



U SCIENCE TECH
FACULTAT DE CIÈNCIES
I TECNOLOGIA
UVIC-UCC

Treball de fi de grau

Creació d'un joc i conversió a multijugador

Josep Sánchez

Grau en Multimèdia

Tutor/a: Enric Vergara Carreres

Vic, juny de 2019

Índex

Índex de figures.....	4
1. Introducció.....	6
2. Motivacions.....	7
3. Objectius.....	8
3.1 Objectius generals:.....	8
3.2 Objectius específics:.....	8
4. Requeriments.....	9
5. Game Design Document.....	10
5.1 Concepte.....	10
5.2 Objectius.....	10
5.3 Normes.....	10
5.4 Propietats dels jugadors.....	11
5.4.1 Poders.....	11
5.5 Jugabilitat:.....	11
5.5.1 Controls.....	11
5.5.2 Mecànica del joc.....	12
5.5.3 Reptes.....	12
5.5.4 Sancions.....	13
5.6 Escenaris.....	13
5.7 Interfície.....	14
5.7 Diagrama de funcionament.....	17
6. Implementació:.....	18
6.1 Implementació audiovisual:.....	18
6.1.1 Menú:.....	18
6.1.2 Avatars:.....	19
6.1.3 Línies volumètriques:.....	20
6.1.4 Pack d'explosions:.....	21
6.1.5 Estètica de l'escenari:.....	22
6.1.6 Murs:.....	23
6.1.7 Sons.....	24
7. Implementació de la programació.....	25
7.1 Lògica d'un jugador:.....	25
7.1.1 Lògica d'inici de la partida.....	26
7.1.2 Lògica de la partida.....	26

7.1.3 Intel·ligència artificial	28
7.1.4 Esdeveniments de finalització de la partida	29
7.2 Lògica de multijugador:	29
7.2.1 Inici de sessió amb Google Play Services.....	32
7.2.2 Sistema de matchmaking o emparellament	33
7.2.3 Sistema de creació de <i>lobbies</i> o sales:	34
7.2.4 Lògica multijugador dintre del joc:	35
7.2.5 Finalització de la partida	37
7.2.6 Interacció amb Google Play Services per a la puntuació.....	38
8. Planificació	39
9. Pressupost.....	41
10. Conclusions.....	43
11. Bibliografia i webgrafia.....	44

Índex de figures

Figura 1: Exemple de la càmera del joc	12
Figura 2: Exemple del primer mapa	13
Figura 3: Exemple del segon mapa	14
Figura 4: Exemple del tercer mapa	14
Figura 5: Interfície del menú.....	15
Figura 6: Interfície del joc quan la partida està en curs	15
Figura 7: Interfície dels botons quan es perd la partida.....	16
Figura 8: Menú del joc.....	18
Figura 9: Avatar 1	19
Figura 10: Avatar 2.....	19
Figura 11: Avatar 3.....	19
Figura 12: Avatar 4.....	20
Figura 13: Exemple d'implementació de línies volumètriques	20
Figura 14: Explosió en esfera.....	21
Figura 15: Explosió en ona.....	21
Figura 16: Exemple de la textura de metall número 37	22
Figura 17: Exemple de la textura de metall número 20	22
Figura 18: Exemple d'una textura aplicada a una columna.....	23
Figura 19: Exemple del mur que cau quan els avatars	23
Figura 20: Panell de control de Google Play Services	32
Figura 21: Inici de sessió amb Google Play Services.....	33
Figura 22: Consola de Google Play.....	38

Resum del treball de fi de grau

Títol: Creació d'un joc i conversió a multijugador

Paraules clau: Game Design, Multijugador, Programació, Videojoc

Autor: Josep Sánchez Serrat

Tutor: Enric Vergara Carreres

Data: Juny de 2018

En aquest treball de fi de grau es documenta la conversió d'un joc d'un jugador a multijugador de dos o quatre jugadors utilitzant Photon Networking ¹, passant per les etapes de realització del joc d'un jugador, implementació de l'inici de sessió seguit del *matchmaking* ² o sistema d'emparellament, sistema de creació de *lobbies* ³ o sales, lògica multijugador quan la partida està en curs i implementació d'un sistema de classificació amb Google Play Services.

L'objectiu d'aquest projecte és aconseguir convertir un joc d'un jugador a multijugador i estudiar la viabilitat, el procés, la dificultat i el temps requerit per a la realització d'aquesta conversió. El resultat serà un prototip de videojoc disponible per a Android.

This Degree's Final Project documents the conversion of a single player game into a multiplayer game for two or four players using Photon Networking as well as the stages of creating a single player game, implementation of the start session followed by matchmaking, creation of lobbies or rooms, multiplayer logic within the game and implementation of a classification system with Google Play Services.

The objective of this project is to convert single player game to a multiplayer game and study the feasibility, process, difficulty and time required to perform this conversion. The result will be a video game prototype available for Android.

¹ Photon Networking: És un conjunt de eines de desenvolupament de jocs multijugador que s'integren a Unity.

² Matchmaking: És el sistema que s'encarrega d'organitzar les partides online contra altres jugadors.

³ Sistema de lobbies o sales: És la zona d'espera prèvia al començament de la partida.

1. Introducció

El projecte realitzat com a treball de fi de grau consisteix en la conversió d'un joc d'un jugador a multijugador anomenat Crush ball.

Aquest joc consisteix en defensar la línia de porteria del jugador, tornant totes les boles cap als altres jugadors. Quan una bola entra a la porteria, es resta una vida de les deu vides inicials amb les quals comença el cada jugador. El joc acaba quan s'han eliminat tots els contrincants de la partida, és a dir, el número de vides dels altres jugador arriba a zero.

El projecte consisteix en el desenvolupament d'un joc d'un jugador i després la conversió d'aquest joc a multijugador, ja que sense el joc prèviament definit i desenvolupat no es pot realitzar aquesta conversió. Per tant, tot el marc d'aquest treball de final de grau inclou la creació del joc en modalitat d'un jugador i després el procés de conversió a multijugador.

S'ha escollit aquest projecte perquè dins del grau multimèdia s'ha estudiat programació de videojocs amb Unity a l'assignatura de programació avançada i posteriorment a l'assignatura d'entorns virtuals. S'ha volgut experimentar i aprofundir aquests coneixements amb la immersió a multijugador que no s'ha experimentat a les assignatures de la carrera.

A la part estètica s'han utilitzat models gratuïts de la botiga de Unity així com alguns elements de pagament de partícules i sistemes d'interfície visual.

A la part auditiva s'han utilitzat diferents sons gratuïts per a donar context a les col·lisions dels diferents elements, explosions, poders i música d'ambient.

Pel que fa a la part lògica s'ha utilitzat Photon Networking, gratuït fins a un nombre determinat d'usuaris actius al mateix moment. Això permet desenvolupar el joc sense cap impediment encara que resultaria de pagament si es volgués preparar per a un nombre d'usuaris més elevat.

2. Motivacions

Les motivacions d'aquest projecte són diverses:

- Primer de tot, aprofundir en les parts que ja s'ha estudiat i treballat en les assignatures de la carrera, com són entorns virtuals i programació avançada amb Unity.
- L'atracció als videojocs també és una motivació clau. La quantitat d'hores que requereix i la complexitat de la tasca ha d'atraure a la persona de manera directa. En aquest cas la motivació a desenvolupar un videojoc sempre s'ha tingut.
- La lògica d'un joc multijugador és complexa i això ens permet aprofundir en tots els seus conceptes, alhora que es treballen coneixements adquirits a la carrera en l'àmbit de programació, de manera que queden reflectits en la realització del joc d'un jugador.
- També és un factor clau la motivació d'aprendre molt més sobre el funcionament dels jocs multijugador que sempre han suposat un repte molt imponent.
- S'ha escollit un joc d'aquestes especificacions perquè és atractiu en l'àmbit audiovisual és atractiu i també en l'àmbit de competitivitat.
- Està inspirat en jocs com el Crash Bash, que consisteix en diferents mini jocs i un d'ells específicament relacionat en defensar la porteria. És un joc del qual se'n coneixia l'existència i sempre hi ha hagut un interès en desenvolupar un joc amb aquesta temàtica.
- És molt important remarcar que aquest projecte està impulsat pel fet de conèixer cada una de les etapes del procés de conversió del joc a multijugador, i així es veurà reflectit al marc teòric, quan es plasmin els diferents passos a seguir que es necessiten per a aquesta conversió.

3. Objectius

Els objectius d'aquest treball de final de grau es poden dividir en objectius generals que s'han tingut presents en el desenvolupament del projecte i objectius específics, que concreten eines o tasques a implementar no compreses en els objectius generals.

3.1 Objectius generals:

- Elaborar un videojoc de mecànica senzilla amb estètica concordant a les mecàniques i l'estil de joc. Encara que s'ha procurat aconseguir una mecànica simple, els objectius del jugador són molt complets i requereixen un desenvolupament complex en un ampli ventall d'àmbits.
- Elaborar una intel·ligència artificial que pugui competir a nivell bàsic contra els usuaris. Aquesta intel·ligència artificial haurà d'estar programada de manera que en uns pocs canvis, intercanviar un jugador per la IA es pugui assolir d'una manera senzilla i instantània. Això s'aconseguirà realitzant una implementació amb jerarquia de classes.
- Aquest joc, un cop acabat, ha de tenir una modalitat multijugador preparada tant per dos com per quatre jugadors, que es desenvoluparà tenint com a base la versió d'un sol jugador.
- Amb l'estudi d'aquest treball s'ha de poder plasmar tant la dificultat com el temps requerit en la conversió d'un joc d'aquestes dimensions. Per tant es marcaran les diferents fases i processos necessaris per al desenvolupament de cadascun dels modes.

3.2 Objectius específics:

- Utilitzar una eina que faciliti la implementació amb Unity del joc multijugador, com és Photon Networking.
- Treballar amb una metodologia Scrum seguint una planificació de tasques per temps.
- Aconseguir un *matchmaking* sense interrupcions entre la cerca de jugadors, la creació de la sala i l'entrada a la partida.
- Utilitzar de Google Play Services per a la gestió de les puntuacions.
- Realitzar pantalles de càrrega entre escenes.
- Disposar d'uns atributs audiovisuals i estètics que puguin cridar l'atenció a l'usuari, adquirits implementant efectes visuals, sons i de partícules per a aconseguir una bona immersió.

4. Requeriments

Els requeriments d'aquest joc són diversos i estan estretament lligats amb els objectius marcats a l'inici. Els més importants i destacats són els següents:

- Realitzar el joc en mode d'un jugador.
- Desenvolupar una intel·ligència artificial que pugui jugar contra els altres usuaris.
- Aconseguir un mode multijugador que permeti simultàniament dos o quatre jugadors.
- Utilitzar el motor de joc Unity.
- Implementar una estètica futurista fent ús dels diferents recursos que ens ofereix la botiga de Unity.
- Utilitzar Photon Networking, una eina de desenvolupament per a Unity.
- Exportar el joc a la plataforma Android.
- Utilitzar GitLab com a eina de gestió de tasques del projecte i per crear un registre amb el seguiment del temps per a l'estudi de la conversió.

5. Game Design Document

5.1 Concepte

Nom: Crush Ball.

Gènere: Arcade.

Plataforma: Android.

Tipus de jugador: *Killer*⁴.

Públic objectiu:

- Estil: Jugadors competitiu.
- Edat: A partir de 7 anys, PEGI 7⁵.
- Gèneres: Esports, *arcade*.

Jugadors: Un jugador, dos jugadors online i quatre jugadors online.

Referències: Crash ball, Ping io.

Estètica: Futurista.

5.2 Objectius

- Ser l'últim de perdre dintre de la partida, i això s'aconsegueix no deixant que les vides arribin a zero.
- Pots utilitzar els poders per tal d'ajudar a marcar a l'adversari i així reduir les seves vides.
- S'ha d'intentar reduir les vides dels adversaris a zero a la vegada que et defenses dels altres contrincants.

5.3 Normes

- Els avatars estan restringits a moure's horitzontalment i no poden desplaçar-se cap endavant ni cap endarrere.
- Els avatars tenen restringit l'ús de l'habilitat especial en un interval de un segon, mai podent repetir la habilitat en aquest període.

⁴ *Killer*: Segons el test de Bartle, l'estudi de la classificació dels jugadors de videojocs, el perfil killer és el que busca competir amb altres jugadors directament.

⁵ PEGI 7: Indica que el joc és apte per a tots els públics però pot contenir imatges o sons que poden sobresaltar a una part dels espectadors.

5.4 Propietats dels jugadors

5.4.1 Poders

Els avatars disposen del poder d'empènyer la bola cap a la direcció on estan col·lidint i d'aquesta manera aconseguir un moviment més ràpid de la bola.

Aquest poder té una limitació, i és la restricció d'ús cada un segon. Idealment el jugador utilitzarà el poder del avatar sempre que la limitació no estigui activa.

El poder només serveix si hi ha un contacte directe entre almenys una bola i l'avatar, i per tant hi ha la dificultat que s'ha de sincronitzar bé el poder amb la col·lisió.

5.5 Jugabilitat:

5.5.1 Controls

El jugador en tot moment utilitzarà la pantalla tàctil del dispositiu mòbil per a desenvolupar les accions que vulgui dur a terme.

Els jugadors disposen d'un *slider*⁶ de moviment horitzontal el qual cada vegada que moguin amb el dit, podran moure el jugador. Caldrà utilitzar aquest element com a control per tal de causar una col·lisió entre la bola o boles i l'avatar, impedit així que les boles no entrin a la porteria pròpia però sí a la dels oponents.

Per aquest motiu, disposaran del moviment mitjançant aquest *slider* per a arribar a aquest fi.

Per tal d'utilitzar el poder especial només caldrà prémer la part superior de la pantalla.

⁶ *Slider*: Element d'interfície visual que serveix per a agafar valors de desplaçament horitzontal.

5.5.2 Mecànica del joc

Càmera: La càmera del joc és en tres dimensions, i està en una posició fixa que proporciona una visió de l'avatar i de tot l'escenari.

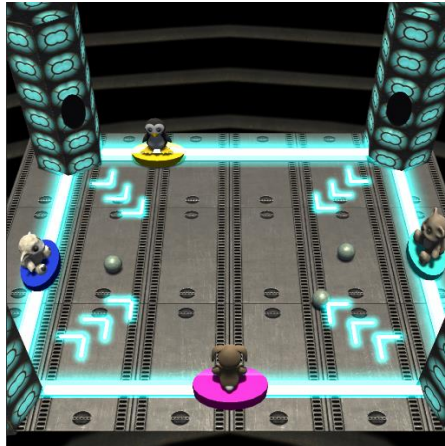


Figura 1: Exemple de la càmera del joc

Puntuació: Els jugadors disposen de deu vides. Cada cop que una bola travessa la seva línia de porteria el total de vides es veu disminuït en una unitat. Quan el total de vides arriben a zero, perden. També existeix un ranking entre usuaris en el mode multijugador que es basa en el número de vides que ha aconseguit mantenir al final el guanyador.

Guardar i carregar partida: Els jugadors, per a guardar i carregar les seves dades, disposaran d'una sincronització automàtica a través del seu compte de Google Play. L'únic procés manual és el d'inici de sessió als serveis de Google Play Games a l'iniciar el joc de manera obligatòria.

5.5.3 Reptes

Els reptes als quals s'enfronten els jugadors són diversos:

- Controlar l'ús dels poders dels oponents per tal de predir la trajectòria de les boles.
- Predir on cauen les boles, determinar on col·lidiran i intentar repel·lir-les.

5.5.4 Sancions

Quan el jugador no pot impedir que una bola travessi la línia de porteria, perd una vida. Aquesta sanció afecta al resultat de manera directe.

Quan ha perdut totes les vides i el seu marcador és zero, perd la partida i la possibilitat de continuar jugant.

5.6 Escenaris

S'han creat tres escenaris diferents en el joc, amb materials diferents però amb la mateixa dinàmica, només amb un canvi estètic.

En els dos primers escenaris, les boles surten disparades de uns pilars disposats a cada vèrtex del mapa. El tercer mapa es distingeix perquè en comptes de pilars hi ha canons que disparen les boles.

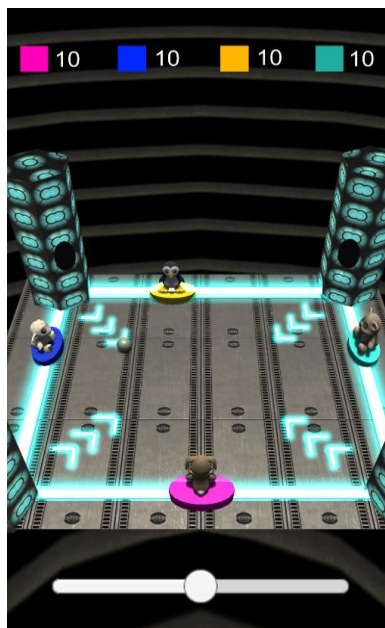


Figura 2: Exemple del primer mapa



Figura 3: Exemple del segon mapa



Figura 4: Exemple del tercer mapa

5.7 Interfície

La interfície del menú es compon de diferents botons que porten a diferents entorns segons el text que hi ha a cada botó. Per una banda tenim la modalitat *Single Player*, la modalitat *Multiplayer*, el menú de configuració *Settings* i l'*about*.

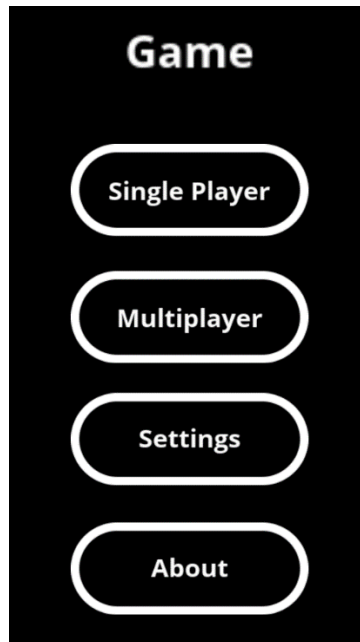


Figura 5: Interfície del menú

La interfície del joc quan la partida s'ha iniciat es compon de dues parts, un *slider* a la part inferior que determina el moviment de l'avatar i és interactiu per al jugador, i la puntuació a la part superior que es mostra amb colors segons la plataforma de cada avatar.



Figura 6: Interfície del joc quan la partida està en curs

Al perdre partida, apareixen dos botons nous a la interfície. Un serveix per retornar al menú principal i l'altre per tornar a iniciar partida.

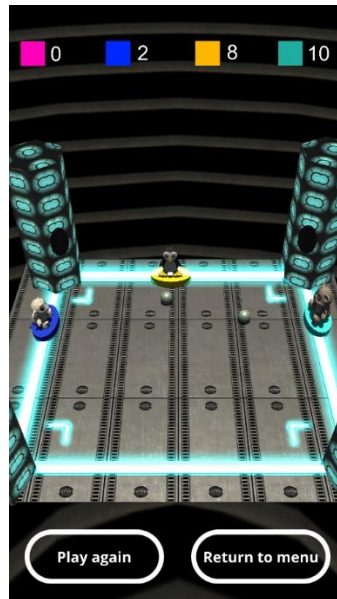
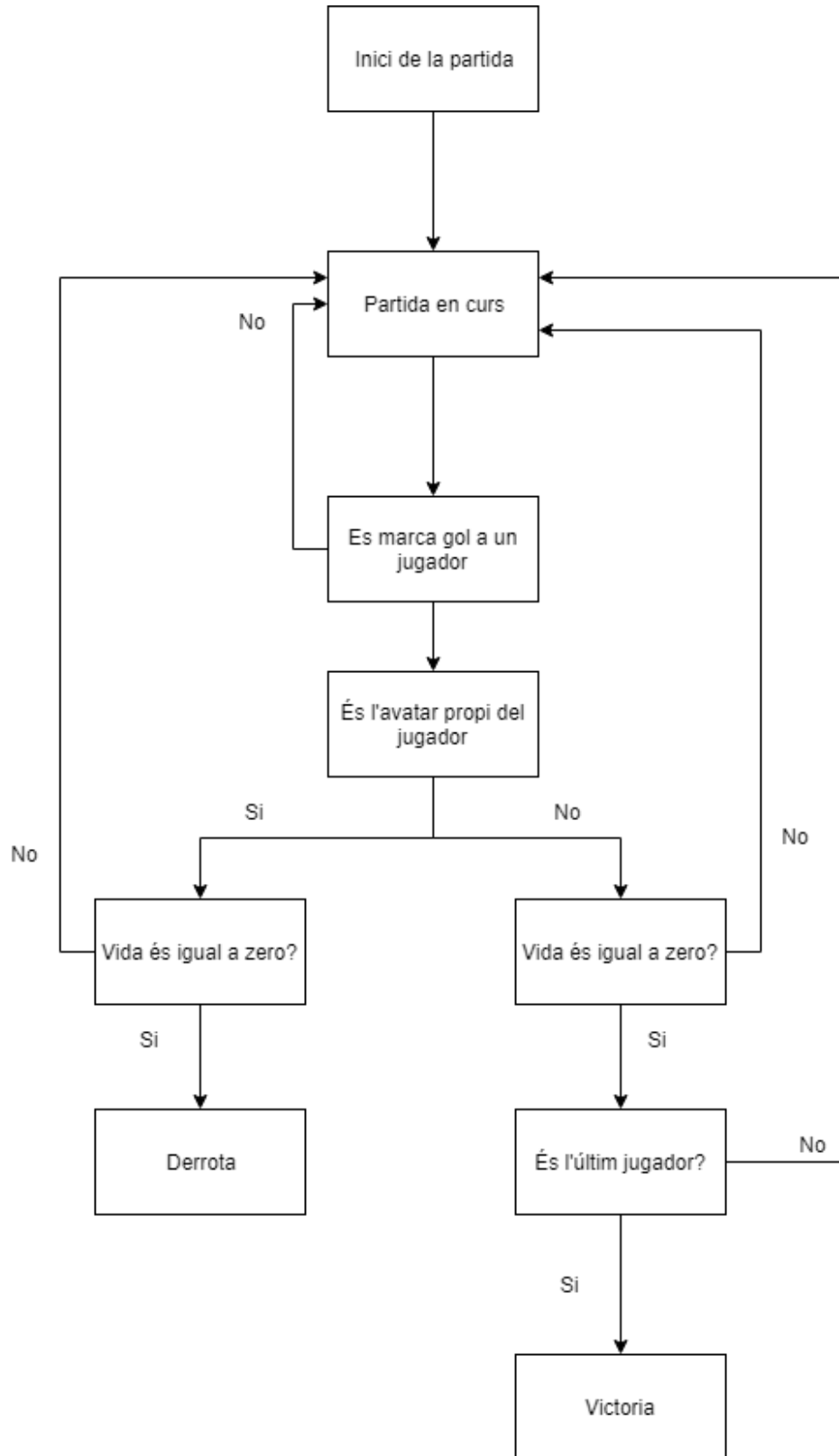


Figura 7: Interfície dels botons quan es perd la partida

5.7 Diagrama de funcionament



6. Implementació:

Per tal d'explicar la implementació, es divideix tot el que s'ha fet en dos grans blocs. El mode d'un jugador i el mode multijugador. Després es detalla en blocs més específics de cada una de les parts per poder abastar tot el procés de la creació del joc d'un jugador fins a la finalització de tota la lògica multijugador, passant per les fases de cada un dels apartats.

Primer es comença amb la implementació audiovisual, que és comuna en tots els modes i necessària per a entendre l'elecció i desenvolupament de tota la matèria audiovisual.

6.1 Implementació audiovisual:

Per a la realització d'aquest projecte, s'han agafat diferents *assets* ⁷ ja desenvolupats per a tal de crear tota l'estètica del joc. Aquests *assets* ens han permès preparar l'àmbit visual de manera fàcil i homogènia. S'han aconseguit de la botiga *asset store* ⁸ de Unity.

6.1.1 Menú:

Per a la creació del menú s'han utilitzat botons predefinits d'un paquet de *l'asset store* ⁹ de pagament:

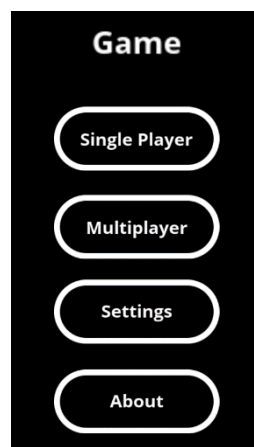


Figura 8: Menú del joc

⁷ Asset: És la representació d'un objecte el qual es pot utilitzar en el joc o projecte.

⁸ Asset Store: Botiga online de la plataforma Unity. <https://assetstore.unity.com/>

⁹ <https://assetstore.unity.com/packages/tools/gui/ui-builder-29757>

6.1.2 Avatars:

Els avatars han estat descarregats de la *asset store*¹⁰ de Unity gratuïtament:



Figura 9: Avatar 1



Figura 10: Avatar 2

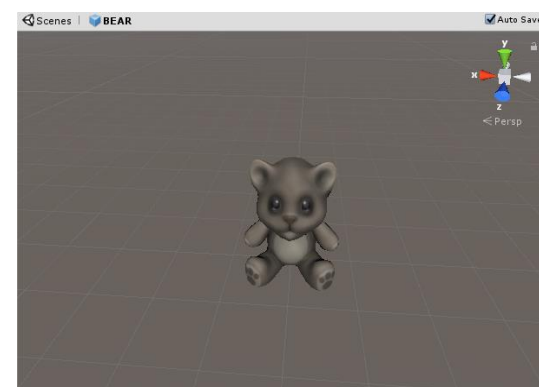


Figura 11: Avatar 3

¹⁰ <https://assetstore.unity.com/packages/3d/characters/6-x-3d-cute-toy-models-105033>

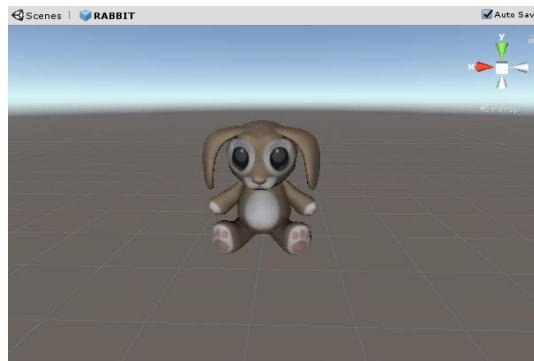


Figura 12: Avatar 4

6.1.3 Línies volumètriques:

Les línies volumètriques són ombrejats que utilitza la GPU ¹¹ basant-se en dos punts, el punt inicial del vèrtex i el punt final.

Aquestes línies volumètriques estan basades en una tècnica creada per Sébastien Hillaire, i es poden trobar a *l'asset store* ¹².

S'han utilitzat per a realitzar un efecte visual que destaquí amb la foscor de l'escenari, i per a crear una sensació de moviment.

També serveixen per identificar dos punts clau de l'escenari: la delimitació de les porteries de cada jugador i la indicació d'on surten les boles que posaran en joc la partida.

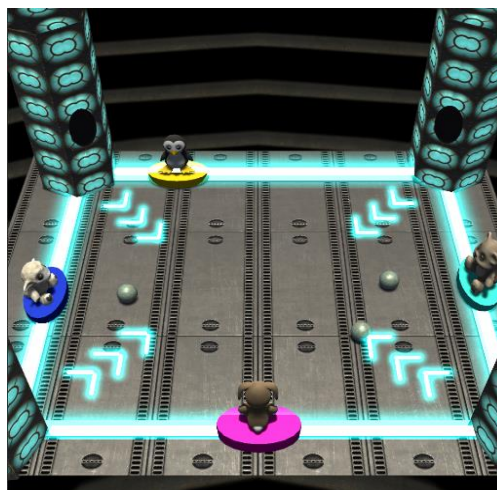


Figura 13: Exemple d'implementació de línies volumètriques

¹¹ GPU: És un coprocessador dedicat al processament de gràfics.

¹² <https://assetstore.unity.com/packages/tools/particles-effects/volumetric-lines-29160>

6.1.4 Pack d'explosions:

Aquest paquet de partícules de pagament es troba a *l'asset store*¹³ i conté més de cent explosions diferents. S'ha utilitzat l'explosió d'esfera per indicar quan un jugador es mor.

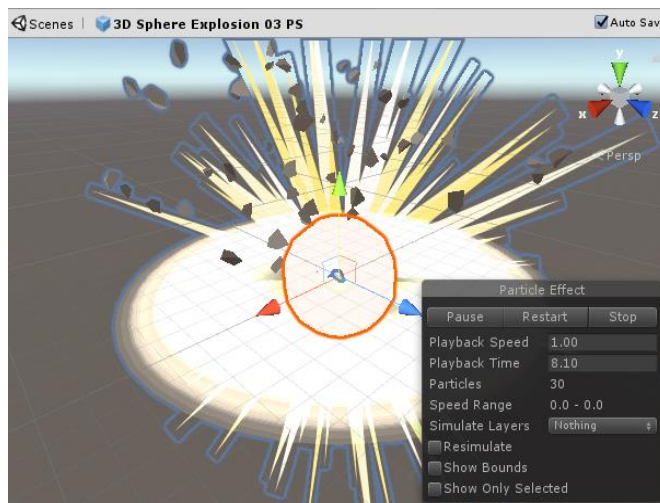


Figura 14: Explosió en esfera

També, d'aquest mateix paquet, s'ha utilitzat un sistema de partícules en forma d'ona per a marcar una habilitat dels jugadors, que és la d'empènyer la bola a més velocitat durant uns segons determinats.

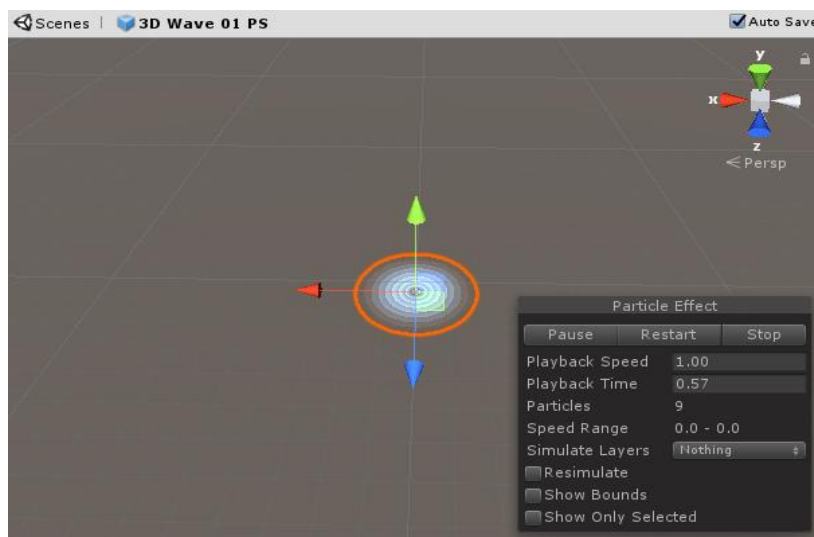


Figura 15: Explosió en ona

¹³ <https://assetstore.unity.com/packages/vfx/particles/spells/fx-mega-pack-2-14208>

6.1.5 Estètica de l'escenari:

S'ha utilitzat un paquet de textures de metall trobat a *l'asset store*¹⁴, que conté més de quaranta tipus de textures diferents.

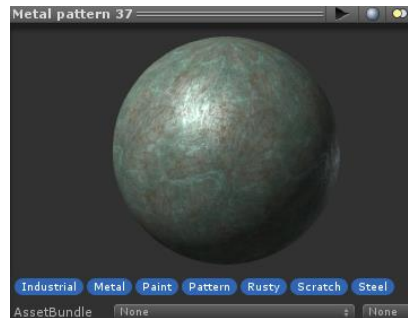


Figura 16: Exemple de la textura de metall número 37

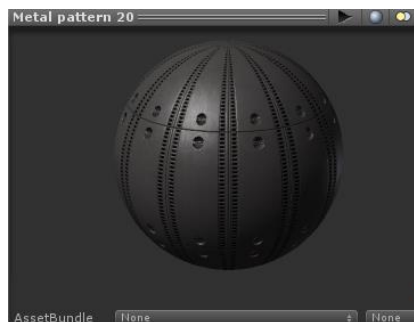


Figura 17: Exemple de la textura de metall número 20

Per a la realització dels pilars s'han unit textures metàl·liques a una columna també importada de *l'asset store* i amb els diferents elements crear una figura pròpia:

¹⁴ <https://assetstore.unity.com/packages/2d/textures-materials/metals/yughues-free-metal-materials-12949>

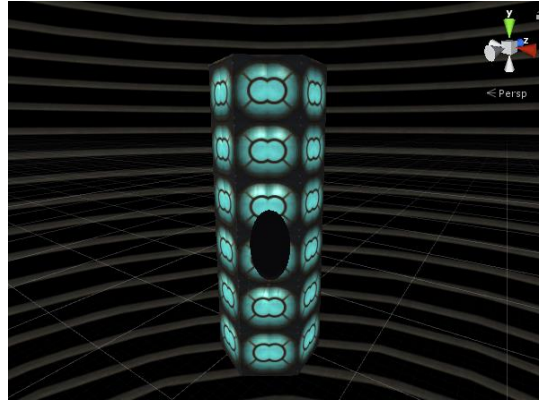


Figura 18: Exemple d'una textura aplicada a una columna.

S'ha utilitzat Blender per a fer una negació booleana entre el pilar i una figura per a tal de realitzar un forat que serà d'on sortiran les boles que posaran en joc la partida.

6.1.6 Murs:

Quan els jugadors moren, uns murs baixen a les seves posicions i fan de col·lisionadors, de manera que encara que mori un jugador, es pugui continuar la partida fins que només en quedi un.

Aquests murs baixen quan el jugador queda a zero vides, amb una animació que utilitza un desplaçament en un eix des d'una posició superior, fins a caure a la posició de l'escenari, bloquejant totes les boles que xoquin amb la seva superfície.

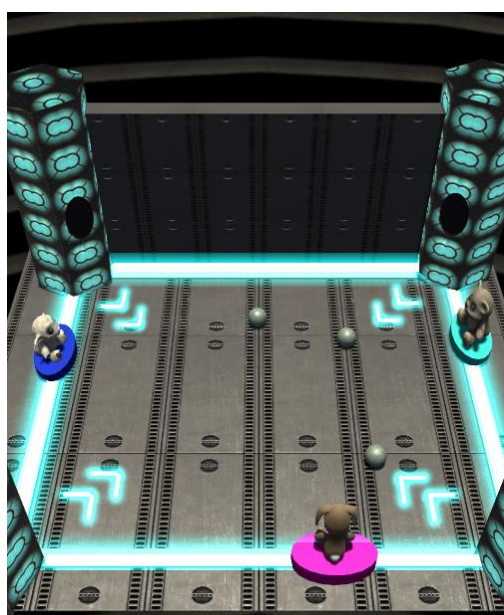


Figura 19: Exemple del mur que cau quan els avatars moren

6.1.7 Sons

S'han implementat diversos sons en aquest projecte, tots gratuïts i utilitzats dintre del joc. N'hi ha diversos, i s'esmentaran tots a continuació:

- **Música d'ambient:** S'ha utilitzat una música gratuïta que és Candlepower¹⁵, de Chris Zabriskie. Aquesta música s'utilitza per a l'ambientació, com a fons per a fomentar la immersió.
- **Col·lisions entre boles:** S'ha utilitzat un so de col·lisions quan les boles xoquen entre elles.
- **Col·lisions amb estructures:** S'ha utilitzat un so de col·lisions quan les boles xoquen contra estructures, per a produir un so més greu i intens.
- **So d'utilització del poder:** Quan els jugadors utilitzen els poders, es produeix un so totalment diferent de tots els altres per a remarcar que s'està utilitzant un poder, acompanyat de l'efecte visual del sistema de partícules.
- **So de marcar:** Quan la bola travessa la porteria de qualsevol jugador, es produeix un so per a remarcar que s'ha produït una resta a les vides d'aquell jugador.
- **So de mort:** Quan mor un jugador, es produeix un so d'explosió i baixa el mur, per tant s'ha cregut convenient utilitzar un so d'explosió en expansió per a remarcar aquest fet.

¹⁵ <https://chriszabriskie.bandcamp.com/track/candlepower>

7. Implementació de la programació

7.1 Lògica d'un jugador:

En aquest treball s'ha realitzat la lògica d'un jugador contra altres tres jugadors controlats per la màquina, per tant, ha implicat una realització i organització dels elements a desenvolupar, que es nombraran a continuació:

- Lògica d'inici de la partida: S'ha necessitat una lògica de càrrega de les escenes.
- Lògica de la partida, que es separa entre els controladors, *scripts*¹⁶ de entorn, *scripts* de les boles i *scripts* dels jugadors i de l'intel·ligència artificial.
- Una intel·ligència artificial senzilla que s'enfronta al jugador. Aquesta intel·ligència artificial s'encarrega de realitzar una cerca de totes les boles que hi ha en aquell moment a la partida, i seguidament detecta quina està més a prop en aquell moment. Quan ha detectat quina està més a prop, es dirigeix a la seva posició per a tal de glopejar-la.
- La lògica de finalització de la partida, és a dir, produir un seguit d'esdeveniments quan la puntuació arriba a zero, que són l'animació del mur, i si el jugador es queda sol i les intel·ligències artificials queden eliminades o el jugador queda eliminat, l'hi apareixen uns botons per tornar a començar partida o tornar al menú principal, mentre a la vegada pot veure el que succeeix a la partida amb total normalitat.

¹⁶ *Script*. És un document que conté instruccions escrites en codi de programació.

7.1.1 Lògica d'inici de la partida

Disposem d'un controlador de joc que gestiona tota la lògica de canvis d'escena. També gestiona la pantalla de càrrega que s'ha realitzat. Totes les ordres en l'àmbit de gestió de l'escena passen per aquest controlador, activant, desactivant i cridant els mètodes necessaris per al transcurs de la partida abans que s'iniciï.

L'usuari, així doncs, interactua amb els botons de la interfície visual per avançar cap a les escenes que ell vol, sigui el propi menú o el botó del començament de la partida, i tota la lògica interna passa per aquest controlador.

S'utilitza, per la transició entre escenes, un *script* que calcula el temps que tarda a carregar la següent escena, i llavors es crea un *canvas*¹⁷ que s'activa i mostra una barra de progrés amb aquest càlcul.

Això permet que hi hagi retroacció cap a l'usuari entre escena i escena, indicant que hi ha un temps de càrrega per a cada acció que faci. Si funcionés d'una altra manera podria donar la sensació a l'usuari que no s'ha produït cap acció al prémer el botó de canviar de escena.

7.1.2 Lògica de la partida

Quan es prem el botó de començar partida, els quatre jugadors es troben repartits en les seves posicions respectives. Es poden veure les diferents tasques en els diferents *scripts* que gestionen tota la lògica:

Controladors:

- GameController: Aquest *script* s'encarrega de gestionar totes les puntuacions i interaccionar amb l'UIController i l'AudioController.
- UIController: S'encarrega d'interactuar amb qualsevol element de la interfície visual, canviant els seus valors.
- AudioController: S'encarrega d'interactuar amb els diferents *AudioSources* de tots els elements i també conté totes les pistes de so que es volen reproduir a l'escena.
- PowerController: S'encarrega de gestionar el poder de l'usuari, que és empènyer la bola més ràpidament. Aquest *script* també limita el temps de recàrrega del poder en un segon.

Scripts de entorn:

¹⁷ Canvas: És l'àrea on tots els elements de la UI han d'estar incrustats.

- **DestroyBalls:** Aquest *script* s'encarrega de gestionar la destrucció de les boles quan passen per una porteria. També, quan es produeix aquesta destrucció, es crida al GameController perquè interaccioni amb l'AudioController i reproduïxi el so de destrucció de boles.
- **SoundCollision:** Aquest *script* assignat a diverses estructures del joc s'encarrega de cridar al GameController perquè interaccioni amb l'AudioController i així reproduir el so de les col·lisions.
- **Arrows:** Aquest *script* assignat a les línies volumètriques s'encarrega de desactivar-les i activar-les contínuament per donar la sensació de moviment, ja que entre totes tenen l'aparença d'una fletxa.

Scripts de les boles:

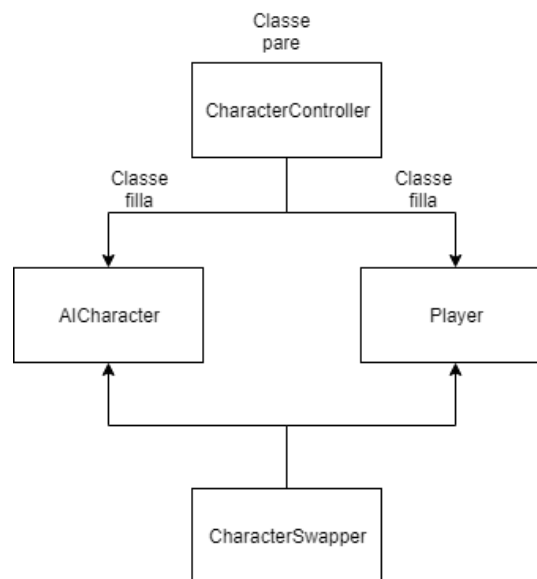
- **CollisionBalls:** Aquest *script* assignat a les boles s'encarrega de cridar al GameController perquè interaccioni amb l'AudioController i així reproduir el so d'impacte entre dues boles.
- **SpeedBalls:** Aquest *script* s'encarrega de que les boles sempre tinguin un valor de moviment mínim en la partida i mai es puguin parar. També s'encarrega de relacionar-se amb el PowerController i així poder augmentar la velocitat durant un temps determinat quan s'activa el poder, i fer que les boles es desplacin més ràpidament.

Scripts dels jugadors i de la intel·ligència artificial:

- **CharacterController:** Aquest *script* és una classe pare abstracta que es fa servir com a esquelet i de la qual l'*script* Player i l'*script* AICharacter hereten els seus mètodes.
- **AICharacter:** Aquest *script* és una classe filla de CharacterController que conté tota la lògica de la intel·ligència artificial que competeix contra els jugadors.
- **Player:** Aquest *script* és una classe filla de CharacterController que conté tota la lògica que serveix perquè els jugadors puguin controlar el seu personatge amb l'*Slider* i interactuar amb ell. Es basa directament en la posició del *Slider* per a ubicar-se en la posició horitzontal amb uns límits definits.

- CharacterSwapper: S'encarrega de canviar entre AI i jugador en el moment que es vulgui de la partida, instanciant cada una de les classes Player i AICharacter i cridant el mètode que les fa funcionar.

Diagrama de la gestió dels canvis entre jugador i AI



7.1.3 Intel·ligència artificial

La intel·ligència artificial que s'enfronta amb el jugador és un *script* que recorre en temps real totes les boles que hi ha a l'escenari, i comprova mitjançant diferents mètodes quina és la bola que està més a prop seu. Seguidament, es mou cap a la seva direcció a una velocitat fixa i para les boles.

De seguida que la bola s'allunya, va cap a la següent més a prop i així contínuament, intentant aturar totes les boles dirigides a ella.

També s'encarrega de delimitar els límits del moviment de la intel·ligència artificial i tenir en compte que n'hi ha una que es mou en l'eix horitzontal i les dues altres en l'eix vertical, ja que unes estan ubicades a l'esquerra i la dreta del jugador i un altre just davant i ha de tenir en compte aquestes restriccions.

7.1.4 Esdeveniments de finalització de la partida

Quan el marcador de tots els rivals arriba a zero o el marcador de l'usuari arriba a zero, apareixen dos elements nous a la interfície visual que substitueixen la part on està ubicat l'*slider*.

Els dos són dos botons situats en paral·lel. Un d'ells serveix per abandonar i tornar al menú principal i l'altre serveix per tornar a començar la partida. El controlador d'escena s'encarrega de gestionar aquesta interacció amb els botons i tornar a recarregar l'escena, jugant de nou, o tornant al menú principal on pot triar diferents opcions.

7.2 Lògica de multijugador:

Per a la conversió del joc a multijugador, s'ha hagut de tenir uns grans blocs molt presents que esmentarem a continuació:

- Inici de sessió amb Google Play Services
- Sistema de *matchmaking* o emparellament
- Sistema de creació de *lobbies* o sales
- Lògica multijugador dintre del joc
- Finalització de la partida
- Interacció amb Google Play Services per a la puntuació

Diagrama de funcionament amb dos jugadors

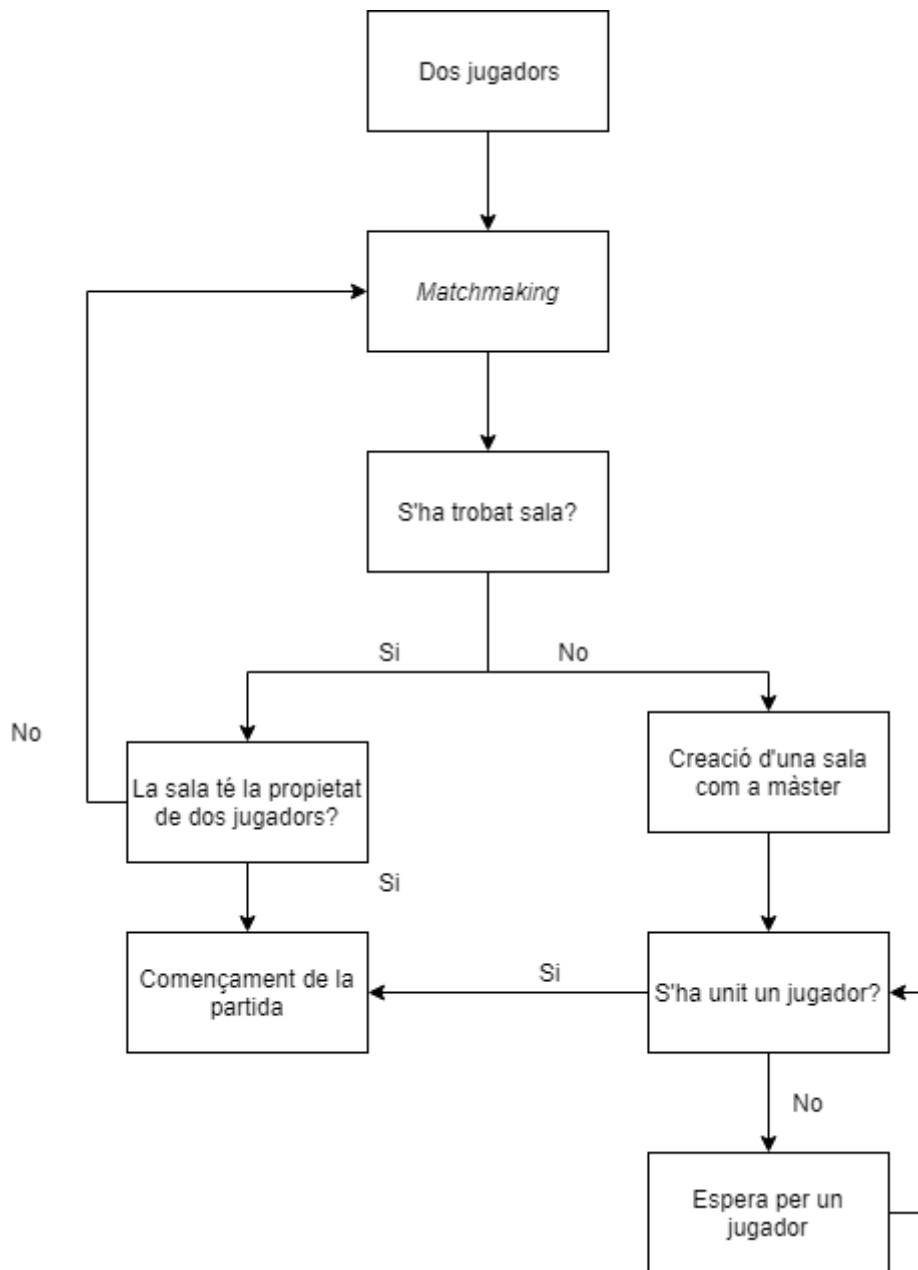
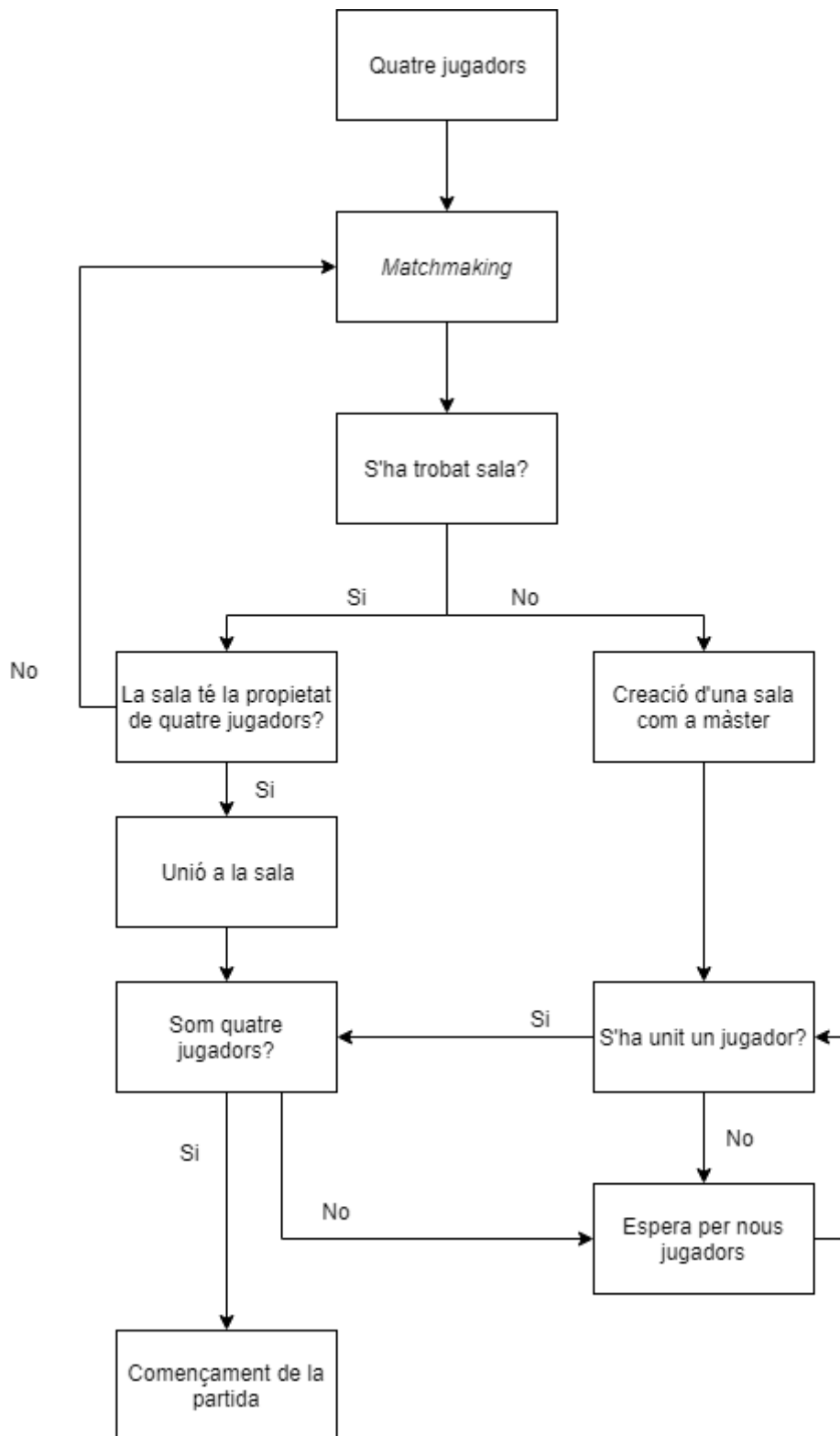


Diagrama de funcionament amb quatre jugadors



7.2.1 Inici de sessió amb Google Play Services

Primer de tot s'ha hagut d'instal·lar l'*SDK*¹⁸ de Google Play Services, per poder procedir a configurar l'inici de sessió. Seguidament, després de un procés de configuració llarg i de pagar la quantitat que demana Google per a ser membre desenvolupador de Google Play Services, es pot procedir a tenir el joc llest per fer-hi les proves necessàries.

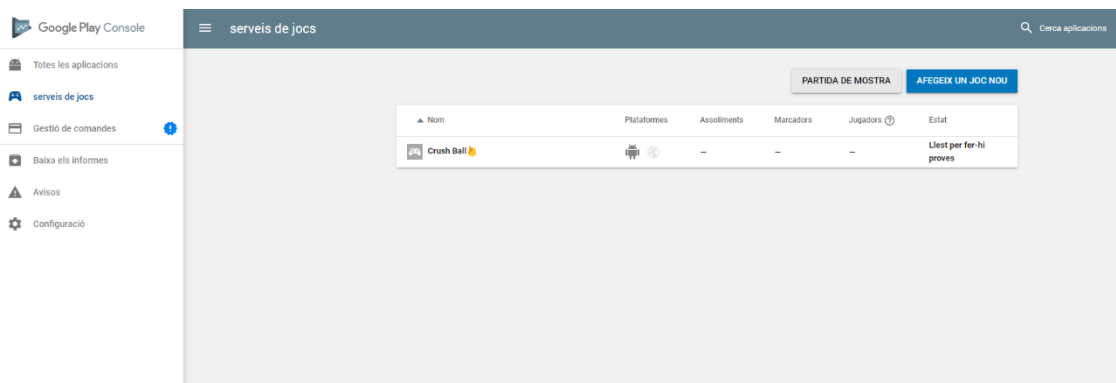


Figura 20: Panell de control de Google Play Services

A Unity, fent ús del *SDK*, es pot procedir a iniciar sessió amb Google Play Services i fer totes les funcions necessàries amb les comandes que ens ha proveït aquest conjunt d'eines de desenvolupament.

Cal dir que el joc està en versió de plataforma de proves, però que si es volgués publicar aquests passos previs són necessaris a la seva publicació. Per tant els passos fets en aquest apartat condueixen, en última instància, a la publicació del joc.

¹⁸ SDK: Conjunt d'eines de desenvolupament.

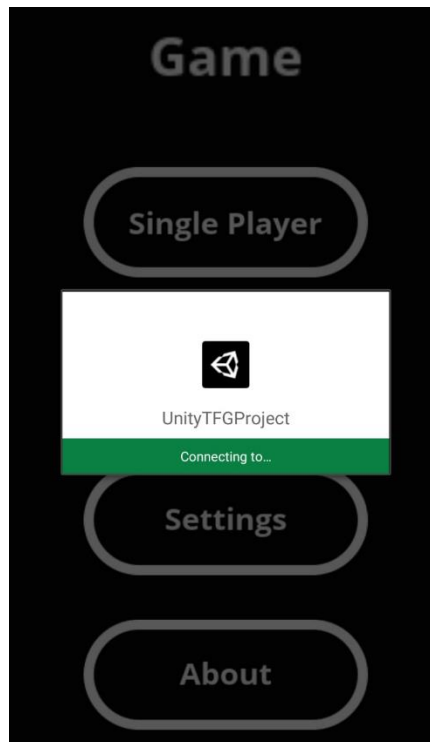


Figura 21: Inici de sessió amb Google Play Services.

7.2.2 Sistema de matchmaking o emparellament

Per a la realització del sistema d'emparellament, fet amb Photon Networking, s'ha hagut d'entendre el sistema automatitzat de Photon Networking per a emparellar els diferents usuaris.

No es fa servir un sistema d'emparellament o *matchmaking* arbitrari, sinó que va per ordre d'arribada sense cap distinció de jugadors o habilitat.

També cal dir que s'ha desenvolupat un sistema de *matchmaking* totalment automàtic, ja que les prestacions característiques d'aquest joc ho poden implementar. S'ha cregut que era millor un sistema que no es veiés el procés, ni la creació de cap tipus de sala, sinó que només en fer la interacció de clicar el botó de començar partida multijugador (sigui de dos o de quatre jugadors), es busqui partida sense cap feedback aparent a l'usuari de si es falla o no en trobar partida.

Els usuaris quan procedeixen a jugar, primer de tot busca quines sales hi ha actives i seguidament, si hi ha sales actives però no han arribat al límit de jugadors indicat. Finalment, s'uneix a la partida.

Si no es dona el cas que no hi ha cap sala activa o hi ha sales jugant però el límit de jugadors està al màxim, passarà a la creació d'un lobby o sala i esperarà a l'usuari.

S'ha de diferenciar, doncs, dos tipus de *matchmaking* totalment diferents que es poden implementar.

- *Matchmaking* totalment aleatori
- *Matchmaking* basat en habilitats de jugadors

El primer, el qual s'ha triat per a aquest projecte, utilitza un emparellament totalment aleatori i entra a qualsevol sala sense cap criteri sempre que compleixi els requisits que siguin sales de dos i quatre jugadors.

El segon sistema es basa a establir certes propietats de la sala i després només s'uneixen els jugadors que compleixin certes propietats.

Això es pot fer gràcies a determinar, durant la creació de les sales o *lobbies*, unes propietats que s'exigirà després que compleixin els jugadors que es vulguin unir.

Per un altra banda, en aquest cas s'ha hagut de tenir en compte per a aquest treball de final de grau és la diferenciació de dos i quatre jugadors, ja que la diferenciació ve marcada per les propietats que definim a les sales, i explicarem de manera més profunda en el següent apartat.

7.2.3 Sistema de creació de *lobbies* o sales:

Quan el *matchmaking* falla, és a dir, no troba cap jugador de les característiques que s'ha buscat, procedeix de manera automàtica a la creació de *lobbies* o sales per a tal d'esperar usuaris que s'uneixin i així començar la partida.

A més s'ha tingut en compte una propietat de la sala, com s'ha mencionat a l'apartat de *matchmaking*, que és si el jugador ha triat dos o quatre jugadors.

La sala, per tant, aplica aquesta distinció i quan nosaltres busquem partida i els jugadors triaran segons la modalitat que hagin escollit.

Aquest mètode serveix per a qualsevol propietat que es vulgui definir a la sala. Si es volgués afegir diferents propietats, com per exemple nivell d'habilitat, simplement hauríem d'aplicar el mateix mètode i afegir aquesta propietat a la sala per fer distinció. Quan els usuaris busquin partida s'ha d'identificar les sales que compleixin amb aquest requisit.

7.2.4 Lògica multijugador dintre del joc:

La lògica del joc multijugador quan la partida està en curs és extensa i complicada, i és el que ha ocupat la major part d'aquest treball de final de grau. Tota aquesta lògica es pot dividir en diferents *scripts* que s'han anat desenvolupant i als quals s'ha anat delegant les tasques i la lògica del joc multijugador.

L'avatar: L'avatar ha de seguir una lògica i un cicle de vida dintre del curs del joc en multijugador. Ha d'anar ubicat a una carpeta específica dintre del projecte com a *prefab*¹⁹ per a la instanciació d'aquest jugador dintre del joc.

S'ha hagut, per tant, de realitzar una lògica molt específica en aquest element prefabricat.

Primer de tot, l'objecte prefabricat és instanciat a l'escena des de la carpeta *resources*²⁰ de Photon Networking en el moment que l'escena es carrega, i aquest avatar ja està disponible per a la interacció amb el jugador. Això és un canvi directe respecte a la lògica d'un jugador per què no necessitàvem instanciar aquest avatar, ja estava actiu a dintre de la partida.

Quan el jugador es desconnecta per algun motiu (tancament forçat de l'aplicació) es crida el mètode automàtic de Photon Networking anomenat *OnLeaveRoom*, on es pot definir el que es vol que passi quan el jugador es desconnecta de manera forçada.

Aquest avatar ha de tenir, també, certs *scripts* relacionats amb la lògica multijugador que esmentarem a continuació:

- *Photon View*: Aquest *script* s'encarrega d'assignar un propietari a l'element el qual està assignat amb un identificador per al seguiment i sincronització d'aquest element en el servidor. Està connectat i lligat a un altre *script* anomenat *Photon Transform View*.
- *Photon Transform View*: Aquest *script* s'encarrega d'establir les propietats de l'objecte i en el seu comportament en la sincronització en l'àmbit de servidor. Es poden definir i activar propietats de sincronització de posició com: *Enable teleport for great distances*, *Interpolation (Ierp, Fixed Speed, Estimated Speed, Synchronize Values)*, *Extrapolation* i *Draw synchronized position error*. Totes aquestes propietats estan lligades directament a com l'avatar actualitzarà la seva posició en l'àmbit de servidor i com ho rebran els altres jugadors que estiguin a partida. Una bona configuració d'aquests paràmetres ajudarà, per tant, a actualitzar millor l'avatar durant la partida i que aquest tingui menys *lag* o latència.

¹⁹ Prefab: Element prefabricat amb unes propietats específiques.

²⁰ Resources: Carpeta ubicada a dintre de les diferents eines importades de Photon Networking.

- *Lag compensation*: Aquest *script* tracta d'una tècnica de correcció de *lag* o latència en l'àmbit de servidor. El que fa aquest *script* és, per tant, predir i compensar la latència generada per la falta de sincronització amb el servidor. Fa una predicció d'on anirà el jugador, veient el salt a la posició final i la posició inicial, i traça una ruta de posicions enviant paquets en la qual la posició final és el destí. Amb aquest *script*, per tant, eviten el que serien els salts que es produeixen de latència quan la connexió és dolenta i no s'actualitza bé la posició dels avatars en el servidor.
- *Player Networking*: En aquest *script* s'ha ubicat la gran part de la lògica del jugador, com és les seves propietats de marcador, el seu moviment o la seva ID. És molt important destacar que pel fet de ser un avatar multijugador i haver marcat a aquest com a tal, es desbloquegen mètodes com *PhotonNetwork.isMine*, mètodes que són essencials per a la lògica i identificació de les accions que es produeixen per cada un dels jugadors. Cal remarcar que al tenir quatre aplicacions corrent, cada una en un dispositiu, les accions es dupliquen quatre vegades, per tant, qualsevol lògica que es fa per cada jugador s'ha d'ubicar per a ell mateix i no pels altres, i així es fa gràcies a la propietat de *isMine*.

Comprovant si l'avatar pertany o no al jugador que està jugant, es pot procedir a qualsevol acció de moviment, gestió de punts, gestió d'usuaris o qualsevol mètode que es vulgui implementar. Pel contrari, tot el que féssim s'aplicaria a tots els avatars multijugador a la vegada i no es podria dur a terme la lògica de manera correcta.

També s'ha de remarcar el concepte de Màster. El concepte de Màster és l'avatar multijugador que entra primer a la partida, propietari de la sala, i el qual volem ficar lògiques que només s'executin una vegada durant la partida. Per exemple, la creació i instanciació del *prefab* o objecte prefabricat bola la dur a terme el màster de la partida, ja que volem que s'instanciïn una vegada en un interval de certs segons a tots els de la partida, i no produir aquesta instanciació quatre vegades diferents. Cal dir, també, que si marxa el jugador màster de la partida, s'assigna de manera automàtica un altre màster a la partida que dur a terme les mateixes funcions sense cap mena d'inconvenient.

Les boles: Per a la sincronització de les boles, també s'ha necessitat els elements *Photon View* i *Photon Transform View* anomenats prèviament a la part de l'avatar, ja que també necessitàvem sincronitzar les seves posicions i tenir un identificador ja que són elements que es mouen en l'àmbit de servidor per a tots els jugadors a la vegada. També s'ha implementat el mateix *script* de *lag compensation* per a tal de gestionar el *lag* o latència que produïen els endarreriments de connexió en l'àmbit de servidor. Aquestes boles també tenen un altre *script* semblant al del *lag compensation* per a la correcció de les col·lisions que funciona d'una manera semblant.

També s'ha de remarcar que aquestes boles són instanciades pel Màster de la partida i sincronitzades en l'àmbit de servidor cap a tots els usuaris.

Les boles han estat tractades d'una manera totalment diferent en comparació al tractament d'un jugador, ja que són elements que es sincronitzen a l'escena i tota la lògica ha de ser sincronitzada amb el servidor. En aquests elements recau la fluïdesa del joc, ja que són els elements que s'instancien i es mouen més a la partida, i això implica que siguin els elements que originen la major despesa a nivell de servidor.

UIPlayer: Aquest *script* serveix per a sincronitzar totes les dades de puntuació de l'interfície de l'usuari amb les del seu propi avatar. Per tant, les dades de l'interfície depenen directament de les dades que es guarden en cada un dels avatars de cada usuari, en l'*script* *PlayerNetworking* explicat anteriorment en l'avatar.

Aquestes dades de l'*script* de *PlayerNetworking* que després es mostren a la puntuació utilitzen un sistema anomenat *Remote Procedure Call*, que serveix per a llançar un mètode cap a tots els clients desde un propi client, en aquest cas el màster. Per tant, tenim un client màster que llança una *Remote Procedure Call* cap al servidor amb un mètode establert, i el servidor contacta amb tots els clients i activa aquest mètode per cada un d'ells.

Això ens serveix per actualitzar totes les puntuacions de tots els clients a la vegada en cada un dels respectius usuaris quan es marca un gol.

7.2.5 Finalització de la partida

Al finalitzar la partida, el jugador perd i pot deixar la sala en qualsevol moment, preparat per a la cerca d'una altra partida, tot i que pot seguir veient com els altres jugadors acaben.

Es desactiva el seu avatar i tota la interacció que hi té, i es produeixen els successos normals de la partida, que són el tancament del mur que delimita el seu perímetre.

En prémer el botó d'abandonar ja no pertany ni té cap implicació en el transcurs de la partida, i abandona la sala.

En aquest moment es produeix la interacció amb Google Play Services per a la gestió dels punts.

7.2.6 Interacció amb Google Play Services per a la puntuació

Dintre de Google Play Services, a l'apartat de marcadors, s'obté informació del nostre marcador amb les puntuacions dels usuaris.

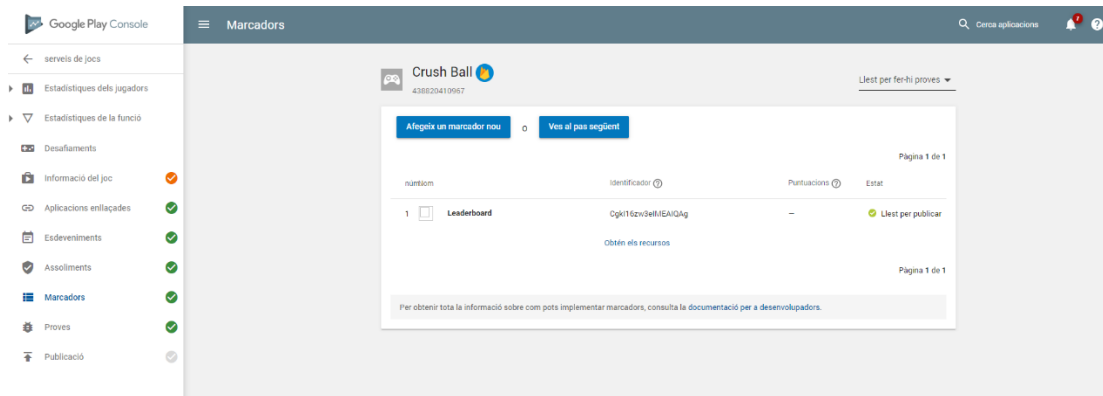


Figura 22: Consola de Google Play

El marcador gestiona les puntuacions de l'usuari, que són visibles a Google Play Games i després recopilades, si es vol, a dins de l'aplicació.

Un cop fet això, es poden mostrar als usuaris aquestes puntuacions. Perquè quan ja s'ha iniciat sessió, disposem de total control sobre de les dades que s'envien a Google Play Services, tant per modificar-les com per recuperar-les.

8. Planificació

La planificació ha sigut molt important per tal de portar un seguiment de totes les tasques i estudiar el temps que s'ha tardat en cada part d'aquest joc.

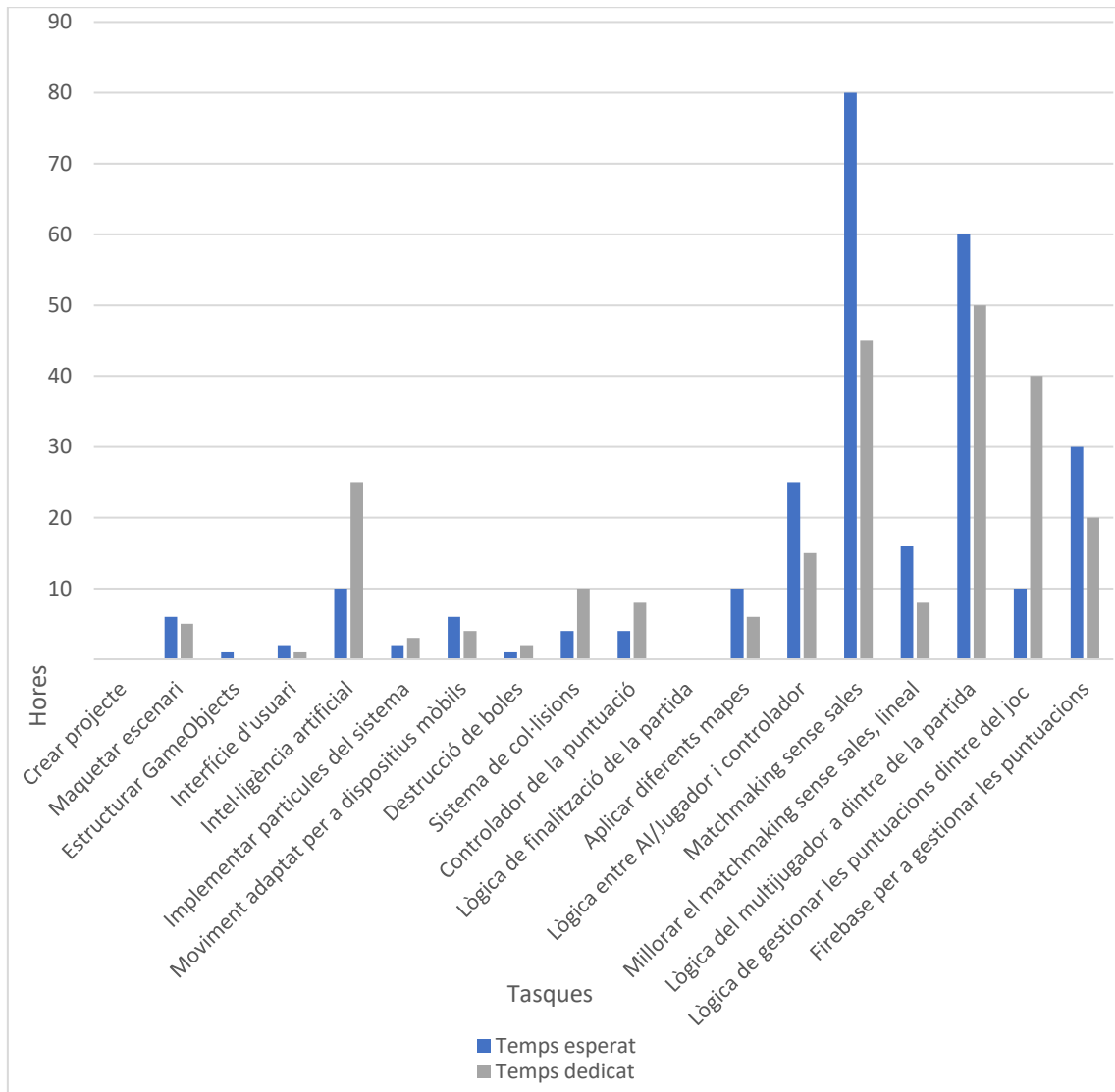
Així, la planificació es parteix en dues parts, la implementació del joc d'un jugador i després la conversió del mateix, sent l'última la que té una importància rellevant en termes d'estudi del temps que es tarda a convertir un joc d'aquestes característiques a multijugador.

Les tasques, per tant, s'han portat a terme a la plataforma Gitlab²¹ i són les que es representen a continuació, com a aproximació:

Tasques:	Temps esperat(hores)	Temps dedicat(hores)
Crear projecte	0.16	0.16
Maquetar escenari	6	5
Estructurar GameObjects	1	0.5
Interfície d'usuari	2	1
Intel·ligència artificial	10	25
Implementar partícules del sistema	2	3
Moviment adaptat per a dispositius mòbils	6	4
Destrucció de boles	1	2
Sistema de col·lisions	4	10
Controlador de la puntuació	4	8
Lògica de finalització de la partida	1.5	2.5
Aplicar diferents mapes	10	6
Lògica entre AI/Jugador i controlador	25	15
Matchmaking	80	45
Millorar el matchmaking sense sales, lineal	16	8
Lògica del multijugador a dintre de la partida	60	50
Lògica de gestionar les puntuacions dintre del joc	10	40
Google Services per a gestionar les puntuacions	30	20

També hi ha hagut moltes hores de investigació i reunions no incloses en aquest temps de desenvolupament de tasques.

²¹ Gitlab: Plataforma per a planificacions de projecte i gestió de codi. <https://gitlab.com/>



9. Pressupost

Hardware	Amortització	Cost
Ordinador de sobretaula	6 mesos	428,53€
Ordinador portàtil	6 mesos	212,85€
Dispositiu mòbil	6 mesos	102,42€

Total: 743,80€

Producte Asset Store Unity	Cost
UIBuilder	48,94€
FX Mega Pack	17,80€
Google Play Services	22,25€

Total: 88,99€

Cost de Personal	Hores totals	Preu Hora	Cost total
Programador	245.16	30	7.354,8€
Game Designer	60	25	1500€

Total: 8.854,8€

Suma total	Cost total
Hardware	743,80€
Llicències	88,99€
Personal	8.854,80€

Cost total del projecte: 9.687,59€

10. Conclusions

En realitzar aquest videojoc, s'ha après molt sobre el funcionament dels jocs multijugador i s'ha aprofundit en coneixements de les assignatures del grau de Multimèdia relacionades amb Unity.

En el desenvolupament del *game design document*, un element molt important en el plantejament de qualsevol videojoc, s'ha pogut plasmar el disseny del joc, les funcionalitats de cada part i quines característiques té en la seva totalitat.

La conversió a multijugador ha sigut extensa i complicada, i la seva transformació ha requerit plantejar de manera totalment diferent el concepte de desenvolupar un videojoc. Les tasques realitzades per aquesta conversió han consumit molta part de la totalitat del temps dedicat. Ha estat molt profitós portar un control de les tasques per tal de marcar els temps estimats i dedicats amb la plataforma Gitlab i és molt important per a qualsevol programador una gestió d'aquestes tasques.

Ha resultat molt difícil intentar plasmar tota la conversió i tots els conceptes del joc multijugador. La lògica que recau al servidor del qual no se'n té el control, implica molta investigació sobre com interaccionaran els elements durant la partida. El testeig també ha resultat complicat, al ser un joc multijugador requeria disposar de varis jugadors fent proves en els seus dispositius de manera simultània.

La conversió ha requerit cent seixanta tres hores aproximadament, un volum molt gran dins la totalitat del treball. Tot i que encara queda molt treball a realitzar en el mode multijugador, al qual l'hi queden moltes millores i molts factors a controlar (com pot ser la optimització quan l'usuari no té connexió o altres aspectes que s'han de tenir molt controlats) es pot dir que aquesta conversió ha estat molt profitosa i és totalment viable amb una major quantitat d'hores. Amb bona part dels coneixements apresos, els canvis seran molt més ràpids d'aplicar en un altre videojoc relacionat amb aquest àmbit.

Com a conclusió final, hi ha diferents implementacions de millora en un futur per a controlar tots aquests factors, però es pot dir que és viable la conversió de un joc d'aquestes característiques amb un temps aproximat superior a unes dues centes hores.

11. Bibliografia i webgrafia

Unity: <https://unity.com/es>

Unity Fòrum: <https://forum.unity.com/>

Photon Networking: <https://www.photonengine.com/en/pun>

Photon Networking Forum: <https://forum.photonengine.com/>

Google Play Services, documentació i *SDK*:

<https://developers.google.com/android/guides/overview>

<https://github.com/playgameservices/play-games-plugin-for-unity>

Creació d'un game design document:

<https://eldocumentalistaudiovisual.com/2015/02/06/documentacion-en-videojuegos-documento-de-diseno-gdd/>

https://www.gamasutra.com/view/feature/131632/creating_a%20_great_document.php

Test de Bartle per a la classificació de jugadors segons el seu perfil:

https://en.wikipedia.org/wiki/Bartle_taxonomy_of_player_types

Àudio:

<https://freesound.org/>

Càlcul de divises:

<https://www.google.com/search?q=EUR+to+USD>