

# Happy Memories

**Alba Raidó Rubinat**

Grau multimèdia

TFG Videojocs

**Consultora:** Gisela Vaquero Juanola

**Professor consultor:** Joan Amedo Moreno

6 de Juny de 2021



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Happy Memories</i>
<b>Nom de l'autor:</b>	<i>Alba Raidó Rubinat</i>
<b>Nom del consultor/a:</b>	<i>Gisela Vaquero Juanola</i>
<b>Nom del PRA:</b>	<i>Joan Amedo Moreno</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>06/06/2021</i>
<b>Titulació o programa:</b>	<i>Grau multimèdia</i>
<b>Àrea del Treball Final:</b>	<i>Videojocs</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<i>videojoc, plataformes, 2D</i>
<b>Resum del Treball (màxim 250 paraules):</b> <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>Des de sempre, el món dels videojocs ha cridat la meva atenció com a jugadora. Per tant, he decidit que el TFG era el moment perfecte per aprofundir en aquesta variant i demostrar les habilitats adquirides durant aquests anys a la Universitat. I a més, reflexionar sobre un nou nivell d'aprenentatge addicional enfocat en la creació de videojocs.</p> <p>Per assolir aquests objectius s'ha decidit crear Happy Memories. És un joc 2D de plataformes que tracta la superació de la depressió.</p> <p>La metodologia emprada en el treball s'ha realitzat d'acord a la duració d'un semestre. Per tant, s'han plantejat uns objectius realistes acord al temps disposat. Acabat el semestre, s'espera que el joc sigui funcional.</p> <p>En aquesta memòria es detalla el procediment que s'ha seguit per desenvolupar el videojoc, des de la conceptualització del disseny fins l'exportació del producte final.</p> <p>En actualitzacions posteriors es pretén poder millorar el contingut i augmentar els nivells del videojoc.</p> <p>No obstant això, estic orgullosa del resultat; ja que és un joc funcional per poder passar una bona estona.</p>	

**Abstract (in English, 250 words or less):**

The world of video games has always attracted my attention as a player. Therefore, I have decided that TFG was the perfect time to go into this variation and to demonstrate the skills acquired during these years at the University. In addition, to reflect on a new level of additional learning focused on the creation of video games.

To achieve these goals, it has been decided to create Happy Memories. It is a 2D platform game that deals with overcoming depression.

The methodology used in the work has been done according to the duration of one semester. Realistic objectives have therefore been set at the time available. After the semester, the game is expected to be functional.

This memory details the procedure that has been followed to develop the game, from conceptualization of design to export of the final product. Later updates are intended to improve the content and increase the levels of the game.

However, I am proud of the result; since it is a functional game to be able to spend a long time.

# Índex

<b>1. Introducció</b> .....	1
1.1. Context i justificació del Treball.....	1
1.2. Objectius del Treball.....	1
1.3. Enfocament i mètode seguit.....	2
1.4. Planificació del Treball.....	2
1.5. Sumari de productes obtinguts.....	3
1.6. Descripció dels capítols de la memòria.....	4
<b>2. Estat de l'art</b> .....	6
2.1. Revisió del gènere.....	6
2.2. Revisió de la tecnologia.....	7
<b>3. Definició del joc</b> .....	8
3.1. Idea i breu descripció.....	8
3.2. Objectius plantejats.....	8
3.3. Interacció entre joc i jugador.....	9
3.4. Sistema de joc.....	9
<b>4. Disseny tècnic</b> .....	10
4.1. Plataforma destí.....	10
4.2. Requisits tècnics.....	10
4.3. Inventari de les eines emprades.....	11
4.4. Inventari d'assets.....	12
4.5. Funcionament IA.....	19
4.6. L'arquitectura del joc.....	21
<b>5. Disseny de nivells</b> .....	23
5.1. Arquitectura de nivells.....	23
5.2. Animator controller.....	27
5.3. Arquitectura de components.....	29
5.3.1. Escenes.....	29
5.3.2. Scripts.....	31
<b>6. Manual d'usuari</b> .....	35
6.1 Requeriments tècnics del maquinari.....	35
6.2. Instruccions del joc.....	35
6.2.1. Objectes a distància.....	35
6.2.2. Relació espai i acceleració.....	36
6.2.3. Criteris de decisió.....	36
6.2.4. Relació dels sons.....	37
<b>7. Conclusions</b> .....	38
<b>8. Glossari</b> .....	40
<b>9. Bibliografia</b> .....	42

## Lista de figures

Figura 1: Diagrama de Gantt.....	3
Figura 2: Hollow knight (Team Cherry, 2017) .....	6
Figura 3: Mario Bros (Nintendo, 1983) .....	6
Figura 4: Estructura de pantalles .....	9
Figura 5: Yari .....	12
Figura 6: Orbe .....	12
Figura 7: IA.....	12
Figura 8: IA.....	12
Figura 9: Obstacle .....	12
Figura 10: Obstacle .....	13
Figura 11: Obstacle .....	13
Figura 12: Plataforma.....	13
Figura 13: Portal.....	13
Figura 14: Fletxa.....	13
Figura 15: Barra de vida .....	13
Figura 16: Procés illustrator .....	14
Figura 17: Creació de polígons .....	15
Figura 18: Inspector gràfic píxel .....	16
Figura 19: Procés audacity .....	18
Figura 20: Procés audacity resultat .....	18
Figura 21: Player tocat per enemies .....	19
Figura 22: Aplicació de collider.....	19
Figura 23: Lògica interacció fantasma.....	20
Figura 24: Mapa de nivells.....	23
Figura 25: Mapa de nivells reduïda.....	23
Figura 26: Menú principal .....	24
Figura 27: Mapa primera part.....	24
Figura 28: Mapa segona part .....	24
Figura 29: Obtenció d'orbes .....	25
Figura 30: Orbe obtinguda .....	25
Figura 31: Nivell 2 i cinemàtica.....	26
Figura 32: yari_idle .....	27
Figura 33: yari_walk .....	27
Figura 34: yari_jump.....	27
Figura 35: Blend Trees .....	28
Figura 36: Sprite-sheet fantasma.....	29
Figura 37: Mapa estats fantasma .....	29
Figura 38: Configuració dels botons.....	31
Figura 39: Configuració dels objectes .....	33
Figura 40: So i control d'orbes.....	37



# 1. Introducció

## 1.1. Context i justificació del Treball

El sector dels videojocs sempre m'ha aportat certa motivació i interès. Actualment, la indústria dels videojocs és altament rellevant degut a l'evolució del producte ofert; la qual calcula una facturació de més de 120.000 milions de dòlars durant l'any 2020. Per aquest motiu, es considera una bona sortida professional, tant per a desenvolupadors de programari, com per a dissenyadors i animadors. És per això que des d'un inici la meua idea ha estat desenvolupar un videojoc.

El treball de final de grau és un projecte que recull tots els coneixements adquirits durant els estudis del grau multimèdia. Aquest projecte permet integrar la gran majoria de coneixements adquirits durant el grau i aplicar-los en un nou nivell de complexitat enfocat a la vessant dels videojocs. Així doncs, el treball inclou tots els processos que determinen la creació del videojoc; des de la selecció del game engine, fins a la exportació del producte final.

Conseqüentment, s'espera adquirir nous coneixements especialitzats en l'entorn del programari Unity, els quals m'obrin portes de cara a un futur professional. I a més, l'obtenció d'un producte d'entreteniment propi.

## 1.2. Objectius del Treball

Durant el desenvolupament del treball, s'ha fixat l'objectiu principal en dissenyar un videojoc des de zero i reproduir el desenvolupament del producte amb la màxima fidelitat als processos i eines utilitzades dins del sector. Així doncs, es pren la determinació d'assolir les següents fites:

- Conèixer com és la planificació del desenvolupament d'un videojoc.
- Avaluació d'eines i programari necessaris.
- Crear el meu primer videojoc i augmentar els coneixements.
- Aprenentatge de l'ús del engine escollit i els components del programari.
- Realitzar i documentar el procés de crear un videojoc.
- Utilitzar metodologies i llenguatges de programació enfocats en la programació de videojocs.



- Planificació del desenvolupament, temps a emprar i els nivells a dissenyar.
- Creació dels elements gràfics propis que siguin necessaris per a l'ambientació, així com també la cerca de material gratuït de sons.
- Gestió de les comunitats d'usuaris de les quals disposa l'engine triat.

### 1.3. Enfocament i mètode seguit

El desenvolupament d'un videojoc es compon per un conjunt de tasques multidisciplinàries on hi participa un equip de professionals amb diverses aptituds destacables dins d'un entorn determinat del projecte. No obstant això, en aquest cas s'ha de desenvolupar de manera individual i durant la limitació d'un trimestre.

Així doncs, es planteja un projecte que no sigui excessivament complex i innovador per tal de poder utilitzar instruccions de programació, que ja són conegudes i presents en altres dissenys de videojoc.

Pel que fa al disseny gràfic i les animacions, s'ha considerat realitzar tots els elements a partir de la producció pròpia; i d'aquesta manera evitar que sorgeixi algun problema en quan als elements necessaris i adequació de l'ambientació respecte a la idea principal del videojoc.

No obstant això, s'ha tingut en compte que siguin elements i animacions simples degut a la limitació del temps que es manté durant tot el projecte.

Respecte a la metodologia emprada, s'ha optat per un model de desenvolupament iteratiu, que permeti disposar en tot moment d'un prototip amb funcionalitat bàsica en cada lliurament; per tal d'obtenir un prototip que permet a l'usuari conèixer i avaluar el producte final i el seu desenvolupament.

### 1.4. Planificació del Treball

Per tal de realitzar el desenvolupament del projecte "Happy Memories", s'ajusta la planificació del treball a una estratègia d'escalabilitat. La planificació es realitza entorn a les fites que s'han determinat.

Així doncs, es divideixen les tasques generals de implementació i desenvolupament al llarg del semestre adaptant-se a les dates de entregues de les proves de l'avaluació continuada. Els últims dies de cada lliurament queden

reservats per revisar i optimitzar el contingut i corregir les funcionalitats que no s’han optimitzat correctament.

A continuació, s’adjunta el Diagrama de Gantt en el qual s’ha realitzat la gestió del projecte.

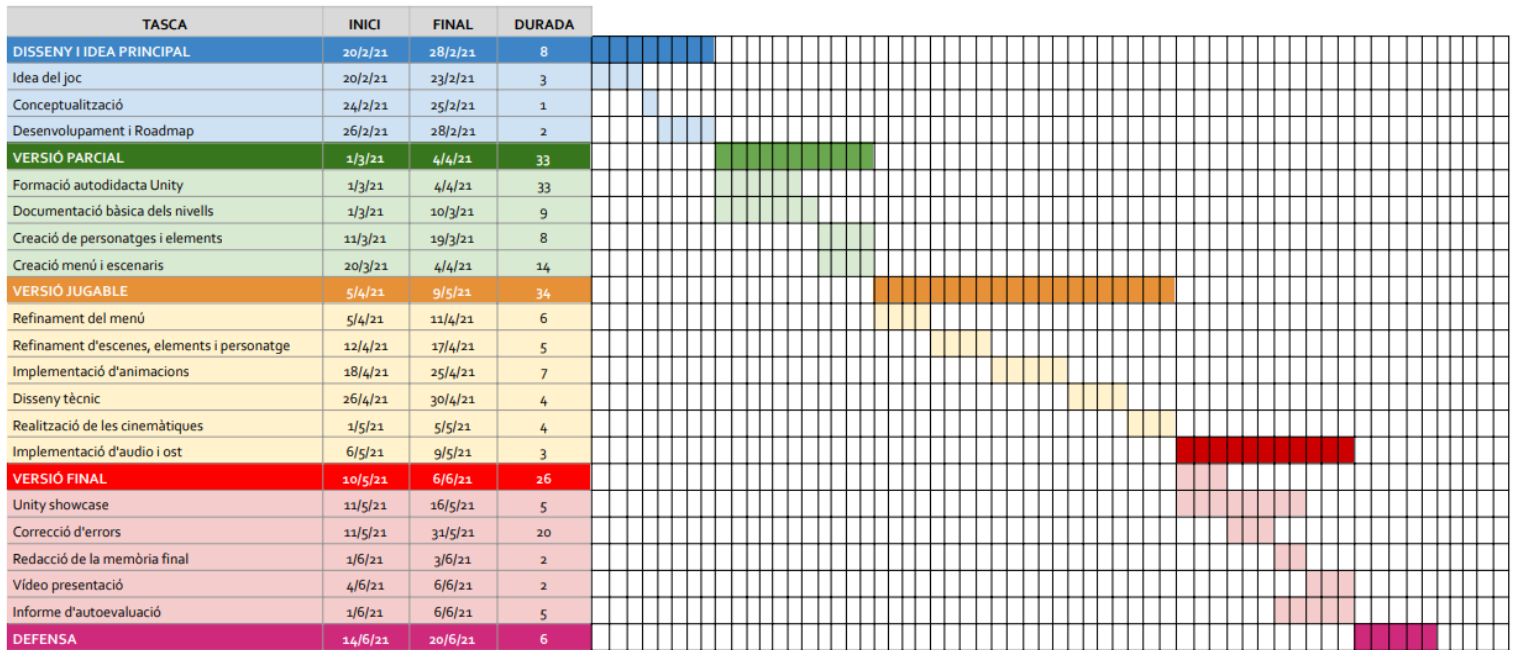


Figura 1: Diagrama de Gantt

Pel que fa als recursos necessaris per a la constitució del projecte, és necessari realitzar la instal·lació del programari durant el procés de versió parcial; el moment durant el qual es realitza l’aprenentatge oportú de l’engine seleccionat.

## 1.5. Sumari de productes obtinguts

Un cop finalitzat el projecte s’espera obtenir els següents productes:

- Executable beta del videojoc per a l’entorn dekstop (Windows).
- Repositori GitHub amb el codi font, recursos gràfics i de so.
- Video explicatiu on s’exposa el procés de desenvolupament.
- Trailer del videojoc.
- Memòria del projecte.

## 1.6. Descripció dels capítols de la memòria

Aquesta memòria es divideix en diferents capítols en els quals s'explica el desenvolupament detallat del videojoc "Happy Memories". Aquests capítols són els següents:

	CAPÍTOL	CONTINGUT
1	<b>Introducció</b>	Capítol introductor que justifica l'elecció del projecte, la planificació del treball, el mètode a seguir i els objectius que determinen el desenvolupament.
2	<b>Estat de l'art</b>	En aquest capítol s'especifica el gènere i la categoria del videojoc. També, s'analitza els diferents motors engines disponibles en el mercat per efectuar la creació del videojoc.
3	<b>Definició del joc</b>	Capítol en el que es defineix la idea del videojoc, la interacció entre joc-jugador, la usabilitat i els objectius plantejats. A més, es detallen inspiracions i referències a altres jocs que han motivat la realització d'aquest projecte.
4	<b>Disseny tècnic</b>	Aquest capítol explica, en major detall, els diferents components del videojoc, funcionament de les IA i inventari dels components. Es detallen els recursos utilitzats, tant externs com propis i s'explica detingudament els diferents elements que componen l'arquitectura del projecte.
5	<b>Disseny de nivells</b>	Aquest capítol es una breu guia explicativa de diferents nivells. Inclou els criteris utilitzat en els aspectes de disseny i la funcionalitat de cadascun dels nivells.
6	<b>Manual d'usuari</b>	Requeriments tècnics del maquinari per jugar, així com també les instruccions del joc.
7	<b>Conclusions</b>	Reflexió del projecte realitzat i tancament de la memòria.
8	<b>Glossari</b>	Glossari de les diferents paraules que no es corresponent al registre general de la llengua catalana.
9	<b>Bibliografia</b>	Llista numerada de les referències bibliogràfiques utilitzades dins de la memòria.

10

**Annexos**

Llista d'apartats que són massa extensos per incloure dins la memòria i tenen caràcter autocontingut.

En la resta de capítols, es realitza una descripció detallada sobre quin ha estat el joc a desenvolupar, les consideracions de disseny seguides i les característiques implementades en diferents elements importants de la programació.

## 2. Estat de l'art

### 2.1. Revisió del gènere

El joc a desenvolupar es classifica dins del gènere d'acció i del subgènere de plataformes-aventura. En aquest tipus de videojocs el personatge principal ha de realitzar la superació de diferents obstacles, posant a prova la consciència espacial de reflexes i agudesa mental del jugador.

L'objectiu del joc es determina a partir de la col·lecció d'un seguit d'elements, els quals desbloquegen la superació dels diferents nivells del videojoc. A més, mitjançant la superació de cada nivell, el jugador descobreix la història o propòsit del personatge i l'objectiu principal com a jugador.

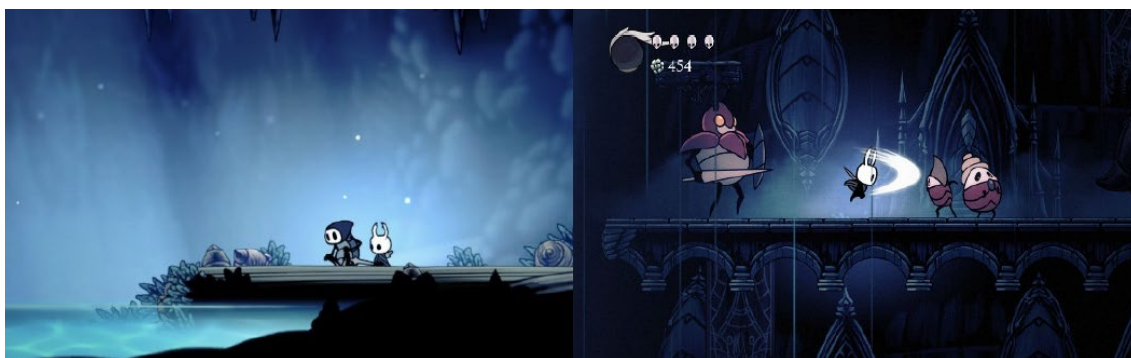


Figura 2: Hollow knight (Team Cherry, 2017)

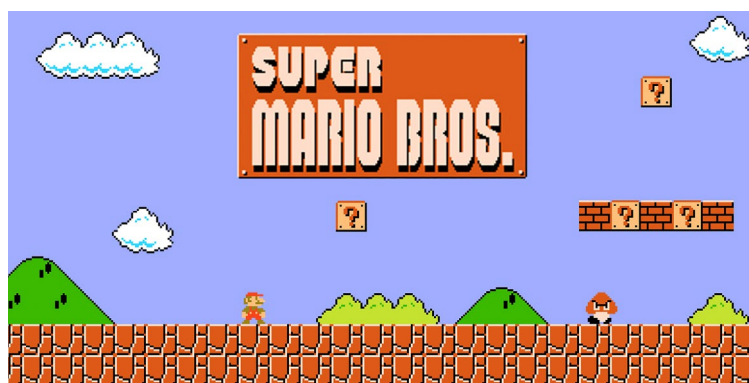


Figura 3: Mario Bros (Nintendo, 1983)

La creació d'aquest videojoc ha estat inspirada en altres que han estat desenvolupats per grans empreses, i de les quals enumero les referències següents:

- **Estil, ambientació i disseny:** Little nightmares i Don't starve
- **Tipus de jugabilitat:** Hollow knight, Mario Bros i Ori

## 2.2. Revisió de la tecnologia

Abans de començar a realitzar el projecte, cal tenir clar una idea i el motor engine en el qual es construeix el joc. Així doncs, es valoren els tres engines més coneguts actualment: Unity, Cocos2D i Unreal Engine. A continuació, s'exposen les diferències entre ells:

- **Unity Technologies**

És un motor engine molt utilitzat per a videojocs 2D o 3D, en els quals predomina el llenguatge de programació C#. A més, disposa d'una amplia comunitat d'usuaris que intercanvien coneixements entre ells. També compta amb recursos propis de l'API, coneguda amb el nom d'Assets Store.

- **Unreal Engine**

A diferència d'Unity, destaca que el llenguatge de programació es desenvolupa en C++ i Blueprints; els quals es diferencien del C# per presentar una curvatura d'aprenentatge molt més pronunciada i complicada. Pel que fa al suport de les plataformes, només disposa de 18 tipus diferents de suport. Per aquest motiu, és un motor menys pobre pel que fa als llançaments del videojoc en altres dispositius.

No obstant això, Unreal manté un bon entorn de desenvolupament i també compta amb un Marketplace de recursos. Pel que fa als beneficis que es poden obtenir a partir de la venda del joc, la plataforma s'emporta un 5% d'aquests beneficis; a diferència d'Unity que deriva de 0€ a 115€ al mes.

- **Cocos 2D**

Tot i ser una bona eina per al desenvolupament de videojocs en 2D, no té una comunitat tant activa com Unity i mostra un problema pel que fa a la preparació autodidacta del codi. Aquest fet es provocat per les poques plataformes de suport de les quals disposa, recursos pobres, i la utilització de Python com a llenguatge de programació.

Finalment, es va decidir emprar Unity com a motor per al desenvolupament del projecte degut als beneficis que aquest aporta. Unity incorpora de manera fàcil i pràctica tots els elements necessaris per a un desenvolupament òptim del videojoc: les físiques per detectar col·lisions, animació de sprits, implementació d'àudio i cinemàtiques, generació de partícules, construcció d'escenaris i altres elements de programació.

## 3. Definició del joc

### 3.1. Idea i breu descripció

“Happy Memories” és un joc de plataformes 2D i d’ambientació fantàstica. Yari és una noia que desperta atrapada en un món que ha estat tergiversat per la falta de llum i que el seu únic objectiu vital és tornar al seu món.

A partir de saltejar de diferents obstacles, Yari ha d’aconseguir recuperar els trossos dels seus records que es troben escampats per tot el mapa.

No obstant això, es presenta la complexitat de superar cada nivell dins d’un temps específic, el qual es controlat per un comptador. Així doncs, Yari ha de descobrir el millor patró per moure’s per les plataformes per tal d’aconseguir tots els records i poder arribar al portal dins de la limitació de temps.

En resum, la història de Yari ens aproxima a la superació dels canvis psicològics que son provocats durant la depressió, els quals es reflexen en el joc mitjançant un canvi progressiu dels escenaris i l’ambientació a partir de la superació de cada nivell.

Així doncs, la protagonista millorarà mitjançant la recuperació de la col·lecció de records positius i superació de nivells.

### 3.2. Objectius plantejats

L’objectiu principal del joc és aconseguir recuperar la totalitat de la felicitat del personatge. De la mateixa manera, es presenta un segon objectiu que implica mantenir viva a Yari en cada nivell del videojoc. Per aconseguir-ho, el jugador ha d’adquirir totes les orbes de llum que es troben repartides en el mapa, i arribar al portal abans de que s’acabi el temps.

Dit això, el personatge ha de saltejar diferents obstacles, tenint en compte un recorregut determinat i una correcta sincronització de moviments, per tal d’evitar que s’acabi el temps. No obstant això, el món de Yari està ple d’enemics que intentaran complicar el procés per tal de que no surti d’aquest món amb vida.

En resum, aquests ítems representen un objectiu que facilita la consecució de l’objectiu principal del videojoc.

### 3.3. Interacció entre joc i jugador

La interacció entre joc i jugador és simple, ja que només existeix un únic protagonista i personatge. El personatge principal crea una directa dependència amb el jugador provocant que aquest entri dins de l'escenari del joc.

Per tal d'obtenir una bona usabilitat, en el present videojoc predomina la interacció a partir d'aplicar la vista 2D, la qual posiciona la càmera en el lateral del personatge. La vista en perspectiva permet que el jugador sempre vegi el perfil el personatge i una millor gestió dels controls del moviment dins d'un escenari d'obstacles i plataformes.

En base a que el jugador pugui percebre la quantitat de temps disponible que li queda al personatge principal, s'ha implementat una barra de vida que actua com a comptador. Aquest element es situa a la part superior esquerra de la pantalla, per tal de que el jugador no la perdi de vista durant l'acció en el joc.

### 3.4. Sistema de joc

El sistema de joc utilitzat és basic, per tal de poder complir amb els requeriments específics d'un videojoc funcional. Així doncs, presenta només un nivell de videojoc.

En resum, el sistema de joc es redueix a les pantalles presentades en el següent gràfic:

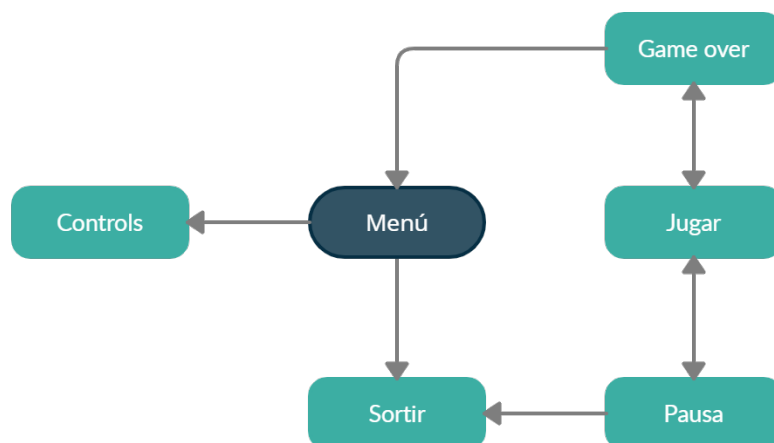


Figura 4: Estructura de pantalles



## 4. Disseny tècnic

### 4.1. Plataforma destí

Inicialment, s'ha considerat triar la plataforma més còmoda per iniciar el projecte. Es considera realitzar el joc per a desktop, ja que possibilita un millor nivell de rendiment i qualitat gràfica. Tot i que per a aquest joc s'han utilitzat gràfics simples, es pretén realitzar millores gràfiques i de components en un futur. També, el fet de realitzar el videojoc per a plataforma desktop, fa més còmoda la realització de proves, actualitzacions i la integració de possibles Mods.

Finalment, el desenvolupament d'un videojoc comporta que el desenvolupador determini la plataforma destí que pugui assolir la quota de mercat més alta. Actualment, els ordenadors són el producte tecnològic del qual tothom disposa, i per tant, la plataforma desktop es considera el mercat més gran dins del sector dels videojocs. A més, possibilita la integració dels comandaments de consola i la possibilitat de visualitzar el joc en qualsevol pantalla, com per exemple un televisor.

No obstant això, la idea de fer-se lloc dins del mercat dels dispositius mòbils pot ser una opció en un futur a partir de realitzar una compatibilitat òptima.

### 4.2. Requisits tècnics

Per tal de realitzar el videojoc a partir de Unity 2020.2.3 es necessiten els següents requeriments mínims:

- Sistema en versió Windows 7
- versió 10.13 High Sierra per a macOS
- CPU x64
- GraphicsAPI DX10, DX100 i DX12-captable GPUS
- AMD GPUs per a macOS
- Hardware que suporti els divers necessaris per funcionar.
- Ratolí i teclat.
- Doble pantalla per tal de facilitar la programació i la visualització del joc, al mateix temps.

### 4.3. Inventari de les eines emprades

A continuació es presenta l'inventari de les eines / programari que s'ha utilitzat al llarg del desenvolupament del videojoc:



- **Unity Technologies:** Motor de desenvolupament de videojocs multi-plataforma creat per la companyia Unity Technologies l'any 2005.
- **Visual Studio 2019:** Entorn integrat de desenvolupament (IDE) que permet el desenvolupament en llenguatge C#, a partir del qual es programa, depura, edita, prova e implementa el codi necessari per executar el videojoc.
- **Pack Adobe:** Programari apte per tal de crear els gràfics del disseny de l'entorn visual i gràfic. En principi s'ha utilitzat Photoshop i Illustrator per tal de realitzar els diferents gràfics i ambientació. També, s'ha utilitzat After effects i Premiere per desenvolupar el tràiler final.
- **Aseprite i Procreate:** programari especialitzar en el disseny i animació de gràfics d'estil píxel-art i producció de *sprites-sheets*.
- **Audacity:** Editor d'àudio de codi obert, a partir del qual es pot modificar un so digitalitzat.
- **TextMesh Pro:** Paquet Unity que permet implementar objectes de text més elaborats.
- **Sourcetree:** Client gratuït que interactua amb el repositori GitHub i és compatible amb Windows i Mac.
- **OBS Studio:** Eina de gravació multi-plataforma i de codi obert que desenvolupa projectes de gravació de pantalla, per tal de poder realitzar minitutorials o streams.
- **Github:** Permet gestionar projectes i controlar versions de codi i administrar els canvis realitzats. És el repositori més utilitzat a nivell mundial per tal de poder pujar i compartir carpetes de projectes grans dins de la xarxa amb altres desenvolupadors.



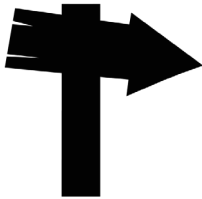
## 4.4. Inventari d'assets

Per realitzar "Happy Memories" s'han utilitzat diversos recursos i assets tant de llicència privada com gratuïta. A continuació es realitza una categorització d'aquests i s'explica una breu descripció de la seva funció dins del videojoc.

### 4.4.1. Recursos gràfics

En aquesta primera taula, s'enumeren tots els recursos gràfics:

ELEMENTS	IMATGE	DESCRIPCIÓ
<b>Yari</b>	 Figura 5: Yari	És el personatge principal. Disposa de diferents sprites per tal de realitzar l'animació de cadascun dels seus moviments. És l'únic personatge del joc.
<b>Orbes</b>	 Figura 6: Orbe	Records positius de Yari que es distribueixen pel mapa, els quals s'han de recuperar en la seva totalitat per poder superar cada nivell i abans d'accedir al portal final.
<b>Enemics IA</b>	 Figura 7: IA	Enemics posicionats en punts essencials, els quals són petits obstacles en moviment que el jugador haurà de saltejar. Es considerat Asset IA degut a que realitza el seu propi moviment.
	 Figura 8: IA	
<b>Obstacles</b>	 Figura 9: Obstacle	Col·lisions físiques distribuïdes pel mapa per tal d'aportar dificultat al joc.

	 <p>Figura 10: Obstacle</p>	<p>El jugador ha de saltejar aquests petits obstacles i en cap cas, tocar-les. Pel que fa a les rodes, aquestes seran dinàmiques i es mouran de dreta-esquerra o amunt-avall. En el moment en que el jugador no pugui evitar-ho, la seva vida es veurà reduïda a 0 i tornarà a començar el joc des de el punt de partida inicial.</p>
	 <p>Figura 11: Obstacle</p>	
<b>Plataformes</b>	 <p>Figura 12: Plataforma</p>	<p>Plataformes estàtiques amb les quals interactua el jugador per tal de poder facilitar el recorregut triat.</p>
<b>Portal</b>	 <p>Figura 13: Portal</p>	<p>Portal per passar al següent nivell. No és possible accedir al portal sense haver recuperat abans totes les orbes col·leccionables.</p>
<b>Senyal</b>	 <p>Figura 14: Fletxa</p>	<p>Forma part de la decoració de l'escenari. No obstant això, senyala el camí per on el jugador ha de dirigir-se dins d'uns paràmetres generals per tal de que s'eviti la desorientació.</p>
<b>Temps</b>	 <p>Figura 15: Barra de vida</p>	<p>El temporitzador es situa a la part superior-esquerra de la pantalla. El funcionament es basa en un compte regressiu fins esgotar el temps. Un cop s'esgota el temps, el joc finalitza.</p>

Tots els assets referent als recursos gràfics s'han creat a partir del programari Adobe; específicament s'ha utilitzat Photoshop i Illustrator.

- El **procés creatiu** segueix l'estètica tètrica per tal d'aconseguir descriure un món de terror, que sigui capaç de captar aquesta essència. Pel que fa a la gama de colors escollida, tant el fons com la decoració es complementen en base als mateixos colors: blaus, negre, gris, i violeta.

No obstant, canviarà a tonalitats mes clares en un segon nivell, en el qual s'inclou el color rosa. D'altra banda, el personatge principal destaca amb colors clars (blau, blanc, vermell i groc) dins de l'escenari fosc que es presenta.

Per aquest motiu, el canvi de colors en el trajecte de la protagonista són importants per al desenvolupament de la trama. Així doncs, per cada nivell superat es realitza un canvi dels colors i elements corresponents a la ambientació. La temàtica canviarà de terrorífica (colors foscos i decoració tenebrosa) a agradable (colors vius i decoració fantàstica) segons el transcurs del joc i la evolució dels nivells.

### ❖ Vectors

Per tal d'aprofundir en el desenvolupament dels gràfics, es realitza a partir de Illustrator.

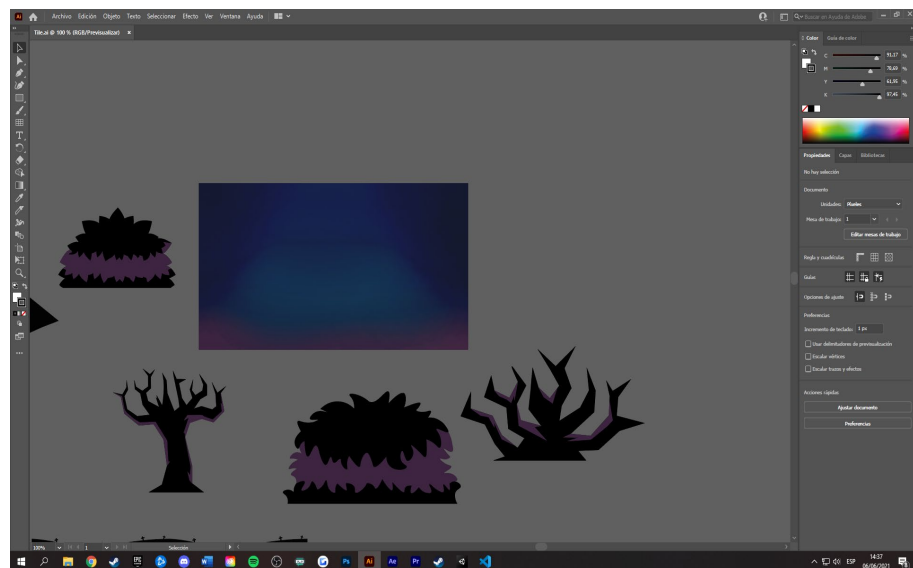



Figura 16: Procés illustrator

A partir de l'eina  es creen els següents polígons que constitueixen el resultat gràfic final elaborat. Aquesta eina permet que punt per punt es realitzi un polígon, el qual es podrà fusionar amb altres i així implementar diferents capes i colors.

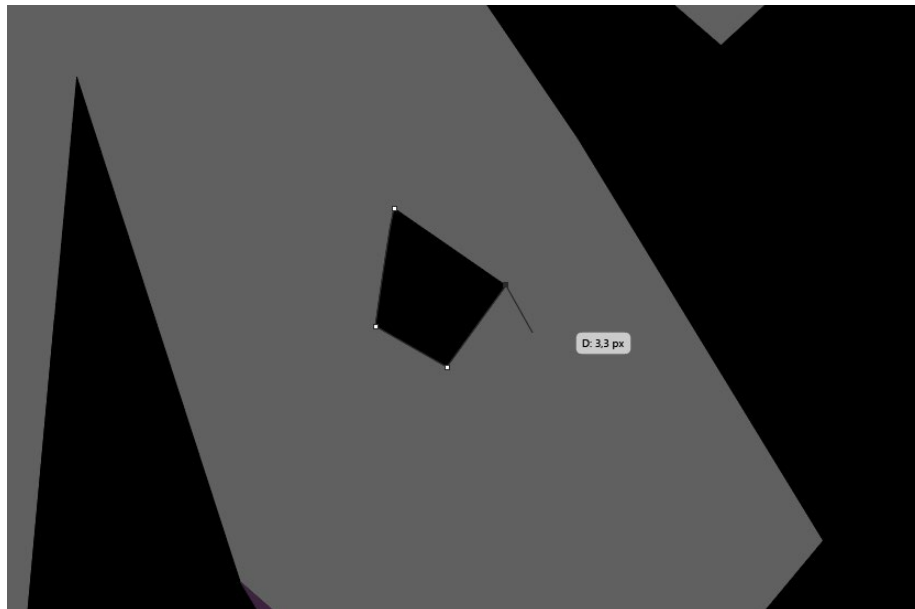


Figura 17: Creació de polígons

El fons i la pantalla del menú principal també s'han realitzat a partir de l'art vectorial d'acord a les dimensions 1020x1080. D'aquesta manera es facilita la implementació d'acord a l'ajust a diferents dimensions dins del joc.

#### ❖ Píxel Art

El disseny en píxel art significa que es crea un gràfic a partir de dibuixar-lo píxel per píxel. A partir del programari Aseprite i Procreate s'han elaborat els fons del videojoc, les orbes i els sprites de personatges i enemics.

Treballar amb aquest tipus de gràfic, facilita la introducció d'elements dins del programari Unity.

A partir de les propietats de cada element, es pot realitzar un augment o reducció del gràfic sense pèrdua de qualitat. Cal tenir en compte d'aplicar aquest tipus de reducció a partir de les propietats 'Inspector'.

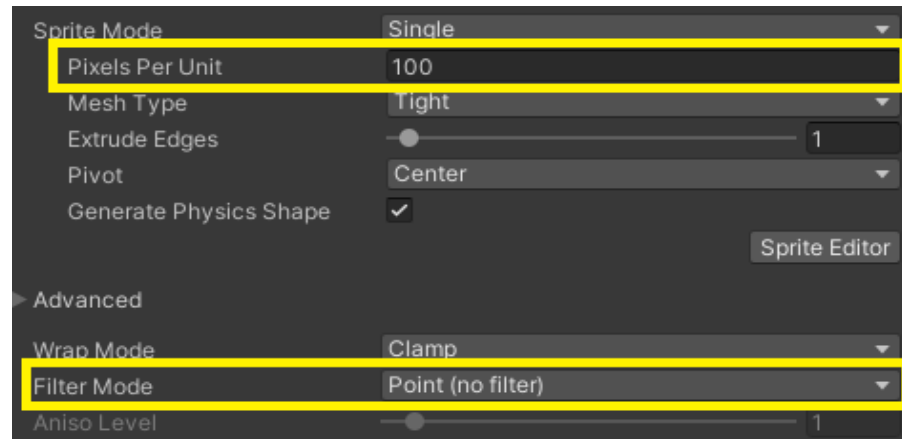


Figura 18: Inspector gràfic píxel

En la imatge anterior es visualitza l'inspector i es senyalen les propietats a modificar per tal de poder aconseguir un resultat òptim dels gràfics en píxel. Per treballar-ho, cal reduir el número 100 a 18 (per exemple), per tal de poder augmentar el píxel de la imatge seleccionada.

A més, és important aplicar el mode 'Point (no filtre)' per tal d'evitar qualsevol distorsió dels elements. El que provoca aquest mètode és la reducció i ampliació dels píxels seguint el mètode de creació a partir del veí. En resum, no es realitza cap pèrdua de qualitat del gràfic.

- Pel que fa a les **característiques tècniques**, s'ha seguit un procés de gràfic vectorial, el qual s'origina a partir de polígons que es creen mitjançant fórmules matemàtiques que divideixen en unitats mínimes la informació que contenen els seus píxels.

Així doncs, els gràfics vectorials es defineixen per una sèrie de punts i tenen la característica de poder redimensionar-se sense ocasionar cap pèrdua de qualitat. Pel que fa al personatge, la IA i les orbes, s'ha utilitzat píxel art en 2D. Aquest disseny complementa amb la tècnica dels gràfics vectorials; i a causa d'això, aquests elements destaquen sobre l'ambientació de l'escenari.

En resum, s'ha tingut en compte que tots dos tipus d'art utilitzats tenen un pes baix en quan a la càrrega d'arxius.

## 4.4.2. Recursos d'àudio

A continuació es mostren els recursos d'àudio que han estat reaprofitats d'altres pàgines amb llicència gratuïta de distribució del material:

<b>BSO del joc</b>
La música ambient del videojoc. Es reproduïx en totes les pantalles del videojoc, tant en la principal com en els respectius nivells.
“Abuse in the Oprhanage” ( <a href="https://patrickdearteaga.com/es/musica-terror/">https://patrickdearteaga.com/es/musica-terror/</a> )
<b>Game over</b>
El so corresponent a la pantalla “Game Over” que s’activa en quan el personatge principal es queda sense vida.
Game over ( <a href="https://mixkit.co/free-sound-effects/game-over/">https://mixkit.co/free-sound-effects/game-over/</a> )
<b>Col·lecció orbes</b>
El so s’activa en el moment que el personatge aconseguix una orbe, per tal de fer-li saber al jugador que s’ha adquirit correctament.
Collect item ( <a href="https://opengameart.org/content/ui-and-item-sound-effect-jingles-sample-2">https://opengameart.org/content/ui-and-item-sound-effect-jingles-sample-2</a> )
<b>Botons del menú principal</b>
Com en el menú principal no compta de música en BSO, s’ha adjuntat un so en el moment en que es prem cada botó del menú. Així doncs, mostra un petit so per tal de que el jugador sàpigues en tot cas l’opció triada.
ButtonClick ( <a href="https://www.soundsnap.com/tags/menu">https://www.soundsnap.com/tags/menu</a> )



## ❖ Audacity

Els recursos d'àudio també han estat afectats per un modificador.

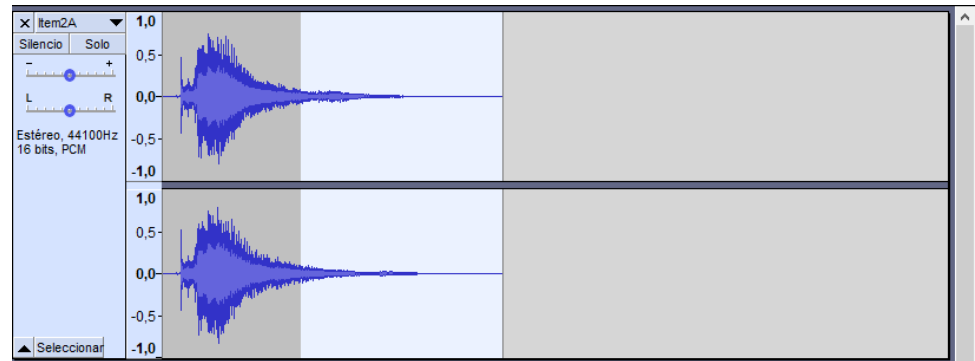


Figura 19: Procés audacity

A partir de la imatge anterior es mostra com s'ha aplicat en el so 'Item' una retallada del so total.

D'aquesta manera, s'aconsegueix un so més òptim per tal de poder implementar-ho dins del joc, ja que es té en compte que no pot durar més de 0,5 segons (duració en temps del moment en que el jugador recol·lecta una orbe).

Per tal d'aconseguir aquest resultat s'ha aconseguit mitjançant 'Efectos > Desvanecer progressivament'. També, s'ha disminuït el volum global del so per tal de que no perjudiqui a la música ambient.

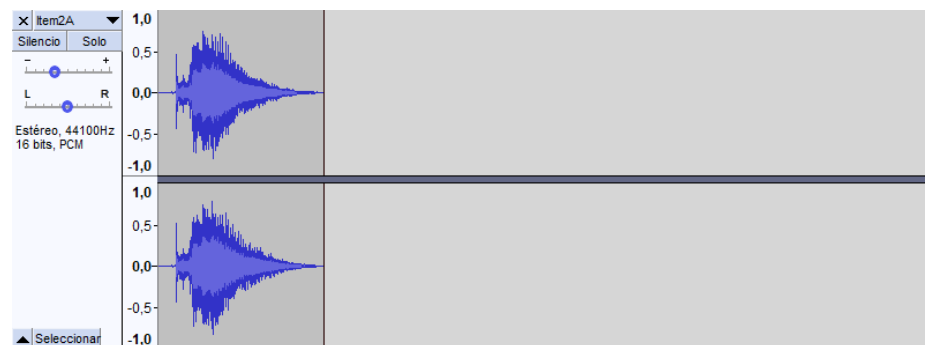


Figura 20: Procés audacity resultat

## 4.5. Funcionament IA

En la taula anterior corresponent al punt 4.4.1. s'ha assignat l'asset IA a l'enemic, el qual es veu afectat per la interacció amb la que actua el jugador.

Un enemic pot entrar en mode de combat per diversos motius. Pot ser per que tenen al jugador a més d'un segon del rang de visió, o perquè entra en contacte directe amb el jugador al rebre la seva presència.

- **L'efecte i lògica d'atac**

En aquest cas, s'utilitza només una lògica d'atac. Ambdós enemics es presenten repartits en el mapa i molesten al jugador a partir de realitzar petites empentes que desconcerten al jugador i provoquen que aquest acabi fora de l'escenari.

En resum, la col·locació d'aquests enemics s'ha realitzat estratègicament per tal de fer tirar al *Player* de les diferents plataformes i aconseguir el *respawn*.

Per tal de que el jugador detecti aquest efecte, s'ha programat un salt en diferents direccions, focalitzat per un canvi de color de 3 segons del personatge.

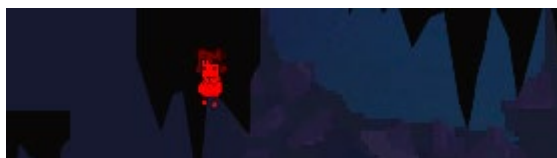


Figura 21: Player tocat per enemics

En el moment en que el personatge torna a tenir moviment normal sobre una plataforma; aquest torna a l'estat natural de color.

Per aconseguir-ho, s'ha implementat un *objecte collider* dins de cadascun dels enemics. A partir d'aquest objecte es reconeix la zona amb la que el *Player* podrà interactuar.

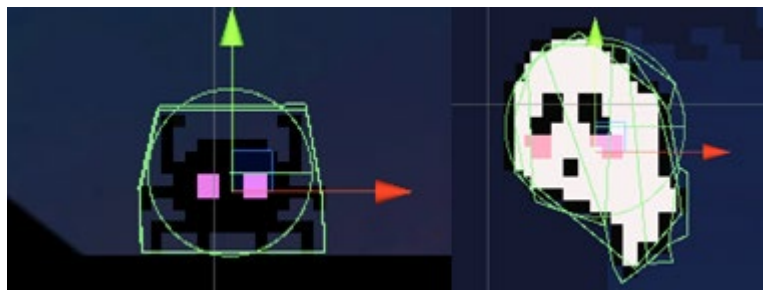


Figura 22: Aplicació de collider

- **Funcionalitat aranya**

A partir d'aquest punt, es busca la manera de realitzar enemics IA diferents. En el cas de les aranyes, es mostra un moviment lògic de dreta a esquerra; el qual segueix el moviment aplicat al *Player*. D'aquesta manera, es marca un tipus de velocitat que quan s'acompleixi; aquesta canvia de direcció.

No obstant això, en el cas de les aranyes aquest moviment es realitza en una mateixa velocitat constant de moviment. No és possible eliminar aquests enemics, per tant el jugador haurà de saltejar-les el màxim possible per evitar caure de les plataformes de l'escenari de joc.

- **Funcionalitat fantasma**

En el cas dels fantasmes, es presenta una funcionalitat diferent. S'ha aplicat *sprites* per tal d'aportar dinamisme al personatge però aquest resta en una única posició.

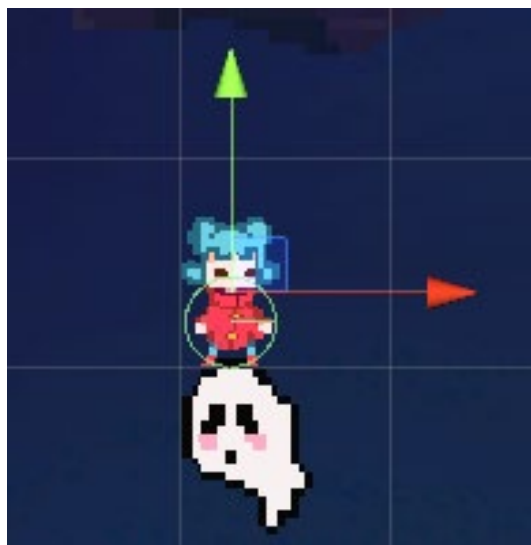


Figura 23: Lògica interacció fantasma

No obstant això, s'aplica la funcionalitat de que el *Player* els pot fer desaparèixer. Tal com es pot comprovar en la imatge, si s'aplica un salt per sobre del enemic, aquest executarà el script i destruirà el enemic.

Per aquest motiu, aquest tipus d'enemics es situen en les plataformes voladores; ja que el *Player* podrà eliminar-los per tal de poder moure's sense cap problema per sobre d'elles.

## 4.6. L'arquitectura del joc

A partir del mapa conceptual presentat en el punt 3.4. d'aquesta memòria, s'ha realitzat una descripció acurada de la divisió d'escenes i pantalles del videojoc.

NÚM	ESCENA	ACCIONS
1	<b>Menú</b>	Menú principal del videojoc compostat per 'Jugar', 'Opcions' i 'Sortir'. S'interactua a partir del desplaçament per ratolí.
2	<b>Opcions</b>	A partir del Menú es pot accedir a "Options", el qual s'activa a partir de fer click en el botó 'Options' del menú principal. El contingut que mostra són els controls del videojoc i un botó de tornada al menú principal.
3	<b>Jugar</b>	Activa el videojoc i executa el primer nivell.
4	<b>Cinemàtica</b>	Diapositiva executada en cada fi de nivell superat. En aquest cas, surt un text que posa 'Nivel 1 superado!'. Aquesta cinemàtica executaria el segon nivell.
5	<b>Pausa</b>	S'utilitza la tecla "ESC" per aturar i activar el joc.
6	<b>Game Over</b>	Si es toca un obstacle/enemic o s'esgota el temps, s'activa la pantalla amb l'opció de tornar al menú principal.

Al iniciar el joc, el primer que es mostra al jugador és el menú principal. Des d'aquesta pantalla pot entrar i sortir del joc o accedir a la pantalla de controls, la qual permet visualitzar els comandaments del videojoc.

Un cop dins de la pantalla de joc, aquesta es correspon amb el primer nivell. Des d'aquesta pantalla es pot jugar i permet accedir a la pantalla de pausa polsant la tecla ESC. La pausa permet al jugador aturar el joc durant un temps i poder tornar a la partida o sortir al menú principal (MainMenu).

També, si el jugador aconsegueix superar el primer nivell; aquest accedeix a una cinemàtica que executa el segon nivell del joc.

No obstant això, en aquest cas només es presenta una imatge conforme el jugador ha superat el primer nivell.

Finalment, la pantalla Game Over apareixerà quan la vida de Yari sigui igual o menor a zero. La pantalla dona la opció al jugador de tornar al menú principal per tornar a començar una nova partida.

## 5. Disseny de nivells

### 5.1. Arquitectura de nivells

El present mapa conceptual mostra l'organització de les pantalles i la relació entre els diferents nivells que es tenia des d'un principi.

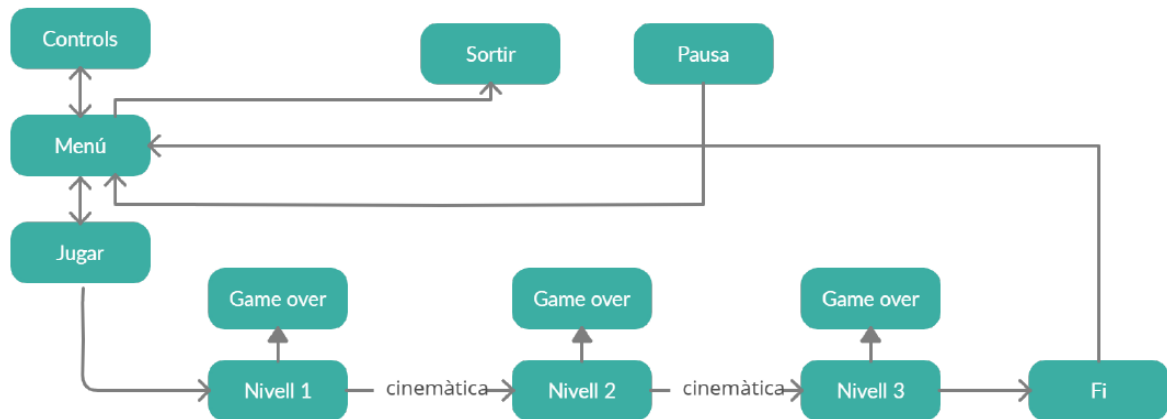


Figura 24: Mapa de nivells

No obstant això, degut a la complexitat del projecte s'han reduït la quantitat de nivells considerablement. Així doncs, es presenta un videojoc amb un únic nivell que demostra el funcionament del videojoc. Es considera l'aportació de la pantalla del nivell 2 com a cinemàtica, per tal de que es visualitzi que s'ha completat el primer nivell satisfactòriament.

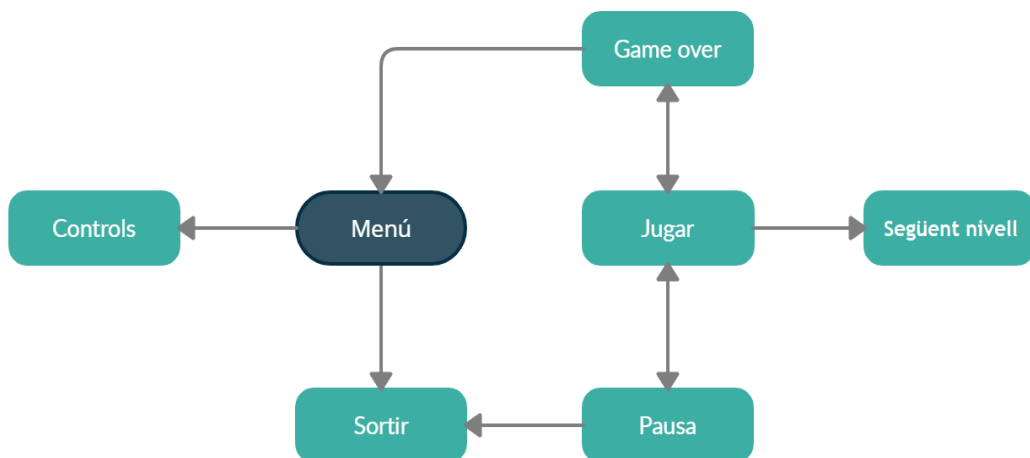


Figura 25: Mapa de nivells reduïda

A continuació, es mostra una breu explicació de cadascun dels nivells exposats en el mapa conceptual anterior:

- **Menú principal**



Figura 26: Menú principal

En principi mostra 3 botons: un per jugar, l'altre per entrar al menú d'opcions i finalment un altre per sortir. El jugador comprova que es marca cada opció amb un rectangle en negre, per tal de que entengui quin es el text seleccionat i quina es la opció que el programari llegeix. També, s'aplica un efecte de so al accedir a l'opció que el jugador ha marcat.

- **Primer nivell**

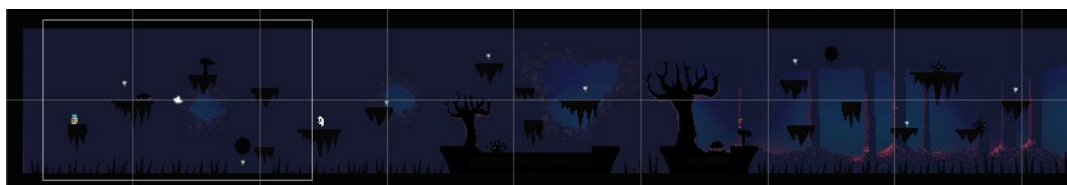


Figura 27: Mapa primera part

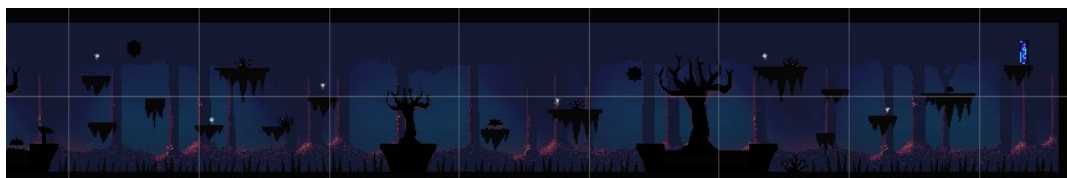


Figura 28: Mapa segona part

Per a aquest projecte només s'ha implementat un nivell de joc que representa el món en el que es troba Yari.

El nivell s'ha realitzat a partir de colors i estètica molt fosca, per tal de correspondre's amb la història plantejada des d'un principi. Així doncs, es dissenya de manera que es correspongui a la ambientació plantejada.

En primer lloc, l'escenari es simula dins d'una cova, en la qual Yari desperta i surt a un bosc frondós i obscur que la condueix fins a una porta secreta que la durà a un altre món.

El personatge ha de recorre el nivell i trobar el portal de sortida, a la vegada que recupera totes les orbes que es troben dins del nivell.

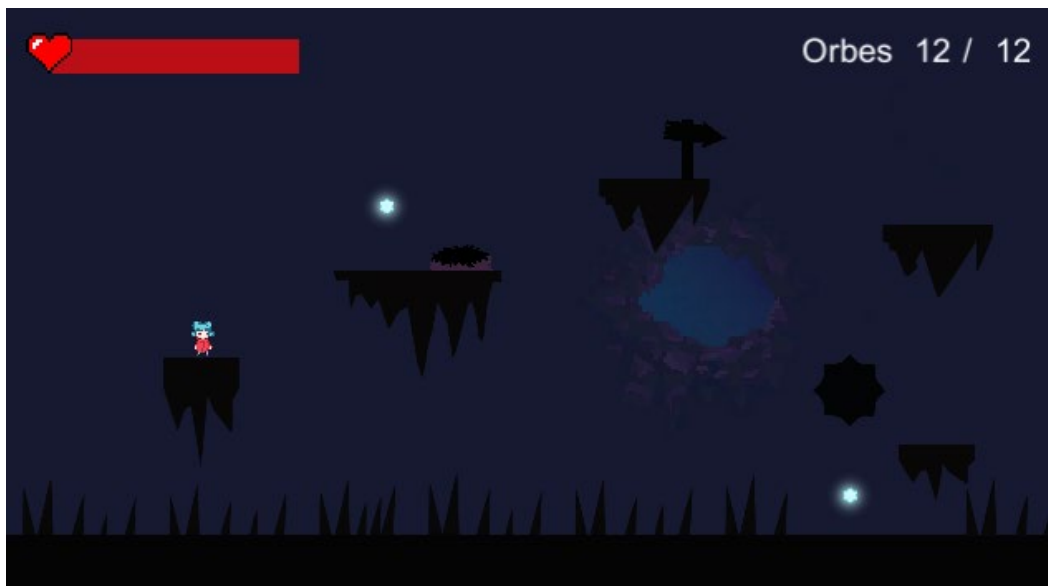


Figura 29: Obtenció d'orbes

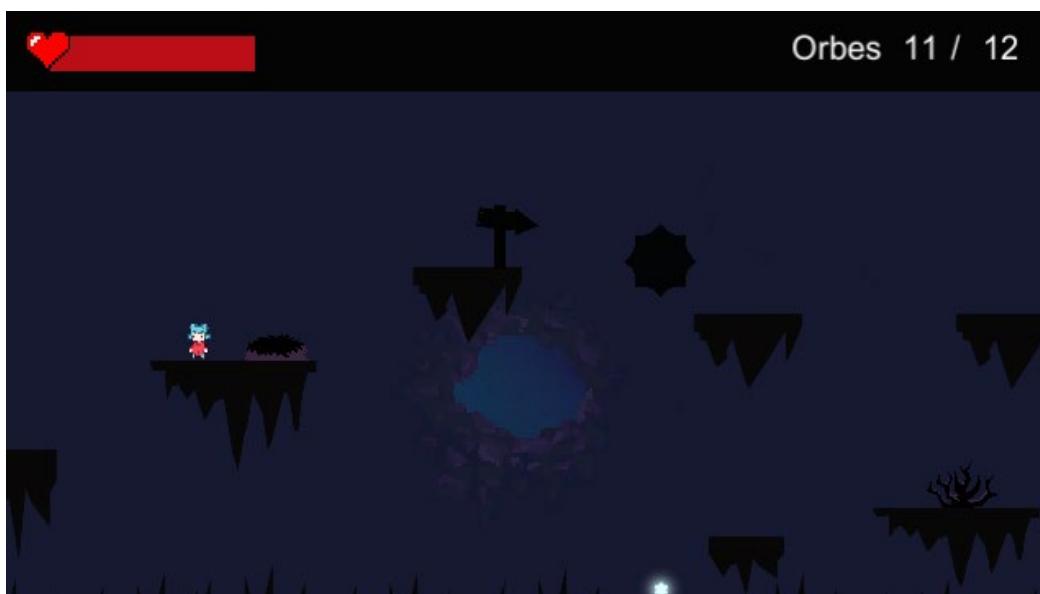


Figura 30: Orbe obtinguda



Com es pot observar, es situa un comptador de les orbes en la part superior-dreta de la pantalla per tal de que el jugador pugui percebre la quantitat d'orbes que li queda per aconseguir (número de l'esquerra). També, es visualitzen les que té en tot moment (número de la dreta).

En segon lloc, es situa a la part superior-esquerra de la pantalla la barra de vida disponible. Aquesta s'actualitza mitjançant el script *HealthBar.cs*. En quan comença el nivell, el compte enredera comença a comptar.

En conclusió, és un nivell petit que requereix aconseguir arribar al portal de manera que s'aprofitin els temps i posicionaments específics que permeten arribar fins al final sense que s'acabi el temps.

- **Cinemàtica i accés a segon nivell**

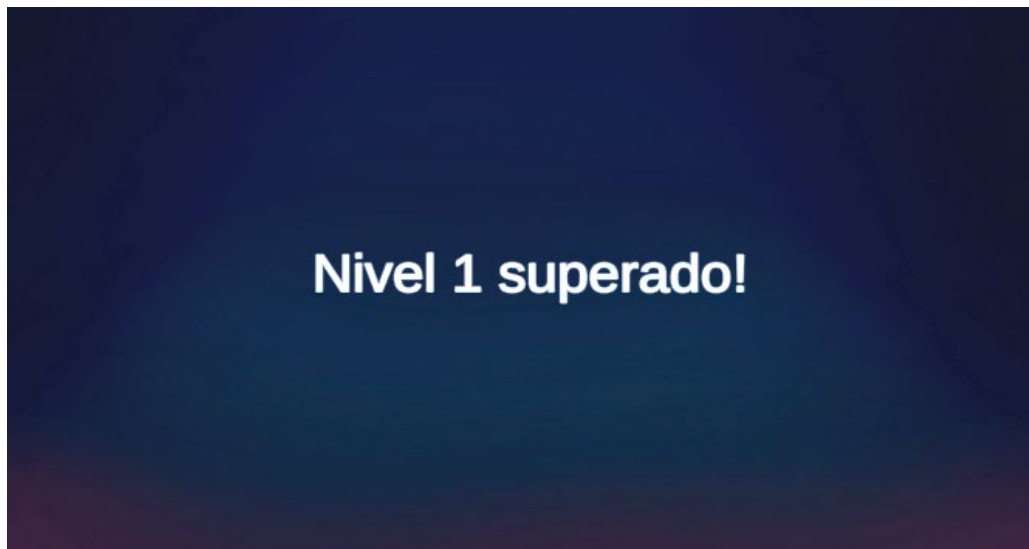


Figura 31: Nivell 2 i cinemàtica

La idea de la cinemàtica era per mostrar els records positius de Yari, que es desbloqueja amb cada superació de nivell. Aquest tipus de records li demostra a Yari que no està sola i sempre ha estat rodejada de bones persones que l'acompanyen durant tots els anys de la seva vida.

El segon nivell pretén ser una rèplica del primer nivell, però es realitza una disposició diferent de les plataformes i el decorat amb una dificultat més elevada. No obstant això, degut al temps limitat del qual consta el projecte, només s'ha pogut crear el nivell per demostrar el funcionament del portal del primer nivell.




En definitiva, es presenta una pantalla en la que es visualitza la superació de nivell.

## 5.2. Animator controller

Per a les animacions de s'ha utilitzat *Animator controller*. Aquest controlador, és un asset natiu que *Unity* permet manipular una sèrie o seqüència de animacions que s'associen a un model 3D. Permet controlar variables referents al tipus d'animació corresponent a l'acció que realitza el personatge.

A partir dels *sprites* s'implementen els moviments del jugador i un dels enemics, el fantasma. Els *sprites* són una sèrie d'imatges en diferents posicions, que es mostren unides en un mateix arxiu animat. Així doncs, s'ha dissenyat els *sprites* que corresponen a cada part del moviment, amb la finalitat de realitzar una acció animada corresponent al moviment establert.

Es mostren les imatges corresponents a l'animació de cada acció:

NOM	FUNCIÓ	SPRITE-SHEET
Yari_idle	Estàtic	 Figura 32: yari_idle
Yari_walk	Caminar	 Figura 33: yari_walk
Yari_jump	Saltar	 Figura 34: yari_jump

A continuació es mostra el mapa *Blend Trees* que identifica la correlació de les diferents funcionalitats. A partir d'aquest mètode es configura l'ordre en que les animacions responen d'acord a la programació especificada en el *script* "Movimiento.cs" corresponent al personatge.

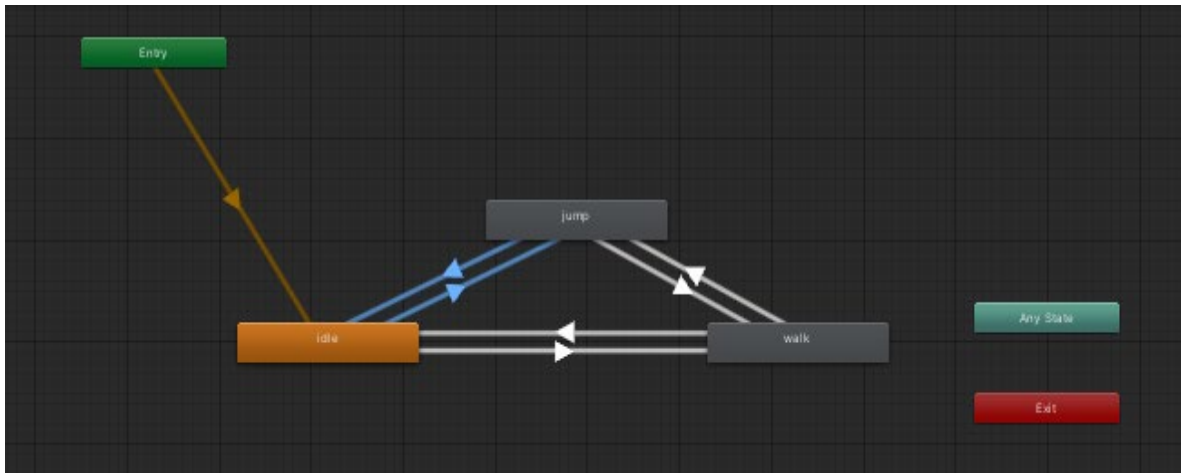


Figura 35: Blend Trees

Per tant, es pot visualitzar que l'animació està composta per una màquina d'estats. Aquesta funcionalitat, activa les diferents animacions intercalant els estats entre ells; per tal d'evitar així una repetició de moviments i augmenti el realisme en les animacions del personatge.

Cada animació es relaciona amb la programació especificada dins del script "Movimiento.cs". A les animacions se'ls ha especificat dos variables:

- **Ground:** Variable de tipus booleà a partir de la qual s'activa o desactiva, ja que només pot tenir dos valors: true i false. En el cas de que el personatge salta, el ground serà desactivat. En el moment en que el personatge cau, es tornarà a activar.

Així doncs, només és possible que es realitzi l'animació de 'salt' en el moment en que el personatge no està tocant cap plataforma.


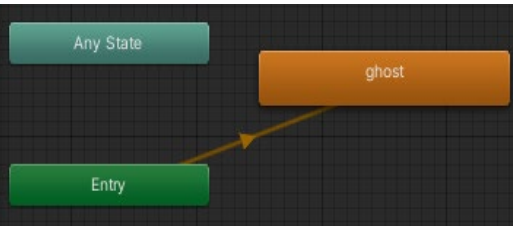
De la mateixa manera, només es possible activar la posició 'walk' i 'idle' quan el personatge està tocant la plataforma.

- **Speed:** Variable de tipus float a partir de la qual s'implementa una velocitat. La lògica funciona de manera que s'executa 'walk' quan el personatge té una velocitat superior a 0.1f. Quan la velocitat és inferior, el personatge retorna a l'animació corresponent a 'idle'.

En quan a les animacions corresponents a la IA, s'ha utilitzat un sistema de màquina d'estat que proporciona el comportament Animator.

De la mateixa manera que en el *Player*, a partir de cada estat es defineix un Blend Tree amb el qual es connecta l'animació que ha de tenir l'enemic segons l'estat en que es defineixi. Dins de la subclasse "enemycontrol.cs" s'ha implementat la lògica del comportament de l'estat de l'aranya i el fantasma.

En la taula següent es mostra el comportament del fantasma, el qual s'ha assignat un mapa d'estats i l'animació del sprite-sheet corresponent.

NOM	FUNCIÓ	SPRITE-SHEET	BLEND TREE
ghost_idle	Estàtic	 <p>Figura 36: Sprite-sheet fantasma</p>	 <p>Figura 37: Mapa estats fantasma</p>

Com ja s'ha comentat anteriorment, en el cas de l'aranya s'ha mantingut només la imatge, per tal de mostrar alguna diferència entre els enemics especificats.

### 5.3. Arquitectura de components

La manera en que s'ha organitzat el joc és a través de la divisió d'escenes. Cadascuna de les escenes té els seus propis objectes de joc, els quals es manipulen a partir de scripts que permeten adherir-se a la lògica de cada objecte.

#### 5.3.1. Escenes

- **MainMenu**

Aquesta escena s'encarrega de monitoritzar el procés del jugador, informar i traslladar les opcions del menú. A més, transmet la música i els sons bàsics que apareixen en totes o casi totes les escenes. Addicionalment, també és l'encarregat de carregar les escenes i activar les pantalles de càrrega durant els processos de càrrega i descàrrega.

És la primera pantalla on el jugador apareix en l'escena del menú principal. Aquesta primera pantalla permet començar el joc, mostra les opcions corresponents i permet sortir del joc. Pel que fa a la interacció, es realitza mitjançant el ratolí.

## ○ **MainPlay**

En aquesta escena es desenvolupa el primer nivell del videojoc. En ella el control del jugador es veu afectat per les diferents tecles ADW i espai, del teclat.

Es mostra el contingut de tot el nivell, així com les diferents plataformes, el jugador, enemics, pantalla Pausa/GameOver, sons, i un seguit d'objectes que es manipulen pels diferents scripts.

Un punt a tractar són els diferents graus d'enemics, ja que existeixen tres modalitats:

- ❖ Les punxes i arbustos: elements estàtics repartits pel mapa sobre i sota de les plataformes.
- ❖ Boles: elements en moviment repartits pel mapa per tal de generar problemes al jugador en el moment de saltar entre plataformes. Aquestes boles són dinàmiques i aporten moviment a l'escena.
- ❖ Els enemics (aranyes i fantasmes): IA que realitzen moviments de dreta a esquerra per tal d'aportar certa dificultat al joc i aplicar moviments inesperats en el personatge; per exemple, el salt enradere en el moment en que col·lisiona el *player* contra un enemic.

## ○ **MainPlay2**

Aquesta escena es correspon al nivell 2 del videojoc. Es compon per les mateixes propietats que el nivell 1.

No obstant això, la realització d'aquest mapa no s'ha gestionat degut a la limitació del temps.

Tot i això, s'ha ajuntat una escena en la que es mostra que s'ha superat el nivell 1 de joc. En aquest cas, si s'hagués tingut més temps, també s'hauria introduït una animació corresponent a un record de Yari.

### 5.3.2. Scripts

- **Cameracontr.cs**

Control de la càmera que provoca un seguiment del moviment del personatge, actualitzant en cada moment la seva posició des de el mètode Update.

- **MainMenu.cs**

Control del menú principal. Permet carregar l'escena del nivell, sortir del joc i obtenir la informació corresponent a l'apartat d'opcions; que també es mostra en la pantalla principal a partir d'utilitzar un Canvas corresponent al nom "Opcions".

- **GameManage.cs**

S'utilitza com a referenciador del objecte "GameManage" des dels objectes de la interfície per tal d'activar altres objectes que es troben desactivats per defecte (com el Canvas del menú de pausa, el qual es busca implementar dins d'aquest script, Opcions i els botons de retorn a Main Menú).

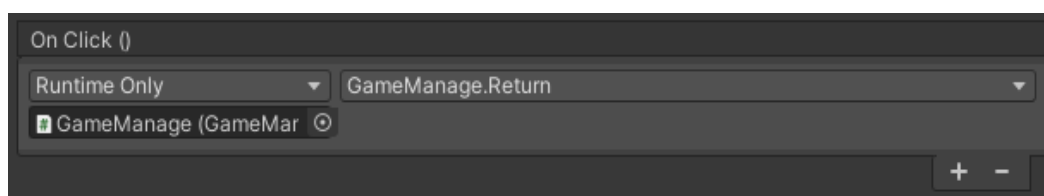


Figura 38: Configuració dels botons

El funcionament és simple ja que el script mai es troba desactivat. Al principi de cada nivell s'inicialitza la seva referència amb el "Gamemanage" a través del loader.

Com es mostra en la imatge anterior, s'introdueix la funcionalitat dels botons a partir de la cerca de la funció específica en el paràmetre 'On click()'.

- **Pause.cs**

Control de l'objecte Canvas corresponent al nom "Pause" que executa el panell de Pausa quan el jugador prem la tecla ESC. També, possibilita

que a partir de l'objecte GameManager, l'usuari pugui retornar a la pantalla del menú principal.

- **Movimiento.cs**

El personatge principal actua com a marioneta manejada per el script, el qual permet controlar el model de manera remota. Així doncs, implementa les funcions de córrer, salt i idle a partir de la menció al paràmetre "Animator", en el qual s'han enregistrat les animacions.

S'apliquen els paràmetres de moviment segons la posició, direcció i condicions especificades al Vector.

A més del control sobre el personatge, s'inclouen funcions referents a la interacció entre el personatge i l'enemic, o sobre altres elements de l'escenari. D'aquesta manera, s'apliquen diferents funcionalitats segons el target al qual el personatge col·lisiona.

Per exemple, el control del personatge al col·lisionar sobre els objectes que tenen el target "obstacle". L'efecte serà que el personatge desapareixi i s'activi la pantalla de GameOver.

També, aquest script implementa una posició determinada com a punt de control personatge i la càmera, per tal d'inicialitzar sempre en el mateix lloc.

- **GroundCollider.cs**

Script que crida al component Pare (Player) corresponent al baking del personatge. La crida es realitza a partir d'un paràmetre private dins del script "Movimiento.cs", el qual conté totes les funcionalitats del Player.

Com a funcionalitat, el collider reconeix quan el Player està en una plataforma. D'aquesta manera, s'ajusta la física del salt i moviment del personatge a un realisme aproximat a altres videojocs. I a més, s'aconsegueix un efecte millorat pel que fa a les animacions.

- **HealtBar.cs**

Script associat al jugador que monitoritza la seva vida i la quantitat de temps que li queda per passar-se el nivell de joc. Es mostra com una barra de vida en la part superior-esquerra de l'escena.

Conté una funció que disminueix la quantitat de vida del jugador segons passa el temps. Al arribar la vida a 0, s'envia un missatge que activa la pantalla de GameOver, per tal d'indicar que el jugador ha perdut.

A més, des de el panell Canvas de GameOver, es permet retornar a l'escena corresponent al menú inicial.

A fi de no complicar el nivell, s'introdueix una vida de 100f per tal d'efectuar una reducció òptima d'acord als obstacles i enemics intruïts dins del nivell.

- **Enemycontrol.cs**

En primer lloc, s'especifica el component d'animació que s'aplica al fantasma.

Aquest script implementa el control de moviment dels enemics. De la mateixa manera, s'executa el target "enemy", que crea una condició entre "Player" i "enemy". D'aquesta manera executa diferents funcionalitats, com és el cas del moviment lateral de l'aranya i el control del Player sobre el fantasma (la destrucció d'aquest objecte).

En el cas dels obstacles, aquest ja s'ha implementat en la part del personatge; ja que els paràmetres recauen directament en el Player.

- **Collectitem.cs**

Control de la recol·lecció de les orbes que es mostren en el mapa, les quals es recullen dins de la carpeta següent:

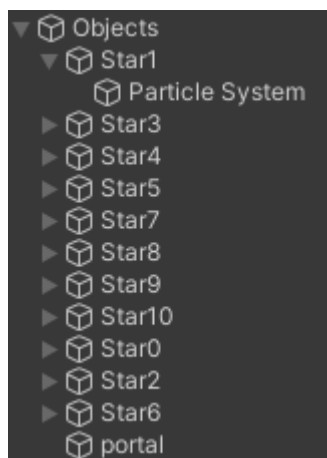


Figura 39: Configuració dels objectes



La gestió s'executa des de l'objecte "Objects". D'aquesta manera, s'aplica el mateix tipus de script en cadascun dels ítems.

Així doncs, el script executa la destrucció de l'objecte quan el *Player* pren contacte directe sobre del ítem.

El script permet mostrar un comptador en la part superior-dreta de la pantalla per tal de que el jugador tingui constància de la quantitat d'orbes que li queda per aconseguir. Un cop recuperades, el jugador haurà de buscar el portal que es troba al final de cada nivell, per tal de superar-lo satisfactòriament.

- **ItemManager.cs**

Per tal de poder executar el següent nivell, és important portar un recompte de les orbes que s'han aconseguit. D'aquesta manera, el joc pot saber en tot moment quantes orbes té el *Player*, i en quin moment executar el següent nivell.

Aquest script es complementa directament amb "Collectitem.cs", des de el qual es criden les funcionalitats.

Permet passar de nivell i visualitzar que el jugador ha arribat al segon nivell a partir d'interpretar l'acció dins del programa.

## 6. Manual d'usuari

### 6.1 Requeriments tècnics del maquinari

Requeriments mínims: 4GB disponibles, sistema operatiu de Windows i un ordinador que pugui executar el videojoc a partir d'uns gràfics mínims i simples.

### 6.2. Instruccions del joc

El principal objectiu de “Happy Memories” és proporcionar al jugador una experiència immersiva que el faci sentir que està vivint la història de Yari. Dit això, es crea una experiència el més pròxim possible a la realitat. El joc té en compte la comoditat i practicitat del jugador referent a la implementació dels controls ja coneguts en altres videojocs, els quals realitzen les mateixes funcions.

Així doncs, el jugador ha d'utilitzar els següents botons per a controlar les diverses opcions i interaccions dins del joc:

- **Botó dret del ratolí:** moviment dins del menú principal i pantalla GameOver o Pausa
- **Teclat alfanumèric – AD:** moviment Yari dreta i esquerra
- **Teclat alfanumèric – WS:** moviment Yari amunt i avall
- **Teclat alfanumèric – D:** accés al portal final
- **Espai:** salt
- **ESC:** pausa

#### 6.2.1. Objectes a distància

Com s'ha explicat en l'anterior apartat d'aquest document, el jugador disposa d'un mecanisme que li permet obtenir objectes que es troben en curta distància sense la necessitat de tocar ningun altre botó.

Tot i que aquesta funcionalitat representa un comportament poc realista, es considerat un punt necessari per a l'entorn de joc. D'aquesta manera es realitzen moviments àgils que es corresponen amb les condicions implicades dins del joc; com per exemple la limitació de temps.

Cal dir que per tal de facilitar la interpretació de recollecció, s'ha adjuntat un objecte 'partícules' que pertany al programari Unity. A partir d'aquesta animació, s'ha pogut realitzar una petita animació.

Tal com es va realitzar en els enemics i *Player*, s'executa com a 'fill' del *Item*. D'aquesta manera, obté els mateixos paràmetres que l'objecte principal.

### 6.2.2. Relació espai i acceleració

La principal causa de cinetosis en ambients virtuals està relacionada amb la velocitat a la que es mouen els elements de l'entorn. Això pot experimentar-ho de primera ma al determinar la velocitat a la que es permet moure al jugador.

Les pantalles més petites o ambients tancats, són més propenses a generar marejos ja que els objectes entren i surten amb més facilitat de la càmera. Un paral·lelisme amb la vida real seria quan un mira per una finestra dins d'un cotxe. Si la persona intenta enfocar els elements més pròxims a la finestra i que passen a una major velocitat, la persona es mareja. Per tant, l'enfoc d'elements es realitza millor a una distancia considerable.

Per aquest motiu, l'acceleració del jugador és de 0.075. Tot i que el jugador no percebi que camina ràpid o lent, la sensació de disconformitat disminueix dràsticament.

### 6.2.3. Criteris de decisió

Els criteris de decisió dins del joc representen un dels punts més rellevants en quan a la dificultat que aquest presenta. Un joc en plataformes 2D té una sèrie d'elements que cal sempre tenir present, així com les pantalles de joc i els menús de selecció. Durant l'execució d'un nivell, un jugador pot realitzar diferents moviments però s'han de qüestionar algunes preguntes: ¿Què farà? Quins elements li fan mal?, Podem aconseguir més temps? Quins criteris de superació té el videojoc? Aquesta sèrie de qüestions es realitzen en qualsevol joc.

El jugador ha de reaccionar hàbilment enfront de la presa de decisions, respecte a la limitació de temps que presenta el concepte del videojoc. Per tant, ha de ser conscient en els moviments a realitzar; per tal de poder calcular el temps que tarda en arribar a l'orbe més pròxima.

Com s'ha explicat anteriorment, l'única manera de poder passar satisfactòriament el nivell del joc; és a partir d'una sincronització entre el temps disponible i la recuperació de les orbes; ja que aquestes permeten obtenir un percentatge de la vida perduda anteriorment.

Per aquest motiu, el criteri de decisió del jugador serà un aspecte important dins del joc.

#### 6.2.4. Relació dels sons

L'adequació d'un so durant la recollecció dels *items* ajuda a la interacció amb el jugador; ja que aquest reconeix el so 'Item' i ho relaciona amb que ha aconseguit una orbe. Així doncs, s'evita que el jugador estigui desconcertat i hagi de mirar constantment el comptador de les orbes.



Figura 40: So i control d'orbes

De la mateixa manera, el so de 'Game Over' es relaciona amb la pantalla de finalització del joc.

Com és habitual, s'han utilitzat sons icònics que ja tenim gravats a la ment; i que per tant són associats a diferents accions ja conegudes per part d'altres videojocs reconeguts.

## 7. Conclusions

A l'inici del projecte s'han enumerat els objectius d'aquest treball, i considero que s'han complert amb èxit; ja que també s'han lliurat tots els documents que conformen el projecte dins del termini establert i s'ha seguit la metodologia i el procés plantejat en la planificació del treball.

En termes generals, el treball ha estat una experiència positiva. El projecte "Happy Memories" s'ha desenvolupat a partir dels coneixements apresos durant el grau i la recerca d'altres fonts d'informació específiques en la programació de jocs.

No obstant això, he hagut de prendre algunes decisions, com la de limitar els nivells i contingut que volia implementar des d'un principi dins del videojoc, per poder aconseguir arribar a temps a tots els lliuraments.

Controlar els aspectes addicionals pel que fa al desenvolupament del disseny, ha estat un dels entrebancs més importants. La decisió d'evitar els assets prèviament construïts i realitzar el disseny propi; m'ha proporcionat desenvolupar un projecte des de 0 i implicar-me en cada aspecte bàsic de la seva creació. Tot i que m'ha dificultat el desenvolupament, estic orgullosa d'haver conegut el procés necessari per a realitzar els meus propis personatges, dissenys d'ambientació, sprite-sheets i animacions.

Certament, pel que fa al procés creatiu, aquest sempre és millorable; ja que en cada projecte els professionals millorem per poder encarar el següent amb millors resultats. Tot i això, estic satisfeta d'haver acomplert el procés de disseny satisfactòriament. La creació d'animacions m'ha obert un nou interès que no tenia des d'un principi. Així doncs, la creació de personatges 2D i 3D; tal com la creació de les seves animacions, podria considerar-se com a una nova afició.

D'altra banda, la programació d'un joc 2D no es considera un projecte complex, però realment ha de considerar moltes variables, scripts i processos. Tot i això, els assets del motor engine Unity Technologies i la comunitat, han facilitat el desenvolupament del videojoc. D'aquesta manera, a partir de mirar alguns tutorials, fòrums i documentació, s'ha pogut posar en marxa el projecte.

No obstant això, considero que les dificultats i entrebancs presentats són reptes que aguditzen la creativitat i l'aprenentatge.

Com a tancament d'aquest document, cal dir que crear el meu primer videojoc era un repte personal que tenia des de que vaig començar la carrera. Han estat

4 mesos que s'ha requerit de moltes hores de treball. No obstant, he millorat dràsticament els meus coneixements de disseny i programació.

En quan al futur de "Happy Memories", considero que s'han assolit molts coneixements i voldré seguir ampliant-los. S'ha deixat una sèrie de línies obertes pendents d'incloure en algun moment. En primer lloc, queda pendent confeccionar uns quants nivells, per tal de facilitar el desenvolupament de l'obra. En segon lloc, millorar el disseny gràfic elaborat per a la realització de l'ambientació del videojoc, ja que sempre és millorable. I finalment, m'agradaria incloure nous objectius, com puzles i altres IA; provocant així noves funcionalitats en el jugador, com per exemple armes.

I ja per acabar, també m'agradaria realitzar les modificacions necessàries per poder fer un desplegament del joc en dispositius mòbils; ja que crec que hi trauria un millor profit al joc.

## 8. Glossari

**2D:** Acrònom utilitzat per referir-se a entorns de 2 dimensions.

**Animator Controller:** Asset natiu de Unity que permet establir seqüències d'animacions

**Assets:** Item que pot ser utilitzat dins d'un projecte.

**Build:** Acció de compilar un programa per obtenir el seu executable.

**Idle:** Tipus de animació utilitzada per representar períodes sense activitat.

**NavMesh:** Malla de navegació que és una estructura de dades abstracta utilitzada per la intel·ligència artificial per trobar un camí que enllaci els dos punts.

**Script:** Fragment de còdi que implementa una funcionalitat.

**UI:** Acrònim utilitzat per a referir-se a la interfície d'usuari

**Game engine:** Aplicació que permet el disseny, la creació i representació d'un videojoc.

**Ítem:** Element de l'escenari del videojoc amb el que el personatge principal pot interactuar.

**Sprite:** Col·lecció d'imatges que mostren la animació d'un element, el qual normalment és el personatge principal o una IA.

**Vista en tercera persona:** Perspectiva del joc on els jugadors veuen el personatge com a una persona al·liena a ell. Pot veure la totalitat de la pantalla.

**Script:** Fitxers de codi amb un conjunt d'instruccions que dona una certa funcionalitat a aquells objectes on s'implementa.

**Collider:** Component assignat als objectes d'Unity per delimitar la zona on es produeixen col·lisions.

**HUD:** Element gràfic que es mostra en tot moment per pantalla durant el nivell i que mostra informació d'utilitat per jugador, com per exemple, el temps de vida que li queda.

**Prefab:** Es tracta d'un GameObject d'Unity on es guarda tota la configuració dels elements que el componen i els seus valors. Serveix com a plantilla per instanciar nous objectes amb la mateixa configuració.

**Dekstop:** Fa referència al escriptori del ordenador, el qual es relaciona amb la pantalla principal que s'observa en qualsevol dispositiu en el que hi predominen les icones i eines disponibles.

**Beta:** La versió beta d'un videojoc generalment és l'última versió de prova abans de la final, la qual serà la comercialitzada. En aquesta versió els provadors de joc poden corregir errors abans de llançar la versió beta tancada (versió definitiva i més estable del joc).

**Baking:** És el càlcul de la escena per determinar l'àrea del escenari en la que es mouen els personatges.

**Respawn:** Regeneració o tornar a aparèixer un personatge que s'ha mort en l'escena.

**Float:** Atribut que defineix un valor sobre una variable.

**Booleà:** Atribut que defineix dos valors possibles sobre una variable, entre els quals només pot ser 'true' o 'false'.

**Canvas:** És l'àrea en el que es troben els elements UI. Es mostra com a un rectangle en la vista de l'escena.

**Elements UI:** Aquests elements s'ajusten al total de la pantalla, i l'escala corresponent. Per exemple, s'utilitza en la pantalla de game over, menú principal i pausa del joc.

**Target:** Indica el nom donat a una propietat per tal d'enllaçar-la a una altra. D'aquesta manera es reconeixen les funcionalitats a aplicar dins de la programació de dos elements.



## 9. Bibliografia

The Indie Game Developer Handbook, Richard Hill-Whittall [2 de Juny, 2021], UOC, Llibre.

Introduction to Game Design, Prototyping and Development: From Concept to Playable Game with Unity and C#, Jeremy Gibson [2 de Juny, 2021], UOC, Llibre.

Unity, Programari lliure de llicència [24 de Maig, 2021], [Web] <<http://unity.com>>.

Game Programming Patterns, Robert Nystrom [3 d'Abril, 2021], UOC, Llibre.

Multiples Autores. List of Unity games. Wikipedia. [27 de Maig, 2021] Web. <[https://en.wikipedia.org/w/index.php?title=List\\_of\\_Unity\\_games](https://en.wikipedia.org/w/index.php?title=List_of_Unity_games)>.

Live Training 5th September 2016 - Creating a Main Menu, Unity. YouTube. [6 de Febrer, 2021] Web. <<https://www.youtube.com/watch?v=OWtQnZsSdEU>>.

How to change scene, GameDevTraum. Youtube. [4 de Març, 2021]. Web. <<https://www.youtube.com/watch?v=05OfmBlf5os>>.

2D Animation in Unity (Tutorial), Brackeys. Youtube. [3 de Juny, 2021]. <<https://www.youtube.com/watch?v=hkaysu1Z-N8&t=152s>>.

How to load levels in Unity/Scenes, The Game Dev\_. Youtube [5 d'Abril, 2021], Web. <<https://www.youtube.com/watch?v=5TOHaygv1IE>>.

Unity 2D Collecting Coins Tutorial, Xlaughts, Youtube. [26 d'Abril, 2021], Web. <<https://www.youtube.com/watch?v=DZ-3g31jk90>>

How to add sound or audio effects SFX to Unity 2D arcade game, Alexander Zotov. Youtube. [17 d'Abril, 2021], Web. <https://www.youtube.com/watch?v=8pFlnyfRfRc>.

How to make a platformer like Mario in Unity, The Weekly Coder. Youtube. [24 de Febrer, 2021] Web. <https://www.youtube.com/watch?v=IMclRECa14o>>.

Health Bar + Lives Counter, Antarsoft, Youtube. [16 de Maig, 2021] Web. <<https://www.youtube.com/watch?v=iX0BEiJTjrE>>.

Points counter, High score and display UI in your game, Coco Code. Youtube. [20 de Maig, 2021] Web. <<https://www.youtube.com/watch?v=YUcvy9PHeXs>>.