

Department 96

Carolina Mengual Manzano

Màster universitari en Disseny i Programació de Videojocs
M7.462 - Treball Final de Màster aula 1

Heliodoro Tejedor Navarro

Joan Arnedo Moreno

6 de juny de 2021



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:	Department 96
Nom de l'autor:	Carolina Mengual Manzano
Nom del consultor/a:	Heliodoro Tejedor Navarro
Nom del PRA:	Joan Arnedo Moreno
Data de lliurament (mm/aaaa):	06/2021
Titulació o programa:	Màster universitari en Disseny i Programació de Videojocs
Àrea del Treball Final:	M7.462 - Treball Final de Màster aula 1
Idioma del treball:	Català
Paraules clau	Videojoc, RPG tàctic, estratègia per torns
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>La finalitat d'aquest treball és comprendre el procés de creació d'un videojoc real en la seva plenitud, tant a nivell creatiu i de disseny com a nivell de desenvolupament.</p> <p>Per fer-ho, s'ha dissenyat un joc de tipus joc de rol tàctic per a ordinador, en el qual el jugador s'enfronta a "la màquina" en un combat per equips, on pot decidir quins membres formaran la seva banda, i a cada ronda podrà indicar on es mourà cada membre i que farà una vegada s'hagi mogut.</p>	
Abstract (in English, 250 words or less):	
<p>The purpose of this work is to understand the process of creating a real video game in its fullness, both at the creative and design level and at the development level.</p> <p>To do this, I have been designing a computer tactical RPG game, in which the player faces "the machine" in a team battle, where he can decide which members will form his band, and in each round, he will be able to indicate where each member will move and what it will do once it has moved.</p>	

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball	1
1.2 Objectius del Treball.....	1
1.3 Enfocament i mètode seguit.....	1
1.4 Planificació del Treball.....	2
1.5 Breu sumari de productes obtinguts	3
1.6 Breu descripció dels altres capítols de la memòria	3
2. Estat de l'art	4
3. Definició del joc	5
3.1. Idea del joc	5
3.1.1. Breu descripció del joc	5
3.1.2. Subgènere i referències a videojocs existents	5
3.1.3. Tipus d'interacció joc-jugador.....	5
3.1.4. Plataforma de destí	5
3.2. Conceptualització	5
3.2.1. Història, ambientació i/o trama.....	6
3.2.2. Definició dels personatges/elements.....	6
3.2.3. Interacció entre els actors del joc.....	7
3.2.4. Objectius plantejats al jugador	7
3.2.5. Concept Art: screenshots, animació... ..	7
4. Disseny tècnic	9
4.1. Eines utilitzades	9
4.1.1. Eines de desenvolupament	9
4.1.2. Altres eines	9
4.2. Assets utilitzats.....	10
4.2.1. Assets de creació pròpia	10
4.2.2. Assets d'altres fonts	11
4.3. Diagrama de classes	12
4.4. Formació de l'equip del jugador	13
4.5. Sistema de canvi de torns	15

4.6. Torn aliat	15
4.6.1. Fase d'escollir moviment.....	16
4.6.2. Primera fase de moviment	16
4.6.3. Segona fase de moviment.....	17
4.6.4. Fase d'escollir acció	18
4.6.5. Fase d'acció	18
4.7. Torn enemic	19
4.7.1. Fase d'escollir moviment.....	19
4.7.2. Fases de moviment	20
4.7.3. Fase d'escollir acció	20
4.7.4. Fase d'acció	21
5. Disseny de nivells.....	22
6. Manual d'usuari	23
7. Conclusions.....	24
7.1. Treball futur	24
8. Glossari	25
9. Bibliografia.....	26

Llista de figures

Il·lustració 1: Diagrama de Gantt.....	2
Il·lustració 2: Icones dels diferents tipus.....	6
Il·lustració 3: Ordre de tipus	7
Il·lustració 4: Screenshot de "Fire Emblem: Three Houses"	8
Il·lustració 5: Icones de tipus de disseny propi.....	10
Il·lustració 6: Altres assets de creació pròpia	11
Il·lustració 7: Diagrama de classes.....	12
Il·lustració 8: Pantalla de selecció d'equip.....	14
Il·lustració 9: Àrea de moviment.....	15
Il·lustració 10: Comparativa línia a destí correcte i incorrecte	16
Il·lustració 11: Comparativa color de l'àrea d'acció	17
Il·lustració 12: Àrea d'acció a distància	18
Il·lustració 13: Línia temporal animació d'acció	18
Il·lustració 14: Àrea d'acció d'un enemic	21
Il·lustració 15: Camp de batalla a l'inici de la partida.....	22

1. Introducció

1.1 Context i justificació del Treball

Amb aquest projecte es pretén crear un videojoc de rol tàctic des de zero, per comprendre de primera mà el procés que comporta la creació d'un videojoc, des de el seu disseny inicial fins la seva implementació final.

S'ha escollit el gènere RPG tàctic ja que està format per diversos nivells, el que permet centrar-se en crear-ne un, que després es pot transformar per crear-ne la resta. A més, aquest gènere sol incloure una intel·ligència artificial per controlar l'equip rival, donant així l'oportunitat de comprendre com es dissenyen i implementen.

1.2 Objectius del Treball

Els objectius principals d'aquest projecte són:

- Creació del camp de batalla on es durà a terme el joc.
- Creació dels personatges (tant aliats com enemics).
- Disseny dels diferents equips d'enemics durant el joc.
- Creació de la pantalla de selecció de membres del nostre equip previ a un enfrontament.
- Disseny i implementació de la resta de menús.
- Desenvolupament de la Intel·ligència Artificial del personatges enemics.
- Disseny i desenvolupament dels enfrontaments (posició inicial dels equips, possibles esdeveniments a meitat d'enfrontament...).
- Implementació dels controls de joc durant els enfrontaments.

1.3 Enfocament i mètode seguit

A l'hora d'enfocar aquest projecte han sorgit diferents qüestions a tindre en compte:

Crear un joc nou o refer un de ja existent?

A l'hora de prendre aquesta decisió es van sospesar les dues opcions, per una banda refer un joc ja existent suposaria estalviar-se tot el temps i recursos que comporta la part més creativa del procés, com el disseny dels escenaris, personatges, armes... o coses tan bàsiques com els controls de joc o els menús, donant així l'oportunitat de centrar-se més en el propi desenvolupament del joc, sobretot en la part de codi.

Però per altre banda, crear un joc des de zero implica dedicar part del temps del projecte en el propi disseny del joc, tot i que el gènere RPG tàctic ja està prou definit com per assolir unes bases comuns en la majoria dels jocs del gènere, com l'existència de torns de joc, els diferents tipus de personatges, etc. Tot i restar temps de desenvolupament, aquesta opció permet veure de forma més real tot el que implica la creació d'un

videojoc, ja que el seu disseny es una part important d'aquest tipus de projecte.

Així que finalment s'ha optat per fer un joc nou (tot i que inspirat en altres títols del gènere), ja que aquesta opció permet obtenir una visió més general però realista del procés de creació d'un videojoc en la seva totalitat, i no només de la part més tècnica del mateix.

Utilitzar assets ja existents o crear-ne de nous?

De forma similar a la qüestió anterior, utilitzar assets ja existents (Asset Store de Unity o Mixamo per les animacions) suposa un estalvi de temps important, ja que la creació de models 3D i les seves animacions és un procés molt laboriós, tant per dissenyar-los com per crear-los.

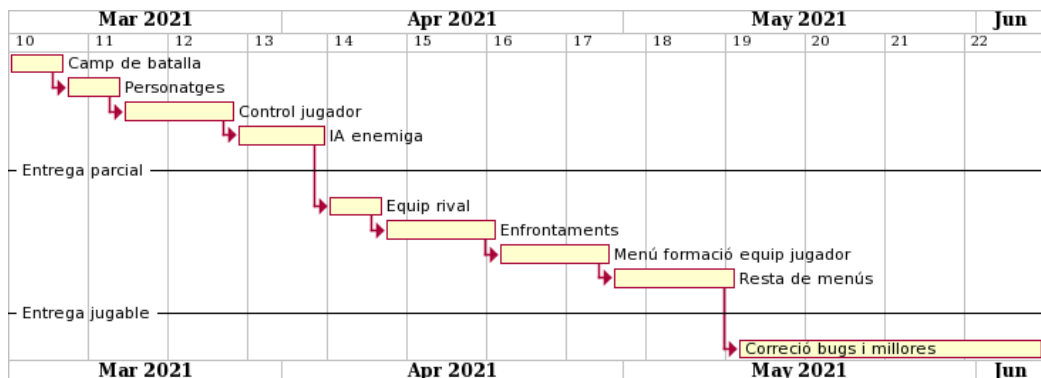
Tot i que utilitzar models i animacions ja creats també implica més rigidesa a l'hora de dissenyar el joc en conjunt, ja que si per exemple s'utilitzen models *LowPoly* per a muntar l'escenari, la resta d'elements del joc (personatges, armes, decoració) també hauran de ser d'estil *LowPoly*.

A diferència de la qüestió anterior, s'ha decidit utilitzar assets ja creats, ja que la creació de models 3D no només implica temps, sinó també coneixements de modelació i animació 3D dels quals no dispo, i que implicarien encara més temps per adquirir-los.

1.4 Planificació del Treball

La planificació s'ha fet sobre els objectius plantejats anteriorment, i quedaria de la següent manera, tot i que es veu més clar al diagrama de Gantt de la II-lustració 1:

- Creació del camp de batalla: 5 dies.
- Creació dels personatges: 5 dies.
- Implementació dels controls de joc: 10 dies.
- Intel·ligència Artificial del personatges enemics: 10 dies.
- Disseny dels equips d'enemics: 5 dies.
- Desenvolupament dels enfrontaments: 10 dies.
- Pantalla de selecció de membres: 10 dies.
- Resta de menús: 10 dies.
- Correcció de bugs i millores: 28 dies.



II-lustració 1: Diagrama de Gantt

1.5 Breu sumari de productes obtinguts

El joc resultat és només una part del que seria el joc real, però en la qual es poden veure els elements clau a la resta del joc, ja que és tracta d'un joc on tots els nivells són semblants, variant el camp i els enemics, però la jugabilitat és sempre la mateixa.

1.6 Breu descripció dels altres capítols de la memòria

A la resta de capítols es parlarà de la part més tècnica del projecte, en concret els capítols seran:

- **Estat de l'art:** anàlisi dels jocs ja existents del mateix gènere.
- **Definició del joc:** breu explicació general del joc creat per poder entendre millor la resta de capítols on s'explica la seva implementació.
- **Disseny tècnic:** part més tècnica del document, on s'explica la implementació del projecte, així com les eines i recursos utilitzades.
- **Disseny de nivells:** explicació del funcionament de la creació dels nivells del joc, tot i que actualment només n'hi ha un de creat.
- **Manual d'usuari:** manual del funcionament del joc, per a que els nous jugadors puguin jugar sense problemes, tot i no conèixer el gènere.

2. Estat de l'art

Els videojocs de rol tàctics, o RPG tàctics, no són els més populars actualment, ja que com es pot veure al rànquing de jocs més venuts a Steam durant el 2020 [1] hi ha molt pocs entre els més venuts, tot i que hi ha títols com *Total War: THREE KINGDOMS* [2] o *Age of Empires II: Definitive Edition* [3] encara es mantenen a la llista, però encara que no són exactament iguals que el joc d'aquest projecte, sí que són del mateix gènere.

Per altra banda, si mirem una llista dels jocs més populars d'aquest gènere com la de Vandal [4], podem trobar jocs més semblants al nostre, com els de la saga *Fire Emblem* [5] o *XCOM* [6].

Entenem com a similar el fet de que els nivells són camps de batalla on s'enfronten dos equips, un d'ells controlat pel jugador i l'altre per una intel·ligència artificial.

A més, aquests equips estan formats per personatges individuals, no per tropes, que són de diferents tipus, i per tant tenen característiques diferents, i normalment existeix un "ordre de tipus", on uns són més forts o més dèbils contra altres, formant un cercle, i amb els quals el jugador pot crear estratègies més complexes, depenent dels tipus del seu equip i de l'equip rival.

Si ens fixem en les entregues més antigues dels jocs mencionats abans, podem veure com el gènere no ha canviat gaire amb el transcurs del temps, més enllà de millores gràfiques evidents, la pròpia jugabilitat dels jocs es conserva pràcticament igual. Això es pot deure a que les bases d'aquest tipus de joc són molt concretes, i no donen gaire joc a modificacions, sense perdre l'essència d'aquests jocs.

3. Definició del joc

3.1. Idea del joc

En aquest apartat es parla de la part més general del joc, com el seu gènere o les seves característiques.

3.1.1. Breu descripció del joc

El projecte és un joc d'estratègia o tàctic on dos "equips" s'enfrontaran en un camp de batalla estil quadricula amb diferents nivells/mapes i amb una petita història que els uneixi i doni coherència a tot el joc.

Cada equip està format per diferents guerrers, que es poden repetir i que tenen diferents habilitats i competències, explicades més endavant.

3.1.2. Subgènere i referències a videojocs existents

La inspiració principal són els jocs de la saga *Fire Emblem* [5] i *Advance Wars* [7], per això el gènere principal del joc és "**Joc de rol tàctic**".

3.1.3. Tipus d'interacció joc-jugador

Ja que els nivells són enfrontaments entre dos equips, un és controlat pel jugador, i l'altre "per la màquina" gràcies a una intel·ligència artificial no gaire complexa però funcional.

En el cas del jugador, durant el seu torn podrà escollir en quin ordre actuen els seus guerrers, on es mourà cadascun i quins acció farà (atacar, curar-se, protegir-se, etc.).

3.1.4. Plataforma de destí

Originalment es tracta d'un joc per ordinador, ja que és la plataforma que dona més lloc a l'hora de treballar amb ella, però en un futur es miraria d'adaptar-lo a més plataformes, tenint en compte que aquest tipus de joc es pot adaptar de forma més o menys senzilla a altres tipus de controls, com una pantalla tàctil o un comandament de consola.

3.2. Conceptualització

En aquest punt ja es parla de forma més concreta del joc dissenyat, explicant la seva història o els seus personatges.

3.2.1. Història, ambientació i/o trama

El joc es centra en una ciutat, on de cop han començat a passar coses estranyes, el protagonista descobrirà que això ja havia passat fa molts anys, i que l'heroi que va salvar el món era un avantpassat seu, però no ho va fer sol, tenia tot un equip amb ell. Així que el nostre protagonista es disposa a reunir un nou equip per salvar el món de nou.

3.2.2. Definició dels personatges/elements

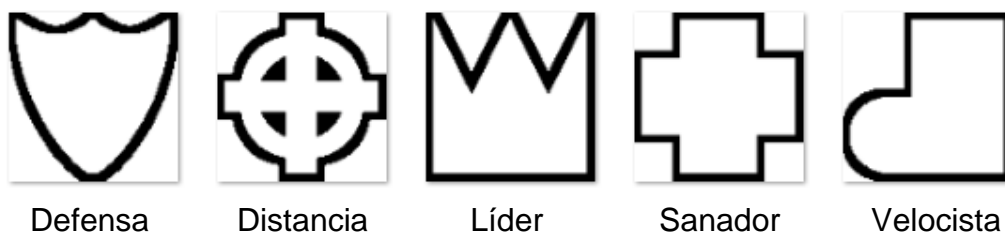
Al joc hi ha 5 tipus de personatges, aquests tipus tenen diferents característiques que es poden veure i comparar de forma més clara a la Taula 1:

Taula 1: Característiques dels tipus de personatges

Nom	Descripció	Força	Velocitat*	Vida	Rang d'acció
<i>Líder</i>	Líder de l'equip, només pot haver un per equip.	Alta	Mitja	Mitja	1
<i>Sanador</i>	No ataca, però cura als seus companys (1 per torn).	-	Mitja	Poca	2
<i>Distància</i>	No pot atacar cos a cos però sí a distància sense tindre en compte obstacles.	Alta	Mitja	Poca	2 a 5
<i>Defensa</i>	Té poca mobilitat, però molta vida, pot ser una primera línia de defensa.	Mitja	Poca	Alta	1
<i>Velocista</i>	Capaç de recorre grans distàncies, però amb poca força d'atac.	Poca	Alta	Mitja	1

* entenem per velocitat la quantitat de caselles que es pot moure en cada torn.

Aquets tipus estan representats per 5 icones, que es poden veure a la Il·lustració 2, que ajuden al jugador a veure en un cop d'ull de quin tipus són els personatges de la partida.

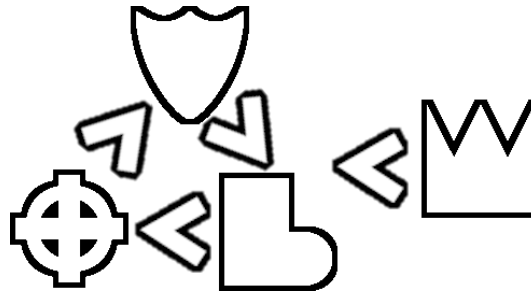


Il·lustració 2: Icones dels diferents tipus

3.2.3. Interacció entre els actors del joc

Aquests tipus no només tenen característiques diferents, sinó que són més forts o més dèbils contra altres tipus, com es pot veure de forma més clara a la Il·lustració 3, que forma part de la UI del joc, els líders són forts contra tots els tipus, excepte sanadors, i ells són dèbils contra el líder.

També es pot veure com els sanadors no entren a l'esquema de tipus, això es deu a que com no poden atacar, no seria just que fossin dèbils contra altres tipus.



Il·lustració 3: Ordre de tipus

La resta de tipus es relacionen de la següent manera:

- A distància: forts contra defensa, però dèbils contra velocistes.
- Defensa: forts contra velocistes, però dèbils contra distància.
- Velocistes: forts contra distància, però dèbils contra defensa.

3.2.4 Objectius plantejats al jugador

L'objectiu principal del jugador és salvar el món, per fer-ho primer haurà de reunir el seu antic equip, però pel camí es trobarà amb grups d'enemics que haurà de vèncer amb ajuda dels companys que ja hagi reunit.

Una vegada els hagi trobat a tots, serà el moment de la gran batalla final.

3.2.5. Concept Art: screenshots, animació...

La zona de combat serà semblant a la dels jocs de *Fire Emblem* [5], un camp obert, on estaran situats els membres dels dos equips, que seran models 3D animats, com es pot veure a la Il·lustració 4.



Il·lustració 4: Screenshot de "Fire Emblem: Three Houses"

Durant el torn del jugador, quan seleccioni algun membre de l'equip, es mostraran en color les possibles caselles on es pot moure aquest personatge. I quan faci una acció com atacar o curar-se el personatge farà una petita animació per mostrar-ho.

4. Disseny tècnic

4.1. Eines utilitzades

Per dur a terme aquest projecte s'han utilitzat diferents eines de desenvolupament, les quals detallaré a continuació, juntament amb la justificació de la meva elecció.

4.1.1. Eines de desenvolupament

L'eina de desenvolupament principal ha sigut Unity [8] juntament amb l'editor de codi C# Rider [9] de JetBrains.

Per escollir el motor de desenvolupament es van estudiar altres opcions com Unreal Engine [10], però finalment es va decidir utilitzar Unity, ja que és l'eina més utilitzada en aquest Màster, i per tant la que es té més per la mà.

En quant a l'editor de codi, es va plantejar l'ús de Visual Studio/Visual Code [11] ja que són les eines predeterminades de Unity (depenent de si es treballa amb Windows o Mac la predeterminada és una o una altra), o Notepad++ [12] per la seva simplicitat, però aquest últim va quedar descartat perquè és més difícil debugar amb ell directament des de Unity.

Però la decisió final va ser utilitzar Rider perquè es tenen més coneixements i pràctica amb els editors de JetBrains que amb els de Microsoft, i per tant és més senzill treballar amb Rider que amb Visual Studio o Visual Code.

4.1.2. Altres eines

En aquest projecte no només s'han utilitzat eines de desenvolupament com a tal, sinó que també s'han utilitzat altres eines per dibuixar, la gestió de codi o per fer el showcase.

En el cas de l'eina de dibuix s'ha utilitzat GIMP [13], ja que estic bastant familiaritzada amb ella, és gratuïta i permet fer dibuixos d'una forma còmoda i senzilla, sobretot en el meu cas que els dibuixos que he fet han sigut bastant simples, però necessitava de certes eines que GIMP m'ofereix i altres programes de dibuix gratuïts no.

Per la gestió de codi s'ha utilitzat GitHub [14], tot i que es va plantejar utilitzar GitLab [15], ja que aquest s'ha utilitzat més durant el Màster, però finalment em vaig decidir per GitHub, perquè també l'he utilitzat molt els últims anys, a més disposa de l'aplicació GitHub Desktop [16], la qual també he utilitzat, i facilita molt la sincronització amb el portal, ja que permet fer els *commits* i *push* amb un parell de clics.

El showcase del projecte s'ha fet amb Google Sites [17], ja que és una plataforma molt senzilla però que permet fer pàgines webs bastant completes i de forma gratuïta, només cal un compte de Google. Inicialment es van plantejar altres plataformes com Blogger [18] o Wordpress [19] que són d'estil blog, però finalment em vaig decidir per Google Sites ja que no calia fer-se un compte nou i és més personalitzable a l'hora de dissenyar l'estructura i el contingut de cada pàgina.

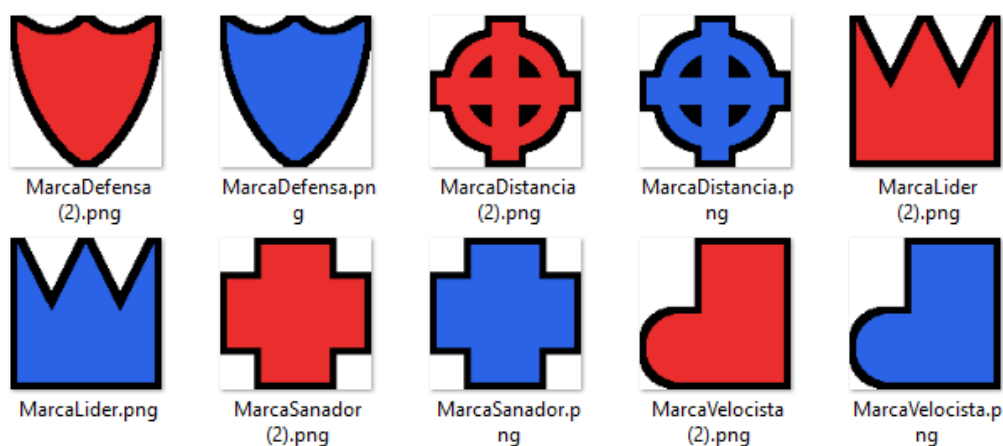
4.2. Assets utilitzats

Per la creació del projecte s'han utilitzat diferents assets d'origens diferents, entre ells els que he dibuixat jo mateixa amb el programa GIMP, com he esmentat a l'anterior apartat.

4.2.1. Assets de creació pròpia

Els assets de creació pròpia han sigut pocs, però necessaris a l'hora de realitzar el projecte, ja que són massa concrets com per trobar-los de forma gratuïta a la xarxa.

Els més visibles són les icones de tipus de jugador, que es poden veure a la II-lustració 5, i que simbolitzen de forma clara però senzilla els 5 tipus diferents de personatges.

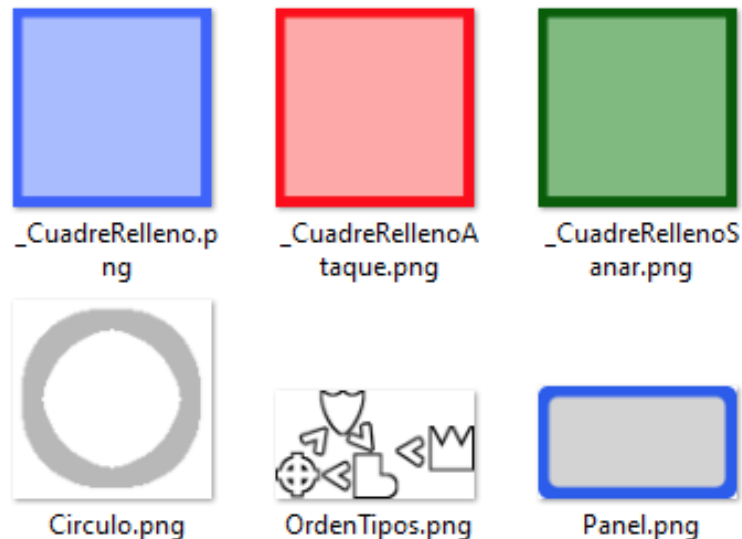


II-lustració 5: Icones de tipus de disseny propi

Inicialment es van fer de color blanc, com es veu al diagrama d'ordre de tipus de la II-lustració 3, però a l'hora d'utilitzar-los per indicar el tipus de cada personatge que hi ha al camp, es van pintar de vermell en el cas dels enemics, i blau en els aliats.

Aquesta elecció de colors es deu a que són els normalment utilitzats per indicar aquest tipus de coses, i per tant els jugadors habituals de videojocs poden reconèixer a que fan referència.

També s'han creat altres assets que formen part del joc, com es veu a la Il·lustració 6, com els quadres que indiquen formen les àrees de moviment, atac i curació, que són pràcticament iguals, però amb colors diferents, ja que indiquen àrees diferents. Els colors s'han seleccionat pensant novament en altres jocs del gènere, per facilitar la comprensió a simple vista dels jugadors habituals d'aquest tipus.



Il·lustració 6: Altres assets de creació pròpia

Per últim, s'han creat elements de UI com el cercle que marca la vida de cada personatge, l'esquema de tipus situat a la cantonada superior esquerra durant la partida o el panell de fons del menú. Aquest últim s'ha creat perquè al paquet d'assets utilitzat per fer els menús que s'especifica al següent apartat, no contenia cap fons que encaixes amb els colors de la resta del joc.

4.2.2. Assets d'altres fonts

Per l'escenari i els personatges s'ha utilitzat el pack *Synty's Polygon City Pack* [20] que inclou una gran quantitat de models per fer una ciutat d'estil LowPoly, a més d'alguns models de personatges del mateix estil. Es va escollir aquest pack tot i ser de pagament perquè el vaig aconseguir a un *bundle* de Humble Bundle [21] a molt bon preu, i el trobo molt complet, tant que s'ha pogut muntar tot l'escenari i els personatges únicament amb els models que inclou.

Ja que l'escenari i els personatges són d'estètica LowPoly, les armes també havien de ser-ho, així que es va buscar un paquet d'assets d'armes d'aquest estil, i es va trobar el *Polygonal Modern Weapons Asset Package* [22] que es pot descarregar de forma gratuïta de l'Asset Store de Unity [23] i ofereix una generosa quantitat d'armes LowPoly de diferents tipus.

Per l'estètica dels menús del joc s'ha utilitzat un altre paquet de l'Asset Store, el *2d Game UI Elements - Complete GUI components* [24], que en

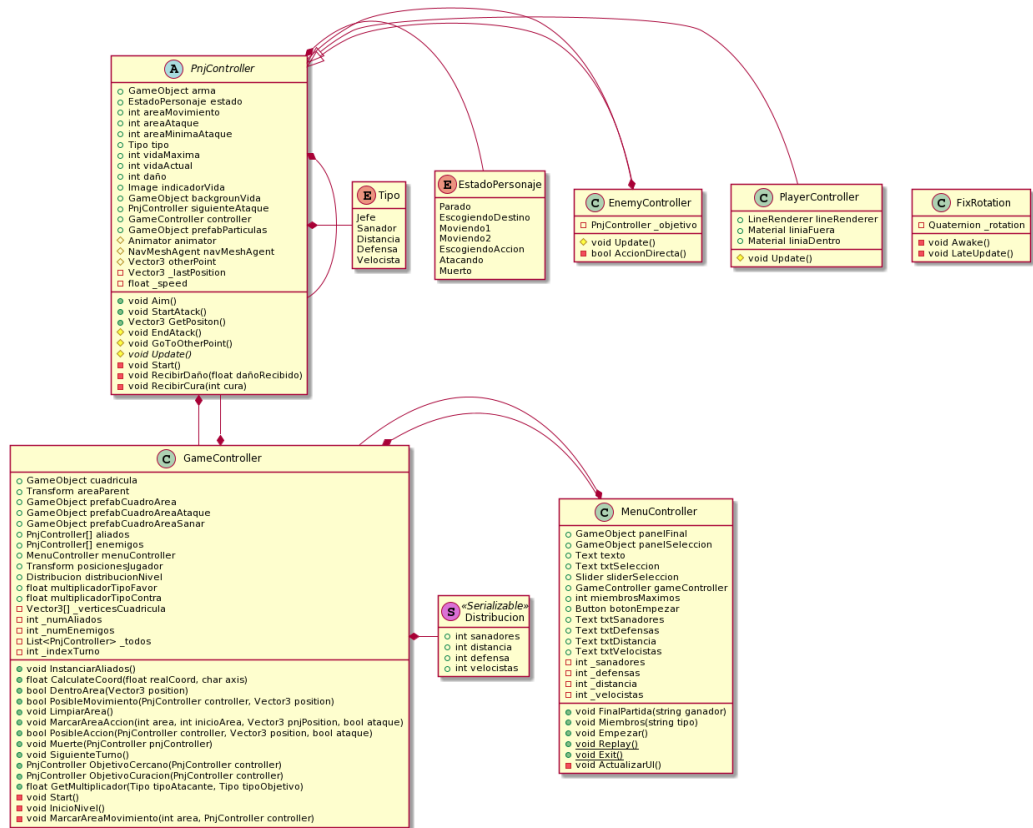
el moment de desenvolupar el projecte era gratuït, tot i que ja no ho és, i que conté una gran quantitat d'element per fer una UI completa i amb un estil senzill que encaixa perfectament amb l'estètica del joc, tot i no ser *LowPoly*.

Per últim, per les animacions dels personatges s'ha utilitzat *Mixamo* [25], que ofereix una gran quantitat d'animacions de tots els tipus, que es poden descarregar, importar als projectes i enllaçar amb els models d'altres paquets, tot i que també es poden descarregar models de la mateixa web, encara que no és el cas en aquest projecte.

4.3. Diagrama de classes

Com es pot veure a la Il·lustració 7, que s'ha generat amb el portal *PlantUML* [26], el projecte té una classe abstracta, *PnjController*, la qual conté els atributs i mètodes comuns entre personatges aliats i enemics, ja que en essència tots els personatges funcionen de forma similar, l'única diferencia es que uns funcionen segons les indicacions del jugador i els altres amb una intel·ligència artificial.

D'aquesta classe hereten dues classes, *PlayerController* i *EnemyController*, les quals sobreescriven el mètode *Update()*, cridant prèviament a l'implementat a la classe mare, i en el qual es troba el codi que correspon al funcionament dels personatges.



Il·lustració 7: Diagrama de classes

També es poden veure dues enumeracions i un struct. Aquest últim s'anomena *Distribucion* i serveix per indicar al *GameController* quants membres de cada tipus té l'equip del jugador, es serializable per poder mostrar les variables d'aquest tipus a l'editor de Unity.

Les enumeracions, *Tipo* i *EstadoPersonaje*, serveixen per aportar informació de cada personatge de forma més accessible, en comptes d'utilitzar strings o int, amb els quals és més fàcil equivocar-se i causar errors.

Per últim, es pot veure que la classe més centralitzada del projecte és *GameController*, això es deu a que es qui controla la jugabilitat general del joc, i les seves funcions són:

- Instanciar els personatges del jugador una vegada a format el seu equip.
- Passar el torn entre els personatges.
- Dibuixar i esborrar les àrees d'atac i acció.
- Calcular el dany que fa un personatge a un altre segons els seus tipus.
- Indicar si una casella es un objectiu vàlid per moure's o fer una acció.
- Indicar a la IA dels enemics els seus objectius més propers, tan per atacar com per curar.

4.4. Formació de l'equip del jugador

Per a que el jugador pugui muntar el seu equip d'una forma fàcil i intuïtiva, a l'inici del nivell se li mostra una pantalla com la que es pot veure a la Il·lustració 8, on pot indicar quants membres de cada tipus vol al seu equip, excepte el líder, ja que és obligatori qui hi hagi exclusivament un, i per tant no té sentit incloure'l a la pantalla de selecció.



II-lustració 8: Pantalla de selecció d'equip

Tota la informació de que es mostra a la pantalla de selecció està representada de dues formes, per exemple el nombre màxim de membres, està indicat tant per text (Miembros restantes: x/y) com de forma més visual gràcies a la barra situada a la dreta del text.

De la mateixa forma, el jugador també pot veure els tipus de membres disponibles, per una banda amb el nom del tipus, i per l'altre amb la icona que els representa, d'aquesta manera el joc és molt més accessible per a persones amb dificultats lectores, ja que amb les icones i la barra obtenen tota la informació necessària per formar el seu equip.

Pel que fa el codi que gestiona el menú de selecció d'equip, està situat a la classe *MenuController*, que s'encarrega d'actualitzar la UI segons l'usuari vagi formant el seu equip, i per últim avisa a la classe *GameController* quan l'equip estigui format, ja que és aquesta classe qui crea les instàncies dels membres de l'equip, i posteriorment comença el joc.

Per crear les instàncies dels membres, es crida al mètode *InstanciarAliados()*, on abans de res s'instancia el líder de l'equip, i després s'utilitza la informació introduïda pel jugador a la pantalla de selecció per crear tants membres de cada tipus com hagi indicat. Aquest mètode recupera de la carpeta *Resources* els prefabs dels diferents tipus, i els acaba de configurar, ja que el seu script de control, *PlayerController*, requereix d'una referència al script *GameController* i al *LineRender* de l'escena actual.

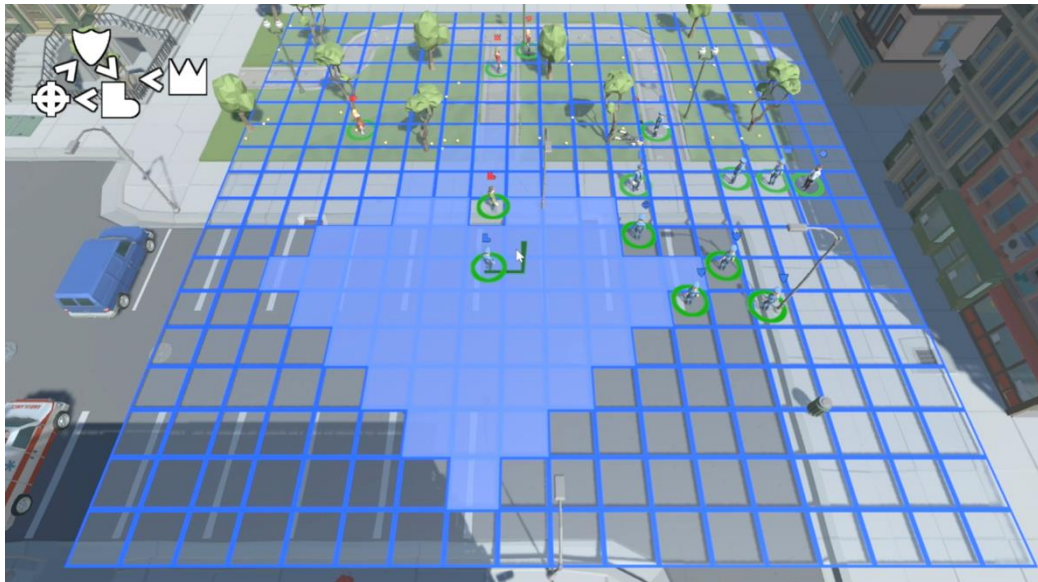
Per situar aquests prefabs s'utilitzen una sèrie de *Transforms* guardats al *GameController* de l'escena, i que indiquen les possibles ubicacions d'spawn dels membres aliats en aquest nivell.

4.5. Sistema de canvi de torns

Durant la partida, el script *GameController* recorre la seva llista de personatges de la partida, on estan inclosos aliats i enemics, per anar passant el torn entre ells, mitjançant el mètode *SiguienteTurno()*.

Cal destacar que els personatges que moren durant la partida no són eliminats d'aquesta llista, es per això que al passar de torn es comprova si el personatge ha mort, en tal cas es passa directament al següent.

Quan s'arriba a un personatge viu es canvia el seu estat a *EscogiendoDestino*, la qual cosa li indica que és el seu torn, a més, en cas que sigui un aliat, es crida al mètode *MarcarAreaMovimiento()* per a marcar les possibles caselles on el jugador podrà dir al personatge que es mogui, com es pot veure a la II-lustració 9.



II-lustració 9: Àrea de moviment

Aquest mètode el que fa és instanciar un prefab d'un quadrat blau a les caselles on es pot moure, utilitzant l'àrea de moviment del tipus de personatge, i evitant les caselles on ja hi hagi algú, sigui aliat o enemic, però que estigui viu, ja que si hi ha un personatge mort es pot ocupar la seva casella.

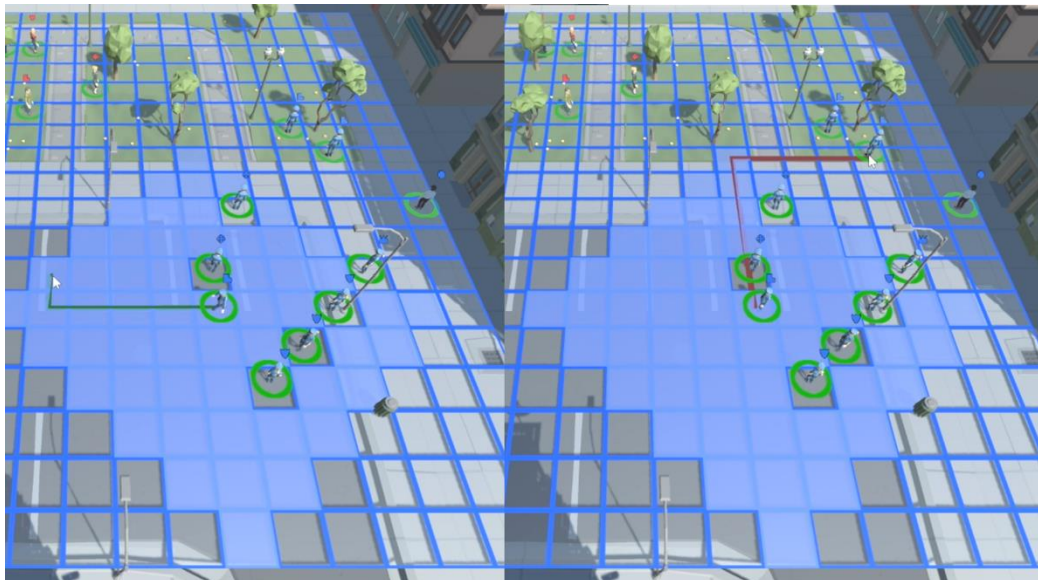
4.6. Torn aliat

Durant el torn de qualsevol dels personatges aliats, es passa per diferents fases, que corresponen als estats de la enumeració *EstadoPersonaje*. Depenent de en quina fase es trobi el personatge, el mètode *Update()* de *PlayerController* farà una cosa o una altra.

4.6.1. Fase d'escollir moviment

Aquesta seria la primera fase per la que passen els personatges aliats, i que correspon a l'estat *EscogiendoDestino*. Durant aquesta fase, per pantalla el jugador pot veure de forma clara totes les possibles caselles on es pot moure, com ja s'ha vist a la Il·lustració 9, a més, segons vagi movent el ratolí per la pantalla veurà una línia des del personatge fins el punt on estigui el ratolí, que li mostrarà quin seria el camí que seguiria el personatge en cas de voler anar a aquell punt, ja que els personatges no es mouen en diagonal, sinó que fan un moviment en forma de L.

També cal destacar que aquesta línia que es dibuixa es de color verd si el destí es possible o vermell si està massa lluny, com es veu a la Il·lustració 10, el que ajuda a l'usuari a veure de forma més clara on es pot moure i on no, juntament amb l'àrea marcada que ja s'ha mencionat anteriorment.



Il·lustració 10: Comparativa línia a destí correcte i incorrecte

Comentar també que si el jugador fa clic a la posició actual del personatge, es considera una opció vàlida, ja que es possible que en aquest torn prefereixi que aquest personatge no es mogui, i passi directament a fer la seva acció principal.

Una vegada el jugador ha fet clic a una posició vàlida, s'assigna aquest punt com a destí final, i el punt entremig com a primer destí, dient al component *NavMeshAgent* que es dirigeixi cap allà, a la vegada que es passa a l'estat *Moviendo1*.

4.6.2. Primera fase de moviment

Aquesta fase correspon a l'estat *Moviendo1*, i durant aquesta el personatge es mou cap al primer punt de destí, el situat a la "cantonada" de L que forma la ruta del personatge.

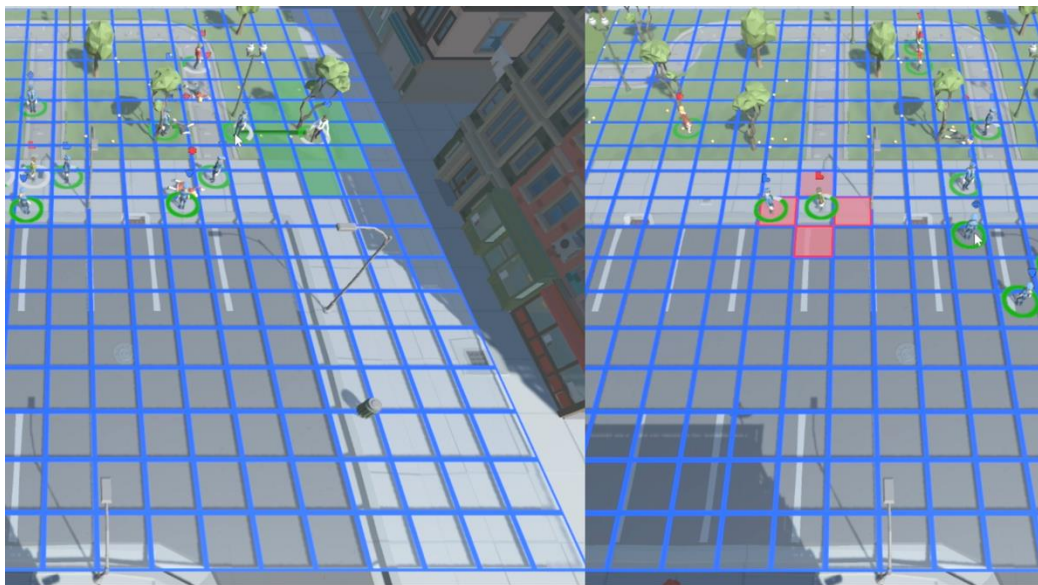
Una vegada el personatge arriba al seu primer destí, es diu al component *NavMeshAgent* que es dirigeixi al segon, i últim, destí.

4.6.3. Segona fase de moviment

Correspon a l'estat *Moviendo2*, i de forma similar a l'anterior comprova quan el personatge ha arribat al seu destí, que en aquest cas és el destí final que el jugador ha seleccionat a la primera fase.

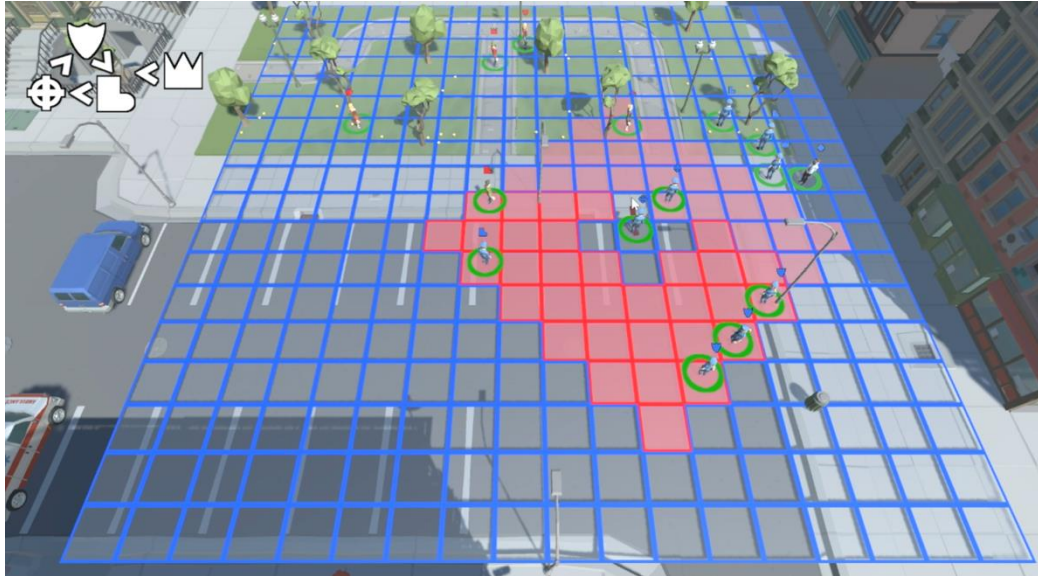
Quan el personatge ja ha arribat es neteja l'àrea de moviment dibuixada anteriorment, i la línia que indicava el seu camí, a més, es dibuixa l'àrea d'acció, gràcies al mètode *MarcarAreaAccion()* de *GameController*.

Aquesta nova àrea es dibuixa de forma similar a la de moviment, però en aquest cas en compte de blau es vermella o verda, com es veu a la Il·lustració 11, segons el tipus de personatge, si és sanador es dibuixarà de color verd, per indicar que la seva acció es curativa i no ofensiva, en cas contrari serà de color vermell, ja que el que farà es atacar.



Il·lustració 11: Comparativa color de l'àrea d'acció

Al dibuixar l'àrea d'acció, a diferència que a la de moviment, es té en compte que alguns tipus de personatges, els que ataquen a distància per ser exactes, no poden atacar a qui tenen massa a prop, i per tant aquesta àrea no ha d'incloure les caselles més properes al personatge, com es veu a la Il·lustració 12.



Il·lustració 12: Àrea d'acció a distància

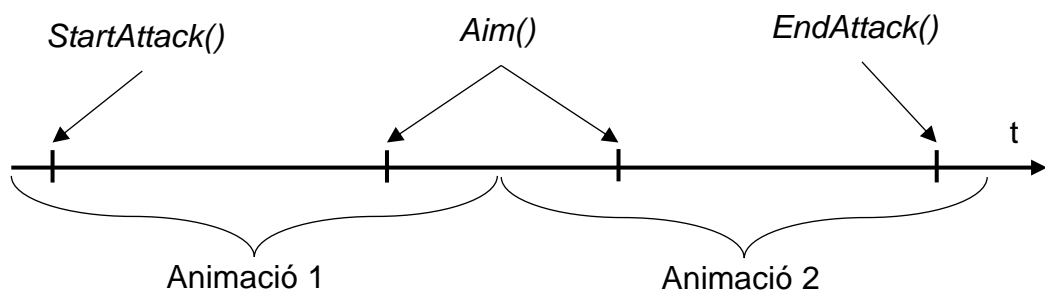
4.6.4. Fase d'escollir acció

Aquesta fase correspon a l'estat *EscogiendoAccion*, a la qual també es mostra la línia verda/vermella que indica al jugador si a la casella a la que té el ratolí és una casella vàlida per fer la seva acció, però en aquest cas no totes les caselles marcades per l'àrea són caselles vàlides, ja que si no hi ha cap personatge al que poder fer-li l'acció no es pot considerar vàlida.

Una vegada el jugador selecciona una casella vàlida, s'esborra la línia, el personatge es gira cap al seu objectiu, es guarda aquest objectiu, i comença l'animació d'acció. En canvi, si el jugador no té a l'abast una casella vàlida, o no vol fer una acció aquest torn, pot seleccionar la casella on està el personatge per passar de torn.

4.6.5. Fase d'acció

Correspon a l'estat *Atacando*, i durant aquesta fase el control es fa mitjançant esdeveniments provocats per les animacions, com es pot veure a la Il·lustració 13, i que recull la classe *PnjController*.



Il·lustració 13: Línia temporal animació d'acció

A l'inici de l'animació es quan es fa realment l'acció, si es sanador es cura al personatge objectiu segons la força del sanador, i si es qualsevol altre tipus de personatge es calcula el dany que farà al seu objectiu, i se li aplica, al fer-ho s'actualitza la vida de l'objectiu, i si arriba a 0 es mor.

Per calcular el dany es multiplica la força del personatge pel multiplicador segons el seu tipus i el de l'objectiu, seguint les següents equacions:

Equació 1: Càlcul de dany si el tipus és a favor

$$danyFinal = dany * 1.5$$

Equació 2: Càlcul de dany si el tipus és en contra

$$danyFinal = dany * 0.5$$

Equació 3: Càlcul de dany si els tipus són neutres entre si

$$danyFinal = dany$$

El següent esdeveniment està situat aproximadament a la meitat de la primera animació, i es repeteix a la meitat de la segona, ja que correspon a desenfundar i enfundar l'arma, el que equival a mostrar-la o no, excepte si es sanador, que no té cap arma.

Per últim, a l'esdeveniment que es produeix pràcticament al final de la segona animació, s'esborra l'àrea d'acció que s'havia dibuixat prèviament a la fase d'escollir acció i es passa al següent torn.

4.7. Torn enemic

De forma similar al control del torn aliat, tot es fa mitjançant els diferents estats relacionats amb l'enumeració *EstadoPersonaje*, i la sobreescritura del mètode *Update()*, però en aquest cas no es requereix la interacció de l'usuari, ja que les decisions les pren la intel·ligència artificial programada.

4.7.1. Fase d'escollir moviment

En arribar a aquesta fase, l'associada amb l'estat *EscogiendoDestino*, el primer que es comprova es si des de la posició actual es pot fer directament l'acció, ja que en tal cas ens saltaríem la resta de fases i passaríem directament a fer l'acció.

En cas contrari, es passa a aproximar-se a l'objectiu més proper, però primer s'ha de saber quin es aquest objectiu, per fer-ho es fa una crida a la classe *GameController*, el mètode que es crida depèn del tipus del personatge, ja que si és sanador es cridarà al mètode *ObjetivoCuracion()* per trobar al seu aliat amb menys vida, però en cas contrari, es crida a *ObjetivoCercano()*, per obtenir al seu enemic més proper.

Una vegada fixat l'objectiu, el següent pas es decidir on es mourà el personatge per poder atacar o curar al seu objectiu, per fer-ho es tenen

en compte totes les possibles caselles de destí del personatge, calculant per a cadascuna el camí que recorreria (en forma de L, al igual que els personatges del jugador), la distància a la que es quedaria de l'objectiu i si el podria atacar/curar des de aquella posició, ja que en el cas dels que ataquen a distància si estan massa a prop no poden atacar.

Mentre es fan aquests càlculs es va guardant la informació de la millor casella en aquell moment, considerant com a millor la que es queda més a prop de l'objectiu, però li permet fer l'acció. Si dues caselles deixen al personatge a la mateixa distància del seu objectiu, es considera millor la que faci recorre menys distància al personatge, per tal de que no faci voltes innecessàries.

Una vegada decidida la casella de destí, es comunica al *NavMeshAgent* el seu primer destí, que serà el vèrtex del camí en forma de L que seguirà, i es passa al següent estat.

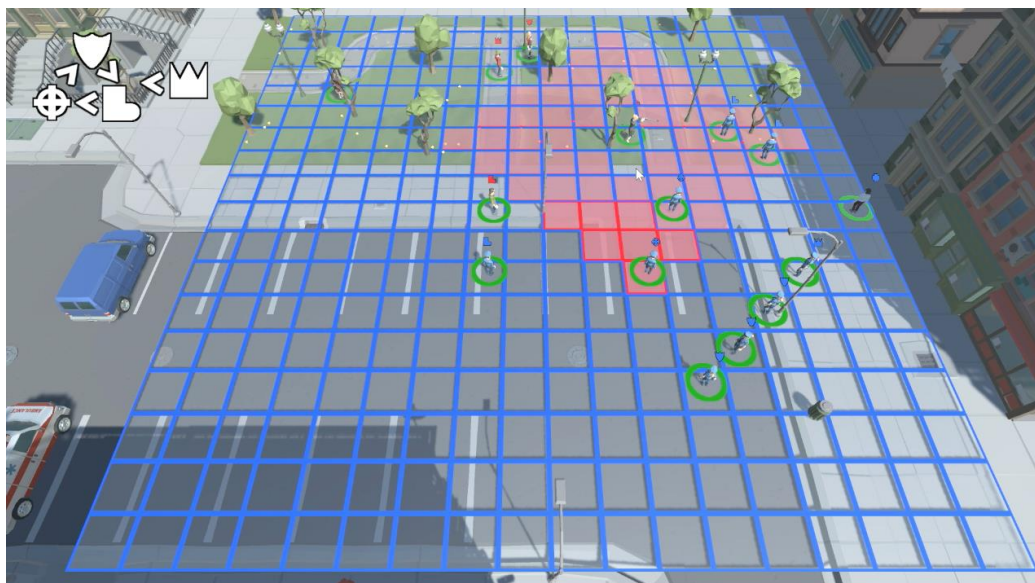
4.7.2. Fases de moviment

Aquestes fases corresponen als estats *Moviendo1* i *Moviendo2*, i són pràcticament iguals que les dels personatges del jugador, ja que en el cas de la primera espera a que el personatge arribi al seu primer destí, per canviar d'estat i indicar al *NavMeshAgent* el nou destí, que serà la casella determinada a la fase d'escollir moviment.

En la segona fase simplement espera a que el personatge arribi al seu destí final per canviar de fase.

4.7.3. Fase d'escollir acció

Correspon a l'estat *EscogiendoAccion*, i el primer que fa es dibuixar l'àrea d'acció del personatge, amb ajuda del *GameController*, per a que el jugador pugui veure de forma clara l'àrea dels enemics, com es veu a la Il·lustració 14, per poder prendre millors decisions durant el seu torn.



Il·lustració 14: Àrea d'acció d'un enemic

Tot seguit es comprova si es pot fer l'acció sobre l'objectiu, ja que es possible que en aquest torn encara estiguem massa lluny de l'objectiu, encara que ens haguem aproximat el màxim, i per tant no es pot realitzar l'acció en aquest torn.

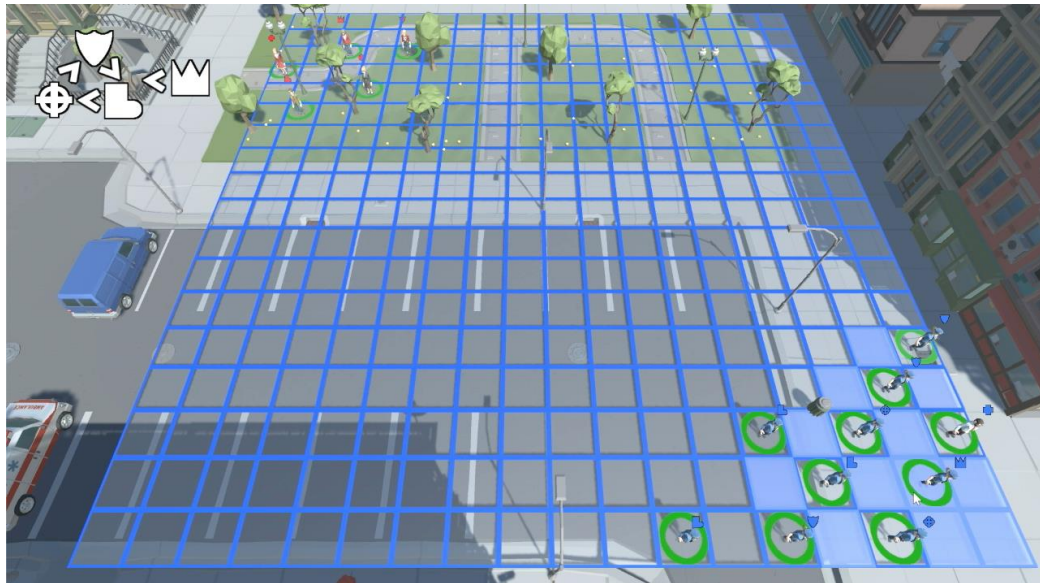
Una vegada ens hem assegurat que l'acció es pot dur a terme, es canvia d'estat i s'inicia l'animació d'acció. Si no es pot fer l'acció es crida a l'esdeveniment de final d'atac.

4.7.4. Fase d'acció

Aquesta fase, que correspon a l'estat *Atacando*, funciona igual que en el cas dels personatges del jugador, mitjançant esdeveniments llençats per les animacions d'acció, com es veu a la Il·lustració 13.

5. Disseny de nivells

El camp de batalla consta d'un escenari urbà, amb una quadricula per la que s'aniran movent els personatges durant el transcurs de la partida, i en la qual es marcaran les àrees de moviment i acció de cada personatge en el seu torn, com es veu a la **¡Error! No se encuentra el origen de la referencia.** Il·lustració 15.



Il·lustració 15: Camp de batalla a l'inici de la partida

En aquest camp de batalla, durant el torn del jugador també es dibuixaran una línia guia per mostrar el camí que seguirà el personatge, com ja s'ha explicat amb més detall a la secció 4.6. Torn aliat.

Tot i que es facin més nivells, els camps de batalla sempre seran del mateix estil, amb una quadricula sobre la que jugar, les úniques diferències poden ser la grandària y el fons sobre el que es situa la quadricula.

6. Manual d'usuari

Per iniciar el joc només cal executar el .exe, en aquest moment s'inicia el joc, i ja es pot escollir els membres de l'equip, per fer-ho, s'utilitza el ratolí, ja que tot el menú de formació d'equip està format per botons.

Durant la partida, a cada torn aliat, el jugador veurà l'àrea de moviment del personatge, i amb el ratolí podrà veure el camí que seguiria per anar a cada casella, i si es possible anar-hi. Una vegada decidida la casella, amb un simple clic es marcarà aquella casella com a destí, però si el jugador volgués saltar-se el torn de moviment, només cal que faci clic a la casella actual del personatge.

Una vegada el personatge arriba al seu destí, és el moment d'escollir l'acció a dur a terme. En aquest moment el jugador podrà veure l'àrea d'acció del personatge, vermella si es d'atac o verda si es de cura. De la mateixa forma en que ha escollit on moure's ara podrà escollir a qui fer-li l'acció, fent clic a la seva casella, o si es vol passar de torn a la casella del personatge actual.

La partida acaba en el moment en que un dels equips es quedi sense membres, considerant guanyador a l'altre equip.

7. Conclusions

Durant la realització d'aquest Treball de Final de Màster, s'ha pogut comprovar com de complex és el desenvolupament d'un videojoc, ja que no només s'ha de programar el seu funcionament, sinó que s'ha de dissenyar tot el joc, buscant una estètica consistent, una jugabilitat divertida, una dificultat que suposi un repte pel jugador, però sense frustrar-lo, i sobretot que no tingui moments avorrits, ja que si un videojoc es torna avorrit deixa d'interessar als jugadors, i probablement l'acabin abandonant sense acabar-lo.

Pel que fa els objectius, la majoria s'han pogut complir, però d'una forma un tant adaptada, ja que per exemple es pretenia crear un parell de nivells, però finalment per falta de temps només s'ha pogut fer un. També es pretenia fer un menú d'opcions i inici de joc, però tampoc ha sigut possible, per la mateixa raó.

Per tant els objectius generals s'han pogut dur a terme, però de forma reduïda a la pensada inicialment.

De forma relacionada al que s'acaba de mencionar, la planificació s'ha pogut dur a terme correctament, però perquè es va decidir reduir la complexitat dels objectius, posant per davant la qualitat que la quantitat.

7.1. Treball futur

Com a treball futur es pretén crear més nivells, tants com per poder-se considerar un joc complet, però per això també cal incloure una historia, sinó seria un seguit de nivells sense cap consistència ni fil conductor, per tant això també es considera un punt pendent en el desenvolupament final del joc.

Com a millores, es pretén millorar la IA dels enemics, ja que actualment es funcional, però encara es podria afinar una mica més, sobretot per a que es tinguessin en compte els tipus dels jugadors, a l'hora d'escollir els objectius d'atac, però sense perdre de vista la seva distancia, per tant s'hauria de trobar un equilibri entre els dos factors.

Per fer el joc més atractiu visualment també es voldria incloure una UI en la que el jugador veies de forma clara l'ordre en la que actuarà cada personatge (tant aliats com enemics), actualitzant-se cada torn per veure remarcat el personatge que té el torn actual. També es podria incloure la vida dels personatges en aquesta UI, ja que tot i que ara es pot veure als peus de cadascun, quedaria més clar i net a la nova UI.

8. Glossari

Asset: són els objectes creats especialment per ser utilitzat a Unity, poden ser models 3D, sons, animacions, scripts...

Script: fitxer de codi.

LowPoly: és un estil de modelatge 3D amb un número de polígons molt baix que fa un efecte de poca definició però de forma estètica.

Prefab: és com una plantilla d'un objecte de Unity, que conté tots els seus components, i valors per defecte, s'utilitza per instanciar còpies d'aquest de forma senzilla, sense tindre que configurar-les cada cop.

Instanciar: afegir un objecte a l'escena de Unity, normalment es refereix a afegir un prefab a l'escena durant el propi joc, mitjançant codi.

NavMeshAgent: component de Unity que serveix per fer que un objecte es mogui per l'escena de forma autònoma, ja que s'indica el seu destí i ell mateix busca el camí òptim, evitant els obstacles.

9. Bibliografia

- [1] Els jocs més venuts a Steam al 2020: <https://store.steampowered.com/sale/BestOf2020> (30-05-2021)
- [2] Web oficial de Total War: <https://www.totalwar.com> (30-05-2021)
- [3] Web oficial de Age of Empires: <https://www.ageofempires.com/games/aoeiide/> (30-05-2021)
- [4] Rànquing dels millors videojocs de rol tàctics en totes les plataformes: <https://vandal.elespanol.com/rankings/videojuegos/tactical-rpg> (30-05-2021)
- [5] Web oficial de Fire Emblem: <https://www.nintendo.es/Juegos/Portal-de-Nintendo/Portal-de-Fire-Emblem/Portal-de-Fire-Emblem-1168499.html> (29-05-2021)
- [6] Web oficial de XCOM: <https://xcom.com/es-ES/> (30-05-2021)
- [7] Web oficial de Advance Wars: <https://www.nintendo.es/Juegos/Game-Boy-Advance/Advance-Wars-866532.html> (29-05-2021)
- [8] Web oficial de Unity: <https://unity.com/es> (29-05-2021)
- [9] Web oficial de Rider: <https://www.jetbrains.com/es-es/rider/> (29-05-2021)
- [10] Web oficial de Unreal Engine: <https://www.unrealengine.com/en-US/> (29-05-2021)
- [11] Web oficial de Visual Studio: <https://visualstudio.microsoft.com/es/> (29-05-2021)
- [12] Web oficial de Notepad++: <https://notepad-plus-plus.org> (29-05-2021)
- [13] Web oficial de Gimp: <http://www.gimp.org.es> (29-05-2021)
- [14] Web oficial de GitHub: <https://github.com> (29-05-2021)
- [15] Web oficial de GitLab: <https://about.gitlab.com> (29-05-2021)
- [16] Web oficial de GitHub Desktop: <https://desktop.github.com> (29-05-2021)

- [17] Web oficial de Google Sites: <https://workspace.google.com/intl/es/products/sites/> (29-05-2021)
- [18] Web oficial de Blogger: <https://www.blogger.com/about/?hl=es> (29-05-2021)
- [19] Web oficial de Wordpress: <https://wordpress.com/es/> (29-05-2021)
- [20] Pack d'assets *Synty's Polygon City Pack*: <https://syntystore.com/products/polygon-city-pack> (29-05-2021)
- [21] Web oficial de Humble Bundle: <https://es.humblebundle.com> (29-05-2021)
- [22] Pack d'assets *Polygonal Modern Weapons Asset Package*: <https://assetstore.unity.com/packages/3d/props/weapons/polygonal-modern-weapons-asset-package-53808> (29-05-2021)
- [23] Asset Store de Unity: <https://assetstore.unity.com> (29-05-2021)
- [24] Pack d'assets *2d Game UI Elements - Complete GUI components*: <https://assetstore.unity.com/packages/2d/gui/icons/2d-game-ui-elements-complete-gui-components-192293> (29-05-2021)
- [25] Mixamo: <https://www.mixamo.com/> (29-05-2021)
- [26] Web de PlantUML: <https://plantuml.com/es/> (26-05-2021)