Tales from Nahelm

Ivan Barrio Muñoz
Grau d'Enginyeria Informàtica
Viedojocs

Consultor: Pere Morales Teixidó
Professor: Joel Servitja Feu

09/06/2019

Aquesta obra està subjecta a una llicència de
Reconeixement-NoComercial-CompartirIgual
3.0 Espanya de Creative Commons

FITXA DEL TREBALL FINAL

Títol del treball:	*Tales from Nahelm*
Nom de l'autor:	*Ivan Barrio Muñoz*
Nom del consultor/a:	*Pere Morales Teixidó*
Nom del PRA:	*Joel Servitja Feu*
Data de lliurament (mm/aaaa):	*06/2019*
Titulació o programa:	*Grau d'Enginyeria Informàtica*
Àrea del Treball Final:	*Videojocs*
Idioma del treball:	*Català*
Paraules clau	*SRPG, Unity, C#*

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

L'objectiu d'aquest projecte ha sigut, el desenvolupament i implementació d'un videojoc per a l'entrega del treball de final de grau de l'enginyeria informàtica. Durant el desenvolupament del projecte s'han intentat aplicar conceptes apareguts durant els estudis realitzats.

El títol del joc és "*Tales from Nahelm*". Aquest joc es pot englobar dins de la categoria de joc de rol d'estratègia per torns d'ambientació medieval fantàstica, on es planteja un repte al jugador per aconseguir superar el nivell implementat derrotant a les forces enemigues amb les unitats disponibles per al jugador.

El joc consta amb un menú inicial i un mapa de batalla amb una petita historia introductòria que posa al jugador en situació, i una petita historia final en cas de victòria del jugador. La batalla es realitza en un terreny on s'han posicionat els enemics manualment, les estadístiques dels quals es generen de forma aleatòria cada cop que s'inicia el nivell.

Per al desenvolupament del projecte s'han emprat eines conegudes amb anterioritat, com "Unity" que ha sigut el motor gràfic triat. Encara que era una eina coneguda, s'ha volgut aprofundir en el coneixement de la mateixa dissenyant una Intel·ligència Artificial capaç d'actuar i generar de forma balancejada, per oferir un repte al jugador.

Podem concloure amb una gran satisfacció del treball realitzat, sobretot respecte a l'aplicació dels procediments apresos en assignatures, com Intel·ligència Artificial. Tot i el treball realitzat, el projecte té molt marge de millora i es podria seguir desenvolupant durant els següents mesos.

Abstract (in English, 250 words or less):

This project objective was the development of a videogame for the final project of my computer engineering. During the development I've tried to apply concepts learned from my studies.

The title of the game is "Tales from Nahelm", this game is a Fantasy Medieval SRPG (Strategy Role-Playing Game), where the Player is challenged to overcome the implemented level by defeating the enemy forces with the available units.

The game has an initial menu and a battle map with an introductory story event that puts the Player in situation. If the Player manages to win the game, there's another story event waiting for him. In the battle all the enemies are positioned manually, but their statistics and weapons are random, so never two plays would be the same.

For the project development, I've been using "Unity", a graphical engine that I've used in the past. Although it was a known tool, I've wanted to deepen the knowledge designing an Artificial Intelligence capable to act to offer a challenge to the Player.

I can conclude with great satisfaction with the work done, especially regarding the application of procedures learned in subjects, such as Artificial Intelligence. Despite the work done, the project has much room for improvement and could continue to develop during the following months.

Índex

1. Introducció 1
1.1 Context i justificació del Treball 1
1.2 Objectius del Treball 1
1.3 Enfocament i mètode seguit 2
1.4 Planificació del Treball 3
1.5 Breu sumari de productes obtinguts 5
1.6 Breu descripció dels altres capítols de la memòria 5
2. Idea inicial del projecte i disseny del videojoc. 6
2.1 Idea inicial. 6
2.2 Valoració de plataformes per al desenvolupament. 16
2.3 Eines emprades i Valoració econòmica del projecte 17
2.4 Plataforma de destí. 19
3.Definició del Videojoc 20
3.1 Historia i ambientació 20
3.2 Objectius del nivell 20
3.3 Personatges 20
3.4 Disseny de personatges 21
4.Disseny del Videojoc 26
4.1 Sistema de torns 26
4.2 Control de Personatges 26
4.3 IA 28
4.4 Generació d'enemics bàsics 29
4.5 Batalles 31
4.6 Configuració de la dificultat 33
4.7 Game Over 34
4.8 Diàlegs, Tutorial i Victoria 34
4.9 GUI i menú d'opcions. 35
5.Conclusions 38
4. Glossari 40
5. Bibliografia 41
6. Annexos 42

Llista de figures

Fig1. Diagrama de Gantt 4
Fig2. Fire Emblem 1 (1990) 7
Fig3. Fire Emblem 2 (1992) 7
Fig4. Fire Emblem 3 (1994) 8
Fig5. Fire Emblem 4 (1996) 8
Fig6. Fire Emblem 5 (1999) 9
Fig7. Fire Emblem 6 (2002) 9
Fig8. Fire Emblem 7 (2003) 10
Fig9. Fire Emblem 8 (2004) 10
Fig10. Fire Emblem 9 (2005) 11
Fig11. Fire Emblem 10 (2007) 11
Fig12. Fire Emblem 11 (2008) 12
Fig13. Fire Emblem 12 (2010) 13
Fig14. Fire Emblem 13 (2013) 14
Fig15. Fire Emblem 14 (2015) 14
Fig16. Fire Emblem 15 (2017) 15
Fig17. Fire Emblem 16 (2019) 15
Fig18. Unity 16
Fig19. Unreal Engine 17
Fig20. CryEngine 17
Fig21. Taula de costos 19
Fig22. Model i Portrait de Niva 21
Fig23. Model i Portrait d’Aki 22
Fig24. Disseny Original, Model i Portrait de Hilda 22
Fig25. Model i Portrait d’Ann 23
Fig26. Model i Portraits de Baagul 23
Fig27. Model i Portrait d’Ardsede 24
Fig28. Model i Portrait d’Omak 24
Fig29. Model i Portrait dels soldats de Terthas 25
Fig30. Pantalla de torn del jugador 26
Fig31. Codi de Character.cs -> funció lvlUp 28

Fig32. Apartat de generació de totes les estadístiques enemigues en GameController. 30
Fig33. Generació d'estadístiques aleatories dels soldats. 31
Fig34. Calculs dels atacs crítics i normals en GameController. 33
Fig35. Historia inicial del joc. 35
Fig36. Menú d'accions i Informació de la unitat 36
Fig37. Selecció d'enemic a atacar amb informació d'ambdues unitats. 36
Fig38. Menú de pausa amb opcions. 37

1. Introducció

1.1 Context i justificació del Treball

La industria del videojoc ha avançat molt ràpidament durant els últims anys. Aquesta industria està molt lligada a la industria tecnològica i a cada avanç tecnològic que es produeix, augmenta la qualitat tècnica que es pot assolir en els videojocs.

Gracies a totes les millores assolides, les plataformes en les que han anat apareixent els videojocs han anat canviant al llarg de la historia.

Començant per el tres en ratlla OXO implementat per Alexander Douglas com a part de la seva tesis doctoral en l'EDSAC en 1952[1], la industria ha avançat en 67 anys de forma tan rapida que ja podem trobar videojocs amb imatges ultra realistes on es poden escanejar persones reals per implementar-les dins dels videojocs, com és el cas de Death Stranding, el nou videojoc que està desenvolupant Hideo Kojima per a la plataforma PS4[2].

Gracies a tots els avenços realitzats i la difusió que s'ha realitzat sobretot els últims 20 anys la industria del videojoc s'ha convertit en una de les principals industries d'oci del mon. Gracies a això les empreses del sector han anat especialitzant les temàtiques dels videojocs que desenvolupen i ara ja no trobem videojocs únicament amb la finalitat d'entretenir a la gent, sinó que també trobem videojocs educatius i fins i tot videojocs competitius, els quals ja s'estan començant a considerar com professions, com qualsevol altre jugador de futbol per exemple[3].

El que s'ha volgut obtenir en aquest projecte és un videojoc amb la finalitat d'entretenir a la gent, presentant un repte al jugador per arribar a aconseguir l'objectiu de la partida realitzant les millors estratègies que pugui per perdre el mínim d'unitats possibles per arribar a superar el repte.

1.2 Objectius del Treball

Per a realitzar aquest projecte s'han definit una sèrie d'objectius a complir per a poder realitzar el desenvolupament de la idea principal de forma estructurada.
Els objectius principals plantejats per a la realització del videojoc han sigut els següents:

- Crear un videojoc en 3D per a dispositius Windows.

- Aprendre a planificar i gestionar tots els passos de disseny i implementació del videojoc amb un equip molt reduït de treball en un espai de temps determinat.
- Aconseguir dissenyar una IA que es presenti com un repte per al jugador i que tingui en compte la situació general de la partida.

A part dels objectius principals, hi ha certs objectius secundaris respecte al desenvolupament:

- Millorar els coneixements existents sobre la programació en el motor gràfic escollit per a la realització del videojoc.
- Saber aplicar els coneixements adquirits durant la enginyeria per a desenvolupar les diferents parts del projecte de forma efectiva.

1.3 Enfocament i mètode seguit

En el desenvolupament d'un videojoc hi ha molts factors a tenir en compte. Primerament, es difícil innovar amb un nou gènere, la industria ha avançat tant i tant de presa que és difícil trobar que una idea no ha sigut implementada amb anterioritat.

També s'ha de tenir en compte que durant el desenvolupament d'un videojoc hi ha diversos perfils implicats, com a mínim podríem trobar el Game Designer, encarregat de definir la idea del projecte, programadors que realitzen el codi de les funcionalitats definides per el Game Designer, Artistes tant 2D com 3D que realitzen tot el treball visual del videojoc i un dissenyador de sons, encarregat de tot l'apartat de música i efectes de so que podem trobar dins d'un videojoc.

Un punt important del desenvolupament d'un videojoc també es la realització de probes sobre el producte amb un equip de probes que sigui capaç de provocar diferents escenaris per comprovar que el videojoc respon de forma correcta a les accions del jugador.

Tenint aquests punts en compte s'ha decidit intentar realitzar un videojoc d'un gènere ja existent, en aquest cas hem volgut realitzar un videojoc de Rol d'estratègia per torns (SRPG).

Respecte a l'equip de treball, com no tenim un equip multidisciplinari per assignar les diferents tasques del desenvolupament al membre de l'equip que correspongui, s'ha definit un pla de treball per anar treballant de forma continua en tots els apartats del desenvolupament, tot i que, per a certs aspectes visuals s'ha procedit a la adquisició de material ofert per part de professionals.

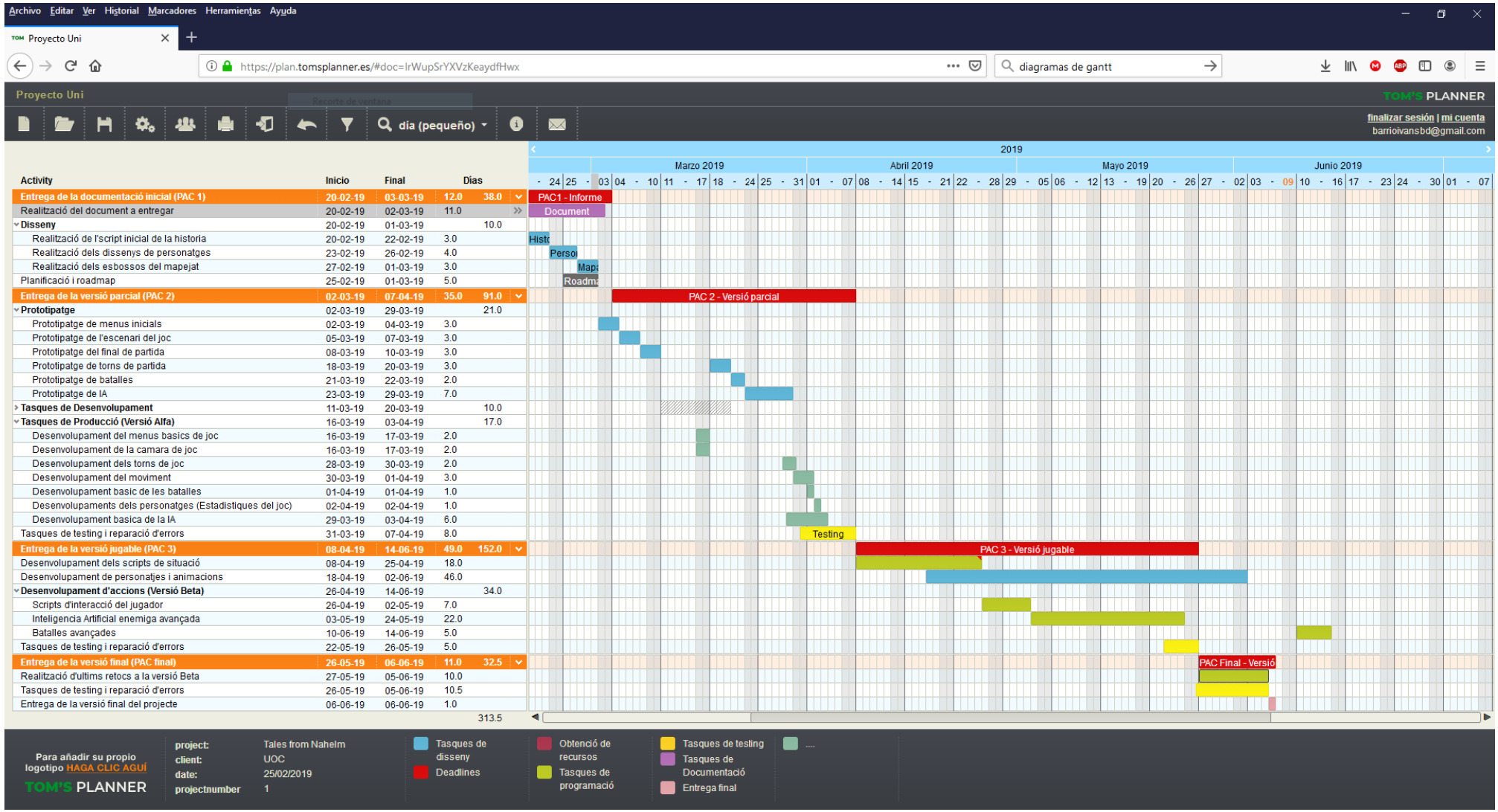
Per a les probes sobre el producte final, s'han realitzat diferents test amb persones conegudes que s'han ofert voluntàries per a provar el producte resultant.

1.4 Planificació del Treball

La planificació del projecte es va realitzar en vista a les entregues que s'havien de realitzar durant la durada proporcionada per a cada entrega.

Durant el desenvolupament del projecte es va haver de modificar la planificació inicial degut a problemes a l'hora de realitzar les implementacions i faltes de temps a l'hora de realitzar certs aspectes del videojoc, i per tant, es va modificar la planificació, quedant finalment de la següent manera:

Fig1. Diagrama de Gantt



1.5 Breu sumari de productes obtinguts

Al final del desenvolupament que s'ha realitzat durant aquests mesos s'han obtingut els següents productes:

- ★ Tales from Nahelm -> Projecte de videojoc realitzat en el motor gràfic Unity.
- ★ Memòria del projecte on es documenta tot el procés de creació del videojoc Tales from Nahelm.
- ★ Vídeos explicatiu comentat de les mecàniques del joc.

1.6 Breu descripció dels altres capítols de la memòria

En els següent capítols de la memòria trobarem un anàlisis detallat de tots els aspectes claus durant el desenvolupament del projecte.

Començarem amb el disseny inicial del joc, estudiant les possibilitats que podia oferir i totes les idees inicials pensades per a introduir en el desenvolupament.

Seguidament detallarem com s'han emprat els softwares triats per al desenvolupament del projecte, fent atenció especial, als punts claus del desenvolupament tant de les mecàniques de joc com al funcionament de la IA.

Finalment es proporcionaran les conclusions realitzades sobre el projecte, juntament amb un glossari, referencies i acreditacions als diferents autors dels materials que s'han anat emprant en el desenvolupament.

2. Idea inicial del projecte i disseny del videojoc.

2.1 Idea inicial.

2.1.1 Gènere del joc.

Tales from Nahelm és un joc de rol d'estratègia per torns, on el jugador disposa de diverses unitats que pot controlar sobre un escenari de joc per conquerir un castell fronterer, ocupat per forces enemigues d'un altre país.

Durant la partida, el jugador tindrà en tot moment una vista de l'escenari de joc amb una perspectiva isomètrica per tenir una idea general de com es troba l'escenari en tot moment, podent així controlar l'estat de les seves unitats com les de l'enemic.

En cada un dels seus torns, el jugador podrà realitzar diferents accions amb les seves unitats i un cop totes les unitats acabin les seves accions o el jugador ho decideixi, passarà ser el torn de l'enemic.

Durant el torn enemic la IA pot fer el mateix que el jugador amb les seves unitats i passar el torn al jugador un cop aquesta hagi acabat de realitzar les seves accions.

2.1.2 Referencies a videojocs existents i evolució del gènere

Hi ha diversos títols i sagues que han servit de referencia per a la creació de Tales from Nahelm, però una saga de videojocs ha sigut el punt de mira durant tot el desenvolupament.

Aquesta saga és Fire Emblem, saga de videojocs realitzada per l'empresa Intelligent Systems i publicada exclusivament en les plataformes proporcionades per Nintendo.

Aquesta saga consta amb 15 títols, pròximament 16, en els quals s'han anat modificant diferents aspectes de la jugabilitat, però sense perdre mai la essència de la saga des de el primer videojoc.

A continuació podem observar quins van ser els orígens d'aquesta saga i quines evolucions han sofert les seves mecàniques durant el pas del temps:

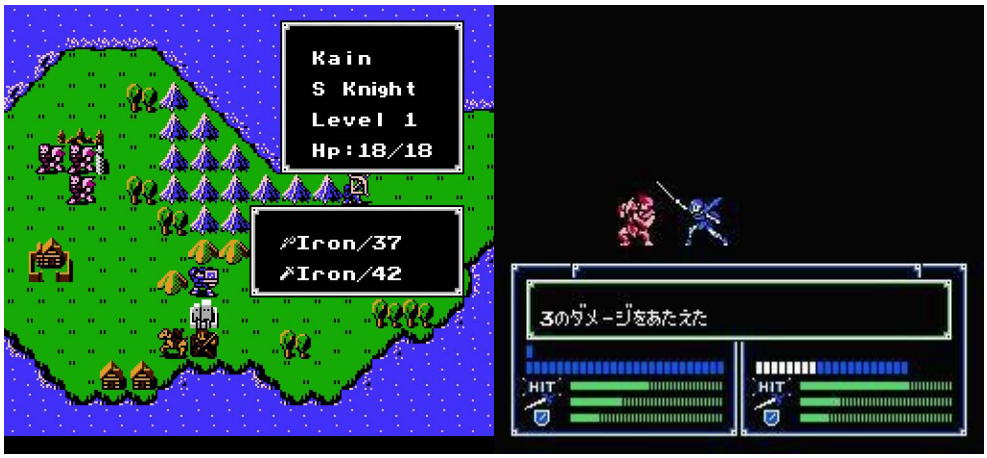


Fig2. Fire Emblem 1 (1990)

Fire Emblem: Shadow Dragon and the Blade of Light (1990)
Primera entrega, sortida per a la plataforma NES, proporcionaria les bases de tota la saga, on trobem els punts clau com el moviment per quadricules, les batalles d'unitat contra unitat, el sistema base de clases i el sistema base d'armes i objectes.

Fig3. Fire Emblem 2 (1992)

Fire Emblem Gaiden (1992)
El segon joc de la saga no continuava la historia original, sinó que era una historia paral·lela en el mateix mon. Aquesta introduïa canvis respecte a la primera entrega, donant llibertat de moviment per el mapa de selecció de capítols, podent repetir batalles, sistema de combat amb màgies diferent a l'original i dos personatges principals diferents.

Fig4. Fire Emblem 3 (1994)

Fire Emblem: Mystery of the Emblem (1994)
Amb la sortida al mercat de la SNES es va realitzar una adaptació del primer Fire Emblem, millorant l'aspecte gràfic i musical, introduint més canvis com la necessitat de desmuntar unitats muntades en espais tancats i ampliant la historia del joc original aprofitant els nous sistemes d'emmagatzemat de la SNES de 24 MB de la època.

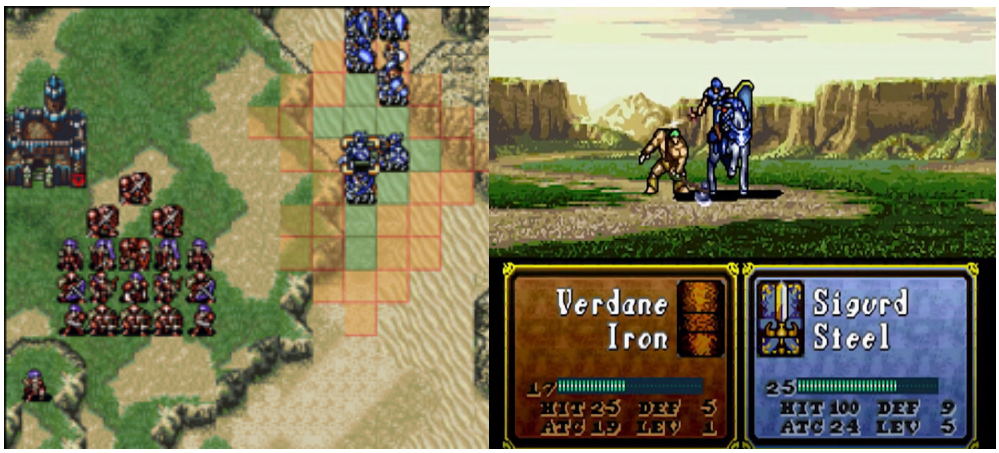


Fig5. Fire Emblem 4 (1996)

Fire Emblem: Geanology of the Holy War (1996)
Primer joc de la saga en introduir un sistema d'habilitats personals per a les unitats que aportarien més versatilitat als combats. També era el primer en incloure un sistema d'amor amb el que casar unitats i així passar habilitats i objectes als fills de les unitats de la primera part del joc, per poder-los emprar en la segona part. Gradualment s'anaven millorant tant els gràfics com la musica emprada en els jocs.

Fig6. Fire Emblem 5 (1999)

Fire Emblem Thracia 776 (1999)
Aquest joc va sortir a finals de la SNES i per tant no va esser molt conegut. No incloïa grans canvis respecte a la jugabilitat de l'anterior títol, però si que ampliava la historia del mateix i es continuava treballant per millorar l'apartat gràfic.

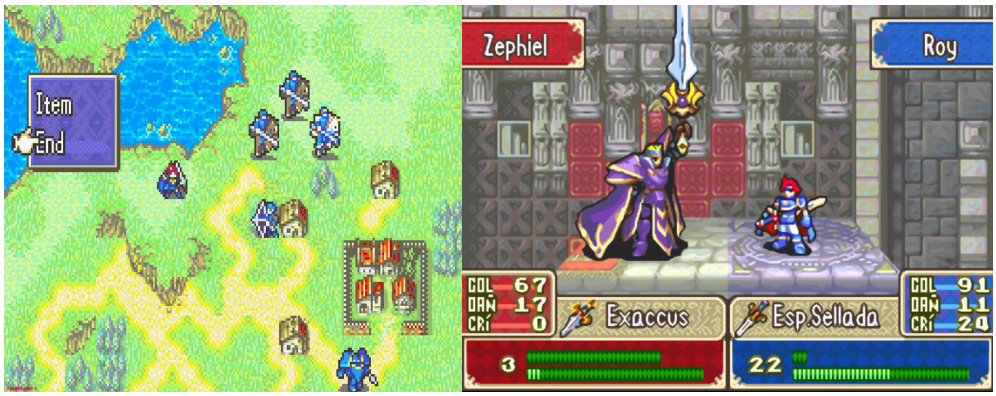


Fig7. Fire Emblem 6 (2002)

Fire Emblem Sword of Seals (2002)
Primer joc en aparèixer en la consola portàtil GBA de Nintendo. Al canvi a GBA es van perdre les habilitats personals de les unitats, però va incloure nous apartats com les conversacions de batalla, que afegien un punt d'història entre una parella de personatges concrets. També va ser el primer en implementar un apartat multijugador, on fins a 4 jugadors podien realitzar batalles amb fins a 5 unitats cada un. També va posar les bases de l'estil visual de tots els jocs que sortirien posteriorment en aquesta mateixa plataforma.

Fig8. Fire Emblem 7 (2003)

Fire Emblem Blazing Sword (2003)
Primer joc de la saga que comptava amb 3 histories diferents i amb un selector de dificultat per jugar en dificultats encara més elevades un cop es completés el joc per primer cop.

Fig9. Fire Emblem 8 (2004)

Fire Emblem The Sacred Stones (2004)
Tercer i últim joc en sortir per a la plataforma GBA, tornava a implementar el sistema de moviment per capítols de Fire Emblem Gaiden, a part de implementar això i la possibilitat de repetir capítols del joc com en Fire Emblem Gaiden no aportava cap nova funcionalitat.

Fig10. Fire Emblem 9 (2005)

Fire Emblem Path of Radiance (2005)
Primera i única entrega en aparèixer en la plataforma Nintendo Gamecube. Va esser el primer joc de la saga en emprar gràfics 3D, incorporant imatges FMV i veus per als personatges. També recuperaven el sistema d'habilitats personals per a les unitats.

Fig11. Fire Emblem 10 (2007)

Fire Emblem Radiant Dawn (2007)
Com en el cas anterior aquesta entrega va ser la única treta en la plataforma Wii de Nintendo. Continuava la historia de la entrega anterior augmentant la qualitat gràfica però sense introduir noves mecàniques de joc.

Fig12. Fire Emblem 11 (2008)

Fire Emblem Shadow Dragon (2008)

Tercera edició del primer joc de la saga, aquest cop per a la plataforma NDS. Implementava un nou sistema de canvi de classe per a les unitats i un sistema de forja per a les armes. El nou sistema multijugador permetia a dos jugadors enfrontar-se en un mapa de conquesta de castells.

Fig13. Fire Emblem 12 (2010)

Fire Emblem New Mystery (2010)
Reedició de la segona part del tercer joc de la saga, la part que ampliava la historia del joc original. Primer joc de la saga on es permet personalitzar una unitat per actuar com a avatar del jugador. Segueix amb la qualitat gràfica del títol anterior. També es el primer joc de la saga en incloure un sistema de contingut descarregable.

Fig14. Fire Emblem 13 (2013)

Fire Emblem Awakening (2013)
Primer joc de la saga en aparèixer en la plataforma 3DS de Nintendo. En aquest joc es retornava al sistema de visita de mapes de batalla anteriors. També retornava el sistema d'amor on els fills de les parelles d'unitats hereten habilitats i estadístiques dels pares. També es segueix implementant el sistema d'avatar. Es el primer joc de la saga en implementar un sistema de batalla per parelles on dues unitats aparellades poden lluitar contra una altre. S'utilitza una combinació de gràfics en 2D i gràfics en 3D que es manté per a les pròximes entregues de la plataforma, alternant entre ambdues de forma dinàmica.

Fig15. Fire Emblem 14 (2015)

Fire Emblem Fates (2015)
Segon joc aparegut en 3DS, primer joc en ser dividit en tres versions diferents, cada una d'elles amb missions i personatges diferents. Millora el sistema de combat per parelles del joc anterior i és el primer joc de la saga on les armes no es trenquen. També torna el component multijugador competitiu i s'afegeix un multijugador cooperatiu on es poden visitar les bases d'altres jugadors per obtenir recursos per millorar les armes.

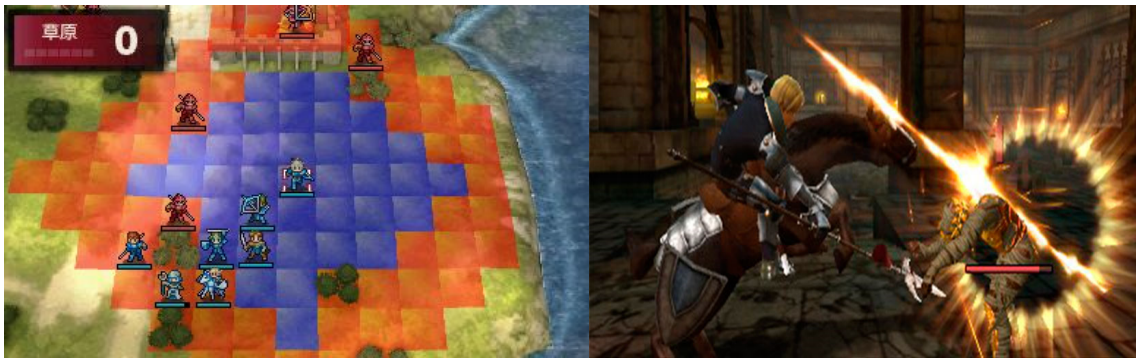


Fig16. Fire Emblem 15 (2017)

Fire Emblem Echoes: Shadows of Valentia (2017)
Reedició del segon joc de la saga per a la plataforma 3DS. Conté les mateixes característiques que la versió original amb nova historia i personatges, afegint un toc d'exploració en vista en tercera persona un cop entrem en les masmorres.

Fig17. Fire Emblem 16 (2019)

Fire Emblem Three Houses (2019)
Últim joc de la saga que sortirà per a la plataforma Nintendo Switch en Juliol. Dins de les novetats mostrades al públic s'inclou la possibilitat d'explorar la base lliurement de la mateixa forma que s'exploraven les masmorres en l'entrega anterior. També es canvia el sistema de canvi de classes i millora de les unitats de forma que sembli que es troben en una acadèmia. Visualment tot l'apartat visual s'ha modificat i està totalment en 3D. Com a detall, ara les unitats representen molts mes soldats en combat que els que apareixen al posicionar, donant la sensació de ser dos exercits de molts soldats combatent.

Repassada tota la saga ens donem compte que alguns dels canvis realitzats en diferents títols apareixen i desapareixen, però la base de la jugabilitat segueix sent la mateixa en tots i cada un dels jocs de la saga, movent les unitats en una graella atacant amb una unitat a una altra de l'enemic.

2.2 Valoració de plataformes per al desenvolupament.

A l'inici del desenvolupament del projecte es va realitzar un petit estudi de les diferents plataformes disponibles per a la realització del projecte, analitzant els avantatges i desavantatges sobre el projecte en cada un dels casos. Les plataformes que es varen considerar son les següents:

2.2.1 Unity 5

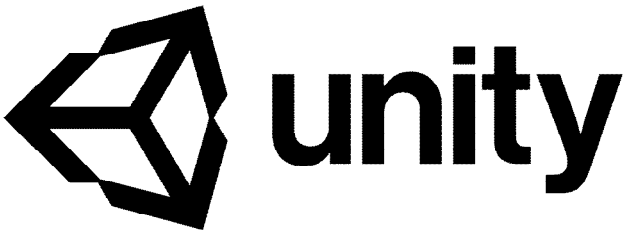


Fig18. Unity

Unity 5 és un motor gratuït per al desenvolupament de jocs tant en 2D com en 3D per a tot tipus de plataformes. A part de la llicencia gratuïta s'ofereix també una llicencia Pro mensual molt assequible, que afegeix unes poques funcionalitats extra per al desenvolupament. Pot emprar-se amb C# o Javascript i es necessari un estudi al detall de les funcionalitats que incorpora donada l'elevada corba d'aprenentatge que té. Tot i així és un dels motors més recomanables per començar a programar donada la seva versatilitat i l'assistència que es proporciona sobre el mateix, ja sigui oficial o per els fòrums de l'empresa.

2.2.2 Unreal Engine 4

Fig19. Unreal Engine

Motor desenvolupat per l'empresa Epic Games totalment gratuït per al desenvolupament de videojocs per a ordinador o consoles. Empra el llenguatge C++ per a la programació dels videojocs i es poden obtenir resultats molt professionals gracies a la potencia que proporciona. Molts jocs actuals empren aquest motor gràfic. Té una corba de dificultat en l'aprenentatge una mica elevada i no és fàcil trobar documentació sobre el mateix, però ofereix actualment uns acabats als productes finals que atrauen a moltes empreses de desenvolupament de videojocs.

2.2.3 CryEngine V

Fig20. CryEngine

Motor de l'empresa Crytek molt potent que empra els llenguatges C# i C++ per a la programació dels videojocs. Es un altre dels motors més potents del mercat, centrat en el desenvolupament de videojocs en 3D amb acabats fotorealistes. Te una corba de dificultat molt elevada però ofereix uns dels millors acabats al producte final.

2.3 Eines emprades i Valoració econòmica del projecte

Els costos del desenvolupament es divideixen en 4 blocs diferenciats, depenent de a que estan associats:

2.3.1 Software

S'ha decidit emprar una llicencia de Unity 5 estàndard i per tant no comporta un cost addicional al projecte.

Per a la edició d'objectes 3D s'ha emprat Blender, un programa d'edició 3D lliure totalment gratuït i tampoc comporta un cost addicional.

Per a la edició de vídeo per a la presentació del projecte i els vídeos explicatius s'ha adquirit una llicencia de Vegas Pro 15.0 valorada en 400€ en un grup de contingut de Software ofert per l'empresa Humble Bundle per un preu de 26€.

Per a l'edició d'imatges s'ha emprat GIMP, un programa d'edició d'imatges lliure i gratuït, que tampoc comporta un cost addicional en Software.

Total costos en Software: 26€

2.3.2 Hardware

S'han emprat dos equips per al desenvolupament del projecte, un d'ells passa les característiques recomanades i l'altre les compleix sense superar-les. El cost mitjà dels equips seria d'uns 800€ aproximadament.

També s'ha utilitzat una tauleta gràfica per a la edició d'imatges i objectes 3D valorada en 200€.

Total costos en Hardware: 1.800€

2.3.3 Recursos

Per estalviar temps en el procés de disseny artístic s'ha decidit adquirir els modelats 3D dels personatges en la pàgina HeroForge. Aquesta pàgina proporciona un editor per crear personatges per imprimir en una impressora 3D. Cada un dels modelats té un cost de 8,87€ havent triat el mètode de descarrega digital per obtenir els arxius dels models. S'han adquirit 7 models diferents per al projecte.

S'han encarregat el disseny dels logotips, fons de pantalla i imatges de conversa dels personatges a un conegut que és dissenyador que no ha volgut cobrar el treball realitzat.

La resta d'objectes, imatges i assets que s'han emprat són gratuïts i es dona crèdit al treball dels autors originals en l'Annex d'aquest mateix document.

Total costos en Recursos: 62,09€

2.3.4 Recursos Humans

Com s'ha comentat a l'inici del document aquests projectes es soler realitzar amb un equip multidisciplinari, tot i que no ha sigut el cas en

aquest projecte. En el cas d'haver-se emprat un equip multidisciplinari per a la realització del projecte, cada un dels integrants del grup tindria uns honoraris diferents. En aquest cas només mirarem els honoraris que s'aplicarien en el treball que s'ha realitzat en aquest projecte.

Les hores invertides en el projecte han sigut: 642h (107 dies treballant 6 hores diàries, caps de setmana inclosos).
Preu per hora de treball: 35€
Total: 22.470€

2.3.5 Cost total del projecte

Amb les dades proporcionades anteriorment podem calcular un cost aproximat del projecte:

Cost del Software	26€
Cost del Hardware	1.800€
Cost dels Recursos	62,09€
Cost en Recursos Humans	22.470€
COST TOTAL DEL PROJECTE	24.358,09€

Fig21. Taula de costos

2.4 Plataforma de destí.

Finalment, un cop decidit el programari a emprar i quin tipus de joc es volia realitzar es va decidir que la plataforma de destí per al videojoc seria l'ordinador. Un joc d'aquestes característiques podria sortir sense problemes en plataformes mòbils ja que un joc amb sistema de moviment per torns permet jugar de forma més despreocupada en un dispositiu mòbil ja que pots començar la partida en un punt i quan no és el teu torn, com no pots actuar de ninguna manera pots fer altres coses. Tot i així no es va decidir emprar-la com plataforma de destí degut a que un joc en 3D carrega molt la memòria del dispositiu, per tant, per jugar d'una forma similar a la d'un dispositiu mòbil, es va decidir emprar l'ordinador com a plataforma de destí ja que amb el ratolí es podria jugar de forma mes fluida deixant els pocs botons del teclat únicament per als moviments de la càmera.

3.Definició del Videojoc

3.1 Historia i ambientació

La historia del jocs ens presenta en un mon de fantasia on els països d'un continent anomenat Nahelm es troben en continues disputes. Dos d'aquests països són Ardgar i Terthas, on les fronteres entre els dos països sempre son punts calents en constant batalla. Una nit tranquil·la, un dels grans ferrers d'Ardgar, Baagul, és cridat a un lloc fronterer per a realitzar el manteniment de les armes del soldats que estaven allà destinats.

Aquella mateixa nit, un general de Terthas assalta el lloc fronterer, conquistant-lo per a l'exercit Tarthés i capturant a tots els soldats supervivents i civils del lloc com a ostatges.

Just abans de ser capturat, Baagul aconsegueix enviar un missatge a la seva filla, Ann, demanant-li que busqués a un grup de mercenaris que coneixia bé per ajudar-la a rescatar-los.

Ann parteix en busca dels Dracs, mercenaris d'Ardgar que treballen per al benestar del país. Un cop els troba, marxen cap el lloc fronterer, amb l'esperança de no arribar massa tard.

3.2 Objectius del nivell

Un cop acabada la introducció del joc i iniciat el primer torn del jugador es donarà la opció de veure un petit tutorial de com jugar, el jugador pot optar per veure l'explicació o saltar-la. En tots dos casos al final s'informarà que l'objectiu es anar a per el cap de l'exercit enemic, ja que un cop l'eliminin tindran via lliure per conquistar el castell un altre cop.

3.3 Personatges

Trobem diversos personatges lligats a la partida en curs, els quals poden aparèixer en el camp de batalla o únicament en trossos de la historia. El llistat de personatges que apareixen és el següent:

Niva: Líder dels Dracs d'Ardgar, és un dels 4 personatges que pot controlar el jugador durant la partida. És capaç d'emprar destrals per combatre i sempre està disposat a ajudar a qui ho necessiti.
Aki: Segona al capdavant dels Dracs d'Ardgar, és un altre dels 4 personatges que pot controlar el jugador. És una mica esquerpa però molt bona guerrera, sobretot si el client paga bé.
Hilda: Nova guerrera dels Dracs d'Ardgar, la tercera del grup que pot controlar el jugador. És una dona alta i forta capaç de brandar espases i llances. No recorda com ha aparegut en Ardgar, però mentre busca una forma de tornar a la seva terra ajudarà tot el que pugui.

Ann: Filla de Baagul, i ultima integrant del grup de personatges que pot utilitzar el jugador. És capaç d'utilitzar espases en combat, tot i que el seu punt fort és realitzar atacs i curacions màgiques. Vol salvar el seu pare captiu costi el que costi.
Baagul: Pare de Ann. Gran ferrer del país d'Ardgar que es troba captiu en un lloc fronterer a mans d'Omak i Ardsede.
Ardsede: Segona al capdavant del destacament d'Omak en l'exercit de Terthas. Es una fetillera capaç de realitzar devastadors conjurs.
Omak: General de Terthas. Ha conquerit el lloc fronterer per seguir avançant en la guerra contra Ardgar. És devastador en l'ús de llances.
Soldats de Terthas: Soldats rasos de Terthas capaços de brandar qualsevol arma física.

3.4 Disseny de personatges

Com s'ha emprat l'eina de HeroForge per a realitzar els personatges primerament s'han generat els personatges i seguidament s'ha procedit a realitzar l'art dels mateixos amb l'excepció de Hilda, la qual és directament la representació directa d'un personatge d'una partida de rol que es retransmet per internet. L'autora del personatge ha donat el seu consentiment per a que aparegui en aquest projecte.

3.4.1 Niva

Fig22. Model i Portrait de Niva

3.4.2 Aki

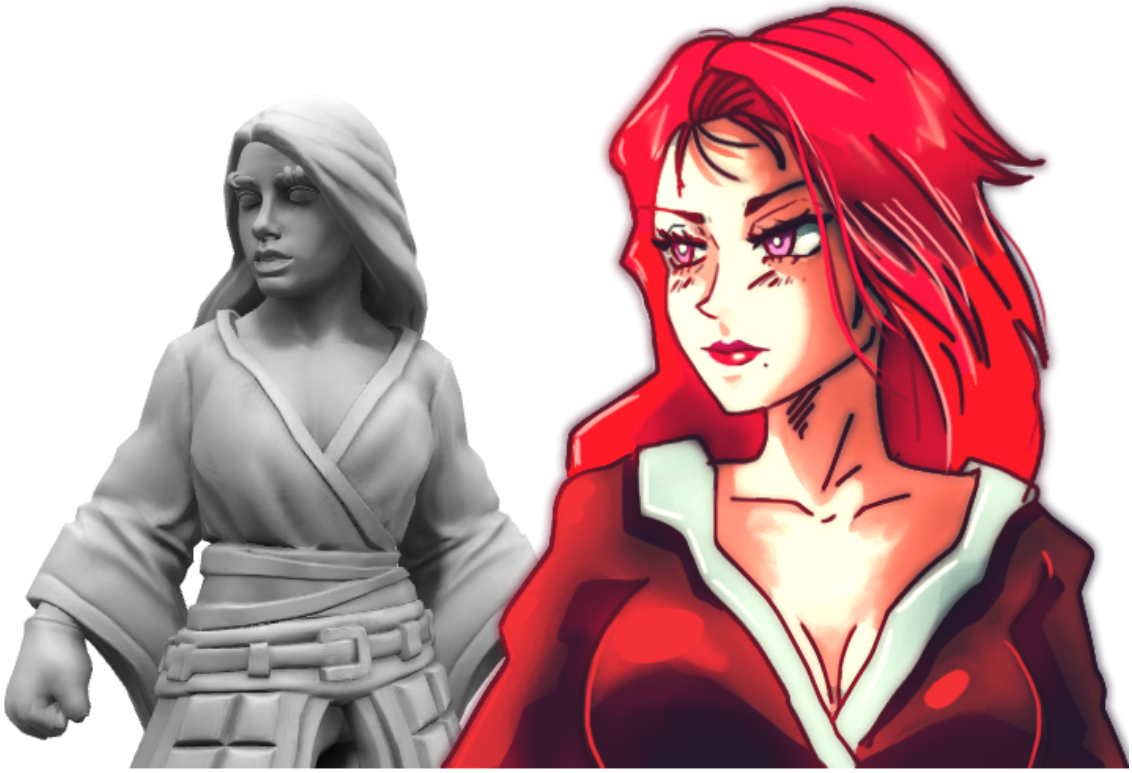


Fig23. Model i Portrait d'Aki

3.4.3 Hilda

Fig24. Disseny Original, Model i Portrait de Hilda

3.4.4 Ann

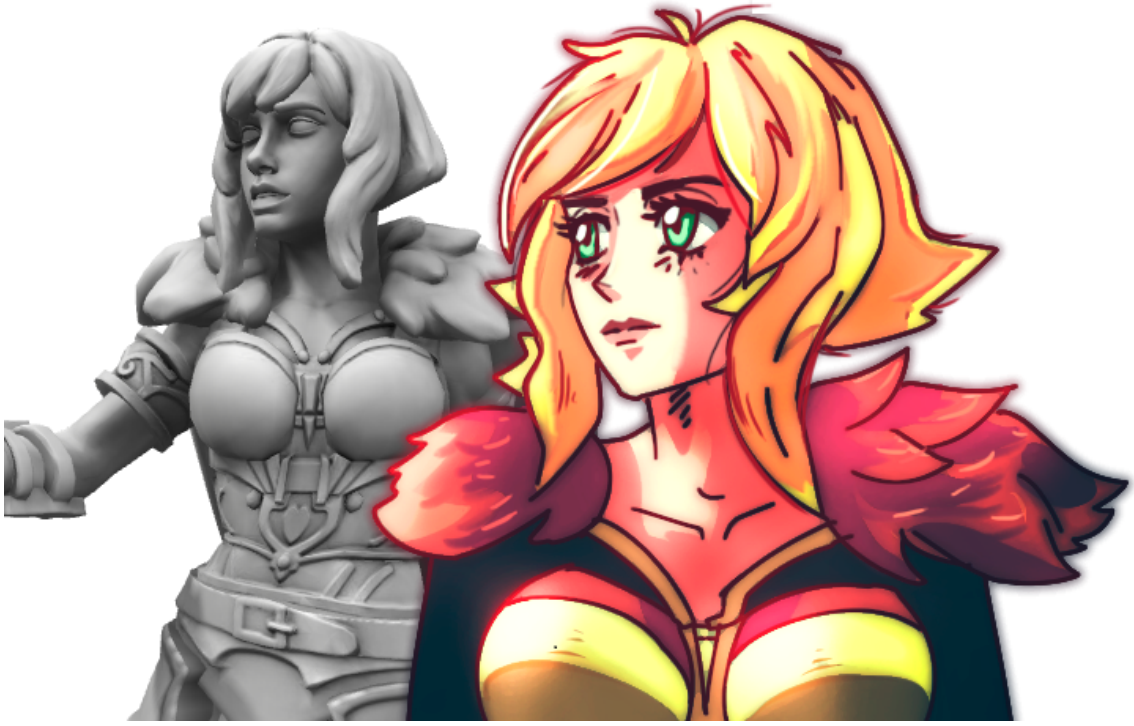


Fig25. Model i Portrait d'Ann

3.4.5 Baagul

Fig26. Model i Portraits de Baagul

3.4.6 Ardsede

Fig27. Model i Portrait d'Ardsede

3.4.7 Omak

Fig28. Model i Portrait d'Omak

3.4.8 Soldats

Fig29. Model i Portrait dels soldats de Terthas

4.Disseny del Videojoc

4.1 Sistema de torns

Fig30. Pantalla de torn del jugador

Per a realitzar el sistema de torns s'han definit dues variables que controlen dos aspectes clau del joc, la primera anomenada "actualTurn" informa de qui es el torn actual si del jugador o de la IA.

La segona s'ha anomenat "turnState" i té el control directament sobre les accions que poden realitzar ambdós jugadors.

Per al torn del jugador, "turnState" comprova si hi ha algun personatge actiu seleccionat, si es troba en els diferents menús d'acció, si el personatge s'està movent, si esta triant algun objecte a utilitzar o escollint una arma per atacar, si està seleccionant a un enemic a atacar o si es troba en combat.

Per a la IA va seleccionant les unitats més amenaçades i les va posicionant sempre intentant obtenir el màxim benefici tant de moviment com d'atacs. Tot això s'explicarà més en detall en l'apartat 4.3.

Per a realitzar totes aquestes funcions de forma fluida es va decidir implementar-les en la funció Update del controlador del joc, el qual hem anomenat "GameController".

4.2 Control de Personatges

Com s'ha explicat amb anterioritat en el sistema de torns, els personatges poden realitzar diverses accions, i totes estan controlades per "GameController".

Tot i que el “GameController” realitza totes les comprovacions pertinents i executa les accions a realitzar, cada unitat conté un script “Character”, el qual conté tota la informació respectiva a la unitat i per tant, totes les accions que depenguin de comprovar o realitzar càlculs amb dades de les estadístiques d’una unitat, aquestes es realitzaran en l’Script “Character”.

Un exemple d’aquest tipus d’accions és el guany d’experiència i la pujada de nivell. “GameController” calcula quina quantitat d’experiència guanya una unitat depenent del resultat del combat i contra quin oponent estava combatent (Si una unitat lluita amb un soldat i no el mata guanyarà menys experiència que si combatis amb el mateix soldat i el matés en combat o que si lluités contra el cap de l’exercit sense matar-lo).

Un cop “GameController” sap l’experiència que es reparteix, se la informa a l’Script “Character” per a que augmenti l’experiència en el nombre informat. Si durant la pujada d’experiència sobrepassem el valor de 100, la unitat puja de nivell. Aquesta pujada de nivell comporta una pujada en diverses característiques que venen determinades per els valors de creixement informats al generar les dades dels personatges i és totalment aleatòria, calculant per a cada estadística un valor aleatori entre 1 i 100 i comparant-lo amb el valor del creixement de la estadística concreta. Si el valor aleatori és inferior o igual al valor informat aquesta estadística pujarà en un punt el seu valor.

```
public void lvlUp(int exp)
{
    //Si no ens trobem a maxim nivell
    if (lvl < 20)
    {
        this.exp += exp;
        if (this.exp >= 100)
        {
            this.exp -= 100;
            //pujar el nivell ja que s'han superat els 100 punts d'experiencia
            int[] rands = new int[8];

            lvl++;

            //Generar 8 aleatoris entre 1 i 100
            for (int i = 0; i < 8; i++)
            {
                rands[i] = Random.Range(1, 100);
            }

            //Comprovem si el creixement ens permet pujar la estadistica per a cada valor generat si no es troba ja en el seu maxim valor
            if (rands[0] <= pvGrowth && pv < maxPv)
            {
                pv += 1;
            }
            if (rands[1] <= strGrowth && str < maxStr)
            {
                str += 1;
            }
            if (rands[2] <= magGrowth && mag < maxMag)
            {
                mag += 1;
            }
            if (rands[3] <= sklGrowth && skl < maxSkl)
            {
                skl += 1;
            }
            if (rands[4] <= spdGrowth && spd < maxSpd)
            {
                spd += 1;
            }
            if (rands[5] <= lckGrowth && lck < maxLck)
            {
                lck += 1;
            }
            if (rands[6] <= defGrowth && def < maxDef)
            {
                def += 1;
            }
            if (rands[7] <= resGrowth && res < maxRes)
            {
                res += 1;
            }
        }
    }
    Debug.Log(charName + " LVL: " + lvl + " Exp: " + this.exp);
```

Fig31. Codi de Character.cs -> funció lvlUp

Totes les funcions que requereixin realitzar calculs amb les estadístiques tant de les propies unitats com dels objectes que porten amb elles passa per l'Script de "Character".

4.3 IA

Com ja s'ha comentat amb anterioritat la IA està implementada dins del "GameController". S'ha pres aquesta decisió ja que en les probes realitzades de la primera idea de la IA que estava totalment separada, aquesta no respectava els torns ni de moviment ni de combat. Introduint-la en el "GameController" hem aconseguit que respecti el torn del jugador i que mogui les unitats d'una en una.

Aquesta IA té diferents punts a l'hora de realitzar les accions.

Primerament realitza una cerca de les unitats que té en el camp de batalla. D'aquestes unitats mira les que té actives (les que encara no han

finalitzat les accions disponibles). I d'aquest grup d'unitats actives mira quines estan a rang de l'enemic, que es denota per l'àrea de moviment de la que disposen sumada a l'àrea d'atac que tenen.

Un cop sap quines unitats estan a rang, per a cada una d'elles comprova quina és l'amenaça mes propera i un cop trobada comença a apropar-se a ella. Aquesta amenaça ve proporcionada per la diferencia d'armes entre les dues unitats.

Si la unitat no queda a rang de la unitat a la que va buscar, i troba per el camí una altra unitat que si queda a rang, atacarà a aquesta segona per estar a rang de combat, sinó, finalitzaria el seu moviment el mes a prop que pugui de la seva amenaça mes propera.

Un cop la IA mou totes les unitats a rang finalitza el seu torn per a que torni a moure el jugador.

4.4 Generació d'enemics bàsics

En general, i com veurem en el següent capítol, les estadístiques de les unitats estan generades de forma manual, i per tant totes les unitats importants (les 4 unitats controlables per el jugador i els 2 enemics diferents de la IA) tenen les seves característiques fixes i sempre començaran sent les mateixes totes les partides (tot i que amb les pujades de nivell de les unitats del jugador poden acabar sent diferents, tot i que molt semblants gracies al sistema de creixements).

Els únics que no tenen les estadístiques ni les armes fixes, son els soldats. Aquestes unitats inicien amb les mateixes estadístiques base, però el sistema les fa pujar de nivell de forma aleatòria per donar diversitat a les estadístiques enemigues.

El mateix passa amb les armes amb les que van equipades, el sistema genera un aleatori que determina quina arma es genera per a la unitat un cop s'inicia la partida i sempre trobarem opcions diferents cada cop que iniciem un nou joc.

```
foreach (GameObject unit in enemies)
{
    switch (unit.name)
    {
        case "Omak":
            unit.GetComponent<Character>().createCharacter("Enemy", "Omak", "Warrior", 41, 18, 0, 13, 12, 11, 13, 1, 20, 1);
            unit.GetComponent<Weapon>().setWeapon("Silver Lance", "Lance", "B", 13, 1, 75, 0);
            unit.GetComponent<Character>().setWeapon(unit.GetComponent<Weapon>());
            break;
        case "Ardsede":
            unit.GetComponent<Character>().createCharacter("Enemy", "Ardsede", "Sorcerer", 24, 4, 8, 4, 6, 3, 8, 7, 20, 6);
            unit.GetComponent<Weapon>().setWeapon("Shadows", "Magic", "C", 4, 2, 60, 50);
            unit.GetComponent<Character>().setWeapon(unit.GetComponent<Weapon>());
            go = new GameObject("ArdsedeStaff");
            go.AddComponent<Item>().setName("Staff");
            go.transform.SetParent(unit.transform);
            unit.GetComponent<Character>().obtainObject(go.GetComponent<Item>());
            break;
        default:
            //Hacer un generador aleatorio de diferentes tipos de unidades para los soldados / Que lo mire dependiendo de las armas que lleve
            randomStats = getRandomSoldierStats();
            unit.GetComponent<Character>().createCharacter("Enemy", "Terthas Soldier", "Soldier", randomStats[0],
                randomStats[1], randomStats[2], randomStats[3], randomStats[4], randomStats[5], randomStats[6], randomStats[7], randomStats[8], randomStats[9]);
            randomWeapon = getRandomWeapon();
            unit.GetComponent<Weapon>().setWeapon(randomWeapon[0], randomWeapon[1], randomWeapon[2], int.Parse(randomWeapon[3]),
                int.Parse(randomWeapon[4]), int.Parse(randomWeapon[5]), int.Parse(randomWeapon[6]));
            unit.GetComponent<Character>().setWeapon(unit.GetComponent<Weapon>());
            break;
    }
}
```

Fig32. Apartat de generació de totes les estadístiques enemigues en GameController.

```
1 referencia
public int[] getRandomSoldierStats()
{
    int[] stats = new int[10];
    //Generem el nivell dels soldats (entre 1 i 5)
    int lvl = Random.Range(1, 5);
    int growth;
    //inicialitzem els valors base de la classe, els quals modificarem depenent del nivell generat automaticament.
    int pv = 16;
    int str = 3;
    int mag = 0;
    int skl = 4;
    int spd = 6;
    int lck = 0;
    int def = 6;
    int res = 6;
    int mov = 20;

    if (lvl != 1)
    {
        for (int i = 1; i < lvl; i++)
        {
            growth = Random.Range(1, 100);
            if (growth <= 40) pv++;

            growth = Random.Range(1, 100);
            if (growth <= 10) str++;

            growth = Random.Range(1, 100);
            if (growth <= 0) mag++;

            growth = Random.Range(1, 100);
            if (growth <= 10) skl++;

            growth = Random.Range(1, 100);
            if (growth <= 10) spd++;

            growth = Random.Range(1, 100);
            if (growth <= 0) lck++;

            growth = Random.Range(1, 100);
            if (growth <= 5) def++;

            growth = Random.Range(1, 100);
            if (growth <= 5) res++;
        }
    }

    stats[0] = pv;
    stats[1] = str;
    stats[2] = mag;
    stats[3] = skl;
    stats[4] = spd;
    stats[5] = lck;
    stats[6] = def;
    stats[7] = res;
    stats[8] = mov;
    stats[9] = lvl;

    return stats;
}
```

Fig33. Generació d'estadístiques aleatories dels soldats.

4.5 Batalles

Totes les batalles de Tales from Nahelm son series de càlculs que determinen des de si alguna de les unitats ataca dos cops, fins a si es realitza un atac crític i quin valor de dany tindrà.

Com ja s'ha comentat, primerament es calcula si alguna de les unitats té la possibilitat de realitzar un atac doble. El "GameController" obté les dades de la característica de Velocitat (Spe) de cada una de les unitats involucrades en la batalla. Si una de les dos unitats té una Velocitat superior a la de l'enemic més 5 aquesta primera unitat realitzarà un atac doble, si per el contrari és la unitat enemiga la que supera la velocitat d'aquesta unitat més 5, serà la unitat enemiga la que realitzi l'atac doble.

Els torns del combat sempre es desenvolupen de la següent forma:

- Ataca la unitat atacant.
- Ataca la unitat atacada.
- Es realitza un atac doble si correspon (en qualsevol dels dos casos plantejats en el paràgraf anterior)

Un cop decidits els torns del combat es comencen a realitzar càlculs per a cada un dels torns. Primerament es comproven les estadístiques de les unitats involucrades i de les armes que porten equipades i es calcula:

- La probabilitat de realitzar un impacte crític sobre l'enemic.
- La probabilitat d'encertar un atac normal sobre l'enemic.
- El valor de dany que es realitzaria a l'enemic en cas d'impactar l'atac.

Amb aquestes dades generades, el "GameController" ja pot començar a triar nombres aleatoris per veure si es realitzen diferents accions en el combat.

Primerament comprova si es realitza un impacte crític, si és així multiplica el valor del dany realitzat per 3 i procedeix a atacar, ja que els impactes crítics no fallen mai.

En cas de no encertar un impacte crític es passa a comprovar si s'encerta l'atac sobre l'enemic, i si s'encerta, es realitza el càlcul de dany.

```
randV = Random.Range(1, 100);
if (randV <= atckStats[2] && randV != 0)
{
    critical = 3;
    isAHit = true;  //Un atac critic sempre encerta
}

if (!isAHit)
{
    //Es realitza l'atac normal?

    randV = Random.Range(1, 100);
    if (randV <= atckStats[0] && randV != 0)
    {
        isAHit = true;
    }
}

if (isAHit)
{
    damageToEnemy = atckStats[1] * critical;
}
if (!isAHit && whoIsAttacking == "Ally" && playerUnitMissed)
{
    playerUnitMissed = true;
}
else
{
    playerUnitMissed = false;
}
```

Fig34. Calculs dels atacs crítics i normals en GameController.

Un cop realitzats els càlculs corresponents es comprova si l'enemic ha mort i es calcula l'experiència guanyada per el combat realitzat (sempre que siguin unitats del jugador, les unitats de la IA no pugen de nivell. Aquesta decisió està explicada en el següent apartat).

4.6 Configuració de la dificultat

La configuració del nivell de dificultat de la partida ha sigut una de les parts que més ha costat realitzar, ja que, l'objectiu de la partida realitzada era presentar un repte al jugador, però que no sigues difícil trobar una estratègia capaç de vèncer i passar-se el nivell.

Per a la configuració de la dificultat, que en gran mesura es concentra en les estadístiques de totes les unitats en joc, s'han fet molts jocs de proves per comprovar que deixant els enemics més forts a estadístiques altes, amb les característiques base dels personatges sigui viable enfrontar-se a ells.

Per a anivellar la balança, es va decidir posar els soldats amb estadístiques aleatòries amb un nivell baix. Aquest baix nivell és el mateix que el de les unitats del jugador, però al tractar-se d'unitats genèriques, tenen estadístiques mes baixes del normal. Per tant, representen una oportunitat per a que el jugador pugui avançar de forma calmada cap a l'exercit enemic i que pugui pujar, com a mínim un parell de nivells per a un parell d'unitats per a poder enfrontar-se a l'enemic més fort amb una mica més de potencia.

Tot i així, quedava penjant un factor que podia fer que el joc fos impossible de guanyar. Si es deixava el sistema de pujada de nivell per als enemics, Omak podia matar a la meitat de les unitats del jugador i pujar de nivell, aportant un valor encara més elevat de les estadístiques altes que ja posseïa. Per tant, per a fer que es pogués guanyar a aquest enemic, es va des habilitar la funció de guany d'experiència per als enemics.

4.7 Game Over

Si totes les unitats del jugador son eliminades el joc passa a estar en mode de "Game Over", en aquest mode, qualsevol acció sobre el teclat o el ratolí ens enviarà a la pantalla d'inici i el jugador haurà de tornar a començar una nova partida per poder tornar a jugar.

4.8 Diàlegs, Tutorial i Victoria

A l'inici de la partida i al final de la mateixa si hem aconseguit vèncer a Omak i prendre el castell, trobarem seqüencies de diàleg que ens comentaran la historia del joc.

Fig35. Historia inicial del joc.

El diàleg inicial, tot i explicar les bases del món del joc, és possible saltar-lo per començar la partida directament.

Un cop ha començat la partida sortirà un diàleg tutorial que, si el jugador vol, pot llegir per saber com funcionen els controls del joc. Abans d'acabar aquest diàleg s'explica al jugador quin és l'objectiu de la partida, que en aquest cas es prendre el castell un cop s'elimina a Omak.

Un cop es compleix l'objectiu de victòria es proporciona a les unitats del jugador l'acció de prendre el castell. Un cop se selecciona aquesta opció, el joc mostra la seqüencia de diàlegs del final d'aquesta historia. Aquesta seqüencia s'ha decidit que no es pot saltar, ja que representa la recompensa a aquells que acaben el nivell. Un cop acabada la seqüencia de diàleg el joc envia al jugador directament a la pantalla del menú inicial del joc.

4.9 GUI i menú d'opcions.

La GUI d'aquest joc s'ha simplificat al màxim per a no carregar la visió del jugador tot i tenir un alt nombre de dades a tenir en compte.

Per una banda tenim els menús d'accions. Aquests menús ens donen les diferents accions que poden realitzar les unitats del jugador. Únicament són visibles quan el jugador selecciona una unitat i aquesta no s'està movent o combatent. Depenent de les accions triades, aquests menús retornaran mes o menys informació ja que per a la selecció d'objectes mostra un segon menú on podem veure tots els objectes que pot tenir una unitat.

Fig36. Menú d'accions i Informació de la unitat

Quan seleccionem una unitat per realitzar accions amb ella trobem en la part superior dreta de la pantalla un petit menú d'estat que ens indica quina unitat hem seleccionat mostrant-nos el "Portrait" corresponent, la vida que té actualment en forma de barra i el tipus d'arma que utilitza amb una icona al costat de la unitat. En cas de seleccionar un enemic també se'ns informarà de quin és l'estat de l'enemic seleccionat.

En batalla, les mateixes dades de la unitat enemiga, apareixen en la part superior esquerre.

Fig37. Selecció d'enemic a atacar amb informació d'ambdues unitats.

Finalment trobem el menú de pausa amb opcions. S'accedeix a aquest menú polsant el botó ESC sense cap unitat seleccionada. Aquest menú

ens deixa activar o desactivar tant la música del joc com els efectes de so del mateix. També trobem un parell d'opcions relacionades amb el jugador.

La primera d'elles és "EXIT GAME", que com bé informa la pròpia opció ens permet sortir de la partida cap al menú principal. Sempre que sortim de la partida en curs perdem els avenços realitzats i per tant no es pot continuar des d'on es va deixar per últim cop i s'ha de començar de nou.

L'altra opció és "END TURN". Aquesta opció finalitza directament el torn del jugador i comença el de la IA. Aquesta opció es va implementar per no haver de passar per totes les unitats i finalitzar-les una per una per acabar un torn si el jugador troba que estan ben posicionades en el mapa i no fa falta moure-les per aguantar al enemics en el seu torn.

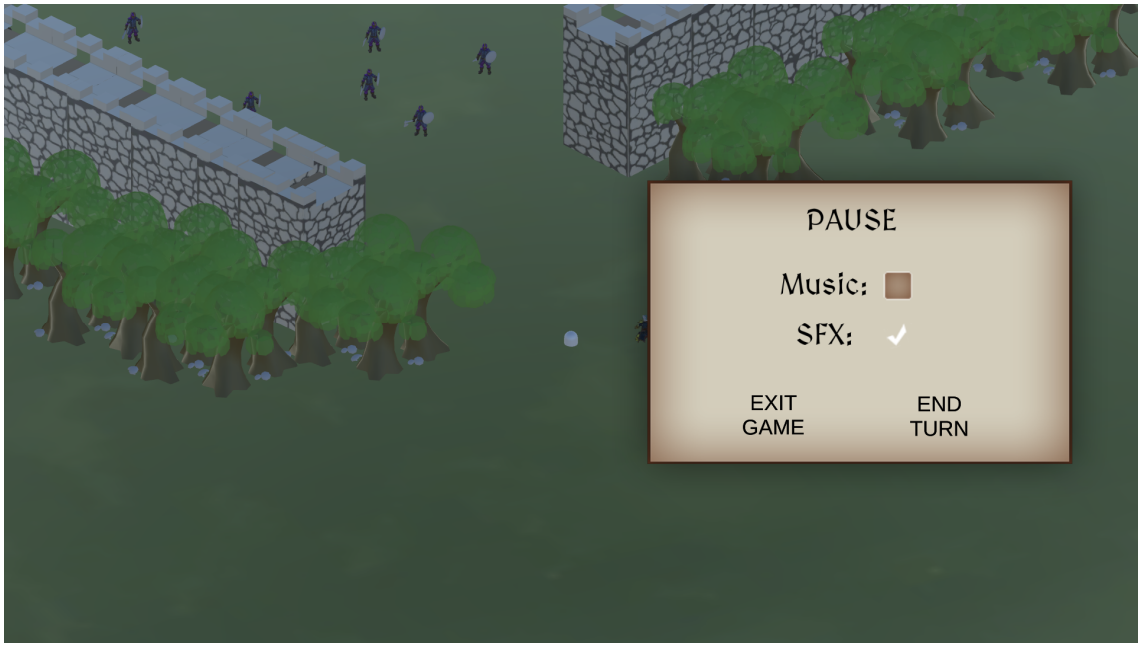


Fig38. Menú de pausa amb opcions.

5.Conclusions

La realització d'aquest treball m'ha permès indagar en l'aprenentatge i implementació de una IA que pugui presentar un repte al jugador sense arribar a ser abusiva. Gracies a aquest resultat s'ha aconseguit realitzar un videojoc que suposi un repte al jugador però que no sigui impossible de guanyar. I com la gran majoria d'enemics canvia entre partides és possible tornar a jugar el nivell sense haver d'aplicar la mateixa estratègia cada cop.

Tot i així, per arribar al punt d'obtenir una IA que generés estadístiques correctes sense arribar a passar-se s'ha agut de realitzar un exhaustiu procés de probes en el que s'han realitzat moltes simulacions amb estadístiques per tal de trobar unes característiques de personatges concretes per a arribar a aquesta sensació de repte. Aquesta part d'investigació personal ha estat molt productiva, ja que m'ha mostrat una forma de balancejar la dificultat d'un joc. Aquest apartat no s'havia tingut en compte i ocupa un gran punt dins del desenvolupament d'un videojoc d'aquestes característiques.

També s'ha aprés que l'apartat gràfic és important i que tot i implementant dissenys senzills es pot arribar a un grau de qualitat elevat. Tot i així, com a autor del videojoc he de reconèixer que no tinc ni experiència ni habilitats artístiques, i per tant, aquesta part del treball m'ha frenat el desenvolupament més de l'esperat. Tot i obtenir tots els models d'altres fonts (ja siguin remunerades o de repositoris de distribució d'objectes) s'ha agut de realitzar un treball artístic que no ha acabat resultant com s'ha esperat. En cas de voler millorar la qualitat del producte final, el més recomanable seria comptar amb una persona experta en aquest camp per a realitzar tot l'apartat gràfic del joc.

Com s'acaba de comentar breument en el paràgraf anterior, hi ha parts del desenvolupament que s'han allargat més de l'esperat i han frenat altres parts del desenvolupament, deixant així un parell dels objectius plantejats en un inici per al videojoc sense implementar.

Per exemple l'intercanvi d'objectes no s'ha pogut implementar ja que no s'ha implementat cap manera d'obtenir objectes nous i per tant no és necessari reorganitzar els inventaris individuals dels personatges per poder obtenir el màxim d'objectes per partida.

Tampoc s'han implementat les habilitats de combat, objectiu plantejat per a realitzar un combat avançat, però que no s'ha pogut realitzar degut als treballs artístics que s'han anat allargant durant el desenvolupament.

Tot i realitzar canvis en la planificació des de la primera entrega per centrar-nos en apartats més importants s'ha intentat seguir la planificació establerta, tot i que no s'ha pogut seguir en la seva totalitat tal i com estava planejat. Per a poder aconseguir un producte final s'ha agut d'encarregar a un conegut tot el treball artístic de la GUI (Dissenys dels

portraits dels personatges i el fons de pantalla del menú inicial) per a poder incloure dins del temps estipulat tot el treball artístic del joc.

Com s'ha comentat anteriorment, com a futures millores del videojoc, s'establiria el sistema d'intercanvi d'objectes i la interacció amb els mateixos en el mapa.

També s'inclouria el sistema d'habilitats, per poder proporcionar al combat una mica més d'estratègia tenint en compte les habilitats disponibles per a cada una de les unitats.

I finalment es milloraria tot l'apartat gràfic del joc, contractant a un Artista i a un Animador per millorar tant totes les textures aplicades als objectes presents al joc, com per millorar tot el moviment dels models 3D presents.

4. Glossari

- **EDSAC:** Acrònim de Electronic Delay Storage Automatic Calculator, un dels primers computadors creats. D'origen britànic.
- **PS4:** Acrònim de Play Station 4, videoconsola desenvolupada per la companyia Sony.
- **SRPG:** Acrònim de Strategy Rol Playing Game, gènere de videojocs dins de l'apartat dels jocs de rol amb un toc d'estratègia en la jugabilitat.
- **IA:** Acrònim d'Intel·ligència Artificial.
- **NES:** Acrònim de Nintendo Entretainment System consola de Nintendo, anomenada Famicom en Japó.
- **SNES:** Acrònim de Super Nintendo Entretainment System, successora de la NES original, anomenada Super Famicom en Japó.
- **GBA:** Acrònim de Game Boy Advance, consola portàtil de Nintendo.
- **FMV:** Acrònim de Full-Motion Video, escenes de vídeo en 3d d'alta calitat.
- **NDS:** Acrònim de Nintendo DS, plataforma portàtil de Nintendo que comptava amb dues pantalles per jugar.
- **Avatar:** Identitat virtual d'un usuari per a actuar com la seva representació.
- **3DS:** Plataforma portàtil de Nintendo, millora de la versió anterior (NDS) que aporta una pantalla que proporciona un efecte 3D en els jocs que el tinguin habilitat.
- **Nintendo Switch:** Plataforma de Nintendo que es pot connectar a una pantalla per jugar en casa i que es pot extreure de la base per jugar de forma portàtil.
- **C#:** Llenguatge de programació.
- **Portrait:** Representació de la part superior d'un personatge.
- **Script:** Fitxer de codi, en aquest cas de llenguatge C#.
- **Spe:** Acrònim del terme anglès Speed, que representa l'estadística de velocitat dels personatges.
- **Game Over:** Terme anglès de finalització de partida.

5. Bibliografia

[1]: Wikipedia, OXO (06/06/2019): https://es.wikipedia.org/wiki/OXO

[2]: Kojima Productions (06/06/2019): http://www.kojimaproductions.jp/en/#works

[3]: The Gamer eSports, Los eSports dan un paso más hacia la profesionalización con la creación de becas para formarse en esta disciplina. (06/06/2019): https://thegamersports.mundodeportivo.com/noticias-industria-esports/los-esports-dan-un-paso-mas-hacia-la-profesionalizacion/

6. Annexos

6.1 Crèdits

A continuació es passarà a donar crèdits als autors originals del contingut que s'ha emprat per a la realització del treball:

- @ChioUyuni: Autora original de Hilda, personatge que pot utilitzar el jugador durant la partida.
- Modelat del castell: Modelat del castell extret del repositori BlendSwap, Model original de l'usuari CopperStache.
- Terreny del mapa: Modelat del terreny extret del repositori BlendSwap, Model original de l'usuari MacroManJr.
- Arbres: Modelat dels arbres extret del repositori BlendSwap, Model original de l'usuari Explosive James.
- Capces del mapa: Modelat de les capces del mapa extret del repositori BlendSwap, Model original de l'usuari Daniel74.
- Llanternes: Modelat de les llanternes extret del repositori BlendSwap, Model original de l'usuari oceanarts.
- Imatge del castell: Extreta de Pexels, autor original: Dominika Roseclay.
- Imatge de l'interior del castell: Extreta de Pexels, autor original: João Gustavo Rezende.
- Fons de batalla entre dos exercits: Extreta de Monodomo, autor original: Steven Riley.
- Mapa generat de Nahelm: Generat en l'aplicació Web Azgaar's Fantasy Map Generator. Tipus de mapa High Island, llavor de generació: 934252262.
- @Chardironic: Autor dels portraits i el fons de pantalla del menú inicial del joc.
- Icona d'una espasa: Icona extreta de Flaticon, autor original: Those Icons.
- Icona d'una llança: Icona extreta de Flaticon, autor original: Freepik.
- Icona d'una destral: Icona extreta de Flaticon, autor original: Freepik.
- Icona d'una flama: Icona extreta de Flaticon, autor original: Those Icons.
- Tots els efectes de so han sigut extrets de la pàgina: https://freesound.org/, concretament son els següents:
    - Flipping Bookpages: autor InsperctorJ.
    - Attack_Sword: autor JoelAudio.
    - Magic: autor RICHERlandTV.
    - Footstep on gravel: autor Ali_6868.
- Per a la música s'ha extret tota de la pàgina YouTube, en pagines de creadors de contingut que proporcionen música sense Copyright, concretament:
    - Musica de la introducció i torn aliat: musica proporcionada per No Copyright Music.

    - Música del torn enemic: musica proporcionada per No Copyright Music.
    - Música de Victoria: musica proporcionada per Stockwaves.
    - Música de Derrota: musica proporcionada per Royalty Free Music Library.
- HeroForge: Pàgina on s'han generat i comprat els models dels personatges del joc.
- Simple Fantasy GUI: Producte ofert en la Asset Store de Unity que proporciona els arxius per a realitzar menús bàsics. Autor original: Nayrissa