Videojoc – Goblin Raider

Manuel Rodríguez Morales
Grau d'Enginyeria Informàtica

Professors: Joan Arnedo Moreno i Joel Servitja Feu

10/06/2018

FITXA DEL TREBALL FINAL

Títol del treball:	*Goblin Raider – Videojoc Arcade*
Nom de l'autor:	*Manuel Rodríguez Morales*
Nom del consultor:	*Joel Servitja Feu*
Data de lliurament (mm/aaaa):	*06/2018*
Àrea del Treball Final:	*Videojocs*
Titulació:	*Grau d'Enginyeria Informàtica*

Resum del Treball (màxim 250 paraules):

L'objectiu d'aquest projecte, que correspon al treball final del grau d'Enginyeria Informàtica, és el d'explicar el procés de creació d'un videojoc. Per tant, analitzarem cadascuna de les etapes que comprenen aquest procés, des del plantejament inicial del projecte fins a l'obtenció d'una versió d'un producte completament jugable. Estudiarem, doncs, les fases de disseny i definició d'un videojoc, així com la planificació i programació necessàries per portar-lo a terme.

La decisió de crear un videojoc en 2D respon a la necessitat d'ajustar-se a un termini d'entre tres i quatre mesos, pel que l'opció de realitzar un joc de tipus arcade/plataformes, amb el motor gràfic de Unity i enfocat al sistema operatiu Windows, semblava la més escaient.

El joc es basa en un petit ésser que ha d'escapar del perill que l'envolta en un món fantàstic de 8 bits, amb estètica Pixel Art. Aquestes amenaces procuraran fer-li mal i posar fi a la seva comesa, així que el protagonista haurà d'esquivar-les o vèncer-les. La seva fita és acabar el nivell sense ser derrotat i obtenir el màxim de punts possibles.

El resultat final respon a aquesta mecànica de joc i té molta jugabilitat, malgrat l'extrema dificultat que ha suposat el seu desenvolupament en el temps establert. Tanmateix, l'ús d'una nova tecnologia i una curta fase de preparació no han sigut impediment a l'hora de programar detalladament cadascun dels components que conformen aquest videojoc que, en definitiva, ha sigut una oportunitat excel·lent per practicar aspectes característics del meu itinerari de grau, el desenvolupament.

Abstract (in English, 250 words or less):

The objective of this Computer Engineering final project is to create a videogame from the very beginning through which we will explain the process of game creation. We will analyse each stage of this development process individually, starting with an initial project approach and concluding with the final version of a fully playable product. Consequently, we will examine the corresponding design and definition phases, as well as the scheduling and development required when making a videogame.

The decision of making a videogame in 2D was taken because of the limited production time —between 3 and 4 months approximately— I had, which also determined me to produce an arcade/platform game with Unity that focuses on the Windows platform.

This game consists of an 8-Bit fantasy world of Pixel art and a small character that runs from danger. He encounters different threats that try to harm and prevent him from carrying out his mission, so he needs to dodge or defeat them in order to achieve his objective: to reach the end of the level alive by getting as many points as possible.

All in all, the final game blends these game mechanics in with an excellent playability. Actually, despite the fact that the usage of a new technology and the mentioned time restrictions turned the development process into a demanding and difficult task, we can conclude that this project has been an excellent opportunity to put into practice most aspects covered in my degree specialisation, i.e., video game development.

Paraules clau (entre 4 i 8):

Videojoc, 2D, Arcade, Plataformes, Unity

Índex

1. Introducció

1.1 Context i justificació del Treball

Els videojocs cada vegada van obtenint més importància, ja que son i seran un entreteniment tal i com ho es el cinema, els llibres o els esports. En aquest sentit, poden arribar fins on s'indiquin els límits a l'hora de desenvolupar-los.

El procés de creació d'un videojoc es el mateix procés que per a qualsevol programari informàtic o altres productes fora d'aquest context, on s'ha de realitzar una planificació, un desenvolupament, uns jocs de proves i una publicació en la plataforma escollida. En aquests casos on nosaltres som directors del nostre propi joc, hem de pensar en tots els aspectes del mateix, incloent la creativitat d'una idea, plasmar-la i afegir contingut audiovisual tenint en compte la usabilitat.

Degut a tot això i la passió que porto pels videojocs, em vaig decantar per fer un treball d'aquestes característiques, aprofitant tot el que he aprés en tots aquests anys dintre de la carrera i aplicant-ho de la manera que més m'agrada.

1.2 Objectius del Treball

- Realitzar una planificació de totes les tasques que comporten realitzar un videojoc en un curt període de temps, des de l'inici fins final.
- Realitzar el primer videojoc oficial a nivell personal.
- Dissenyar un joc desafiant i senzill en mecànica a mesura que anem avançant.
- Realitzar un videojoc amb bona rejugabilitat gràcies al desenvolupament i la creació automàtica i aleatòria d'escenaris.
- Obtenir coneixements d'una nova tecnologia en alça per a la creació de videojocs com es el cas de Unity, aprofitant que es compatible amb el llenguatge C#, el qual es semblant al llenguatge més utilitzat de la carrera (Java).
- Obtenir una base per aplicar contingut Audiovisual en un videojoc.

1.3 Enfocament i mètode seguit

Des del principi s'ha volgut agafar diferents components d'alguns jocs com es la mecànica i la jugabilitat dels Super Mario Bros i l'acció dels Metal Slug, jocs que van marcar un abans i un després en l'historia dels videojocs, degut a les característiques presentades en el seu moment. Volia un joc progressivament difícil, com els típics jocs Arcade on havies de ficar una moneda cada vegada que perdies, però sense aquesta metodologia.

D'aquests pensaments ha sorgit Goblin Raider, un joc Arcade de plataformes 2D, amb controls senzills i una progressió d'escenaris aleatoris.

1.4 Planificació del Treball

El primer que es va realitzar va ser el document de disseny del videojoc, on es troben totes les característiques, incloent jugabilitat, interacció del jugador, tipus de gràfics i progressió.

Seqüencialment, es van decidir els components que hi haurien al nostre joc, incloent personatges, trampes, enemics i escenaris.

Altres elements, com la totalitat d'enemics, paràmetres, interfície i sons es van deixar per més endavant.

Per un bon començament i degut al temps establert del projecte, es va decidir utilitzar recursos online sense copyright per a la majoria de gràfics utilitzats (enemics, escenaris, protagonista, etc). Seguidament de tenir els sprites i les diferents animacions creades, vaig procedir a la creació del moviment del personatge principal i el de un enemic en un escenari simple predefinit.

Un cop creada una base amb una mínima interacció entre les diferents entitats, es va procedir a la generació del mapa, incloent la generació del mapa automàtica i el fons dimensional que li aporta identitat i profunditat ambiental. A

partir d'aquí es comencen a polir detalls, aplicar sons, interfícies i millorar la quantitat d'enemics.

Finalment ens dediquem a fer més nivells i més dificultosos augmentant paràmetres.

1.5 Breu sumari de productes obtinguts

- Executable per a PC de la versió final del producte.
- Un APK instal·lable en Android i preparat per a ser publicat.
- Tres vídeos relatius a les 3 PACS entregades on s'expliquen els progressos.
- Un vídeo resum explicant tot el procés de creació del videojoc.
- Un diagrama de Gantt amb la planificació del projecte.
- Un document explicant els possibles estats i transicions de pantalla del videojoc.
- Memòria final del projecte.

1.6 Breu descripció dels altres capítols de la memòria

La resta de capítols mostraran els diferents aspectes del videojoc i com s'han anat creant cadascun d'ells. Des de l'escolliment i creació dels elements audiovisuals fins la lògica de la programació.

2. Resta de capítols

2.1 Elements principals

Els elements principals d'un joc 2D els veurem a primera vista gràficament. Es llistaran una sèrie de **sprites** utilitzats, els quals s'han tret majoritàriament de recursos online gratuïts i s'han retocat segons la finalitat. Cadascun dels personatges que surten al joc, tenen el seu full de múltiples sprites, per a poder fer les animacions. Des de Unity tenim la oportunitat de importar-los i separar-los per trossos automàticament, segons els paràmetres que indiquem.
Els sprites al final només seran els components gràfics dels contenidors o **prefabs,** que contenen a tot un objecte o **gameObject** en Unity. Aquest tipus d'objectes seran **instanciables** en qualsevol moment.

A continuació faig el detall dels objectes més importants del joc:

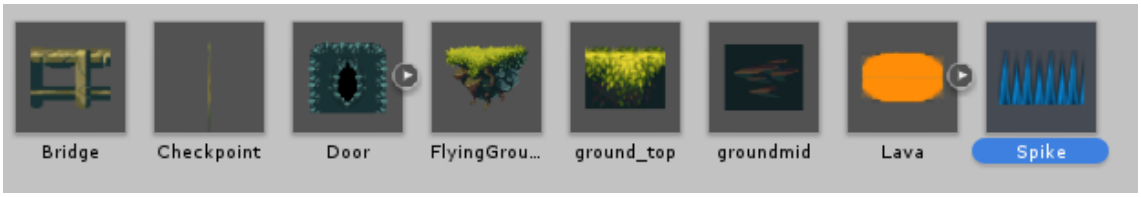
- **Goblin**:

  Representa al jugador. Pot moure's horitzontalment i saltar de manera vertical. Pot disparar en qualsevol moment una o varies boles de foc.

- **Escenari**:

  El mapa es representat seguint uns patrons d'aleatorietat. Tindrà un comportament de terra estàtica per on el Goblin podrà anar passant segons avanç pel nivell, fent de col·lisionador. Els enemics ho tindran com a obstacle per a la recerca del jugador principal i a part, els hi farà de límit.

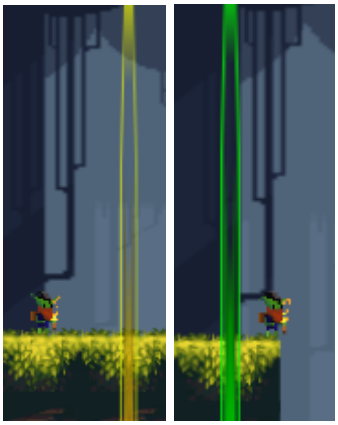
  Aquests son tots els tipus de prefabs utilitzats per a la creació del mapa:



Com podem veure, les trampes de pinxos i lava faran mal al jugador. Els pinxos son estàtics i la lava té un moviment vertical constant. El pont caurà quan el jugador el trepitgi. Finalment, la porta serà al final del nivell per passar al següent.

- **Checkpoints**:

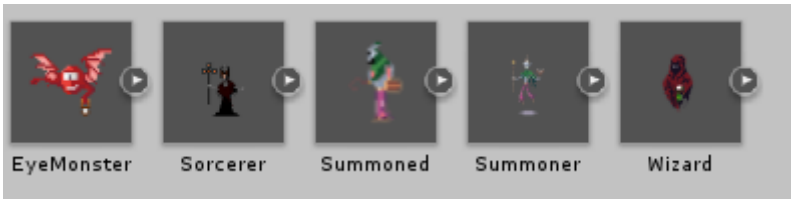
  Els checkpoints son els punts de guardat al mateix nivell. Cada vegada que es passa per damunt d'un d'ells, es guarda la posició del jugador i dels enemics, juntament amb la quantitat de vida que tenen. Si per casualitat morim, podrem tornar a la última posició guardada amb la vida d'aquell moment.



- **Enemics**:

  Cadascun dels enemics del joc tenen la seva forma d'atacar i un comportament independent dels altres.

  Els podem veure a la imatge:



La definició de cadascun d'ells seria la següent:

1. EyeMonster: Un monstre volador que dispara boles d'energia amb un temps lent. Vola, s'apropa, dispara i persegueix.

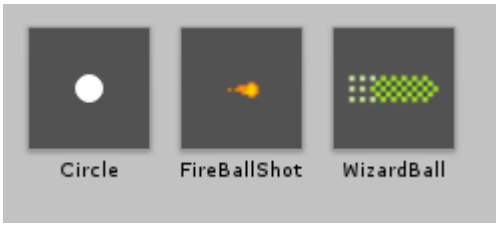
2. Sorcerer: Ataca des de aprop, fuig desapareixent i torna a atacar des de un dels costats.

3. Summoner: Invoca a enemics més petits que ell (Summoned). Segueix de forma horitzontal e invoca als petits que intentaran apropar-se al personatge fent salts cap a la seva direcció.

4. Wizard: Enemic final que s'anirà trobant contínuament al acabar els nivells. Es un enemic que farà atacs tant a distància com de proximitat.

- **Bales de tirs o atacs:**

  Seran instanciades pel jugador (bola de foc) com per els enemics que disparen. Tindran una força i velocitat constants. S'eliminen de l'escenari un cop no es veuen a la pantalla, per tal de minimitzar càrrega de treball. S'utilitzen els següents prefabs:



2.2 Elements secundaris

Per un altre part, tenim els objectes que faran de l'experiència més immersiva, els quals interactuen amb l'usuari i fan entendre les diferents situacions del joc:

- **Barres de vida d'enemics i de personatge:**

  S'ha creat un objecte per a representar la vida dels enemics, que estarà situada a sobre d'ells. Amb aquesta barra estem dient que els podem destruir si els hi baixem tota la barra i a part ens servirà per veure en tot moment quant els hi falta. S'ajusta a les dimensions dels diferents enemics i cadascun d'ells tindrà una quantitat determinada.

  Per contrabanda, el jugador també tindrà la seva pròpia barra de vida, situada a la part posterior esquerra de la pantalla. El jugador tindrà també un màxim de vida.

- **Vides del jugador**

  Les vides del jugador es mostren just a sota de la barra de vida, a la part posterior esquerra de la pantalla. Com a màxim tindrem 5 vides, les quals s'aniran perdent quan la barra de vida arribi a zero o be, quan el personatge caigui fora de l'escenari. Només es guanyarà una vida al completar el nivell. Pels nivells següents, tindrem les vides que ens quedin al completar l'anterior.

- **Textos flotants:**

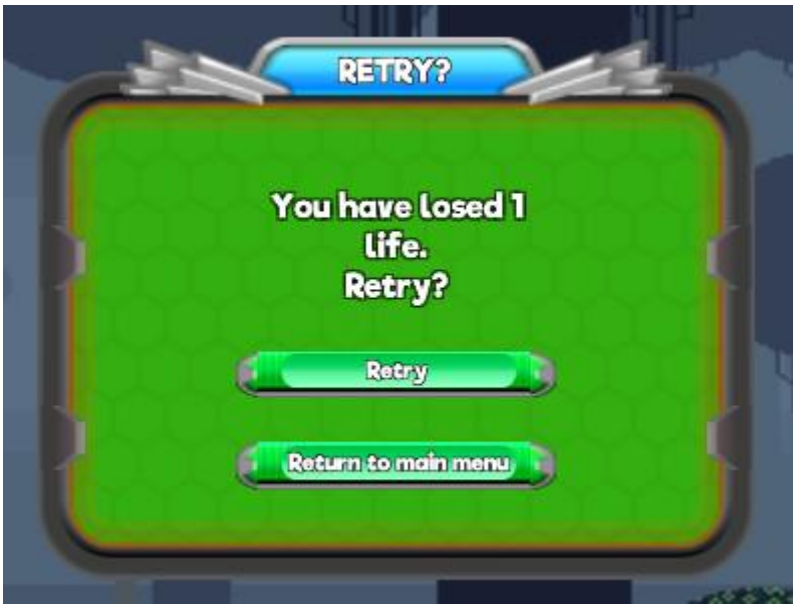
  S'han afegit textos flotants per a indicar la quantitat de mal que fem, rebem o ens curem. Si es una cura, el text serà verd, d'altra banda, serà roig.
  Es realitza una animació per a que el text faci una transició a transparent i aparegui a diferents llocs per a que siguin llegibles, sempre al costat del lloc d'impacte.

- **Boosts o millores:**

  Es creen els objectes de Boost amb tres tipus de millores. Per una banda estan els dispars amb múltiples bales, i la invencibilitat. Aquestes millores seran exclusivament temporals. Per l'altra banda, hi haurà cures que seran permanents. La quantitat de bales, de vida i de temps en segons d'invulnerabilitat seran aleatoris dintre d'un rang de valors.
  El prefab té una animació incorporada per donar un efecte de potenciador i es el següent:

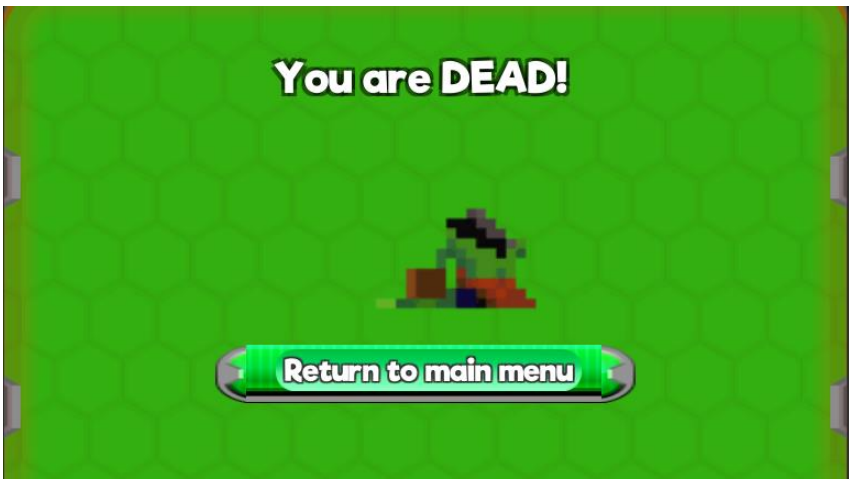
- **Pantalla de retry:**

  Amb la pantalla de retry, al perdre una vida, podrem tornar-ho a intentar accedint a l'últim checkpoint.



- **Pantalla de game over:**

  La pantalla de game over es donarà quan ens hagim quedat sense vides:



- **Pantalla de final de nivell:**

  La pantalla de final de nivell ens servirà per veure els punts obtinguts dels monstres derrotats i els cristalls en possessió:

- **Pantalla de tenda de millores:**

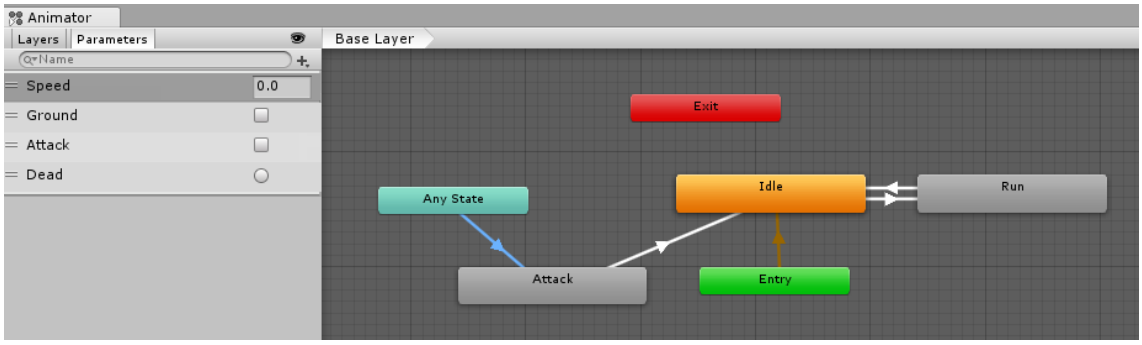
  Es crea una pantalla per la tenda de millores, on hi hauran 6 millores desbloquejables. Aquesta pantalla apareixerà al final del nivell, si li donem al botó de tenda.

  Les millores s'aniran desbloquejant seqüencialment. Podrem canviar una bona quantitat de punts per cristalls per a poder aconseguir les millores:

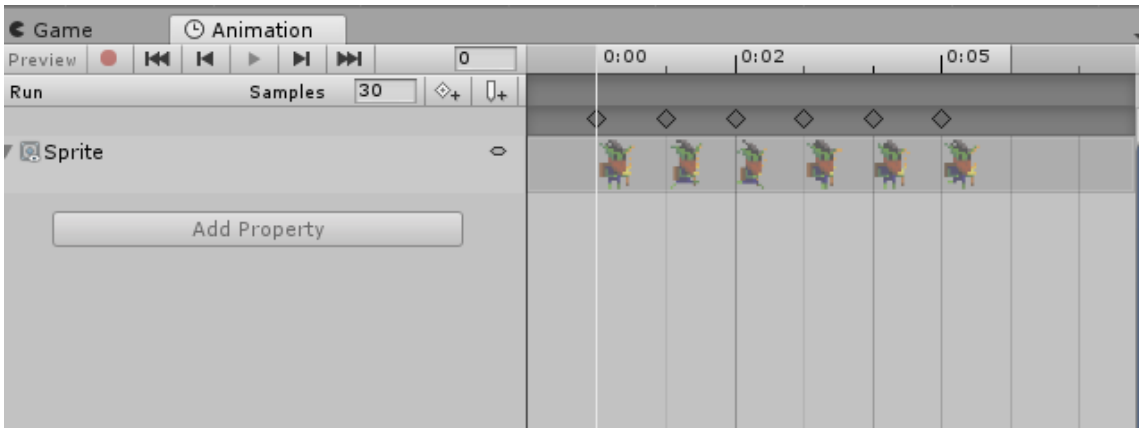
- **Animacions:**

S'han creat animacions pel moviment dels enemics, jugador i diferents atacs. Cada acció serà una animació vinculada a un controlador d'animació, que aquest a la vegada serà un component de l'objecte el qual volem animar. Té la finalitat d'establir un flux des de inici a fi, segons uns paràmetres donats i la definició del seu comportament.

En la següent imatge podem veure una captura de la vista del animator, dintre de Unity, amb l'exemple del Goblin:

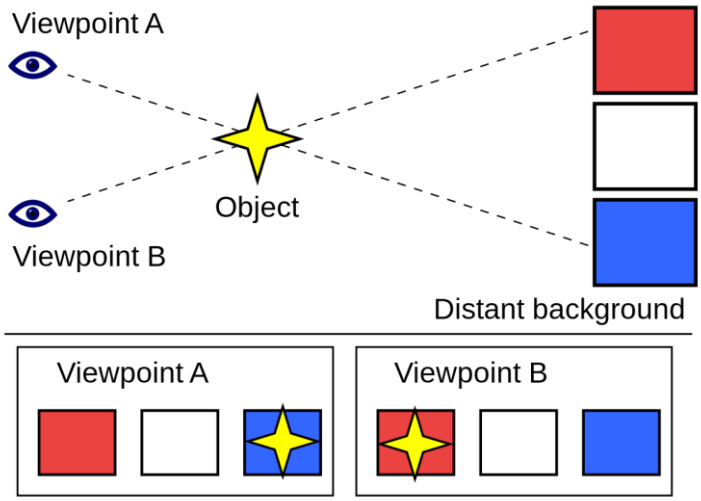


Per l'altra banda, fiquem un exemple d'animació, com pot ser la de córrer. Podem veure tots els detalls, incloent una vista prèvia de l'animació:



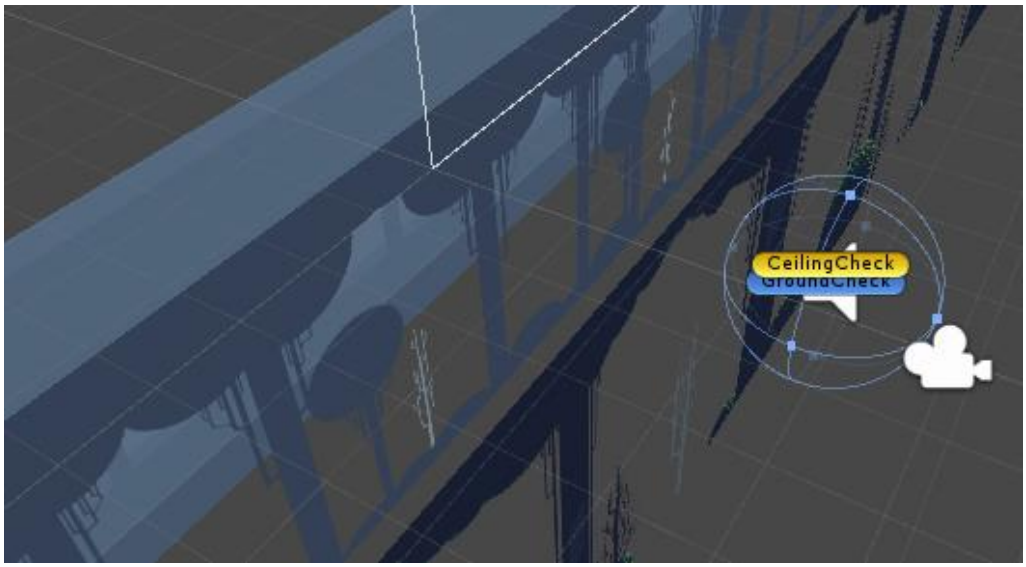
- **Efecte Parallax (Animació de fons):**

Aquest efecte anomenat Parallax o Parallaxing, es un efecte que consta de 2 o més plànols en paral·lel que semblen que tinguin profunditat quan hi ha moviment, degut a la distància entre ells i el punt de vista. Un exemple il·lustrat de la tècnica es aquest:

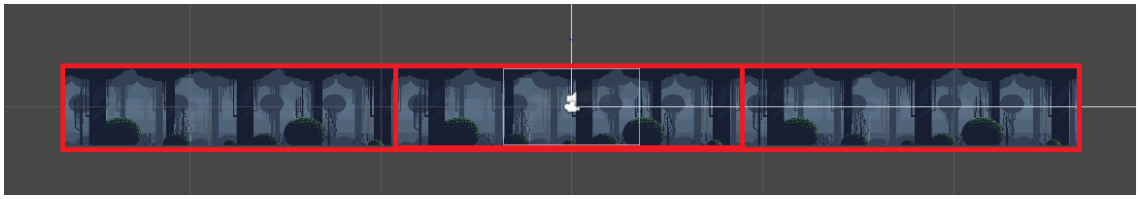


Ilustració 1 - Parallax – Wikipedia

Per al nostre cas, tenim un fons amb 3 capes i diferents distàncies entre elles que equivalen al distant background, un objecte càmera que serà la visió del joc (viewpoint), i el personatge (object). Si fiquem l'editor del Unity en mode 3D, veurem les diferents profunditats de l'eix Z:



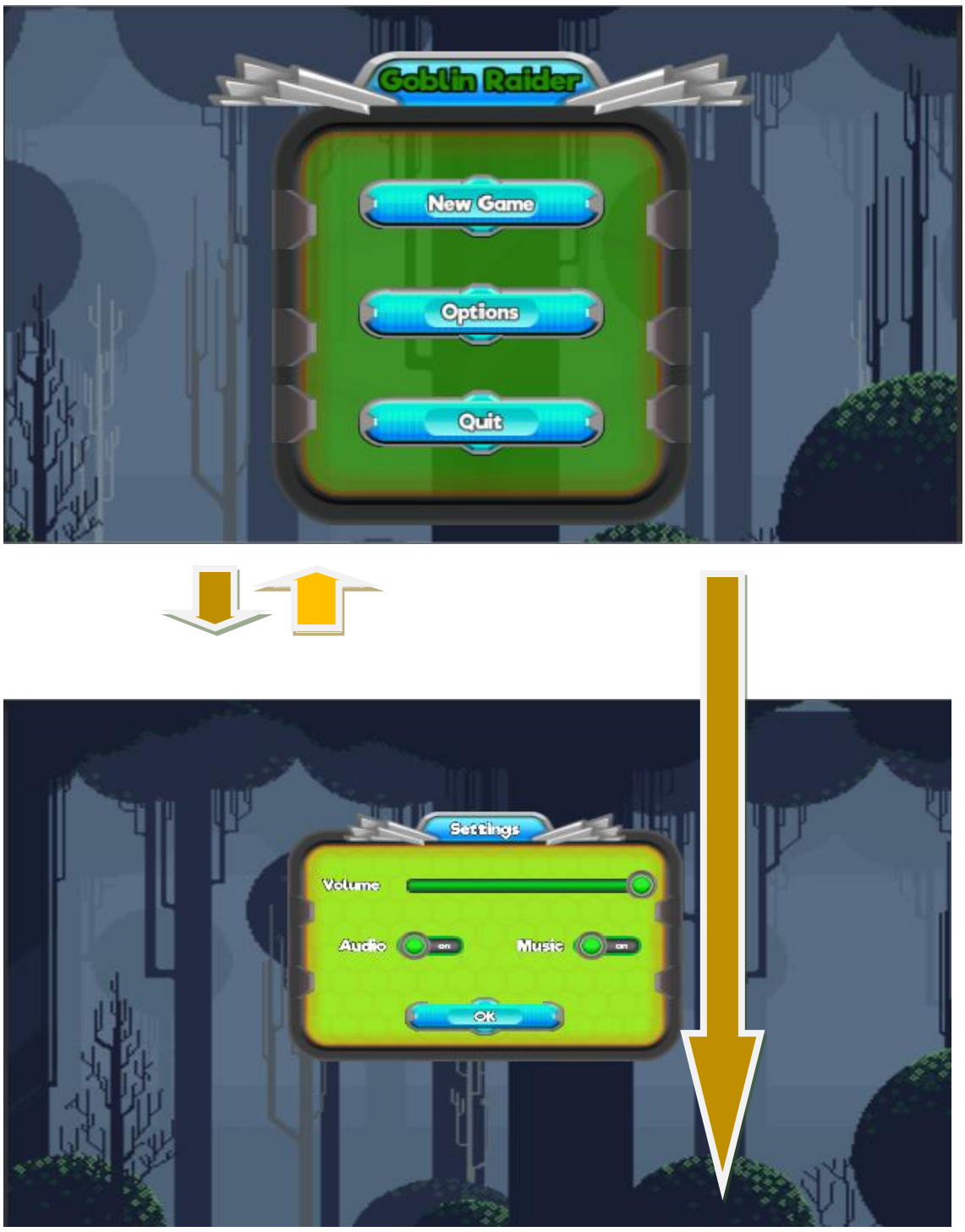
No es només l'efecte el que s'ha implementat, sinó que també s'ha hagut d'implementar un Scroller o "desplaçador d'imatges" a cadascun dels plànols per tal de que el fons mai acabi. El sistema es basa en replicar el fons tant a l'esquerra com a la dreta de l'existent, de manera que necessitarem tres plànols junts:



Tenint això i controlant els límits de la posició dels plànols, podem fer aparèixer una imatge que s'utilitzarà i, a l'hora, fer desaparèixer aquella imatge que no hi estem utilitzant, quan hi hagi el moviment de càmera. Aquest moviment sempre serà en qualsevol de les dues direccions horitzontals.

- **Pantalles de transició:**

  A continuació es mostren les pantalles i les transicions bàsiques que hi ha entre elles:



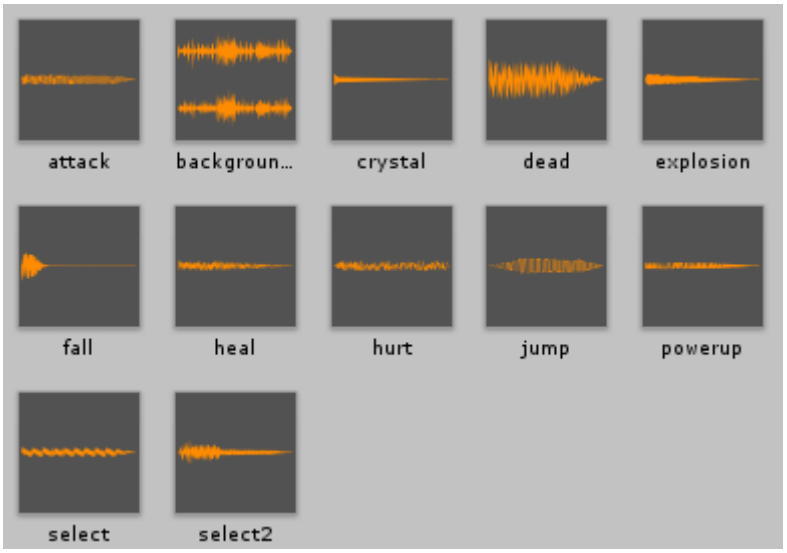
Depenent de si tornem al menú (primera pantalla), ens apareixerà un botó per continuar al següent nivell:

Aquest botó desapareixerà si morim o si no tenim cap joc guardat.

- **Sons:**

S'han afegit efectes de so tant pels menús de navegació com per efectes dintre del joc. Es podran activar o desactivar al gust del jugador, juntament amb la música de fons, la qual podrem parar-la o reduir-li el volum. Se'ls hi ajusta el volum i altres aspectes per a que encaixin dins del joc.

Aquests son els sons utilitzats al joc:



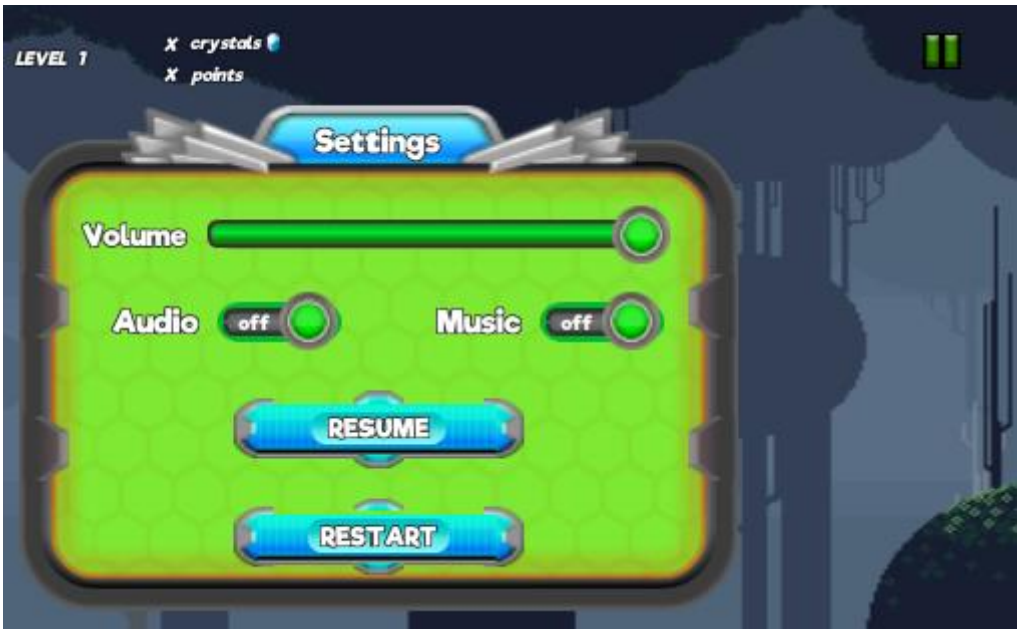
- **Botons i controlador:**

Per a la part d'Android, ha calgut ficar un Joystick per tal de controlar el moviment horitzontal del personatge a la part inferior esquerra i un botó de salt a la part inferior dreta. Els botons s'ajusten a la resolució de la pantalla, encara que al editor Unity apareguin bastant grans:

Captura de la versió Android

- **Menú de pausa:**

  Per a les dues versions, s'ha fet un botó per tal d'obrir el menú de pausa, a la part superior esquerra de la pantalla. Des de ell, podrem modificar paràmetres de so en qualsevol moment i tornar al menú principal:



2.3 Programació i Lògica

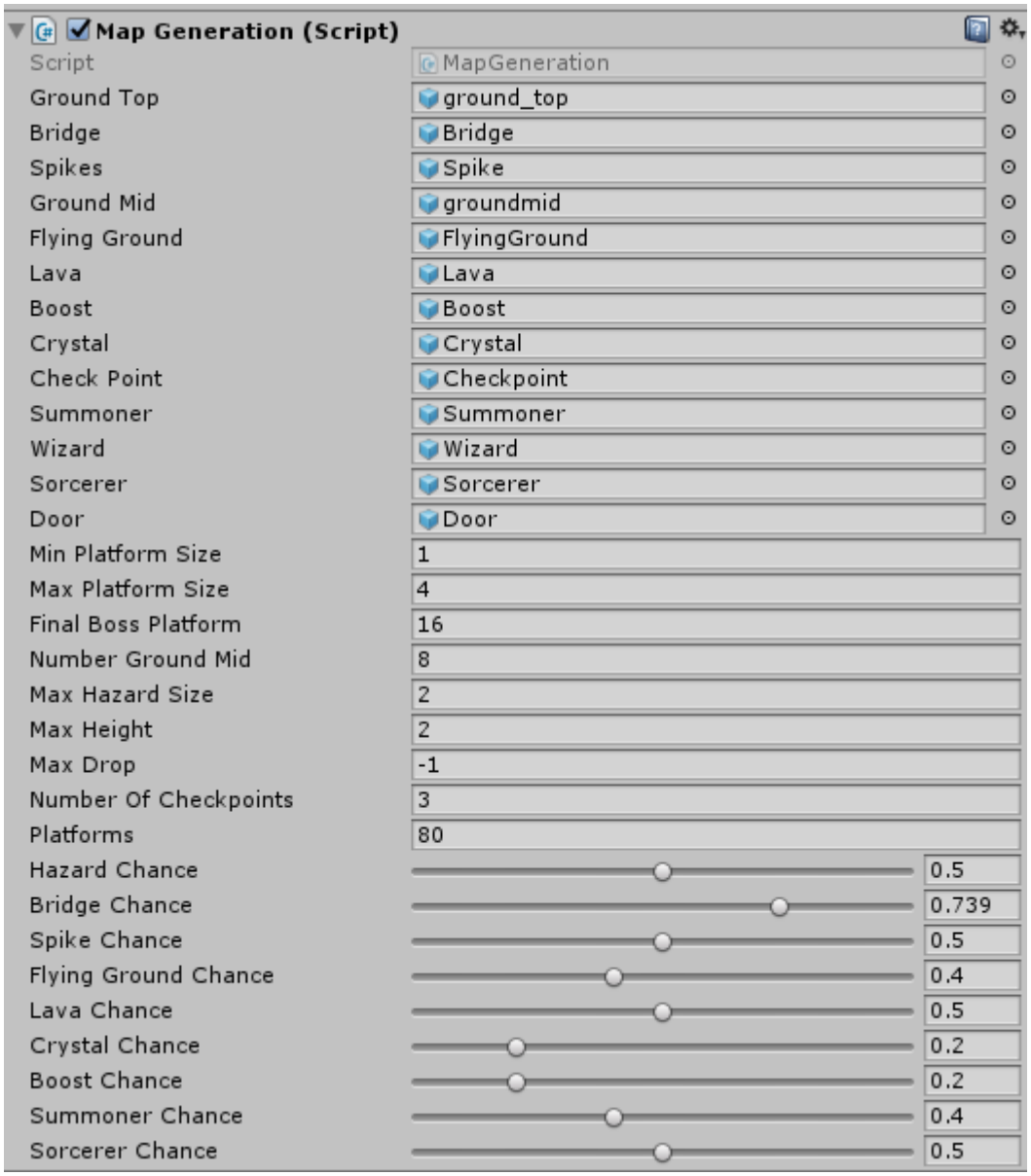
A continuació descriuré les classes creades, així com el detall de les seves funcionalitats i el propòsit de les implementacions. Els diferents **scripts** s'han agrupat per carpetes/categories, per facilitat d'accés.

Les classes principals son les següents:

MapGeneration:

Es la classe principal de creació del mapa i tot el que li envolta. El script arranca un cop està vinculat a l'objecte MapGeneration. En ell li facilitarem tots els objectes del mapa comentats anteriorment, a més a més d'alguns enemics que instanciaran a l'hora. A part, en aquest script, s'indiquen les diferents

aleatorietats per a la construcció de trampes, boosts, plataformes flotants, forats, lava, aparició de monstres, etc.



Alguns enemics també estan inclosos en la generació del mapa, per tal de facilitar la posició i l'aleatorietat dels diferents sprites.

Per un altra banda, cada vegada que s'avança de nivell, es modificaran alguns paràmetres de aleatorietat per fer el joc més difícil.

CheckpointManager:

El CheckpointManager serà el controlador del nivell, des del qual podrem accedir a les dades de les diferents posicions i vides dels enemics i del jugador. Estarà acoblat a un objecte amb el mateix nom per a poder iniciar-lo. S'utilitzarà cada vegada que es guardi o es carregui un estat, quan es passi per damunt del objecte CheckpointItem.

CheckpointItem

Es l'script realitzat per a l'objecte del mapa. Aquest tipus d'objecte cridarà al manager quan el disparador s'activi, que serà el primer cop que el personatge principal toqui o passi per damunt d'ell. A part, tindrà la transició de color a verd quan s'hagi activat i el so implementat per a l'activació.

SceneLoader:

Aquesta classe es l'encarregada de la transició de les diferents escenes o pantalles del joc, des de que comença fins que termina. Aquesta classe serà única i haurà d'estar present en totes les fases del joc, per tant ha de ser accessible des de qualsevol lloc. Algun paràmetre que també ha d'estar present en tot moment també hi serà en aquesta classe, com es el nivell actual que s'està jugant o el que està guardat.

ScoreManager:

Aquesta també es una de les classes que també han d'estar permanents en totes o bé en quasi totes les fases del joc. Serà l'encarregada de guardar la quantitat de punts i de cristalls que s'aniran recollint durant el joc.

AudioController:

Classe encarregada de l'àudio del joc. També serà única per arribar a totes les fases del joc. Respon tant als sons implementats dintre del joc, com a les diferents pantalles on es poden modificar els paràmetres de volum i silenci.

EnemyManager:

L'EnemyManager serà l'encarregat de fer aparèixer els monstres més bàsics (EyeMonster) segons uns paràmetres preestablerts. Entre aquests està el temps d'aparició dels monstres, tipus de monstre, temps entre aparicions i les seves localitzacions.

Per aquest cas, l'objecte que conté el script, estarà acoblat a la càmera del joc per a tindre un punt de referència i per a que els monstres apareguin dintre de la visualització del jugador.

EnemyBehaviour:

En aquest script es defineix el comportament que tenen tots els enemics. Cadascun dels objectes dels enemics hauran de contenir aquest component per a realitzar les seves accions. Es guarden tots els paràmetres requerits per a realitzar els moviments i els atacs, segons la posició de l'objectiu.

Es bastant complex degut a la metodologia utilitzada per a la recerca del jugador, basada en una llibreria externa que dota als enemics d'una recerca de tipus A*. Aquest algoritme de computació es basa en calcular un camí directe a un objectiu mitjançant un comportament. Per al nostre cas, l'enemic buscarà un camí directe cap al jugador i es mourà, però en el moment que es trobi una col·lisió que li impedeixi passar, farà un re càlcul d'aquesta recerca per tornar al camí sense aquesta col·lisió. Per tant, si l'objectiu es un objecte en moviment, s'haurà de cridar a l'algoritme de nou per a fer un nou re càlcul de la posició final del camí, apart del càlcul predeterminat de les col·lisions. Amb això

aconseguim que si hi ha un objecte que interfereixi com es el cas del mapa, el mateix enemic busqui una sortida que el condueixi de nou cap a l'objectiu.

AstarPath:

Script de la llibreria externa comentada, que conté la lògica del algoritme de computació A*. Com a script, estarà acoblat al objecte A*map, on s'indica la grandària del mapa de recerca per nodes i altres paràmetres com l'activació de les col·lisions, les capes de Unity que faran d'obstacles i el numero de connexions per node. Com estem generant un mapa d'objectes a l'hora que aquesta lògica s'executa, haurem de fer un escaneig del mapa d'objectes per mapejar i definir els obstacles al mapa de A*.

BGParallax:

Script amb la lògica del efecte Parallax, explicat prèviament. Estarà dins de l'objecte BGParallax i tindrà associats els diferents backgrounds amb els que volem realitzar l'efecte, juntament amb la quantitat de moviment amb la que volem moure els plànols. Aquest objecte juntament amb els objectes que contenen les diferents capes dels fons estaran dintre del contenidor WholeEscenario, amb la finalitat de tindre-ho empaquetat tot en un mateix prefab. Aquest prefab anirà acoblat a la càmera per a sincronitzar l'eix vertical i realitzar l'efecte de manera indefinida i sense problemes gràfics.

Scroller:

Script que controla els tres fons per tal de fer l'efecte scroll prèviament explicat. Estarà assignat a cadascun dels plànols.

GoblinCharacterController:

Es el controlador principal del jugador, on estan totes les característiques referents al moviment, als atacs, als items que pot rebre i als controls del

personatge. Aquest script està vinculat a l'objecte GoblinCharacter, que serà amb el que interactuarem com jugadors.

En ell es defineix la comprensibilitat de què és el terra pel jugador, on aquest podrà saltar si hi es damunt, i de totes aquelles físiques que li envolten. Un exemple seria el control de si està saltant o si està caient per un forat per tal d'activar diferents events.

A part, es defineixen els diferents tipus de controls que hi hauran per a Windows i Android.

DamageAbleItem:

Aquest script es com una propietat per a tots aquells objectes que podran rebre mal i destruir-se, incloent tant al Goblin com als Enemics. En ell es controla la mort de cada personatge i el seu comportament.

CollisionItem:

Script com a propietat de tot objecte que tingui la capacitat de col·lisionar amb altres objectes, amb la finalitat de controlar la causa i efecte de ambdues parts. Un altre finalitat d'aquest script es poder definir un mateix comportament entre objectes que tenen un component **Collider2D** (col·lisionador), sigui o no de tipus **trigger** (disparador).

DropRandomItem:

Es una classe simple que s'utilitzarà com a propietat dels enemics. Tal i com el nom indica, es una classe per a dotar a un objecte la possibilitat de deixar anar ítems (o objectes) predefinits de manera aleatòria. Per al nostre cas, s'aplicarà als enemics un cop morin.

PickupItem:

Es un script simple que atorga la propietat de poder obtenir l'objecte. En ell estan definits els tipus d'objecte que es podran agafar (cristalls i boosts).

Invulnerability:

Dota a un objecte de poder tindre la propietat de ser invulnerable durant un cert temps. Aquest serà un component de l'objecte GoblinCharacter i s'executarà cada vegada que el personatge rebi mal o bé mitjançant millores (boosts).

FlashingItem:

Script que serveix per a fer que un objecte amb un component **Renderer** faci intermitències gràfiques amb una determinada freqüència. Es atribuït al script d'invulnerabilitat per a representar-la, aconseguint l'efecte gràfic durant uns segons fins a que pari, on serem de nou vulnerables a dany.

LavaMovement:

Script simple on es controla el moviment de cada objecte lava. S'aplica una forca segons el moviment vertical i la distancia màxima de caiguda.

PlatformDown:

Script simple per controlar quan ha de caure un objecte de tipus pont o Bridge, que serà sempre i quan el jugador el trepitgi.

ApplicationUtil:

Classe d'utilitats que s'utilitza merament per veure en quin tipus de plataforma s'està executant el codi. Es utilitzada pels controls, dintre del controlador del Goblin.

Destroyable:

Propietat per a destruir un objecte que contingui aquest script quan marxi de la visió de la càmera, a través del mètode OnBecameInvisible.

FloatingDamageText:

Script simple per a l'animació dels textos flotants i el seu mostreig a pantalla. Estarà assignat al prefab PopupTextParent, que conté un canvas.

FloatingTextController:

Classe amb propietats estàtiques per a generar un prefab PopupTextParent cada vegada que es necessiti. Es el controlador dels textos flotants.

Shop:

Classe encarregada de la interactivitat de la tenda de millores del joc. Estarà aplicada al objecte ShopPanel, dintre del layout de les diferents ranures de millores. Els continguts que apareixen per pantalla aniran vinculats al script.

ContinueScript:

Script simple per a saber si hi ha partida guardada. Utilitzat per la primera pantalla.

CustomToogle:

Script simple per a controlar l'activació i desactivació de les diferents opcions sonores, dintre dels menús d'interfície habilitats. Utilitza la classe AudioController per a realitzar les activacions.

HealthBarScript:

Script per controlar gràficament la barra de vida del jugador. Estarà assignat al objecte de la barra de vida HealthBarBackground.

EnemyHealthBarScript:

Semblant a HealthBarScript però aplicat a cadascun dels enemics. Es podria haver utilitzat el mateix script que per a la vida del jugador, però la vida del jugador volia que fos estàtica per a poder rebre-la des de qualsevol lloc.

LifeScript:

Script controlador del sistema de vides. S'actualitza cada vegada que hi ha una diferència de vides i es comprova si realment n'hem perdut una, n'hem guanyat una o potser hem arribat al límit.

El paràmetre de vides el volem estàtic per tal de poder-hi accedir en tot moment.

RendererExtensions:

Script simple per tal d'utilitzar una funcionalitat extendida dels components de tipus Renderer. S'utilitza per veure si hi han objectes dintre de la visió de la càmera.

2.4 Interfície de l'usuari

La interfície de l'usuari està dissenyada amb l'objectiu de que sigui senzilla i vistosa, amb un estil retro.

La part posterior de la pantalla conté els elements més importants:

- Vida del jugador:

  Mostra la quantitat de vida restant del jugador. Si arriba a buidar-se, el jugador perd una vida de les que hi ha a sota de la mateixa barra de vida.

  A part, el jugador tindrà 5 vides com a màxim, i estaran situades just a sota de la pantalla.

- Nivell:

  Mostra el nivell actual. Estarà just a la dreta de la barra de vida:

- Punts:

  Mostra la quantitat de punts obtinguts per fer mal als enemics.

- Cristalls:

  Mostra la quantitat de cristalls obtinguts directament dels enemics, de l'escenari o de la tenda de millores.

- Botó de pausa:

  Aquest botó compleix la funció de parar el joc momentàniament i fa aparèixer un menú d'opcions on es poden desactivar els sons del joc, sortir al menú o continuar.

2.5 Mecànica del joc

El joc es basa en escapar del perill que l'envolta al jugador, a la vegada que es fa camí, derrotant els enemics. La fita és acabar el nivell sense ser derrotat i obtenir el màxim de punts possibles. D'aquesta manera, el joc consta de les següents parts principals:

- **El jugador**:

  El jugador ha de controlar el Goblin de manera àgil per preveure els diferents atacs i no col·lisionar amb trampes i altres monstres. Podrà moure's, saltar i disparar projectils. Aquest projectils faran mal als enemics cada vegada que col·lisionin.

  Per un altra banda, el jugador podrà anar beneficiant-se de les millores que li vagin caient dels enemics derrotats, o bé, de les millores generades al crear el mapa. Si estan generades, es trobaran a algunes plataformes flotants. Aquestes millores li permetran curar-se durant la partida, disparar múltiples projectils o invencibilitat temporal.

- **Els enemics:**

  Els enemics faran els seus propis moviments definits, amb els objectius d'atacar al personatge principal i derrotar-lo.

Hi hauran com a màxim 5 enemics del tipus EyeMonster, els quals reapareixeran constantment si son derrotats. Per l'altra banda, els enemics generats per nivell, son completament aleatoris donat un rang de valors, i son limitats pel nivell.

- **Pantalla de menú:**

  El menú principal del joc, que serà el primer que veurem quan l'executem. Podem començar una partida nova, reprendre la última partida guardada, accedir al menú d'opcions d'àudio o sortir de l'aplicació.

- **Pantalla de Retry:**

  Pantalla que ens apareixerà al perdre una vida i la que permetrà tornar-ho a intentar o tornar al menú principal.

- **Pantalla de mort:**

  Apareixerà quan al jugador no li quedin vides. A mode informatiu, es la pantalla de game over. Ens permetrà anar directes a la pantalla de menú principal.

- **Nivell completat:**

  Quan el jugador elimina l'últim enemic del nivell (Wizard), el jugador haurà completat el nivell, haurà guanyat una vida d'un màxim de 5 i s'haurà guardat la partida. En aquest moment, es veurà que s'activa el resplendor de la porta, per indicar-nos que s'ha obert per completar el nivell. Un cop anem a la porta, es mostren les diferents estadístiques i diferents opcions a escollir. Podrem anar al menú principal, anar a la tenda de millores o seguir jugant al següent nivell.

- **Pantalla de pausa:**

  En qualsevol moment, el jugador podrà polsar el botó de pausa per parar el joc, apareixent una finestra amb les diferents opcions

sonores, un botó per reprendre la partida i un botó per tornar al menú principal.

- **Progressió e increment de la dificultat:**

  El numero d'enemics augmentarà en cada nivell, pujant la freqüència de l'aleatorietat. Alguns paràmetres de generació del mapa, que també depenen de l'aleatorietat de l'aparició de trampes, etc, també s'incrementarà a mesura que hi hagin més nivells. El repte del jugador serà aconseguir cristalls per comprar les millores a partir de la tenda, i ser suficientment àgil per anar superant els diferents nivells.

3. Conclusions

De manera general, estic satisfet amb el resultat del joc en el temps establert. Tinc decidit que continuaré millorant-lo i dedicant-li més temps personal amb idees i pensaments que han anat sorgint durant el desenvolupament, per així publicar-ho en Google Play o en alguna plataforma i començar a omplir el dossier personal.

Per a poder arribar-ho a acabar, s'han hagut de planificar les diferents tasques de manera que es poguessin fer en un temps determinat, i ha calgut no fallar en els diferents temps. Per a això s'ha realitzat un diagrama de Gantt del projecte, que està publicat al Tomsplanner: **https://www.tomsplanner.es/public/arxy**

Per un altre part, el aprendre la eina de Unity ha fet que aquesta planificació potser balli al inici del desenvolupament. El aprendre un motor gràfic pot ser relativament fàcil si ja tens experiència en algun altre, però no ha sigut el meu cas, ja que es el primer motor que aprenc. Cal dir que ha sigut voluntari el escollir aquest motor gràfic per tal d'aprendre a donar-li us per a futurs jocs, així que podem dir que l'objectiu està assolit.

He de dir que l'apartat Audiovisual es el més complicat per nosaltres els programadors, bàsicament perquè no es realment la nostra feina. En aquest projecte em venien moltes idees per a fer diferents enemics o personatges, però com se que no hi havia temps suficient, vaig recórrer a els recursos online gratuïts. Encara que no ho sembli, si tingués experiència en aquest sector, podria haver fet els meus propis dissenys i no hauria invertit certa quantitat de temps buscant sprites amb aquestes idees que em sorgien al cap. Encara i això, vaig trobar força contingut de la comunitat on realment he pogut escollir el que he volgut ficar i n'estic molt satisfet. Per a l'apartat del Àudio vaig trobar un generador de sons i vaig anar fent els diferents àudios a oïda, fent sons aleatoris segons com anaven sonant. Si hi havia algun que em recordava a alguna acció del joc, l'editava i l'exportava.

Amb l'ajuda del consultor, he estat obert a modificacions que ell mateix i altres coneguts m'han anat dient segons anava desenvolupant. Crec que es clau per tal d'evolucionar i adonar-sen de les carències i de la obertura de noves possibilitats.

Finalment, en la meva opinió, un videojoc es un procés molt difícil per a una sola persona, a menys que tinguem unes idees molt bàsiques de vídeojoc i aquest no requereixi gaires recursos. Si volem ser originals, haurem de portar per la mà molts aspectes d'aquest procés que no només tracten en programar. Seguiré fent jocs de manera autodidacta per tal d'aprendre més conceptes que potser no s'han arribat a fer servir en aquest projecte, però sí que en un futur m'agradaria fer un projecte col·laboratiu amb gent especialitzada en disseny, so, gràfics, i altres camps. La creació d'aquest joc m'ha fet evolucionar en molts àmbits que desconeixia i en general ha sigut molt positiu per la meva carrera.

4. Codi Font e instal·lació

Per al codi, he utilitzat Unity 2017.3.1f1 Personal. Amb el projecte obert dintre del client, necessitarem instal·lar les llibreries externes que s'indiquen a la bibliografia, baixant-les directament de l'asset Store o desde la web del creador. Un cop descarregades les podem executar e importar directament.

5. Glossari

Sprite: Imatge visual amb estructura d'un mapa de bits.

Prefab: Contenidor de Unity a nivell de disc, per definir entitats i juntar continguts. Es poden utilitzar a nivell d'edició i execució, ja que seran **instanciables** en qualsevol moment.

GameObject: Contenidor de Unity a nivell de client. Li podem afegir fills, pares i components de Unity, que poden ser scripts o característiques. La diferència més característica amb prefab es que el gameObject no es guarda a disc sinó a la configuració del projecte de Unity.

Instanciable: Propietat dels gameObjects i dels prefabs. Es tracta de qualsevol objecte que pot desplegar-se tant a nivell d'edició com a nivell d'execució.

Script: Programa o classe que realitza diferents accions.

Collider: Es un component que poden tindre alguns objectes per saber si han hagut col·lisions amb altres objectes.

6. Bibliografia

- Sprites i animacions

  Web: https://lionheart963.itch.io/ - *Autor: Warren Clark*
  Web: https://opengameart.org/ - *Autor: ansimuz*

- Llibreries externes

  Web: https://arongranberg.com/astar/download *(Free Version) – Autor: Aron Granberg*

Web: https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-32351
Web: https://assetstore.unity.com/packages/2d/gui/casual-game-gui-skin-67196 - *Autor: Dead Mosquito Games*

- Instal·lació i us de l'entorn de desenvolupament Unity
  Web: https://unity3d.com/es
  Llibre: Learning C# Programming with Unity 3D - Autor: Alex Okita. Editorial CRC Press, 20140707, Boca Raton, EEUU.

- Audio
  Web: http://www.drpetter.se/project_sfxr.html
  Web: https://www.bfxr.net/
  Web: https://opengameart.org/content/menu-music

- Diagrama de Gantt
  Web: https://www.tomsplanner.es/

- Consultes sobre programació i scripts
  Web: https://es.stackoverflow.com/
  Llibre: Learning C# Programming with Unity 3D - Autor: Alex Okita. Editorial CRC Press, 20140707, Boca Raton, EEUU.

- Consultes sobre la creació i administració d'elements en Unity
  Web: https://forum.unity.com/
  Llibre: Learning C# Programming with Unity 3D - Autor: Alex Okita. Editorial CRC Press, 20140707, Boca Raton, EEUU.

7. Presentació

El vídeo de la presentació es pot trobar en el següent enllaç de Youtube:

https://www.youtube.com/watch?v=HP3sxM6QhGY&feature=youtu.be