

## Treball final de grau

**Estudi:** Grau en Disseny i Desenvolupament de Videojocs

**Títol:** Desenvolupament d'un videojoc per a mòbils destinat a infants amb Trastorn de l'Espectre Autista.

**Document:** Memòria

**Alumne:** Marc Piferrer Sagrera

**Tutor:** Gustavo Patow

**Departament:** Informàtica, Matemàtiques Aplicada i Estadística

**Àrea:** Llenguatges i Sistemes Informàtics

**Tutora:** Anna Valls

**Ocupació:** Terapeuta

**Empresa:** "Tots hi som"

**Convocatòria (mes/any):** Juny /2022

## AGRAÏMENTS

En primer lloc, vull agrair als meus tutors, en Gustavo Patow, els seus consells i suport m'han ajudat a veure que el meu projecte pot ser diferent i m'ha tranquil·litzat en els moments més difícils del desenvolupament d'aquest projecte, i a l'Anna Valls, el pilar més important d'aquest projecte i la seva font d'inspiració. També agrair que hagi estat tant disponible per resoldre els meus dubtes i mostrar tant d'interès en el desenvolupament del projecte, sense ella aquest projecte no hagués estat possible.

També vull dedicar una part d'aquest agraïment als meus pares, a la meva germana per a suportar el meu mal humor en moments de molta pressió per la feina, la Universitat i el TFG. Moltes gràcies també per mostrar tant d'interès en el que feia i deixava de fer i per els vostres consells i opinions que molts d'ells han acabat resultant molt útils. Crec que cal afegir la resta de la meva família com poden ser els meus avis per a mostrar sempre orgull cap a la meva persona i cuidar-me com ho han fet sempre.

A la Judit, per entendre que el temps que tenia era molt limitat i animar-me cada dia a treballar amb aquest projecte. Et prometo que recuperarem el temps que he invertit en aquest treball. Gràcies per l'afecte donat tots aquest mesos i escoltar-me en els moments més baixos.

Agrair també als amics fets a la Universitat que en pocs anys hem esdevingut un gran grup dins i fora de l'entorn universitari, gràcies per totes les nits de Discord treballant junts en els nostres projectes i pels plans de desconexió que hem anat organitzant. Moltes gràcies sobretot a vosaltres David, Pau i Quim. A la Marina, que tot i que ja no ens veiem tant com ens agradaria, les conversacions amb tu sempre han estat un moment de tranquil·litat. Més que una amiga ets una germana.

Sense vosaltres aquest projecte no hagués estat el mateix i no l'hauria desenvolupat amb tant d'amor.

## ÍNDEX

1. INTRODUCCIÓ.....	6
1.1 MOTIVACIONS.....	7
1.2 PROPÒSIT .....	7
1.3 OBJECTIUS .....	8
1.4 QUADRE D'AUTOVALORACIÓ.....	8
2. ESTUDI DE VIABILITAT .....	10
2.1 RECURSOS TÈCNICS.....	10
2.2 RECURSOS HUMANS .....	10
2.3 REQUISITS TECNOLÒGICS.....	11
2.4 COST ECONÒMIC.....	11
2.5 ESTUDI DE MERCAT .....	13
2.5.1 CERQUES REALITZADES .....	13
2.5.2 RESULTATS OBTINGUTS .....	14
2.5.3 MATRIU DE COMPETITIVITAT.....	17
2.5.4 CONCLUSIONS ESTUDI DE MERCAT .....	17
2.6 PÚBLIC OBJECTIU I PERFIL DE JUGADOR.....	18
2.7 CONCLUSIÓ DE LA VIABILITAT DEL PROJECTE.....	19
3. PLANIFICACIÓ .....	20
3.1 PLA DE TREBALL.....	21
3.1.1 ABAST DEL PROJECTE .....	22
3.1.2 TASQUES PLANIFICADES.....	23
3.1.3 CRONOGRAMA.....	24
4. MARC DE TREBALL I CONCEPTES PREVIS.....	25
4.1 TRASTORN DE L'ESPECTRE AUTISTA (TEA).....	25
4.2 EXPERIÈNCIA AMB TERAPEUTES I PACIENTS.....	28
4.3 REFERÈNCIES .....	29
4.3.1 DICTAPICTO .....	29
4.3.2 DÍA A DÍA .....	30
4.3.3 CONCLUSIONS REFERÈNCIES.....	31
4.4 MOTOR DE JOC.....	32
4.4.1 UNITY.....	33
4.5 PEGI .....	35
5. DISSENY DEL VIDEOJOC.....	36
5.1 ESTIL .....	36

5.1.1 GÈNERE.....	36
5.1.2 CÀMERA.....	37
5.1.3 GRÀFICS.....	38
5.2 GAMEPLAY.....	39
5.2.1 DEFINICIÓ DE REPTES.....	39
5.2.2 JERARQUIA DE REPTES.....	39
5.2.3 DEFINICIÓ D'ACCIONS.....	45
5.3 MECÀNIQUES.....	45
5.3.1 RECURSOS.....	45
5.3.2 ENTITIES.....	48
5.3.3 DESCRIPCIÓ DE LES MECÀNIQUES.....	51
5.3.4 OBJECTIUS.....	53
5.3.5 CONTROLS BÀSICS.....	53
5.4 FLOWCHART.....	54
5.5 LEVEL DESIGN.....	55
5.5.1 ECONOMIA INTERNA.....	55
5.6 CANVIS REALITZATS DE DISSENY.....	56
5.7 HISTÒRIA.....	57
5.7.1 MÓN DEL JOC.....	57
5.8 INTERFÍCIE GRÀFICA.....	58
5.8.1 MENÚ PRINCIPAL.....	58
5.8.2 CREADOR CALENDARI.....	60
5.8.3 CALENDARI.....	60
5.8.4 MENÚ DE PAUSA.....	61
5.8.5 MENÚ DE PUNTUACIÓ.....	62
5.8.6 NIVELLS.....	62
5.8.6.1 MINIJOC VESTIR-SE.....	63
5.8.6.2 MINIJOC FER LA MOTXILLA.....	63
5.8.6.3 MINIJOC ORDENAR HABITACIÓ.....	64
5.8.6.4 MINIJOC RENTAR DENTS.....	64
5.8.6.5 MINIJOC MENJAR.....	65
5.8.6.6 MINIJOC ORDENAR OBJECTES.....	65
5.8.6.7 MINIJOC DUTXAR-SE.....	66
5.8.6.7 PART D'ORDENAR.....	66
5.9 ART DEL JOC.....	67
5.9.1 ESCENARIS.....	67

5.9.2 PERSONATGES .....	68
5.9.3 PICTOGRAMES.....	68
5.9.4 ALTRES SPRITES .....	69
5.10 EFECTES DE SO .....	69
5.11 PLA DE MÀRQUETING .....	70
5.11.1 PLATAFORMES DE DISTRIBUCIÓ I CONSUM.....	70
5.11.2 COMUNICACIÓ .....	70
6. IMPLEMENTACIÓ I PROVES .....	71
6.1 IMPLEMENTACIÓ.....	72
6.1.1 MENÚ PRINCIPAL (menuManager.cs).....	72
6.1.2 CREADOR DE CALENDARI (CalendarMaker.cs).....	74
6.1.3 CALENDARI (CalendarManager.cs).....	76
6.1.4 MINIJOCS.....	78
6.1.4.1 MINIJOC FER MOTXILLA .....	78
6.1.4.2 MINIJOC VESTIR-SE.....	81
6.1.4.3 MINIJOC ORDENAR HABITACIÓ.....	85
6.1.4.4 MINIJOC RENTAR DENTS .....	87
6.1.4.5 MINIJOC ORDENAR OBJECTES.....	91
6.1.4.6 MINIJOC MENJAR .....	93
6.1.4.7 MINIJOC DUTXAR-SE .....	95
6.2 PROVES.....	98
7. RESULTATS.....	99
8. CONCLUSIONS .....	105
9. TREBALL FUTUR .....	106
10. BIBLIOGRAFIA .....	107
11. ANNEXOS.....	108
11.1 CODI EXPLICACIÓ.....	108
11.2 CODI EXPLICACIÓ SONOR.....	109
11.3 CODI PUNTUACIÓ .....	110
11.3 CODI MENÚ DE PAUSA.....	112
11.4 CODI TEMPORITZADOR .....	113
12. MANUAL D'INSTAL·LACIÓ I D'USUARI.....	114
12.1 INSTALACIÓ ANDROID.....	114
12.2 INSTALACIÓ WINDOWS.....	114
12.3 MANUAL D'USUARI .....	115

## 1. INTRODUCCIÓ

En la indústria dels videojocs hi ha una gran varietat de títols pensats per diferents tipus de públics. Existeixen videojocs d'aventures, viatges on el protagonista ha de complir algun objectiu passant per diferents obstacles i aventures; videojocs de simulacions, que tenen un gran àmbit de possibilitats, des de simuladors de conducció, fins a simuladors d'esports com poden ser el futbol o el bàsquet... Altres videojocs que existeixen són els casual, jocs pensats per a ser usats en moments en el que el jugador té una estona en la que no té res a fer i necessita una distracció. Aquests jocs poden tenir partides ràpides en les que no es perdi una gran quantitat de temps.

No obstant, una característica que comparteixen tots aquests videojocs és que no deixen de ser jocs lúdics, sense cap altre objectiu al seu darrera que el de fer que l'usuari gaudeixi una estona d'aquesta activitat. Aquests jocs necessiten unes certes habilitats tant motrius com mentals que poder no tots els usuaris poden tenir.

Actualment hi ha diferents aplicacions, com pot ser "Dictapicto", creat per la "Fundación Orange", per a ajudar als pacients de Trastorn de l'Espectre Autista. Per ajudar a entendre el que pot dir una persona. No obstant no existeix una varietat gaire gran de videojocs adaptats per aquest tipus de públic.



Figura 1 Captura de Dictapicto

En aquest projecte s'estudiarà la malaltia del TEA per fer la creació d'un joc terapèutic format per diferents minijocs que ajudin als terapeutes a entrenar les accions quotidianes d'una persona.

## 1.1 MOTIVACIONS

La majoria d'estudis, treballs o idees sorgeixen al descobrir un necessitat que no ha estat tractada, o al menys no ho ha estat d'una forma encertada, o sorgeixen al descobrir que hi ha un gran interès o mercat en algun tipus de producte. En el primer cas un exemple seria el desenvolupament de dispositius per facilitar la vida a gent amb alguna discapacitat, mentre que en el segon cas un exemple seria el desenvolupar un joc de tipus "shooter" competitiu, ja que actualment tenen un gran mercat i poden tenir un gran nombre de jugadors.

En el meu cas considero que el meu treball sorgeix de la necessitat de crear una eina per a ajudar a terapeutes a ajudar a nens amb TEA, a recordar i entendre la seva rutina diària per a permetre assolir un mínim de independència. Aquest projecte ha estat desenvolupat per mirar d'ajudar al centre "Tots hi som" en la seva tasca d'ajudar al pacient a gaudir d'una vida independent i de qualitat. Al haver-hi una gran varietat de pacients amb diferents graus de TEA és molt difícil saber on acotar per a que puguin jugar el màxim de persones possibles.

Aquesta malaltia està present en un de cada 68 nens. Per tant, considero que és important donar-li visibilitat i desenvolupar eines per mirar d'ajudar a les persones que tenen la malaltia. Això ho farem mitjançant un joc desenvolupat específicament per aquest col·lectiu.

## 1.2 PROPÒSIT

Vista la baixa quantitat de videojocs pensats per a gent amb TEA, o videojocs que hagin estat adaptats per que els puguin gaudir usuaris amb aquest trastorn, hem pensat que seria interessant crear un joc dissenyat expressament per aquest col·lectiu. El propòsit principal d'aquest projecte serà desenvolupar el prototip d'un videojoc que ajudi a que pacients amb TEA puguin entrenar les activitats de la vida diària i les habilitats necessàries per a dur-les a terme.

Aquest videojoc té la intenció de convertir-se en un Serious Game, no obstant s'ha mirat de fer de manera que pel nen no sigui una teràpia, si no que sigui un moment lúdic.

Com hem dit anteriorment el joc està pensat per un target molt concret, per nens amb TEA. Ja que aquesta patologia té un gran espectre de possibilitats, i no tots els pacients comparteixen les mateixes patologies ni de la mateixa gravetat, he d'aclarir que aquest joc està pensat per pacients amb certes afectacions cognitives.

### 1.3 OBJECTIUS

L'objectiu d'aquest Treball Final de Grau és desenvolupar el prototip d'un videojoc format per minijocs 2D, adaptant les mecàniques i el disseny dels minijocs a les característiques dels pacients que ens indiquin els terapeutes del centre "Tots hi som". Aquest videojoc ha de ser adaptat per dispositius Android, ja que el centre treballen amb Tauletes. Per aconseguir-ho les mecàniques principals del joc seran (arrossegar i col·locar, arrossegar i eliminar, clicar botons,..). També haurem de crear un sistema en el que el terapeuta o el pare del pacient puguin crear un calendari que s'adeqüi a cada pacient per tal de que a base de repetició quedi interioritzat al pacient.

Per a dur a terme el projecte es mantindrà un contacte constant amb els terapeutes del centre, més concretament amb la Senyora Anna Valls, per tal d'obtenir assessorament i un feedback del que es va fent i de les característiques que necessiten que tingui el nostre prototip. El projecte està centrat en l'apartat tècnic, no obstant també s'haurà de treballar l'art i l'apartat gràfica, ja que aquests pacients usen molts de pictogrames per tal d'entendre i comunicar-se més fàcilment. El joc es programarà en Unity i els elements gràfics es realitzaran amb Photoshop. Les tasques a desenvolupar inclouen, però no es limiten a:

- Estudiar els símptomes del Trastorn de l'Espectre Autista.
- Implementar mecàniques bàsiques per minijocs 2D.
- Implementar un sistema per a realitzar calendaris.
- Dissenyar minijocs adaptats al públic target.
- Adaptar el joc amb pictogrames perquè el públic targuet entengui el que s'ha de fer.
- Documentar el projecte.

### 1.4 QUADRE D'AUTOVALORACIÓ

L'equip que realitza un videojoc està format normalment per un equip multidisciplinari, no obstant en aquest cas tant sols i haurà una persona que serà l'encarregada de les diferents parts del projecte. A la taula 1 es mostra una valoració de la quantitat d'esforç que s'ha necessitat en cada aspecte.

Estètica	5%
Narrativa	0%
Mecàniques	65%
Tecnologia	30%

*Taula 1 Quadre d'autoavaluació*



Com podem veure a la taula anterior, a l'apartat on es gastaran més recursos és clarament el de les mecàniques. Com s'ha dit anteriorment, la proposta és que, a través de les mecàniques, els pacients amb l'ajuda del terapeuta, entrenin les tasques més habituals que té una persona en el seu dia a dia. L'ajuda de la terapeuta Anna Valls ens servirà per decidir quines són les activitats que haurem de fer minijoc i de quina manera podem representar-les adequadament.

La tecnologia és el segon bloc en el que si ha de dedicar més esforç ja que treballarem amb el motor Unity i l'editor d'imatges Photoshop. A més a més, s'hauran de tenir en compte les característiques de la malaltia per programar el joc i els minijocs.

En el cas de la estètica, tot i que no és un punt en el que hi haguem de dedicar una gran quantitat de temps, és important ja que hem d'usar els pictogrames perquè els pacients entenguin que és cada cosa. S'haurà de fer un joc molt pla en l'apartat estètic per no confondre al nen i fer que l'escenari es pugui confondre el mínim possible. Tot i això no es un aspecte que s'hagi de dedicar molt d'esforç, com s'exposa a l'apartat 5.1.3 GRÀFICS.

Finalment la Narrativa en aquest treball no té pes ja que simplement s'han de realitzar tasques en un ordre determinat pel terapeuta o el tutor per a que el nen, a base de repeticions, les dugui a terme. Tampoc pot haver-hi cap component Narratiu ja que la repetició és un fet molt important que ajudarà al nen i per tant no pot haver-hi cap història al darrera.

## 2. ESTUDI DE VIABILITAT

En aquest capítol valorarem si el projecte és viable o no observant diferents aspectes: els recursos tècnics, els recursos humans, els requisits tecnològics, el cost econòmic i un estudi de mercat.

### 2.1 RECURSOS TÈCNICS

Aquest treball s'ha dut a terme amb dos ordinadors, ja que durant el curs he estat vivint a Girona i a Calonge. Els dos ordinadors tenen un sistema operatiu Windows 10 de 64 bits. Els components dels ordinadors són els següents:

	ORDINADOR DE SOBRETAULA	ORDINADOR PORTÀTIL
Processador	AMD Ryzen 7 5800X	Intel® Core™ i7-7700HQ
Tarjeta gràfica	Nvidia Geforce RTX 3070	Nvidia Geforce GTX 1050
Memòria	32GB RAM	16GB RAM

*Taula 2 Quadre d'especificacions dels PC's*

### 2.2 RECURSOS HUMANS

En el desenvolupament d'un videojoc és requereixen diferents perfils per al desenvolupament.

- **Cap del projecte:** Aquest serà el responsable de coordinar el projecte, dictar les tasques a realitzar i si les que s'han realitzat estan correctes o no.
- **Dissenyador principal:** Aquest, serà l'encarregat de plantejar com seran els minijocs del projecte, quines mecàniques seran les adequades per cada un d'aquests i de quina manera. Serà el que parlarà amb la terapeuta i estudiarà quin comportament pot tenir el públic target. A més a més redactarà la memòria del projecte.
- **Programador i tester:** Serà el responsable de desenvolupar l'algorisme per a poder realitzar les funcions que vulguem que el joc tingui. A més a més, aquest serà el responsable de comprovar que el joc funciona correctament.
- **Artista principal i guionista:** Aquest, serà l'encarregat de decidir l'art que tindrà el videojoc i fer el que es necessiti.

Aquest projecte ha estat desenvolupat per una sola persona que ha hagut de fer de tots els perfils. No obstant, les tasques del de cap del projecte s'ha dut a terme conjuntament amb el tutor, i les de dissenyador amb la terapeuta.

## 2.3 REQUISITS TECNOLÒGICS

Per desenvolupar aquest projecte tant sols ha calgut els dos ordinadors descrits a l'apartat de 'Recursos tècnics'. S'ha fet servir el motor de desenvolupament Unity, en la versió 2020.3.25f1 (64-bit); dins d'aquest, l'entorn de programació Visual Studio amb el llenguatge C#. S'ha usat el Photoshop per editar imatges i crear algunes altres.

## 2.4 COST ECONÒMIC

En aquest apartat mostrarem el cost econòmic de tots els recursos esmentats anteriorment.

En el cas dels recursos tècnics, com que hem treballat amb els nostres ordinadors personals ens estalviarem el cost de comprar-lo. No obstant l'haurèm de tenir en compte per l'amortització dels recursos utilitzats.

$$\text{Amortització} = \frac{\text{mesosfeina} * \text{preuordinadors}}{36} = \frac{10 * (1963.58 + 899)}{36} = 795,16\text{€}$$

En el cas dels programes utilitzats són gratuïts excepte el Photoshop que s'ha usat la llicència d'un familiar.

En el cas dels recursos humans hem esmentat que hi ha diferents perfils de treballadors que són necessaris per tal de dur a terme el projecte. Com hem dit anteriorment, excepte el perfil de cap del projecte, els perfils han estat desenvolupats per la mateixa persona. No obstant, si haguéssim de calcular el cost per cada perfil de forma independent, tindríem els següents sous:

- Cap del projecte: 25€/h
- Sou del dissenyador: 20€/h
- Sou del programador: 15€/h
- Sou de l'artista: 10€/h

Hem considerat que el programador ja coneixerà les eines necessàries per a produir el joc, ja que considerem que no contractaríem a una persona sense aquests coneixements bàsic. Tenint en compte això, les feines a realitzar serien les següents:

- Estudi de la malaltia.
- Planificació del projecte
- Sessions amb la terapeuta per guiar el nostre projecte.
- Disseny dels Scripts per implementar les mecàniques del joc.
- Disseny dels minijocs.
- Disseny dels escenaris i les interfícies.
- Documentació.
- Cerca dels pictogrames necessaris.
- Adaptació dels pictogrames
- Cerca de sons necessaris pel joc.

Per tant, el cost en funció de les tasques seria el següent:

Tasques	Perfil	Hores	Cost/Hora	Cost Total
Estudi de la malaltia	Dissenyador	20	20	400
Planificació del projecte	Cap projecte	120	25	3000
Reunions amb terapeuta	Dissenyador	25	20	500
Implementació scripts	Programador	120	15	1800
Disseny minijocs	Dissenyador	100	20	2000
Implementar interfícies	Programador	20	15	300
Disseny escenaris i interfícies	Artista	20	10	200
Documentació	Dissenyador	120	20	2400
Cerca pictogrames	Dissenyador	5	20	100
Adaptació pictogrames	Artista	15	10	150
Cerca sons	Dissenyador	2	20	40
<b>Amortització</b>				795,16
<b>TOTAL</b>				11.685,16 €

*Taula 3 Cost econòmic*

## 2.5 ESTUDI DE MERCAT

Abans d'invertir en qualsevol projecte hem d'estudiar la possible cabuda que tindrà el producte en el mercat un cop finalitzat. En aquest apartat descobrirem quin és el target del nostre videojoc. A més a més estudiarem els productes similars que existeixen en el mercat per a descobrir si el nostre producte tindria gaire competència en el mercat i així poder tindre una estimació de com rebria el públic el nostre producte en comparació amb els actuals.

### 2.5.1 CERQUES REALITZADES

Identificarem com a videojocs similars al nostre, els que tenen com objectiu alguna de les següents descripcions:

- Joc del gènere del minijocs.
- Joc sobre alguna malaltia mental.
- Joc sobre TEA.
- Aplicació sobre TEA.

Tenint en compte aquestes descripcions hem utilitzat les següents paraules clau per fer cerques:

- Minigames.
- Mental Health Game
- TEA Game
- TEA Aplicacion

S'han utilitzat aquestes paraules claus soles i convinant-les entre elles. El cercador que hem utilitzat ha estat el Google.

## 2.5.2 RESULTATS OBTINGUTS

Els resultat trobats s'han analitzat i s'han classificat en els següents videojocs i aplicacions:

- **Child of light:** un RPG 2D on es tracta la depressió de la protagonista al perdre un familiar proper.
  - Cercador: Google
  - Paraules Clau: Videogame mental Health
  - Resultats 95.500.000



*Figura 2 Banner Child of Light*

- **Dicta Picto:** una aplicació que permet passar la informació de veu a imatges en format de pictograma de manera automàtica.
  - Cercador: Google
  - Paraules Clau: Aplicaciones TEA
  - Resultats: 23.000.000

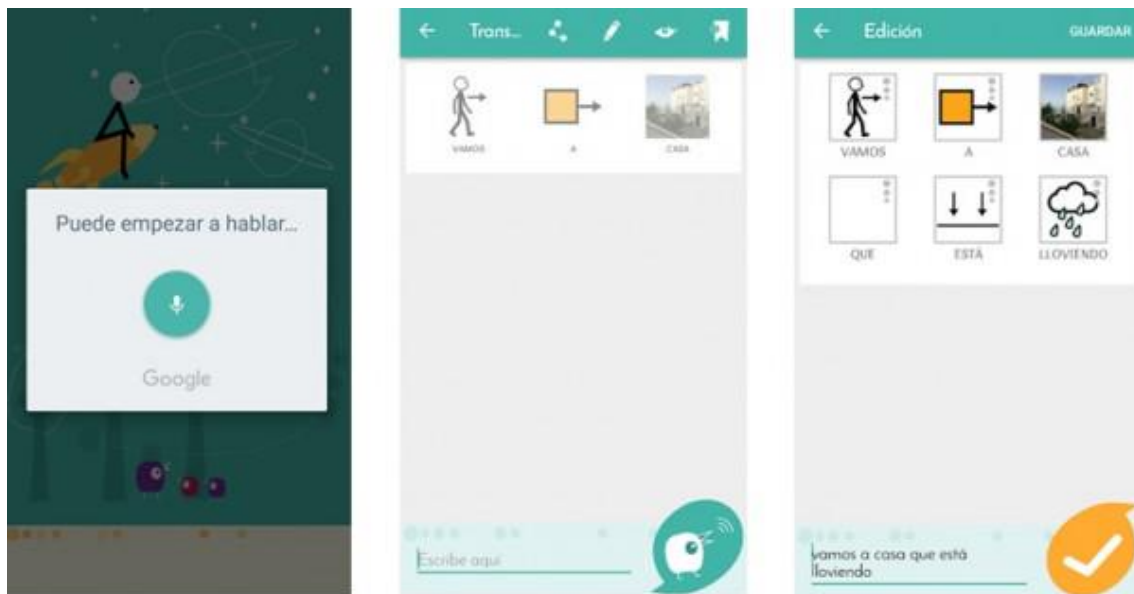


Figura 3 Captures DictaPicto

- **Día a Día:** una aplicació que funciona com a un simple calendari en el qual podem guardar i revisar les activitats diàries de forma visual i estructurades
  - Cercador: Google
  - Paraules Clau: Aplicaciones TEA
  - Resultats: 23.000.000



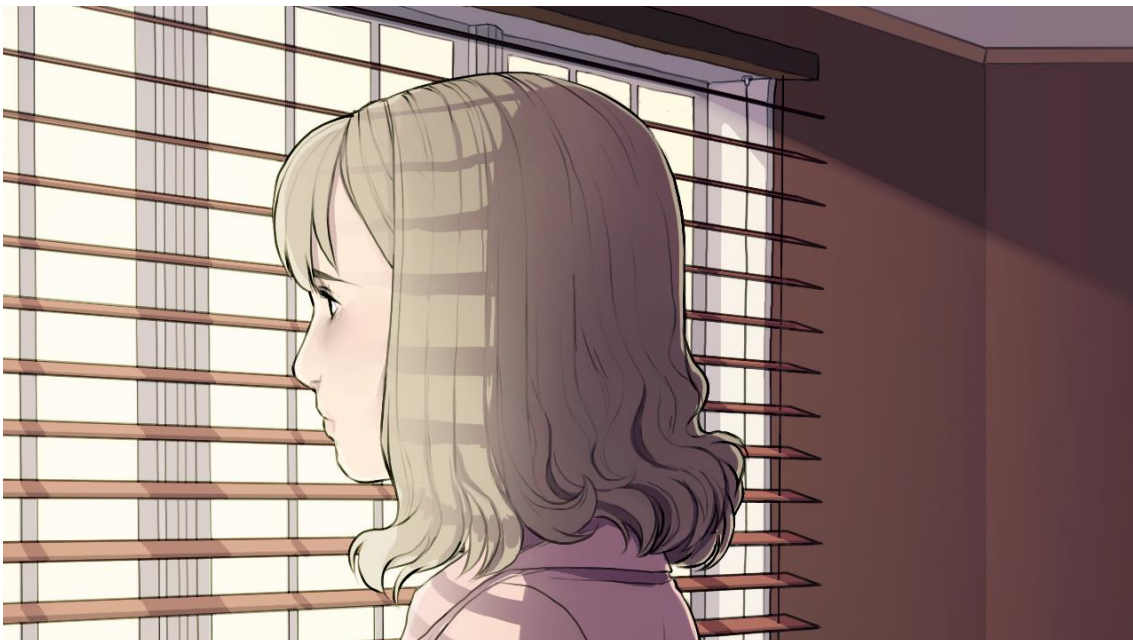
Figura 4 Captures Día a Día

- **Treasure Rangers:** un videojoc d'aventures, exploració i puzzles en el que un grup d'amics, en el que un es autista, han de descobrir un misteri que envolta la seva ciutat.
  - Cercador: Google
  - Paraules Clau: Videojuegos sobre TEA
  - Resultats: 10.400.000



*Figura 5 Captura Treasure Rangers*

- **Actual Sunlight:** un videojoc narratiu en la que el jugador haurà de veure com el seu personatge tracta la seva soledat, els seus problemes mentals i el suïcidi.
  - Cercador: Google
  - Paraules Clau: Videogame mental Health
  - Resultats 95.500.000



*Figura 6 Captura Actual Sunlight*



### 2.5.3 MATRIU DE COMPETITIVITAT

A partir dels videojocs i les aplicacions trobades, i tenint en compte els propòsits de les aplicacions, hem definit els següents trets característics:

- Ha de ser un videojoc.
- Ha de ser un joc educatiu o terapèutic (encara que ha de divertir als nens).
- Ha de ser un joc amb mecàniques senzilles i minijocs.
- Ha d'estar pensat per jugar gent amb TEA.
- El públic objectiu és gent amb patologia tractada.
- Ha de ser una eina que puguin usar els terapeutes.

Tenint en compte els propòsits esmentats anteriorment hem creat la taula de competitivitat, que podem veure a la Taula.

Aplicació	Videojoc?	Tipus	TEA	Minijocs?	Públic objectiu	Eina terapeutes
Child of light	Si	Lúdic	No	No	No Patologia	No
Dicta Picto	No	Teràpia	Si	No	Patologia	Si
Día a Día	No	-	Si	No	Patologia	Si
Treasure Rangers	Si	Lúdic	Si	No	No Patologia	No
Actual Sunlight	Si	Lúdic	No	No	No Patologia	No
Joc Proposat	Si	Teràpia	Si	Si	Patologia	Si

*Figura 7 Taula de Competitivitat*

### 2.5.4 CONCLUSIONS ESTUDI DE MERCAT

Un cop realitzada l'estudi, hem pogut veure que tot i que hi ha varies aplicacions que es poden fer servir com a teràpia per pacients amb TEA, cap d'elles és un videojoc i s'usen per tasques molt concretes. A part d'això, sí que existeixen videojocs que tractin sobre el TEA o altres malalties mentals, no obstant aquests fiquen al jugador en la pell d'un pacient o d'algun personatge amb aquest problema.

Això dista de la nostra idea, ja que el que pretén el nostre videojoc és el de poder interioritzar als pacients la rutina que han de seguir, o si més no, entrenar-la, amb petits minijocs que estiguin adaptats a ells.

## 2.6 PÚBLIC OBJECTIU I PERFIL DE JUGADOR

Per a dur a terme un projecte, en aquest cas un videojoc, hem de mirar qui serà el consumidor, o si més no, quina persona serà la interessada en comprar-lo. En el cas del nostre projecte, el jugador no serà qui el compri. El dissenyador del videojoc ha de tenir en compte que ell no serà el target pel producte. Per tant s'haurà d'assessorar o mirar d'acostar-se el més possible al punt de vista del consumidor. S'ha de tenir en compte que, potser el que per nosaltres és una cosa molt bàsica, pel jugador potser no ho és. Per tant hem de mirar d'evitar punts morts en el projecte en el que el jugador no sàpiga que ha de fer en aquell punt.

Com que el nostre videojoc no deixa de ser terapèutic o educatiu, està clar que el públic target serà el pacient i el seu terapeuta. Per tant haurem de fer que el terapeuta pugui configurar el joc segons vulgui i el pacient que tingui. A més a més, haurem d'adaptar les mecàniques perquè les puguin realitzar el màxim nombre possible de pacients i seguint sempre les indicacions que ens facin els terapeutes segons el que necessitin.

Per tant els usuaris seran:

- **Terapeuta:** Serà l'encarregat de configurar el joc i supervisar al jugador que faci la tasca de la manera que correspon, i ajudar-lo en cas que no entengui o no pugui realitzar alguna de les tasques assignades.
  
- **Pacient:** Serà el jugador. L'objectiu serà aprendre a base de repeticions el que ha de fer al seu dia a dia.

Com podem veure, una peculiaritat d'aquest projecte és que el jugador final del joc no serà el comprador del mateix. Per tant, tot i que hem d'adaptar el joc als nostres jugadors, haurem d'aplicar les característiques que creguin convenientes els seus terapeutes.

## 2.7 CONCLUSIÓ DE LA VIABILITAT DEL PROJECTE

En aquest projecte s'usen programes de pagament. No obstant, al tindre llicències d'estudiant o gratuïtes, excepte en el cas del Photoshop que podrem usar la llicència d'un familiar, el cost del software serà inexistent.

En el cas del Hardware sí que hi ha un cost, el que hem vist en apartats anteriors. No obstant, en aquest cas concret, al no haver de fer una inversió ja que aquests ordinadors eren personals i ja es tenien per altres projectes, no haurem de fer cap inversió.

Pel que fa al sou dels treballadors, com hem comentat anteriorment, aquest projecte està realitzat per un sol estudiant, i per tant els sous a pagar en aquest cas també seran 0.

No obstant, si féssim aquest projecte amb professionals, no creiem que sigui un pressupost molt elevat, ja que a causa de les característiques necessàries del propi joc, és senzill de programar i amb una complexitat relativament baixa.

L'estudi de mercat ens ha mostrat que el nostre projecte és viable ja que no n'hi ha cap de semblant. Tot i que s'ha vist algun joc que tractava les rutines dels nens amb TEA, aquest ho feia massa complicat pel seu enteniment, i aquest projecte no ho farà.

Creiem que aquest projecte, en el cas de haver estat dut a terme dintre d'algun estudi professional, podria haver optat a alguna subvenció, ja que es tracta d'un joc terapèutic.

Com a conclusió, ja que és un projecte d'estudiant i es disposa de tots els aspectes necessaris per al seu desenvolupament, el treball serà viable.

### 3. PLANIFICACIÓ

En aquest apartat veurem quina estratègia hem seguit per a poder complir els objectius plantejats. Descriurem el pla de treball, les tasques planificades, el temps estimat i els resultats esperats de cada tasca.

Normalment en els projectes de disseny d'un videojoc és realitzen varies tasques de forma paral·lela, aquestes tasques es reparteixen segons de quin tipus siguin i es consensuen dates de terminis per a tenir cadascuna d'aquestes tasques resoltes. No obstant, com hem anat comentant al llarg d'aquesta memòria, aquest treball el realitzarà una sola persona i, per tant les tasques les realitzarà un sol individu. En aquest cas les reunions es faran amb el tutor del treball i amb la terapeuta que actuarà també com a tutora.

En qualsevol projecte hi ha 3 etapes principals. Aquestes etapes són:

- Etapa de disseny o planificació del projecte.
- Etapa d'elaboració del projecte.
- Etapa de retocs, testeig i proves.

No obstant, en el món dels videojocs, al realitzar varies tasques alhora, el seu desenvolupament es du a terme iterativament. Per tant, quan s'ha elaborat una etapa pot ser que s'hagi de tornar a dissenyar ja que aquesta no concorda amb qualsevol altre part, o s'ha vist que aquesta es podria fer d'alguna altra manera més adient.

En el meu cas concret, cada vegada que realitzava alguna part del meu projecte ho comentava al meu tutor i miràvem si estava correcte o no. Un cop el meu tutor ho donava com a correcte, li ensenyava a la terapeuta i ella m'explicava com podia millorar-ho o que havia de canviar per a complir les seves necessitats. Per tant, podem dir que en tot el procés del projecte hem estat en un bucle constant d'assaig i error fins a aconseguir els resultats necessaris.

He pogut veure que una bona planificació inicial combinada amb una bona comunicació entre els tutors i l'autor ha estat la clau per a obtenir un prototip funcional per la tasca que volíem aconseguir.

### 3.1 PLA DE TREBALL

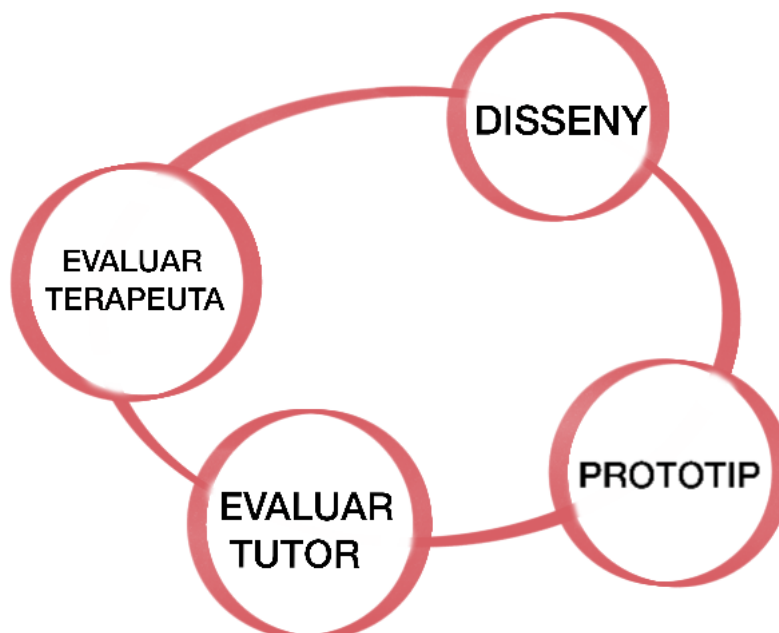
En aquest punt definirem les tasques que haurem de realitzar i la seva. Haurem de tenir en compte diferents factors, com poden ser els imprevistos que puguin sorgir, sobreestimar alguna tasca i al final realitzar-la en un temps menor, etc.

Per exemple, ens hem trobat diferents problemes a l'hora de provar alguna mecànica amb algun usuari o per motius de feina hem hagut de posposar reunions.

Com hem comentat anteriorment, el procés d'aquest projecte és iteratiu. No obstant, en aquest cas hi ha una peculiaritat ja que s'ha hagut de validar amb els dos tutors.

Un exemple pot ser el primer cop que vam veure que havíem d'adaptar el nostre joc. Vam decidir amb el tutor preparar el moviment entre els diferents minijocs del nostre personatge, pel que vam acordar dissenyar una casa en la que es veiessin diferents portes amb les sales de la casa. Vam començar a implementar-ho fent un "Click and Go" i animant el personatge. Un cop vam veure amb el tutor que era correcte li vam ensenyar a la terapeuta, ella va considerar que era massa complicat pels nens amb TEA i que ella simplement volia un calendari amb un pictograma que indiqués la tasca a fer (ja que és amb el format que treballen ells al centre) i al clicar la imatge ens redirigís directament al minijoc. Allà vam acordar amb la terapeuta en fer un joc molt estàtic en el que no hi haguessin animacions que poguessin distreure al nen i que tant sols es mogués l'element que utilitza el nen cada vegada.

La mostra a la figura 8:



*Figura 8 Figura pla treball*

### 3.1.1 ABAST DEL PROJECTE

Per a planificar correctament un projecte primer s'ha de tenir clar quin és el seu abast. S'ha de pensar i definir totes les funcionalitats necessàries que haurà de tindre el videojoc. D'aquesta manera podrem organitzar d'una manera més correcte el projecte i saber el que necessitarem per a poder desenvolupar-lo correctament. Es cert que mentre es realitza el projecte podem anar trobant característiques extremes o algunes que creiem que eren necessàries ho deixin de ser.

Considerem que el nostre videojoc necessita tenir els següents punts:

- Videojoc format per un conjunt de minijocs
- L'objectiu d'aquest videojoc és el de ser utilitzat com una eina terapèutica.
- El videojoc ha de servir com un entrenament de les tasques rutinàries.
- Les tasques a realitzar les determinarà el terapeuta per cada nen.
- Les mecàniques dels minijocs han de ser molt senzilles i adaptables a una tauleta o dispositiu mòbil.
- Tot i que el joc tindrà una puntuació, cada minijoc serà sempre un èxit per a aportar feedback positiu al pacient, ja que si necessita ajuda li donarà el terapeuta.
- Les interfícies han de ser senzilles.
- Ha de ser impossible pel pacient canviar la rutina establerta pel terapeuta.
- S'ha de representar la rutina d'una forma clara pel pacient.
- S'han de representar els minijocs de les següents tasques:
  - Vestir-se.
  - Parar la taula.
  - Rentar-se les dents.
  - Dutxar-se.
  - Ordenar habitació.
  - Saber on van diferents objectes.
  - Fer-se la motxilla.
- S'ha d'implementar un sistema sonor que expliqui al pacient el que ha de fer.

## 3.1.2 TASQUES PLANIFICADES

Un cop tenim descrits els punts que haurà de tenir el projecte, ja podem definir més detalladament les tasques que haurem de realitzar. Aquestes seran les següents:

Número	Tasca	Temps estimat
1	Presentació proposta a terapeutes especialitzats	2 hores
2	Reunió cada dos setmanes terapeuta i el tutor. (Reunions separades)	40 hores
3	Implementació minijocs	40 hores
4	Disseny minijocs	20 hores
5	Disseny habitacions	8 hores
6	Buscar i adaptar pictogrames	5 hores
7	Implementació interfícies	10 hores
8	Gravació sons	1 hora
9	Proves amb pacients	0 hores
10	Reunió terapeuta sobre les proves	2 hores
11	Implementació canvis	8 hores
12	Pulir bugs i ultims canvis	15 hores
<b>TOTAL</b>		<b>141 hores</b>

*Figura 9 Taula de tasques*

### 3.1.3 CRONOGRAMA

S’ha fet un cronograma per a donar una visió global del projecte. Podem veure que per a fer la implementació de cada minijoc hem anat tenint reunions cada dues setmanes amb el tutor i la terapeuta. Podem veure el cronograma global del projecte (Figura 10) i el cronograma setmanal del desenvolupament del videojoc (Figura 11).

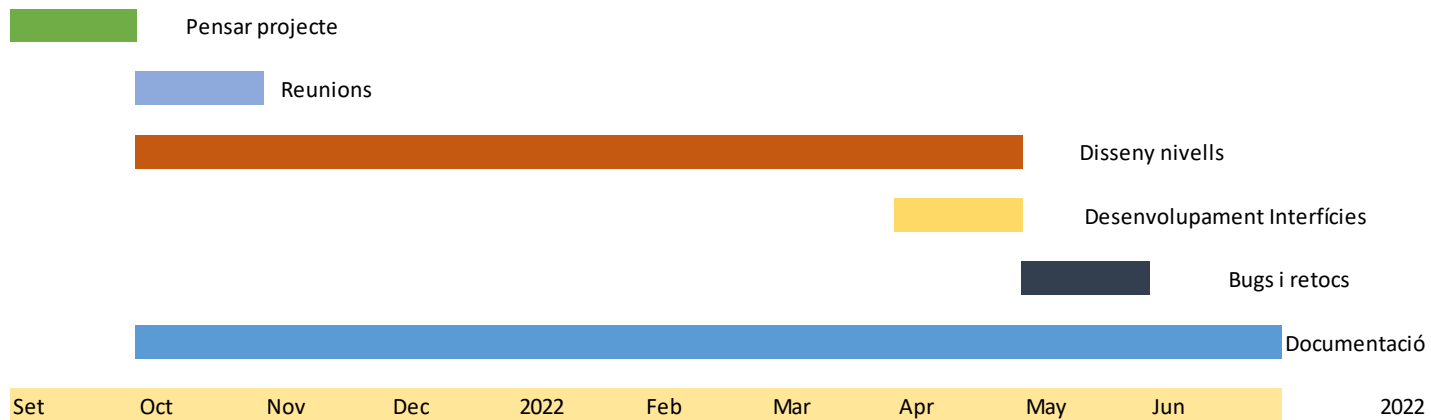


Figura 10 Cronograma global

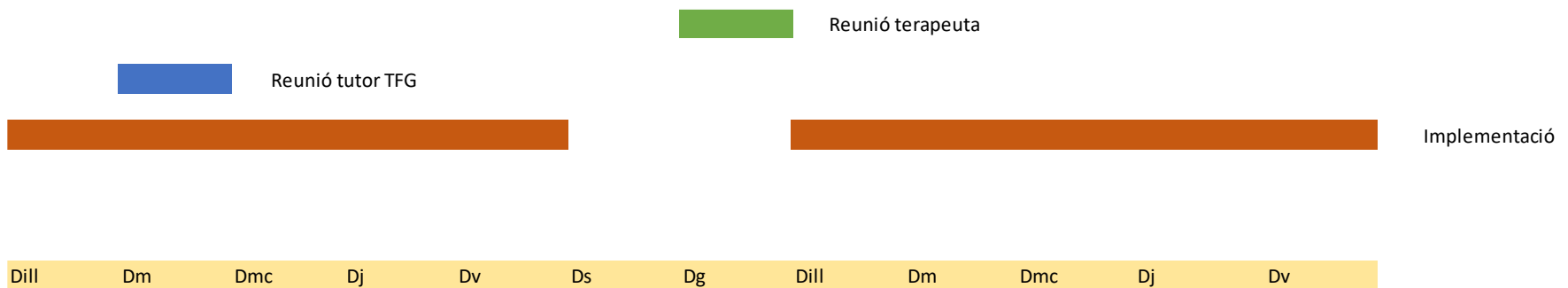


Figura 11 Cronograma setmanal



## 4. MARC DE TREBALL I CONCEPTES PREVIS

Aquest apartat ens servirà per a posar en context al lector i explicar els que és necessari per comprendre el motiu pel qual s'han pres algunes decisions. Sabem que aquest apartat és molt important en tots els projectes, no obstant creiem que en aquest cas és més necessari per la possible desconexió de la malaltia i de com interactuen els terapeutes amb els pacients.

Per tant, introduïrem quina mena de joc s'ha realitzat. Això ho farem explicant quines són les referències que hem tingut per a realitzar el projecte. Com hem comentat anteriorment, ens hem inspirat en diferents aplicacions més que en videojocs, ja que no n'hem trobat cap comercialitzat amb les característiques requerides.

A part d'això explicarem amb quin motor hem desenvolupat el projecte i les raons per les quals creiem que ha estat la decisió més correcta.

### 4.1 TRASTORN DE L'ESPECTRE AUTISTA (TEA)

Creiem que el més important que cal saber per entendre el motiu del desenvolupament d'aquest videojoc i poder entendre totes les decisions que s'han pres és el fet d'entendre per quin tipus de persona s'ha desenvolupat aquest videojoc. Com hem comentat anteriorment aquest projecte està pensat per ser una eina terapèutica i educativa per persones amb una patologia concreta: El trastorn de l'espectre autista (TEA). Es caracteritza per greus dèficits del desenvolupament, permanents i profunds, que afecten la socialització, la comunicació, la imaginació i la conducta. Les causes de l'autisme encara no es coneixen, encara que es barregen diverses hipòtesis com les causes genètiques, la presència d'alteracions en la connectivitat cerebral o en el sistema immunitari. Una detecció primerenca de l'autisme en nens proporciona la possibilitat d'iniciar un tractament el més aviat possible i, d'aquesta manera, reduir els símptomes.

Els TEAs es caracteritzen per les dificultats en la interacció social i en la comunicació i, també, per la tendència a les conductes repetitives. No obstant això, els símptomes i la seva gravetat poden variar en aquestes tres àrees principals. Així, en algunes persones els símptomes poden produir menor impediment i ser més subtils, sobretot, en aquelles persones situades a l'extrem de l'espectre caracteritzat per l'alt funcionament. Malgrat això, en altres, els símptomes poden ser més greus i no arribar a desenvolupar el llenguatge o mostrar conductes repetitives que interfereixen en la vida quotidiana.

En el gràfic de la figura 2 es mostra com els símptomes bàsics de l'autisme s'acostumen a acompanyar d'altres malalties i trastorns que també poden variar en gravetat.

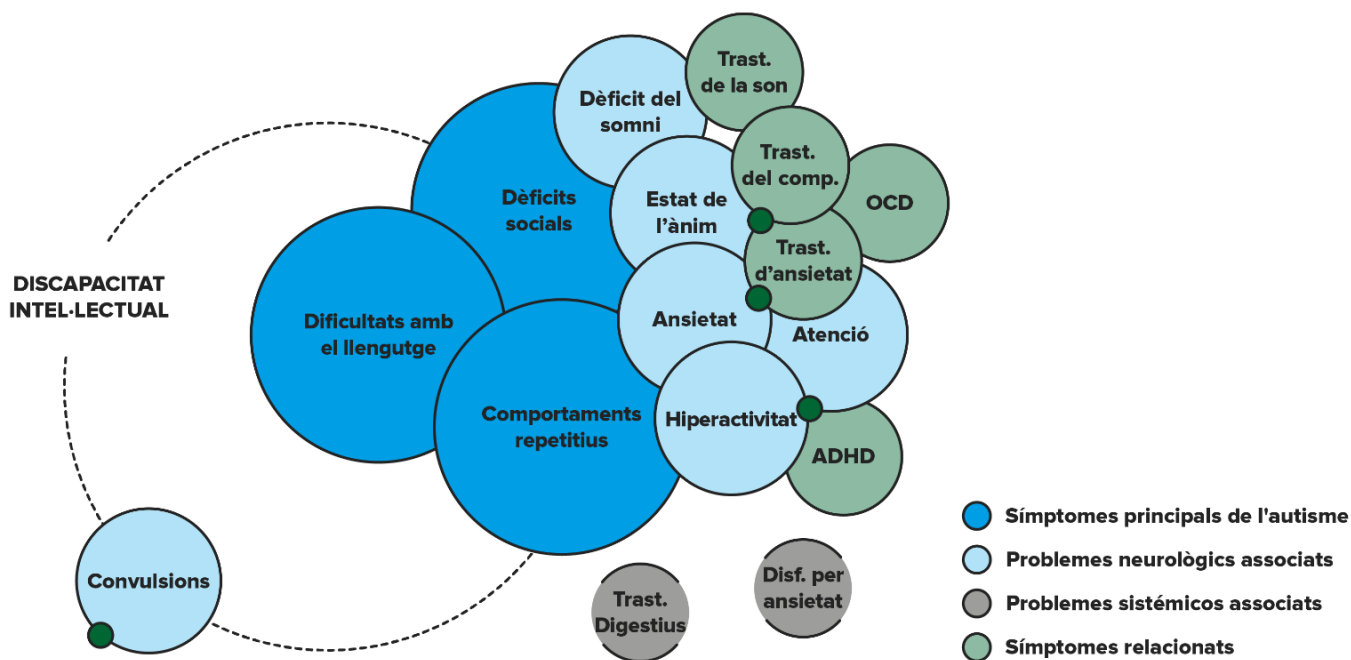


Figura 12 Diagrama Simptomes TEA (Clínic Barcelona)

Tota la informació explicada fins aquí i aquest diagrama han estat obtinguts de la web oficial de l'hospital Clínic de Barcelona\*.

A part dels símptomes esmentats en el diagrama anterior, trobem les següents dificultats:

- Dificultats en la interacció social.
  - Dificultats en les interaccions habituals.
  - Dificultats per interpretar les expressions facials i els gestos habituals que acompanyen la interacció
  - Dificultats per regular les emocions.
- Dificultats en la comunicació.
  - Retard en el desenvolupament del llenguatge i l'ús de gestos per comunicar-se.
  - Característiques peculiars del llenguatge.
  - Dificultats per comprendre el llenguatge corporal, el to de veu, les expressions o frases fetes.
- Conductes repetitives:
  - Conductes repetitives inusuals i tendència a restringir els interessos i les activitats.
  - Col·locar objectes d'una manera determinada, repetir sons, paraules o frases.
  - Interessos diferents i molt exagerats.
  - Hipersensibilitat.
- Altres símptomes:
  - Síndromes genètiques.
  - Convulsions.
  - Trastorn del son.
  - Menjar coses que no són aliments.
- Trastorns psiquiàtrics:
  - Discapacitat intel·lectual.
  - Trastorn per Dèficit d'Atenció i Hiperactivitat (TDAH).
  - Trastorns d'ansietat.
  - Trastorns afectius.

## 4.2 EXPERIÈNCIA AMB TERAPEUTES I PACIENTS

Com s'ha comentat en anteriors apartats, en aquest projecte hi ha hagut una constant interacció i iteració amb els terapeutes que faran servir la nostra eina en el futur.

La terapeuta que ens ha fet de tutora del TFG, la senyora Anna Valls, ha mantingut una via directa de comunicació i ha fet una reunió setmanal amb l'autor per a veure el progrés del projecte i les coses que ells canviarien.

Vam començar implementant les mecàniques dels minijocs i, per cada reunió, havíem de fer el minijoc que havíem acordat. Si li semblava bé, passàvem al següent i si no canviàvem les coses que ella considerava necessàries i fèiem un altre minijoc. Tots els minijocs han estat desenvolupats amb la seva autorització i acotant el més possible les seves necessitats, tot i que és cert que ens ha donat molta llibertat per a implementar el que ella considerava oportú.

Al fer les reunions amb ella, ens vam adonar que aquesta patologia era molt complexa i que era molt difícil fer alguna cosa que puguin jugar tots els nens amb TEA. Vam arribar a la conclusió que desenvoluparíem un joc en el qual els nens amb més problemes puguin jugar amb l'assistència de la terapeuta o dels pares, i que els que no tinguin problemes tant severos puguin jugar ells sols encara que al inici pot ser que necessitin una petita empenta.

Al inici vam veure que les idees de minijocs que volíem desenvolupar els havíem de simplificar molt perquè el pacient pogués complir les tasques de forma satisfactòria. Aquest fet de simplificar el vam veure mes clar al debatre com s'havia de passar d'un minijoc a l'altre, ja que amb el tutor vam creure que el més idoni era fer una animació en la que el personatge es mogués per la casa i tries la porta que ell volgués. No obstant, al moment en el que la terapeuta va veure el que estàvem fent, ens fa comentar que ella prefereix un videojoc estàtic en el que tant sols es moguin les coses amb les que poden interactuar els nens.

Com a conclusió final, vam veure que era molt complex fer un joc per a tots els nens amb TEA ja que l'espectre es tant gros que sempre es quedarà algú fora. Per tant, es va decidir enfocar-lo a pacients que tinguin la suficient capacitat com per a resoldre puzles d'una complexitat bàsica.

## 4.3 REFERÈNCIES

Les idees d'un projecte poden sorgir de moltes maneres. Un exemple podria ser que es tingui una idea a afegir a un estil de videojoc que estigui molt de moda que el pugui fer millor. Un altre podria ser el de tenir una idea completament nova que es cregui que pugui tenir potencial. En el nostre cas, podem dir que la idea és completament nova, no obstant, és ben cert que ens hem inspirat en una sèrie de coses que ja existeixen en altres aplicacions. A més a més, els minijocs implementats que ja existeixen, o les seves mecàniques, no s'utilitzen en aquest tipus d'aplicacions. Creiem que el tret característic del nostre videojoc és que sigui una eina de treball pels terapeutes.

A continuació exposarem unes aplicacions que han servit d'inspiració pel videojoc.

### 4.3.1 DICTAPICTO

DictaPicto és una app que permet passar un missatge de veu o escrit a imatges de manera immediata. Pensada per ajudar les persones amb autisme o aquelles que usen sistemes pictogràfics per millorar la seva comunicació, el seu objectiu és millorar l'accés a la informació i facilitar la comprensió de l'entorn amb independència que les persones que envolten la persona amb TEA coneguin aquests sistemes de comunicació augmentatius i alternatius.

La traducció de veu a pictogrames es produeix automàticament a partir d'una frase parlada de l'usuari; convertint-la inicialment en text i, per això, en els pictogrames/imatges que representen les paraules.

Desenvolupada per BJ-Adaptaciones i Doble Equipo, Dictapicto permet anticipar i seqüenciar activitats de la vida diària de forma flexible i en diferents escenaris; facilitant així la participació i la interacció de l'usuari amb el seu entorn.

A més, es presenta com una eina senzilla i pràctica amb què preparar de forma àgil materials bàsics per, per exemple, la preparació de normes o històries socials senzilles.

#### **Funcionalitats:**

- Traducció de veu a pictogrames: La traducció es produeix automàticament a partir d'una frase parlada de l'usuari; convertint-la inicialment en text i, d'aquí, en els pictogrames/imatges que representen les paraules
- Gestió de traduccions: Quan es presenta la frase traduïda, l'usuari té diverses opcions; Seleccionar la imatge que representa una paraula en concret, etiquetar la frase per facilitar-ne l'ús posterior i fins i tot compartir-la amb altres dispositius.
- Fitxer de traduccions: L'eina consta d'un fitxer de traduccions amb exemples per defecte. Aquest fitxer és editable i l'usuari podrà esborrar o afegir traduccions noves. Aquestes traduccions es poden etiquetar.

- Gestió de vocabulari: L'aplicació inclou un corpus de vocabulari basat en els pictogrames d'ARASAAC. Aquest pot ampliar-se i/o actualitzar-se, editant o eliminant els termes que s'hi inclouen.



*Figura 12 Banner DictaPicto*

D'aquesta aplicació ens vam inspirar en el fet de comunicar-nos amb el pacient mitjançant pictogrames. Es cert que en el joc hi han interfícies i explicacions que és fa servir text normal o botons. No obstant, en els jocs els elements que han de saber que són estan representats amb pictogrames per a facilitar la jugabilitat del pacient.

#### 4.3.2 DÍA A DÍA

Dia a dia és un diari visual pensat especialment per a persones amb autisme o amb dificultats de comunicació. Desenvolupada per la Fundació Orange i BJ Adaptaciones, l'aplicació permet treballar de manera fàcil i intuïtiva, donant molta rellevància als elements visuals i oferint diferents opcions de personalització.

L'aplicació consta d'un senzill calendari on és possible:

- Guardar i revisar les activitats realitzades durant un dia, de manera visual i estructurada, afegint imatges, fotos, vídeos...
- Anticipar activitats o esdeveniments recurrents en el temps que ja s'han realitzat i que s'han anotat al diari.
- Fomentar la comunicació a través de les activitats realitzades.

A l'hora de fer servir l'aplicació, l'usuari pot incloure l'activitat en diferents moments del dia (matí, tarda, nit) i descriure-la mitjançant imatges, vídeos, sons, textos... A més, un cop creada l'activitat, es pot associar fàcilment a persones i llocs amb qui s'ha compartit.

El diari és personalitzable i es pot adequar a les necessitats de cadascú, per exemple, categoritzant les persones de l'entorn o els llocs més freqüents, ocultant vistes i elements que no es facin servir, etc. Veure figura 13.

El desenvolupament de l'aplicació ha comptat amb la col·laboració de l'associació Aprenem.



Figura 13 Captura Día a Día

D'aquesta aplicació ens hem inspirat en la creació de calendaris amb pictogrames. Aquest calendari el farem servir per fer que el jugador entri al minijoc de la tasca corresponent. La idea original era fer que el personatge és moguéssim d'habitació a habitació, no obstant la terapeuta ens va dir que havíem de buscar una altra manera i li va semblar interessant el fer un calendari com el del "Día a Día" ja que ells a la consulta treballen d'aquesta manera.

#### 4.3.3 CONCLUSIONES REFERÈNCIES

En aquest apartat hem pogut veure que tot i que hem extret coses de diferents aplicacions, el joc en sí no està inspirat en cap que estigui actualment al mercat. No obstant, com s'ha dit anteriorment, les mecàniques i alguns minijocs ja existeixen en altres videojocs. No ens hem inspirat en cap joc en concret si no que hem agafat mecàniques o minijocs que tots coneixem i els hem adaptat pels nens amb TEA. Les aplicacions esmentades van ser recomanades per la terapeuta ja que és com treballen i es volia una eina el més semblant possible al que fan amb el pacient.

## 4.4 MOTOR DE JOC

Un motor de videojocs és un terme que fa referència a una sèrie de llibreries de programació que permeten el disseny, creació i representació d'un videojoc. A l'hora de l'elecció del motor s'ha de tenir en compte la facilitat d'aprendre a fer servir el motor de videojocs i la facilitat per exportar el joc a diferents plataformes.

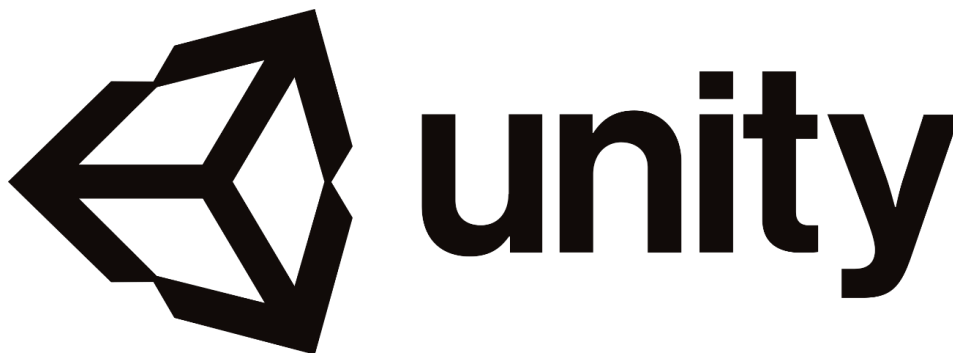
Algunes de les funcionalitats més importants són:

- **El motor de físiques:** és el que fa possible aplicar aproximacions físiques als videojocs per a què tinguin una sensació més realista a la interacció dels objectes amb l'entorn. En altres paraules, és l'encarregat de fer els càlculs necessaris perquè un objecte simuli tenir atributs físics com pes, volum, acceleració, gravetat...
- **El motor de so:** Els sons i la banda sonora d'un videojoc és també una part molt important. El motor de sons és l'encarregat de carregar pistes, modificar-ne la taxa de bits, treure-les de reproducció i sincronitzar-les, entre altres coses.
- **L'scripting:** Tots els motors de videojocs tenen un llenguatge de programació que permet implementar el funcionament dels personatges i objectes que formen part del videojoc.

Els motors de videojocs més coneguts poden ser: L'Unity 3D, l'Unreal Engine, el Frostbite Engine,...



#### 4.4.1 UNITY



*Figura 14 Logo Unity*

Aquest projecte s'ha decidit realitzar amb el motor de videojocs Unity. Aquest motor va ser desenvolupat per Unity Technologies. Unity té dues versions: Unity Professional (pro) i Unity Personal. La versió professional l'hauríem de pagar si guanyéssim \$200k anualment, al pagar ens donarien suport Premium i escalabilitat. Nosaltres farem servir la segona versió, ja que no tenim els guanys que obligarien a pagar llicència. A continuació exposarem els motius pels quals s'ha triat Unity envers algun altre motors i les parts bàsiques que necessitem saber.

Una de les raons principals per les quals s'ha decidit usar aquest motor és perquè aquest és el que més s'utilitza per a desenvolupar videojocs. Segons els mateixos creadors, més de la meitat dels videojocs del món han estat creats amb Unity. Un altre fet molt important és que aquesta tecnologia ja ens era coneguda, el que ens va permetre guanyar temps al no haver d'aprendre de 0 el funcionament d'aquesta tecnologia. A part de ser conegut, és amb el motor de joc que hem provat que més còmode ens hem sentit fins al moment.

A part de l'esmentat anteriorment, un altre factor va ser que, si es feia bé el projecte, resultaria molt simple poder compilar una versió de PC i una per Android. Això ha estat un fet molt important ja que, al lloc on es vol provar amb els pacients, tenen sales amb tauletes i altres amb PC i els era interessant poder provar-ho amb les dues plataformes.

A continuació exposem algunes característiques d' aquest motor:

- Permet el desenvolupament multiplataforma per a gairebé totes les opcions que hi ha en el mercat, tant de consoles com dispositius mòbils. A més, exportar el mateix videojoc a diferents plataformes és molt fàcil de realitzar i ho fa a molta velocitat, deixant gairebé tota la feina restant a l'adaptació de controls i interfícies.
- Permet desenvolupar tota mena de videojocs. Té diferents mòduls per treballar tant en 3D com en 2D. Integra exemples de molts tipus de gèneres i implementacions bàsiques per començar amb el desenvolupament.

- Té una versió gratuïta que és suficientment capaç per si mateixa. Es pot fer qualsevol videojoc sense cap restricció, però apareixen limitacions com que apareix el logo d'Unity en començar el joc (encara que es pot treure), o els serveis que ofereixen estan limitats.
- L'editor és molt intuïtiu i senzill d'utilitzar. Permet treballar amb molts tipus de fitxers i per tant es pot complementar el motor amb molts programes externs.
- Al ser un dels motors més usats té una gran quantitat de documentació i molts desenvolupadors que han preguntat i respost dubtes per internet i això fa que hi hagi una gran quantitat d'informació. Es pot veure molt clarament a Stack Overflow.
- El llenguatge que s'utilitza per a programar en aquest motor és el C# un llenguatge que, igual que el que hem comentat anteriorment, al ser molt usat hi ha una gran quantitat d'informació per internet i molta gent el sap fer servir.

A la figura 15 adjuntem una captura de pantalla del Unity. Aquí podem veure l'inspector que és la taula que es mostra a la part dreta de la imatge. L'inspector ens permet afegir i modificar els paràmetres d'un objecte seleccionat.

A la part esquerra tenim dues finestretes. La que està a dalt ens mostra la jerarquia dels elements que hi ha a l'escena. La de sota ens mostra els fitxers i arxius que hi ha dintre del projecte.

A la pantalla central podem veure l'escena i en cas de donar-li al play veuríem una previsualització i podríem jugar al que hem estat desenvolupant.

A part de el comentat existeixen moltes més finestres que es poden ficar a la pantalla com poden ser l'Animator, el ShaderGraph,...

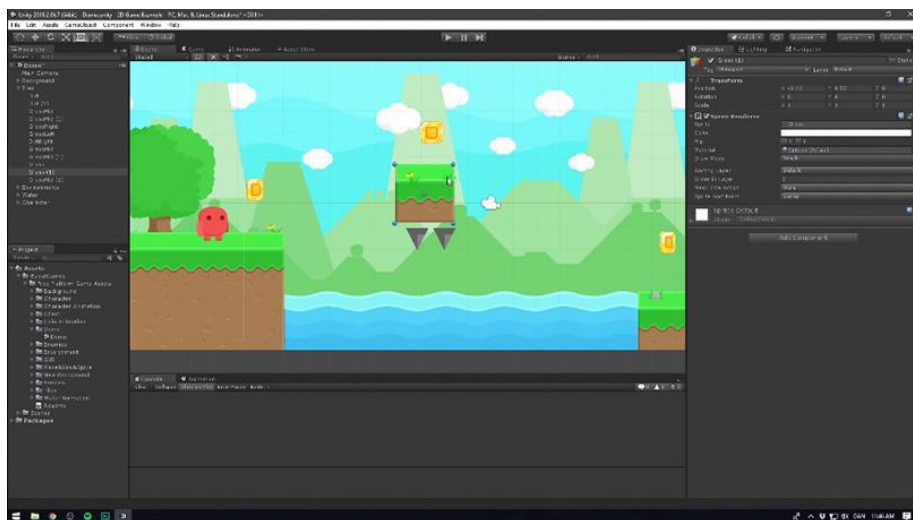


Figura 15 Interfícies Unity

## 4.5 PEGI



Figura 15 Logo PEGI amb els seus codis

El Sistema PEGI (Pan European Game Information) és el mecanisme d'autoregulació dissenyat per la indústria per dotar els seus productes d'informació orientativa sobre l'edat adequada per al consum. El Sistema PEGI està integrat per dos tipus d'icones descriptores, una relativa a l'edat recomanada i una altra al contingut específic susceptible d'anàlisi. El disseny dels logotips informatius es basa en els llums de seguretat viària, fent-ne més fàcil i visual la interpretació. Els segells van acompanyats de termes pictogràfics sobre el contingut del joc. D'aquesta manera, no hi ha errors d'interpretació sobre el que és apte per a cada consumidor, ja que aquest podrà comprendre per endavant el tipus de continguts que trobarem i realitzar una elecció lliure i informada.

En el cas del nostre projecte, per l'estil de videojoc i pels elements que conté, s'hauria d'assenyalar que és un **PEGI 3**. No hi hauria cap pictogràfic sobre contingut de joc ja que aquest no conté cap element que calgui informar. Per tan el nostre joc seria apte per tots els públics a partir de 3 anys.

## 5. DISSENY DEL VIDEOJOC

En aquest apartat haurem d'explicar tot allò que hem fet perquè el projecte compleixi els objectius que ens hem marcat. Detallarem les decisions que hem pres i quin ha estat el motiu per a prendre cada decisió.

### 5.1 ESTIL

En aquest subapartat explicarem l'estil que hem decidit aplicar en aquest videojoc. Creiem que l'estil del videojoc és un dels elements més importants, ja que es pot tenir un joc amb unes mecàniques espectaculars, que si no entra pels ulls o si l'estil emprat no és el més correcte, aquest videojoc perdrà molt de valor.

#### 5.1.1 GÈNERE

El projecte que hem proposat és del tipus dels minijocs 2D. Podríem considerar com si fos un puzzle 2D, ja que el jugador ha de complir tasques com ficar coses en un lloc determinat.

Les mecàniques dels puzzles 2D poden ser clicar en el lloc correcte, arrossegat i deixar un objecte en un lloc específic, gratar fins a eliminar o netejar un objecte, etc.

Un exemple molt clar d'aquest tipus de videojocs en els quals hi ha diferents minijocs de l'estil puzzle serien la saga de videojocs de "Imagina Ser" de la NintendoDS o el "Cooking mama" (Veure Figura 16). Aquests jocs, mitjançant el panell tàctil de la NintendoDS, permetien resoldre diferents puzzles o fer diferents accions per superar els nivells.

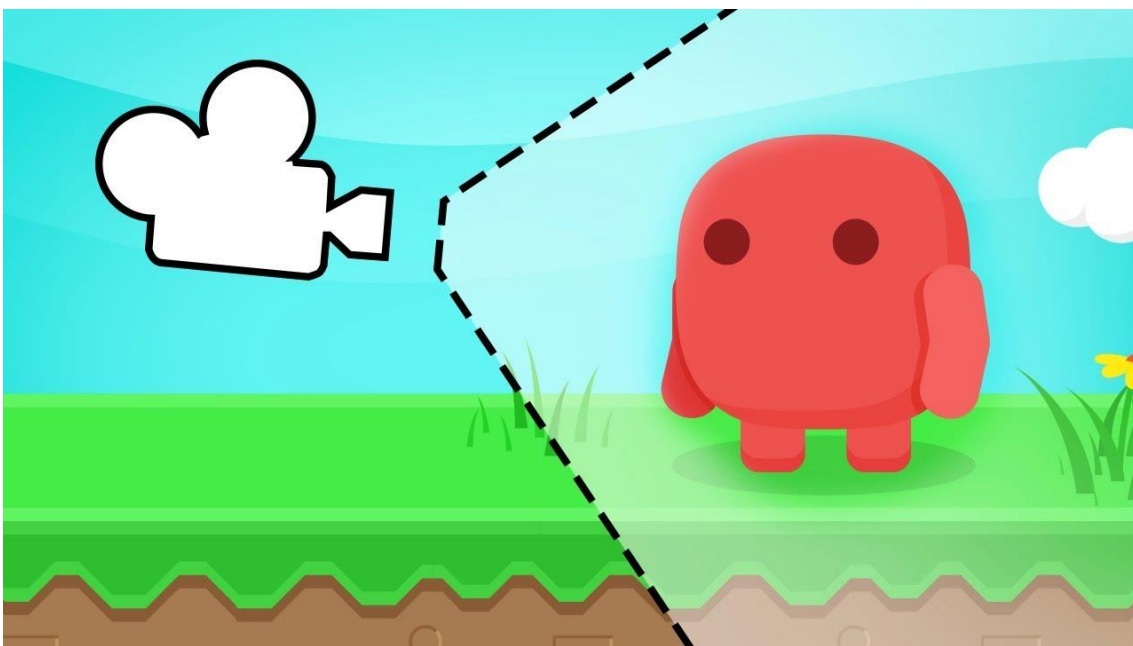


Figura 16 Captura Cooking Mama

### 5.1.2 CÀMERA

La càmera que s'utilitza en el videojoc és possiblement la més utilitzada al llarg de la història de la indústria: la perspectiva 2D i frontal. Aquest tipus de perspectiva ha estat la dominant durant molts d'anys, ja que abans era molt complicat tecnològicament suportar altres tipus de càmeres. Aquesta perspectiva s'utilitza molt en videojocs de plataformes, puzles i minijocs. No obstant, tots aquests gèneres s'estan passant al 3D en l'actualitat. Un exemple pot ser la saga de Nintendo Mario, pel fet que aquests van començar essent jocs de plataformes 2D i s'han anat convertint fins a arribar a ser jocs de plataforma 3D.

A la Figura 17 és mostra el tipus de càmera que usarem en aquest projecte.



*Figura 17 Figura que mostra el tipus de càmera.*

A l'inici d'aquest projecte volíem realitzar un joc 3D en primera persona en el que es veies les mans del nen indicant l'acció a fer. No obstant després de la reunió que vam fer amb la terapeuta vam decidir que no seria la millor opció, ja que ells no ho interpretarien adequadament i que la millor opció seria que no hi hagués cap personatge en concret i que el jugador tan sols interactués amb l'escenari. En el cas en què en algun minijoc hagi de sortir algun personatge aquest hauria de ser en tercera persona i que es vegi clarament que és un ésser humà perquè puguin entendre que és el personatge sobre el qual s'ha de realitzar certa activitat. En el nostre joc aquest cas seria els minijocs de vestir-se i de dutxar-se.

### 5.1.3 GRÀFICS

En aquest apartat explicarem quin tipus d'estil artístic tindrà el nostre videojoc. Cal aclarir que l'estil artístic del videojoc no ve determinat pel gènere d'aquest o el tipus de joc que serà. Sí que és cert que hi ha jocs que els queda més bé un estil artístic que un altre, però no hauria de ser un fet que el definís. En el nostre projecte ens veiem obligats a usar uns pictogrames molt concrets. Aquests pictogrames estan dissenyats amb uns dibuixos molt simples i amb colors plans. Per a no trencar amb l'estètica d'aquests, vam decidir seguir amb aquest estil i fer un estil "cartoon". Tot i que els pictogrames tenien un petit contorn negre, vam decidir no fer-ho als escenaris per a poder dissenyar-los més fàcilment. A part de seguir l'estètica pels pictogrames, la terapeuta ens va dir que l'estil del videojoc havia de ser el més bàsic possible, per que s'entengui que és cada objecte.



*Figura 18 Pictograma usat al joc*

Els pictogrames del projecte s'han obtingut d'un gran repertori ofert per la terapeuta. Les habitacions, igual que les escenes, han estat realitzades per l'autor. Els objectes que hi ha a cada escena que no siguin els pictogrames s'han aconseguit de diferents planes webs. Tanmateix, la majoria d'aquests han estat modificats per l'autor perquè encaixin de forma correcta, per exemple, en el cas de l'escriptori del joc d'ordenar l'habitació, tenia les línies de contorn molt gruixudes i les vam eliminar.

Les interfícies també han estat fetes per l'autor del projecte, no obstant el disseny dels botons s'han obtingut d'un asset de l'Asset Store del Unity\*.

\* Referència bibliografia 2

## 5.2 GAMEPLAY

La paraula gameplay significa en anglès reproducció de joc. Representa la manera com el jugador interactua amb el joc o a la manera com el joc interactua amb el jugador.

Una traducció de gameplay seria jugabilitat.

### 5.2.1 DEFINICIÓ DE REPTES

El repte principal de tots els videojocs és el de completar-lo amb èxit. Malgrat això, per a arribar a assolir aquest repte s'hauran d'anar aconseguint altres reptes diferents, així successivament fins a arribar els reptes més senzills que són aquells que anomenem reptes atòmics (no divisibles).

A l'hora de crear aquests reptes, el dissenyador s'ha de ficar a la pell de qui serà la persona que jugarà. Tanmateix, en aquest cas no hem de tenir en compte tan sols l'opinió del dissenyador, ja que s'ha de tenir en compte que és una eina terapèutica o educativa. Per tant, també haurem de tenir en compte l'opinió de la persona que donarà suport al jugador, en aquest cas, els terapeutes.

A causa de les limitacions del jugador, els reptes es poden arribar a repetir entre minijoc o entre reptes més grossos. En conseqüència, tot i que el repte pugui arribar a ser el mateix, hem d'aconseguir que entre un i altre sigui el suficientment diferent perquè no arribi a avorrir al jugador i que aporti un valor educatiu diferent en cada cas.

### 5.2.2 JERARQUIA DE REPTES

Com s'ha comentat en l'apartat anterior, el repte final és molt clar, completar el videojoc. En uns videojocs pot ser arribant a l'últim nivell, en uns altres pot ser completar tots els minijocs o en altres eliminar tots els enemics. En el cas del nostre joc el propòsit serà completar el dia satisfactòriament completant tots els minijocs que hagi assignat el terapeuta.

En la Figura 19 podem veure el funcionament d'aquestes jerarquies:

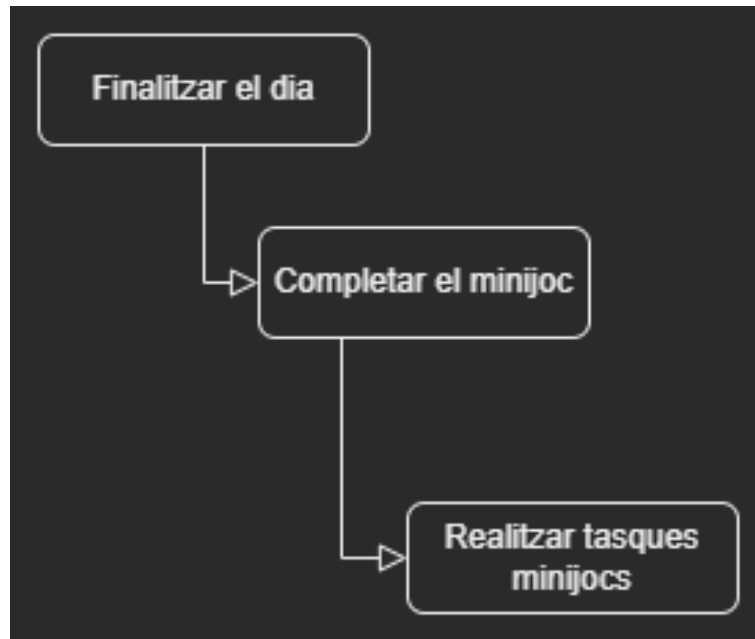


Figura 19 Diagrama jerarquia de reptes

No obstant, cada minijoc afegeix un repte específic. A continuació es mostrarà el repte de cada minijoc.

### MINIJOC VESTIR-SE:

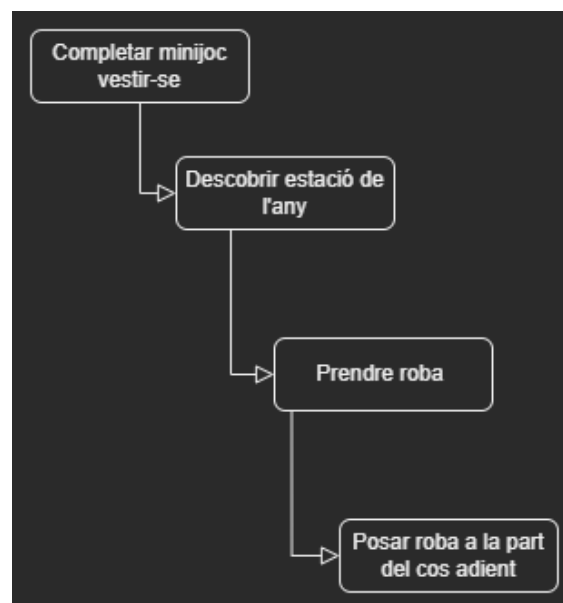


Figura 20 Diagrama jerarquia de reptes minijoc vestir-se

En aquest minijoc el nen ha d'escollir la roba adient per a vestir el personatge. Ha de col·locar-la a la part del cos que correspongui. Si el nen fica tota la roba correctament, haurà completat el minijoc i, si s'equivoca de roba, haurà de buscar una altra peça més adient. Veure la Figura 20.



### MINIJOC FER LA MOTXILLA:

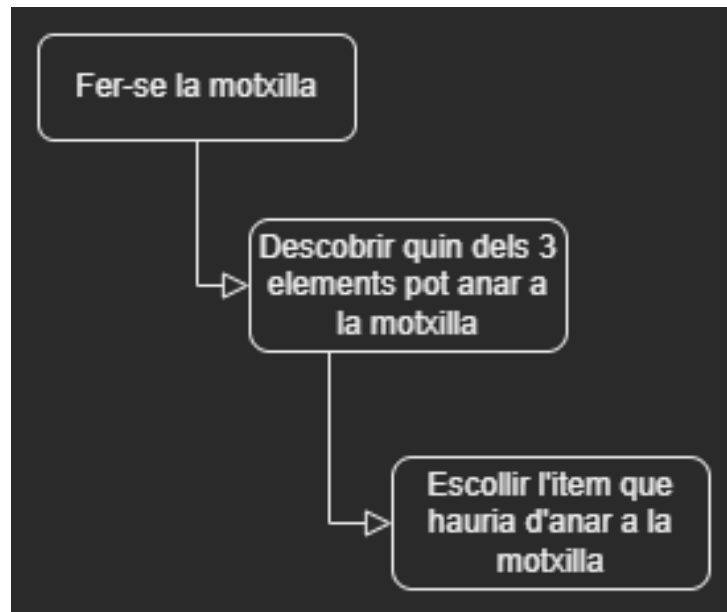


Figura 21 Diagrama jerarquia de reptes minijoc fer motxilla

En aquest minijoc el jugador veurà 3 botons amb 3 pictogrames que representen diferents objectes. El jugador haurà d'escollir quin dels 3 elements és el que hauríem de ficar a la motxilla. Si el jugador respon correctament es veuran 3 pictogrames diferents, altrament haurà de tornar a escollir l'element. Veure Figura 21.

### MINIJOC ORDENAR HABITACIÓ:

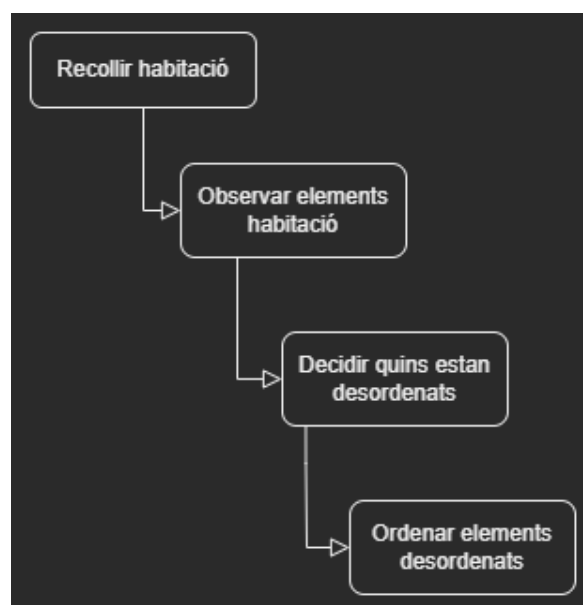
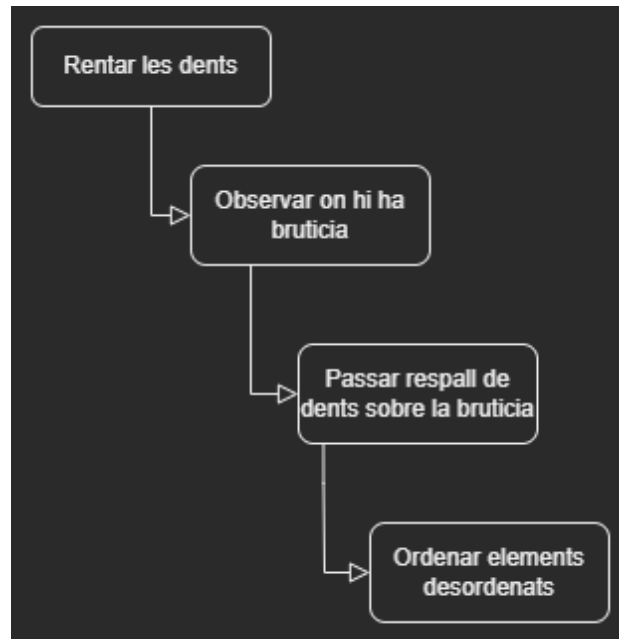


Figura 22 Diagrama jerarquia de reptes minijoc ordenar habitació

En aquest minijoc el jugador veurà la seva habitació i haurà de clicar sobre els elements que estiguin desordenats, per a així ordenar-los. Cada vegada que el jugador entri en aquest minijoc els elements desordenats seran diferents. Si clica sobre l'element desordenat, aquest s'ordenarà, altrament no passarà res. Veure Figura 22.

### **MINIJOC RENTAR DENTS:**



*Figura 23 Diagrama jerarquia de reptes minijoc rentar dents*

En aquest minijoc el jugador haurà de passar el raspall de dents sobre la brutícia que veiem entre les dents. Cada vegada que passem el raspall de dents anirà desapareixent la brutícia. Finalment, la brutícia desapareixerà completament i, si s'ha eliminat tota, finalitzarà el minijoc. Veure Figura 23.

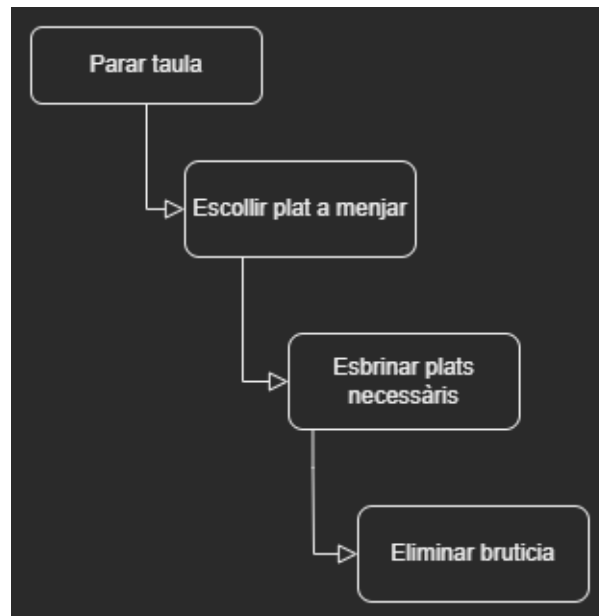
**MINIJOC PARAR TAULA:**

Figura 24 Diagrama jerarquia de reptes minijoc menjar

En aquest minijoc el jugador haurà d'escollir quins plats necessitarà per a menjar l'aliment triat. Si el plat escollit és correcte, aquest anirà a la taula ocupant la posició de la seva silueta, altrament haurà d'escollir un altre plat. Veure figura 24.

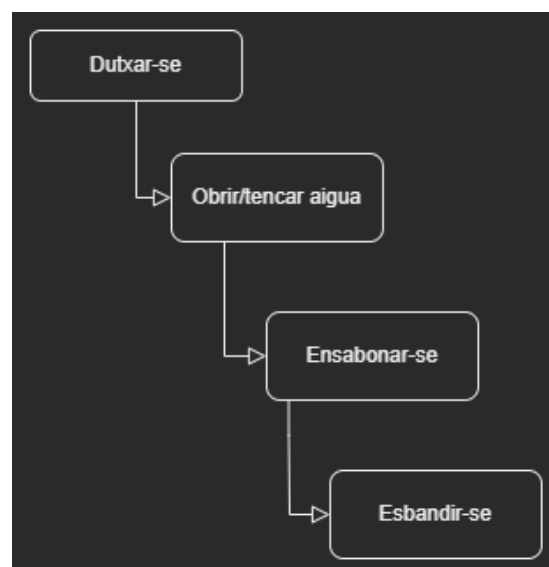
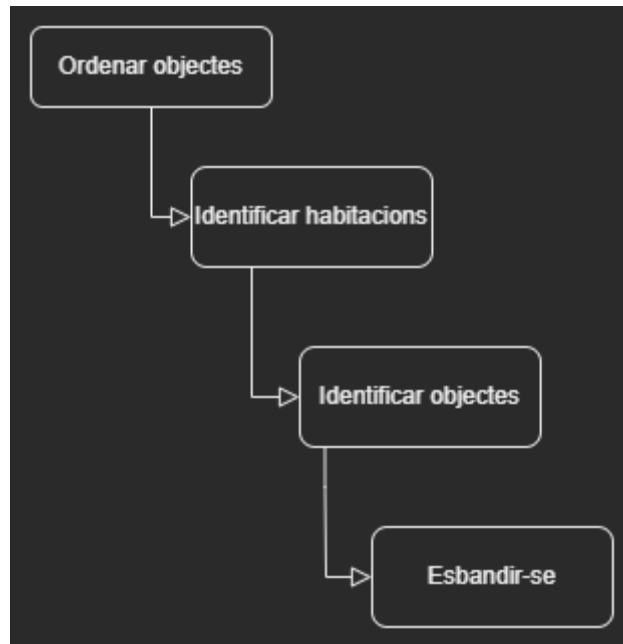
**MINIJOC DUTXAR-SE:**

Figura 25 Diagrama jerarquia de reptes minijoc dutxar-se

En aquest minijoc el jugador haurà de dutxar al personatge. Haurà d'alternar entre aigua i sabó per a netejar la brutícia i eliminar el sabó. El jugador no podrà ensabonar al personatge si l'aigua esta engegada. Un cop el personatge estigui ensabonat el jugador haurà d'obrir l'aigua per a retirar el sabó. Veure Figura 25.

### **MINIJOC ORDENAR OBJECTES:**



*Figura 26 Diagrama jeràrquia de reptes minijoc ordenar habitació*

En aquest minijoc el jugador veurà diferents portes d'habitacions de la casa i un objecte. El jugador haurà d'arrossegar l'element a la porta que toca. Si s'ha equivocat, l'element tornarà a la seva posició inicial, altrament apareixeria un nou objecte a col·locar. Veure la Figura 26.

En els minijocs de dutxar-se i el de parar taula, abans de dutxar-se i parar taula, hi ha un petit minijoc en el que s'ordenen les passes a seguir per a menjar o per a dutxar-se. Apareixeran diferents pictogrames que el jugador haurà d'arrossegar a la posició que correspongui. Si es fiquen a una posició que no toca, tornarà a la posició inicial, altrament es quedarà fixat a la posició que li correspon.

A més a més, en el minijoc de rentar-se les dents, hi ha unes preguntes prèvies relacionades amb la higiene bucal.

A part dels minijocs esmentats també hi ha un apartat en la que el terapeuta decideix les coses que ha de fer el nen al seu dia a dia i un apartat en la que el nen va d'un minijoc a un altre. Considerem que aquestes dues parts no deixen de ser configuració i interfícies.

### 5.2.3 DEFINICIÓ D'ACCIONS

Les accions són el que el jugador pot fer en el món del joc, com podria ser agafar roba, posar-se la roba, escollir item, etc. Per definir quines són les accions ens hauríem de fixar en els reptes de menor nivell, els reptes atòmics, i haurem de pensar quina és l'acció per a superar cadascun d'ells. En el nostre projecte hi ha les següents accions per a poder superar els reptes:

- **Escollir Roba:** Triar peça de roba a agafar de l'armari.
- **Posar Roba:** Posar la peça de roba a la part del cos que toca.
- **Escollir Item:** Decidir quin objecte toca a cada moment.
- **Arrossegar item:** Arrossegar l'objecte al lloc que toca.
- **Rentar dents:** Restregar el raspall de dents on hi ha brutícia.
- **Ensabonar:** Restregar l'esponja on hi ha brutícia.
- **Obrir aigua:** Obrir aigua per a que ens tregui l'espuma del sabó.
- **Escollir activitat:** Escollir l'activitat a fer (acció que farà el terapeuta).

### 5.3 MECÀNIQUES

Als jocs de taula i als videojocs, la mecànica és la regla que regula i guia les accions del jugador, així com la corresponent resposta del joc. La mecànica d'un joc especifica de manera efectiva com funcionarà el joc per a les persones que el juguen.

#### 5.3.1 RECURSOS

Els recursos són objectes del joc o materials que es poden crear, utilitzar, destruir, etc. Són el que en programació entendríem com a classes. A les següents taules podrem veure els recursos del joc.

##### Minijoc motxilla:

###### Nivell motxilla

Qüestions correctes
Qüestions incorrectes
Resposta 1
Resposta 2
Resposta 3

###### Elements Motxilla

Es cert?
----------

**Minijoc vestir-se:****Nivell vestir-se**

Estació de l'any

En el nivell de vestir-se també hi ha imatges que representen l'estació del any.

**Botons roba**

Llista tipus roba

...

Tipus roba

...

Tipus roba posada

...

Els botons de la roba tenen un atribut com el mostrat a la taula anterior per cada tipus de roba que hi ha.

**Drag roba**

Anterior

Eliminat

Puc posar-me?

Objectiu

Tipus roba posada

Posició inicial

**Minijoc ordenar habitació:****Nivell ordenar habitació**

Imatge Correcte

Imatge Incorrecte

Estat

**Minijoc rentar dents:****Nivell rentar dents(preguntes)**

Llista preguntes

Llista respostes

Pregunta actual

Llista preguntes creades

**Nivell rentar dents(rentat)**

Llista brutícia (part alta)

Llista brutícia (part baixa)

Temporitzador

**Arrossegar i eliminar**

Posició inicial Objectes Nombre eliminats
-------------------------------------------------

**Brutícia**

Estat Puc baixar més?
--------------------------

**Minijoc menjar:****Nivell parar taula (part ordenar)**

Nombre ordenats Tipus ordenar Llista pictogrames Llista objectius Llista elements
-----------------------------------------------------------------------------------------------

**Plat**

Plat afegit Posició array Objectiu
------------------------------------------

**Nivell menjar (controlador)**

Puntuació Llista plats a posar
-----------------------------------

**Minijoc ordenar objectes:****Nivell ordenar objectes**

Llista objectes a ordenar Llista objectius
-----------------------------------------------

**Objecte ordenar**

Objectiu
----------

**Minijoc dutxar-se:****Nivell dutxar-se (part ordenar)**

Nombre ordenats
Tipus ordenar
Llista pictogrames
Llista objectius
Llista elements

**Nivell dutxar-se**

Temps
Dutxa activa?

**Brutícia**

Estat
Puc baixar més?

Com hem pogut veure, tenim un gran nombre de recursos. Això és a causa que cada nivell és diferent i s'ha d'interactuar amb altres elements.

Cada minijoc tindrà un controlador que serà l'encarregat de gestionar els minijocs. Com que cada minijoc és diferent i necessiten coses diferents els hem separat per cada un d'ells.

## 5.3.2 ENTITIES

Els entities són les instàncies d'un recurs o l'estat d'algun element del món del joc. En les següents taules podrem veure els entities del joc.

**Minijoc motxilla:**

<b>Recurs: Nivell motxilla</b>	<b>Entitat: Quiz Manager</b>
Qüestions correctes	List<Image>
Qüestions incorrectes	List<Image>
Resposta 1	Button 1
Resposta 2	Button 2
Resposta 3	Button 3
Puntuació	100

<b>Recurs: Elements Motxilla</b>	<b>Entitat: questionButton</b>
Es cert?	true



**Minijoc vestir-se:**

<b>Recurs: Nivell vestir-se</b>	<b>Entitat: ClotheGameController</b>
Estació del any	2

<b>Recurs: Botons roba</b>	<b>Entitat: ChangeClothe</b>
Llista tipus roba	List<image>
...	List<image>
Tipus roba	List<image>
...	List<image>
Tipus roba posada?	true
...	false

<b>Recurs: Drag roba</b>	<b>Entitat: Drag</b>
Anterior	GameObject Ex:Interior
Eliminat	None
Puc posar-me?	true
Objectiu	GameObject Ex:interiorCol
Segons	List<GameObject> [Pant. Ll, Pant. c]
Posició inicial	Vec2(0,0)

**Minijoc ordenar habitació:**

En aquest minijoc el controlador tan sols escull els elements a desordenar, i mira quins s'han ordenat. Per tant, tindrà una llista de gameobjects i un enter amb el nombre d'elements ordenats.

<b>Recurs: Nivell ordenar habitació</b>	<b>Entitat: DifferenceElement</b>
Imatge Correcte	Image
Imatge Incorrecte	Image
Estat	0

**Minijoc rentar dents:**

<b>Recurs: Nivell rentar dents</b>	<b>Entitat: questionsTeethcontroller</b>
Llista preguntes	List<string>
Llista respostes	List<bool>
Pregunta actual	"Ens hem de rentar les dents...?"
Llista preguntes creades	List<string>

<b>Recurs: Arrossegar i eliminar</b>	<b>Entitat: DragRemove</b>
Posició inicial	Vec2(0,0)
Objectes	List<GameObj>
Nombre eliminats	4

<b>Recurs: Brutícia</b>	<b>Entitat: Dirt</b>
Estat	2
Puc baixar més?	true

### Minijoc menjar:

<b>Recurs: Nivell parar taula (part ordenar)</b>	<b>Entitat: Ordercontroller</b>
Nombre ordenats	0
Tipus ordenar	1
Llista pictogrames	List<Image>
Llista objectius	List<GameObj>
Llista elements	List<GameObj>

<b>Recurs: Plat</b>	<b>Entitat: Dishadder</b>
Plat afegit	Image
Posició array	1
Objectiu	GameObj (knife collider)

<b>Recurs: Nivell menjar (controlador)</b>	<b>Entitat: FoodManager</b>
Puntuació	100
Llista plats a posar	List<bool>

### Minijoc ordenar objectes:

<b>Recurs: Nivell ordenar objectes</b>	<b>Entitat: OrderObectsController</b>
Llista objectes a ordenar	List<GameObj>
Llista objectius	List<GameObj>

<b>Recurs: Objecte ordenar</b>	<b>Entitat: DragObject</b>
Objectiu	GameObject (collider cuina)

**Minijoc dutxar-se:**

Recurs: <b>Nivell dutxar-se (part ordenar)</b>	Entitat: <b>OrderController</b>
Nombre ordenats	0
Tipus ordenar	1
Llista pictogrames	List<Image>
Llista objectius	List<GameObj>
Llista elements	List<GameObj>

Recurs: <b>Nivell dutxar-se</b>	Entitat: <b>ShowerController</b>
Temps	1:29
Dutxa activa?	true

Recurs: <b>Brutícia</b>	Entitat: <b>Dirt</b>
Estat	2
Puc baixar més?	true

## 5.3.3 DESCRIPCIÓ DE LES MECÀNIQUES

En aquest apartat farem un breu resum de les mecàniques que hi ha a cada minijoc.

**Minijoc motxilla:**

- Clicar botó objecte: En clicar aquest botó, si aquest representa un element que pot ser necessària per a la motxilla passarem a la següent selecció de botons amb resultat correcte. En cas contrari haurà de tornar a escollir un botó fins a obtenir la resposta correcta.

**Minijoc vestir-se:**

- Clicar botó roba: Al clicar aquest botó apareixerà la peça de roba representada. Si la peça ja està a la pantalla, canviarà el model de roba.
- Arrossegar peça roba: Al clicar sobre la peça de roba i moure el ratolí, mourem la peça de roba. Si la deixem anar al lloc indicat, si es pot posar, es quedarà allà. Altrament, tornarà a la posició inicial.

**Minijoc ordenar habitació:**

- Clicar element: Al clicar sobre un element de l'habitació, si aquest està desordenat s'ordenarà automàticament. Si ja està ordenat, ens constarà com a error.

**Minijoc rentar dents:**

- Clicar botó verdader/fals: Al clicar sobre el botó, si responem correctament a la pregunta, passarem a la següent. Si no se'ns restarà punts.
- Arrossegar raspall de dents: Al clicar sobre el raspall i moure el ratolí, aquest es mourà. Si el raspall passa per sobre brutícia aquest l'anirà fent desaparèixer.

**Minijoc menjar:**

- Clicar botó plat: Al clicar sobre el botó, si escollim un plat necessari per a l'aliment escollit, aquest apareixerà a la taula en el seu lloc.
- Arrossegar tasques: Al clicar sobre la tasca a fer per menjar i moure el ratolí, aquest es mourà. Si deixem anar el pictograma amb l'acció sobre el lloc on ha d'anar aquest, es quedarà allà. Altrament, tornarà a la posició inicial.

**Minijoc ordenar objectes:**

- Arrossegar objectes: Al clicar un objecte i moure el ratolí, aquest es mourà. Si deixem anar el pictograma amb l'objecte sobre el lloc on ha d'anar, aquest es quedarà allà. Altrament, tornarà a la posició inicial.

**Minijoc dutxar-se:**

- Arrossegar tasques: Al clicar sobre la tasca a fer per menjar i moure el ratolí, es mourà. Si deixem anar el pictograma amb l'acció sobre el lloc on ha d'anar, es quedarà allà. Altrament, tornarà a la posició inicial.
- Arrossegar esponja: Al clicar sobre l'esponja i moure el ratolí, es mourà. Si l'esponja passa per sobre brutícia, l'anirà fent desaparèixer.
- Apagar/encendre aigua: Al clicar el botó d'engegar/apagar aigua, si s'engega l'aigua i el personatge té sabó al cos, aquest anirà desapareixent a poc a poc fins a desaparèixer.

### 5.3.4 OBJECTIUS

L'objectiu del joc, és finalitzar el dia havent completat totes les tasques assignades pel tutor. En aquest joc, al ser una eina terapèutica i educativa, el jugador no tindrà l'opció de perdre. Tanmateix, sí que se l'avisarà si fa alguna cosa malament, però haurà d'acabar realitzant bé la tasca per a passar de nivell. Això ho hem fet, ja que, parlant amb la terapeuta, vam arribar a la conclusió de que sí que s'havia d'avisar al jugador si feia alguna cosa malament, no obstant, han d'aprendre el que han fet malament i arreglar-ho.

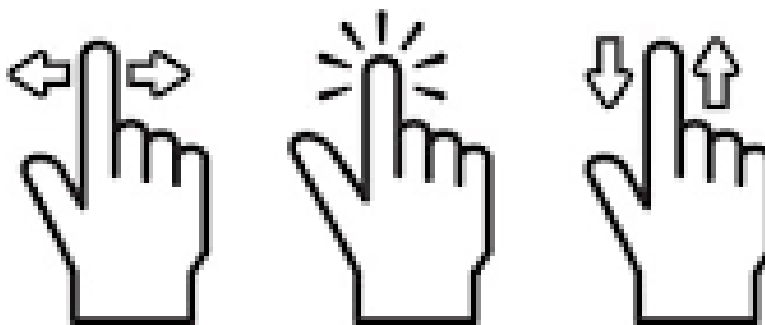
Si féssim que perdessin el joc, podrien perdre la motivació per a continuar.

### 5.3.5 CONTROLS BÀSICS

Aquest prototip ha estat desenvolupat perquè es pugui jugar tant amb PC com amb dispositius mòbils i tauletes. Tot i que la idea principal és que el joc sigui tan sols per mòbils i tauletes, hem fet que els controls siguin també bons per al PC, ja que als terapeutes, actualment, els és més fàcil provar-los d'aquesta manera.

Les accions a realitzar per als pacients són molt senzilles i no les volem dificultar més per si algun té problemes motors. Com s'ha dit a l'inici de la memòria, volem que el joc sigui apte per tants pacients com sigui possible.

Com podem veure a la Figura 27, els gestos que poden fer els jugadors són: prémer la pantalla, i moure el dit per la pantalla.



*Figura 27 Figura que mostra els controls tàctils que es fan servir*

A part dels controls tàctils, per al moment, si juguem en PC es poden fer els mateixos gestos que es fan amb el dit amb el cursor del PC.

## 5.4 FLOWCHART

El flowchart és un tipus de diagrama que representa un procés o un flux de treball. En la figura 28 es pot veure el flowchart del joc, ensenyant les diferents pantalles que hi ha, i com s'accedeixen a elles.

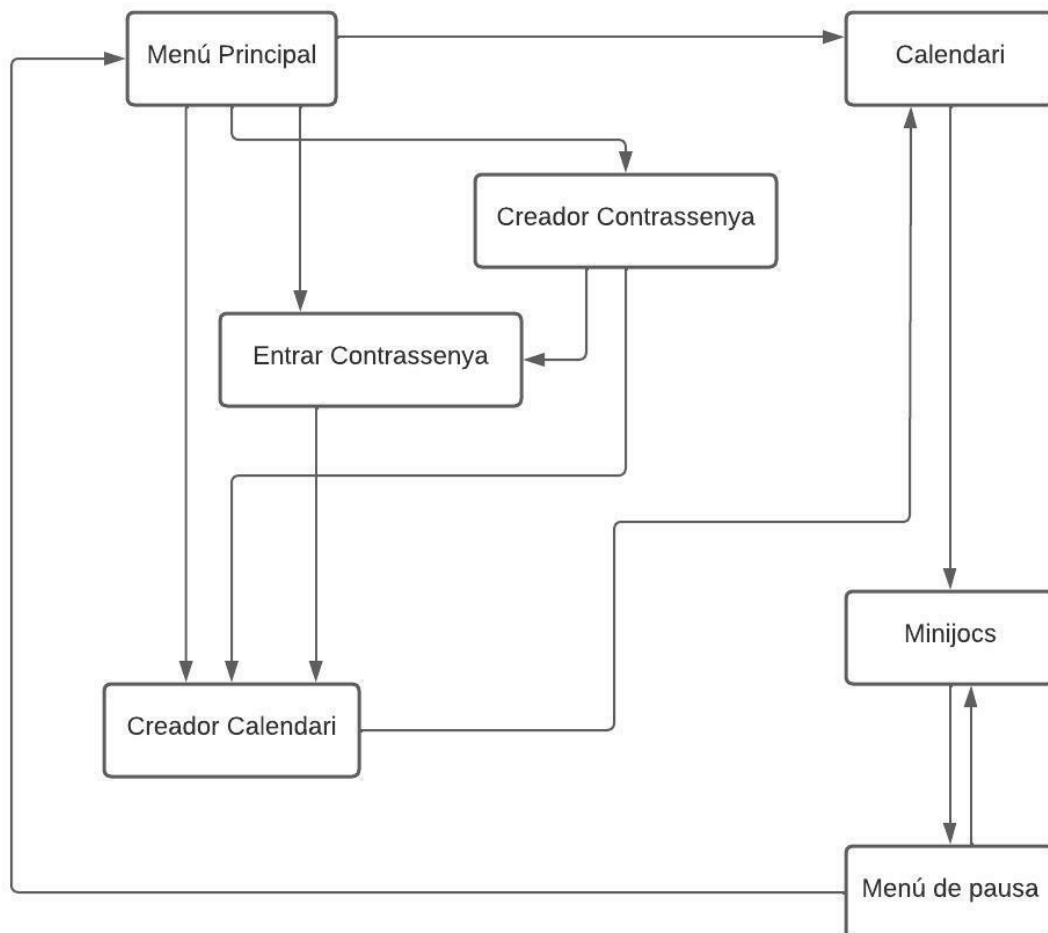


Figura 28 Figura que mostra el flowchart

Com podem veure en la Figura 28, al flowchart, per a fer o refer el calendari només hi podem accedir des del menú principal. Ja que considerem que tan sols el tutor pot escollir el calendari, hem fet que s'hagi de fer una contrassenya. Un cop hi hagi un calendari creat, el jugador pot entrar al calendari, i des del calendari pot anar als diferents minijocs que estiguin definits. Mentre estem als minijocs, podem entrar al menú de pausa. Des d'allà, podem tornar al minijoc de dues maneres: reiniciant-lo o continuant on estàvem. Al menú de pausa podem tornar al menú principal.

## 5.5 LEVEL DESIGN

Normalment, quan parlem del disseny de nivells, creiem que hi ha una successió de nivells en què es va avançant el joc i la història. Tot i que hi ha jocs que són 100% lineals, n'hi ha que es poden anar completant amb ordres diferents, o fins i tot amb finals diferents. En el cas del nostre videojoc, al no tenir una història concreta, i que es pot personalitzar tant l'ordre com les tasques a realitzar, l'ordre no està definit.

A part d'això, en el disseny dels nivells l'escenari és molt important, ja que aquest pot decidir els reptes que es poden trobar a cada nivell. Com hem explicat anteriorment, cada minijoc té una diferenciació amb els altres, i busquem que la dificultat de passar o no el joc es relacioni amb la dificultat que podria tenir el nen en realitzar les activitats a casa o recordar que les hagi de fer.

### 5.5.1 ECONOMIA INTERNA

L'economia és el sistema pel qual recursos i entitats són produïts, consumits, i intercanviats en quantitats quantificables. Un exemple seria la vida d'un personatge en un joc de plataformes, que en caure al buit se li resta una vida de les que té per a passar el nivell. El nostre joc no té una economia pensada com a tal. No obstant això, la puntuació pot ser representada com a tal.

#### **Puntuació:**

El jugador començarà cada minijoc amb la puntuació de 100. Aquesta puntuació anirà disminuint si el jugador fa accions incorrectes, o respon de manera incorrecta alguna pregunta.

Un cop finalitzar el minijoc apareixerà un panell, que detallarem més endavant en l'apartat de les interfícies, amb estrelles. Depenent de la puntuació que tingui el jugador a l'acabar el minijoc aquesta puntuació es canviarà per estrelles de la següent manera:

- Puntuació superior a 75: 3 estrelles.
- Puntuació entre 50 i 75: 2 estrelles.
- Puntuació entre 0 i 50: 1 estrella.

Com hem comentat en altres apartats, mai hi haurà un minijoc sense completar. Tanmateix, les estrelles definiran de quina manera el pacient ha superat la tasca.

## 5.6 CANVIS REALITZATS DE DISSENY

Com hem comentat a l'apartat de planificació del projecte, hi havia un contacte constant tant amb la terapeuta com amb el tutor del TFG. Cada dues setmanes tenia una reunió amb la terapeuta. Durant aquesta reunió li ensenyava el prototip de cada minijoc i ella em retornava un feedback, si aquest era positiu passàvem a implementar el joc final, altrament modificàvem el prototip i anàvem iterant fins a obtenir el resultat idoni. A més a més de les reunions, tenia un contacte directe amb la Sra. Anna Valls, cosa que permetia anar avançant tot i no tenir reunions.

Alguns dels canvis més importants realitzats després de debatre amb els tutors han estat els següents:

- La idea inicial era que el videojoc estigues enfocat per adolescents, no obstant, a l'inici del desenvolupament, vam decidir enfocar-ho a nens, ja que era amb el grup que podíem provar-ho, i els podria interessar més el videojoc.
- En el minijoc de parar taula, al principi estaven tots els plats a una espècie d'armari, i el jugador havia d'arrossegar els necessaris a la taula. No obstant, pensant com millorar, l'Anna ens va comentar que podríem fer que a la taula ja hi hagués les siluetes dels plats, i que tant sols clicant un botó amb el pictograma del plat a ficar ja anés directament allà. Ens va argumentar que això aniria bé ja que el nen interioritzaria on ha d'anar cada cosa a la taula.
- En el minijoc de vestir-se, inicialment volíem que el nen es vestís de la manera que ell preferís. No obstant, la terapeuta ens va dir que li agradaria que el nen es vestís segons l'estació de l'any. Vam afegir una finestra en la qual es veia fora i el nen pot interpretar el temps que fa.
- Va haver-hi una decisió que va canviar bastant com teníem pensat que seria el joc. Aquesta decisió va ser la de fer un joc pla sense animacions, en el que tan sols es moguessin els elements amb els quals pot interactuar el pacient. Aquesta decisió es va prendre al provar un petit prototip i adonar-nos que, al fer que el personatge es mogués per la casa, el pacient es confonia. Parlant amb la Sra. Anna, vam consensuar que el millor seria deixar el joc sense animacions i centrar-nos en facilitar al pacient entendre la mecànica i el perquè de cada cosa.

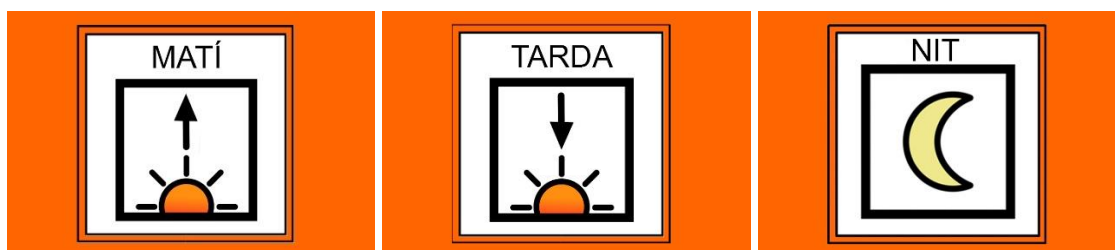


## 5.7 HISTÒRIA

Com hem comentat abans, aquest joc és educatiu i terapèutic. Tot i que sí que volem que sigui lúdic per tal d'atrapar al pacient i que aquest vulgui entrenar, no està pensat per a ser un joc lúdic com a tal. En aquest cas no hem considerat que sigui necessari desenvolupar una història més enllà de què hem de realitzar les tasques que el tutor o terapeuta hagi assignat.

Sí que és cert que podríem fer que el nen triés la tasca que volgués en cada moment, però vam creure més adient ficar d'excusa que s'ha de completar un dia sencer, perquè d'aquesta manera el nen té clar un objectiu, i així, a base de repetició, aprèn que ha de fer cada dia i en quin moment.

Un cop el nen va realitzant les tasques, el dia va avançant. Això ho veurem en el calendari on hi ha pictogrames que representen el matí, la tarda i la nit. Podem veure aquests pictogrames a la Figura 29.



*Figura 29 Pictogrames que representen el matí, la tarda i la nit.*

### 5.7.1 MÓN DEL JOC

**DIMENSIÓ FÍSICA:** El prototip contindrà minijocs que representaran les diferents habitacions de la casa segons el minijoc. Aquestes habitacions estaran dissenyades en un món 2D on els objectes es poden desplaçar tan verticalment com horitzontalment.

**DIMENSIÓ TEMPORAL:** Cada partida d'aquest prototip representarà un dia a la vida del pacient que hi juga. Per a fer-lo més real pel pacient, fem que el tutor o terapeuta pugui configurar les tasques a realitzar i l'ordre. Per tant, pot haver-hi una gran varietat d'ordre a l'hora de realitzar les tasques.

**DIMENSIÓ AMBIENTAL:** L'estètica del joc és actual. Això ho hem fet, ja que una de les preocupacions principals dels terapeutes era que el pacient entengués que el representat podria ser casa seva i què és el que ha de fer ell a la vida real.

**DIMENSIÓ EMOCIONAL:** L'emoció més present en el transcurs de la història és la superació. Això és a causa que probablement molts dels pacients els costarà realitzar algunes tasques. No obstant, amb l'ajuda del tutor i a mesura que passi el temps, ho aniran aconseguint i interioritzant, que és l'objectiu terapèutic del joc.

## 5.8 INTERFÍCIE GRÀFICA

En aquest apartat explicarem el conjunt d'escenes que té el projecte i els elements més importants que anirem trobant al llarg del joc.

### 5.8.1 MENÚ PRINCIPAL

Com hem comentat en altres apartats, el que realitzarà el calendari del pacient serà el terapeuta. Per a poder modificar o crear un calendari, aquest haurà de posar una contrasenya que primer haurà de ser creada.

En les següents figures veurem totes les pantalles que podrem trobar en el menú principal.

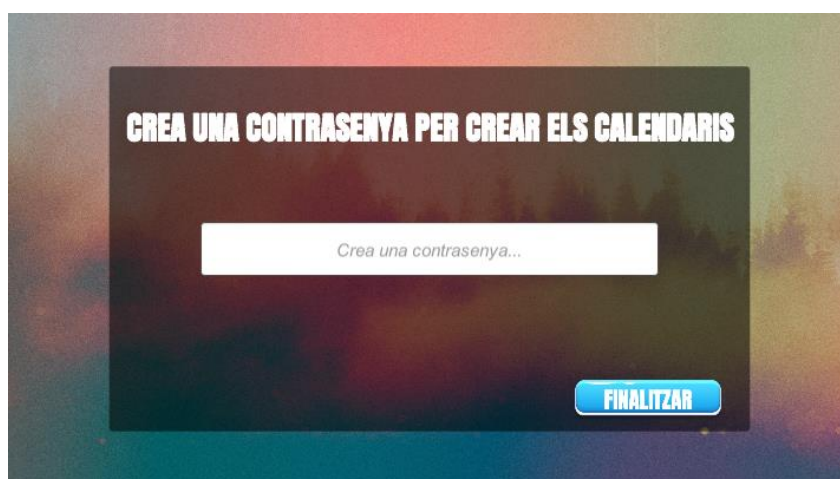


Figura 30 Figura que mostra la pantalla per crear la contrasenya

En iniciar el joc per primera vegada, ens sortirà una pantalla per crear la contrasenya per a poder fer els calendaris. Veure Figura 30.



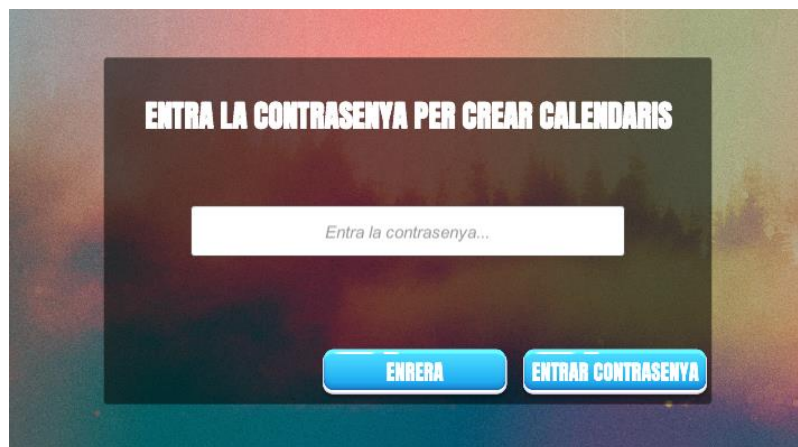
Figura 31 Figura que mostra les dos possibilitats de menús principals que trobarem

El primer cop que entrem el joc i encara no haguem fet cap calendari, veurem el menú d'inici com la part esquerra de la Figura 30, ja que no podem jugar. En canvi si hem fet algun calendari, podrem començar a jugar.



*Figura 32 Figura que mostra que passa al clicar començar*

Si cliquem a començar, s'obrirà aquest menú. Si cliquem a començar des de punt guardat, continuarem des d'on hagem acabat l'última vegada que vam jugar. Si cliquem a començar dia de nou, començarem de nou el dia de l'últim calendari creat. Veure la Figura 32.



*Figura 33 Figura que mostra que passa al clicar fer/refer calendari*

Si cliquem a refer/fer calendari, anirem a la pantalla que es mostra a la figura 33. Aquí podem veure que es demana una contrasenya per a crear el calendari. Aquesta contrasenya és la que hem creat a la Figura 30.

Com hem comentat en altres apartats, hem fet que, per a crear un calendari, s'hagi d'entrar una contrasenya, per així si algun pacient no vol realitzar alguna tasca, aquest no la pugui eliminar o canviar.

### 5.8.2 CREADOR CALENDARI

Per a poder crear el calendari hem creat una escena en la qual ens diu quin moment del dia estem. A més a més, a la Figura 34 podem veure que hi ha 4 barres d'opcions. Aquí podrem seleccionar la tasca a fer, per tant, com a molt podrem fer 4 per cada moment del dia.

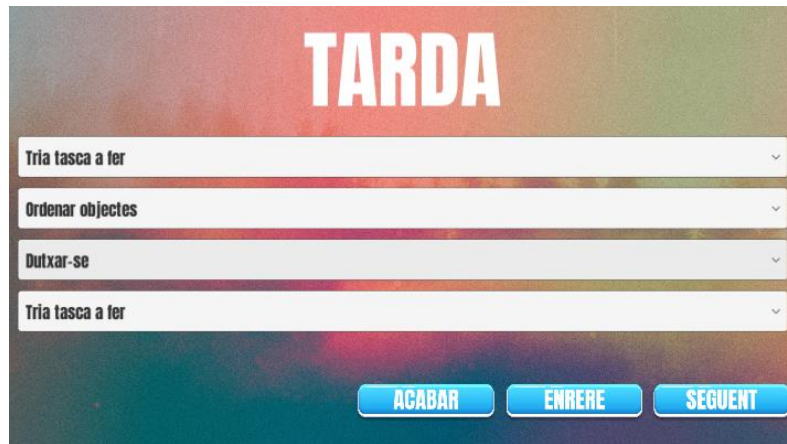


Figura 34 Figura que mostra el creador de calendari

Com podem veure a la Figura 34, se'ns indica que és la tarda. En aquestes 4 barres d'opcions podem triar l'activitat que vulguem. Un cop fet la tria, si cliquem a "següent" passarem a la nit, si cliquem a "acabar" haurem acabat el calendari, i "enrere", si ens hem equivocat, podem modificar el matí. S'ha decidit fer la imatge de la tarda, ja que surten trossos dels botons, perquè quan estiguem fent matí no podrem tornar enrere i quan estiguem a nit no podrem passar al següent.

### 5.8.3 CALENDARI

El calendari, com hem comentat en altres apartats, és l'eina que té el jugador per a anar al minijoc que li toca a cada moment. A la Figura 35 podem veure com és el calendari.



Figura 35 Figura que mostra el calendari

Com podem veure a la Figura 35, es veu un calendari. En aquest cas podem veure que estem al matí i que hem completat la tasca de vestir-se. Les tasques que ja hem realitzat es veuen amb una opacitat menor i amb un tick verd a sobre. Com podem veure, la següent tasca a realitzar és la d'esmorzar.

#### 5.8.4 MENÚ DE PAUSA

S'ha creat un petit menú de pausa per si el pacient vol parar un moment el joc, o per si ha de sortir. També hem permès que es pugui reiniciar el nivell per si el jugador vol obtenir les 3 estrelles i veu que s'ha equivocat. A la figura 36 veurem com hem fet aquest menú de pausa.

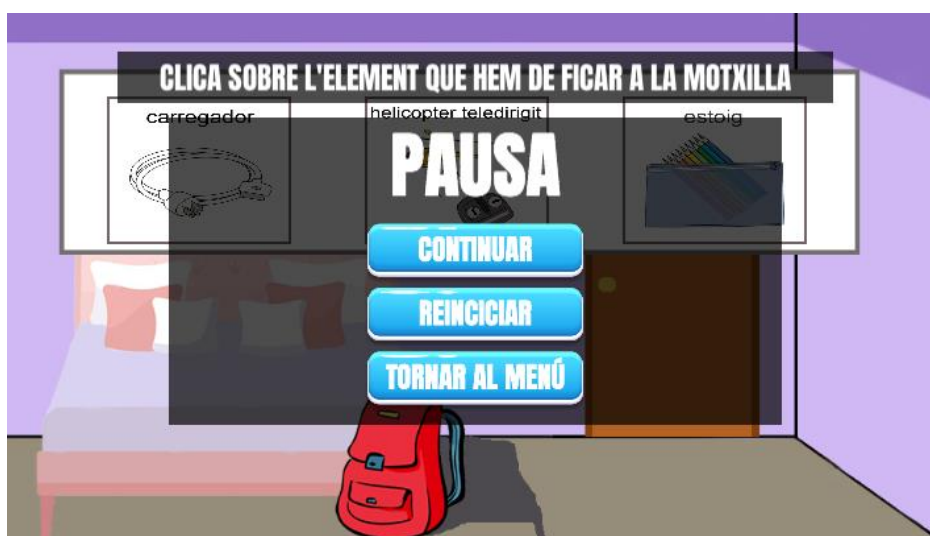


Figura 36 Figura que mostra el calendari

Com podem veure a la Figura 36, un cop som dintre del menú de pausa podem continuar, reiniciar o tornar al menú inicial. Si continuem, el minijoc continuarà en el mateix punt que quan hem premut el botó de pausa. Si reiniciem, el minijoc s'iniciarà de nou i el jugador haurà de començar el minijoc de nou sense tenir en compte la puntuació que ja portava en el moment de parar el joc. En el cas de tornar al menú tornariem al menú principal.

### 5.8.5 MENÚ DE PUNTUACIÓ

Com hem comentat en altres apartats, tot i que cada minijoc tindrà la seva puntuació, el jugador la veurà reflectida en estrelles. Un cop finalitzat el minijoc transformarem els punts en estrelles, tal com hem comentat en l'apartat d'economia, i es veuran en una interfície senzilla que podem veure a la figura 37.



*Figura 37 Figura que mostra la interfície de puntuació*

Com podem veure a la Figura 37 i tenint en compte el comentat a l'apartat d'economia, podem veure que el jugador ha obtingut tres estrelles i, per tant, entre 75 i 100 punts.

### 5.8.6 NIVELLS

En aquest apartat mostrarem la interfície gràfica de cada minijoc. Com ja hem comentat anteriorment, cada mijoc és diferent i, per tant, haurem d'explicar cada un d'aquests de forma separada.

### 5.8.6.1 MINIJOC VESTIR-SE



Figura 38 Minijoc vestir-se

La Figura 38 mostra el que veurà el jugador en aquest minijoc. Com podem veure a la part esquerra, hi ha un armari amb pictogrames de roba. Al clicar aquest pictograma, apareixerà la roba al centre de la pantalla, en aquest cas veiem un jersei rosa. La roba que aparegui en el centre de la pantalla, la podrem agafar i arrossegar fins a la part del cos on creiem que ha d'anar posada. A la part de dalt a la dreta podem veure una rodona blava, al clicar allà s'obrirà el menú de pausa.

### 5.8.6.2 MINIJOC FER LA MOTXILLA



Figura 39 Minijoc fer la motxilla

La Figura 39 mostra el que veurà el jugador en aquest minijoc. Com podem veure a la part de dalt veurem 3 botons amb diferents objectes. El jugador haurà de clicar sobre l'element que cregui que ha d'anar dintre la motxilla. A la part de dalt a la dreta podem veure una rodona blava, al clicar allà s'obrirà el menú de pausa.

### 5.8.6.3 MINIJOC ORDENAR HABITACIÓ



Figura 40 Minijoc ordenar habitació

La Figura 40 mostra el que veurà el jugador en aquest minijoc. Com podem veure hi ha diferents elements a l'habitació. Si el jugador creu que l'element està desordenat, l'haurà de clicar, i si està desordenat aquest s'ordenarà. A la part de dalt a la dreta podem veure una rodona blava, al clicar allà s'obrirà el menú de pausa.

### 5.8.6.4 MINIJOC RENTAR DENTS

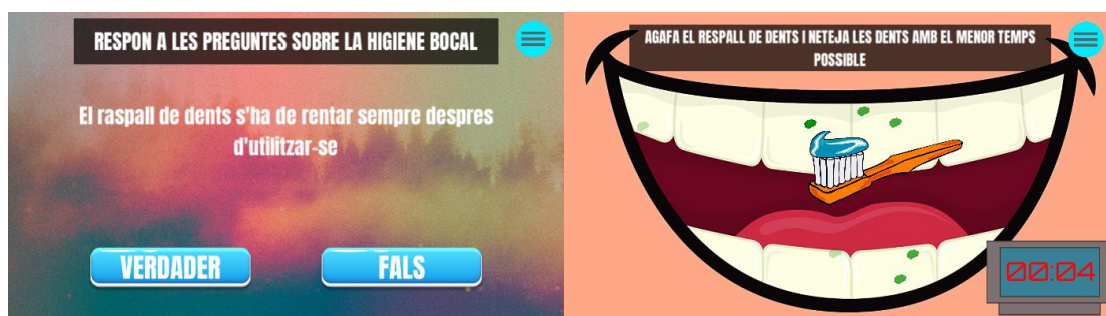


Figura 41 Preguntes higiene bocal

Figura 42 Minijoc Rentar dents

La Figura 41 mostra la part de les preguntes en el minijoc de rentar-se les dents. El jugador haurà de clicar verdader o fals segons les preguntes que li facin. En la Figura 42 podem veure una boca oberta amb brutícia. El jugador haurà d'agafar el raspall de dents i eliminar la brutícia que té el personatge a les dents. A més a més, hi ha un rellotge que marca el temps que tarda a rentar les dents. Segons el temps que hagi estat el jugador la puntuació serà més alta o més baixa. A la part de dalt a la dreta de les dues imatges podem veure una rodona blava, al clicar allà s'obrirà el menú de pausa.



### 5.8.6.5 MINIJOC MENJAR



Figura 43 Pantalla escollir aliment

Figura 44 Minijoc Parar taula

La Figura 43 mostra la pantalla que tindrem per a escollir l'aliment. En canvi, la Figura 44 mostra una taula amb elements col·locats i les siluetes dels altres. A la part dreta podem veure pictogrames amb plats. El jugador ha de clicar sobre la imatge que cregui que necessita per a menjar el plat triat. A la part de dalt a la dreta de les dues imatges podem veure una rodona blava, al clicar allà s'obrirà el menú de pausa. Aquest minijoc serà igual per a esmorzar, dinar i sopar.

### 5.8.6.6 MINIJOC ORDENAR OBJECTES



Figura 45 Preguntes ordenar objectes

La Figura 45 mostra el minijoc d'ordenar els objectes. A la imatge es veuen una sèrie de portes amb el pictograma de l'habitació que hi ha al seu darrere. A dalt al centre es veu un pictograma amb un objecte. El jugador haurà d'arrossegar l'objecte fins a la porta on creu que ha d'anar. A la part de dalt a la dreta de les dues imatges podem veure una rodona blava, al clicar allà s'obrirà el menú de pausa.

## 5.8.6.7 MINIJOC DUTXAR-SE

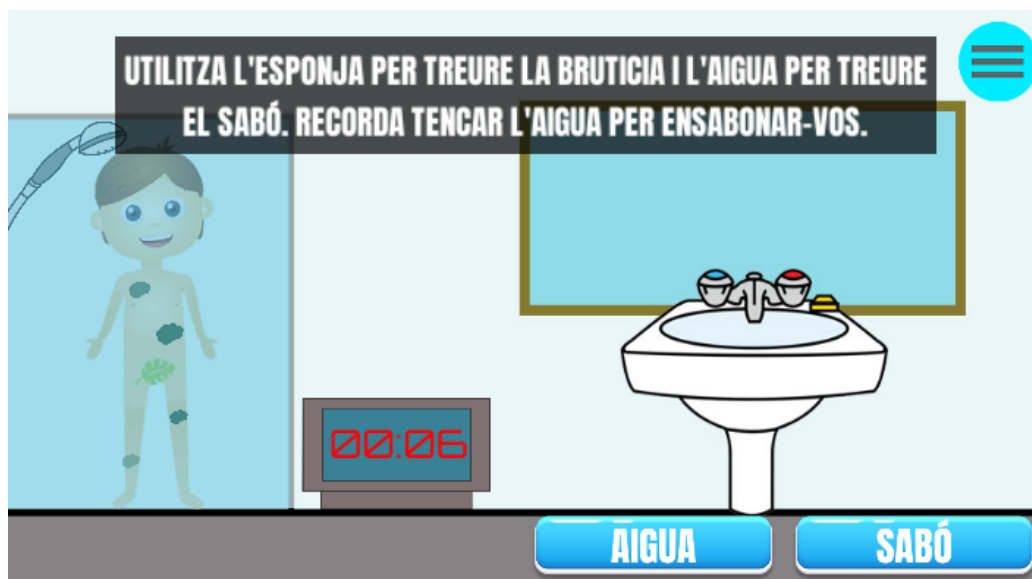


Figura 46 Minijoc dutxar-se

La Figura 46 mostra el minijoc de dutxar-se. Podem veure el mateix nen que al minijoc de vestir-se, però aquest nen té brutícia al cos. Si cliquem sabó sortirà una esponja que hem de passar per sobre la brutícia. Un cop aparegui sabó on hi havia brutícia, haurem d'obrir l'aigua per a treure'l. A la part de dalt a la dreta de les dues imatges podem veure una rodona blava, al clicar allà s'obrirà el menú de pausa.

## 5.8.6.7 PART D'ORDENAR

En el minijoc de menjar i el de dutxar-se hi ha una part prèvia en la qual el nen ha d'ordenar tasques. En les Figures 47 i 48 podem veure com són aquestes interfícies.

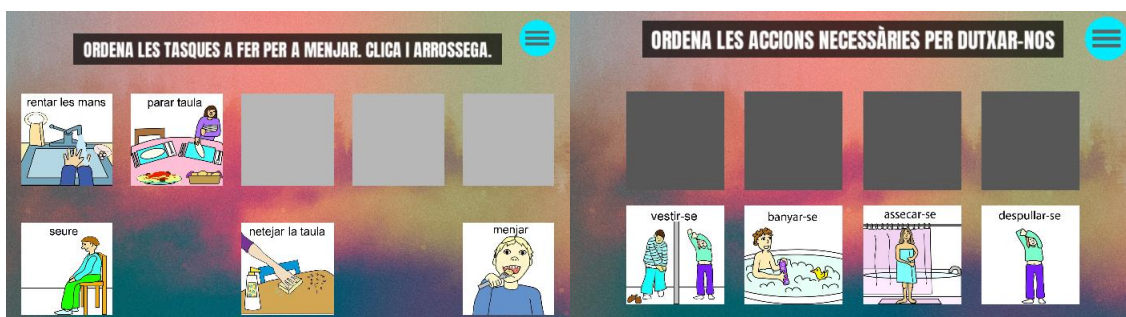


Figura 47 Part ordenar menjar

Figura 48 part ordenar dutxar-se

## 5.9 ART DEL JOC

En aquest projecte s'ha treballat el disseny dels escenaris. La majoria dels altres Sprites han estat obtinguts d'artistes trobats per internet o dels pictogrames usualment emprats pels terapeutes.

### 5.9.1 ESCENARIS

En el cas dels escenaris, han estat dissenyats per l'autor del projecte. Com s'ha comentat en altres apartats, s'ha dissenyat amb Photoshop. En les Figures 49, 50, 51, 52 i 53 podem veure els escenaris dissenyats sense els elements interactuables del joc:

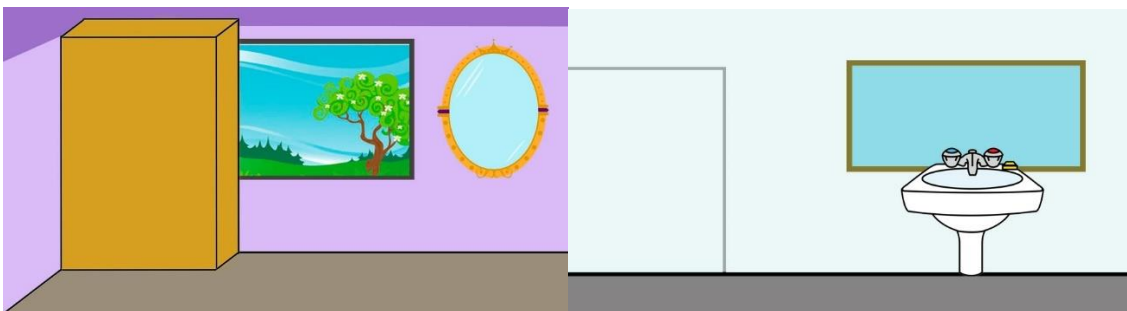


Figura 49 Part ordenar menjar

Figura 50 part ordenar dutxar-se

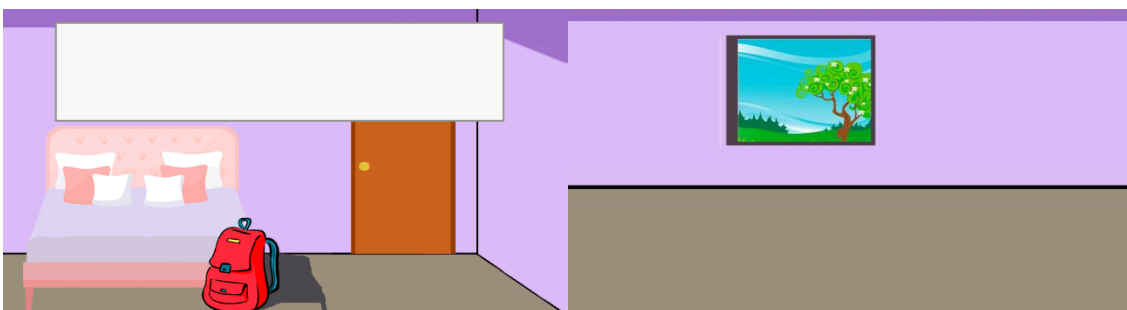


Figura 51 Part ordenar menjar

Figura 52 part ordenar dutxar-se

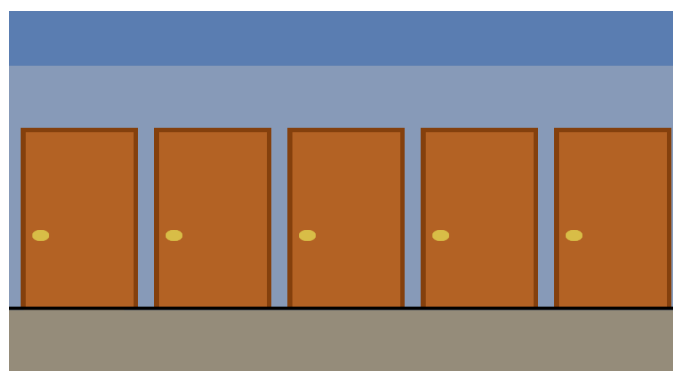


Figura 53 Part ordenar menjar

### 5.9.2 PERSONATGES

Com s'ha comentat en altres apartats, aquest joc no té personatges com a tal. No obstant, en els minijocs de vestir-se i dutxar-se necessitem un personatge per a poder interactuar. El personatge usat l'hem trobat per internet<sup>1</sup>. Aquest personatge és el que veiem a la Figura 54. El personatge no és important pel joc, no té cap pes a la història ni és necessari descriure'l. És un simple element del videojoc.



Figura 54 Personatge minijocs vestir-se i dutxa

### 5.9.3 PICTOGRAMES

En altres apartats hem comentat que hem fet servir pictogrames perquè els pacients puguin interpretar correctament diferents elements. Aquests pictogrames han estat subministrats per la Sra. Anna Vall. Els pictogrames han estat traduïts del castellà al català per l'autor amb Photoshop. Podem veure un exemple a la Figura 55.

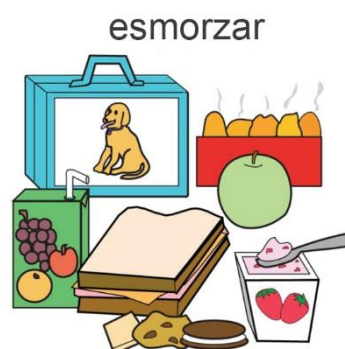
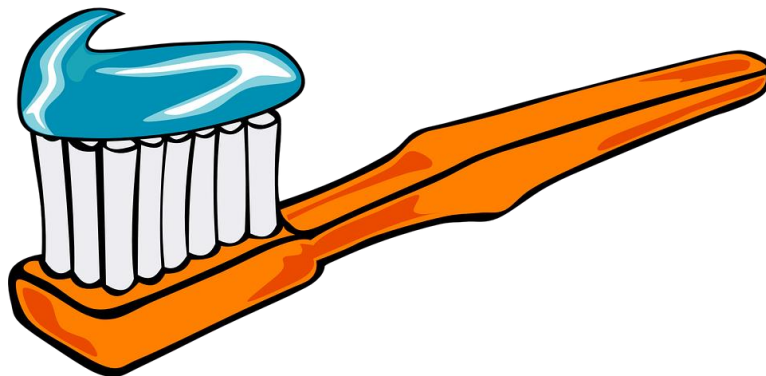


Figura 55 Pictograma que representa esmorzar

<sup>1</sup> <https://www.freejpg.com.ar/imagenes/premium/1202807146/nir-os-de-dibujos-animados-en-la-ilustracion-vectorial-de-ropa-interior>

#### 5.9.4 ALTRES SPRITES

En alguns minijocs hem utilitzat Sprites que no han estat realitzades per l'autor. La majoria dels trobats per internet han estat modificats per l'autor per a no ser iguals. Aquests Sprites els hem obtingut de pixabay<sup>1</sup>, a la Figura 56 podem veure l'exemple d'un Sprite utilitzat en el projecte desenvolupat.



*Figura 56 Asset d'un raspall de dents.*

#### 5.10 EFECTES DE SO

En aquest prototip hem implementat una sèrie de sons. Els que consten són els que hem cregut necessaris per als jugadors. Primer, hem cregut necessari un efecte sonor que ens indiqui si hem realitzat bé una tasca o no. Aquests dos sons els hem trobat a la biblioteca de sons freesound<sup>1</sup>. D'altra banda, la terapeuta va veure necessària una petita explicació sonora del minijoc perquè els nens que no sàpiguen llegir puguin entendre el que s'ha de fer a cada un d'ells. Aquests sorolls han estat gravats per l'autor i més endavant ens agradaria gravar-ho per un professional en un entorn adequat.

<sup>1</sup> <https://pixabay.com/no/vectors/tannb%C3%B8rste-tannpasta-tenner-297768/>

<sup>2</sup> <https://freesound.org/>

## 5.11 PLA DE MÀRQUETING

Com tots els projectes, s'ha de fer un estudi del pla de màrqueting. No obstant, com veurem en aquest apartat, aquest joc, al tenir un públic objectiu molt marcat, veurem que serà una mica diferent als jocs regulars.

### 5.11.1 PLATAFORMES DE DISTRIBUCIÓ I CONSUM

La majoria de videojocs es distribueixen en les principals plataformes de distribució. En el cas de les consoles, podrien ser les botigues físiques o les pròpies de la consola. D'altra banda per PC, les principals plataformes serien les online com són: Steam, Origin, Epic Games, etc. Finalment pels dispositius mòbils i tauletes es distribueixen per Play Store, en cas d'Android, o Apple Store en el cas de Apple.

En el cas del nostre projecte, al ser enfocat per a centres terapèutics, volem que sigui distribuït com a llicències. Per tant, si un centre volgués comprar el nostre videojoc, podrien comprar una llicència i utilitzar-lo en tots els seus dispositius. En el cas que els pares volguessin que el seu fill pogués jugar al nostre joc a casa per practicar ens podrien comprar una llicència per a un sol dispositiu amb un preu més baix. Per tant, el distribuïdor seríem nosaltres mateixos.

### 5.11.2 COMUNICACIÓ

Com hem comentat en la introducció d'aquest apartat, aquest joc al tenir un públic molt concret, no es distribuirà ni s'anunciarà com un joc normal. Aquest joc, en comptes de promocionar-se en revistes de videojocs o fòrums de videojocs, serà promocionat en mitjans que tinguin alguna cosa a veure amb el TEA o relacionats amb la medicina.

A més a més, intentariem posar-nos en contacte amb diferents centres i els presentariem el nostre producte. Un cop entréssim en qualsevol centre, els comentariem que els pares també poden comprar el joc per a practicar des de casa, i d'aquesta manera podríem arribar també a ells.

Per tant, el nostre objectiu principal serà exposar-nos en llocs que tinguin interès el nostre producte i anar a vendre'l als mateixos centres.

## 6. IMPLEMENTACIÓ I PROVES

En aquest apartat explicarem tot el que hem desenvolupat en aquest projecte i els problemes que hem pogut tenir per a resoldre'ls. Tot i que aquest projecte és un prototip, podem veure que la idea principal del projecte s'ha pogut desenvolupar i que la seva utilitat ha estat l'esperada. En aquest prototip hem desenvolupat 7 minijocs diferents amb diferents tasques quotidianes, els menús i les interfícies necessaris per a permetre als terapeutes usar aquest joc com a eina per a practicar les tasques amb els seus pacients.

En les Figures 57 i 58 podem veure l'estructura de les carpetes del Unity .



*Figura 57 Captura estructura carpetes Unity.      Figura 58 Captura estructura carpeta Art.*

Com podem veure a la Figura 57, hem separat les diferents parts del projecte en carpetes. Això ho hem fet per a organitzar millor els elements per a poder accedir més fàcilment a ells. A més a més, dins de la carpeta “Art” hem organitzat els Sprites, pictogrames i fonts de text per a mantenir un ordre més clar.

A la Figura 57 podem veure una carpeta anomenada “Buttons”. En aquesta carpeta hi ha el paquet de Unity baixat d'internet que hem utilitzat per a dissenyar els botons del joc.

## 6.1 IMPLEMENTACIÓ

En aquest apartat explicarem com hem implementat les diferents funcionalitats que hem volgut que tingués el nostre joc. Explicarem com hem implementat els minijocs, menús i interfícies més importants.

### 6.1.1 MENÚ PRINCIPAL (menuManager.cs)

El primer que veurà el jugador a l'hora d'obrir el joc serà el menú principal. Com hem comentat en altres apartats, des d'aquí podrem crear i modificar els calendaris, o accedir al calendari per a poder jugar.

A continuació explicarem les funcions més importants del menuManager.cs:

```
void Start()
{
    if (PlayerPrefs.GetString("psswd") == "")
    {
        pausebuttons();
        passwordCreatorPanel.SetActive(true);
    }
    else
    {
        passwordCreatorPanel.SetActive(false);
    }

    if (PlayerPrefs.GetInt("numberActivities") <= 0 || !PlayerPrefs.HasKey("numberActivities"))
    {
        playButton.SetActive(false);
        calendarButton.GetComponentInChildren<Text>().text = "FER CALENDARI";
    }
}
```

Figura 59 Captura funció start() menuManager.cs

Com podem veure a la Figura 59, en la funció start(), a l'entrar al menú principal hem de mirar si el calendari i la contrasenya estan creats o no. En cas que no existeixi la contrasenya veurem una pantalla que ens la farà crear i, en el cas que el calendari no existeixi, no podrem jugar, ja que és indispensable tenir un calendari per a poder jugar.

```
public void createpassword()
{
    PlayerPrefs.SetString("psswd", password.text);
    PlayerPrefs.Save();
    passwordCreatorPanel.SetActive(false);
    title.SetActive(true);
    calendarButton.SetActive(true);
    exitButton.SetActive(true);
}
```

Figura 60 Captura funció createpassword() menuManager.cs



```

public void checkpassword()
{
    string enteredpassword = accespassword.text;

    if (enteredpassword == PlayerPrefs.GetString("psswd"))
    {
        SceneManager.LoadScene("CalendarMaker");
    }
    else
    {
        accespassword.text = "";
    }
}

```

Figura 61 Captura funció checkpassword() menuManager.cs

Com podem veure a les Figures 60 i 61, en les funcions createpassword() i checkpassword(), gestionem la contrasenya amb els PlayerPrefs. El primer cop que entri el jugador al joc, es cridarà la funció createpassword() i es veurà la pantalla per a crear-la. La contrasenya només es crearà el primer cop que s'obri el joc i els PlayerPrefs guarden les dades entre partides.

Totes les altres funcions que té el menuManager.cs són simplement funcions que mostren o amaguen diferents interfícies o que canvien l'escena segons el botó que premi el jugador.

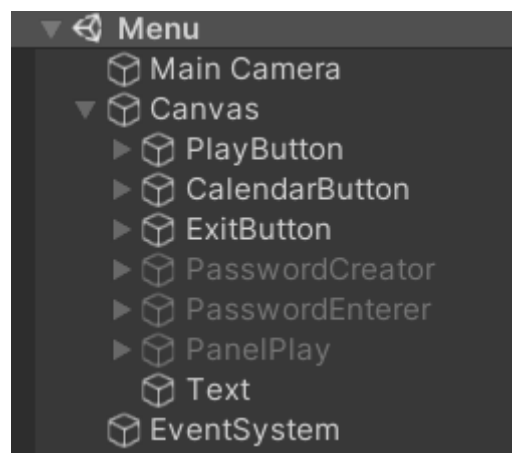


Figura 62 Captura estructura Menú principal

En la Figura 62 podem veure l'estructura del Menú principal. Simplement, hi ha un canvas amb tots els botons i interfícies dissenyades que mostrarem o deixarem de mostrar segon les característiques comentades anteriorment.

### 6.1.2 CREADOR DE CALENDARI (CalendarMaker.cs)

Un cop el jugador hagi creat la contrasenya, si volem que es pugui jugar, el terapeuta o el tutor haurà de crear un calendari amb les tasques que el nen haurà de realitzar.

A continuació explicarem les funcions més importants del CalendarMaker.cs:

```
public List<string> list = new List<string> { "Tria tasca a fer", "Vestir-se", "Rentar-se les dents", "Fer la motxilla",
    "Dutxar-se", "Dinar", "Esmorzar", "Sopar", "Ordenar objectes", "Ordenar habitació" };

void setdropdownoptions()
{
    foreach (Dropdown dropdown in dropdown)
    {
        dropdown.options.RemoveRange(0, dropdown.options.Count);
    }

    foreach (Dropdown drop in dropdown)
    {
        foreach (string option in list)
        {
            drop.options.Add(new Dropdown.OptionData(option));
        }
    }
}
```

Figura 63 Captura funció setdropdownoptions() i list CalendarMaker.cs

A la Figura 63 veiem la funció setdropdownoptions(), emplenem tots els Dropdowns amb el contingut que tenim a la llista "list". Segons el moment del dia, eliminarem o afegirem tasques. Per exemple dinar, esmorzar i sopar només seran visibles en el moment del dia que correspongui.

```
public void next()
{
    foreach (Dropdown drop in dropdown)
    {
        if (drop.options[drop.value].text != "Tria tasca a fer")
        {
            fillPlayerPrefs(drop.options[drop.value].text);
        }
    }
    foreach (Dropdown drop in dropdown)
    {
        drop.value = 0;
    }
    changetime();
}
```

Figura 64 Captura funció next() CalendarMaker.cs

A la Figura 64 podem veure la funció next(). Aquesta serà cridada quan el jugador premi el botó “següent”. El codi llegirà els valors de tots els Dropdowns i si són diferents a “Tria tasca a fer” agafarem la tasca i l’omplirem. A l’acabar reiniciarà els valors del dropdown i canviarem de moment del dia. Com podem veure es crida a una funció anomenada changetime(). Aquesta afegeix a la llista les activitats noves que es poden fer en aquest temps i afegeix el pictograma que mostra el moment del dia en què estem.

```
void fillPlayerPrefs(string content)
{
    if (position == 1)
    {
        PlayerPrefs.SetString("1", content);
    }
    else if (position == 2)
    {
        PlayerPrefs.SetString("2", content);
    }
    else if (position == 3)
    {
        PlayerPrefs.SetString("3", content);
    }
    . . . . .
    PlayerPrefs.Save();
    position++;
}
```

Figura 65 Captura funció fillPlayerPrefs() CalendarMaker.cs

Igual que hem fet en el menú inicial guardarem les activitats en PlayerPrefs. La funció fillPlayerPrefs emplena aquests PlayerPrefs amb les tasques triades pel terapeuta. Aquesta funció es crida un cop hem premut el botó next i tenim els Dropdowns amb alguna tasca posada. Un cop guardada la tasca, guardem el PlayerPrefs i augmentem la posició per a guardar la següent tasca en una nova PlayerPrefs.

Les altres funcions que conté el CalendarMaker.cs serveixen per gestionar quan s’han de veure i quan no els botons o els casos especials com quan acabem el calendari. S’han fet servir els PlayerPrefs ja que es guarden entre partides i aquest és un element que ens interessava. No obstant si ho tornéssim a fer ho faríem amb un json que guardés les dades, cosa que seria més senzilla d’implementar i més correcte.

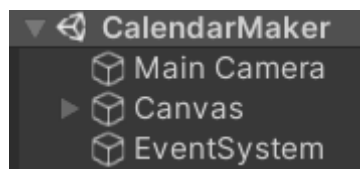


Figura 66 Captura estructura Calendar Maker

A la figura 66 podem veure l’estructura del CalendarMaker. Dintre del Canvas hi ha els Dropdown i els botons que utilitzem.

### 6.1.3 CALENDARI (CalendarManager.cs)

Com hem comentat en altres apartats, el calendari servirà perquè el nen sàpiga quina tasca ha de fer en cada moment i iniciar els minijocs. En l'apartat anterior hem vist que hem guardat les tasques en PlayerPrefs, on el nom era l'ordre de cadascuna d'aquestes. En aquesta secció explicarem com transformarem aquestes dades en un calendari interactiu per als nens.

A continuació explicarem les funcions més importants del CalendarManager.cs:

```

void Start()
{
    numberactivities = PlayerPrefs.GetInt("numberActivities");
    actualact = PlayerPrefs.GetInt("Actualactivity");

    string actualimage = searchImage(actualact+1);

    while (actualimage=="Mati" || actualimage=="Tarda" || actualimage == "Nit") {
        actualact++;
        actualimage = searchImage(actualact + 1);
    }

    PlayerPrefs.SetInt("Actualactivity", actualact);
    PlayerPrefs.Save();

    for (int i = 0; i<actualact; i++){
        Color col = listofButtons[i].GetComponent<Image>().color;
        col.a = 0.25f;
        listofButtons[i].GetComponent<Image>().color = col;
        listofButtons[i].GetComponent<Button>().interactable = false;
        correctsprites[i].SetActive(true);
    }

    for (int j = actualact+1; j <listofButtons.Count; j++) {
        Color col = listofButtons[j].GetComponent<Image>().color;
        col.a = 0.25f;
        listofButtons[j].GetComponent<Image>().color = col;
        listofButtons[j].GetComponent<Button>().interactable = false;
    }

    foreach (Button but in listofButtons)
    {
        if (counter <= numberactivities)
        {
            string imagename = searchImage(counter);
            fillbutton(imagename, but.gameObject);
        }
        if (counter > numberactivities)
        {
            Destroy(but.gameObject);
        }
        counter++;
    }
}

```

Figura 67 Captura funció start() CalendarManager.cs

En la Figura 67, podem veure la funció start() del CalendarManager.cs. En aquesta funció mirem quina activitat és l'actual i baixem l'opacitat i inhabilem els botons que no siguin la posició actual de l'activitat. Els botons anteriors a l'activitat actual ficarem la imatge d'un correcte si són activitats, ja que ja estaran realitzades.

Un cop tractats els botons, recorrerem cada botó i buscarem quina activitat representa, i segons l'activitat que sigui, cridarem la funció fillbutton() per a canviar l'Sprite del botó i assignar que farà cada botó.

```
string searchImage(int counter) {
    string imagename;
    switch (counter) {
        case 1:
            imagename = PlayerPrefs.GetString("1");
            break;
        case 2:
            imagename = PlayerPrefs.GetString("2");
            break;
        . . . . .
    }
    return imagename;
}
```

Figura 68 Captura funció searchImage() CalendarManager.cs

En la Figura 68, podem veure la funció searchImage. Aquí podem veure que aquesta funció recupera el nom de la tasca que hem guardat en els PlayerPrefs quan hem creat el calendari.

```
void fillbutton(string imagename, GameObject but) {
    switch (imagename)
    {
        case "Rentar-se les dents":
            but.GetComponent<Image>().sprite = images[2];
            but.GetComponent<Button>().onClick.AddListener(gototeeth);
            break;
        case "Vestir-se":
            but.GetComponent<Image>().sprite = images[0];
            but.GetComponent<Button>().onClick.AddListener(gotoclothe);
            break;
        case "Fer la motxilla":
            but.GetComponent<Image>().sprite = images[1];
            but.GetComponent<Button>().onClick.AddListener(gotobag);
            break;
        case "Dinar":
            but.GetComponent<Image>().sprite = images[9];
            PlayerPrefs.SetString("FoodTime", "Dinner1");
            PlayerPrefs.Save();
            but.GetComponent<Button>().onClick.AddListener(gotodinar);
            break;
        case "Mati":
            correctsprites[counter - 1].SetActive(false);
            but.GetComponent<RectTransform>().sizeDelta = new Vector2(350, 200);
            but.GetComponent<Image>().sprite = images[3];
            but.GetComponent<Button>().interactable = false;
            break;
    }
}
```

Figura 69 Captura fillbutton() CalendarManager.cs

En la Figura 69, podem veure la funció `fillbutton()`. Aquí podem veure que aquesta funció canvia l'Sprite del botó segons la tasca que li hem assignat. En el cas que el botó no representi una tasca, fem un escalat per a fer el botó més gran. Això ho fem ja que seran els marcadors de canvi de dia i creiem que són molt importants. A més a més, si el botó representa una tasca, afegirem un `AddListener` amb una funció que serà el que farà el joc al prémer el botó. Per exemple si la tasca és fer-se la motxilla, el botó ens portarà a l'escena de fer-se la motxilla.

Hi ha minijocs que tenen característiques diferents, com per exemple en el minijoc de dinar, esmorzar o sopar, que hem de dir quin àpat és i guardar-ho en un altre `PlayerPrefs`, ja que tot i que el minijoc és el mateix, els plats que hi apareixen no.

#### 6.1.4 MINIJOCS

En aquest apartat veurem com hem desenvolupat els diferents minijocs.

##### 6.1.4.1 *MINIJOC FER MOTXILLA*

En el minijoc de fer-se la motxilla, es vol representar que el nen ha de fer-se la motxilla per a anar a l'escola. En aquest cas volem practicar amb el nen quins són els elements que hem de posar a la motxilla per anar a l'escola.

Això ho farem amb una escena en el que hi haurà la motxilla de l'escola del nen i 3 botons que amb pictogrames que representaran diferents objectes. Dos d'aquests objectes no són necessaris per anar a l'escola i el restant si ho és. El nen haurà de clicar sobre el botó que representi l'objecte correcte.

A la Figura 70 podem veure la funció `fillQuestions()`, aquesta funció escollirà quin és l'element correcte que mostrarem cada vegada. A més a més, controla que no repetim objecte correcte i decideix quins elements incorrectes mostrar.

A continuació explicarem les funcions més importants del `QuizManager.cs`.

```

void fillQuestions()
{
    int correct = Random.Range(0, 3);
    int anteriorImage = -1;
    for (int i=0; i<=2; i++)
    {
        Sprite actualImage;
        if (correct == i)
        {
            if (i == 0)
            {
                answer1.GetComponent<questionButton>().setTrue();
            }
            else if (i == 1)
            {
                answer2.GetComponent<questionButton>().setTrue();
            }
            else
            {
                answer3.GetComponent<questionButton>().setTrue();
            }
            int correctImage = Random.Range(0, 4);
            while (correctQuestionsUsed[correctImage] == 1) {
                correctImage = Random.Range(0, 4);
            }
            correctQuestionsUsed[correctImage] = 1;
            actualImage = QuestionsCorrect[correctImage];
        }
        else
        {
            int inCorrectImage = Random.Range(0, 8);
            while (inCorrectImage == anteriorImage) {
                inCorrectImage = Random.Range(0, 8);
            }
            anteriorImage = inCorrectImage;
            actualImage = QuestionsIncorrect[inCorrectImage];
        }
        if (i == 0) {
            answer1.image.sprite = actualImage;
        }
        else if (i==1)
        {
            answer2.image.sprite = actualImage;
        }
        else
        {
            answer3.image.sprite = actualImage;
        }
    }
}

```

Figura 70 Captura fillQuestions() QuizManager.cs

En aquesta funció primer de tot escollirem a quin dels 3 botons anirà l'objecte correcte. Un cop escollit el botó, recorrerem els 3 botons que hi ha a l'escena. Si la posició del botó (assignada a un enter) és la mateixa que on haurà d'anar l'objecte correcte, assignem amb setTrue() que aquest botó serà el correcte. Un cop hem assignat el botó com a correcte, agafem un pictograma d'un objecte que no haguem fet servir i li assignem. En el cas que el botó sigui un incorrecte, li assignarem un pictograma d'un objecte que no hagi de portar a l'escola. A més a més, vigilem que no es repeteixi aquest objecte en l'altre botó incorrecte el mateix objecte. Les altres funcions d'aquest Script controlen el nombre de preguntes que podem fer, el canvi d'escena i la puntuació en cas de superar el minijoc.

A part del manager tenim un Script anomenat questionButton.cs. Aquest Script controlarà els botons que clicarà el jugador. A la figura 71 podem veure la funció que comprova si el botó clicat és el correcte o no.

```

public void itsTrue()
{
    int pause = PlayerPrefs.GetInt("pause");
    if (pause==0)
    {
        if (isTrue)
        {
            SoundManagerScript.PlaySound("correct");
            StartCoroutine(restart());
        }
        else
        {
            if (qzmnger.GetComponent<QuizManager>().puntuacion >= 50)
            {
                qzmnger.GetComponent<QuizManager>().puntuacion -= 10;
            }
            SoundManagerScript.PlaySound("wrong");
            StartCoroutine(fail());
        }
    }
}

```

Figura 71 Captura fillQuestions() QuizManager.cs

La funció itsTrue() comprova si el botó clicat és el correcte o no. Això ho fa mirant una variable booleana que té aquest Script. Aquesta variable l'assignem a true a la funció fillQuestions() que hem vist a la figura 70. Si és el correcte cridem la funció restar, i si és incorrecte ens resta 10 punts de la puntuació.

```

IEnumerator restart()
{
    correct.SetActive(true);
    yield return new WaitForSeconds(0.5f);
    correct.SetActive(false);
    qzmnger.GetComponent<QuizManager>().Restart();
}

```

Figura 72 Captura fillQuestions() QuizManager.cs

Com podem veure a la figura 72, la funció restart() crida una funció del QuizManager anomenada Restart(). Aquesta simplement mira si hem respost totes les preguntes. Si és així mostrem la puntuació i tornem al calendari, altrament tornem a cridar fillQuestions del QuizManager.cs i tornem a començar.



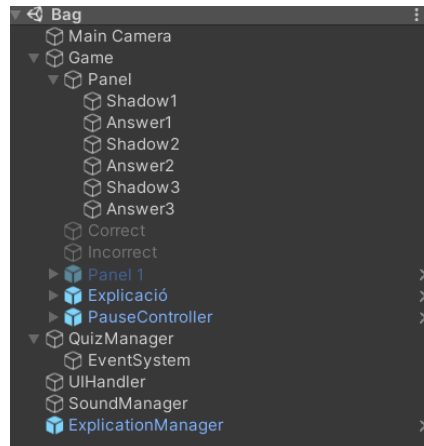


Figura 73 Captura fillQuestions() QuizManager.cs

A la figura 73 podem veure l'estructura de l'escena de la motxilla. Cal destacar els gameObjects Answer1, Answer2 i Answer3 que són els botons que contenen el questionButton.cs i el QuizManager que és qui conté el QuizManager.cs.

#### 6.1.4.2 MINIJOC VESTIR-SE

En el minijoc de vestir-se, el nen haurà de vestir el personatge tenint en compte el temps que es veu a través de la finestra. Per a escollir la peça de roba que el nen vol col·locar haurà de clicar el botó amb el pictograma de les peces de roba. Al clicar apareixerà la roba al centre de l'escena i l'haurà d'arrossegar fins al lloc on ha d'anar la peça de roba.

En el controlador del joc, anomenat ClotheGameController.cs, simplement escollim amb un random quina estació de l'any tindrem i mostra un escenari o un altre. A més a més, serà l'encarregat de controlar la puntuació.

L'Script ChangeClothe.cs controlarà quan volem canviar una peça de roba si ho podem fer o no. A la figura 74 podem veure un exemple.

```

public void changeclothe(string clothe)
{
    if (!final && PlayerPrefs.GetInt("pause") == 0)
    {
        if (interiorput == false)
        {
            interior.gameObject.SetActive(false);
        }
        if (tshirtput == false)
        {
            tshirt.gameObject.SetActive(false);
        }
        if (trouserput == false)
        {
            trouser.gameObject.SetActive(false);
        }
        if (shorttrouserput == false)
        {
            shorttrouser.gameObject.SetActive(false);
        }
        if (shoesput == false)
        {
            shoes.gameObject.SetActive(false);
        }
        if (jerseyput == false)
        {
            jersey.gameObject.SetActive(false);
        }
        switch (clothe)
        {
            case "interior":
                if (interiorput == false)
                {
                    interior.gameObject.SetActive(true);
                    int actual = Random.Range(0, interiorlist.Count);

                    while (actual == anterior)
                    {
                        actual = Random.Range(0, interiorlist.Count);
                    }
                    interior.sprite = interiorlist[actual];
                    anterior = actual;
                }
                break;
        }
    }
}

```

Figura 74 Captura changeclothe() ChangeClothe.cs

La funció changeclothe() és la que es cridarà en el cas que el jugador cliqui sobre qualsevol pictograma de roba. Aquesta funció farà el setActive(false) de totes les peces de roba diferents a la clicada, perquè es vegi tan sols la que ens interessa, i les que ja estiguin col·locades. En el cas que la peça ja estigui seleccionada i el nen cliqui el botó de la peça, canviarà l'Sprite corresponent. Tenim un switch case que farà aparèixer la roba segons el botó clicat.

```

private void Update() {
    if (season == 1 && (trouserput && tshirtput && interiorput && shoesput && jerseyput)) {
        finalbutton.SetActive(true);
    }
    if (season == 2 && (shorttrouserput && tshirtput && interiorput && shoesput))
    {
        finalbutton.SetActive(true);
    }
    if (season == 3 && ((shorttrouserput || trouserput) && tshirtput && interiorput && shoesput))
    {
        finalbutton.SetActive(true);
    }
}

```

Figura 75 Captura update() ChangeClothe.cs

A la figura 75 podem veure la funció Update() de ChangeClothe.cs. Aquesta funció comprovarà si s'han ficat les peces de roba que requereix cada estació. En el cas que sí, es mostrarà un botó per a finalitzar el joc. Fem que el jugador hagi de prémer el botó, ja que pot ficar més peces de roba per a guanyar més punts i ha de ser ell que ha de saber quan parar.

Un cop veiem a l'escena la peça de roba que hem seleccionat, l'Script Drag.cs serà l'encarregat de gestionar com interactuem amb aquesta peça de roba.

```

public void OnPointerUp(PointerEventData eventData)
{
    if(rectTransform.position.x >= objective.GetComponent<RectTransform>().position.x - tolerancia && rectTransform.position.x <= objective.GetComponent<RectTransform>().position.x + tolerancia && rectTransform.position.y >=
    objective.GetComponent<RectTransform>().position.y - tolerancia && rectTransform.position.y <= objective.GetComponent<RectTransform>().position.y + tolerancia && icanfit)
    {
        SoundManagerScript.PlaySound("correct");
        gamecontroller.GetComponent<ClotheGameController>().addScore();
        if (second != null) {
            foreach (GameObject sec in second)
            {
                if ((sec.name == "Trousers" && season == 2) || (sec.name == "ShortTrousers" && season==1))
                {
                    sec.GetComponent<Drag>().icanfit = false;
                }
                else {
                    sec.GetComponent<Drag>().icanfit = true;
                }
            }
        }
        if (anterior != null){
            anterior.SetActive(false);
        }
        if (deleted != null)
        {
            deleted.GetComponent<Drag>().icanfit = false;
        }
        rectTransform.position = objective.GetComponent<RectTransform>().position;

        switch (this.gameObject.name) {
            case "Shirt":
                GameObject.Find("Buttons").GetComponent<ChangeClothe>().tshirtput = true;
                break;
            case "Interior":
                GameObject.Find("Buttons").GetComponent<ChangeClothe>().interiorput = true;
                break;
            case "Trousers":
                GameObject.Find("Buttons").GetComponent<ChangeClothe>().trouserput = true;
                break;
            case "ShortTrousers":
                GameObject.Find("Buttons").GetComponent<ChangeClothe>().shorttrouserput = true;
                break;
            case "Shoes":
                GameObject.Find("Buttons").GetComponent<ChangeClothe>().shoesput = true;
                break;
            case "Jersey":
                GameObject.Find("Buttons").GetComponent<ChangeClothe>().jerseyput = true;
                break;
            default:
                break;
        }
        this.gameObject.GetComponent<Drag>().enabled = false;
    }
    else
    {
        //Debug.Log("not okay");
        rectTransform.position = iniPos;
        SoundManagerScript.PlaySound("wrong");
        gamecontroller.GetComponent<ClotheGameController>().removeScore();
    }
}

```

Figura 75 Captura OnPointerUp() Drag.cs

Com podem veure a la Figura 75, usem el OnPointerUp per a veure si el jugador ha col·locat la peça de roba en el lloc que toca, i si compleix els requisits per a poder ser ficada. En el cas que el jugador hagi col·locat la roba en el collider que toca, la peça es quedarà allà i ja no es podrà moure més, en cas contrari la peça tornarà a la posició inicial.

El Drag.cs té els atributs anterior, second i icanfit. L'atribut anterior és un atribut que mira quin objecte ha d'estar posat per a poder ficar la peça de roba actual, si aquesta és null vol dir que no necessita cap. En el cas del second, és un atribut que indica quina peça de roba es podrà ficar quan l'actual es fiqui. L'atribut icanfit serà true quan la peça de roba es pugui ficar. Un cop la peça de roba està col·locada, per molt que cliquem el pictograma amb la imatge corresponent no passarà res.

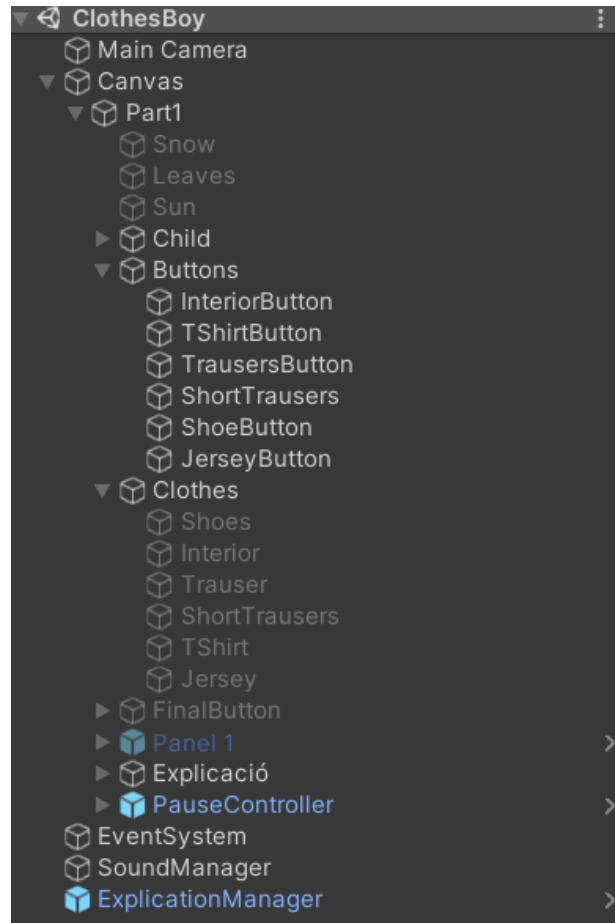


Figura 76 estructura escena ClothesBoy

A la Figura 76 podem veure l'estructura de l'escena ClothesBoy. Podem veure un grup de Botons anomenat "Buttons", allà dintre tindrem els botons que cridaran a la funció que permetrà visualitzar o canviar la peça que representen. A més a més, podem veure un grup anomenat "Clothes". Cada element d'allà dintre tindrà la imatge de l'element que representa i tindran l'Script Drag.cs, ja que seran els objectes a col·locar al nostre personatge.

### 6.1.4.3 MINIJOC ORDENAR HABITACIÓ

En el minijoc d'ordenar l'habitació el nen ha d'identificar quins elements de l'habitació estan desordenats. El nen haurà de fer click sobre els elements que estiguin desordenats i aquests s'ordenaran. El minijoc acabarà quan el nen hagi identificat tots els elements desordenats.

Primer de tot hem d'escollir quins elements seran els que estaran desordenats. Això ho farem en iniciar el joc i ho farà el differenceController.cs.

```
void Start()
{
    while (selected.Count < 4) {
        int number = Random.Range(0, items.Count);
        bool found = selected.Contains(number);
        while (found) {
            number = Random.Range(0, items.Count);
            found = selected.Contains(number);
        }
        selected.Add(number);
    }

    foreach (int obj in selected) {
        items[obj].gameObject.GetComponent<differenceElement>().setWrong();
    }
}
```

Figura 77 Captura Start() differenceController.cs

A la figura 77 podem veure la funció start() del differenceController.cs. Com podem veure, mitjançant números random obtenim els elements a desordenar i els fem a una llista anomenada selected. Per a evitar que hi hagin duplicats, cada vegada que seleccionem un element a desordenar mitjançant el Contains(), mirem si aquest està dintre de la llista.

Un cop tenim la llista amb els 4 elements a desordenar, cridem la funció setWrong() del differenceElement.cs per tal de desordenar-los. L'Script differenceController.cs, també és l'encarregat de controlar quan s'han trobat totes les diferències i la puntuació. L'Script differenceElement.cs el tenen tots els gameObjects que són susceptibles a ser desordenats. És l'encarregat de desordenar i ordenar l'objecte en cas que sigui necessari.

```

public void setWrong() {
    myself.image.sprite = wrong;
    state = 1;
}

0 referencias
public void changeState() {
    if (PlayerPrefs.GetInt("pause") == 0)
    {
        if (state == 1)
        {
            StartCoroutine(success());
            myself.image.sprite = correct;
            state = 0;
        }
        else
        {
            StartCoroutine(fail());
        }
    }
}

```

Figura 78 Captura setWrong() i changeState() differenceElement.cs

A la figura 78 podem veure la funció setWrong() que cridem a l'Start() del differenceController.cs pels elements a desordenar. Aquesta funció fica l'Sprite desordenat de l'element i el fica en estat 1 (desordenat). En canvi, la funció changeState(), és a dir, és la que cridem quan cliquem sobre l'objecte. Si aquest està desordenat s'ordena, es canvia l'estat i s'augmenta el nombre d'elements ordenats. En cas contrari ens haurem equivocat i restarem punts.

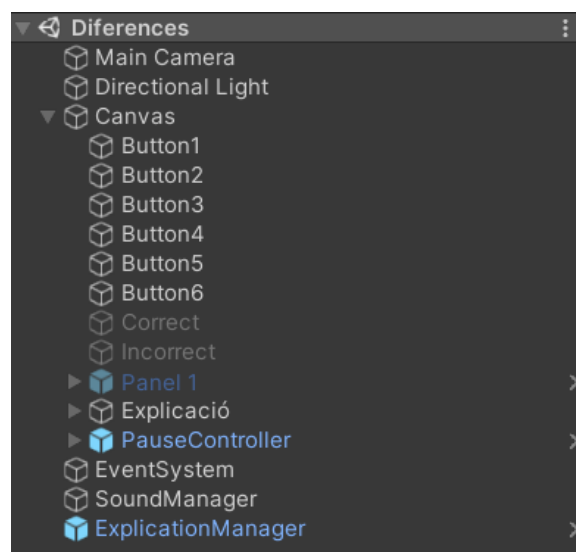


Figura 79 Captura estructura escena Diferences

A la figura 79 podem veure els Botons que seran els que tenen les imatges dels objectes que es poden desordenar. Aquests seran els que tindran l'Script differenceElement.cs. El Canvas serà l'element que tindrà el differenceController.cs.

#### 6.1.4.4 MINIJOC RENTAR DENTS

El minijoc de rentar-se les dents consta de dues parts. La primera part consta d'unes preguntes de verdader i fals que ha de respondre al jugar. L'Script que implementa aquesta part és el `questionsTeethcontroller.cs`.

```
void Start()
{
    createdquestions.Add(Random.Range(0, questions.Count));
    createdquestions.Add(Random.Range(0, questions.Count));
    while (createdquestions[0] == createdquestions[1])
    {
        createdquestions[1] = Random.Range(0, questions.Count);
    }
    createdquestions.Add(Random.Range(0, questions.Count));
    while (createdquestions[0] == createdquestions[2] || createdquestions[1] == createdquestions[2])
    {
        createdquestions[2] = Random.Range(0, questions.Count);
    }
    createdquestions.Add(Random.Range(0, questions.Count));
    while (createdquestions[0] == createdquestions[3] || createdquestions[1] == createdquestions[3]
        || createdquestions[2] == createdquestions[3])
    {
        createdquestions[3] = Random.Range(0, questions.Count);
    }
    actualQuestion.text = questions[createdquestions[actualquestion]];
}
```

Figura 80 Captura `Start()` de `questionsTeethcontroller.cs`

A la Figura 80 podem veure la funció `Start()` de l'Script `questionsTeethcontroller.cs`. Com podem veure, aquesta funció crea una llista de les preguntes que hem triat a l'atzar. Aquestes les guarda en una llista anomenada `createdquestions`. Com podem veure, evitem que estiguin repetides les preguntes, per la qual cosa cada vegada tindrem 4 preguntes diferents entre elles.

```
public void iscorrect()
{
    if (answers[createdquestions[actualquestion]])
    {
        actualquestion++;
        StartCoroutine(fine());
    }
    else
    {
        StartCoroutine(fail());
    }
}

0 referencias
public void iswrong()
{
    if (!answers[createdquestions[actualquestion]])
    {
        actualquestion++;
        StartCoroutine(fine());
    }
    else
    {
        StartCoroutine(fail());
    }
}
```

Figura 81 Captura `Start()` de `questionsTeethcontroller.cs`

A la Figura 81 podem veure la funció `isincorrect()` i `iswrong()`. Aquestes funcions es cridaran segons el botó que cliqui el jugador. Sabrem si hem encertat o no, ja que hi ha una array que, si es té el nombre de la pregunta, retorna si la pregunta és certa o falsa. Si el jugador ha encertat, augmentarem la pregunta en què estem i cridarem `fine()`. Aquesta funció canvia el text de la pregunta actual i així passaríem a la següent pregunta. Si el jugador s'equivoca, simplement cridarem `fail()` que donarà feedback al jugador que ha respost malament i haurà de contestar fins a fer-ho correctament. La resta d'aquest Script gestionarà el canvi de part del joc i saber quan ha respost totes les preguntes correctament.

La segona part consta en rentar les dents. Veurem una boca amb brutícia i el jugador haurà de passar el raspall de dents a la brutícia, que anirà desapareixent a poc a poc, fins que desaparegui completament. Veure Figura 42. L'Script que determinarà la posició de la brutícia serà el `toothcleanercontroller.cs`.

```
void Start()
{
    foreach (GameObject dirttop in TopDirt)
    {
        int x = Random.Range(-343, 352);
        int y = Random.Range(63, 96);
        dirttop.transform.localPosition = new Vector3(x, y, 1);
    }
    foreach (GameObject dirtdown in DownDirt)
    {
        int x = Random.Range(-124, 152);
        int y = Random.Range(-188, -138);
        dirtdown.transform.localPosition = new Vector3(x, y, 1);
    }
}
```

Figura 82 Captura `Start()` de `toothcleanercontroller.cs`

A la Figura 82 podem veure la funció `Start()` de l'Script `toothcleanercontroller.cs`. Aquesta funció agafa dues llistes amb `gameObject` i, mitjançant randoms, els col·locarà a posicions aleatòries. Hem separat en dues llistes per a tenir brutícia tant a les dents de la part superior com a les de la part inferior.

Un cop tenim la brutícia col·locada a l'escena vam implementar com podíem eliminar-la. Vam voler fer que cada vegada que passes el raspall de dents per sobre aquest anés desapareixent fins a finalment desaparèixer del tot. L'Script que controla el raspall de dents és el `DragRemove.cs`.



```

public void OnDrag(PointerEventData eventData)
{
    if (PlayerPrefs.GetInt("pause") == 0)
    {
        transform.position += (Vector3)eventData.delta;

        foreach (GameObject objective in Objects)
        {
            if (rectTransform.position.x >= objective.GetComponent<RectTransform>().position.x - tolerancia && rectTransform.position.x <= objective.GetComponent<RectTransform>().position.x +
                tolerancia && rectTransform.position.y >= objective.GetComponent<RectTransform>().position.y - tolerancia && rectTransform.position.y
                <= objective.GetComponent<RectTransform>().position.y + tolerancia)
            {
                if (objective != null)
                {
                    if (minigame == 1)
                    {
                        objective.GetComponent<DirtShower>().removeLevel();
                    }
                    else if (minigame == 0)
                    {
                        objective.GetComponent<Dirt>().removeLevel();
                    }
                }
            }
        }
    }
}

```

Figura 83 Captura OnDrag() de DragRemove.cs

A la Figura 83, podem veure la funció OnDrag() de DragRemove.cs. En aquesta funció podem veure que, per a cada element d'una llista, que representa la brutícia, mirem si el raspall de dents passa per sobre. En cas que passi per sobre, cridem la funció removeLevel(). Com podem veure hi ha un if amb la variable minigame. Això és degut al fet que aprofitem aquest script en el minijoc de la dutxa, en aquest cas minigame = 0. Per acabar tenim un altre Script que fa que baixi l'opacitat de la brutícia i s'elimini. Aquest Script és el Dirt.cs.

```

public void removeLevel()
{
    if (canDowngrade) {
        if (estat > 0)
        {
            bubbles.SetActive(true);
            estat = estat - 1;
        }
        canDowngrade = false;
        getopacity();
        StartCoroutine(changedowngradestate());
    }
}

```

Figura 84 Captura removeLevel() de Dirt.cs

A la Figura 84, podem veure la funció removeLevel() de Dirt.cs. Aquest Script el tenim a cada gameObject que representa la brutícia i té els atributs estat, que representa el que falta per desaparèixer, i canDowngrade, que és un bool que ens diu si pot baixar l'opacitat o no. Un cop el raspall passa per sobre la brutícia i es crida aquesta funció, apareixen unes bombolles sobre la brutícia i baixa l'opacitat. El StartCoroutine que veiem a la Figura és perquè tinguem un temps en què no baixi l'opacitat de l'objecte, ja que sinó, desapareixeria de cop.

```

public void getopacity()
{
    image = this.GetComponent<Image>();
    var tempColor = image.color;
    if (estat == 4)
    {
        tempColor.a = 1f;
    }
    else if (estat == 3) {
        tempColor.a = 0.75f;
    }
    else if (estat == 2)
    {
        tempColor.a = 0.5f;
    }
    else if (estat == 1)
    {
        tempColor.a = 0.25f;
    }
    else if (estat == 0)
    {
        toothbrush.GetComponent<DragRemove>().addremoved();
        estat = -1;
        tempColor.a = 0f;
    }
    image.color = tempColor;
}
}

```

Figura 85 Captura getopacity() de Dirt.cs

A la Figura 85, podem veure la funció getopacity() de Dirt.cs. La funció removeLevel crida aquesta funció per tal de canviar l'opacitat. Si l'estat arriba a 0, cridem una funció anomenada addremoved() del DragRemove. Aquest, afegeix un element esborrat que, si arriba al nombre de brutícia que tenim a l'inici, finalitzarà el minijoc.

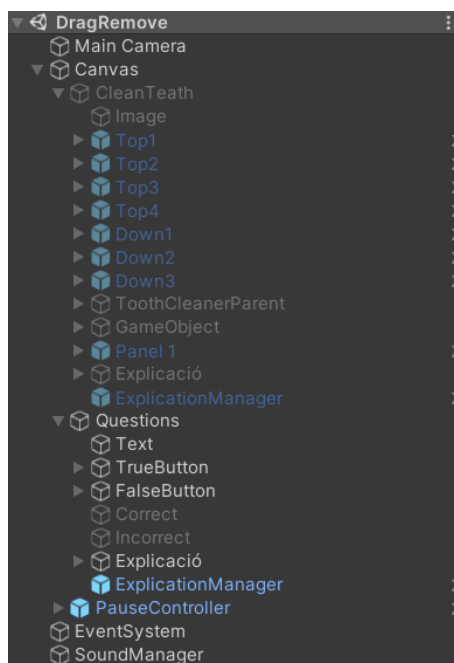


Figura 86 Captura estructura DragRemove

A la Figura 86 podem veure l'estructura del DragRemove. Podem veure els anomenats Top i Down que són la brutícia i contindran l'Script Dirt.cs. També podem veure el ToothCleanerParent que contindrà l'Script toothcleanercontroller.cs. Finalment, dins de Questions tindrem el questionsTeethcontroller.cs.

#### 6.1.4.5 MINIJOC ORDENAR OBJECTES

El minijoc d'ordenar objectes consisteix en arrossegant l'objecte que representa un pictograma fins a la porta de l'habitació a la que hauria d'anar. L'Script que controla quin objecte hem d'ordenar i a quin lloc ha d'anar és el `orderObjectsController.cs`.

```
public void changeobject() {
    actualobject = Random.Range(0, target.Count);
    obj.gameObject.GetComponent<Image>().sprite = objectimage[actualobject];
    obj.gameObject.GetComponent<dragobject>().objective = target[actualobject];
}
```

Figura 87 Captura `changeobject()` `orderObjectsController.cs`

A la figura 87 podem veure la funció `changeobject()` de l'Script `orderObjectsController`. Aquesta funció escull aleatòriament un element de la llista "target". Un cop escollit aquest objecte, li canviem l'imatge pel pictograma que el representa i li assignem quin és l'objectiu. Aquesta funció es crida també al `Start()` per a començar el minijoc.

```
public void feedbackAnswer(int feedback) {
    if (feedback == 0)
    {
        SoundManagerScript.PlaySound("correct");
        correct.SetActive(true);
        wrong.SetActive(false);

        target.RemoveAt(actualobject);
        objectimage.RemoveAt(actualobject);

        StartCoroutine(finishfeedback());
        correctanswers += 1;
        if (correctanswers >= 5)
        {
            StartCoroutine(starspanel());
        }
        else {
            changeobject();
        }
    }
    else {
        SoundManagerScript.PlaySound("wrong");
        wrong.SetActive(true);
        correct.SetActive(false);
        StartCoroutine(finishfeedback());
    }
}
```

Figura 88 Captura `feedbackAnswer()` `orderObjectsController.cs`

A la Figura 88 podem veure la funció `feedbackAnswer()`. Aquesta funció, segons si hem posat l'objecte al lloc correcte, eliminarà de la llista d'objectes l'element per a no repetir-lo i cridarà de nou `changeobject()`. Un cop haguem respost correctament 5 vegades, el minijoc acabarà. En cas de respondre malament haurem de tornar a fer-ho.

A part de l'Script explicat anteriorment n'hi ha un altre que controla l'objecte a col·locar a lloc. Aquest Script és dragobject.cs.

```
public void OnPointerUp(PointerEventData eventData) {
    //Debug.Log(this.gameObject.name + " Was Released.");

    if (rectTransform.position.x >= objective.GetComponent<RectTransform>().position.x - 50 && rectTransform.position.x
        <= objective.GetComponent<RectTransform>().position.x + 50 && rectTransform.position.y >= objective.GetComponent<RectTransform>().position.y - 182 && rectTransform.position.y
        <= objective.GetComponent<RectTransform>().position.y + 182) {
        controller.gameObject.GetComponent<OrderObjectsController>().feedbackAnswer(0);
        rectTransform.position = iniPos;
        SoundManagerScript.PlaySound("correct");
        //Debug.Log("OKEY");
    }
    else {
        controller.gameObject.GetComponent<OrderObjectsController>().feedbackAnswer(1);
        rectTransform.position = iniPos;
    }
}
```

Figura 89 Captura OnPointerUp() dragobject.cs

A la figura 89 podem veure la funció OnPointerUp() de l'Script dragobject.cs. Aquest Script té un atribut que representa l'objectiu on ha d'anar l'objecte. Aquesta funció comproba, un cop deixem anar l'objecte, si l'hem deixat sobre l'element que toca. En cas afirmatiu crida feedbackAnswer() i aquest canviarà d'objecte. En cas contrari, també cridarà feedbackAnswer() però mantindrem el mateix objecte i aquest tornarà a la posició inicial.

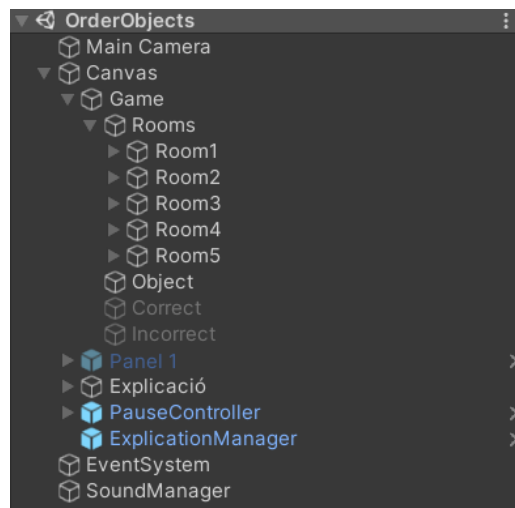


Figura 90 Captura estructura OrderObjects

A la figura 90 podem veure l'estructura del minijoc OrderObjects. Podem veure que hi ha un conjunt de Rooms. Aquestes seran els objectius de l'Object. L'element Object serà l'objecte a col·locar i tindrà l'Script dragobject.cs. El canvas serà el que tindrà l'Script orderObjectsController.cs.

#### 6.1.4.6 MINIJOC MENJAR

El minijoc de menjar consta de dues parts. La primera consisteix en ordenar diferents pictogrames que representen les tasques a fer per a menjar. Com que és molt similar al minijoc d'ordenar objectes, no ho explicarem en aquest apartat. Si es vol veure com s'ha codificat, es podrà veure al Annex. La segona part consta en escollir un plat, que seran diferents segons l'àpat que haguem de realitzar, i parar la taula segons el plat escollit.

L'Script FoodManager.cs serà l'encarregat de mostrar la pantalla que pertoca. Un cop ordenades les tasques, ens mostrarà una pantalla que sortiran diferents menjars segons l'àpat en el que estem. Aquí haurem de prémer sobre el plat que escollim i guardarem els elements que necessitem per aquest plat. L'Script Dish.cs serà el que ens dirà quins plats necessitem.

```
public void foodchoose()
{
    if (PlayerPrefs.GetInt("pause") == 0)
    {
        controller.gameObject.GetComponent<FoodManager>().filldishes(spoon, knife, dish, mug, fork, glass);
        controller.gameObject.GetComponent<FoodManager>().setnumberelements(numberelements);

        controller.gameObject.GetComponent<FoodManager>().setImageObjective(this.gameObject.GetComponent<Image>().sprite);
    }
}
```

Figura 91 Captura foodchosed() Dish.cs

A la Figura 91 podem veure la funció foodchoose() de l'Script Dish.cs. Aquesta funció li diu al FoodManager quins plats necessita pel plat escollit. A la funció filldishes li passem els atributs booleans que representen la vaixela que té cada plat i a setnumberelements() li passem el nombre d'elements que necessitem. Ara l'Script FoodManager.cs tindrem els plats necessaris per a cada aliment.

```
public void adddish() {
    if (!added && PlayerPrefs.GetInt("pasue")==0)
    {
        bool need = gamecontroller.GetComponent<FoodManager>().weneed(arrayposition);
        if (need)
        {
            added = true;
            objective.GetComponent<Image>().sprite = dishon;
            gamecontroller.GetComponent<FoodManager>().adddish();
        }
    }
}
```

Figura 92 Captura adddish() dishadder.cs

A la Figura 92 podem veure la funció adddish() de l'Script dishadder.cs. Aquesta funció la cridarem quan cliquem el botó amb el pictograma d'un plat. La funció cridarà la funció weneed() del FoodManager.cs i ens retornarà si necessitem el plat del botó triat o no. En cas afirmatiu mostrarem el plat a la taula i afegirem el nombre de plats triats. La funció adddish() del FoodManager.cs simplement afegeix 1 a una variable que representa els plats ficats. Si aquests són iguals als plats necessaris, el minijoc finalitzarà.

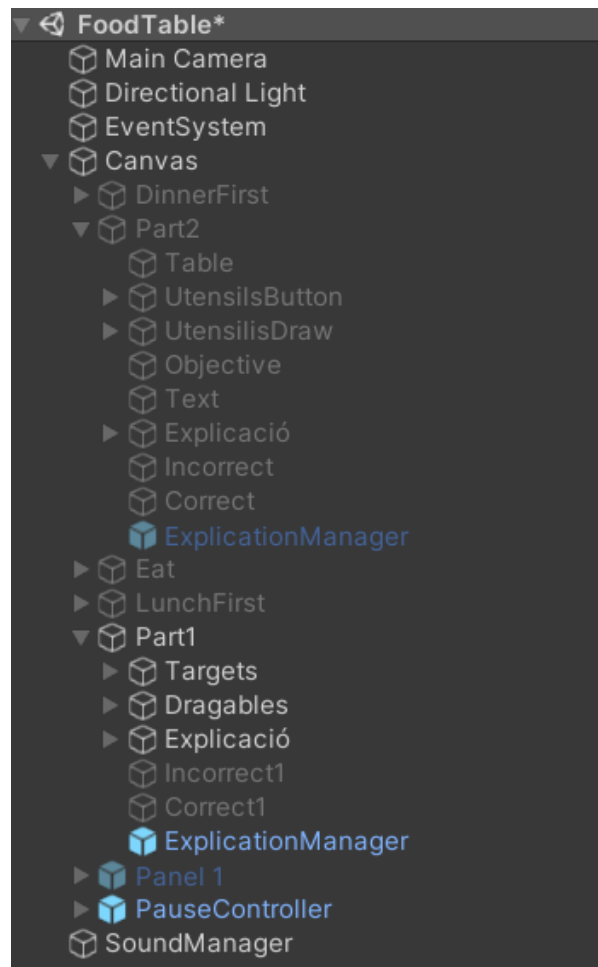


Figura 93 Captura estructura FoodTable

A la Figura 93 podem veure l'estructura del minijoc FoodTable. Podem veure dues parts la Part1, que és la part d'ordenar accions, i la Part2 que és la part de parar taula. També podem veure Dinner, Eat i Lunch. Aquestes pantalles ens mostraran els plats a triar en cada àpat. Finalment, a la part 2 tenim "UtensilisButton" que seran els botons que ens crearan els plats i "UtensilisDraw" que seran els mateixos plats.

#### 6.1.4.7 MINIJOC DUTXAR-SE

El minijoc de dutxar-se consta de dues parts. La primera consisteix en ordenar diferents pictogrames que representen les tasques a fer per a dutxar-se, igual que el minijoc de menjar. Com que és molt similar al minijoc d'ordenar objectes, no ho explicarem en aquest apartat. Si es vol veure com s'ha codificat es podrà veure a l'Annex. La segona part consisteix en treure la brutícia que té pel cos el personatge. A més a més, a diferència del minijoc de rentar-se les dents, a l'ensabonar-se es quedarà el sabó i li haurèm de treure. L'Script ShowerController.cs, serà l'encarregat de controlar si es pot fer servir l'esponja o no, ja que el jugador no podrà usar l'esponja si l'aigua està activada.

```
public void activeshower()
{
    if (!activeSponge)
    {
        if (activeShower)
        {
            Shower.GetComponent<Image>().sprite = showerOff;
            activeShower = false;
        }
        else
        {
            Shower.GetComponent<Image>().sprite = showerOn;
            activeShower = true;
        }
    }
    else {
        StartCoroutine(fail());
    }
}
```

Figura 94 Captura activeshower() ShowerController.cs

A la figura 94 podem veure la funció activeshower() de l'Script ShowerController.cs. Aquesta funció activa la dutxa si no hi ha el sabó activat. A més a més, fica a true la variable que diu si la dutxa esta activa o no. Existeix la mateixa funció per a l'esponja.

Un cop activem l'esponja hem de detectar si aquesta passa per sobre de la brutícia o no. L'encarregat de controlar això és l'Script DragRemove.cs. Aquest Script és el mateix que utilitzem en el minijoc de rentar-nos les dents.

```
public void OnDrag(PointerEventData eventData)
{
    if (PlayerPrefs.GetInt("pause") == 0)
    {
        transform.position += (Vector3)eventData.delta;

        foreach (GameObject objective in Objects)
        {
            if (rectTransform.position.x >= objective.GetComponent<RectTransform>().position.x - tolerancia && rectTransform.position.x <= objective.GetComponent<RectTransform>().position.x + tolerancia && rectTransform.position.y >= objective.GetComponent<RectTransform>().position.y - tolerancia && rectTransform.position.y <= objective.GetComponent<RectTransform>().position.y + tolerancia)
            {
                if (objective != null)
                {
                    if (minigame == 1)
                    {
                        objective.GetComponent<DirtShower>().removeLevel();
                    }
                    else if (minigame == 0)
                    {
                        objective.GetComponent<Dirt>().removeLevel();
                    }
                }
            }
        }
    }
}
```

Figura 95 Captura OnDrag() de DragRemove.cs

A la Figura 95, podem veure la funció OnDrag() de DragRemove.cs. En aquesta funció podem veure que, per cada element d'una llista, que guarda la informació de la brutícia, mirem si el raspall de dents passa per sobre seu. En cas que el raspall passi per sobre la brutícia cridem la funció removeLevel(). Com podem veure, hi ha un if amb la variable minigame. Això és a causa que aprofitem aquest script en el minijoc de rentar-se les dents, en aquest cas minigame = 1. Per acabar tenim un altre Script que fa que baixi l'opacitat de la brutícia i s'elimini. Aquest Script és el Dirt.cs.

```
public void removeLevel()
{
    if (canDowngrade) {
        if (estat > 0)
        {
            bubbles.SetActive(true);
            estat = estat - 1;
        }
        canDowngrade = false;
        getopacity();
        StartCoroutine(changedowngradestate());
    }
}
```

Figura 96 Captura removeLevel() de dirtshower.cs

A la Figura 96, podem veure la funció removeLevel() de dirtshower.cs. Aquest Script el tenim a cada gameObject que representa la brutícia i té els atributs estat, que representa el que falta per desaparèixer, i canDowngrade, que és un bool que ens diu si pot baixar l'opacitat o no. Un cop el raspall passa per sobre la brutícia i es crida aquesta funció, apareixen unes bombolles sobre la brutícia i baixa l'opacitat. El StartCoroutine que veiem a la Figura és perquè tinguem un temps en què no baixi l'opacitat del objecte ja que si no, desapareixeria de cop.

```
public void getopacity()
{
    image = this.GetComponent<Image>();
    var tempColor = image.color;
    if (estat == 4)
    {
        tempColor.a = 1f;
    }
    else if (estat == 3) {
        tempColor.a = 0.75f;
    }
    else if (estat == 2)
    {
        tempColor.a = 0.5f;
    }
    else if (estat == 1)
    {
        tempColor.a = 0.25f;
    }
    else if (estat == 0)
    {
        toothbrush.GetComponent<DragRemove>().addremoved();
        estat = -1;
        tempColor.a = 0f;
    }
    image.color = tempColor;
}
```

Figura 97 Captura getopacity() de dirtshower.cs



A la Figura 97, podem veure la funció `getopacity()` de `dirtshower.cs`. La funció `removeLevel` crida aquesta funció per tal de canviar l'opacitat. Si l'estat arriba a 0 cridem una funció anomenada `addremoved()` del `DragRemove`. Les funcions de les Figures 96 i 97 són les mateixes que vam fer servir en el minijoc de rentar-nos les dents.

```
public void removebubblelevel()
{
    if (estat <= -1)
    {
        bubbles.SetActive(true);
        if (canDowngrade && controller.GetComponent<ShowerController>().isactiveshower())
        {
            estat = estat - 1;
            canDowngrade = false;
            getbubbleopacity();
            StartCoroutine(changedowngradestatebubble());
        }
    }
}
```

Figura 98 Captura `removebubblelevel()` de `dirtshower.cs`

A la Figura 98 podem veure la funció `removebubblelevel()` de l'Script `dirtshower.cs`. Aquesta funció és la que fa el minijoc diferent que el de rentar-se les dents. Un cop eliminada la brutícia, sortirà sabó on hi havia la brutícia. Aquesta funció es cridarà si tenim l'aigua de la dutxa oberta. El sabó anirà desapareixent amb una funció igual que el `getopacity` de la Figura 97. El minijoc acabarà quan no hi hagi ni brutícia ni sabó.

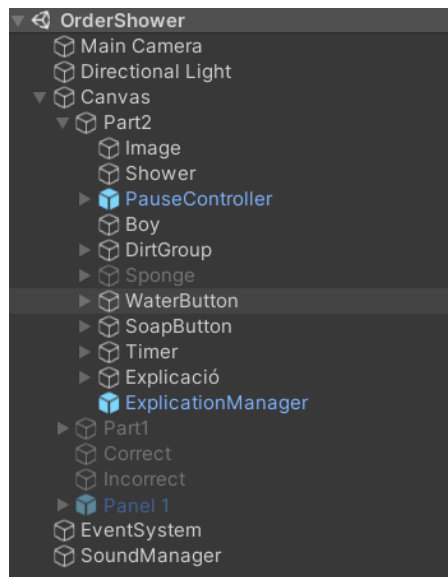


Figura 99 Captura estructura minijoc `OrderShower`

A la Figura 99 podem veure l'estructura del minijoc de dutxar-se. Podem veure que hi ha dues parts. La d'ordenar accions i la de dutxar-se. Dintre de la `Part2` podem veure "DirtGroup" allà hi hauran tots els objectes que tenen l'Script `dirtshower.cs` i actuen com a brutícia. A més a més, veiem "Sponge" que tindrà l'Script `DragRemove.cs`.

## 6.2 PROVES

A mesura que anàvem desenvolupant el projecte, anàvem provant tots els minijocs. Primer hi jugaven els terapeutes i un cop refeiem les coses que consideraven necessàries passàvem a provar-ho amb els pacients. Hi havia vegades que, quan es testejava amb els pacients, ens donàvem compte que, coses que poder per nosaltres eren molt clares, per a ells no ho eren i ho havíem de refer.

Un cop finalitzat el projecte vam fer proves amb el joc complet a l'entorn pel qual estava pensat. Primer de tot vam mirar de que fes el calendari el propi terapeuta i després jugues el nen. El resultat va ser molt bo, ja que la majoria de nens gaudien i els servia per practicar les tasques diàries. Mentre testejàvem, a la Sra. Anna Valls se li va ocórrer provar de muntar el calendari amb els nens, i vam veure que això també anava bé per acabar de repassar amb el nen l'ordre de les tasques que ha de fer.

Finalment, tant nosaltres com la Sra. Anna creiem que les proves van resultar satisfactòries i que seria viable treure al mercat aquest joc i a usar-lo com a eina de treball pels terapeutes.

## 7. RESULTATS

En aquest apartat es comprovarà si s'han complert els objectius que es van marcar en un principi i de quina manera ho hem fet.

Recordem que l'objectiu principal del projecte era el de desenvolupar un joc que ajudés als nens amb trastorn de l'espectre autista a interioritzar les tasques que han de realitzar cada dia i de quina manera les han de realitzar. A més a més, un altre dels objectius principals era que aquest joc servis com a eina per als terapeutes.

Per a aquests objectius es van marcar uns objectius concrets:

- Desenvolupament d'un videojoc mòbil per a nens amb Trastorn de l'Espectre de l'Autisme.

Com hem dit en altres apartats, la idea principal al començar a desenvolupar aquest videojoc era que estigues enfocat a adolescents. No obstant, a l'inici del projecte i parlant amb la terapeuta, vam decidir canviar el target a nens. Això ho vam fer ja que en aquells moments era amb el grup que podíem anar provant els nostres prototips i el grup que els terapeutes creien que rebrien més bé el videojoc.

- Estudiar els diferents aspectes del TEA, amb l'assessorament de terapeutes.

Com hem indicat en altres apartats, hem tingut un contacte constant amb la Sra. Anna Valls. A l'inici del projecte, ens va explicar les característiques d'aquest trastorn fins que vam comprendre els problemes que poden tenir al seu dia a dia. A part de les seves explicacions, es van fer lectures de diferents estudis i articles\*.

- Estudiar com els terapeutes interaccionen amb ells per a fer el joc més fàcil.

Tot i que al moment de dir els objectius que havíem de complir ja sabíem de la seva existència, vam veure que la forma més eficaç d'interactuar amb els nens era mitjançant els pictogrames. Aquests pictogrames són molt senzills i els permet identificar de manera fàcil i ràpida els elements o accions que el terapeuta vol explicar al nen. A més a més, el calendari que s'ha realitzat per a entrar als minijocs, és el mateix tipus de calendari que usen aquests nens amb els terapeutes. A la Figura 100 podem veure un calendari com el que utilitzen els nens.



Figura 100 Imatge que mostra un calendari usat per un nen amb TEA

\* Referència bibliografia: Allà trobarem els diferents articles que hem consultat

- Desenvolupar un formulari que ens indiqui com és el dia a dia del jugador.

Tot i que la idea inicial era realitzar un petit formulari que crees un horari fix pels nens, al final vam desenvolupar un altre formulari que es podia entrar directament les activitats a realitzar. Vam veure necessari posar una contrasenya per a realitzar el formulari, ja que si no el jugador podria canviar les accions i no fer les activitats que el terapeuta creu que són necessàries per ell.

- Fer recerca d'un conjunt de pictogrames que siguin comprensibles pel grup desitjat.

Quan vam començar el projecte, la Sra. Anna ens va facilitar una gran font de pictogrames, allà vam escollir els que vam creure necessaris per al nostre projecte. Com que realitzàvem el projecte per a provar-ho amb el centre "Tots hi som", els vam preguntar amb quin idioma ho volien i ens van dir que en català. Com que tots els pictogrames estaven en castellà els vam demanar permís per a traduir-los i ens el van concedir. Com que els pictogrames són específics pel nostre públic target, sí que són comprensibles per ells.

- Desenvolupar un joc que sigui comprensible pel grup desitjat, amb especial èmfasi en les mecàniques que poden realitzar cada grup de nens amb TEA.

Cada vegada que realitzàvem un minijoc li fèiem una petita demo a la terapeuta. Ella s'encarregava de provar-lo, tant ella com amb pacients, per a decidir si era comprensible o no per a ells. Com que utilitzàvem un mètode iteratiu, quan ella trobava alguna cosa a millorar m'ho comentava i s'arreglava fins a arribar al resultat desitjat. Un cop acabat el projecte, vam provar el joc final amb un grup del centre i els resultats van ser satisfactoris.

- Intentar treballar encara que sigui una mica l'art del videojoc.

En aquest projecte l'autor ha dissenyat els escenaris i les interfícies d'aquest. No obstant, si continuéssim amb aquest projecte l'art seria 100% de l'autor.

- Finalitzar i arrodonir la documentació del projecte.

Tot i que s'ha anat realitzant parts de la memòria mentre es duia a terme el projecte, la major part s'ha fet al finalitzar el joc, ja que s'ha acabat amb prou antelació per poder dedicar temps a la memòria del projecte.

A continuació mostrarem les captures finals de tots els minijocs en acció:

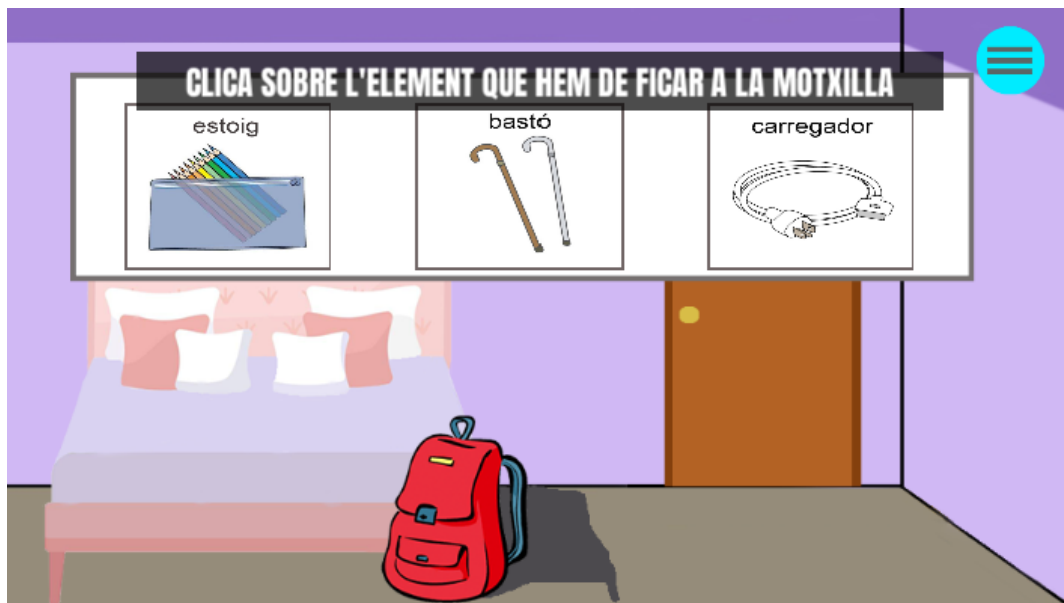


Figura 101 Captura minijoc Ordenar motxilla

L'objectiu del minijoc que veiem a la Figura 101, és que el nen interioritzi quins elements necessita per anar a l'escola.

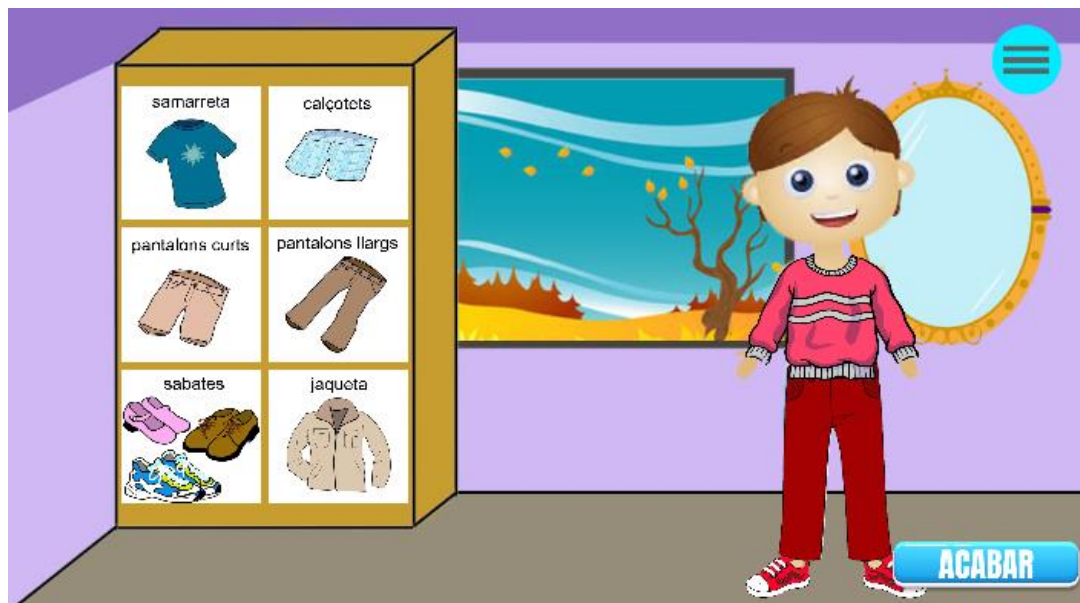


Figura 102 Captura minijoc Vestir-se

L'objectiu del minijoc que veiem a la Figura 102, és que el nen reconegui quines peces de roba s'ha de posar i diferenciar segons l'època de l'any.



Figura 103 Captura minijoc Ordenar habitació

L'objectiu del minijoc que veiem a la Figura 103, és que el nen sigui capaç de reconèixer quins elements estan desordenats i quins no.



Figura 104 Captura part 1 minijoc rentar dents

L'objectiu de la part 1 del minijoc que veiem a la Figura 104, és que el nen interioritzi els aspectes més bàsics relacionats amb la higiene bucal.



Figura 105 Captura part 2 minijoc rentar dents

L'objectiu de la part 2 del minijoc que veiem a la Figura 105, és que el nen interioritzi que és molt important rentar-se les dents, i que ho ha de fer fins a tenir-les ben netes.



Figura 106 Captura minijoc parar taula

L'objectiu del minijoc que veiem a la Figura 106, és que el nen sigui de poder decidir quina coberteria necessita per a menjar el plat que ell ha escollit.



Figura 107 Captura minijoc Ordenar objectes

L'objectiu del minijoc que veiem a la Figura 107, és que el nen interioritzi a quina part de la casa ha d'anar cada element.



Figura 108 Captura minijoc Dutxar-se

L'objectiu del minijoc que veiem a la Figura 108, és que el nen aprengui que s'ha d'apagar l'aigua mentre t'ensabones, i que no pots deixar ni sabó ni brutícia a l'hora de dutxar-se.

Com podem veure a les Figures des de la 101 la 108, el resultat del joc és bastant polit. Tot i les limitacions que hem tingut a l'hora del disseny artístic del joc, creiem que hem obtingut un resultat bastant atractiu, més tenint en compte el nostre públic objectiu.



## 8. CONCLUSIONS

Amb aquest treball hem pogut posar a pràctica gran part de les coses que s'han après durant el Grau de Disseny i Desenvolupament de Videojocs. No obstant, com en tots els projectes sempre s'acaben aprenent coses noves.

En el moment en el qual havíem de decidir que faríem de TFG, nosaltres ja teníem molt clar que havíem de fer un joc per nens amb TEA. Això és a causa d'una conversa que vam tenir amb la nostra tutora, la Sra. Anna. Ella ens va comentar que hi havia molt poques aplicacions i jocs adequats per als seus pacients i que podríem desenvolupar alguna cosa que els hi fos útil. Amb les seves paraules vam veure clar que el nostre joc havia de tenir un objectiu més enllà que el d'entretenir als jugadors i li vam proposar dissenyar un joc amb les directrius que ella marqués, i va acceptar.

Vam començar a reunir-nos per a acabar d'enfocar que era ben bé el que volíem desenvolupar i finalment, en una reunió, vam arribar a la conclusió que el més important era que els nens poguessin ser mínimament independents. En aquest moment vam decidir que faríem un joc amb diferents minijocs que representarien les tasques de la casa. D'aquesta manera i a base de repeticions, que és com més aprenen els nens amb TEA, interioritzarem el que ha de fer cada nen cada dia.

Tot i que no s'ha pogut realitzar tot el que es volia fer, i amb alguns aspectes a millorar, aquest projecte m'ha sigut molt reconfortant, ja que hem vist amb els nostres propis ulls pacients fent-lo servir i veient que veritablement podia resultar útil.

Creiem que tot i la limitació que tenim pel tipus de jugador i els controls que podem tenir en un mòbil o una tauleta, s'han pogut desenvolupar diferents mecàniques bastant diferenciades entre elles i que, tot i que el nen hagi de realitzar el mateix moviment repetidament, acabi treballant diferents coses.

Tot i que les mecàniques d'aquest projecte són relativament senzilles, hi ha hagut diferents problemes a l'hora de dur-les a terme i adaptar-les en el que volíem fer, o per a reaprofitar algunes parts del codi. Com a crítica personal podríem haver fet el codi una mica més aprofitable, al llarg del projecte hem anat rectificat en aquest aspecte.

En el prototip presentat es poden veure complerts tots els requisits previs que es van presentar al començar el projecte. A més a més, s'han desenvolupat les tasques més importants del dia a dia dels nens, que era un objectiu que volíem aconseguir.

Per acabar, podem dir que el fet d'acabar aquest projecte i veure com els nens amb TEA han pogut jugar i l'han pogut comprendre demostra que hem assolit el nostre objectiu final.

## 9. TREBALL FUTUR

En aquest capítol parlarem dels nous apartats o característiques que voldríem incorporar en el projecte en cas de continuar amb el desenvolupament del joc.

- Incorporar noves tasques diàries que puguin realitzar els nens tenint en compte les seves característiques.
- Canviar tot l'art que no hagi estat dissenyat per l'autor, per altres propis.
- Fer unes petites pantalles de càrrega amb un petit vídeo explicatiu de què s'ha de realitzar a cada tasca. D'aquesta manera podríem fer que nens que no entenguessin algun joc, no haguessin de preguntar a la pedagoga que s'ha de fer.
- Aplicar l'Script, ja desenvolupat, que tradueix a diferents idiomes el que nosaltres demanem, i traduir el joc almenys al castellà i a l'anglès.
- Fer que el personatge pugui ser tant noi com noia, i que pugui ser mínimament personalitzable per a que el nen se senti més identificat, tot i que és cert que el personatge es veu en molt pocs casos.

## 10. BIBLIOGRAFIA

1. Rosa Calvo. (2018). *Trastorn de l'Espectre Autista*. (Consulta 25 de novembre de 2021) <https://www.clinicbarcelona.org/ca/asistencia/malalties/trastorn-de-lespectre-autista>
2. KartInnka. (2022). *Buttons Set*. KartInnka. (Consulta 20 abril 2022) <https://assetstore.unity.com/packages/2d/gui/buttons-set-211824>
3. Fundación Orange. (2022). *Dicta Picto*. (Consulta 15 desembre 2021) <https://www.fundacionorange.es/aplicaciones/dictapicto-tea/>
4. Fundación Orange. (2022). *Día a Día*. (Consulta 15 desembre 2021) <https://www.fundacionorange.es/aplicaciones/dia-a-dia/>
5. Tots hi som. (2020). *Tots hi som web*. (Consulta 20 desembre 2021) <https://www.totshisom.cat/>
6. Unity technologies.(2021). *Unity Manual*. (Consulta durant tot el projecte) <https://docs.unity3d.com/Manual/index.html>
7. Unity technologies. (2020). *Unity Answers*. (Consulta durant tot el projecte) <https://answers.unity.com/>
8. Stack Exchange, Inc. (2022). *Stack Overflow*. (Consulta durant tot el projecte) <https://stackoverflow.com/>
9. Brackeys. (2017). *Canal de youtube "Brackeys"*. (Consulta durant tot el projecte) <https://www.youtube.com/brackeys>
10. Universidad Internacional de Valencia. (2015). *Recopilación de artículos sobre el espectro autista*. (Consulta 22 desembre 2021) <https://www.universidadviu.com/es/actualidad/nuestros-expertos/recopilacion-de-articulos-sobre-el-espectro-autista>
11. Anna Valls. *Diverses explicacions sobre la malaltia i les seves afectacions*. (Consultes al llarg del projecte)

## 11. ANNEXOS

Com a Annex, hem adjuntat el projecte sencer de Unity. Creiem que tot i que ja s'han explicat els elements més importants dels Scripts desenvolupats, pot ser interessant donar una ullada a tot el contingut.

Dintre de la carpeta del projecte podem trobar tots els Scripts creats i els assets que hem fet servir. Per accedir als Scripts hem d'anar a "TFG\Assets\Script". Per accedir a tots els Sprites i Pictogrames fets servir hem d'anar a "TFG\Assets\Art". No obstant, volem adjuntar en aquest annex els Scripts dels Prefabs que no hem comentat a l'apartat d'implementació. També hem adjuntat un vídeo en el que es mostra un exemple d'una partida completa del nostre projecte.

A part d'aquest vídeo s'han fet altres vídeos amb nens amb TEA jugant al prototip. No obstant, tot i que tenim el permís dels pares per a fer aquests vídeos, no els hem adjuntat a l'Annex per evitar problemes.

### 11.1 CODI EXPLICACIÓ

Cada vegada que entrem en un minijoc sortirà un petit text amb un fons negre que explicarà al nen el que ha de fer a cada activitat. Aquest és el codi.

```
public class explicacion: MonoBehaviour
{
    public GameObject expl;

    IEnumerator endexp()
    {
        yield return new WaitForSeconds(10f);
        expl.SetActive(false);
    }

    // Start is called before the first frame update
    void Start()
    {
        StartCoroutine(endexp());
    }
}
```

## 11.2 CODI EXPLICACIÓ SONOR

Cada vegada que entrem en un minijoc es narrarà l'explicació que veiem per pantalla. Aquest és el codi.

```
public class soundExplication: MonoBehaviour
{
    public static AudioClip bagSound, dressSound, diferenceSound,
    cleanteethSound, showerSound, orderSound, orderEatSound, ordenShower, TableSound,
    QuestionsMouthSound;
    static AudioSource audioSrc;

    public string sound;
    // Start is called before the first frame update
    void Start()
    {
        bagSound = Resources.Load<AudioClip>("Motxilla");
        dressSound = Resources.Load<AudioClip>("VestirNen");
        cleanteethSound = Resources.Load<AudioClip>("RentarDents");
        showerSound = Resources.Load<AudioClip>("Dutxa");
        orderSound = Resources.Load<AudioClip>("Ordenar");
        orderEatSound = Resources.Load<AudioClip>("OrdenMenjar");
        ordenShower = Resources.Load<AudioClip>("OrdenDutxa");
        TableSound = Resources.Load<AudioClip>("PararTaula");
        diferenceSound = Resources.Load<AudioClip>("Diferences");
        QuestionsMouthSound = Resources.Load<AudioClip>("PreguntesBoca");

        audioSrc = GetComponent<AudioSource>();
        playsound(sound);
    }

    // Update is called once per frame
    public static void playsound(string clip)
    {
        switch (clip)
        {
            case "bag":
                audioSrc.PlayOneShot(bagSound);
                break;
            case "vestir":
                audioSrc.PlayOneShot(dressSound);
                break;
            case "diference":
                audioSrc.PlayOneShot(diferenceSound);
                break;
            case "rentarDents":
                audioSrc.PlayOneShot(cleanteethSound);
                break;
            case "dutxa":
                audioSrc.PlayOneShot(showerSound);
                break;
            case "ordenar":
                audioSrc.PlayOneShot(orderSound);
                break;
            case "ordMenjar":
                audioSrc.PlayOneShot(orderEatSound);
                break;
        }
    }
}
```

```

        break;
    case "ordDutxa":
        audioSrc.PlayOneShot(ordenShower);
        break;
    case "pararTaula":
        audioSrc.PlayOneShot(TableSound);
        break;
    case "preguntesBoca":
        audioSrc.PlayOneShot(QuestionsMouthSound);
        break;
    }
}
}

```

### 11.3 CODI PUNTUACIÓ

Un cop acabem cada minijoc sortirà un panell amb la puntuació de cada minijoc. Aquest és el codi.

```

public class scorePanelController : MonoBehaviour
{
    public GameObject Star1;
    public GameObject Star2;
    public GameObject Star3;

    public void backtomenu()
    {
        int actualact = PlayerPrefs.GetInt("Actualactivity");
        actualact++;
        PlayerPrefs.SetInt("Actualactivity", actualact);
        PlayerPrefs.Save();
        SceneManager.LoadScene("DailyCalendar");
    }

    public void setscore(int score) {

        print("He entrat aqui amb score:");
        print(score);

        if (score < 50)
        {
            Star1.SetActive(true);
            Star2.SetActive(false);
            Star3.SetActive(false);
        }

        else if (score >= 50 && score<=75) {
            Star1.SetActive(true);
            Star2.SetActive(true);
            Star3.SetActive(false);
        }

        else if (score >= 75)
        {
            Star1.SetActive(true);

```

```
        Star2.SetActive(true);  
        Star3.SetActive(true);  
    }  
}
```



*Figura 109 Imatge que mostra com és gràficament el prefab de la puntuació*

A la figura 109 podem veure com és gràficament el menú de pausa del joc.

### 11.3 CODI MENÚ DE PAUSA

El nostre joc té un menú de pausa, aquest és el codi que el gestiona.

```
public class pauseController: MonoBehaviour
{

    public GameObject panelPause;
    public GameObject buttonPause;

    string actualscene;
    string actualfoodtime;

    // Start is called before the first frame update
    void Start()
    {
        actualscene = SceneManager.GetActiveScene().name;
        actualfoodtime = PlayerPrefs.GetString("FoodTime");
    }

    // Update is called once per frame
    void Update()
    {

    }

    public void startpause() {
        PlayerPrefs.SetInt("pause", 1);
        PlayerPrefs.Save();
        panelPause.SetActive(true);
        buttonPause.SetActive(false);
    }

    public void resume() {
        PlayerPrefs.SetInt("pause", 0);
        PlayerPrefs.Save();
        panelPause.SetActive(false);
        buttonPause.SetActive(true);
    }

    public void restartminigame()
    {
        PlayerPrefs.SetInt("pause", 0);
        PlayerPrefs.Save();
        SceneManager.LoadScene(actualscene);
        PlayerPrefs.SetString("FoodTime", actualfoodtime);
        PlayerPrefs.Save();
    }

    public void backtomenu()
    {
        PlayerPrefs.SetInt("pause", 0);
        PlayerPrefs.Save();
        SceneManager.LoadScene("Menu");
    }
}
```



## 11.4 CODI TEMPORITZADOR

Aquest codi es el que fem servir a tots els rellotges que fem servir en el joc.

```

public class timer : MonoBehaviour
{
    public Text Uitext;
    private int segons = 0;
    private int minuts = 0;

    bool timeractive = true;
    private void Awake()
    {
        InvokeRepeating("Cronometro", 0f, 1f);
    }

    public void stoptimer()
    {
        timeractive = false;
    }

    public int getScore()
    {
        return (minuts * 10) + segons;
    }
    void Cronometro()
    {
        if (timeractive && PlayerPrefs.GetInt("pause")==0)
        {
            if (segons == 60)
            {
                segons = 0;
                minuts++;
            }
            if (minuts < 5)
            {
                segons++;
                if (segons >= 10)
                {
                    Uitext.text = "0" + minuts.ToString() + ":" +
segons.ToString();
                }
                else
                {
                    Uitext.text = "0" + minuts.ToString() + ":0" +
segons.ToString();
                }
            }
        }
    }
}

```

## 12. MANUAL D'INSTAL·LACIÓ I D'USUARI

### 12.1 INSTAL·LACIÓ ANDROID

En aquest apartat explicarem com instal·lar el prototip que hem desenvolupat en un dispositiu Android, que és el pensat per al joc final.

1. El primer que hem de fer és descarregar l'APK.
2. Un cop descarregat aquest arxiu, hem de clicar a l'arxiu per a instal·lar-lo en el dispositiu.

### 12.2 INSTAL·LACIÓ WINDOWS

En aquest apartat explicarem com instal·lar el prototip que hem desenvolupat en un dispositiu Windows (Per si es vol provar allà).

1. El primer que hem de fer és descarregar l'arxiu comprimit on hi ha el prototip.

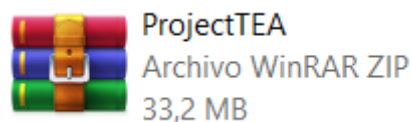


Figura 110 Arxiu comprimir del projecte

2. Un cop descarregat aquest arxiu, hem de fer clic dret a sobre i buscar l'opció de descomprimir.

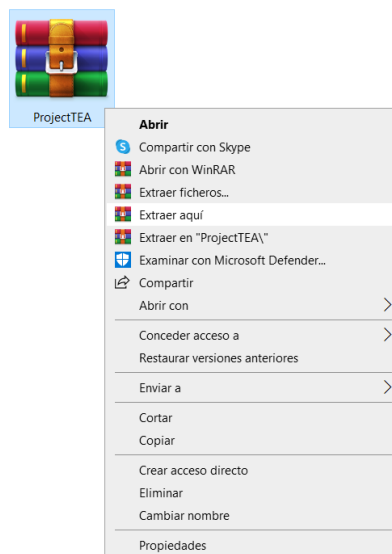


Figura 111 Captura descomprimir arxiu

3. Finalment, entrarem a la carpeta extreta i allà trobarem l'executable anomenat LaMevaRutina.exe. Al clicar dos cops se'ns obrirà el joc.

LaMevaRutina_Data	21/05/2022 19:32	Carpeta de archivos	
MonoBleedingEdge	21/05/2022 19:32	Carpeta de archivos	
LaMevaRutina	08/12/2021 23:59	Aplicación	639 KB
UnityCrashHandler64	09/12/2021 0:00	Aplicación	1.204 KB
UnityPlayer.dll	09/12/2021 0:00	Extensión de la ap...	27.567 KB

*Figura 112 Captura executable del joc*

### 12.3 MANUAL D'USUARI

A continuació adjuntem un petit manual d'usuari mostrant com funciona el prototip i ensenyant les tasques que es poden dur a terme.

Un cop executem el joc, apareixerà una pantalla amb el logotip de Unity, ja que sortirà sempre si tenim la versió gratuïta d'aquest motor de joc. Podem veure-ho a la Figura 113.



*Figura 113 Captura pantalla amb logotip del Unity*

Un cop acabi aquesta imatge veurem el menú principal del joc. Si no hem creat cap contrasenya, l'haurem de crear. Si no hem creat cap calendari, també l'haurem de crear. Des del menú principal podem crear un nou calendari o mostrar el calendari creat, veure Figura 114.



Figura 114 Captura Menú principal

Si hem de crear un calendari, haurem d'escollir entre les diferents tasques que ens surten als desplegable. Podrem fer 4 tasques per moment del dia. Aquests moments seran matí, tarda i nit. Veure la Figura 115.

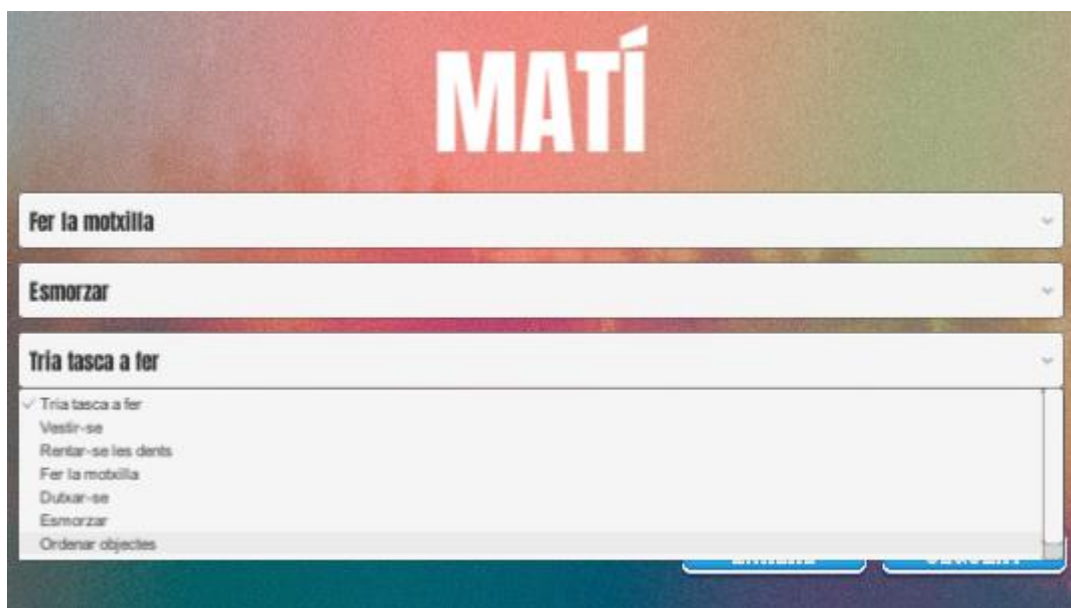


Figura 115 Captura Creador Calendari

Un cop hem creat aquest calendari o hem clicat directament a començar dia, veurem el calendari que ha creat el terapeuta o el tutor. Veure Figura 116.



Figura X Captura Creador Calendari

Un cop hem clicat el pictograma de la tasca anirem al minijoc que hem clicat. Veure Figura 116.

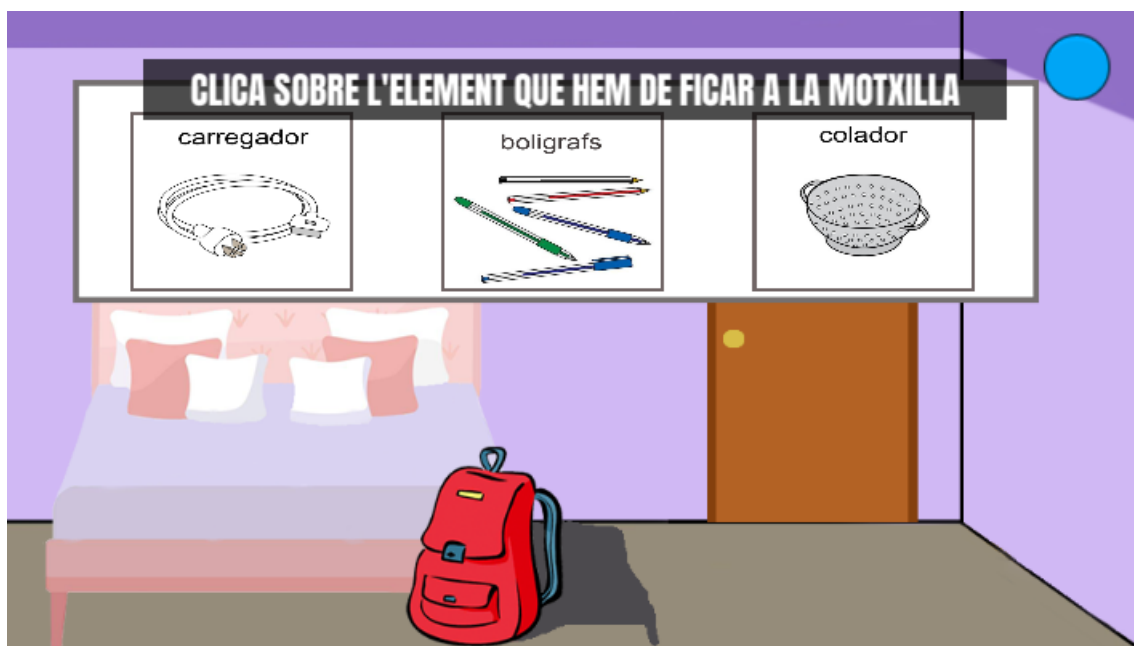
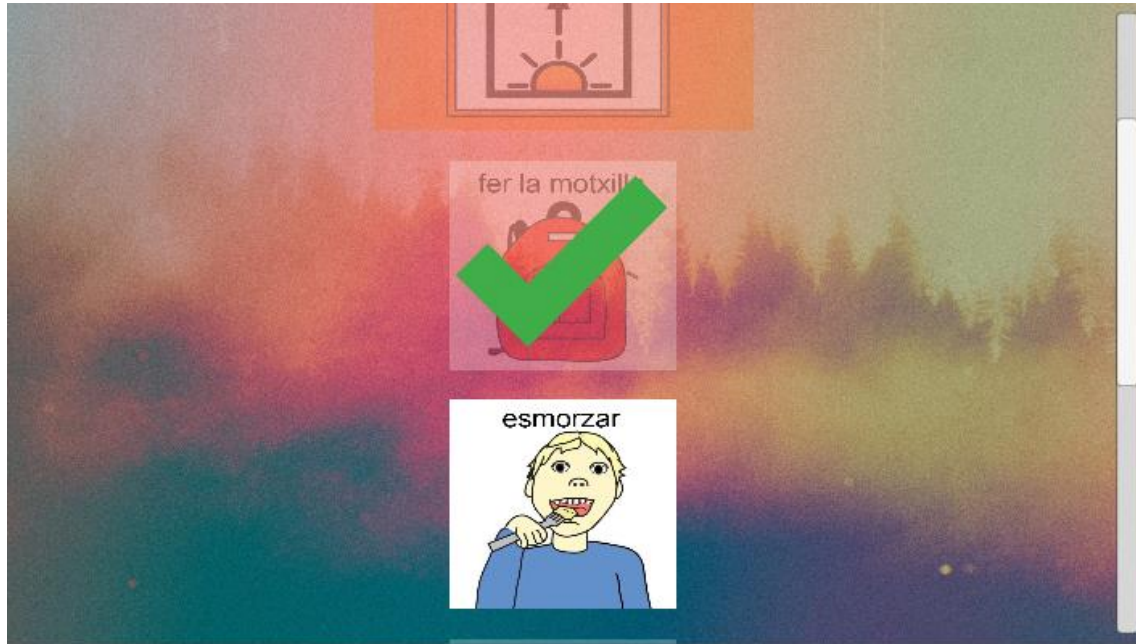


Figura 117 Captura Creador Calendari

Un cop finalitzat el minijoc, tornarem a la pantalla del calendari. Podrem veure que sobre la tasca que hem realitzat hi ha un tick verd que ens indica que ja hem realitzat la tasca i veurem que podrem fer la següent. Veure Figura 118.



*Figura 118 Captura Creador Calendari*

Un cop realitzem totes les tasques, podrem tornar al menú principal o tornar a començar el dia de nou sense haver de crear de nou el calendari.