

PLANE SURVIVOR

Cristian Alarcón Sanabria
Grau d'Enginyeria Informàtica
Videojocs

Professor: Javier Luis Cánovas Izquierdo
Professor Col·laborador: Joel Servitja Feu

09/06/2019



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>Plane Survivor</i> |
| Nom de l'autor: | <i>Cristian Alarcón Sanabria</i> |
| Nom del consultor/a: | <i>Joel Servitja Feu</i> |
| Nom del PRA: | <i>Javier Luis Cánovas Izquierdo</i> |
| Data de lliurament (mm/aaaa): | <i>06/2019</i> |
| Titulació o programa: | <i>Grau d'Enginyeria Informàtica</i> |
| Àrea del Treball Final: | <i>Videojocs</i> |
| Idioma del treball: | <i>Català</i> |
| Paraules clau | <i>Videojoc, 2D, Plataformes, Unity</i> |
| Resum del Treball (màxim 250 paraules): | |
| <p>El projecte final de grau Plane Survivor és un videojoc de plataformes en dos dimensions creat amb el motor Unity (versió 2018.2). L'objectiu d'aquest projecte a més del propi videojoc és la planificació i documentació de les diferents etapes durant el desenvolupament d'aquest.</p> <p>El joc tracta d'una pilot que s'ha estavellat en un volcà i ha d'escapar dels diferents perills que li proporcionarà el nivell de joc, com una lava que anirà pujant, enemics, punxes i estalactites que cauen, a més de totes les plataformes a saltar per a poder avançar.</p> <p>Degut a que fer un videojoc suposa un treball força ampli i només es disposa de poc més de 3 mesos per al projecte es van prendre una sèrie de decisions per a delimitar l'abast del projecte: només es faria un nivell complet, el projecte es faria en Unity i en 2D.</p> <p>La decisió d'escollir Unity és deguda a ser el motor més conegut, gratuït i amb molta documentació en línia, ja que abans de començar el projecte no es tenien coneixements previs en la creació de videojocs, el fet centrar-se en 2D també és deguda a la falta de coneixements previs.</p> <p>El resultat obtingut és un joc senzill on s'han intentat introduir diferents mecàniques com una lava que va pujant per a donar una sensació d'urgència al jugador i suposi un repte on la resta d'elements suposin un obstacle per a la nostra finalitat: sobreviure i escapar del volcà.</p> | |

Abstract (in English, 250 words or less):

The final grade project Plane Survivor is a 2D platform game created with Unity (version 2018.2). The objective of this project besides the game itself is the planning and documentation of the different stages during its development.

The game is about a pilot who crashes his plane inside a volcano and must escape from a distinct range of dangers provided in the level, for example a rising lava, enemies, spikes or even falling stalactites, besides the platform challenges themselves in order to advance.

Because the creation of a game requires a lot of time and we only have 3 months for this project a few decisions were taken before the beginning in order to focus the scope: only one level will be created, Unity will be the engine used and 2D will be the graphical choice.

The Unity choice is due to Unity itself being the most known free game engine with lots of documentation available online and prior to this project the knowledge for creating a game was non-existent, the same reason can be applied to the 2D choice.

The final build is a simple game where many mechanics have been introduced, a rising lava as a main focus to give a sense of urgency to the player where the rest of the elements are different obstacles to our goal: survive and escape from the volcano.

DEDICATORIA

Per a la Marta i l'Antei,
he fet un somni realitat vivint
una vida de somni amb vosaltres.

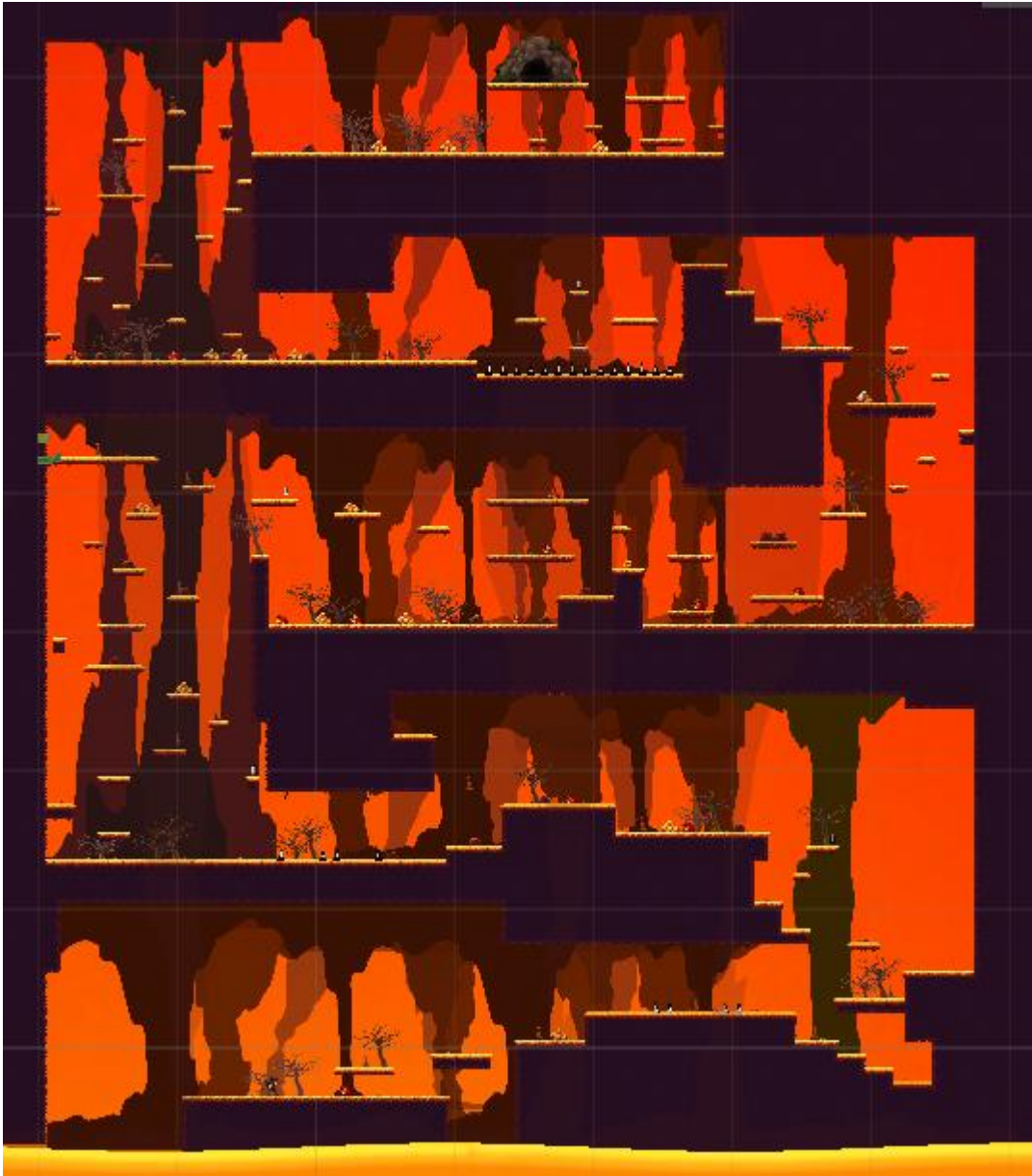
Índex

| | |
|--|----|
| 1. Introducció..... | 1 |
| 1.1 Context i justificació del Treball | 1 |
| 1.2 Objectius del Treball..... | 2 |
| 1.3 Enfocament i mètode seguit | 2 |
| 1.4 Planificació del Treball..... | 3 |
| 1.5 Breu sumari de productes obtinguts | 6 |
| 1.6 Breu descripció dels altres capítols de la memòria..... | 7 |
| 2. Estat de l'art | 8 |
| 2.1 Revisió del gènere..... | 8 |
| 2.2 Revisió sobre la tecnologia..... | 8 |
| 3. Definició del joc | 9 |
| 3.1 Canvis respecte a la proposta inicial | 9 |
| 4. Disseny tècnic | 12 |
| 4.1 Aspectes bàsics de Unity | 12 |
| 4.2 Entorn de Unity..... | 12 |
| 4.3 Els components d'un objecte més utilitzats | 13 |
| 4.4 Scripts | 15 |
| 4.5 Assets..... | 17 |
| 5. Disseny del personatge principal i enemics..... | 18 |
| 5.1 Personatge principal..... | 18 |
| 5.2 Script del personatge principal | 18 |
| 5.3 Animacions i animador del personatge principal | 20 |
| 5.4 Altres aspectes destacables del personatge principal | 21 |
| 5.5 Enemics..... | 21 |
| 6. Disseny del nivell principal | 23 |
| 6.1 Càmera del nivell..... | 23 |

| | |
|--|----|
| 6.2 Fons amb efecte Parallax..... | 23 |
| 6.3 Objectes del nivell: Grid..... | 24 |
| 6.4 Objectes del nivell: GameSession..... | 25 |
| 6.5 Resta d'objectes del nivell..... | 27 |
| 7. Altres escenes i interfície de joc..... | 30 |
| 7.1 Menú principal..... | 30 |
| 7.2 Menú d'opcions..... | 30 |
| 7.3 Pantalla de victòria..... | 31 |
| 8. Manual d'usuari..... | 32 |
| 8.1 Possibles interaccions..... | 32 |
| 8.2 Elements del HUD..... | 32 |
| 9. Conclusions..... | 33 |
| 10. Glossari..... | 35 |
| 11. Bibliografia..... | 37 |
| 11.1 Assets..... | 37 |
| 11.2 Consultes extra..... | 38 |
| 12. Annexos..... | 39 |
| 12.1 Com descarregar i executar el projecte..... | 39 |
| 12.2 Llistat d'errors solventats al llarg del desenvolupament..... | 41 |

Llista de figures

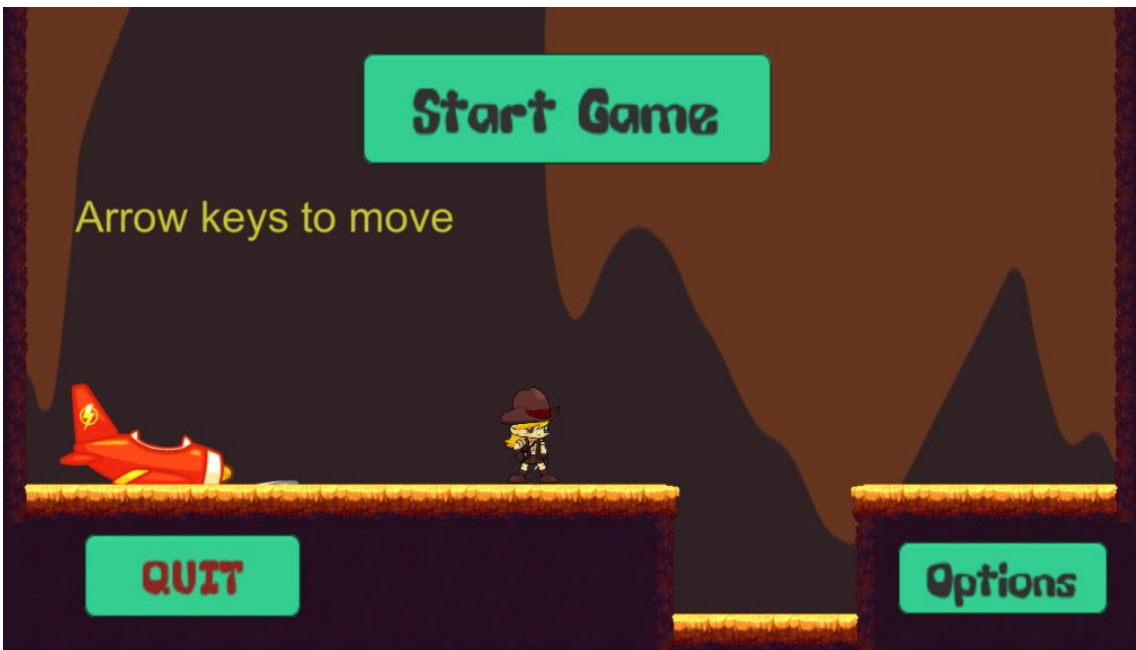
Nivell principal (i complet) del joc.



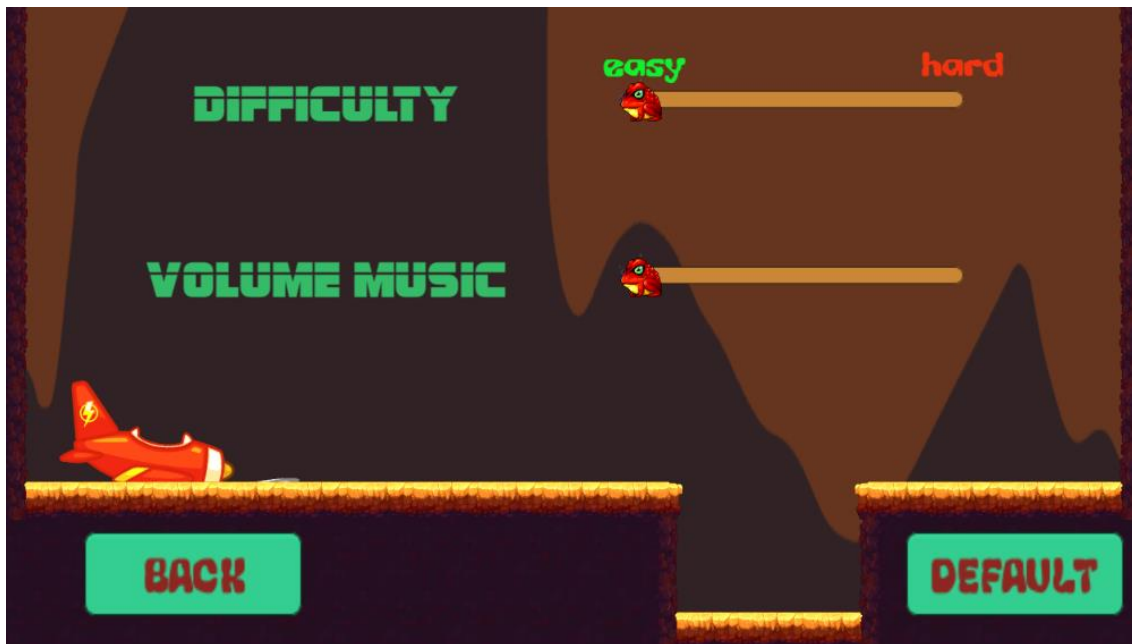
Personatge principal



Menú principal



Menú d'opcions



Pantalla de victòria



1. Introducció

1.1 Context i justificació del Treball

Els videojocs en l'actualitat són una eina d'entreteniment cada cop més universal que mou milions de persones ja siguin tant usuaris com creadors, el mercat no para de créixer i cada cop hi tenen cabuda un major nombre de títols i un major tipus de jocs, això implica que ara fer jocs tot i continuar essent una tasca complexa hi ha moltes més eines per a poder desenvolupar videojocs que fa uns anys, motors de desenvolupament com Unity, Unreal Engine o RPG Maker han esdevingut molt populars aquests últims anys i han democratitzat els videojocs aconseguint un corrent de jocs anomenats indies (individuals i molt personals d'autor) que estan definint més que mai la corrent artística de molts videojocs.

En aquest treball es veurà el procés de creació d'un videojoc, de totes les responsabilitats i etapes per a aconseguir treure endavant un títol que per molt modest que sigui ha de complir uns mínims des d'un punt de vista individual. Per tant no només hem de planificar el títol sinó encarregar-nos del disseny, les mecàniques, els gràfics, els menús, els sons, etc... tot això en un període de temps acotat a 3 mesos i mig més o menys.

Tot i les dificultats exposades hi ha varis motius que m'han portat a decidir fer aquest projecte:

- Sóc un apassionat als videojocs des de petit i el meu somni sempre havia estat poder fer algun videojoc (i treballar d'això tot i que ara sóc més realista i ho veig complicat).
- La meva recent paternitat m'ha fet replantejar el que deixem als nostres fills, el que influïm amb les nostres accions i la imatge que projectem en ells, per aquest motiu volia no només donar un sentit acadèmic a aquest grau sinó poder en un futur mostrar un producte de creació pròpia que sintetitzés l'esforç i dedicació dedicats aquests anys.
- Com a fanàtic dels videojocs volia fer un joc senzill però que suposés un repte i que fos una mica difícil per al jugador, que tinc la sensació que els videojocs s'han fet més fàcils i s'ha perdut la sensació d'emoció i recompensa que suposava el finalitzar un joc, així que m'he proposat que el joc sigui un repte.
- Aprendre i entendre el disseny de nivells en els videojocs i com les decisions més simples afecten al nostre projecte.

1.2 Objectius del Treball

- Planificar totes les etapes de creació d'un videojoc acotant-nos en els 3 mesos i mig imposats pel projecte i establir uns objectius a complir durant el desenvolupament per a tenir una visió real de l'estat del projecte.
- Aconseguir els coneixements necessaris per a poder utilitzar l'entorn de desenvolupament de videojocs Unity i C# per a la part de programació, per sort C# és molt similar a Java, el llenguatge que més utilitzat al grau.
- Dissenyar un únic nivell que suposés un repte per al jugador i doni una idea de com podria ser un joc complet.
- Establir una metodologia de treball setmanal basada en objectius per a tenir una sensació d'avenç real.
- Recerca d'assets a utilitzar al projecte per a aconseguir un grafisme agradable.

1.3 Enfocament i mètode seguit

L'enfoc del projecte ha estat primer obtenir cert coneixement practicant amb tutorials disponibles online per a poder tenir certa pràctica, en el meu cas he seguit un curs complet de la web didàctica Udemy.org amb 35 hores de vídeo, molt complet, l'havia comprat feia cert temps i ha sigut un molt bon moment per a treure-li profit (a més d'haver-se actualitzat el propi curs a Unity 2018), aquest curs tenia un enfoc pràctic en realitzar 3 videojocs diferents (una aventura només de text, un joc de naus tipus Space Invaders i un altre estratègic molt similar al Plants vs Zombies). Amb aquest curs ja s'obté una bona base per a saber com començar a fer el nostre joc.

El producte a desenvolupar és un joc de plataformes dels de sempre però canviant el desplaçament només horitzontal a tenir un enfoc sobretot vertical, ja que la lava va pujant i si ens toca morim (perdem una vida més bé) directament, es podria dir que està basat més bé en alguns dels nivells especials d'algun joc de plataformes on has d'escapar ja sigui d'una inundació o un escenari que es va destruint (com les inundacions del més recent Ori & the Blind Forest per exemple), la decisió d'afegir lava s'ha fet per a aportar certa sensació d'urgència al jugador i disposar d'uns quants elements pel nostre joc i no només les plataformes, els enemics i les trampes com a la gran majoria de títols... en definitiva per a diferenciar-se una miqueta ni que sigui de tots els jocs de plataformes clònics.

S'ha escollit primer invertir unes poques setmanes a aprendre Unity per a intentar evitar haver d'improvisar sobre la marxa i poder fer una millor planificació del joc tenint un mínim de coneixement del que es pensava desenvolupar ja que si no tenim una idea global acabem posant uns criteris una mica arbitraris, tot i així crec que en el moment de planificar encara no tenia una idea global del projecte com la tinc amb posterioritat.

Un cop finalitzat el curs, desenvolupar ha estat força més senzill i s'ha pogut trobar solució a tots els problemes que han anat sorgint i s'ha pogut complir tota la planificació amb força rigor i en el temps previst.

1.4 Planificació del Treball (en blau el temps real i en negre el planificat)

| Període | Fase | Data Inicial | Data Final | |
|---------|---|-------------------------------------|--------------------------|---|
| PAC 1 | Principi de la planificació | 20/02/2019 | 03/03/2019 | Complet |
| PAC 1 | Elaboració informe, cerca possibles assets i avanç al curs d'Udemy de Unity 2D | 20/02/2019 | 03/03/2019 | Apartat fet amb més aleatorietat que altre cosa, molt poc temps en la meua opinió, sense coneixements previs i improvisant. |
| PAC 2 | Versió Parcial | 04/03/2019 | 07/04/2019 | Complet |
| PAC 2 | Estudi de Unity: Acabar el curset que estic portant a terme per a poder tenir un coneixement necessari per al projecte, creació del punt de partida a GitHub. | 04/03/2019 Data d'inici correcta | 24/03/2019 01/04/2019 | El curs va esdevenir força més llarg (i complet). |
| PAC 2 | Cerca d'assets d'escenari, personatges i enemics per a poder almenys tenir mostres en imatge del joc. | 04/03/2019 Data d'inici correcta | 17/03/2019 15/04/2019 | Els primers assets escollits es van acabar canviant igualment. |
| PAC 2 | Inserció dels primers conceptes jugables: moviment del | 17/03/2019 01/04/2019 | 25/03/2019 15/05/2019 | Degut al canvi d'assets i a que la càmera va ser problemàtica durant tot el |

| | | | | |
|-------|--|-------------------------------------|--------------------------|---|
| | personatge, càmera que el segueixi... | | | desenvolupament no ho vam enllestir fins força tard. |
| PAC 2 | Disseny del nivell amb les plataformes i la lava pujant, elaboració informe | 25/03/2019 01/04/2019 | 07/04/2019 30/05/2019 | El disseny original (acabat a l'abril) es va decidir ampliar i fer el nivell més llarg. |
| PAC 3 | Versió Jugable | 08/04/2019 | 26/05/2019 | Complet |
| PAC 3 | Inserció del enemics i les rutines de moviment | 08/04/2019 Data d'inici correcta | 21/04/2019 31/05/2019 | Tot i completar a temps un dels enemics en el temps establert a última hora es va decidir afegir un enemic més. |
| PAC 3 | Col·lisions entre enemics i personatge, també cal incloure l'arma del personatge | 22/04/2019 15/04/2019 | 05/05/2019 20/04/2019 | Completat abans de temps, l'enemic afegit amb posterioritat no es compta perquè es ve reaprofitar el script de l'altre enemic. |
| PAC 3 | Animacions personatge protagonista i enemics | 06/05/2019 20/03/2019 | 19/05/2019 15/05/2019 | Es va començar abans per agilitzar feina, es va animar manualment el caràcter previ, però es va descartar i el nou ja venia amb animacions. |
| PAC 3 | Selecció i addició efectes de so i música, elaboració informe | 20/05/2019 20/05/2019 | 26/05/2019 22/05/2019 | L'apartat més senzill, tot i semblar fet força ràpid vaig estar un parell de dies escoltant una bona quantitat de sons i |

| | | | | |
|-----------|---|--------------------------|--------------------------|--|
| | | | | músiques. |
| PAC Final | Versió Final | 27/05/2019 | 09/06/2019 | Complet |
| PAC Final | Menús i interfície | 27/05/2019 10/05/2019 | 02/06/2019 25/05/2019 | Enllestit amb força facilitat (gràcies al curs), modificat a posteriori per suggeriment del tutor |
| PAC Final | Testeig i retocs finals, elaboració informe | 03/06/2019 01/04/2019 | 09/06/2019 04/06/2019 | Aquest apartat realment ha format part del projecte de bon principi, m'hauré passat el nivell 200 vegades tranquil·lament. |

Com es pot veure entre la planificació i el temps real hi ha moltes diferències en quant a temps, però com ja he comentat la planificació es va fer sense coneixement i era més una improvisació que altra cosa, ja que ni jo esperava fer aquests temps, però havia de posar alguns per a fer la primera entrega, tot i així estic feliç d'haver aconseguit fer-ho tot i no haver-me deixat gairebé res.

El projecte va començar una mica precipitat, jo ja havia començat el curs de Unity però encara l'havia de finalitzar, per tant per a la preparació de la primera PAC que era una elaboració de la planificació del projecte no em vaig voler arriscar a intentar fer un joc molt ambiciós per por a no ser capaç, vist amb perspectiva no me'n penedeixo perquè tot i haver fet el curs la feina que requereix fer qualsevol aspecte del joc és força gran i haver omplert amb més objectius la planificació hagués aconseguit desmotivar-me per no poder anar complint les dades marcades, ja que algunes setmanes he anat just tot i així.

L'elaboració del joc va començar amb l'elecció d'assets tant pel personatge principal i enemics com per a l'escenari, vaig començar amb un personatge principal masculí que no pegava gaire amb la resta, el professor m'ho va fer veure, la veritat és que m'hagués costat desfer-me del personatge per la feinada que va ser animar-ho manualment extremitat per extremitat com vaig fer (no gaire bé tot i així degut a la meva inexperiència) durant una setmana. Per sort l'asset que va substituir el protagonista acompanya molt millor l'escenari i enemics i no s'havia d'animar ja que venia amb uns sprites amb animacions frame a frame.

Un cop tenim el personatge i assets del nivell es comença amb el disseny del nivell en sí, mitjançant l'ús de tilesets per a poder ficar les primeres plataformes

i poder interactuar amb el nivell i crear la lava que pugui, a més de poder inserir ja per fi els enemics i punxes com a obstacles.

Amb gairebé tots els elements del joc es comença a treballar amb la interacció entre ells com les col·lisions entre objectes, les rutines de moviment dels enemics, treballar amb l'efecte parallax del fons i la càmera que seguirà al jugador.

L'efecte parallax i la càmera són els dos aspectes que més maldecaps han generat, el primer perquè treballar amb diferents plànols provoca que quedi tot molt irreal si que no s'ajusta bé, la superposició de diferents fons provoca (segons l'angle) alguns objectes fora de posició i queden lletjos en perspectiva, la càmera en canvi no només ha generat molts problemes sinó que ha afectat a altres aspectes com les animacions del personatge protagonista, ja que intentant fer que la càmera es posés darrera la posició del personatge segons la seva direcció va provocar una reprogramació de la direcció del personatge per a poder controlar-ho fàcilment (que ja de pas s'ha aprofitat alhora de calcular la trajectòria del tret i l'impacte amb els enemics), però després es va modificar igualment a una única càmera intel·ligent.

Un cop ja tenim completa la interacció i la càmera ens vam posar a refinar el nivell amb cada plataforma posada en una distància correcta, amb un desplaçament i una direcció concreta que anés en relació a la velocitat de pujada de la lava i la capacitat de salt decidida, juntament amb això vam afegir una acció que baixés el nivell de lava una mica en un punt concret si s'agafen 5 o més flascons durant el desenvolupament de la partida, és possible finalitzar el nivell sense fer això però força complicat...

Amb aquestes addicions només restava desenvolupar els menús de joc i les diferents pantalles auxiliars: pantalla inicial, pantalla d'opcions i la pantalla final on es veurà la puntuació de col·leccionables recollits.

A més de tot l'esmentat prèviament a la fase de depuració final s'han modificat totes les col·lisions de l'escenari ja que provocava que el personatge es rossegés una mica al saltar molt a prop de les parets, així que manualment s'han creat de zero també, amb tot el nivell ja dissenyat hem creat un checkpoint a mig nivell per si es mor i encara li queden vides al jugador pugui començar a un punt avançat del nivell.

1.5 Breu sumari de productes obtinguts

- Personatge principal amb animacions de repòs, córrer, saltar i disparar.
- Enemics amb un desplaçament lateral amb animacions de caminar i de morir per l'impacte d'un tret (no moren al saltar al damunt), al entrar en contacte amb el personatge principal li resten una porció de vida.
- Punxes que al entrar en contacte amb el personatge li treuen una porció de vida.
- Lava creada principalment com un quadrat simple que es desplaçava i poc més, però que ha acabat evolucionant en una malla tridimensional

(plana això sí) per a simular un efecte d'ona, al entrar en contacte amb el personatge principal aquest directament mor i es reinicia el nivell, la lava a més puja a diferent velocitat segons el punt on es trobi el personatge protagonista, a més es pot baixar el seu nivell si recol·lectem flascons.

- Estalactites amb punxes que cauen al apropar-nos que al entrar en contacte amb el protagonista aquest perd una porció de vida.
- 2 elements permanents que un cop recollits durant la mateixa partida no tornen a sortir: bales per a augmentar el carregador de la pistola i flascons que serà el col·leccionable del nivell.
- Pantalla d'inici de joc que contextualitza l'accident (visualment) i fa de petit tutorial.
- Pantalla d'opcions on podem seleccionar la dificultat i el volum de la música.
- Nivell complet amb checkpoint intermedi amb tots els elements de joc creats i una sortida a final del nivell.
- Pantalla de felicitació on es veu la puntuació (els flascons recollits).
- Menú de pausa al prémer la tecla p durant la partida, el joc es pausa i podem tornar al menú, tornar al joc o reiniciar el nivell (els objectes permanents tornen i comencem de bon principi el nivell).
- Menú de game over cada cop que morim i no ens queden vides restants, com a opcions tenim reiniciar el nivell o tornar al menú principal.

1.6 Breu descripció dels altres capítols de la memòria

- Capítol 2: Estat de l'art: revisió del gènere i revisió sobre la tecnologia
- Capítol 3: Definició del joc: idea inicial i canvis al llarg del projecte.
- Capítol 4: Disseny tècnic: Aspectes bàsics de Unity – Estructura alhora de crear un videojoc, interacció entre objectes, aspectes destacables.
- Capítol 5: Disseny del personatge principal i enemics, animacions, col·lisions i programació en C#.
- Capítol 6: Disseny del nivell principal: fons parallax, control de la lava, tilesets i càmera.
- Capítol 7: Altres escenes i interfície de joc, on comentarem els menús que es despleguen, el control que es té sobre ells, l'elecció de fonts, les opcions de que es disposen i com afecten al nivell principal.
- Capítol 8: Manual d'usuari, explicació dels controls.
- Capítol 9: Conclusions del projecte.
- Capítol 10: Glossari amb tot el vocabulari tècnic o referencial d'ús.
- Capítol 11: Bibliografia.
- Capítol 12: Annexos. Com descarregar el joc i llista d'errors trobats al llarg del desenvolupament.

2. Estat de l'art

2.1 Revisió del gènere

El gènere de plataformes 2D a l'actualitat no és un gènere gaire predominant, Nintendo és qui treu més jocs d'aquest gènere ja que va desenvolupant entregues 2D de Mario a gairebé cada plataforma, també té els Donkey Kong Country o els jocs de Yoshi i Kirby que tot i tenir un enfoc més en la exploració són referents vàlids.

Altres companyies grans pràcticament no toquen el gènere, que recordi Ubisoft va treure fa uns pocs anys l'últim Rayman (un joc preciós amb un motor gràfic UbiArt propi que sembla una delícia, tot i que Ubisoft ha reconegut que no l'ha pogut utilitzar gaire degut a la dificultat de treure-li profit) i Electronic Arts va treure no fa gaire Unravel 2, un joc amb un personatge que és un ninot de fil.

En quant a jocs indie sí que hi han hagut jocs de plataformes aquests darrers anys, tot i que han acostumat a ser propostes poc comercials i normalment amb una dificultat infernal com el cas del Super Meat Boy, en canvi a dispositius mòbils sí que s'han prodigat més però no com les propostes clàssiques sinó jocs amb personatges de moviment automàtic on nosaltres escollim el moment de salt o moure'ns per uns camins prefixats, però sense la llibertat de control del gènere.

2.2 Revisió sobre la tecnologia

Nintendo no acostuma a parlar mai dels seus motors de desenvolupament, tot i que sorprenentment l'últim Yoshi (Yoshi's Crafted World de Switch) utilitza Unreal Engine, cosa que suposo és justificable per l'estil artístic escollit amb una forta presència de materials com paper, cartró i objectes de manualitats.

Ubisoft com ja he comentat utilitza el seu motor propi UbiArt i sempre hi ha hagut veus a favor d'una comercialització o una esperança a un major ús del motor però fa poc Ubisoft va comentar que es troba en desús per la seva complicada interfície, poc més que els desenvolupadors originals l'han utilitzat i els pocs jocs externs van necessitar de molta assistència de l'equip creador original, una llàstima.

Unravel utilitza un motor gràfic propi anomenat Phyre Engine, tot i ser un joc distribuït per Electronic Arts es podria dir que gairebé és un projecte indie si no fos per la financiació al darrere d'EA.

Altres títols indie han estat desenvolupats amb Unity com Hollow Knight que és un dels més bonics tot i estar més centrat en l'exploració, però segueixen havent-hi títols amb motor propi, ja sigui per la seva aposta pel píxel art o perquè els desenvolupadors tenen molta destresa i no volen donar una part del benefici de la seva creació a l'empresa creadora del motor de desenvolupament.

3. Definició del joc

El concepte és un joc de plataformes 2D de tota la vida d'avançar i saltar, on la majoria de títols sempre el desenvolupament era d'esquerra a dreta aquí el volia fer una mica més vertical i a mi personalment sempre m'han agradat els nivells on no tens temps per a parar i mirar l'escenari, sinó que hi ha una sensació d'urgència provocada perquè o bé tot s'està inundant o has d'escapar d'algun altre perill, així que per a no fer tan simple el títol de només desplaçar-te i avançar fins al final doncs posant la lava aconseguim un joc menys convencional, on el repte no és matar als enemics sinó arribar al final on els enemics i perills són merament obstacles per al teu objectiu.

El context del joc és el següent:

- Som una pilot que acaba d'estavellar l'avió després de problemes de motor, amb tan mala sort que aterrem a dins d'un volcà, on en breus minuts entrarà en erupció amb nosaltres a dins, així que avançant pel terreny i esquivant enemics hem d'anar pujant fins trobar la sortida del volcà.

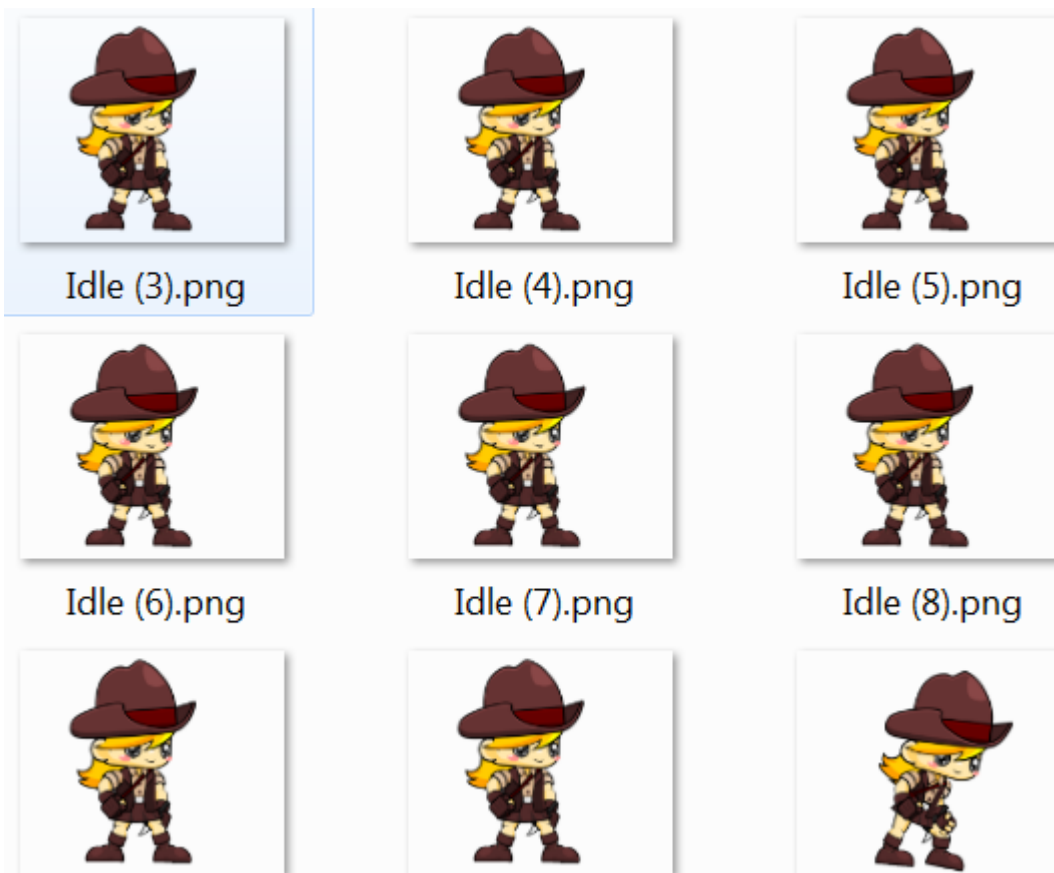
3.1 Canvis respecte a la proposta inicial

Primer canvi: personatge principal

El nostre protagonista anava a ser un personatge que havia trobat a parts i el vaig començar a animar, però la meua inexperiència va provocar que les animacions quedessin completament irreal, a més el personatge no s'acabava d'integrar a la resta d'assets, ja que eren colors molt plans i en canvi els assets de l'escenari eren més colorits i sobretot a més resolució.



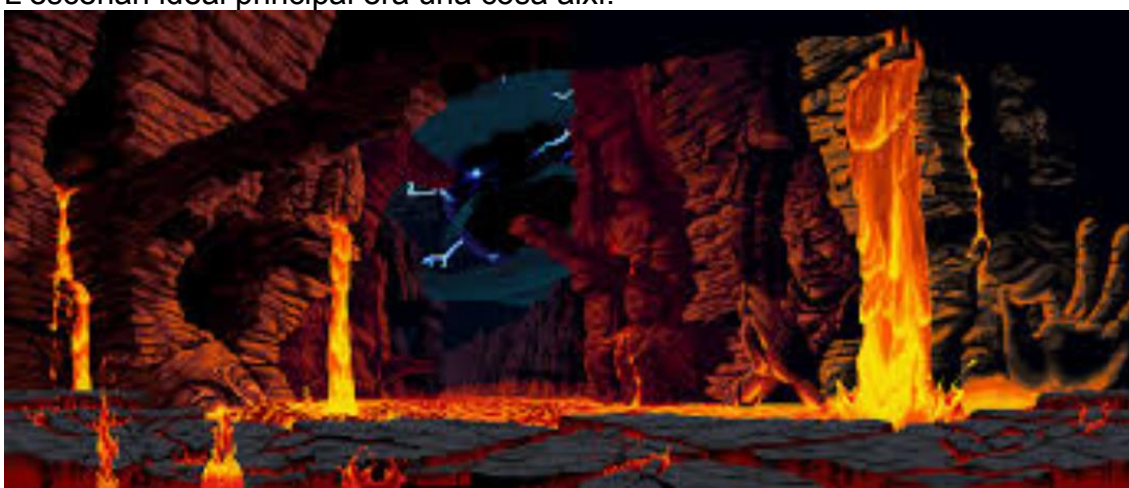
Al final vam acabar amb aquest personatge:



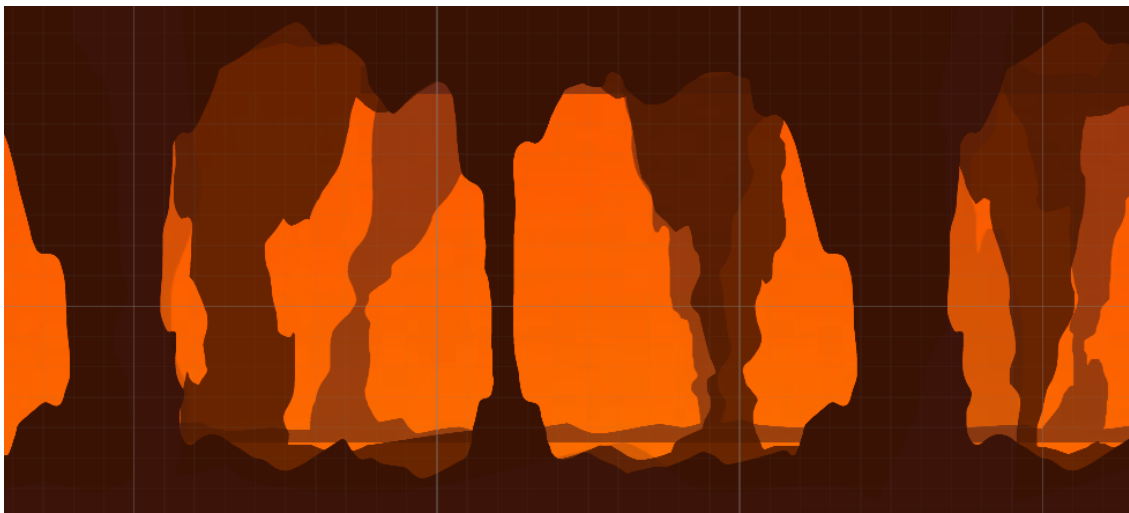
Un personatge amb molta més resolució, més simpàtic i més colorit, a part de venir amb un set bàsic d'animacions.

Canvis a l'escenari

L'escenari ideal principal era una cosa així:



Però clar, no vaig trobar res gratuït a la altura, al final vaig agafar un asset de cova que vaig trobar i li vaig modificar el color, de pas vaig posar una animació al fons simulant com la llum de la pròpia lava...



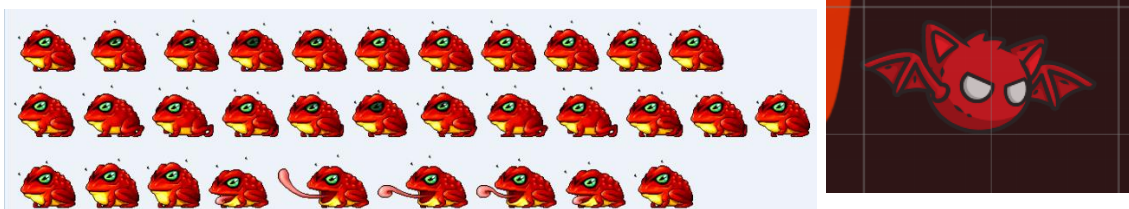
Tot i no ser el mateix que de bon principi no està malament i ens va força bé per a l'efecte Parallax.

L'objectiu i possibilitats del joc no s'han vist reduïdes, ja que la Lava es desplaça bé i els salts són prou ajustats, en aquest aspecte diria que no hi ha canvis. A la llista de figures es pot veure el nivell sencer en una única captura.

Sobre els enemics durant la primera PAC encara no sabia que agafaria, vaig posar una captura d'uns ossos i pensava animar manualment també els enemics, al final per sort es va descartar veient els resultats de l'animació del personatge principal descartat.



Per sort els enemic trobats han estat força resultants, ja que es mouen prou bé i queden millor sobre la resta d'assets...



4. Disseny tècnic

4.1 Aspectes bàsics de Unity

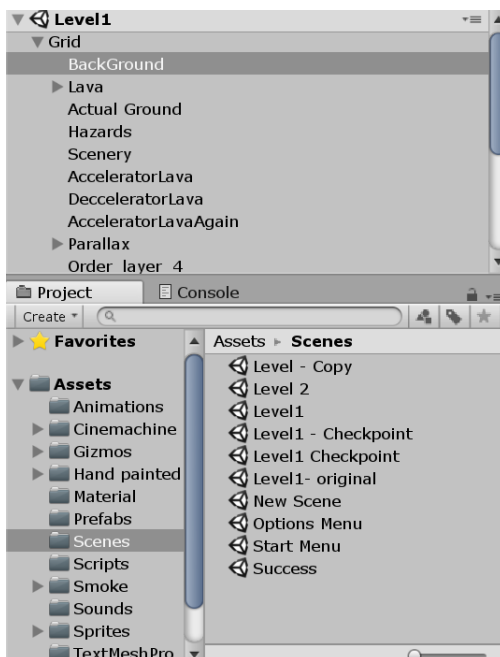
Unity es pot resumir en dos aspectes principals: les escenes i els objectes, es podria dir que tot videojoc creat en Unity s'estructura en que cada escena sigui un dels nivells del joc i que el contingut de cada nivell siguin els objectes. Una escena a Unity és principalment un nivell de joc, un entorn on ficarem els objectes que defineixen el joc.

Un objecte a Unity és la unitat atòmica més petita amb la que treballar i sobre la qual li podem afegir tants components com volguem, un objecte l'únic que té definit és una posició dins de la escena (component Transform), tota la resta que tingui són components, a més els objectes els podem englobar dins de tags o layers, que els podem crear i després englobar-los, això ens servirà per a en un futur les interaccions entre objectes, per exemple si volem una interacció entre el escenari (que poden ser un munt d'objectes) i el personatge principal podem controlar el comportament entre el personatge i algun tag o layer que englobi tot un conjunt d'objectes estalviant-nos haver d'especificar comportaments individuals similars entre varis objectes.

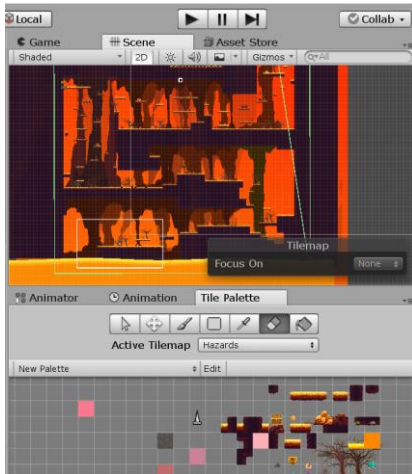
A més dels objectes i escenes també existeixen controladors d'animacions i animacions, que són però que s'associen a un objecte. Hi haurà més coses però no han sigut necessàries per al projecte.

4.2 Entorn de Unity

Cada escena de Unity disposa d'un entorn de treball que podem dividir en quatre parts:

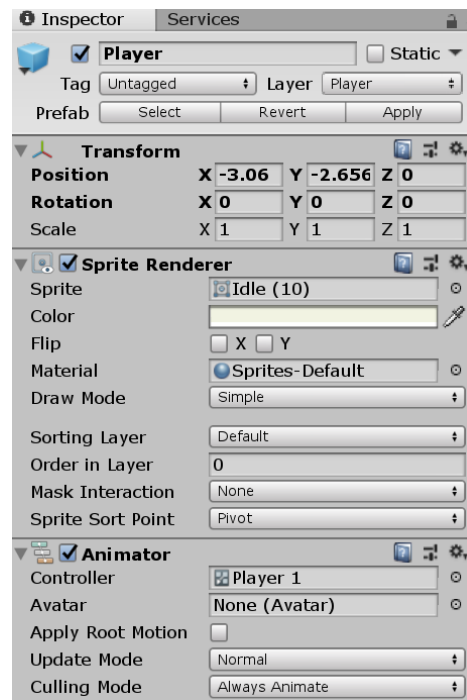


- De primeres a la part esquerra tenim aquest entorn, on tenim tots els objectes de la escena i a la part de sota tenim l'organització del propi projecte on veiem en una estructura de carpetes el contingut de tot el projecte separat per les escenes, els scripts o qualsevol carpeta que creem que pugui afegir una millor organització al projecte. A més a més tenim la pestanya console on veurem totes les notificacions i errors que es produeixen en l'execució del projecte, aquests errors vindran marcats en vermell i els avisos en groc, qualsevol objecte creat si l'arrosseguem a la carpeta Prefab ens quedarà com un objecte especial molt més fàcil de clonar



- La part central del menú de Unity disposa de la pantalla del nivell, separada per game, que és el que es veurà per la càmera, Scene que és la disposició de la escena on ens podem moure lliurement. Damunt de la pantalla tenim un botó de plaer que ens executa la escena. A la part de sota tenim un menú on veure la paleta de tiles, l'animador i les animacions.

- Per últim tenim a la part dreta el que anomenem Inspector on cada cop que seleccionem un dels objectes podem veure tots els seus components, aquí podem afegir tants components com ens facin falta, si per exemple tenim el personatge principal es pot veure que tenim el Sprite Renderer (que és el que ens permet afegir una imatge a l'objecte) i l'Animator, que és un controlador per a gestionar les animacions que volem que el nostre objecte pugui fer.



4.3 Els components d'un objecte més utilitzats

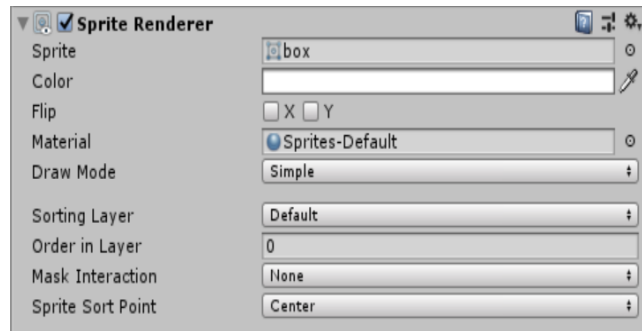
En el moment de crear una escena Unity per defecte ja crea un objecte com a mínim anomenat Main Camera que no és més que un objecte amb el component càmera, com ja hem comentat abans els components són els que defineixen que podrà fer i que no podrà fer l'objecte, així que posarem un llistat dels components utilitzats i una breu explicació de la utilitat que els hi hem tret.



- Càmera: Component ja nombrat, utilitzat per a que el joc utilitzi aquest objecte com a punt de vista, podem escollir la mida del focus, els objectes a visionar segons capes, el tipus de projecció (completament plana o tenint en compte la profunditat), a més d'altres aspectes que ja no s'han aprofundit al projecte com els d'aquí a l'esquerra.

- AudioListener: Component bàsic per a la reproducció de so, si hi ha objectes que generen so però la càmera no té AudioListener no s'escoltarà res al joc.

- SpriteRenderer: és el component que ens permet introduir un sprite a on es troba situat el objecte, podem afegir qualsevol imatge, modificar el color i especificar la mida del sprite, al igual que l'ordre per capes (per a controlar la superposició visual d'objectes, etc...).

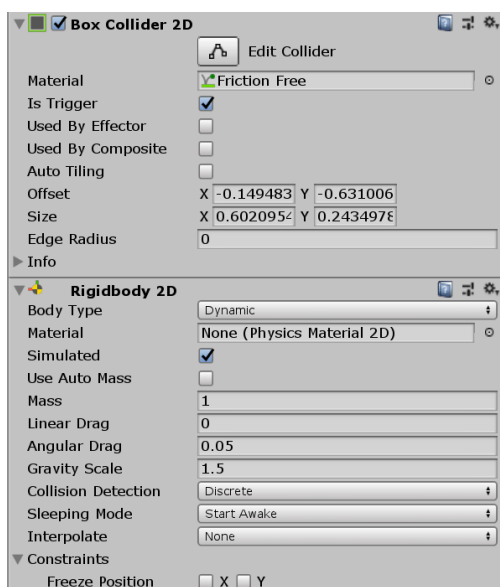


- Grid: Component per a especificar una quadrícula sobre la qual utilitzarem un altre component (els tiles) alhora de dissenyar l'escenari del joc.

- Tilemap: Conjunt de tiles (que haurem exportat a la paleta de tiles prèviament) que podrem posar sobre la quadrícula, els objectes amb tilemaps és recomanable posar-los dins d'objectes amb el component Grid (sí, podem ficar objectes dins d'altres objectes com si fos una estructura de carpeta), la diferència entre tenir un tile o ficar imatges directes a la escena és que si posem una imatge es crea un objecte individual amb el seu sprite renderer, en canvi un objecte amb el seu tilemap pot tenir tants tiles com posem de la paleta de tiles com vulguem i tots formaran part d'un únic objecte, això ens permetrà ficar un collider que englobi tots els objectes de cop per exemple.

- Collider: de primeres hi ha diferents tipus: segons les formes poden ser BoxCollider, SphereCollider, PolygonCollider, etc... són bàsicament les línies de col·lisió que establim per a qualsevol objecte, podem fer que sigui un obstacle (que no es pugui travessar) o que no sigui un obstacle

però activi un disparador (trigger) que després mitjançant un script podrem programar-li accions, si per exemple volem controlar que la lava pugui més ràpid en cert moment llavors podem posar un BoxCollider que cobreixi una zona i que al entrar el personatge la velocitat de pujada sigui una nova establerta. També hi ha un parell de colliders anomenats tilemap Collider i Composite Collider que serveixen per a fer que tot un tilemap tingui un collider global per a totes les tiles que englobi.



- Rigidbody: component que transforma un objecte en una entitat física amb la possible aplicació de les lleis de la física (aspecte que Unity controla internament permetent modificar la gravetat, la massa, la detecció de col·lisions, etc...), els cossos rígids poden ser dinàmics (on tenen pes per exemple subjecte a la gravetat), kinematic (on no li afecten les lleis de la física però es poden moure) o static on són objectes fixos.
- Text i Canvas: El component de text permet crear línies de text a l'escena, tenen una particularitat, cada cop que creem un text l'escena crearà un objecte anomenat Canvas i tots els objectes de text aniran dins el Canvas, perquè això? Perquè els textos seran independents de la càmera, seran objectes que es superposen per damunt de la càmera, diguem que el Canvas és una capa de superposició per damunt de la càmera, suposo que també es poden treure els objectes amb text fora del Canvas (ho hem fet afegint un text a la zona d'activació de l'aigua). El propi objecte Canvas a més porta components propis com el Canvas Scaler que adapta la posició dels objectes dins el Canvas segons la posició o el número de píxels que li diguem i un component anomenat Graphic Raycaster que és un calculador d'iluminació que calcula la projecció dels rajos de llum, bé, en el nostre cas no l'hem tocat perquè no tenim fonts de llum ni res. El sentit que li veig al Canvas és com a capa per a interfície, trobant-se independent del propi joc i els seus elements.

4.4 Scripts

L'aspecte primordial de Unity per a controlar objectes, a cada objecte li podem ficar tants scripts com volem i ens permetran establir paràmetres, comportaments i la gestió i relació amb altres objectes i els seus respectius scripts.

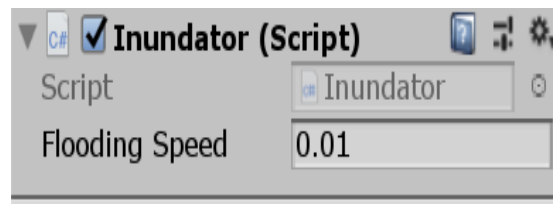
Els scripts funcionen en llenguatge C#: un llenguatge de Microsoft orientat a objectes molt similar a Java, una sort ja que aquest ha estat el llenguatge més utilitzat durant el grau i no ha calgut gairebé cap aprenentatge, només tenir en compte les funcions més utilitzades per defecte a Unity úniques de Unity.

- Awake: la funció Awake és una funció que s'executa abans de la pròpia càrrega de l'objecte i els seus paràmetres al script, aquesta funció va molt bé si volem per exemple que un objecte no es destrueixi quan carreguem una altra escena o que es mantingui en altra escena destruint el que es crea en aquella, simplement fem una funció que controli en nombre d'objectes que hi hagi i li afegim la funció DontDestroyOnLoad per a que no es destrueixi, això ens va molt bé per a tenir en compte la persistència d'objectes o tenir el mateix objecte en varies escenes (conservant el valor dels atributs nivell rere nivell, com per exemple per a la puntuació dels flascons al nostre títol o la vida del personatge al nivell, cada cop que morim i es reinicia el nivell hem de restar una vida i carregar de nou.

- Start: funció que s'utilitza un cop s'ha carregat l'objecte que conté el script, on podem inicialitzar paràmetres o establir les relacions amb altres objectes o cridar funcions necessàries a l'inici d'una escena.
- Update: aquesta funció s'utilitza per a interaccions o accions que volguem actualitzar constantment, ja que s'executa a cada frame del joc, per tant per a tenir per exemple inputs de teclat (per a moure el personatge) o per a tenir moviments predefinits (com fem amb la lava, que movem la posició de l'objecte (el seu component transform) fent que s'auto sumi un valor molt petit a cada frame en l'eix d'ordenades.
- OnTriggerXXX (OnTriggerEnter2D, OnTriggerExit2D...) aquesta funció s'activa cada cop que entrem en contacte amb un collider, ja sigui perquè entrem o sortim o altres opcions (que nosaltres no hem utilitzat), a més la funció ens retorna el objecte amb el que col·lionem, per si volem interactuar amb ell, per exemple si un enemic col·lionava amb una bala disparada doncs llavors aquí faríem que el personatge morís.

A més de les funcions esmentades podem crear tantes funcions com considerem com a Java, a més de paràmetres a modificar, que funcionen igual amb una particularitat, si en el moment de declarar un paràmetre el fem com a públic o amb un [SerializeField] al davant aquest paràmetre es podrà modificar des de l'inspector de Unity a l'objecte on sigui el script.

Aquí tenim per exemple el script Inundator que forma part de l'objecte Lava, aquest és un script senzill que controla la posició de la lava, on al propi script hem posat un paràmetre que ens dirà l'espai que es desplaça l'objecte cada frame i el podem modificar des del propi inspector de Unity.



Altres aspectes a tenir en compte en quant a usos que hem fet al projecte és la possibilitat de relacionar objectes o directament scripts, si per exemple volem que al passar per un lloc pugi més ràpid la lava podem posar un objecte amb el seu propi collider i un script que quan detecti al jugador cridi a una funció del script pertanyent a Lava per a accelerar la velocitat de la pujada de la lava. Això ho podem fer o bé buscant l'objecte en qüestió i agafar el seu component i emmagatzemar-lo a un paràmetre declarat prèviament de la mateixa classe o bé buscant i cridant alguna funció pròpia del script buscat.

Emmagatzemar direccions d'objectes és útil quan per exemple volem activar o desactivar objectes en moments concrets (com per exemple per a visualitzar menús al prémer el botó p per a posar en pausa el joc), ja que podem així des d'un únic script controlar molts aspectes i no haver de tenir un munt de scripts.

Una altra funció important a Unity són les corutines que són funcions on podem controlar el temps, per exemple si volem activar un esdeveniment i que duri 3 segons doncs utilitzem una corutina on es cridarà l'esdeveniment, s'espera 3 segons i després es desactiva l'esdeveniment.

La resta de components utilitzats els veurem a mida que parlem dels objectes que s'han anat creant a mida que el projecte anava prenent forma, a més també tenim altres aspectes no contemplats com per exemple els materials o les partícules que els veurem en els apartats posteriors on han esdevingut importants únicament en certs aspectes i hem considerat que no tenien cabuda a l'apartat inicial per ser menys bàsics que els contemplats aquí.

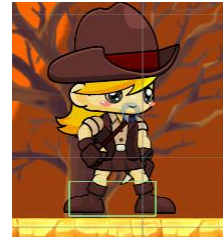
4.5 Assets

Durant el desenvolupament d'aquest joc no s'ha creat cap asset, tots provenen de diferents webs, a la bibliografia es pot veure d'on prové cadasc, són gratuïts i lliures de llicència.

5. Disseny del personatge principal i enemics

5.1 Personatge principal

El nostre personatge principal és una simpàtica aventurera que s'estavella amb l'avió com podem veure al menú principal, aquest personatge principal no és més que un objecte amb una sèrie de components que anirem desgranant a continuació:



- Sprite Renderer: component que conté la imatge del personatge.
- Animator: component que conté el control d'animacions (animator controller) on hi ficarem les animacions com si fos una màquina d'estats, aquest controlador té un estat inicial i cada animació és com un estat i per a moure'ns entre estats tenim unes transicions que podem activar mitjançant activadors, booleans o altres paràmetres dins el controlador.
- Rigidbody 2D: component que li dona les propietats físiques al personatge.
- Capsule Collider: component on controlem l'espai on rebrà impactes el personatge.
- Box Collider 2D: és el rectangle que tenim a la part inferior del personatge a la captura, ho tenim per a poder controlar quan el nostre personatge es troba amb els peus al terra i així permetre o no que pugui saltar.

5.2 Script del personatge principal

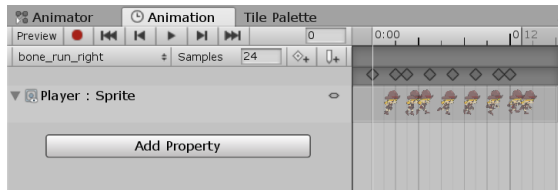
El nostre personatge principal només té un script amb el que controlarem les accions que pot realitzar el nostre mitjançant les següents accions les quals ficarem dins la funció update per a que es comprovin frame a frame:

- Córrer: mitjançant una funció Run creada es comprovarà primer si hi ha moviment horitzontal, és a dir, si prenem les tecles esquerra o dreta, en el moment en que es detecti el que fem és proporcionar velocitat al personatge en la direcció premuda, també controlarem la direcció per a posteriorment saber la direcció en la que dispararem, a més, en el moment de moure'ns també activarem un paràmetre de l'animador del personatge que activarà l'estat de córrer en una direcció concreta.
- Saltar: gràcies al boxcollider als peus del personatge sabem si podem saltar o no, per tant la nostra primera condició és si no estem en contacte amb el terra no saltarem (de ser cert és que estarem saltant i per tant mantindrem l'animació de salt activada) així que si estem al terra (on tenim l'estat de córrer en fals a l'animador) i comprovem que es prem el botó de salt -espai en la nostra configuració- llavors se li aplica una velocitat vertical al personatge principal i se li activa l'estat de salt a l'animador.

- Rebre un cop: a aquesta funció (hit) el primer que fem és comprovar que el collider del cos (no el dels peus) entri en contacte amb bé la capa enemig o la capa perills i que es trobi en un estat on pugui rebre dany, ja que mentre s'acaba de rebre un cop fa menys de 3 segons el personatge és invulnerable a rebre'n mentre, en cas contrari cridem a una funció de rebre cop a l'objecte que controla la sessió de joc on se'ns restarà una porció de vida i comprovem si ens queda vida per a fer una de les possibles dues accions:
 - Activar l'acció de rebre un cop si encara ens queda vida, on es crida la corutina de rebre cop on no només restarem sinó que farem al personatge moure'l cap amunt degut a l'impacte, aplicar un so d'impacte i fer aparèixer i desaparèixer el personatge per a simular un parpadeig durant 3 segons on el personatge serà invulnerable i tindrà desactivat l'estat de rebre un cop.
 - Activar l'acció de morir en cas de que no quedin més porcions de vida, on ens ficarem a l'estat de mort a la màquina d'estats de l'animador, alentirem el temps amb Time.timeScale (paràmetre on si és a 1 el joc va a velocitat normal però si el posem inferior el joc va més lent) i cridarem a una funció del joc on se'ns restarà una vida i el joc tornarà a carregar el nivell (o pantalla de gameOver en cas de no quedar-ne vides.
- Girar el personatge: aquesta és una funció simple on s'apuntarà en una direcció al personatge mitjançant un càlcul matemàtic on es detecti que hi ha moviment i que aquest sigui en una direcció concreta, que llavors gira el personatge apuntant en aquella direcció.
- Disparar: aquesta funció si premem el botó Ctrl i estem en un estat de poder disparar (que és un estat determinat per estar al terra, tenir bales - que ho mirem a la sessió de joc- i no haver disparat fa menys de 3 segons abans controlat amb una corutina també) activarem l'acció de disparar a la màquina d'estats del controlador, disminuïm el comptador de bales, farem un so i crearem un nou objecte dels prefabricats prèviament anomenat projectile que és un simple objecte amb un script que es mou a dreta o esquerra segons la direcció en el moment del tret i si entra en contacte amb un enemic llavors activa la funció de morir de l'enemic.

A més de les funcions descrites s'utilitzen uns quants paràmetres que declarem a l'inici del script per a poder controlar factors com la velocitat de moviment, la força del salt, si es pot disparar, saltar o rebre un cop... ja sigui mitjançant booleans o números reals (float) o enters (int).

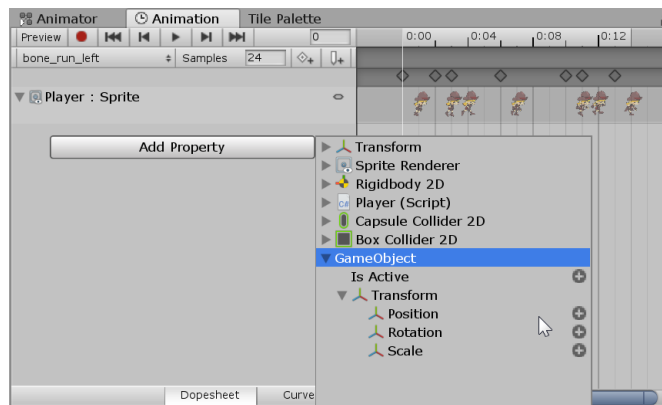
5.3 Animacions i animador del personatge principal



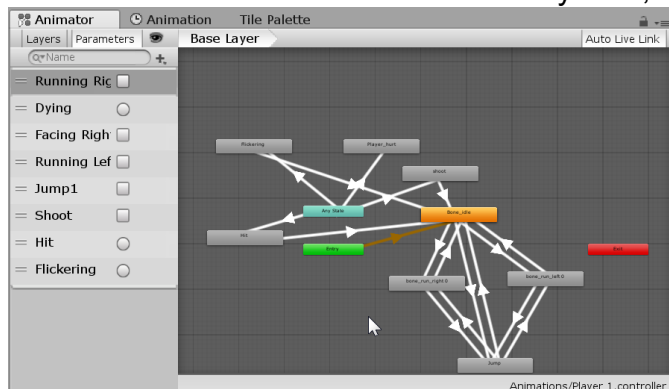
El nostre personatge principal disposa d'un conjunt d'animacions que s'han creat de forma molt senzilla, com podem veure a la captura a la part d'animation escollim el tipus d'animació (on posa bone_run_right si

cliquem veurem les que hi han ja creades per a l'objecte seleccionat o l'opció de crear una de nova). Nosaltres com ja tenim el conjunt d'animacions per separat de l'asset descarregat només afegim animació, fem les imatges que formin la animació completa i l'arrosseguem directament sobre la part dreta (les imatges s'han d'haver arrossegat prèviament al directori d'assets del projecte de Unity) i ens sortirà ja directament, aquí podem separar més cada frame o apropar-lo, donant-li al play podem visualitzar la animació al complet, normalment acostuma a sortir molt ràpida la animació així que baixem el número de samples per segon (per defecte està a 60 que per a alguns jocs està genial però el nostre personatge no està tan bé animat).

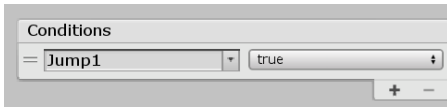
El menú d'animacions ens permet a més a més afegir qualsevol propietat a qualsevol frame, ja sigui al desplaçament (Transform), al propi sprite per a modificar el color, el tamany, etc... o qualsevol altre component que formi part del objecte, per tant si haguéssim volgut podríem haver establir propietats sense haver de passar-hi tant pels scripts, però bé, en el meu cas és més senzill l'entorn conegut i en aquest cas si es pot fer en C# allà es farà.



Tota animació com hem esmentat formarà part d'un controlador d'animacions, un animator controller exactament, en el nostre cas l'animador del jugador té uns quants estats ja que té en compte si corre cap a un costat o cap a un altre, si salta, si dispara, si rep un cop, si mor o si es troba en repòs, tots aquests estats tenen la seva animació pròpia i es relacionen les unes amb les altres mitjançant les fletxes que es diuen transicions. El controlador té dos estats bàsics: un és l'estat inicial i l'altre anystate, al primer si posem una transició directa a una animació aquella animació quedarà com a animació per defecte si no canviem cap estat, per tant l'animació de repòs lògicament (idle) és la que hi va, a partir d'aquesta animació es pot córrer i saltar i les transicions s'activen segons l'activació de



paràmetres, si per exemple apremem el botó de salt el script del jugador activa el booleà Jump1 (que és un paràmetre pertanyent a l'animador, no és cap paràmetre del script) i com podem veure a la imatge la condició per a que la transició de l'estat de repòs (bone_idle) a salt (jump) és que Jump1 passi a estat cert. En cas que sigui fals tornarà a l'estat iddle (a no ser que s'activi també la condició de córrer cap un costat o un altre on llavors ens anirem a l'estat de córrer).



Altres paràmetres com Flickering són els que activen el parpadeig del personatge quan rep un cop, el paràmetre Facing Right al principi l'utilitzavem per la càmera i per a la direcció en que disparem, però al final només s'utilitza per la direcció de les bales disparades ja que la càmera vam posar una intel·ligent que segueix i prediu la futura posició anticipant el moviment del jugador.

5.4 Altres aspectes destacables del personatge principal

Per saltar ens hem trobat amb problemes ja que el personatge s'arrossegava amb les parets si per exemple preiem el botó de salt i a la vegada ens desplaçàvem cap a la paret, per a evitar aquest arrossegament el que hem fet és crear un material sense fricció i enllaçar-lo als dos col·lidors del personatge, així qualsevol col·lisió amb un collider estarà lliure de friccions, però tot i així el problema persistia i vam haver de modificar el collider de tot l'escenari (ja que no eren parets planes i es continuaven generant friccions), la solució a aquets problema derivat el comentem a l'apartat d'escenaris.

5.5 Enemics

Sobre els enemics parlaré molt per damunt perquè principalment són personatges animats igual que el principal, amb dos col·lidors també (un per a detectar si poden continuar avançant i un altre de col·lisió que és el que pot entrar en contacte amb les bales o amb el personatge principal provocant-li mal).



A la captura podem veure els dos col·lidors, un que toca el terra i un altre que envolta al personatge, doncs bé, aquest objecte té un animador controller també però només amb dos estats o animacions pròpies: l'animació de moure's cap endavant i l'animació de morir en cas de rebre un impacte de bala.

Referents al script és molt més senzillet i se centra en moure el personatge mentre estigui viu (un estat que canvia en el moment de rebre un tret), cap a una direcció que canviarà segons el que indiqui la funció OnTriggerExit2D que controla el moment en que sortim de la capa d'escenari (el que ens indicarà que o bé hem arribat a la punta del terra o estem a punt de travessar una paret, per tant, el que farem és modificar el signe de la velocitat de l'eix X per a canviar de direcció el moviment del personatge.

A més d'això la funció de morir no s'utilitza directament aquí sinó que com hem vist abans la crida el objecte projectil al impactar amb el collider de l'enemic, no

ho hem dit abans amb el personatge protagonista però destruïm l'objecte un cop ha completat la seva animació mitjançant una corutina que esperi a la finalització d'aquesta.

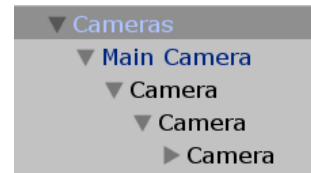
Tenim també un segon enemic que té les mateixes accions que l'altre enemic però amb unes particularitats, de primeres l'altre gira quan deixa de detectar terra, aquest en canvi gira quan el seu collider (que va davant d'ell) es troba amb algun objecte amb la capa "Ground", que és qualsevol part del nivell considerada interactiva. També encara que no es nota gaire quan el personatge mor es canvia el seu comportament físic i passa de ser un rigidbody de tipus kinematic a dinamic, per a que li afecti així la gravetat al morir i vagi baixant (tot i que ho fa força lent).



Sobre els enemics tot i tenir disponibles animacions d'atac i haver-ho testejat feien al joc més difícil en algunes parts amb la lava pujant i els enemics atacant, així que com ja hem justificat algun cop els enemics d'aquest joc són poc més un obstacle que altra cosa i s'han acabat traient, ja que sinó ajustar la lava segons les zones on siguin els enemics voladors (que tenen sentit només als espais de desplaçament vertical) li treia importància a la lava si la deixàvem més lenta.

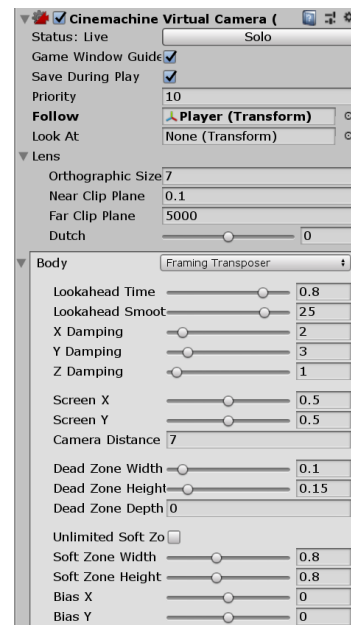
6. Disseny del nivell principal

L'escenari del nivell està compost per un objecte que hem anomenat Grid que conté tot els elements visuals de l'escenari com els diferents fons, els diferents obstacles, les plataformes del nivell, la lava, les col·lisions o colliders per a cada plataforma o alguns ficats per a pujar més ràpid el nivell de lava, però començarem parlant d'un personatge clau en qualsevol joc que sempre passa desapercebut: la càmera.



6.1 Càmera del nivell

La càmera ha estat un dels majors maldecaps del joc, aconseguir que el personatge seguís al personatge protagonista ha estat fàcil mitjançant un script al principi, però anticipar-se i moure la càmera amb anticipació no acabava d'encaixar, per sort Unity disposa d'un bon grapat de paquets que afegeixen contingut al programa, en el nostre cas vam utilitzar el paquet cinemachine que afegeix varies classes per a crear càmeres de diferents tipus: càmeres que segueixen, configuració de punts morts on no es mou la càmera si el personatge no surt del punt mort central del·limitat, capacitat per a moure's segons cap a on es mourà el personatge, velocitat de la càmera i de transició entre càmeres, etc... en el nostre cas teníem varies càmeres una per cada direcció de desplaçament, una altra per als salts però les transicions acabaven resultant problemàtiques així que al final s'ha escollit una càmera predictiva única que funciona prou bé.



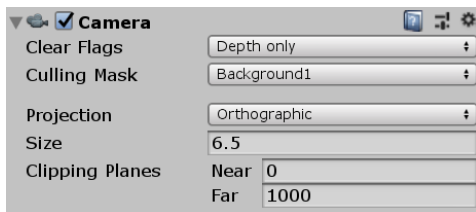
6.2 Fons amb efecte Parallax

L'escenari volíem que donés sensació de profunditat però el ser un joc en dos dimensions això esdevé impossible realment, però per sort podem enganyar una mica al nostre cervell i vista amb trucs com l'efecte Parallax: un efecte que consisteix en controlar els plànols de fons i fer-los moure a diferents velocitats, per exemple si tenim un escenari amb varies capes d'arbres i ens desplaçem a la dreta fem que el arbres més propers es moguin a una velocitat però els de



més darrera vagin més lents, això ens donarà una sensació de que els del fons es troben més a lluny ja que al moure's més lents vol dir que no ens hem desplaçat tant en relació amb els punts llunyans i això passa a la vida real, per tant és fàcil d'entendre i molt efectiu

sempre que es configuri bé, en el nostre cas el que hem fet ha sigut introduir diverses càmeres dins la càmera principal i que només es veïés una part de l'escenari en cadascuna d'aquestes càmeres, hem fet que cada càmera contingui un zoom diferent i problema resolt. Com es pot veure a la imatge cada requadre és una càmera i cadascuna d'elles només visualitza un element del fons i té una mida de visió (o el que diríem de forma més simple el zoom una mica diferent, per exemple el primer fons el veiem amb una mida de 6.5, la càmera principal a 7 (la de més amplitud) i la resta més petites encara, al tenir una amplitud diferents els fons es mouen a una velocitat diferent i això ja dona una bona sensació de profunditat.

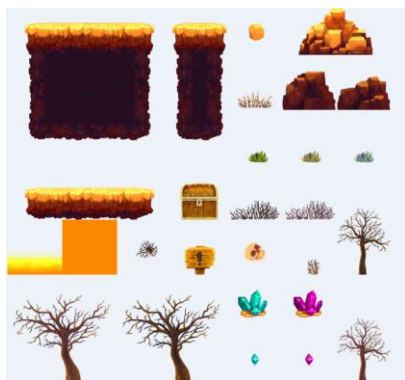


6.3 Objectes del nivell: Grid

Primer de tot tenim la quadrícula (Grid) on ficarem tots els elements que formin part de l'escenari a excepció d'enemics, càmeres, el jugador, la sessió de joc i elements específics com els col·leccionables, la fi del nivell, el checkpoint o certs esdeveniments.

La quadrícula està formada pels següents objectes:

- Background: aquí tenim un únic objecte amb el fons a un color entre vermell i groc que li hem afegit una animació on el color es va modificant per a simular una mica la llum emesa per la lava.
- Lava: objecte que conté un script (que fa pujar la lava a certa velocitat i que es para si posem en pausa el joc i s'accelera o desaccelera si cridem a alguna de les funcions de que disposa) i dos objectes amb un component anomenat Mesh Renderer que l'he trobat a un paquet per a simular banderes però que modificant una mica la disposició crec que s'ha aconseguit un bon efecte d'ona per a simular la Lava.
- Actual Ground: bàsicament és un mapa de tiles on tenim totes les plataformes dibuixades del nostre joc, s'han utilitzat els següents assets per a omplir l'escenari:



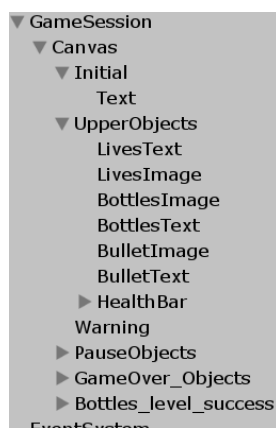
Podem veure que tenim per a construir roques, un tile per al color de la lava, un cartellet que utilitzarem de checkpoint, herba, roques i arbres per a donar-li una mica de vida (no massa) a l'entorn volcànic que tenim com a objectiu.

- Hazards: bàsicament les punxes del propi mapa de tiles, tot objecte pertanyent a la capa de Hazard fa sempre dany al personatge protagonista, no tenen cap script ni tenen res més destacable.
- Scenery: Un altre mapa de tiles on s'han posat tots els arbres, roques i resta d'elements de l'escenari no interactius.
- AcceleratorLava i DeceleratorLava: són bàsicament dos objectes amb un collider cadascú situat a punts estratègics de l'escenari que mitjançant el script propi quan el nostre personatge arriba al collider el script acaba cridant l'acceleració o desceleració de la lava al script de l'objecte Lava.
- Parallax: Aquí tenim dos o tres plànols de fons per cada altura del nivell separats i un plànol per davant del tot semi-transparent, classificats per capes per a que les càmeres produeixin l'efecte parallax.
- Order_layer_4: aquest objecte és una capa per a tapar certs objectes amb l'ordre de visualització 4, s'utilitza per un parell de canonades que no sobresurtin, res més.
- Colliders: degut a que els tiles tenen moltes irregularitats això provocava que les col·lisions al salt no fossin adequades, arrossegant el personatge més d'un cop, així que es van desactivar totes i s'han fet els colliders necessaris per a cobrir totes les plataformes i parets manualment, millorant molt el control a les plataformes del personatge principal, ja que ens estalviem una bona pila de problemes de col·lisions, en un joc complet això no seria viable fer totes les col·lisions manuals, ja que ha estat una feinada.

6.4 Objectes del nivell: GameSession

Aquest objecte controla gairebé totes les variables del joc i tota la interfície visible i no visible, és el que dona sentit al nivell.

Per tant com es pot veure tot els objectes d'interfície (UI) es troben dins del Canvas, tenim en total tots aquests objectes:



- **Initial:** missatge inicial dient que escapis de la lava pujant i que intentis aconseguir almenys 5 flascos, aquest objecte a més conté una animació que fa desaparèixer lentament el missatge, ja que no té sentit tenir el missatge a tot el nivell, només al principi i poc més. Igualment el propi script de GameSession desactiva el missatge als 3 segons.
- **UpperObjects:** conté tota la interfície indicativa del nivell, on veiem la nostra barra de vida (mitjançant un Slider al que li apliquem el valor del paràmetre

que tenim com la vida del personatge dins el script de `GameSession`), també veiem el número de vides totals i a la part superior dreta veiem el número de bales que tenim i els flascons recollits. Els flascons en cert moment del joc activen un mecanisme que fa baixar l'aigua, això ja ho veurem en l'objecte on es troba aquest mecanisme més endavant.

- `Warning` és el missatge a la zona de sortida d'aigua en cas de que arribem i no tinguem els 5 flascons mínims, sortirà només en el moment necessari mentrestant romandrà desactivat.
- `PauseObjects` és un objecte que conté tota la interfície del menú pausa amb els botons d'opció disponibles, va desactivat per defecte a `GameSession` i s'activa en cas de prémer la lletra p, també conté un script que s'activa al detectar la p polsada.
- `GameOver_Objects` conté un menú similar al de `PauseObjects` però s'activa quan morim i no ens queden vides, activa una música trista (desactiva prèviament la música del nivell) i tot per a dramatitzar la nostra mort, aquesta no té script i es porta tot des de `GameSession`.
- `Bottles_level_success` en canvi és el menú que es mostra en el nivell de victòria on es veu el número de col·leccionables recollits.

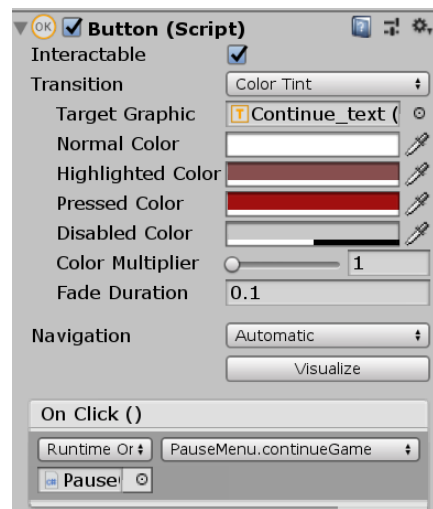
A més de tot això `GameSession` consta d'un component que no havíem vist fins ara: `AudioSource`, que és on afegim la música del joc, en altres scripts on activem un so l'únic que fem és cridar la funció `PlayClipAtPoint` i indiquem el so a reproduir i la posició, amb l'`AudioSource` podem també configurar el volum i gràcies a això veurem com a les opcions tenim una opció de volum de la música, a més recordar que cada cop que morim es para i es torna a encendre i la música torna a començar.

El script de `GameSession` controla un bon grapat de variables: els flascons, la vida, les bales, si s'ha arribat al checkpoint i tot els menús, a la funció `Start` d'inici es desactiva tota interfície excepte la inicial i els indicadors de les cantonades (vida, bales, etc...), però crida la corutina que desactiva la interfície inicial als 3 segons, a més inicialitza les porcions de vida al slider i les vides depenent dels paràmetres escollits a opcions (segons la dificultat tindrem més o menys vides), a la funció `Update` només controlarem si ens trobem a la pantalla de victòria per a activar la interfície que mostra el recompte de col·leccionables al mig de la pantalla.

`GameSession` també controla com fan altres scripts mitjançant una funció `Awake` que no es destrueixi pel canvi d'escena l'objecte, en canvi si tornem al menú principal o reiniciem nivell la funció sí el destrueix, ja que sinó mantindrà les vides de la partida anterior i la porció de salut que es tingués, a més de no tenir cap sentit mantenir l'objecte al menú.

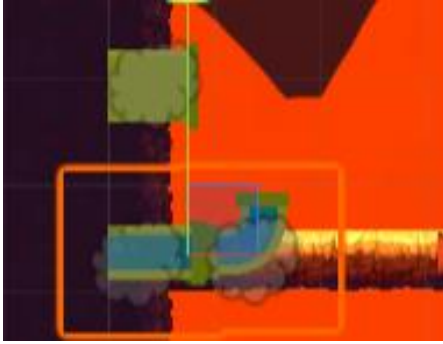
A més tenim una sèrie de funcions on activem unes pantalles o altres, a més que poden ser cridades per altres scripts:

- Tenim una sèrie de funcions per a controlar la pèrdua de vides: de primeres restem un al comptador de vides i comprovem si s'ha arribat al checkpoint, en cas afirmatiu carreguem el nivell des del checkpoint (que és una altra escena exacta a la del nivell però amb el personatge i la lava al mig del nivell, conservant els objectes persistents) sinó carreguem l'inici de nivell, però a més d'això sempre reiniciem les porcions de vida del slider i actualitzem la interfície que ens mostra les vides, a més de normalitzar la velocitat del joc de nou.
- Tenim funcions per a incrementar els comptadors de flascons i bales, un altre per restar 5 flascons i un per disminuir el de bales cada cop que desaparem, simplement sumen o resten i actualitzen la UI.
- Tenim una funció que en cas de morir i no tenir més vides crida al menú game over, una altra que en el cas de detectar la pulsació de la tecla p com hem dit abans s'atura el joc i s'activa el menú de pausa, aquests dos menús disposen de botons que al ser seleccionats activen altres funcions (l'enllaç entre pulsació i crida a la funció es fa des del propi Unity com es veu en aquesta captura amb el component Button). Aquests botons es poden configurar el color segons si passem per damunt el ratolí o si el prenem, i a la part de On Click configurem el que passarà al clicar, en aquest cas es cridarà a la funció de continuar el joc al script de menú de pausa.
- La resta de funcions són molt simples, algunes aturen la música, una altra comprova si hi ha bales per saber si pot disparar i d'altres retornen diferents paràmetres per a consultes a altres scripts. Hi ha també que retornen a l'escena del menú principal o a l'escena de la pantalla de victòria (que es crida des del script que forma part de l'objecte End_Level).



6.5 Resta d'objectes del nivell

ScenePersist: En aquest objecte englobem tots els objectes que el personatge pugui recollir i no volem que tornin a sortir si el personatge mor, de primeres utilitzem Awake al script per a mantenir l'objecte encara que carreguem el nivell un altre cop o el següent del checkpoint, el script no fa res més, només es manté durant tota la partida per tant tot objecte que estigui dins aquest objecte si es destrueix o qualsevol cosa es mantindrà igual la resta de partida. Dins l'objecte distingim dos tipus:



- Flascons: són el col·leccionable del joc i a la vegada serveixen en un punt concret per a activar un mecanisme que abaixi el nivell, cada cop que entrem en contacte amb un amb la funció OnTriggerEnter2D incrementem el comptador de flascons cridant a la funció que ho incrementa a GameSession, reproduïm un so i destruïm l'objecte i com es troba a dins de l'objecte ScenePersist llavors cada cop que morim no torna a aparèixer l'objecte tot i carregar-se el nivell.

- Bales: són la munició de la pistola, i al igual que amb els flascons fem el mateix però incrementem les bales, la resta és igual: reproduïm un so relacionat i ens carreguem l'objecte.

Estalactites: Aquest objecte consta de varis subobjectes:

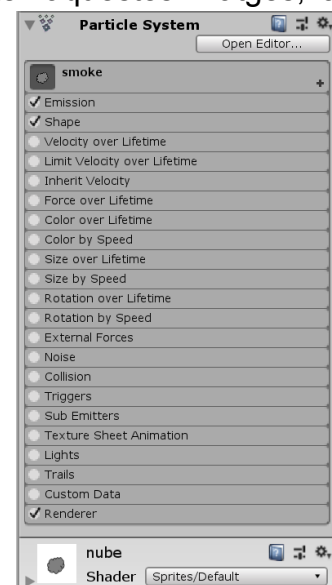
Varis objectes Activation que el que fan és mitjançant un collider en detectar el personatge converteixen el rigidbody de l'objecte de estàtic a dinàmic mitjançant un script propi i llavors per la força de la gravetat comença a caure.

- o Cada objecte Activation conté un objecte stalactite que és la pròpia estalactica amb el seu sprite i un script que detecta quan entra en contacte amb el terra per activar una animació de desaparició, simplement el color alpha comença a disminuir fins fer-se invisible, això en poc més de mig segon, temps que triga en destruir-se mitjançant una corutina.

Activation_stuff: Aquest objecte també conté varis subobjectes i un aspecte nou de Unity que encara no hem vist: els efectes de partícules.

Els **efectes de partícules** són exactament el que imaginem, una successió d'imatges on controlem la direcció, el volum en que surten aquestes imatges, la seva mida, la seva forma d'emissió... el que triguen a desaparèixer, la direcció, si els hi pot afectar o no la gravetat...

En el nostre cas només hem modificat aspectes bàsics i hem modificat la forma original per un núvol de fum, realment he agafat la imatge de paint d'afegir una vinyeta i he retallat la punta i ha quedat com un núvol, força pràctic. Podem veure que hem tocat l'apartat d'emissió fent que s'emetés en un temps concret, amb molt poca vida cada emissió però emetent un munt de cop. L'apartat shape ens ha permès configurar la forma d'emissió (sphere en el nostre cas) i Renderer ens ha permès modificar la forma inicial emesa pels núvols comentats anteriorment.



També hi ha un altre efecte de partícules per a simular l'aigua on s'ha modificat la forma d'emissió en con, un sprite diferent i encara més partícules emeses de cop per a simular un torrent d'aigua.

Tots els objectes d'activation_stuff actuen en un ordre determinat al script: Al entrar en contacte amb el collider s'activa tot en el següent ordre si es compleixen dues condicions primer: que es detecti al jugador i que aquest tingui més de 5 flascons.

- Primer es destrueix el collider per a evitar l'activació dues vegades de l'event i que alhora de restar els flascons no ens faci la resta varies vegades.
- Es desactiva el text que indica els 5 flascons necessaris.
- S'activa el fum a les dues canonades.
- S'activa un so d'explosió.
- Es resten 5 flascons dels flascons recol·lectats.
- Es modifiquen els sprites de canonades per uns on es veu l'entrada de líquid (el fum continua per sobre per a dissimular la transició).
- So de sortida d'aigua potent.
- L'efecte de partícules de degoteig que hi ha previ s'atura i entra el potent, a la vegada desapareix la canonada per on surt l'aigua.
- Es crida al script Lava per a que baixi el nivell de lava.
- Desapareix el sprite de canonada plena de líquid.
- S'atura l'efecte de fum de la canonada.
- S'atura la sortida d'aigua.
- Es destrueix el collider que detecta tot això.

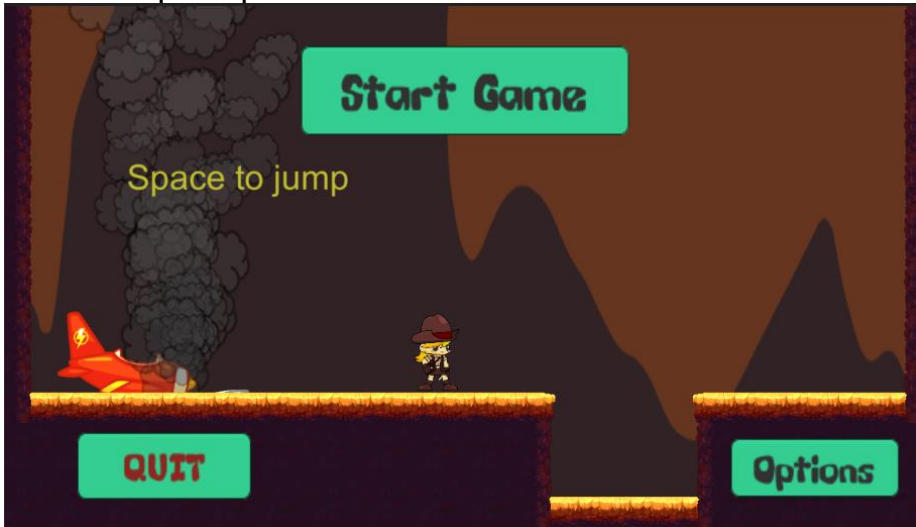
Totes aquestes operacions es fan en una corutina separades per intervals de temps d'un segon o mig segon per a quadrar els temps, la baixada, el temps que dura cada so i donar una sensació més o menys correcta o creïble del que ha passat.

End_Level: Aquest objecte és un sprite en forma de cova que simula la sortida del volcà, té un collider a la porta que arribant alenteix el joc i carrega el nivell de victòria, a més carrega una música coherent al moment de victòria (atura la música d'abans), durant els moments d'entrada a la porta el personatge esdevé immortal, això pot semblar una tonteria però em donava problemes arribar justet amb la lava ja que si era la teva última vida et plantava el gameover tot i haver-hi arribat, així que un cop s'arriba es considera que no pots morir encara que la lava et toqui en el moment immediatament posterior a arribar al collider.

Checkpoint: En el nostre cas hem utilitzat el tile amb un cartell com a checkpoint (la imatge del tile com a objecte en concret), l'hem animat una mica i emet un so quan passem per la seva posició (encara que passem per damunt seu) i es difumina fins desaparèixer emetent una llum verda, llavors el script crida al script GameSession per a que quedi constància que s'ha arribat al checkpoint i en cas de morir el joc es carregarà des de l'escena de checkpoint, que com ja havíem comentat és el mateix nivell però amb la posició inicial del personatge al punt de checkpoint i la lava a un punt proper.

7. Altres escenes i interfície de joc

7.1 Menú principal



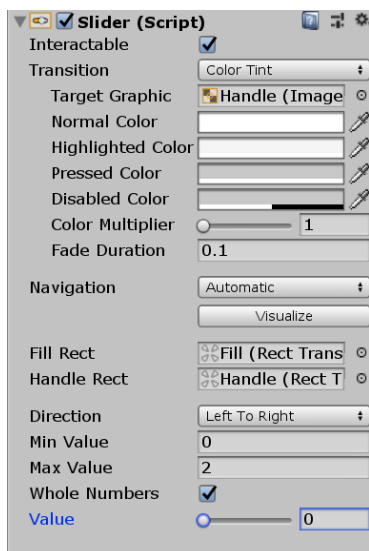
El menú principal a part dels botons que ja hem comentat en apartats anteriors destaca per poder controlar al personatge principal (tot i que no té habilitada l'habilitat de disparar), per tenir l'avió estavellat amb molt de fum sortint del motor i pels missatges que expliquen els controls del joc.

- El personatge principal no és més que una còpia del mateix personatge del nivell de joc, així que poc a explicar, però el pots moure i saltar pel menú, ja que el controles amb les fletxes i l'espai però selecciones les opcions amb el ratolí.
- L'avió estavellat és bàsicament un asset d'un avió agafat d'Internet que tenia les peces separades per a poder muntar animacions per trossos, en el nostre cas vam col·locar les parts que vam trobar necessàries i vam posar un efecte de partícules de fum similar al del nivell però aquest cop amb un sprite en forma de núvol modificant el color.
- Els missatges de control contenen una sèrie d'animacions que s'activen mitjançant corutines que comencen i s'acaben cada 3 segons, es veu un text desplaçant-se d'esquerra a dreta, evitant així carregar massa el menú principal de text.

7.2 Menú d'opcions

El menú d'opcions és un menú molt similar al menú principal però s'ha tret el personatge i hi ha dos sliders amb les opcions de dificultat i volum de la música.

La elecció de la dificultat suposa modificar uns pocs aspectes del joc: de primeres el número de vides es redueix, tens 3 vides a dificultat fàcil, 2 a mig i 1 a difícil (la salut no es veu modificada), també canvia el nombre de bales al nostre carregador inicial, comencem amb 5 en fàcil i tenim 4 en normal i 3 en difícil, això ho calculem com es pot veure restant al valor estàndard que tenim configurat el valor de la dificultat al script de GameSession, o sigui, si la



dificultat és 2 les vides seran 3-2 vides = 1 vida, GameSession només accedirà al script PlayerPrefs (que és el script que tenim al menú d'opcions per a emmagatzemar les configuracions o retornar els valors per defecte) que té funcions per a retornar els valors de dificultat i volum.

A més a més la velocitat de pujada de la Lava també varia, a més dificultat més ràpid puja, no gaire però just per a que no puguis perdre temps.

Per al volum de la música el mateix que hem especificat a dalt, però en el moment d'inicialitzar la música establim el volum.

Un altre aspecte a destacar és que al clicar default les opcions tornen a la seva posició original (el volum al mig i la dificultat en fàcil), això es fa mantenint dos paràmetres amb els valors per defecte i posant-los cada cop que cliquem.

7.3 Pantalla de victòria

La pantalla de victòria conté un fons nou on es veu el volcà des del fons, una interfície amb la paraula d'enhorabona on ens mostra el recompte de col·leccionables i ens permet tornar al menú. Aquesta pantalla no té cap particularitat en comparació amb les altres, per tant no destacaré res, a la llista de figures tenim una captura de la escena, no es veu la puntuació perquè està presa des del propi Unity fora de partida, però aquesta es mostra sota les lletres grogues.

8. Manual d'usuari

El joc és un simple plataformes en dues dimensions, els controls són ben simples:

- És recomanable escollir la resolució 1280x720 ja que els menús estan adaptats per aquesta mida.
- Utilitzem les fletxes de direcció per a moure'ns a esquerra o dreta.
- Si premem el botó ctrl disparem la pistola.
- Si premem el botó d'espai saltem.
- Si premem el botó p es pausa el joc.

8.1 Possibles interaccions

- Si entrem en contacte amb un enemic ens restarà una porció de vida, el mateix passarà amb les punxes, en canvi, si ens toca la lava no se'ns restarà cap porció sinó que perdrem una vida al complet.
- Si entrem en contacte amb flascons o amb bales els recollirem i s'afegiran al nostre comptador, aquests objectes no sortiran de nou si morim.
- Si arribem a un cartell amb exclamació (no cal el contacte, podem passar per la plataforma d'amunt) el joc ens activarà el checkpoint.

8.2 Elements del HUD



- 1- Número de vides i barra de vida sota (on indica que podem rebre dos cops abans de morir, quan es queda com es veu al següent cop morim, això no afecta a la lava que directament ens mata sinó als enemics i objectes punxants de l'entorn).
- 2- Número de bales disponibles (en arribar a 0 no podem disparar).
- 3- Número de flascons (en arribar a un punt concret es necessitaran un mínim de 5 per a poder avançar, ho hem vist a la pròpia escena del joc).

9. Conclusions

Tenint en compte el resultat i la perspectiva que tenia abans de començar el projecte crec que la falta d'experiència ha jugat un paper primordial, a la primera PAC vaig fer unes previsions sense cap tipus de coneixement, així que de l'únic que en puc estar orgullós és d'haver estat modest alhora de marcar-me unes fites perquè ja he anat molt just amb aspectes que no vaig valorar bé com la càmera o les col·lisions, però tot i així he de reconèixer que el curs de Unity fet al principi m'ha ajudat moltíssim ja que es fan varis jocs i aprens una mica de tot però sobretot veus plantejaments de problemes que mai t'imagines poder resoldre'ls tal i com es resolen, això sí, aquest curs es força llarg i jo el vaig començar abans de tenir el projecte, sinó s'hagués retardat més encara el projecte.

També ha après a gestionar una mica millor el temps ja que com tenia el full de fites quan em bloquejava en alguna cosa em posava amb altre i anava pensant en el problema previ i així no perdia tant de temps bloquejat, això no ho he pogut aplicar sempre però anar avançant en altres aspectes evita la frustració d'un problema, l'important ha sigut poder avançar, a les últimes setmanes vaig penjar a l'habitació on treballa les fites escrites en paper i anava tatxant les que anava fent i tenia una visió actualitzada i evitava centrar-me només en el problema del moment per a no perdre perspectiva, dormir bé ha estat molt productiu per a les solucions, ja que molts cops durant el dia no tenia cap idea i a l'aixecar-me tenia la solució moltes vegades.

Sobre aspectes que no han tingut cabuda per sort no han estat gaires, una versió mòbil la tenia plantejada i al final només hi ha la versió per Windows, pensava posar un segon idioma però tampoc m'ha donat temps i s'ha quedat només en anglès tot i que vaig mirar com fer un menú multi-idioma, però el temps és el temps, m'hagués agradat ficar algun enemic més per a afegir varietat, però almenys el nivell és prou jugable i és un repte intentar finalitzar el nivell, així que estic força satisfet.

M'hagués agradat poder utilitzar també diferents càmeres i no haver-me de conformar amb la càmera intel·ligent afegida al final, que va molt bé però vaig estar moltes hores intentant aconseguir que les transicions entre les diferents càmeres fos suau i tot i així no vaig aconseguir evitar les estrebades de càmera en certs punts. M'agradaria aprofundir força més en les càmeres.

També com a crítica personal m'hagués haver tingut algun tipus de formació artística per a poder crear assets propis, perquè per molt que es busqui mai trobes el que tens al teu cap i també es perd molt temps intentant aconseguir assets que entre ells quedin bé.

De cada al futur intentaré aprendre a fer assets, tinc un parell de cursos de gimp que m'he de mirar per a començar i sobretot no perdre pràctica amb Unity ara que he après, volia fer un videojoc pels motius exposats al principi però també volia saber si m'agradaria realment desenvolupar i tot i ésser una feinada m'ha acabat agradant força, així que em plantejo en un futur intentar

dedicar-me al desenvolupament de videojocs, a més, es podria dir que aquest projecte serà el primer ítem del meu futur portfolio.

Una sort que sí he tingut és d'haver completat el curs de la pàgina web d'Udemy de Unity que el vaig pagar fa un temps abans del projecte però l'objectiu era el d'aprendre Unity per a poder fer videojocs, per això no he fet cap tipus de fulla de costos del projecte perquè va ser una decisió personal meva anterior, ja disposava del curs, però sincerament amb el que he après si hagués de tornar a fer un projecte d'alguna eina que desconegués buscaria algun curs d'Udemy també i el pagaria amb tot el gust del món perquè ha estat meravellós: molt didàctic, entretingut i sobretot enfocat a aprendre i fer les coses per un mateix.

Per últim agrair al consultor les seves idees i suggeriments amb cada entrega que m'han ajudat a portar el projecte per millor camí, la seva visió ha tingut molt pes en els canvis fets, ja que els he trobat sempre molt justificats, a vegades no s'han pogut complir però s'han redissenyat altres aspectes per a evitar les situacions problemàtiques descrites tot i així.

En resum i com a conclusió molt personal crec que fer un projecte de creació d'un videojoc ha suposat un esforç major que haver realitzat altre tipus de projecte únicament de programació com seria el més comú en un treball de fi de grau d'enginyeria informàtica i que en la situació personal on he sigut pare fa poc potser haver escollit un altre projecte més senzill hagués estat una millor decisió de cara a tenir més temps, però no puc dir que me'n penedeixi de la meva decisió, m'ha agradat molt la experiència i em puc acomiadar del grau amb un bon sabor de boca i l'orgull d'haver creat un producte (molt modest això sí) sobre el meu hobby preferit, el que tantes hores de diversió m'ha aportat i m'ha descobert persones i històries que m'han influenciat i m'han portat a aquest punt de la vida, on no només he aconseguit finalitzar uns estudis que fa anys vaig haver d'abandonar sinó en una situació familiar que no canviaria per res, que és a qui dedico aquest projecte: a la meva dona Marta qui m'ha donat forces durant tots aquests anys d'esforç i al meu fill Antei que m'ha descobert un nou món de felicitat i responsabilitat.

10. Glossari

- **Unity:** motor de videojocs multiplataforma creat per Unity Technologies, actualment gratuït i un dels més utilitzats al món, estrenat al 2005.
- **Unreal Engine:** motor de videojocs multiplataforma creat per Epic Games, estrenat amb el videojoc Unreal al 1998, és potser el motor més utilitzat i conegut en l'actualitat, es programa amb C++.
- **RPG Maker:** motor de videojocs orientat en un principi a jocs de rol clàssics en 2D únicament que s'ha anat transformant en un motor multi ús 2D per a diferents gèneres, desenvolupat per una companyia japonesa anomenada ASCII Corporation al 1988, al principi era un joc per a crear jocs de rol com els de l'època i va acabar mutant en un motor amb moltes més possibilitats.
- **Asset:** la traducció directa seria actiu, que és certa, però en el nostre context acostuma a ser qualsevol element ja siguin sprites, animacions o imatges creats per algú que la resta por aprofitar.
- **Sprite:** imatge visual amb estructura d'un mapa de bits.
- **Tilemap:** mapa de tiles.
- **Tileset o tile:** sprites preparats per a ser utilitzats a un tilemap, no tenen sentit fora d'aquest ús per tenir unes mides preparades a encaixar en quadrícula.
- **Frame:** equivalent a un fotograma, és la imatge generada en un moment concret o en una unitat de temps del processador.
- **Grid:** Component quadrícula on tota la resta d'objectes dins podran tenir tilemaps que es col·locaran en les posicions de les que disposi.
- **Videojoc indie:** és el nom que se li posa a jocs d'un estudi independent, normalment de baix pressupost o amb un únic creador moltes vegades.
- **Escena:** segons la pròpia definició de Unity una escena és un nivell únic que conté els entorns i menús del joc pertanyents a aquell nivell.
- **Objecte (o GameObject):** és el concepte més important de l'editor de Unity, tot el que hi sigui a l'entorn de l'escena (ja sigui UI, sigui música, sigui un sprite, un tilemap, etc...) és un objecte a Unity, al que li afegim components per a definir característiques pròpies.
- **Space Invaders:** joc de naus dels anys 80 on amb la teva nau has d'aniquilar tots els enemics del nivell, un clàssic que va fundar el gènere de jocs de naus pràcticament.

- **Plants vs zombies:** joc de PopCap Games del gènere defensa de territori molt similar als jocs de taula clàssics on mitjançant la gestió de recursos pots anar emplaçant personatges atacants o defensors, amb l'objectiu de defensar la posició d'enemics, en aquest cas zombis que volen menjar el cervell del veí, com a particularitat les defenses funcionen per sí soles, no les controles directament, només encarregant-nos de la gestió.
- **Ori & the Blind Forest:** videojoc indie de no tan baix pressupost (ja que va acabar sent finançat per Microsoft) de plataformes i exploració en 2D amb un estil artístic i unes animacions molt cuidades.
- **GitHub:** Repositori on es pot pujar el codi de qualsevol desenvolupament mitjançant versionat, emmagatzemant com en una línia de temps els canvis de codi per a tenir una orientació espacial del nostre avenç.
- **Udemy:** pàgina web de formació bastant coneguda, ofereix un ampli catàleg de cursos.
- **Parallax:** efecte que simula un efecte de profunditat a partir de moure capes de fons pla a diferents velocitats cadascú.
- **Checkpoint:** aspecte dels nivells en el que el joc (en cas de morir) es reinicia en un punt més avançat que no pas l'inicial.
- **UI:** User Interface, és una característica que engloba els objectes que porten text o algun tipus per a la interfície d'usuari.
- **Tag:** Etiqueta que li posem a un objecte per a poder englobar amb posterioritat comportaments entre conjunts d'objectes.
- **Layer:** Capa on englobem a objectes, tot i semblar el mateix que els tags quan per exemple utilitzem el tileset aquest permet dibuixar tiles directament a una de les capes i no pas a tags.

11. Bibliografia

Tots els recursos han estat en línia, però primer de tot nombrar el curs d'Udemy que ha sigut bàsicament la font de gairebé tota consulta, un dels cursos de més èxit de la pàgina amb 35 hores de contingut en vídeo, un cop completat ha estat fins i tot senzill l'avenç al llarg del projecte perquè tenia la seguretat de poder fer gairebé qualsevol cosa que em plantegés a petita escala.

Ben Tristem, Rick Davidson. (última actualització abril 2019) Complete C# Unity Developer 2D: learn to Code Making Games [en línia] Udemy, Inc <<https://www.udemy.com/unitycourse/>>

11.1 Assets

Kenney.(2010-2019) Generic Items. [en línia] Kenney Assets [última consulta maig 2019] <<https://kenney.nl/assets/generic-items>>

Lucid Arts. (Abril 2016) Free Volcano Platform Tileset. [en línia] OpenGameArt.org [última consulta abril 2019] <<https://opengameart.org/content/free-volcano-platform-tileset>>

Adventurer Girl – Free Sprites. (2014) [en línia] GameArt2D. [última consulta maig 2019] <<https://www.gameart2d.com/adventurer-girl---free-sprites.html#>>

Freepng.es (2016) [en línia] <<https://www.freepng.es/png-0axjy9/>>

bevouliin.com (Març 2015) Free ingame items – Spike Monsters. [en línia] OpenGameArt.org [última consulta Abril 2019] <<https://opengameart.org/content/bevouliin-free-ingame-items-spike-monsters>>

PWL. (Octubre 2016) Seamless Cave in Parts. [en línia] OpenGameArt.org [última consulta abril 2019] <<https://opengameart.org/content/seamless-cave-in-parts>>

Phoenix1291 (Desembre 2017) SFX: The Ultimate 2017 8 bit mini pack [en línia] OpenGameArt.org [última consulta maig 2019] <<https://opengameart.org/content/sfx-the-ultimate-2017-8-bit-mini-pack>>

Spring (Setembre 2017) “Fire Level” [en línia] OpenGameArt.org [última consulta maig 2019] <<https://opengameart.org/content/fire-level>>

Tomasz Skowronski (2018) zai Lava [en línia] dafont.com [última consulta abril 2019] <<https://www.dafont.com/es/zai-lava.font>>

11.2 Consultes extra

Pere Martra (Gener 2018) Como crear un efecto Parallax en Unity, de una forma acojonantemente sencilla. [en línia] WordPress [última consulta abril 2019] <<https://martra.uadla.com/como-crear-un-efecto-parallax-en-unity-de-una-forma-acojonantemente-sencilla/>>

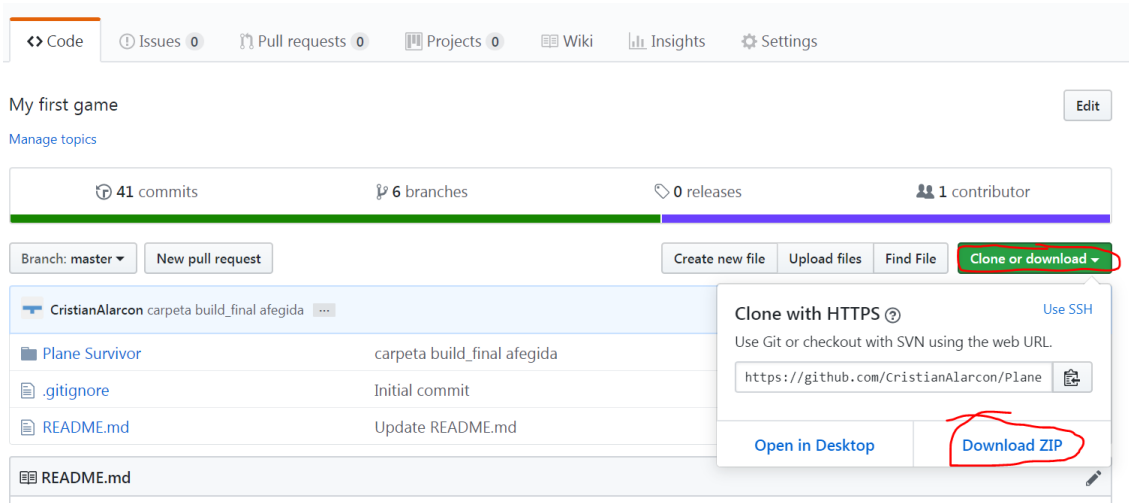
“Unity User Manual (2018.2)”. (Maig 2018) [en línia] Unity Technologies. [última consulta maig 2019] <<https://docs.unity3d.com/2018.2/Documentation/Manual/>>

12. Annexos

12.1 Com descarregar i executar el projecte

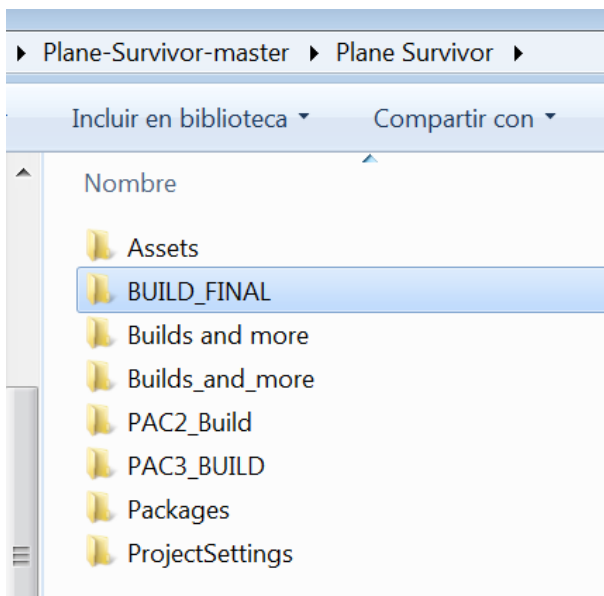
1.- Anar a <https://github.com/CristianAlarcon/Plane-Survivor>

2.- Cliquem a Clone or download i després a Download ZIP. Amb això ens descarregarem el projecte.

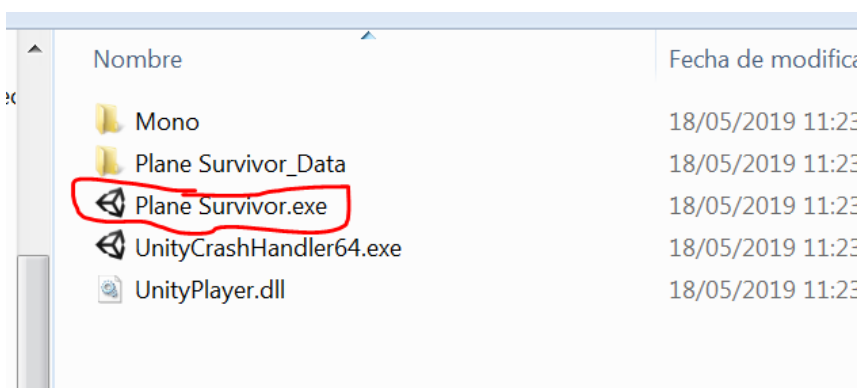


3.- Descomprimir el ZIP descarregat a qualsevol carpeta i obrim la carpeta.

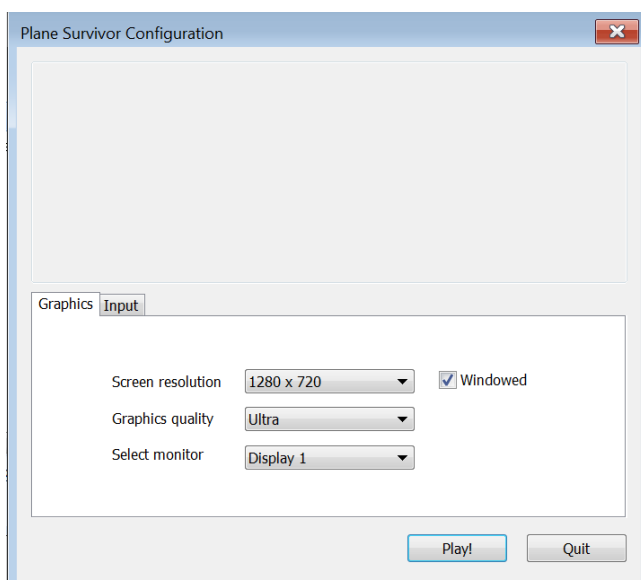
4.- Naveguem fins a Plane Survivor i entrem a BUILD_FINAL



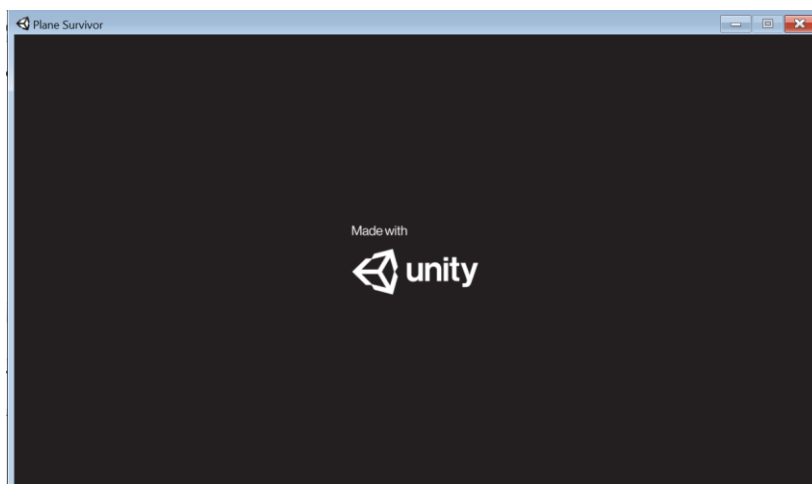
5.- Fem doble clic a Plane Survivor.exe



6.- Recomano escollir la resolució 1280 x 720 sobretot, és la resolució dels assets i com millor es veu. Cliqueu a Play i...



7.- A jugar!



12.2 Llistat d'errors solventats al llarg del desenvolupament.

- El collider de la porta final ens portava al final del nivell si moríem a prop i la lava ens arrossegava: aquest problema s'ha arreglat impedint que de cap forma s'arribi mort al collider, s'ha pujat una mica també per a evitar qualsevol tipus de contacte i es pausa el joc també al poc de morir impedint l'arrossegament del jugador.
- Contacte entre la bala i l'enemic inexistent, tot i que es detectava el contacte no s'activava la mort des del propi script de l'enemic, al final al script de la bala es crida el script de l'enemic amb el que es col·lisiona (s'agafen el de cada enemic i el que existeixi llavors crida al mètode de morir).
- L'animació de fer un tret s'ha ampliat una mica, ja que quedava molt brusca al principi, s'ha fet més lenta perquè sinó sembla que tregui la pistola John Wayne i l'amagui en menys d'un segon.
- L'animació del personatge segons el costat és una mica diferent, es sentia millor retocant els sprites intermedis així, això s'havia fet en principi per a poder utilitzar una càmera o l'altre segons la direcció però després ja no feia falta, però a última hora revisant una mica les animacions s'ha decidit deixar per aquest motiu.
- La direcció del tret també ha estat un dels motius per a mantenir la direcció del personatge en un booleà, ja que si ho sincronitzava amb el moviment la bala si girava el personatge canviava de direcció també.
- La física de la lava no segueix exactament les ones, si posava un collider amb la malla les físiques es tornaven boges així que realment el contacte de la lava és un entremig entre la ona més alta i la més baixa, no és perfecte però genera molts menys problemes.
- La càmera ja s'ha comentat al seu apartat, així que seré breu, hi ha hagut primer un conjunt de càmeres associades a cada animació, per a després treure's degut a que les transicions eren problemàtiques, llavors es va passar a una càmera mitjançant un script, però no quedava tan bé com la última implementada que és predictiva i s'avança una mica segons el moviment que estàs fent, en la meva opinió és la millor de les 3 implementades.
- Durant el menú de game over si premies la lletra P s'activava el menú de pausa al damunt, per a evitar això s'ha posat al script de PauseMenu una nova condició booleana prèvia a detectar la lletra p per activar-se, aquesta condició (pausable) ens permet controlar quan podem pausar i quan no, així que durant el menú de game over es posa en fals el booleà i així no permetem que es pugui posar el joc en pausa.

- El recompte de vides tenia un bug, ja que vaig estar fent proves un dia amb el nombre de vides i al principi es mostren les vides però després es mostraven les vides -1 per a que al joc es veiés un 0 al morir, no vaig pensar en posar la condició de mort quan les vides fossin 0 sinó 1 i fins que no me'n vaig assabentar no ho vaig solucionar.
- Durant la resta de flascons per l'activació de l'aigua es generava un problema i és que com és una funció que s'activava al detectar l'entrada del jugador al collider de l'objecte es repetia varies vegades provocant que sempre s'acabés a 0 el comptador de flascons, per a evitar-ho el que s'ha fet és eliminar el collider un cop activat l'aigua i llavors fer la resta de flascons.