



Tunnel Viking

Bernardo Gual de Torrella Cerdà
Grau en Enginyeria Informàtica

Consultor: Joel Servitja Feu

07 de gener del 2018



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

Títol del treball:	<i>Tunnel Viking</i>
Nom de l'autor:	<i>Bernardo Gual de Torrella Cerdà</i>
Nom del consultor:	<i>Joel Servitja Feu</i>
Data de lliurament (mm/aaaa):	<i>01/2018</i>
Àrea del Treball Final:	<i>Videojocs Educatius</i>
Titulació:	<i>Pla d'Estudis de l'Estudiant</i>
Resum del Treball (màxim 250 paraules):	
<p>El treball es enfocat a la realització d'un videojoc dins del gènere <i>endless runner</i>, en el qual el joc continua indefinidament fins que s'acaba el temps o les vides, en el cas del treball, les vides. L'objectiu d'aquest tipus de jocs, i per tant del joc del treball, es obtenir la millor puntuació possible. La temàtica del joc del treball es vikinga/medieval i està realitzat en Unity.</p> <p>El joc consta d'un menú principal i el nivell infinit. Aquest joc es basa en cilindres buits per els quals el jugador es mourà. En aquets cilindres es generaran aleatòriament obstacles a evitar, les monedes que augmentaran la puntuació i diversos <i>power-ups</i> que ens donaran certs beneficis. Quan es col·lisiona amb un obstacle ens resta puntuació.</p> <p>A més de les monedes s'aplicarà un multiplicador al temps per obtenir més puntuació. La dificultat anirà augmentant de dues maneres: per distancia recorreguda i quan s'arribi a dues puntuacions específiques. A més, quan augmenta la dificultat, també augmenta el valor de les monedes i canvia la distribució dels obstacles i la taxa d'aparició dels <i>power-ups</i>.</p> <p>Els <i>power-ups</i> consten de: vides addicionals, un escut que ens protegeix d'una col·lisió, un multiplicador que augmentarà el multiplicador del temps i dues fletxes, una vermella que augmenta temporalment la velocitat del jugador i una altra verda que el fa més lent.</p> <p>Quan s'acaben les vides el joc terminarà i calcularà la puntuació total tenint en compte les monedes i el temps que s'ha estat jugant.</p>	

Abstract (in English, 250 words or less):

This project is focused in the development of an endless runner video-game, where the game continues indefinitely until the player runs out of lives. The goal of this game is to obtain a high score. The theme of the game is Norse/ Dark Ages and is developed in Unity.

The game consists of a main menu and the infinite level. The infinite level has a number of hollow cylinders where the player moves. In those cylinders spawn the obstacles to avoid, the coins to pick and the power-ups that affect the player. Crashing into an obstacle subtracts on the score.

In addition to the coins, there are a multiplier applied to the total game time to calculate the total score. The difficulty will increase by two objectives: by distance and by score. When the difficulty rises, the coin value also rises and the spawning of the obstacles and power-ups.

These power-ups consist of: additional lives, a shield that prevents losing a life and score when colliding, a multiplier that increments the time multiplier and 2 arrows, one red that increments the player's speed, and a green one decreasing the speed.

When there are no remaining live the game ends and proceeds to calculate the final score.

Paraules clau (entre 4 i 8):

Endless runner Tunnel Viking

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball	1
1.2 Objectius del Treball.....	1
1.3 Enfocament i mètode seguit	1
1.4 Planificació del Treball.....	2
1.5 Breu sumari de productes obtinguts.....	4
1.6 Breu descripció dels altres capítols de la memòria	4
2. Disseny.....	5
2.1 Idea del joc	5
2.2 Conceptualització	8
2.3 Desenvolupament	9
3. Implementació	10
3.1 Introducció.....	10
3.2 Controladors.....	10
3.3 Escenari del joc	12
3.4 Els obstacles	14
3.5 Els <i>collectibles</i>	15
3.6 El jugador	17
3.7 Configuració de la dificultat	19
3.8 GUI.....	24
3.9 Aparença visual i sons.....	27
4. Conclusions.....	28
5. Glossari	30
6. Bibliografia.....	31
7. Credits	32
7.1 So.....	32
7.2 Imatges.....	32
7.4 Tests.....	32

Llista de figures

1: Diagrama de Gantt del Treball	3
2: Captura del minijoc del Sonic 2	6
3: Captura del joc Jetpack Joyride	6
4: Els 10 cilindres	12
5: Cilindre i el seu <i>collider</i> , la caixa verda	13
6: Els 4 tipus d'obstacles	14
7: Els <i>power-ups</i>	15
8: Gràfics de l'escut	16
9: El jugador i la zona jugable	17
10 Animació de col·lisió	18
11: Wall2 i el segons obstacle a 45°	20
12: Wall 3 amb un simètric en el primer obstacle	21
13: Wall3 amb la diferencia de 20°	22
14: El menú principal	24
15: el Hud del joc	25
16: Pantalla de Game Over	26

1. Introducció

1.1 Context i justificació del Treball

Els videojocs son una forma de oci digital popular i amb gran nombre de consumidors. Per tant es un camp d'estudi i de negoci interessant per a la realització d'una carrera universitària.

En el treball s'ha realitzat un videojoc basat en un gènere bastant popular, sobre tot en smartphones, el *endless runner*. En aquest gènere es consisteix en arribar a la millor puntuació esquivant obstacles i agafant *items*. L'ideal seria desenvolupar un joc que serveixi tant per a un ordinador domèstic o un smartphone i per això al joc s'han implementat tant controls per a teclat i pantalla tàctil.

Per a realitzar aquests punts s'ha optat per utilitzar Unity[1] com a motor gràfic.

1.2 Objectius del Treball

1. Capacitat per planificar i gestionar un projecte
2. Aprendre a dissenyar una idea de videojoc
3. Aprendre a implementar un videojoc
4. Aprendre a utilitzar un motor gràfic i les seves eines

1.3 Enfocament i mètode seguit

Com a opcions per a realitzar el treball tenim que es podria fer un joc innovador, sense estar basat en cap altre o intentant crear un gènere nou, o fer un joc basat en un gènere existent (en aquest cas, un *endless runner*) i tractant de donar-li una mica d'innovació. Aquest darrer cas es el que s'ha realitzat per al treball de final de grau, fer un joc de un gènere popular i tractant de donar-li una mica d'originalitat fent-ho en 3D ja que hi ha més quantitat d'aquests jocs en 2D.

El tractar de fer un joc completament innovar en el món actual on ja s'han realitzat milers de jocs pot resultar molt difícil o directament impossible.

Per tant s'ha decidit realitzar un joc basat en un tipus de jugabilitat ja existent, però tractant de donar-li una mica d'originalitat per a no crear una simple còpia.

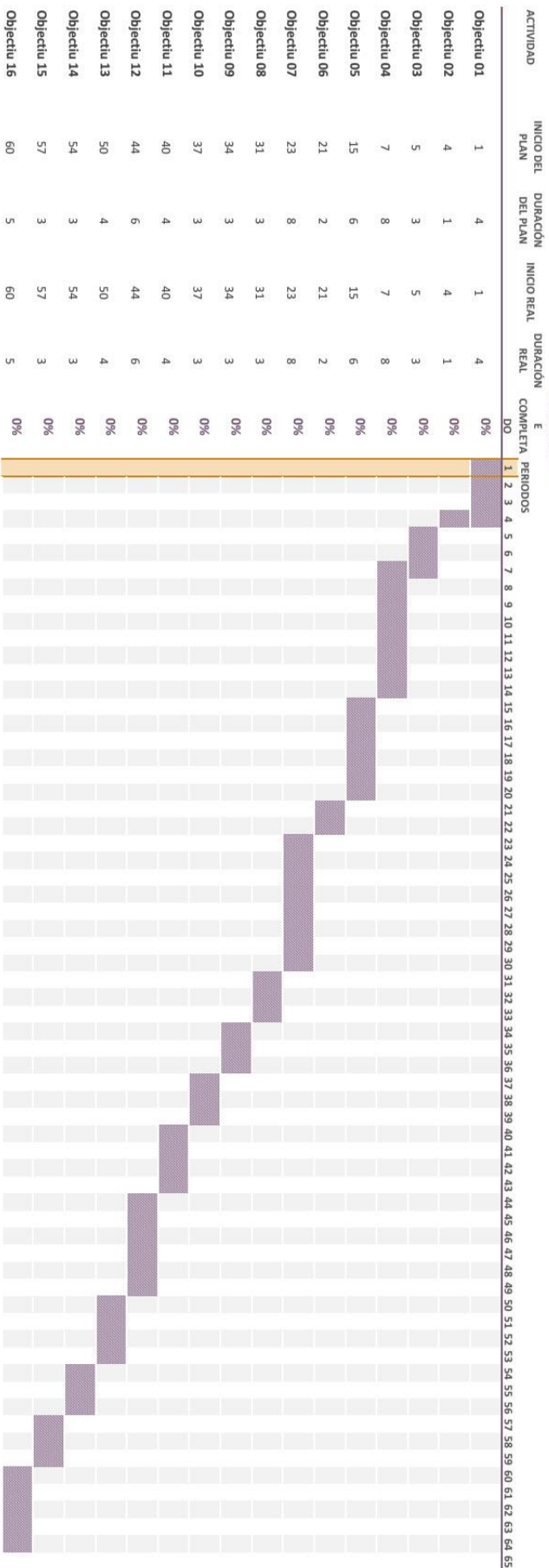
1.4 Planificació del Treball

El treball s'ha dividit en 4 entregues i aquestes en objectius a realitzar:

- PAC1, disseny:
 - Obtenir la idea del joc
 - Conceptualització
 - Planificació
- PAC2, versió parcial:
 - O1: Generació d'un nivell infinit sense obstacles
 - O2: Generació d'un objecte que serà el jugador
 - O3: Programar el moviment del jugador pensant en controls mòbils (fent clic al costat esquerre per anar a la esquerra, etc...)
 - O4: Afegir els obstacles de manera semi-aleatòria (evitant casos on sigui impossible avançar)
 - Els principals obstacles serien porcions de murada o murs complets amb un únic espai per passar
 - O5: Afegir la programació per a que el jugador col·lisió amb els obstacles i perdi una vida, a més de programar el sistema de vides
 - O6: Afegir un petit menú principal i una pantalla de final de joc per a poder a començar a jugar
 - O7: Inclusió dels objectes que augmenten la puntuació i la programació del sistema de puntuació (els multiplicadors per distancia i la quantitat de punts que afegeix obtenir un objecte)
 - També, per a donar importància a un objecte que s'implementarà més endavant, restar puntuació al col·lidir amb un obstacle
- PAC3, versió jugable:
 - O8: Implementar l'augment de velocitat del jugador en el temps per a augmentar la dificultat
 - O9: Implementar un power-up que ens permeti recuperar una vida
 - O10: Implementar un power-up que augmenti el multiplicador de la puntuació actual
 - O11: Implementar un power-up que ens permeti impactar amb un obstacle
 - O12: Canviar tots els elements placeholder per altres que siguin finals (el jugador per un viking, afegir textures)
 - O13: Animació del jugador
 - O14: Implementació d'altres possibles power-up
 - O15: Afegir distintes aparences per als obstacles
 - O16: Afegir animacions als obstacles
- PAC4, versió final:
 - Realitzar les millores proposades per el professor.

TunnelViking

Seleccione un periodo para resaltarlo a la derecha. A continuación hay una leyenda. **Periodo mes 1.**



1: Diagrama de Gantt del Treball

1.5 Breu sumari de productes obtinguts

El joc consta de 2 nivells: un petit menú principal i el nivell infinit que es la base del joc

El nivell infinit es una sèrie de cilindres buits on el jugador es desplaça evitant els obstacles i agafant objectes que l'ajudaran a obtenir una millor puntuació fins que ens quedem sense vides.

Aquesta puntuació augmenta cada vegada que s'agafa una moneda i al final del joc també té en compte la quantitat de temps que hem estat jugant i el multiplica per un valor que augmenta per distancia recorreguda pel jugador i també gràcies a un *power-up*.

Però també disminueix cada vegada que col·lionem amb un obstacle, juntament amb una vida.

Si perdem totes les vides el joc acabarà i ens mostrarà una pantalla on es veuran les distintes parts que conformen la puntuació final.

1.6 Breu descripció dels altres capítols de la memòria

- Disseny: com es va arribar a l'idea del joc i les decisions de motor gràfic i plataforma
- Implementació: explicació dels elements que conformen el joc i les decisions preses.
 - Controladors: explicació els scripts del joc.
 - Escenari de joc: com funcionen els cilindres que conformen el món del joc.
 - Obstacles: explicació del "enemic" del joc.
 - *Collectibles*: com funcionen els objectes trobats al joc i com interactuen amb el jugador.
 - Jugador: com es controla i com interactua amb el món
 - Configuració de la dificultat: com s'incrementa la dificultat i com canvien els processos de generació d'obstacles i objectes quan aquesta augmenta.
 - GUI: explicació de la interfície d'usuari.
 - Aparença visual: per què es varen tomar les decisions sobre els gràfics.

2. Disseny

2.1 Idea del joc

2.1.1 Descripció

El joc “Tunnel Viking” es un “*endless runner*” 3D per a ordinador o mòbil amb diversos *power-ups* que proporcionaran al jugador diversos poders per a superar els diferents obstacles que hi haurà als nivells. El joc tindrà un únic nivell cilíndric que serà infinit i que tindrà obstacles generats proceduralment.

La perspectiva de càmera serà en 3ra persona i dirigida cap a la direcció del moviment. El jugador podrà moure's 90 graus cap als costats per a poder esquivar els obstacles. L'objectiu serà obtenir la major quantitat de punts possibles. Aquests punts s'obtingran de dues maneres: agafant les monedes que hi ha al nivell i per quantitat de temps jugat. La dificultat anirà augmentant en el temps, augmentant la velocitat amb la qual el jugador avançarà per el nivell, modificant l'aparició d'obstacles i *power-ups* i augmentant un multiplicador que afectarà a la puntuació total.

2.1.2 Subgènere i referències a videojocs existents

Com s'ha comentat amb anterioritat, el gènere es un “*endless runner*”, es a dir, un joc on el jugador sempre es en moviment i haurà d'esquivar els obstacles per anant obtenint punts. Com inspiració de la perspectiva de càmera i jugabilitat s'ha utilitzat el mini-joc per aconseguir les esmeraldes del caos que es troba al joc de Sega Mega-Drive Sonic 2[2].

Com al mini-joc el jugador recorrerà un escenari cilíndric i es podrà desplaçar per el perímetre del cilindre. A diferencia d'aquest mini-joc, no hi haurà opció de saltar ni tampoc es podrà anar pel sostre. Pel que fa a la gravetat només afectarà al jugador quan estigui en un extrem del cilindre(-90 o 90 graus) i no continuï movent-se cap a aquet extrem, es a dir, que si no continuem prement la tecla dreta (per exemple) el jugador automàticament retornarà a la posició neutra.

A més per facilitar la jugabilitat, ja que es anirà augmentant la dificultat, no hi haurà ni girs ni corbes.



2: Captura del minijoc del Sonic 2

A més, s'agafa com inspiració per als power-ups un "endless runner" conegut com Jetpack Joyride de Halfbrick[3]. D'aquest joc s'extreuen la possibilitat d'agafar power-ups per a facilitar un poc el joc, la velocitat en augment i la mecànica d'agafar objectes a la vegada d'esquivar obstacles. Però, a diferència del joc original, a més de que el joc no serà un scroller lateral, es que les monedes no serviran per a comprar millores, si no per a millorar la puntuació.



3: Captura del joc Jetpack Joyride

2.1.3 Tipus d'interacció joc-jugador

El jugador desplaçarà el personatge a la dreta o a la esquerra per a evitar els obstacles. Prement la fletxa dreta del teclat ens mourà cap a la dreta i la fletxa esquerra cap a la esquerra. Per iniciar el moviment s'ha de polsar la barra espaiadora. Pel que fa a la interacció amb els menús es fa amb el ratolí.

2.1.4 Plataforma de destí

La plataforma triada es l'ordinador, ja que es una plataforma estesa i fàcil de manejar i en la que desenvolupar jocs. Es important comentar que s'ha tingut la consideració de implementar controls per a mòbil ja que el joc es ideal per a jugar-lo en un smartphone. Tothom té un smartphone i hi ha un augment de les persones que juguen amb el telèfon. A més que el format de joc de "*endless runner*" es més popular a les plataformes mòbils que als ordinadors personals. També es té en compte que només es necessari moure's o dreta o esquerra, el joc es pot escalar perfectament a un mòbil per la senzillesa dels controls.

2.2 Conceptualització

2.2.1 Història, ambientació i/o trama

La història serà senzilla: un viking ha de recórrer un túnel agafant objectes per guanyar-se el favor dels déus del Valhalla. La temàtica serà nòrdica o de l'època medieval, però els items seran d'aparença genèrica per a no confondre al jugador.

2.2.2 Definició dels personatges/elements

- Jugador: Un viking que tracta d'arribar al Valhalla. Per motius de temps s'ha utilitzat un model gratuït[4] de la Asset Store de Unity[5]. Tindrà 3 vides, per tant podrà col·lisionar amb els obstacles 2 vegades abans de terminar el joc.
- Túnel/nivell: Es un cilindre buit on aniran els obstacles i el jugador es mourà.
- Seccions de nivell: Son les seccions en les quals estarà dividit el nivell. Cada 20 seccions completades augmentarà el multiplicador del temps i cada 30 la velocitat del jugador.
- Obstacles: elements al nivell que si fem contacte amb ells s'acaba la partida o perdem una vida.
- Objectes de puntuació: Son els objectes que augmenten la nostra puntuació, en el joc, son monedes.
- Power-ups: Diferents elements que ens proporcionen millores o ajudes per a progressar per el nivell

2.2.3 Interacció entre els actors del joc

El jugador interactuarà amb els obstacles de manera senzilla: si col·lisiona amb un obstacle, perd una vida i una quantitat de la puntuació que depèn de la dificultat actual. Si perd totes 3, el joc acaba i mostra una pantalla amb el resum de la puntuació.

Quan el jugador obté un *power-up* el utilitzarà automàticament. Aquets *power-ups* son: un escut que impedeix que es perdi una vida o puntuació al col·lisionar amb un obstacle, cors o altres objectes per recuperar una vida, multiplicadors que augmentaran la quantitat de punts que obtenim per el temps jugat, una fletxa verda que reduirà temporalment la velocitat del jugador i una fletxa vermella que serà un power-up maliciós que augmentarà temporalment la velocitat.

2.2.4 Objectius plantejats al jugador

Obtenir una millor puntuació cada vegada que es jugui al joc. Aquesta puntuació serà més gran quants més monedes, multiplicadors i quan més temps duri el joc.

2.3 Desenvolupament

2.3.1 Evaluació d'engines i kits de desenvolupament

CryEngine

Aquest motor té moltes característiques interessants i una gran potencia per que fa als aspectes tècnics dels jocs. El kit de desenvolupament "sandbox" que conté sembla molt potent, però el principal problema d'aquest motor gràfic es que no suporta el mòbil com a plataforma i com s'ha comentat abans es vol pensar en el mòbil com a futura plataforma. Llicència gratuïta per a estudiants.

Unreal Engine 4

Un altre motor que permet gran potencia tècnica i té un kit de desenvolupament que no necessita conèixer programació. També es important la quantitat de tutorials que hi ha a la xarxa. Permet fer jocs per a mòbil. Es té una mica d'experiència amb una versió anterior d'aquest motor. També es de llicència gratuïta per a estudiants.

Unity

Tècnicament no es tan impressionant com els altres 2 però permet fer jocs a un nivell bastant alt. Es el motor gràfic que té major suport de plataformes. Té integrats sistemes de monetització per a jocs mòbils. Hi ha gran quantitat de tutorials a la xarxa. El kit de desenvolupament es senzill i permet fer tot el necessari per a fer un joc mòbil.

Tenint en compte aquests detalls el motor triat per al desenvolupament del projecte es Unity..

3. Implementació

3.1 Introducció

Per a facilitar la comprensió d'aquest capítol la implementació s'explicarà variant l'ordre dels objectius del treball exposats al punt 1.4

3.2 Controladors

Els controladors son elements sense aparença física que contenen els scripts que controlen el joc. S'entrarà en detall de les distintes funcions en els capítols enfocats als distintes elements del joc. Els més rellevants son:

3.2.1 GameController

Es el nexa del joc, que inicialitza el joc, controla la part de la GUI que conté la puntuació i les vides i a més conté funcions que controlen la música del joc. Tots els elements estan enllaçats aquí, ja que té les operacions necessàries per transmetre informació entre els distintes controladors i elements del joc.

3.2.2 Timer

Controlador que conté les rutines per als diferents temporitzadors dels *power-ups* i que controla el temps total de joc. També controla la visibilitat dels panells de la GUI fan ús d'una conta enrere com els *power-ups* de l'escut, d'alentiment i d'acceleració del jugador.

3.2.3 PlayerController

On es controla el moviment del jugador i les interaccions amb els *collectibles* i els sons que produeixen. També controla la quantitat de vides que té el jugador i avisa al GameController de que ha de modificar la GUI quan s'agafen els *power-ups* i monedes, quan es perd una vida i que el joc acaba quan s'arriba a 0 vides.

3.2.4 TubeController

Es un controlador clau, ja que es el que controla la generació dels obstacles i dels *power-ups* als cilindres. També controla la generació indefinida del nivell, reposicionant els cilindres quan han sigut travessats.

3.2.5 WallController

Es el controlador dels obstacles. Detecta quan el jugador ha col·lisionat i li ordena perdre una vida i aturar-se.

3.2.6 CrashController

Controla la animació del jugador al col·lisionar amb un obstacle. Aquesta animació es un *sprite* o imatge 2D que apareix darrera el jugador i dona l'efecte de col·lisió.

3.2.7 GameOverController

Aquest controlador s'activa quan el joc ha finalitzat i rep tota la informació necessària per a donar els resultats totals de la puntuació per part del GameController. A més mostra 2 botons per a reiniciar la partida o per tornar al menú principal.

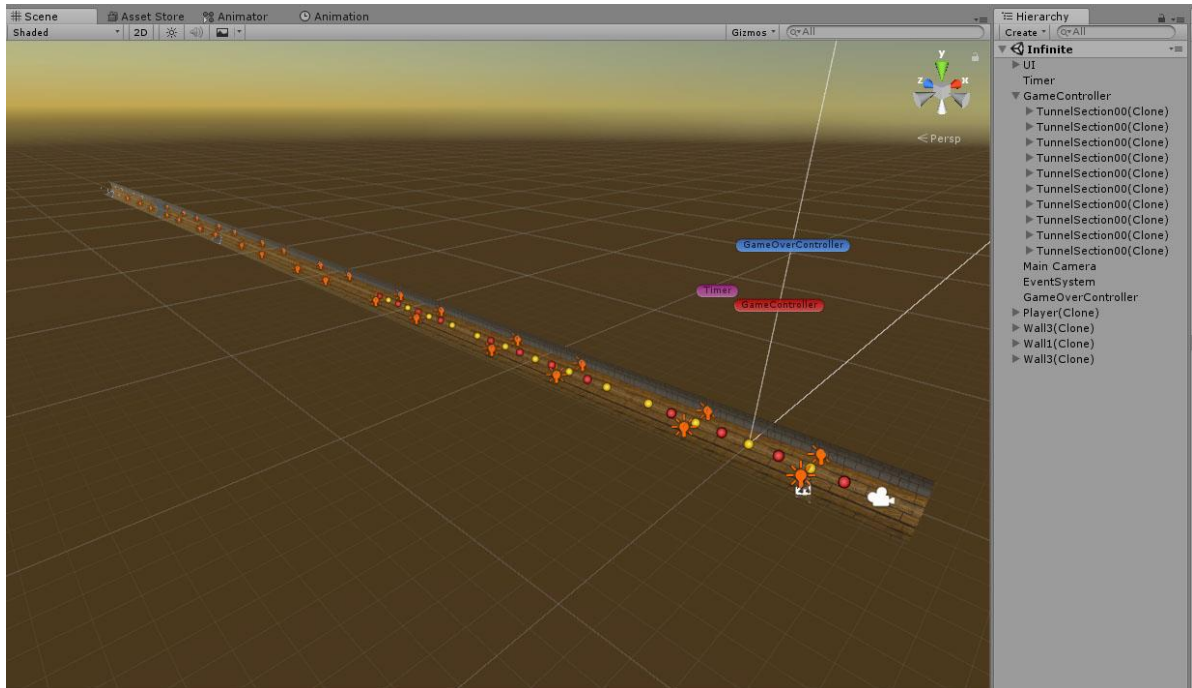
3.2.8 Alternatives

Les opcions alternatives en el tema de scripts controladors eren o centralitzar més tots els processos en menys scripts o descentralitzar-los separant més les funcions en més scripts.

La decisió tomada es un punt entremig, tenim un script central que es el GameController que porta gran part de la gestió del joc però transmet la informació als altres elements. Aquests elements són els altres controladors associats als altres components essencials del joc com el jugador, el cilindre i el temporitzador.

3.3 Escenari del joc

L'escenari del joc son una sèrie de cilindres buits on el jugador es mou cap endavant i la càmera el segueix. L'escenari sempre es compost per 10 cilindres, es a dir, només hi ha 10 cilindres (figura 4) en el joc i aquests es desplacen cap endavant quan el jugador els supera.

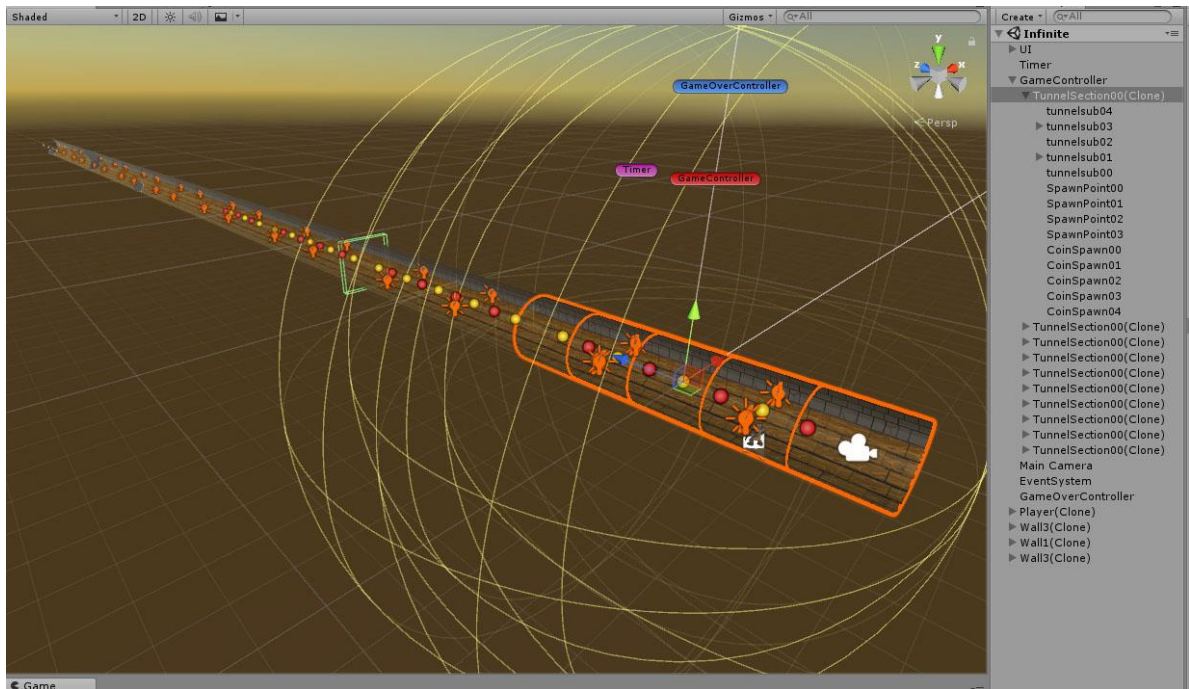


4: Els 10 cilindres

Per a reconèixer quan el jugador supera un cilindre s'ha implementat un *collider* (figura 5) que detecta quan entra el jugador i executa un codi. Aquest codi destrueix tots els objectes instanciats al cilindre, siguin monedes, obstacles o power-ups i el desplaça cap al final del nivell i procedeix a crear-ne nous elements dins el cilindre. Així reutilitzem assets i alleugerem la carga del joc. El codi també informa al *gamecontroller* que s'ha superat un cilindre per a controlar l'increment de la velocitat del jugador i del multiplicador del temps.

Pel que fa a l'estructura d'un cilindre, aquest es dividit en 5 sub-cilindres, conté 5 punts on poden aparèixer els obstacles i els *collectibles* (els punts vermells i grocs, respectivament, de la figura 4 i 5). També contenen 2 torxes a cada sub-secció parella.

L'script associat als cilindres (*TubeController*) també controla si un cilindre conté un tipus determinat d'obstacle per no fer el joc impossible.



5: Cilindre i el seu *collider*, la caixa verda

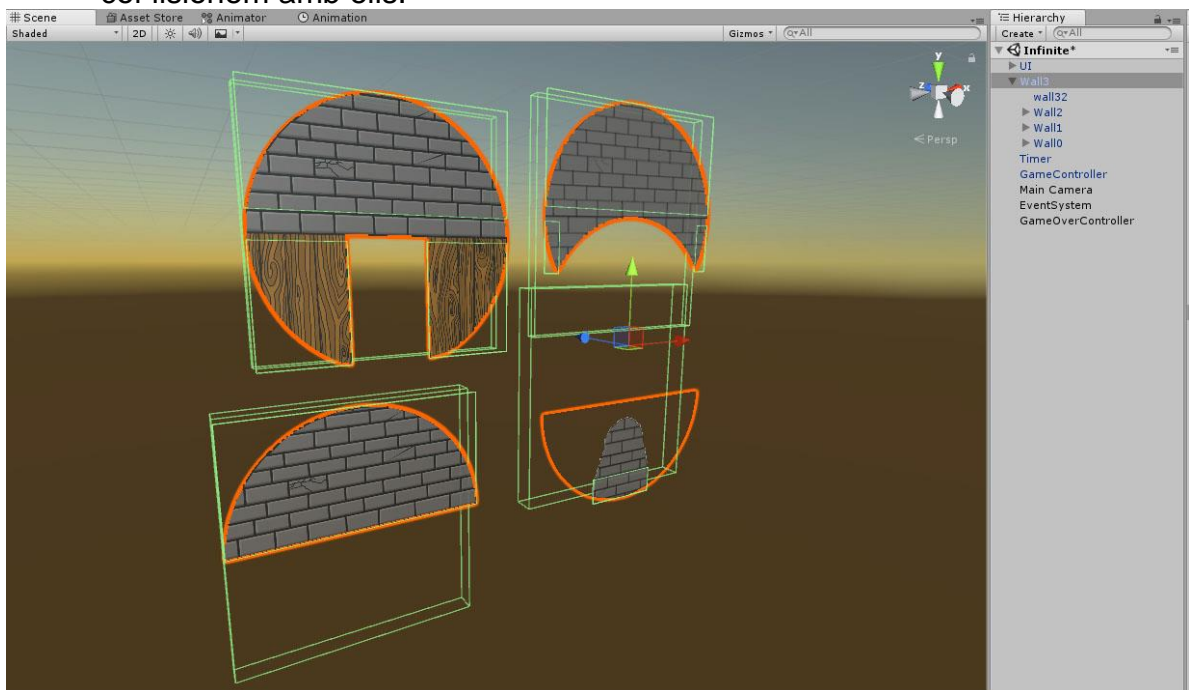
3.3.1 Alternatives

L'alternativa en aquest punt era fer nivells predefinits envers d'un nivell infinit que es anés construint proceduralment. Un nivell predefinit alleujaria la tasca de programació, però faria que la tasca de disseny fos important. A més d'això, un nivell predefinit acabaria sent molt avorrit per al jugador, però es interessant per a tenir en compte ja que si en futur es volgués fer nivells que fossin un repte per al jugador.

La decisió presa de fer-ho infinit era la idònia ja que amb un únic nivell es podria aconseguir el prototipus d'un joc complet senzill típic dels smartphones.

3.4 Els obstacles

Els obstacles son els elements que ens faran perdre vides i puntuació si col·lissem amb ells.



6: Els 4 tipus d'obstacles

Hi ha 4 tipus d'obstacles (figura 6) i cadascun té unes pautes d'aparició distintes per a cada nivell de dificultat.

Pel que fa al obstacle Wall0 (superior esquerra a la figura 6) es controla per l'script dels cilindres per evitar crear 2 cilindres consecutius amb aquest obstacle i fer impossible el joc. Els altres obstacles son Wall1 (superior dreta), Wall2 (inferior esquerra) i Wall3 (inferior dreta).

Els obstacles es componen per un pla que conté la textura i 2 tipus de *collider*, un que funciona com *trigger* o disparador i d'altres que serveixen per donar-li "massa" i que controlen la col·lisió amb el jugador. El *trigger* serveix per detectar quan un obstacle ha estat superat i, per tant, ja no ens es útil i podem fer-ho inactiu. El fer-lo inactiu evita tenir un obstacle bloquejant la càmera.

3.4.1 Alternatives

Les alternatives eren el nombre de tipus d'obstacles. Es podria haver fet un obstacle mòbil al estil del Wall3 però que es mogués 180 graus com el jugador, o un altre com Wall0 que també es mogués. Però no s'ha realitzat perquè això hauria augmentat considerablement la dificultat del joc i per tant amb quatre obstacles diferents el joc tindria certa dificultat però sense fer-ho impossible i frustrant per al jugador.

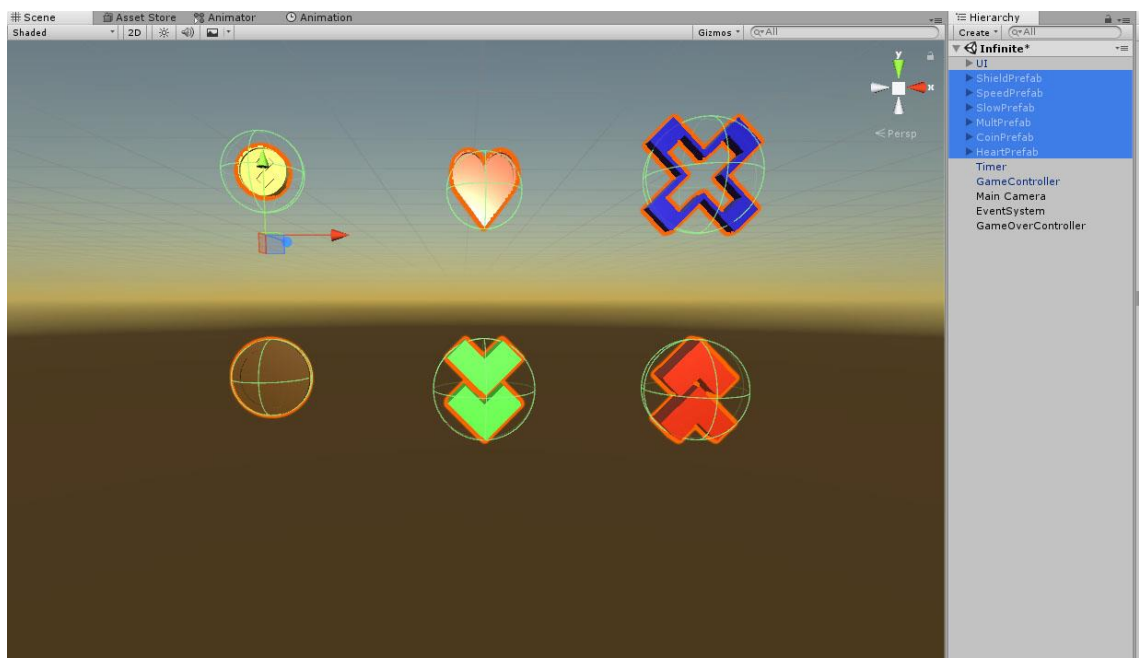
3.5 Els *collectibles*

Els *collectibles* consten de els objectes que el jugador pot recollir durant la partida. Cada *collectible* te un collider definit en forma de *trigger* que proporciona al jugador la informació necessària per a processar que ha recollit un objecte i aplicar el codi corresponent.

Aquests objectes (figura 7) son, per ordre de esquerra a dreta i de a dalt a baix:

- Moneda: augmenta la puntuació total depenent de la dificultat actual
- Cor: incrementa el nombre de vides fins a 3. Aquest nombre no es pot superar.
- Multiplicador: augmenta el valor que es multiplica al total de temps jugat. També depèn de la dificultat actual.
- Escut: protegeix al jugador d'una col·lisió durant 10 segons.
- Alentidor: fa que el jugador es desplaci més lentament durant 5 segons. Acumulable
- Accelerador: fa que el jugador avanci a més velocitat durant 5 segons. També acumulable.

Tots aquests items, a més de emetre un so quan son recollits, contenen un script que els fa rotar sobre l'eix Y per motius estètic. Tots el objectes són desactivats al ser recollits i el jugador comunica la funció adient al GameController.



7: Els *power-ups*

3.5.1 Les monedes

Les monedes son l'element més comú al joc. Aquests objectes al recollir-los ens augmenten la puntuació. La seva aparició està programada per a que s'instancien a posicions que faciliten el esquivar els obstacles. El `PlayerController` comunica al `GameController` que actualitzi la puntuació.

3.5.2 Els cors

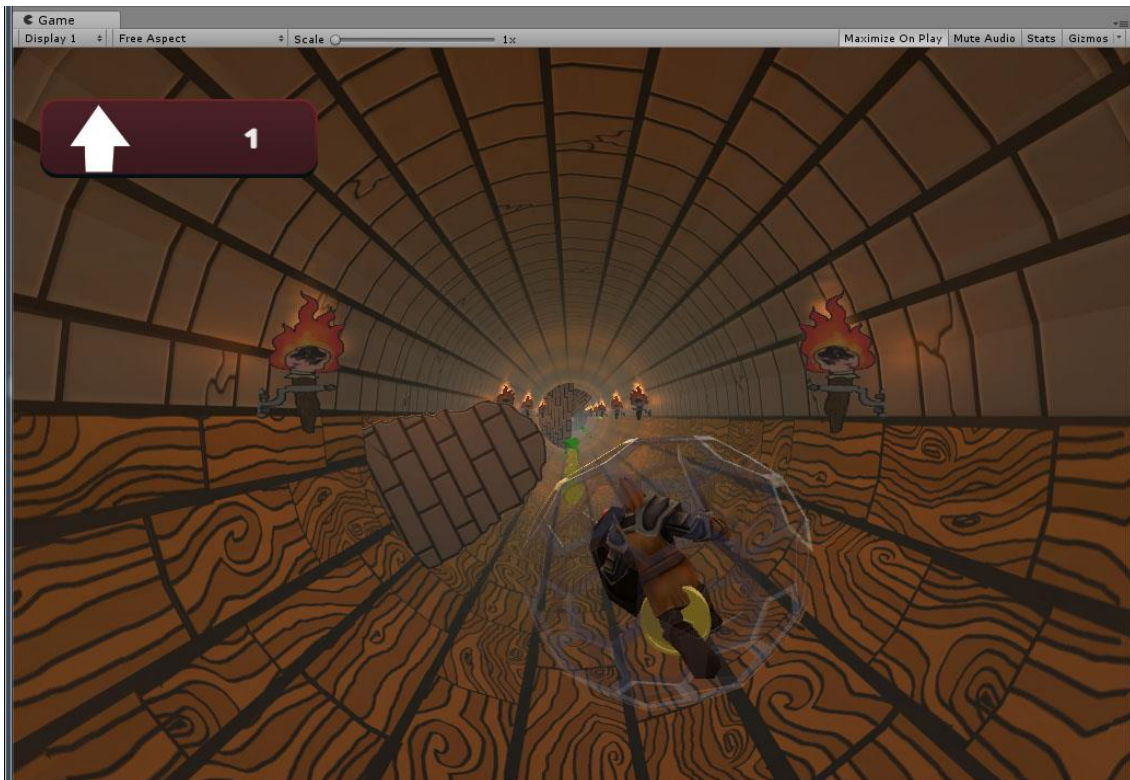
Un dels objectes més importants. Al recollir-los incrementen la quantitat de vides del jugador fins al màxim de 3. Al igual que les monedes quan es recull el controlador del jugador comunica al GameController que ha de mostrar una altra vida a la GUI.

3.5.3 El multiplicador

Aquest objecte augmenta el multiplicador que s'aplicarà al temps total de joc per a la puntuació total al final del joc. Es comunica al controlador del joc que augmenti el valor del multiplicador.

3.5.4 L'escut

Al recollir aquest objecte apareix un efecte d'escut (figura 8) a més d'un efecte de so i un panell a la GUI que ens mostra el temps actiu del *power-up*. Quan aquest es actiu durant 10 segons, al col·lisionar no es perd una vida ni es resta la puntuació, només es perd l'escut. Si s'agafa un altre escut tenint actiu un, es reinicia el comptador de temps a 10.



8: Gràfics de l'escut

3.5.5 Les fletxes de velocitat

Aquests 2 *power-ups* funcionen de la següent manera: quan son recollits modificant la velocitat amb la que sona la música del joc i modifiquen la velocitat amb la que es mou el jugador durant 5 segons.

La fletxa verda redueix la velocitat del jugador i redueix el pitch de la música i la fletxa vermella augmenta la velocitat i el pitch.

Després dels 5 segons la velocitat i la musica tornen a la normalitat.

Aquests dos objectes interactuen entre sí de la següent manera:

- Si s'agafa una fletxa tenint activa la contraria, els efectes s'anul·len entre sí.
- Si s'agafen dues fletxes iguals estant actives es va sumant 5 segons al temporitzador.
- Al col·lisionar amb un obstacle les dues fletxes es desactiven.

Per a donar èmfasis a que la fletxa vermella es un item maliciós s'ha implementat un altre so.

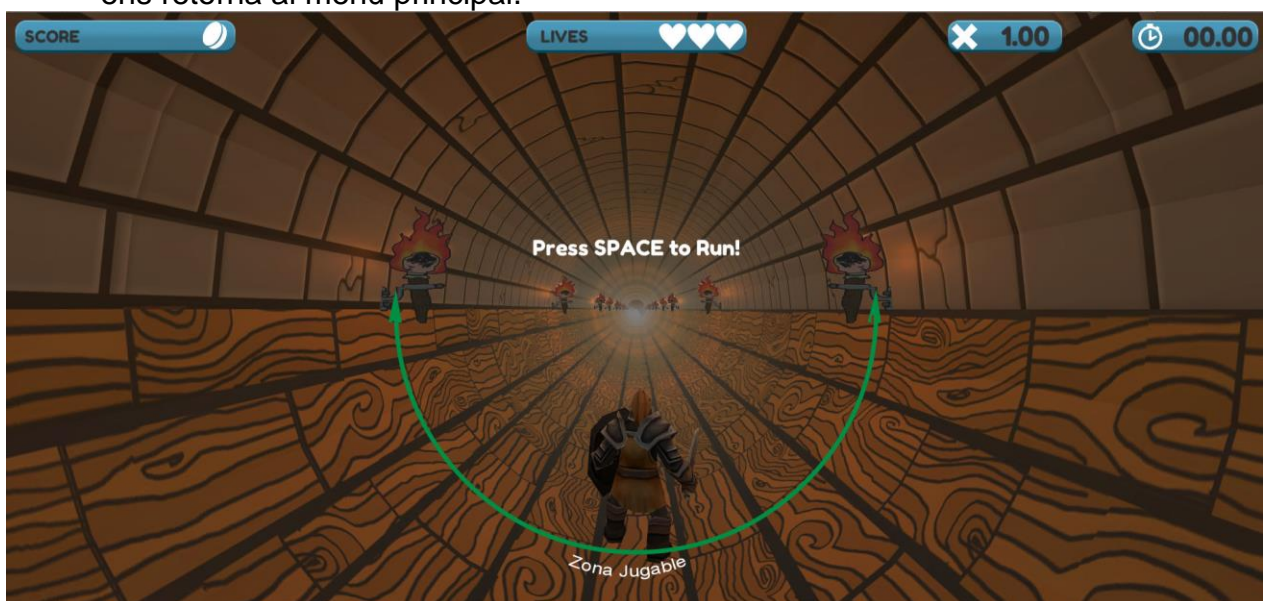
3.5.6 Alternatives

En aquest cas les alternatives eren saber quan parar d'afegir *collectibles* i que siguin originals. Es podria haver afegit distints tipus de monedes amb valors diferents o versions malicioses dels *power-ups*, però es va decidir seria millor mantenir-ho simple per a no augmentar de manera considerable la dificultat del joc

3.6 El jugador

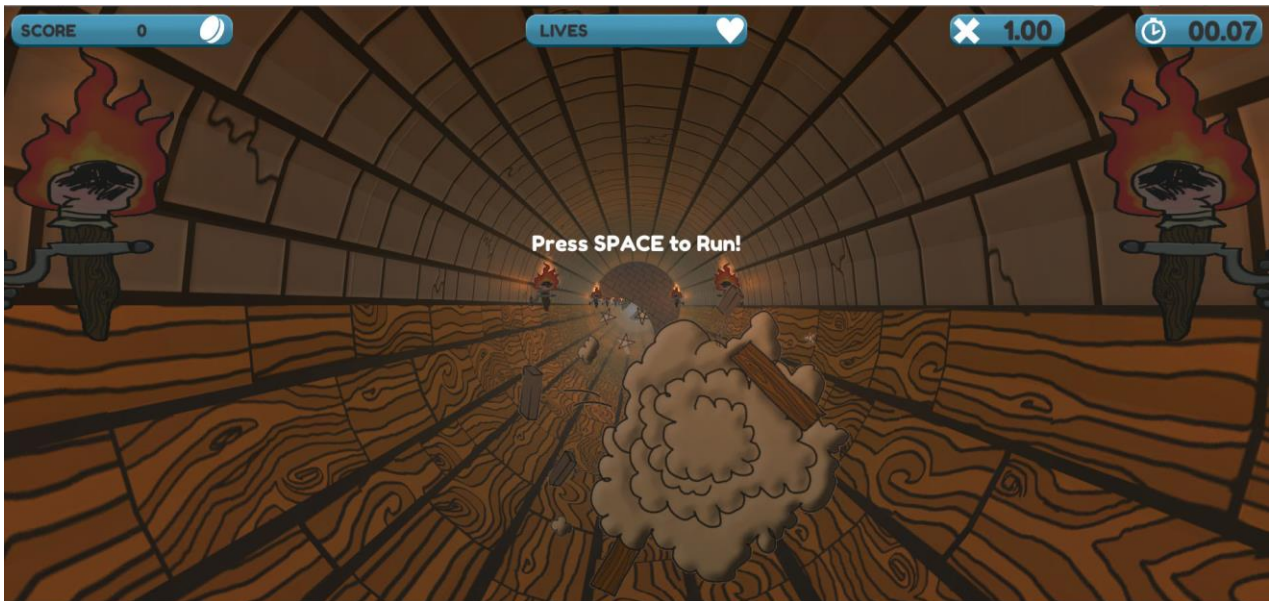
El guerrer medieval que representa al jugador. Aquest només es pot moure en els 180 graus inferiors del cilindre (figura 9). Si es prem la fletxa de direcció dreta o esquerra (o A o D per utilitzar el WASD) es mourà com a màxim 90 graus cap a la dreta o esquerra. Per iniciar el moviment i el joc s'ha de prémer la barra espaciadora. Després d'això només es necessari pulsar o dreta o esquerra per a moure el personatge i evitar els obstacles.

Una altre tecla important a tenir en consideració es la tecla Escape que ens retorna al menú principal.



9: El jugador i la zona jugable

Quan fem contacte amb un obstacle aquest envia un missatge al controlador del jugador i aquest es deté, es desplaça 10 unitats cap enrere (per facilitar el tornar a començar el moviment, activa el so de col·lisió, fa activa l'animació de col·lisió i torna a activar el missatge de tornar a pulsar la barra espaciadora per tornar a començar el moviment (figura 10). Si el jugador no té activat un escut, perdrà una vida i veurà disminuïda la seva puntuació. El decrement de la puntuació serà controlada per el GameController que a més controlarà la par de la GUI de les vides.



10 Animació de col·lisió

3.6.1 Alternatives

Inicialment el joc permetia anar per el sostre i la càmera rotava però el jugador es mantenia fixat en pantalla. Per motius de possibles marejos als possibles jugadors i gràcies al consell del consultor es va canviar a com es ara. La decisió inicial permetia més varietat d'obstacles a costa de ser menys agradable de jugar.

3.7 Configuració de la dificultat

Aquest capítol tractarà sobre la configuració de la dificultat del joc i com es tracta. S'entrarà en detall sobre la creació dels obstacles i la aparició dels *collectibles*.

Primer mencioner que la dificultat augmenta de dues maneres: quan s'arriba a un valor fixat de puntuació i quan es travessen 30 cilindres.

Per puntuació el primer increment es als 750 punts i per al segon son 5000 punts. Aquests increments de dificultat afecten a l'aparició de obstacles i items, el increment per cilindres travessats augmenta només la velocitat del jugador.

Per a tractar de tenir una valoració objectiva de quant difícil era el joc, a més de l'opinió del autor i el consultor, es va consultar a una 3ra persona. Es va mostrar versions del joc distintes per a veure si el joc era impossible o massa fàcil.

3.7.1 La dificultat i la seva associació amb la puntuació

L'increment de dificultat, a més, varia el valor de les monedes i multiplicadors i quanta puntuació es perd quan el jugador interactua amb un obstacle. Els canvis de valor son mostrats a la següent tabla

Tabla 1: Canvis de valor segons dificultat

Element	Dificultat fàcil	Dificultat mitjana	Dificultat difícil
Valor Moneda	10	25	50
Valor Multiplicador	0.10	0.20	0.30
Puntuació restada per obstacle	250	500	1000

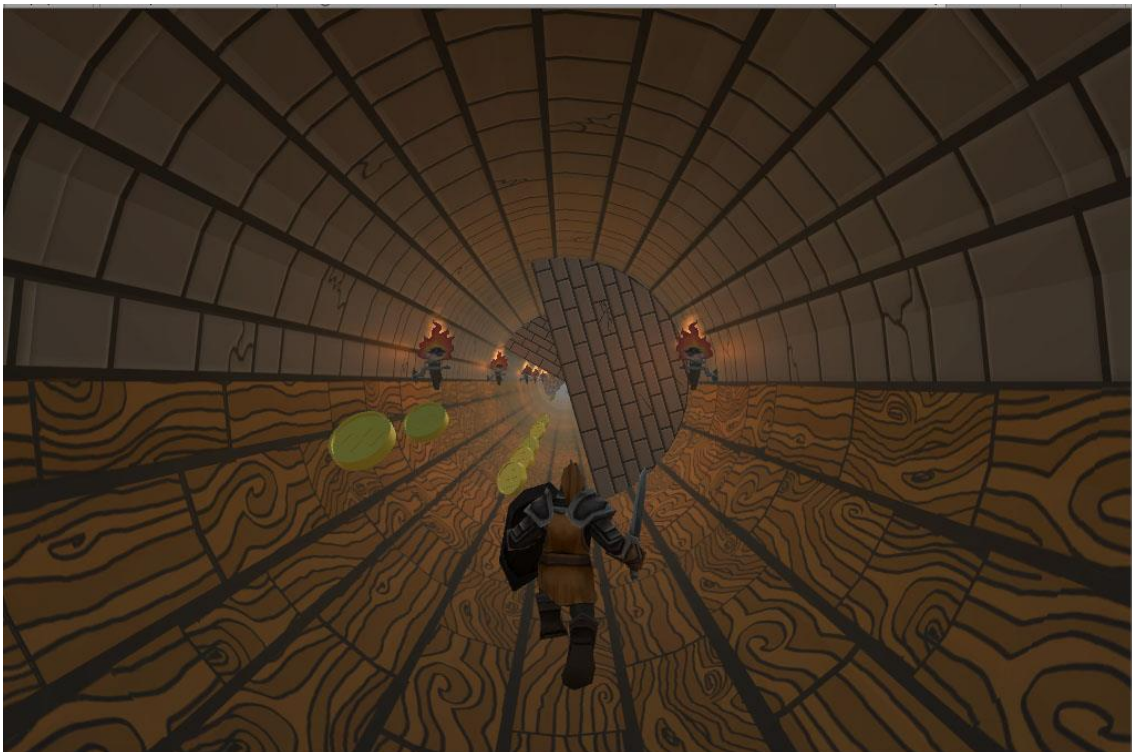
3.7.2 La dificultat i la taxa d'aparició dels obstacles

Al augmentar la dificultat, el GameController modifica tots els cilindres per a que instancien els obstacles i aquests els creen depenent de la dificultat.

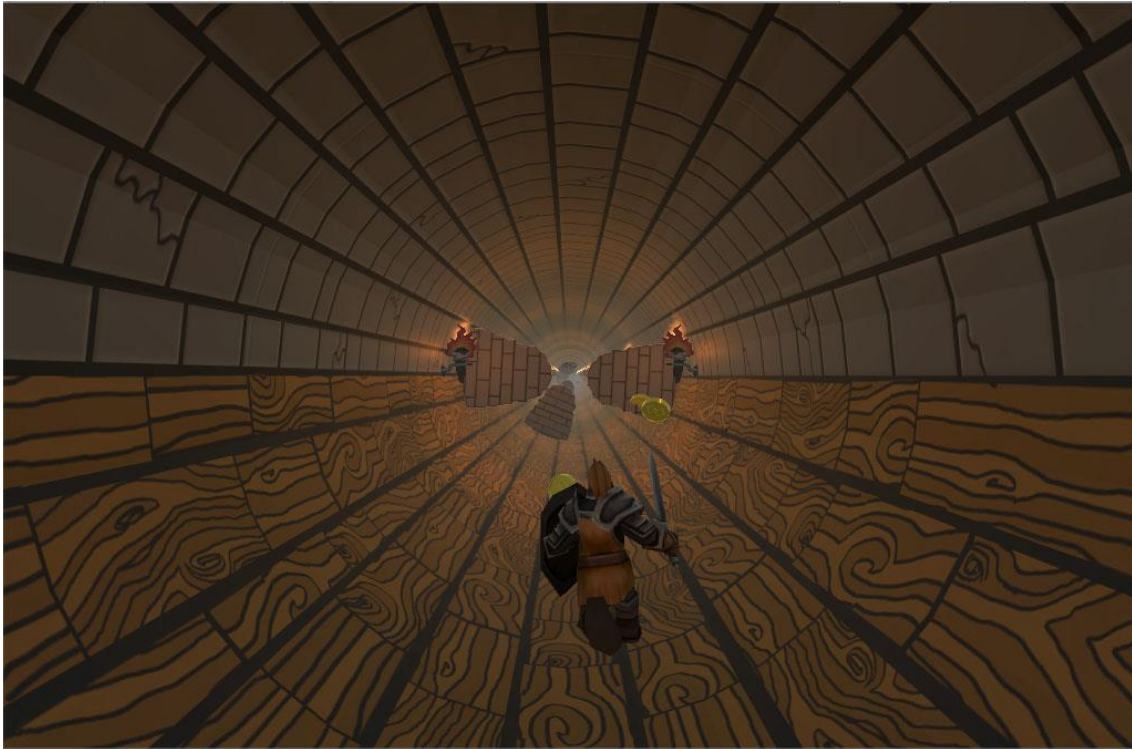
El procés de creació funciona de la següent manera: es crea un nombre aleatori entre 0 i 8 i es fa el mòdul de 4 (així tenim valors més aleatoris), per tant tenim un valor entre 0 i 3 que serà l'índex d'una llista d'obstacles. Amb l'obstacle agafat de la llista instanciem els obstacles del cilindre. A més mai es crearan dos cilindres seguits amb el obstacle que només té una porta per a poder passar (Wall0).

A la dificultat inicial, a més de la restricció dels 2 Wall0 consecutius:

- Només es crearà un Wall0 per cilindre.
- Només un Wall1 per cilindre.
- Wall2 té tres possibilitats aleatòries:
 - Es creen dos obstacles, el segon a 90 graus del primer .
 - Igual que l'anterior però el segon es a 45 graus (figura11).
 - Només crea un obstacle.
- El Wall3 te cinc opcions:
 - Només un obstacle.
 - Fer dos però el primer te un simètric a 180 graus.
 - Fer dos però el simètric el té el segon obstacle (figura12).
 - Fer tots dos amb simètrics.
 - Fer tres obstacles en la mateixa posició (un a 180 graus del primer i el tercer a 90 graus).



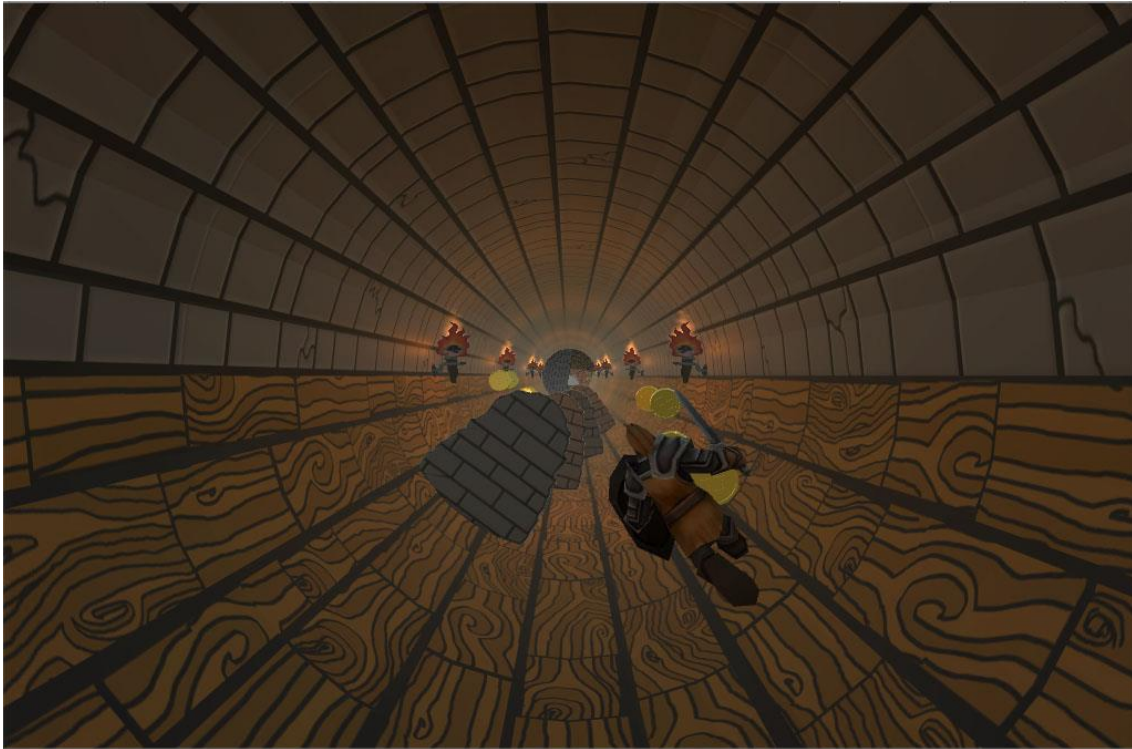
11: Wall2 i el segons obstacle a 45°



12: Wall 3 amb un simètric en el primer obstacle

A la dificultat mitjana:

- Wall0 es instanciat a dues posicions amb una diferencia de 60 graus.
- Wall1 es instanciat dues vegades en dues posicions distintes amb una diferencia de 45°.
- Wall 2 té tres opcions:
 - Crear tres obstacles amb una diferencia de 45° entre cada un d'ells.
 - Crear un segon obstacle amb una diferencia de 90°.
 - Crear només un obstacle.
- Wall3 té cinc opcions:
 - Crear quatre obstacles en diferents posicions però només amb una diferencia de 20° entre ells (figura 13).
 - Crear dos obstacles en dues posicions distintes amb simètrics amb una diferencia de 80° sent assequible per al jugador.
 - Dos obstacles amb simètric com l'anterior però amb angle completament aleatori.
 - Tres obstacles en la mateixa posició com a la dificultat fàcil però afegint un quart obstacle en una altra posició .
 - Només un obstacle.



13: Wall3 amb la diferencia de 20°

A la màxima dificultat:

- Per a Wall0 crea sempre tres obstacles a tres posicions distintes, el segon a 60° del primer i el tercer en el mateix angle que el segon.
- Per a Wall1 es creen quatre obstacles a quatre posicions distintes però cada següent obstacle es a 20° del anterior.
- Per a Wall2 hi ha tres opcions:
 - Es creen quatre obstacles exactament iguals al patró de Wall1.
 - Es creen tres obstacles a tres posicions distintes amb una diferencia de 45° però el tercer està una mica més allunyat dels dos primers.
 - Igual al segon però els tres obstacles son a posicions adjacents.
- Per a Wall3 hi ha quatre opcions:
 - Dos obstacles en dues posicions distintes amb simètrics i angle aleatori entre ells.
 - Dues posicions amb tres obstacles a cada una (el simètric i un a 90°).
 - Els 4 obstacles amb diferencia de 20° de la figura 13.
 - Quatre obstacles a quatre posicions amb angles completament aleatoris.

3.7.3 La dificultat i la aparició de *collectibles*

La aparició de monedes funciona de la següent manera: a les 5 posicions que hi ha a cada cilindre (punts grocs a la figura 4 o 5) es crearan els *collectibles*, a les 4 primeres posicions sempre seran monedes i a la última es on pot aparèixer un *power-up* o ser una altra

moneda. Les monedes sempre apareixen a certes posicions i angles de tal manera que generin una ruta que guiï al jugador.

La taxa d'aparició dels power-ups depèn de la dificultat actual. A la taula següent es mostren les probabilitats:

Tabla 2: Probabilitats d'aparició

Collectible	Dificultat fàcil	Dificultat mitjana	Dificultat difícil
Moneda	40%	45%	50%
Cor	20%	10%	5%
Multiplicador	15%	15%	15%
Escut	10%	10%	10%
Alentidor	10%	10%	10%
Accelerador	5%	10%	10%

3.7.4 Alternatives

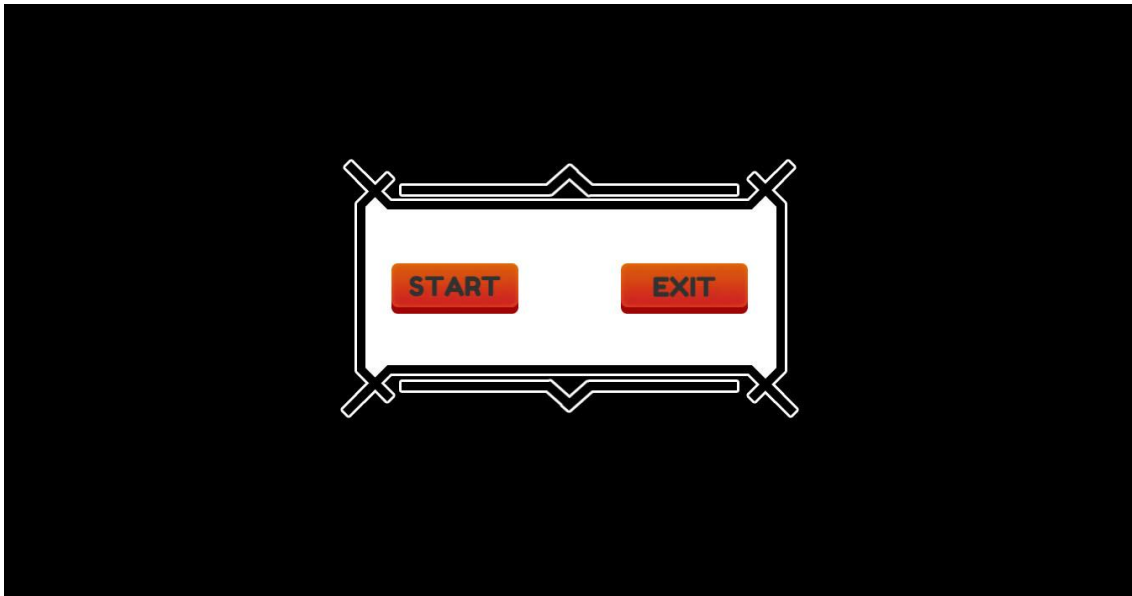
Les distintes alternatives eren diferents maneres de presentar els obstacles i les probabilitats de que fos un mètode o un altre al mostrar un tipus d'obstacle. S'ha decidit de la manera actual ja que presenta un repte però no es impossible gràcies a la aparició constant de *power-ups*.

3.8 GUI

Per a fer la GUI s'han utilitzat dos paquets de icones i interfícies de la Asset Store [6] i [7]

3.8.1 Menú principal

El menú principal es molt senzill: una pantalla amb art simple i 2 botons, un per començar el joc i altre per sortir de l'aplicació (figura 12).

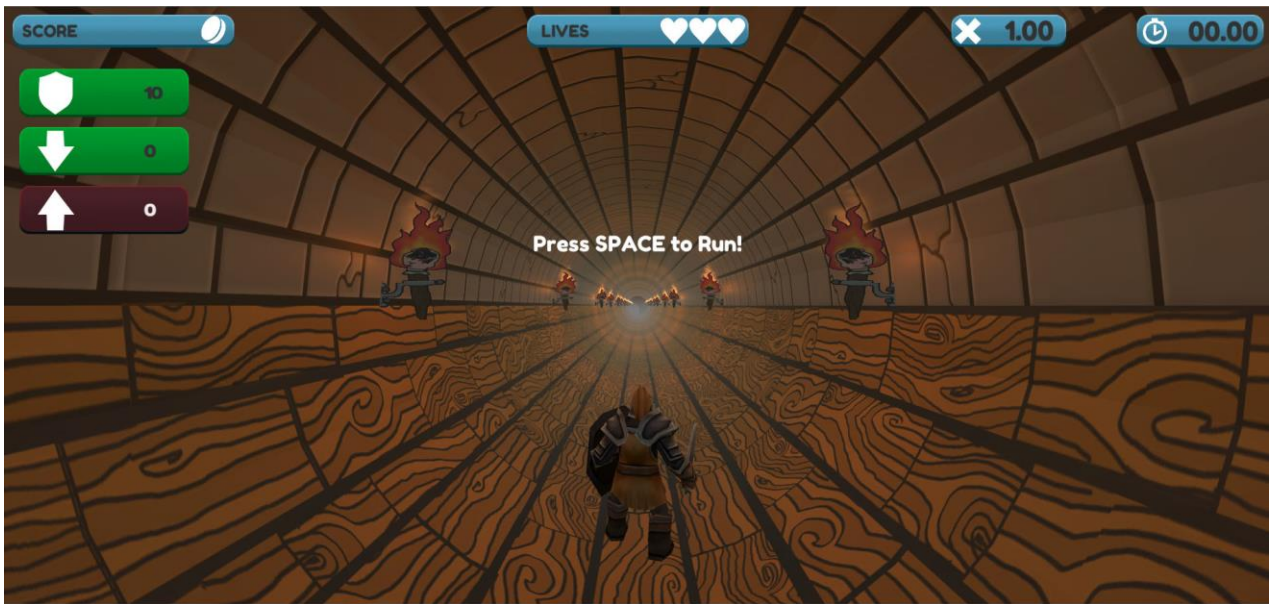


14: El menú principal

Per començar el joc al pulsar el botó START activa la funció LoadScene(1) al petit script ButtonController (no mencionat a la part de controladors per la senzillesa de l'script) i carga l'escena 1 que es el nivell infinit.

3.8.2 HUD

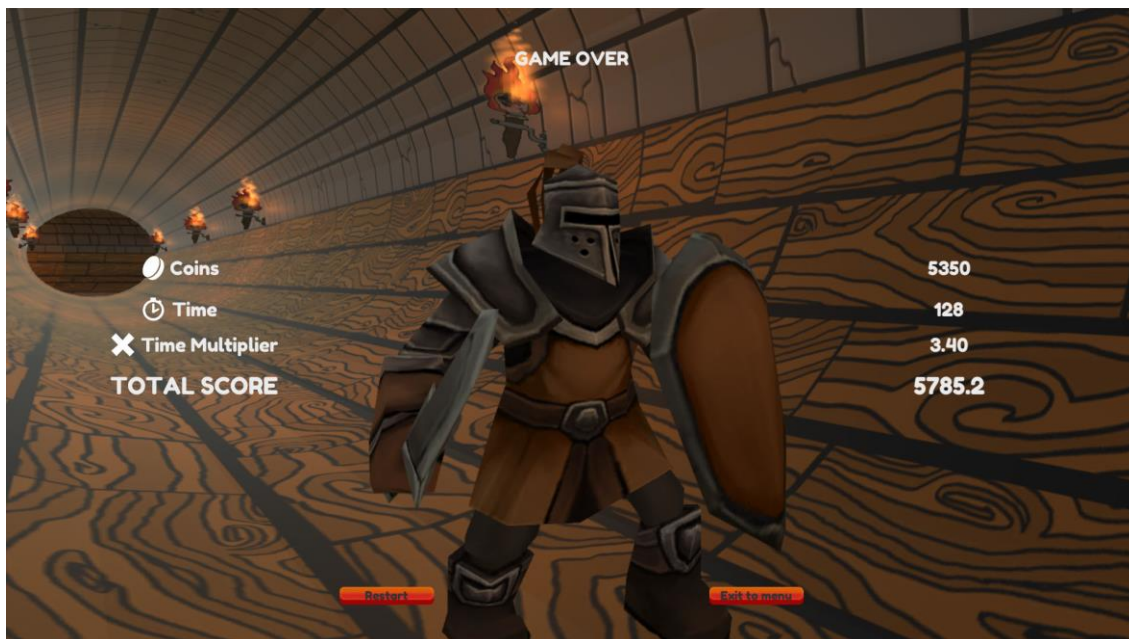
La informació important que necessita el jugador (puntuació, vides, multiplicador i temps de joc) sempre es mostra a la part superior de la pantalla. La informació corresponent als *power-ups* es mostra a la part esquerra de la pantalla amb una mida més gran i d'un color verd per als 2 objectes beneficiosos (escut i ralentització) i vermell per al maliciós (acceleració)



15: el Hud del joc

Com podem veure a la figura 12, que a efectes de l'explicació s'ha activat la part dels *power-ups*, a la part superior de la pantalla de esquerra a dreta sempre seran visibles el valors de puntuació, vides, el multiplicador i el temps. També es pot veure el missatge per començar el moviment. A la part esquerra d'amunt cap avall es troben els panells dels *power-ups* de escut, alentiment i acceleració i el seu temps actiu. Aquests panells només apareixen en el joc quan agafem el *power-up* relacionat. Quan aquest acaba el panell es torna inactiu i no es mostra. Mai es pot mostrar el panell d'alentiment i el d'acceleració a la vegada ja que els 2 objectes s'anul·len entre ells.

3.7.3 Pantalla de Game Over



16: Pantalla de Game Over

En aquesta pantalla es mostra un resum de la puntuació obtinguda a la partida. Mostra la puntuació aconseguida per monedes, temps realitzat, el multiplicador i la puntuació total. Aquesta puntuació total és la suma de les monedes i el temps per el multiplicador.

Es mostren avall dos botons un per tornar a jugar i l'altre que retorna al menú principal.

3.8.4 Alternatives

Com es volia fer un joc d'habilitat no es volia sobrecarregar la pantalla d'informació i per tant es va decidir col·locar els elements de la GUI a posicions no molestes i de fàcil accés. Una alternativa seria haver fet els panells dels *power-ups* sempre presents i a la part inferior de la pantalla però s'ha preferit la austeritat.

3.9 Aparença visual i sons

S'ha tractat de donar-li una aparencia una mica infantil per fer-ho més agradable a la vista. A aquesta estètica, a més, s'afegeix la intenció de no fer servir en excés els *assets* gratuïts i genèrics d'Unity. Per això s'ha realitzat textures en un editor de imatges des de zero. Les textures realitzades son per a les parets dels cilindres, els obstacles i les torxes. Aquestes textures han sigut realitzades per l'il·lustrador Adrián Sánchez-Cabezudo i per l'autor.

Els clips de so s'han triat de llibreries de sons gratuïts anomenada freesound.org [8] o [incompetech](http://incompetech.com) [9], tractant de que siguin precisos amb el que es suposa que representen.

3.9.1 Alternatives

En aquest cas les alternatives eren fer servir integralment *assets* de la Asset Store d'Unity i no haver fet elements 3D des de zero amb blender ni certes textures amb un programa d'edició de imatges. Fent-ho amb *assets* genèrics li hauria donat una aparencia més ajustada però perdria personalitat ja que hi ha molts petits jocs que els utilitzen. Creant els models des de zero, a més de proporcionar a l'autor experiència amb aquestes eines i el procés de creació de models 3D, dota al joc d'una aparencia menys genèrica.

4. Conclusions

En aquest treball s'ha aconseguit un joc senzill amb la gran majoria d'elements que formen un joc complet. Aquest permet jugar de principi a fi i al ser generat proceduralment es pot jugar múltiples vegades que sempre serà diferent.

Una de les lliçons apreses més important es com fer un nivell proceduralment. Aquests nivells necessiten un treball previ de planificació molt important ja que s'ha de crear un codi que creï el nivell indefinidament, que funcioni consistentment i que no faci que el nivell, a llarg termini, sobrecarregui la plataforma on es juga.

Una altra lliçó es com es vol controlar la dificultat, ja que un joc molt difícil frustrarà al jugador i massa fàcil l'avorrirà. A més d'aquesta consideració també es important tenir en compte que a un joc procedural, la programació de la dificultat necessita diferents tests fets per diferents persones per a tractar de fer-la realment equilibrada.

La última lliçó apresada es la de l'aparença gràfica. Si es vol un joc visualment agradable sense que pareixi completament genèric i sense personalitat es necessari utilitzar el menys possible els assets gratuïts de Unity. Hi ha assets a la venda, i molts d'aquests tenen una qualitat elevada, però a la llarga fan al joc menys únic. L'opció ideal seria fer-los des de zero, però això implica tenir coneixements de disseny 3D i una quantitat de temps considerable per a crear-los. En aquest projecte, com s'ha mencionat al capítol 3.9, es va tractar de fer certs elements des de zero, sobre tot textures, però per falta de temps i de coneixements de l'autor, aquestes textures son massa simples.

Pel que fa als objectius del treball exposats al capítol 1.2 s'han complert tots. La planificació i gestió del projecte s'ha complert segons el planning fet a la PAC1.

Per a la idea de videojoc també s'ha complert satisfactòriament, ja que s'ha realitzat un joc simple però entretingut. Hauria sigut molt més interessant haver dissenyat alguna idea més innovadora, però probablement el resultat hauria fos molt confús ja que al món actual es molt difícil tenir una idea nova, sobre tot en termes de jugabilitat.

En la part d'implementació s'ha complert l'objectiu. S'ha après a utilitzar tota la informació disponible a fòrums i a la documentació d'Unity. També a cercar elements per al joc com els clips de so gratuïts de freesound.org i incompetech.com. Alternativament i com afegit al desenvolupament, es va aprendre a utilitzar un control de versions per a guardar el projecte.

Per a acabar amb els objectius, s'ha après a utilitzar Unity i gran part de les seves eines. A més de les eines bàsiques com l'editor i l'asset Store, també s'ha utilitzat la part de control d'animacions Animator, l'editor de sprites i l'eina de "cuinar" la il·luminació per a no sobrecarregar la plataforma on s'executa el joc.

Tractant el tema de la planificació, en el treball s'ha seguit la planificació vista al capítol 1.4 sense gaires modificacions importants (com a molt tardar menys en un objectiu i una mica més en altre). Hauria sigut més adient fer més tests amb més gent externa per a haver trobat uns controls millors per al jugador i una

dificultat més ajustada (la 3ra persona té una opinió molt semblant a l'autor en quant dificultat i controls) ja que gran parts de les opinions diferents varen venir del consultor. També hagués sigut desitjable dedicar-li més temps a la part visual ja que les textures es varen fer de zero i no son d'un nivell normal per a un joc, o haver utilitzat més els assets gratuïts d'Unity per donar-li més qualitat gràfica.

Per concloure aquest capítol, es mencionarà una idea que no s'ha explorat en aquest treball i es haver realitzat el joc per nivells fixos. L'idea dels nivells fixos es adient si aquests nivells son extres al joc normal, es a dir, mentre el joc es un nivell infinit i procedural, els nivells extres son nivells que proporcionen un repte al jugador. Aquests nivells poden ser realment curts però al mateix temps realment difícils i que requereixin una habilitat superior.

5. Glossari

- Asset: Representació de qualsevol item que pot ser utilitzat al joc
- Collectible: Objecte que pot ser recollit per el personatge jugador
- Collider: Objecte que defineix la forma d'un altre objecte per a les col·lisions físiques
- GUI: Interfície Gràfica d'Usuari, el model visual amb el que el jugador rep la informació i en el cas del menú, interactua.
- HUD: Head-Up Display, es la informació que proporciona el joc a l'usuari. En el nostre cas la puntuació, vides, multiplicador i el temps de joc y *power-ups*.
- Power-Up: Objecte que millora al jugador o li dóna un nou poder.
- Procedural: En el cas del treball es el nivell fet per un procés i no fet a mà directament en l'editor d'Unity
- Sprite: Objecte gràfic 2D
- Trigger: Disparador, element que al ser activat (entrant a dins, estant a dins o inclús sortint) emet una senyal.

6. Bibliografía

- [1]: <https://unity3d.com/es/> (22/09/2017)
- [2]: [https://es.wikipedia.org/wiki/Sonic_the_Hedgehog_2_\(Mega_Drive\)](https://es.wikipedia.org/wiki/Sonic_the_Hedgehog_2_(Mega_Drive)) (22/09/2017)
- [3]: <https://halfbrick.com/our-games/jetpack-joyride/> (23/09/2017)
- [4]: Toon RTS Units Demo by Polygon Blacksmith
<https://www.assetstore.unity3d.com/en/#!/content/69687> (20/11/2017)
- [5]: <https://www.assetstore.unity3d.com/en/> (20/11/2017)
- [6]: SimpleUI by Unruly Games
<https://www.assetstore.unity3d.com/en/#!/content/103969> (27/12/2017)
- [7]: Simple Vector Icons by Unruly Games
<https://www.assetstore.unity3d.com/en/#!/content/101218> (27/12/2017)
- [8]: <https://freesound.org/> (1/12/2017)
- [9]: <https://incompetech.com/> (1/12/2017)
- [10]: <https://docs.unity3d.com/Manual/index.html> (22/09/2017)

7.Credits

7.1 So

Bushwick Tarantella Loop Kevin MacLeod (incompetech.com)
Licensed under Creative Commons: By Attribution 3.0 License
<http://creativecommons.org/licenses/by/3.0/>

PowerUp2.wav AbbasGamez (freesound.org)
Licensed under Creative Commons: By Attribution 3.0 License
<http://creativecommons.org/licenses/by/3.0/>

coin.wav ProjectsU012 (freesound.org)
Licensed under Creative Commons: By Attribution 3.0 License
<http://creativecommons.org/licenses/by/3.0/>

crash.aiff bone666138 (freesound.org)
Licensed under Creative Commons: By Attribution 3.0 License
<http://creativecommons.org/licenses/by/3.0/>

pickups-shield-belt runningmind (freesound.org)
Licensed under Creative Commons: By Attribution 3.0 License
<http://creativecommons.org/licenses/by/3.0/>

pickup-ammo-shock.wav runningmind (freesound.org)
Licensed under Creative Commons: By Attribution 3.0 License
<http://creativecommons.org/licenses/by/3.0/>

7.2 Imatges

- Els panells i botons de la GUI - SimpleUI by Unruly Games
- Les icones de la GUI - Simple Vector Icons by Unruly Games
- Les versions digitals sense donar color de les torxes, textures dels obstacles i cilindre – Adrián Sánchez-Cabezudo

7.3 Models 3D

El model del jugador i les seves animacions – Toon RTS Units Demo by Polygon Blacksmith

7.4 Tests

Juan Luis Ladaria
Joel Servitja Feu – Consultor de la UOC