les inversions reals dels acords (no les triades a
l'esbós, sinó les deduïdes de la posició de les notes).
    Array<Character> cInv = new Array<Character>(); //correcció de les inversions. Equivalent al de Pant1, però
marcaré també la divergència respecte les inversions escollides a l'esbós.
    Array<CorPuntual> cTessitures = new Array<CorPuntual>(); //correcció de les tessitures
    Array<CorPuntual> cCreuades = new Array<CorPuntual>(); //correcció dels creuaments
    Array<CorPuntual> cCreuadesH = new Array<CorPuntual>(); //correcció dels creuaments horitzontals
    Array<CorPuntual> cSeparades = new Array<CorPuntual>(); //correcció de les distàncies de més d'una octava
    Array<CorPuntual> cPar8 = new Array<CorPuntual>(); //octaves paral·leles
    Array<CorPuntual> cPar5 = new Array<CorPuntual>(); //quintes paral·leles
    Array<CorPuntual> cPar7 = new Array<CorPuntual>(); //sèptimes paral·leles
    Array<CorPuntual> lligadures = new Array<CorPuntual>(); //per guardar les ubicacions de les lligadures
    static Array<CorPuntual> solapades = new Array<CorPuntual>();
    Array<CorPuntual> cPrep7 = new Array<CorPuntual>(); //preparació de les sèptimes
    boolean septMajor; //per marcar si la sèptima trobada és major (requereix preparació especial.
    Array<CorPuntual> cRes7 = new Array<CorPuntual>(); //resolució de les sèptimes
    Array<CorPuntual> cRI7 = new Array<CorPuntual>(); //resolucions irregulars de la sèptima i correcció
d'aquestes.
    byte maxRI7 = 2; //màxim de resolucions irregulars que permetré a cada esquema.
    Array<CorPuntual> c64PrepB = new Array<CorPuntual>(); //preparació del baix al 6/4
    Array<CorPuntual> c64ResB = new Array<CorPuntual>(); //resolució del baix al 6/4
    Array<CorPuntual> c64Prep4 = new Array<CorPuntual>(); //preparació de la quarta al 6/4
    Array<CorPuntual> c64Res4 = new Array<CorPuntual>(); //resolució de la quarta al 6/4
```

```
    Array<CorPuntual> cAug = new Array<CorPuntual>(); //correcció dels intervals augmentats (horitzontal)
    Array<CorPuntual> cDim = new Array<CorPuntual>(); //correcció dels intervals disminuïts (horitzontal)
    Array<CorPuntual> cItv = new Array<CorPuntual>(); //per si hi ha algun interval molt estrany.
    Array<Character> cCad = new Array<Character>(); //deso la correcció de les cadències (no em cal desar on són ni
res perquè ja surt a "Pant.cadencies").

    static Array<Note[]> esquemaEnNotes = new Array<Note[]>(); //pels càlculs de la correcció em caldrà disposar
de les notes reals que hi ha a l'esquema ("esquema" només guarda quina posició té la nota dins l'acord, no el valor
real d'aquesta nota).
    //la matriu esquemaEnNotes es calcula a partir de les dades conegudes del context i la matriu "esquema" a la
funció "tradueixNotes()" ubicada dins les funcions de correcció.

    boolean puc7a, puc9a; //per desactivar els botons si no puc posar la nota a l'acord
    byte altOp; //alteracions opcionals (guardo resultat de H.opcionals)

    static boolean nouEsquema = true; //per què la primera vegada que obri aquesta pantalla es generi l'esquema
amb tot zeros.

    int nErrors; //per desar les quantitats d'errors i alertes que hi ha a la realització
    int nAlertes;

    Array<Log> logE = new Array<Log>();
    Array<Log> logA = new Array<Log>();

    /** - - - Coses de botons - - - */
    Array<BotQ> bVeus = new Array<BotQ>(); //botons de les veus
    Array<BotQ> bNotes = new Array<BotQ>(); //botons de les notes de l'acord
    BotQ bEsq, bDr; //botons de navegar per l'esquema
    BotQ bEsborra;
    BotQ bMenu;

    BotQ bAlt1,bAlt2;

    boolean clickBot = false; //com a la pantalla de l'esbós
    boolean calCorregir = false;

    //botons del menú
    boolean mostraMenu = false;

    BotQ bErrors,bAlertes; //botons que clicaré per veure quins errors i alertes hi ha
    boolean mostraErrors = false, mostraAlertes = false;
    int errorAct, alertaAct;
    int xLogs; //posició esquerra del rectangle dels registres (centre de les circumferències)
    int yLogs; //posició inferior del rectangle dels registres (centre de les circumferències)
    BotQ bAnt, bSeg;

    int dBM = 120; //distància entre botons del menú
    BotQ bBuida;
    BotQ bEsbos;
    BotQ bCancel;
    boolean vullBuidar=false;
    boolean vullTornar=false;

    /**- - Reproducció d'àudio - -*/
    BotQ bEscolta;
    boolean pucEscoltar = false; //marca si el botó d'escoltar ja permet que es reprodueixi (i.e. ja està tot
carregat)
    boolean escoltant = false; //marca si s'està reproduint l'esquema
    float tAcord;
    byte lastAct; //per desar l'últim actual abans de reproduir (i tornar-hi)

    /****** CONSTRUCTOR ******/
    public Pant2(hEditor app){
        this.app = app;

        lote = new SpriteBatch();

        camera = new OrthographicCamera();
        encuadre = new ExtendViewport(0,Pant.screenH, camera);
        encuadre.apply(false);

        System.out.println("\nInicialitzant en segon pla la pantalla SATB...");

        //botons
        for(int i=0;i<4;i++){
            bVeus.add(new BotQ(80f,Pant.screenH*((5-i)/6f+0.04f),120f,120f));
        }
        for(int i=0;i<5;i++){
            bNotes.add(new BotQ(Pant.realW*Pant.ratio-75,Pant.screenH*((5f-i)/6.15f)+80,120,120,(9-2*i))); //x a
resize
```

```
        }
        bEsborra = new BotQ(Pant.realW*Pant.ratio-75,80,120,120,"X"); //x a resize
        bEsq = new BotQ(290,75,120,120);
        bDr = new BotQ(430,75,120,120);

        bAlt1 = new BotQ(Pant.realW*Pant.ratio-375,0,120,120);
        bAlt2 = new BotQ(Pant.realW*Pant.ratio-225,0,120,120);

        //botons del menú
        bMenu = new BotQ(110f,110f,170f,170f);
        bErrors = new BotQ(110f,Pant.screenH-110f,130f,130f);
        bAlertes = new BotQ(110f,Pant.screenH-270f,130f,130f);

        xLogs = (int) (bErrors.x+bErrors.w*1.38f);
        yLogs = (int) (Pant.screenH*0.3f);

        bAnt = new BotQ(xLogs+(Pant.realW*Pant.ratio-bErrors.w*0.75f-xLogs)/4f  -20f, Pant.screenH*0.11f,
(Pant.realW*Pant.ratio-bErrors.w*0.75f-xLogs)*0.37f,110,"Anterior");
        bSeg = new BotQ(xLogs+(Pant.realW*Pant.ratio-bErrors.w*0.75f-xLogs)*3/4f+20f, Pant.screenH*0.11f,
(Pant.realW*Pant.ratio-bErrors.w*0.75f-xLogs)*0.37f,110,"Següent");

        bEscolta = new BotQ((Pant.realW*Pant.ratio-bErrors.w*0.75f+xLogs)/2f,bMenu.y+3.5f*dBM,400,80,"Escolta-
ho!");
        bEsbos = new BotQ((Pant.realW*Pant.ratio-bErrors.w*0.75f+xLogs)/2f,bMenu.y+dBM,400,80,"Confirmar");
//posició i w irrellevants, surten a posaElsBotonsPer();

        bBuida = new BotQ((Pant.realW*Pant.ratio-bErrors.w*0.75f+xLogs)/2f,bMenu.y,400,80,"Buidar realització");

        bCancel = new BotQ(Pant.realW*Pant.ratio/2f+180,bMenu.y+dBM,300,80,"Cancel·lar");

        //comprovo que les funcions H estan correctament implementades (trec resultats per pantalla per si vull
mirar què passa, vaja).
        System.out.println("Comprovant funcions musicals de la realització...");
        System.out.println("\nLes tessitures de les veus són:\nS: "+H.notes[H.tMinS.nom]+" "+H.tMinS.ia+" -
"+H.notes[H.tMaxS.nom]+" "+H.tMaxS.ia);
        System.out.println("A: "+H.notes[H.tMinA.nom]+" "+H.tMinA.ia+" - "+H.notes[H.tMaxA.nom]+" "+H.tMaxA.ia);
        System.out.println("T: "+H.notes[H.tMinT.nom]+" "+H.tMinT.ia+" - "+H.notes[H.tMaxT.nom]+" "+H.tMaxT.ia);
        System.out.println("B: "+H.notes[H.tMinB.nom]+" "+H.tMinB.ia+" - "+H.notes[H.tMaxB.nom]+" "+H.tMaxB.ia);
    }


    /*********** RENDER ***********/
    @Override
    public void render(float delta) {
        delta=Math.min(delta, 0.25f);

        Gdx.gl.glClearColor(S.fons.r,S.fons.g,S.fons.b,S.fons.a);
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);

        camera.update();
        lote.setProjectionMatrix(camera.combined);

        /**- Execució de la correcció -*/
        if(calCorregir){
            corregeix();
            pucEscoltar=false; //si he corregit és que he modificat (i podria no tenir tots els sons carregats).
            calCorregir=false;
            System.out.println("Delta: "+delta);
        }

        lote.begin();
        if(!mostraMenu){

            /** - - Pentagrames, claus i armadura - -*/
            lote.setColor(S.pentagrama);
            for(int i=0; i<5; i++){ //dibuixo els tres pentagrames (Sol, fa i l'esbós).
                lote.draw(Recursos.rect, 0, ySol+dEspai*i, Pant.realW*Pant.ratio, 3f);
                lote.draw(Recursos.rect, 0, yFa+dEspai*i, Pant.realW*Pant.ratio, 3f);
                lote.draw(Recursos.rect, 0, yEsb+dEspaiEsb*i, Pant.realW*Pant.ratio, 2f);
            }
            lote.draw(Recursos.sol, xSolFa+15f+(scroll-lLim), ySol-56*r12, dEspai*4f, dEspai*8f);
            lote.draw(Recursos.fa, xSolFa+15f+(scroll-lLim), yFa-56*r12, dEspai*4f, dEspai*8f);
            lote.draw(Recursos.rect, xSolFa+(scroll-lLim), yFa, 5f, ySol+dEspai*4f-yFa); //barra que ajunta les
claus
            lote.draw(Recursos.rect, scroll+dEAc*(esquema.size)-10f, yFa, 12f, ySol+dEspai*4f-yFa); //doble barra
final
            lote.draw(Recursos.rect, scroll+dEAc*(esquema.size)-18f, yFa, 3f, ySol+dEspai*4f-yFa);
            lote.draw(Recursos.sol, xEsb+10f+(scrollEsb-lLimEsb), yEsb-56*r1e,dEspaiEsb*4f,dEspaiEsb*8f); //clau de
l'esbós
            lote.draw(Recursos.rect, scrollEsb+dEAcEsb*esquema.size-8f, yEsb, 12f, dEspaiEsb*4f); //doble barra
```

```
final esbós
            lote.draw(Recursos.rect, scrollEsb+dEAcEsb*esquema.size-14f, yEsb, 2f, dEspaiEsb*4f);
            if(H.armadura>0){ //dibuixo l'armadura. TODO mateix que a la pantalla de l'esbós. (gestió de x i bb)
                ii=0;
                while(ii<H.armadura){
                    lote.draw(Recursos.sharp, xSolFa+15f+(scroll-lLim)+dEspai*3.5f+ii*dArm,
ySol+H.armPos((3+4*ii)%7,true,false)-dNota*0.38f, dNota*0.5f*1.7f, dNota*1.7f);
                    lote.draw(Recursos.sharp, xSolFa+15f+(scroll-lLim)+dEspai*3.5f+ii*dArm,
yFa+H.armPos((3+4*ii)%7,true,true)-dNota*0.38f, dNota*0.5f*1.7f, dNota*1.7f);
                    lote.draw(Recursos.sharp, xEsb+10f+(scrollEsb-lLimEsb)+dEspaiEsb*3.5f+ii*dArmEsb,
yEsb+H.armPos((3+4*ii)%7,true,dEspaiEsb)-dNotaEsb*0.38f, dNotaEsb*0.5f*1.7f, dNotaEsb*1.7f);
                    ii++;
                }
            }else if(H.armadura<0){
                ii=0;
                while(ii>H.armadura){
                    lote.draw(Recursos.bemoll, xSolFa+15f+(scroll-lLim)+dEspai*3.5f+(-ii)*dArm, ySol+H.armPos((6+3*(-
ii))%7,false,false)-dNota*0.05f, dNota*0.5f*1.7f, dNota*1.7f);
                    lote.draw(Recursos.bemoll, xSolFa+15f+(scroll-lLim)+dEspai*3.5f+(-ii)*dArm, yFa+H.armPos((6+3*(-
ii))%7,false,true)-dNota*0.05f, dNota*0.5f*1.7f, dNota*1.7f);
                    lote.draw(Recursos.bemoll, xEsb+10f+(scrollEsb-lLimEsb)+dEspaiEsb*3.5f+(-ii)*dArmEsb,
yEsb+H.armPos((6+3*(-ii))%7, false,dEspaiEsb)-dNotaEsb*0.05f, dNotaEsb*0.5f*1.7f, dNotaEsb*1.7f);
                    ii--;
                }
            }


            ii=0;
            for(Note[] ac: esquemaEnNotes){ //per cada acord desplegat (per cada acord a l'esbós)

                /** - - - DIBUIX NOTES REALITZACIÓ - - -*/

                for(int veu=0;veu<4;veu++){ //escric les notes de la realització
                    if(ac[veu].nom!=-1){ //si hi ha nota (-1 és que no hi ha res definit)
                        if(hiHaCor(cTessitures,ii,veu)){ //si tinc una correcció sobre la tessitura, pinto la nota
del color de la correcció
                            lote.setColor(S.cColor(F.lastCor.cor));
                            lote.setColor(lote.getColor().r,lote.getColor().g,lote.getColor().b,0.4f); //atenuat del
color (poso transparència)
                            lote.draw(Recursos.cir, scroll+dEAc*ii-dNota/2+dNota*H.ajustSolapades(ii,veu),
H.ySolFa(veu)+H.notePos(ac[veu], veu), dNota, (dNota+dEspai)/2f);
                        }else if(veu==vAct){ //si no tinc correcció, omplo les notes de la veu actual
                            if(ii==aAct){ //(i marco diferent si a més són de l'acord actual)
                                lote.setColor(S.actual);
                            }else{
                                lote.setColor(S.pentagrama);
                            }
                            lote.setColor(lote.getColor().r,lote.getColor().g,lote.getColor().b,0.2f); //atenuat del
color
                            lote.draw(Recursos.cir, scroll+dEAc*ii-dNota/2+dNota*H.ajustSolapades(ii,veu),
H.ySolFa(veu)+H.notePos(ac[veu], veu), dNota, (dNota+dEspai)/2f);
                        }


                        //pinto la vora de la nota
                        if(ii==aAct){ //marco de color l'acord actual.
                            lote.setColor(S.actual);
                        }else{
                            lote.setColor(S.pentagrama);
                        }
                        lote.draw(Recursos.nota, scroll+dEAc*ii-dNota/2+dNota*H.ajustSolapades(ii,veu),
H.ySolFa(veu)+H.notePos(F.keep(ac[veu]), veu), dNota, dNota);

                        if(H.lastNotePos>dEspai*4f){ //línia addicional del La4 (i Do3 en clau de fa)
                            lote.draw(Recursos.rect, scroll+dEAc*ii-dNota*0.75f+dNota*H.lastSolapades,
H.ySolFa(veu)+dEspai*5f, dNota*1.5f, 3f);

                            if(H.lastNotePos>dEspai*5f){ //línia addicional del Mi3 (clau de fa)
                                lote.draw(Recursos.rect, scroll+dEAc*ii-dNota*0.75f+dNota*H.lastSolapades,
H.ySolFa(veu)+dEspai*6f, dNota*1.5f, 3f);

                                if(H.lastNotePos>dEspai*6f){ //línia addicional del Sol3 (clau de fa)
                                    lote.draw(Recursos.rect, scroll+dEAc*ii-dNota*0.75f+dNota*H.lastSolapades,
H.ySolFa(veu)+dEspai*7f, dNota*1.5f, 3f);
                                }
                            }
                        }else{ //línies addicionals per sota
                            if(H.lastNotePos<dEspai*(-1f)){ //línia del Do3 (i Mi1 en clau de fa)
                                lote.draw(Recursos.rect, scroll+dEAc*ii-dNota*0.75f+dNota*H.lastSolapades,
H.ySolFa(veu)-dEspai, dNota*1.5f, 3f);
```

```
                            if(H.lastNotePos<dEspai*(-2f)){ //línia del La2 (clau de sol)
                                lote.draw(Recursos.rect, scroll+dEAc*ii-dNota*0.75f+dNota*H.lastSolapades,
H.ySolFa(veu)-dEspai*2f, dNota*1.5f, 3f);

                                if(H.lastNotePos<dEspai*(-3f)){ //línia del Fa2 (clau de sol)
                                    lote.draw(Recursos.rect, scroll+dEAc*ii-dNota*0.75f+dNota*H.lastSolapades,
H.ySolFa(veu)-dEspai*3f, dNota*1.5f, 3f);
                                }
                            }
                        }
                    } //fi línies addicionals

                    /** - - - Alteracions accidentals - - -*/

                    if(H.accidental(F.lastNote)){ //Afegeixo les alteracions accidentals.
                        switch(H.getAltNum(F.lastNote)){
                        case 1:
                            lote.draw(Recursos.sharp, scroll+dEAc*ii-dAlt, H.ySolFa(veu)+H.lastNotePos-
dNota*0.38f, dNota*0.5f*1.7f, dNota*1.7f);
                            break;
                        case 0:
                            lote.draw(Recursos.natural, scroll+dEAc*ii-dAlt, H.ySolFa(veu)+H.lastNotePos-
dNota*0.38f, dNota*0.5f*1.7f, dNota*1.7f);
                            break;
                        case -1:
                            lote.draw(Recursos.bemoll, scroll+dEAc*ii-dAlt, H.ySolFa(veu)+H.lastNotePos-
dNota*0.05f, dNota*0.5f*1.5f, dNota*1.5f);
                            break;
                        default:
                            System.out.println("Accidental sense dibuix!!!
"+H.notes[F.lastNote.nom]+H.getAlt(F.lastNote));
                        }
                    } //fi accidentals

                    /**- - -Lligadures - - -*/
                    if(hiHaCor(lligadures,ii,veu)){ //TODO ajustar les lligadures a les notes solapades.
                        lote.setColor(S.pentagrama);
                        if(veu%2==0){ // Veu superior del pentagrama (S i T)
                            lote.draw(Recursos.lligat, scroll+dEAc*ii+dNota*H.lastSolapades,
H.ySolFa(veu)+H.lastNotePos+dEspai,dEAc-dNota*H.lastSolapades+dNota*H.ajustSolapades(ii+1, veu, false),dEAc*0.2f);
//la proporció natural de la imatge és 4x1
                        }else if(veu==1){ //veu inferior (lligadura invertida) (A) (no lligo el baix)
                            lote.draw(Recursos.lligat,scroll+dEAc*ii, H.ySolFa(veu)+H.lastNotePos,dEAc,-
dEAc*0.2f);
                        }
                    }

                    /**- - - Correccions locals - - -*/

                    if(veu>0){ //Part de les correccions que mai van ancorades a la soprano (intervals
verticals).
                        //Execució de la correcció de veus separades
                        if(hiHaCor(cSeparades,ii,veu)){ //si la nota està a més d'una octava de la veu
superior...
                            Recursos.xifrat.setColor(S.error); //TODO calcular distància per centrar el text.
                            Recursos.xifrat.draw(lote,F.keep("+8!"),
scroll+dEAc*ii+F.align("xifrat"),H.ySolFa(veu)+H.lastNotePos+dEspai*(1+F.lastCor.aux*0.5f)*0.5f+F.vAlign("xifrat"))
;
                        }

                        //execució de la correcció de moviments paral·lels
                        if(hiHaCor(cPar8,ii,veu)){ // Octaves
                            lote.setColor(S.cColor(F.lastCor.cor));
                            lote.draw(Recursos.rect,scroll+dEAc*ii+dNota+18,H.ySolFa(veu)+H.lastNotePos, 5f,
(H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii)[F.lastCor.aux], F.lastCor.aux, false))-
(H.ySolFa(veu)+H.lastNotePos) +dEspai); //marco l'interval. (el false de notePos és perquè no la guardi a
lastNotePos (per mantenir a lastNotePos la posició de la nota actual).

    lote.draw(Recursos.rect,scroll+dEAc*(ii+1)+dNota+18,H.ySolFa(veu)+H.notePos(esquemaEnNotes.get(ii+1)[veu], veu,
false), 5f, (H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii+1)[F.lastCor.aux], F.lastCor.aux, false))-
(H.ySolFa(veu)+H.notePos(esquemaEnNotes.get(ii+1)[veu], veu, false)) +dEspai);
                        }
                        if(hiHaCor(cPar5,ii,veu)){ // Quintes
                            lote.setColor(S.cColor(F.lastCor.cor));
                            lote.draw(Recursos.rect,scroll+dEAc*ii+dNota+9,H.ySolFa(veu)+H.lastNotePos, 5f,
(H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii)[F.lastCor.aux], F.lastCor.aux, false))-
(H.ySolFa(veu)+H.lastNotePos) +dEspai);

    lote.draw(Recursos.rect,scroll+dEAc*(ii+1)+dNota+9,H.ySolFa(veu)+H.notePos(esquemaEnNotes.get(ii+1)[veu], veu,
```

```
false), 5f, (H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii+1)[F.lastCor.aux], F.lastCor.aux, false))-
(H.ySolFa(veu)+H.notePos(esquemaEnNotes.get(ii+1)[veu], veu, false)) +dEspai);
                        }
                        if(hiHaCor(cPar7,ii,veu)){ // Sèptimes
                            lote.setColor(S.cColor(F.lastCor.cor));
                            lote.draw(Recursos.rect,scroll+dEAc*ii+dNota,H.ySolFa(veu)+H.lastNotePos, 5f,
(H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii)[F.lastCor.aux], F.lastCor.aux, false))-
(H.ySolFa(veu)+H.lastNotePos) +dEspai); //marco l'interval. (el false de notePos és perquè no la guardi a
lastNotePos (per mantenir a lastNotePos la posició de la nota actual).

    lote.draw(Recursos.rect,scroll+dEAc*(ii+1)+dNota,H.ySolFa(veu)+H.notePos(esquemaEnNotes.get(ii+1)[veu], veu,
false), 5f, (H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii+1)[F.lastCor.aux], F.lastCor.aux, false))-
(H.ySolFa(veu)+H.notePos(esquemaEnNotes.get(ii+1)[veu], veu, false)) +dEspai);
                        }

                        //execució de la correcció de veus creuades
                        if(hiHaCor(cCreuades,ii,veu)){
                            if(veu%2==1){ //si és veu imparell (A o B) significa que és dins el mateix pentagrama.
Marco alerta per si es vol corregir canviant les notes de veu.
                                Recursos.xifrat.setColor(S.alerta);
                                Recursos.xifrat.draw(lote,F.keep("x"), scroll+dEAc*ii-dAlt,
H.ySolFa(veu)+H.lastNotePos+dEspai*(1+F.lastCor.aux*0.5f)*0.5f+F.vAlign("xifrat")+4f);
                            }else{ //si és entre pentagrames (T) marco error.
                                Recursos.xifrat.setColor(S.error);
                                Recursos.xifrat.draw(lote,F.keep("x"), scroll+dEAc*ii-dAlt,
H.ySolFa(veu)+H.lastNotePos+dEspai*(1+(F.lastCor.aux+6f)*0.5f)*0.5f+F.vAlign("xifrat")+4f);
                            }
                        }
                    }
                    //execució de la correcció de creuaments horitzontals
                    if(hiHaCor(cCreuadesH,ii,veu)){
                        Recursos.font.setColor(S.error);
                        if(F.lastCor.cor=='+'){ //si és ascendent...
                            Recursos.font.draw(lote,F.keep("-------"), scroll+dEAc*ii+F.align('r')*0.8f,
H.ySolFa(veu)+H.notePos(esquemaEnNotes.get(ii-1)[F.lastCor.aux], veu,
false)+dEspai*0.5f*(1+0.5f*H.sobreLinia(esquemaEnNotes.get(ii-1)[F.lastCor.aux]))+7f+F.vAlign());
                        }else{ //canvio l'efecte de "sobreLinia" per deixar la línia a baix si és descendent.
                            Recursos.font.draw(lote,F.keep("-------"), scroll+dEAc*ii+F.align('r')*0.8f,
H.ySolFa(veu)+H.notePos(esquemaEnNotes.get(ii-1)[F.lastCor.aux], veu, false)+dEspai*0.5f*(1-
0.5f*H.sobreLinia(esquemaEnNotes.get(ii-1)[F.lastCor.aux]))+7f+F.vAlign());
                        }
                    }

                    //execució de la correcció de preparació de sèptimes
                    if(hiHaCor(cPrep7,ii,veu)){
                        lote.setColor(S.cColor(F.lastCor.cor));
                        if(veu==vAct){

    lote.setColor(lote.getColor().r*0.85f,lote.getColor().g*0.85f,lote.getColor().b*0.85f,1f);
                        }
                        if(veu%2==0){ //veu de dalt del pentagrama (moc una mica els punts quan se solapa perquè
no se solapin els punts)
                            lote.draw(Recursos.cir, scroll+dEAc*ii-dAlt*0.8f+5*H.lastSolapades,
H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, 10f, 10f);
                        }else{ //veu de baix
                            lote.draw(Recursos.cir, scroll+dEAc*ii-dAlt*0.8f-6*H.ajustSolapades(ii,veu-1,false),
H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, 10f, 10f);
                        }

                        if(F.lastCor.aux!=F.lastCor.veu){ //si també està implicada la fonamental...
                            lote.setColor(S.cColor(F.lastCor.cor));
                            if(F.lastCor.aux==vAct){

    lote.setColor(lote.getColor().r*0.85f,lote.getColor().g*0.85f,lote.getColor().b*0.85f,1f);
                            }
                            if(F.lastCor.aux%2==0){ //veu superior
                                lote.draw(Recursos.cir, scroll+dEAc*ii-
dAlt*0.8f+5*H.ajustSolapades(ii,F.lastCor.aux,false),
H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii)[F.lastCor.aux], F.lastCor.aux,
false)+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[F.lastCor.aux]))-3.5f, 10f, 10f);
                            }else{ //veu inferior
                                lote.draw(Recursos.cir, scroll+dEAc*ii-dAlt*0.8f-
6*H.ajustSolapades(ii,F.lastCor.aux-1,false),
H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii)[F.lastCor.aux], F.lastCor.aux,
false)+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[F.lastCor.aux]))-3.5f, 10f, 10f);
                            }
                        }
                    }
                    //execució de la correcció de resolució de sèptimes
                    if(hiHaCor(cRes7,ii,veu)){
```

```
                        lote.setColor(S.cColor(F.lastCor.cor));
                        if(veu==vAct){

    lote.setColor(lote.getColor().r*0.85f,lote.getColor().g*0.85f,lote.getColor().b*0.85f,1f);
                        }
                        if(veu%2==0){ //veu de dalt del pentagrama
                            lote.draw(Recursos.cir, scroll+dEAc*ii+dAlt*0.8f+(dNota+6)*(H.lastSolapades),
H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, -10f, 10f);
                        }else{ //veu de baix (separo per l'ajust per solapament)
                            lote.draw(Recursos.cir, scroll+dEAc*ii+dAlt*0.8f+(dNota-5)*(H.ajustSolapades(ii, veu-
1, false)), H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, -10f,
10f);
                        }

                    }
                    //dibuix de les resolucions irregulars TODO en algun moment incorporar un dibuix decent d'una
línia de punts.
                    if(hiHaCor(cRI7,ii,veu)){
                        lote.setColor(S.cColor(F.lastCor.cor));
                        for(int n=2;n<7;n++){
                            lote.draw(Recursos.cir, scroll+dEAc*(ii+n/8f), (H.ySolFa(veu)+H.lastNotePos)*((8-
n)/8f)+(n/8f)*(H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii+1)[F.lastCor.aux], F.lastCor.aux,
false))+dEspai*0.5f, -6f, 6f);
                        }
                    }
                    //execució de la correcció de 6/4
                    if(veu==3){ //preparació i resolució del baix.
                        if(hiHaCor(c64PrepB,ii,veu)){
                            lote.setColor(S.cColor(F.lastCor.cor));
                            if(veu==vAct){

    lote.setColor(lote.getColor().r*0.85f,lote.getColor().g*0.85f,lote.getColor().b*0.85f,1f);
                            }
                            if(veu%2==0){ //veu de dalt del pentagrama
                                lote.draw(Recursos.cir, scroll+dEAc*ii-dAlt*0.8f+5*H.lastSolapades,
H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, 10f, 10f);
                            }else{ //veu de baix
                                lote.draw(Recursos.cir, scroll+dEAc*ii-dAlt*0.8f-6*H.ajustSolapades(ii,veu-
1,false), H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, 10f, 10f);
                            }
                        }
                        if(hiHaCor(c64ResB,ii,veu)){
                            lote.setColor(S.cColor(F.lastCor.cor));
                            if(veu==vAct){

    lote.setColor(lote.getColor().r*0.85f,lote.getColor().g*0.85f,lote.getColor().b*0.85f,1f);
                            }
                            if(veu%2==0){
                                lote.draw(Recursos.cir, scroll+dEAc*ii+dAlt*0.8f+(dNota+6)*H.lastSolapades,
H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, -10f, 10f);
                            }else{
                                lote.draw(Recursos.cir, scroll+dEAc*ii+dAlt*0.8f+(dNota-5)*H.ajustSolapades(ii,
veu-1, false), H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, -10f,
10f);
                            }
                        }
                    }else{ //preparació i resolució de la quarta.
                        if(hiHaCor(c64Prep4,ii,veu)){
                            lote.setColor(S.cColor(F.lastCor.cor));
                            if(veu==vAct){

    lote.setColor(lote.getColor().r*0.85f,lote.getColor().g*0.85f,lote.getColor().b*0.85f,1f);
                            }
                            if(veu%2==0){ //veu de dalt del pentagrama
                                lote.draw(Recursos.cir, scroll+dEAc*ii-dAlt*0.8f+5*H.lastSolapades,
H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, 10f, 10f);
                            }else{ //veu de baix
                                lote.draw(Recursos.cir, scroll+dEAc*ii-dAlt*0.8f-6*H.ajustSolapades(ii,veu-
1,false), H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, 10f, 10f);
                            }
                            if(F.lastCor.aux!=F.lastCor.veu){ //si hi ha una segona quarta implicada...
                                lote.setColor(S.cColor(F.lastCor.cor));
                                if(F.lastCor.aux==vAct){

    lote.setColor(lote.getColor().r*0.85f,lote.getColor().g*0.85f,lote.getColor().b*0.85f,1f);
                                }
                                if(F.lastCor.aux%2==0){ //veu superior
                                    lote.draw(Recursos.cir, scroll+dEAc*ii-
dAlt*0.8f+5*H.ajustSolapades(ii,F.lastCor.aux,false),
H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii)[F.lastCor.aux], F.lastCor.aux,
```

```
false)+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[F.lastCor.aux]))-3.5f, 10f, 10f);
                                }else{ //veu inferior
                                    lote.draw(Recursos.cir, scroll+dEAc*ii-dAlt*0.8f-
6*H.ajustSolapades(ii,F.lastCor.aux-1,false),
H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii)[F.lastCor.aux], F.lastCor.aux,
false)+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[F.lastCor.aux]))-3.5f, 10f, 10f);
                                }
                            }
                        }
                        if(hiHaCor(c64Res4,ii,veu)){
                            lote.setColor(S.cColor(F.lastCor.cor));
                            if(veu==vAct){

    lote.setColor(lote.getColor().r*0.85f,lote.getColor().g*0.85f,lote.getColor().b*0.85f,1f);
                            }
                            if(veu%2==0){
                                lote.draw(Recursos.cir, scroll+dEAc*ii+dAlt*0.8f+(dNota+6)*H.lastSolapades,
H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, -10f, 10f);
                            }else{
                                lote.draw(Recursos.cir, scroll+dEAc*ii+dAlt*0.8f+(dNota-5)*H.ajustSolapades(ii,
veu-1, false), H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, -10f,
10f);
                            }
                            if(F.lastCor.aux!=F.lastCor.veu){ //si hi ha una segona quarta implicada...
                                lote.setColor(S.cColor(F.lastCor.cor));
                                if(F.lastCor.aux==vAct){

    lote.setColor(lote.getColor().r*0.85f,lote.getColor().g*0.85f,lote.getColor().b*0.85f,1f);
                                }
                                if(F.lastCor.aux%2==0){
                                    lote.draw(Recursos.cir,
scroll+dEAc*ii+dAlt*0.8f+(dNota+6)*H.ajustSolapades(ii,F.lastCor.aux,false),
H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii)[F.lastCor.aux], F.lastCor.aux,
false)+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[F.lastCor.aux]))-3.5f, -10f, 10f);
                                }else{
                                    lote.draw(Recursos.cir, scroll+dEAc*ii+dAlt*0.8f+(dNota-
5)*H.ajustSolapades(ii,F.lastCor.aux-1,false),
H.ySolFa(F.lastCor.aux)+H.notePos(esquemaEnNotes.get(ii)[F.lastCor.aux], F.lastCor.aux,
false)+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[F.lastCor.aux]))-3.5f, -10f, 10f);
                                }
                            }
                        }
                    }
                    //correcció d'augmentats
                    if(hiHaCor(cAug,ii,veu)){
                        Recursos.xifrat.setColor(S.error);
                        Recursos.xifrat.draw(lote,F.keep("A"), scroll+dEAc*(ii+0.5f)+F.align("xifrat"),
H.ySolFa(veu)+dEspai*0.5f+(H.lastNotePos+H.notePos(esquemaEnNotes.get(ii+1)[veu], veu,
false))*0.5f+F.vAlign("xifrat"));
                    }
                    //correcció d'intervals estranys TODO acabar de mirar si està tot malament o si hi ha alguna
excepció amb els intervals petits
                    if(hiHaCor(cItv,ii,veu)){
                        Recursos.xifrat.setColor(S.error);
                        Recursos.xifrat.draw(lote,F.keep("?!"), scroll+dEAc*(ii+0.5f)+F.align("xifrat"),
H.ySolFa(veu)+dEspai*0.5f+(H.lastNotePos+H.notePos(esquemaEnNotes.get(ii+1)[veu], veu,
false))*0.5f+F.vAlign("xifrat"));
                    }
                    //correcció de disminuïts
                    if(hiHaCor(cDim,ii,veu)){
                        lote.setColor(S.cColor(F.lastCor.cor));
                        if(veu%2==0){ //veu de dalt del pentagrama
                            lote.draw(Recursos.cir, scroll+dEAc*ii+dAlt*0.8f+(dNota+6)*(H.lastSolapades),
H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, -10f, 10f);
                        }else{ //veu de baix (separo per l'ajust per solapament)
                            lote.draw(Recursos.cir, scroll+dEAc*ii+dAlt*0.8f+(dNota-5)*(H.ajustSolapades(ii, veu-
1, false)), H.ySolFa(veu)+H.lastNotePos+dEspai*0.5f*(1+H.sobreLinia(esquemaEnNotes.get(ii)[veu]))-3.5f, -10f,
10f);
                        }
                    }
                }
            }

            /**- - - DIBUIX NOTES ESBÓS - - -*/

            H.resetStack=true;
            if(Pant.esbos.get(ii).dom){ //calculo quantes notes hauré de pintar (per limitar el "for")
                nAc = 5;
            }else if(Pant.esbos.get(ii).sept){
                nAc = 4;
```

```
                }else{
                    nAc = 3;
                }
                if(ii==aAct){ //marco de color l'acord actual.
                    lote.setColor(S.actual);
                    Recursos.eXifrat.setColor(S.actual); //és la font de posar "M" a les sèptimes majors. La faig
servir més avall, però així ja tinc el color posat.
                }else{
                    lote.setColor(S.pentagrama);
                    Recursos.eXifrat.setColor(S.pentagrama);
                    if(ii<cInv.size){
                        if(tincPle.get(ii) && cInv.get(ii)!='n'){ //pinto d'alerta l'acord de l'esbós si la inversió
està diferent a la realització (i marco si hi ha error)
                            lote.setColor(S.cColor(cInv.get(ii)));
                            Recursos.eXifrat.setColor(S.cColor(cInv.get(ii)));
                        }
                    }
                }
                for(byte j=0;j<nAc;j++){
                    if(j==Pant.esbos.get(ii).inv){
                        lote.setColor(lote.getColor().r,lote.getColor().g,lote.getColor().b,0.2f); //marco la nota
que havia posat al baix (mateix color, quasi transparent).
                        lote.draw(Recursos.cir, scrollEsb+dEAcEsb*ii-dNotaEsb/2,
yEsb+H.nStackPos(F.keep(H.getChordNote(Pant.esbos.get(ii), 1+2*j)),dEspaiEsb), dNotaEsb, (dNotaEsb+dEspaiEsb)/2f);
                        lote.setColor(lote.getColor().r,lote.getColor().g,lote.getColor().b,1f);
                        lote.draw(Recursos.nota, scrollEsb+dEAcEsb*ii-dNotaEsb/2, yEsb+H.stackH, dNotaEsb, dNotaEsb);
//aquests valors de offset y han estat una qüestió empírica. TODO Treure o ajustar si cal o si en realitat no fan
res.
                    }else{
                        lote.draw(Recursos.nota, scrollEsb+dEAcEsb*ii-dNotaEsb/2,
yEsb+H.nStackPos(F.keep(H.getChordNote(Pant.esbos.get(ii), 1+2*j)),dEspaiEsb)+1f, dNotaEsb,
(dNotaEsb+dEspaiEsb)/2f);
                    }

                    if(H.stackH<(-dEspaiEsb)){ //dibuix línies addicionals. Primera inferior (do3)
                        lote.draw(Recursos.rect, scrollEsb+dEAcEsb*ii-dNotaEsb*0.75f,yEsb-
dEspaiEsb,dNotaEsb*1.5f,2f);
                    }else if(H.stackH>dEspaiEsb*5f){ //segona superior (do5)
                        lote.draw(Recursos.rect, scrollEsb+dEAcEsb*ii-dNotaEsb*0.75f, yEsb+dEspaiEsb*6f,
dNotaEsb*1.5f, 2f);
                    }else if(H.stackH>dEspaiEsb*4f){ //primera superior (la4)
                        lote.draw(Recursos.rect, scrollEsb+dEAcEsb*ii-dNotaEsb*0.75f, yEsb+dEspaiEsb*5f,
dNotaEsb*1.5f, 2f);
                    }

                    /**- - - Alteracions accidentals - - -*/
                    if(H.accidental(F.lastNote)){ //pinto l'alteració de la nota en cas que calgui l'accidental
                        switch(H.getAltNum(F.lastNote)){
                        case 1:
                            lote.draw(Recursos.sharp, scrollEsb+dEAcEsb*ii-dAltEsb, yEsb+H.stackH-dNotaEsb*0.38f,
dNotaEsb*0.5f*1.7f, dNotaEsb*1.7f);
                            break;
                        case 0:
                            lote.draw(Recursos.natural, scrollEsb+dEAcEsb*ii-dAltEsb, yEsb+H.stackH-dNotaEsb*0.38f,
dNotaEsb*0.5f*1.7f, dNotaEsb*1.7f);
                            break;
                        case -1:
                            lote.draw(Recursos.bemoll, scrollEsb+dEAcEsb*ii-dAltEsb-15f*(j%2)+5f, yEsb+H.stackH-
dNotaEsb*0.05f, dNotaEsb*0.5f*1.5f, dNotaEsb*1.5f); //alterno amb el %2 les alteracions que es molesten entre elles
                            break;
                        default:
                            System.out.println("Accidental sense dibuix!!!
"+H.notes[F.lastNote.nom]+H.getAlt(F.lastNote));
                        }
                    }
                }

                /**- - Indicacions de l'esbós - -*/
                if(Pant.esbos.get(ii).sept){ //sèptima major (el color ve de dalt)

    if(F.keep(H.ascendent(H.getInterval(H.getChordNote(Pant.esbos.get(ii),1),H.getChordNote(Pant.esbos.get(ii),7)))
).nom==6 && F.lastNote.so==11){ //si la sèptima és major...

    Recursos.eXifrat.draw(lote,F.keep("M"),scrollEsb+dEAcEsb*ii+F.align("eXifrat"),yEsb+H.stackH+80f*r1e);
                }
                }

                /** - - - - TEXT GRAU I XIFRAT - - - -*/

                if(ii==aAct){ //escric els graus
```

```
                    Recursos.font.setColor(S.actual);
                    Recursos.xifrat.setColor(S.actual);
                }else{
                    Recursos.font.setColor(S.graus);
                    Recursos.xifrat.setColor(S.graus);
                }
                Recursos.font.draw(lote,F.keep(H.roman[Pant.esbos.get(ii).grau]),
scroll+dEAc*ii+F.gAlign(Pant.esbos.get(ii)), yFa-75f);
                Recursos.xifrat.draw(lote,F.keep(H.textGrau(Pant.esbos.get(ii))),
scroll+dEAc*ii+F.gAlign(Pant.esbos.get(ii))-F.align('r',H.roman[Pant.esbos.get(ii).grau]),yFa-
75f+F.vAlign("xifrat",'d')-F.vAlign('d',H.roman[Pant.esbos.get(ii).grau]));

                if(ii<tincPle.size){ //vigilo no sortir de la correcció que tinc calculada
                    if(tincPle.get(ii)){ //pels acords que ja tenen totes les notes escric la inversió real (en lloc
de la decidida a l'esbós)

                        if(ii<inversions.size){
                            if(ii<cInv.size){
                                if(cInv.get(ii)!='n'){ //si passa alguna cosa fora el normal, marco amb color
                                    Recursos.xifrat.setColor(S.cColor(cInv.get(ii)));
                                    if(inversions.get(ii)==0 && cInv.get(ii)=='e'){ //si era un error i no hi ha
xifrat (E.F. sense 7), escric "!!" per poder pintar alguna cosa de vermell xD
                                        Recursos.xifrat.draw(lote,F.keep("!!"), scroll+dEAc*ii-
F.gAlign(Pant.esbos.get(ii))+10f, yFa-75f+F.vAlign("xifrat"));
                                    }
                                }
                            }
                            Recursos.xifrat.draw(lote,F.keep(H.xifrat(inversions.get(ii), false)), scroll+dEAc*ii-
F.gAlign(Pant.esbos.get(ii))+10f, yFa-75f+F.vAlign("xifrat"));
                            Recursos.xifrat.draw(lote,F.keep(H.xifrat(inversions.get(ii), true)),  scroll+dEAc*ii-
F.gAlign(Pant.esbos.get(ii))+10f, yFa-75f+F.vAlign("xifrat")-F.vAlign('d', H.roman[Pant.esbos.get(ii).grau]));
                        }

                        //pinto també una "i" quan queden incomplets, en color de correcció.
                        Recursos.xifrat.setColor(S.cColor(cComplet.get(ii)));
                        if(!complet.get(ii)){
                            Recursos.xifrat.draw(lote,F.keep("i"), scroll+dEAc*ii+F.gAlign(Pant.esbos.get(ii))-20f,
yFa-75f+F.vAlign("xifrat")-F.vAlign('d', H.roman[Pant.esbos.get(ii).grau]));
                        }

                        //i pinto dos punts per marcar quan passen coses amb la duplicació.
                        if(ii<cDuplicades.size){ //vigilo els marges
                            if(cDuplicades.get(ii)=='e' || cDuplicades.get(ii)=='a'){ //si tinc correcció rellevant
marco
                                lote.setColor(S.cColor(cDuplicades.get(ii)));
                                lote.draw(Recursos.cir, scroll+dEAc*ii+F.gAlign(Pant.esbos.get(ii))-20f, yFa-75f,
6f, 6f);
                                lote.draw(Recursos.cir, scroll+dEAc*ii+F.gAlign(Pant.esbos.get(ii))-20f, yFa-75f-10f,
6f, 6f);
                            }
                        }

                }else{ //si no està l'acord ple, pinto el xifrat, però en gris.
                        Recursos.xifrat.setColor(S.irrellevant);
                        Recursos.xifrat.draw(lote,F.keep(H.xifrat(Pant.esbos.get(ii), false)), scroll+dEAc*ii-
F.gAlign(Pant.esbos.get(ii))+10f, yFa-75f+F.vAlign("xifrat"));
                        Recursos.xifrat.draw(lote,F.keep(H.xifrat(Pant.esbos.get(ii), true)), scroll+dEAc*ii-
F.gAlign(Pant.esbos.get(ii))+10f, yFa-75f+F.vAlign("xifrat")-F.vAlign('d', H.roman[Pant.esbos.get(ii).grau]));
                    }
                }

                ii++;
            }

            /**- - - Cadències - - -*/
            ii=0;
            for(Cadencia cad: Pant.cadencies){
                lote.setColor(S.cadencies);
                Recursos.font.setColor(S.cadencies);

                if(cCad.size==Pant.cadencies.size){ //si tinc correcció de les cadències, aplico els colors de la
correcció.
                    if(cCad.get(ii)!='n'){
                        lote.setColor(S.cColor(cCad.get(ii)));
                        Recursos.font.setColor(S.cColor(cCad.get(ii)));
                    }
                }

                lote.draw(Recursos.rect, scroll+dEAc*(cad.n)-40f, yFa-dCad, dEAc*(cad.w/2f), 6f);
                lote.draw(Recursos.rect, scroll+dEAc*(cad.n+cad.w/2f)+40f, yFa-dCad, dEAc*(cad.w/2f), 6f);
```

```
                lote.draw(Recursos.rect, scroll+dEAc*(cad.n)-40f, yFa-dCad+6f, 6f, 12f);
                lote.draw(Recursos.rect, scroll+dEAc*(cad.n+cad.w)+40f, yFa-dCad+6f, -6f, 12f);
                if(cad.cp){
                    Recursos.font.draw(lote,F.keep("CP"),scroll+dEAc*(cad.n+cad.w/2f)+F.align(),yFa-dCad+F.vAlign());
                }else{
                    Recursos.font.draw(lote,F.keep("CT"),scroll+dEAc*(cad.n+cad.w/2f)+F.align(),yFa-dCad+F.vAlign());
                }
                ii++;
            }

            if(escoltant){
                Recursos.xifrat.setColor(S.b_normal);
                Recursos.xifrat.draw(lote,F.keep("- - Reproduint... (toca la pantalla per aturar) - -
"),Pant.realW*Pant.ratio/2+F.align("xifrat"),Pant.screenH*0.2f+25+F.vAlign("xifrat"));
            }

            /**- - Cortines - -*/
            lote.setColor(S.fons); //pinto les cortines que tapen l'esquema
            lote.draw(Recursos.rect, 0, 0, xSolFa, Pant.screenH); //cortines de la realització
            lote.draw(Recursos.rect, rLim, 0, rxSolFa, Pant.screenH);
            lote.draw(Recursos.rect, 0, 0, xEsb, 150); //cortines de l'esbós
            lote.draw(Recursos.rect, rLimEsb, 0, rxEsb, 150);

            /**- - - - - BOTONS - - - - -*/
            //pinto botons de les veus
            ii=0;
            for(BotQ b: bVeus){
                if(ii==vAct){
                    lote.setColor(S.b_marc_ressaltat);
                }else{
                    lote.setColor(S.b_marc);
                }
                if(escoltant){
                    lote.setColor(S.greyScale(lote.getColor()));
                }
                lote.draw(Recursos.cir,b.x-b.w/2f,b.y-b.h/2f,b.w,b.h);
                if(esquema.get(aAct).satb[ii]!=0){
                    lote.setColor(S.b_premut);
                }else{
                    if(ii==vAct){
                        lote.setColor(S.b_ressaltat);
                    }else{
                        lote.setColor(S.b_normal);
                    }
                }
                if(escoltant){
                    lote.setColor(S.greyScale(lote.getColor()));
                }
                lote.draw(Recursos.cir,b.x-b.w/2f+8,b.y-b.h/2f+8,b.w-16,b.h-16);
                Recursos.font.setColor(S.b_text);
                Recursos.font.draw(lote,F.keep(H.rVeus[ii]),b.x+F.align(),b.y+F.vAlign());
                ii++;
            }
            ii=0;
            for(BotQ b: bNotes){
                dAux = esquema.get(aAct);
                if(dAux.satb[vAct]==b.grau){
                    lote.setColor(S.b_marc_ressaltat);
                }else{
                    lote.setColor(S.b_marc);
                }
                if(escoltant){
                    lote.setColor(S.greyScale(lote.getColor()));
                }
                lote.draw(Recursos.cir,b.x-b.w/2f,b.y-b.h/2f,b.w,b.h);

                if(b.grau==9 && !puc9a || b.grau==7 && !puc7a){
                    lote.setColor(S.b_inactiu);
                    Recursos.font.setColor(S.b_text_inactiu);
                    Recursos.bNotes.setColor(S.b_text_inactiu);
                }else{ //TODO canviar les alteracions en text per dibuixos
                    if(dAux.satb[0]==b.grau || dAux.satb[1]==b.grau || dAux.satb[2]==b.grau || dAux.satb[3]==b.grau){
//si la nota és en alguna veu
                        lote.setColor(S.b_premut);
                    }else{
                        lote.setColor(S.b_normal);
                    }
                    Recursos.font.setColor(S.b_text);
                    Recursos.bNotes.setColor(S.b_text);
                }
```

```
                if(escoltant){
                    lote.setColor(S.greyScale(lote.getColor()));
                }
                lote.draw(Recursos.cir,b.x-b.w/2f+8,b.y-b.h/2f+8,b.w-16,b.h-16); //omplo el botó

                if(altOp/10==b.grau){ //marco quan hi ha alteracions opcionals
                    if(dAux.satb[vAct]==b.grau){
                        lote.setColor(S.b_marc_ressaltat);
                    }else{
                        lote.setColor(S.b_marc);
                    }
                    if(escoltant){
                        lote.setColor(S.greyScale(lote.getColor()));
                    }
                    lote.draw(Recursos.cir,b.x-b.w/2f+10,b.y-b.h/2f+10,b.w-20,b.h-20);
                    if(dAux.satb[0]==b.grau || dAux.satb[1]==b.grau || dAux.satb[2]==b.grau || dAux.satb[3]==b.grau){
//si la nota és en alguna veu
                        lote.setColor(S.b_premut);
                    }else{
                        lote.setColor(S.b_normal);
                    }
                    if(escoltant){
                        lote.setColor(S.greyScale(lote.getColor()));
                    }
                    lote.draw(Recursos.cir,b.x-b.w/2f+14,b.y-b.h/2f+14,b.w-28,b.h-28);
                }
                if(b.grau==9 && !puc9a || b.grau==7 && !puc7a){
                    Recursos.font.draw(lote,F.keep(Integer.toString(b.grau)),b.x+F.align(),b.y+F.vAlign());
                }else{
                    Recursos.bNotes.draw(lote,F.keep(H.notes[F.keep(H.getChordNote(Pant.esbos.get(aAct),
b.grau)).nom]+""+H.getAlt(F.lastNote)),b.x+F.align("bNotes"),b.y+F.vAlign("bNotes"));
                }

                ii++;
            }
            //botons de notes d'alteracions opcionals
            if(A.tAltOp>A.tHold){
                lote.setColor(S.fons);
                lote.draw(Recursos.cir, bAlt1.x-bAlt1.w/2f-8, bAlt1.y-bAlt1.h/2f-8, bAlt1.w+16, bAlt1.h+16);
                lote.draw(Recursos.rect, bAlt1.x, bAlt2.y-bAlt2.h/2f-8, rLim-bAlt1.x, bAlt2.h+16);
                lote.setColor(S.b_marc);
                lote.draw(Recursos.cir, bAlt1.x-bAlt1.w/2f, bAlt1.y-bAlt1.h/2f, bAlt1.w, bAlt1.h);
                lote.draw(Recursos.cir, bAlt2.x-bAlt2.w/2f, bAlt2.y-bAlt2.h/2f, bAlt2.w, bAlt2.h);
                if(altOp%10==9 && Pant.esbos.get(aAct).alt9
                || altOp%10==7 && Pant.esbos.get(aAct).alt7
                || altOp%10==6 && Pant.esbos.get(aAct).alt6){ //pinto premut el botó de l'alteració opcional que hi
ha posada
                    lote.setColor(S.b_normal);
                    lote.draw(Recursos.cir, bAlt1.x-bAlt1.w/2f+8, bAlt1.y-bAlt1.h/2f+8, bAlt1.w-16, bAlt1.h-16);
                    lote.setColor(S.b_premut);
                    lote.draw(Recursos.cir, bAlt2.x-bAlt2.w/2f+8, bAlt2.y-bAlt2.h/2f+8, bAlt2.w-16, bAlt2.h-16);
                }else{
                    lote.setColor(S.b_premut);
                    lote.draw(Recursos.cir, bAlt1.x-bAlt1.w/2f+8, bAlt1.y-bAlt1.h/2f+8, bAlt1.w-16, bAlt1.h-16);
                    lote.setColor(S.b_normal);
                    lote.draw(Recursos.cir, bAlt2.x-bAlt2.w/2f+8, bAlt2.y-bAlt2.h/2f+8, bAlt2.w-16, bAlt2.h-16);
                }
                Recursos.bNotes.draw(lote,F.keep(H.notes[F.keep(H.getChordNote(H.altChord(Pant.esbos.get(aAct),
altOp%10, false), altOp/10)).nom]+""+H.getAlt(F.lastNote)),bAlt1.x+F.align("bNotes"),bAlt1.y+F.vAlign("bNotes"));
                Recursos.bNotes.draw(lote,F.keep(H.notes[F.keep(H.getChordNote(H.altChord(Pant.esbos.get(aAct),
altOp%10, true), altOp/10)).nom]+""+H.getAlt(F.lastNote)),bAlt2.x+F.align("bNotes"),bAlt2.y+F.vAlign("bNotes"));
            }

            //botó d'esborrar
            lote.setColor(S.error);
            if(escoltant){
                lote.setColor(S.greyScale(lote.getColor()));
            }
            lote.draw(Recursos.cir, bEsborra.x-bEsborra.w/2f, bEsborra.y-bEsborra.h/2f, bEsborra.w, bEsborra.h);
            lote.setColor(S.fons);
            lote.draw(Recursos.cir, bEsborra.x-bEsborra.w/2f+8, bEsborra.y-bEsborra.h/2f+8, bEsborra.w-16,
bEsborra.h-16);
            Recursos.bNotes.setColor(S.error);
            if(escoltant){
                Recursos.bNotes.setColor(S.greyScale(Recursos.bNotes.getColor()));
            }
            Recursos.bNotes.draw(lote,F.keep("Esb"),bEsborra.x+F.align("bNotes"),bEsborra.y+F.vAlign("bNotes"));

            //botons navegació
            lote.setColor(S.b_marc);
```

```
            lote.draw(Recursos.arrow, bEsq.x-bEsq.w/2f, bEsq.y-bEsq.h/2f, bEsq.w, bEsq.h);
            lote.draw(Recursos.arrow, bDr.x+bDr.w/2f, bDr.y-bDr.h/2f, -bDr.w, bDr.h);

        }else{ //si mostraMenu

            //cercle d'errors
            lote.setColor(S.error);
            lote.draw(Recursos.cir, bErrors.x-bErrors.w/2f, bErrors.y-bErrors.h/2f, bErrors.w, bErrors.h);
            lote.setColor(S.fons);
            lote.draw(Recursos.cir, bErrors.x-bErrors.w*0.8f/2f, bErrors.y-bErrors.h*0.8f/2f, bErrors.w*0.8f,
bErrors.h*0.8f);
            if(mostraErrors){
                lote.setColor(S.newAlpha(S.error, 0.1f));
            }
            lote.draw(Recursos.cir, bErrors.x-bErrors.w*0.8f/2f, bErrors.y-bErrors.h*0.8f/2f, bErrors.w*0.8f,
bErrors.h*0.8f);
            Recursos.font.setColor(S.error);
            Recursos.font.draw(lote,F.keep(Integer.toString(nErrors)),bErrors.x+F.align(),bErrors.y+F.vAlign());


            //cercle d'alertes
            lote.setColor(S.alerta);
            lote.draw(Recursos.cir, bAlertes.x-bAlertes.w/2f, bAlertes.y-bAlertes.h/2f, bAlertes.w, bAlertes.h);
            lote.setColor(S.fons);
            lote.draw(Recursos.cir, bAlertes.x-bAlertes.w*0.8f/2f, bAlertes.y-bAlertes.h*0.8f/2f, bAlertes.w*0.8f,
bAlertes.h*0.8f);
            if(mostraAlertes){
                lote.setColor(S.newAlpha(S.alerta, 0.1f));
            }
            lote.draw(Recursos.cir, bAlertes.x-bAlertes.w*0.8f/2f, bAlertes.y-bAlertes.h*0.8f/2f, bAlertes.w*0.8f,
bAlertes.h*0.8f);
            Recursos.font.setColor(S.alerta);
            Recursos.font.draw(lote,F.keep(Integer.toString(nAlertes)),bAlertes.x+F.align(),bAlertes.y+F.vAlign());

            if(mostraErrors || mostraAlertes){ //registre d'errors i registre d'alertes
                if(mostraErrors){
                    lote.setColor(S.error);
                }else{
                    lote.setColor(S.alerta);
                }
                //cantonades cercle
                lote.draw(Recursos.cir, xLogs-bErrors.w/2f, bErrors.y-bErrors.h/2f, bErrors.w, bErrors.h); //TODO
potser canviar per un rectangle arrodonit directament
                lote.draw(Recursos.cir, Pant.realW*Pant.ratio-bErrors.w/2-bErrors.w*0.75f, bErrors.y-bErrors.h/2f,
bErrors.w, bErrors.h);
                lote.draw(Recursos.cir, xLogs-bErrors.w/2, yLogs-bErrors.h/2f, bErrors.w, bErrors.h);
                lote.draw(Recursos.cir, Pant.realW*Pant.ratio-bErrors.w/2-bErrors.w*0.75f, yLogs-bErrors.h/2f,
bErrors.w, bErrors.h);
                //rectangles
                lote.draw(Recursos.rect, xLogs, yLogs-bErrors.h/2f,(Pant.realW*Pant.ratio-bErrors.w*0.75f)-(xLogs),
bErrors.y-yLogs+bErrors.h);
                lote.draw(Recursos.rect, xLogs-bErrors.w/2f, yLogs,(Pant.realW*Pant.ratio-bErrors.w*0.75f)-
(xLogs)+bErrors.w, bErrors.y-yLogs);
                //botons anterior i següent
                lote.draw(Recursos.cir, bAnt.x-bAnt.w/2f-bAnt.h/2f,bAnt.y-bAnt.h/2f, bAnt.h, bAnt.h);
                lote.draw(Recursos.cir, bAnt.x+bAnt.w/2f-bAnt.h/2f,bAnt.y-bAnt.h/2f, bAnt.h, bAnt.h);
                lote.draw(Recursos.rect, bAnt.x-bAnt.w/2f,bAnt.y-bAnt.h/2f, bAnt.w, bAnt.h);
                lote.draw(Recursos.cir, bSeg.x-bSeg.w/2f-bSeg.h/2f,bSeg.y-bSeg.h/2f, bSeg.h, bSeg.h);
                lote.draw(Recursos.cir, bSeg.x+bSeg.w/2f-bSeg.h/2f,bSeg.y-bSeg.h/2f, bSeg.h, bSeg.h);
                lote.draw(Recursos.rect, bSeg.x-bSeg.w/2f,bSeg.y-bSeg.h/2f, bSeg.w, bSeg.h);

                //(part interior)
                lote.setColor(S.fons);
                //cantonades cercle
                lote.draw(Recursos.cir, xLogs-bErrors.w*0.8f/2f, bErrors.y-bErrors.h*0.8f/2f, bErrors.w*0.8f,
bErrors.h*0.8f);
                lote.draw(Recursos.cir, Pant.realW*Pant.ratio-bErrors.w*0.8f/2-bErrors.w*0.75f, bErrors.y-
bErrors.h*0.8f/2f, bErrors.w*0.8f, bErrors.h*0.8f);
                lote.draw(Recursos.cir, xLogs-bErrors.w*0.8f/2, yLogs-bErrors.h*0.8f/2f, bErrors.w*0.8f,
bErrors.h*0.8f);
                lote.draw(Recursos.cir, Pant.realW*Pant.ratio-bErrors.w*0.8f/2-bErrors.w*0.75f, yLogs-
bErrors.h*0.8f/2f, bErrors.w*0.8f, bErrors.h*0.8f);
                //rectangles
                lote.draw(Recursos.rect, xLogs, yLogs-bErrors.h*0.8f/2f,(Pant.realW*Pant.ratio-bErrors.w*0.75f)-
(xLogs), bErrors.y-yLogs+bErrors.h*0.8f);
                lote.draw(Recursos.rect, xLogs-bErrors.w*0.8f/2f, yLogs,(Pant.realW*Pant.ratio-bErrors.w*0.75f)-
(xLogs)+bErrors.w*0.8f, bErrors.y-yLogs);
                //botons anterior i següent
                //TODO enfosquir quan no els puc polsar (compte que amb transparències no puc perquè se solapen les
formes)
```

```
                lote.draw(Recursos.cir, bAnt.x-bAnt.w/2f-bAnt.h*0.75f/2f,bAnt.y-bAnt.h*0.75f/2f, bAnt.h*0.75f,
bAnt.h*0.75f);
                lote.draw(Recursos.cir, bAnt.x+bAnt.w/2f-bAnt.h*0.75f/2f,bAnt.y-bAnt.h*0.75f/2f, bAnt.h*0.75f,
bAnt.h*0.75f);
                lote.draw(Recursos.rect, bAnt.x-bAnt.w/2f,bAnt.y-bAnt.h*0.75f/2f, bAnt.w, bAnt.h*0.75f);
                lote.draw(Recursos.cir, bSeg.x-bSeg.w/2f-bSeg.h*0.75f/2f,bSeg.y-bSeg.h*0.75f/2f, bSeg.h*0.75f,
bSeg.h*0.75f);
                lote.draw(Recursos.cir, bSeg.x+bSeg.w/2f-bSeg.h*0.75f/2f,bSeg.y-bSeg.h*0.75f/2f, bSeg.h*0.75f,
bSeg.h*0.75f);
                lote.draw(Recursos.rect, bSeg.x-bSeg.w/2f,bSeg.y-bSeg.h*0.75f/2f, bSeg.w, bSeg.h*0.75f);
                if(mostraErrors){
                    Recursos.bNotes.setColor(S.error);
                }else{
                    Recursos.bNotes.setColor(S.alerta);
                }
                Recursos.bNotes.draw(lote, F.keep(bAnt.text), bAnt.x+F.align("bNotes"), bAnt.y+F.vAlign("bNotes"));
                Recursos.bNotes.draw(lote, F.keep(bSeg.text), bSeg.x+F.align("bNotes"), bSeg.y+F.vAlign("bNotes"));

                if(mostraErrors){
                    Recursos.bNotes.setColor(S.error);
                    Recursos.bNotes.draw(lote, F.keep("Registre d'errors
"+F.ifText(logE.size>0,"("+(errorAct+1)+"/"+logE.size+")")), xLogs-bErrors.w*0.10f, bErrors.y+F.vAlign("bNotes"));
                    if(nErrors>0){
                        if(errorAct<logE.size){ //asseguro els límits
                            Recursos.xifrat.setColor(S.error);
                            Recursos.xifrat.draw(lote, F.keep("acord nº: "+(logE.get(errorAct).nAc+1)),
Pant.realW*Pant.ratio-bErrors.w*0.65f+F.align("xifrat",'r'), bErrors.y+F.vAlign("xifrat"));
                            Recursos.iQuant.setColor(S.error);
                            Recursos.iQuant.draw(lote, F.keep(N.ambContext(logE.get(errorAct).ref)), xLogs-
bErrors.w*0.10f, bErrors.y-bErrors.h*0.5f, (Pant.realW*Pant.ratio-bErrors.w*0.75f)-(xLogs), Align.left, true);
                        }
                    }else{
                        Recursos.iQuant.setColor(S.error);
                        Recursos.iQuant.draw(lote, F.keep("No s'han detectat errors a l'esquema."), xLogs-
bErrors.w*0.10f, bErrors.y-bErrors.h*0.5f, (Pant.realW*Pant.ratio-bErrors.w*0.75f)-(xLogs), Align.left, true);
                    }
                }else{ //mostraAlertes
                    Recursos.bNotes.setColor(S.alerta);
                    Recursos.bNotes.draw(lote, F.keep("Registre d'alertes
"+F.ifText(logA.size>0,"("+(alertaAct+1)+"/"+logA.size+")")), xLogs-bErrors.w*0.10f, bErrors.y+F.vAlign("bNotes"));
                    if(nAlertes>0){
                        if(alertaAct<logA.size){
                            Recursos.xifrat.setColor(S.alerta);
                            Recursos.xifrat.draw(lote, F.keep("acord nº: "+(logA.get(alertaAct).nAc+1)),
Pant.realW*Pant.ratio-bErrors.w*0.65f+F.align("xifrat",'r'), bErrors.y+F.vAlign("xifrat"));
                            Recursos.iQuant.setColor(S.alerta);
                            Recursos.iQuant.draw(lote, F.keep(N.ambContext(logA.get(alertaAct).ref)), xLogs-
bErrors.w*0.10f, bErrors.y-bErrors.h*0.5f, (Pant.realW*Pant.ratio-bErrors.w*0.75f)-(xLogs), Align.left, true);
                        }else{
                            Recursos.iQuant.setColor(S.alerta);
                            Recursos.iQuant.draw(lote, F.keep("No hi ha descripcions per les alertes generades."),
xLogs-bErrors.w*0.10f, bErrors.y-bErrors.h*0.5f, (Pant.realW*Pant.ratio-bErrors.w*0.75f)-(xLogs), Align.left,
true);
                        }
                    }else{
                        Recursos.iQuant.setColor(S.alerta);
                        Recursos.iQuant.draw(lote, F.keep("No s'ha generat cap alerta."), xLogs-bErrors.w*0.10f,
bErrors.y-bErrors.h*0.5f, (Pant.realW*Pant.ratio-bErrors.w*0.75f)-(xLogs), Align.left, true);
                    }
                }
            }else{ //el que hi hagi sota dels registres quan no estan oberts.

                //Escoltar l'esquema
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, bEscolta.x-bEscolta.w/2f, bEscolta.y-bEscolta.h/2f, bEscolta.w, bEscolta.h);
                if(pucEscoltar){
                    lote.setColor(S.b_premut);
                }else{
                    lote.setColor(S.b_normal);
                }
                lote.draw(Recursos.rect, bEscolta.x-bEscolta.w/2f+8f, bEscolta.y-bEscolta.h/2f+8f, bEscolta.w-16f,
bEscolta.h-16f);
                Recursos.font.setColor(S.b_text);
                if(!pucEscoltar){ //no estan carregats els àudios
                    if(Recursos.carregantSons){ //però s'estan carregant
                        Recursos.bMenu.setColor(S.b_text);
                        Recursos.bMenu.draw(lote, F.keep("Carregant...
"+Integer.toString(Math.round(Recursos.manager.getProgress()*100))),
bEscolta.x+F.align("bMenu"),bEscolta.y+F.vAlign("bMenu"));
                    }else{ //ni s'han carregat ni s'estan carregant
```

```
                    Recursos.font.draw(lote, F.keep(bEscolta.text), bEscolta.x+F.align(),bEscolta.y+F.vAlign());
                }
            }else{ //està preparat per reproduir
                Recursos.font.draw(lote, F.keep("Reprodueix"), bEscolta.x+F.align(),bEscolta.y+F.vAlign());
            }

            //Buidar realització
            lote.setColor(S.error);
            lote.draw(Recursos.rect, bBuida.x-bBuida.w/2f, bBuida.y-bBuida.h/2f, bBuida.w, bBuida.h);
            lote.setColor(S.fons);
            lote.draw(Recursos.rect, bBuida.x-bBuida.w/2f+8f, bBuida.y-bBuida.h/2f+8f, bBuida.w-16f, bBuida.h-
16f);
            if(vullBuidar){ //s'ha polsat el botó de buidar la realització i estic demanant confirmació
                Recursos.xifrat.setColor(S.error);
                Recursos.xifrat.draw(lote, "Segur que vols esborrar la realització? No es podrà desfer.", xLogs,
Pant.screenH*0.5f, (Pant.realW*Pant.ratio-bErrors.w*0.75f-xLogs), Align.center, true);

                Recursos.font.setColor(S.error);
                Recursos.font.draw(lote, F.keep("Confirmar"), bBuida.x+F.align(),bBuida.y+F.vAlign());
                lote.setColor(S.error);
                lote.draw(Recursos.rect, bCancel.x-bCancel.w/2f, bCancel.y-bCancel.h/2f, bCancel.w, bCancel.h);
                lote.setColor(S.fons);
                lote.draw(Recursos.rect, bCancel.x-bCancel.w/2f+8f, bCancel.y-bCancel.h/2f+8f, bCancel.w-16f,
bCancel.h-16f);
                Recursos.font.draw(lote, F.keep(bCancel.text), bCancel.x+F.align(),bCancel.y+F.vAlign());
            }else{
                Recursos.bNotes.setColor(S.error);
                Recursos.bNotes.draw(lote, F.keep(bBuida.text),
bBuida.x+F.align("bNotes"),bBuida.y+F.vAlign("bNotes"));
            }

            //editar esbós
            lote.setColor(S.b_marc);
            lote.draw(Recursos.rect, bEsbos.x-bEsbos.w/2f, bEsbos.y-bEsbos.h/2f, bEsbos.w, bEsbos.h);
            lote.setColor(S.b_normal);
            lote.draw(Recursos.rect, bEsbos.x-bEsbos.w/2f+8f, bEsbos.y-bEsbos.h/2f+8f, bEsbos.w-16f, bEsbos.h-
16f);
            Recursos.font.setColor(S.b_text);
            if(vullTornar){ //s'ha polsat el botó d'Editar esbós i estic demanant confirmació
                Recursos.xifrat.setColor(S.alerta);
                Recursos.xifrat.draw(lote, "Esborrar parts de l'esbós n'esborrarà la realització corresponent.
Segur que vols continuar?", xLogs, Pant.screenH*0.5f, (Pant.realW*Pant.ratio-bErrors.w*0.75f-xLogs), Align.center,
true);

                Recursos.font.draw(lote, F.keep(bEsbos.text), bEsbos.x+F.align(),bEsbos.y+F.vAlign());
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, bCancel.x-bCancel.w/2f, bCancel.y-bCancel.h/2f, bCancel.w, bCancel.h);
                lote.setColor(S.b_normal);
                lote.draw(Recursos.rect, bCancel.x-bCancel.w/2f+8f, bCancel.y-bCancel.h/2f+8f, bCancel.w-16f,
bCancel.h-16f);
                Recursos.font.draw(lote, F.keep(bCancel.text), bCancel.x+F.align(),bCancel.y+F.vAlign());
            }else{
                Recursos.font.draw(lote, F.keep("Editar Esbós"), bEsbos.x+F.align(),bEsbos.y+F.vAlign());
            }
        }
    }
    if(A.tMenu>A.tHold){ //pantalla de les referències

        lote.setColor(S.newAlpha(S.fons,0.97f)); //requadre que tapa mig transparent
        lote.draw(Recursos.rect, 0, 0, Pant.realW*Pant.ratio, Pant.screenH);

        Recursos.bNotes.setColor(S.graus);
        Recursos.bNotes.draw(lote,"Ingredients de l'enunciat:",50,Pant.screenH*0.9f);

        if(PIng.bTipusAcord.size+PIng.bTipusCadencia.size+PIng.bTipusRes.size>0){
            Recursos.font.setColor(S.graus);
            Recursos.iQuant.setColor(S.graus);
            Recursos.iXifrat.setColor(S.graus);

            for(BotIA b: PIng.bTipusAcord){
                lote.setColor(S.ingredients);

                lote.draw(Recursos.rect, b.x-b.h*0.2f, b.y-b.h*0.2f, b.w+b.h*0.4f, b.h);
                //quantitat (p. ex. 2x)
                if(b.iAc.n>1 && !b.iAc.tcp){
                    Recursos.iQuant.draw(lote, b.iAc.n+"x", b.x, b.y+F.vAlign("iQuant", 'd'));
                }
                //grau
                Recursos.font.draw(lote, F.keep(H.roman[b.iAc.acord.grau]), b.x+b.d[0], b.y+F.vAlign("font",
'd'));
```

```
                    //xifrat
                    Recursos.iXifrat.draw(lote, F.keep(F.ifText(b.iAc.triada && b.iAc.acord.inv<1, "5",
H.xifrat(b.iAc.acord, false))), b.x+b.d[1],
b.y+F.vAlign("font",'d',H.roman[b.iAc.acord.grau])+F.vAlign("iXifrat")*0.7f);
                    Recursos.iXifrat.draw(lote, F.keep(H.xifrat(b.iAc.acord, true)), b.x+b.d[1],
b.y+F.vAlign("iXifrat")*1.3f);
                    //carro
                    Recursos.iXifrat.draw(lote, F.keep(b.carro), b.x+b.d[2],
b.y+F.vAlign("font",'c',H.roman[b.iAc.acord.grau])+F.vAlign("iXifrat",'c'));
                    Recursos.iQuant.draw(lote, F.keep(b.carro2), b.x+b.d[3], b.y+F.vAlign("iQuant",'d'));
                }

                for(BotIC b: PIng.bTipusCadencia){
                    lote.setColor(S.ingredients);

                    lote.draw(Recursos.rect, b.x-b.h*0.2f, b.y-b.h*0.2f, b.w+b.h*0.4f, b.h);
                    //quantitat (p. ex. 2x)
                    if(b.iCad.n>1 && !b.iCad.tcp){
                        Recursos.iQuant.draw(lote, b.iCad.n+"x", b.x, b.y+F.vAlign("iQuant", 'd'));
                    }
                    //text "CT"
                    Recursos.font.draw(lote, F.keep("CT"), b.x+b.d[0], b.y+F.vAlign("font",'d'));
                    //"xifrat"
                    Recursos.iXifrat.draw(lote, F.keep(F.ifText(b.iCad.sept, "7",F.ifText(b.iCad.triada,
"5"))),b.x+b.d[1],b.y+F.vAlign("font",'d',"CT")+F.vAlign("iXifrat")*0.7f);
                    //diferent i tcp
                    Recursos.iQuant.draw(lote, F.keep(F.ifText(b.iCad.dif, " d")+F.ifText(b.iCad.tcp, " !")),
b.x+b.d[2], b.y+F.vAlign("iQuant",'d'));
                }

                for(BotIR b: PIng.bTipusRes){
                    lote.setColor(S.ingredients);

                    lote.draw(Recursos.rect, b.x-b.h*0.2f, b.y-b.h*0.2f, b.w+b.h*0.4f, b.h);
                    //text "RI"
                    Recursos.font.draw(lote, F.keep("RI"), b.x, b.y+F.vAlign("font",'d'));
                    //"xifrat"
                    Recursos.iXifrat.draw(lote, F.keep(F.ifText(b.iRI.sens,
"S","7")),b.x+b.d*1.1f,b.y+F.vAlign("iXifrat")*1.4f);
                }
            }else{ //si no hi ha ingredients
                Recursos.iQuant.setColor(S.graus);
                Recursos.iQuant.draw(lote,"No s'ha definit cap ingredient (enunciat
lliure).",50,Pant.screenH*0.83f);
            }

            Recursos.bNotes.draw(lote,"Tessitures dels cantants:",50, yTess+Pant.screenH*0.23f);
            lote.setColor(S.graus);

            //pentagrames
            for(int i=0;i<5;i++){
                for(int j=0;j<4;j++){
                    lote.draw(Recursos.rect, dTess*j+(mTess), yTess+dEspaiEsb*i, wTess, 2f);
                }
            }

            //soprano (S)
            Recursos.font.setColor(S.graus);
            Recursos.font.draw(lote,"S",mTess-50f,yTess+50f);
            lote.draw(Recursos.sol, mTess+10f, yTess-56f*r1e, dEspaiEsb*4f, dEspaiEsb*8f);
            lote.draw(Recursos.nota, mTess+140f, yTess+H.notePos(H.tMinS.nom,H.tMinS.ia, false,dEspaiEsb),
dNotaEsb, dNotaEsb);
            lote.draw(Recursos.rect, mTess+140f-dNotaEsb*0.25f,yTess-dEspaiEsb, dNotaEsb*1.5f, 2f);
            lote.draw(Recursos.nota, mTess+85f, yTess+H.notePos(H.tMaxS.nom,H.tMaxS.ia, false,dEspaiEsb),
dNotaEsb, dNotaEsb);
            lote.draw(Recursos.rect, mTess+85f-dNotaEsb*0.25f,yTess+dEspaiEsb*5f, dNotaEsb*1.5f, 2f);

            //contralt (A)
            Recursos.font.draw(lote,"A",mTess+dTess-50f,yTess+50f);
            lote.draw(Recursos.sol, mTess+dTess+10f, yTess-56f*r1e, dEspaiEsb*4f, dEspaiEsb*8f);
            lote.draw(Recursos.nota, mTess+dTess+140f, yTess+H.notePos(H.tMinA.nom,H.tMinA.ia, false,dEspaiEsb),
dNotaEsb, dNotaEsb);
            lote.draw(Recursos.rect, mTess+dTess+140f-dNotaEsb*0.25f,yTess-dEspaiEsb, dNotaEsb*1.5f, 2f);
            lote.draw(Recursos.rect, mTess+dTess+140f-dNotaEsb*0.25f,yTess-dEspaiEsb*2f, dNotaEsb*1.5f, 2f);
            lote.draw(Recursos.rect, mTess+dTess+140f-dNotaEsb*0.25f,yTess-dEspaiEsb*3f, dNotaEsb*1.5f, 2f);
            lote.draw(Recursos.nota, mTess+dTess+85f, yTess+H.notePos(H.tMaxA.nom,H.tMaxA.ia, false,dEspaiEsb),
dNotaEsb, dNotaEsb);

            //tenor (T)
            Recursos.font.draw(lote,"T",mTess+2*dTess-50f,yTess+50f);
```

```
                lote.draw(Recursos.fa, mTess+2*dTess+10f, yTess-56f*r1e, dEspaiEsb*4f, dEspaiEsb*8f);
                lote.draw(Recursos.nota, mTess+2*dTess+140f, yTess+H.notePos(H.tMinT.nom,H.tMinT.ia,
true,dEspaiEsb), dNotaEsb, dNotaEsb);
                lote.draw(Recursos.nota, mTess+2*dTess+85f, yTess+H.notePos(H.tMaxT.nom,H.tMaxT.ia, true,dEspaiEsb),
dNotaEsb, dNotaEsb);
                lote.draw(Recursos.rect, mTess+2*dTess+85f-dNotaEsb*0.25f,yTess+dEspaiEsb*5f, dNotaEsb*1.5f, 2f);
                lote.draw(Recursos.rect, mTess+2*dTess+85f-dNotaEsb*0.25f,yTess+dEspaiEsb*6f, dNotaEsb*1.5f, 2f);
                lote.draw(Recursos.rect, mTess+2*dTess+85f-dNotaEsb*0.25f,yTess+dEspaiEsb*7f, dNotaEsb*1.5f, 2f);

                //baix (B)
                Recursos.font.draw(lote,"B",mTess+3*dTess-50f,yTess+50f);
                lote.draw(Recursos.fa, mTess+3*dTess+10f, yTess-56f*r1e, dEspaiEsb*4f, dEspaiEsb*8f);
                lote.draw(Recursos.nota, mTess+3*dTess+140f, yTess+H.notePos(H.tMinB.nom,H.tMinB.ia,
true,dEspaiEsb), dNotaEsb, dNotaEsb);
                lote.draw(Recursos.nota, mTess+3*dTess+85f, yTess+H.notePos(H.tMaxB.nom,H.tMaxB.ia, true,dEspaiEsb),
dNotaEsb, dNotaEsb);
                lote.draw(Recursos.rect, mTess+3*dTess+85f-dNotaEsb*0.25f,yTess+dEspaiEsb*5f, dNotaEsb*1.5f, 2f);
            }

            lote.setColor(S.fons); //cercle del to (botó del menú) TODO ajustar mida de lletra com a l'altra
pantalla per les tonalitats llargues xD
            lote.draw(Recursos.cir, bMenu.x-bMenu.w*1.2f/2f, bMenu.y-bMenu.h*1.2f/2f, bMenu.w*1.2f, bMenu.h*1.2f);
            lote.setColor(S.tonalitat);
            if(escoltant){
                lote.setColor(S.greyScale(lote.getColor()));
            }
            lote.draw(Recursos.cir, bMenu.x-bMenu.w/2f, bMenu.y-bMenu.h/2f, bMenu.w, bMenu.h);
            lote.setColor(S.fons);
            lote.draw(Recursos.cir, bMenu.x-bMenu.w*A.rMenu/2f, bMenu.y-bMenu.h*A.rMenu/2f, bMenu.w*A.rMenu,
bMenu.h*A.rMenu);
            Recursos.font.setColor(0f,0f,0f,1f);
            Recursos.font.draw(lote, F.keep(H.modeNotes(H.tonalitat.nom,H.mode)+H.getAlt(H.tonalitat)+"
"+H.rModes[H.mode]), bMenu.x+F.align(), bMenu.y+F.vAlign());

        lote.end();

        /**- Actualització de les animacions - */
        A.update(delta);

        /**- Actualització del gestor de sons i la reproducció d'aquests -*/
        if(Recursos.carregantSons){
            if(Recursos.manager.update()){ //això fa un trosset de la càrrega, i retorna false si encara no està
                System.out.println("Càrrega de sons acabada. Pots reproduir");
                pucEscoltar = true;
                Recursos.carregantSons=false;
            }
        }else if(escoltant){ //estic escoltant l'esquema
            if(tAcord>1.4f){ //s'ha acabat l'acord anterior
                if(aAct+1<=Recursos.ultimAmbNotes){ //queden acords (amb alguna nota: si tota la resta és buida m'ho
estalvio)
                    aAct++;
                    fixScroll();
                    permisosBotons();
                    Recursos.playChord(esquemaEnNotes.get(aAct));
                    tAcord=0;
                }else{
                    aAct=lastAct;
                    escoltant=false;
                    mostraMenu=true;
                }
            }
            tAcord+=delta; //incremento
        }
    }

    /********** CLICK **********/

    @Override
    public boolean touchDown(int screenX, int screenY, int pointer, int button){
        clickBot = false;
        click.x = screenX;
        click.y = screenY;
        encuadre.unproject(click);

        //System.out.println(click.x+","+click.y);
        if(!escoltant){
            if(!mostraMenu){
                //botons de veu
                if(click.x<300){ //miro d'acotar-ho abans per no obligar-lo a fer les potències amb cada click.
                    ii=0;
```

```
                for(BotQ b: bVeus){
                    if(!clickBot){
                        if(Math.pow((click.x-b.x),2)+Math.pow((click.y-b.y),2)<Math.pow(b.w/2f, 2)){ //pitàgores.
Perquè igual apropo massa els botons rodons i més val que no detecti quadrats.
                            vAct= (byte) ii;

                            clickBot=true; //per estalviar-li comprovar els altres botons si ja sap que n'he
clicat un
                        }
                    }
                    ii++;
                }
            }
            //botons de nota TODO potser valdria la pena comprovar-los enrere perquè si no estic comprovant
sempre primer la 9a i la 7a, que en molts acords no hi són.
            if(click.x>Pant.realW*Pant.ratio-300){ //només comprovo els botons si s'ha polsat al marge dret
                ii=0;
                for(BotQ b: bNotes){
                    if(!clickBot){
                        if(Math.pow((click.x-b.x),2)+Math.pow((click.y-b.y),2)<Math.pow(b.w/2f,2)){
                            if(b.grau==7 && !puc7a || b.grau==9 && !puc9a){
                                System.out.println("L'acord escollit no té la nota demanada");
                            }else{
                                if(altOp/10!=b.grau){
                                    if(esquema.get(aAct).satb[vAct]!=b.grau){ //si no hi ha cap nota a la veu, la
poso a l'alçada per defecte

                                        System.out.println("Nota "+b.grau+" a la veu "+vAct);
                                        esquema.get(aAct).set(vAct, b.grau, H.defaultIA(Pant.esbos.get(aAct),
b.grau%7, vAct));
                                    }else{ //en cas contrari pujo octava (ja torna a baix si és necessari perquè
surto dels marges)
                                        dAux = esquema.get(aAct);
                                        dAux.ia[vAct] = (byte)
H.canviaOctava(H.getChordNote(Pant.esbos.get(aAct),dAux.satb[vAct]).nom, dAux.ia[vAct], vAct);
                                    }
                                    tradueixNotes(false); //actualitzo la traducció de l'acord actual per
reflectir el canvi.
                                    calCorregir = true;
                                }else{
                                    System.out.println("Nota amb opcionals. Esperant resolució.");
                                    A.bAltOp=(byte) ii; //marco quin botó he polsat
                                    A.tAltOp=0; //engego el temporitzador
                                }
                            }

                            clickBot=true;
                            fixScroll();
                        }
                    }
                    ii++;
                }
            }
            if(!clickBot){
                if(Math.pow((click.x-bEsborra.x),2)+Math.pow((click.y-bEsborra.y),2)<Math.pow(bEsborra.w/2f,2)){
                    dAux.set(vAct,0,0);

                    tradueixNotes(false); //actualitzo la traducció de l'acord actual per reflectir el canvi.
                    calCorregir=true;
                    clickBot=true;
                    fixScroll();
                }
            }
            if(!clickBot){ //botons de dreta i esquerra (navegació per l'esquema).
                if(Math.pow((click.x-bEsq.x),2)+Math.pow((click.y-bEsq.y),2)<Math.pow(bEsq.w/2f, 2)){
                    if(aAct>0){
                        aAct--;
                        System.out.println("Esquerra. Acord actual: "+aAct);
                        fixScroll();
                    }else{
                        System.out.println("Ja ets al primer acord");
                    }
                    //amb el canvi d'acord recalculo els permisos de 7a i 9a
                    permisosBotons();
                }else if(Math.pow((click.x-bDr.x),2)+Math.pow((click.y-bDr.y),2)<Math.pow(bDr.w/2f, 2)){
                    if(aAct<esquema.size-1){
                        aAct++;
                        System.out.println("Dreta. Acord actual: "+aAct);
                        fixScroll();
                    }else{
                        System.out.println("Ja ets a l'últim acord");
```

```
                }
                //amb el canvi d'acord recalculo els permisos de 7a i 9a
                permisosBotons();
            }
        }
    }else{ //aquí van els botons de la pantalla del menú
        if(Math.pow(click.x-bErrors.x, 2)+Math.pow(click.y-bErrors.y, 2)<Math.pow(bErrors.w/2, 2)){
            if(!mostraErrors){
                System.out.println("Veure errors.");
                mostraErrors=true;
                mostraAlertes=false;
            }else{
                System.out.println("Tancar errors");
                mostraErrors=false;
            }

            clickBot=true;
        }else if(Math.pow(click.x-bAlertes.x, 2)+Math.pow(click.y-bAlertes.y, 2)<Math.pow(bAlertes.w/2, 2)){
            if(!mostraAlertes){
                System.out.println("Veure alertes.");
                mostraAlertes=true;
                mostraErrors=false;
            }else{
                System.out.println("Tancar alertes.");
                mostraAlertes=false;
            }

            clickBot=true;
        }
        if(mostraErrors || mostraAlertes){ //quan veig registres
            if(!clickBot){
                if(click.x>bAnt.x-bAnt.w/2-bAnt.h/2 && click.x<bAnt.x+bAnt.w/2+bAnt.h/2
                && click.y>bAnt.y-bAnt.h/2 && click.y<bAnt.y+bAnt.h/2){ //botó anterior registre
                    System.out.println("Registre anterior.");
                    if(mostraErrors){
                        if(logE.size>1){
                            if(errorAct>0){
                                errorAct--;
                            }else{
                                errorAct=logE.size-1;
                            }
                            System.out.println("- Nou error actual: "+errorAct);
                        }else{
                            System.out.println("- Només hi ha una entrada al registre.");
                        }
                    }else{
                        if(logA.size>1){
                            if(alertaAct>0){
                                alertaAct--;
                            }else{
                                alertaAct=logA.size-1;
                            }
                            System.out.println("- Nova alerta actual: "+alertaAct);
                        }else{
                            System.out.println("- Només hi ha una entrada al registre.");
                        }
                    }
                }else if(click.x>bSeg.x-bSeg.w/2-bSeg.h/2 && click.x<bSeg.x+bSeg.w/2+bSeg.h/2
                && click.y>bSeg.y-bSeg.h/2 && click.y<bSeg.y+bSeg.h/2){ //botó següent registre
                    System.out.println("Següent registre.");
                    if(mostraErrors){
                        if(logE.size>1){
                            if(errorAct<logE.size-1){
                                errorAct++;
                            }else{
                                errorAct=0;
                            }
                            System.out.println("- Nou error actual: "+errorAct);
                        }else{
                            System.out.println("- Només hi ha una entrada al registre.");
                        }
                    }else{
                        if(logA.size>1){
                            if(alertaAct<logA.size-1){
                                alertaAct++;
                            }else{
                                alertaAct=0;
                            }
                            System.out.println("- Nova alerta actual: "+alertaAct);
                        }else{
```

```
                                System.out.println("- Només hi ha una entrada al registre.");
                            }
                        }
                    }
                }
            }else{ //quan no estic veient registres
                if(!clickBot){
                    if(click.x>bEscolta.x-bEscolta.w/2f && click.x<bEscolta.x+bEscolta.w/2f &&
click.y>bEscolta.y-bEscolta.h/2f && click.y<bEscolta.y+bEscolta.h/2f){
                        System.out.println("Vull escoltar");
                        if(!pucEscoltar){
                            System.out.println("Carregant notes necessàries");
                            Recursos.carregaSons();
                        }else{
                            if(!escoltant){ //enceto només si no s'està escoltant ja
                                System.out.println("Reproduint esquema...");
                                lastAct=aAct; //deso l'últim actual per recuperar-lo en acabat.
                                aAct=0;
                                fixScroll();
                                permisosBotons();
                                Recursos.playChord(esquemaEnNotes.get(aAct)); //reprodueixo l'acord inicial
                                tAcord = 0; //i poso a zero el temporitzador
                                escoltant = true;
                                mostraMenu = false;
                            }
                        }

                        clickBot=true;
                    }else if(click.x>bEsbos.x-bEsbos.w/2f && click.x<bEsbos.x+bEsbos.w/2f && click.y>bEsbos.y-
bEsbos.h/2f && click.y<bEsbos.y+bEsbos.h/2f){
                        if(vullTornar){
                            System.out.println("Tornant a la pantalla de l'esbós.");
                            app.setScreen(app.pant);
                        }else{ //demano confirmació per si ha clicat sense voler
                            System.out.println("Es demana tornar a l'esbós. Confirmar la tria.");
                            posaElsBotonsPer("vullTornar");
                            vullBuidar=false;
                            vullTornar=true;
                        }
                        clickBot=true;
                    }else if(click.x>bBuida.x-bBuida.w/2f && click.x<bBuida.x+bBuida.w/2f && click.y>bBuida.y-
bBuida.h/2f && click.y<bBuida.y+bBuida.h/2f){
                        if(vullBuidar){
                            System.out.println("Buidant la realització.");
                            nouEsquema(esquema.size);
                            tradueixNotes(true);
                            corregeix();
                            permisosBotons();

                            posaElsBotonsPer("menu");
                            vullBuidar=false;
                            mostraMenu=false;
                        }else{ //demano confirmació per si ha clicat sense voler
                            System.out.println("Es demana buidar la realització. Confirmar la tria.");
                            posaElsBotonsPer("vullBuidar");
                            vullTornar=false;
                            vullBuidar=true;
                        }
                        clickBot=true;
                    }else if(click.x>bCancel.x-bCancel.w/2f && click.x<bCancel.x+bCancel.w/2f &&
click.y>bCancel.y-bCancel.h/2f && click.y<bCancel.y+bCancel.h/2f){
                        if(vullTornar){
                            System.out.println("Tornada a l'esbós cancel·lada.");
                            posaElsBotonsPer("menu");
                            vullTornar=false;

                            clickBot=true;
                        }else if(vullBuidar){
                            System.out.println("Buidatge de la realització cancel·lat.");
                            posaElsBotonsPer("menu");
                            vullBuidar=false;

                            clickBot=true;
                        }
                    }
                }
            }
        }
        //botó rodó del menú (que es veu sempre)
        if(!clickBot && click.x>bMenu.x-bMenu.w/2f && click.x<bMenu.x+bMenu.w/2f && click.y>bMenu.y-bMenu.h/2f
```

```
&& click.y<bMenu.y+bMenu.h/2f){
                if(!mostraMenu){
                    A.tMenu = 0;
                }else{
                    mostraMenu = false;
                }
                clickBot=true;
            }
        }else{
            escoltant=false;
        }

        return true;
    }

    @Override
    public boolean touchUp(int screenX, int screenY, int pointer, int button) {
        click2.x=screenX;
        click2.y=screenY;
        encuadre.unproject(click2);



        /**- - - Gestió de les alteracions opcionals - - */
        if(A.tAltOp>=0f){

            int clickAlt = 0; //em guardo a quin botó d'opcional he deixat anar per no haver de fer el càlcul de la
col·lisió moltes vegades
            if(Math.pow(click2.x-bAlt1.x, 2)+Math.pow(click2.y-bAlt1.y, 2)<Math.pow(bAlt1.w/2f, 2)){
                clickAlt=1;
            }else if(Math.pow(click2.x-bAlt2.x, 2)+Math.pow(click2.y-bAlt2.y, 2)<Math.pow(bAlt2.w/2f, 2)){
                clickAlt=2;
            }

            if(A.tAltOp<A.tHold || clickAlt!=0){ //no s'ha aguantat o bé s'ha aguantat i deixat anar sobre les
alteracions opcionals
                if(esquema.get(aAct).satb[vAct]!=bNotes.get(A.bAltOp).grau){ //si no hi ha cap nota a la veu, la
poso a l'alçada per defecte
                    System.out.println("Nota "+bNotes.get(A.bAltOp).grau+" a la veu "+vAct);
                    esquema.get(aAct).set(vAct, bNotes.get(A.bAltOp).grau, H.defaultIA(Pant.esbos.get(aAct),
bNotes.get(A.bAltOp).grau%7, vAct));
                }else if(clickAlt==0){ //en cas contrari pujo octava (si només estic canviant l'alteració no).
                    dAux = esquema.get(aAct);
                    dAux.ia[vAct] = (byte) H.canviaOctava(H.getChordNote(Pant.esbos.get(aAct),dAux.satb[vAct]).nom,
dAux.ia[vAct], vAct);
                }
                //si estava triant alteració la part anterior haurà assegurat que hi ha nota posada. El següent
assigna l'alteració opcional escollida a l'acord.
                if(clickAlt==1){ //alteració 1: altN=false
                    if(altOp%10==9){
                        Pant.esbos.get(aAct).alt9=false;
                    }else if(altOp%10==7){
                        Pant.esbos.get(aAct).alt7=false;
                    }else if(altOp%10==6){
                        Pant.esbos.get(aAct).alt6=false;
                    }else{
                        System.out.println("Alteració no reconeguda");
                    }
                }else if(clickAlt==2){ //alteració 2: altN=true
                    if(altOp%10==9){
                        Pant.esbos.get(aAct).alt9=true;
                    }else if(altOp%10==7){
                        Pant.esbos.get(aAct).alt7=true;
                    }else if(altOp%10==6){
                        Pant.esbos.get(aAct).alt6=true;
                    }else{
                        System.out.println("Alteració no reconeguda");
                    }
                }

                tradueixNotes(false); //actualitzo la traducció de l'acord actual per reflectir el canvi.
                calCorregir = true;
            }else{
                System.out.println("Canvi d'alteració descartat");
            }
        }
        A.tAltOp = -1;

        /**- - - Gestió del botó del menú - - -*/
        if(A.tMenu>=0f && A.tMenu<A.tHold){
```

```
            System.out.println("Mostra el menú");
            A.rMenu = 0.8f;
            mostraMenu = true;
        }
        A.tMenu = -1;

        return true;
    };

    @Override
    public boolean touchDragged(int x, int y, int pointer) {
        click2.x=x;
        click2.y=y;
        encuadre.unproject(click2);

        if(!mostraMenu){
            if (click.y > Pant.screenH*0.15f && click.y < Pant.screenH*0.95f && click.x > 25 && click.x < rLim){
//si els clicks inicial i final...
                if(click2.y > Pant.screenH*0.15f && click2.y < Pant.screenH*0.95 && click2.x > 25 && click2.x <
rLim){ //són dins l'àrea mòbil...
                    if(Math.abs(click2.x-click.x)>5 || Math.abs(click2.y-click.y)>5){ //restringeixo a les vegades
que realment s'ha intentat moure expressament.
                        scroll+=click2.x-click.x; //TODO donar inèrcia a l'scroll. Ja hi ha fetes les variables, però
s'ha d'implementar.
                        System.out.println("\nArrossegat " + (click2.x-click.x));
                        System.out.println("Nou valor proposat: "+scroll);
                        System.out.println("Hi ha "+esquema.size+" acords i en caben "+ (rLim-lLim)/dEAc);
                        System.out.println("Hi ha ocupat "+(dEAc*esquema.size)+" i es veu "+(rLim-lLim));
                        if(esquema.size*dEAc<(rLim-lLim)){
                            System.out.println("Moviment impedit. Tots els acords són visibles.");
                            scroll=lLim;
                        }else if(scroll>lLim){
                            System.out.println("Ajustat al límit esquerra: "+scroll);
                            scroll=lLim;
                        }else if(scroll<((rLim)-dEAc*esquema.size)){ //TODO afegir alguna mena d'histèresi petita
per evitar que es torni boig. Investigar el comportament i això..
                            scroll=(int) ((rLim)-dEAc*esquema.size);
                            System.out.println("Ajustat al límit dret: "+scroll);
                        }else{
                            System.out.println("Moviment permès");
                        }
                        click.x=click2.x; //actualitzo la última posició clicada per poder fer una arrossegada
contínua fiable.
                        click.y=click2.y;

                        /**- - Ajust de l'esbós - -*/ //TODO igual es podria mirar de suavitzar el salt brusc que fa
quan està desfassat per culpa de l'ajust a fixScroll.
                        if((dEAc*esquema.size-(rLim-lLim))!=0){ //no vull dividir per zero
                            scrollEsb = (int) (lLimEsb+(scroll-lLim)*(dEAcEsb*esquema.size-(rLimEsb-
lLimEsb))/(dEAc*esquema.size-(rLim-lLim))); //faig la ràtio entre les longituds ocultes en cada cas, i trasllado la
posició del pentagrama gros a partir d'aquesta ràtio a la posició que vull al petit.
                        }
                    }else{
                        //aquí hi anirà el que sigui que vulgui fer amb touchDown sobre la zona arrossegable.
                    }
                }
            }
        } //fi if(!mostraMenu)

        return true;
    }

    public void fixScroll(){ //ajust de l'scroll després de remenar botons (per si queda l'acord actual fora).
        if(scroll+(aAct+1)*dEAc > rLim){
            scroll = (int) (rLim-dEAc*(aAct+1));
        }else if(scroll+aAct*dEAc < lLim){
            scroll = (int) (lLim-aAct*dEAc);
        }
        if((dEAc*esquema.size-(rLim-lLim))!=0){ //no vull dividir per zero (sí, ho he trobat perquè just ha
coincidit i ha petat tot)
            scrollEsb = (int) (lLimEsb+(scroll-lLim)*(dEAcEsb*esquema.size-(rLimEsb-lLimEsb))/(dEAc*esquema.size-
(rLim-lLim)));
        }
        if(scrollEsb+(aAct+1)*dEAcEsb > rLimEsb){ //ajusto també independentment l'esbós.
            scrollEsb = (int) (rLimEsb-dEAcEsb*(aAct+1));
        }else if(scrollEsb+aAct*dEAcEsb < lLimEsb){
            scrollEsb = (int) (lLimEsb-aAct*dEAcEsb);
        }
    }
```

```
    @Override
    public void resize(int width, int height) {
        encuadre.update(width, height);

        Pant.realW = Gdx.graphics.getWidth(); //faig els càlculs directament sobre les variables de la primera
pantalla
        Pant.realH = Gdx.graphics.getHeight();
        Pant.ratio = (float) Pant.screenH/Pant.realH;

        for(BotQ b: bNotes){
            b.x= (int) (Pant.realW*Pant.ratio-75f);
        }
        bEsborra.x = (int) (Pant.realW*Pant.ratio-75f);

        /**- - -Botons del menú - - -*/
        bAnt.x = (int) (xLogs+(Pant.realW*Pant.ratio-bErrors.w*0.75f-xLogs)/4f  -20f);
        bSeg.x = (int) (xLogs+(Pant.realW*Pant.ratio-bErrors.w*0.75f-xLogs)*3/4f+20f);
        bAnt.w = (int) ((Pant.realW*Pant.ratio-bErrors.w*0.75f-xLogs)*0.37f);
        bSeg.w = (int) ((Pant.realW*Pant.ratio-bErrors.w*0.75f-xLogs)*0.37f);

        bCancel.x = (int) ((Pant.realW*Pant.ratio-bErrors.w*0.75f+xLogs)/2f+180);

        bEscolta.x = (int) ((Pant.realW*Pant.ratio-bErrors.w*0.75f+xLogs)/2f);
        posaElsBotonsPer();

        rLim = (int) (Pant.realW*Pant.ratio-rxSolFa); //com a les cortines.
        rLimEsb = (int) (Pant.realW*Pant.ratio-rxEsb);

        dTess = (int) (Pant.realW*Pant.ratio/4.2f);
        mTess = (int) ((Pant.realW*Pant.ratio-(3*dTess+wTess+50))*0.8f);

        camera.position.set(Pant.realW/2f*Pant.ratio,Pant.screenH/2f,0);

        Pant.posicionaIngs();
    }

    @Override
    public void show() {
        posaElsBotonsPer("menu"); //recoloco les coses que puguin haver quedat malament en quedar el menú obert quan
canvio de pantalla
        vullTornar=false;
        mostraMenu=false;
        Gdx.input.setInputProcessor(this); //canvio el receptor d'entrades a aquesta pantalla

        if(nouEsquema){
            System.out.println("Generant nou esquema buit a partir dels "+Pant.esbos.size+" acords...");
            nouEsquema(Pant.esbos.size); //afegeix tants acords desplegats buits com acords hi ha a l'esbós.

            nouEsquema = false;
        }else{
            aAct = Pant.aAct; //si vinc d'editar i no el faig nou, mantinc l'actual que tenia a l'altra pantalla
        }
        tradueixNotes(true); //quan entro a la pantalla de la realització, recalculo des de zero la traducció dels
números de l'esquema a variables nota.
        corregeix(); //pre-correcció per tenir generades les variables que en surten (si no hi són, algunes coses no
es dibuixen).
        permisosBotons(); //càlcul dels permisos de 7 i 9 per l'acord actual (normalment els calculo en canviar
d'acord, però aquí si vinc d'editar l'esbós poden estar malament

        lLim = (int) (310+dArm*Math.abs(H.armadura)); //per ajustar el límit a l'armadura si l'he canviat abans de
venir a aquesta pantalla
        lLimEsb = (int) (xEsb+120f+dArmEsb*Math.abs(H.armadura));

        scroll = lLim;
        scrollEsb = lLimEsb;

        if(Pant.fila==0){
            yTess = (int) (Pant.screenH*0.41f);
        }else{
            yTess = (int) (Pant.screenH*0.36f);
        }
    }

    @Override
    public void pause() {
    }
    @Override
    public void resume() {
    }
    @Override
```

```
    public void hide() { //em desfaig dels àudios que havia carregat
        pucEscoltar = false;
        Recursos.manager.clear();
    }
    @Override
    public void dispose() {
    }

    /**** POSICIONAMENT DELS ELEMENTS ****/
    public void posaElsBotonsPer(){
        if(vullTornar){
            posaElsBotonsPer("vullTornar");
        }else if(vullBuidar){
            posaElsBotonsPer("vullBuidar");
        }else{
            posaElsBotonsPer("menu");
        }
    }
    public void posaElsBotonsPer(String config){
        if(config == "vullTornar"){
            bEsbos.x  = (int) ((Pant.realW*Pant.ratio-bErrors.w*0.75f+xLogs)/2f-180);
            bEsbos.w=300;

            bBuida.x = (int) ((Pant.realW*Pant.ratio-bErrors.w*0.75f+xLogs)/2f);
            bBuida.w=400;

            bCancel.y = bMenu.y+dBM;
        }else if(config == "vullBuidar"){
            bEsbos.x = (int) ((Pant.realW*Pant.ratio-bErrors.w*0.75f+xLogs)/2f);
            bEsbos.w=400;

            bBuida.x = (int) ((Pant.realW*Pant.ratio-bErrors.w*0.75f+xLogs)/2f-180);
            bBuida.w = 300;

            bCancel.y = bMenu.y;
        }else if(config == "menu"){
            bEsbos.x = (int) ((Pant.realW*Pant.ratio-bErrors.w*0.75f+xLogs)/2f);
            bEsbos.w=400;

            bBuida.x = (int) ((Pant.realW*Pant.ratio-bErrors.w*0.75f+xLogs)/2f);
            bBuida.w=400;
        }
    }

    /***************** FUNCIONS DE CÀLCUL I CORRECCIÓ ***************/

    /** - - - Generació d'un esquema buit - - - */
    public void nouEsquema(int n){
        esquema.clear();
        for(int i=0;i<n;i++){
            esquema.add(new Desplegat());
        }
        aAct=0; //poso l'actual a zero per començar pel principi
    }

    public void permisosBotons(){ //càlcul de permisos de sèptima i novena
        puc7a = false;
        puc9a = false;
        if(Pant.esbos.get(aAct).sept){
            puc7a = true;
        }
        if(Pant.esbos.get(aAct).dom){
            puc7a = true;
            puc9a = true;
        }
        altOp = H.opcionals(Pant.esbos.get(aAct));
        if(altOp!=0){ //si tinc opcionals en alguna nota ajusto la posició dels botons extra
            bAlt1.y=bNotes.get(4-altOp/10/2).y; //l'array comença a dalt i les desenes de altOp són 1 3 5 7 9 (les
he de passar als índex 4 3 2 1 0)
            bAlt2.y=bNotes.get(4-altOp/10/2).y;
        }

        System.out.println("Nou codi altOp: "+altOp);
    }

    /** - - - Traducció dels números a notes - - - */
    public void tradueixNotes(boolean reset){ //passo els valors numèrics relatius de "esquema" a valors absoluts
de nota ("esquemaEnNotes").
        System.out.println("Traduïnt posicions a notes reals...");
        if(reset){ //en cas que vulgui recalcular-ho des de zero.
```

```
            System.out.println("- Renovant tot l'esquema");
            esquemaEnNotes.clear();
            int i=0;
            for(Desplegat d: esquema){
                esquemaEnNotes.add(new Note[4]); //creo una Nota[4] corresponent a l'acord desplegat
                for(int veu=0;veu<4;veu++){ //i l'omplo amb la informació de les diferents veus.
                    if(d.satb[veu]!=0){ //si hi ha alguna nota marcada a l'esquema
                        esquemaEnNotes.get(i)[veu] = H.getChordNote(Pant.esbos.get(i), d.satb[veu]);
                        esquemaEnNotes.get(i)[veu].ia = d.ia[veu];
                    }else{
                        esquemaEnNotes.get(i)[veu] = new Note(-1,-1,-1); //poso -1 per saber que no hi ha res (les
notes sempre seran >0).
                    }
                }
                i++;
            }
        }else{ //si no demano refer tot, calculo l'acord actual i prou.
            System.out.println("- Càlcul només per l'acord actual");
            for(int veu=0;veu<4;veu++){
                if(esquema.get(aAct).satb[veu]!=0){
                    esquemaEnNotes.get(aAct)[veu] = H.getChordNote(Pant.esbos.get(aAct),
esquema.get(aAct).satb[veu]);
                    esquemaEnNotes.get(aAct)[veu].ia = esquema.get(aAct).ia[veu];
                }else{
                    esquemaEnNotes.get(aAct)[veu] = new Note(-1,-1,-1);
                }
            }
        }
    }

    /********* Correcció de la realització ********/

    public void corregeix(){ //TODO si he d'acabar permetent una correcció al final, hauria de treure fora d'això
la part que fa càlculs previs (perquè els dibuixos en depenen), i hauria de donar color per defecte als dibuixos,
també, perquè voldré veure'ls encara que no hi hagi correcció. :_D
        System.out.println("\nCorregint realització...");

        System.out.println("Posant a zero el registre...");
        logE.clear();
        logA.clear();

        System.out.println("Renovant llistes de notes utilitzades i duplicades...");
        tincNotes.clear();
        tincPle.clear();
        duplicades.clear();
        int i=0;
        for(Desplegat d: esquema){
            tincNotes.add(new boolean[]{false,false,false,false,false});
            duplicades.add((byte) 0);
            for(int veu=0;veu<4;veu++){
                if(d.satb[veu]!=0){ //si a la veu hi ha una nota...
                    if(!tincNotes.get(i)[d.satb[veu]/2]){ //i encara no la tenia...
                        tincNotes.get(i)[d.satb[veu]/2]=true; //marco que la nota està posada
                    }else{ //si ja tenia la nota que he trobat...
                        if(duplicades.get(i)==1 || (!tincNotes.get(i)[0] && duplicades.get(i)==3)){ //si ja tinc la
fonamental duplicada (o bé de moment no tinc fonamental i està duplicada la tercera (per si és la tercera, la
fonamental, en realitat))
                            duplicades.add((byte) (duplicades.pop()+10*d.satb[veu])); //afegeixo la segona nota
duplicada a les desenes (per mantenir la fonamental a la posició important)
                        }else{ //si no, passo a desenes el que hi havia i entro com a unitats la nota nova (si no hi
havia res, 10*0 = 0 i m'és igual).
                            duplicades.add((byte) (duplicades.pop()*10+d.satb[veu]));
                        }
                    }
                }
            }
            if(d.satb[0]!=0 && d.satb[1]!=0 && d.satb[2]!=0 && d.satb[3]!=0){ //si hi ha alguna cosa a tot arreu
                tincPle.add(true);
            }else{
                tincPle.add(false);
            }
            i++;
        }
        System.out.println(duplicades);
        //System.out.println(tincPle);

        /**- - - Càlcul de les inversions generades amb la realització - - -*/
        System.out.println("Calculant inversions dels acords realitzats...");
        inversions.clear();
        i=0;
```

```
        for(Desplegat d: esquema){
            if(tincPle.get(i)){ //només comprovo la inversió de l'acord si ja té totes les notes
                if(tincNotes.get(i)[4]){ //l'acord té novena
                    inversions.add((byte) (10+d.satb[3]/2-1)); //com amb la sèptima, però li resto 1 a la inversió,
perquè en realitat la fonamental no compta
                }else if(tincNotes.get(i)[3]){ //l'acord té sèptima
                    inversions.add((byte) (10+d.satb[3]/2)); //afegeixo 10 per marcar d'alguna manera que és amb
sèptima
                }else{ //l'acord és tríade
                    inversions.add((byte) (d.satb[3]/2)); //directament truncant el baix trobo la inversió
                }
            }else{
                inversions.add((byte) (-1)); //si no té totes les notes, afegeixo un qualsevol a la llista per
ocupar l'espai i prou.
            }
            i++;
        }
        System.out.println(inversions);

        /**- - - Comprovació de les notes utilitzades en cada acord - - -*/
        System.out.println("Calculant completesa dels acords...");
        complet.clear();
        cComplet.clear();
        for(i=0;i<tincNotes.size;i++){
            if(tincPle.get(i)){ //només comprovo la completesa si ja tinc totes les veus plenes
                if(tincNotes.get(i)[4]){ //l'acord té novena
                    if(tincNotes.get(i)[1] && tincNotes.get(i)[2] && tincNotes.get(i)[3]){ //si té 3, 5 i 7... (que
són totes, perquè aquí no hi haurà Fonamental).
                        complet.add(true); //marco complet
                        cComplet.add('n');
                    }else{
                        complet.add(false); //si no, marco incomplet.
                        if(tincNotes.get(i)[1] && tincNotes.get(i)[2]){ //si tinc 3 i 5 (que aquí són 1 i 3), vol dir
que falta la quinta de l'acord, que és la que es pot treure.
                            cComplet.add('a'); //En principi no és un drama truere la 5a, però com que no és
l'habitual aviso a l'usuari.
                            logA.add(new Log(i,1104));
                        }else{
                            cComplet.add('e');
                            logE.add(new Log(i,104));
                        }
                    }
                }else{ //no té novena TODO ajustar per si és tríade de sensible (tenir en compte dom, que si no
està malament)
                    if(tincNotes.get(i)[0] && tincNotes.get(i)[1] && tincNotes.get(i)[2]){ //si hi ha totes les notes
del tríade (la sèptima per al càlcul m'és igual, perquè si és un acord de sèptima ja voldrà dir que la té)
                        complet.add(true); //marco complet
                        cComplet.add('n');
                    }else{
                        complet.add(false); //si en falta alguna marco incomplet (aquí m'és igual quina. Només miro
que hi sigui tot).
                        if(tincNotes.get(i)[0] && tincNotes.get(i)[1]){ //I aquí ja miro que s'hagi tret la que
tocava (si tinc 1 i 3 he tret la quinta).
                            if(inversions.get(i)==0 || inversions.get(i)>5){ //en EF o amb sèptima treure la quinta
no és un drama (mateix que abans, marco alerta)
                                cComplet.add('a');
                                logA.add(new Log(i,1104));
                            }else{ //en 6 (aquí no entra 6/4, perquè 6/4 sempre té 5a perquè la té al baix)
                                cComplet.add('e'); //no permeto fer incomplet el 6.
                                logE.add(new Log(i,0));
                            }
                        }else{
                            cComplet.add('e');
                            logE.add(new Log(i,104));
                        }
                    }
                }
            }else{
                complet.add(false); //per omplir (en principi no ho vindré a buscar en cap moment)
                cComplet.add('o'); //ocult (per omplir l'espai amb alguna cosa).
            }
        }
        //ajustos degut a la CP
        if(Pant.tincCP){ //si per algun motiu no tingués CP, deixaria tal com ha quedat de la correcció normal
            if(cComplet.get(cComplet.size-2)=='a'){ //si el V de la CP és incomplet però correcte (he posat alerta
perquè faltava la 5a).
                System.out.println("V cadencial incomplet");
                if(cComplet.get(cComplet.size-1)=='a'){ //si el I també és incomplet, deixo tot com està (amb
alertes) perquè en principi se suposa que un ha d'anar compet.
                    System.out.println("- Compte. L'últim I també ha quedat incomplet!");
```

```
            }else if(cComplet.get(cComplet.size-1)=='n'){ //si el I és complet, marco que el V anterior estava
bé.
                System.out.println("- I complet. V cadencial correcte.");
                cComplet.removeIndex(cComplet.size-2); //trec el del V
                cComplet.insert(cComplet.size-1, 'c'); //el marco correcte
            }
        }else if(cComplet.get(cComplet.size-2)=='n'){ //si el V està complet, comprovo el I, per marcar bé si
està incomplet.
            if(cComplet.get(cComplet.size-1)=='a'){ //El I estava incomplet però correcte.
                System.out.println("Últim I incomplet.\nVe d'un V complet. Correcte.");
                cComplet.removeIndex(cComplet.size-1);
                cComplet.add('c');
            }
        }
    }
    System.out.println(complet);
    System.out.println(cComplet);

    /**- - - - Correcció de les inversions i comparació amb les de l'esbós - - - -*/
    System.out.println("Calculant correcció de les inversions...");
    cInv.clear();
    if(tincPle.get(0) && inversions.get(0)!=0){ //comprovo que el primer estigui en E.F.
        cInv.add('e');
        logE.add(new Log(i,0));
    }else{
        cInv.add('n');
    }
    i=1; //a partir del segon...
    while(i<inversions.size-3 || (!Pant.tincCP && i<inversions.size)){ //deixo els tres últims per calcular-los
a part amb la CP, a menys que no tingui CP
        if(tincPle.get(i)){ //les inversions només estan calculades pels plens, així que no té sentit corregir
els que no ho estan
            if(inversions.get(i)==2){ //si trobo un 6/4...
                System.out.println("Trobat 6/4 a la posició "+i);
                if(inversions.get(i-1)==2){ //i el d'abans també és un 6/4...
                    System.out.println("Incorrecte. Dos 6/4 seguits.");
                    cInv.add('e');
                    logE.add(new Log(i,27));
                }else{ //si el d'abans no porta problema...
                    if(Pant.esbos.get(i).inv==2 && !Pant.esbos.get(i).sept && !Pant.esbos.get(i).dom){ //si ja
estava planejat, deixo normal.
                        cInv.add('n');
                    }else{
                        cInv.add('a'); //si no marco alerta TODO al menú, posar una opció per ajustar l'esbós
d'acord amb les inversions escollides aquí.
                        logA.add(new Log(i,1001));
                    }
                }
            }else{ //si no és un 6/4
                if(Pant.esbos.get(i).inv!=0){ //si a l'esbós havia marcat alguna inversió expressament
                    if(Pant.esbos.get(i).sept || Pant.esbos.get(i).dom){ //si és de sèptima
                        if(inversions.get(i)>5 && Pant.esbos.get(i).inv==inversions.get(i)%10){ //està igual que
a l'esbós (a la realització també té sèptima i la inversió coincideix).
                            cInv.add('n');
                        }else{
                            cInv.add('a');
                            logA.add(new Log(i,1001));
                        }
                    }else{ //si és tríade
                        if(Pant.esbos.get(i).inv==inversions.get(i)){ //està igual que a l'esbós
                            cInv.add('n');
                        }else{
                            cInv.add('a');
                            logA.add(new Log(i,1001));
                        }
                    }
                }else{ //si no havia marcat res, deixo normal
                    cInv.add('n');
                }
            }
        }else{
            cInv.add('n');
        }
        i++;
    }
    if(Pant.tincCP){
        System.out.println("- Inversions a la CP..."); //TODO en algun moment adaptar a les CP sense I_6/4
        if(inversions.get(inversions.size-3)!=2){ //el tercer per la cua ha de ser 6/4 (I_6/4)
            if(tincPle.get(inversions.size-3)){ //si no és ple no té sentit
                cInv.add('e'); //(és ple i no 6/4)
```

```
                    logE.add(new Log(i,0));
                }else{
                    cInv.add('n');
                }
            }else{ //el tercer per la cua és 6/4...
                if(inversions.get(inversions.size-4)==2){ //vigilo que no hagi posat la subdominant també amb 6/4
(que faria dos 6/4 seguits). Si s'ha comès l'error, marco error al primer 6/4, perquè el segon és obligat.
                    cInv.removeIndex(cInv.size-1);
                    cInv.add('e');
                    logE.add(new Log(cInv.size-1,27));
                }
                cInv.add('n');
            }
            if(tincPle.get(inversions.size-2) && inversions.get(inversions.size-2)%10!=0){ //el segon per la cua ha
de ser en E.F. (V o V7)
                cInv.add('e');
                logE.add(new Log(inversions.size-2,0));
            }else{
                cInv.add('n');
            }
            if(tincPle.get(inversions.size-1) && inversions.get(inversions.size-1)!=0){ //l'últim ha de ser en E.F.
i sense sèptima ni res. (I)
                cInv.add('e');
                logE.add(new Log(inversions.size-1,0));
            }else{
                cInv.add('n');
            }
        }
        System.out.println(cInv);

        /**- - - Correcció de les notes duplicades - - -*/
        System.out.println("Comprovant correcció de les duplicacions...");
        cDuplicades.clear();
        Note notaDuplicada;
        i=0;
        for(byte d: duplicades){
            if(d!=0){ //si tinc informació d'alguna nota duplicada...

                notaDuplicada = H.getChordNote(Pant.esbos.get(i), d%10); //trobo quina és

                if(notaDuplicada.mateixaNota(H.getNote(6, H.tonalitat, 0))){ //si és la sensible (el 6 del major és
la sensible)
                    System.out.println("- ("+i+") Error. No pots duplicar la sensible.");
                    cDuplicades.add('e'); //no puc duplicar la sensible.
                    logE.add(new Log(i,0));
                }else{ //no és la sensible
                    if(inversions.get(i)<5){ //és un acord tríade (sèptima és 10,11,12,13)
                        if(inversions.get(i)==0){ //tríade E.F.
                            if(d%10==1){ //tinc duplicada la fonamental. TODO ajustar per tríade de sensible
                                cDuplicades.add('n');
                            }else{ //tinc duplicada una altra nota (aquí si tinc dues notes diferents duplicades pot
ser que es deixi alguna cosa sense comprovar, però com que estarà malament igualment per un altre lloc tant hi fa).
                                System.out.println("- ("+i+") Duplicació poc habitual.");
                                cDuplicades.add('a'); //pot ser que estigui bé, segons el cas, així que deixo en
alerta.
                                logA.add(new Log(i,1001));
                            }
                        }else if(inversions.get(i)==1){ //si és en 6
                            if(d%10!=esquema.get(i).satb[3]){ //si NO he duplicat el baix...
                                cDuplicades.add('n'); //està bé (perquè sé que no és la sensible, que ho he comprovat
abans).
                            }else{ //he duplicat el baix, comprovo si la nota correspon a un dels graus tonals
(I/IV/V)
                                System.out.println("- ("+i+") Baix duplicat en un 6.");
                                if(notaDuplicada.equals(H.getNote(0)) || notaDuplicada.equals(H.getNote(3)) ||
notaDuplicada.equals(H.getNote(4))){
                                    System.out.println("- - Correcte. El baix és grau tonal.");
                                    cDuplicades.add('n'); //hi correspon, correcte.
                                }else{
                                    System.out.println("- - Error. El baix no és grau tonal.");
                                    cDuplicades.add('e'); //és una qualsevol, malament.
                                    logE.add(new Log(i,0));
                                }
                            }
                        }else{ //6/4
                            cDuplicades.add('i');
                            //TODO mirar si hi ha alguna norma sobre duplicacions pel 6/4 (penso que a part de la
sensible res, però potser passa com amb els de 6)
                        }
                        for(Cadencia c: Pant.cadencies){ //ajusto per no marcar alerta quan és tònica duplicada en CT
```

```
tríada
                        if(i==c.n+c.w && !c.cp){ //si sóc al segon acord d'una cadència trencada
                            if(notaDuplicada.mateixaNota(H.tonalitat)){
                                System.out.println("- - Ajustant: tònica duplicada en CT tríada. Correcte.");
                                cDuplicades.removeIndex(cDuplicades.size-1);
                                cDuplicades.add('n');
                                break;
                            }
                        }
                    }
                }else{ //és un acord amb sèptima
                    if(d==7){
                        System.out.println("- ("+i+") Error. No pots duplicar la sèptima.");
                        cDuplicades.add('e');
                    }else{
                        System.out.println("- ("+i+") Duplicació en un acord de sèptima (poc habitual).");
                        cDuplicades.add('a'); //marco alerta perquè és estrany que la sèptima tingui nota
duplicada.
                        logA.add(new Log(i,1102));
                    }
                }
            }

            }else{
                cDuplicades.add('i'); //irrellevant
            }
            i++;
        }
        System.out.println(cDuplicades);

        /**- - - Correcció dels límits de les veus- - -*/
        System.out.println("Comprovant adequació a les tessitures...");
        cTessitures.clear();
        byte tOffset;
        i=0;
        for(Note[] ac: esquemaEnNotes){
            for(int veu=0; veu<4; veu++){
                if(ac[veu].nom!=-1){ //si hi ha una nota posada
                    tOffset = (byte) H.dinsTessitura(ac[veu], veu);
                    if(tOffset != 0){ //si la nota és fora la tessitura
                        switch(veu){
                        case 0: //S (poso de moment alerta o error segons com d'estricte considero que és el límit)
TODO en algun moment investigar com d'estrictes es consideren en general
                            if(tOffset>0){ //surt per dalt
                                cTessitures.add(new CorPuntual(i,veu,'e')); //la soprano per dalt la considero error
                                logE.add(new Log(i,4));
                            }else{ //surt per baix
                                if(tOffset<-1){
                                    cTessitures.add(new CorPuntual(i,veu,'e')); //la soprano més d'una nota per baix
la considero error (cap de les dues referències ho permet)
                                    logE.add(new Log(i,4));
                                }else{
                                    cTessitures.add(new CorPuntual(i,veu,'a')); //la soprano una nota per baix la
considero alerta (perquè el Zamacois permet una nota més que els meus apunts).
                                    logA.add(new Log(i,4));
                                }
                            }
                            break;
                        case 1: //A
                            if(tOffset>0){ //surt per dalt
                                if(tOffset>1){
                                    cTessitures.add(new CorPuntual(i,veu,'e')); //la contralt més d'una nota per dalt
la considero error (cap de les dues referències ho permet)
                                    logE.add(new Log(i,3));
                                }else{
                                    cTessitures.add(new CorPuntual(i,veu,'a')); //la contralt una nota per dalt la
considero alerta (el Zamacois permet una nota més que els meus apunts)
                                    logA.add(new Log(i,3));
                                }
                            }else{ //surt per baix
                                cTessitures.add(new CorPuntual(i,veu,'e')); //la contralt per baix la considero error
(el Zamacois de fet permet menys, en aquest cas).
                                logE.add(new Log(i,3));
                            }
                            break;
                        case 2: //T
                            if(tOffset>0){ //surt per dalt
                                if(tOffset>1){
                                    cTessitures.add(new CorPuntual(i,veu,'e')); //el tenor més d'una nota per dalt el
considero error (cap de les dues referències ho permet)
```

```
                                logE.add(new Log(i,2));
                            }else{
                                cTessitures.add(new CorPuntual(i,veu,'a')); //el tenor una nota per dalt el
considero alerta (el Zamacois permet una nota més)
                                logA.add(new Log(i,2));
                            }
                        }else{ //surt per baix
                            if(tOffset<-1){
                                cTessitures.add(new CorPuntual(i,veu,'e')); //el tenor més d'una nota per baix el
considero error (cap de les dues referències ho permet)
                                logE.add(new Log(i,2));
                            }else{
                                cTessitures.add(new CorPuntual(i,veu,'a')); //el tenor una nota per baix el
considero alerta (el Zamacois permet una nota més).
                                logA.add(new Log(i,2));
                            }
                        }
                        break;
                    case 3: //B
                        if(tOffset>0){ //surt per dalt
                            if(tOffset>1){
                                cTessitures.add(new CorPuntual(i,veu,'e')); //el baix més d'una nota per dalt el
considero error (cap de les dues referències ho permet)
                                logE.add(new Log(i,1));
                            }else{
                                cTessitures.add(new CorPuntual(i,veu,'a')); //el baix una nota per dalt el
considero alerta (el Zamacois permet una nota més)
                                logA.add(new Log(i,1));
                            }
                        }else{ //surt per baix
                            if(tOffset<-1){
                                cTessitures.add(new CorPuntual(i,veu,'e')); //el baix més d'una nota per baix el
considero error (cap de les dues referències ho permet)
                                logE.add(new Log(i,1));
                            }else{
                                cTessitures.add(new CorPuntual(i,veu,'a')); //el baix una nota per baix el
considero alerta (el Zamacois permet una nota més).
                                logA.add(new Log(i,1));
                            }
                        }
                        break;
                    }
                }
            }
        }
        i++;
    }

    System.out.println("Comprovant creuaments verticals, separació i moviments paral·lels...");
    cCreuades.clear();
    cSeparades.clear();
    solapades.clear();
    cPar8.clear();
    cPar5.clear();
    cPar7.clear();
    Note itv; //per desar l'interval mentre comprovo coses.
    Note itv2; //per desar l'interval de l'acord següent.
    i=0;
    for(Note[] ac: esquemaEnNotes){ //per cada acord de la realització
        for(int veu=0;veu<4;veu++){ //començo per S... (i vaig baixant)
            if(ac[veu].nom!=-1){ // (si no hi ha nota m'estalvio el càlcul)
                for(int veu2=(veu+1);veu2<4;veu2++){ // ...i la comparo amb A (i amb les altres, també baixant)
                    if(ac[veu2].nom!=-1){ // (si no hi ha nota amb què comparar també m'estalvio el càlcul)
                        itv = H.getInterval(ac[veu2], ac[veu]); //l'interval el calculo cap amunt, que serà més
pràctic.

                        // VEUS ADJACENTS
                        if(Math.abs(veu2-veu)==1){
                            if(H.compost(itv)>7){ //si hi ha més d'una octava //TODO potser mirar què passa en
l'improbable cas de tenir 8A i ajustar codi en conseqüència (tot i que probablement no passa mai).
                                if(veu2!=3){ //i no és amb el baix
                                    System.out.println("- Més d'una octava a "+i+" (entre "+H.rVeus[veu2]+" i
"+H.rVeus[veu]+")");
                                    cSeparades.add(new CorPuntual(i,veu2,H.compost(itv))); //sempre marcaré a la
veu de sota i se sobreentén que és pujant. L'auxiliar l'aprofito per guardar la separació de les notes.
                                    logE.add(new Log(i,5));
                                }
                            }else if(H.compost(itv)<0){ //si estan creuades (si surt negatiu havent fet
l'interval expressament ascendent és que està creuat). TODO potser mirar què passa en l'improbable cas que sigui un
unísson disminuït i ajustar.
```

```
                                System.out.println("- Veus creuades a "+i+" ("+H.rVeus[veu2]+" amb
"+H.rVeus[veu]+")");
                                cCreuades.add(new CorPuntual(i,veu2,H.compost(itv))); //com abans, sempre serà amb
la següent superior.
                                if(veu%2!=0){
                                    logE.add(new Log(i,6));
                                }else{
                                    logA.add(new Log(i,1006));
                                }
                                //TODO afegir un botó al menú que ajusti les veus creuades dins el mateix
pentagrama (que les intercanvii entre elles)
                            }else if(H.compost(itv)<2 && veu%2==0){ //si estan tan a prop i són al mateix
pentagrama...
                                System.out.println("- Notes solapades a "+i+" ("+H.rVeus[veu2]+" amb
"+H.rVeus[veu]+")");
                                solapades.add(new CorPuntual(i, veu, H.compost(itv))); //al valor auxiliar deso
l'interval (desplaçaré més o menys segons aquest).
                            }
                        }

                        // CAS GENERAL
                        if(i<esquemaEnNotes.size-1){ //m'asseguro de no sortir-me de la matriu pel final (no
buscaré moviment paral·lel que comenci a l'últim acord, perquè no hi ha res després xD)
                            if(esquemaEnNotes.get(i+1)[veu].nom!=-1 && esquemaEnNotes.get(i+1)[veu2].nom!=-1){
//miro que el següent acord també tingui notes (perquè si no igualment no podré comparar).
                                itv2 = H.getInterval(esquemaEnNotes.get(i+1)[veu2], esquemaEnNotes.get(i+1)[veu]);
                                if(H.compost(itv)==H.compost(itv2)){ //penso que m'estalvio càlcul si el primer
que miro és que siguin el mateix interval (no miro el qualificatiu perquè m'interessarà marcar les quintes amb
qualificatius diferents).
                                    if(itv.nom==0){ //si l'interval és d'octava
                                        System.out.println("- Octaves seguides a "+i);
                                        if(H.getInterval(ac[veu], esquemaEnNotes.get(i+1)[veu]).so==0 &&
H.getInterval(ac[veu2], esquemaEnNotes.get(i+1)[veu2]).so==0){
                                            System.out.println("- - Són lligades. Correcte."); //Entenc que si és
tot lligat no passa res (Això ni ho guardo com a correcció).
                                        }else{
                                            cPar8.add(new CorPuntual(i,veu2,veu,'e'));
                                            logE.add(new Log(i,8));
                                        }
                                    }else if(itv.nom==4){ //si és de quinta
                                        System.out.println("- Quintes seguides a "+i);
                                        if(itv.so!=itv2.so){ //quintes amb qualificatius diferents. Correcte.
                                            System.out.println("- - Qualidicatius diferents. Correcte."); //no ho
guardo
                                            //cPar5.add(new CorPuntual(i,veu2,veu,'c'));
                                        }else if(ac[veu]==esquemaEnNotes.get(i+1)[veu2] ||
ac[veu2]==esquemaEnNotes.get(i+1)[veu]){ //comparteixen un so (Com les quintes de Rimsky). Correcte.
                                            System.out.println("- - Comparteixen un so. Correcte.");
                                            cPar5.add(new CorPuntual(i,veu2,veu,'c'));
                                        }else if(H.getInterval(ac[veu], esquemaEnNotes.get(i+1)[veu]).so==1){ //es
mouen mig to. Correcte.
                                            System.out.println("- - Es mouen mig to. Correcte."); //no ho guardo
                                        }else if(H.getInterval(ac[veu], esquemaEnNotes.get(i+1)[veu]).so==0 &&
H.getInterval(ac[veu2], esquemaEnNotes.get(i+1)[veu2]).so==0){
                                            System.out.println("- - Van lligades. Correcte."); //això no cal
guardar-ho, tampoc.
                                        }else{
                                            cPar5.add(new CorPuntual(i,veu2,veu,'e'));
                                            logE.add(new Log(i,10));
                                        }
                                    }else if(itv.nom==6){ //si és de sèptima
                                        System.out.println("- Sèptimes seguides a "+i);
                                        if(H.getInterval(ac[veu], esquemaEnNotes.get(i+1)[veu]).so==0 &&
H.getInterval(ac[veu2], esquemaEnNotes.get(i+1)[veu2]).so==0){
                                            System.out.println("- - Van lligades. Correcte."); //i, altra vegada no
ho guardo, perquè és irrellevant.
                                        }else{ //Normal. Error.
                                            cPar7.add(new CorPuntual(i,veu2,veu,'e'));
                                            logE.add(new Log(i,9));
                                        }
                                    }
                                }
                            }
                        }

                    }
                }
            }
        }
        i++;
```

```
        }

        System.out.println("Comprovant lligadures, creuaments horitzontals i sèptimes...");
        cPrep7.clear();
        cRes7.clear();
        cRI7.clear();
        lligadures.clear();
        cCreuadesH.clear();
        cAug.clear();
        cDim.clear();
        cItv.clear();
        int veuAux;
        int j;
        CorPuntual cAux; //valor auxiliar de correcció per guardar la correcció provisional durant les cerques que
poden trobar diferents opcions.
        for(int veu=0;veu<4;veu++){ //Correccions horitzontals
            i=0;
            for(Note[] ac: esquemaEnNotes){ //dins de cada veu recorro l'esquema d'esquerra a dreta
                if(ac[veu].nom!=-1){ //només corregeixo quan hi ha nota

                    //detecció de lligadures i acords augmentats i disminuïts
                    if(i<esquema.size-1){ //si hi ha acord següent
                        if(esquema.get(i+1).satb[veu]!=0){ //i hi ha nota al següent acord
                            if(ac[veu].equals(esquemaEnNotes.get(i+1)[veu])){ //s'ha de fer amb equals perquè "=="
compara referències, no paràmetres (mira si és la mateixa instància, no si les dues instàncies són equivalents).
                                System.out.println("- Creant lligadura a "+i+"-"+H.rVeus[veu]);
                                lligadures.add(new CorPuntual(i,veu));
                            }else{ //no va lligada. Comprovo A i d.
                                itv = H.getInterval(ac[veu], esquemaEnNotes.get(i+1)[veu]); //faig l'interval amb la
nota següent
                                if(H.augmentat(itv)){ //si és augmentat
                                    if(H.compost(itv)!=0){ //si és un unísson augmentat (un cromatisme) no passa res.
                                        System.out.println("- Interval augmentat a "+i+"-"+H.rVeus[veu]+"
("+(itv.nom+1)+"A)");
                                        cAug.add(new CorPuntual(i,veu)); //marco l'error (no hi ha augmentats bons,
així que no guardo el char
                                        logE.add(new Log(i, 11));
                                    }
                                }else if(H.disminuit(itv)){ // (i) - dim - (i+1)
                                    if(H.compost(itv)!=1){ //si és enharmonia (2d) no m'interessa (ho comprovaré en un
altre moment).
                                        System.out.println("- Interval disminuït a "+i+"-"+H.rVeus[veu]+"
("+(itv.nom+1)+"d)");
                                        j=1;
                                        while(i+1+j<esquema.size){ //asseguro els límits de la cerca
                                            if(esquemaEnNotes.get(i+1)[veu].equals(esquemaEnNotes.get(i+1+j)[veu])){ //
(i+1) és igual que (i+1+j)
                                                j++; //si la nota s'ha mantingut, s'aplaça la resolució, així que
segueixo buscant
                                            }else if(esquemaEnNotes.get(i+1+j)[veu].nom!=-1){ // aquí (i+1+j) ja és
diferent de (i+1), així que comprovo l'interval entre aquests
                                                itv2 = H.getInterval(esquemaEnNotes.get(i+1)[veu],
esquemaEnNotes.get(i+1+j)[veu]);
                                                if(H.compost(itv2)==-Math.signum(itv.so)){ //segona (compost==1) en la
direcció contrària a l'interval disminuït TODO investigar les normes al respecte (la normativa que tinc no
contempla tornada per cromatisme i no sé si estaria bé)
                                                    System.out.println("- - Correcte. Direcció contrària per graus
conjunts.");
                                                }else{
                                                    System.out.println("- - Resolució incorrecta.");
                                                    cDim.add(new CorPuntual(i+1,veu,'e')); //guardo la correcció a
(i+1) perquè és la nota que s'ha de resoldre
                                                    logE.add(new Log(i+1,12));
                                                }
                                                j=-j; //marca de trobat
                                                break;
                                            }else{ //no hi havia nota i he quedat penjat.
                                                System.out.println("- - Pendent de resoldre.");
                                                cDim.add(new CorPuntual(i+1,veu,'a'));
                                                logA.add(new Log(i+1,12));
                                                j=-j; //marca de trobat
                                                break;
                                            }
                                        }
                                        if(j>0){ //no he trobat res (vol dir que s'ha acabat l'esquema a base de
lligar la nota eternament xD)
                                            System.out.println("- - Impossible resoldre.");
                                            cDim.add(new CorPuntual(i+1,veu,'e'));
                                            logE.add(new Log(i+1,12));
                                        }
```

```
                                }
                            }else if(!H.mMJ(itv)){ //si és més que augmentat o menys que disminuït (perquè A i d
ja els he contemplat abans)
                                System.out.println("- Trobat un interval estrany a "+i+"-"+H.rVeus[veu]);
                                cItv.add(new CorPuntual(i,veu,'e'));
                                logE.add(new Log(i,0));
                            }

                            //comprovo creuaments horitzontals
                            if(itv.nom>0 && veu>0){ //l'interval amb la nota següent és ascendent (si és la
soprano m'ho salto, perquè no té ningú a dalt)
                                if(ac[veu-1].nom!=-1){ //tinc una nota a la veu superior (superior és menys
número) que em limitarà el moviment
                                    itv2 = H.getInterval(ac[veu-1], esquemaEnNotes.get(i+1)[veu]); //del límit a la
meva nota.
                                    if(itv2.so>0){ //ascendent (he creuat el límit)
                                        System.out.println("- Creuament horitzontal ascendent a "+(i+1)+"-
"+H.rVeus[veu]+"/"+H.rVeus[veu-1]);
                                        cCreuadesH.add(new CorPuntual((i+1),veu,(veu-1),'+')); // (i+1) perquè és
on passa el creuament, i després guardo la veu que creua i la veu creuada.
                                        logE.add(new Log(i+1,7));
                                    }
                                }
                            }else if(itv.nom<0 && veu<3){ //l'interval amb la nota següent és descendent (si és
el baix m'ho salto, perquè no té ningú a baix)
                                if(ac[veu+1].nom!=-1){ //tinc nota a la veu inferior que em limitarà el moviment
                                    itv2 = H.getInterval(ac[veu+1], esquemaEnNotes.get(i+1)[veu]); //del límit a la
meva nota
                                    if(itv2.so<0){ //descendent (he creuat el límit)
                                        System.out.println("- Creuament horitzontal descendent a "+(i+1)+"-
"+H.rVeus[veu]+"/"+H.rVeus[veu+1]);
                                        cCreuadesH.add(new CorPuntual((i+1),veu,(veu+1),'-')); //acord, veu que
creua, veu creuada.
                                        logE.add(new Log(i+1,7));
                                    }
                                }
                            }
                        }// fi "no va lligada"
                    }
                }

                //Correcció de sèptimes
                if(esquema.get(i).satb[veu]==9 || esquema.get(i).satb[veu]==7 && !tincNotes.get(i)[4]){ //si és
la 9 o bé és la 7 i no tinc 9 (si és la sèptima que compta dins l'acord)
                    System.out.println("- Sèptima trobada a "+i+"-"+H.rVeus[veu]);
                    if(F.keep(H.ascendent(H.getInterval(H.getChordNote(Pant.esbos.get(i),1),
H.getChordNote(Pant.esbos.get(i),7)))).nom==6 && F.lastNote.so==11){
                        System.out.println("- Sèptima Major");
                        septMajor=true; //marco prèviament si la sèptima és major per ajustar la correcció en
conseqüència.
                    }else{
                        septMajor=false;
                    }

                    System.out.println("- Comprovant preparació...");
                    if(i>0){ //en principi no hauria d'entrar mai igualment perquè el 1r acord no té mai sèptima,
però per si de cas canvio coses més endavant.
                        if(esquemaEnNotes.get(i-1)[veu].nom!=-1){ //comprovo abans que tinc nota anterior
                            itv = H.getInterval(esquemaEnNotes.get(i-1)[veu], ac[veu]); //interval des de la nota
anterior
                            if(H.compost(itv)==0 && itv.so==0 || (!septMajor) && Math.abs(H.compost(itv))<2){ //ve
lligada o bé ve per graus conjunts i no és major.
                                if(septMajor){
                                    System.out.println("- - Ve lligada. Correcte."); //És l'alternativa normal. No
la guardo.
                                }else{
                                    System.out.println("- - Lligada o per graus conjunts. Correcte.");
                                }
                            }else if(H.compost(itv)>0 || (septMajor && Math.abs(H.compost(itv))<2)){ //salta,
però cap amunt (o era sèptima major que venia per graus conjunts en lloc de lligada)
                                System.out.println("- - Salt ascendent. Comprovant fonamental...");
                                cAux = new CorPuntual(42,42,'i'); //marco irrellevant com a posada a zero (els
números són arbitraris).
                                for(int veu2=0;veu2<4;veu2++){ //cerca de la fonamental corresponent
                                    if(veu2!=veu){
                                        if(esquema.get(i).satb[veu2]==1 || tincNotes.get(i)[4] &&
esquema.get(i).satb[veu2]==3){ //si trobo la fonamental
                                            if(esquemaEnNotes.get(i-1)[veu2].nom!=-1){ //miro que a l'acord
anterior aquesta veu també tingui nota
                                                itv2 = H.getInterval(esquemaEnNotes.get(i-1)[veu2], ac[veu2]);
```

```
//calculo l'interval per la veu de la fonamental
                                        if(H.compost(itv2)==0 && itv2.so==0 || (!septMajor) &&
Math.abs(H.compost(itv2))<2){ //la fonamental entra lligada o per graus conjunts (només lligada si és 7M)
                                            if(septMajor){
                                                System.out.println("- - Fonamental lligada. Correcte.");
                                            }else{
                                                System.out.println("- - Fonamental lligada o per graus
conjunts. Correcte.");
                                            }
                                            cAux.cor = 'c'; //marco correcte. No guardo veus ni res perquè
no guardaré correcció.
                                            break; //com que ja he trobat una solució correcta, surto.
                                        }else{ //la fonamental també salta. Malament.
                                            System.out.println("- - Fonamental mal preparada. Error.");
                                            if(cAux.cor!='a' && cAux.cor!='e'){ //només sobreescric amb un
error si no hi havia res
                                                cAux = new CorPuntual(i,veu,veu2,'e'); //deixo les dades a
la variable auxiliar de correcció
                                            }
                                        }
                                    }else{ //tinc fonamental, però de moment no ve d'enlloc
                                        System.out.println("- - Fonamental pendent de preparar.");
                                        if(cAux.cor!='a'){ //només sobreescric si no hi havia res o si hi
havia un error (perquè aquesta via dóna una possibilitat millor que l'error pel fet d'estar pendent)
                                            cAux = new CorPuntual(i,veu,veu2,'a');
                                        }
                                    }
                                }
                            }
                            if(cAux.cor=='i'){ //vol dir que no he trobat res a la cerca.
                                System.out.println("- - Fonamental no ubicada.");
                                cPrep7.add(new CorPuntual(i,veu,veu,'a')); //per coherència amb la quantitat
de dades d'alguns valors dins cPrep7, omplo també la dada auxiliar les vegades que no em cal
                                if(septMajor){
                                    logA.add(new Log(cAux.ac,31));
                                }else{
                                    logA.add(new Log(cAux.ac,23));
                                }
                            }else if(cAux.cor=='e' || cAux.cor=='a'){
                                cPrep7.add(new CorPuntual(cAux.ac,cAux.veu,cAux.aux,cAux.cor)); //passo les
dades de l'auxiliar a la correcció d'aquesta sèptima.
                                if(cAux.cor=='e'){
                                    if(septMajor){
                                        logE.add(new Log(cAux.ac,31));
                                    }else{
                                        logE.add(new Log(cAux.ac,23));
                                    }
                                }else{
                                    if(septMajor){
                                        logA.add(new Log(cAux.ac,31));
                                    }else{
                                        logA.add(new Log(cAux.ac,23));
                                    }
                                }
                            }
                        }else{ //la sèptima salta cap a baix.
                            System.out.println("- - Salt descendent. Incorrecte.");
                            cPrep7.add(new CorPuntual(i,veu,veu,'e'));
                            logE.add(new Log(i,231));
                        }
                    }else{ //no hi ha nota anterior.
                        System.out.println("- - La nota anterior encara no hi és.");
                        cPrep7.add(new CorPuntual(i,veu,veu,'a'));
                        if(septMajor){
                            logA.add(new Log(i,31));
                        }else{
                            logA.add(new Log(i,23));
                        }
                    }
                }

                System.out.println("- Comprovant resolució...");
                if(i<esquema.size-1){ //asseguro els marges
                    if(esquemaEnNotes.get(i+1)[veu].nom!=-1){ //comprovo que hi ha nota al darrere
                        itv = H.getInterval(ac[veu], esquemaEnNotes.get(i+1)[veu]); //faig l'interval amb la
següent
                        if(H.compost(itv)==0){ //unísson...
                            System.out.println("- - Unisson. Comprovant qualificatiu...");
                            if(itv.so==0){ //mateixa nota.
```

```
                                System.out.println("- - Sèptima aguantada. Correcte.");
                            }else if(Math.abs(itv.so)<2){ //cromatisme.
                                System.out.println("- - Resolt per cromatisme. Correcte."); //TODO mirar si
passa res per fer el cromatisme avall (fa pinta que és correcte, però ho tenies pendent als apunts)
                            }
                        }else if(H.compost(itv)==-1){ //segona descendent
                            System.out.println("- - Resol baixant. Correcte.");
                        }else{ //no resol. Comprovo possibilitat de RI.
                            System.out.println("- - Sèptima no resolta. Comprovant RI.");
                            for(veuAux=0;veuAux<4;veuAux++){ //cerca d'una nota que permeti la RI
                                if(veuAux!=veu){ //comprovo les altres tres veus
                                    if(esquemaEnNotes.get(i+1)[veuAux].nom!=-1){ //si hi ha nota...
                                        itv2 = H.getInterval(ac[veu], esquemaEnNotes.get(i+1)[veuAux]);
                                        if(itv2.nom==0 && Math.abs(itv2.so%12)<2 || H.descendent(itv2).nom==-
1){ //comprovo que en alguna de les altres veus sí que hi ha la nota que permetria les condicions anteriors.
                                            if(cRI7.size<maxRI7){ //comprovo el límit de Resolucions irregulars.
                                                System.out.println("- - Resolució irregular correcta.");
                                                cRI7.add(new CorPuntual(i,veu,veuAux,'n')); //guardo les dues
veus per saber d'on a on va la línia de punts.
                                            }else{ //si ja m'he passat de la quantitat permesa,
                                                System.out.println("- - Superat el límit de RI/esquema.");
                                                cRI7.add(new CorPuntual(i,veu,veuAux,'e'));
                                                logE.add(new Log(i,251));
                                            }
                                            veuAux=-veuAux; //marca de trobat (deixa veuAux entre 0 i -3)
                                            break;
                                        }
                                    }
                                }
                            }
                            if(veuAux>0){ //no he trobat RI
                                if(tincPle.get(i)){ //si ja tinc l'acord ple, marco error.
                                    System.out.println("- - No hi ha RI. Incorrecte.");
                                    cRes7.add(new CorPuntual(i,veu,'e'));
                                    logE.add(new Log(i,25));
                                }else{ //si encara falten notes, deixo l'alerta.
                                    System.out.println("- - RI pendent.");
                                    cRes7.add(new CorPuntual(i,veu,'a'));
                                    logA.add(new Log(i,25));
                                }
                            }
                        }
                    }else{ //si encara no hi ha nota al darrere (vol dir que no comprovaré RI fins que sàpiga
que no s'ha posat resolució normal).
                        System.out.println("- - Pendent de resoldre."); //marco alerta, com amb la preparació
                        cRes7.add(new CorPuntual(i,veu,'a'));
                        logA.add(new Log(i,24));
                    }
                }
            }
        }
        i++;
    }
}
System.out.println("RI x"+cRI7.size);

System.out.println("Coprovant realització dels acords en 6/4...");
c64PrepB.clear();
c64ResB.clear();
c64Prep4.clear();
c64Res4.clear();
i=0;
for(Note[] ac: esquemaEnNotes){ //per cada acord
    if(!tincNotes.get(i)[4] && !tincNotes.get(i)[3] && esquema.get(i).satb[3]==5){ //no hi ha sèptima (ni de
sensible) i estic en segona inversió
        System.out.println("- Trobat 6/4 a "+i);

        System.out.println("- Comprovant preparació del baix...");
        if(i==0){ //si estic al primer acord segur que no he pogut preparar-lo, així que malament.
            System.out.println("- - Impossible preparar el baix.");
            c64PrepB.add(new CorPuntual(i, 3,'e'));
            logE.add(new Log(i,26));
        }else{
            if(esquemaEnNotes.get(i-1)[3].nom!=-1){ //hi ha nota al baix anterior
                itv = H.getInterval(esquemaEnNotes.get(i-1)[3], ac[3]); //interval entre els dos baixos
                if(Math.abs(H.compost(itv))<2){ //lligat o per graus conjunts
                    System.out.println("- - Baix ben preparat."); //no guardo res
                }else{
                    System.out.println("- - Baix mal preparat.");
                    c64PrepB.add(new CorPuntual(i, 3,'e'));
```

```
                        logE.add(new Log(i,26));
                    }
                }else{
                    System.out.println("- - Baix pendent de preparar.");
                    c64PrepB.add(new CorPuntual(i, 3,'a'));
                    logA.add(new Log(i,26));
                }
            }

            System.out.println("- Comprovant resolució del baix...");
            if(i==esquema.size-1){ //si és l'últim segur que no el puc resoldre
                System.out.println("- - Impossible resoldre el baix.");
                c64ResB.add(new CorPuntual(i, 3,'e'));
                logE.add(new Log(i,26));
            }else{
                if(esquemaEnNotes.get(i+1)[3].nom!=-1){ //hi ha nota al baix següent
                    itv = H.getInterval(ac[3], esquemaEnNotes.get(i+1)[3]); //interval entre els dos baixos
                    if(Math.abs(H.compost(itv))<2){ //lligat o per graus conjunts
                        System.out.println("- - Baix ben resolt."); //no guardo res
                    }else{
                        System.out.println("- - Baix mal resolt.");
                        c64ResB.add(new CorPuntual(i, 3,'e'));
                        logE.add(new Log(i,26));
                    }
                }else{
                    System.out.println("- - Baix pendent de resoldre.");
                    c64ResB.add(new CorPuntual(i, 3,'a'));
                    logA.add(new Log(i,26));
                }
            }

            System.out.println("- Comprovant preparació de la quarta...");
            if(i==0){ //mateixa lògica que amb el baix.
                System.out.println("- - Impossible preparar la quarta.");
                c64Prep4.add(new CorPuntual(i, 3,'e'));
                logE.add(new Log(i,26));
            }else{
                cAux = new CorPuntual(42,42,'i'); //marco diferent veu i aux.
                for(int veu=0;veu<3;veu++){ //em salto comprovar el baix, que ja l'he fet abans.
                    if(esquema.get(i).satb[veu]==1){ //trobo la fonamental (que serà la quarta del 6/4)
                        System.out.println("- - Quarta trobada a "+H.rVeus[veu]);
                        if(esquemaEnNotes.get(i-1)[veu].nom!=-1){ //si hi ha nota a l'acord anterior
                            itv = H.getInterval(esquemaEnNotes.get(i-1)[veu], ac[veu]); //interval entre les dues
                            if(Math.abs(H.compost(itv))<2){ //lligada o per graus conjunts
                                System.out.println("- - Preparació correcta.");
                                cAux.cor = 'c'; //marco que he trobat abans de sortir.
                                break;
                            }else{
                                System.out.println("- - Preparació incorrecta.");
                                if(cAux.cor!='a' && cAux.cor!='e'){
                                    cAux = new CorPuntual(i,veu,veu,'e');
                                }else if(cAux.cor=='e'){
                                    cAux.aux = (byte) veu; //si ja tenia una quarta amb error i resulta que estan
malament les dues, afegeixo la info de la segona a "aux".
                                }
                            }
                        }else{
                            System.out.println("- - Pendent de preparar.");
                            if(cAux.cor!='a'){
                                cAux = new CorPuntual(i,veu,veu,'a');
                            }else{
                                cAux.aux = (byte) veu; //mateix concepte. Guardo la veu de la segona quarta amb
alerta a "aux".
                            }
                        }
                    }
                }
                if(cAux.cor=='i'){
                    System.out.println("- - No s'ha trobat cap quarta.");
                    //Aquí no faig res perquè ha suprimit la fonamental i ja rebrà per algun altre lloc
                }else if(cAux.cor=='e' || cAux.cor=='a'){
                    c64Prep4.add(new CorPuntual(cAux.ac,cAux.veu,cAux.aux,cAux.cor));
                    if(cAux.cor=='e'){
                        if(cAux.veu!=cAux.aux){ //hi ha dues quartes
                            logE.add(new Log(cAux.ac,261));
                        }else{
                            logE.add(new Log(cAux.ac,26));
                        }
                    }else{
                        if(cAux.veu!=cAux.aux){ //hi ha dues quartes
```

```
                            logA.add(new Log(cAux.ac,261));
                        }else{
                            logA.add(new Log(cAux.ac,26));
                        }
                    }
                }
                System.out.println("- Comprovant resolució de la quarta...");
                cAux = new CorPuntual(42,42,'i');
                for(int veu=0;veu<3;veu++){ //com abans, el baix ja està fet.
                    if(esquema.get(i).satb[veu]==1){ //trobo una quarta
                        System.out.println("- - Quarta trobada a "+H.rVeus[veu]);
                        if(i<esquema.size-1){ //miro que no sigui l'últim acord
                            if(esquemaEnNotes.get(i+1)[veu].nom!=-1){ //si hi ha nota a l'acord següent
                                itv = H.getInterval(ac[veu], esquemaEnNotes.get(i+1)[veu]);
                                if(Math.abs(H.compost(itv))<2){ //lligada o per graus conjunts
                                    System.out.println("- - Resolució correcta");
                                    cAux.cor = 'c'; //marca de trobat
                                    break;
                                }else{
                                    System.out.println("- - Resolució incorrecta.");
                                    if(cAux.cor!='a' && cAux.cor!='e'){
                                        cAux = new CorPuntual(i,veu,veu,'e');
                                    }else if(cAux.cor=='e'){
                                        cAux.aux = (byte) veu; //si trobo una segona quarta amb error afegeixo la dada
de la seva veu a "aux"
                                    }
                                }
                            }else{
                                System.out.println("- - Pendent de resoldre.");
                                if(cAux.cor!='a'){
                                    cAux = new CorPuntual(i,veu,veu,'a');
                                }else{
                                    cAux.aux = (byte) veu; //mateixa filosofia.
                                }
                            }
                        }else{ //si és l'últim acord ja sé que serà impossible
                            System.out.println("- - Impossible resoldre la quarta.");
                            if(cAux.cor!='e'){ //no cal que comprovi si he guardat una alerta, perquè si és l'últim
acord totes tindran el mateix problema.
                                cAux = new CorPuntual(i,veu,veu,'e');
                            }else if(cAux.cor=='e'){
                                cAux.aux = (byte) veu; //mateixa filosofia.
                            }
                        }
                    }
                }
                if(cAux.cor=='i'){
                    System.out.println("- - No s'ha trobat cap quarta.");
                    //Aquí no faig res perquè ha suprimit la fonamental i ja rebrà per un altre lloc
                }else if(cAux.cor=='e' || cAux.cor=='a'){
                    c64Res4.add(new CorPuntual(cAux.ac,cAux.veu,cAux.aux,cAux.cor));
                    if(cAux.cor=='e'){
                        if(cAux.veu!=cAux.aux){ //hi ha dues quartes
                            logE.add(new Log(cAux.ac,261));
                        }else{
                            logE.add(new Log(cAux.ac,26));
                        }
                    }else{
                        if(cAux.veu!=cAux.aux){ //hi ha dues quartes
                            logA.add(new Log(cAux.ac,261));
                        }else{
                            logA.add(new Log(cAux.ac,26));
                        }
                    }
                }
            }
            i++;
        }

        System.out.println("Comprovant realització de les cadències...");
        cCad.clear();
        byte mvtCT;
        for(Cadencia c: Pant.cadencies){ //vaig directament a buscar les cadències.
            if(!c.cp){ //CT
                if(!c.ds){ //no tinc III- a la cadència TODO comprovar que no s'hagi de fer res especial amb el III-
(que només cal resoldre la DS)
                    System.out.println("- Cadència trencada a "+(c.n));
                    if(tincPle.get(c.n+c.w)){ //té totes les veus decidides (si no, no tindré dades de l'inversió)
                        if(inversions.get(c.n+c.w)<5){ //el IV/VI no porta sèptima (les inversions amb sèptima les he
```

```
guardat sumant 10)
                        if(complet.get(c.n+c.w)){ //i surten totes les notes del IV/VI
                            if(Pant.esbos.get(c.n+c.w).grau==3 && duplicades.get(c.n+c.w)==5
                            || Pant.esbos.get(c.n+c.w).grau==5 && duplicades.get(c.n+c.w)==3){ //he duplicat la
tònica del IV/VI
                                System.out.println("- - Tònica duplicada. Comprovant moviments.");
                                if(tincPle.get(c.n+c.w-1)){ //el V té totes les veus decidides
                                    if(complet.get(c.n+c.w-1)){ //i hi surten totes les notes
                                        mvtCT = 0;
                                        for(int veu=0;veu<4;veu++){ //recorro les veus sumant el nom dels
intervals (per mirar si s'han fet els mínims moviments possibles).
                                            itv = H.getInterval(esquemaEnNotes.get(c.n+c.w-1)[veu],
esquemaEnNotes.get(c.n+c.w)[veu]);
                                            mvtCT += Math.abs(itv.nom);
                                        }
                                        if(inversions.get(c.n+c.w-1)<5){ //el V no porta sèptima
                                            if(Pant.esbos.get(c.n+c.w).grau==5 && mvtCT<6
                                            || Pant.esbos.get(c.n+c.w).grau==3 && mvtCT<5){
                                                System.out.println("- - Resolució òptima."); //si no se m'escapa
res, al menys (i aplica per sèptima de sensible, en principi)
                                                cCad.add('n'); //correcte
                                            }else{
                                                System.out.println("- - Resolució no òptima. Alerta.");
                                                cCad.add('a'); //hi ha millors maneres de fer-ho, però com no sé el
context, deixo alerta i prou
                                                logA.add(new Log(i,1017));
                                            }
                                        }else{
                                            if(Pant.esbos.get(c.n+c.w).grau==5 && mvtCT<5
                                            || Pant.esbos.get(c.n+c.w).grau==3 && mvtCT<4){
                                                System.out.println("- - Resolució òptima."); //si no se m'escapa
res, al menys
                                                cCad.add('n'); //correcte
                                            }else{
                                                System.out.println("- - Resolució no òptima. Alerta.");
                                                cCad.add('a');
                                                logA.add(new Log(i,1017));
                                            }
                                        }
                                    }else{
                                        System.out.println("- - V incomplet.");
                                        cCad.add('e'); //TODO mirar si això hauria de passar a alerta (no sé com de
greu és).
                                        logE.add(new Log(i,0));
                                    }
                                }else{
                                    System.out.println("- - V pendent de realitzar.");
                                    cCad.add('i');
                                }
                                //TODO mirar com comprovar "resoldre per sentit comú/mínims moviments possibles"
xD
                            }else{
                                System.out.println("- - Tònica no duplicada. Error.");
                                cCad.add('e');
                                logE.add(new Log(i,17));
                            }
                        }else{
                            System.out.println("- - Acord de resolució incomplet. Error."); //TODO investigar què
passa amb la quinta. (De moment deixo impossible incomplet).
                            cCad.add('e');
                            logE.add(new Log(i,0));
                        }
                    }else{
                        System.out.println("- - Porta sèptima. No aplica.");
                        cCad.add('n'); //posar a normal si estan plens o el que sigui
                    }
                }else{
                    System.out.println("- - Pendent de realitzar.");
                    cCad.add('i');
                }
            }else{ //tinc III-
                cCad.add('n'); //deixo per defecte (no corregeixo res, de moment, aquí).
            }
        }else{ //CP
            //TODO mirar què fa que estigui bé i què vull comprovar abans de donar-la per bona
            cCad.add('n'); //de moment marco irrellevant fins que la corregeixi
        }
    }
    System.out.println(cCad);
```

```
        cRecompte(); //faig el recompte de quantitats d'errors i alertes.

        //TODO Comprovacions de melodia
        //TODO Comprovacions alt segona 6-7 (menor)
        //...

    } //fi corregeix

    public void cRecompte(){ //TODO anar actualitzant quan vagi afegint més correccions.
        nErrors = 0;
        nAlertes = 0;
        for(Character c: cComplet){
            if(c=='e'){
                nErrors++;
            }else if(c=='a'){
                nAlertes++;
            }
        }
        for(Character c: cDuplicades){
            if(c=='e'){
                nErrors++;
            }else if(c=='a'){
                nAlertes++;
            }
        }
        for(Character c: cInv){
            if(c=='e'){
                nErrors++;
            }else if(c=='a'){
                nAlertes++;
            }
        }
        for(CorPuntual c: cTessitures){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }

        for(CorPuntual c: cCreuades){
            if(c.veu%2==0){ //és la que allà era veu2, així que la comparació va al revés
                nErrors++;
            }else{
                nAlertes++;
            }
        }

        nErrors += cCreuadesH.size; //no està marcat amb 'e', però són errors.
        nErrors += cSeparades.size;

        for(CorPuntual c: cPar8){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }
        for(CorPuntual c: cPar5){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }
        for(CorPuntual c: cPar7){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }
        for(CorPuntual c: cPrep7){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }
        for(CorPuntual c: cRes7){
```

```
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }
        for(CorPuntual c: cRI7){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }
        for(CorPuntual c: c64PrepB){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }
        for(CorPuntual c: c64ResB){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }
        for(CorPuntual c: c64Prep4){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }
        for(CorPuntual c: c64Res4){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }

        nErrors += cAug.size;

        for(CorPuntual c: cDim){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }
        for(CorPuntual c: cItv){
            if(c.cor=='e'){
                nErrors++;
            }else if(c.cor=='a'){
                nAlertes++;
            }
        }
        for(Character c: cCad){
            if(c=='e'){
                nErrors++;
            }else if(c=='a'){
                nAlertes++;
            }
        }
        System.out.println("S'han trobat "+nErrors+" errors i "+nAlertes+" alertes.");

        errorAct = 0; //poso a zero per no sortir-me dels límits anant a veure una correcció que ja no hi és
        alertaAct = 0;
        System.out.println("- Desats "+logE.size+" registres d'errors i "+logA.size+" d'alertes.");
    }

    public static boolean hiHaCor(Array<CorPuntual> cp, int ac, int veu){ //comprova si hi ha correcció puntual
en una llista de correccions puntuals per a un acord i veu concrets.
        for(CorPuntual c: cp){
            if(c.ac==ac && c.veu==veu){
                F.lastCor = c; //aprofito i guardo la correcció que he trobat que hi havia.
                return true;
            }
        }
```

```
        return false; //només arriba fins aquí si no ha retornat res abans, així que ja va bé.
    }
}
```

A.1.23. [PIng.java] – Pantalla dels ingredients

```
package com.rusca8.hEditor;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.InputAdapter;
import com.badlogic.gdx.Screen;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.OrthographicCamera;
import com.badlogic.gdx.graphics.g2d.SpriteBatch;
import com.badlogic.gdx.math.Vector3;
import com.badlogic.gdx.utils.Align;
import com.badlogic.gdx.utils.Array;
import com.badlogic.gdx.utils.viewport.ExtendViewport;
import com.badlogic.gdx.utils.viewport.Viewport;
import com.rusca8.hEditor.A;
import com.rusca8.hEditor.BotQ;
import com.rusca8.hEditor.BotIA;
import com.rusca8.hEditor.BotIC;
import com.rusca8.hEditor.BotIR;
import com.rusca8.hEditor.F;
import com.rusca8.hEditor.H;
import com.rusca8.hEditor.IngAcord;
import com.rusca8.hEditor.IngCadencia;
import com.rusca8.hEditor.IngRes;
import com.rusca8.hEditor.Pant;
import com.rusca8.hEditor.Recursos;
import com.rusca8.hEditor.S;
import com.rusca8.hEditor.hEditor;

public class PIng extends InputAdapter implements Screen {
    hEditor app;

    OrthographicCamera camera; //explicats a la pantalla de l'esbós (Pant)
    Viewport encuadre;
    SpriteBatch lote;

    Vector3 click = new Vector3(0,0,0);

    int ii=0; //iterador global
    int idGen=1; //generador de números identificadors (s'incrementarà cada vegada que n'hagi d'assignar un)

    Array<IngRes> tipusRes = new Array<IngRes>(); //per desar els ingredients tipus resolució irregular
    static Array<BotIR> bTipusRes = new Array<BotIR>(); //per fer els botons associats

    Array<IngCadencia> tipusCadencia = new Array<IngCadencia>(); //per desar els ingredients tipus cadència
    static Array<BotIC> bTipusCadencia = new Array<BotIC>(); //per fer els botons associats

    Array<IngAcord> tipusAcord = new Array<IngAcord>(); //per desar els ingredients tipus acord
    static Array<BotIA> bTipusAcord = new Array<BotIA>(); //per fer els botons associats

    String desc="Un acord...";

    /**- - - Tria tipus d'ingredient - - -*/
    BotQ bTipusR, bTipusC, bTipusA; //RI, Cadència, Acord

    /**- - -Ingredient tipus resolució irregular- - -*/
    IngRes auxTR;
    BotIR bAuxTR;
    BotQ bRIs, bRI7;

    /**- - -Ingredient tipus cadència- - -*/
    int xIngCad;
    int yIngCad = (int) (Pant.screenH*0.6f);
    byte nTCPCad = 3; //número a partir del qual es dispara el tcp
    IngCadencia auxTC;
    BotIC bAuxTC;
    boolean pucDif=false;

    /**- - - Ingredient tipus acord - - -*/
    int xBarraE = 100; //posició de la barra de botons esquerra
    int xIngAc;
    int yIngAc = (int) (Pant.screenH*0.6f);

    Array<BotQ> bGraus = new Array<BotQ>();
```

```
    BotQ b7a, bDom, bInv, bAlt;
    boolean pucDom=false, pucAlt=false; //permisos dels botons
    byte altOp=0; //codi d'alteracions opcionals
    byte nTCPAc = 5; //número a partir del qual es dispara el tcp
    BotQ bDif, bEnll; //diferents, enllaçats
    boolean pucDifAc=false, pucDifInv=false, pucEnll=false; //com que per defecte hi ha x1 no es poden posar

    IngAcord auxTA; //auxiliar
    BotIA bAuxTA; //botó auxiliar associat


    /**- - - Pantalla dels ingredients - - -*/
    int yIngs=(int) (Pant.screenH*0.57f);
    int dIngs = 70; //distància entre ingredients
    int dMI=150; //distància del menú als ingredients (poso valor inicial, però la canviaré més a baix).

    int yClau=0;
    int hClau=0;

    int ingAct=-1; //id de l'ingredient actual (per seleccionar-los i esborrar-los i coses així).
    byte maxIngs=6; //màxim d'ingredients que deixo posar
    BotQ bAfegeix; //per afegir
    BotQ bEdita; //per editar
    BotQ bEAmb; //per enllaçar requeriments entre ells
    BotQ bEsborra; //per esborrar requeriments

    BotQ bFet; //botó per quan he acabat
    boolean vullEsbos; //per demanar confirmació
    BotQ bCancel; //per desfer quan estic demanant confirmació

    /**- - - Coses de botons en general - - -*/
    BotQ bMes, bMenys, bTCP; //afegir quantitat, restar quantitat, botó "tants com puguis" (surten a diverses
pantalles)
    boolean autoTCP = true; //marca per tal que es pugui desactivar l'aplicació automàtica del tcp

    boolean editant=false;
    BotQ bDesa;

    BotQ bTorna;
    BotQ bMenu;


    boolean triaTipus=false; //triant el ripus d'ingredient
    boolean triaRes = false; //triant un ingredient tipus resolució irregular
    boolean triaCadencia=false; //triant un ingredient tipus cadència
    boolean triaAcord=false; //triant un ingredient tipus acord

    boolean clickBot;

    /****** CONSTRUCTOR ******/
    public PIng(hEditor app){
        this.app = app;

        lote = new SpriteBatch();

        camera = new OrthographicCamera();
        encuadre = new ExtendViewport(0,Pant.screenH, camera);
        encuadre.apply(false);

        System.out.println("Inicialitzant pantalla dels requeriments...");

        /**- Botons -*/ //Compte, canvis de X a resize.
        //tria tipus
        bTipusR = new BotQ(Pant.realW*Pant.ratio*0.2f,Pant.screenH/2f,270,270,"RI"); //TODO fer mida proporcional a
amplada
        bTipusC = new BotQ(Pant.realW*Pant.ratio*0.5f,Pant.screenH/2f,270,270,"Cad");
        bTipusA = new BotQ(Pant.realW*Pant.ratio*0.8f,Pant.screenH/2f,270,270,"Acord");

        //ingredient tipus resolució irregular
        bRIs = new BotQ(Pant.realW*Pant.ratio*3/6f,Pant.screenH/2f,270,270,"RIs");
        bRI7 = new BotQ(Pant.realW*Pant.ratio*5/6f,Pant.screenH/2f,270,270,"RI7");

        //ingredient tipus acord (i compartits amb tipus cadència)
        for(int i=0;i<8;i++){
            bGraus.add(new BotQ(Pant.realW*Pant.ratio-75f, Pant.screenH*((8f-i)/9),i));
        }

        bMes   = new BotQ(xBarraE, Pant.screenH/2f+Pant.screenH*2/9f+14f,"+"); //les y d'alguns canvien segons
pantalla
        bMenys = new BotQ(xBarraE, Pant.screenH/2f+Pant.screenH/9f+14f,  "-");
```

```
        bTCP   = new BotQ(xBarraE, Pant.screenH/2f+4, "!");
        bDif  = new BotQ(xBarraE, Pant.screenH/2f-Pant.screenH/9f-6f, 130, 80,"Dif");
        bEnll   = new BotQ(xBarraE, Pant.screenH/2f-Pant.screenH*2/9f-6f, 130, 80,"Enll");

        xIngAc = (int) (((bDif.x+bDif.w/2f)+(bGraus.get(0).x-bGraus.get(0).w/2f))/2); //punt mig de l'espai entre
botons (canvia a resize)
        xIngCad = (int) (((bDif.x+bDif.w/2f)+(Pant.realW*Pant.ratio))/2); //punt mig de l'espai dret (canvia a
resize)

        b7a  = new BotQ(xIngAc-225, Pant.screenH*1/9f, 130, 80, "7/5"); //x dels botons a resize
        bDom = new BotQ(xIngAc-75,  Pant.screenH*1/9f, 130, 80, "DS" );
        bInv = new BotQ(xIngAc+75,  Pant.screenH*1/9f, 130, 80, "Inv");
        bAlt = new BotQ(xIngAc+225, Pant.screenH*1/9f, 130, 80, "Alt");

        bDesa = new BotQ(xBarraE,Pant.screenH/9f,120,120,"V"); //TODO canviar per un dibuix decent d'un tick


        //pantalla ingredients
        bAfegeix = new BotQ(100,100,120,120,"+");
        bEdita = new BotQ(250,100,120,120,"Edit");
        bEAmb = new BotQ(400,100,120,120,"Enll"); //TODO implementar el botó
        bEsborra = new BotQ(Pant.realW*Pant.ratio-100,100,120,120,"Esb");

        bFet = new BotQ((bEAmb.x+bEsborra.x)/2, 90, bEsborra.x-bEAmb.x-bEAmb.w*2, 80, "Fet"); //canvis segons
botons i a resize
        bCancel = new BotQ(Pant.realW*Pant.ratio*4/5f,90,Pant.realW*Pant.ratio*1.5f/5f,80,"Cancel·lar");

        //general
        bMenu = new BotQ(Pant.realW*Pant.ratio/4f,yIngs,170f,170f);
        bTorna = new BotQ(100,100,115,115); //compte, canvis x i y des del menú

    }

    /*********** RENDER ***********/
    @Override
    public void render(float delta) {
        delta=Math.min(delta, 0.25f);

        Gdx.gl.glClearColor(S.fons.r,S.fons.g,S.fons.b,S.fons.a);
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);

        camera.update();
        lote.setProjectionMatrix(camera.combined);

        lote.begin();
            if(triaTipus){
                /**- - - TRIA DE TIPUS D'INGREDIENT - - -*/
                Recursos.bMenu.setColor(S.graus);
                Recursos.bMenu.draw(lote,"Tria el tipus d'ingredient que vols
afegir:",15f,Pant.screenH*0.85f,Pant.realW*Pant.ratio-30f, Align.center, true);

                //botons tipus
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, bTipusR.x-bTipusR.w/2f, bTipusR.y-bTipusR.h/2f, bTipusR.w, bTipusR.h);
                lote.draw(Recursos.rect, bTipusC.x-bTipusC.w/2f, bTipusC.y-bTipusC.h/2f, bTipusC.w, bTipusC.h);
                lote.draw(Recursos.rect, bTipusA.x-bTipusA.w/2f, bTipusA.y-bTipusA.h/2f, bTipusA.w, bTipusA.h);
                lote.setColor(S.b_normal);
                lote.draw(Recursos.rect, bTipusR.x-bTipusR.w/2f+8, bTipusR.y-bTipusR.h/2f+8, bTipusR.w-16, bTipusR.h-
16);
                lote.draw(Recursos.rect, bTipusC.x-bTipusC.w/2f+8, bTipusC.y-bTipusC.h/2f+8, bTipusC.w-16, bTipusC.h-
16);
                lote.draw(Recursos.rect, bTipusA.x-bTipusA.w/2f+8, bTipusA.y-bTipusA.h/2f+8, bTipusA.w-16, bTipusA.h-
16);

                Recursos.font.setColor(S.graus);
                Recursos.font.draw(lote,F.keep(bTipusR.text),bTipusR.x+F.align(),bTipusR.y+F.vAlign());
                Recursos.font.draw(lote,F.keep(bTipusC.text),bTipusC.x+F.align(),bTipusC.y+F.vAlign());
                Recursos.font.draw(lote,F.keep(bTipusA.text),bTipusA.x+F.align(),bTipusA.y+F.vAlign());

                //botó tornar
                lote.setColor(S.graus);
                lote.draw(Recursos.arrow, bTorna.x-bTorna.w/2f, bTorna.y-bTorna.h/2f, bTorna.w, bTorna.h);

                //TODO afegir en simètric el botó de requeriment aleatori

            }else if(triaRes){
                /**- - - INGREDIENT TIPUS RESOLUCIÓ IRREGULAR - - -*/
                //botó RIs
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, bRIs.x-bRIs.w/2f, bRIs.y-bRIs.h/2f, bRIs.w, bRIs.h);
```

```
                lote.setColor(S.b_normal);
                lote.draw(Recursos.rect, bRIs.x-bRIs.w/2f+8, bRIs.y-bRIs.h/2f+8, bRIs.w-16, bRIs.h-16);
                Recursos.font.setColor(S.b_text);
                Recursos.font.draw(lote, F.keep("Sensible"), bRIs.x+F.align("font"), bRIs.y+F.vAlign("font"));

                //botó RI7
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, bRI7.x-bRI7.w/2f, bRI7.y-bRI7.h/2f, bRI7.w, bRI7.h);
                lote.setColor(S.b_normal);
                lote.draw(Recursos.rect, bRI7.x-bRI7.w/2f+8, bRI7.y-bRI7.h/2f+8, bRI7.w-16, bRI7.h-16);
                Recursos.font.setColor(S.b_text);
                Recursos.font.draw(lote, F.keep("Sèptima"), bRI7.x+F.align("font"), bRI7.y+F.vAlign("font"));

                //botó tornar
                lote.setColor(S.graus);
                lote.draw(Recursos.arrow, bTorna.x-bTorna.w/2f, bTorna.y-bTorna.h/2f, bTorna.w, bTorna.h);
            }else if(triaCadencia){
                /**- - - INGREDIENT TIPUS CADENCIA - - -*/

                Recursos.bigFont.setColor(S.graus);
                Recursos.font.setColor(S.graus);
                Recursos.bMenu.setColor(S.graus);
                Recursos.bNotes.setColor(S.graus);

                //quantitat (p. ex. 2x)
                if(auxTC.n>1 && !auxTC.tcp){
                    Recursos.font.draw(lote, F.keep(auxTC.n+"x"), xIngCad-bAuxTC.w/2f, yIngCad+F.vAlign("font",'d'));
                }
                //text CT
                Recursos.bigFont.draw(lote, F.keep("CT"), xIngCad-bAuxTC.w/2f+bAuxTC.d[0],
yIngCad+F.vAlign("bigFont",'d'));
                //"xifrat"
                Recursos.bMenu.draw(lote, F.keep(F.ifText(auxTC.sept, "7",F.ifText(auxTC.triada, "5"))), xIngCad-
bAuxTC.w/2f+bAuxTC.d[1], yIngCad+F.vAlign("bigFont",'d',"CT")+F.vAlign("bMenu")*0.7f);
                //diferent i tcp
                Recursos.font.draw(lote, F.keep(F.ifText(auxTC.dif, " d")+F.ifText(auxTC.tcp, " !")), xIngCad-
bAuxTC.w/2f+bAuxTC.d[2], yIngCad+F.vAlign("font",'d'));

                //descripció
                Recursos.bNotes.draw(lote, desc, xBarraE+bDif.w*(0.5f+0.2f), Pant.screenH*0.5f,
(Pant.realW*Pant.ratio-bDif.w*0.2f)-(xBarraE+bDif.w*(0.5f+0.2f)), Align.center, true);

                /**- - - Botons opcions - - -*/
                //botó sèptima/tríada
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, b7a.x-b7a.w/2f, b7a.y-b7a.h/2f, b7a.w, b7a.h);
                if(auxTC.sept || auxTC.triada){
                    lote.setColor(S.b_premut);
                }else{
                    lote.setColor(S.b_normal);
                }
                lote.draw(Recursos.rect, b7a.x-b7a.w/2f+8, b7a.y-b7a.h/2f+8, b7a.w-16, b7a.h-16);
                Recursos.font.setColor(S.b_text);
                if(auxTC.sept){
                    Recursos.font.draw(lote,F.keep("Amb sèptima"),b7a.x+F.align(),b7a.y+F.vAlign());
                }else if(auxTC.triada){
                    Recursos.font.draw(lote,F.keep("Tríada"),b7a.x+F.align(),b7a.y+F.vAlign());
                }else{
                    Recursos.font.draw(lote,F.keep("Tríada / Sèptima"),b7a.x+F.align(),b7a.y+F.vAlign());
                }

                /**- - - Botons opcions avançades - - -*/
                //botó més quantitat
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, bMes.x-bMes.w/2f, bMes.y-bMes.h/2f, bMes.w, bMes.h);
                if(!auxTC.tcp){
                    lote.setColor(S.b_normal);
                    Recursos.font.setColor(S.b_text);
                }else{
                    lote.setColor(S.b_inactiu);
                    Recursos.font.setColor(S.b_text_inactiu);
                }
                lote.draw(Recursos.rect, bMes.x-bMes.w/2f+8, bMes.y-bMes.h/2f+8, bMes.w-16, bMes.h-16);
                Recursos.font.draw(lote,F.keep(bMes.text),bMes.x+F.align(),bMes.y+F.vAlign());

                //botó menys quantitat
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, bMenys.x-bMenys.w/2f, bMenys.y-bMenys.h/2f, bMenys.w, bMenys.h);
                if(auxTC.tcp && !(autoTCP && auxTC.n==nTCPCad) || auxTC.n==1){
                    lote.setColor(S.b_inactiu);
```

```
                Recursos.font.setColor(S.b_text_inactiu);
            }else{
                lote.setColor(S.b_normal);
                Recursos.font.setColor(S.b_text);
            }
            lote.draw(Recursos.rect, bMenys.x-bMenys.w/2f+8, bMenys.y-bMenys.h/2f+8, bMenys.w-16, bMenys.h-16);
            Recursos.font.draw(lote,F.keep(bMenys.text),bMenys.x+F.align(),bMenys.y+F.vAlign());

            //botó tants com puguis
            lote.setColor(S.b_marc);
            lote.draw(Recursos.rect, bTCP.x-bTCP.w/2f, bTCP.y-bTCP.h/2f, bTCP.w, bTCP.h);
            if(auxTC.tcp){
                lote.setColor(S.b_premut);
            }else{
                lote.setColor(S.b_normal);
            }
            lote.draw(Recursos.rect, bTCP.x-bTCP.w/2f+8, bTCP.y-bTCP.h/2f+8, bTCP.w-16, bTCP.h-16);
            Recursos.font.setColor(S.b_text);
            Recursos.font.draw(lote,F.keep(bTCP.text),bTCP.x+F.align(),bTCP.y+F.vAlign());

            //botí diferents
            lote.setColor(S.b_marc);
            lote.draw(Recursos.rect, bDif.x-bDif.w/2f, bDif.y-bDif.h/2f, bDif.w, bDif.h);
            if(pucDif){
                if(auxTC.dif){
                    lote.setColor(S.b_premut);
                }else{
                    lote.setColor(S.b_normal);
                }
                Recursos.font.setColor(S.b_text);
            }else{
                lote.setColor(S.b_inactiu);
                Recursos.font.setColor(S.b_text_inactiu);
            }
            lote.draw(Recursos.rect, bDif.x-bDif.w/2f+8, bDif.y-bDif.h/2f+8, bDif.w-16, bDif.h-16);
            Recursos.font.draw(lote,F.keep(bDif.text),bDif.x+F.align(),bDif.y+F.vAlign());

            /**- - - Botons genèrics - - -*/
            //botó tornar
            lote.setColor(S.graus);
            lote.draw(Recursos.arrow, bTorna.x-bTorna.w/2f, bTorna.y-bTorna.h/2f, bTorna.w, bTorna.h);

            //botó desar
            lote.setColor(S.correcte);
            lote.draw(Recursos.cir, bDesa.x-bDesa.w/2f, bDesa.y-bDesa.h/2f, bDesa.w, bDesa.h);
            lote.setColor(S.fons);
            lote.draw(Recursos.cir, bDesa.x-bDesa.w*0.82f/2f, bDesa.y-bDesa.h*0.82f/2f, bDesa.w*0.82f,
bDesa.h*0.82f);
            lote.setColor(S.correcte);
            lote.draw(Recursos.tick, bDesa.x-bDesa.w/2f+12, bDesa.y-bDesa.h/2f+11, bDesa.w-24, bDesa.h-24);

        }else if(triaAcord){
            /**- - - INGREDIENT TIPUS ACORD - - -*/

            Recursos.bigFont.setColor(S.graus);
            Recursos.font.setColor(S.graus);
            Recursos.bMenu.setColor(S.graus);
            Recursos.bNotes.setColor(S.graus);
            //quantitat (p. ex. 2x)
            if(auxTA.n>1 && !auxTA.tcp){
                Recursos.font.draw(lote, F.keep(auxTA.n+"x"), xIngAc-bAuxTA.w/2f, yIngAc+F.vAlign("font", 'd'));
            }
            //grau
            Recursos.bigFont.draw(lote, F.keep(H.roman[auxTA.acord.grau]), xIngAc-bAuxTA.w/2f+bAuxTA.d[0],
yIngAc+F.vAlign("bigFont", 'd'));
            //xifrat
            Recursos.bMenu.draw(lote, F.keep(F.ifText(auxTA.triada && auxTA.acord.inv<1, "5",
H.xifrat(auxTA.acord, false))), xIngAc-bAuxTA.w/2f+bAuxTA.d[1],
yIngAc+F.vAlign("bigFont",'d',H.roman[auxTA.acord.grau])+F.vAlign("bMenu")*0.7f);
            Recursos.bMenu.draw(lote, F.keep(H.xifrat(auxTA.acord, true)), xIngAc-bAuxTA.w/2f+bAuxTA.d[1],
yIngAc+F.vAlign("bMenu")*1.3f);
            //carro
            Recursos.bMenu.draw(lote, F.keep(bAuxTA.carro), xIngAc-bAuxTA.w/2f+bAuxTA.d[2],
yIngAc+F.vAlign("bigFont",'c',H.roman[auxTA.acord.grau])+F.vAlign("bMenu",'c'));
            Recursos.font.draw(lote, F.keep(bAuxTA.carro2), xIngAc-bAuxTA.w/2f+bAuxTA.d[3],
yIngAc+F.vAlign("font",'d'));

            //descripció
            Recursos.bNotes.draw(lote, desc, xBarraE+bDif.w*(0.5f+0.2f), Pant.screenH*0.4f, (bGraus.get(0).x-
bGraus.get(0).w*0.5f-bDif.w*0.2f)-(xBarraE+bDif.w*(0.5f+0.2f)), Align.center, true);
```

```
/**- - - Botons de grau - - -*/
ii=0;
for(BotQ b: bGraus){ //botons de graus
    lote.setColor(S.b_marc);
    lote.draw(Recursos.rect, b.x-b.w/2f, b.y-b.h/2f, b.w, b.h);
    if(auxTA.acord.grau==b.grau){
        lote.setColor(S.b_premut);
    }else{
        lote.setColor(S.b_normal);
    }
    lote.draw(Recursos.rect, b.x-b.w/2f+8f, b.y-b.h/2f+8f, b.w-16f, b.h-16f);

    Recursos.font.setColor(S.b_text);
    Recursos.font.draw(lote,F.keep(H.roman[b.grau]), b.x+F.align(), b.y+F.vAlign());
    ii++;
}

/**- - - Botons opcions d'acord - - -*/
//botó sèptima
lote.setColor(S.b_marc);
lote.draw(Recursos.rect, b7a.x-b7a.w/2f, b7a.y-b7a.h/2f, b7a.w, b7a.h);
if(auxTA.acord.sept || auxTA.triada){
    lote.setColor(S.b_premut);
}else{
    lote.setColor(S.b_normal);
}
lote.draw(Recursos.rect, b7a.x-b7a.w/2f+8, b7a.y-b7a.h/2f+8, b7a.w-16, b7a.h-16);
Recursos.font.setColor(S.b_text);
if(auxTA.acord.sept){
    Recursos.font.draw(lote,F.keep("7a"),b7a.x+F.align(),b7a.y+F.vAlign());
}else if(auxTA.triada){
    Recursos.font.draw(lote,F.keep("Tri"),b7a.x+F.align(),b7a.y+F.vAlign());
}else{
    Recursos.font.draw(lote,F.keep(b7a.text),b7a.x+F.align(),b7a.y+F.vAlign());
}

//botó dominants
lote.setColor(S.b_marc);
lote.draw(Recursos.rect, bDom.x-bDom.w/2f, bDom.y-bDom.h/2f, bDom.w, bDom.h);
if(auxTA.acord.dom){
    lote.setColor(S.b_premut);
}else if(pucDom){
    lote.setColor(S.b_normal);
}else{
    lote.setColor(S.b_inactiu);
}
lote.draw(Recursos.rect, bDom.x-bDom.w/2f+8, bDom.y-bDom.h/2f+8, bDom.w-16, bDom.h-16);
if(pucDom){
    Recursos.font.setColor(S.b_text);
}else{
    Recursos.font.setColor(S.b_text_inactiu);
}
if(auxTA.acord.grau==4){
    Recursos.font.draw(lote,F.keep("V9"),bDom.x+F.align(),bDom.y+F.vAlign());
}else if(auxTA.acord.grau==1){
    Recursos.font.draw(lote,F.keep("DD"),bDom.x+F.align(),bDom.y+F.vAlign());
}else{
    Recursos.font.draw(lote,F.keep(bDom.text),bDom.x+F.align(),bDom.y+F.vAlign());
}

//botó inversions
lote.setColor(S.b_marc);
lote.draw(Recursos.rect, bInv.x-bInv.w/2f, bInv.y-bInv.h/2f, bInv.w, bInv.h);
if(auxTA.acord.inv==0){
    lote.setColor(S.b_normal);
}else{
    lote.setColor(S.b_premut);
}
lote.draw(Recursos.rect, bInv.x-bInv.w/2f+8, bInv.y-bInv.h/2f+8, bInv.w-16, bInv.h-16);
Recursos.font.setColor(S.b_text);
switch(auxTA.acord.inv){
case -1:
    Recursos.font.draw(lote,F.keep("EF"),bInv.x+F.align(),bInv.y+F.vAlign());
    break;
case 0:
    Recursos.font.draw(lote,F.keep(bInv.text),bInv.x+F.align(),bInv.y+F.vAlign());
    break;
case 1:
    Recursos.font.draw(lote,F.keep("1a"),bInv.x+F.align(),bInv.y+F.vAlign());
```

```
    break;
case 2:
    Recursos.font.draw(lote,F.keep("2a"),bInv.x+F.align(),bInv.y+F.vAlign());
    break;
case 3:
    Recursos.font.draw(lote,F.keep("3a"),bInv.x+F.align(),bInv.y+F.vAlign());
    break;
}

//botó alteracions
lote.setColor(S.b_marc);
lote.draw(Recursos.rect, bAlt.x-bAlt.w/2f, bAlt.y-bAlt.h/2f, bAlt.w, bAlt.h);
if(altOp!=0){ //hi ha opcionals disponibles
    if(auxTA.alt){ //tinc alteració obligada
        lote.setColor(S.b_premut);
    }else{
        lote.setColor(S.b_normal);
    }
}else if(auxTA.acord.grau==7 && auxTA.acord.sept){ //puc posar 7M (és un X amb 7a)
    if(auxTA.septMajor){ //tinc la 7M posada
        lote.setColor(S.b_premut);
    }else{
        lote.setColor(S.b_normal);
    }
}else{ //no puc fer res amb el botó
    lote.setColor(S.b_inactiu);
}
lote.draw(Recursos.rect, bAlt.x-bAlt.w/2f+8, bAlt.y-bAlt.h/2f+8, bAlt.w-16, bAlt.h-16);
if(altOp!=0 || (auxTA.acord.grau==7 && auxTA.acord.sept)){
    Recursos.font.setColor(S.b_text);
}else{
    Recursos.font.setColor(S.b_text_inactiu);
}
if(auxTA.alt){
    Recursos.font.draw(lote,F.keep(H.altOpText(auxTA.acord)),bAlt.x+F.align(),bAlt.y+F.vAlign());
}else if(altOp!=0){
    Recursos.font.draw(lote,F.keep(bAlt.text+(altOp%10)),bAlt.x+F.align(),bAlt.y+F.vAlign());
}else if(auxTA.acord.grau==7 && auxTA.acord.sept){
    Recursos.font.draw(lote,F.keep("7M"),bAlt.x+F.align(),bAlt.y+F.vAlign());
}else{
    Recursos.font.draw(lote,F.keep(bAlt.text),bAlt.x+F.align(),bAlt.y+F.vAlign());
}
/**- - - Botons opcions avançades - - -*/
//botó més quantitat
lote.setColor(S.b_marc);
lote.draw(Recursos.rect, bMes.x-bMes.w/2f, bMes.y-bMes.h/2f, bMes.w, bMes.h);
if(!auxTA.tcp){
    lote.setColor(S.b_normal);
    Recursos.font.setColor(S.b_text);
}else{
    lote.setColor(S.b_inactiu);
    Recursos.font.setColor(S.b_text_inactiu);
}
lote.draw(Recursos.rect, bMes.x-bMes.w/2f+8, bMes.y-bMes.h/2f+8, bMes.w-16, bMes.h-16);
Recursos.font.draw(lote,F.keep(bMes.text),bMes.x+F.align(),bMes.y+F.vAlign());

//botó menys quantitat
lote.setColor(S.b_marc);
lote.draw(Recursos.rect, bMenys.x-bMenys.w/2f, bMenys.y-bMenys.h/2f, bMenys.w, bMenys.h);
if(auxTA.tcp && !(autoTCP && auxTA.n==nTCPAc) || auxTA.n==1){
    lote.setColor(S.b_inactiu);
    Recursos.font.setColor(S.b_text_inactiu);
}else{
    lote.setColor(S.b_normal);
    Recursos.font.setColor(S.b_text);
}
lote.draw(Recursos.rect, bMenys.x-bMenys.w/2f+8, bMenys.y-bMenys.h/2f+8, bMenys.w-16, bMenys.h-16);
Recursos.font.draw(lote,F.keep(bMenys.text),bMenys.x+F.align(),bMenys.y+F.vAlign());

//botó tants com puguis
lote.setColor(S.b_marc);
lote.draw(Recursos.rect, bTCP.x-bTCP.w/2f, bTCP.y-bTCP.h/2f, bTCP.w, bTCP.h);
if(auxTA.tcp){
    lote.setColor(S.b_premut);
}else{
    lote.setColor(S.b_normal);
}
lote.draw(Recursos.rect, bTCP.x-bTCP.w/2f+8, bTCP.y-bTCP.h/2f+8, bTCP.w-16, bTCP.h-16);
Recursos.font.setColor(S.b_text);
Recursos.font.draw(lote,F.keep(bTCP.text),bTCP.x+F.align(),bTCP.y+F.vAlign());
```

```
                //botó diferents
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, bDif.x-bDif.w/2f, bDif.y-bDif.h/2f, bDif.w, bDif.h);
                if(pucDifAc || pucDifInv){
                    if(auxTA.difAc || auxTA.difInv){
                        lote.setColor(S.b_premut);
                    }else{
                        lote.setColor(S.b_normal);
                    }
                    Recursos.font.setColor(S.b_text);
                }else{
                    lote.setColor(S.b_inactiu);
                    Recursos.font.setColor(S.b_text_inactiu);
                }
                lote.draw(Recursos.rect, bDif.x-bDif.w/2f+8, bDif.y-bDif.h/2f+8, bDif.w-16, bDif.h-16);
                if(auxTA.difAc){
                    if(auxTA.difInv){ //da + di
                        Recursos.font.draw(lote,F.keep("A+I"),bDif.x+F.align(),bDif.y+F.vAlign());
                    }else{ //da
                        Recursos.font.draw(lote,F.keep("Ac"),bDif.x+F.align(),bDif.y+F.vAlign());
                    }
                }else{
                    if(auxTA.difInv){
                        Recursos.font.draw(lote,F.keep("Inv"),bDif.x+F.align(),bDif.y+F.vAlign());
                    }else{
                        Recursos.font.draw(lote,F.keep(bDif.text),bDif.x+F.align(),bDif.y+F.vAlign());
                    }
                }

                //botó enllaçats
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, bEnll.x-bEnll.w/2f, bEnll.y-bEnll.h/2f, bEnll.w, bEnll.h);
                if(pucEnll){
                    if(auxTA.enll){
                        lote.setColor(S.b_premut);
                    }else{
                        lote.setColor(S.b_normal);
                    }
                }else{
                    lote.setColor(S.b_inactiu);
                }
                lote.draw(Recursos.rect, bEnll.x-bEnll.w/2f+8, bEnll.y-bEnll.h/2f+8, bEnll.w-16, bEnll.h-16);
                if(pucEnll){
                    Recursos.font.setColor(S.b_text);
                }else{
                    Recursos.font.setColor(S.b_text_inactiu);
                }
                Recursos.font.draw(lote,F.keep(bEnll.text),bEnll.x+F.align(),bEnll.y+F.vAlign());

                /**- - - Botons genèrics - - -*/
                //botó tornar
                lote.setColor(S.graus);
                lote.draw(Recursos.arrow, bTorna.x-bTorna.w/2f, bTorna.y-bTorna.h/2f, bTorna.w, bTorna.h);

                //botó desar
                lote.setColor(S.correcte);
                lote.draw(Recursos.cir, bDesa.x-bDesa.w/2f, bDesa.y-bDesa.h/2f, bDesa.w, bDesa.h);
                lote.setColor(S.fons);
                lote.draw(Recursos.cir, bDesa.x-bDesa.w*0.82f/2f, bDesa.y-bDesa.h*0.82f/2f, bDesa.w*0.82f,
bDesa.h*0.82f);
                lote.setColor(S.correcte);
                lote.draw(Recursos.tick, bDesa.x-bDesa.w/2f+12, bDesa.y-bDesa.h/2f+11, bDesa.w-24, bDesa.h-24);


            }else{
                /**- - - PANTALLA DELS INGREDIENTS - - -*/
                if(!vullEsbos){
                    //Botó afegeix
                    lote.setColor(S.b_marc);
                    lote.draw(Recursos.cir, bAfegeix.x-bAfegeix.w/2f, bAfegeix.y-bAfegeix.h/2, bAfegeix.w,
bAfegeix.h);
                    if(tipusAcord.size+tipusCadencia.size+tipusRes.size<maxIngs){
                        lote.setColor(S.b_normal);
                        Recursos.bigFont.setColor(S.b_text);
                    }else{
                        lote.setColor(S.b_inactiu);
                        Recursos.bigFont.setColor(S.b_text_inactiu);
                    }
                    lote.draw(Recursos.cir, bAfegeix.x-bAfegeix.w/2f+8, bAfegeix.y-bAfegeix.h/2+8, bAfegeix.w-16,
```

```
bAfegeix.h-16);

    Recursos.bigFont.draw(lote,F.keep(bAfegeix.text),bAfegeix.x+F.align("bigFont"),bAfegeix.y+F.vAlign("bigFont"));
//TODO canviar per dibuixos o alguna cosa que es vegi millor

                    //Botó edita
                    lote.setColor(S.b_marc);
                    lote.draw(Recursos.cir, bEdita.x-bEdita.w/2f, bEdita.y-bEdita.h/2, bEdita.w, bEdita.h);
                    if(ingAct!=-1){
                        lote.setColor(S.b_normal);
                        Recursos.bNotes.setColor(S.b_text);
                    }else{
                        lote.setColor(S.b_inactiu);
                        Recursos.bNotes.setColor(S.b_text_inactiu);
                    }
                    lote.draw(Recursos.cir, bEdita.x-bEdita.w/2f+8, bEdita.y-bEdita.h/2+8, bEdita.w-16, bEdita.h-16);

    Recursos.bNotes.draw(lote,F.keep(bEdita.text),bEdita.x+F.align("bNotes"),bEdita.y+F.vAlign("bNotes"));

                    //Botó enllaçat amb
                    lote.setColor(S.b_marc);
                    lote.draw(Recursos.cir, bEAmb.x-bEAmb.w/2f, bEAmb.y-bEAmb.h/2, bEAmb.w, bEAmb.h);
                    lote.setColor(S.b_inactiu);
                    lote.draw(Recursos.cir, bEAmb.x-bEAmb.w/2f+8, bEAmb.y-bEAmb.h/2+8, bEAmb.w-16, bEAmb.h-16);
                    Recursos.bNotes.setColor(S.b_text_inactiu);

    Recursos.bNotes.draw(lote,F.keep(bEAmb.text),bEAmb.x+F.align("bNotes"),bEAmb.y+F.vAlign("bNotes"));

                    //Botó esborra
                    if(ingAct!=-1){
                        lote.setColor(S.error);
                    }else{
                        lote.setColor(S.b_inactiu);
                    }
                    lote.draw(Recursos.cir, bEsborra.x-bEsborra.w/2f, bEsborra.y-bEsborra.h/2, bEsborra.w,
bEsborra.h);
                    lote.setColor(S.fons);
                    lote.draw(Recursos.cir, bEsborra.x-bEsborra.w/2f+8, bEsborra.y-bEsborra.h/2+8, bEsborra.w-16,
bEsborra.h-16);
                    if(ingAct!=-1){
                        Recursos.bNotes.setColor(S.error);
                    }else{
                        Recursos.bNotes.setColor(S.b_inactiu);
                    }

    Recursos.bNotes.draw(lote,F.keep(bEsborra.text),bEsborra.x+F.align("bNotes"),bEsborra.y+F.vAlign("bNotes"));
                }else{
                    lote.setColor(S.b_marc); //botó de cancel·lar el pas a esbós
                    lote.draw(Recursos.rect, bCancel.x-bCancel.w/2f, bCancel.y-bCancel.h/2, bCancel.w, bCancel.h);
                    lote.setColor(S.b_normal);
                    lote.draw(Recursos.rect, bCancel.x-bCancel.w/2f+8, bCancel.y-bCancel.h/2+8, bCancel.w-16,
bCancel.h-16);
                    Recursos.font.setColor(S.b_text);
                    Recursos.font.draw(lote,F.keep(bCancel.text),bCancel.x+F.align(),bCancel.y+F.vAlign());
                }

                //Botó fet
                lote.setColor(S.b_marc);
                lote.draw(Recursos.rect, bFet.x-bFet.w/2f, bFet.y-bFet.h/2, bFet.w, bFet.h);
                lote.setColor(S.b_normal);
                lote.draw(Recursos.rect, bFet.x-bFet.w/2f+8, bFet.y-bFet.h/2+8, bFet.w-16, bFet.h-16);
                Recursos.font.setColor(S.b_text);
                if(vullEsbos){
                    Recursos.font.draw(lote,F.keep("Passar a fer l'Esbós"),bFet.x+F.align(),bFet.y+F.vAlign());
                }else{
                    Recursos.font.draw(lote,F.keep(bFet.text),bFet.x+F.align(),bFet.y+F.vAlign());
                }

                //botó del to
                lote.setColor(S.tonalitat);
                lote.draw(Recursos.cir, bMenu.x-bMenu.w/2f, bMenu.y-bMenu.h/2f, bMenu.w, bMenu.h);
                lote.setColor(S.fons);
                lote.draw(Recursos.cir, bMenu.x-bMenu.w*A.rMenu/2f, bMenu.y-bMenu.h*A.rMenu/2f, bMenu.w*A.rMenu,
bMenu.h*A.rMenu);
                if(H.getAlt(H.tonalitat)==""){
                    Recursos.font.setColor(S.graus);
                    Recursos.font.draw(lote, F.keep(H.modeNotes(H.tonalitat.nom,H.mode)+H.getAlt(H.tonalitat)+"
"+H.rModes[H.mode]), bMenu.x+F.align(), bMenu.y+F.vAlign());
                }else{ //si el nom de la tonalitat és massa llarg, faig més petita la lletra.
                    Recursos.bNotes.setColor(S.graus);
```

```
                Recursos.bNotes.draw(lote, F.keep(H.modeNotes(H.tonalitat.nom,H.mode)+H.getAlt(H.tonalitat)+"
"+H.rModes[H.mode]), bMenu.x+F.align("bNotes"), bMenu.y+F.vAlign("bNotes"));
            }
            /**- - - ingredients - - -*/
            if(bTipusAcord.size+bTipusCadencia.size+bTipusRes.size>0){
                Recursos.font.setColor(S.graus);
                Recursos.iQuant.setColor(S.graus);
                Recursos.iXifrat.setColor(S.graus);
                for(BotIA b: bTipusAcord){
                    if(vullEsbos){
                        lote.setColor(S.ingredients);
                    }else{
                        if(b.iAc.id==ingAct){
                            lote.setColor(S.b_premut);
                        }else{
                            lote.setColor(S.b_normal);
                        }
                    }
                    lote.draw(Recursos.rect, b.x-b.h*0.2f, b.y-b.h*0.2f, b.w+b.h*0.4f, b.h);
                    //quantitat (p. ex. 2x)
                    if(b.iAc.n>1 && !b.iAc.tcp){
                        Recursos.iQuant.draw(lote, b.iAc.n+"x", b.x, b.y+F.vAlign("iQuant", 'd'));
                    }
                    //grau
                    Recursos.font.draw(lote, F.keep(H.roman[b.iAc.acord.grau]), b.x+b.d[0], b.y+F.vAlign("font",
'd'));
                    //xifrat
                    Recursos.iXifrat.draw(lote, F.keep(F.ifText(b.iAc.triada && b.iAc.acord.inv<1, "5",
H.xifrat(b.iAc.acord, false))), b.x+b.d[1],
b.y+F.vAlign("font",'d',H.roman[b.iAc.acord.grau])+F.vAlign("iXifrat")*0.7f);
                    Recursos.iXifrat.draw(lote, F.keep(H.xifrat(b.iAc.acord, true)), b.x+b.d[1],
b.y+F.vAlign("iXifrat")*1.3f);
                    //carro
                    Recursos.iXifrat.draw(lote, F.keep(b.carro), b.x+b.d[2],
b.y+F.vAlign("font",'c',H.roman[b.iAc.acord.grau])+F.vAlign("iXifrat",'c'));
                    Recursos.iQuant.draw(lote, F.keep(b.carro2), b.x+b.d[3], b.y+F.vAlign("iQuant",'d'));
                }
                for(BotIC b: bTipusCadencia){
                    if(vullEsbos){
                        lote.setColor(S.ingredients);
                    }else{
                        if(b.iCad.id==ingAct){
                            lote.setColor(S.b_premut);
                        }else{
                            lote.setColor(S.b_normal);
                        }
                    }
                    lote.draw(Recursos.rect, b.x-b.h*0.2f, b.y-b.h*0.2f, b.w+b.h*0.4f, b.h);
                    //quantitat (p. ex. 2x)
                    if(b.iCad.n>1 && !b.iCad.tcp){
                        Recursos.iQuant.draw(lote, b.iCad.n+"x", b.x, b.y+F.vAlign("iQuant", 'd'));
                    }
                    //text "CT"
                    Recursos.font.draw(lote, F.keep("CT"), b.x+b.d[0], b.y+F.vAlign("font",'d'));
                    //"xifrat"
                    Recursos.iXifrat.draw(lote, F.keep(F.ifText(b.iCad.sept, "7",F.ifText(b.iCad.triada,
"5"))),b.x+b.d[1],b.y+F.vAlign("font",'d',"CT")+F.vAlign("iXifrat")*0.7f);
                    //diferent i tcp
                    Recursos.iQuant.draw(lote, F.keep(F.ifText(b.iCad.dif, " d")+F.ifText(b.iCad.tcp, " !")),
b.x+b.d[2], b.y+F.vAlign("iQuant",'d'));
                }
                for(BotIR b: bTipusRes){
                    if(vullEsbos){
                        lote.setColor(S.ingredients);
                    }else{
                        if(b.iRI.id==ingAct){
                            lote.setColor(S.b_premut);
                        }else{
                            lote.setColor(S.b_normal);
                        }
                    }
                    lote.draw(Recursos.rect, b.x-b.h*0.2f, b.y-b.h*0.2f, b.w+b.h*0.4f, b.h);
                    //text "RI"
                    Recursos.font.draw(lote, F.keep("RI"), b.x, b.y+F.vAlign("font",'d'));
                    //"xifrat"
                    Recursos.iXifrat.draw(lote, F.keep(F.ifText(b.iRI.sens,
"S","7")),b.x+b.d*1.1f,b.y+F.vAlign("iXifrat")*1.4f);
                }
                lote.setColor(S.tonalitat);
                lote.draw(Recursos.rect, bMenu.x+dMI*0.8f, yClau, 10f, hClau);
```

```
                lote.draw(Recursos.rect, bMenu.x+dMI*0.8f, yClau, 30f, -10f);
                lote.draw(Recursos.rect, bMenu.x+dMI*0.8f, yClau+hClau, 30f, 10f);
            }else{
                Recursos.bNotes.setColor(S.graus); //TODO arreglar posicions en proporció a text incloent botó
                Recursos.bNotes.draw(lote,F.keep("Encara no hi ha
ingredients."),bMenu.x+dMI,yIngs+F.vAlign("bNotes"));
                if(vullEsbos){
                    Recursos.xifrat.setColor(S.alerta);
                    Recursos.xifrat.draw(lote, "Deixaràs l'enunciat lliure. Segur que vols continuar?",
bAfegeix.x, Pant.screenH*0.35f, bEsborra.x-bAfegeix.x, Align.center, true);
                }
            }
        }
        lote.end();
    }

    /********** CLICK **********/
    @Override
    public boolean touchDown(int screenX, int screenY, int pointer, int button) {
        clickBot = false;
        click.x = screenX;
        click.y = screenY;
        encuadre.unproject(click);

        if(triaTipus){
            /**- - - TRIA DE TIPUS D'INGREDIENT - - -*/
            if(click.y>bTipusR.y-bTipusR.w/2f && click.y<bTipusR.y+bTipusR.w/2f){ //tots tres són dins les mateixes
cotes verticals
                if(click.x>bTipusA.x-bTipusA.w/2f && click.x<bTipusA.x+bTipusA.w/2f){ //Tipus Acord (el comprovo
primer perquè serà el predominant)

                    tipusAcord.add(new IngAcord(idGen++)); //sembla que així envia idGen al constructor i fa
l'increment a posteriori (em sembla bé).
                    auxTA = tipusAcord.get(tipusAcord.size-1); //ho guardo en auxiliar per referència fàcil
                    System.out.println("Nou ingredient de tipus acord ("+tipusAcord.size+"/"+tipusAcord.size+") amb
id="+auxTA.id);

                    bTipusAcord.add(new BotIA(Pant.realW*Pant.ratio/2, Pant.screenH/2, auxTA));
                    System.out.println("- Creat el botó associat a l'ingredient
("+bTipusAcord.size+"/"+bTipusAcord.size+").");
                    bAuxTA = bTipusAcord.get(bTipusAcord.size-1); //mateixa filosofia

                    permisosBotons("Ac");
                    descriu("Ac");
                    posaElsBotonsPer("triaAcord");
                    triaAcord = true;
                    triaTipus = false;

                }else if(click.x>bTipusC.x-bTipusC.w/2f && click.x<bTipusC.x+bTipusC.w/2f){ //Tipus Cadència (ordeno
segons freqüència en què apareixeran, que estalviarà una mica de càlcul)

                    tipusCadencia.add(new IngCadencia(idGen++));
                    auxTC = tipusCadencia.get(tipusCadencia.size-1);
                    System.out.println("Nou ingredient de tipus cadencia
("+tipusCadencia.size+"/"+tipusCadencia.size+") amb id="+auxTC.id);

                    bTipusCadencia.add(new BotIC(Pant.realW*Pant.ratio/2, Pant.screenH/2, auxTC));
                    System.out.println("- Creat el botó associat a l'ingredient
("+bTipusCadencia.size+"/"+bTipusCadencia.size+").");
                    bAuxTC = bTipusCadencia.get(bTipusCadencia.size-1);

                    permisosBotons("Cad");
                    descriu("Cad");
                    posaElsBotonsPer("triaCadencia");
                    triaCadencia = true;
                    triaTipus=false;
                }else if(click.x>bTipusR.x-bTipusR.w/2f && click.x<bTipusR.x+bTipusR.w/2f){ //Tipus RI

                    tipusRes.add(new IngRes(idGen++));
                    auxTR = tipusRes.get(tipusRes.size-1);
                    System.out.println("Nou ingredient de tipus RI ("+tipusRes.size+"/"+tipusRes.size+") amb
id="+auxTR.id);

                    bTipusRes.add(new BotIR(0,0,auxTR));
                    System.out.println("- Creat el botó associat a l'ingredient
("+bTipusRes.size+"/"+bTipusRes.size+").");
                    bAuxTR = bTipusRes.get(bTipusRes.size-1);

                    posaElsBotonsPer("triaRes");
                    triaRes=true;
```

```
                triaTipus=false;
            }
        }else if(Math.pow(click.x-bTorna.x, 2)+Math.pow(click.y-bTorna.y, 2)<Math.pow(bTorna.w/2f,2)){
            System.out.println("Tornar a pantalla d'ingredients.");
            triaTipus=false;
        }
    }else if(triaRes){
        /**- - - INGREDIENT TIPUS RESOLUCIÓ IRREGULAR - - -*/
        if(click.x>bRI7.x-bRI7.w/2f && click.x<bRI7.x+bRI7.w/2f
        && click.y>bRI7.y-bRI7.h/2f && click.y<bRI7.y+bRI7.h/2f){ //botó RI de la Sèptima
            System.out.println("\nDesar ingredient:");
            System.out.println("ID="+(auxTR.id)+" | Resolució irregular de la sèptima.");
            auxTR.sens=false;

            posiciona();
            triaRes = false;

            clickBot=true;
        }else if(click.x>bRIs.x-bRIs.w/2f && click.x<bRIs.x+bRIs.w/2f
              && click.y>bRIs.y-bRIs.h/2f && click.y<bRIs.y+bRIs.h/2f){ //botó RI de la Sensible
            System.out.println("\nDesar ingredient:");
            System.out.println("ID="+(auxTR.id)+" | Resolució irregular de la sensible.");
            auxTR.sens=true;

            posiciona();
            triaRes = false;

            clickBot=true;
        }else if(Math.pow(click.x-bTorna.x, 2)+Math.pow(click.y-bTorna.y, 2)<Math.pow(bTorna.w/2f,2)){
            if(editant){
                System.out.println("Descartar canvis. Tornant a pantalla d'ingredients.");
                posiciona();
                editant = false;
            }else{
                System.out.println("Tornar a tria de tipus. Descartar ingredient.");
                bTipusRes.removeValue(bAuxTR, true);
                tipusRes.removeValue(auxTR, true);

                posaElsBotonsPer("triaTipus");
                triaTipus = true;
            }
            triaRes = false;

            clickBot=true;
        }
    }else if(triaCadencia){
        /**- - - INGREDIENT TIPUS CADÈNCIA - - -*/
        if(click.x>b7a.x-b7a.w/2f && click.y<b7a.y+b7a.h/2f){ //marge inferior (opcions)
            if(click.x<b7a.x+b7a.w/2f){ //botó de sèptimes i tríades
                if(auxTC.sept){   // 7a --> Res
                    System.out.println("ID="+(auxTC.id)+" | 7/5: Cap obligació.");
                    auxTC.sept=false;
                }else if(auxTC.triada){ // Tríada obligat --> Sèptima
                    System.out.println("ID="+(auxTC.id)+" | 7/5: Amb sèptima.");
                    auxTC.triada=false;
                    auxTC.sept=true;
                }else{                  // res --> Tríada obligat
                    System.out.println("ID="+(auxTC.id)+" | 7/5: Tríada obligat.");
                    auxTC.triada=true;
                }
                permisosBotons("Cad"); //recalculo permisos (de vegades 7a decideix si hi puc Dif)
                bAuxTC.updateWH(); //actualitzo les mides i distàncies del botó
                descriu("Cad");

                clickBot = true;
            }
        }
        if(!clickBot && click.x<bDif.x+bDif.w/2f && click.y<bMes.y+bMes.h/2f){ //marge esquerra (opcions
avançades)
            if(click.y>bMes.y-bMes.h/2f){ //botó més quantitat
                if(!auxTC.tcp){ //només faig anar els botons + - quan no hi ha tcp
                    if(auxTC.n<nTCPCad || (!autoTCP)){ //sóc per sota el límit o tinc desactivat el tcp automàtic
                        auxTC.n++;
                        System.out.println("ID="+(auxTC.id)+" | n="+(auxTC.n));
                    }else{ //tinc tcp automàtic i he arribat al límit
                        auxTC.tcp=true; //activo tcp
                        System.out.println("N elevada. Proposta de TCP.\n"
                                + "ID="+(auxTC.id)+" | TCP: "+F.siNo(auxTC.tcp));
                    }
                    permisosBotons("Cad");
```

```
                    bAuxTC.updateWH();
                    descriu("Cad");
                }else{
                    System.out.println("TCP activat. Desactiva'l per fer canvis de quantitat.");
                }
            }else if(click.y>bMenys.y-bMenys.h/2f){ //botó menys quantitat
                if(!auxTC.tcp){ //només faig anar els botons + - quan no hi ha tcp
                    if(auxTC.n>1){
                        auxTC.n--;
                        System.out.println("ID="+(auxTC.id)+" | n="+(auxTC.n));
                    }else{
                        System.out.println("No pots posar-ne menys.");
                    }
                    permisosBotons("Cad");
                    bAuxTC.updateWH();
                    descriu("Cad");
                }else{
                    if(autoTCP && auxTC.n>=nTCPCad){ //permeto tornar enrere quan el tcp s'ha activat
automàticament
                        auxTC.tcp=false;
                        System.out.println("Enrere.\n"
                                + "ID="+(auxTC.id)+" | TCP: "+F.siNo(auxTC.tcp));
                        permisosBotons("Cad");
                        bAuxTC.updateWH();
                        descriu("Cad");
                    }else{
                        System.out.println("TCP activat. Desactiva'l per fer canvis de quantitat.");
                    }
                }
            }else if(click.y>bTCP.y-bTCP.h/2f){ //botó tants com puguis
                if(auxTC.tcp){
                    auxTC.tcp=false;
                    if(auxTC.n>=nTCPCad){
                        autoTCP=false;
                    }
                }else{
                    auxTC.tcp=true;
                }
                permisosBotons("Cad");
                bAuxTC.updateWH();
                descriu("Cad");
                System.out.println("ID="+(auxTC.id)+" | TCP: "+F.siNo(auxTC.tcp));

            }else if(click.y>bDif.y-bDif.h/2f){ //botó diferents
                if(pucDif){
                    if(auxTC.dif){
                        auxTC.dif=false;
                    }else{
                        auxTC.dif=true;
                    }
                    System.out.println("ID="+(auxTC.id)+" | Diferents: "+F.siNo(auxTC.dif));
                }else{
                    System.out.println("No pots demanar diferents en aquest context");
                }
                bAuxTC.updateWH();
                descriu("Cad");
            }
        }
        if(!clickBot){
            if(Math.pow(click.x-bTorna.x, 2)+Math.pow(click.y-bTorna.y, 2)<Math.pow(bTorna.w/2f,2)){
                if(editant){
                    System.out.println("Descartar canvis. Tornant a pantalla d'ingredients.");
                    bAuxTC.shrink();
                    posiciona();
                    editant=false;
                }else{
                    System.out.println("Tornar a tria de tipus. Descartar ingredient.");
                    bTipusCadencia.removeValue(bAuxTC, true);
                    tipusCadencia.removeValue(auxTC, true);

                    posaElsBotonsPer("triaTipus");
                    triaTipus = true;
                }
                triaCadencia = false;

                clickBot=true;
            }else if(Math.pow(click.x-bDesa.x, 2)+Math.pow(click.y-bDesa.y, 2)<Math.pow(bDesa.w/2f,2)){
                System.out.println("\nDesar ingredient:");
                bAuxTC.printLn();
```

```
                bAuxTC.shrink();
                posiciona();
                triaCadencia = false;

                clickBot=true;
            }
        }
    }else if(triaAcord){
        /**- - - INGREDIENT TIPUS ACORD - - -*/
        if(click.x>bGraus.get(0).x-bGraus.get(0).w/2f){ //si pico al marge dret
            for(BotQ b: bGraus){
                if(click.y>b.y-b.h/2f){ //he clicat un botó de grau
                    if(auxTA.acord.grau!=b.grau){ //i he canviat respecte al grau que tenia
                        auxTA.acord.grau = (byte) b.grau; //assigno a l'ingredient el grau triat
                        System.out.println("ID="+(auxTA.id)+" | Grau="+H.roman[b.grau]);
                        permisosBotons("Ac"); //recalculo els permisos dels botons (Dom, alt, dif, enll)

                        auxTA.acord.alt6=false; //trec els paràmetres que depenen del grau
                        auxTA.acord.alt7=false;
                        auxTA.acord.alt9=false;
                        auxTA.alt=false;
                        auxTA.septMajor=false;

                        bAuxTA.updateWH(); //actualitzo les mides i distàncies del botó
                        descriu("Ac");
                    }
                    clickBot = true;
                    break;
                }
            }
        }
        if(!clickBot && click.x>b7a.x-b7a.w/2f && click.y<b7a.y+b7a.h/2f){ //marge inferior (opcions)
            if(click.x<b7a.x+b7a.w/2f){ //botó de sèptimes i tríades
                if(auxTA.acord.sept){   // 7a --> Tríada obligat
                    System.out.println("ID="+(auxTA.id)+" | 7/5: Tríada obligat.");
                    if(auxTA.acord.inv==3){
                        auxTA.acord.inv=-1; //trec la inversió si esdevé impossible
                    }
                    auxTA.acord.sept=false;
                    auxTA.septMajor=false;
                    auxTA.triada=true;
                }else if(auxTA.triada){ // Tríada obligat --> res
                    System.out.println("ID="+(auxTA.id)+" | 7/5: Cap obligació.");
                    auxTA.triada=false;
                }else{                  // res --> 7a
                    System.out.println("ID="+(auxTA.id)+" | 7/5: Amb sèptima.");
                    auxTA.triada=false;
                    auxTA.acord.sept=true;
                }
                permisosBotons("Ac"); //recalculo permisos (de vegades 7a decideix si hi ha Alt)
                bAuxTA.updateWH(); //actualitzo les mides i distàncies del botó
                descriu("Ac");

                clickBot=true;

            }else if(click.x<bDom.x+bDom.w/2f){ //botó família de la dominant
                if(pucDom){
                    if(auxTA.acord.dom){
                        auxTA.acord.dom=false;
                    }else{
                        auxTA.acord.dom=true;
                    }
                    System.out.println("ID="+(auxTA.id)+" | Dom: "+F.siNo(auxTA.acord.dom));
                    permisosBotons("Ac");
                    bAuxTA.updateWH(); //actualitzo les mides i distàncies del botó
                    descriu("Ac");

                    clickBot=true;
                }else{
                    System.out.println("Encara no pots posar DS en aquest acord.");
                }

            }else if(click.x<bInv.x+bInv.w/2f){ //botó inversions
                if(auxTA.acord.sept){ //si té sèptima
                    if(auxTA.acord.inv<3){
                        auxTA.acord.inv++;
                    }else{
                        auxTA.acord.inv=-1;
                    }
                }else{ //si no té sèptima
```

```
                    if(auxTA.acord.inv<2){
                        auxTA.acord.inv++;
                    }else{
                        auxTA.acord.inv=-1;
                    }
                }
                if(auxTA.acord.inv==-1){
                    System.out.println("ID="+(auxTA.id)+" | Inv: EF obligatori.");
                }else{
                    System.out.println("ID="+(auxTA.id)+" | Inv: "+H.xifrat(auxTA.acord,
false)+"/"+H.xifrat(auxTA.acord, true));
                }
                permisosBotons("Ac");
                bAuxTA.updateWH();
                descriu("Ac");

                clickBot=true;
            }else if(click.x<bAlt.x+bAlt.w/2f){ //botó d'alteracions
                if(altOp!=0){
                    switch(altOp%10){
                    case 9:
                        if(auxTA.acord.alt9){
                            auxTA.acord.alt9=false;
                        }else{
                            if(auxTA.alt){
                                auxTA.alt=false;
                            }else{
                                auxTA.acord.alt9=true;
                                auxTA.alt=true;
                            }
                        }
                        break;
                    case 6:
                        if(auxTA.acord.alt6){
                            auxTA.acord.alt6=false;
                        }else{
                            if(auxTA.alt){
                                auxTA.alt=false;
                            }else{
                                auxTA.acord.alt6=true;
                                auxTA.alt=true;
                            }
                        }
                        break;
                    case 7:
                        if(auxTA.acord.alt7){
                            auxTA.acord.alt7=false;
                        }else{
                            if(auxTA.alt){
                                auxTA.alt=false;
                            }else{
                                auxTA.acord.alt7=true;
                                auxTA.alt=true;
                            }
                        }
                        break;
                    default:
                        System.out.println("Alteració no reconeguda.");
                        break;
                    }
                    permisosBotons("Ac");
                    bAuxTA.updateWH();
                    descriu("Ac");

                }else if(auxTA.acord.grau==7 && auxTA.acord.sept){ //si és grau X i té 7a, tria 7M (no marco de
cap manera la 7m específica, perquè demanant 7 qualsevol és el mateix).
                    if(auxTA.septMajor){
                        auxTA.septMajor=false;
                    }else{
                        auxTA.septMajor=true;
                    }
                    System.out.println("ID="+(auxTA.id)+" | Sèptima Major: "+F.siNo(auxTA.septMajor));
                    bAuxTA.updateWH();
                    descriu("Ac");
                }else{
                    System.out.println("No hi ha alteracions opcionals disponibles.");
                }

                clickBot=true;
```

```
                if(altOp!=0){ //(això només escriu)
                    System.out.println("ID="+(auxTA.id)+" | Alteració obligada: "+F.siNo(auxTA.alt));
                    if(auxTA.alt){
                        switch(altOp%10){
                        case 9:
                            System.out.println("     | 9a rebaixada: "+F.siNo(auxTA.acord.alt9));
                            break;
                        case 6:
                            System.out.println("     | 6è pujat: "+F.siNo(auxTA.acord.alt6));
                            break;
                        case 7:
                            System.out.println("     | Sensible: "+F.siNo(auxTA.acord.alt7));
                            break;
                        }
                    }
                }
            }
            if(!clickBot && click.x<bDif.x+bDif.w/2f && click.y<bMes.y+bMes.h/2f){ //marge esquerra (opcions
avançades)
                if(click.y>bMes.y-bMes.h/2f){ //botó més quantitat
                    if(!auxTA.tcp){ //només faig anar els botons + - quan no hi ha tcp
                        if(auxTA.n<nTCPAc || (!autoTCP)){ //sóc per sota el límit o tinc desactivat el tcp automàtic
                            auxTA.n++;
                            System.out.println("ID="+(auxTA.id)+" | n="+(auxTA.n));
                        }else{ //tinc tcp automàtic i he arribat al límit
                            auxTA.tcp=true; //activo tcp
                            System.out.println("N elevada. Proposta de TCP.\n"
                                    + "ID="+(auxTA.id)+" | TCP: "+F.siNo(auxTA.tcp));
                        }
                        permisosBotons("Ac");
                        bAuxTA.updateWH();
                        descriu("Ac");
                    }else{
                        System.out.println("TCP activat. Desactiva'l per fer canvis de quantitat.");
                    }
                }else if(click.y>bMenys.y-bMenys.h/2f){ //botó menys quantitat
                    if(!auxTA.tcp){ //només faig anar els botons + - quan no hi ha tcp
                        if(auxTA.n>1){
                            auxTA.n--;
                            System.out.println("ID="+(auxTA.id)+" | n="+(auxTA.n));
                        }else{
                            System.out.println("No pots posar-ne menys.");
                        }
                        permisosBotons("Ac");
                        bAuxTA.updateWH();
                        descriu("Ac");
                    }else{
                        if(autoTCP && auxTA.n>=nTCPAc){ //permeto tornar enrere quan el tcp s'ha activat
automàticament
                            auxTA.tcp=false;
                            System.out.println("Enrere.\n"
                                    + "ID="+(auxTA.id)+" | TCP: "+F.siNo(auxTA.tcp));
                            permisosBotons("Ac");
                            bAuxTA.updateWH();
                            descriu("Ac");
                        }else{
                            System.out.println("TCP activat. Desactiva'l per fer canvis de quantitat.");
                        }
                    }
                }else if(click.y>bTCP.y-bTCP.h/2f){ //botó tants com puguis
                    if(auxTA.tcp){
                        auxTA.tcp=false;
                        if(auxTA.n>=nTCPAc){
                            autoTCP=false;
                        }
                    }else{
                        auxTA.tcp=true;
                    }
                    permisosBotons("Ac");
                    bAuxTA.updateWH();
                    descriu("Ac");
                    System.out.println("ID="+(auxTA.id)+" | TCP: "+F.siNo(auxTA.tcp));

                }else if(click.y>bDif.y-bDif.h/2f){ //botó diferents
                    if(pucDifAc){ //puc diferent acord
                        if(pucDifInv){ //puc les dues coses
                            if(auxTA.difAc){ //da
                                if(auxTA.difInv){ // da+di -> res
                                    auxTA.difAc=false;
```

```
                            auxTA.difInv=false;
                        }else{ // da -> di
                            auxTA.difAc=false;
                            auxTA.difInv=true;
                        }
                    }else{ //no da
                        if(auxTA.difInv){ //di -> da+di
                            auxTA.difAc=true;
                        }else{ //res -> da
                            auxTA.difAc = true;
                        }
                    }
                    System.out.println("ID="+(auxTA.id)+" | Diferent Acord: "+F.siNo(auxTA.difAc)+"\n"
                            + "     | Diferent Inversió: "+F.siNo(auxTA.difInv));
                }else{ //només acord
                    if(auxTA.difAc){
                        auxTA.difAc=false;
                    }else{
                        auxTA.difAc=true;
                    }
                    System.out.println("ID="+(auxTA.id)+" | Diferent Acord: "+F.siNo(auxTA.difAc));
                }
            }else if(pucDifInv){ //puc diferent inversió
                if(auxTA.difInv){
                    auxTA.difInv=false;
                }else{
                    auxTA.difInv=true;
                }
                System.out.println("ID="+(auxTA.id)+" | Diferent Inversió: "+F.siNo(auxTA.difInv));
            }else{ //no puc res
                System.out.println("No pots demanar diferents en aquest context");
            }
            bAuxTA.updateWH();
            descriu("Ac");

        }else if(click.y>bEnll.y-bEnll.h/2f){ //botó enllaçats
            if(pucEnll){
                if(auxTA.enll){
                    auxTA.enll=false;
                }else{
                    auxTA.enll=true;
                }
                bAuxTA.updateWH();
                descriu("Ac");
            }else{
                System.out.println("No pots demanar enllaçats en aquest context");
            }
        }
    }
    if(!clickBot){
        if(Math.pow(click.x-bTorna.x, 2)+Math.pow(click.y-bTorna.y, 2)<Math.pow(bTorna.w/2f,2)){
            if(editant){
                System.out.println("Descartar canvis. Tornant a pantalla d'ingredients");
                bAuxTA.shrink();
                posiciona();
                editant=false;
            }else{
                System.out.println("Tornar a tria de tipus. Descartar ingredient.");
                bTipusAcord.removeValue(bAuxTA, true); //esborro el botó de l'actual
                tipusAcord.removeValue(auxTA, true); //i esborro l'actual (true és fer servir == en lloc de
equals)

                posaElsBotonsPer("triaTipus");
                triaTipus = true;
            }
            triaAcord = false;

            clickBot=true;
        }else if(Math.pow(click.x-bDesa.x, 2)+Math.pow(click.y-bDesa.y, 2)<Math.pow(bDesa.w/2f,2)){
            System.out.println("\nDesar ingredient:");
            bAuxTA.printLn(); //escriu el text d'ingredient per consola

            bAuxTA.shrink(); //faig petites les mides del botó per poder-lo dibuixar petit a la pantalla
d'ingredients.
            posiciona();
            triaAcord=false;

            clickBot=true;
        }
    }
```

```
        }else{
            /**- - - PANTALLA INGREDIENTS - - -*/
            if(!vullEsbos){
                if(click.y>bAfegeix.y-bAfegeix.w/2f && click.y<bAfegeix.y+bAfegeix.w/2f){ //tots dos botons són dins
les mateixes cotes verticals
                    if(click.x>bAfegeix.x-bAfegeix.w/2f && click.x<bAfegeix.x+bAfegeix.w/2f){ //botó d'afegir
                        if(bTipusAcord.size+bTipusCadencia.size+bTipusRes.size<maxIngs){
                            System.out.println("Afegir ingredient. Mostrant tipus disponibles.");
                            posaElsBotonsPer("triaTipus");
                            triaTipus = true;
                            clickBot = true;
                        }else{
                            System.out.println("Arribat al màxim nombre d'ingredients permesos.");
                        }
                    }else if(click.x>bEdita.x-bEdita.w/2f && click.x<bEdita.x+bEdita.w/2f){ //botó d'editar
                        if(ingAct!=-1){
                            System.out.println("Editant l'ingredient amb ID="+ingAct);
                            if(tipusDe(ingAct)=="Ac"){ //si l'ingredient que vull editar és de tipus Acord...
                                auxTA=tipusAcord.get(tipusAcord.indexOf(new IngAcord(ingAct), false));
                                bAuxTA=bTipusAcord.get(bTipusAcord.indexOf(new BotIA(ingAct), false));

                                permisosBotons("Ac");
                                descriu("Ac");
                                bAuxTA.updateWH();

                                editant=true;
                                posaElsBotonsPer("triaAcord");
                                triaAcord = true;
                            }else if(tipusDe(ingAct)=="Cad"){ //equivalent per tipus Cadència
                                auxTC=tipusCadencia.get(tipusCadencia.indexOf(new IngCadencia(ingAct), false));
                                bAuxTC=bTipusCadencia.get(bTipusCadencia.indexOf(new BotIC(ingAct), false));

                                permisosBotons("Cad");
                                descriu("Cad");
                                bAuxTC.updateWH();

                                editant=true;
                                posaElsBotonsPer("triaCadencia");
                                triaCadencia = true;
                            }else if(tipusDe(ingAct)=="Res"){
                                auxTR=tipusRes.get(tipusRes.indexOf(new IngRes(ingAct), false));
                                bAuxTR=bTipusRes.get(bTipusRes.indexOf(new BotIR(ingAct), false));

                                editant=true;
                                posaElsBotonsPer("triaRes");
                                triaRes = true;
                            }

                        }else{
                            System.out.println("No has seleccionat cap ingredient per editar.");
                        }
                        clickBot = true;
                    }else if(click.x>bEAmb.x-bEAmb.w/2f && click.x<bEAmb.x+bEAmb.w/2f){ //botó enllaç amb..
                        System.out.println("La funció d'enllaçar ingredients encara no ha estat implementada.");
                    }else if(click.x>bEsborra.x-bEsborra.w/2f && click.x<bEsborra.x+bEsborra.w/2f){ //botó
d'esborrar
                        if(ingAct!=-1){
                            System.out.println("Esborrant l'ingredient amb ID="+ingAct);
                            bTipusAcord.removeValue(new BotIA(ingAct), false); //demano que esborri dins de tots els
tipus
                            tipusAcord.removeValue(new IngAcord(ingAct), false); //(com que l'id és únic dins el
conjunt de tots els tipus no esborrarà res més)
                            bTipusCadencia.removeValue(new BotIC(ingAct), false);
                            tipusCadencia.removeValue(new IngCadencia(ingAct), false);
                            bTipusRes.removeValue(new BotIR(ingAct), false);
                            tipusRes.removeValue(new IngRes(ingAct), false);
                            posiciona();
                        }else{
                            System.out.println("No has seleccionat cap ingredient per esborrar.");
                        }
                        clickBot = true;
                    }
                }
                ingAct=-1; //trec l'actual (ja se'n posarà un de nou si he polsat un ingredient)
                if(!clickBot){
                    ii=0;
                    for(BotIA b: bTipusAcord){
                        if(click.x>b.x-b.h*0.2f && click.x<b.x+b.w+b.h*0.2f
                        && click.y>b.y-b.h*0.2f && click.y<b.y+b.h*0.8){
                            System.out.println("Seleccionat ingredient nº"+ii+" amb ID="+b.iAc.id);
```

```
                        ingAct=b.iAc.id;
                        clickBot=true;
                        break;
                    }
                    ii++;
                }
                if(ingAct==-1){ //comprovo si he clicat alguna cadència (si no havia clicat cap acord, només)
                    for(BotIC b: bTipusCadencia){
                        if(click.x>b.x-b.h*0.2f && click.x<b.x+b.w+b.h*0.2f
                        && click.y>b.y-b.h*0.2f && click.y<b.y+b.h*0.8){
                            System.out.println("Seleccionat ingredient n°"+ii+" amb ID="+b.iCad.id);
                            ingAct=b.iCad.id;
                            clickBot=true;
                            break;
                        }
                        ii++;
                    }
                }
                if(ingAct==-1){ //mateix per les RI
                    for(BotIR b: bTipusRes){
                        if(click.x>b.x-b.h*0.2f && click.x<b.x+b.w+b.h*0.2f
                        && click.y>b.y-b.h*0.2f && click.y<b.y+b.h*0.8){
                            System.out.println("Seleccionat ingredient n°"+ii+" amb ID="+b.iRI.id);
                            ingAct=b.iRI.id;
                            clickBot=true;
                            break;
                        }
                        ii++;
                    }
                }
            }
        }else{ // (si vullEsbos)
            if(click.x>bCancel.x-bCancel.w/2f && click.x<bCancel.x+bCancel.w/2f //botó de cancel·lar
            && click.y>bCancel.y-bCancel.h/2f && click.y<bCancel.y+bCancel.h/2f){
                System.out.println("Desfer pas a pantalla de l'esbós.");
                bFet.x = (bEAmb.x+bEsborra.x)/2;
                bFet.w = (bEsborra.x-bEAmb.x-bEAmb.w*2);

                vullEsbos = false;
                clickBot = true;
            }
        }
        if(!clickBot && click.x>bFet.x-bFet.w/2f && click.x<bFet.x+bFet.w/2f
                    && click.y>bFet.y-bFet.h/2f && click.y<bFet.y+bFet.h/2f){ //botó "Fet"
            if(vullEsbos){
                System.out.println("Passant a pantalla de l'esbós.");
                app.setScreen(app.pant); //salto a la pantalla de l'esbós
            }else{
                System.out.println("Tria d'ingredients acabada. Demanant confirmació per anar a l'esbós.");
                bFet.x = (int) (Pant.realW*Pant.ratio*1.5f/5f);
                bFet.w = (int) (Pant.realW*Pant.ratio*2.5f/5f);
                vullEsbos = true;
            }
        }

    }
    return true;
};


@Override
public void resize(int width, int height) {
    encuadre.update(width, height);

    Pant.realW = Gdx.graphics.getWidth(); //faig els càlculs directament sobre les variables de la pantalla
principal
    Pant.realH = Gdx.graphics.getHeight();
    Pant.ratio = (float) Pant.screenH/Pant.realH;

    /**- - Tria del tipus d'ingredient - -*/
    bTipusR.x = (int) (Pant.realW*Pant.ratio*0.2f);
    bTipusC.x = (int) (Pant.realW*Pant.ratio*0.5f);
    bTipusA.x = (int) (Pant.realW*Pant.ratio*0.8f);

    /**- - Ingredient tipus cadència - -*/
    xIngCad = (int) (((bDif.x+bDif.w/2f)+(Pant.realW*Pant.ratio))/2);

    /**- - Ingredient tipus acord - -*/
    for(BotQ b: bGraus){
        b.x = (int) (Pant.realW*Pant.ratio-75f);
```

```
        }

        xIngAc = (int) (((bDif.x+bDif.w/2f)+(bGraus.get(0).x-bGraus.get(0).w/2f))/2);

        b7a.x  = (int) (xIngAc-225);
        bDom.x = (int) (xIngAc-75 );
        bInv.x = (int) (xIngAc+75);
        bAlt.x = (int) (xIngAc+225);


        /**- - Pantalla dels ingredients - -*/
        posiciona();
        bEsborra.x = (int) (Pant.realW*Pant.ratio-100);

        if(vullEsbos){
            bFet.x = (int) (Pant.realW*Pant.ratio*1.5f/5f);
            bFet.w = (int) (Pant.realW*Pant.ratio*2.5f/5f);

            bCancel.x = (int) (Pant.realW*Pant.ratio*4/5f);
            bCancel.w = (int) (Pant.realW*Pant.ratio*1.5f/5f);
        }else{
            bFet.x = (bEAmb.x+bEsborra.x)/2;
            bFet.w = (bEsborra.x-bEAmb.x-bEAmb.w*2);
        }


        /**- - General - -*/
        posaElsBotonsPer();

        camera.position.set(Pant.realW/2f*Pant.ratio,Pant.screenH/2f,0);
    }

    @Override
    public void show() {
        vullEsbos = false; //poso les coses a zero per si estaven mogudes.
        posaElsBotonsPer();

        Gdx.input.setInputProcessor(this); //canvio el receptor d'entrades a aquesta pantalla

        posiciona(); //per si estic tornant d'una pantalla diferent.
    }

    @Override
    public void pause() {
    }
    @Override
    public void resume() {
    }
    @Override
    public void hide() {
    }
    @Override
    public void dispose() {
    }

    /******* COMPROVACIONS DE PERMISOS *******/

    public void permisosBotons(String tipus){
        if(tipus=="Cad"){
            if(auxTC.sept || auxTC.triada || (auxTC.n<2 && !auxTC.tcp)){ //TODO canviar això quan permeti DS
                pucDif = false;
                auxTC.dif = false;
            }else{
                pucDif = true;
            }
        }else if(tipus=="Ac"){ //permisos per la pantalla d'acords
            //permís dominant
            pucDom = false; //de moment no permeto Dominants Secundàries
            //permís alteracions opcionals
            altOp = H.opcionals(auxTA.acord);
            if(auxTA.acord.grau!=7 && altOp!=0){ //si no és qualsevol (X) i té opcional...
                pucAlt=true;
            }else{
                pucAlt=false;
            }
            //permís diferents acords
            if((auxTA.acord.grau==7 || (altOp!=0 && !auxTA.alt)) && (auxTA.n>1 || auxTA.tcp)){ //si demano grau
qualsevol (X) (o té opcionals no obligades) i en demano més d'un, puc demanar que siguin diferents
                pucDifAc=true;
            }else{
```

```
                pucDifAc=false;
                auxTA.difAc=false; //ho trec si estava posat
            }
            //permís diferents inversions
            if(auxTA.acord.inv==0 && (auxTA.n>1  || auxTA.tcp)){ //si no demano cap inversió en concret, i demano
més d'un acord, puc demanar diferents inversions
                pucDifInv=true;
            }else{
                pucDifInv=false;
                auxTA.difInv=false; //ho trec si estava posat
            }
            //permís enllaçades
            if(auxTA.acord.grau==7 && auxTA.n==2 && !auxTA.tcp){ //si demano grau qualsevol (X) i quantitat 2, puc
demanar que siguin enllaçats
                pucEnll=true;
            }else{
                pucEnll=false;
                auxTA.enll=false; //ho trec si estava posat
            }
        }
    }

    /****** CÀLCUL DE LA POSICIÓ DELS ELEMENTS *****/

    /**- - Posició dels ingredients i el botó del menú a pantalla d'ingredients - -*/

    public void posiciona(){
        if(bTipusAcord.size+bTipusCadencia.size+bTipusRes.size>0){
            int wMax=0;
            for(BotIA b: bTipusAcord){
                wMax = (int) Math.max(wMax, b.w+b.h*0.2f);
            }
            for(BotIC b: bTipusCadencia){
                wMax = (int) Math.max(wMax, b.w+b.h*0.2f);
            }
            for(BotIR b: bTipusRes){
                wMax = (int) Math.max(wMax, b.w+b.h*0.2f);
            }

            bMenu.x = (int) (Pant.realW*Pant.ratio/2f-(wMax+dMI)/2f+0.25f*bMenu.w);

            ii=0;
            for(BotIA b: bTipusAcord){
                b.x = bMenu.x + dMI;
                b.y = (int) (yIngs+dIngs*((bTipusAcord.size+bTipusCadencia.size+bTipusRes.size-1)/2f-ii)-0.3f*b.h);
//no són 0.5h perquè la y dels botons és a la base del grau
                ii++;
            }
            //mantinc la ii, de manera que a efectes pràctics és com un sol bucle amb tots els botons
            for(BotIC b: bTipusCadencia){
                b.x = bMenu.x + dMI;
                b.y = (int) (yIngs+dIngs*((bTipusAcord.size+bTipusCadencia.size+bTipusRes.size-1)/2f-ii)-0.3f*b.h);
                ii++;
            }
            for(BotIR b: bTipusRes){
                b.x = bMenu.x + dMI;
                b.y = (int) (yIngs+dIngs*((bTipusAcord.size+bTipusCadencia.size+bTipusRes.size-1)/2f-ii)-0.3f*b.h);
                ii++;
            }
            yClau = (int) (yIngs-dIngs*((bTipusAcord.size+bTipusCadencia.size+bTipusRes.size)/2f)-4);
            hClau = (int) (dIngs*(bTipusAcord.size+bTipusCadencia.size+bTipusRes.size)+9);
        }else{
            bMenu.x = (int) (Pant.realW*Pant.ratio/2f-(-F.align("bNotes",'r',"Encara no hi ha
ingredients")+dMI)/2f+0.25f*bMenu.w);
        }
    }

    /**- - - Posició dels botons mòbils segons pantalla - - -*/

    public void posaElsBotonsPer(){
        if(triaCadencia){
            posaElsBotonsPer("triaCadencia");
        }else if(triaRes){
            posaElsBotonsPer("triaRes");
        }else if(triaAcord){
            posaElsBotonsPer("triaAcord");
        }else if(triaTipus){
            posaElsBotonsPer("triaTipus");
        }
    }
```

```
    public void posaElsBotonsPer(String pantalla){
        if(pantalla=="triaRes"){
            bRIs.x = (int) (Pant.realW*Pant.ratio*3/6f);
            bRI7.x = (int) (Pant.realW*Pant.ratio*5/6f);

            bTorna.y = (int) (Pant.screenH/2);
            bTorna.x = (int) (Pant.realW*Pant.ratio*1/6f);
        }else if(pantalla=="triaCadencia"){
            b7a.x = xIngCad;
            b7a.w = (int) ((Pant.realW*Pant.ratio-bDif.w*1f)-(xBarraE+bDif.w*(0.5f+1f)));

            bMes.y = (int) (Pant.screenH/2f+Pant.screenH*1.5f/9f+10f);
            bMenys.y = (int) (Pant.screenH/2f+Pant.screenH*0.5f/9f+10f);
            bTCP.y = (int) (Pant.screenH/2f-Pant.screenH*0.5f/9f);
            bDif.y = (int) (Pant.screenH/2f-Pant.screenH*1.5f/9f-10f);


            bTorna.y = (int) (Pant.screenH*8/9f);
            bTorna.x = (int) (xBarraE);
        }else if(pantalla=="triaAcord"){
            b7a.x = xIngAc-225;
            b7a.w = 130;

            bMes.y = (int) (Pant.screenH/2f+Pant.screenH*2/9f+14f);
            bMenys.y = (int) (Pant.screenH/2f+Pant.screenH/9f+14f);
            bTCP.y = (int) (Pant.screenH/2f+4);
            bDif.y = (int) (Pant.screenH/2f-Pant.screenH/9f-6f);

            bTorna.y = (int) (Pant.screenH*8/9f);
            bTorna.x = (int) (xBarraE);
        }else if(pantalla=="triaTipus"){
            bTorna.y = (int) (100);
            bTorna.x = (int) (100);
        }
    }

    /******* CÀLCUL DE LA DESCRIPCIÓ ******/
    public void descriu(String tipus){
        desc=""; //buido la variable
        if(tipus=="Cad"){
            /**- - - Descipció de cadències - - -*/

            //Tram 1: Número/"Tantes"
            if(auxTC.tcp){
                desc+="Tantes";
            }else{
                desc+=F.fem(F.nomNums(auxTC.n));
            }
            //Tram 2: "cadències trencades"
            if(auxTC.tcp || auxTC.n>1){
                desc+=" cadències trencades";
            }else{
                desc+=" cadència trencada";
            }
            //Tram 3: "tríada"/"amb sèptima"
            if(auxTC.sept){
                desc+=" amb sèptima";
            }else if(auxTC.triada){
                desc+=" ";
                if(auxTC.tcp){
                    desc+=F.plurals("tríada",2);
                }else{
                    desc+=F.plurals("tríada",auxTC.n);
                }
            }
            //Tram 4: diferents
            if(auxTC.dif){
                desc+=" diferents";
            }
            //Tram8: final de TCP
            if(auxTC.tcp){
                desc+=" com puguis";
            }
            desc+=".";

        }else if(tipus=="Ac"){
            /**- - - Descipció d'acords - - -*/

            //Tram 1: Número/"Tants"
            if(auxTA.tcp){
```

```
        desc+="Tants";
    }else{
        desc+=F.nomNums(auxTA.n);
    }
    //Tram2: Grau/"Acord"
    if(auxTA.acord.grau==7){ //qualsevol
        if(auxTA.tcp){
            desc+=F.plurals(" acord", 2); //si he dit "Tants" també vull plural, encara que estigui marcat un
        }else{
            desc+=F.plurals(" acord", auxTA.n);
        }
    }else{ //grau concret
        desc+=" ";
        if(auxTA.tcp){
            desc+=F.plurals(H.graus[auxTA.acord.grau], 2);
        }else{
            desc+=F.plurals(H.graus[auxTA.acord.grau], auxTA.n);
        }
    }
    //Tram3: "tríada"/"amb sèptima" (+"major")
    if(auxTA.acord.sept){
        desc+=" amb sèptima";
        if(auxTA.septMajor){
            desc+=" major";
        }
    }else if(auxTA.triada){
        desc+=" ";
        if(auxTA.tcp){
            desc+=F.plurals("tríada",2);
        }else{
            desc+=F.plurals("tríada",auxTA.n);
        }
    }
    //Tram4: inversions
    if(auxTA.acord.inv!=0){
        desc+=" en ";
        if(auxTA.acord.inv==-1){
            desc+=H.inversions[0]; //estat fonamental
        }else{
            desc+=H.inversions[auxTA.acord.inv];
        }
    }
    //Tram5: altOp
    if(auxTA.alt){
        desc+=" amb";
        switch(H.opcionals(auxTA.acord)%10){
        //TODO afegir case 9 quan tingui manera de marcar sèptima de sensible (7s/7ds)
        case 6:
            if(auxTA.acord.alt6){
                desc+=" el sisè pujat";
            }else{
                desc+=" el sisè natural";
            }
            break;
        case 7:
            if(auxTA.acord.alt7){
                desc+=" sensible";
            }else{
                desc+=" subtònica";
            }
            break;
        }
    }
    //Tram6: Diferents
    if(auxTA.difAc){
        desc+=" diferents";
    }
    if(auxTA.difInv){
        if(auxTA.difAc){
            desc+=" i";
        }
        desc+=" en diferents inversions";
    }
    //Tram7: Enllaçats
    if(auxTA.enll){
        desc+=" enllaçats";
    }
    //Tram8: final de TCP
    if(auxTA.tcp){
        desc+=" com puguis";
```

```
            }
            desc+=".";
        }
    }

    /**- - RETORNA EL TIPUS D'UN INGREDIENT DONAT EL SEU ID - -*/

    public String tipusDe(int id){
        if(tipusAcord.contains(new IngAcord(id), false)){
            return "Ac";
        }else if(tipusCadencia.contains(new IngCadencia(id), false)){
            return "Cad";
        }else if(tipusRes.contains(new IngRes(id), false)){
            return "Res";
        }else{
            System.out.println("No existeix cap ingredient amb ID="+id);
            return null;
        }
    }
}
```

A.1.24. [PInici.java] – Pantalla Inicial

```
package com.rusca8.hEditor;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.Screen;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.OrthographicCamera;
import com.badlogic.gdx.graphics.g2d.SpriteBatch;
import com.badlogic.gdx.utils.viewport.FitViewport;
import com.badlogic.gdx.utils.viewport.Viewport;

public class PInici implements Screen {
    hEditor app;

    OrthographicCamera camera;
    Viewport encuadre;
    SpriteBatch lote;

    float integral; //suma de deltes (per assegurar que es mostra el títol una mica de temps)
    boolean tincImatge; //per carregar la resta quan ja hi ha una imatge mostrant-se (i.e. el primer fotograma ha
passat)
    boolean carregat;

    public PInici(hEditor app){
        this.app = app;
        camera = new OrthographicCamera();
        encuadre = new FitViewport(800,800,camera);
        lote = new SpriteBatch();
        encuadre.apply(true);

        integral=0;
        tincImatge=false;
        carregat=false;

        Recursos.preCarrega();
    }
    @Override
    public void render(float delta) {
        Gdx.gl.glClearColor(S.fons.r,S.fons.g,S.fons.b,S.fons.a);
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);

        integral+=delta;

        camera.update();
        lote.setProjectionMatrix(camera.combined);

        lote.begin();
            lote.draw(Recursos.inici, 0, 0, 800, 800);
        lote.end();

        if(tincImatge && !carregat){ //un cop tinc el primer fotograma, carrego l'aplicació
            carrega();

            carregat=true;
        }

        if(carregat && integral>0.5f){ //quan estigui carregada o bé després de 2 segons, salto a la primera
```

```
pantalla de debò
            this.dispose();
            app.setScreen(new PMenu(app)); //tampoc deso aquesta pantalla
        }

        tincImatge=true; //tan bon punt acabi el primer fotograma tindrem imatge fins a nova ordre
    }

    @Override
    public void resize(int width, int height) {
        encuadre.update(width, height);
    }
    @Override
    public void show() {
    }

    @Override
    public void pause() {
    }
    @Override
    public void resume() {
    }
    @Override
    public void hide() {
    }
    @Override
    public void dispose() { //coses de què ens hem de desfer per alliberar recursos
        lote.dispose();
    }

    public void carrega(){
        Recursos.carrega();
        N.carrega();

        app.pIng = new PIng(app);
        app.pant = new Pant(app);
        app.pant2 = new Pant2(app);

        app.setScreen(this); //sembla que quan inicialitza les altres pantalles perd el nort. Això ho ha arreglat.
    }
}
```

A.1.25. [PMenu.java] – Pantalla del menú

```
package com.rusca8.hEditor;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.InputAdapter;
import com.badlogic.gdx.Screen;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.OrthographicCamera;
import com.badlogic.gdx.graphics.g2d.SpriteBatch;
import com.badlogic.gdx.math.Vector3;
import com.badlogic.gdx.utils.viewport.ExtendViewport;
import com.badlogic.gdx.utils.viewport.Viewport;

public class PMenu extends InputAdapter implements Screen {
    hEditor app;

    OrthographicCamera camera;
    Viewport encuadre;
    SpriteBatch lote;

    Vector3 click = new Vector3(0,0,0);

    int xFlam = 50;
    int wFlam = 350;

    //botons
    int dB = 25; //distància entre botons
    BotQ bNou;
    int pesNou = 5;
    BotQ bAbout;
    int pesAbout = 2;

    public PMenu(hEditor app){
        this.app = app;

        lote = new SpriteBatch();
```

```
        camera = new OrthographicCamera();
        encuadre = new ExtendViewport(0,Pant.screenH,camera);
        encuadre.apply(false); //explicacions de tot això a "Pant"

        //botons
        bNou = new BotQ((Pant.realW*Pant.ratio+(2*xFlam+0.69f*wFlam))/2,0,Pant.realW*Pant.ratio-
(2*xFlam+wFlam*0.69f)-2*dB,0,"Nou Esquema");
        bAbout = new BotQ(bNou.x, 0, bNou.w, 0, "Quant a...");

        posiciona();
    }

    @Override
    public void render(float delta) {
        delta = Math.min(delta, 0.25f);

        Gdx.gl.glClearColor(S.fons.r,S.fons.g,S.fons.b,S.fons.a);
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);

        camera.update();
        lote.setProjectionMatrix(camera.combined);

        lote.begin();
            lote.setColor(S.b_marc);
            lote.draw(Recursos.rect, 0, 0, 2*xFlam+wFlam*0.69f, Pant.screenH);
            lote.setColor(S.fons);
            lote.draw(Recursos.flamingo, xFlam, 50, wFlam, 2*wFlam);

            /**** BOTONS ****/
            //botó de nou esquema
            lote.setColor(S.b_marc);
            lote.draw(Recursos.rect, bNou.x-bNou.w/2f, bNou.y-bNou.h/2f, bNou.w, bNou.h);
            lote.setColor(S.b_normal);
            lote.draw(Recursos.rect, bNou.x-bNou.w/2f+12f, bNou.y-bNou.h/2f+12f, bNou.w-24f, bNou.h-24f);
            Recursos.font.setColor(S.b_text);
            Recursos.font.draw(lote, F.keep(bNou.text),bNou.x+F.align(),bNou.y+F.vAlign());

            //botó d'informació
            lote.setColor(S.b_marc);
            lote.draw(Recursos.rect, bAbout.x-bAbout.w/2f, bAbout.y-bAbout.h/2f, bAbout.w, bAbout.h);
            lote.setColor(S.b_normal);
            lote.draw(Recursos.rect, bAbout.x-bAbout.w/2f+12f, bAbout.y-bAbout.h/2f+12f, bAbout.w-24f, bAbout.h-
24f);
            Recursos.font.setColor(S.b_text);
            Recursos.font.draw(lote, F.keep(bAbout.text),bAbout.x+F.align(),bAbout.y+F.vAlign());
        lote.end();
    }

    @Override
    public boolean touchDown(int screenX, int screenY, int pointer, int button) {
        click.x = screenX;
        click.y = screenY;
        encuadre.unproject(click);

        if(click.x>2*xFlam+0.69*wFlam){ //a la dreta del marge (als botons)
            if(click.y>bNou.y-bNou.h/2f-dB/2f){ //clic a "Nou esquema"
                System.out.println("Començant nou esquema...");
                this.dispose();
                app.setScreen(new PTon(app));
            }else{ //clic a "Quant a..."
                System.out.println("Secció about...");
                Recursos.carregaAbout();
                app.setScreen(new PAbout(app,this)); //conservo la pantalla del menú perquè sé segur que hi tornaré
en acabat
            }
        }else{
            //aquí el que vulguis fer si clica el flamenc (en cas que vulguis fer res)
        }
        return true;
    };

    @Override
    public void resize(int width, int height) {
        encuadre.update(width, height);

        Pant.realW = Gdx.graphics.getWidth();
        Pant.realH = Gdx.graphics.getHeight();
        Pant.ratio = (float) Pant.screenH/Pant.realH;
```

```
        posiciona();

        camera.position.set(Pant.realW/2f*Pant.ratio,Pant.screenH/2f,0);
    }

    @Override
    public void show() {
        Gdx.input.setInputProcessor(this);
    }

    @Override
    public void pause() {
    }
    @Override
    public void resume() {
    }
    @Override
    public void hide() {
    }

    @Override
    public void dispose() {
        lote.dispose();
    }

    public void posiciona(){
        bNou.h = (Pant.screenH-3*dB)/(pesNou+pesAbout)*pesNou;
        bAbout.h = (Pant.screenH-3*dB)/(pesNou+pesAbout)*pesAbout;

        bNou.y = Pant.screenH-dB-bNou.h/2;
        bAbout.y = dB+bAbout.h/2;

        bNou.x = (int) ((Pant.realW*Pant.ratio+(2*xFlam+0.69f*wFlam))/2);
        bAbout.x = bNou.x;

        bNou.w = (int) (Pant.realW*Pant.ratio-(2*xFlam+wFlam*0.69f)-2*dB);
        bAbout.w = bNou.w;
    }
}
```

A.1.26. [PTon] – Pantalla de la tria de mode

```
package com.rusca8.hEditor;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.InputAdapter;
import com.badlogic.gdx.Screen;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.OrthographicCamera;
import com.badlogic.gdx.graphics.g2d.SpriteBatch;
import com.badlogic.gdx.math.Vector3;
import com.badlogic.gdx.utils.viewport.ExtendViewport;
import com.badlogic.gdx.utils.viewport.Viewport;

public class PTon extends InputAdapter implements Screen {
    hEditor app;

    OrthographicCamera camera;
    Viewport encuadre;
    SpriteBatch lote;

    Vector3 click = new Vector3(0,0,0);

    int rCir = (int) (Pant.realW*Pant.ratio);
    int xCir = (int) (Pant.realW*Pant.ratio/2+(Math.sqrt(Math.pow(rCir, 2)-Math.pow(Pant.screenH/2f, 2))+rCir)/2);
    int yCir = (int) ((Pant.screenH)/2);

    public PTon(hEditor app){
        System.out.println("Carregant tria del mode...");
        this.app = app;

        lote = new SpriteBatch();

        camera = new OrthographicCamera();
        encuadre = new ExtendViewport(0,Pant.screenH,camera);
        encuadre.apply(false); //explicacions de tot això a "Pant"
    }

    @Override
```

```
    public void render(float delta) {
        delta = Math.min(delta, 0.25f);

        Gdx.gl.glClearColor(S.fons.r,S.fons.g,S.fons.b,S.fons.a);
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);

        camera.update();
        lote.setProjectionMatrix(camera.combined);

        lote.begin();
            //Seccions
            lote.setColor(S.fons);
            lote.draw(Recursos.rect,0,0,Pant.realW*Pant.ratio,Pant.screenH);
            lote.setColor(S.b_normal);
            lote.draw(Recursos.cir, xCir-rCir, yCir-rCir, 2*rCir, 2*rCir);

            Recursos.font.setColor(S.b_marc);
            Recursos.font.draw(lote,"Tria el mode:",30,Pant.screenH-30);
            Recursos.xifrat.setColor(S.b_marc);
            Recursos.xifrat.draw(lote,F.keep("*Podràs triar\nla tonalitat més endavant"), 30,
30+F.vAlign("xifrat",'d'));

            //lletres seccions
            Recursos.bigFont.setColor(S.b_marc);

    Recursos.bigFont.draw(lote,F.keep("Major"),Pant.realW*Pant.ratio*0.25f+F.align("bigFont"),yCir+F.vAlign("bigFon
t"));
            Recursos.bigFont.setColor(S.b_marc);

    Recursos.bigFont.draw(lote,F.keep("menor"),Pant.realW*Pant.ratio*0.75f+F.align("bigFont"),yCir+F.vAlign("bigFon
t"));

        lote.end();
    }

    @Override
    public boolean touchDown(int screenX, int screenY, int pointer, int button) {
        click.x = screenX;
        click.y = screenY;
        encuadre.unproject(click);

        if(Math.pow(click.x-xCir, 2)+Math.pow(click.y-yCir, 2)>Math.pow(rCir, 2)){
            System.out.println("Mode Major. Passant a triar ingredients.");
            H.mode=0;
        }else{
            System.out.println("Mode menor. Passant a triar ingredients.");
            H.mode=1;
        }
        this.dispose();
        app.setScreen(app.pIng);
        return true;
    };

    @Override
    public void resize(int width, int height) {
        encuadre.update(width, height);

        Pant.realW = Gdx.graphics.getWidth();
        Pant.realH = Gdx.graphics.getHeight();
        Pant.ratio = (float) Pant.screenH/Pant.realH;

        rCir = (int) (Pant.realW*Pant.ratio);
        xCir = (int) (Pant.realW*Pant.ratio/2+(Math.sqrt(Math.pow(rCir, 2)-Math.pow(Pant.screenH/2f, 2))+rCir)/2);

        camera.position.set(Pant.realW/2f*Pant.ratio,Pant.screenH/2f,0);
    }

    @Override
    public void show() {
        Gdx.input.setInputProcessor(this);
    }

    @Override
    public void pause() {
    }

    @Override
    public void resume() {
    }
```

```
    @Override
    public void hide() {
    }

    @Override
    public void dispose() {
        lote.dispose();
    }

}
```

A.1.27. [Recursos] – Gestió d'imatges, fonts i sons

```
package com.rusca8.hEditor;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.assets.AssetManager;
import com.badlogic.gdx.audio.Sound;
import com.badlogic.gdx.graphics.Color;
import com.badlogic.gdx.graphics.Texture;
import com.badlogic.gdx.graphics.Texture.TextureFilter;
import com.badlogic.gdx.graphics.g2d.BitmapFont;
import com.badlogic.gdx.graphics.g2d.Sprite;
import com.badlogic.gdx.graphics.g2d.freetype.FreeTypeFontGenerator;
import com.badlogic.gdx.graphics.g2d.freetype.FreeTypeFontGenerator.FreeTypeFontParameter;
import com.rusca8.hEditor.Pant2;

public class Recursos {

    /**- - Imatges - -*/
    //pantalla inicial
    public static Texture tex_inici; //pantalla que es mostra abans que comenci tot
    public static Sprite inici;

    public static Texture tex_flamingo; //clau del logo
    public static Sprite flamingo;

    //genèric
    public static Texture tex_rect; //les imatges tenen totes proporcions que permeten mides potència de 2, perquè
si no no sempre funciona el mipMap (crec)
    public static Sprite rect;

    public static Texture tex_cir;
    public static Sprite cir;

    public static Texture tex_nota;
    public static Sprite nota;

    public static Texture tex_sharp; //TODO dibuixar això més maco xD -Agafar l'equivalent als altres dibuixos de
Wikimedia Commons.
    public static Sprite sharp;

    public static Texture tex_natural; //De Wikimedia Commons
    public static Sprite natural;

    public static Texture tex_bemoll; //De Wikimedia Commons
    public static Sprite bemoll;

    public static Texture tex_sol;
    public static Sprite sol;

    public static Texture tex_fa;
    public static Sprite fa;

    public static Texture tex_lligat;
    public static Sprite lligat;

    public static Texture tex_arrow; //icona no elegible per a copyright (elements massa comuns)
    public static Sprite arrow;

    public static Texture tex_tick; //tick de icons8
    public static Sprite tick;

    //per la secció "Quant a.."
    public static Texture tex_avatar; //foto d'avatar
    public static Sprite avatar;

    public static Texture tex_launcher; //icona de l'aplicació (només interior)
    public static Sprite launcher;
```

```
    public static Texture tex_folder; //carpeta, per parlar dels recursos i tal (de icons8)
    public static Sprite folder;

    public static Texture tex_twitter; //per la icona de twitter
    public static Sprite twitter;

    public static Texture tex_github; //per la icona de github
    public static Sprite github;

    /**- - Fonts - -*/
    public static BitmapFont bigFont; //pel grau a la tria d'ingredient acord
    public static BitmapFont font; //estàndard
    public static BitmapFont bMenu; //pel botó del menú quan font és massa gran
    public static BitmapFont bNotes; //pels botons de les notes a la realització
    public static BitmapFont xifrat; //pels xifrats
    public static BitmapFont iQuant; //per la quantitat dels ingredients
    public static BitmapFont iXifrat; //pel xifrat dels ingredients
    public static BitmapFont eXifrat; //Equivalent a la mida xifrat per l'esbós de Pant2.
    /**
    * Copyright Steve Matteson
    *
    * Licensed under the Apache License, Version 2.0 (the "License");
    * you may not use this file except in compliance with the License.
    * You may obtain a copy of the License at
    *
    * http://www.apache.org/licenses/LICENSE-2.0
    *
    * Unless required by applicable law or agreed to in writing, software
    * distributed under the License is distributed on an "AS IS" BASIS,
    * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    * See the License for the specific language governing permissions and
    * limitations under the License.
    **/ //sobre la llicència de la font (Open Sans, de Steve Matteson)

    /**- - Sons - -*/
    public static AssetManager manager = new AssetManager();
    public static boolean carregantSons = false;
    public static int ia=4; //variables per parametritzar la càrrega de sons (índex acústic i so cromàtic)
    public static int so=9;
    public static Sound[] cantants = new Sound[4]; //més aviat violins, en realitat. Quatre: un per hospedar cada
nota de l'acord.

    public static void preCarrega(){
        System.out.println("Carregant pantalla inicial...");
        tex_inici = new Texture(Gdx.files.internal("logoInici.png"),true);
        tex_inici.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        inici = new Sprite(tex_inici);
    }

    public static void carregaAbout(){
        System.out.println("Carregant recursos per about...");

        tex_avatar = new Texture(Gdx.files.internal("yo.png"),true);
        tex_avatar.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        avatar = new Sprite(tex_avatar);

        tex_launcher = new Texture(Gdx.files.internal("launcher_interior.png"),true);
        tex_launcher.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        launcher = new Sprite(tex_launcher);

        tex_folder = new Texture(Gdx.files.internal("icons8_folder.png"),true);
        tex_folder.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        folder = new Sprite(tex_folder);

        tex_twitter = new Texture(Gdx.files.internal("twitterLogo.png"),true);
        tex_twitter.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        twitter = new Sprite(tex_twitter);

        tex_github = new Texture(Gdx.files.internal("githubLogo.png"),true);
        tex_github.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        github = new Sprite(tex_github);
    }

    public static void carrega(){
        System.out.println("Carregant recursos...");

        /**- - Imatges - -*/
        tex_flamingo = new Texture(Gdx.files.internal("flamingoclave.png"), true); // true significa fer MipMaps
        tex_flamingo.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
```

```
        flamingo = new Sprite(tex_flamingo);

        tex_rect = new Texture(Gdx.files.internal("rect.png"), true);
        tex_rect.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        rect = new Sprite(tex_rect);

        tex_cir = new Texture(Gdx.files.internal("circle.png"), true);
        tex_cir.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        cir = new Sprite(tex_cir);

        tex_nota = new Texture(Gdx.files.internal("nota.png"), true);
        tex_nota.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        nota = new Sprite(tex_nota);

        tex_sharp = new Texture(Gdx.files.internal("sharp.png"), true);
        tex_sharp.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        sharp = new Sprite(tex_sharp);

        tex_natural = new Texture(Gdx.files.internal("natural.png"), true);
        tex_natural.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        natural = new Sprite(tex_natural);

        tex_bemoll = new Texture(Gdx.files.internal("bemoll.png"), true);
        tex_bemoll.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        bemoll = new Sprite(tex_bemoll);

        tex_sol = new Texture(Gdx.files.internal("sol.png"), true);
        tex_sol.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        sol = new Sprite(tex_sol);

        tex_fa = new Texture(Gdx.files.internal("fa.png"), true);
        tex_fa.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        fa = new Sprite(tex_fa);

        tex_lligat = new Texture(Gdx.files.internal("lligat.png"), true);
        tex_lligat.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        lligat = new Sprite(tex_lligat);

        tex_arrow = new Texture(Gdx.files.internal("arrow.png"), true);
        tex_arrow.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        arrow = new Sprite(tex_arrow);

        tex_tick = new Texture(Gdx.files.internal("icons8_checkmark_modified.png"), true);
        tex_tick.setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);
        tick = new Sprite(tex_tick);

        //text
        FreeTypeFontGenerator fontGen = new FreeTypeFontGenerator(Gdx.files.internal("OpenSans-Regular.ttf"));
//free font by Google from fontsquirrel.com
        FreeTypeFontParameter fontPar = new FreeTypeFontParameter();

        fontPar.color = Color.valueOf("ffffff");
        fontPar.genMipMaps = true;

        fontPar.size = 80;
        bigFont = fontGen.generateFont(fontPar);
        bigFont.getRegion().getTexture().setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);

        fontPar.size = 50;
        font = fontGen.generateFont(fontPar);
        font.getRegion().getTexture().setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);

        fontPar.size = 45;
        bMenu = fontGen.generateFont(fontPar);
        bMenu.getRegion().getTexture().setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);

        fontPar.size = 39;
        bNotes = fontGen.generateFont(fontPar);
        bNotes.getRegion().getTexture().setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);

        fontPar.size = 35;
        xifrat = fontGen.generateFont(fontPar);
        xifrat.getRegion().getTexture().setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);

        fontPar.size = 31;
        iQuant = fontGen.generateFont(fontPar);
        iQuant.getRegion().getTexture().setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);

        fontPar.size = 28;
        iXifrat = fontGen.generateFont(fontPar);
```

```
        iXifrat.getRegion().getTexture().setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);

        fontPar.size = (int) (35*Pant2.r1e);
        eXifrat = fontGen.generateFont(fontPar);
        eXifrat.getRegion().getTexture().setFilter(TextureFilter.MipMapLinearNearest, TextureFilter.Linear);

        fontGen.dispose();
    }

    static byte ultimAmbNotes;

    public static void carregaSons(){ //funció que gestiona(rà) quins sons s'han de carregar en funció de quines
notes té l'esquema.
        //fitxers d'audio provinents de la Philarmonia Orchestra (violí i contrabaix), sota llicència CC 3.0 BY SA
        byte ii=0;
        ultimAmbNotes=0;

        for(Note[] ac: Pant2.esquemaEnNotes){ //carrego els sons de totes les notes que s'han utilitzat
            for(int i=0;i<4;i++){
                if(ac[i].nom!=-1){ //només faig alguna cosa si hi ha nota
                    so=ac[i].so;
                    ia=ac[i].ia;
                    manager.load("audio/s"+(ia*100+so)+".mp3", Sound.class); //el manager ja té en compte aprofitar
els repetits
                    ultimAmbNotes = ii;
                }
            }
            ii++;
        }

        carregantSons = true;
    }

    public static void playChord(Note[] notesAcord){ //funció que reproduirà un acord donat el seu vector de
notes
        for(int i=0;i<4;i++){
            if(notesAcord[i].nom!=-1){ //si no hi ha nota no faig res
                so=notesAcord[i].so;
                ia=notesAcord[i].ia;
                cantants[i] = manager.get("audio/s"+(ia*100+so)+".mp3", Sound.class);
                cantants[i].play(1f,1f,0.25f*i); //el faig cantar (volum: 0 a 1, alçada: 0=original, panorama: 0 a 1)
            }
        }
    }
}
```

A.1.28. [S.java] – Biblioteca per a la gestió del color

```
package com.rusca8.hEditor;

import com.badlogic.gdx.graphics.Color;

public class S { //Variables i funcions relacionades amb els colors (S de sinestesia).

    /**- - - Colors genèrics - - -*/

    static Color fons = new Color(1f,0.95f,0.86f,1f);
    static Color pentagrama = new Color(0f,0f,0f,1f);
    static Color ingredients = new Color(0.8f,0.8f,0.8f,0.8f);

    static Color tonalitat = new Color(0f,0.44f,0.60f,1f); //color del cercle de la tonalitat i altres elements de
l'entorn gràfic.

    static Color flamingo = new Color(0.6f,0.4f,0f,1f);
    static Color blanc = new Color(1f,1f,1f,1f);

    /**- - - Colors dels botons - - -*/

    static Color b_marc = new Color(0.24f,0.18f,0f,1f);
    static Color b_marc_ressaltat = new Color(0.60f,0.47f,0f,1f);
    static Color b_normal = new Color(0.84f,0.65f,0.31f,1f);
    static Color b_ressaltat = new Color(0.87f,0.69f,0.32f,1f);
    static Color b_premut = new Color(0.96f,0.80f,0.35f,1f);
    static Color b_inactiu = new Color(0.38f,0.33f,0.18f,1f);
    static Color b_text = new Color(0.11f,0.08f,0f,1f);
    static Color b_text_inactiu = new Color(0.17f,0.13f,0f,0.5f); //TODO potser ajustar una mica les lletres en
consonància a la iluminació dels botons.

    /**- - - Colors de les correccions - - -*/
```

```
    static Color graus = new Color(0f,0f,0f,1f);
    static Color correcte = new Color(0.24f,0.63f,0.28f,1f);
    static Color alerta = new Color(0.82f,0.58f,0f,1f); //abans estava a 87 65 0 (m'agrada més el nou)
    static Color error = new Color(0.6f,0f,0f,1f);
    static Color irrellevant = new Color(0f,0f,0f,0.2f);

    static Color cadencies = new Color(0.50f,0.37f,0.19f,1f);

    /**- - - Colors de la interfície - - -*/

    static Color actual = new Color(0f,0.44f,0.60f,1f);
    static Color cursor = new Color(b_marc_ressaltat);

    /**- - - Color segons correcció - - -*/

    public static Color cColor(char c){ //envia el color corresponent a un valor de correcció.
        switch(c){
        case 'c':
            return correcte;
        case 'a':
            return alerta;
        case 'e':
            return error;
        case 'i':
            return irrellevant;
        default:
            return graus;
        }
    }

    /**- - - Canvi de transparència - - -*/

    public static Color newAlpha(Color c, float a){
        return (new Color(c.r,c.g,c.b,a));
    }

    public static Color greyScale(Color c){
        float m = (c.r+c.g+c.b)/3f;
        return new Color(m,m,m,c.a);
    }
}
```

A.2. Codi específic de la versió d'escriptori (HemioliaEditor-desktop)

A.1.29. [DesktopLauncher.java] – Llançador de la versió d'escriptori

```
package com.rusca8.hEditor.desktop;

import com.badlogic.gdx.backends.lwjgl.LwjglApplication;
import com.badlogic.gdx.backends.lwjgl.LwjglApplicationConfiguration;
import com.rusca8.hEditor.hEditor;

public class DesktopLauncher {
    public static void main (String[] arg) {
        LwjglApplicationConfiguration config = new LwjglApplicationConfiguration();
        config.title = "Hemiolia Editor - Rusca8";
        config.width = 750; // 800
        //neutro 600 400, uso 750 450
        config.height = 450; // min pan 600, max pan 450
        new LwjglApplication(new hEditor(), config);
    }
}
```

A.3. Codi específic de la versió per a mòbils (HemioliaEditor-android)

A.1.30. [AndroidLauncher.java] – Llançador de la versió per Android

```
package com.rusca8.learningGDX.android;

import android.os.Bundle;

import com.badlogic.gdx.backends.android.AndroidApplication;
import com.badlogic.gdx.backends.android.AndroidApplicationConfiguration;
import com.rusca8.learningGDX.LearningGDX;
```

```
public class AndroidLauncher extends AndroidApplication {
    @Override
    protected void onCreate (Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        AndroidApplicationConfiguration config = new AndroidApplicationConfiguration();
        initialize(new LearningGDX(), config);
        config.useAccelerometer = false;
        config.useCompass = false;
    }
}
```

A.1.31. [AndroidManifest.xml] – Informació essencial sobre l'app per al sistema

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:installLocation="auto"
    package="com.rusca8.hEditor.android"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="9" android:targetSdkVersion="20" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/GdxTheme" >
        <activity
            android:name="com.rusca8.hEditor.android.AndroidLauncher"
            android:label="@string/app_name"
            android:screenOrientation="landscape"
            android:configChanges="keyboard|keyboardHidden|orientation|screenSize">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```