



Universitat Oberta  
de Catalunya

# DISSENY I DESENVOLUPAMENT D'UN JOC EN 2D

**Jaume Martí Vallès**  
Grau Multimèdia  
Videojocs

**Helio Tejedor Navarro**  
**Jordi Duch Gavalrà**

18/01/2018



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

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Creació d'un joc 2D en Unity</i>
<b>Nom de l'autor:</b>	<i>Jaume Martí Vallès</i>
<b>Nom del consultor/a:</b>	<i>Helio Tejedor Navarro</i>
<b>Nom del PRA:</b>	<i>Jordi Duch Gavaldà</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>01/2018</i>
<b>Titulació o programa:</b>	<i>Grau Multimèdia</i>
<b>Àrea del Treball Final:</b>	<i>Videojocs</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<i>Acció, Sci-Fi, Unity</i>
<b>Resum del Treball (màxim 250 paraules):</b> <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>Aquest treball és un projecte de final de Grau Multimèdia de la Universitat Oberta de Catalunya. En aquesta memòria s'expliquen els procediments utilitzats per a dur a terme aquest treball.</p> <p>El nom del treball és Disseny i Desenvolupament d'un joc 2D en Unity, El treball tracta sobre el disseny i desenvolupament del joc, en aquest document trobem tota la documentació referent a les fases del projecte i els passos seguits per arribar al producte final, des de la implementació d'aquest com també les diferents etapes que s'han seguit, el procés de planificació, disseny i implementació, fins al raonament de les decisions preses durant el treball.</p> <p>Aquest document també conté les valoracions personals i tot tipus d'informació que pugui fer referència a la documentació i consecució del Videojoc.</p>	

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

This work is a project of final of Multimedia Degree of the Open University of Catalonia. This memory explains the procedures used to carry out this work.

The name of the work is Design and Development of a 2D game in Unity, The work treats on the design and development of the game. This document contains all the documentation referent to the phases of the project and the steps followed to arrive to the final product, starting with the implementation of this and also the different stages that followed, the process of planning, design and implementation, until the reasoning of the decisions made during the work.

This document also contains the personal estimations and any useful information related to the documentation and achievement of the Video game.

# Índex

1.	<b>Introducció</b> .....	3
1.1.	Context i Justificació .....	3
1.2.	Objectius.....	4
1.3.	Enfocament i mètode seguit .....	4
1.4.	Planificació del Treball .....	5
1.4.1.	Recursos.....	5
1.4.2.	Tasques .....	7
1.4.3.	Diagrama de Gantt i estimació de temps .....	8
2.	<b>Disseny del joc</b> .....	9
2.1.	<b>Especificacions del Projecte</b> .....	9
2.1.1.	Gènere i Tipus de joc.....	9
2.1.2.	Història del joc.....	10
2.1.3.	Objectiu del joc.....	10
2.1.4.	Target al que està dirigit.....	10
2.1.5.	Requeriments del joc .....	11
2.1.6.	Descripció del Funcionament i de la Mecànica del joc.....	12
2.1.7.	Controls del Joc .....	13
2.2.	<b>Interfície Gràfica del Joc</b> .....	14
2.2.1.	Menu Principal .....	14
2.2.2.	Menú de Selecció de Nau .....	15
2.2.3.	Pantalla de puntuació.....	15
2.2.4.	Pantalla de Final de Partida o GameOver.....	15
2.2.5.	Pantalla de Pausa.....	16
2.2.6.	Elements audiovisuals de l'interfície gràfica.....	16
2.2.7.	Taula amb els elements gràfics que conformen l'interfície.....	17
2.3.	<b>Elements durant la partida</b> .....	17
2.3.1.	Elements Estàtics durant la partida.....	18
2.3.2.	Background.....	18
2.3.3.	Health Bar o Barra de Vida .....	18
2.3.4.	Puntuació de la partida .....	19
2.3.5.	Taula amb els elements estàtics .....	19
2.4.	<b>Elements Dinàmics durant la partida</b> .....	20
2.5.	<b>Naus del jugador</b> .....	20
2.5.1.	Nau-1 .....	20
2.5.2.	Nau-2 .....	21
2.5.3.	Nau-3 .....	21
2.5.4.	Taula amb les característiques de les naus del jugador .....	21
2.6.	<b>Naus enemigues</b> .....	22
2.6.1.	Enemic 1 E-1.....	22
2.6.1.	Enemic2 mina E-2.....	22
2.6.2.	Enemic 3 E-3.....	23
2.6.3.	Enemic 4 Sniper E-4 .....	23
2.6.4.	Enemic 5 Mina cluster E-5 .....	23
2.6.5.	Enemic 6 (4 projectils) E-6 .....	23
2.6.6.	Enemic 7 (8 projectils) E-7 .....	23
2.6.7.	Enemic Roca.....	24
2.6.8.	Taula amb les característiques de les diferents naus enemigues.....	25
2.6.9.	Enemic Boss .....	26
2.7.	<b>Projectils</b> .....	27
2.7.1.	Projectils del jugador.....	27
2.7.2.	Projectils Enemics.....	28
2.7.3.	Taula amb els projectils enemics .....	28
2.8.	<b>Sistema d'augmentos o Power up's</b> .....	28
2.8.1.	Power up Foc.....	28
2.8.2.	Power up Shield.....	29
2.8.3.	Power up Vida.....	29
2.9.	<b>Disseny Audiovisual dels Elements del joc</b> .....	30
2.9.1.	Disseny gràfic.....	30
2.9.2.	Tipografies del joc .....	30
2.9.3.	Disseny de l'Audio.....	31

2.9.4.	Banda Sonora .....	31
2.9.5.	Efectes de So.....	31
<b>3.</b>	<b>Desenvolupament i Implementació .....</b>	<b>33</b>
3.1.	Introducció a Unity i el seu funcionament .....	33
3.2.	Estructura de fitxers del joc.....	34
3.3.	Scripts del joc .....	34
3.3.1.1.	Estructura de fitxers dels Scripts .....	35
3.4.	Implementació del codi .....	35
3.4.1.	Controladors.....	35
3.4.1.1.	ControladorJoc.....	35
3.4.1.2.	ControladorBackground .....	35
3.4.1.3.	DobleAudioSource .....	36
3.4.1.4.	ControladorSeleccióDeNau.....	36
3.4.1.5.	ScriptMostraPuntuacions .....	36
3.4.2.	Enemies .....	37
3.4.2.1.	ScriptEnemy .....	37
3.4.2.2.	ScriptEnemy4Disparos .....	40
3.4.2.1.	ScriptEnemy8Disparos .....	40
3.4.2.2.	ScriptEnemyMina .....	40
3.4.2.3.	ScriptEnemyClusterMina .....	41
3.4.2.4.	ScriptEnemyMovil.....	41
3.4.2.5.	ScriptEnemySniper.....	41
3.4.2.6.	ScriptFocEnemy .....	42
3.4.2.7.	ScriptFocEnemyNoVelocity .....	43
3.4.2.8.	ScriptMisilEnemy .....	43
3.4.2.9.	ScriptRoca.....	44
3.4.2.10.	ScriptSpawn .....	45
3.4.2.11.	ScriptBossBase .....	47
3.4.2.12.	BossComponentBase.....	47
3.4.2.13.	BossFaseBase .....	49
3.4.2.14.	ScriptFocNau.....	49
3.4.2.15.	ScriptNauBasic.....	49
3.4.2.16.	ScriptNauGran.....	52
3.4.2.17.	ScriptNauLaser.....	52
3.4.3.	Menus .....	52
3.4.4.	Objectes.....	53
3.4.4.1.	Ddol.....	53
3.4.4.2.	laserScript .....	53
3.4.4.3.	ScriptPowerUpBase .....	55
3.4.4.4.	ScriptPowerUpAtac,criptPowerUpShield, ScriptPowerUpVida.....	56
3.5.	Implementació d'efectes del joc.....	56
3.5.1.	Bump Mapping .....	57
3.5.2.	Animacions .....	58
3.5.3.	Sistema de Partícules .....	59
3.5.3.1.	Projectil nau jugador N1 i N2.....	59
3.5.3.2.	Efecte impacte projectil .....	60
3.5.3.3.	Explosió.....	60
3.5.3.4.	Propulsió de la Nau .....	61
3.5.4.	Altres Efectes .....	61
3.5.4.1.	Escut de les naus .....	61
3.5.4.2.	Raig laser .....	61
3.5.4.3.	Il·luminació .....	62
<b>4.</b>	<b>Millors del Joc.....</b>	<b>63</b>
<b>5.</b>	<b>Conclusions del Projecte.....</b>	<b>64</b>
<b>6.</b>	<b>Glossari .....</b>	<b>66</b>
<b>7.</b>	<b>Bibliografia .....</b>	<b>67</b>

# 1. Introducció

## 1.1. Context i Justificació

El punt de partida del treball es focalitza amb el gran consum que actualment els usuaris realitzen de les aplicacions per a dispositius mòbils, en concret els videojocs són les 'App' amb més consum que trobem en les stores de les diferents plataformes per a dispositius mòbils i del desig de conèixer el món del disseny i desenvolupament de videojocs.

Aquest és el meu primer videojoc que desenvolupo, ja tenia una bona base del joc en la qual treballar, així doncs la intenció d'aquest projecte és la d'obtenir un producte amb el qual jo hem sentís satisfet i el qual fos un joc al que jo jugaria. Com ja tenia una mica d'experiència he intentat anar una mica més lluny i aportar més funcionalitats i una millor experiència de joc.

L'objectiu del projecte és el d'aconseguir un producte depurat el qual es pogués jugar en diferents plataformes i compleixi les expectatives marcades a l'inici del projecte, un dels altres requeriments importants és que el joc tingués un bon nivell de jugabilitat i d'aspecte gràfic.

El motor de joc amb el qual s'ha desenvolupat el joc és Unity 5, el qual és multiplataforma i un dels motors més utilitzats actualment per a la creació i desenvolupament de jocs en plataformes mobile.

Pel que fa al llenguatge utilitzat és el C# en detriment del UnityScript el qual desapareixerà del motor Unity.

## 1.2. Objectius

Els objectius d'aquest projecte són:

- Desenvolupar un joc multiplataforma 2D en Unity.
- Iniciar i aprofundir en el disseny i desenvolupament de videojocs.
- Creació d'un art audiovisual espacial i futurístic.
- Implementar els recursos multimèdia treballats durant el grau.
- Complir amb els requisits que un TFG suposa.

## 1.3. Enfocament i mètode seguit

L'enfocament seguit en aquest projecte ha sigut el de millorar el producte el qual s'havia treballat anteriorment, per això s'han hagut de prendre diferents punts de vista i valorar quins eren els punts més fluixos del joc. Un cop decidits els punts més fluixos s'ha hagut de decidir a quins aspectes del joc es donaria prioritat, ja que el temps és limitat i l'objectiu és aconseguir un videojoc amb la qualitat desitjada.

S'ha decidit enfocar el desenvolupament del joc en millorar la jugabilitat i aportar elements distintius que dotessin de personalitat al joc, així doncs l'enfocament de partida ha marcat el mètode a seguir durant les diferents fases del projecte.

En la primera fase del projecte es va decidir el camí de ruta del projecte, i la idea del joc va prendre forma, es van marcar els objectius i es van planificar les tasques a dur a terme.

En la fase de desenvolupament i implementació s'han afegit les diferents funcionalitats del joc, aquestes fases són retroactives, ja que moltes d'aquestes implementacions s'han anat modificant a mesura que



s'afegien funcionalitats i elements al joc o aquestes requerien canvis o millores.

D'altra banda, durant el projecte s'ha hagut de realitzar un treball de documentació i investigació, cal dir que el motor de desenvolupament d'Unity 5 i el llenguatge C# estan molt ben documentats, factor que facilita molt la tasca de l'estudiant.

## 1.4. Planificació del Treball

### 1.4.1. Recursos

Aquests són els recursos multimèdia utilitzats per a la realització del projecte.

**Photoshop**<sup>[1]</sup> és un editor de gràfics rasteritzats creat per l'empresa Adobe Systems Incorporated llançat per primer cop l'any 1991 i ja des dels seus inicis ha sigut un estàndard per al retoc fotogràfic, però també és extensament utilitzat en altres camps com són el disseny web la 'fotocomposició', edició de gràfics i en qualsevol procés de tractament d'imatges digital.

En un començament el programa treballava en un espai 'bitmap' o mapa de bits/pixels i en una sola capa, en l'actualitat treballa amb múltiples capes i suporta una gran varietat de fitxers d'imatge diferents, apart de disposar d'una gran quantitat d'efectes i eines, cal dir que la interfície de treball es troba molt desenvolupada.

**Unity5**<sup>[2]</sup> és un motor de videojocs multiplataforma creat per Unity Technologies, aquesta va ser fundada l'any 2004 amb l'objectiu de crear un motor de joc que estés a l'abast de tothom i a un preu accessible. Unity suporta varis sistemes operatius com Microsoft Windows, OS X i Linux i s'ha convertit en una referència per als desenvolupadors

independents que no poden crear el seu propi motor de joc i no disposen de les eines necessàries per desenvolupar un joc.

Unity és utilitzat conjuntament en una gran varietat de programes com són editors de gràfics, editors de gràfics 3D, editors de gràfics vectorials entre d'altres i suporta varis tipus de mapejats diferents. El motor gràfic utilitza 'OpenGL', (per a Windows, Mac i Linux) 'Direct3D' (sols en Windows) i 'OpenGL Es' (per a Android i IOS).

Unity ofereix dos tipus de llicències, una professional que és de pagament i gaudeix de varies millores i la llicència gratuïta, que mostra un missatge de benvinguda en els jocs independents i una marca d'aigua en els jocs web, la qual no es pot ni modificar ni desactivar, també ofereixen llicències educatives.

**Audacity** és un editor d'àudio digital i gravació per ordinador de codi obert, multiplataforma suportant Windows, GNU/Linux, i Mac, inclou una gran quantitat d'idiomes, apart que el lloc web ofereix tutorials amb varis idiomes. Va ser publicat l'any 2000 com a software lliure, actualment és l'editor d'àudio més utilitzat en els sistemes GNU/Linux i gaudeix d'una gran popularitat.

Entre les seves principals característiques trobem la gravació d'àudio en temps real, suporta una gran varietat de fitxes d'àudio com són 'Ogg Vorbis', el qual utilitzarem en els fitxers d'àudio, MP3, WAV, AIFF, AU, LOF i WMP, permet la conversió entre formats, l'edició de pistes múltiples i una gran varietat d'efectes de so els quals ens permeten modificar i crear els sons a gust, també ofereix la possibilitat d'instal·lar 'plug-ins' per augmentar les funcionalitats.

### 1.4.2. Tasques

El desenvolupament d'un videojoc es divideix en diferents fases les quals acaben donant forma al producte final. Aquestes es poden dividir en diferents etapes però ho podem resumir en cinc diferents fases que són el disseny, el desenvolupament, la implementació, el testeig i la publicació del producte final, aquestes no és realitzen de manera seqüencial sinó que es van realitzant a mesura que es va desenvolupant el joc.

La fase de disseny del joc conté el procés creatiu mitjançant el qual es desenvolupa la idea del joc, es concreta la mecànica del videojoc i es concreta el seu aspecte gràfic, aquesta fase marcarà la planificació del projecte.

En la fase de Desenvolupament s'estudiaran els recursos multimèdia i es dissenyaran les solucions del joc seguint les pautes marcades en la fase de disseny, aquesta fase conté la recerca d'informació necessària per poder dur a terme el videojoc.

Durant la implementació es generaran els recursos multimèdia que conformen el joc i s'incorporaran al joc, és generaran els scrips els quals contenen la lògica mitjançant la qual el joc funciona.

El testeig és la fase en la qual es realitzen les diferents proves del joc com són el seu funcionament, és busquen possibles errors i és testeja que la jugabilitat i l'aspecte global del videojoc siguin els desitjats. Aquesta fase ens permetrà depurar el joc i ajustar la jugabilitat i l'experiència del jugador.

L'últim pas és la publicació del Videojoc, en aquesta fase el joc surt a la llum i és publicat en les diferents plataformes per a la seva descàrrega o

venda física, en aquesta fase es realitzen els manuals d'ajuda del joc i la documentació del projecte.

### 1.4.3. Diagrama de Gantt i estimació de temps

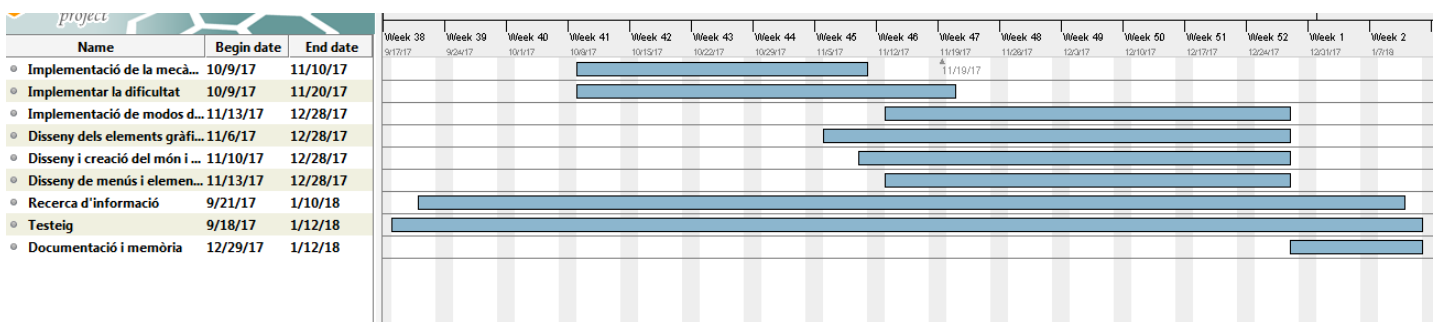
En la fase de disseny del joc és va realitzar un diagrama de Gantt és van definir les tasques i els seus temps aproximats necessaris per a realitzar cadascuna d'aquestes.

En el diagrama de Gantt podem veure la distribució d'aquestes tasques previstes durant el disseny i desenvolupament del projecte.

Les tasques previstes al diagrama de Gantt són:

- Disseny i creació del món i del joc
- Disseny dels elements gràfics
- Disseny dels menús i altres elements
- Implementació de la mecànica del joc
- Implementació de la dificultat
- Implementació de modes de joc o altres millores
- Recerca d'informació
- Testeig
- Documentació i memòria

Pel que fa al compte del temps no s'han comptabilitzat les hores però sens dubte s'ha superat l'estimació en varies de les fases previstes al començament del projecte, d'altra banda alguna de les tasques durant el projecte ha canviat, a causa de decisions que s'han pres durant el projecte.



## 2. Disseny del joc

### 2.1. Especificacions del Projecte

#### 2.1.1. Gènere i Tipus de joc

El joc és un background scroller en dues dimensions o 2D, ambientat amb la ciència ficció, aquests són més col·loquialment coneguts com space shooters i pertanyen al subgènere dels shooters, els quals pertanyen al gènere d'acció.

Dins del gènere dels shooters trobem molts subgèneres, aquests es classifiquen mitjançant diferents característiques del joc, com la perspectiva, el seu nivell de realisme, el nombre de personatges els quals poden jugar, si és multi jugador (online) i per últim la temàtica del joc, la majoria de shooters els podem classificar en dues categories en primera persona o FPS i en tercera persona.

Pel que fa als tipus de joc el qual és un space shooter o *Shoot 'em up*, trobem un gran nombre de referències d'aquesta temàtica de jocs ja que va ser de les primeres en aparèixer en els inicis dels videojocs, un dels primers jocs d'aquest gènere és *Spacewar* el qual es va publicar en màquines arcade al 1961. Tot i que durant el temps s'ha sabut renovar el gènere i ocupar el seu lloc al mercat en les diferents plataformes.

Algunes de les referències que trobem són *Space Invaders*, *Defender* els quals van fer créixer el gènere fins arribar a la seva màxima esplendor en la dècada dels 80 i 90, actualment aquest gènere cada cop es veu més apartat degut a què el públic demana altres gèneres, tot i que sempre ha estat present en el món dels videojocs.

### 2.1.2. Història del joc

Aquests jocs no solen tenir molta història, poden contar una petita història però normalment aquesta és secundària i té molt poca presència en el joc, ja que aquests jocs enganxen al jugador mitjançant altres mitjans com són una gran quantitat de acció en pantalla i la necessitat de destruir enemics sense parar.

Tot i això, el joc té una història que és més un objectiu, el jugador és un pilot solitari el qual es veu immers en una batalla per la supervivència del seu univers, que es veu envaït per les forces enemigues, el joc està ambientat en una batalla a l'espai en un futur llunyà, ets l'última esperança per derrotar les forces enemigues.

### 2.1.3. Objectiu del joc

Normalment en un joc l'objectiu d'aquest és finalitzar el joc o poder veure l'esperat final del joc, ja que aquest conta una història durant la partida o els diferents nivells de joc, però el tipus de joc que es desenvolupa en aquest projecte, a l'igual que el seu gènere no es caracteritzen per gaudir d'una gran història o guió, el seu objectiu radica en destruir el màxim nombre d'enemics o superar els diferents nivells de joc.

Aquest joc té un objectiu clar, aconseguir la màxima puntuació possible i superar els diferents Boss que apareixen durant la partida. Però sobretot l'objectiu principal és que el jugador gaudeixi l'estona en la qual juga al joc i que aquest tingui una gran quantitat d'acció la qual enganxi al possible target.

### 2.1.4. Target al que està dirigit

El joc està dirigit a tots els públics, ja que des d'un nen petit fins a una persona de la tercera edat poden jugar, així doncs aquest tipus de joc

pot ajudar a millorar la motricitat i els reflexos del jugador, ja que requereix d'una gran concentració, reaccions ràpides i bona coordinació per a poder superar les diferents dificultats que van apareixen al joc.

Així doncs podríem dir que el target és dels 6 - 65 o més anys.

### 2.1.5. Requeriments del joc

El joc ha de complir una sèrie de requeriments establerts al començament del projecte els quals determinaran el nivell de qualitat i la correcta consecució d'aquest.

- Implementació de un nivell de dificultat gradual que augmentes a mesura que la partida avança i dotes al joc d'un nivell adequat d'experiència de joc.
- Un nivell de jugabilitat alta, per aconseguir un joc dinàmic el qual no es torni monòton al cap d'una estona.
- Aconseguir un producte depurat i funcional el qual estigui lliure d'errors.
- Un aspecte gràfic futurístic, atractiu i depurat.
- Funcionalitats extra que aportin un toc distintiu al joc com poden ser animacions, efectes especials, funcionalitats distintives entre d'altres...
- Complir els objectius plantejats en la fase de disseny del projecte.

Els requeriments mínims per a poder jugar al joc, són els que ens demana Unity per executar un joc en PC o MAC, tot i que Unity ens permet compilar per un gran nombre de plataformes el joc no està destinat a sortir en plataformes online així doncs compilarem per poder executar-lo als sistemes de sobretaula.

- Sistema operatiu Windows XP SP2+, Mac OS X 10.8+, Ubuntu 12.04+, SteamOS+.

- Targeta gràfica amb DirectX9 o superior.
- Cpu compatible amb les instruccions SSE2.

### 2.1.6. Descripció del Funcionament i de la Mecànica del joc

El joc és d'acció i es juga en tercera persona el punt de vista es aeri, d'aquesta manera al ser un scroller 2D aquesta vista és la més adequada i la que dota al joc d'una millor experiència, ja que garanteix al jugador un bon coneixement de la seva posició i del seu entorn.

El jugador controla la seva nau mitjançant l'input del teclat, amb el qual pot desplaçar la nau en els eixos x i y i disparar diferents tipus de projectils als seus enemics.

El tipus de partida és deathmatch en el qual l'objectiu es anar superant les dificultats, derrotar el enemics i aconseguir la màxima puntuació possible, aquest es sumen al marcador de puntuació.

El joc comença amb una dificultat baixa en la qual el nombre i varietat d'enemics que apareix es reduït, a mesura que el jugador va augmentant els seus punts apareixeran més enemics i mes diversitat d'aquestos.

El joc té implementat un sistema de nivells intern els quals determinen la dificultat del joc en relació a la puntuació aconseguida en la partida actual. El nivell de dificultat augmenta cada cop que el jugador aconsegueix mil punts, quan el jugador finalitza la partida i comença de nou el nivell de dificultat torna a ser el més fàcil, així doncs el nivell de dificultat sols té efecte durant la partida, aquesta continua augmentant indefinidament cada cop que se sumen mil punts al comptador de puntuació, cada cop que el nivell de dificultat augmenta desbloqueja nous enemics i aquests apareixen més freqüentment, a partir del sisè nivell de dificultat apareixen tots els diferents tipus d'enemics. Cada cop que el jugador suma cinc-mil punts el fons i la música del joc s'aturen i



apareix un enemic especial o Boss, el qual té una mecànica de fases en les que el Boss dispara diferents tipus de projectils i enemics, el jugador ha de superar les diferents fases del boss destruint els diferents canons que el boss té per derrotar-lo i sumar una gran quantitat de punts.

En el joc tot no són dificultats, també trobem implementat un sistema d'augmentos o power up's mitjançant la qual el jugador pot recuperar la vida perduda, augmentar la seva potencia de foc o aconseguir un escut d'energia que protegeix la nau del jugador, aquests augmentos apareixen cada cop que el nivell de dificultat puja, també apareixen de manera aleatòria al destruir els meteorits que viatgen per l'espai però la probabilitat que donin un power up és baixa.

Altres de les mecàniques que s'han dissenyat al joc són les animacions i els efectes visuals especials els quals mitjançant les eines que ens facilita el Unity i un bon editor de gràfics com és el Photoshop podem aconseguir.

### **2.1.7. Controls del Joc**

Els controls del joc són configurables al principi del joc mitjançant les opcions de configuració, el jugador pot elegir el tipus de control i les tecles o botons amb les quals vol jugar. Els controls del joc estan optimitzats per controlar la nau amb el analògic dels comandaments o joystick, de manera que com mes desplaçem el control a un costat més ràpid és el moviment de la nau o menys.

El joc està dissenyat perquè la resposta un cop el jugador ha fet una pulsació sigui immediata de manera que no hi hagi delay entre la pulsació del teclat i l'acció que succeeix en pantalla, en aquest tipus de joc és molt important que el jugador tingui una resposta immediata i el

control sigui adequat ja que es requereixen de moviments molt ràpids i precisos.

## 2.2. Interfície Gràfica del Joc

En primer lloc el primer que veiem quant iniciem un joc a banda de la seva introducció és la interfície gràfica, la qual és molt important a l'hora de garantir una bona experiència de joc, ja que aquesta és la que garanteix una comunicació amb la màquina. Ha de ser clara i descriptiva i que permeti al jugador navegar intuïtivament de manera natural a través dels menús del joc, els quals ens donaran accés a la partida i a les seves diferents opcions.

La interfície del joc està formada pels diferents elements gràfics que la componen i la lògica que la fa funcionar. El jugador podrà accedir als diferents menús i pantalles del joc mitjançant els botons disposats.

### 2.2.1. Menu Principal

Un cop iniciem el videojoc el primer que el jugador es troba és el menú principal, la seva funció és la de guiar al jugador per les diferents opcions que disposa el joc i facilitar la tasca del jugador.

En el menú principal del joc podem trobar el nom del joc i les tres següents opcions de partida les qual podem accedir en forma de botons:

- Jugar aquesta opció ens envia a una nova escena on apareix el menú de selecció de nau per al jugador.
- Puntuacions aquesta opció ens porta a una pantalla on veiem les màximes puntuacions aconseguides al joc.

- Sortir quant cliquem el botó sortir el joc s'atura i es deixa d'executar.

### 2.2.2. Menú de Selecció de Nau

Aquest menú permet elegir la nau amb la que el jugador vol jugar la partida, mostra una imatge de la nau i una petita descripció de les seves característiques, la interfície d'aquest menú ens permet navegar mitjançant els botons laterals. El menú de selecció de nau conte les següents opcions:

- Enrere ens retorna al menú principal tal i com el seu nom indica.
- Els botons amb les fletxes ens permeten canviar la nau amb la qual volem jugar la partida.
- Start quan cliquem aquest boto la partida s'inicia.

### 2.2.3. Pantalla de puntació

Aquest pantalla mostra les màximes puntuacions aconseguides al joc aquestes es guarden en un fitxer de text. Aquest menú sols té una opció que sols ens permet tornar enrera al menú principal.

### 2.2.4. Pantalla de Final de Partida o GameOver

Aquest pantalla és el que menys volem veure ja que apareix un cop la nau del jugador ha sigut destruïda i significa que la partida ha acabat. Per pantalla podem veure el text de GAMEOVER significant que la partida ha acabat i s'ens mostraran les opcions següents:

- Reintentar la qual tornar a començar la partida de nou amb la puntuació a zero i amb la mateixa selecció de nau que el jugador havia realitzat anteriorment.
- Tornar al Menú ens retorna al menú principal del joc.

### 2.2.5. Pantalla de Pausa

Aquest menú sols pot aparèixer durant la partida clicant la tecla 'p' o 'ESC', un cop aparegui aquest menú la partida quedarà aturada en mode pausa i es mostrarà el text PAUSA en pantalla i les següents opcions de partida:

- Continuar si polsem la tecla 'p' o 'ESC' o la opció la partida és resumirà exactament on l'havíem aturat.
- Torna al Menú ens retorna al menú principal del joc

### 2.2.6. Elements audiovisuals de l'interfície gràfica

Els diferents elements gràfics són els que componen la interfície gràfica del joc per a construir-la s'han utilitzat diferents recursos:

- Background és el fons de pantalla que podem veure en els menús principal, selecció de nau i puntuacions aquest té una ambientació en un espai nebulós i està composta per un planeta, la seva funció es totalment estètica.
- Frame és el enquadrat exterior que veiem a sobre el fons, la seva funció és la d'indicar al jugador que es troba dins d'una interfície gràfica, apart millora el aspecte visual d'aquesta.
- Botons trobem dos estils diferents de botons la seva funció és la de permetre la navegació pels diferents menús del joc i poder seleccionar les opcions de partida. Així doncs, són els encarregats de garantir la navegació pels diferents menús del joc. Aquest quan són premuts es tornen més foscos i es reproduïx un efecte sonor indicant que s'ha clicat un botó.

En la interfície del joc trobem dos tipus d'àudio diferents, un són els efectes sonors els quals es reproduïxen al clicar un botó i l'altre és la banda musical sonora que ens acompanya durant aquesta fase del joc.

### 2.2.7. Taula amb els elements gràfics que conformen l'interfície

**FRAME**



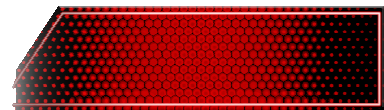
**BACKGROUND**



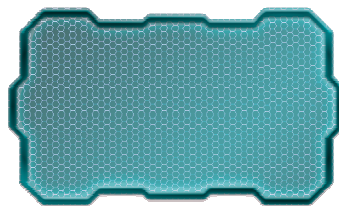
**BOTO MENU**



**BOTO SELECCIO**



**FRAME MENU SELECCIO**



## 2.3. Elements durant la partida

Un cop hem començat la partida l'escena canvia i deixem enrere la interfície gràfica per donar pas a la partida, on el joc comença, aquests

són els elements que trobem dins del joc i els quals interactuen amb la partida.

Elements del joc

Tots els jocs estan formats per un conjunt d'elements els quals donen forma al joc de manera que aquest pren vida, en el cas d'Unity aquest elements són assets els quals contenen els diferents elements que conformen el joc com els audiovisuals, la lògica i les interfícies del joc, aquests elements els afegim a les escenes per a donar vida al videojoc.

### **2.3.1. Elements Estàtics durant la partida**

Aquest elements sempre estaran presents durant la partida de manera que no desapareixen mentres la partida es desenvolupa.

### **2.3.2. Background**

El background o fons és l' element que ens crea l'ambientació del joc i a més ens crea la sensació de moviment mentre la nau viatja per l'espai, ja que el fons és va movent en un loop de manera que ens crea la sensació d'avançar, en termes de videojocs és un scroll lateral. El fons és una imatge png com totes les utilitzades en Unity en aquest projecte, cal recordar que el format .png suporta transparències, el fons està dissenyat perquè es repeteixi indefinidament de manera que el començament i el final d'aquest són iguals, mitjançant aquest truc i la lògica en els scrips podem realitzar l'efecte d'un fons en moviment continu, el qual mai s'acaba.

### **2.3.3. Health Bar o Barra de Vida**

Aquest element el podem trobar a la part superior esquerra i ens indica el nivell de vida o energia que la nau té, quan aquesta desapareix

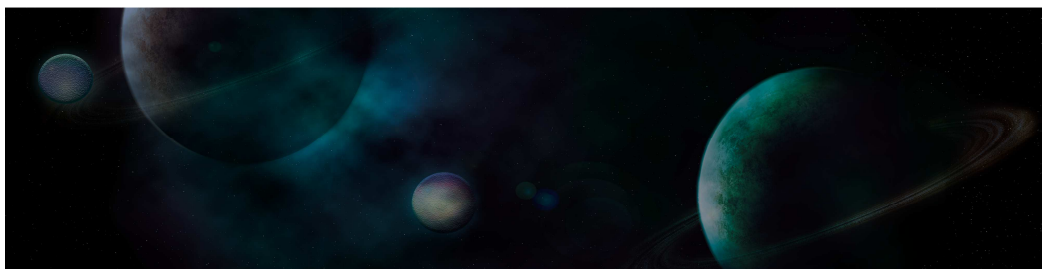
significa que la nau ha sigut destruïda i la partida ha acabat. La barra de vida està formada per dues imatges, una és el contenidor el qual mai varia, l'altre element és la barra d'energia la qual varia a mesura que la nostra energia augmenta o disminueix omplint o buidant el contenidor.

### 2.3.4. Puntuació de la partida

A la part superior dreta podem trobar el comptador de punts aquest és un simple text que ens mostra la puntuació actual de la partida, aquesta augmenta a mesura que derrotem enemics. Un cop ha acabat la partida si la puntuació és una de les màximes 'emmagatzema i la podem veure a la pantalla de puntuacions al menú principal.

### 2.3.5. Taula amb els elements estàtics

#### BACKGROUND



#### HEALTHBAR

##### CONTENEDOR



##### BARRA VIDA



## 2.4. Elements Dinàmics durant la partida

Aquest elements poden aparèixer o desaparèixer durant la partida o no aparèixer. En aquest apartat englobarem tots els demés elements que intervenen durant el joc com son les naus que controla el jugador els enemics i els diferents elements que trobem al joc.

## 2.5. Naus del jugador

En el joc trobem tres naus diferents les quals el jugador pot elegir amb quina vol jugar la partida, aquestes naus tenen un sistema de foc distintiu per a cada d'una d'elles creat mitjançant el sistema de partícules d'unity i uns atributs diferents els quals són:

- Vida o energia indica la resistència de la nau com més alta més impactes enemics pot resistir.
- Velocitat aquesta propietat indica la rapidesa en la que la nau es mou en els eixos X i Y.
- Potència de foc indica la quantitat de dany que la nau realitza als seus enemics.

### 2.5.1. Nau-1

Aquesta és la nau bàsica del joc és caracteritza per ser la nau més ràpida que el jugador pot elegir, alhora també és una de les més resistents per contra és la nau amb menys potencia de foc, sols dispara un projectil.




### 2.5.2. Nau-2

Aquesta nau en aparença és la mateixa que la bàsica però la seva potencia de foc es veu augmentada ja que aquesta nau disposa de dos canons de manera que dispara dos projectils tot i que per contra la seva velocitat es veu fortament afectada.

### 2.5.3. Nau-3

Aquesta nau disposa d'un sistema de foc especial el qual és continu, de manera que mantenint la tecla de foc premuda aquesta nau dispara un raig de plasma que destrueix tot el que es posa pel seu camí. Aquesta nau gaudeix de l'avantatge de no haver d'obrir foc contínuament gaudir d'un armament tant poderós li comporta una velocitat molt reduïda i un nivell de vida mes baix que les seves homònimes.

### 2.5.4. Taula amb les característiques de les naus del jugador

NOM	IMATGE	VIDA	VELOCITAT	P. FOC
N1		3	10	1.5
N2		3	5.5	1
N3		2	6	0.06

## 2.6. Naus enemigues

Durant la partida ens apareixeran diferents tipus d'enemics cadascun amb un sprite característic, el objectiu que tenen és destruir la nau del jugador cada una gaudeix d'un comportament i atributs propis el qual les diferencien de les demés. A mesura que la dificultat augmenta i el joc avança aniran apareixen nous tipus d'enemics els quals dificultaran la tasca del jugador i li suposaran un nou repte que superar, tots els enemics tenen la mateixa possibilitat d'aparèixer però perquè apareguin tots els enemics el jugador ha d'augmentar la seva puntuació fins als sis mils punts.

### 2.6.1. Enemic 1 E-1

Aquest enemic és el més bàsic i apareix des del començament de la partida sols es desplaça amb l'eix x a una velocitat constant i el temps d'obrir foc és aleatori entre un temps mínim i màxim, els projectils van dirigits a la posició on es troba la nau del jugador.

### 2.6.1. Enemic2 mina E-2

Aquest enemic té una velocitat constant en l'eix x i no dispara cap tipus de projectils sinó que explota creant una explosió en àrea que destrueix la nau del jugador si aquest es troba dins del rang de l'explosió, el temps d'explosió aleatòria entre un temps mínim i màxim determinat.

### 2.6.2. Enemic 3 E-3

Aquest enemic es mou en els eixos x i y, i es dirigeix cap a la posició de l'enemic a alta velocitat quan s'apropa a l'enemic disminueix dràsticament la seva velocitat i comença a disparar projectils al jugador aquesta nau continua disparant seguint al jugador fins que es destruïda.

### 2.6.3. Enemic 4 Sniper E-4

Aquesta nau apareix en pantalla i es queda al costat dret de la pantalla i sols es desplaça amb l'eix y seguint al jugador i dispara projectils cap a la posició on es troba la nau del jugador, aquest enemic és molt resistent i el fet que es queda allunyat del jugador el fa difícil de destruir.

### 2.6.4. Enemic 5 Mina cluster E-5

Aquest enemic té el mateix funcionament que la mina amb la diferència que és més gran i que quan explota allibera 4 clústers en forma de mina cap a totes direccions.

### 2.6.5. Enemic 6 (4 projectils) E-6

Aquesta nau enemiga viatja en una velocitat constant en eix x i no es desplaça amb el eix y, té la peculiaritat que dispara quatre projectils en tots costats el seu temps de foc es aleatori entre dos valors màxim i mínim.

### 2.6.6. Enemic 7 (8 projectils) E-7

Aquesta nau té el mateix funcionament que el enemic4projectils però dispara vuit projectils en tots costats perpendiculars i en diagonal a la seva pròpia posició.

### 2.6.7. Enemic Roca

Els meteorits són roques que viatgen a gran velocitat per l'espai, aquests no són enemics pròpiament dits però sí que són molt perillosos ja que si un impacta amb la nau la destrueix, l'única possibilitat de sobreviure a un impacte d'un meteorit és mitjançant un escut d'energia. Aquests es generen de manera aleatòria a l'igual que la seva mida, sempre és mouen en el eix x.

Destruir els meteorits no aporta punts al joc però sí que hi ha un incentiu al destruir-los i és el fet que tenen una possibilitat entre setanta i cinc de generar un augment aleatori per al jugador.

2.6.8. Taula amb les caracterítiques de les diferents naus enemigues.

ENEMIC	IMATGE	VIDA	PUNTUACIO
E1		1	75
E2		2	90
E3		1	25
E4		6	150
E5		1	50
E6		3	105
E7		4	135
ROCA		1	0

### 2.6.9. Enemy Boss

Aquest enemic és un enemic final el qual apareix cada cop que el jugador aconseguix cinc-mil punts. Quan el boss apareix el moviment del background s'aturava en versions anteriors ara és mou més ràpidament ja que dona més impressió i la banda sonora canvia. Aquest està format per diferents fases les quals hem de superar per a poder derrotar-lo.

Cada fase té un canons els quals es troben situats en diferent parts de la nau i obren foc contra el jugador, l'objectiu es destruir els diferents canons de la nau per poder derrotar les diferents fases de l'enemic i destruir la nau. En cada fase l'enemic obre foc amb diferents projectils oferint nous reptes al jugador, el funcionament del Boss està dividit en les següents fases.

**Fase 1** el boss dispara projectils en la direcció on es troba el jugador, el temps de foc és aleatori entre un rang de valors màxim i mínim.

**Fase 2** en aquesta fase el boss dispara projectils com a la fase anterior i genera enemics del tipus E5 del tipus mina cluster aquests no afegixen punts al marcador del jugador ja que formen part del boss.

**Fase 3** aquesta és la última etapa del boss on dispara projectils del tipus míssil els quals es mouen en els seus eixos x i y dibuixant una trajectòria en forma d'onada, apareixen múltiples míssils en pantalla i el jugador ha d'intentar esquivar-los i destruir els canons.

## 2.7. Projectils

Els projectils són els trets de foc que disparen les naus del joc, quan impacten a una nau causen dany, si el dany és major o igual a la vida aquesta és destruïda. Cada projectil té propietats com són el dany i la velocitat depèn de la nau que el dispara ja siguin naus del jugador o enemigues.

En el joc trobem dos tipus diferents de projectils els que utilitzen un sprite o imatge o els que han estat creats mitjançant efectes visuals amb el sistema de partícules d'unity.

Quan un projectil impacta amb un cos aquest generà un efecte visual del color del projectil mitjançant el sistema de partícules, la funcionalitat d'aquest efecte és visual i la d'informar al jugador que els projectils han impactat en un cos.

### 2.7.1. Projectils del jugador

Els projectils que disparen les naus del jugador estan formats per sprites els quals manipulem i afegim partícules mitjançant el sistema de partícules d'Unity. Utilitzem dos mètodes de foc diferents, el projectil de llum que disparen les naus N-1 i N-2, i el raig làser que dispara la nau N-3, aquests dos projectils funcionen de manera oposada. El primer és un sol projectil el qual impacta i causa un dany, el raig làser impacta contínuament a l'enemic ininterrompudament mentre el jugador obri foc, el raig làser genera una quantitat de dany molt més petita per impacte. Una altra diferència important és que els projectils del jugador destrueixen els projectils enemics però el raig làser no, sinó que continuen la seva trajectòria, d'altra banda el raig làser seria massa fàcil i la nau N-3 seria massa poderosa.

### 2.7.2. Projectils Enemics

Pel que fa als projectils enemics trobem tres diferents tipus de projectils els quals són sprites formats per imatges, el funcionament és el mateix que el de les naus del jugador quan impacten causen dany, si aquest es igual o superior a la vida, la nau es destruïda.

### 2.7.3. Taula amb els projectils enemics

NOM	IMATGE	VIDA
PE 1		3
PE 2		3
PE 3		2

## 2.8. Sistema d'augments o Power up's

Durant la partida el jugador té la possibilitat d'aconseguir millores per a la seva nau les quals tenen diferents efectes, que apareixen aleatòriament cada cop que el jugador suma mil punts al marcador, també hi ha una petita possibilitat que al destruir un meteorit ens aparegui un power up. Aquest és classifiquen pel seu color i efecte.

### 2.8.1. Power up Foc

Aquest augment és representa amb el color vermell, l'efecte que produeix és el de millorar la potencia de foc i el dany que els projectils causen, a l'absorbir una millora la grandària del projectil de la nau del jugador no sols causarà més dany també augmentarà la seva mida.






### 2.8.2. Power up Shield

Aquest augment el podem identificar pel seu color blau representatiu, l'efecte que generà en la nau del jugador és el de crear un escut d'energia que protegeix a la nau del jugador. Aquest escut és un efecte generat mitjançant la manipulació de textures amb el sistema de partícules d'Unity.

### 2.8.3. Power up Vida

Aquest augment és representat pel color verd i té la propietat de curar una unitat de vida al jugador. D'altra banda si la vida del jugador es troba al màxim, sumarà una quantitat de punts al marcador.

NOM	IMATGE	EFFECTE
P FOC		MES DANY
P SHIELD		ESCU PROTECTOR
P VIDA		CURA LA VIDA

## 2.9. Disseny Audiovisual dels Elements del joc

### 2.9.1. Disseny gràfic

Pel que fa a les solucions gràfiques creades al joc tots els elements s'han creat mitjançant l'editor de gràfics photoshop en la seva versió 2017, aquesta eina és molt potent i ens permet treballar amb mapes de bits, mitjançant el qual podem aconseguir els resultats desitjats.

Tots els elements del joc i la interfície gràfica han sigut generats utilitzant diferents procediments amb aquesta eina, i s'han exportat com a fitxer del tipus .png, format que ens permet treballar amb transparències i el Unity suporta.

Gràcies al treball amb capes, màscares, les opacitats i els efectes que ens proporciona el photoshop podem aconseguir els resultats que trobem en les diferents solucions gràfiques.

### 2.9.2. Tipografies del joc

Per al joc s'han seleccionat dues tipografies, les quals s'ha considerat adequades al estil i ambientació del joc una d'aquestes fonts l'utilitzem ens els títols i els menús, l'altra font s'ha seleccionat com una font de text per a les descripcions de text entre d'altres la seva funció és complementar la font principal.

La font principal del joc és cgf locust resistance<sup>[7]</sup>, descarregada de la pàgina DaFont.com, és una font de ciència ficció la qual s'adapta molt bé al estil del joc, per als nostres títols i menús, és una font amb estil artístic.

L'altra font utilitzada és la Arial degut a que és una font que es llegeix molt bé i extensament utilitzada. Des de la seva aparició és un tipus de

lletra sans serif la qual ve inclosa al sistema operatiu Windows i s'ha tornat una font de referència en el món digital.

### 2.9.3. Disseny de l'Audio

Per a la creació del àudio hem treballat amb l'eina Audacity aquesta eina ens permet modificar les ones sonores mitjançant un conjunt de tècniques i efectes que ofereix l'eina.

### 2.9.4. Banda Sonora

Pel que fa a les melodies s'han utilitzat quatre temes diferents d'editoria lliure del web OpenGameArt.org, el qual té gran quantitat d'art realitzat per altres usuaris el que és de lliure ús i gratuït. Aquest temes són de temàtica espacial s'han ajustat els nivells de so i s'han convertit al format .ogg Vorbis, també s'ha conservat la auditoria dels creador.

- Tema del menú de puntuacions: *Enchanted tiki 86* [3]
- Tema durant la partida in-game: *Kim Lightyear - The Others* [4]
- Tema del menú principal i de selecció: *Maxstack - Through space* [5]
- Tema del Boss *Clearside - Assimilator* [6]

### 2.9.5. Efectes de So

Aquests àudios són els que conformen els diferents efectes que sentim al joc, engloben els que és reproduïxen al clicar a un boto tant com els sons durant la partida. Per a realitzar aquests sons s'ha grabat la veu humana i s'han utilitzat diferents mètodes per tal d'aconseguir els diferents efectes que componen el joc, els principals són el canvi de freqüència de ona i de to, diferents filtres com són LFO o el reverb

entre d'altres, també s'ha modificat la velocitat de reproducció d'algunes pistes i s'ha ajustat la compressió i l'equalització en totes elles.

<b>Foc 1</b>	Sò que utilitza la nau N 1 al obrir foc
<b>Foc 2</b>	Sò que utilitzat la nau N 2 al obrir foc
<b>Foc 3</b>	Sò que utilitzen les naus enemigues al obrir foc
<b>Laser</b>	Sò que utilitza la nau N3 al obrir foc
<b>Clickmenu</b>	Sò que es reproduueix al clicar un botó
<b>Explosió</b>	Sò que es reproduueix quan una nau és destruïda

## 3. Desenvolupament i Implementació

### 3.1. Introducció a Unity i el seu funcionament

En la fase de desenvolupament i implementació és portaran a terme les diferents mecàniques i idees dissenyades en la fase de disseny del joc. La implementació del joc s'ha realitzat amb Unity amb la seva versió més actualitzada i el llenguatge utilitzat és el C#. En aquest apartat de la memòria s'expliquen els passos i mètodes seguits per implementar les mecàniques i els diferents elements del joc.

Abans d'aprofundir en les diferents implementacions hem de parlar del funcionament del motor del joc i dels seus aspectes que el caracteritzen. Unity ens ofereix la capacitat de treballar juntament amb diferents programes de disseny gràfic i disseny 3D, d'altra banda ens ofereix una mecànica de *ShaderLab* la qual ens permet la creació d'ombrejats i realitzar diferents efectes també ofereix un motor de físiques de Nvidia i gaudeix d'una tecnologia d'animació pròpia, *Mecanim* la qual permet aconseguir un moviment fluid i natural dels personatges i cossos.

La interfície d'Unity ens ofereix un conjunt d'eines les quals ens ajudaran a muntar les escenes del joc, les quals estan formades pels *GameObject*, són l'objecte bàsic d'Unity el qual ens permet afegir diferents components permet modificar el comportament de l'objecte per ajustar-lo als requeriments del disseny.

### 3.2. Estructura de fitxers del joc

Els diferents fitxers que contenen els elements del joc en Unity es troben dins de la carpeta *Assets* on podem organitzar les carpetes lliurement depenent de les necessitats del nostre projecte. Trobem les diferents carpetes.

- **Escenes** contenen les diferents escenes del projecte.
- **Materials** conte materials utilitzats al joc creat amb l'editor d'Unity.
- **Resources** aquesta carpeta conte tots els assets que podríem necessitar utilitzar durant l'execució del joc.
- **Scripts** conté tota la lògica del joc.
- **SFX** conte els elements necessaris per als efectes especials.
- **Shader** conté fitxers de shader.

*Resources* aquesta carpeta és molt important ja que conté els *prefabs* dels joc, aquests són *GameObjects* prefabricats amb components afegits i les seves propietats ajustades, això ens permet cridar aquests objectes mitjançant els scripts i utilitzar-los durant l'execució del codi.

### 3.3. Scripts del joc

Els scripts del joc contenen la lògica del joc, el llenguatge utilitzat és el C# el qual és un llenguatge orientat a objectes.

Els diferents scripts del projecte estan organitzats en carpetes segons la seva funció. Per tal de reutilitzar el codi de manera eficient, alguns dels scripts del joc són heretats per altres scripts que aprofiten la funcionalitat del pare i modifiquen o implementen noves funcionalitats d'aquest.

### 3.3.1.1. Estructura de fitxers dels Scripts

- Controladors
- Enemics
- Jugador
- Menus
- Objectes

## 3.4. Implementació del codi

Aquest apartat seguirà l'estructura de fitxers dels scripts.

### 3.4.1. Controladors

Són els scripts encarregats de controlar les escenes del joc.

#### 3.4.1.1. ControladorJoc

Aquest és un dels scripts més importants del joc ja que conté la lògica que controla els diferents aspectes bàsics de la partida com la puntuació, els nivell de dificultat intern del joc, les fases de l'enemic Boss, controla la pausa i l'estat de la partida com el GameOver.

#### 3.4.1.2. ControladorBackground

Aquest scrip controla el moviment del background, el background es pot aturar mitjançant la funció 'PauseScroll'.

Aquesta funció és la que fa moure el background.

```
private void Update()
{
    if (!isPaused) {
        scroll += Time.deltaTime;
        newPosition = Mathf.Repeat(scroll * scrollSpeed,
            pixel2units);
        transform.position = startPosition + new Vector3(-1, 0, 0)
            * newPosition;
    }
}
```

El més important d'aquesta funció es 'Mathf.Repeat' el que fa és lopejar el primer dels paràmetres que rep de manera que aquest mai és major que el segon paràmetre ni menor de zero.

#### 3.4.1.3. DobleAudioSource

Aquest scrip s'encarrega de realitzar una transició entre pistes d'audio amb *crossfading* d'aquesta manera quant la pista s'acaba aquesta disminueix el seu volum mentre l'altra pista comença i augmenta el seu volum.

Aquest codi forma part del fòrum d'Unity i ha estat realitzat per *Igor Aherne*.

#### 3.4.1.4. ControladorSeleccióDeNau

Aquest scrip s'encarrega de controlar l'escena de selecció de nau.

#### 3.4.1.5. ScriptMostraPuntuacions

S'encarrega de controlar la pantalla de Puntuacions.



### 3.4.2. Enemies

Són els scrips que s'encarreguen de controlar els enemics.

#### 3.4.2.1. ScriptEnemy

Aquest és el scrip que defineix el enemic més bàsic apartir d'aquest scrip tota la resta d'enemics aconseguen un conjunt de funcionalitats bàsiques comuns amb la resta de scrips així doncs aquest és el scrip Pare dels enemics.

```
protected virtual void Start()
{
    rigidNau = GetComponent<Rigidbody2D>();
    rigidNau.velocity = new Vector2(speed,
    rigidNau.velocity.y);
    animador = GetComponent<Animator>();
    spriteRenderer = GetComponent<SpriteRenderer>();
    isDead = false;
    //Mentres l'animador estigui activat, no podrem canviar els
    sprites de la nau, degut a que l'animació de
    //l'explosió te una curva sprite. El desactivem fins que
    necessitem fer- ho servir.
    animador.enabled = false;

    //DisparoPeriodic es un mètode recursiu i, per tant, en
    cridarla per primer cop
    //inicia el procediment que fa disparar la nau cada cert t
    temps aleatori entre dos valors.
    Invoke("DisparoPeriodic", Random.Range(minFireTime,
    maxFireTime));
}
```

Aquesta funció inicialitza tots els valors de la nau enemiga. És important notar que desactivem l'animador per a poder canviar els sprites de les naus enemigues durant la seva execució.

```
public virtual void DisparoPeriodic()
{
    if (!isDead)
    {
        //Aconseguim l'objecte de la nau del jugador
        GameObject nauJugador =
        GameObject.Find("spaceship");
        if (nauJugador != null)
        {
            // Transform.position és la posició del objecte
            al qual el script està referenciat
            // Quaternion.identity = afegeix la bala sense
            rotació
        }
    }
}
```

```

GameObject bala = Instantiate(bullet,
transform.position, transform.rotation);
ScriptFocEnemy balaScript =
bala.GetComponent<ScriptFocEnemy>();

//L'hi donem al projectil les propietats que
hem configurat per la nau
balaScript.attack = attack;
balaScript.vida = vidaBala;

//Fem sonar l'audio de disparo de la nau
AudioSource.PlayClipAtPoint(bulletSound,
transform.position, 1.3f);
//Encenem les llums de la nau (alternem entre
sprite 1 i 2)
turnOnLights();

//Apaguem les llums de la nau al cap de mitg
segon
Invoke("turnOffLights", 0.5f);
}

// L'enemic basic dispara de forma periòdica, per
tant executem la funció de forma recursiva cada x
temps aleatori
Invoke("DisparoPeriodic", Random.Range(minFireTime,
maxFireTime));
}
}

```

Aquesta funció s'executa de manera recursiva per a que les naus enemigues continuïn obrint foc.

```

// Aquesta funció s'executa al colisionar amb un altre
objecte.
private void OnTriggerEnter2D(Collider2D obj)
{
    // nom de l'objecte que ha xocat
    var name = obj.gameObject.name;

    // si ha xocat amb una bala
    if ((name == "bullet(Clone)" || name ==
"laser(Clone)") && !isDead)
    {
        haveBeenShot(obj);
    }

    // si la roca colisiona contra la nau del jugador,
es destrueix la roca i la nau, mitjançant la funció
GameOver
    if (name == "spaceship")
    {
        playerCollision(obj);
    }
}
}

```

Aquesta funció controla les col·lisions dels elements del joc.

```

protected virtual void haveBeenShot(Collider2D obj)
{
    //aconseguim l'atac de la bala
    ScriptFocNau objBala =
obj.gameObject.GetComponent<ScriptFocNau>();
    float attack = objBala.GetAttack();
    GameObject efecteImpacte =
obj.gameObject.GetComponent<ScriptFocNau>().efecteImpa
cte;

    //el restem a la vida
    vida -= attack;

    //efecte de hit
    Instantiate(efecteImpacte, obj.bounds.center,
Quaternion.Inverse(transform.rotation));

    //si ja no te vida
    if (vida <= 0)
    {
        //Hem d'activar l'animador al morir.
        animador.enabled = true;
        isDead = true;
        gameObject.layer = 2;
        spriteRenderer.color = Color.white;
        animador.SetBool("IsGameOver", true);

        AudioSource.PlayClipAtPoint(Resources.Load<AudioClip>(
"Audio/Sounds/explosio"), transform.position);
        afegirPunts();
        Destroy(obj.gameObject);
        // Es destrueix a si mateix i a l'objecte que
        ha colisionat. Al cap d'un segon, que es el que
        dura l'animació.
        Invoke("selfDestroyCoroutine", 1);
    }
    else
    {
        //hem de destruir la bala sempre
        Destroy(obj.gameObject);
    }
}
}

```

Aquesta funció es crida quan un projectil del jugador impacta amb una de les naus enemigues, en resta la vida a la nau, genera l'efecte hit, activa l'animador si aquest ha mort, també canvia la capa a la que es troba el GameObject i la canvia a la capa 'Ignore Raycast', això ho fem perquè una nau que ha mort no pugui interferir en un Raycast.

```
private void afegirPunts()
{
    GameObject controlador =
    GameObject.Find("Controlador");
    ControladorJoc scriptControlador =
    controlador.GetComponent<ControladorJoc>();
    scriptControlador.AfegirPunts(valorPunts);
}
```

Aquesta funció crida al script ControladorJoc i afegeix els punts de la nau enemiga que ha sigut destruïda.

### 3.4.2.2. ScriptEnemic4Disparos

Aquest script modifica la funcionalitat del DisparoPeriodic per a què l'enemic cada cop que dispari es generin quatre projectils i viatgin en els quatre punts cardinals.

#### 3.4.2.1. ScriptEnemic8Disparos

Aquest script fa el mateix que l'anterior però en comptes de generar quatre projectils en genera vuit que viatgen en els quatre punts cardinals i en diagonal.

### 3.4.2.2. ScriptEnemicMina

Aquest script fa que l'enemic exploti al cap d'una estona generant una explosió que destrueix a l'enemic i a la nau del jugador si es veu atrapada en ella.

```
protected virtual void ExplodeMine()
{
    isDead = true;
    spriteRenderer.color = Color.white;
    //Per tal de donar una mica mes de dificultat, abans
    d'explotar la mina augmentem l'escala per tenir una
    explosió mes gran.
    transform.localScale = new Vector2(1.5f,1.5f);
    GetComponent<BoxCollider2D>().size = new Vector2(2,2);
    animador.SetBool("IsGameOver", true);

    AudioSource.PlayClipAtPoint(Resources.Load<AudioClip>("Audio/
    Sounds/explosio"), transform.position);
}
```

```

        // Es destrueix a si mateix i a l'objecte que ha
        // colisionat.
        Invoke("selfDestroyCoroutine", 1);
    }

```

### 3.4.2.3. ScriptEnemyClusterMina

Aquest script és un fill de ScriptEnemyMina, i la seva funcionalitat és la de generar quatre mines al moment que explota i les envia en diferents direccions de manera aleatòria.

### 3.4.2.4. ScriptEnemyMovel

Aquest script generarà un enemic que segueix al jugador i quan s'apropa al jugador disminueix la seva velocitat.

```

public override void UpdateIfAlive()
{
    base.UpdateIfAlive();

    GameObject player = GameObject.Find("spaceship");
    Rigidbody2D rigidEnemy =
    GetComponent<Rigidbody2D>();

    //Apuntem la nau cap al jugador

    Vector3 dir = player.transform.position -
    transform.position;

    //I ens movem cap allí

    rigidEnemy.velocity = dir;
}
}

```

Aquest codi afegeix funcionalitats a sobre de les que ja té l'objecte pare.

### 3.4.2.5. ScriptEnemySniper

Aquest script genera un enemic que es queda allunyat del jugador i el segueix en el eix y.

### 3.4.2.6. ScriptFocEnemy

Aquest script controla el foc enemic. Al iniciar-se s'aplica una velocitat al projectil i apunta cap a la direcció on es troba el jugador en el moment que es crea el projectil, també controla si el projectil ha col·lisionat.

```

public virtual void Start()
{
    Rigidbody2D rigidBala = GetComponent<Rigidbody2D>();
    GameObject player = GameObject.Find("spaceship");

    //Apuntem la bala cap a puesto

    Vector3 dir = player.transform.position -
transform.position;

    // Mou la bala

    rigidBala.velocity = dir;
}

protected virtual void OnTriggerEnter2D(Collider2D obj)
{
    var name = obj.gameObject.name;

    //comprovem si hem colisionat amb una bala del
jugador o amb el jugador.
    if (name == "bullet(Clone)") hitByOtherBullet(obj);
    else if (name == "spaceship") hitPlayer(obj);
}

protected virtual void hitByOtherBullet(Collider2D obj)
{
    vida--;
    if (vida <= 0)
    {
        GameObject efecte =
obj.gameObject.GetComponent<ScriptFocNau>().efecteImpa
cte;
        //Instanciem l'efecte de hit de la bala del
jugador
        Instantiate(efecte, obj.bounds.center,
Quaternion.Inverse(transform.rotation));
        Destroy(obj.gameObject);
        dead();
    }
    else
    {
        //Instanciem l'efecte de hit de la bala de
l'enemic
        Instantiate(efecteImpacte, obj.bounds.center,
Quaternion.Inverse(transform.rotation));
    }
}

```

```

        Destroy(obj.gameObject);
    }
}

protected virtual void dead()
{
    Destroy(gameObject);
}

protected virtual void hitPlayer(Collider2D obj)
{
    ScriptNauBasic nauScrip =
    obj.gameObject.GetComponent<ScriptNauBasic>();

    Destroy(gameObject);
    Instantiate(efecteImpacte, obj.bounds.center,
    Quaternion.Inverse(transform.rotation));
    nauScrip.Colision(attack);
}

```

### 3.4.2.7. ScriptFocEnemyNoVelocity

Aquest script està generat perquè el projectil no apunti el jugador, aquest foc el utilitzen les naus E6 i E7 les quals disparen quatre i vuit projectils respectivament.

### 3.4.2.8. ScriptMissilEnemy

Aquest script conté la lògica el qual controla el missil el qual dispara l'enemic Boss en la seva tercera fase. Aquest script és fill de ScriptFocEnemy.

```

    public override void Start()
    {
        rigidMissil = GetComponent<Rigidbody2D>();
        initialPositionY = transform.position.y;

        rigidMissil.velocity = transform.forward;

        newPosition = transform.position;
        newPosition.y -= 0.01f;

        //El bool baixant controla si el missil esta descrivint la curva
        cap amunt o cap abaix
        baixant = true;

        //El tipus de moviment (semicercles o recte) es tria de forma
        aleatoria al instanciar el missil.
        moviment = Random.Range(0,2);

        if (moviment == 0) rigidMissil.velocity = Vector2.left * speed;

        animador = GetComponent<Animator>();
    }

```

```

        spriteRenderer = GetComponent<SpriteRenderer>();
    }

    // Update is called once per frame
    void Update () {
        if (!isDead)
        {
            if (moviment == 1)
            {
                //Es fa moure de forma similar a la nau que persegueix al
                jugador.
                //Persegueix un punt que es troba per davant del missil, i
                que va oscilant de forma que descriu el moviment que ha de
                fer el missil
                //Es similar a un ruc perseguint una pastanaga lligada a
                un fil
                if (baixant) newposition = new
                Vector3(transform.position.x - speed, newposition.y - (3.75f
                * Time.smoothDeltaTime), newposition.z);
                else newposition = new Vector3(transform.position.x -
                speed, newposition.y + (3.75f * Time.smoothDeltaTime),
                newposition.z);
                Vector3 dir = newposition - transform.position;
                float angle = Mathf.Atan2(dir.y, dir.x) * Mathf.Rad2Deg;
                transform.rotation = Quaternion.AngleAxis(angle - 180,
                Vector3.forward);

                //I ens movem cap allí
                rigidMisil.velocity = dir;
                if (newposition.y <= initialPositionY - 2.75f) baixant =
                false;
                if (newposition.y >= initialPositionY + 2.75f) baixant =
                true;
            }
        }
        else rigidMisil.velocity = new Vector2(0, 0);
    }
}

```

Aquestes dues funcions són les que modifiquen el comportament del scrip pare per aconseguir el comportament desitjat.

### 3.4.2.9. ScriptRoca

Aquest script controla el comportament dels meteòrits que apareixen al joc.



### 3.4.2.10. ScriptSpawn

Aquest script genera els enemics que apareixen al joc durant la partida, també genera els power ups que apareixen cada mil punts al pujar el nivell de dificultat intern del joc.

```
private void addEnemyShip()
{
    if (isSpawning)
    {
        Renderer ship = GetComponent<Renderer>();
        float x1 = transform.position.y - ship.bounds.size.y / 2;
        float x2 = transform.position.y + ship.bounds.size.y / 2;
        int i = 0;

        Vector2 spawnPoint = new Vector2(transform.position.x,
        Random.Range(x1, x2));

        //Segons el level de la partida actual, s'instancien mes o
        //menys enemics diferents
        switch (cj.GetLevel())
        {
            case 0:
                i = 0;
                break;

            case 1:
                i = Random.Range(0, enemyShip.Length - 4);
                break;

            case 2:
                i = Random.Range(0, enemyShip.Length - 3);
                break;

            case 3:
                i = Random.Range(0, enemyShip.Length - 2);
                break;

            case 4:
                i = Random.Range(0, enemyShip.Length - 1);
                break;

            default:
                i = Random.Range(0, enemyShip.Length);
                break;
        }

        Instantiate(enemyShip[i], spawnPoint,
        Quaternion.identity);

        //També a partir de nivell 4, es generen enemics
        //vadicionals de forma aleatoria segons el nivell.
        if (cj.GetLevel() > 3)
        {
            //El bucle es repeteix un nombre de cops equivalent a
            //la meitat del nivell.
            for (int repeat = 0; repeat < cj.GetLevel() / 2;
            repeat++)
            {
```

```

        int level = cj.GetLevel();
        if (level > 11) level = 11; //A partir de level
        12, ja no es aleatori: es genera un enemic cada
        cop que s'executa aquest bucle.
        int random = Random.Range(level, 12);

        //Si el numero generat es 11, el enemic adicional
        es genera
        if (random == 11) addEnemyCoroutine(new
        Vector2(transform.position.x, Random.Range(x1,
        x2)), Random.Range(0, enemyShip.Length));
    }
}
//Aquesta funció es crida recursivament per anar generant
enemics.
//Es crida fins i tot quan el spawn esta desactivat (p.ex. un
boss)
//per a que en quant s'activi un altre cop es tornin a generar
enemics.
if (cj.GetLevel() > 5) Invoke("addEnemyShip",
    Random.Range(0.5f, 3.5f));
else Invoke("addEnemyShip", Random.Range(2f, 7f));
}

IEnumerator addEnemyCoroutine(Vector2 spawnPoint, int enemy)
{
    float wait = Random.Range(0.3f,1.7f);
    yield return new WaitForSeconds(wait);
    Instantiate(enemyShip[enemy], spawnPoint,
        Quaternion.identity);
}

```

Aquesta Coroutine ens permet a l'hora de generar varis enemics addicionals, que aquest no aperiguin tots alhora.

```

private void addEnemyMine()
{
    if (isSpawning)
    {
        if (cj.GetLevel() >= 0)
        {
            Renderer ship = GetComponent<Renderer>();
            var x1 = transform.position.y - ship.bounds.size.y /
                2;
            var x2 = transform.position.y + ship.bounds.size.y /
                2;

            Vector2 spawnPoint = new Vector2(transform.position.x,
                Random.Range(x1, x2));

            int i = Random.Range(0, 2);
            int mina = Random.Range(0, mines.Length);

            if (cj.GetLevel() < 3) mina = 0;

            if (i == 1) Instantiate(mines[mina], spawnPoint,
                Quaternion.identity);
        }
    }
}

```

```

//Aquesta funció es crida recursivament per anar generant
enemics.
Invoke("addEnemyShip", Random.Range(5f, 12f));
}

```

### 3.4.2.11. ScriptBossBase

Aquest script té la funció de fer arribar el boss al seu punt, iniciar les diferents fases que té el boss

```

private void nextFase()
{
    turnOnLights();
    faseActual++;
    //comprovem si queden fases per completar. De ser així,
    iniciem la següent fase al cap d'un segon.
    if (faseActual < fases.Count -1)
        Invoke("nextFaseLightsOff",1f);
    else
    {
        //activem l'animador de nou
        animador.enabled = true;
        //animacio destruccio boss
        animador.SetBool("IsGameOver", true);

        AudioSource.PlayClipAtPoint(Resources.Load<AudioClip>("Audio/Sounds
        /explosio"), transform.position);
        //informem al controlador del joc
        controladorJoc.FinalitzaMonstre();
        //destruim el boss
        Destroy(gameObject);
    }
}
}

```

Aquesta funció és la que s'encarrega de canviar la fase del boss.

### 3.4.2.12. BossComponentBase

Aquest script s'encarrega de controlar els canons que conformen cada fase del Boss.

```

protected virtual void shoot()
{
    if (active && !isDead)
    {
        //Aconseguim l'objecte de la nau del jugador
        GameObject nauJugador = GameObject.Find("spaceship");
    }
}

```

```

if (nauJugador != null)
{
    // Transform.position és la posició del objecte al
    // qual el script està referenciat
    // Quaternion.identity = afegeix la bala sense
    // rotació
    GameObject cbullet = Instantiate(bullet,
    transform.position, transform.rotation);
    ScriptEnemy scriptBullet =
    cbullet.GetComponent<ScriptEnemy>();

    turnOnLights();

    //Si es null, aquest component no dispara un
    //enemic.
    //El boss pot disparar bales o enemics, però si
    //disparem un enemic hem de fer que no doni punts.
    if (scriptBullet != null) scriptBullet.valorPunts =
    0;

    ScriptFocEnemy balaScript =
    cbullet.GetComponent<ScriptFocEnemy>();
    //Si es null, aquest component no dispara una bala
    if(balaScript != null)
    {
        balaScript.attack = attack;
        balaScript.vida = vidaBala;
        AudioSource.PlayClipAtPoint(bulletSound,
        transform.position, 1.3f);
        Invoke("turnOffLights", 0.5f);
    }
}

// L'enemic basic dispara de forma periòdica, per tant
// executem la funció de forma recursiva cada x temps
// aleatori
Invoke("shoot", Random.Range(minFireTime,
maxFireTime));
}
}

```

Aquesta funció és molt similar al foc dels altres enemics però afegeix comprovacions per tal que els canons puguin disparar projectils i enemics. Addicionalment aquesta funció també s'encarrega de no sumar punts si es generà un enemic per tal de que no es descontrolï la puntuació durant la fase de Boss.

### 3.4.2.13. BossFaseBase

Aquest script s'encarrega de inicialitzar els components de la fase i durant la fase comprovar si s'han destruït els canons que conformen la fase.

### 3.4.2.14. ScriptFocNau

Aquest script s'encarrega de inicialitzar el projectil del jugador i donar-li velocitat.

### 3.4.2.15. ScriptNauBasic

Aquest scrip s'encarrega de controlar el moviment, el foc, les col·lisions, si la partida ha acabat entre d'altres de la nau del jugador, com són els power up's.

```
private void Update()
{
    // Aconseguim el component Rigidbody per als controls
    // de la nau
    Rigidbody2D rigidNau = GetComponent<Rigidbody2D>();
    if (canControl)
    {
        //Donem moviment a la nau en els seus eixos
        if (Input.GetAxis("Horizontal") > 0 &&
            rigidNau.position.x < 8.4)
            rigidNau.velocity = new
            Vector2(Input.GetAxis("Horizontal") *
            speed, rigidNau.velocity.y); // = speed;
        else if (Input.GetAxis("Horizontal") < 0 &&
            rigidNau.position.x > - 8.4)
            rigidNau.velocity = new
            Vector2(Input.GetAxis("Horizontal") * speed,
            rigidNau.velocity.y);
        else
            rigidNau.velocity = new Vector2(0,
            rigidNau.velocity.y);
    }
}
```

```

    if (Input.GetAxis("Vertical") > 0 &&
        rigidNau.position.y < 4.4)
        rigidNau.velocity = new
        Vector2(rigidNau.velocity.x,
        Input.GetAxis("Vertical") * speed);
    else if (Input.GetAxis("Vertical") < 0 &&
        rigidNau.position.y > -4.4)
        rigidNau.velocity = new
        Vector2(rigidNau.velocity.x,
        Input.GetAxis("Vertical") * speed);
    else
        rigidNau.velocity = new
        Vector2(rigidNau.velocity.x, 0);

        if (ShootOrNot())
        {
            Fire();
        }
    }
    else
    {
        rigidNau.velocity.Set(0, 0);
    }
}

```

Aquesta funció controla el moviment de la nau en els seus eixos.

```

public virtual void Fire()
{
    // Transform.position és la posició del objecte al
    // qual el script està referenciat
    // Quaternion.identity = afegeix la bala sense
    // rotació
    Vector3 sortidaBala = transform.position;
    GameObject bala;
    ScriptFocNau scriptBala;

    sortidaBala.x += 0.75f;

    bala = Instantiate(bullet, sortidaBala,
    Quaternion.Inverse(transform.rotation));

    //la bala pot augmentar de mida amb els powerups.
    //Per tant, despres d'instanciar-la, canviem l'escala
    //dels dos
    //gameobjects que componen el projectil
    bala.transform.Find("Particle System").localScale =
    new Vector3(1+bulletScale, 1+bulletScale,
    1+bulletScale);
    bala.transform.localScale = new
    Vector3(bala.transform.localScale.x + bulletScale,
    bala.transform.localScale.y + bulletScale,
    bala.transform.localScale.z + bulletScale);

    scriptBala = bala.GetComponent<ScriptFocNau>();
    scriptBala.SetAttack(atac);

    //fem sonar el so que hem assignat al inspector de
    //Unity
}

```

```

        AudioSource.PlayClipAtPoint(bulletSound,
        transform.position);
    }

```

Aquesta funció controla el foc de la nau del jugador.

```

void OnCollisionEnter2D(Collision2D obj)
{
    Rigidbody2D rigidNau =
    GetComponent<Rigidbody2D>();

    string name = obj.gameObject.tag;

    // Evitarem que la nau marxi fora dels limits
    que hem especificat mitjançant objectes
    invisibles
    if (name == "limitx")
    {
        rigidNau.velocity.Set(0,
        rigidNau.velocity.y);
    }
    if (name == "limity")
    {
        rigidNau.velocity.Set(rigidNau.velocity.x,
        0);
    }
}

```

Aquesta funció s'encarrega que la nau del jugador no surti dels límits de la pantalla.

```

public void Colision(int attack)
{
    if (!shield)
    {
        vida -= attack;
        barraVida.sizeDelta = new Vector2( (vida *
        rectInicialX) / vidaInicial,
        barraVida.sizeDelta.y);
        if (vida <= 0)
        {
            GameOver();
        }
    }
    else RemoveShield();
}

```

Aquesta funció controla quant un enemic o projectil impacta amb la nau del jugador.

#### 3.4.2.16. ScriptNauGran

Aquesta script modifica el ScriptNauBasic el qual és el pare, per dotar a la nau N 2 de dos projectils.

#### 3.4.2.17. ScriptNauLaser

Aquest script s'encarrega de modificar el ScriptNauBasic de manera que cada cop que dispari aquesta cridi a la funció Firelaser, del script laserScript, el qual conte lògica del foc làser.

### 3.4.3. Menus

Aquesta carpeta conte els scripts que s'encarreguen de controlar les interfícies dels menus i la navegació entre ells, també el controlen la música inicial de cada escena i les pantalles de Pausa i GameOver.

- **UIManagerIngame**
- **UIManagerMenu**
- **UIManagerPuntuacions**
- **UIManagerSeleccoNau**



### 3.4.4. Objectes

#### 3.4.4.1. Ddol

Aquest script del objecte Ddol (Don't destroy on Load) és un objecte que no es destrueix al canviar entre escenes, per tant ens és molt útil per fer sonar la música o passar paràmetres entre escenes.

#### 3.4.4.2. laserScript

Aquest script implementa tota la mecànica per al foc làser.

```
public void FireLaser()
{
    //No hauria de passar mai, però parem primer el
    //disparo per no estar disparant dos cops alhora
    StopCoroutine("Fire");
    StartCoroutine("Fire");
}
```

Aquesta és la funció que es crida quant volem disparar el laser. Es crida primer StopCoroutine perquè en cap cas sigui possible disparar dos làsers.

```
public IEnumerator Fire()
{
    //Per disparar el laser, aquest metode utilitza mes
    //endavant un raycast per calcular fins on pot arribar
    //el laser fins
    //trobar un obstacle. A mes, utilitza el punt de
    //colisió amb el raycast per crear-hi a la seva
    //posició projectils de forma sostinguda
    laserLine.enabled = true;
    sonidoLaser.Play();
    scriptNau.turnOnLights();

    //Mentres tinguem la tecla de disparo presionada
    while(Input.GetButton("Fire1")) {
        //creem el raycast
        Ray2D ray = new Ray2D(transform.position,
            Vector2.right);
        RaycastHit2D hit;

        hit = Physics2D.Raycast(transform.position,
            Vector2.right, 16.3f);
    }
}
```

```

//Indiquem la posició de sortida del laserLine
(l'efecte visual del laser).
//En aquest cas, l'inici del raycast
laserLine.SetPosition(0,ray.origin);

//si el raycast no ha impactat amb cap enemic
fem un laser de tota la pantalla (arriba fins
la posició 16.3, mes enlla del que es veu)
if (hit.collider == null)
laserLine.SetPosition(1, ray.GetPoint(16.3f));
else
{
    //si ha impactat, la posició final del
    laser serà el lloc de impacte
    laserLine.SetPosition(1, hit.point);

    GameObject bala;
    ScriptFocNau scriptBala;

    //creem una bala a la posició d'on impacta
    el laser
    bala = Instantiate(bullet,
    transform.position,
    Quaternion.Inverse(transform.rotation));
    scriptBala =
    bala.GetComponent<ScriptFocNau>();
    scriptBala.SetAttack(scriptNau.atac);

    scriptBala.efecteImpacte.transform.localScale
    = new Vector3(.25f,.25f,.25f);
    foreach(Transform t in
    scriptBala.efecteImpacte.transform)
    {
        t.localScale = new Vector3(.25f, .25f,
        .25f);
    }

    bala.transform.position = hit.point;
}
}

```

Aquesta funció s'encarrega de disparar el làser i de fer totes les comprovacions necessàries, també s'encarrega dels efectes audiovisuals del làser.

```

yield return null;
}
laserLine.enabled = false;
sonidoLaser.Stop();
scriptNau.turnOffLights();
}

public void GrowLaser(float growth)
{
    laserLine.startWidth = laserLine.startWidth +
    growth;
    laserLine.endWidth = laserLine.endWidth + growth;
}

```

Aquesta funció s'encarrega de fer més gran el làser.

### 3.4.4.3. ScriptPowerUpBase

El script s'encarrega de controlar el sistema d'augmentos o power up's.

```

// Aquesta funció s'executa al colisionar amb un
// altre objecte.
private void OnTriggerEnter2D(Collider2D obj)
{
    // nom de l'objecte que ha xocat
    var name = obj.gameObject.name;

    // si la roca colisiona contra la nau del
    // jugador, es destrueix la roca i la nau,
    // mitjançant la funció GameOver
    if (name == "spaceship")
    {
        ScriptNauBasic ns =
        obj.gameObject.GetComponent<ScriptNauBasic>();
        addToShip(ns);
    }
}

public virtual void addToShip(ScriptNauBasic
nauScript) {}
}

```

Aquesta funció controla el tipus de power up que ha col·lisionat i l'afegeix a la nau.

#### 3.4.4.4. **ScriptPowerUpAtac,criptPowerUpShield, ScriptPowerUpVida**

Aquests scrips modifiquen la funcionalitat de la funcio addToShip del ScriptNauBasic, per afegir el agument corresponent.

### 3.5. **Implementació d'efectes del joc**

Unity ens permet realitzar una gran varietat d'efectes mitjançant una gran quantitat d'eines que tenim a l'abast, mitjançant els materials, les textures i a banda el sistema de partícules ens permeten aconseguir les diferents solucions aplicades al joc.

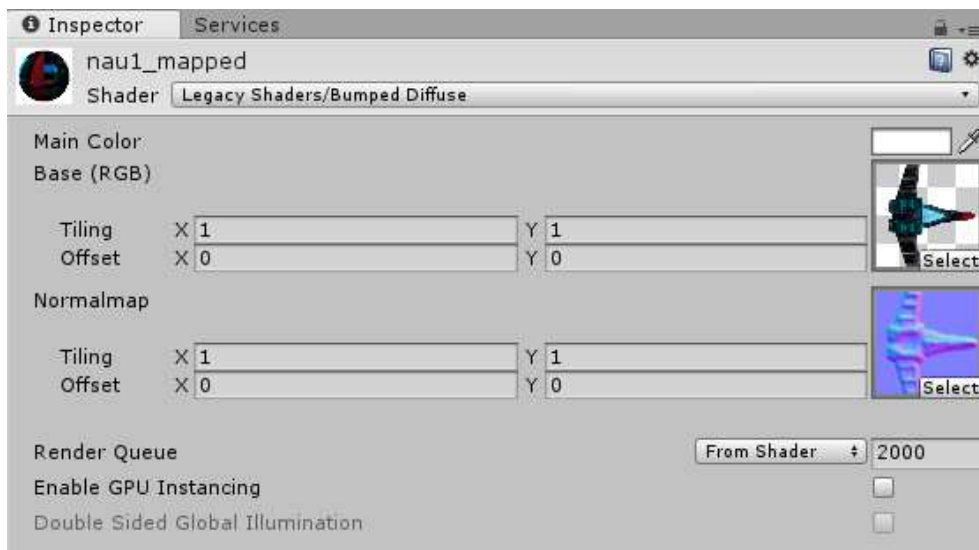
### 3.5.1. Bump Mapping

Aquesta tècnica de computació de gràfics permet simular relleus i profunditats en una superfície mitjançant la il.luminació.

En el nostre aplicarem la tècnica a les textures planes de les nostres naus, per poder aconseguir donar efecte de profunditat i que aquestes responguin a la il.luminació, en primer hem de crear una textura amb blanc i negre de la nostra nau on aplicarem ombra, en les zones més elevades aplicarem un amb to més clar i les zones més enfonsades més opac, aquest ha de ser amb escala de grisos.

Un cop hem creat la nostra textura per al bump mapping, hem de crear un material al que afegirem les nostres textures, el tipus de shader que utilitzarem per aquest material és el Bumped Diffuse que és l'efecte que volem donar al nostre material, un cop creat el material tenim que afegir el sprite de la nostra nau i la textura que hem creat amb l'ombra.

## MATERIAL AMB BUMP MAPPING



### 3.5.2. Animacions

En les seves versions anteriors el joc tenia una animació la qual s'utilitzava per a realitzar les explosions de les diferents naus del joc tant les enemigues com les del jugador en la versió final les explosions es generen mitjançant el sistema de partícules, d'altra banda el codi que fa referència a les explosions està comentat en els scripts, aquesta ha sigut una de les últimes modificacions que s'han fet al joc.

Aquesta decisió s'ha pres, ja que l'estil de l'explosió no acompanyava amb el joc i donava problemes de visualització amb el bump mapping de les naus.

D'altra banda s'ha de dir que les naus del jugador i l'enemic boss canvien el seu sprite, mitjançant el codi en els scripts .

Els Enemics Mina i Mina cluster tenen una animació, mitjançant uns varis sprite i les animacions aconseguim donar l'efecte de rotació als dos enemics, però aquesta animació és molt senzilla.

#### ANIMACIO ENEMIC MINA I CLUSTER



### 3.5.3. Sistema de Partícules

El sistema de partícules ens permet crear una gran varietat d'efectes mitjançant l'ús de sprites en el nostre cas, ja que treballem en un joc 2D, el sistema de partícules és molt adequat per crear efectes de moviment com poden ser núvols, flames atacs màgics entre d'altres.

En aquest projecte hem creat una sèrie d'efectes els quals tenen diferents funcions, és la propulsió de les naus, les explosions, l'impacte dels projectils a una nau i els projectils de les naus del jugador.

#### 3.5.3.1. Projectil nau jugador N1 i N2

Per aquest efecte hem creat tres tipus de partícules diferents les quals tenen diferents funcions, el tipus de shader utilitzat en aquestes els particle Additive de manera que el material que creem mitjançant les textures utilitzades s'adhereixi a les partícules.

Així doncs trobem el sistema de partícules principal hem de dir que aquests són GameObjects el qual és el pare i conté dos sistemes de partícules més, aquest sistema conté el material amb la textura del projectil la qual és una transparència amb la forma difusa del projectil en blanc ombrejat, per a poder configurar el nostre efecte hem de configurar els paràmetres del particle sistema com són el color l'emissió de partícules, el color sobre el temps la rotació i el render al qual afegirem el nostre material amb la textura del projectil.

Un cop realitzat això ja tenim un efecte que sembla un projectil però encara és molt pla, així doncs crearem una còpia del sistema de partícules original i el situarem dins d'aquest en aquests efecte canviarem el material que es renderitza per una textura semblant a

l'anterior però redona de manera que crearem un punt de llum com en el cas anterior modificarem les propietats perquè aquest punt de llum creixi un cop es creï el projectil i disminueixi.

Un cop fet aquests 2 passos crearem el trails i la cua del projectil mitjançant una forma de con el ús de l'opció trails i color overlifetime i cover over trail en els dos sistemes de partícules.

### **3.5.3.2. Efecte impacte projectil**

Aquest efecte és l'efecte que es produeix quan un projectil impacta amb un cos ja sigui un projectil, nau o enemic. Aquest efecte és un efecte compost com l'anterior per varis sistemes de partícules tot i que aquest és més senzill, ja que sols utilitza un material amb textura, per a crear un punt de llum com amb el cas anterior i el seu funcionament és més simple, aquest efecte no té moviment i el seu temps d'actuació és ràpid.

Les altres parts són partícules amb diferents colors i renderitzats per aconseguir l'efecte desitjat, cal dir que aquest efecte treballa amb una malla esfèrica, en la qual es renderitzen les partícules perquè apareguin amb totes direccions fent una forma esfèrica.

### **3.5.3.3. Explosió**

Aquest efecte i el següent ha estat descarregat dels Assets d'Unity<sup>[8]</sup>, és gratuït i de lliure ús creat pels propis desenvolupadors d'Unity. Així i tot el seu funcionament és molt paregut als sistemes de partícules anteriors.



#### **3.5.3.4. Propulsió de la Nau**

Aquest efecte de flama també forma part del paquet d'efectes anterior [8], és molt senzill el qual s'ha modificat per adapta'l a les necessitats del joc, aquest efecte emet partícules en forma de flama i el seu color varia segons el temps.

#### **3.5.4. Altres Efectes**

##### **3.5.4.1. Escut de les naus**

Aquest no és un sistema de partícules sinó un sistema de shader que ja té l'Unity preparat. Aquest efecte està compost d'un material el qual li hem afegit dues textures diferents un per cada efecte de shield diferent, el tipus de shader el qual ens permet realitzar aquestes funcionalitats Unlit//ShieldFX, aconseguim els efectes d'escut de les naus.

##### **3.5.4.2. Raig laser**

El raig làser no és un efecte pròpiament dit, és simplement un line render al qual hem afegit un material amb una textura la qual té un gradient del color i aspecte desitjat per al làser, aquest material s'adhereix al line render amb un shader del tipus Particles/additive.

### 3.5.4.3. Il·luminació

Aquesta implementació no estava, prevista al joc s'ha realitzat a última hora per aprofitar i millorar els efectes que la llum crea en interactuar amb el bump mapping de les naus i aconseguir un millor efecte visual.

S'han afegit diferents punts d'il·luminació al joc. En primera instància tenim una il·luminació estàtica, a la part superior del background això ens aporta que els cossos de les naus tinguin punts amb il·luminació i d'altres amb ombres aconseguint una sensació major de volum.

Les altres il·luminacions que s'han afegit al joc no són estàtiques, apareixen i desapareixen. Trobem il·luminació en les explosions, i els projectils del jugador i el làser. Aquests tenen il·luminació i quant es generen produeixen un punt d'il·luminació del color de l'efecte, amb una intensitat adequada a les necessitats de l'efecte, s'aconsegueix un resultat molt bo i millora l'estètica global del joc.

## 4. Millores del Joc

Sempre es pot millorar un producte i és molt difícil donar-lo per acabat, normalment les millores han de millorar algun dels aspectes del joc, com potser els aspecte audiovisual, la jugabilitat, noves característiques entre d'altres...

Una de les implementacions que hem va restar fer al joc, és que els enemics es tornessin més forts a mesura que el joc avança, si que el joc té una curva de dificultat gradual, però els enemics sempre són igual de forts, en el cas del Boss cada cop que apareix és mes fàcil ja que la nostra nau té més potencia de foc.

Possibles millores del projecte, treballar en l'aspecte gràfic del joc afegint més animacions, com poden ser animar el moviment de les naus, crear més varietat d'explosions.

Enriquir el joc amb més elements, més implementacions d'enemics, i afegir-los a un nivell de joc personalitzat on es controla l'aparició dels enemics i com aquests es distribueixen en el mapa, per així poder crear diferents nivells de joc, tot i que això ja seria un altre joc diferent.

Una altra de les millores que s'han tingut en compte i no s'han dut a terme degut a la manca de temps i la complicació que comporta, és el treball amb intel·ligència artificial, per a poder crear enemics amb comportaments intel·ligents i més complexes. Actualment s'està treballant molt per intentar recrear bots o autòmats que juguin al nivell del jugador humà, crec que aquesta àrea és molt interessant i en pocs anys veurem moltes novetats en aquest àmbit.

També es va plantejar al principi del projecte treball amb cossos 3D ja que aquest aportarien un nivell de qualitat superior al joc i permeten animar els cossos realísticament aconseguint resultats de gran qualitat.

## 5. Conclusions del Projecte

Durant la consecució d'aquest projecte s'ha intentat anar un pas sempre més enllà i intentar millorar el producte en cada entrega de manera qualitativa, a mesura que he anat realitzant les diferents entregues també ha millorat els meus coneixements amb els diferents programaris i el motor Unity, de manera que s'han intentat buscar noves solucions o millorar les que ja estaven implementades, sempre intentant millorar el joc i la jugabilitat d'aquest.

En aquest projecte s'han intentat utilitzar els recursos treballats durant el grau multimèdia, els quals han servit de base per a realitzar aquest projecte d'altra banda he de dir que la base que ja tenia de Unity m'ha permès utilitzar el temps en intentar aportar millores al joc i elements interessants i que dotesin al joc de jugabilitat i un aspecte distintiu, fet que crec que he aconseguit.

D'altra banda Unity ofereix un gran nombre de possibilitats a l'hora de desenvolupar jocs, també ens ofereix molt bones eines i un motor potent, treballar amb Unity és molt còmode i és una gran eina crec per iniciar-se en el món del disseny i creació de Videojocs, però a l'hora ens ofereix els mitjans per aconseguir resultats professionals, tenim clars exemples de jocs molt populars que ho coroboren com poden ser Rus' per a la plataforma Steam o Heartstone de l'empresa Blizzard, tots dos amb

mecàniques molt dispars, però amb gran quantitat de seguidors.

Durant el projecte s'ha hagut de realitzar molta feina de documentació la qual m'ha permès obtenir molts coneixements nous en les diferents àries del projecte. La gran popularitat d'Unity i plataformes com Youtube ajuden a trobar moltes fonts d'informació, tutorials, explicacions, d'altres per a poder realitzar les diferents solucions implementades al joc.

Voldria remarcar que el meu objectiu principal era el d'aconseguir un videojoc el qual a mi hem fos divertit i crees un repte per al jugador, també aquest projecte és un petit homenatge als inicis dels videojocs i a un gènere tan important en la història com són els shot em up.

Per a finalitzar m'agradaria comentar que els últims dies del projecte s'ha investigat sobre els bump mapping i he pensat que seria un valor afegit al joc que en millora molt l'aspecte visual i l'experiència de joc. Aquest fet ha comportat que s'hagin realitzat un gran nombre de canvis i millores d'última hora al joc, com és el d'afegir il·luminació per aprofitar els efectes d'il·luminació que podem aconseguir amb el bump mapping, i realitzar totes les màscares a les naus del joc.

## 6. Glossari

**App:** fa referència a un programa informàtic dissenyat per aplicacions mòbil.

**bitmap:** o imatge de mapa de bits és una estructura de fitxers de dades que representa una retícula rectangular de píxels o punts de color, la qual s'anomena matriu.

**Fotocomposició:** és la capacitat de compondre pàgines a partir de matrius fotogràfiques o negatius de lletres, per a produir cintes fotogràfiques compostes.

**OpenGL:** és una especificació estàndard que defineix una interfície de programació d'aplicacions multilinguatge i multiplataforma per escriure aplicacions que produeixen gràfics 2D i 3D. La interfície consisteix en més de 250 funcions diferents destinades a dibuixar escenes tridimensionals.

**Direct3D:** és una especificació propietat de Microsoft que defineix una interfície de programació de gràfics 3D és el principal competidor de OpenGL.

**OpenGL ES:** és una variant simplificada del OpenGL ES dissenyada per a dispositius integrats com telèfons mòbils, tablets i videoconsols.

**C#:** Fa referència al llenguatge de programació C Sharp orientat a objectes.

**.Net Framework** de Microsoft dissenyat per emfatitzar la transparència en les xarxes amb independència de plataforma de hardware que permet el ràpid desenvolupament d'aplicacions, és la resposta al creixent negoci en el web.

**Ogg Vorbis** és un còdec d'àudio digital amb pèrdues totalment lliure, llançat al 2002 el qual gràcies a la seva compressió amb pèrdues permet produir una menor quantitat d'informació resultant ens uns fitxers menys pesats.

**Plug-in** és una aplicació informàtica que es relaciona amb un altre per agregar-li una funcionalitat nova, normalment aquesta aplicació és executada per la aplicació principal, també denominats com add-on o extensions.

**Scroller** es denomina desplaçar en anglès, fa referència al moviment en 2D dels continguts que formen el escenari del videojoc o la finestra en la que es mostra una aplicació informàtica.

**Shooter** són jocs d'acció que testejen la velocitat i reflexes del jugador, l'objectiu d'aquests joc es destruir els oponents i anar avançant.

**sprite** és un tipus de mapa bits que representa un objecte amb moviment mitjançant diferents imatges d'aquest, aquest van ser molt utilitzats en videojocs 2D, actualment no s'utilitzen tant degut als videojocs amb modelats 3D.

**LFO** low frequency oscillation es refereix a un senyal d'àudio amb una freqüència inferior als 20Hz que crea un ritme palpitant, amb un so de *Whao Whao* típic.

## 7. Bibliografia

- [1] <https://www.adobe.com/es/products/photoshop.html> [octubre 2017]
- [2] <https://unity3d.com/es> [gener 2018]
- [3] <https://opengameart.org/content/enchanted-tiki-86> [decembre 2017]
- [4] <https://opengameart.org/content/the-others> [decembre 2017]
- [5] <https://opengameart.org/content/through-space> [maig] [decembre 2017]
- [6] <https://opengameart.org/content/clearside-assimilator> [gener 2017]
- [7] <http://www.dafont.com/es/cgf-locust-resistance.font> [novembre 2017]
- [8] <https://www.assetstore.unity3d.com/en/#!/content/3045> [gener 2017]