

BigBug

Disseny i Desenvolupament d'un videojoc

Memòria del Projecte Final de Grau

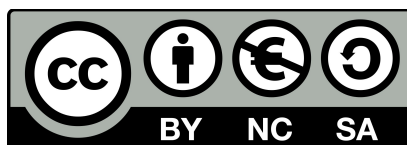
Grau en Multimèdia

Especialitat en Comunicació Visual i Creativitat

Autor: ALBERTO ARANDA OSUNA

Consultor: Llogari Casas

GENER DE 2016



Aquest projecte està subjecte a una llicència [Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](#)

No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de realitzar amb una llicència igual a la que regula l'obra original (en aquest cas el present projecte).

La present memòria, així com el projecte BigBug (continguts, disseny, desenvolupament i producció), han estat realitzats exclusivament i íntegrament per l'Alberto Aranda Osuna.

La música (música del menú i música principal del joc) és obra de l'Oriol Granados (oriolgb1@gmail.com).

Els efectes sonors (descarregats de <http://www.freesound.org>) es troben sota llicències CC0 1.0 (Domini públic) i CC BY 3.0 (Reconeixement).

Les tipografies utilitzades han estat la Grinched (<http://www.sharkshock.net/fontpage.html>) i la Wicked Scary Movie (<http://cargocollective.com/ajpaglia/De-Commissioned-Fonts>), ambdues sota llicències per usos no comercials.

Per tal d'aplicar textures tipus *cartoon* als personatges, he fet servir un recurs de l'Asset Store de Unity anomenat Toon Shader Free (<https://www.assetstore.unity3d.com/en/#!/content/21288>)

Abstract

L'objectiu del present projecte és la realització d'un petit videojoc de tipus *survival shooter* en tercera persona, on l'objectiu principal és eliminar els enemics mitjançant la interacció amb el personatge principal a través de controladors físics (teclat d'ordinador i ratolí) mitjançant els quals el personatge es podrà moure i disparar. El videojoc inclou el seu propi *character design*, el modelatge i l'animació 3D, l'apartat gràfic, així com la programació, conceptes que han estat treballats i adquirits al llarg del grau en Multimèdia.

Paraules clau: animació 3D, Autodesk Maya, art conceptual, modelatge 3D, rigging, programació, C#, survival shooter, TFG, Unity, UOC, Photoshop, videojoc.

Index

1. Introducció	8
2. Descripció	9
2.1 Definició	9
2.2 Estètica	10
2.3 Jugabilitat	12
3. Objectius	14
3.1 Principals	14
3.2 Secundaris	14
4. Marc teòric	15
5. Continguts	17
5.1 Continguts del projecte	17
5.2 Continguts del videojoc	17
6. Metodologia	20
7. Arquitectura de l'aplicació	22
8. Tecnologia	23
9. Planificació	24
10. Procés de treball	25
10.1 Preproducció	25
10.2 Producció	25
10.3 Postproducció	25
11. Prototips	26
11.1 Lo-Fi	26
11.2 Hi-Fi	28
12. Perfils d'usuari	34
13. Usabilitat	35
13.1 Proves	36
14. Test d'usuari	38
15. Versions de l'aplicació	44
15.1 Pre-Alfa	44
15.2 Alfa	44
15.3 Beta	48
16. Bugs	52
17. Projecció a futur	53

18. Pressupost	55
19. Conclusions	56
Annex 1. Lliurables del projecte	58
Annex 2. Codi font (extractes)	59
Annex 3. Llibre d'estil	64
Annex 4. Glossari	68
Annex 5. Bibliografia	69

Figures i taules

Index de figures

Figura 1: <i>Concept art</i> del protagonista	10
Figura 2: Model 3D del protagonista	10
Figura 3: <i>Concept art</i> de l'enemic	11
Figura 4: Model 3D de l'enemic	11
<i>Figura 5: Vista general de l'escenari</i>	12
Figura 6: Arbres amb formes punxegudes i flors triangulars	12
Figura 7: Navegació de l'aplicació	22
Figura 8: Navegació de la narració del joc	22
Figura 9: Prototip Lo-Fi de la pantalla inicial del videojoc	26
Figura 10: Prototip Lo-Fi de la pantalla de selecció del nivell de dificultat	27
Figura 11: Prototip Lo-Fi de la pantalla de joc amb la GUI	28
Figura 12: Prototip Hi-Fi del menú principal	28
Figura 13: Prototip Hi-Fi de la pantalla d'instruccions	29
Figura 14: Prototip Hi-Fi de la pantalla de llicències i recursos utilitzats	29
Figura 15: Prototip Hi-Fi de la pantalla de crèdits	30
Figura 16: Prototip Hi-Fi de la pantalla amb l'argument del videojoc	30
Figura 17: Prototip Hi-Fi del menú de selecció del nivell de dificultat	31
Figura 18: Prototip Hi-Fi de la pantalla de joc amb la GUI	32
Figura 19: Prototip Hi-Fi del menú de pausa	32
Figura 20: Prototip Hi-Fi del menú de fi de partida	33
Figura 21: Gènere	38
Figura 22: Perfils de jugador	39
Figura 23: Menús i navegació	40
Figura 24: Comentaris sobre menús i navegació	41
Figura 25: Jugabilitat	41
Figura 26: Comentaris sobre la jugabilitat	41
Figura 27: Diversió	42
Figura 28: Comentaris sobre la diversió	42
Figura 29: Estètica	43
Figura 30: Valoració global	43
Figura 31: Comentaris finals	43
Figura 32: Blend Tree a Unity	45
Figura 33: Tecles de moviment	45
Figura 34: Posició del ratolí	45
Figura 35: Moviment de la càmera	46
Figura 36: Moviment de l'enemic	46
Figura 37: Funció per treure vida	47
Figura 38: Enemics genèrics	47
Figura 39: Generar enemics	48

Figura 40: Funció perquè el jugador canviï de color	49
Figura 41: Indicadors de la GUI	49
Figura 42: Build Settings	51
Figura 43: Carregar una escena	51

Index de taules

Taula 1: Dates clau de la planificació del projecte	24
Taula 2: Característiques tècniques dels models utilitzats	37
Taula 3: Dispositius utilitzats pels usuaris	40
Taula 4: Pressupost hipotètic del projecte. Perfils	55
Taula 5: Pressupost hipotètic del projecte. Número d'hores i cost total	55

1. Introducció

Des de fa molts anys sóc un gran aficionat als videojocs, i al llarg de la meua vida he jugat de tot tipus, en diferents plataformes i formats. Tot i que sempre he tingut molta curiositat per saber com es feia un videojoc, mai m'havia plantejat la possibilitat de crear-ne un fins que he tingut l'opció d'escollir la temàtica del TFG. Després d'haver fet totes les assignatures del grau en Multimèdia i gràcies a haver adquirit coneixements en modelatge/animació 3D (el que més m'agrada) i en programació, vaig creure en la viabilitat de crear el meu propi videojoc, podent conèixer com és el procés de creació des d'un punt de vista teòric-pràctic.

La idea de combinar models en 3D amb interacció és una assignatura que encara tenia pendent i que no es toca en el grau, la qual cosa em va empènyer una mica més a prendre aquesta decisió. Volia treballar en un projecte que integrés l'ús de personatges 3D i la possibilitat de programar un entorn en el qual poder interactuar amb aquests. La idea de crear un món virtual on poder veure els meus propis personatges en moviment i poder interactuar amb ells era una cosa que em seduïa, i va provocar que escollís l'opció del videojoc en comptes del curt d'animació 3D (era l'altra idea que tenia en ment) la qual cosa ja havia fet (més o menys) en altres assignatures del grau.

A demés, és la primera vegada que he pogut modelar els meus propis personatges i dotar-los de vida mitjançant l'animació. Durant el grau només havia animat personatges de tercers, ja fossin descarregats d'Internet o bé facilitats per la universitat. Així doncs, era una oportunitat que no volia deixar escapar, ja que tot i que comporta una càrrega de feina considerable, volia ser capaç de poder realitzar tot el procés de creació de personatges, des de la idea fins a l'animació, passant pel modelatge i el *rigging*.

No obstant, aquest projecte no pretén esdevenir un videojoc innovador en quant a jugabilitat i interacció amb l'usuari. D'una banda perquè no es disposa del temps suficient i perquè no tinc tots els coneixements de programació necessaris per portar-ho a terme; d'altra banda perquè des de que vaig decidir fer el videojoc, tenia clar que on volia invertir la major part del temps era en el *game design* i l'animació de personatges, i donar així un toc personal i creatiu en aquest aspecte.

2. Descripció

2.1. Definició

El present projecte consisteix en la creació d'un videojoc (producte multimèdia) que incorpora contingut digital 3D (imatge 3D i animació), música i efectes sonors (àudio), menús i indicadors gràfics d'interfície (disseny gràfic i text), i interacció (programació). És un videojoc tipus *survival shooter* en tercera persona dissenyat i creat per ser jugat amb dispositius d'escriptori (ordinadors de sobretaula i portàtils) per les plataformes Windows i MAC OS X. Vaig escollir aquest gènere de videojoc perquè, donat el temps disponible, vaig pensar que seria una idea viable i realista, tenint en compte que també tenia que dissenyar i crear tota la part de 3D. A demés, els *shooters* han estat sempre un format que m'ha agradat i entretingut molt, i volia crear el meu propi.

L'argument del videojoc gira entorn a la figura d'un jardiner que planta unes flors que provoquen que els insectes augmentin les seves dimensions i es tornin agressius al menjar o entrar en contacte amb aquestes. A causa d'això, el protagonista inventa una màquina (un tipus de pistola fumigadora) amb un disparador molt potent i una reserva de pesticida gran, per fer així front als insectes i aconseguir que tot torni a la normalitat. El rerefons de l'argument és l'actual/futura situació de la indústria agrícola, on s'investiga i es treballa en modificacions genètiques de les llavors per augmentar beneficis empresarials (un exemple seria l'empresa Monsanto¹).

En el videojoc hi ha només 2 tipus de personatges:

- el jardiner (protagonista), un home gran i corpulent que s'encarrega d'eliminar els insectes mutants amb la seva pistola fumigadora. L'escenari on es desenvolupa el joc és la casa del protagonista, la qual es troba dins d'un gran jardí.

- els insectes (enemics), són una barreja entre formiga i mosca, els quals han mutat a causa del contacte amb les plantes i han crescut de manera desorbitada. El seu caràcter ha esdevingut agressiu i violent. Hi ha de dos colors diferents.

1 <https://es.wikipedia.org/wiki/Monsanto>

El nom de BIGBUG se'm va ocórrer arrel de voler barrejar els conceptes INSECTES i GRANS, caient en el compte de que insectes (o bitxos) en anglès és *bug* i gran és *big*. A demés, penso que com a nom comercial també pot ser adequat, ja que és curt, fàcil de pronunciar i és en anglès.

2.2. Estètica

L'estètica del videojoc és de tipus *cartoon*, tant pel que fa als personatges i les animacions, com a l'escenari i els objectes. Això ho he aconseguit amb proporcions corporals exagerades i un pel irreal en els personatges i formes geomètriques bàsiques en els objectes, a demés de crear animacions que no persegueixen de manera obsessionada el realisme, sinó un grau de personalitat alt. Per remarcar més aquest tipus d'estètica, he utilitzat un estil *low poly* en la resolució de les malles, a demés d'aplicar unes textures que semblen extretes de dibuixos animats.

En el cas del protagonista, volia crear un individu gran i corpulent, amb braços i mans exageradament grans i cames i peus més petits. D'aquesta manera, l'objectiu era aconseguir que quan correués, ho fes d'una manera més "simpàtica" i un pel "irreal" (cosa que deriva en un estil *cartoon*), ja que al tenir les cames curtes, les ha d'aixecar molt alhora de córrer, cosa que provoca uns moviments exagerats. A demés, el moviment accentuat del cap també és una característica força *cartoon*, ja que l'ésser humà real gairebé no el mou quan es desplaça. Això crea personalitat en el personatge, que era l'objectiu principal alhora de dissenyar-lo.

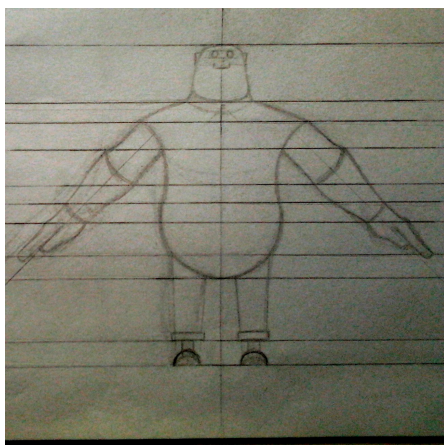


Figura 1: *Concept art* del protagonista



Figura 2: Model 3D del protagonista

D'altra banda, pel que fa al disseny dels enemics, tot i estar inspirat en insectes reals, tenen un cert comportament i disseny humanoide, ja que en comptes de tenir varies potes, només en tenen 4 (dos braços i dos cames). El motiu principal d'això és que alhora de fer el modelatge i el *rigging* és molt més pràctic treballar amb poques articulacions i amb personatges bípedes. El disseny també és força *cartoon* (cap molt gran i extremitats com peus i cames molt petits), la qual cosa ajuda a que el projecte tingui més coherència visual.



Figura 3: *Concept art* de l'enemic

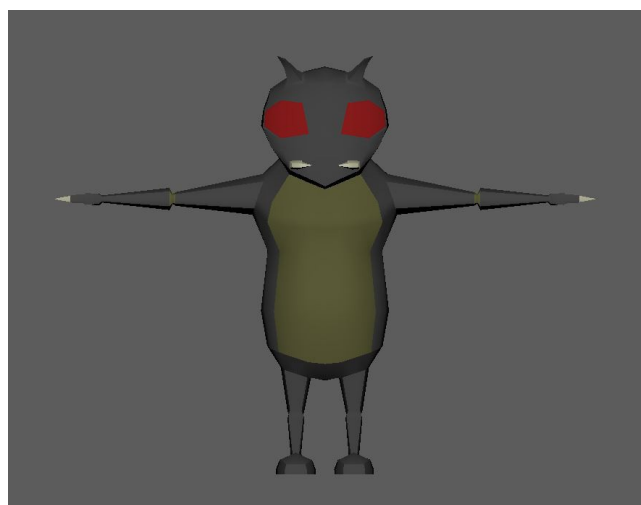


Figura 4: Model 3D de l'enemic

L'escenari està format pel jardí i la casa on viu el jardiner. A demés, hi ha altres objectes relacionats amb la temàtica com tallagespes, una mànega, bancs, arbustos, arbres, pedres i flors. Tots aquests objectes estan dissenyats i modelats en format *low poly* (al igual que els personatges), deformant lleugerament formes geomètriques bàsiques i aconseguint arbres amb copes punxegudes, flors d'aspecte triangular o cases molt quadriculades.



Figura 5: Vista general de l'escenari



Figura 6: Arbres amb formes punxegudes i flors triangulars

2.3. Jugabilitat

Pel que fa al funcionament i jugabilitat del videojoc, aquest ha estat dissenyat per ser jugat amb ordinador. Mitjançant les tecles WASD del teclat, l'usuari controla el moviment del protagonista, desplaçant-lo a través de tot l'escenari, i amb el moviment del ratolí es controla la rotació (cap a quina direcció es vol anar) i a la vegada s'apunta als enemics. Per disparar, s'ha de fer clic en el botó esquerre del ratolí.

Al principi de la partida apareix un marcador de puntuació a la cantonada superior dreta de la pantalla, el qual s'anirà incrementant a mesura que es maten insectes. D'aquesta manera, l'objectiu del jugador és eliminar el màxim número d'enemics evitant que aquests l'ataquin, ja que només hi ha una única (i no gaire gran) barra de vida sense opció a regenerar-se. Mentre aquesta no s'acabi, el joc està programat perquè es continuï creant més enemics que apareixeran per la pantalla. Al ser un *survival*, un cop que la vida del personatge s'acaba, finalitza la partida. Al final de la partida es pot veure quants enemics s'han matat, reptant a l'usuari a millorar la seva millor puntuació.

El videojoc està pensat perquè l'usuari no es pugui quedar quiet disparant als enemics (la qual cosa seria massa fàcil), ja que l'escenari tampoc és gegant i a més els enemics estan programats perquè sempre persegueixin al jugador, desplaçant-se a la posició actual que aquest ocupa. L'usuari, per tant, ha d'anar recorrent el mapa, trobant petits amagatalls i obstacles (tallagespes, bancs...) que li serviran per guanyar una mica de temps mentre segueix disparant als enemics.

Per qüestió de temps, només hi ha una única pantalla amb tres nivells de dificultat (fàcil, intermedi i difícil), els quals es diferencien en funció dels cops que el jugador ha de disparar a un enemic per matar-lo i la freqüència amb la que aquests es van generant en la pantalla (hi ha dos punts de generació de personatges en l'escenari).

He de dir que, tot i que m'hagués agradat poder crear també el videojoc per dispositius mòbils, això comportava un redisseny de la jugabilitat, la qual cosa sobrepassava el límit de temps disponible pel present projecte. En un futur, i amb més temps, potser seria més factible adaptar les característiques del videojoc perquè pugui ser jugat en dispositius tàctils.

3. Objectius

3.1. Principals

- Crear un producte multimèdia viable, en aquest cas un videojoc
- Dotar al projecte d'un sentit, creant un argument i uns continguts que plasmin la idea inicial
- Dissenyar, modelar, *riguejar* i animar personatges propis
- Dissenyar i modelar escenaris i objectes inorgànics propis que contribueixin en una posada en escena òptima del videojoc
- Aconseguir que tot el conjunt del projecte adquireixi una coherència visual i artística, així com desenvolupar un estil propi pel que fa a l'estètica i al disseny
- Dissenyar i desenvolupar una estructura de navegació i interacció a través de la qual l'usuari pugui descobrir totes les possibilitats del videojoc
- Saber programar un disseny d'interacció entre usuari i màquina per mitjà del programa Unity, integrant models 3D, sistemes de partícules i elements gràfics

3.2. Secundaris

- Descobrir d'una manera teòric-pràctica com es treballa en la creació i desenvolupament d'un videojoc
- Adquirir experiència i destresa en la direcció de projectes
- Aconseguir que el videojoc esdevingui un producte amb garanties d'èxit comercial

4. Marc teòric

L'elecció de la temàtica del TFG té lloc en un moment de revolució en el món dels videojocs.

Si bé és cert que fa uns anys la creació de videojocs era només cosa de grans estudis i productores (recolzats econòmicament per distribuïdores i altres grans empreses) aquesta tendència ha anat canviant fins a l'actualitat gràcies a factors com els avenços tecnològics, l'abaratiment del hardware i software, i sobretot l'aparició de sistemes de distribució digital a Internet. Tot plegat ha derivat en l'aparició d'una escena anomenada "independent" (o *indie games* en anglès), on ja no fa falta una gran quantitat de diners per crear i distribuir un videojoc. Fent una mica d'història, i tal i com apareix en un article de la web EuroGamer.es², va ser el 2008 quan l'escena *indie* va cobrar un gran impuls gràcies al llançament del videojoc Braid³ del desenvolupador Jonathan Blow dins de la plataforma de Microsoft *Xbox Live*. Va ser a partir d'aquí (gràcies a la gran acollida i a les bones crítiques del videojoc) quan els dissenyadors, usuaris i *gamers* es van començar a plantejar el fet de que potser la llavors indústria del videojoc podria tenir una alternativa viable. Degut a la impossibilitat de que els desenvolupadors *indie* poguessin competir amb les grans productores pel que fa a nivell de qualitat gràfica del producte, aquests van començar a apostar fort per la innovació i l'experimentació artística i creativa.

És difícil establir una definició exacta del que és un joc *indie*, ja que no hi ha cap llibre ni manual on aparegui, potser perquè el concepte en sí mateix és canviant i ambigu. Popularment parlant s'ha arribat a un tipus d'acord on es defineix a un videojoc *indie* segons els següents atributs:

- un videojoc que està fet per poques persones o grups molt petits
- un videojoc amb gràfics de baixa qualitat
- un videojoc on la història és brillant
- un videojoc que es considera art

2 <http://www.eurogamer.es/articulos/la-generacion-indie>

3 [https://es.wikipedia.org/wiki/Braid_\(videojuego\)](https://es.wikipedia.org/wiki/Braid_(videojuego))

Però actualment aquests punts no són patrimoni únic dels videojocs *indies*. Si bé és cert que molts títols contenen alguns o tots els punts anteriors, hi ha jocs “tradicionals” (no *indies*) recolzats per grans editores que també poden estar fets per grups petits, o que els gràfics poden ser dolents o bé que l'argument pot ser boníssim. El que realment fa a un joc *indie* és l'explicació que es troba a la definició d'aquest en la web GamerDic⁴ on es cita el següent:

“Del inglés indie, abreviatura de independent (independiente).

*1. Videojuego realizado por un estudio que trabaja de manera independiente, es decir, **sin ceñirse a las órdenes o pautas de un tercero (generalmente una editora) que influya en el proceso de creación.***

2. Estudio de desarrollo que realiza videojuegos indie.”

El que està ressaltat en negreta és el que fa realment a un joc independent. Ja pot ser realitzat per moltes persones, ja pot tenir molt bons gràfics, l'argument pot ser dolentíssim, però si el videojoc està dirigit tal i com vol el creador o creadors, fent totalment el que volen i sense cap intervenció d'interessos comercials d'entremig, és un videojoc *indie*.

4 <http://www.gamerdic.es/termino/indie>

5. Continguts

Pel que fa al contingut del projecte, hi ha dos grans apartats: continguts del projecte i continguts del videojoc.

5.1. Continguts del projecte

En aquest apartat es llisten els documents que integren el projecte a nivell global:

- La memòria del projecte
- El videojoc en format .exe (Windows) i .app (Mac)

5.2. Continguts del videojoc

A continuació es citen els diferents elements necessaris per a la producció del videojoc:

- El *concept art*. Disseny dels personatges que després són modelats en 3D
 - Esbós del protagonista
 - Esbós del insecte
 - Esbós de l'escenari i objectes inorgànics
- Models 3D. En aquest apartat trobem els elements 3D necessaris
 - Model 3D del protagonista
 - Model 3D de la pistola fumigadora
 - Model 3D de l'insecte
 - Escenari i models inorgànics 3D
 - Cases
 - Arbres
 - Arbustos
 - Tallagespes
 - Tanques
 - Bancs
 - Flors
 - Pedres
 - Mànega

- Interfície gràfica del joc
 - Menú d'inici
 - Fons de pantalla
 - Botons
 - Títols
 - Textos
 - Logotip BigBug (flor amb calavera al mig)
 - Barra de vida
 - Barra de vida
 - Imatge del personatge protagonista (avatar)
 - Puntuació
 - Text amb la puntuació
 - Imatge dels enemics (avatar)
 - Menú de pausa
 - Botons
 - Títols
 - Textos
 - Menú de fi de la partida
 - Botons
 - Títols
 - Textos

- Tipografies
 - Tipografia del títol del videojoc: Grinched
 - Tipografia de títols, textos i botons: Wicked Scary Movie

- L'àudio del videojoc
 - Efectes sonors
 - Música del menú principal i del videojoc

- La documentació de codi C# (scripts)
 - PlayerMovement.cs, el qual s'encarrega del moviment del jugador
 - PlayerHealth.cs, el qual s'encarrega de controlar la salut del jugador
 - PlayerShoot, el qual s'encarrega de que el jugador pugui disparar i fer mal als enemics

- CameraMovement.cs, el qual s'encarrega del moviment de la càmera
- EnemyMovement.cs, el qual s'encarrega del moviment dels enemics
- EnemyAttack.cs, el qual s'encarrega de que l'enemic pugui atacar al jugador i fer-li mal
- EnemyHealth.cs, el qual s'encarrega de controlar la salut dels enemics
- SpawnEnemy.cs, el qual s'encarrega de generar enemics
- NumKilledEnemies.cs, el qual s'encarrega de portar el compte dels enemics que el jugador ha matat
- PauseMenu.cs, el qual gestiona el menú de pausa del joc
- GameOverMenu.cs, el qual gestiona el menú de fi de la partida (scripts GameOverMenuEasy, GameOverMenuMedium i GameOverMenuHard pels diferents nivells de difucltat)
- MenuButtons.cs, el qual gestiona el comportament dels botons i la navegació per la interfície (MenuButtonsEasy, MenuButtonsMedium i MenuButtonsHard pels diferents nivells de dificultat)

6. Metodologia

Per tal d'aconseguir els millors resultats i trobar el menor número d'inconvenients possibles, el mètode que s'ha seguit en aquest projecte pel que fa al disseny i a la programació consisteix en la realització de les diferents tasques o processos amb elements de prova, els quals es troben sense acabar o en una forma molt bàsica o genèrica. En el desenvolupament d'un projecte com és un videojoc cal optimitzar el temps, cal fer moltes proves abans de treballar amb els elements finals, ja que si falla alguna cosa, sempre hi haurà més marge d'error per solucionar-ho. És un tipus de metodologia que he après a utilitzar un cop iniciat el projecte, ja que al principi no ho vaig contemplar i vaig perdre un temps que després és irrecuperable (especialment amb el modelatge i *rigging* del protagonista).

Unity és un software ideal per seguir aquest tipus de metodologia, ja que és bastant senzill reemplaçar elements i valors. En el cas dels enemics, al principi vaig fer servir formes geomètriques bàsiques pròpies d'aquest software en comptes dels models definitius. Un cop tota la interacció amb el personatge principal estava creada, vaig modelar l'enemic en 3D i el vaig substituir a Unity enlloc de la forma geomètrica. Així mateix, en el cas dels menús i la interfície gràfica d'usuari vaig fer el mateix, creant en primer lloc models amb elements genèrics del programa, i substituint-los més tard per les formes definitives fetes amb Photoshop i Illustrator.

Pel que fa als elements 3D han estat modelats utilitzant Autodesk Maya (versió 2014 i 2016). És convenient afegir que hi ha unes quantes diferències entre modelar per un curt o per un videojoc, i la principal radica en el número de polígons que conté el model. En un videojoc, tot polígon conta, i s'ha d'intentar que la malla del model sigui el més baixa possible en número de polígons (en el modelatge per curts també, però no és tant crucial). En el cas dels videojocs el render és continu, té lloc a mesura que l'usuari interactua amb joc, i per tant la capacitat computacional de les màquines (ja sigui una videoconsola, ordinador, etc.) s'ha de poder fer càrrec de l'apartat gràfic a cada fotograma. A més s'ha de tenir en compte que Unity, un cop importat el model, converteix els *Quads* de Maya a triangles, per tant hi haurà encara un número de polígons més alt.

Pel que fa al *rigging*, s'ha utilitzat un esquelet *HumanIK* de Maya pel protagonista i l'enemic, afegint un *joint* més al final dels peus per no tenir problemes alhora de crear els controladors. Per l' *skinning* s'ha fet servir l'opció *Smooth Bind* de Maya, i assignar així les àrees d'influència de cada *joint*. Els controladors han estat creats manualment.

Les animacions dels personatges han estat realitzades amb Maya 2016. Pel protagonista han estat necessàries 4 (córrer cap endavant, córrer cap endarrere, córrer de costat i morir); pels enemics només ha fet falta crear l'animació de caminar atacant amb els braços. Seguint la mateixa metodologia citada anteriorment, les primeres animacions es van realitzar de forma bàsica per comprovar si tot funcionava correctament a Unity. Un cop testejat, es van crear les animacions finals.

Finalment, abans de crear l'escenari final, vaig crear un simple pla a Unity per tal de trobar les mides adequades corresponents a les de Maya, ja que no són les mateixes, i fent proves vaig trobar l'escala corresponent. Després vaig crear l'escenari definitiu.

7. Arquitectura de l'aplicació

L'arquitectura de l'aplicació consisteix en un menú inicial que incorpora una navegació molt senzilla, i que porta a l'usuari a la pantalla de joc, la qual incorpora un botó de pausa amb el seu corresponent menú emergent, i el menú de fi de partida quan el jugador mor.

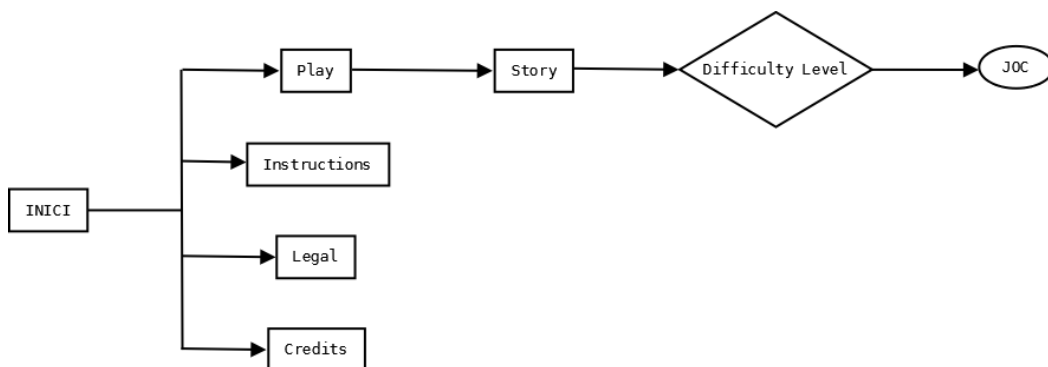


Figura 7: Navegació de l'aplicació

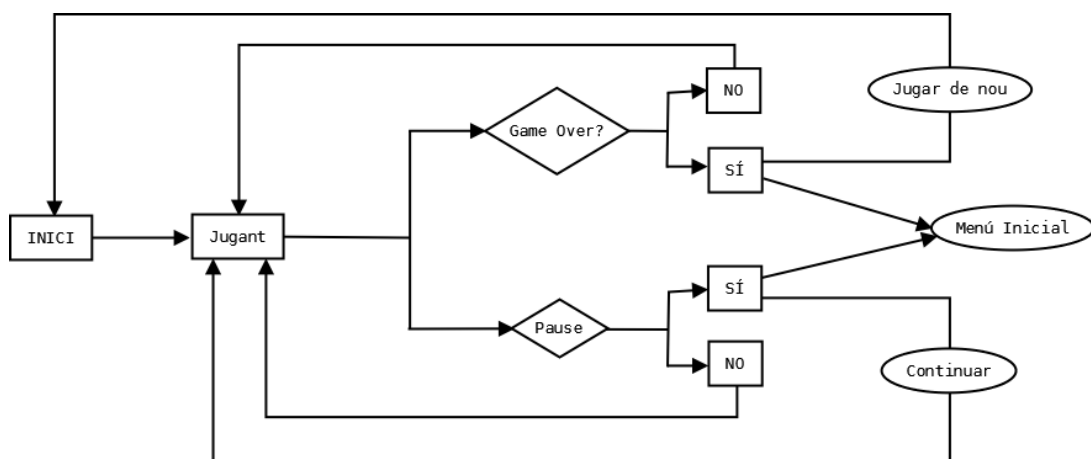


Figura 8: Navegació de la narració del joc

8. Tecnologia

A continuació s'enumeren i expliquen els softwares utilitzats pel present projecte, així com les diferents tasques per les quals s'han fet servir. També es cita el hardware (en aquest cas el meu ordinador) amb el qual s'ha portat a terme la creació de l'aplicació.

Software

Autodesk Maya 2014 i 2016. Software de modelatge i animació 3D amb el qual s'han realitzat les següents tasques:

- Modelatge del personatge principal, enemics, escenari i objectes inorgànics
- *Rigging* dels personatges i creació de controladors
- Animació dels personatges

Unity 5.1.3f1. Software de disseny i desenvolupament de videojocs, amb el qual s'han realitzat les següents tasques:

- Tota la part de programació del videojoc, mitjançant el seu editor i Visual Studio 2015 (ús del llenguatge C#)
- Creació dels sistemes de partícules corresponents a les explosions dels enemics quan moren i quan són ferits
- Aplicació de les textures tipus *cartoon*

Adobe Photoshop. Software d'edició d'imatges i creació gràfica, amb el qual s'han realitzat les següents tasques:

- Creació dels títols, textos, fons de pantalla i botons dels menus del videojoc
- Creació de les imatges (avatars) de la interfície gràfica d'usuari del videojoc

Adobe Illustrator. Software de creació gràfica vectorial, amb el qual s'han realitzat les següents tasques:

- Creació del logotip (calavera), així com les icones del menú d'instruccions

Hardware

PC M150SM Windows 7 Professional (64 bits). Processador Intel Core i7-4700MQ CPU @ 2.40GHz. Memòria RAM 16 GB. NVIDIA Quadro K3100M

9. Planificació

La planificació del projecte ha estat pensada i analitzada amb cautela, ja que normalment en el disseny i desenvolupament d'un videojoc participa més d'una persona, sobretot si els continguts són de creació pròpia. En el present projecte, àmbits tan amplis i diferents com el modelatge o la programació han estat duts a terme per una sola persona, amb la qual cosa la organització prèvia ha estat molt important. Com es pot veure en les dates clau, vaig preferir realitzar el *concept art*, modelatge i *rigging* del personatge en primer lloc, així com les seves animacions de moviment, per així començar a treballar amb Unity en el desenvolupament del videojoc. Crec que és millor enfocar-ho així que no pas fer tots els personatges i les seves animacions de cop, corrent el risc que després alguna cosa pugui estar malament o no funcioni amb Unity i s'hagi de refer tot amb la pèrdua de temps que comporta.

TASQUES	INICI	FINAL
<i>Concept art</i> del protagonista	17/09/2015	18/09/2015
PAC 01	24/09/2015	28/09/2015
Modelatge del protagonista	25/09/2015	09/10/2015
<i>Rigging</i> del protagonista	12/10/2015	28/10/2015
PAC 02	23/10/2015	26/10/2015
Modelatge de la pistola	28/10/2015	29/10/2015
Animacions de moviment del personatge	01/11/2015	6/11/2015
Moviment del personatge i enemics, matar enemics i rebre mal (Unity)	10/11/2015	26/11/2015
PAC 03	27/11/2015	30/11/2015
<i>Concept art</i> de l'enemic	01/12/2015	01/12/2015
Modelatge de l'enemic	01/12/2015	06/12/2015
<i>Rigging</i> de l'enemic	07/12/2015	10/12/2016
Animació de moviment de l'enemic	11/12/2016	11/12/2016
Animació de mort del personatge	12/12/2016	12/12/2016
Modelatge de l'escenari i objectes inorgànics	13/12/2015	17/12/2015
Puntuació, generació d'enemics i Game Over (Unity)	18/12/2015	21/12/2015
Creació de sistemes de partícules a Unity	21/12/2015	23/12/2015
Menus i GUI. Disseny i producció amb Photoshop i Illustrator, i integració amb Unity	24/12/2015	04/01/2016
Integració de l'àudio i la música	05/01/2016	06/01/2016
Test d'usuari	07/01/2015	08/01/2015
Entrega Final	07/01/2015	11/01/2015

Taula 1: Dates clau de la planificació del projecte

10. Procés de treball

El procés de treball emprat en aquest projecte (en termes de flux de treball) és igual o similar al procés que es pot seguir en un estudi d'animació o videojocs. Per tant, les diferents etapes del projecte són les següents: Preproducció, Producció i Postproducció.

10.1. Preproducció

- Definició de la idea i justificació del projecte
- Definició del gènere del videojoc
- Elaboració de l'argument del videojoc
- *Concept art* dels personatges
- Entrega de la PAC 1

10.2. Producció

- Modelatge 3D dels personatges
- *Rigging* dels personatges
- Animacions dels personatges
- Programació i desenvolupament del funcionament i interacció del joc
- Modelatge 3D de l'escenari i objectes inorgànics
- Creació dels sistemes de partícules
- Disseny, creació i integració de la interfície gràfica d'usuari i els menus
- Entrega de la PAC 2 i PAC 3

10.3. Postproducció

- Integració de l'àudio
- Test d'usuaris i usabilitat
- Entrega final del projecte

11. Prototips

11.1. Lo-Fi

A continuació es mostren els prototips de baixa fidelitat del projecte. Aquests són necessaris per poder dissenyar, d'una manera bàsica, l'aspecte gràfic de les pantalles dels diferents menús i la interfície gràfica d'usuari del joc, així com planificar la distribució dels diferents continguts i seccions. Com es pot observar, si es comparen amb els prototips d'alta fidelitat finals, han patit variacions respecte certs aspectes com algun canvi de nom en els títols i els botons, així com la inclusió d'alguns elements nous que sorgeixen del desenvolupament normal del projecte.

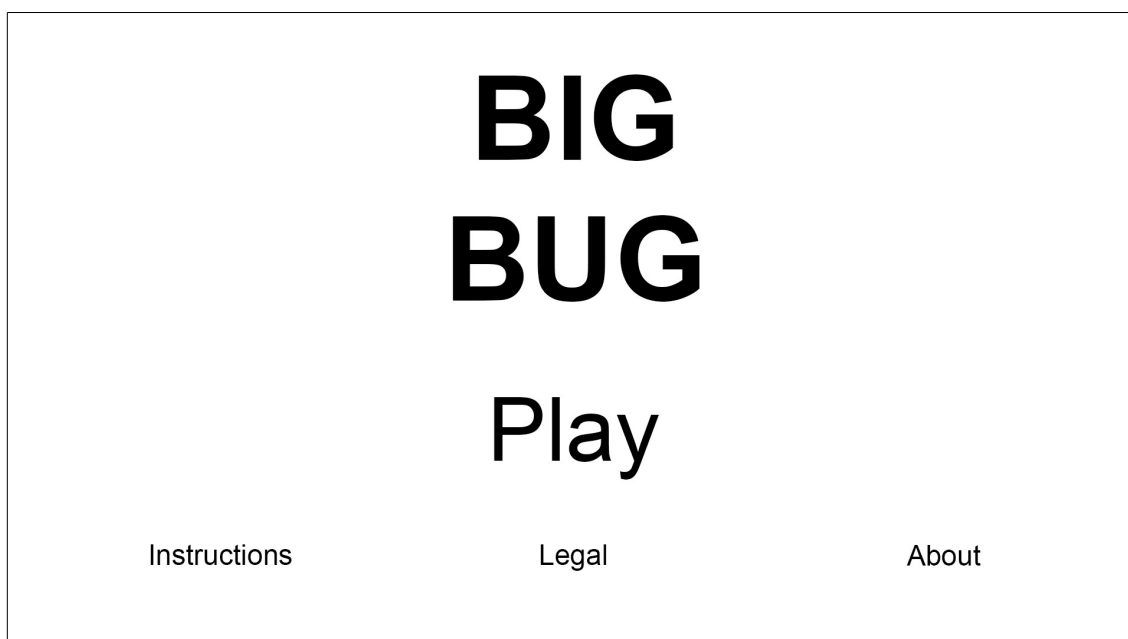


Figura 9: Prototip Lo-Fi de la pantalla inicial del videojoc

Aquesta primera pantalla inicial està formada per 4 botons. El botó *Play* porta a la selecció del nivell de dificultat, i posteriorment al joc en sí. El botó *Instructions* mostra una pantalla on s'explica com es juga al videojoc i quins controladors o tecles s'han de fer servir. El botó *Legal* mostra una pantalla que ofereix informació sobre l'apartat legal del projecte, així com el tipus de llicència i els recursos de tercers empleats. Per últim, el botó *About* (en el prototip d'alta fidelitat es pot veure com aquest nom ha estat substituït pel de *Credits*) mostra una pantalla amb informació sobre els crèdits del projecte.

En la imatge següent es pot veure la pantalla de selecció del nivell de dificultat (en el prototip final *Level* ha estat substituït per *Difficulty*, ja que em va semblar més adient), en la qual es pot escollir quin nivell es vol per començar a jugar. Com es pot observar hi ha 3 nivells: *Gardener* (fàcil), *Bug Killer* (Intermedi) i *Exterminator* (Difícil).



Figura 10: Prototip Lo-Fi de la pantalla de selecció del nivell de dificultat

Pel que fa a la interfície gràfica d'usuari (GUI) de la pantalla de joc, està formada per dos indicadors. A la cantonada superior esquerra hi ha la barra de vida del jugador i a la cantonada superior dreta hi ha la puntuació. Aquest primer prototip funciona només com a referència alhora de dissenyar el prototip d'alta fidelitat, ja que en aquest s'han afegit icones i elements gràfics que ajuden a l'usuari alhora de jugar per primer cop. S'ha de trobar un equilibri entre informar de la millor manera possible a l'usuari i deixar espai suficient a la pantalla perquè pugui jugar d'una manera òptima. En aquest cas en particular ha estat fàcil, ja que només es necessitava col·locar dos indicadors en pantalla, i per evitar confusions els he col·locat separats, però ambdós a la part superior.

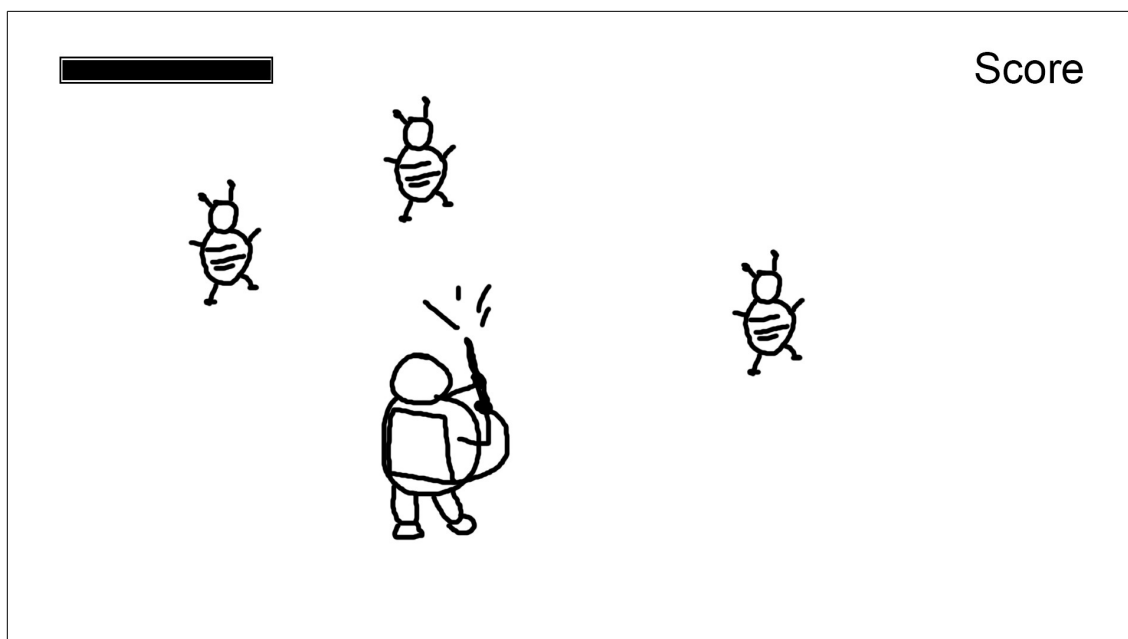


Figura 11: Prototip Lo-Fi de la pantalla de joc amb la GUI

11.2. Hi-Fi

A continuació es mostren els prototips d'alta fidelitat en l'estat final de l'aplicació.



Figura 12: Prototip Hi-Fi del menú principal

Respecte el prototip de baixa fidelitat, en aquest s'ha afegit un fons de pantalla, un títol amb una tipografia de tercers (tipografia *Grinched*) i la insígnia del videojoc (una icona que representa una flor amb una calavera al mig, fent referència a l'argument). A demés també s'han afegit els botons i una sortida del videojoc (botó *EXIT* a la part superior dreta) la qual no estava representada en el prototip de baixa fidelitat.

A continuació es mostren les pantalles de *INSTRUCTIONS*, *LEGAL* i *CREDITS*. En aquestes s'ha afegit un botó *BACK* per tal de poder tornar endarrere sempre que es vulgui.

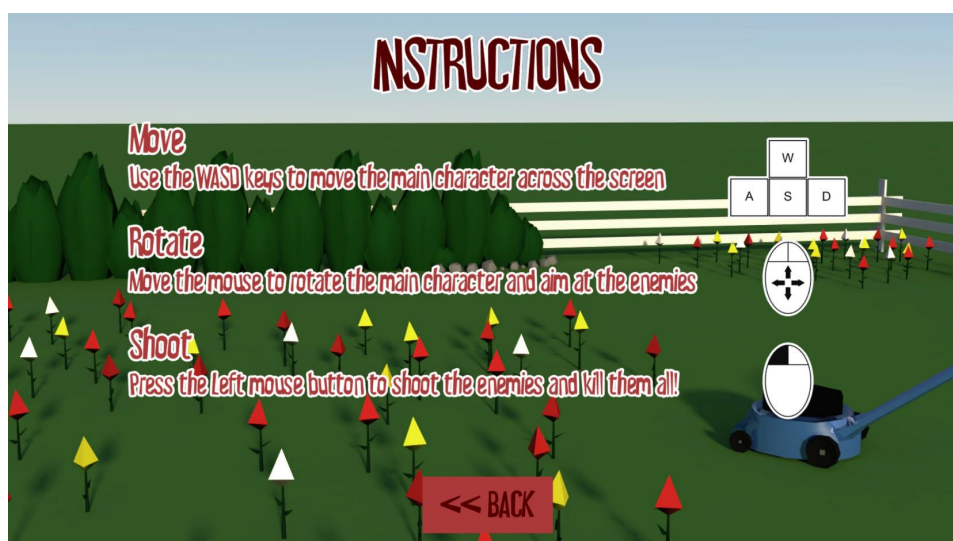


Figura 13: Prototip Hi-Fi de la pantalla d'instruccions

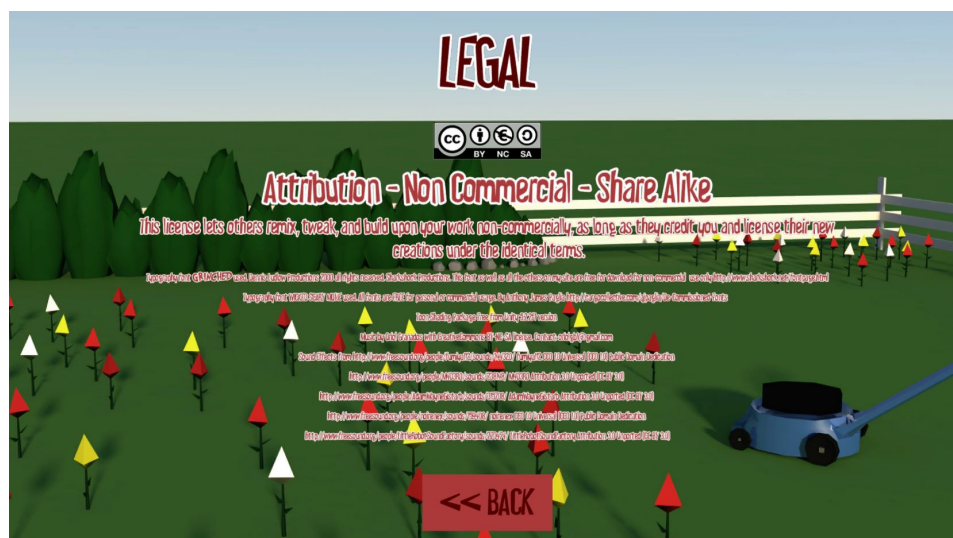


Figura 14: Prototip Hi-Fi de la pantalla de llicències i recursos utilitzats



Figura 15: Prototip Hi-Fi de la pantalla de crèdits

Respecte els prototips de baixa fidelitat, també s'ha afegit una pantalla amb una breu història sobre l'argument que envolta al videojoc. A més, hi ha dos botons, un per anar cap endarrere per si es vol consultar alguna cosa del menú principal, i un altre per anar a la pantalla de selecció del nivell de dificultat.



Figura 16: Prototip Hi-Fi de la pantalla amb l'argument del videojoc

Pel que fa a la pantalla de selecció del nivell de dificultat els únics canvis respecte el prototip de baixa fidelitat són la inclusió dels botons, el fons de pantalla i el canvi del títol. Si es fa clic en un dels botons, l'usuari accedeix directament a jugar, tot i que també hi ha l'opció de tornar endarrere si es vol.



Figura 17: Prototip Hi-Fi del menú de selecció del nivell de dificultat

A continuació es mostra el prototip d'alta fidelitat de la interfície gràfica de l'usuari en la pantalla de joc. Com s'ha comentat anteriorment, en aquest cas s'ha afegit elements gràfics com els avatars dels personatges, així com un botó de pausa a la cantonada inferior dreta (*PAUSE*). Mitjançant aquest botó l'usuari pot detenir el joc si ho necessita, oferint també la possibilitat d'abandonar la partida abans d'acabar-la.



Figura 18: Prototip Hi-Fi de la pantalla de joc amb la GUI

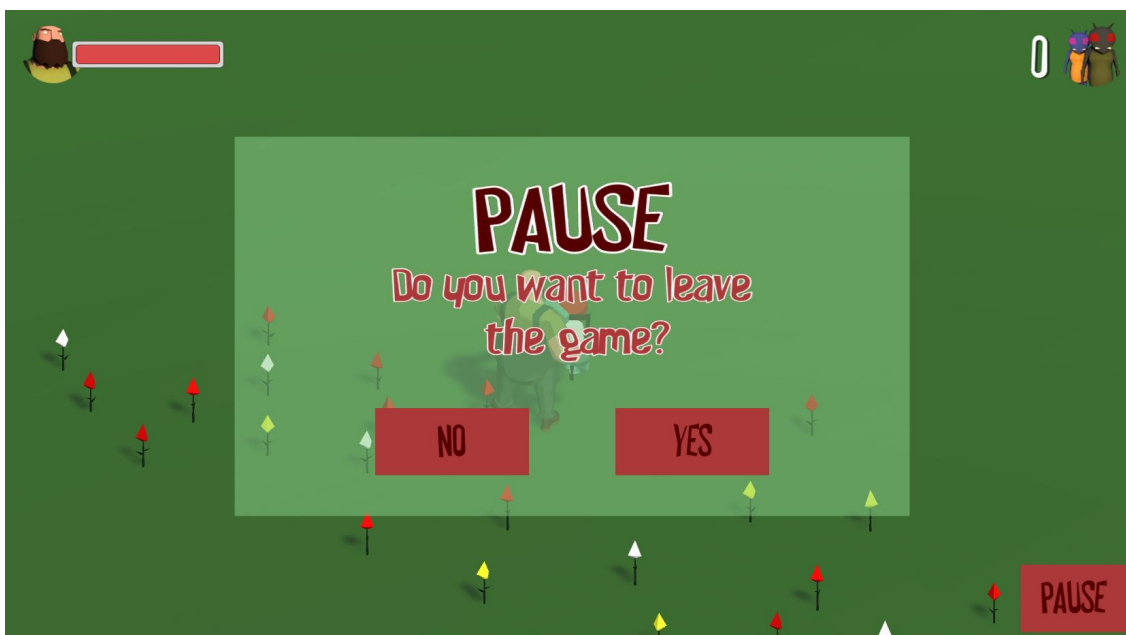


Figura 19: Prototip Hi-Fi del menú de pausa

Per últim es mostra el prototip d'alta fidelitat de la pantalla de fi de la partida, la qual apareix quan el jugador mor. Hi ha dues opcions: o bé es juga de nou fent clic a *PLAY AGAIN* o bé es torna al menú principal a través del botó *QUIT*.

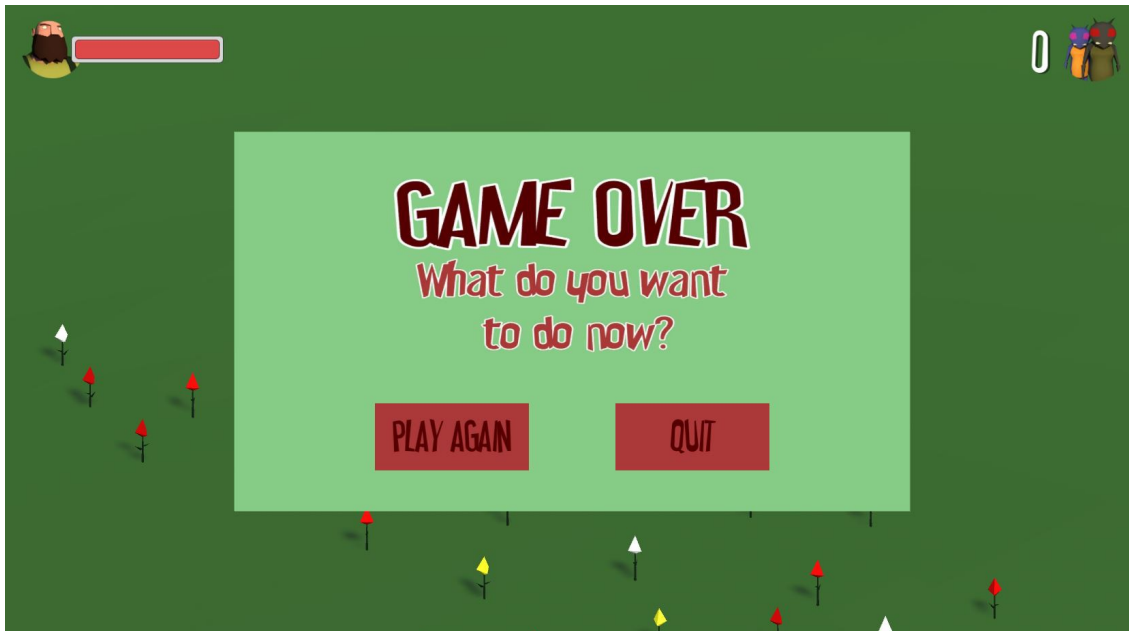


Figura 20: Prototip Hi-Fi del menú de fi de partida

12. Perfils d'usuari

BigBug és un videojoc amb un únic nivell (pantalla) i una dinàmica de joc força senzilla (ja s'han explicat els motius en apartats anteriors), i per tant, la llista de perfils d'usuari potencials és força limitada. Òbviament, aquests s'analitzen sense tenir en compte possibles i futures millores en el videojoc, ja que això es comenta en apartats següents. En aquest apartat només s'enumeren els perfils prenent com a única referència el projecte actual.

- Usuaris infantils (aproximadament entre 6 i 10 anys sense importar el gènere) que no tenen una llarga experiència en el món dels videojocs i volen començar a interactuar amb aquests. BigBug és un videojoc amb una jugabilitat senzilla (ús de 4 tecles i el ratolí), amb una funcionalitat fàcil i uns objectius de partida clars (eliminar el màxim número d'enemics sense que et matin), per tant és força ideal per aquest segment poblacional. A més, incorpora un disseny i uns personatges amb una estètica força infantil, la qual cosa també és un reclam pels més petits.

-Usuaris joves i adults (un ampli rang que va des dels 10 fins als 45 anys sense importar el gènere) que simplement volen un videojoc curt que no impliqui grans dosis de complexitat per així poder jugar en petites estones lliures a casa (fent un descans alhora d'estudiar) o a la feina. És força idoni per aquests casos, ja que servirà per descarregar una mica d'adrenalina en moments en que els usuaris puguin estar saturats i vulguin aclarir una mica els seus caps.

13. Usabilitat

Pel que fa al disseny de la usabilitat i experiència de l'usuari, l'aplicació ha estat concebuda perquè l'usuari pugui interactuar amb la interfície gràfica i els menús de forma senzilla i intuïtiva. En un videojoc, el que l'usuari vol principalment és jugar, per tant les estructures de navegació prèvia i posterior al joc han de ser el més simple possible en quant a funcionalitat, però atenent a unes demandes de disseny que han de contribuir a una coherència visual al llarg de tota l'aplicació. El que no es vol en un videojoc és que l'usuari s'avorreixi inclús abans de començar a jugar, per això els menús de navegació han de ser clars. Pel que fa a la interfície gràfica d'usuari del joc, els indicadors també han de ser precisos, intentant ocupar el menor espai de la pantalla de joc possible. En el present projecte s'han empleat alguns dels principis heurístics del guru de la usabilitat Jacob Nielsen⁵ que més s'adapten al món dels videojocs.

Els principis són els següents (extrets íntegrament de la seva pàgina web *NNGroup*⁶):

“Adequació entre el sistema i el món real

El sistema ha de parlar el llenguatge dels usuaris, amb paraules, frases i conceptes familiars a aquests, en comptes de parlar en termes propis del sistema. S'ha de seguir les convencions del món real, fent que la informació aparegui en un ordre natural i lògic.”

Aquest principi s'ha aplicat alhora de crear el contingut textual dels menús. Les paraules escollides han estat simples i familiars en el món dels videojocs, com ara *Play*, *Instructions* o *Back*. A més, la informació apareix en un ordre lògic, ja que, per exemple, no es pot començar a jugar sense saber les instruccions del joc, i per això aquestes apareixen en el menú inicial.

5 https://es.wikipedia.org/wiki/Jakob_Nielsen

6 <https://www.nngroup.com/articles/ten-usability-heuristics/>

“Disseny estètic i minimalista

Els diàlegs no han de contenir informació irrellevant o que rarament serà necessària. Cada fragment d'informació no necessària competeix amb fragments d'informació que sí ho és, la qual cosa fa que en disminueixi la seva visibilitat.”

És un videojoc senzill, per tant l'usuari no necessita informació extra que lluny de proporcionar ajuda el pot confondre més.

“Ajuda i documentació

Tot i que és millor que el sistema pugui ser utilitzat sense documentació, és possible que pugui ser necessari proveir ajuda i documentació. De ser així, aquesta informació ha de ser fàcil de buscar, exposant els passos necessaris i no ha de ser gaire extensa.”

En el món dels videojocs, la primera frase d'aquest principi podria no tenir gaire sentit, ja que sempre hi ha d'haver un mínim d'instruccions perquè l'usuari no tingui problemes alhora de començar a jugar. Tot i així, el que sí que s'ha aplicat d'aquest principi al present projecte és que aquesta informació ha de ser fàcilment localitzable (al menú principal el botó *INSTRUCTIONS*), exposant els passos necessaris (s'explica amb text i icones) i no gaire llarga (la jugabilitat és senzilla, per tant les instruccions han de ser curtes).

13.1. Proves

Les proves d'usabilitat/funcionament del videojoc han estat realitzades amb tres dispositius diferents. Amb aquestes proves volia comprovar que, en tres models diferents de màquina, l'aplicació funcionava sense importar la qualitat del dispositiu i el sistema operatiu. Òbviament, hagués estat molt millor fer proves en molts altres models diferents, però per qüestió de temps i recursos ha estat impossible. A continuació es mostra una taula comparativa amb les característiques dels models utilitzats en aquestes proves.

COMPONENT	DISPOSITIU 1	DISPOSITIU 2	DISPOSITIU 3
Marca	Azken Muga	Samsung	Apple
Model	M150SM	R540	I Mac 21,5"
Sistema operatiu	Windows 7 Professional 64 bits	Windows 7 Home Premium 64 bits	Mac OS X Yosemite
Processador	Intel Core i7-4700MQ CPU @ 2.40GHz	Intel Core i3 Dual Core 370M @ 2.26GHz	Intel Core 2 Duo
Memòria RAM	16 GB	4 GB	4GB
Targeta gràfica	NVIDIA Quadro K3100M 4GB GDDR5	ATI Mobility Radeon HD545v 512MB GDDR3	Ati Radeon HD 4670 512MB
Pantalla	15.6" 1920x1080	15.6" 1366 x 768	21,5" 1920x1080

Taula 2: Característiques tècniques dels models utilitzats

L'elecció d'aquests 3 dispositius permet provar el videojoc en Windows i en MAC OS X (les dues plataformes per les quals ha estat publicat el joc), a més de tenir la possibilitat de veure com es visualitza a diferents resolucions i mides de pantalla, així com amb diferents processadors i targetes gràfiques.

Després de provar el videojoc en els tres dispositius, vaig arribar a les conclusions següents:

→ El videojoc funciona correctament en els 3 dispositius (inclús en el segon, que és el menys potent). Això és degut, en part, a que el modelatge 3D ha estat creat en *low poly* (menys polígons que renderitzar, per tant, menys càrrega computacional) i a que els enemics, un cop morts, s'eliminen del programa gràcies a la programació amb C# a Unity.

→ Tot i així, i degut principalment al processador i la targeta gràfica, el dispositiu on funciona millor el videojoc és en l'1. Principalment, la diferència es nota en l'apartat gràfic, on els polígons estan millors definits, ja que tant en el 2 com en el 3 aquests tenen una mica d'efecte "dents de serra" (és a dir, no es veuen suavitzats). Això s'hagués pogut solucionar a Unity amb l'aplicació d'un modificador de antialiasing, però comportava molta càrrega computacional i, tenint en compte que el videojoc havia de ser jugat per diferents persones, vaig optar per no utilitzar-lo.

14. Test d'usuari

En aquest apartat s'ha fet una enquesta mitjançant l'eina online Survey Monkey a través de la qual algunes de les persones que han provat el joc (14) han valorat la seva experiència (les imatges següents han estat extretes directament d'aquesta web).

En l'enquesta es van plantejar diferents preguntes per tenir coneixement del que opinaven els usuaris en referència al funcionament de l'aplicació, la jugabilitat i l'apartat estètic i gràfic. A més, es volia conèixer també els valors demogràfics (edat i sexe), el perfil de jugador i el dispositiu que han utilitzat per jugar. Tot i que els tests d'usuari serveixen normalment per millorar l'aplicació abans d'acabar-la definitivament, en el present projecte s'han realitzat un cop el videojoc estava ja acabat, per tant aquestes proves servien per unes hipotètiques millores de cara al futur (m'hagués agradat haver-los fet abans d'acabar definitivament el joc, però per qüestió de temps no ha estat possible).

En primer lloc, comentar que el rang d'edat no ha estat una dada gaire significativa, ja que les edats dels usuaris es troben compreses entre 22 i 39 anys, amb la qual cosa hi ha segments de la població que es queden sense analitzar.

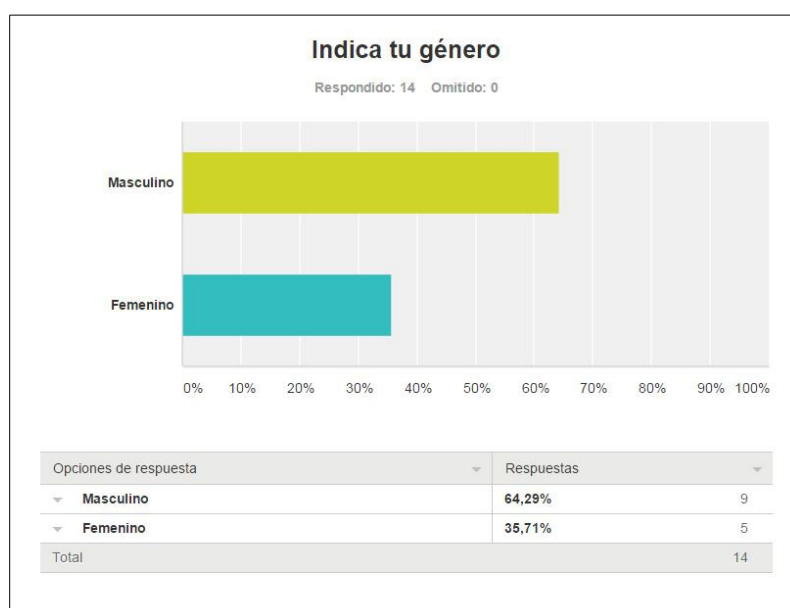


Figura 21: Gènere

Com es pot observar en la figura anterior, pel que fa al gènere dels enquestats el gènere masculí ha guanyat al femení en número de participants. A continuació es mostra una imatge amb els diferents perfils de jugador dels usuaris.

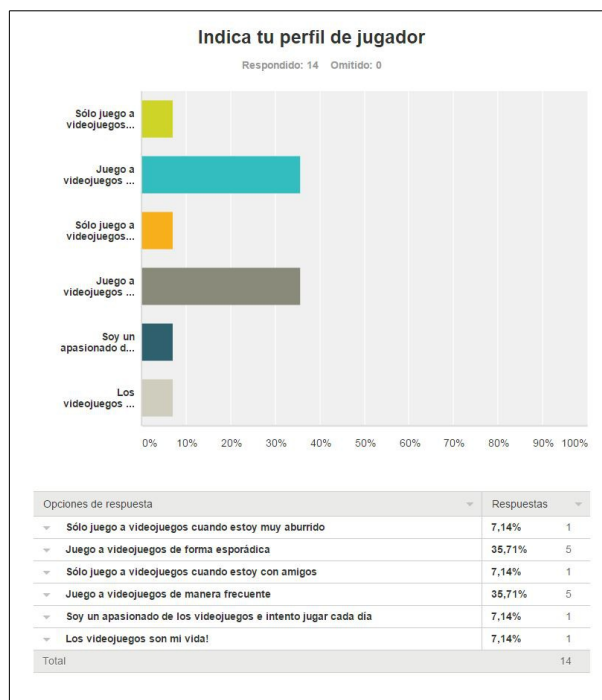


Figura 22: Perfils de jugador

Els perfils que més es repeteixen són aquells usuaris que juguen a videojocs de forma esporàdica i aquells que ho fan de manera freqüent (gairebé un 36%).

Els dispositius amb els quals ha estat provat el videojoc varien molt, i en alguns dels casos es difícil saber exactament de quin model es tracta, ja que no tots els usuaris han escrit completament les característiques.

DISPOSITIUS	SISTEMA OPERATIU
MacBook Pro	Mac OS X 10.11 El Capitan
Windows AMD FX-6300	Windows
i5	Windows 7
Intel Core i7-4790	Windows 10
-	Windows 7
Fujitsu Lifebook A series i5	Windows
Intel i7 32 GB RAM Nvidia 770	Windows 10
i5 GTX 650	Windows 7
-	Windows 10
Apple iMac	-
Dell Inspiron	Windows 10
Dell Inspiron 17"	Windows 10
Imac 21.5" Intel Core 2 Duo	Mac OS X 10.8.5

Taula 3: Dispositius utilitzats pels usuaris

Un cop acabades les preguntes sobre les dades de l'usuari, a continuació es continua amb les preguntes d'opinió de diferents aspectes de l'aplicació. La primera fa referència als menús del videojoc i la navegació.

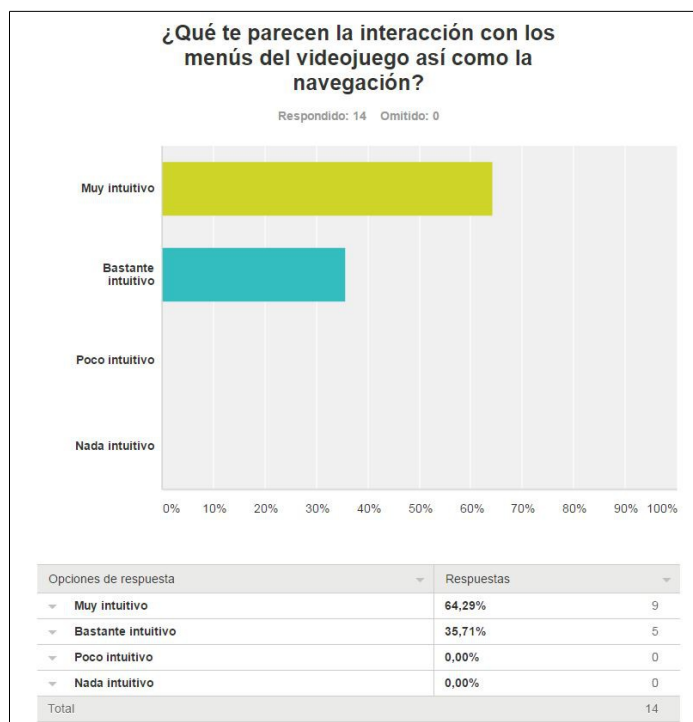


Figura 23: Menús i navegació

En general, els usuaris troben intuïtius els menús i la navegació de l'aplicació. Tot i així, alguns van deixar comentaris personals força interessants de cara a millores en el futur.

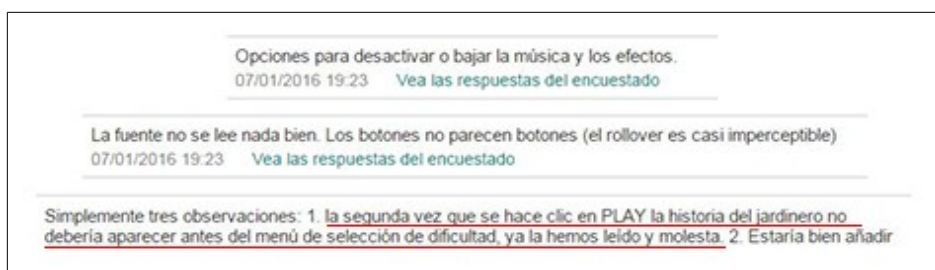


Figura 24: Comentaris sobre menús i navegació

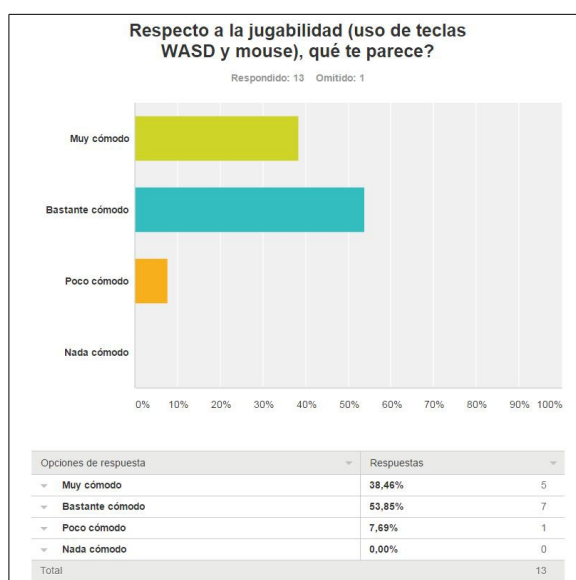


Figura 25: Jugabilitat

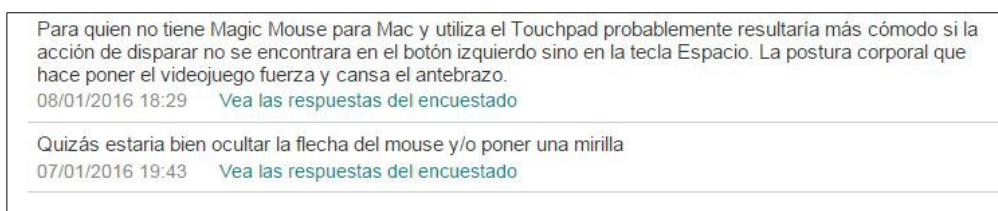


Figura 26: Comentaris sobre la jugabilitat

Pel que fa a la jugabilitat, la majoria dels usuaris pensen que és força còmode l'ús de les tecles WASD i el ratolí. Tot i així val la pena comentar el que exposa un usuari en els comentaris de la figura anterior sobre l'ús del *Touchpad* als ordinadors Mac. Personalment no sóc usuari d'Apple, i per tant és un fet amb el qual no vaig contar, però és molt cert que ahora de dissenyar un producte com un videojoc s'ha de tenir en compte tots els tipus de dispositius cap als quals pot anar dirigit. És una qüestió de la qual prenc molta nota per futures millores.

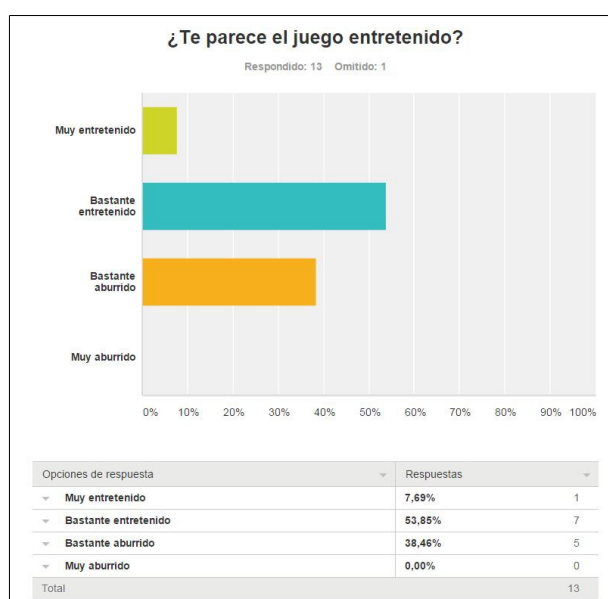


Figura 27: Diversió

Pel que fa a l'entreteniment que proporciona el joc, malgrat molts opinen que és bastant entretingut, hi ha una gran part dels usuaris que opinen que és força avorrit. En els comentaris que es mostren a continuació els usuaris suggereixen coses com la creació de més pantalles, diferents tipus d'enemics o comportaments més complexes d'aquests.

Las partidas resultan relativamente cortas independientemente de la destreza del jugador, debido a la estructuración de la aparición de personajes enemigos. 08/01/2016 18:29 Vea las respuestas del encuestado
Faltaria añadir cosas como mejoras de armas, diferentes enemigos, logros etc. 08/01/2016 16:24 Vea las respuestas del encuestado
No se si hay diferentes escenarios, sino estaria bien un ranking de puntuaciones para cada nivel de dificultad. 07/01/2016 19:43 Vea las respuestas del encuestado
Se hace un poco repetitivo al poder seguir jugando hasta que mueras... si eres bueno no se termina nunca. 07/01/2016 19:23 Vea las respuestas del encuestado

Figura 28: Comentaris sobre la diversió

En referència a l'estètica i al disseny de l'aplicació, als usuaris els hi ha agradat força, tot i que també hi ha opinions crítiques, tal i com es pot veure en la següent imatge.

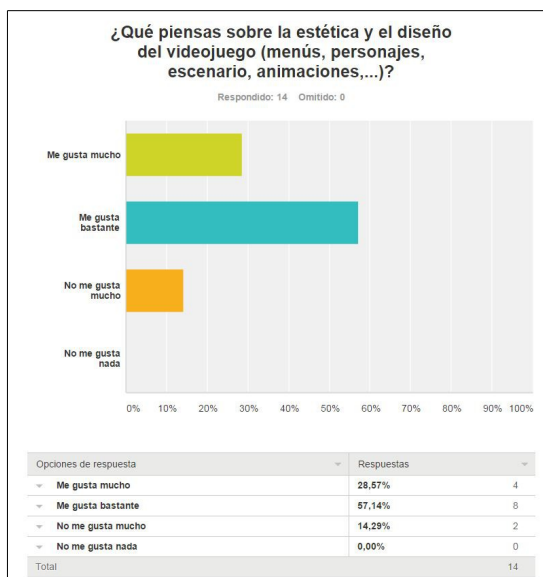


Figura 29: Estètica

En conclusió, podríem dir que la valoració global del videojoc ha estat positiva. Les parts crítiques s'han dirigit bàsicament a la necessitat d'incloure més personatges, més escenaris i pantalles, millorar el menú de navegació i crear comportaments de l'enemic més "intel·ligents".

	1	2	3	4	5	6	7	8	9	10	Total	Promedio ponderado
Valoración global	0,00% 0	0,00% 0	0,00% 0	7,14% 1	7,14% 1	14,29% 2	28,57% 4	21,43% 3	14,29% 2	7,14% 1	14	7,21

Figura 30: Valoració global

<p>Simplemente tres observaciones: 1. la segunda vez que se hace clic en PLAY la historia del jardinero no debería aparecer antes del menú de selección de dificultad, ya la hemos leído y molesta. 2. Estaría bien añadir una opción gráfica para que el jardinero recupere la salud durante la partida. 3. Los personajes enemigos aparecen desde las esquinas del escenario con una periodicidad determinada y se dirigen todos ellos hacia el personaje del jardinero, personalmente ampliaría dicha periodicidad, ya que apenas transcurre menos de 1 minuto y la partida se complica porque los enemigos van cerrando al jardinero en una zona del escenario y resulta complicado cambiar de posición, o bien haría que los personajes enemigos aparecieran con la misma periodicidad actual pero que en lugar de dirigirse directamente al jardinero quedaran pululando por el escenario y solamente atacaran al jardinero si este se encontrara a un radio determinado de ellos. En general, pienso que es un videojuego bien desarrollado, correctamente funcional y mejorable en los aspectos que he comentado. 08/01/2016 18:29 Vea las respuestas del encuestado</p>
<p>Me parece un juego muy entretenido, sobretodo para jugar con el móvil cuando vas en el metro, tren, etc... 07/01/2016 19:53 Vea las respuestas del encuestado</p>
<p>Buen trabajo simplemente añadirle algún factor más que ayude a rejugarlo varias veces. 07/01/2016 19:43 Vea las respuestas del encuestado</p>
<p>Muy entretenido y engancha un montón!!!! 07/01/2016 17:37 Vea las respuestas del encuestado</p>

Figura 31: Comentaris finals

15. Versions de l'aplicació

15.1. Pre-Alfa

En aquesta versió, vaig modelar i *riguejar* el personatge principal amb una senzilla animació per exportar-ho a Unity. D'aquesta manera es va comprovar que tot funcionés correctament, ja que tot i que Maya i Unity són compatibles, s'havia d'investigar sobre els processos necessaris per fer de forma correcta les importacions i exportacions dels models i animacions. En aquest punt s'havia de tenir en compte el format comú i el nombre de *fps* en les animacions de Maya, així com vigilar el número de polígons del model 3D. Aquesta versió, per tant, ha estat molt important alhora de poder establir la metodologia final que s'ha seguit en els processos posteriors del projecte, i tot i que la part de programació encara no s'havia començat, es van fer grans avenços respecte el modelatge, el *rigging* i els passos necessaris per establir una correcta comunicació entre els softwares.

15.2. Alfa

En aquesta versió es va crear un escenari provisional senzill a Unity sobre el qual poder treballar, i es van crear les animacions de moviment definitives del protagonista (quiet, córrer endavant, córrer endarrere i córrer lateralment). També es va començar amb la programació a Unity, creant en primer lloc la interacció de l'usuari amb el personatge principal a través de l'ús de les tecles WASD (moviment) i ratolí (rotació) per tal de desplaçar-lo per l'escenari. En aquest punt, va ser necessari utilitzar una opció de Unity anomenada *Blend Tree* amb la qual es pot establir que, segons certs paràmetres (en aquest cas la direcció cap a la qual es dirigeix el jugador) es reproduceixi una animació o una altra (si el personatge es mou cap endavant, cal reproduir l'animació de córrer cap endavant, si es mou cap a la seva dreta s'havia de reproduir l'animació de córrer lateralment, etc).

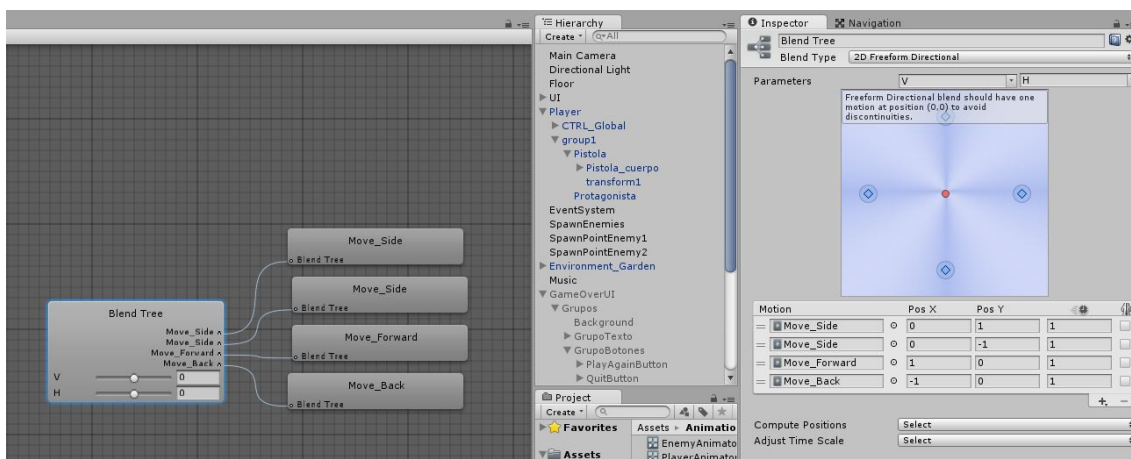


Figura 32: Blend Tree a Unity

Abans de començar amb la interacció del protagonista amb els enemics, calia crear i configurar la càmera del videojoc. Aquest punt va ser un dels més importants, ja que la càmera havia de seguir el moviment del jugador però no la seva rotació. Per tant vaig decidir crear una càmera amb vista ortogràfica (no de perspectiva) de manera que aquesta es mouria sempre en els eixos horitzontal i vertical (no en profunditat, ja que no es volia zoom) i no podria rotar. Així doncs, la càmera seguiria el moviment del jugador (tenint en compte la seva posició) a través de les tecles WASD, i a través del moviment del ratolí el jugador rotaria sobre el seu propi eix.

```
float horz = Input.GetAxisRaw("Horizontal"); //Captem els valors de les tecles esquerra/dreta o bé A/D
float vert = Input.GetAxisRaw("Vertical"); //Captem els valors de les tecles adalt/abaix o bé W/S
```

Figura 33: Tecles de moviment

```
Ray cameraRay = Camera.main.ScreenPointToRay(Input.mousePosition);
```

Figura 34: Posició del ratolí

```

void Start()
{
    offset = transform.position - target.position; //L'offset o diferència de distància serà la posició de la càmera menys la posició
                                                    //del objectiu a seguir (en aquest cas el jugador).
}

void FixedUpdate()
{
    Vector3 targetPos = target.position + offset; //A cada frame s'anirà actualitzant la
                                                    //posició del target (jugador)
    transform.position = Vector3.Lerp(transform.position, targetPos, cameraSmooth * Time.deltaTime); //S'utilitza Lerp per poder interpolat la
                                                    //posició de la càmera
}

```

Figura 35: Moviment de la càmera

El següent pas a donar era crear la interacció entre el personatge principal i els enemics. En aquest punt, donat que el model 3D de l'enemic encara no estava creat (seguint les pautes de la metodologia que s'ha explicat en un apartat anterior) es van fer servir objectes genèrics de Unity (un tipus de càpsules) per simular la presència dels enemics. En primer lloc es va crear el moviment dels enemics, cosa que es va crear utilitzant un component de Unity anomenat *Nav Mesh Agent*.

```

void Update()
{
    if (enemyHealth.currentHealth > 0 && playerHealth.currentHealth > 0) //L'enemic només es mourà quan tan l'enemic com el jugador estiguin
    { //vius
        navAI.SetDestination(player.position); //En cada actualització del Update se li diu al NavMeshAgent quina és
                                                //la posició del jugador, i a través del SetDestination anirà cap
                                                //aquesta. Per tant, l'enemic es dirigirà en tot moment a la posició
                                                //del jugador
    }
}

```

Figura 36: Moviment de l'enemic

En la imatge anterior es pot veure com funciona aquest component. Per tal que l'enemic segueixi al jugador, el programa ha de comprovar en cada moment quina és la posició del jugador, la qual s'estableix com a posició de destí de l'enemic. Aquesta posició de destí es va actualitzant constantment, ja que en el cas de que el jugador es mogui (cosa molt freqüent) aquesta haurà de canviar també.

Un cop l'enemic podia perseguir al jugador, el següent punt a resoldre era fer possible que ambdós es poguessin eliminar mútuament (treure's vida). Pel que fa al jugador, es va crear el raig (dispar) de la pistola mitjançant un component de Unity anomenat *Line Renderer*. La funció d'aquest és crear una línia a l'escenari que va des d'un punt d'inici (el canó de la pistola) fins a un punt final (l'enemic o qualsevol objecte de l'escenari amb el qual impacti el raig). A demés, es van establir dues variables que corresponien

a la salut dels personatges (jugador i enemic), per tal de poder restar vida en cas d'atac.

```
public void Damage(int qtty, Vector3 hitPoint)
{
    if (isDead) //Si l'enemic ja està mort, llavors es surt de la funció Damage perquè ja no se li pot atacar més
        return;

    enemyHurtSound.Play(); //Es reproduueix l'audio quan es fereix l'enemic

    currentHealth -= qtty; //Es resta la quantitat de dany a la salut actual

    hurtEnemyParticles.transform.position = hitPoint; //Les partícules apareixen al lloc de l'enemic on es dispara
    hurtEnemyParticles.Play();

    if (currentHealth <= 0) //Si la salut actual és menor que 0 (l'enemic ha mort)...
    {
        Death(); //...llavors es crida a la funció Death
    }
}
```

Figura 37: Funció per treure vida

Aquesta funció és una de les més importants en aquest pas (es fa servir tant pel jugador com pels enemics), ja que permet disminuir la salut en el cas que s'hagi estat atacat i a més es comprova si els personatges estan morts. En aquest punt també es va crear la barra de vida del personatge, la qual va suposar el primer element de la interfície gràfica d'usuari.



Figura 38: Enemics genèrics

15.3. Beta

En aquesta versió (la final) es van acabar de crear els aspectes de programació necessaris, es va crear el model final de l'enemic i l'escenari, i també el menú de navegació de l'aplicació i la GUI definitiva.

En primer lloc es va crear la generació automàtica d'enemics en dos punts diferents de l'escenari, de manera que fins que el jugador no estigui mort, els enemics es continuen generant seguint uns paràmetres concrets de temps. Això es porta a terme mitjançant la funció *InvokeRepeating*, la qual crida a una funció cada cert temps la qual és l'encarregada d'instanciar els enemics.

```
void Start () {  
    InvokeRepeating("Spawn", spawnTime, spawnTime); //Es crida la funció Spawn, que començara passat cert temps, i es seguirà cridant un cop  
                                                    //passa aquest mateix temps. Per exemple, ("Spawn", 3 segons, 3 segons).  
}  
  
void Spawn () {  
    if(playerHealth.currentHealth <= 0f) //Si el jugador ha mort...  
    {  
        return; //...llavors es surt de la funció  
    }  
  
    int randomPoint = Random.Range(0, spawnPoints.Length); //Es genera un punt aleatòriament  
  
    Instantiate(enemy, spawnPoints[randomPoint].position, spawnPoints[randomPoint].rotation); //S'instancia l'enemic en la posició i rotació  
                                                    //del punt aleatori anterior  
}
```

Figura 39: Generar enemics

Un cop la interacció entre jugador i enemic estava completa, era hora de crear la figura de l'enemic final a Maya, així com l'escenari amb tots els objectes inorgànics necessaris.

Després, un cop ja estaven tots els elements 3D importats a Unity, vaig crear el sistema de puntuació i el game over. Cada cop que el jugador mata a un enemic, el marcador s'incrementa en 1, i si el jugador mor, llavors es reproduïx l'animació de la mort del personatge que havia fet a Maya. Una vegada fet això, el videojoc ja era totalment funcional en termes de jugabilitat i interacció, i només faltava afegir els efectes visuals com ara els sistemes de partícules i fer que el jugador adquirís un to vermellós al ser atacat.

Per aconseguir això últim, vaig crear uns *arrays* amb els materials del jugador i la pistola, de manera que cada cop que aquest fos ferit, canviessin tots a un to vermellós durant un temps determinat, i que un cop passat aquest temps, tornessin a les seves tonalitats originals.

```
IEnumerator HurtChangeColor()
{
    for (int i = 0; i < playerMesh.materials.Length; i++)
    {
        playerMesh.materials[i].color = new Color(           //Quan el jugador es tocat per un enemic, es canvien els colors dels materials de
                                                             //la malla d'aquest a un to vermellós
            playerMesh.materials[i].color.r,
            playerMesh.materials[i].color.g / 2,
            playerMesh.materials[i].color.b / 2
        );
    }
    for (int i = 0; i < gunMesh.materials.Length; i++)
    {
        gunMesh.materials[i].color = new Color(             //El mateix passa amb els materials de la pistola
            gunMesh.materials[i].color.r,
            gunMesh.materials[i].color.g / 2,
            gunMesh.materials[i].color.b / 2);
    }
    for (int i = 0; i < gunEndMesh.materials.Length; i++)
    {
        gunEndMesh.materials[i].color = new Color(         //El mateix passa amb els materials del canó de la pistola
            gunEndMesh.materials[i].color.r,
            gunEndMesh.materials[i].color.g / 2,
            gunEndMesh.materials[i].color.b / 2);
    }
    yield return new WaitForSeconds(0.1f);                 //Amb aquesta instrucció, el to vermellós es veurà durant una dècima de segon
    playerMesh.materials = materialsPlayer;                //Un cop passat aquest temps, els materials tornen al seu color original
    gunMesh.materials = materialsGun;
    gunEndMesh.materials = materialsGunEnd;
}
}
```

Figura 40: Funció perquè el jugador canviï de color

Pel que fa a les partícules, han estat necessaris tres sistemes diferents:

- explosió de l'enemic quan mor
- el jugador fereix a l'enemic amb el raig de la pistola
- feix de partícules que surt de la pistola (al voltant del canó) quan aquesta es dispara

Per últim, es van crear els elements gràfics dels menús així com la navegació de l'aplicació, i es va acabar la interfície gràfica d'usuari afegint unes icones com a ajuda visual per l'usuari final.



Figura 41: Indicadors de la GUI

Pel que fa a la navegació de l'aplicació, es va estructurar en dues escenes diferents. En la primera es va crear el menú principal, el qual estava format per la pantalla inicial que conté 4 botons:

- INSTRUCTIONS
- LEGAL
- CREDITS
- EXIT

Aquests botons (exceptuant EXIT, amb el qual es surt directament de l'aplicació) porten a altres menús (en aquest cas pantalles amb informació rellevant) que formen part de la mateixa escena. El mateix ocorre amb el menú de l'argument i el de selecció del nivell de dificultat.

Ara bé, quan s'escull el nivell de dificultat, llavors l'aplicació porta a l'usuari a una segona escena (les escenes són "invisibles" per l'usuari, és a dir, només tenen sentit a Unity) que és el joc. El menú de pausa i el menú de fi de la partida es troben en la mateixa escena que el joc. Aquesta estructura de la navegació és possible gràcies a dos opcions que proporciona Unity:

→ *Build Settings*. Gràcies a aquest menú els diferents nivells o escenes de l'aplicació es poden organitzar en l'ordre que es vol que apareguin. En aquest cas, la primera escena que ha d'aparèixer quan l'usuari iniciï l'aplicació és el menú principal, i la segona escena o nivell serà el joc. A més, aquest menú també permet exportar l'aplicació, escollint la plataforma a la qual es dirigeix amb les configuracions corresponents, com ara l'arquitectura del sistema operatiu, les resolucions i formats de pantalla, la icona d'escriptori de l'aplicació i la qualitat gràfica (entre d'altres). Totes aquestes opcions es poden configurar a través de la pestanya *Player Settings*.

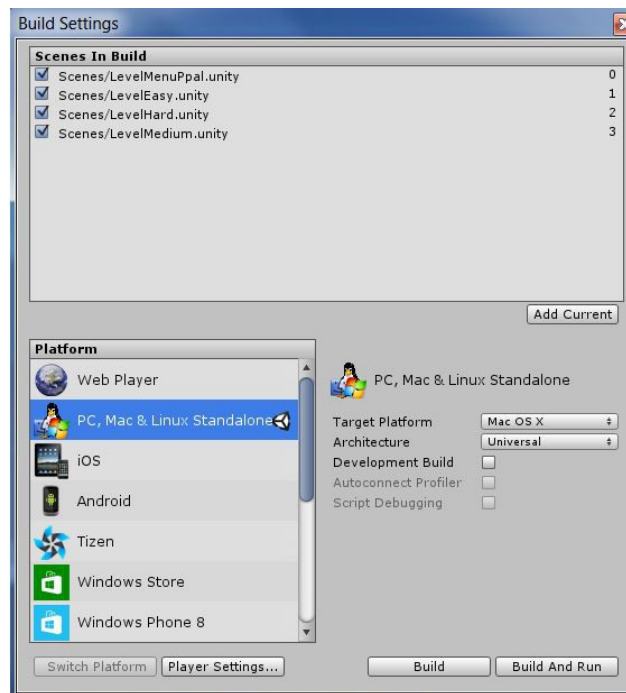


Figura 42: Build Settings

→ *LoadLevel()*. Aquesta funció permet dirigir-se a l'escena que estigui entre els parèntesis.

```
public void PlayButton()
{
    Application.LoadLevel("LevelEasy");
}

public void ExitButton()
{
    Application.Quit();
}
```

Figura 43: Carregar una escena

Ja per últim, i abans d'exportar l'aplicació, es va afegir la música i els efectes sonors.

16. Bugs

El principal error que s'ha detectat ha estat en el moviment del personatge, ja que a vegades quan passa d'una animació a una altra, degut a les transicions entre aquestes, es produeix un efecte de moviment "estrany" (tot i que tampoc es nota gaire). Això es solucionaria polint les transicions entre animacions i provant punts inicials i finals diferents.

Un altre error que s'hauria de solucionar seria evitar que quan es fa clic al botó de pausa, el jugador dispari a la vegada. No influeix en la dinàmica del joc, però no queda bé estèticament.

17. Projecció a futur

El motiu principal pel qual vaig escollir la creació d'un videojoc com a temàtica del present projecte va ser el desig de crear els meus propis personatges i poder interactuar amb aquests. Aquesta tasca ha estat portada a terme, i en un pla personal crec que els objectius principals que es van establir alhora de començar el projecte han estat assolits. Ara bé, crec també que els resultats finals han estat força condicionats pel factor temps, cosa lògica no obstant, ja que es tracta d'un treball de final de grau. Quan es comença un projecte personal, sempre hi ha moltes idees en ment, idees bones i potser no tan bones, però que mai veuran la llum degut a la limitació temporal. Doncs bé, en aquest apartat exposaré aquelles idees que crec que podrien millorar l'aplicació amb més temps i recursos.

Primer de tot començaré amb el disseny i el contingut. Per un joc d'aquestes característiques, un requisit indispensable és que hi hagi una gran varietat de personatges, ja que és un aspecte que enriqueix no només l'apartat visual, sinó també la jugabilitat. La possibilitat de tenir enemics (en aquest cas insectes) que pertanyin a diferents tipus, com podrien ser formigues, abelles o mosques (entre d'altres) faria que BigBug guanyés. Cadascun d'aquests es comportaria de manera diferent, i per tant caldrien diferents formes d'eliminar-los, fent que l'usuari es sentís més immers i participatiu en el joc. Un altre aspecte a millorar seria la possibilitat de contar amb més tipus d'armes, sent algunes més efectives que les altres alhora d'eliminar a certs tipus d'enemics.

Sens dubte, un altre punt que milloraria molt el videojoc seria la inclusió de diferents nivells o pantalles. Abans de començar el projecte, la meva idea era crear diferents nivells (ordenats de forma progressiva) de manera que l'usuari hagués de superar un per passar al següent. En cada nivell hi hauria un número d'enemics que l'usuari tindria que eliminar per superar-lo (es podria utilitzar inclús un compte endarrere per afegir més emoció a la partida). Estaria bé pensar en la idea de que certs tipus d'insectes poguessin ser bons, ajudant al jugador a acabar amb els dolents, i afegint així un toc d'estratègia a BigBug.

En cada nivell, l'escenari seria diferent. La primera pantalla podria ser la casa del protagonista (és a dir, la pantalla del joc actual) i la resta podrien ser diferents

localitzacions d'un poble o ciutat (des d'una plaça a un supermercat passant per un parc d'atraccions). Això també oferiria la possibilitat de crear noves conductes en els enemics, com podria ser la possibilitat d'escapar o d'amagar-se. D'aquesta manera, els enemics no perseguirien al protagonista durant tota la partida, i es milloraria així la seva intel·ligència de forma que l'usuari hagués de ser millor estratega. També es podria pensar en l'opció d'incloure diferents ítems en les pantalles, com ara eines per millorar les armes o recuperació de vida. A més també podrien haver-hi missions secundaries, com per exemple salvar a altres personatges de situacions difícils i així rebre bonificacions, amb les quals poder personalitzar la indumentària i aspecte del personatge principal i l'estètica de les armes. La inclusió de dolents finals al acabar una ronda de nivells també és un factor que es podria tenir en compte.

Els menús de navegació també haurien de ser lleugerament millorats. Es podria afegir l'opció de carregar partides per tal que l'usuari no perdés el progrés aconseguit i opcions de personalització (això sempre és ben rebut per l'usuari). A més els botons haurien de ser redissenyats (en els tests d'usuari hi ha hagut comentaris al respecte) per tal que l'experiència d'usuari fos millor, afegint àudio i un disseny més contrastat. Un aspecte que també m'agradaria afegir en un futur seria una animació sobre l'argument del videojoc en comptes del text que hi ha actualment, explicant la història que va portar al desastre en el qual està ubicat el món de BigBug.

D'altra banda, pel que fa a la jugabilitat, una cosa que m'agradaria fer en un futur seria adaptar el joc perquè funcionés a dispositius mòbils com ara *smartphones* o *tablets*, ja que actualment són molts els usuaris que per jugar i per entretenir-se utilitzen únicament dispositius d'aquest tipus (de camí a la feina, esperant al dentista, etc.). La mobilitat forma part de la societat actual, i és una oportunitat que s'ha d'aprofitar.

18. Pressupost

Donat que aquest projecte és un treball final de grau, i com a tal, totes les tasques han estat dutes a terme per mi sol (excepte l'àudio, del qual s'ha encarregat un amic el qual no em cobra res) el cost és 0 €. No obstant, a continuació es planteja un pressupost derivat d'una situació hipotètica on el videojoc seria produït per un equip multidisciplinar, tenint en compte el preu en euros per hora en cada tasca i el número total d'hores.

PERFIL	QUANTITAT	SOU PER HORA (€)
Il·lustrador	1	35
Modelador de personatges 3D	1	32
Modelador d'escenaris 3D	1	30
<i>Rigger</i>	1	22
Animador 3D	1	35
Dissenyador gràfic	1	20
Programador	1	20
Especialista en so	1	30
Especialista en màrqueting	1	15

Taula 4: Pressupost hipotètic del projecte. Perfils

ACTIVITAT	HORES	PERFIL	COST (€)
<i>Concept art</i>	35	Il·lustrador	700
Modelatge 3D de personatges	80	Modelador de personatges 3D	2560
<i>Rigging</i>	30	<i>Rigger</i>	660
Animació 3D	40	Animador 3D	1400
Modelatge 3D d'escenaris	65	Modelador d'escenaris 3D	1950
Programació	100	Programador	2000
Disseny d'interfície gràfica	20	Dissenyador gràfic	400
Producció de l'àudio	25	Especialista en so	750
Estudi de mercat i posicionament	15	Especialista en màrqueting	225
		TOTAL	10.645 €

Taula 5: Pressupost hipotètic del projecte. Número d'hores i cost total

19. Conclusions

La realització del Treball Final de Grau ha estat una gran barreja de sensacions i emocions, algunes molt gratificants, altres no tant, però en general han estat noves experiències per seguir creixent professionalment, artísticament i personalment. Des d'un pla personal i per la meua pròpia experiència, diria que he viscut 3 etapes clau en aquest procés (aquestes etapes no tenen res a veure amb l'ordre d'entrega de PAC's o les versions de l'aplicació).

La primera va ser la de recerca de la idea i argument del videojoc, la qual va ser molt motivadora ja que, per primer cop en el grau, tenia gairebé total llibertat alhora d'escollir el meu projecte. Va ser una fase en la que multitud d'idees brotaven del meu cap, així com possibles dissenys de personatges i escenaris, colors, músiques, etc. Després de passar bona part de l'estiu pensant sobre el tema, finalment vaig decidir crear BigBug.

Però després arriba la segona etapa, que és la més dura emocionalment i psicològicament. És quan comences a baixar dels núvols i toques amb els peus a terra, quan te n'adones que estàs tu sol, que tens bones idees però poc temps i recursos, i llavors la frustració invadeix per moments les motivacions i ganes inicials. En aquesta etapa vaig haver de donar prioritat a certs aspectes del projecte, així com abandonar idees que estaven a l'inici i adaptar-ho tot a un calendari molt ajustat.

La tercera etapa és l'etapa en la que assimiles que el teu joc no serà exactament igual a com ho havies plantejat al principi de tot. Però llavors intentes ser coherent, i amb el temps i els recursos disponibles utilitzes la creativitat i l'enginyer per intentar que el resultat final sigui igualment bo. En aquesta etapa la motivació em va tornar a invadir, ja que comences a veure que el projecte va prenent forma. Els teus personatges es mouen, comences a veure colors, a jugar amb el teu propi videojoc, i llavors les ganes inicials tornen per enllestir finalment el projecte.

Un cop he arribat a aquest punt he de dir que estic orgullós d'haver escollit aquest projecte, ja que tot i que no he pogut fer tot el que volia en un principi, he après moltíssimes coses noves i he aprofundit en d'altres. Quan vaig començar el projecte, els meus coneixements en programació eren força bàsics, i tot i que ara mateix

tampoc són un mestre en la matèria, si que puc dir orgullós que acabo el grau tenint uns bons fonaments de programació. El mateix ha passat amb el *rigging* (és un dels aspectes que més m'ha costat, especialment l'esquelet i els controladors del protagonista), ja que si bé es toca en assignatures del grau, mai havia tingut que realitzar un complet. També he adquirit més destresa en modelatge i animació 3D, així com en disseny gràfic.

En conclusió, acabo aquest projecte i el grau amb bon sabor de boca i content pel tot el que he après, que al final és el més important que queda.

Annex 1. Lliurables del projecte

Videojoc BigBug

- *BigBugWin.exe*
Arxiu executable de l'aplicació per la plataforma Windows.
- *BigBugMac.app*
Arxiu de l'aplicació per la plataforma Mac.
- *Projecte de Unity*
Arxiu amb tots els components del projecte per Unity. Aquí s'inclou els scripts, l'àudio, imatges gràfiques, personatges i objectes en format .fbx, tipografies, materials i escenes.
- *Arxius .ma de Maya*
Arxius que contenen els models en 3D, els *riggings* i les animacions.
- *Arxius .psd i .ai*
Arxius de Photoshop i Illustrator dels menús i GUI.

Presentació del projecte

- *Vídeo de presentació*
Vídeo de presentació del projecte pel tribunal.
- *Presentació en format lliure del projecte*
Presentació que es dirigeix al públic en general, no especialitzat.

Informació del projecte

- *Memòria*
Document en format .pdf que inclou la memòria del projecte.
- *Autoinforme d'avaluació*
Document en format .pdf que incou un autoinforme d'avaluació del projecte.

Annex 2. Codi font (extractes)

A continuació es mostren els extractes més importants dels scripts de programació en C# que han esdevingut les peces clau pel desenvolupament de l'aplicació. En les imatges apareixen comentaris a la part dreta per ajudar a entendre el funcionament.

Les tres imatges següents corresponen al moviment del personatge principal. Aquest està dividit en tres funcions que posteriorment es criden al mètode FixedUpdate de Unity.

La funció de moviment:

```
void Move(float horz, float vert)
{
    movement.Set(horz, 0f, vert); //Els valors de moviment del jugador seran les tecles A/D per X, res per
    //Y (no volem que es mogui en vertical), i les tecles W/S per l'eix Z
    movement = movement.normalized * moveSpeed * Time.deltaTime; //Amb normalized aconseguim que el vector en diagonal de moviment es
    //realitzi a la mateixa velocitat que els horitzontal i vertical
    playerRBody.MovePosition(transform.position + movement); //Utilitzem el MovePosition del rigidbody perquè així el jugador no pugui
    //atravessar objectes. Aleshores, el moviment serà igual a la posició actual
    //del jugador més el moviment calculat anteriorment.
}
```

La funció de rotació:

```
void Turn()
{
    Ray cameraRay = Camera.main.ScreenPointToRay(Input.mousePosition); //Es crea un raig des de la càmera en funció del punt on tinguem
    //el mouse
    RaycastHit floorHit;
    if(Physics.Raycast(cameraRay, out floorHit, cameraRayDistance, floorMask))
    {
        Vector3 playerToMouse = floorHit.point - transform.position; //Obtenim el vector des de la posició del personatge a la posició
        //que hem tocat amb el mouse
        playerToMouse.y = 0f; //No volem que el personatge miri cap al terra
        Quaternion playerRotation = Quaternion.LookRotation(playerToMouse); //S'utilitza Quaternion perquè Unity no treballa amb vectors pel
        //que fa a mesures de rotació
        playerRBody.MoveRotation(playerRotation); //establim la rotació al rigidbody del personatge
    }
}
```

La funció que reproduïx les animacions:

```
void Animating(float horz, float vert)
{
    bool idle = ((horz == 0) && (vert == 0));
    animCtrl.SetBool("IsMoving", !idle); //Si el jugador no està quiet, llavors s'aplica l'animació de moviment
    //gràcies al BlendTree, es mourà de forma diferent depenent si és cap
    //endavant, endarrere o lateralment
}
```

La funció que crida a les 3 anteriors i que es va actualitzant a cada fotograma:

```
//Utilitzem FixedUpdate en comptes de Update perquè és més adient alhora de treballar amb el motor de físiques
void FixedUpdate()
{
    float horz = Input.GetAxisRaw("Horizontal"); //Captem els valors de les teclcs esquerra/dreta o bé A/D
    float vert = Input.GetAxisRaw("Vertical"); //Captem els valors de les teclcs adalt/abaix o bé W/S

    float moveDirAngle = Mathf.Rad2Deg * (Mathf.Atan2(horz, vert)); //S'ha de saber la direcció del jugador per aplicar l'animació concreta
    float shootDirAngle = transform.rotation.eulerAngles.y; //S'ha de saber també la rotació del jugador
    float playerDirection = moveDirAngle - shootDirAngle; //La resta d'ambdues dóna la posició/orientació del jugador

    float sin = -Mathf.Sin(Mathf.Deg2Rad * playerDirection); //S'estableix el sinus i el cosinus per saber on es dirigeix el jugador
    float cos = Mathf.Cos(Mathf.Deg2Rad * playerDirection);

    animCtrl.SetFloat("H", sin); //Mitjançant H i V podem establir a través del BlendTree del Animator
    //l'animació que es necessita en funció de la posició i rotació del jugador
    animCtrl.SetFloat("V", cos); //si es mou cap endavant, endarrere o lateralment

    Move(horz, vert);
    Turn();
    Animating(horz, vert);
}
```

La següent imatge correspon al script del moviment de la càmera. Aquesta ha de seguir al jugador en la seva translació horitzontal i vertical, però no en la seva rotació.

```
void Start()
{
    offset = transform.position - target.position; //L'offset o diferència de distància serà la posició de la càmera menys la posició
    //del objectiu a seguir (en aquest cas el jugador).
}

void FixedUpdate()
{
    Vector3 targetPos = target.position + offset; //A cada frame s'anirà actualitzant la
    //posició del target (jugador)
    transform.position = Vector3.Lerp(transform.position, targetPos, cameraSmooth * Time.deltaTime); //S'utilitza Lerp per poder interpolat la
    //posició de la càmera
}
```

Tot seguit es pot veure el codi que permet que els enemics es moguin cap a la posició del jugador.

```
Transform player; //Es necessita saber el transform del jugador perquè l'enemic el pugui seguir
//La variable no es fa pública perquè l'enemic és un Prefab, i no és recomanable que un Prefab tingui
//referències amb objectes de la jerarquia (en aquest cas el jugador)

PlayerHealth playerHealth;
EnemyHealth enemyHealth;
NavMeshAgent navAI; //Es crea un objecte NavMeshAgent per tal d'establir una Intel·ligència Artificial pels enemics

void Awake()
{
    player = GameObject.FindGameObjectWithTag("Player").transform; //S'obté la referència del transform del jugador. Com s'ha comentat més
    //amunt, al no obtenir la referència d'aquest per mitjà d'una
    //variable pública, s'ha d'obtenir mitjançant el mètode FindGameObject
    //WithTag(...)

    playerHealth = player.GetComponent<PlayerHealth>();
    enemyHealth = GetComponent<EnemyHealth>();
    navAI = GetComponent<NavMeshAgent>(); //S'obté la referència del NavMeshAgent
}

void Update()
{
    if (enemyHealth.currentHealth > 0 && playerHealth.currentHealth > 0) //L'enemic només es mourà quan tan l'enemic com el jugador estiguin
    //vius
    {
        navAI.SetDestination(player.position); //En cada actualització del Update se li diu al NavMeshAgent quina és
    //la posició del jugador, i a través del SetDestination anirà cap
    //aquesta. Per tant, l'enemic es dirigirà en tot moment a la posició
    //del jugador
    }
}
```

A continuació es mostra el codi més rellevant pel que fa a la salut del personatge principal i la dels enemics, i com es fa possible que es puguin treure vida.

En primer lloc s'exposen una imatge de les funcions més importants de la salut del jugador.

```

public void Damage (int qty) //Aquesta funció és la que s'encarrega de treure vida. És pública perquè s'ha de poder
                             //accedir des d'altres components i scripts d'altres objectes (bàsicament l'enemic)
{
    damaged = true; //El jugador és ferit, i es canvia la variable damaged a true
    currentHealth -= qty; //La quantitat de dany es resta a la salut actual
    healthUI.value = currentHealth; //La salut actual s'actualitza al Slider de vida del jugador
    playerHurtSound.Play(); //Es reproduueix el so del jugador ferit

    if(currentHealth <= 0 && !isDead) //Cada cop que es decrementa la salut del jugador, s'ha de comprovar si la salut
                                     //d'aquest és menor o igual que 0 i si no estava mort. Si ambdues opcions es compleixen
                                     //llavors es crida a la funció Death
    {
        Death();
        StartCoroutine(gameOverMenuEnabled()); //Es crida a la corutina perquè aparegui el menu Game Over un cop mort el jugador
    }
}

void Death()
{
    isDead = true; //El jugador passa a estar mort

    playerShoot.NoEffects();

    animCtrl.SetTrigger("Die"); //Es reproduueix l'animació de mort del jugador

    playerHurtSound.clip = deathSound; //Quan el jugador està mort, s'assigna el clip d'àudio de mort del jugador a l'audio source
    playerHurtSound.Play(); //Llavors es reproduueix

    playerMovement.enabled = false; //Mitjançant la referència a PlayerMovement, desactivem aquest perquè el jugador no
    //es pugui moure un cop ha mort

    playerShoot.enabled = false;
}

```

La funció que permet al jugador disparar i treure vida a l'enemic.

```

void Shoot()
{
    timer = 0f; //Es posa el timer a 0

    shootSound.Play(); //Es reproduueix el so del tret

    shootLight.enabled = true; //S'activa la llum

    shootParticles.Stop();
    shootParticles.Play();

    shootLine.enabled = true; //S'activa el render del tret
    shootLine.SetPosition(0, transform.position); //Es posiciona l'extrem inicial (element 0) del tret a la posició del canó
    //de la pistola (transform.position)

    shootRay.origin = transform.position; //Punt inicial del tret
    shootRay.direction = transform.forward; //Direcció on anirà el tret (sempre cap endavant)

    if(Physics.Raycast(shootRay, out shootHit, shootDistance, shootableMask)) //Comprovem si el tret colisiona amb algun objecte de la capa
    //Shootable. El shootHit guarda la info de l'objecte
    //amb el que s'ha colisionat
    {
        EnemyHealth enemyHealth = shootHit.collider.GetComponent<EnemyHealth>(); //Si es colisiona, s'accedeix al colider de l'objecte
    //amb el que s'ha colisionat, i així podem accedir al
    //script EnemyHealth

        if(enemyHealth != null)
        {
            enemyHealth.Damage(shootDamage, shootHit.point); //Si enemyHealth no és null vol dir que s'ha colisionat
    //amb un enemic, i que per tant, se li treurà vida,
    //obtenint també el punt de colisió

        }
        shootLine.SetPosition(1, shootHit.point); //En el cas d'haver-hi colisió amb un enemic o objecte de la capa shootable (si no és enemic no se li pot
    //treure vida) es situa l'extrem final (element1) del tret en aquesta posició
    }
    else
    {
        shootLine.SetPosition(1, shootRay.origin + shootRay.direction * shootDistance); //Sino es colisiona amb cap objecte de la capa Shootable llavors es dibuixa el render del tret posicionant l'
    //extrem final (element1) en una posició que va des de la posició inicial seguint la mateixa direcció per la
    //longitud del tret (en aquest cas 100 unitats)
    }
}

```

En segon lloc es mostra el codi més rellevant pel que fa a la salut dels enemics.

```

public void Damage(int qty, Vector3 hitPoint)
{
    if (isDead) //Si l'enemic ja està mort, llavors es surt de la funció Damage perquè ja no se li pot atacar més
        return;

    enemyHurtSound.Play(); //Es reproduïx l'audio quan es fereix l'enemic

    currentHealth -= qty; //Es resta la quantitat de dany a la salut actual

    hurtEnemyParticles.transform.position = hitPoint; //Les partícules apareixen al lloc de l'enemic on es dispara
    hurtEnemyParticles.Play();

    if (currentHealth <= 0) //Si la salut actual és menor que 0 (l'enemic ha mort)...
    {
        Death(); //...llavors es crida a la funció Death
    }
}

void Death()
{
    isDead = true; //Quan l'enemic mor, isDead passa a ser cert perquè així l'enemic no pugui morir de nou

    capsuleCollider.isTrigger = true; //També desactivem el seu collider, fent que el component isTrigger passi a cert, de manera
    //que el collider no podrà xocar físicament amb altres colliders de l'escena

    GetComponent<NavMeshAgent>().enabled = false; //Accedim al NavMeshAgent de l'enemic i el deshabilitem per tal que no pugui moure's més
    GetComponent<Rigidbody>().isKinematic = true; //Accedim al Rigidbody de l'enemic i el marquem com a Is Kinematic, de manera que deixarà
    //de veure's influenciat per les físiques

    NumKilledEnemies.numEnemies += scoreEnemyKilled; //Cada cop que un enemic mort, el marcador s'incrementa en 1
}

```

I per últim les funcions que permeten a l'enemic atacar al personatge i treure-li vida. Primer el programa ha de saber si l'enemic entra en contacte amb el jugador o pel contrari es tracta d'un altre objecte de l'escenari.

```

void OnTriggerEnter (Collider coll) //Cada cop que un objecte amb collider entri dins del collider de l'enemic (el collider que té
//isTrigger activat) es cridarà aquest mètode
{
    if(coll.gameObject == player) //Si el collider del gameobject que entra al collider de l'enemic és el jugador...
    {
        isInRange = true; //...llavors el jugador està al costat de l'enemic (està dins el seu alçanc)
    }
}

void OnTriggerExit (Collider coll) //Cada cop que un objecte amb collider surti del collider de l'enemic (el collider que té
//isTrigger activat) es cridarà aquest mètode
{
    if (coll.gameObject == player) //Si el collider del gameobject que surt del collider de l'enemic és el jugador...
    {
        isInRange = false; //...llavors el jugador està fora de l'alçanc de l'enemic, aquest ja no li pot fer mal
    }
}

```

A continuació es crea la funció d'atac.

```
void Attack()
{
    timer = 0f; //El contador es posa a 0 cada cop que l'enemic ataca al jugador

    if(playerHealth.currentHealth > 0) //Aquí es comprova si el jugador té vida. En el cas que així sigui...
    {
        playerHealth.Damage(damageQtty); //...es crida al mètode Damage del script PlayerHealth
    }
}
```

I finalment, es crida la funció anterior sempre que el jugador estigui viu, estigui en contacte amb l'enemic i hagi passat el temps mínim entre atac i atac.

```
void Update()
{
    timer += Time.deltaTime; //Es va contant el temps que passa des de l'últim cop que l'enemic ha atacat al jugador

    if(timer >= attackTime && isInRange && enemyHealth.currentHealth > 0) //Si el temps que ha passat és major o igual que el temps que s'ha
    //d'esperar entre atac i atac i el jugador es troba a l'alcanc de //l'enemic, i la salut de l'enemic és major que 0 (està viu)...
    {
        Attack(); //...llavors s'ataca
    }
}
```

Annex 3. Llibre d'estil

L'estil de l'aplicació es compon dels següents elements:

→ **Isotip de l'aplicació.** És una flor vermella amb una calavera al mig. La calavera representa que les flors no són naturals, que tenen un component tòxic i malèvol (en referència a l'argument del videojoc).



→ **Logotip de l'aplicació.** Títol en verd amb la paraula BigBug.



→ **Paleta de colors.** S'han escollit dos colors que són el verd i el vermell (complementaris). Pels diferents títols, botons i textos s'han fet servir diferents tonalitats d'aquests.

Pels fons de pantalla dels menús de pausa i game over s'ha fet servir el següent color:



RGB [Hex] #88cc88
RGB [0-255] 136 204 136

Pels botons, l'isotip i el text s'ha fet servir el següent color:



RGB [Hex] #aa3939
RGB [0-255] 170 57 57

Pels títols i el text dels botons s'ha fet servir el color següent:



RGB [Hex] #550000
RGB [0-255] 85 0 0

Pel logotip de BigBug s'ha fet servir el color següent:



RGB [Hex] #2d882d
RGB [0-255] 45 136 45

→ **Tipografies**. Les fonts tipogràfiques utilitzades en aquest projecte han estat la Grinched (només utilitzada pel logotip de l'aplicació) i la Wicked Scary Movie (utilitzada per la resta d'elements).

GRINCHED

Wicked Scary Movie

Annex 4. Glossari

survival shooter ... Tipus de videojoc on el jugador ha de sobreviure matant a enemics mitjançant l'ús d'una arma que dispara.

character

- design* Disseny dels personatges d'un videojoc o un curt d'animació
- rigging* Procés pel qual es dota a un model 3D d'un esquelet per les futures animacions
- game design* Disseny de la part creativa del videojoc (personatges, escenaris, etc.)
- cartoon* Tipus d'il·lustració que està basada en l'anatomia humana però amb grans dosis de surrealisme i ficció
- concept art* Il·lustracions que representen visualment un disseny, idea o personatge pel seu ús en pel·lícules, videojocs, còmics, etc
- gamers* Tota aquella persona que juga a videojocs
- poligonatge* Número de polígons que té un objecte 3D
- quad* Polígon de 4 arestes en els programes de 3D
- low poly* Model en 3D que té un baix número de polígons
- humanIK* Sistema de construcció d'esquelet automàtic de Maya
- joint* Articulació o ós que forma part d'un esquelet a Maya
- skinning* Tècnica per la qual la malla del model 3D s'aplica a l'esquelet
- smooth bind* Tècnica de *skinning* pròpia de Autodesk Maya
- fps* Fotogrames per segon (en anglès *frames per second*)
- aliàsing* Efecte visual en el qual les vores dels polígons no estan suavitzades
- Touchpad* Panell tàctil per ordinadors de la marca Apple
- Blend Tree* Eina de Unity per gestionar animacions
- Nav Mesh Agent* ... Component de Unity per establir que un element es mogui per una superfície determinada
- GUI* Interfície Gràfica d'Usuari (de l'anglès *Graphical User Interface*)
- array* Element utilitzat en programació. Associa valors amb claus

Annex 5. Bibliografia

Autodesk. *Maya User's Guide*.

<[http://download.autodesk.com/global/docs/maya2014/en_us/index.html?
=contextId=BULLETNODES](http://download.autodesk.com/global/docs/maya2014/en_us/index.html?contextId=BULLETNODES)>

Microsoft. *C# Programming Guide*. <<https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>>

Vaughan, W. (2012). *Modelado Digital*. Anaya Multimedia.

Unity. Manual.

<<http://docs.unity3d.com/Manual/index.html>>

Nielsen Norman Group. 10 Usability Heuristics for User Interface Design.

<<https://www.nngroup.com/articles/ten-usability-heuristics/>>