Universitat de Girona
Escola Politècnica Superior

Treball final de grau

Estudi: Grau en Disseny i Desenvolupament de Videojocs

Títol: Desenvolupament i implementació d'un set d'animacions per el personatge d'un videojoc

Document: Memòria

Alumne: Carla Maya Rey

Tutor: Francesc Xavier Costa Brugué
Departament: OGEDP
Àrea: LSI

Convocatòria (mes/any): Septembre 2021

ÍNDEX

1. INTRODUCCIÓ

El mercat dels videojocs és un sector que ha anat creixent i evolucionant des de la seva primera aparició l'any 1946. Tot i que no va ser fins els anys 60 que van apareixer els primers videojocs moderns. Aquest creixement només es veu aturat per la creativitat dels desenvolupadors i la tecnologia.

És per això que a mesura que ha anat avançant la tecnologia ho han fet també els videojocs. Una de les formes d'evolució més evidents són els gràfics, al principi tots els jocs utilitzaven gràfics 2D ja que no hi havia capacitat per més, no va ser fins la dècada dels 90 que es van començar a veure videojocs en 3 dimensions.

Analitzant aquests aspectes es pot veure que des dels inicis de la industria moderna una part molt important en els videojocs ha estat l'apartat visual. Aquest no només consisteix en els entorns d'un videojoc o els disseny dels personatges, sinó també en les animacions dels mateixos. Les animacions dels personatges, enemics i diferents criatures del món d'un videojoc és el que dóna sensació de vida en el mateix.

Tenint en compte la importància de les animacions he decidit centrar aquest projecte en la creació i animació d'un personatge i un petit entorn de proves per poder veure i provar les diferents animacions desenvolupades.

1.1. Objectiu

L'objectiu d'aquest projecte és dissenyar un personatge en 3D amb les diferents animacions que necessita el personatge principal d'un videojoc, concretament es vol dissenyar el personatge d'un joc estil *RPG* en tercera persona i tot el set d'animacions d'aquest, des de les més bàsiques com saltar, correr, etc. fins altres més secundàries com fer diferents *idles*, entre altres.

Posteriorment es vol implementar totes les animacions en un petit escenari de proves on es puguin veure i provar.

Per aconseguir aquest objectiu s'ha de dissenyar el personatge de zero, es faran diferents dissenys i es triarà un, fent les diferents vistes (frontal i lateral) del disseny triat per poder fer el model en 3D, que es farà amb el programa Autodesk Maya. Amb el model fet es procedirà a texturitzar i fer el *rigging*. Un cop completat el procés anterior es començaran a fer les diferents animacions del personatge mencionades anteriorment. Per últim es farà la demo, que es farà utilitzant el motor de jocs Unity 3D, per fer-la s'exportaran les animacions

del Maya i s'importaran al motor de jocs. En la demo es podrà controlar el personatge i provar els diferents moviments que es poden fer, també es disposarà d'un punt amb un enemic per provar el combat. Per aconseguir això s'hauran de programar els diferents inputs amb els quals es controlarà el personatge, així com el moviment del mateix (moure's, saltar, etc.) també s'hauran d'implementar les diferents col·lisions (terra, parets, enemics) i les animacions corresponents en conseqüència si s'escau. Per la zona del combat s'haurà de programar un enemic que haurà de fer un atac contra el personatge. A més s'haurà de programar i construir l'arbre d'animacions del personatge, perquè es vegin correctament i fer que les diferents animacions s'activin en el moment adient. I altres detalls com la càmera del joc.

Per altra banda amb aquest projecte s'espera:

1. Dissenyar des de zero un personatge i les seves animacions.
2. Millorar en àmbits tècnics i en l'ús de Unity 3D.
3. Aprendre a utilitzar programes nous (Substance Painter).
4. Millorar en l'ús de programes ja coneguts (Autodesk Maya).
5. Millorar en la creació d'animacions per videojocs.

D'aquesta manera es vol aconseguir un projecte amb els següents pesos a cada apartat:

Estètica	80%
Narrativa	0%
Mecàniques	0%
Tecnologia	20%

Com es pot veure aquest treball té un form component estètic (80%) quedant la tecnologia en 20% ja que per la demo és necessari programar tots els moviments i animacions corresponents. Pel que fa a la narrativa s'ha deixat a 0% ja que no es té pensat donar-li una narració a l'entorn dissenyat. Les mecàniques seran les típiques d'un videojoc RPG és per aquesta manca d'innovació que s'ha decidit donar un pes del 0%.

2. ESTUDI DE VIABILITAT

L'estudi de viabilitat és necessari en l'àmbit professional per poder saber si un videojoc serà rentable. Per aquest motiu s'ha de tenir en compte diferents aspectes com la tecnologia, l'economia o la legalitat.

2.1. Recursos

2.1.1. Estudi de viabilitat tecnològica

- Hardware
    - Ordinador 1: Eina de treball de la part artística del projecte
    - Ordinador 2: Eina de treball de la part tècnica del projecte
- Software
    - Krita: Editor gràfic
    - Autodesk Maya: Entorn de desenvolupament de gràfics 3D
    - Substance Painter: Eina de texturitzat
    - Audacity: Eina d'edició de so
    - Unity: Motor de videojoc
    - Visual Studio: Entorn de programació
    - Google Drive: Eina d'emmagatzemament al núvol

2.1.2. Estudi de viabilitat econòmica

- Hardware
    - Ordinador 1: Cost aproximat en el moment de compra 800€ [disponible]
    - Ordinador 2: Cos de 2499€ [comprat durant el desenvolupament del projecte]
- Software
    - Krita: Cost 0€, codi obert
    - Autodesk Maya: Cost 0€, llicència d'estudiant gratuïta
    - Substance Painter: Cost 0€, llicència d'estudiant
    - Audacity: Cost 0€, codi obert
    - Unity: Cost 0€, llicència gratuïta dintre un llindar
    - Visual Studio: Cost 0€
    - Google Drive: Cost 0€, gratuït dintre d'un llindar

- Humà[1]
    - Dissenyador 3D: 12€/hora
    - Programador junior: 9’39€/hora

2.1.3. Estudi de viabilitat legal

El projecte dut a terme no presenta cap problema pel que fa als principals aspectes legals:

- **LOPD (Llei Orgànica de Protecció de Dades)**

  Aquesta llei no afecta al projecte desenvolupat, ja que aquest no guardarà cap dada de caràcter personal del jugador

- **LSSICE (Llei de Serveis de la Societat de la Informació i Comerç electrònic)**

  Aquesta llei tampoc és aplicable, ja que el projecte està pensat únicament com a treball educatiu sense intenció de ser comercialitzat.

- **PEGI(Pan European Game Information)**

  PEGI és un sistema de classificació europeu voluntari. El sistema informa als compradors de l’edat recomanada i de contingut sensible del joc, aquesta informació ve donada a través d’unes icones.

  En el cas d’aquest projecte es necessitaria l’etiqueta de violència, ja que es lluita contra enemics. Per això es considera que es podria classificar dintre de *PEGI* 12.

Figura 1: Icones de la classificació PEGI
(https://www.elsotanoperdido.com/noticias/el-codigo-regulador-bbfc-deja-de-existir/2012073032063)

[1] Sous extrets de la pàgina: es.indeed.com

- **Drets d'autor i Copyright**

  El desenvolupament del projecte ha estat dut a terme íntegrament per mi amb l'ajut del tutor i algun company a excepció dels sons i música utilitzats que s'han buscat amb llicència Creative Commons.

  Tot i que no es té la intenció de comercialitzar el projecte en cas d'un canvi d'opinió s'hauria de tenir en compte alguns aspectes del software utilitzat:

  - Autodesk Maya: La llicència utilitzada per el desenvolupament és d'estudiant, per tant en cas de comercialització s'hauria d'accedir a una llicència comercial.
  - Substance Painter: Com en el cas anterior, la llicència utilitzada és d'estudiant i per tant també s'hauria d'accedir a una llicència comercial, en aquest cas, però, també s'ha de tenir en compte que hi ha diferents llicències segons un llindar d'ingressos i en cas de superar-lo s'ha d'actualitzar la llicència a una apta.
  - Unity 3D: Aquest cas és similar a l'anterior. Unity disposa de llicències gratuïtes fins a un cert llindar d'ingressos en els últims 12 mesos, en cas de superar-lo s'ha de pagar el preu de la llicència.

2.1.4. Conclusions

Com es pot veure hi ha viabilitat econòmica, ja que es disposa de les eines i dispositius necessaris per el desenvolupament del projecte.

A més el cost humà és de 0€, ja que en ser un treball universitari hi dedicaré les hores necessàries sense fer cap contractació.

2.2. Estudi de mercat

2.2.1. Estat de l'art

El gènere RPG és molt estès a la indústria dels videojocs, hi ha molt títols classificats dintre aquest gènere. Després d'analitzar diversos jocs s'ha arribat a la conclusió de que les mecàniques principals estan molt arraigades al gènere i hi ha molt poca variació entre els diferents títols del mercat. Hi ha 2 grans grups de RPG, cadascun amb unes mecàniques diferents en combat, aquests grups són els següents:

RPG de combat per torns

Com indica el seu nom en aquest tipus de RPG els combats es fan per torns, normalment el diferents personatges i enemics tenen una velocitat i el primer torn l'obté qui tingui la velocitat més alta. A partir d'aquí el combat es desenvolupa seguint l'ordre establert. Aquest ordre pot ser modificat amb moviment que pugin o baixin la velocitat dels personatges.

Figura 2: Imatge del joc De Fobos y Deimos (http://magcedonia.com/alexander-rodriguez-queria-que-en-de-fobos-y-deimos-la-sexualidad-fuese-un-problema/de-fobos-y-deimos-batalla/)

Aquests jocs disposen d'un menú des d'on es selecciona que farà cada personatge, ja sigui indicant totes les accions de con com en el cas del Final Fantasy o en el torn de cada personatge com en el cas del Persona 5.

Figura 3: Imatge del joc Final Fantasy X (https://uvejuegos.com/guia/Final-Fantasy-X/Guia-completa/3042/26/8)

Una mecànica diferent que tenen alguns d'aquests jocs és tenir la capacitat, ja sigui sempre o durant un temps, de saber l'ordre dels torns en que es desenvoluparà el combat. El que tots tenen en comú és que donen temps per pensar en una estrategia que podria decidir el combat.

Figura 4: Imatge del joc Pokemon Rojo Fuego
(https://vandal.elespanol.com/noticia/1350710505/verano-de-pokemon-pokemon-rojo-fuego-y-verde-hoja/)

Com es pot veure a les diferents figures presentades el jugador pot controlar tant un grup de personatges (figura 1 i 2) que normalment no es podrà canviar durant el combat o controlarà només un personatge (figura 4) que serà possible canviar per un altre durant el combat.

RPG de combat a temps real o ARPG (Action RPG)

En aquest tipus de RPG el jugador té sempre el control del moviment del personatge, això implica que durant els combats el jugador s'haurà de moure per esquivar atacs, haurà d'apropar-se als enemics per atacar, etc. tot això en el moment que el jugador vulgui.

Aquest tipus de combat posa a prova les reaccions del jugador que ha d'estar atent per no rebre mal i per atacar en el moment adient als enemics. A més s'ha de tenir en compte la manca de temps per pensar en una estratègia, com també diferents aspectes com el patró d'atac dels enemics o buscar un lloc allunyat de la lluita en cas de necessitar recuperar vitalitat. Alguns exemples d'aquest estil de RPG són la majoria de jocs de la saga Kingdom Hearts o el Code Vein entre molts.

Figura 5: Imatge del videojoc Kingdom Hearts 3
(https://kingdomofhart.wordpress.com/2019/08/11/kingdom-hearts-iii-parte-3/)

Figura 6: Imatge del joc Code Vein
(https://www.koi-nya.net/2017/09/18/nuevo-video-code-vein-centrado-combate-jefe/)

2.2.2. Públic objectiu i tipologia de jugador

Aquest projecte està destinat per tots els públics, ja que no es desenvoluparà un joc com a tal sinó un entorn per veure i provar les mecàniques d'aquest.

En cas de que es volgués desenvolupar un joc RPG sencer a partir d'aquest treball es destinaria a un públic jove que li agradés jugar jocs llargs i amb un cert pes narratiu.

Per poder fer un anàlisi més profund del públic objectiu s'ha agafat la classificació creada per Richard Bartle anomenada taxonomia de Bartle, aquesta classificació consisteix en dividir els jugadors en 4 grups segons les seves preferències d'interacció i accions.

- **Achiever/Triunfador:** Són els jugadors que prefereixen guanyar la major quantitat de punts, equip, nivells i assoliments del joc. Els seus esforços gairebé no tenen benefici en el joc, busquen el prestigi que els otorga aquests assoliments. Són atrets per jocs que tenen una gran varietat d'assoliments o una taula de classificació.
- **Explorer/Explorador:** Són els jugadors que els hi agrada l'exploració, descobrir àrees noves i llocs ocults. El gènere que més els atreu és el de món obert per la llibertat que ofereix en l'exploració.
- **Socializer/Social:** Són els jugadors que disfruten el joc per la seva part social, ja sigui interactuant amb altres jugadors o amb personatges no jugables.
- **Killers/Assassí:** Són els jugadors que prefereixen la competència, ja sigui amb altres jugadors o amb personatges no jugables.

Figura 7: Imatge de la classificació de la taxonomia de Batler
(https://commons.wikimedia.org/wiki/File:Taxonom%C3%ADa_de_Bartle.png)

Tenint en compte aquesta classificació el perfil que més serà atret per aquest projecte és el dels exploradors. El jugador podrà moure's per tot l'escenari i provar les diferents mecàniques.

En el cas que es fes un videojoc sencer a partir del projecte, aquest perfil es veuria substituït per el perfil dels assassins i el dels triomfadors, això és degut a que el joc es tornaria molt més lineal amb menys possibilitats d'explorar i obriria les portes a la competició contra altres personatges no jugables (perfil dels assassins) i a poder aconseguir diferents assoliments (perfil dels triomfadors).

3. PLANIFICACIÓ

Per fer la planificació del projecte s'han tingut en compte les diferents tasques que es duran a terme, agrupant-les en diferents grups com es pot veure a la següent figura (figura 8):

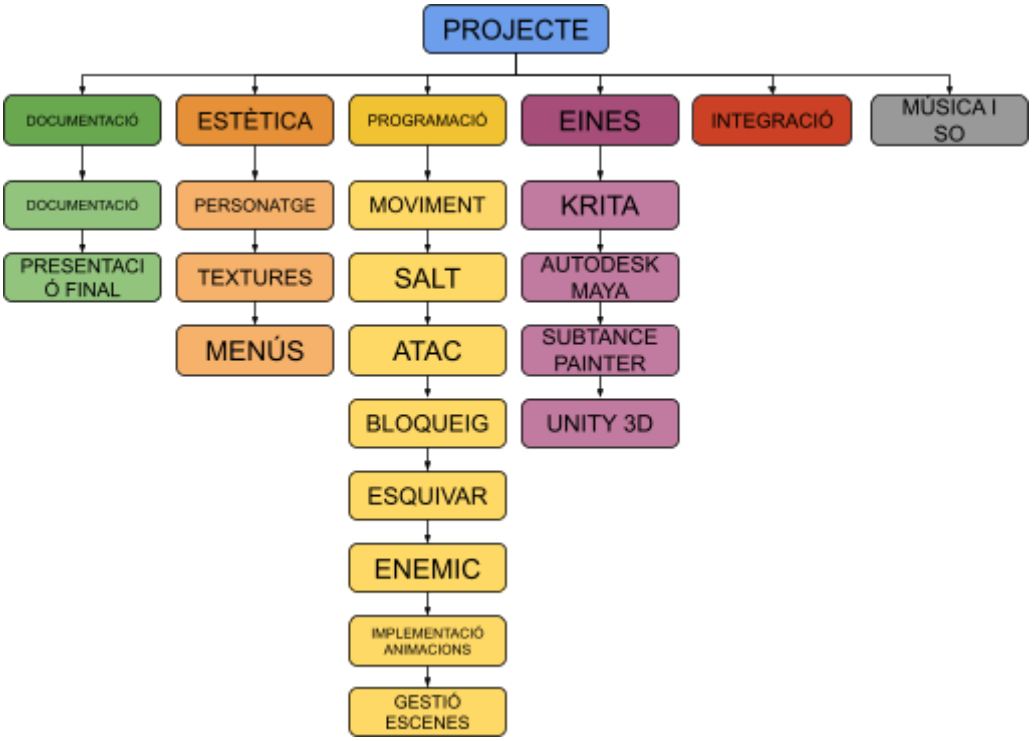


Figura 8: Representació dels paquets de treball en forma d'esquema

3.1. Tasques

- **Programació**
  - Moviment del personatge: Creació del sistema que permet al jugador moure el personatge.
  - Accions del personatge: Creació de les diferents accions que pot executar el jugador
  - Accions de l'enemic: Creació el comportament de l'enemic.
  - Gestió d'escenes: Creació i gestió de les escenes del joc.

- **Art**
    - Disseny personatge: Creació del personatge del joc. Disseny de les diferents vistes (frontal i lateral), modelat i rigging.
    - Disseny textures personatge: Creació de les textures per el personatge.
    - Disseny textures enemic: Creació de les textures de l'enemic.
    - Disseny animacions: Creació de les animacions del personatge del joc.
- **Programació i art**
    - Menús: Creació dels menús que dónen accés a la partida i sortida del programa.
    - Implementació animacions: Implementació de les animacions de manera que es vegin correctament i es transicioni d'una a l'altre degudament.
- **Miscel·lani**
    - Documentació: Crear i preparar la documentació per incloure a la memòria.
    - Presentació final: Presentar el projecte.

3.2. Paquets de treball

- **Paquet de treball 1** - Documentació
    - **Tasques:** Documentació i presentació final.
    - **Temporalització:** Durant tot el projecte.
    - **Fita:** Documentació i presentació final.

- **Paquet de treball 2** - Estètica
    - **Tasques:**
        - **Sub-paquet de treball 2.1:** Personatge - Disseny
            - **Tasques:** Disseny del personatge.
            - **Fita:** Personatge dissenyat.
        - **Sub-paquet de treball 2.2:** Personatge - Modelat
            - **Taques:** Modelat del personatge i l'arma que porta.
            - **Fita:** Personatge modelat.
        - **Sub-paquet de treball 2.3:** Personatge - Rigging
            - **Taques:** Riggejat del personatge.
            - **Fita:** Personatge amb rigging.
        - **Sub-paquet de treball 2.4:** Personatge - Animacions
            - **Tasques:** Disseny i creació de les animacions.
            - **Fita:** Animacions fetes.
        - **Sub-paquet de treball 2.5:** Texturitzat - UV
            - **Tasques:** Desplegament de les uv's del personatge.
            - **Fita:** Desplegades les uv's.
        - **Sub-paquet de treball 2.6:** Texturitzat - Textures personatge
            - **Tasques:** Disseny i creació de les textures del personatge.
            - **Fita:** Textures exportades.

        - **Sub-paquet de treball 2.7:** Texturitzat - Textures enemic
            - **Tasques:** Disseny i creació de les textures de l'enemic.
            - **Fita:** Textures exportades.
        - **Sub-paquet de treball 2.8:** Menús
            - **Tasques:** Menú principal, menú de pausa i menú de fi de partida.
            - **Fita:** Menús del joc dissenyats.
    - **Temporalització:** Durant tot el projecte
    - **Fita:** Dissenyat tot l'art del projecte

- **Paquet de treball 3** - Programació
    - **Tasques:**
        - **Sub-paquet de treball 3.1:** Moviment
            - **Tasques:** Moviment del personatge i interacció.
            - **Fita:** Generat tot el moviment i la interacció del personatge.
        - **Sub-paquet de treball 3.2:** Salt
            - **Taques:** Mecànica de salt del personatge.
            - **Fita:** Generada la mecànica de salt.
        - **Sub-paquet de treball 3.3:** Atac
            - **Taques:** Mecànica d'atac del personatge, fins a una combinació de 3 atacs seguits.
            - **Fita:** Generada la mecànica d'atac.
        - **Sub-paquet de treball 3.4:** Bloqueig
            - **Tasques:** Mecànica de bloqueig.
            - **Fita:** Generada la mecànica de bloqueig.
        - **Sub-paquet de treball 3.5:** Esquivar
            - **Tasques:** Mecànica d'esquivar atacs.
            - **Fita:** Generada la mecànica d'esquivar.
        - **Sub-paquet de treball 3.6:** Enemic
            - **Tasques:** Zona d'interacció amb l'enemic i atac.
            - **Fita:** Generada tota la lògica de l'enemic.
        - **Sub-paquet de treball 3.7:** Implementació animacions
            - **Tasques:** Implementar les animacions a l'entorn.
            - **Fita:** Implementades les animacions.
        - **Sub-paquet de treball 3.8:** Gestió escenes
            - **Tasques:** Gestionar les diferents escenes.
            - **Fita:** Generada la lògica de la gestió d'escenes.
        - **Sub-paquet de treball 3.9:** Build i test
            - **Tasques:** Generar la build, testejar i solucionar errors.
            - **Fita:** Generada la build sense bugs.
    - **Temporalització:** Segona part del projecte.
    - **Fita:** Generada tota la lògica del projecte amb l'art implementat.

- **Paquet de treball 4** - Eines
    - **Tasques:** Aprenentatge de les eines utilitzades, si s'escau.
    - **Temporalització:** Durant tot el projecte.
    - **Fita:** Assoliment dels coneixements necessaris per el projecte

- **Paquet de treball 5** - Integració
    - **Tasques:** Integrar tots els elements del projecte i combinar-los.
    - **Temporalització:** Segona part del projecte.
    - **Fita:** Elements correctament introduïts al projecte.

- **Paquet de treball 6** - Música i so
    - **Tasques:** Trobar música ambiental i sons per les accions del personatge.
    - **Temporalització:** Segona part del projecte.
    - **Fita:** Obtenció de la música i sons necessaris.

3.3. Metodologia de treball

En ser un projecte individual no ha estat necessari fer reunions per planificar-lo. S'ha decidit fraccionar el projecte en paquets (veure punt anterior) per tal de tenir control de les tasques fetes i les que encara queden per fer. De manera setmanal es valorarà el progrés fet i, si cal, es modificarà el cronograma, sempre intentant mantenir la data d'entrega.

Tot i ser individual el treball és tutoritzat, per això es faran reunions amb el tutor per informar dels progressos fets, plantejar dubtes que surgeixin i demanar opinió i consell en cas de ser requerit.

Per el desenvolupament s'ha decidit fer una metodologia de treball lineal, és a dir, començar amb una tasca i no passar a la següent fins acabar-la. S'ha decidit fer així ja que el procediment del disseny del personatge ha de seguir aquest mètode, això és degut a que, per exemple, no es pot començar a animar sense tenir el rigging prèviament fet i aquest no es pot fer sense el model, etc.

Per això el projecte es dividirà en dos parts, la primera on es realitzarà tot el procediment artístic i la segona on es desenvoluparà la part més tècnica i de programació.

3.3.1. Cronograma

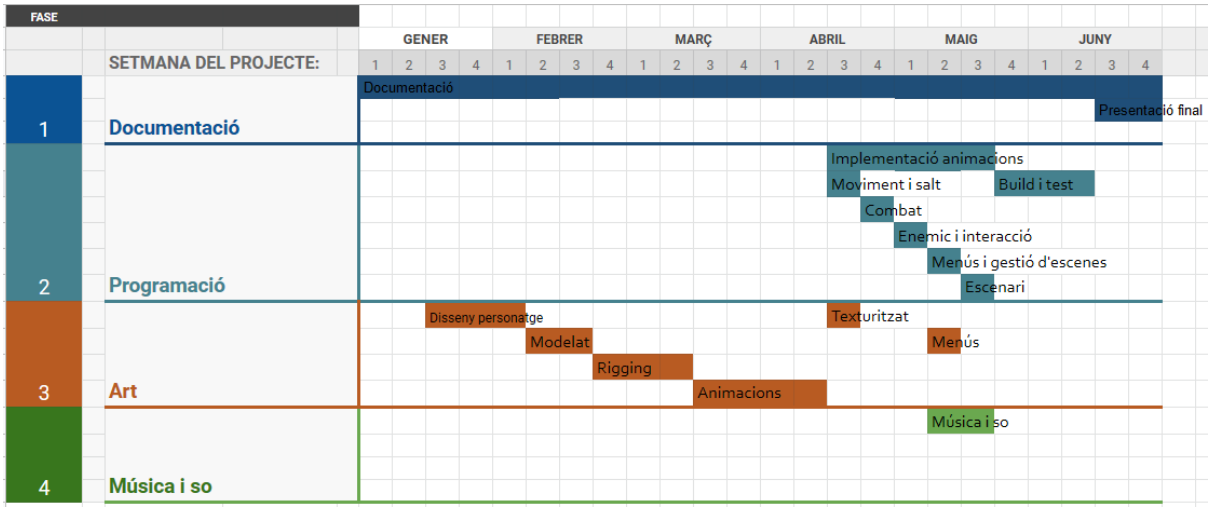


Figura 9: Cronograma de la planificació

3.4. Resultats esperats del projecte

S'espera crear una escena de testeig on poder provar els diferents moviments i accions del personatge i les seves respectives animacions, implementades correctament. Amb la possibilitat d'ampliar l'avast d'accions.

4. MARC DE TREBALL I CONCEPTES PREVIS

A continuació es presentaran els entorns de treball utilitzats en el desenvolupament del projecte i s'analitzaran amb més detall alguns dels jocs que es considerarien competència directe.

4.1. Entorns de treball

Tots els entorns utilitzats en el projecte han estat gratuïts

4.1.1. Krita

Krita és un software gratuït i de codi lliure per pintura digital i il·lustració. És distribuït sota la llicència GNU Public License versió 2 o més actual.

S'ha triat aquest software perquè és gratuït i s'estava familiaritzat amb ell. Les altres possibles eleccions eren de pagament o menys conegudes.

4.1.2. Autodesk Maya 2020

Autodesk Maya 2020 és un software de desenvolupament de gràfics 3D, animació, efectes especials i renderitzat. La primera versió de Maya va ser llançada el 1998 amb el nom de Maya per l'empresa Alias-Wavefront, renombrada més tard com a Alias. El nom d'Autodesk Maya es va adoptar després de la compra d'Alias per part d'Autodesk.

S'ha triat aquest software per que és amb el que s'estava més familiaritzat i per tenir una llicència gratuïta per estudiants.

4.1.3. Substance Painter

Substance Painter és un software de texturitzat 3D. Adquirit per Adobe.

Aquest software ha estat triat per la facilitat que atorga a l'hora de fer textures i per tenir una llicència gratuïta per estudiants.

4.1.4. Audacity

Audacity és un software d'edició i gravació de so de codi obert sota la llicència GNU Public License versió 2.

S'ha triat aquest software per ser gratuït i per la seva facilitat d'edició.

4.1.5. Google Drive

Google Drive és un sistema d'emmagatzemament al núvol propietat de Google. També disposa d'editor de text i full de càlcul entre altres. Va ser llançat al 2012 com a substitut de Google Docs.

S'ha triat aquest software per que ja es tenia un compte de Google, el qual dóna accés als diferents serveis de Google.

4.1.6. Visual Studio

Visual Studio és un entorn de desenvolupament integrat o IDE per les seves sigles en anglès. El qual dóna suport a molts llenguatges de programació. És propietat de Micrsoft.

S'ha triat aquest software ja que és el que recomana Unity per desenvolupar el codi.

4.1.7. Unity 3D

Unity 3D és un motor desenvolupament de videojocs 2D i 3D creat per Unity Technologies i tret al mercat al 2005. Unity té suport de compilació per diferents plataformes.

Ha estat el motor de jocs triat per la familiaritat que és té i per tenir una llicència gratuïta per persones o organitzacions que no superin els 100.000 USD de beneficis en els últims 12 mesos.

4.2. Competència directa

En l'estudi de viabilitat (veure punt 2.2) s'han analitzat mínimament alguns videojocs semblants al desenvolupament d'aquest projecte en cas de fer-ne un joc sencer.

El joc proposat estaria dintre la categoria de RPG de combat a temps real, per això a continuació s'analitzaran amb més profunditat els exemples proposats anteriorment per tal de veure les diferències i poder-los comparar.

Kingdom Hearts 3

Com ja s'ha dit *Kingdom Hearts 3* (com la majoria de jocs de la saga) és un videojoc RPG amb combat a temps real.

Les característiques del combat d'aquest videojoc són que el personatge té una combinació bàsica d'atac de fins a 3 moviments a terra i una altra diferent en cas d'estar a l'aire, aquesta combinació es pot ampliar a mesura que es puja de nivell, a més de desbloquejar altres habilitats útils per el combat. El sistema de combat disposa també de màgia, ja sigui per atacar els enemics, curar els aliats o inflingir canvis d'estat. I per finalitzar hi ha la utilització d'objectes.

Durant el combat es disposa també d'un equip de personatges no jugables que pot ser de fins a 4 integrants.

Figura 10: Imatge del joc Kingdom Hearts 3
(https://www.hobbyconsolas.com/guias-trucos/kingdom-hearts-3/kingdom-hearts-iii-15-cosas-me-hubiera-gustado-saber-antes-empezar-jugar-364321)

A diferència del Kingdom Hearts en el cas de desenvolupar un joc a partir del projecte, aquest no disposaria d'un sistema màgic ni es tindrien ajudants. Sí que hi hauria objectes especials per el combat així com de cura de vitalitat. A més no hi hauria un sistema de nivells, el creixement del personatge vindria donat per la narrativa.

Pel que fa l'estètica Kingdom Hearts en ser una saga que es basa en els mons de les pel·lícules de Disney presenta una estètica infantil que pot fer allunyar-se al públic objectiu. En el cas del projecte aquest aposta per una estètica de textures semi-realistes i de colors més apagats.

Code Vein

Code Vein és un joc RPG d'acció a temps real amb la particularitat que és un souls-like, això vol dir que moltes de les mecàniques del joc estan basades en les mecàniques de la saga *Dark Souls.* Aquest fet implica que seran molt diferents de les mecàniques proposades pel projecte.

El sistema de combat del Code Vein es basa en la gestió de l'energia. El personatge té un màxim d'energia que pot utilitzar tant per atacar com per esquivar atacs, aquesta energia s'omple si no es fa cap acció que en requereixi. També hi ha una energia especial, que equival als punts d'experiència, que es perd en cas de morir en un combat i és necessari tornar al lloc de la derrota per recuperar-la, aquesta energia especial no serveix de manera directa en el combat sinó que s'utilitza per poder pujar de nivell.

El combat disposa també d'uns atacs especials i uns objectes equipables (fins un màxim de 8) que serveixen per millorar temporalment els estats del personatge.

Com en el cas del Kingdom Hearts, durant el combat es disposa de l'ajut de dos personatges no jugables.

Figura 11: Imatge del joc Code Vein
(https://geemugeemu.com/noticia/nuevos-detalles-del-sistema-de-combate-de-code-vein/)

Com ja s'ha comentat al principi el sistema de combat del Code Vein respecte al projecte són molt diferents, ja que en aquest treball s'aposta per les mecàniques típiques dels combats dels RPG. Les diferències més notables són que en el projecte no hi ha gestió d'energia en el combat, a més no es podran utilitzar objectes d'augment d'estats. També, com en el cas anterior, no es disposarà d'un equip de personatges no jugables.

L'estètica del Code Vein, però, s'assembla més a l'estètica proposada, amb unes textures semi-realistes.

5. DISSENY DEL VIDEOJOC

A continuació es presentaran tots els detalls de la funcionalitat del joc.

5.1. Funcionalitat del joc

- Menú principal
    - Accedir al joc: El jugador començarà el joc.
    - Sortir del joc: El jugador sortirà del joc.

- Pantalla de joc
    - Menú de pausa: El jugador accedirà al menú de pausa de la partida.
        - Retornar a la partida: El jugador retornarà a la partida.
        - Tornar al menú principal: El jugador anirà al menú principal.
    - Controlar el personatge: El jugador controlarà el personatge, el control inclou moviment, salt i dash (només en combat).
    - Executar les habilitats del personatge: El jugador controlarà les habilitats del personatge, les habilitats inclouen atacar (combinació màxima de 3 atacs, només en combat) i bloquejar atacs enemics (només en combat).

- Menú de fi de partida
    - Accedir a una nova partida: El jugador tornarà a començar l'experiència del joc.
    - Sortir del joc: El jugador sortirà del joc.

5.2. Hardware

Aquest projecte ha estat desenvolupat en dos ordinadors diferents.

Per la part d'art del projecte (exceptuant el texturitzat) s'ha utilitzat un equip amb les següents característiques:

- CPU: Intel Core i7
- GPU: Nvidia Geforce 820M
- Memòria: 8GB
- SO: Windows 8.1

Per la part tècnica del projecte s'ha utilitzat el següent equip

- CPU: Intel Core i7
- GPU: Nvidia Geforce GTX 3070
- Memòria: 32GB
- SO: Windows 10

El joc ha pogut ser executat sense problema en l'equip de treball de la primera part.

5.3. Disseny del videojoc

Tot seguit es podran veure tots els detalls del disseny del joc per el projecte.

5.3.1. Descripció general

Com ja s'ha dit el joc creat es basa en els jocs RPG d'acció real, concretament en les escenes de testeig dels mateixos. Amb una càmera mòbil situada primerament darrere el personatge i que el segueix. El jugador podrà controlar el personatge i realitzar les accions bàsiques dels jocs RPG.

Una escena de testeig és aquella que serveix als desenvolupadors per provar les mecàniques del joc, les animacions, les simulacions, etc. En cap cas aquestes escenes estan destinades a ser incloses al producte final.

Figura 12: Imatge d'una escena de testeig del joc Kingdom Herats 3
(https://www.youtube.com/watch?v=JvkomcIA2W0)

5.3.2. Mecàniques

Espai

Visió global

A continuació es presenta un esquema amb la definició d'espais que seguirà el projecte:

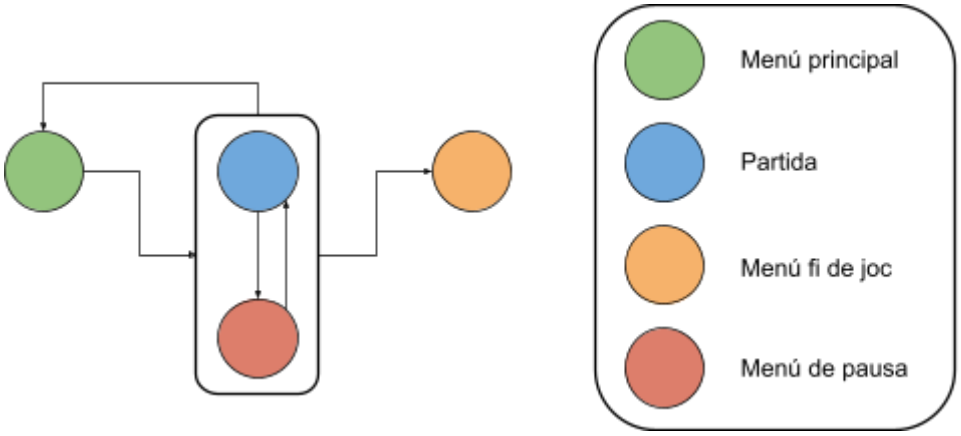


Figura 13: Definició dels espais en forma d'esquema

Tot seguit es presenta la visió global del que seria el menú principal en cas de desenvolupar el joc com a tal. Per el projecte no s'han creat totes les pantalles del menú, es per això que en l'esquema s'utilitza un codi de colors, així doncs els elements de color verd són els inclosos al projecte, mentre que els vermells són els que no s'han desenvolupat.

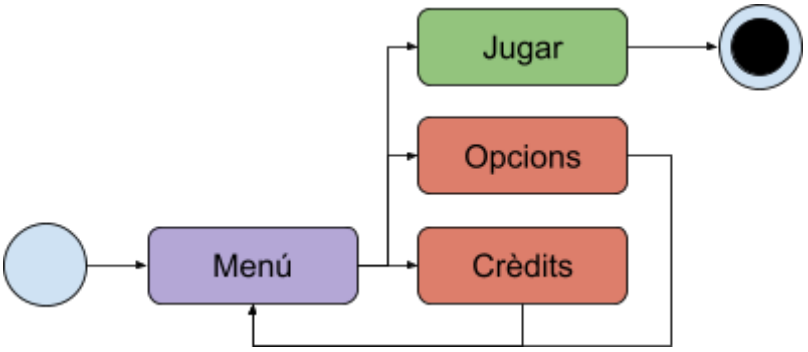


Figura 14: Visió global del menú principal

Definició dels espais

En començar el joc s'entra al menú principal, aquí el jugador pot escollir entre començar la partida, canviar les opcions, veure els crèdits i sortir del joc:

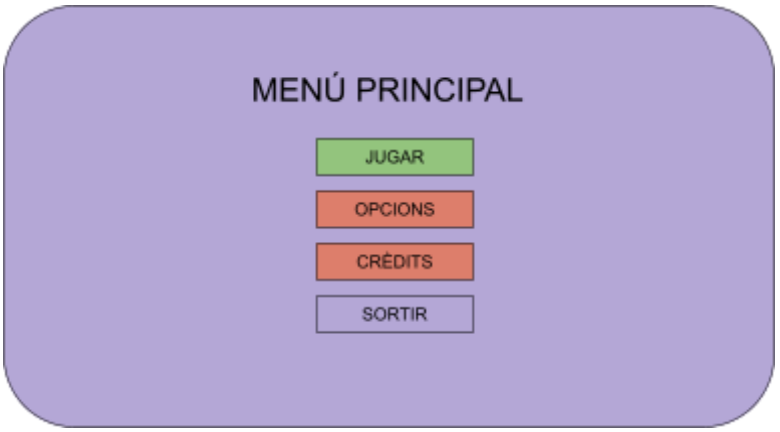


Figura 15: Representació del menú principal

En l'espai del menú d'opcions es pot canviar el volum del la música i els sons i també la resolució de pantalla.

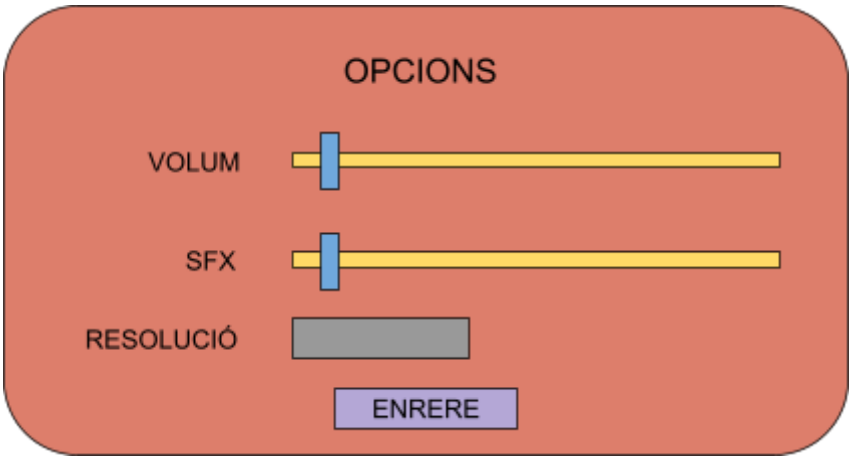


Figura 16: Representació del menú d'opcions

A l'espai de crèdits es podrà veure una llista amb els noms dels participants en el joc i la tasque desenvolupada.

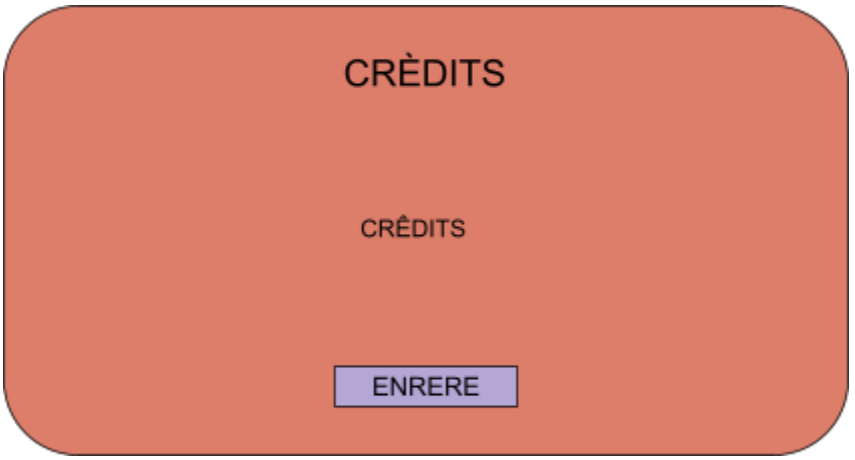


Figura 17: Representació del menú de crèdits

En entrar a la partida el jugador es trobarà en un entorn 3D on hi ha 4 zones, 3 d'elles amb diferents estructures i la última amb un enemic per provar les mecàniques de combat. A continuació es mostra un esquema del mapa amb vista des de dalt:

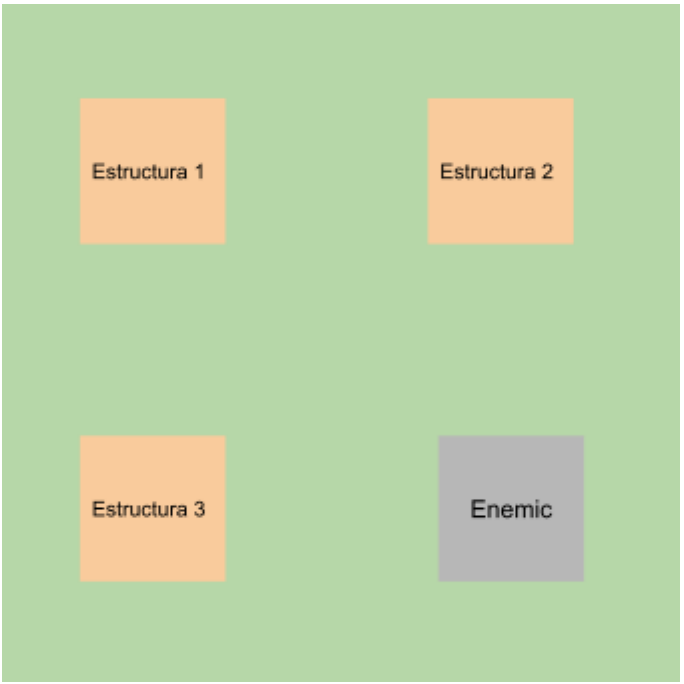


Figura 18: Representació de l'espai de joc

En l'espai de joc l'usuari pot accedir en qualsevol moment a un menú de pausa, en aquest menú podrà retornar a la partida o anar al menú principal:

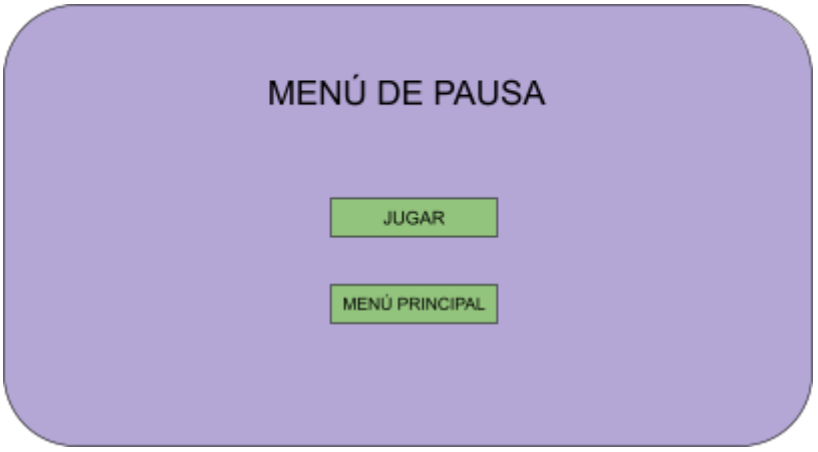


Figura 19: Representació del menú de pausa

Finalment l'últim espai del joc és el menú de final de partida, a aquest només es pot arribar en cas de que l'enemic mati el personatge:

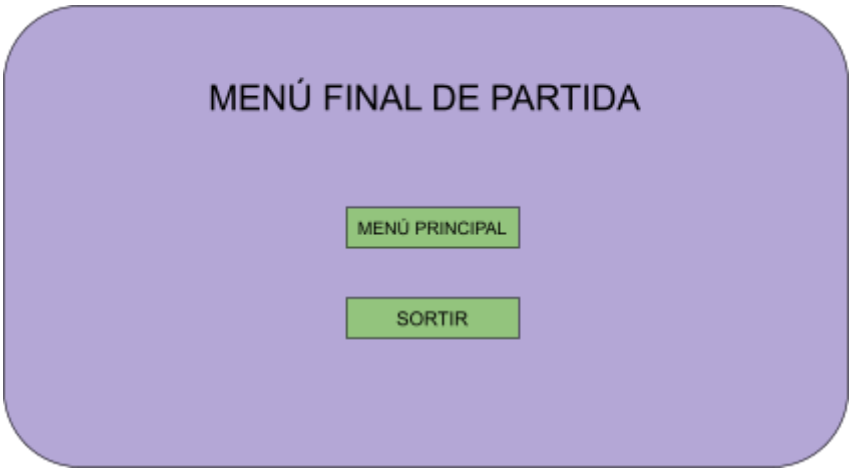


Figura 20: Representació del menú de final de partida

Accions i lògica

A continuació es podrà veure una llista de les accions que podria dur a terme el jugador en el diferents espais mencionats en el punt anterior.

Menú principal

- **Accedir al joc:** Apretant un botó específic per aquesta funcionalitat.
- **Accedir a les opcions:** Apretant un botó específic per aquesta funcionalitat
- **Accedir als crèdits:** Apretant un botó específic per aquests funcionalitat.
- **Sortir del joc:** Apretant un botó específic per aquesta funcionalitat.

Menú d'opcions

- **Modificar valors:** El jugador podrà modificar les configuracions apretant els botons o utilitzant els lliscadors.
- **Tornar al menú principal:** Apretant un botó específic per aquesta funcionalitat.

Crèdits

- **Tornar al menú principal:** Apretant un botó específic per aquesta funcionalitat.

Partida

- **Moviment:** El jugador es podrà moure per l'escenari.
- **Salt:** El jugador podrà saltar, meitat d'un salt no es pot moure.
- **Atacar:** El jugador podrà atacar si es troba a una zona amb enemics. Es pot fer un combo de fins a 3 atacs.

    Si protagonista és a zona enemic i input atac:

        Executa atac 1

        Si protagonista és a zona enemic i input atac avans de finalizar atac 1:

            Executa atac 2

            Si protagonista és a zona enemic i input atac avans de finalizar atac 1:

                Executa atac 3
- **Bloquejar:** El jugador podrà bloquejar atacs enemics.
- **Esquivar:** El jugador podrà esquivar atacs enemics.
- **Rebre un cop:** El jugador pot ser atacat per l'enemic.

    Si enemic colpeja protagonista i aquest no està defensant o esquivant:

protagonista.vida -= enemic.atac

- **Morir:** El jugador podrà morir si és colpejat diverses vegades.

  Si protagonista.vida = 0:

  Fi de joc i càrrega de Menú fi de partida

- **Obrir menú de pausa:** El jugador podrà obrir el menú de pausa.

Jerarquia de reptes

Degut a la naturalesa d'aquest projecte el jugador no ha de superar cap repte en el joc desenvolupat. Per aquest motiu a continuació es presentarà la jerarquía de reptes que tindria en cas de desenvolupar-se com a joc complet:

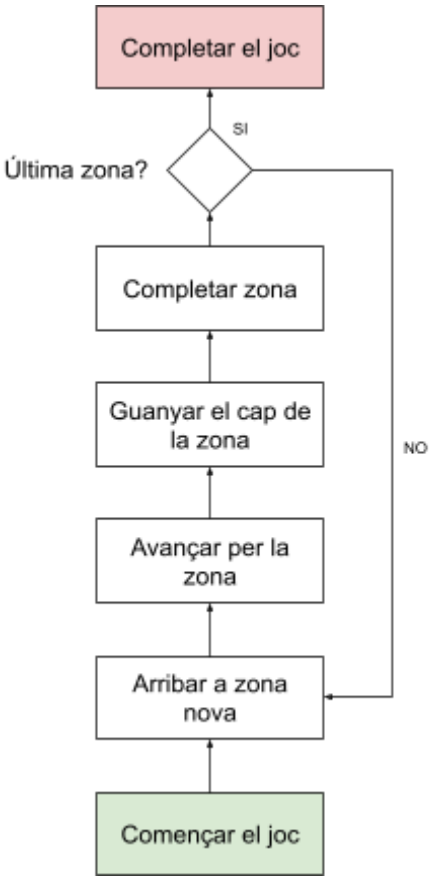


Figura 21: Esquema de la jerarquia de reptes

Objectes

A continuació es poden veure els objectes i els atributs del joc.

Protagonista		
Vida	Cops que pot rebre el personatge sense morir	int [0...3]
Posició	Localització espacial en el mapa	Vector3(posició dins l'escenari)

Enemic		
Atac	Punts de vida que treu al protagonista	int [0...3]
Posició	Localització espacial en el mapa	Vector3(posició dins l'escenari)

Estructures		
Posició	Localització espacial en el mapa	Vector3(posició dins l'escenari)

Economia del joc

En aquest projecte l'únic factor d'economia és la vida del personatge que pot disminuir quan l'enemic el colpeja. En el cas d'haver desenvolupat un joc els elements de l'economia haguessin estat els següents:

- **Monedes:** Sistema econòmic del món del joc.
  - Sources: Combats.
  - Drains: Cap.
  - Traders: A les botigues es poden gastar les monedes per diferents objectes.
  - Converters: Cap.
- **Vida:** Número de cops que pot rebre el personatge abans de morir.
  - Sources: Combats i alguns objectes.
  - Drains: Atacas enemics exitosos.
  - Traders: Cap.
  - Converters: Cap.

- **Punts d'habilitat:** Punts que permeten desbloquejar habilitats.
    - Sources: Pujar de nivell
    - Drains: Cap.
    - Traders: Cap.
    - Converters: Activar habilitats del personatge

5.3.3. Estudi i disseny de personatges

El personatge principal és l'element més important del desenvolupament del projecte, ja que aquest és centra en el disseny i les animacions del personatge i per tant tot el projecte gira entorn a aquest fet.

Com a tal el disseny del personatge està pensat per ser el protagonista del joc. Això vol dir que seria la base per avançar en la història del joc complet. El personatge també és la base per tota la jugabilitat del joc, ja que és l'únic que l'usuari podrà controlar.

A continuació es poden veure tots els elements del disseny del personatge per el desenvolupament del projecte.

Estètica

S'ha decidit utilitzar una estètica 3D estil anime japonès, ja que el 3D facilita fer les animacions i és una estètica molt utilitzada en els videojocs RPG actuals. A més es creu que l'estil 3D agradarà més al públic objectiu.

Les textures dels personatges són semi realistes i de colors apagats, mentre que les textures de l'escenari són de colors més vius per destacar d'aquesta manera els diferents personatges i enemics.

Figura 22: Imatge del joc Genshin Impact, referencia per l'estil de personatge
(https://www.eurogamer.es/articles/2020-10-30-genshin-impact-recibira-la-actualizacion-1-1-en-noviembre)

Característiques del personatge

El personatge que controla el jugador és un noi jove de 17 anys d'1m i 75 cm d'altura i 80 kg de pes. Té els ulls liles amb el cabell bru i punxegut. És un noi alegre i segur de sí mateix al que no li agrada rendir-se, que ha sortit del seu poble natal per saciar les ànsies d'explorar món que sempre ha tingut des que era petit.

En sortir al món ha pogut comprovar amb la seva pròpia pell que viure tot sol és difícil, ja que ara ha d'aconseguir ell mateix els diners per poder satisfer les necessitats més bàsiques. És per aquest motiu que decideix acceptar treballs com a caçador de monstres i bestioles.

Llenguatge corporal

Com s'ha dit abans és un noi segur de sí mateix i aquest fet es vol transmetre amb les animacions, per aquest motiu en l'animació de caminar es pot veure com es desplaça alçat transmetent seguretat i força a cada passa. És el mateix cas en l'animació de correr. Aquesta seguretat també es pot apreciar en els atacs que són ràpids i precisos. Per últim amb l'animació de mort s'ha volgut transmetre el comportament de no voler rendir-se, ja que abans de caure derrotat intenta mantenir-se en peu.

Amb les dos animacions aleatòries d'idle es pot veure també un caràcter despreocupat, això és com a conseqüència del sentiment de seguretat, aquest fa que es preocupi molt difícilment.

Vestimenta

El personatge es vesteix amb una samarreta blanca sense mànigues, a sobre d'aquesta porta una jaqueta de color negre de màniga llarga fins els canells i d'alçada fins a sobre del genoll. Du uns pantalons llargs marrons lligats amb un cinturó de cuir amb sivella metàl·lica.

Com a calçat porta unes botes altes de pell tintades d'un color blau fosc gairebé negre amb la punta d'un material semblant a la goma de color vermell i amb dos tires a cada bota d'un blau més clar que el de les botes que acaben en una peça metàl·lica per ajustar-les a les cames.

Al voltant del coll porta una bufanda de color vermell feta amb una tela comfortable i suau.

Per últim el personatge porta una espasa d'un sol tall d'un metre de llarg, aquest és irregular siguent prim a la part més propera al mànec i ample a la zona central i la punta. La fulla de l'espasa és d'un metall negre, mentre que el mànec és de fusta. Cal tenir en compte que la jaqueta al costat esquerra a l'alçada de la cintura està teixida amb una tela especial amb propietats imantades, d'aquesta manera aparenta que l'espasa s'aguanta sola.

Figura 23: Imatge del disseny final del personatge.

Objectes característics

Els objectes que caracteritzen el personatge són la bufanda vermella record que va rebre de la seva família en emprendre el viatge i l'espasa tant irregular que utilitza per lluitar.

5.3.4. Narrativa

Per aquest projecte no s'ha desenvolupat cap element narratiu més enllà dels presentats pel que fa el personatge. En el cas, però, que es volgués desenvolupar un videojoc complet a partir del treball fet sí s'haurien d'incloure elements narratius ja que són un element important en els jocs d'estil RPG.

5.3.5. Interfícies

El joc en desenvolupar-se en una escena de prova no compta amb elements informatius. No obstant això, en un joc complet s'hauria de donar informació de la vitalitat del personatge, les monedes que es té, la vida dels enemics i altres informacions importants.

El que s'ha fet en el joc desenvolupat és un canvi de música quan el personatge s'apropa a la zona on es troba l'enemic, d'aquesta manera a més de veure com desenveina l'espasa hi ha un canvi sonor per avisar el jugador. L'enemic es pot distingit també en el seu aspecte totalment negre i en les partícules del mateix color que li surten del cos.

5.3.6. Ambient

Com ja s'ha dit abans l'escenari que es vol fer és aquell d'un escenari de proves, per aquest motiu només consta de 3 estructures i un enemic. La primera de les estructures és un conjunt de dos rampes, per veure el comportament en aquestes.

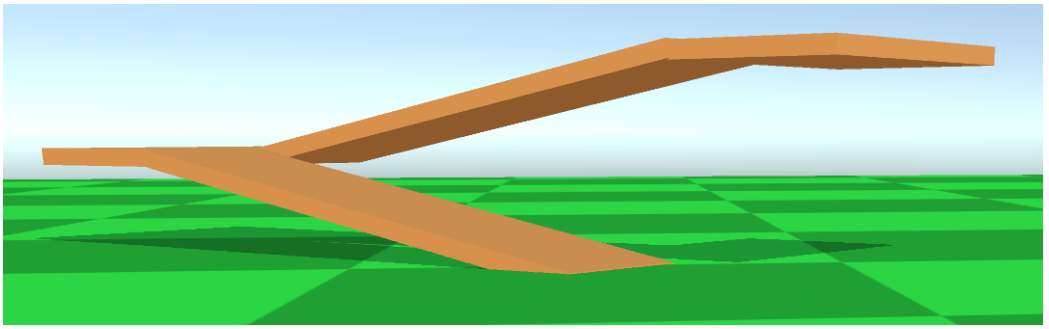


Figura 24: Estructura 1

Amb la segona estructura es pot veure el comportament del *Character Controller* en unes escales, les vermelles serien les escales incòmodes per la jugabilitat, ja que en pujar i baixar la càmera fa un moviment molt brusc. Les blaves serien més aptes ja que tenen un pla per sobre que fa de rampa, el fet de no renderitzar el pla dona encara la sensació d'escales.

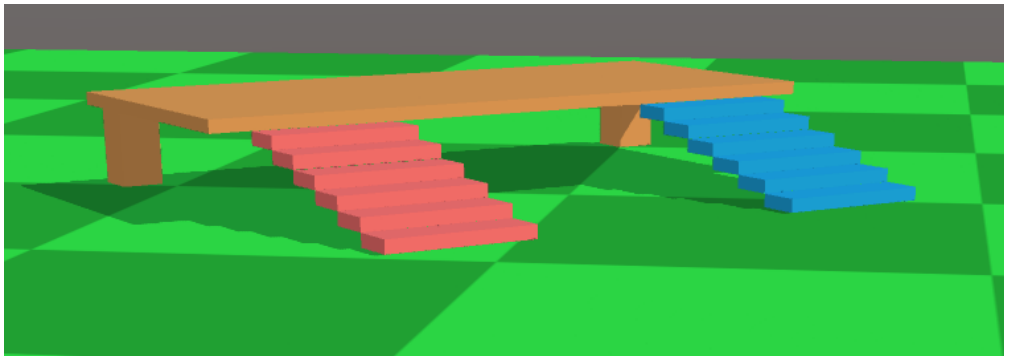


Figura 25: Estructura 2

Per últim la tercera estructura és una zona per provar el salt.

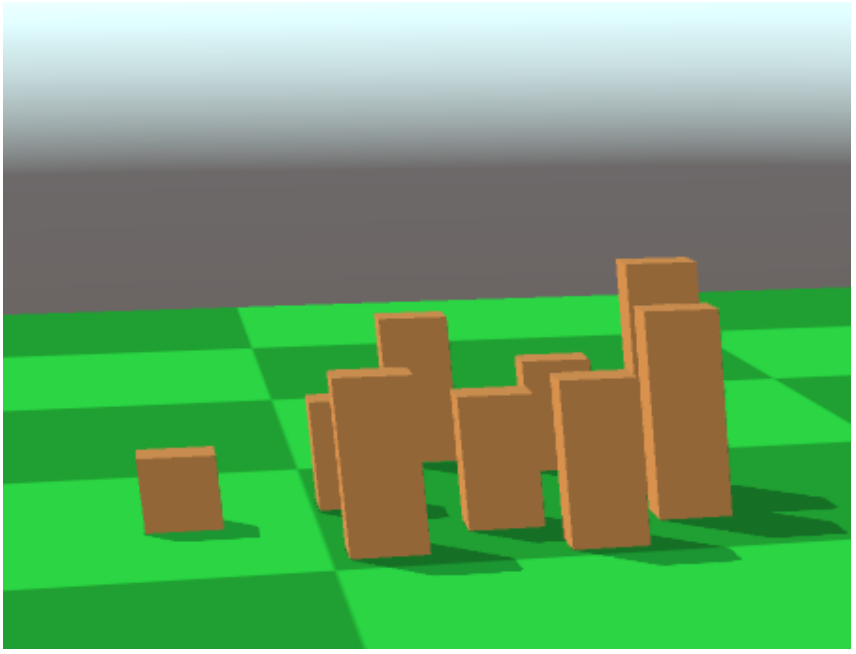


Figura 26: Estructura 3

La quarta zona és on podem trobar l'enemic per provar les mecàniques del combat.

6. IMPLEMENTACIÓ I PROVES

En ser aquest un projecte amb una forta component artística acompanyat d'alguns aspectes tècnics es presentarà primerament de manera detalla el procediment seguit per la creació del personatge i tot seguit la i integració del mateix i els elements tècnics.

6.1. Personatge

El procediment seguit per desenvolupament del personatge ha estat el següent.

6.1.1. Pluja d'idees

El primer pas per la creació del personatge era definir que es volia com a protagonista. No va ser molt difícil escollir fer un noi jove com a personatge.

Un cop escollit aquest aspecte es van dibuixar diferents propostes de personatge, tant facials amb diferents pentinats i característiques com corporals, de constirució similar totes però amb diferents dissenys de roba.

Es van fer també esbossos de l'arma del personatge.

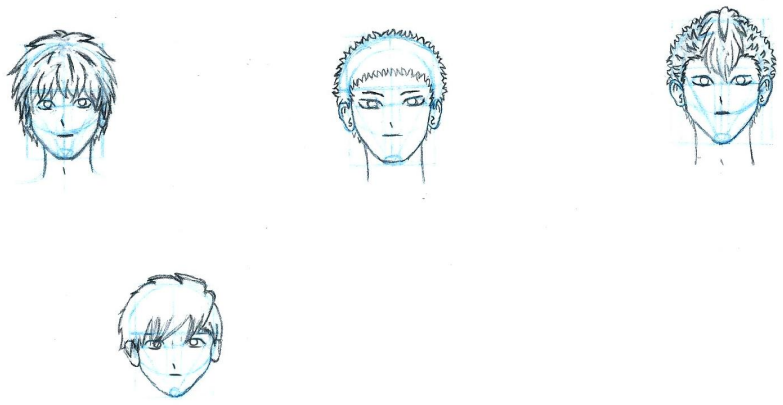


Figura 27: Primers dissenys de cares i pentinats

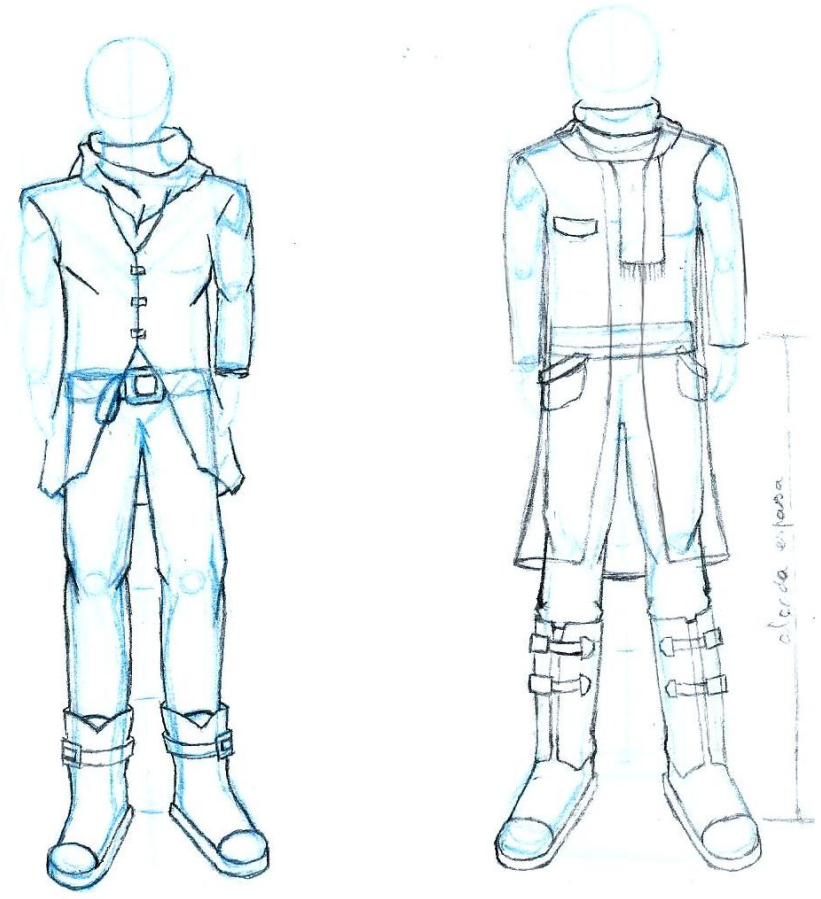


Figura 28: Primers dissenys de roba

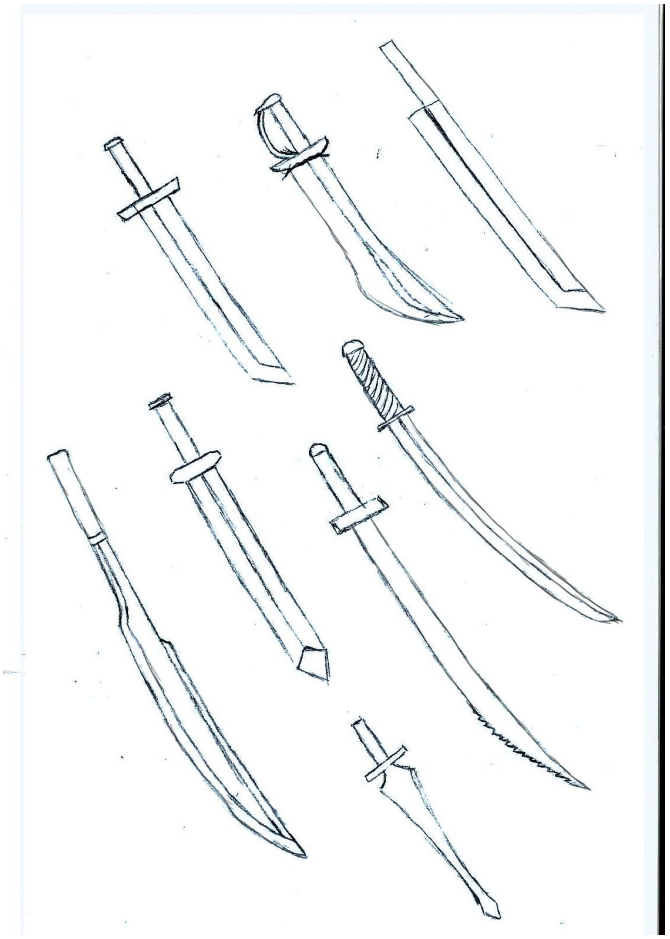


Figura 29: Primers dissenys d’espases

6.1.2. Disseny final

Amb els diferents dissenys proposats es va procedir a triar el disseny final. En aquest cas aquest consisteix en una de les 4 cares proposades i amb una combinació de roba dels dos dissenys primers.

Es va escollir també una de les espases com a definitiva.

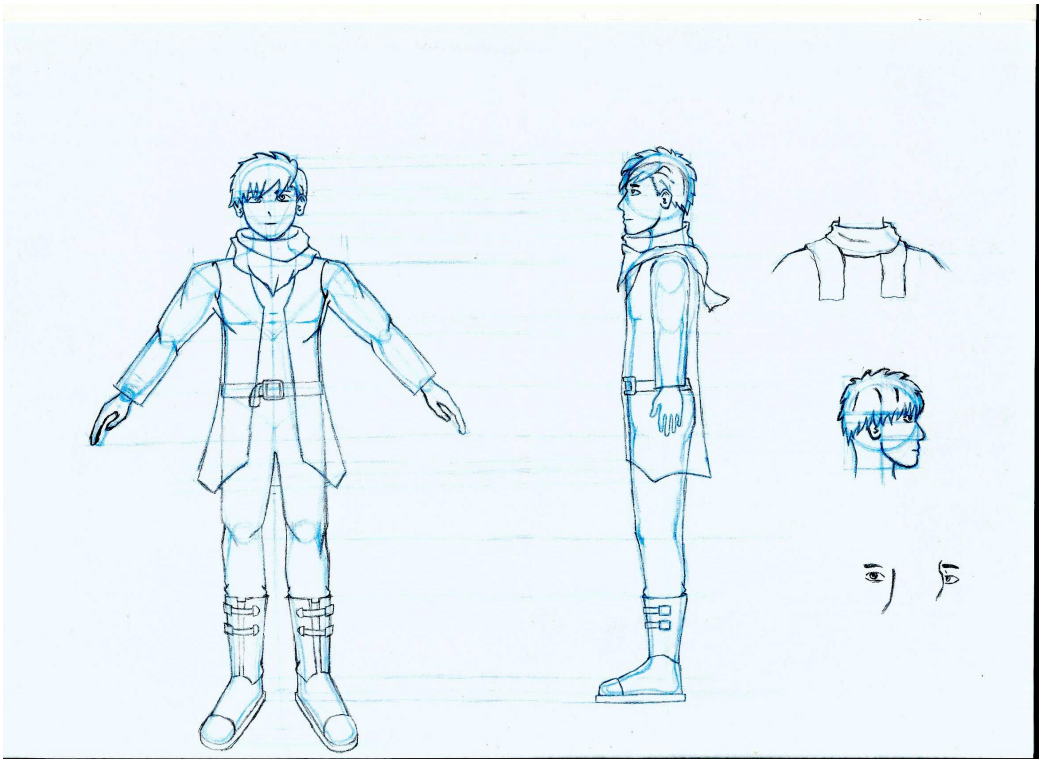


Figura 30: Vista frontal i lateral del disseny final junt amb detalls més concrets d'altres vistes

Com es pot veure en la figura anterior (Figura 30) la roba del personatge és una combinació dels primers dissenys proposats (Figura 28) on s'ha escollit la bufanda i el cinturó del disseny esquerra i les botes del dret, pel que fa la jaqueta és una combinació de les dos proposades on la llargada i acabat inferior són els del disseny esquerra mentre que les mànigues i el fet que no tingui botons són del disseny dret, tampoc té butxaques com en el cas de l'esquerra. Per últim s'ha decidit que dugués una samarreta com en el disseny dret, ja que en l'esquerra no en porta.

6.1.3. Digitalització

Un cop fet el disseny final, es va digitalitzar les diferents vistes per tal de tenir una línea de dibuix més clara, aquest fet ajudaria més endavant en el procés de modelat.

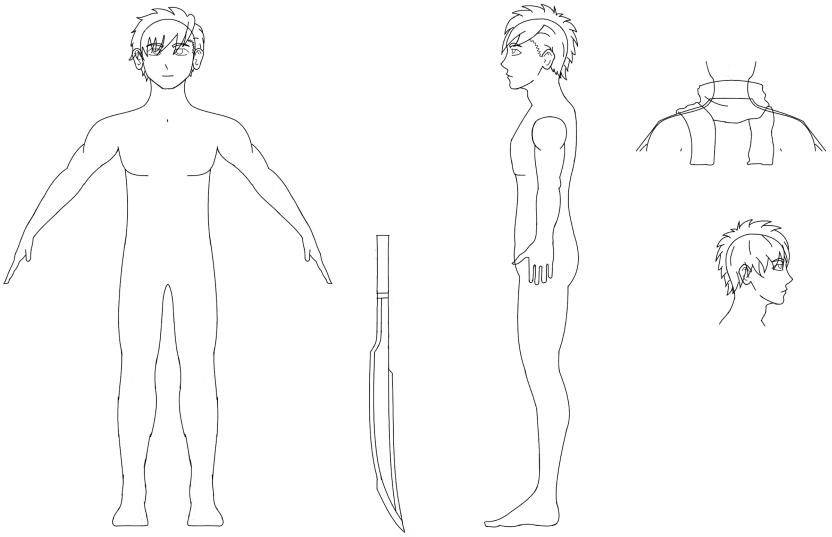


Figura 31: Digitalització del cos del personatge

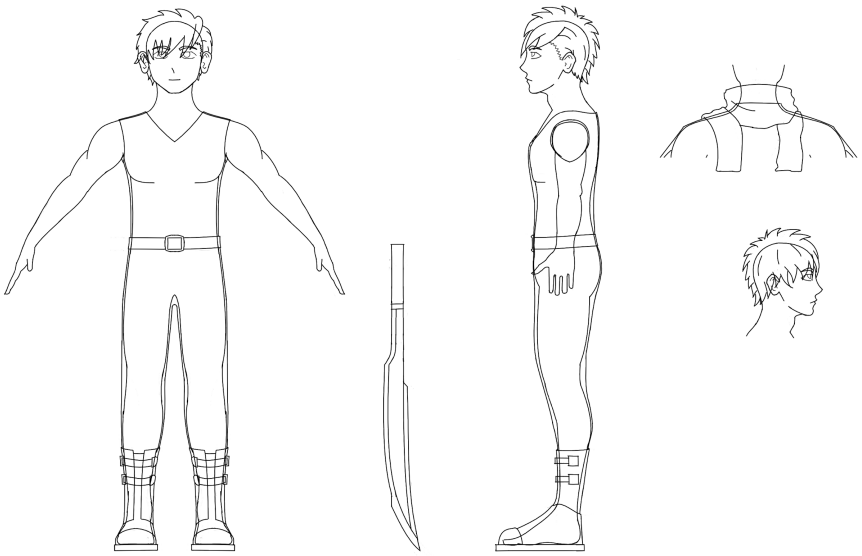


Figura 32: Digitalització de la capa interna de roba

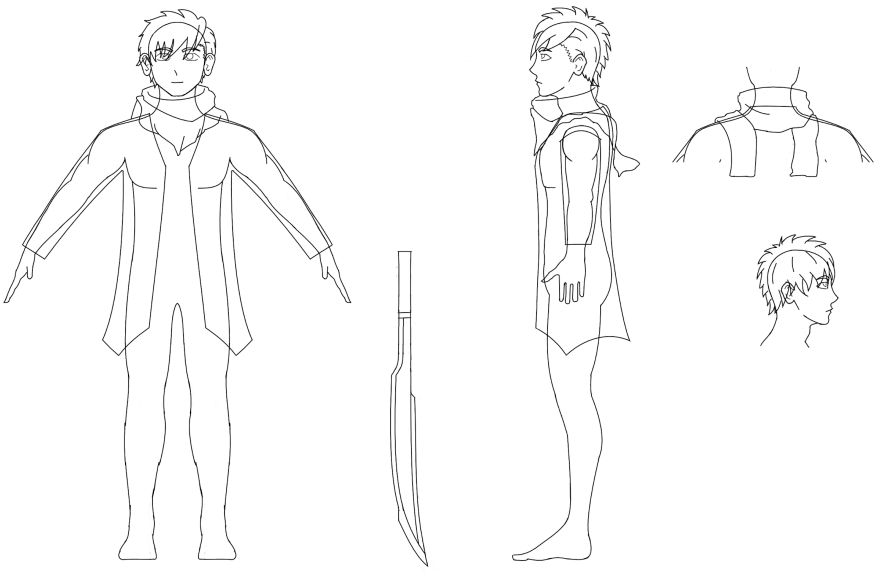


Figura 33: Digitalització de la capa externa de roba

6.1.4. Coloració

Per fer la coloració d'una manera senzilla es va utilitzar el sistema de màscares de colors del programa Krita, aquest sistema permet amb només una línea del color desitjat pintar l'espai fins el contorn, semblant a l'eina del cubell de pintura però sense necessitat que le líneas es trobin en la mateixa capa o totalment tancades.

Tot i això l'eina no és del tot fiable i a vegades falla, però per donar la idea dels colors que es vol utilitzar és prou util per el projecte.

Figura 34: Personatge colorejat

6.1.5. Modelat

A continuació es va dur a terme el modelat, en aquest procés primer es modela una versió amb pocs polígons.

Aquest procediment consisteix a crear el personatge amb poc nivell de detall.

Per facilitar el procés de modelat s'importa la imatge del personatge frontal i lateral dintre l'Autodesk Maya, d'aquesta manera es poden seguir les marques per fer un modelat més precís del disseny.

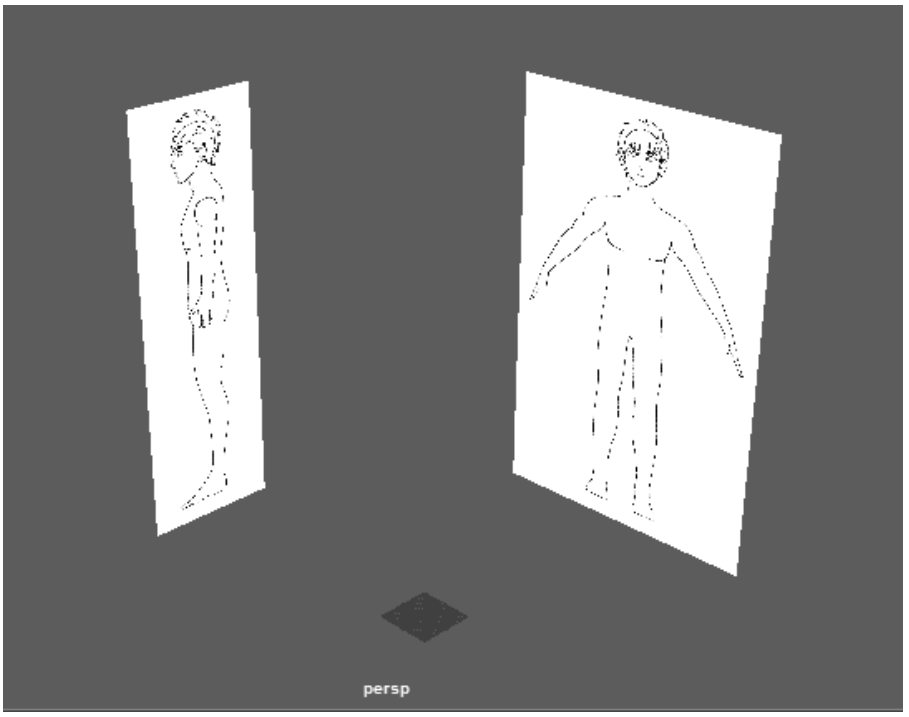


Figura 35: Espai de treball del modelat

El model final del personatge consta de 18604 triangles. Més endavant aquest model es suavitzarà de manera que tingui més polígons i es pugui fer més detall que s'utilitzarà en el mapa de normals.

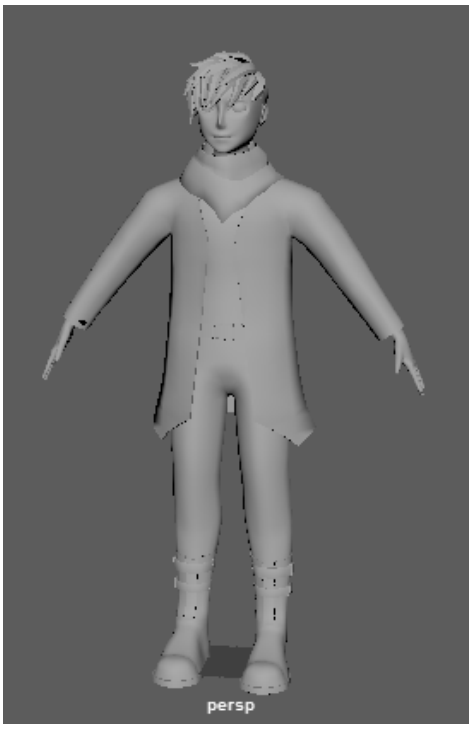


Figura 36: Model final del personatge

6.1.6. Texturitzat

Una vegada fet el model i per poder fer el rigging de manera correcte és necessari desplegar el mapa de uv's, ja que les influències dels ossos s'han de pintar i tenen com a base les uv's del model.

Aquestes uv's seran també les que s'utilitzaran en el procés de pintar les textures.

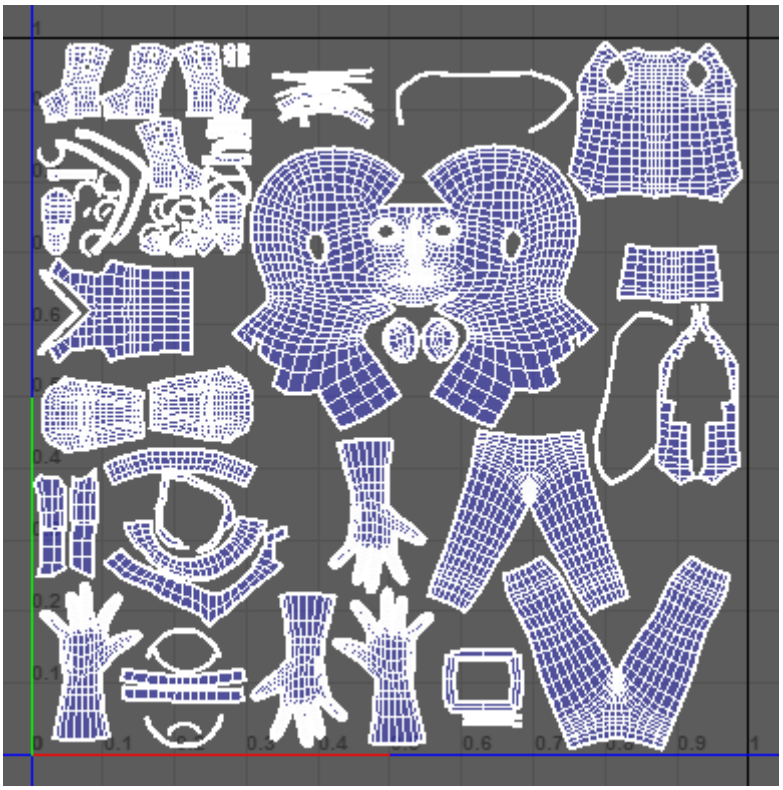


Figura 37: Mapa d'uv's del cos del personatge

Una vegada fetes les uv's del model es va procedir a augmentar el número de polígons del model d'aquesta manera el desplegat es manté en la nova versió.

Tot seguit es van exportar les dos versions per importar la de baixa resolució al Substance Painter i utilitzar la d'alta resolució per aconseguir els diferents mapes (normals, oclusió ambiental, etc.).

Figura 38: Model texturitzat al Substance Painter

Per fer les textures es van utilitzar materials disponibles al programa i es van modificar alguns paràmetres per ajustar-se a l'estil que es volia aconseguir.

6.1.7. Rigging

A continuació es va procedir a construir l'esquelet del model, aquest és necessari per moure la malla del model.

Una vegada fet l'esquelet de manera correcta s'han de pintar les influències per aconseguir una correcta deformació de la malla.

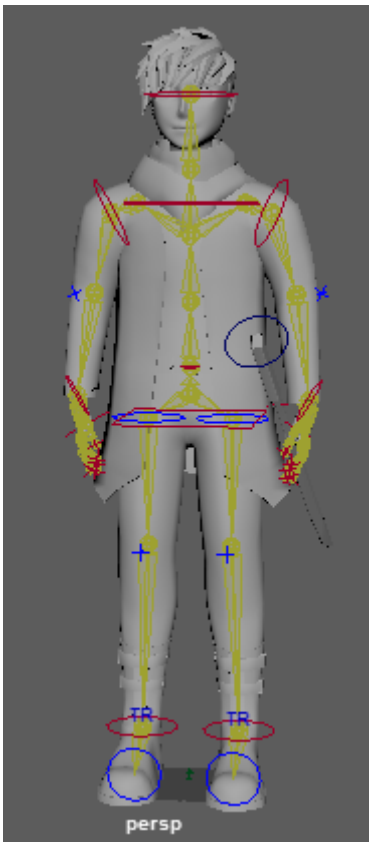


Figura 39: Rigging finalitzat del personatge

Per fer el rigging es va utilitzar el Human IK de l'Autodesk Maya. Aquesta eina dona un esquelet humà de base, reduint així el procediment a situar al lloc correcte els ossos.

6.1.8. Animacions

Finalment després del rigging arriba el procés central del projecte. Fer les diferents animacions que necessita el personatge. Les animacions realitzades han estat les següents.

Caminar

La part més important d'aquesta animació són les cames. Les cames han de seguir un cicle tancat per poder reproduir l'animació en bucle. Mentre que la part superior del cos s'utilitza per transmetre l'actitud del personatge.

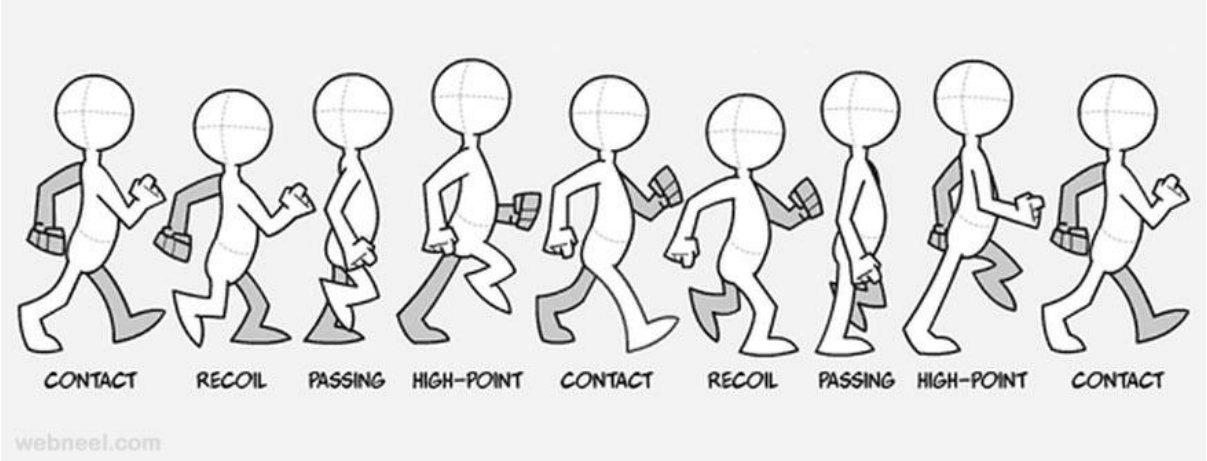


Figura 40: Imantge de referencia utilitzada per fer el cicle (https://webneel.com/walk-cycle-animation)

Correr

Per aquesta es va utilitzar de base l'animació de caminar, ja que també consisteix en un cicle de les mateixes característiques.

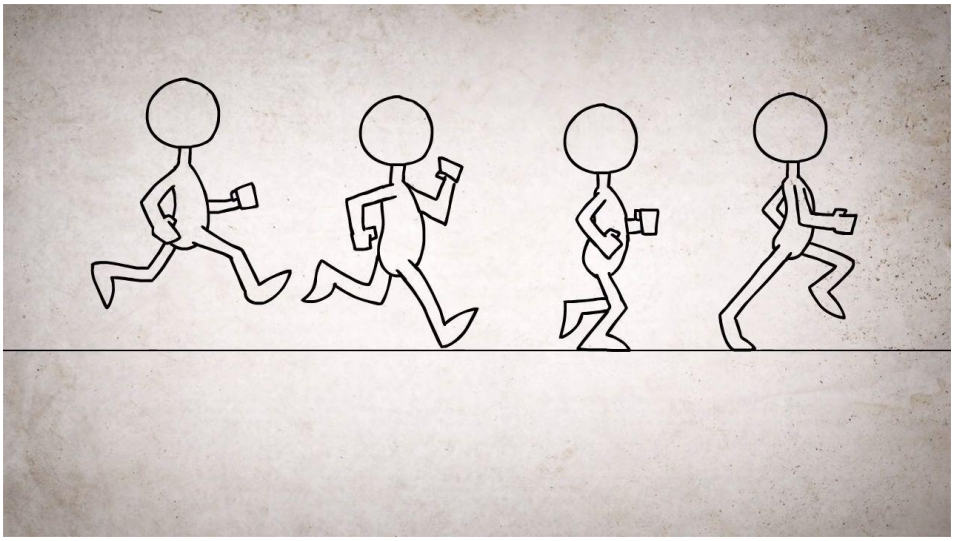


Figura 41: Imatge de referencia utilitzada per el cicle de correr
(https://www.pinterest.es/pin/863776403500428997/)

Idle

L'idle és l'animació que fa el personatge quan està quiet en un lloc.

De l'idle hi ha dos versions, una per els moments fora de batalla més tranquila, de moviments suaus gairebé imperceptibles. I la segona per els moments de combat, aquesta és més activa transmetent així el fet d'estar en una situació poc amigable.

L'idle, igual que passa amb el caminar i el correr ha de ser una animació que es pugui posar en bucle, ja que mentre el jugador estigui quiet s'haurà d'anar reproduint.

Idle alternatiu

S'han fet dos animacions alternatives. Aquestes són produïdes per trencar la monotonia de l'idle, es reprodueix una de les dos de manera aleatoria cada 10 segons.

Aquestes són accions que es desenvolupen una vegada trencant el bucle de l'idle i tornant a ell una vegada finalitzades.

Salt

Moviment bàsic a la majoria de videojocs. Aquest es divideix en 4 parts, anticipació on el personatge agafa impuls, pujada on el personatge es mou amb un impuls vertical, caiguda on el personatge cau de retorn a terra per efecte de la gravetat i aterratge on el personatge s'ajup per efecte de la força aturada.

La caiguda i l'aterratge del joc es poden utilitzar també per quan el personatge cau des de un punt elevat al terra.

Envainar i desenvainar

Accions contràries entre elles. Per facilitar l'animació es podia haver optat per utilitzar la mateixa animació reproduïda endavant o enrere segons convingui. Però es va optar per dos animacions diferents, donant d'aquesta menera més personalitat al personatge.

Atac

L'acció d'atacar consta d'una combinació de fins a tres moviments. Començant sempre en el primer moviment es pot decidir si fer-ne només un, fer-ne dos o completar els tres moviments de l'atac.

Bloquejar

Acció que permet bloquejar atacs posant l'espasa davant del personatge. Aquesta acció no permet el moviment i pot durar tant temps com el jugador vulgui.

Esquivar

Acció que permet esquivar un atac fent un moviment ràpid. Una vegada iniciada l'acció no es pot modificar la trajectòria fins que no acaba.

Rebre mal

Acció que es produeix automàticament quan el personatge rep un cop. En aquesta animació el personatge es retira lleument cap enrere tornant de seguida a la posició d'atac.

Morir

Quan el personatge es queda sense vitalitat mor. Com s'ha dit anteriorment en el disseny de personatge (veure punt 5) aquesta acció consisteix en el personatge intentant mantenir-se dempeus fins que inevitablement cau a terra.

Caminar i córrer amb espasa

Aquestes animacions s'han realitzat utilitzant les màscares de Unity, permetent d'aquesta manera combinar el moviment de l'animació de caminar o córrer normals amb el moviment de la posició base de lluita.

Per totes les animacions s'ha utilitzat alguna referència normalement en forma de vídeo, ja fos buscat a internet o gravat a casa per destacar el moviment buscat.

6.2. Implementació

A continuació es veurà el procediment seguit a la implementació. I els problemes sorgits durant aquesta.

6.2.1. Personatge

El personatge és l'element principal del projecte i com a tal moltes de les accions programades recauen en ell.

Per començar es va fer el moviment del personatge, aquest es fa llegint l'imput del jugador i multiplican-lo per la velocitat utilitzant la funció *Move()* del *CharacterControlller*. El moviment està limitat a 1 ja que si no es fa això en el moviment diagonal es va més depresa perque es sumen els valors dels imputs dels dos eixos. A més el moviment s'ha fet relatiu a la càmera d'aquesta manera si li donem enrere el personatge es girarà i caminarà cap a la càmera.

Hi ha diferents condicions per moure el personatge, la primera i més important que el personatge encara estigui viu, la segona és que no hagi estat colpejat, que sigui a terra i per últim que no estigui esquivant.

El moviment es calcula des de l'Update ja que s'ha de mirar a cada frame si el personatge es mou.

```
void Update () {
    if (dead) {
        if (!dying) {
            StartCoroutine(death());
        }

    } else {
        if (!hit && player.isGrounded) {
            horizontalMove = Input.GetAxis("Horizontal");
            verticalMove = Input.GetAxis("Vertical");
        }

        if (!drawn) {
            sword.transform.SetParent(swordSheathed);
        } else {
            sword.transform.SetParent(swordDrawn);
        }

        playerInput = new Vector3(horizontalMove, 0, verticalMove);
        playerInput = Vector3.ClampMagnitude(playerInput, 1); //This clamps the value to 1 avoiding greater values when going diagonally

        playerAnimatorController.SetFloat("playerWalkVelocity", playerInput.magnitude * presentSpeed);
        if (!dash) {
            if (horizontalMove < 0 || horizontalMove > 0 || verticalMove < 0 || verticalMove > 0) {
                canDash = true;
            } else {
                canDash = false;
            }

            camDirection();

            movePlayer = playerInput.x * camRight + playerInput.z * camForward;
            movePlayer *= presentSpeed;

            player.transform.LookAt(player.transform.position + movePlayer);

            setGravity();

            if (!hit) {
                playerSkills();
            }

            player.Move(movePlayer * Time.deltaTime);

        } else {
            StartCoroutine(Dash());
        }
    }
}
```

Figura 42: Codi del moviment del personatge entre altres funcions

Tot seguit trobem les habilitats del jugador que consisteixen en saltar si està tocant a terra, caminar també si està tocant al terra, i en cas d'estar en combat bloquejar o esquivar depenent si el personatge es mou o no i atacar.

```
public void playerSkills () {
    if (player.isGrounded && Input.GetButtonDown("Jump")) {
        fallVelocity = jumpForce;
        movePlayer.y = fallVelocity;
        playerAnimatorController.Play("jumpCrouch");
        jumpSound();
        stopIdle();
    }

    if (player.isGrounded && Input.GetButton("Crouch")) {
        presentSpeed -= 0.05f;
        if (presentSpeed < crouchSpeed) {
            presentSpeed = crouchSpeed;
        }
    } else {
        presentSpeed += 0.05f;
        if (presentSpeed > playerSpeed) {
            presentSpeed = playerSpeed;
        }
    }

    if (combat && !canDash && Input.GetButton("block_dodge")) {
        presentSpeed = 0;
        blockDash = true;
        playerAnimatorController.SetBool("block", blockDash);

    } else if (combat && canDash && Input.GetButtonDown("block_dodge")) {
        initDash();
    } else {
        blockDash = false;
        playerAnimatorController.SetBool("block", blockDash);
    }

    if (combat && Input.GetButtonDown("attack")) {
        attack();
    }
}
```

Figura 43: Codi de les habilitats que pot executar el jugador

A continuació es pot veure el codi utilitzat per donar gravetat al personatge, el comportament canvia depenent de si s'està a l'aire o a terra.

```
public void setGravity () {
    if (player.isGrounded) {
        fallVelocity = -gravity * Time.deltaTime;
        movePlayer.y = fallVelocity;
        playerAnimatorController.SetFloat("playerVerticalVelocity", player.velocity.y);
    } else {
        fallVelocity -= gravity * Time.deltaTime;
        movePlayer.y = fallVelocity;
        playerAnimatorController.SetFloat("playerVerticalVelocity", player.velocity.y);
    }
    playerAnimatorController.SetBool("isGrounded", player.isGrounded);
    onRamp();
}
```

Figura 44: Codi per donar gravetat

A la figura següent (figura 45) es poden veure les funcions implicades a l'acció d'esquivar. Primerament es crida una funció que posa a punt les variables necessàries per executar el moviment. Seguit d'una corrutina que indica el temps que el moviment s'està executant.

```
private void initDash () {
    blockDash = true;
    dash = true;
    currentDashTime = 0;
    playerAnimatorController.SetBool("dodge", dash);
    soundsManager.playDash();
}

IEnumerator Dash () {
    float startTime = Time.time;

    while (Time.time < startTime + maxDashTime) {
        player.Move(movePlayer * dashVelocity * Time.deltaTime);
        playerAnimatorController.SetBool("dodge", true);
        yield return null;
    }

    yield return null;

    blockDash = false;
    dash = false;
    playerAnimatorController.SetBool("dodge", blockDash);
}
```

Figura 45: Codi de l'acció d'esquivar

Tot seguit es poden veure les funcions per atacar i fer una combinació d'atacs. Aquestes funcions es criden mitjançant *Animation events* exceptuant el primer moviment que es crida estant en combat i quan l'usuari apreta el botó corresponent.

```
public void attack () {
    if (comboCount == 0) {
        playerAnimatorController.Play("combo1");
        comboCount = 1;
        return;
    } else {
        if (comboPossible) {
            comboPossible = false;
            comboCount += 1;
        }
    }
}

public void ComboPossible () {
    comboPossible = true;
}

public void combo () {
    if (comboCount == 2) {
        playerAnimatorController.Play("combo2");
    }

    if (comboCount == 3) {
        playerAnimatorController.Play("combo3");
    }
}

public void comboReset () {
    comboPossible = false;
    comboCount = 0;
}
```

Figura 46: Codi de l'acció d'atacar

A continuació dues funcions que donen una mica d'estètica al projecte. Es criden des de animation events i activen o desactiven les partícules de l'arma segons s'entra o surt del combat.

```
private void activateParticles () {
    swordParticles.SetActive(true);
}

private void deactivateParticles () {
    swordParticles.SetActive(false);
}
```

Figura 47: Codi d'activació i desactivació de partícules

Es pot veure a continuació les funcions que permeten trencar el bucle del idle i cridar una de les altres dos animacions aleatòriament. Aquest codi consisteix en una corrutina que es crida una vegada cada 10 segons. El que fa llavors és triar una animació de les dos disponibles aleatòriament i reproduir-la. Hi ha també dos funcions que s'executen amb animation events i paren o inicien la corrutina segons si el personatge es mou o es torna a quedar quiet.

```
IEnumerator idle () {
    idleing = true;
    yield return new WaitForSeconds(10);
    pickIdle();
    idleing = false;
}

private void pickIdle () {
    var numIdle = Random.Range(1, 3);
    if (numIdle == 1) {
        playerAnimatorController.Play("idle2");
    } else {
        playerAnimatorController.Play("idle3");
    }
}

public void startIdle () {
    if (!idleing && !combat) {
        doIdle = StartCoroutine(idle());
    }
}

public void stopIdle () {
    StopCoroutine(doIdle);
    if (idleing) {
        idleing = false;
    }
}
```

Figura 48: Codi d'activació d'animació especial

Les dos funcions a continuació serveixen per indicar a quina posició ha d'estar emparentada l'espasa, això és degut a que fora de combat l'espasa es situa a la cintura i es conduïda i moguda per aquesta, però en combat es busca que l'espasa estigui "enganxada" a la mà del personatge.

```
public void drawnSword () {
    drawn = true;
    swordParticles.SetActive(drawn);
}

public void sheathedSword () {
    drawn = false;
    swordParticles.SetActive(drawn);
}
```

Figura 49: Codi que activa les emparentacions de l'espasa

A continuació es pot veure la corrutina que crida el menú de fi de partida al morir. S'ha decidit utilitzar una corrutina per fer que el joc esperi uns segons abans de canviar de pantalla, d'aquesta manera l'usuari té temps d'entendre que ha passat.

```
IEnumerator death () {
    dying = true;
    yield return new WaitForSeconds(deathTime);
    SceneManager.LoadScene("GameOver");
    dying = false;
}
```

Figura 50: Codi per canviar al menú de fi de partida

Tot seguit es poden veure unes funcions que detecten si el personatge està en una rampa, això es va fer degut a que en baixar una rampa el personatge reproduia l'animació de caure ja que en baixa una rampa no es detecta sempre la col·lisió amb el terra.

```
public void onRamp () {
    ramp = Vector3.Angle(Vector3.up, normal) < player.slopeLimit;
    if (Vector3.Angle(Vector3.up, normal) == 0 || player.velocity.y < -2f) {
        ramp = false;
    }
    playerAnimatorController.SetBool("ramp", ramp);
}

private void OnControllerColliderHit (ControllerColliderHit hit) {
        normal = hit.normal;
}
```

Figura 51: Codi per detectar si s'està a una rampa

A continuació es troben funcions per cridar sons mitjançant animation events.

```
public void footstepSound () {
    soundsManager.playStep();
}

public void sowrdSound1 () {
    soundsManager.playSword1();
}

public void sowrdSound2 () {
    soundsManager.playSword2();
}

public void sowrdSound3 () {
    soundsManager.playSword3();
}

public void jumpSound () {
    soundsManager.playJump();
}

public void landSpound () {
    soundsManager.playLand();
}
```

Figura 52: Codi per cridar sons

Per últim hi ha la detecció d'entrada i sortida de la zona de l'enemic. Entrar en aquesta zona implica entrar en mode combat permetent al jugador provar totes aquelles mecàniques restringides fora d'aquest.

```
private void OnTriggerEnter (Collider c) {
    if (c.tag == "Enemy") {
        combat = true;
        playerAnimatorController.SetBool("fighting", combat);
        audioManager.changeBGM(battleBGM);ç
    }
}

private void OnTriggerExit (Collider c) {
    if (c.tag == "Enemy") {
        combat = false;
        playerAnimatorController.SetBool("fighting", combat);
        audioManager.changeBGM(peaceBGM);
    }
}
```

Figura 53: Codi per detectar la zona de combat

6.2.2. Enemic

Pel que fa l'enemic no és massa complexe, tot el que fa és atacar cada 5 segons activant i desactivant un col·lisionador situat devant seu per saber si li ha donat al jugador tot això ajudat per dos animation events que indiquen quan s'està atacant i quan es para de fer-ho.

```
void Start() {
    enemyAnimatorController = GetComponent<Animator>();
    enemyCollider = GetComponent<BoxCollider>();
}

void Update() {
    if (!wait) {
        StartCoroutine(attack());
    }
}

private void OnTriggerEnter (Collider c) {
    if (c.tag == "Player" && attacking) {
        c.GetComponent<DamageController>().takeDamage(damage);
        Debug.Log("attacked");
    }
}

private void playAnimation () {
    enemyAnimatorController.SetTrigger("attack");
}

IEnumerator attack () {
    wait = true;
    playAnimation();
    yield return new WaitForSeconds(waitAttack);
    wait = false;
}

public void Attacking () {
    this.GetComponent<BoxCollider>().enabled = true;
    attacking = true;
}

public void NotAttacking () {
    this.GetComponent<BoxCollider>().enabled = false;
    attacking = false;
}
```

Figura 54: Codi de l'enemic

6.2.3. Control de danys al personatge

És l'encarregat de la gestió de la vida del personatge, quan rep un atac i no estava bloquejant es resta un punt de vida i en cas de morir crida una funció per alentir el temps. També dóna durant un temps curt invencibilitat al jugador i el fa parar de moure's.

```
void Start() {
    playerAnimator = GetComponent<Animator>();
    controller = GetComponent<PlayerController>();
}

public void takeDamage (int damage) {
    if (!invencibility && health > 0 && !controller.blockDash) {
        health -= damage;
        if (health <= 0) {
            controller.dead = true;
            playerAnimator.Play("death");
            soundsManager.playDeath();
            timeManager.slowDown();
        } else {
            playerAnimator.Play("pain");
            soundsManager.playPain();
            StartCoroutine(invencibilityMoment());
            StartCoroutine(stop());
        }

    } else if (!controller.dash) {
        soundsManager.playBlock();
    }
}

IEnumerator invencibilityMoment () {
    invencibility = true;
    yield return new WaitForSeconds(invencibilityTime);
    invencibility = false;
}

IEnumerator stop () {
    controller.hit = true;
    yield return new WaitForSeconds(stopTime);
    controller.hit = false;
}
```

Figura 55: Codi del manejador de mal

6.2.4. Gestor de temps

Aquesta funció va ser creada per donar un efecte de càmera lenta en el moment de morir similar al moment de mort en els jocs de la saga Kingdom Hearts.

```
void Start() {
    Time.timeScale = 1;
}

public void slowDown () {
    Time.timeScale = slowedDown;
    Time.fixedDeltaTime = Time.timeScale * 0.02f;
}
```

Figura 56: Codi del manejador de temps

6.2.5. Menús

Per acabar hi ha els menús. El menú principal i el de fi de partida comparteixen codi, ja que les seves funcions són exactament igual, entrar al joc o sortir d'aquest. Mentre que el menú de pausa s'ha d'activar, desactivar i parar el temps mentre està activat.

```
public void play () {
    SceneManager.LoadScene("Game");
}

public void quit () {
    Debug.Log("quit");
    Application.Quit();
}
```

Figura 57: Codi menú principal

```
void Update() {
    if (Input.GetButtonDown("Cancel")) {
        if (paused) {
            resume();
        } else {
            pause();
        }
    }
}

public void resume () {
    pauseMenu.SetActive(false);
    Time.timeScale = 1f;
    paused = false;
    pj.playerAnimatorController.SetBool("isGrounded", !paused);
}

private void pause () {
    pauseMenu.SetActive(true);
    Time.timeScale = 0f;
    paused = true;
}

public void mainMenu () {
    SceneManager.LoadScene("MainMenu");
}
```

Figura 58: Codi del menú de pausa

6.3. Problemes i solucions

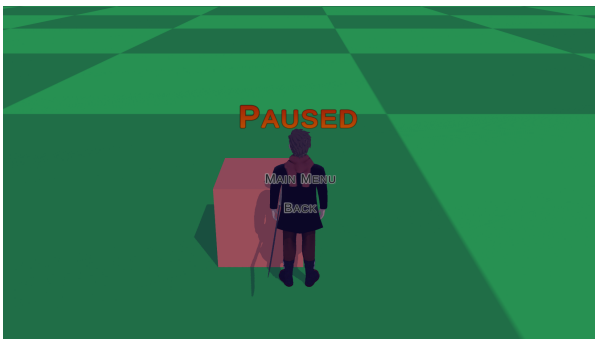
En aquest apartat es detallaran els problemes trobats al llarg de la implementació i les solucions adoptades per cadascun d'ells. Per facilitar l'enteniment, alguns dels errors es van gravar en vídeo, aquestes gravacions seran adjuntes al projecte (veure annex 2).

- A l'inici de la implementació vaig veure que la funció *isGrounded()* del *Character controller* fluctuava entre vertader i fals impedint executar la mecànica de salt. La solució es va trobar al forum de Unity on s'indicava que s'havia de posar el valor *Min Move Distance* a 0.
- El segon inconvenient va ser degut a les animacions. S'havia importat l'espasa animada junt amb el model fent que en utilitzar les màscares de Unity l'espasa quedés desplaçada en l'animació. Això es va solucionar fent que l'espasa i les animacions s'importessin per separat i emparentant l'espasa al lloc indicat segons la situació.
- En utilitzar el component *Cloth* per fer la roba i afegir els col·lisionadors per que no atrevasses el cos del personatge va resultar en que el personatge comencés a girar caòticament sobre si mateix això es devia a que el *Rigidbody* no tenia activades les restriccions de rotació.
- En fer el moviment d'esquivar entrava en un bucle infinit ja que en un principi la corrutina no retornava el control al codi. A més l'animació no es reproduia bé però es va arreglar reexportant i reimportant.
- Un altre petit problema és que les animacions importades són només de lectura, això vol dir que si es vol crear animation events s'han de copiar els fotogrames i enganxar en un clip d'animació nou.
- A més els animation events no es criden en l'últim frame de l'animació.
- En fer les diferents animacions d'idle el principal problema va ser el desconeixement de com fer i que utilitzar en el codi. Primer es van buscar solucions en la web, ja fos al forum o a les respostes de Unity. Després de donar-li voltes als diferents codis vistos i arribar a la conclusió que eren molt complicats i que no s'entenien fàcilment es va tenir la idea d'utilitzar animation events, aquests van resultar senzills i funcionals.
- En morir el personatge es movia tot sol, això es devia xocs entre els col·lisionadors de la roba i el rigidbody.

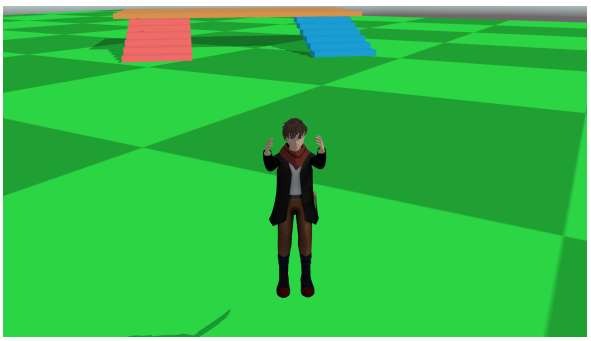
- Per últim cap al final del desnvolupament el moviment d'esquivar es va deixar d'executar es devia a que el rigidbody del personatge va deixar de detectar el moviment. Es creu que això va succeir en copiar i enganxar el component. Simplement esborrar el component i tornar-lo a col·locar no canviava res. En vistes de no trobar cap solució es va decidir fer la comprovació manualment i es va afegir una variable i una comparació per saber si és possible fer el moviment d'esquivar per part de l'input.

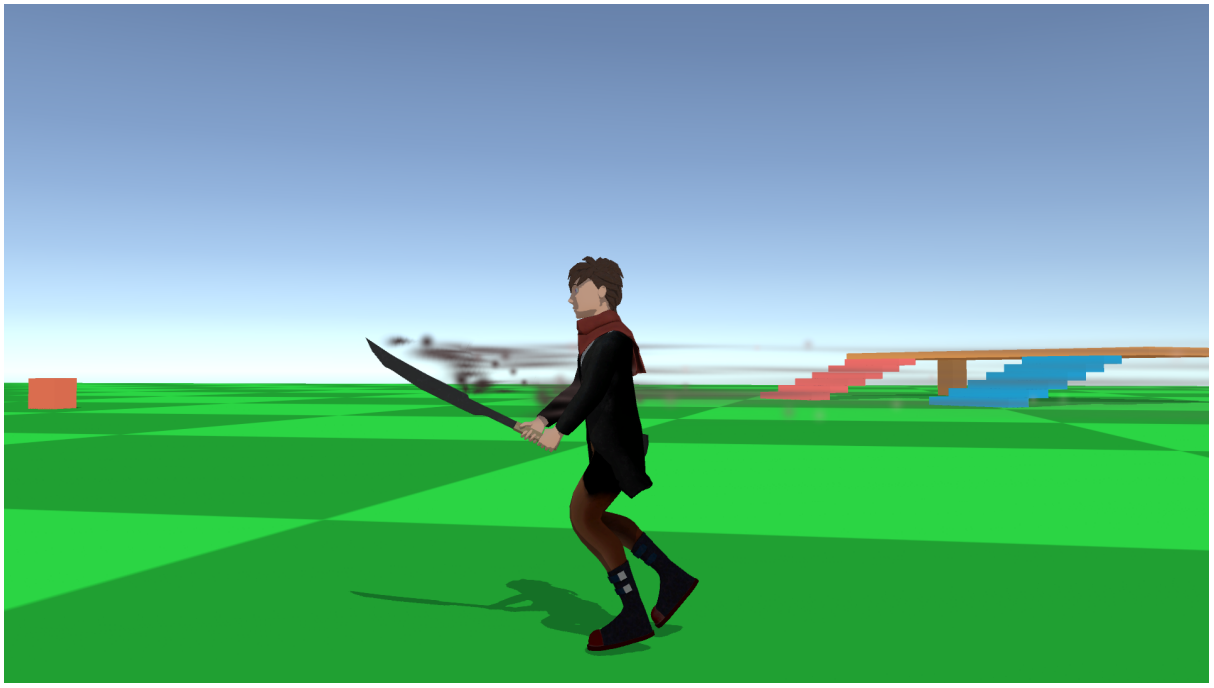
7. RESULTATS

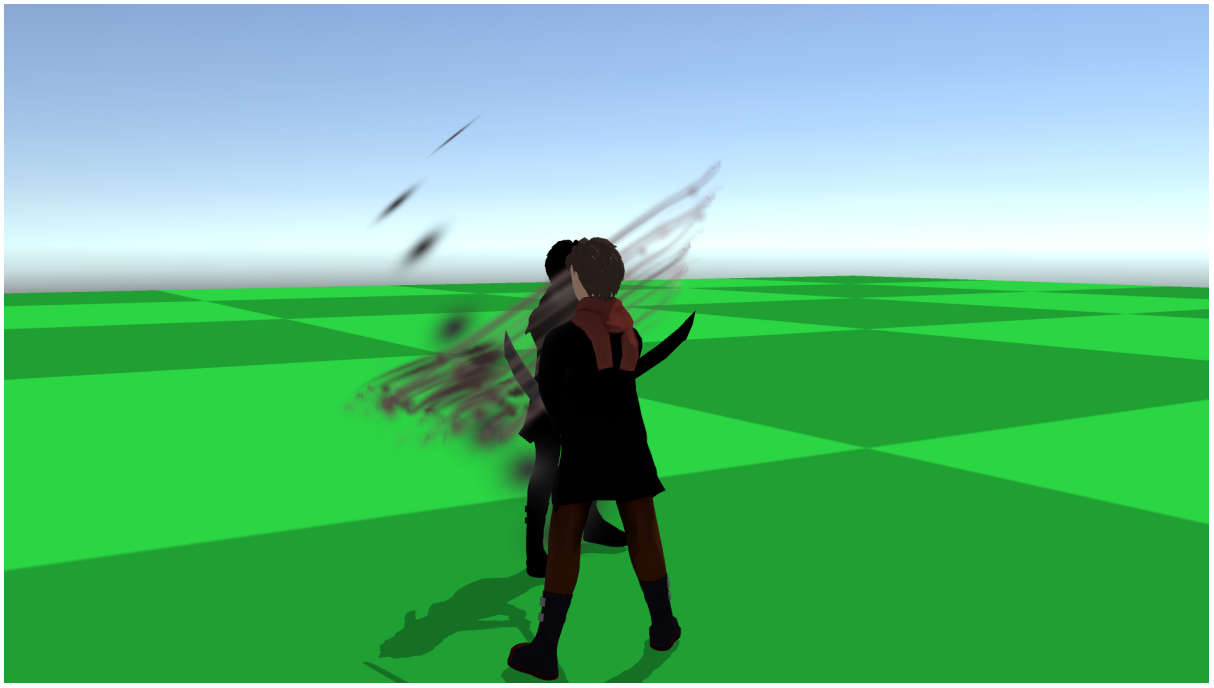
En aquest apartat es mostraran diferents captures de pantalla del joc per veure el resultat final del producte. Tot i això en ser un treball centrat en les animacions els resultats obtinguts no seran del tot apreciables per això s'ha decidit adjuntar l'enllaç a un video amb el resultats finals (veure annex 3).

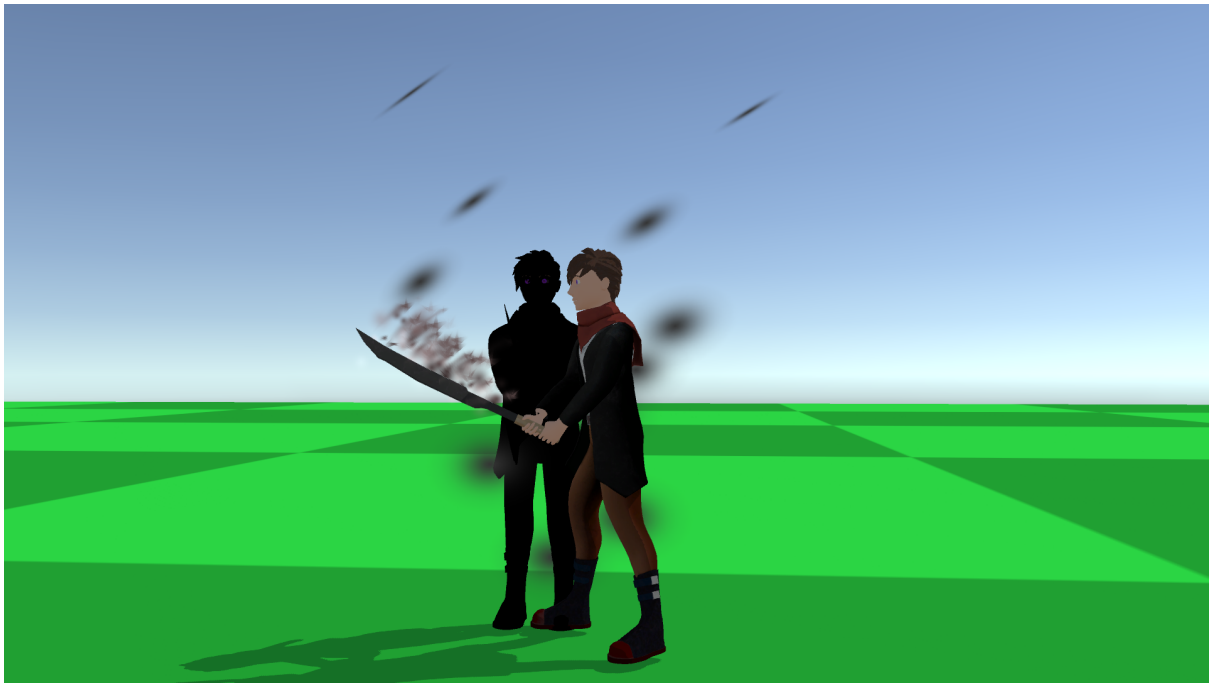


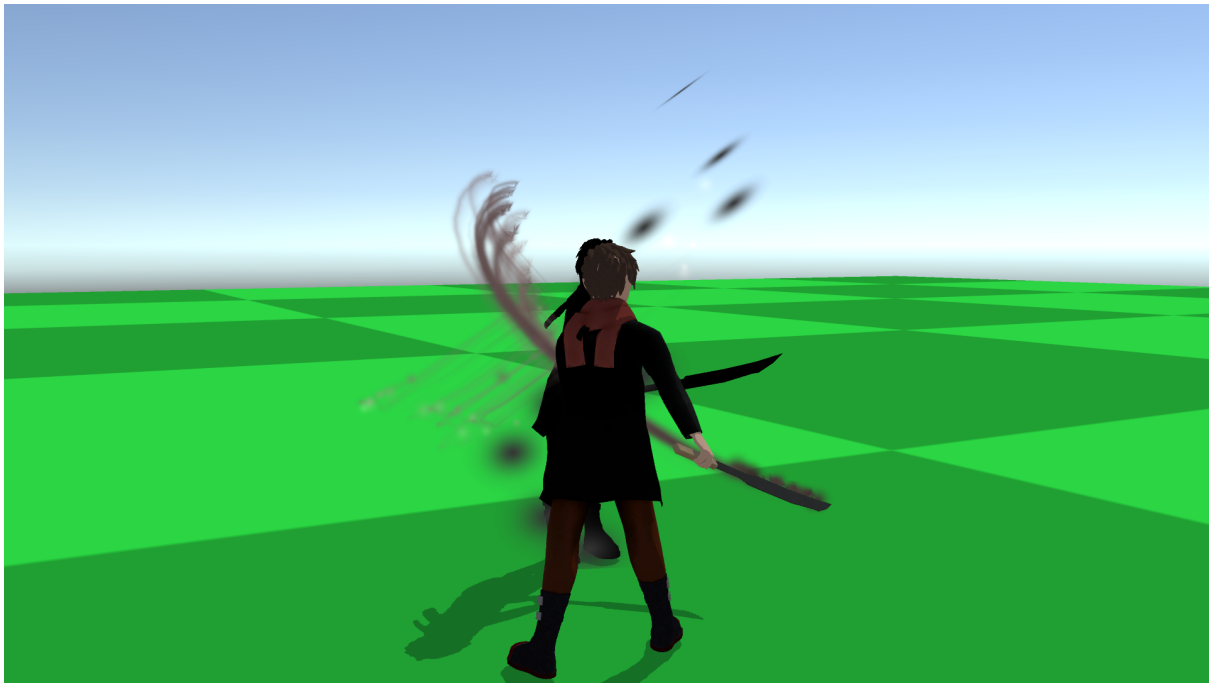

GAME OVER
RETRY
QUIT











8. CONCLUSIONS

8.1. Desviació de la planificació

Inicialment el treball estava ideat de manera que s'acabés al juny però degut a problemes mèdics, en concret un quiste al canell, que va impossibilitar el treballar en el projecte durant dos mesos, fent d'aquesta manera que no es pogués assolir la data prevista en una primera instància.

El cronograma final ha estat el següent, en vermell es veuen marcades les setmanes on no es va poder treballar.

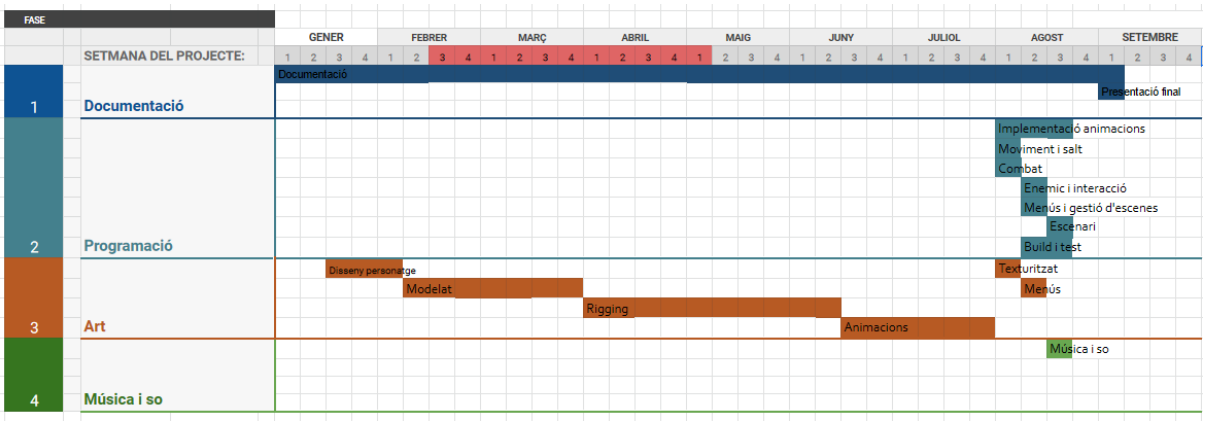


Figura 59: Cronograma final del projecte

8.2. Conclusions

En iniciar aquest projecte em vaig plantejar assolir uns objectius durant el desenvolupament del projecte i en acabar aquest. Ara després d'haver acabat tot el desenvolupament puc afirmar que he aconseguit arribar als objectius plantejats.

Gràcies a aquest treball he pogut experimentar el procés de creació d'un personatge en 3D per a un videojoc, no només això, si no que a més he pogut aprendre diversos aspectes desconeguts mentre el duia a terme.

Entre aquests he de destacar els *Animation Layers* de l'Autodesk Maya que permeten animar una part concreta sense que es vegi afectada o afecti a altres parts del model. També cal mencionar les màscares d'animació del Unity de les quals no en sabia res, només sabia que existia una manera de combinar animacions. Del Unity també he après sobre els *Animation events* que són esdeveniments de programació que es criden quan l'animació arriba a un cert frame.

Tots aquests coneixements no els hagués assolit si no hagués desenvolupat un projecte amb aquestes característiques.

A més aquest treball m'ha obligat a programar mecàniques, treballar-les i fer-les funcionar de manera correcte. Si es té en compte que la programació no és ni de prop el meu fort puc dir sense penedir-me que estic satisfeta amb el resultat final del desenvolupament.

Per acabar simplement vull agrair al meu tutor, en Xavier Costa, per l'ajut durant tot el procés del desenvolupament del treball. Agrair també als meus companys que m'han ajudat quan tenia dubtes i m'han ofert la seva ajuda si tenia algun problema. També agrair als meus pares que cada vegada que em veien treballant volien veure i saber que estava fent i que sense saber-ho m'han donat esperança quan creia que no ho podia acabar. Per últim una menció especial al meu gos, que va ser la única companyia i recolzament emocional que vaig tenir durant els moments més durs del desenvolupament en el mes d'agost quan estava sola.

9. TREBALL FUTUR

El joc desenvolupat és només una escena de proves i com a tal hi ha molt marge de millora o augment en la jugabilitat. En cas de que es volgués desenvolupar un videojoc a partir d'aquest treball s'haurien d'afegir diversos aspectes, una llista d'aquests seria:

- **Narrativa:** Actualment el treball no disposa de cap tipus de narrativa, per poder desenvolupar un joc s'hauria de crear de zero.
- **Mecàniques:** El nivell de desenvolupament actual dona peu a poder afegir més mecàniques, ja siguin de moviment com agafar-se i moure's per un sortint d'una paret o mecàniques de joc com diferents habilitats o l'ús d'objectes.
- **Música i sons:** Tot i que el treball disposa d'aquests, són millorables i ampliables. Amb un bon equip es podrien produir sons propis per el joc.
- **Interfícies:** Com s'ha dit al punt 5 el joc no disposa d'interfícies i aquestes són importants per informar al jugador.
- **Menús:** Com s'ha vist també al punt 5, els menús no estan del tot implementats. Això és degut a que no feien cap aportació a l'escena.

A nivell del desenvolupat en el treball es considera que el codi tot i ser funcional hi ha marge de millora en cost i manera de fer algunes de les implementacions.

10. BIBLIOGRAFIA

- Unity technologies. Unity Manual: https://docs.unity3d.com/
- Unity technologies. Unity Blog: https://blogs.unity3d.com/
- Unity technologies. Unity Forum: https://forum.unity.com/
- Unity technologies. Unity Answers: https://answers.unity.com/
- Unity technologies. Unity API: https://docs.unity3d.com/ScriptReference/
- Brackeys. Canal de Youtube “Brackeys”: https://www.youtube.com/brackeys
- Sir Wade Neistadt. Canal de Youtube “Sir Wade Neistadt”: https://www.youtube.com/channel/UCH64i_nEITFZIYE98oN8Idw
- GamerGarage. Canal de Youtube “GamerGarage”: https://www.youtube.com/channel/UCBgm2BugU2rMA8AnQlH7G2Q
- FreeSound. Sons creative commons: https://freesound.org

Altres pàgines consultades per la realització d’aquesta memòria:

https://es.wikipedia.org/wiki/Industria_de_los_videojuegos

https://www.gaboxreviews.site/historia/cual-fue-el-primer-videojuego-en-3d-de-la-historia/

https://es.wikipedia.org/wiki/Pan_European_Game_Information

https://es.wikipedia.org/wiki/Taxonom%C3%ADa_de_Bartle

https://www.eurogamer.es/articles/2017-09-07-code-vein-detalla-su-sistema-de-combate

11. ANNEXOS

Annex 1

S'ha adjuntat el producte final del projecte desenvolupat.

Annex 2

Enllaç a la carpeta de Google Drive on es poden consultar els errors gravats.

https://drive.google.com/drive/folders/1xeSucwOGPdGQ9QqMZ_PDrrfwNTvb6MH8?usp=sharing

Annex 3

Enllaç al vídeo on es mostra el resultat final obtingut.

https://drive.google.com/file/d/1-PBZB_bJwoCNI45Qtgqw6_kOZ5q2ukng/view?usp=sharing

12. MANUAL D'USUARI I D'INSTAL·LACIÓ

12.1. Manual d'instal·lació

Aquest projecte no requereix d'instal·lació en el sistema, només és necessari descarregar el projecte, descomprimir-lo i executar-lo. Només per sistemes Windows, ja que no es disposa d'un Mac per provar-lo.

És possible que en executar el joc per primera vegada surti un missatge de Windows advertint de que l'autor de l'aplicació és desconegut i aquesta pot ser perillosa, en aquest cas s'ha de donar a *Més informació* i *Executar de totes formes*.

Per evitar que passi això és necessari tenir un certificat de Microsoft indicant que el software és segur, aquest certificat s'ha de configurar a Unity per fer una exportació certificada i evitar l'avís.

12.2. Manual d'usuari

Els controls del joc són els següents:

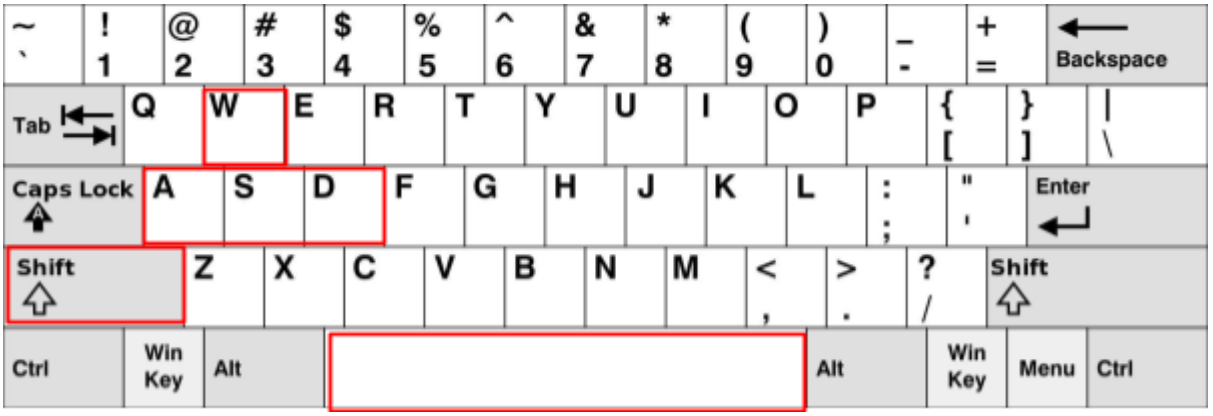


Figura : Tecles utilitzades en el joc(imatge base: https://www.profesionalreview.com/2017/12/13/historia-del-teclado-qwerty/)

- **W, A, S, D:** Moure el personatge.
- **Shift esquerra:** Caminar.
- **Espai:** Saltar

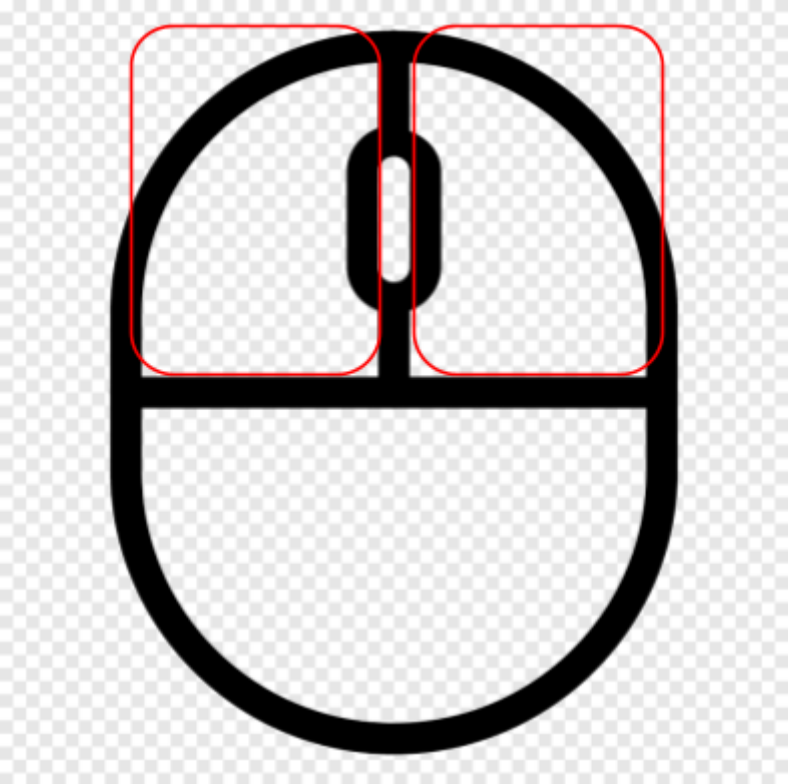


Figura : Botons utilitzatsen el joc (imatge base: https://www.pngegg.com/es/png-exejb)

- **Moviment del ratolí:** Moure la càmera.
- **Botó esquerra:** Atacar (en combat)
- **Botó dret:** Esquivar (si el personatge es mou en combat) o bloquejar (si el personatge és quiet en combat)

13. GLOSARI

- **RPG:** Sigles en anglès de *Rol Playing Game*. Literalment 'joc d'interpretació de rols'. És una classe de videojoc que adapta els jocs de rol de sobretaula.
- **Idle:** Paraula anglesa per inactiu/va. En el context dels videojocs fa referència a l'estat en que el jugador no mou el personatge, està inactiu.
- **Rigging:** Nom que rep el procés de crear un sistema de control per els models 3D. Utilitzat normalment per animals creant un esquelet semblant al de la realitat.
- **UV:** Fa referència a la malla desplegada d'un model 3D. S'utilitzen les lletres 'u' i 'v' per identificar els eixos de coordenades. No s'utilitzen la 'x', la 'y' o la 'z' perque aquestes fan referència als eixos 3D.
- **Souls-like:** Literalment 'com els Souls', Souls fent referència a una saga de videojocs anomenada *Dark Souls*.
- **Input:** Dades que entren en un sistema. En el cas dels videojocs cada 'input' equival a una acció de part del jugador.
- **Character Controller:** Component de Unity per facilitar el moviment i controlar les col·lisions d'un personatge.
- **Animation events:** Literalment 'esdeveniments d'animació'. Són funcions que es criden en un frame específic d'una animeció.
- **Rigidbody:** Component de Unity per controlar la física d'un objecte.