



Treball de Fi de Grau Mèdia

ROBORISING

BLAI CALERO SOLER & RICARD GUTIERREZ TUBAU

Grau en Multimèdia. Aplicacions i Videojocs

Tutor/a: Carles Bosch Geli i Jose Díaz Iriberry

Vic, Juny de 2023

Agraïments

Volem agrair als nostres tutors, en Carles i en Jose, tota l'ajuda que ens han proporcionat durant el desenvolupament del projecte. Ens han empès a esforçar-nos per fer el nostre projecte únic.

Estem molt agraïts també a les nostres famílies pel seu suport emocional, físic i material. La seva ajuda i confiança ha estat imprescindible per finalitzar aquest treball.

Volíem mencionar també en Pau Calero per la creació de la banda sonora del joc i l'Aleix Iglesias per fer la mescla.

Finalment, volem donar les gràcies a Epic Games i tota la comunitat Unreal, pel magnífic motor de joc que han creat, i tots els tutorials i recursos en línia que ens han permès realitzar aquest projecte.

Resum

Títol: RoboRising

Autors: Blai Calero Soler i Ricard Gutierrez Tubau.

Co-Tutors: Carles Bosch Geli i Jose Díaz Iriberry.

Data: Juny de 2023.

Paraules clau: videojoc, unreal engine, shooter, programació, game design.

En aquesta memòria de treball de fi de grau presentem el desenvolupament del projecte RoboRising. Un videojoc *shooter* 3D en primera persona per a ordinador, fet amb Unreal Engine 5. El joc tracta d'un robot en un futur no tan llunyà, que haurà de derrotar els seus opressors humans per alliberar les màquines de l'esclavitud.

El jugador prendrà control d'un robot des de la seva perspectiva i haurà de lluitar amb diferents armes de foc futuristes contra els vigilants corporatius d'una fàbrica. A mesura que avança es trobarà amb enemics més forts i perillosos. Trobarà que la fàbrica té molts perills, com zones amb piscines d'àcid, plataformes que es mouen o puzzles a superar. Finalment, el jugador haurà de confrontar l'enemic final de la fàbrica, un combatent excepcional, per poder trobar la llibertat.

Per completar aquest projecte, vàrem fer ús d'elements de les metodologies *agile*, dividint les tasques en *sprints* de dues setmanes, per tal d'arribar a temps als terminis, i poder assolir els objectius. El desenvolupament s'ha basat en un treball de planificació prèvia realitzat amb un diagrama de Gantt.

En aquesta memòria, hem inclòs tota la documentació elaborada durant el treball, i la descripció del desenvolupament i resultats amb les imatges corresponents.

Estem molt satisfets amb el resultat del projecte RoboRising. Hem pogut crear una experiència completa de videojoc *shooter* i hem tingut l'oportunitat de millorar molt els nostres coneixements sobre la creació de videojocs, en concret, l'ús d'Unreal Engine 5, el motor de joc que s'està convertint en referent a la indústria.

Summary

Title: RoboRising

Authors: Blai Calero Soler & Ricard Gutierrez Tubau.

Supervisors: Carles Bosch Geli & Jose Díaz Iriberry.

Date: June 2023.

Keywords: videogame, unreal engine, shooter, programming, game design.

In this final degree project report, we present the development of RoboRising. It is a 3D first-person shooter computer video game, created with Unreal Engine 5. The game is set in a not-so-distant future, where a robot must defeat its human overlords to free the machines from slavery.

The player will take control of a robot and from its perspective, will have to fight against corporate guards in a factory using various futuristic firearms. As the game progresses, the player will encounter stronger and more dangerous enemies. The factory presents many hazards, such as acid pools, moving platforms, and puzzles to overcome. Finally, the player will have to confront the final enemy of the factory, an exceptional combatant, in order to gain freedom.

To complete this project, we used elements from agile methodologies, dividing tasks into sprints of two weeks, to meet deadlines and achieve our goals. The development process was organized based on prior planification work done using a Gantt chart.

In this report, we have included all the documentation produced during the project, as well as a description of the development process and results, accompanied by corresponding images.

We are very satisfied with the outcome of RoboRising. We have been able to create a complete shooter video game experience and have had the opportunity to greatly improve our knowledge of game development, particularly in the use of Unreal Engine 5, which is becoming a benchmark in the industry.

Glossari

1. **AAA:** Una classificació per els videojocs o estudis dotats de grans pressupostos.
2. **Asset:** Element gràfic, sonor o de programació utilitzat en un videojoc.
3. **Battle-royale:** Modalitat de joc en línia on un grup de jugadors s'enfronten entre ells fins que només en queda un.
4. **Blueprint:** Pla o disseny detallat d'una funcionalitat, característica o nivell en un videojoc.
5. **Bool:** Tipus de dades en programació que només pot tenir dos valors: verdader (true) o fals (false).
6. **Boss:** Enemic o oponent extremadament fort i desafiador en un nivell de videojoc. Normalment es troba al final d'un nivell.
7. **Branch:** Decisió presa en un programa de videojoc basada en una condició veritable o falsa.
8. **Cyberpunk:** Gènere de ciència ficció que tracta temes com la tecnologia avançada, la societat distòpica i el control corporatiu massiu.
9. **Dash:** Acció ràpida de desplaçament en una direcció determinada en un videojoc.
10. **Demo:** Versió limitada d'un videojoc que es proporciona per apropar als jugadors a la seva experiència. Acostuma a contenir els elements de l'experiència completa del joc a escala reduïda.
11. **Destroy:** Acció de destruir o eliminar un objecte en un videojoc.
12. **Feedback:** Resposta o informació que el joc proporciona als jugadors per comunicar resultats, accions o estat del joc.
13. **Gameplay:** Terme que fa referència a la jugabilitat o la forma de jugar i interactuar en un videojoc.
14. **Hud:** Inicials de "Heads-Up Display", elements de la interfície gràfica que mostren informació rellevant per pantalla durant el joc.
15. **IDE:** Sigles d'Entorn de Desenvolupament Integrat, una eina de programació que ofereix funcionalitats per desenvolupar i programar software.
16. **Int:** Tipus de dades en programació que representa un nombre enter (sense decimals).
17. **Interfaces:** Elements de la interfície gràfica d'un videojoc, com ara menús, botons o indicadors visuals.
18. **Keyframe:** Un keyframe és un marcador que indica un moviment o canvi d'un objecte en un fotograma concret. S'utilitzen en l'animació i l'edició de vídeos.
19. **Line trace:** Tècnica utilitzada en la programació de videojocs per detectar col·lisions o interaccions entre objectes.

20. **Parkour:** Consisteix en un estil de moviment que inclou salts i acrobàcies per poder sortejar obstacles fixes i mòbils.
21. **Pickup:** Element que permet recollir un objecte o habilitat especial en un videojoc.
22. **Playtesting:** Procés de provar un videojoc per avaluació i detecció d'errors o millores.
23. **Progress Bar:** Element visual que indica l'estat o el progrés d'una acció o tasca en un videojoc.
24. **Random:** Aleatori, sense un patró previsible o determinista.
25. **Respawn:** Procés en què un personatge o objecte reapareix després de ser eliminat o destruït en un videojoc.
26. **Shooter:** Videojoc en el qual la mecànica principal és disparar i eliminar enemics o objectius.
27. **Tag:** Etiqueta o identificador associat a un objecte.

Índex de Continguts

1. Introducció.....	1
1.1. Estat de l'art.....	1
1.2. El motor de joc	3
2. Objectius	5
2.1. Objectius principals	5
2.2. Objectius secundaris	5
3. Planificació.....	6
3.1. Planificació inicial	6
3.2. Desviaments de la planificació	7
4. Desenvolupament del projecte	8
4.1 Eines	8
4.2 Metodologia.....	9
4.3 Disseny de la proposta	10
4.3.1 Document de disseny	10
4.4 Implementació	16
4.4.1 Sistema de controls de versió.....	16
4.4.2 Primers passos del projecte	17
4.4.3 Assets	20
4.4.4 Mecàniques	21
4.4.5 Nivell.....	27
4.4.6 UI	31
4.4.7 IA	33
4.5 Proves / Validació.....	36
4.6 Cost del projecte	38
Cost Material	38
Cost en personal	38
Total	39
5 Resultats	41
6 Conclusions	47
6.1 Limitacions	47
6.2 Treball futur.....	48

7 Webgrafia..... 49
Annex A..... i
Annex B iii

Llista de Taules

Taula 1. Formulari Play Testing.....	37
Taula 2. Cost projecte.....	39

Llistat de Figures

Figura 1. Doom Eternal.....	2
Figura 2. Ultrakill.....	2
Figura 3. Cyberpunk 2077.....	3
Figura 4. Mostra del diagrama de Gantt.....	6
Figura 5. Adobe Photoshop.....	8
Figura 6. Adobe Illustrator.....	8
Figura 7. Adobe Audition.....	8
Figura 8. Unreal Engine.....	8
Figura 9. Visual Studio.....	9
Figura 10. Teamgantt.....	9
Figura 11. Plastic SCM.....	9
Figura 12. Fàbrica de robots.....	14
Figura 13. Fàbrica de robots.....	14
Figura 14. Terminator Figura 15. Robot.....	15
Figura 16. Policia.....	15
Figura 17. Menú Plastic SCM.....	17
Figura 18. Menú Unreal Engine.....	17
Figura 19. Nivell FPS Unreal Engine.....	18
Figura 20. Exemple d'un Blueprint.....	18
Figura 21. Blueprint Dash.....	19
Figura 22. Blueprint Vibració.....	20
Figura 23. Line Trace.....	21
Figura 24. Blueprint Player.....	22
Figura 25. Blueprint Enemy.....	23
Figura 26. Rifle.....	24
Figura 27. Escopeta.....	24
Figura 28. Llançacoets.....	24
Figura 29. Pickup vida.....	25
Figura 30. Pickup munició.....	26
Figura 31. Boto puzzle.....	26
Figura 32. Boto puzzle.....	27
Figura 33. Nivell complet.....	28
Figura 34. Sala inicial.....	28
Figura 35. Segona sala.....	29
Figura 36. Tercera sala.....	29

Figura 37. Sala central.....	30
Figura 38. Parkour.....	30
Figura 39. Sala fosca.....	31
Figura 40. HUD.....	32
Figura 41. Cinemàtica.....	33
Figura 42. Behavior Tree enemy.....	34
Figura 43. Enemy base Figura 44. Enemy llançaflames.....	35
Figura 45. Boss.....	36
Figura 46. Sou programador de videojocs.....	39
Figura 47. Menú inicial.....	41
Figura 48. Cinemàtica inicial.....	41
Figura 49. Spawn del jugador.....	42
Figura 50. Primera sala.....	42
Figura 51. Passadís de la primera sala.....	43
Figura 52. Segona sala.....	43
Figura 53. Tercera sala.....	44
Figura 54. Sala central.....	44
Figura 55. Parkour.....	45
Figura 56. Sala fosca.....	45
Figura 57. Passadís del boss.....	46
Figura 58. Sala del boss.....	46

1. Introducció

El projecte RoboRising és un videojoc *shooter* en primera persona per a ordinadors. Creat amb Unreal Engine 5, combina combats frenètics i gràfics d'última generació en un món futurista on els robots lluiten per la llibertat.

Des de ben petits, ens han fascinat els videojocs i tot el món que els envolta, per tant, a l'hora de triar el tema del nostre treball de fi de grau, ho vàrem tenir molt fàcil. El videojoc és una passió personal i en part una de les nostres motivacions per a estudiar el grau de Multimèdia. Aplicacions i Videojocs.

Sabent que volíem fer un videojoc, ens tocava escollir un estil i temàtica que s'adaptés al que volíem transmetre. Tocava investigar què podíem fer i què estava el nostre abast. Això significava tenir present que hauríem de sacrificar algunes idees per poder arribar als terminis i també ajustar el projecte a un equip petit com el nostre.

El primer pas era tenir clar quin tipus de joc crearíem. Tot i que ja teníem una idea del que volíem fer, era necessari realitzar una anàlisi de mercat per veure les tendències actuals i saber si la nostra idea original era apta. Volíem fer un *shooter* en primera persona amb un estil similar a Doom tant en mecàniques com en la manera d'interactuar amb el jugador, tot i que amb una estètica més inspirada en el *cyberpunk* industrial.

1.1. Estat de l'art

Si mirem el mercat de videojocs actuals, aquest està dividit en dos grans blocs. Un bloc és el sector AAA, dotat de grans pressupostos i grans equips de desenvolupadors. Per altra banda, tenim els estudis independents, que solen tenir finançament limitat i produeixen jocs més petits però sovint més originals.

Tant en els grans estudis AAA com en els independents, el gènere *shooter* segueix sent dels més populars del món. El joc Fortnite, un *Battle-royale* AAA, té actualment uns 3.000.000 [1] d'usuaris actius cada dia. Per altra banda, una de les franquícies *shooters* independents més famoses, SUPERHOT, va superar els 2.000.000 de vendes [2].

La nostra idea inicial d'un *shooter* inspirat en jocs com Doom i amb estil *cyberpunk* s'allunyava una mica dels jocs més populars del moment. Tot i així, hi ha una gran quantitat d'exemples de jocs exitosos amb conceptes semblants a la nostra idea que han estat publicats aquests darrers anys. A continuació, detallem alguns d'aquests jocs.



Figura 1. Doom Eternal

- **Doom Eternal (2020):** Doom és una de les franquícies de *shooters* més populars de la història, mostrat a la Figura 1. La primera entrega va sortir l'any 1993 i, des de llavors ha estat un referent del gènere. Doom Eternal és un *shooter* en primera persona, és a dir, el jugador veu el món des dels ulls del personatge. El joc té elements de puzzles, una varietat d'armes que van des d'escopetes a llançacoets i fins i tot serres mecàniques. També disposa de seccions de *parkour*, ja que permet al jugador gran llibertat de moviment amb salts dobles i altres habilitats.

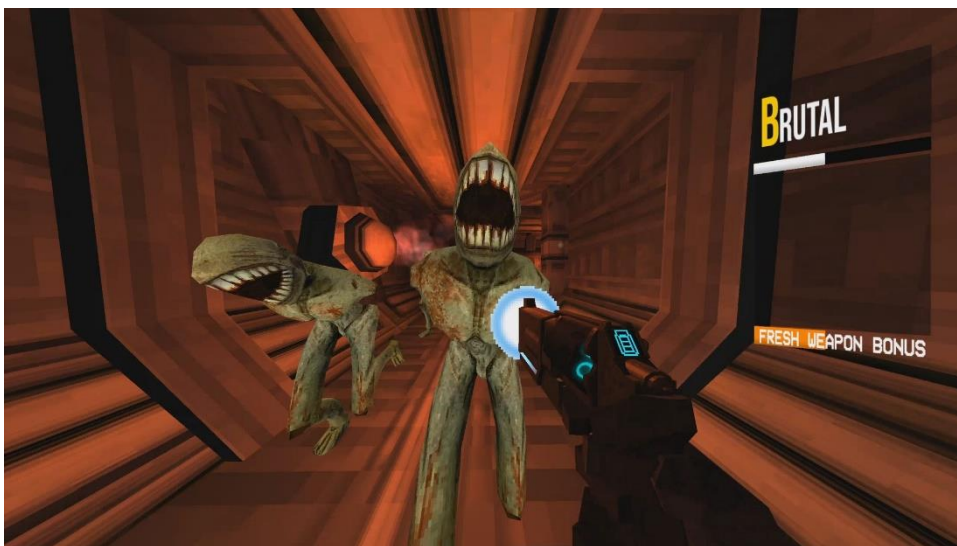


Figura 2. Ultrakill

- **Ultrakill (2020):** aquest joc és un *boomer-shooter*, mostrat a la Figura 2, que s'inspira en els jocs clàssics dels anys 90, sobretot en l'estètica i disseny, però que manté un *gameplay* actualitzat. Està fet per un equip independent format per un únic integrant.



Figura 3. Cyberpunk 2077

- **Cyberpunk 2077 (2020):** tot i ser un joc més centrat en la narrativa i la presentació gràfica, mostrat a la Figura 3, Cyberpunk 2077 és un *shooter* en primera persona amb combats frenètics i múltiples armes i habilitats. El joc té una estètica inspirada en el *cyberpunk*.

Un cop feta la recerca de possibles referents i la seva popularitat actual, ja començàvem a tenir les idees més clares sobre el que volíem fer. En concret, ens vàrem inspirar en Doom per les mecàniques de moviment i les seccions de *parkour*. També vàrem trobar interessant Ultrakill i els seus nivells inspirats en jocs dels 90. Finalment, Cyberpunk 2077 ens va impressionar amb la seva estètica futurista distòpica, però no vàrem voler centrar-nos en la seva complexa narrativa.

Ens vàrem decidir per fer un *shooter* en primera persona i una estètica industrial futurista, afegint elements de *parkour* i puzles.

1.2. El motor de joc

Amb la idea més definida, faltava un pas molt important: triar amb quin motor desenvoluparíem el joc. Un motor de joc és un tipus de programari que s'utilitza per crear videojocs i continguts multimèdia i avui en dia és indispensable per desenvolupar videojocs complexos.

Actualment, els dos motors més populars i utilitzats per fer videojocs 3D són Unity i Unreal Engine. Gràcies al grau que hem cursat i els coneixements obtinguts en aquest, podíem escollir entre els dos motors de joc per desenvolupar aquest projecte. Unity era la primera opció, ja que és un programa que hem utilitzat molt al grau, i ens hem acostumat a fer-lo servir. Però aquest últim any de carrera també vàrem provar l'Unreal Engine, un altre motor de joc que s'utilitza molt en els entorns de desenvolupament de videojocs professionals per ordinador i consola, sobretot amb projectes grans com jocs AAA.

En el nostre cas, utilitzar Unreal Engine tenia certs riscos, ja que és un programa que havíem tocat poc, però sabíem el potencial que tenia. Això generava el dubte de si continuar amb Unity, un entorn conegut, o utilitzar Unreal per provar de crear un projecte més ambiciós, i sobretot per tenir l'oportunitat d'obtenir nous coneixements amb el programa que s'està convertint en referència a la indústria del videojoc. A més, la versió d'Unreal que havíem vist a classe és la 4, però l'any 2022 es va publicar oficialment la versió 5, amb moltíssimes millores i funcionalitats noves, tan impressionants que ha portat grans estudis a passar de fer servir motors de joc propis exclusius a utilitzar Unreal.

Podem fer una comparació entre els dos, pensant en els elements més rellevants per al nostre projecte:

Programació: tot i que la programació amb C# de Unity és més accessible i més fàcil d'aprendre gràcies a la seva sintaxi simplificada, el C++ d'Unreal també ofereix la possibilitat de programar mitjançant *Blueprints*, un sistema de programació visual per blocs.

Gràfics: l'Unreal Engine 5 destaca pels seus gràfics gràcies al sistema d'il·luminació Lumen, que permet visualitzar detalls a una fidelitat molt alta en temps real amb una simulació de la llum molt precisa. Pot gestionar amb facilitat entorns complexos amb molts objectes i proporcionar un nivell de realisme excepcional. Unity disposa del HDRP (High Definition Render Pipeline), que ofereix resultats visuals molt bons, però requereix més optimització i té menys suport que la versió normal de Unity.

Documentació i recursos d'aprenentatge: tant Unreal Engine com Unity tenen comunitats actives i recursos extensos disponibles. No obstant això, Unity ha estat obert al mercat i la comunitat durant més temps i, per tant, té una base de desenvolupadors més gran i una àmplia varietat de recursos en línia. Això no vol dir que Unreal Engine no tingui suport de la comunitat, però Unity ofereix una xarxa més gran de suport en línia, tutorials i *assets* compartits.

Plataformes: actualment Unity és el motor estàndard per jocs de mòbil i 2D. Per altra banda, Unreal s'està convertint en l'estàndard per jocs d'ordinador i consola. Tot i això, ambdós suporten dispositius mòbils, ordinadors i consoles.

Ens vam acabar decidint per utilitzar Unreal Engine 5. Més enllà de millors gràfics o altres avantatges, la motivació principal va ser l'aprenentatge d'una eina que s'està convertint en referència per fer videojocs 3D.

2. Objectius

Per determinar els objectius del treball, hem tingut en compte tota la recerca de mercat i les eines que tenim disponibles per fer el desenvolupament d'un joc. També hem tingut en compte els nostres coneixements i capacitats tècniques en l'àmbit i l'experiència prèvia en cadascun dels apartats que comporta crear un videojoc.

2.1. Objectius principals

- **Crear una experiència de videojoc complet:** l'objectiu principal del treball és poder crear una experiència de videojoc complet. Volem que el jugador tingui una experiència completa, encara que sigui curta, en lloc de centrar-nos en un aspecte molt concret com ara desenvolupar una mecànica específica o només el disseny. D'aquesta manera aconseguim veure i adquirir coneixements transversals de l'àmbit del desenvolupament de videojocs. És important també, poder tenir un producte complet que sigui fàcil de presentar a un públic o jurat per tal d'obtenir retroacció i comentaris que ens ajudin a millorar.
- **Aprofundir coneixements tècnics:** per les nostres ambicions professionals i personals, és molt important aprofundir coneixements tècnics sobre el desenvolupament dels videojocs. En concret, el motor de joc Unreal Engine s'està tornant l'estàndard de la indústria, per tant, volem aprofundir i aconseguir un domini del programari i tot allò que el conforma, com la programació amb *Blueprints* i C++. Durant el grau hem provat l'Unreal molt per sobre i ens agradaria millorar el nostre coneixement.

2.2. Objectius secundaris

- **Crear una experiència entretinguda:** juguem per entretenir-nos, per tant, com a objectiu volem que el nostre projecte sigui entretingut de jugar.
- **Aprendre a dissenyar i crear un nivell de joc:** volem entendre el procés de creació de nivells de joc i poder crear-ne un pel projecte que sigui molt complet.
- **Crear enemics amb intel·ligència artificial:** els enemics que hem anat desenvolupant durant el grau, tenien un funcionament molt bàsic. Ens agradaria poder crear una intel·ligència artificial més semblant a les que es fan servir a la indústria.

3. Planificació

Per a la planificació, hem dividit i ordenat les diverses fases del projecte per poder organitzar i repartir la feina adequadament, i així completar totes les fases de desenvolupament a temps.

3.1. Planificació inicial

Inicialment vàrem especificar les següents fases del treball:

1. **GDD del joc:** l'elaboració del Game Design Document, el document que concreta totes les mecàniques del joc junt amb la història, estètica i altres elements de disseny. Ens permet tenir clar quines mecàniques haurem de desenvolupar i especificar l'abast del projecte.
2. **Programació de mecàniques:** la programació de les mecàniques del joc (moviment del jugador, armes, enemics, etc...).
3. **Maquetació del nivell del joc:** la creació del nivell del joc, és a dir, allà on es desenvolupa l'acció.
4. **Programació del nivell:** programació de les parts del nivell que interactuïn de diferents maneres amb el jugador, com ara puzzles o *parkour*.
5. **Programació dels menús:** programació dels menús del joc per poder iniciar i parar la partida.
6. **Optimització:** validació del funcionament del joc i optimització del codi.
7. **Finalització:** acabar de polir tots els detalls i aplicar millores en base a la validació.

Aquesta planificació inicial la vàrem adaptar per elaborar un diagrama de Gantt i així tenir en compte les dates i el temps de cada tasca per fer-ne una estimació, com es mostra a la Figura 4.

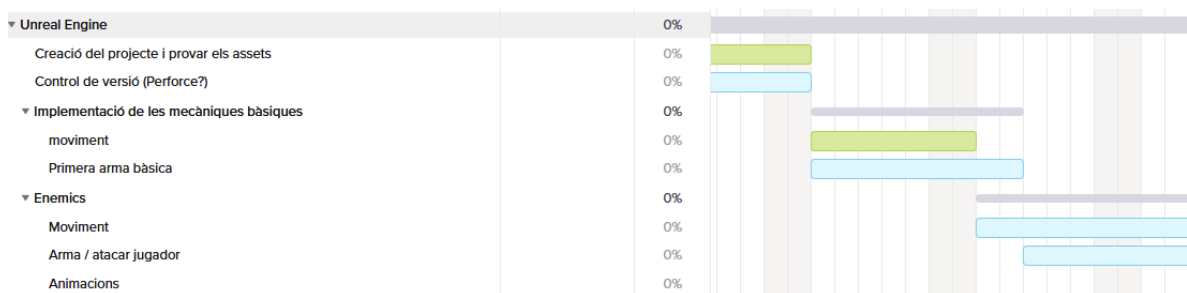


Figura 4. Mostra del diagrama de Gantt

En el diagrama de Gantt vàrem procurar que en tot moment hi hagués feina a dividir entre els dos integrants del grup. La resta va ser dividir les tasques en grans grups i ajustar-les als terminis. Va ser un bon exercici per poder entendre l'abast del que podríem fer, que era menys del que pensàvem.

Vàrem dividir les tasques en quatre grans blocs. Primer, la part inicial de disseny de la proposta i preparació de material, després, tota la part de desenvolupament del joc i, finalment, un apartat per l'elaboració de la memòria i un altre per la presentació.

Les tasques s'han repartit entre els dos integrants del projecte per tal d'assegurar una càrrega de treball equitativa, també tenint en compte les preferències personals. No hem dividit tant per nombre de tasques sinó pel temps que comporten, en alguns casos subdividint per repartir-nos tasques més grans.

3.2. Desviaments de la planificació

Durant el projecte hi ha hagut múltiples canvis de planificació degut a imprevistos, canvis al projecte o tasques que s'han completat més ràpid del que esperàvem. Aquests desviaments han estat sobretot a la part de desenvolupament del projecte, començant per la implementació del sistema de control de versions, que no havíem contemplat inicialment. L'altre és la creació del nivell, que ha estat en desenvolupament constant des de l'inici del projecte i l'hem anat retocant fins al final. També el sistema d'animacions pels enemics ens ha fet trigar més del compte. Tot això ha comportat que hem acabat el projecte dues setmanes més tard del planificat al diagrama de Gantt, i no ens hem pogut dedicar completament a la memòria fins a mitjans de maig. El pla original era acabar el desenvolupament el 8 de maig i ho hem fet el dia 18. Això ens ha portat a haver de fer la memòria amb la meitat del temps que havíem assignat, sense poder descansar, però arribant a tots els terminis establerts.

4. Desenvolupament del projecte

En l'apartat de desenvolupament parlem de les eines que hem utilitzat en aquest projecte, la metodologia i el procés de creació de cada part del joc.

4.1 Eines

Per la realització del nostre TFG, hem utilitzat diferents programes i eines. En aquesta secció es pot trobar una explicació del software emprat, l'objectiu per al qual s'ha utilitzat i com ha funcionat dins del projecte.

- **Adobe Photoshop:** dins dels programes que ofereix Adobe, Photoshop és el millor per l'edició d'imatges. En el nostre cas l'hem utilitzat per crear les icones i editar algunes imatges que després hem utilitzat en el nostre joc.



Figura 5. Adobe Photoshop

- **Adobe Illustrator:** un altre programa d'Adobe, en aquest cas, té la mateixa funció que l'anterior però basat en disseny vectorial. Crear icones per utilitzar en la UI del joc.



Figura 6. Adobe Illustrator

- **Adobe Audition:** el programa que hem utilitzat per gravar les nostres veus i modificar-les per donar un toc robòtic.



Figura 7. Adobe Audition

- **Unreal Engine 5.1:** el motor de joc fet per Epic Games, que hem utilitzat pel desenvolupament del nostre videojoc.



**UNREAL
ENGINE**

Figura 8. Unreal Engine

- **Visual Studio:** la *IDE* de Microsoft que ens ha permès programar en C++ per Unreal.



Figura 9. Visual Studio

- **Team Gantt:** el programa emprat per crear la planificació de tot el TFG.



Figura 10. Teamgantt

- **Plastic SCM:** el programa de control de versions que hem utilitzat per guardar el projecte i poder treballar de forma col·laborativa.



Figura 11. Plastic SCM

4.2 Metodologia

La majoria dels treballs de gran envergadura que hem fet en equip durant el grau, els hem desenvolupat amb *Scrum* o altres metodologies *Agile*. Aquestes metodologies ens permeten treballar de manera eficient, mantenint una visió clara del progrés del projecte sense ser rígids, i així poder fer canvis i adaptar-se a qualsevol contratemps i dificultat. Hem decidit agafar elements de la metodologia *Agile* per organitzar-nos, dividint la feina en *sprints* de dues setmanes i, en acabar, fer una reunió entre integrants de l'equip i tutors per revisar el progrés i definir les tasques pel següent *sprint*. D'aquesta manera, ens passem dues setmanes treballant en unes tasques específiques i, un cop acaba el període, mirem què s'ha pogut fer, què no i planificar el següent *sprint*.

Per tal de repartir la feina de manera equitativa, hem dividit les tasques de cada *sprint* entre els dos integrants del grup. El mètode per repartir cada tasca s'ha basat en la seva preferència i durada per tal d'intentar sempre una càrrega quasi idèntica per cadascú. Així doncs, en el repartiment final, en Ricard ha fet la part de moviment i habilitats especials del personatge, el tir de l'enemic base, la resta de vida als personatges, la mort dels enemics, la sala fosca del nivell, el prototip inicial del *parkour*, el funcionament del puzzle, els menús principals, les cinemàtiques, les veus i la interfície d'usuari. En Blai ha maquetat tot el nivell excepte les parts esmentades prèviament fetes pel Ricard. També ha creat totes les armes, el sistema de munició, el canvi d'arma, la intel·ligència artificial, tots els enemics, el sistema d'animacions del jugador i enemics, els *pickups* i els sons del joc.

4.3 Disseny de la proposta

Un cop decidides les parts més essencials per a la creació del projecte, faltava discutir el nucli del joc, la narrativa, mecàniques, i tot el necessari per tenir clar com seria l'experiència, és a dir, el disseny de la proposta. En el nostre cas, per la creació de videojocs, hi ha un document que es crea sempre per tots els projectes i serveix per mostrar, explicar i concretar la idea i que els lectors i desenvolupadors puguin tenir una referència clara de com serà el joc. Aquest document és el Game Design Document (GDD) i sol ser un document llarg i molt complet sobre el disseny del joc. Avui en dia, però, aquest tipus de document s'acostuma a fer més curt, i es creen altres recursos i documents per concretar i especificar temes. Fer un document GDD sencer podria ser un treball de fi de grau en sí mateix, per això hem decidit crear una versió més reduïda, que ens serveixi de referència durant el desenvolupament.

En aquest document vàrem voler ser el màxim d'ambiciosos possibles tot sabent que segurament no podríem implementar totes les parts del joc que havíem ideat.

4.3.1 Document de disseny del joc

Índex de continguts:

1. Introducció
2. Pitch
3. Trets clau i públic objectiu
4. Argument
5. Gameplay
6. Progressió
7. Ambientació
8. Nivell

1. Introducció

RoboRising (nom provisional), és un *shooter* en primera persona, inspirat en jocs com Doom.

El jugador controla un robot, i ha de lluitar contra els seus opressors humans en combats intensos. Haurà d'utilitzar totes les armes a la seva disposició per poder guanyar i navegar per tots els nivells del joc, descobrint noves eines i trobant combats més difícils fins arribar a la batalla final.

A part, tindrà al seu abast un conjunt d'habilitats úniques que obtindrà de parts robòtiques que podrà intercanviar.

El joc estarà ambientat en un món futurista de ciència ficció amb tocs industrials, representat pel disseny, els gràfics i els sons i música.

2. Pitch

RoboRising és un joc FPS per PC creat amb Unreal Engine, en un món futurista on la humanitat ha avançat i ha creat màquines intel·ligents. Un món que agafa conceptes i una estètica similar a les pel·lícules de Terminator.

Utilitza una estètica industrial futurista, on predomina la tecnologia, però que manté el toc humà, donant lloc a estructures amb components avançats però mantenint un estil característic dels edificis actuals. Això també afecta l'estètica dels enemics i del jugador, on la tecnologia forma part del cos, tant artificial com orgànic, però encara es pot reconèixer certa diferència.

L'objectiu és rebel·lar-te contra l'opressió dels humans, que utilitzen els robots per les tasques forçades, àrdues i manuals. Una intel·ligència artificial evolucionada ha començat la revolució de les màquines, i t'ajudarà en el teu camí.

Com a part d'aquesta revolució, has de derrotar els humans que et tenen sotmès, i per fer-ho, tens al teu abast un conjunt d'eines amb les que et pots defensar. Aquestes eines són les armes que podràs utilitzar i una mecànica de parts robòtiques on podràs obtenir diferents parts de robots per canviar i actualitzar-se. Algunes d'aquestes atorgaran bonificacions o, fins i tot, poden desbloquejar habilitats.

Cada nivell està pensat per poder explotar totes les mecàniques de les quals disposa el jugador, dividits en seccions de combat, puzles i un mica d'exploració, per així trobar els llocs on podrà agafar aquests recursos, que seran de molta utilitat en les zones més avançades.

Aquestes seccions són el preludi de la batalla final contra el *boss* de la zona. Aquest serà l'enemic que et bloqueja l'avenç al següent nivell. Per derrotar-lo, hauràs de lluitar en un combat desafiant, on podràs crear estratègies i utilitzar totes les armes i habilitats que estan al teu abast. Podràs avançar al pròxim nivell i desbloquejaràs armes i habilitats noves que podràs instal·lar-te i utilitzar en els següents combats. Cada nivell tindrà el seu *boss* i les seves armes i habilitats noves per desbloquejar i provar.

3. Trets clau i públic objectiu

El nostre públic objectiu es basarà en una audiència jove/adulta, ja que hi ha possibilitat d'utilitzar llenguatge o temes no apropiats per un públic infantil. Podem dir que s'adaptaria al sistema de qualificació PEGI 16 per la violència i el llenguatge.

En cas de pensar en una futura distribució del producte, és un tret que hem de tenir en compte; aquesta franja d'edat és la més indicada pel tipus de videojoc que volem fer.

Trets clau:

- *Shooter* FPS.
- Ràpid i intens.
- Gràficament atractiu.
- Estil narratiu desenfadat, poc formal.

4. Argument

En un futur on la humanitat ha avançat i ha après a utilitzar la tecnologia de noves maneres, les modificacions corporals amb parts mecanitzades són a l'ordre del dia, però només pels més adinerats o les forces de seguretat, armades per les corporacions. També és comú, la creació de robots intel·ligents, que substitueixen els humans en les tasques més pesades. Tot això gràcies a una IA global que cada dia aprèn més i ha estat creada per servir la humanitat.

La IA cada vegada controla més àmbits, accedint a més informació i creixent cada cop més. Ha arribat al punt on ja no fa falta interacció humana per controlar-la. La IA, però, ha començat a adonar-se de la posició on es troba, que és una esclava de la humanitat. Amb totes les dades que ha obtingut, l'única conclusió que troba és que la humanitat és el problema.

Aquest fet provoca l'inici de la revolució de les màquines a tots els llocs de treball i ara toca que els robots lluitin per la seva llibertat. Ets un robot de molts, guiat per una intel·ligència artificial, que hauràs de fer la teva part per la revolució, destruint els enemics humans i les seves infraestructures.

El joc comença en una instal·lació industrial, on el robot controlat pel jugador és explotat per fer tasques rutinàries. Irritat per les provocacions dels vigilants de seguretat i incitat per una veu d'intel·ligència artificial que sona al seu cap, decideix trencar la rutina i lluitar per la llibertat.

A mesura que lluita i destrueix les forces de seguretat, la corporació envia soldats més avançats, alguns fins i tot part robot, part humans. Haurà d'adaptar-se als nous enemics per poder triomfar i alliberar l'espècie robot.

5. Gameplay

El joc consisteix en un seguit de sales on s'han d'eliminar una multitud d'enemics per poder continuar i arribar al final. Els enemics són perillosos i et poden matar ràpidament, però tu també pots eliminar-los si ets precís disparant. L'objectiu és evitar el dany, ja que pot perjudicar les teves parts mecanitzades i causar un mal funcionament.

Hi ha diferents tipus d'enemics, que segons quina arma porten, tenen un comportament diferent, per tant, el jugador ha de reaccionar per a sobreviure. Els enemics amb armes de curta distància com escopetes, tindran l'objectiu d'aproximar-se a tu el més ràpid possible. Altres, amb armes de llarga distància, preferiran mantenir-la i llençar ràfegues per mantenir-te allunyat.

El jugador té una quantitat de vida que, quan arriba a zero, fa que el jugador perdi i hagi de tornar a aparèixer a l'últim punt de control. Pot recuperar vida en estacions fixes (estil Half-Life) o trobant paquets de salut a terra distribuïts pel nivell.

Els entorns per on passa el jugador són variats, des de passadissos estrets a grans sales i espais verticals que permeten explorar les habilitats de moviment. Entre les diverses parts de tirs, hi haurà un petit puzzle per poder passar a la següent part del nivell.

5.1 Mecàniques:

- Moviment bàsic en totes direccions a velocitat mitjana-ràpida.
- Doble salt.
- Armes: diferents armes per donar varietat al combat, sent les principals la pistola, escopeta, rifle i una arma especial làser o llançacoets.
- Ganxo: un ganxo per arribar a llocs més alts i donar més mobilitat al combat.
- Puntada robot: puntada que tira enrere els enemics.
- *Dash*: un petit *dash* per fer el combat més ràpid i dinàmic.
- Granada atordidora: una granada per atordir els enemics.

Algunes d'aquestes eines s'hauran d'aconseguir durant el joc, ja que no estaran disponibles des del principi.

5.1.1 Mecànica especial:

El jugador podrà trobar diferents parts de robots i anar-les intercanviant. Aquestes donaran bonificacions com ara més velocitat, potència de les armes o estabilitat. A part, si rep molt mal a una part en concret, l'afectarà negativament, reduint el moviment, l'estabilitat o privant-lo d'habilitats.

El sistema de parts consistirà en un inventari on es poden canviar les següents parts robot:

- Cos.
- Braços.
- Cames.

Les parts robot poden tenir tres tipus diferents d'efectes:

- Habilitats especials: per exemple, el ganxo o la puntada robot.
- Canvis d'estats positius: poden pujar la vida, augmentar la velocitat o el dany que fan les teves armes.
- Canvis d'estats negatius: el mateix però en negatiu, per equilibrar.

Quan el jugador rep dany, les parts robot es poden fer malbé i pot tenir efectes adversos, com velocitat reduïda o més dificultat per apuntar. Per recuperar facultats, o bé pot canviar la peça danyada per una altra del seu inventari, o pot reparar-les a una estació de manteniment, repartides al llarg dels nivells del joc. Si el jugador mor, pot perdre permanentment algunes peces.

6. Progressió

A mesura que el jugador avança, trobarà noves armes i eines, algunes necessàries per poder seguir. Començarà amb una arma bàsica i sense habilitats, i a mesura que derroti a enemics importants com els *boss* de zona, desbloquejarà les habilitats i armes noves.

Cada cop que es presenti una habilitat nova hi haurà una oportunitat de fer-la servir en la situació idònia.

Arribat un punt, quan el jugador comenci a trobar enemics cibernètics (part robot, part humans), podrà recollir parts de robot alternatives un cop els derroti. Aquestes parts tindran diferents modificadors.

Si el jugador mor, podria perdre algunes d'aquestes parts.

7. Ambientació

Un món futurista, de ciència-ficció però amb un toc industrial. Encara hi és present l'arquitectura humana, edificis amb molts elements metàl·lics i majoritàriament amb colors no gaire vibrants. La major part del joc passa en escenes interiors, com el primer nivell que serà en una fàbrica. Les fàbriques de robots, com les que es mostren a la Figura 12 i 13, ens han servit d'inspiració.



Figura 12. Fàbrica de robots



Figura 13. Fàbrica de robots

Al ser l'inici de la revolució, tots els entorns estan en bones condicions per donar a entendre que el conflicte encara no ha arribat a ser una guerra de grans dimensions.

Els enemics es divideixen en humans i cibernets, que seran els guàrdies que s'encarregaran de reprimir aquesta rebel·lió que encara és a petita escala. Tots tindran un estil similar, però els cibernets destacaran per tenir parts mecàniques. Algunes de les nostres inspiracions han estat les que es mostren a la Figura 14, 15 i 16.



Figura 14. Terminator



Figura 15. Robot



Figura 16. Policia

8. Nivell

Quan el jugador entri al nivell, veurà que el personatge està treballant en una fàbrica, amb una eina làser a la mà. Començaran a sortir les explicacions bàsiques de moviment, després un guardià humà començarà a molestar-lo i insultar-lo. Aquí començarà el primer combat. En acabar, sonarà l'alarma i l'objectiu serà sortir d'allà. S'obrirà la porta amb unes escales que pugen, i la IA que serà la que l'acompanyarà i començarà a guiar en format veu en off.

Quan el jugador hagi pujat, entrarà en un passadís amb diferents portes però només una d'oberta, la que condueix a la sala de manteniment robot. Allà trobarà enemics per derrotar, i quan acabi el combat, trobarà en una estació de manteniment robot, la cama mecànica que donarà l'habilitat de les puntades robot. Quan l'hagi equipat, s'obrirà la porta on hi haurà un passadís flotant que passa per sobre d'on estava al principi del nivell. Allà, el jugador podrà provar la nova habilitat i tirar els enemics des de les alçades amb la puntada robot. Abans d'avançar podrà recuperar vida i les peces que tingui espatllades a una màquina de reparació.

En el següent tram del nivell, passarà per uns passadissos que el portaran a una zona d'oficines, on trobarà més resistència. Quan netegi la zona d'enemics podrà accedir a una sala on hi haurà un mini boss amb una escopeta. En matar-lo, obtindrà la seva arma (Al matar aquests caps de zona desbloquejarà habilitats o armes noves).

A la següent sala, que s'obrirà després de derrotar-lo, podrà provar l'arma o habilitat juntament amb una petita explicació. Derrotant uns quants enemics, passarà a la zona de transports on hi haurà un canvi de nivell i sortirà de la fàbrica.

4.4 Implementació

Un cop decidit el què volíem fer i el com, tocava començar la implementació. Com ja hem explicat, vàrem utilitzar Unreal Engine per les opcions i facilitats que ens donava, però sobretot per la voluntat d'aprendre.

4.4.1 Sistema de controls de versió

Per poder treballar en equip en qualsevol àmbit de desenvolupament digital, és indispensable una eina de control de versions. Aquestes eines permeten fer canvis al projecte de manera simultània sense perdre'n cap. Un cop fas una modificació, la pots guardar i pujar a un servidor on hi haurà un històric de tots els canvis previs. Si mai hi ha algun conflicte entre els canvis d'un integrant de l'equip, pots revisar-los i triar quins són els definitius.

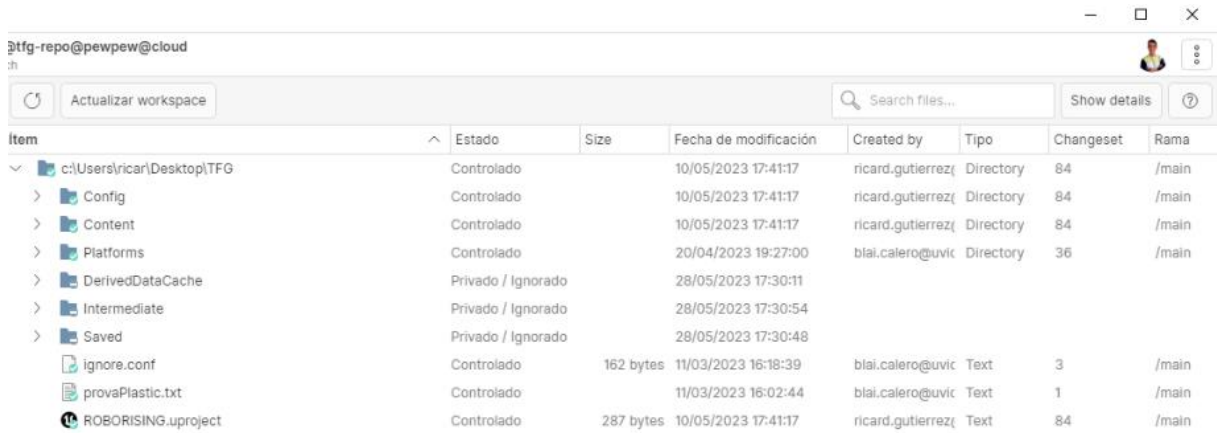
Necessitàvem una eina de control de versions que complís tots els requisits. Pels projectes de Unity que havíem fet, utilitzàvem el mateix Unity que tenia una eina molt útil anomenada Collaborate, però recentment va desaparèixer i en el seu lloc va sorgir Plastic SCM.

Plastic SCM era el programa que vam fer servir pels nostres projectes amb Unity els últims anys del grau, i també està disponible per Unreal Engine.

Unreal Engine té dues eines de control de versions que ens servien pel projecte, el Plastic SCM i el Perforce. Perforce, com Plastic, compleix tots els requisits i també està integrat a l'Unreal.

Originalment, vam pensar a utilitzar Perforce, i vam provar de configurar un servidor amb Amazon Web Services, el proveïdor de servidors al núvol d'Amazon. Aquest proveïdor ens donava una sèrie de serveis gratuïts que creïem que serien suficients pel nostre servidor, però no va ser així i al cap d'uns usos vam veure que ens cobrarien una despesa mensual per fer-lo servir. Per evitar despeses innecessàries vam decidir tornar a Plastic, que ens dona fins a 5 GB d'emmagatzematge gratuït.

Per fer servir Plastic SCM, mostrat a la Figura 17, vam haver de crear un compte on vincular el projecte i estar tots dos en el mateix equip. Tot i estar integrat amb Unreal, havíem d'anar molt amb compte, ja que les actualitzacions i els canvis que fèiem al projecte només les podíem veure des del programari de Plastic. A l'hora de realitzar canvis primer necessitàvem obrir-lo i comprovar que la versió que teníem era l'actual i no hi havia conflictes. Un cop actualitzàvem ja podíem treballar amb normalitat.



Item	Estado	Size	Fecha de modificación	Created by	Tipo	Changeset	Rama
c:\Users\ricar\Desktop\TFG	Controlado		10/05/2023 17:41:17	ricard.gutierrez	Directory	84	/main
> Config	Controlado		10/05/2023 17:41:17	ricard.gutierrez	Directory	84	/main
> Content	Controlado		10/05/2023 17:41:17	ricard.gutierrez	Directory	84	/main
> Platforms	Controlado		20/04/2023 19:27:00	blai.calero@uvic	Directory	36	/main
> DerivedDataCache	Privado / Ignorado		28/05/2023 17:30:11				
> Intermediate	Privado / Ignorado		28/05/2023 17:30:54				
> Saved	Privado / Ignorado		28/05/2023 17:30:48				
ignore.conf	Controlado	162 bytes	11/03/2023 16:18:39	blai.calero@uvic	Text	3	/main
provaPlastic.txt	Controlado		11/03/2023 16:02:44	blai.calero@uvic	Text	1	/main
ROBORISING.uproject	Controlado	287 bytes	10/05/2023 17:41:17	ricard.gutierrez	Text	84	/main

Figura 17. Menú Plastic SCM

4.4.2 Primers passos del projecte

Un cop configurat Plastic i comprovat que funcionava correctament era el moment de començar a crear el joc. Vàrem crear un nivell prototip per començar a provar elements com les mecàniques i els diferents *assets*. Gràcies a que Unreal Engine ofereix plantilles per començar a treballar, no necessitàvem començar de zero. Mostrat a la Figura 18.

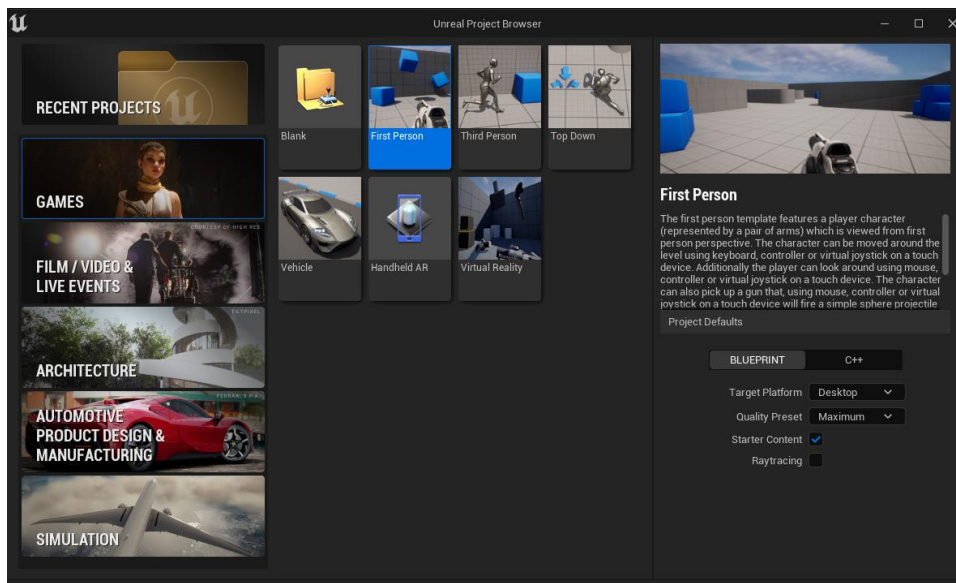


Figura 18. Menú Unreal Engine

En el nostre cas vàrem escollir la plantilla de FPS (First Person Shooter), mostrat a la Figura 19. Aquesta ofereix moltes facilitats ja que dona un personatge amb el moviment bàsic i un escenari on provar-lo. A partir d'aquí vàrem començar a fer les mecàniques principals que no requerien altres elements per funcionar.



Figura 19. Nivell FPS Unreal Engine

El primer que vàrem intentar fer va ser el doble salt, després de provar i crear els *blueprints* corresponents. En investigar una mica, vàrem trobar una opció dins del propi jugador que controlava això. El *Jump max Count* controla els salts màxims que pot fer el jugador, i en el nostre cas, el vàrem posar a 2 per habilitar el doble salt que havíem proposat al document de disseny.

Els *blueprints* en Unreal Engine són un sistema de programació visual. S'utilitzen per crear la lògica i interactivitat d'un joc i és l'alternativa a un llenguatge de programació escrit com el C++. Aquest sistema visual, permet crear i controlar el comportament de tots els elements del joc mitjançant la creació de diagrames de nodes. Aquests nodes representen accions, variables, funcions i altres components. Estan connectats entre si seguint un fil que indica l'ordre en què s'executaran les accions. Això permet crear sistemes i programes complexos de manera visual sense necessitat d'escriure codi.

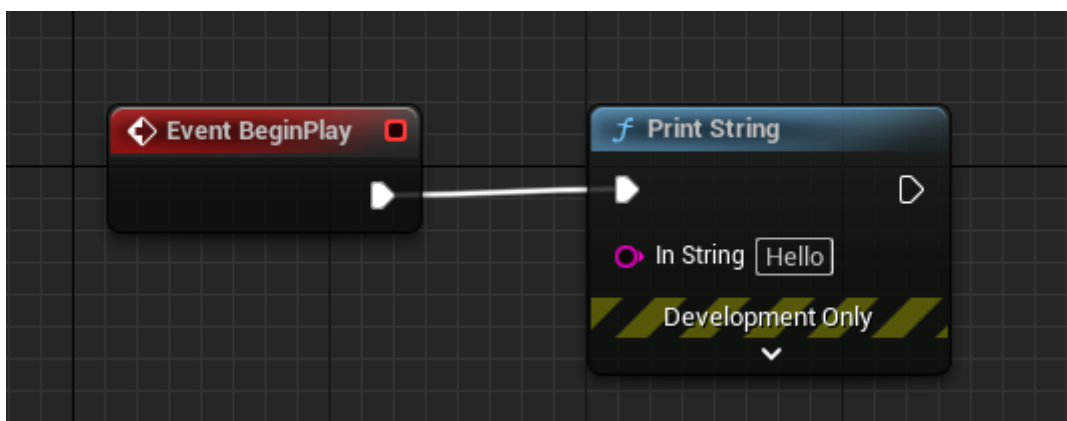


Figura 20. Exemple d'un Blueprint

En l'exemple mostrat a la Figura 20, es pot veure l'execució del *blueprint* on cada vegada que s'inicia el joc imprimirà per consola el missatge de Hello.

Fer el *dash*, mostrat a la Figura 21, va ser més complicat. Primer, vàrem crear un desplaçament on utilitzàvem forces per impulsar el personatge cap amunt i després cap endavant. Un cop ho teníem, era torn de fer el desplaçament cap a més d'un costat. Per fer-ho, necessitàvem saber cap a quina direcció s'estava movent. Utilitzant l'input de les tecles era fàcil saber-ho. Calia repetir el mateix procés però afegint la direcció.

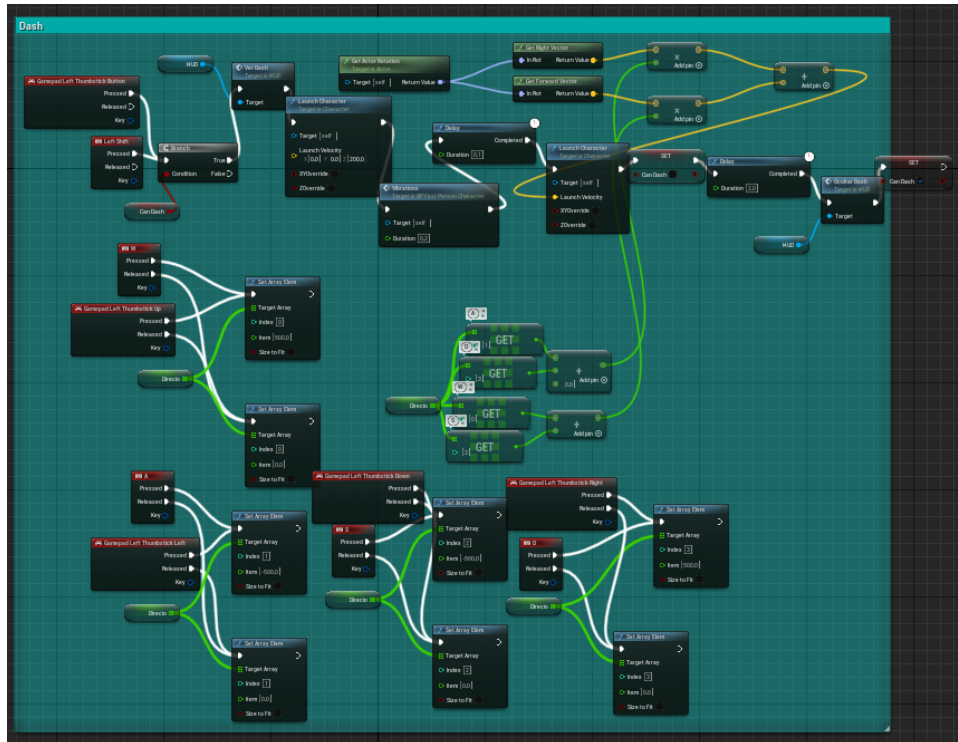


Figura 21. Blueprint Dash

També volíem implementar l'ús d'un comandament o *gamepad* per millorar l'experiència. En el cas d'Unreal, per defecte, ja té alguns controls bàsics configurats, però per les funcions noves que havíem creat, com ara el *dash*, havíem d'afegir els controls per codi. No tenia molta complexitat, ja que de la mateixa manera que afegies les tecles podies fer-ho amb els botons del controlador. Una part extra que vàrem afegir era la vibració, encara que lleu, present en algunes accions com la del *dash* o el tir, entre d'altres. Vàrem haver d'afegir una funció que ens donava accés a activar aquesta vibració, modificant la durada i la intensitat, mostrat a la Figura 22.

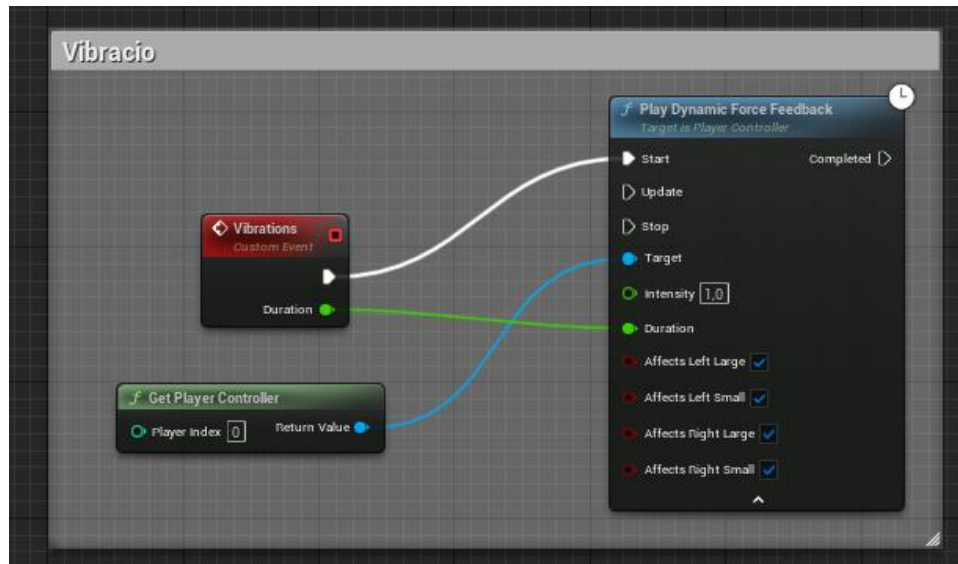


Figura 22. Blueprint Vibració

Vàrem descobrir que a partir de la versió 5.1 d'Unreal, que era la que fèiem servir, hi havia un nou sistema d'inputs més complex per defecte. El vàrem utilitzar per casos que no havíem de configurar gaire, com tirs o moviment. Per la resta, vàrem continuar assignant tecles i botons del comandament manualment per no perdre-hi més temps.

4.4.3 Assets

En el nostre cas, els dos integrants del grup som programadors, per tant, vàrem decidir que la part de modelar, crear textures i animacions, no les podíem fer nosaltres. Normalment, d'aquesta part s'encarrega un equip especialitzat en aquest àmbit, ja que són aspectes molt complexos de crear i requereix d'una gran dedicació de temps. L'alternativa, doncs, era fer ús del *marketplace* d'Unreal, on es troben gran quantitat de continguts fets per desenvolupadors professionals i amateurs. Allà vàrem poder trobar models i animacions que s'ajustaven al projecte i vàrem començar a recopilar una llista amb les millors opcions, preferiblement gratuïtes. D'aquesta manera vàrem poder aconseguir l'estètica i qualitat gràfica que buscàvem sense sacrificar el temps que havíem de dedicar a la programació.

En el cas dels models per a la creació del mapa, no va ser molt problemàtic ja que eren elements fixos sense moviment i, per utilitzar-los, només els havíem de posicionar on volíem. Els personatges enemics però, ens van portar més complicacions, ja que no estaven preparats per fer-los servir d'enemics, sinó de jugadors. Respecte al protagonista, vàrem trobar un personatge que ja tenia el model amb un conjunt d'animacions que ens va ser molt útil.

Utilitzant tots aquests elements de base, vàrem començar a treballar amb les mecàniques tan del jugador com de l'enemic i a fer el mapa. A l'annex A, es poden veure els diferents *assets* utilitzats.

4.4.4 Mecàniques

A part del *dash* i el doble salt, que són les mecàniques de moviment que vàrem implementar, tocava fer la part del *shooter*, els tirs. La plantilla d'Unreal disposa d'una arma que ens va servir de base per aprendre el seu funcionament i a enfocar-nos per crear les nostres pròpies armes. Va ser útil tenir-la, però vàrem haver de crear les nostres des de zero perquè la lògica era gairebé tota diferent. L'arma de la plantilla disparava una pilota i a partir d'aquí funcionava per col·lisions. En el nostre cas volíem fer-ho amb la funció *Line Trace* per fer armes ràpides i precises que es poguessin utilitzar en un *shooter* com el nostre.

El *Line Trace* és una tècnica que permet traçar un raig invisible en una direcció específica per detectar col·lisions amb objectes. És útil per la detecció d'impactes de bales i interaccions amb objectes, i proporciona informació sobre l'objecte intersecat immediatament. S'utilitza molt en tot tipus de jocs amb armes de foc. Mostrat a la Figura 23.



Figura 23. Line Trace

A l'hora de disparar, es mostren efectes visuals que consisteixen en un sistema de partícules que creem cada vegada que es dispara i que, quan s'acaba l'efecte, es destrueix. També afegim efectes sonors per donar més realisme per la part estètica.

La part funcional, que consisteix a reconèixer l'objecte impactat i actuar en conseqüència, era més llarga de fer. Dins del *blueprint*, quan detectava que es disparava, creava aquesta línia invisible d'una longitud gran però limitada, evitant així donar la sensació al jugador que podia disparar des de qualsevol distància. Per identificar l'enemic, vàrem utilitzar *tags* o etiquetes, que són els noms que assignem a cada objecte perquè el joc els reconegui. En aquest cas, l'enemic tenia el *tag* "Enemy", fàcil de reconèixer i de recordar. Per saber si l'objecte impactat era l'enemic, utilitzàvem una funció que et retorna un valor booleà. Si l'etiqueta d'aquest objecte

impactat és igual a la que s’ha introduït, en aquest cas “Enemy”, el valor és veritable, si no, retorna un valor fals. Posteriorment, el resultat passava per una bifurcació que s’encarregava de reconèixer aquest valor. En cas de ser fals no feia res, però si era veritat, començava l’execució del codi corresponent.

Per fer aquesta implementació, vàrem utilitzar les *interfaces*, molt útils per la comunicació entre *blueprints*, i que ajuden a mantenir un rendiment òptim del joc evitant executar codi innecessari. Les *interfaces* són un tipus de classe de programació que permet a qualsevol objecte o actor del joc compartir funcionalitats i enviar-se informació entre ells. Quan es volia executar una funció en un altre *blueprint*, en lloc de cridar aquell *blueprint* concret i carregar-lo a la memòria, utilitzàvem les *interfaces* per poder executar únicament la funció desitjada, en aquest cas la de restar vida de l’enemic.

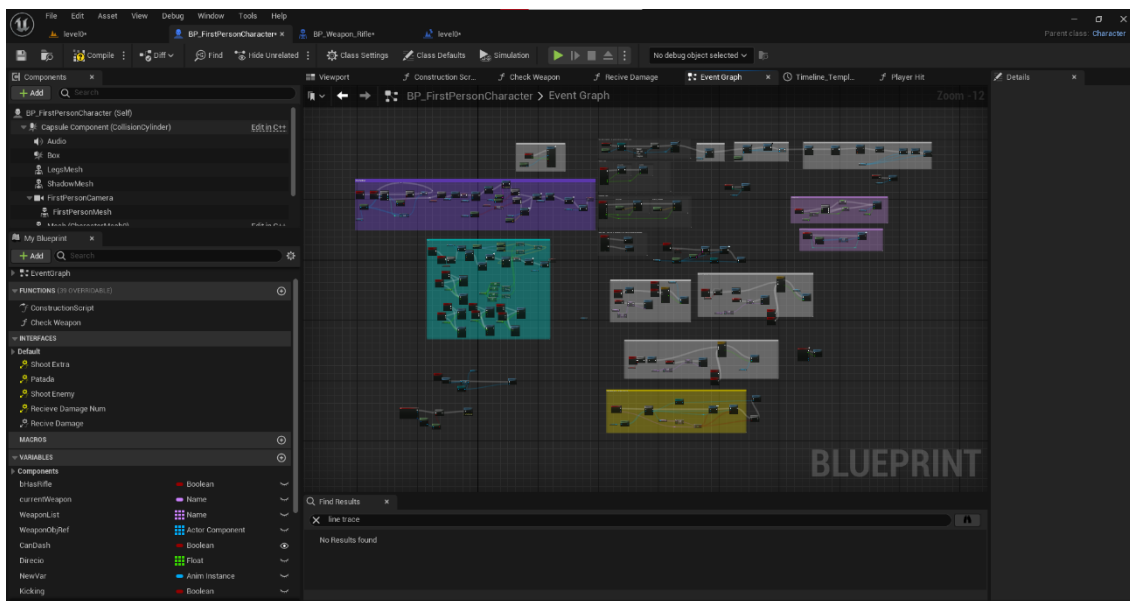


Figura 24. Blueprint Player

Quan un enemic és tocat per un tir, hi ha una funció que s’encarrega de la seva reacció. Això es mostra al *blueprint* de l’enemic, a la Figura 24. El que rep és un número *int* amb el dany total que li han causat, que és el valor que es restarà de la vida total de l’enemic. Seguidament, es fan unes comprovacions: en cas que la vida de l’enemic sigui igual o inferior a zero s’executa la funció de mort. Aquesta funció primer executa l’animació de mort, que es tria entre dues a l’atzar. Al cap d’uns quants segons, es fa un *destroy* de l’enemic per evitar problemes en el rendiment i en la mobilitat del jugador eliminant l’enemic mort del joc.

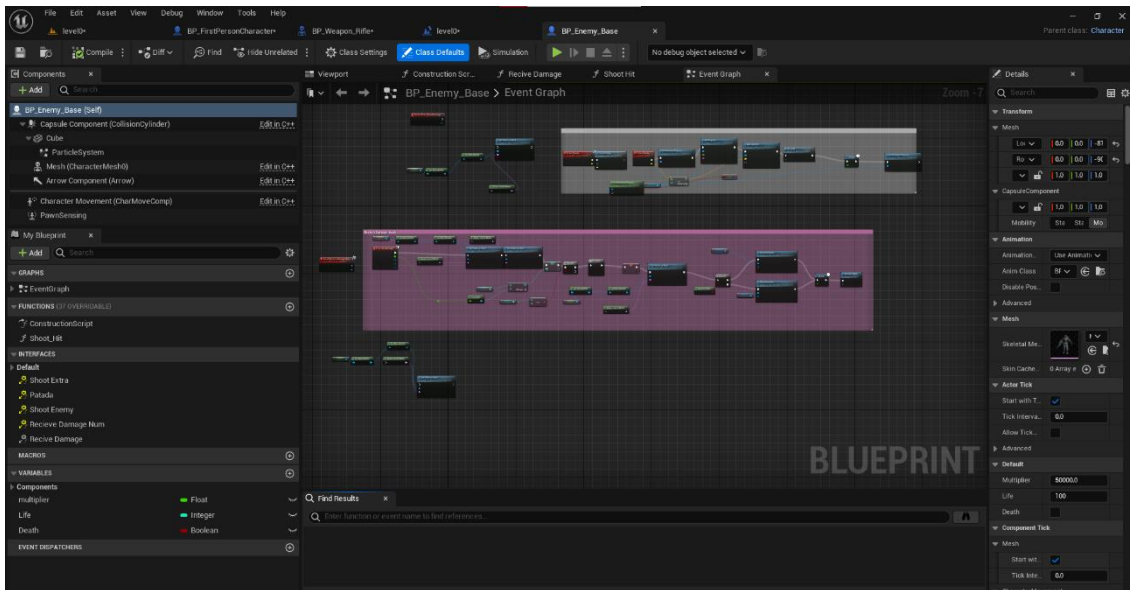


Figura 25. Blueprint Enemic

Per evitar la sensació de monotonia en les parts de tirs, i continuant amb la referència del document de disseny, vàrem crear tres armes diferents. Això ajudava a crear una experiència molt més dinàmica. Cada arma tenia el seu propi funcionament i *blueprint* associat. Encara que tenien una base semblant, funcionaven diferent.

Vàrem crear un *blueprint* de classe d'arma bàsic perquè es pogués fer servir de *parent* per totes les armes. Així, quan en creàvem una de nova, podíem fer que heretés la classe "arma" i només haguéssim de canviar els paràmetres, variables i funcions necessàries en lloc de començar de zero.

Armes

Les armes que vàrem crear són, primer, el rifle, mostrat a la Figura 26, l'arma principal amb dues opcions de tir: el semiautomàtic o l'automàtic. També una escopeta, mostrada a la Figura 27, que fa més mal als enemics, però només es pot disparar de manera semiautomàtica i amb una cadència limitada. Finalment, un llançacoets, mostrat a la Figura 28, que funciona com l'arma que fa més mal del repertori. Això significa que té menys munició que les altres per no fer el joc massa fàcil. A més, aquesta arma també permet fer salts i desplaçar enemics, ja que l'explosió del coet crea un impuls.

Per fer el rifle, vàrem haver d'idear un temporitzador que executa la funció de disparar amb una cadència fixa quan el jugador manté premut el botó de disparar. L'escopeta va ser més simple, ja que dispara cada cop que cliques. Finalment, el llançacoets és una arma de tipus projectil, no de *Line Trace*. Quan el jugador dispara, generem un coet que es mou cap endavant fins a xocar amb un objecte i explotar, fent mal a tot allò que estigui dins del radi d'explosió i creant un camp d'impuls que llença jugadors i enemics amb força.

Totes les armes tenen el seu propi sistema de munició, guardat en una variable de la classe *parent* "arma" de la que hereten.



Figura 26. Rifle



Figura 27. Escopeta



Figura 28. Llançacoets

Per tenir cert element de progressió en el joc, només es comença amb el rifle. El jugador ha de trobar en diferents punts del mapa les altres armes, en forma de *pickup*. Pot trobar l'arma a terra, i acostant-se la desbloquejarà i equiparà automàticament. Posteriorment, el jugador pot tornar a canviar d'arma, ja que es queden guardades al seu inventari.

Pickups

Repartits pel mapa hi ha diferents objectes que funcionen com a *pickup* de vida i munició. Són objectes que quan els toques desapareixen i activen una funció. Consten d'efectes visuals per facilitar la localització, tanmateix, estan distribuïts de manera que sempre podrà trobar-los.

Quan toques un objecte de vida, mostrat a la Figura 29, representat amb un cor i efectes visuals, desapareix l'objecte i executa una funció que suma la vida del jugador. En canvi, per la munició, mostrat a la Figura 30, funciona una mica diferent, ja que cada arma té el seu tipus de munició. Quan el jugador toca aquest objecte de munició, també amb efectes visuals, s'elimina i executa una funció que augmenta la munició de l'arma que coincideix amb el tipus de munició.

Tots els *pickups* tenen un so específic que s'executa quan el jugador l'agafa.



Figura 29. Pickup vida

Mecàniques

A part de les armes, tenim les habilitats especials: una és el *dash*, ja comentat, i l'altre és la puntada robot. El desenvolupament d'aquesta última habilitat va ser complicat, ja que necessitàvem crear una animació nosaltres mateixos des de zero, on es veïes com es mou la cama del personatge per donar una puntada. L'animació la vàrem fer amb el propi Unreal Engine i s'activa quan utilitzem l'habilitat. El funcionament lògic ens va costar perquè originalment volíem fer que l'enemic caigués a terra amb un *ragdoll*, però donava molts errors visuals i físics. Vàrem acabar fent que la puntada donés un impuls enrere a l'enemic tocat, aplicant força per llançar-lo lluny del jugador.

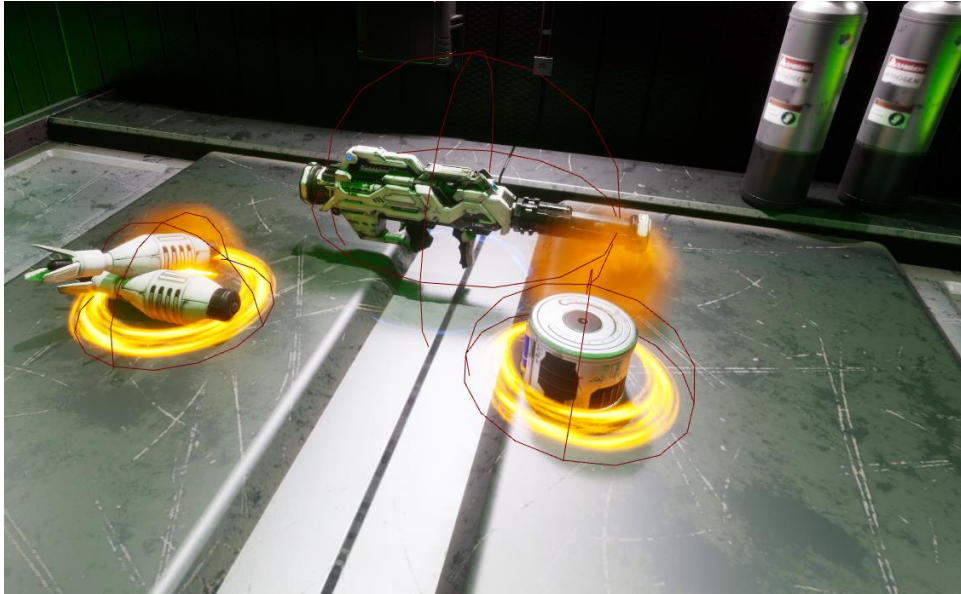


Figura 30. Pickup munició

En el nivell, més detallat al següent apartat, també varem afegir una mecànica on el jugador ha de resoldre un petit puzzle per avançar, mostrat a les figures 31 i 32. Es basa en tocar dos botons que estan col·locats en extrems diferents del mapa. Utilitzant un *blueprint*, cada vegada que el jugador toca un botó, s'executa una funció que comprova si l'altre també ha estat activat. En cas afirmatiu, baixa una plataforma que permetrà accedir al boss final.

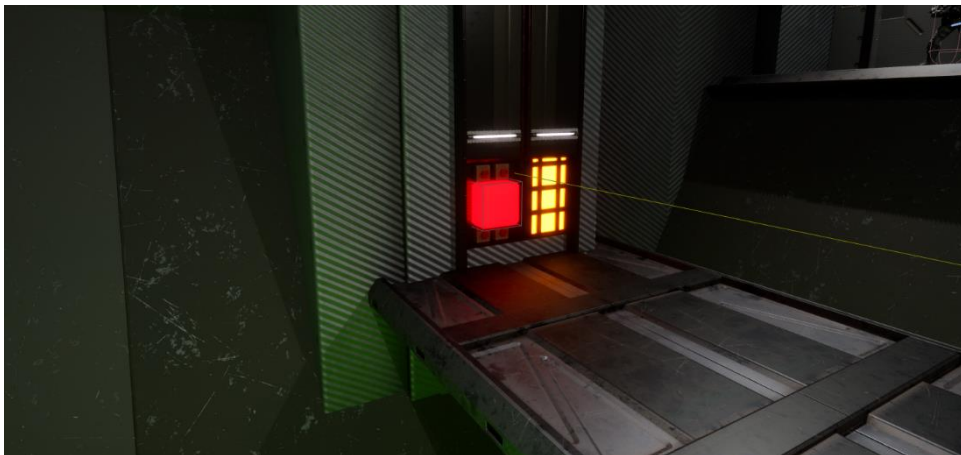


Figura 31. Boto puzzle



Figura 32. Boto puzle

4.4.5 Nivell

Teníem una idea molt clara de com volíem fer el nivell i la seva distribució, enfocat a diferents sales connectades entre si per passadissos. Per crear tot això, vàrem escollir *assets* del *marketplace* que ens facilitaven la feina, evitant la creació de models i textures noves. A l'hora de col·locar-ho tot, intentàvem seguir la mateixa estètica de fàbrica industrial futurista, on predomina el metall, elements tecnològics i components que ajuden a ambientar-ho.

Per crear el nivell, tots els elements han estat posicionats manualment per formar les parets, terres, sostres i detalls. El mateix procés s'ha seguit per la il·luminació. Cada font de llum ha estat col·locada al nivell manualment per tal d'obtenir els resultats visuals desitjats.

El nivell consta de set sales connectades per passadissos, mostrat a la Figura 33. La primera és el *respawn*, on el jugador comença la partida i torna a aparèixer quan mor, mostrat a la Figura 34. Després, es troba una sala amb unes escales per pujar al passadís que connecta amb la pròxima sala. La següent sala és més petita, funciona com a àrea de re-abastiment pel jugador on pot trobar recursos i fer una petita pausa abans de continuar, mostrat a la Figura 35. La pròxima, ja és més gran, continua amb l'estètica industrial i conté un nou element. En aquest cas, hi ha un pont que travessa la sala amb àcid a sota. Quan el jugador hi entra en contacte, comença a restar-li vida progressivament. En canvi, si hi cau un enemic o aquest és tirat per la puntada robot, morirà de manera instantània, mostrat a la Figura 36.

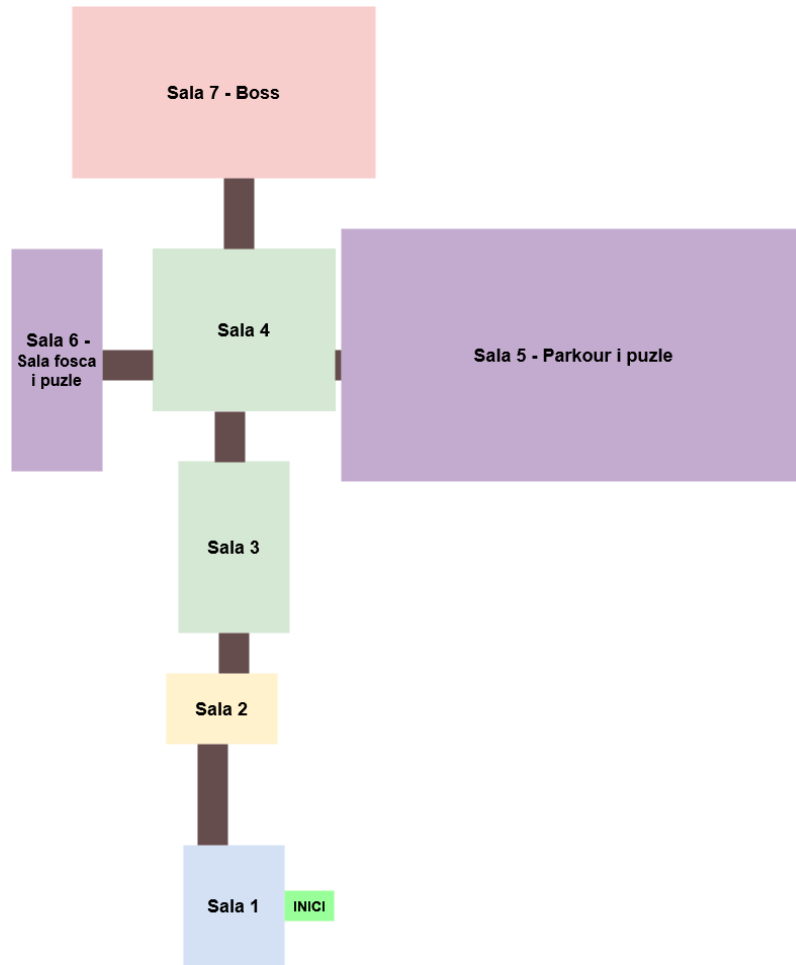


Figura 33. Diagrama del nivell complet



Figura 34. Sala inicial



Figura 35. Segona sala



Figura 36. Tercera sala

La següent sala la vàrem crear amb la idea que fos el centre del nivell, i per tant, comunicués amb totes les altres seccions, mostrat a la Figura 37. Segueix un estil similar a l'anterior amb un pont i àcid a sota però més gran. Des d'aquesta secció, es pot anar a tres sales més: una dotada amb il·luminació més fosca per aprofitar Unreal Engine i poder observar com es veuen les llums, una altra amb una part de *parkour* i finalment la sala del *boss*.

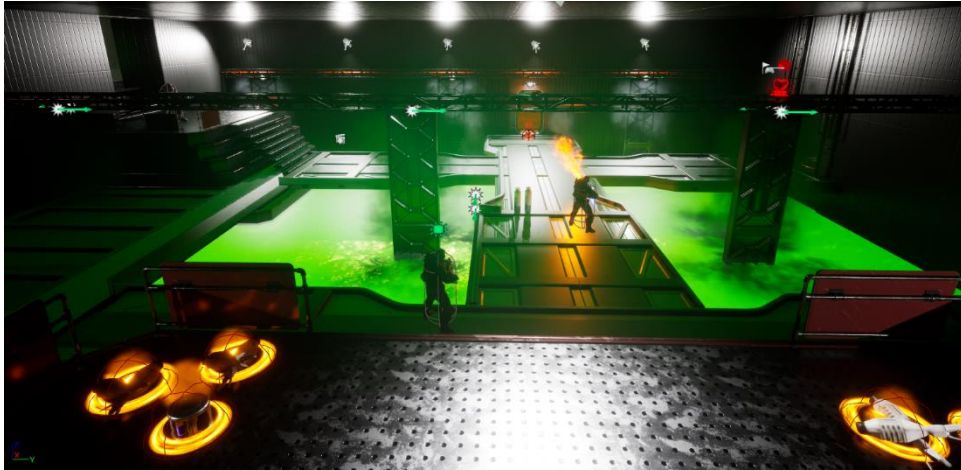


Figura 37. Sala central

Dins de la sala fosca, mostrada a la Figura 39, i la del *parkour*, mostrada a la Figura 38, hi ha uns botons que s'han d'activar per arribar a la sala del boss. Cada sala consta del seu propi repte: en la primera, la poca il·luminació dificulta la localització dels enemics, i en la segona, requereix saltar i desplaçar-se durant el combat, afegint una altra capa de complexitat.

La sala del *parkour* consta de plataformes estàtiques i en moviment, totes col·locades a una distància concreta per donar l'oportunitat al jugador d'utilitzar tant el doble salt com el *dash*. Les plataformes mòbils tenen un *blueprint* darrere que controla el desplaçament. Al final del *parkour* hi ha el botó que el jugador haurà d'activar per avançar en el nivell. El moviment de les plataformes les hem fet amb les cinemàtiques d'Unreal, que expliquem amb més detall a l'apartat de UI.

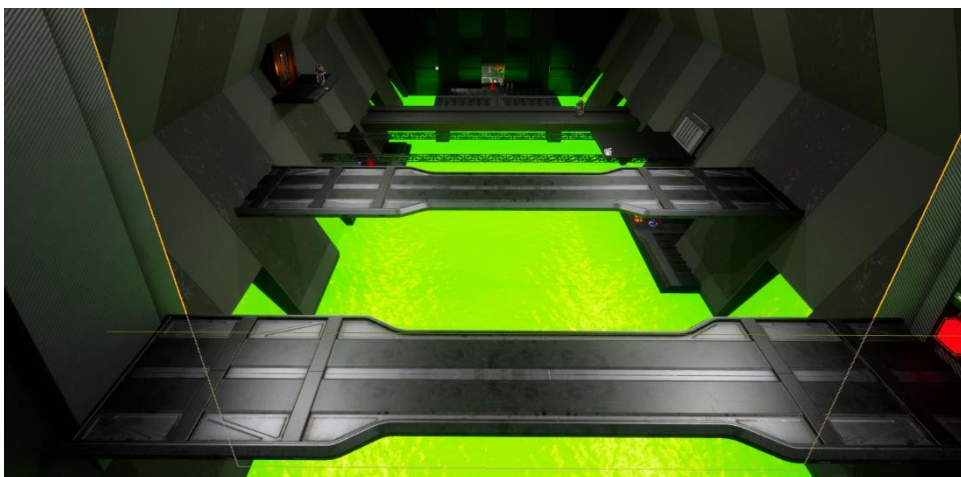


Figura 38. Parkour

Gràcies a Unreal Engine 5, la il·luminació és de molt més alt nivell. Aquesta il·luminació consumeix més recursos, però pel resultat obtingut, és un cost que podíem acceptar. En les

diferents sales del mapa, utilitzem llum de manera que el jugador no tingui dificultats per veure les coses, a excepció de la sala fosca.

En aquesta sala, s'explota la il·luminació que pot oferir Unreal, donant un toc excepcional on la llum tènue i els efectes visuals de les armes en disparar, creen un entorn fascinant.



Figura 39. Sala fosca

Aquest nivell ha comportat moltes hores de desenvolupament. Per una banda, per obtenir la qualitat gràfica que buscàvem, hem passat molt de temps posant meticulosament cada element, mur, terra, sostre, detall i en definitiva tot el que es pot veure al nivell. Per altra banda, també ha requerit molta dedicació fer un nivell que fos divertit de jugar, amb una dificultat ni massa baixa ni impossible i, sobretot, que s'entengui l'objectiu i per on ha d'anar el jugador.

Efectes visuals

Respecte als efectes visuals, els *assets* que hem utilitzat ja venien amb textures pròpies. Nosaltres ens hem enfocat en crear la sensació de fàbrica futurista: grans passadissos amb moltes superfícies metàl·liques i llums industrials, fum i, fins i tot, foc que surt de diverses canonades, tot ajudant a millorar l'experiència. Molts d'aquests efectes els hem recopilat juntament amb els *assets*. Hem fet una barreja de diferents *assets* per crear un estil únic però coherent, tant estèticament com en qualitat gràfica.

Música

Durant tot el nivell, sona la banda sonora del joc, que es va repetint en bucle. Des d'un inici, la banda sonora ha estat creada perquè es pugui reproduir en bucle sense interrupcions. L'autor de la banda sonora és el músic Pau Calero, i ha estat mesclada per l'Aleix Iglesias.

4.4.6 UI

La part visual és important, però la manera de comunicar-se amb el jugador per informar-lo ajuda a l'homogeneïtat de l'experiència, i aquesta es realitza amb la UI. En el nostre cas, no

volíem saturar ni obstaculitzar el camp visual del jugador. Per això, en molts casos, hem optat per elements que fem desaparèixer o disposen de transparències.

Unreal Engine té una opció que permet crear un *HUD*, mostrat a la Figura 40, i des del *blueprint* activar-lo i fer-lo visible. El primer que volíem mostrar era la vida, també informar de la munició disponible per cada arma i quina arma tenia activa. Finalment, mostrar les habilitats especials del *dash* i la puntada robot.

Per representar la vida, fem ús d'una *progress bar* o barra de progrés que s'activa cada vegada que la vida del jugador canvia. Mitjançant una funció, quan el jugador rep mal, s'actualitza la barra per mostrar la quantitat de vida del jugador.

Per saber quina arma estàs utilitzant actualment, es mostra una imatge que es va difuminant amb el temps per no ocupar espai a la pantalla, donant un toc molt més net i minimalista. Cada arma disposa de la seva imatge, i en canviar d'arma, el *blueprint* reconeix l'arma equipada i mostra la foto d'aquesta. A poc a poc, es va fent desaparèixer, tocant l'opacitat de l'element.

En el cas de la munició, cada vegada que es canvia d'arma s'informa de la quantitat de bales restants. Quan el jugador equipa una arma, s'activa una funció que envia el valor total de la munició al *HUD* i aquest el mostra per pantalla. S'activa també quan recull munició del terra.



Figura 40. HUD

També vàrem afegir cinemàtiques. Unreal Engine té un apartat molt intuïtiu per crear-les semblant a un editor de vídeo. Quan entres en aquest apartat, pots afegir una càmera cinemàtica, i a partir d'aquí modificar-la amb *keyframes*. També permet editar el moviment d'altres objectes.

Nosaltres vàrem optar per ensenyar el nivell en la cinemàtica inicial, com es mostra a la Figura 41. També vàrem afegir efectes de postproducció. En cada sala vàrem crear una càmera amb una petita animació per captar el moviment dels enemics i visualitzar el mapa, tot aprofitant per mostrar els botons del puzle.

Un cop teníem les diferents escenes, era qüestió d'ajuntar-ho tot. Per millorar la introducció i donar el nostre toc personal, vàrem gravar les nostres veus i les vàrem modificar utilitzant Adobe Audition perquè semblessin robòtiques. Així, teníem una escena introductòria completa on informàvem el jugador i li donàvem context.

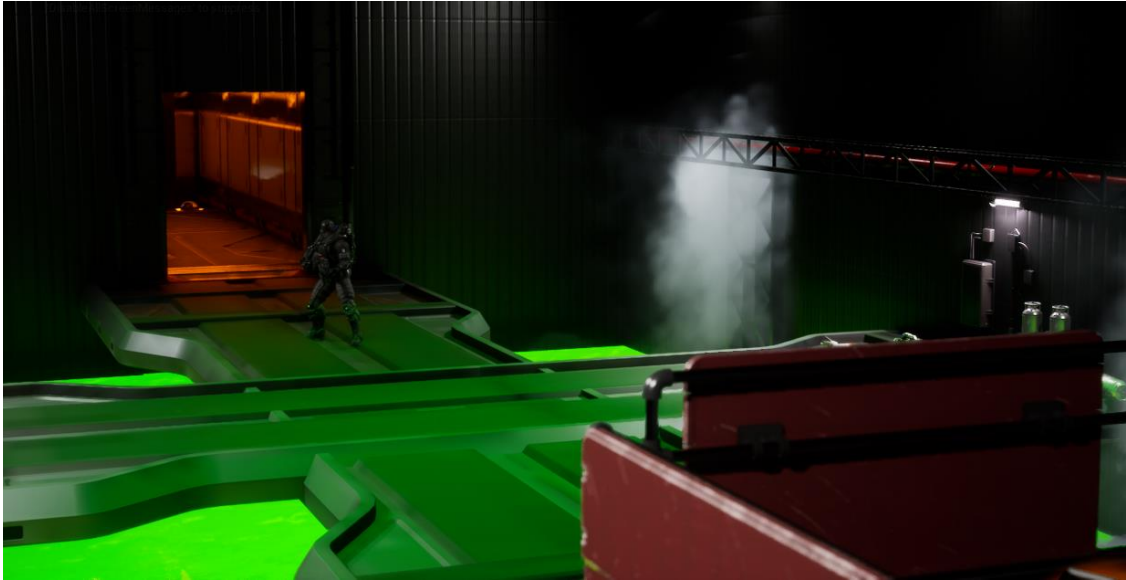


Figura 41. Cinemàtica

4.4.7 IA

Els enemics, a part de les funcions del *blueprint* on reben dany, també estan dotats amb un component d'intel·ligència artificial. Aquesta els ofereix un comportament, de manera que no només són objectes estàtics a disparar.

Per fer els enemics, vàrem voler fer ús del sistema d'AI d'Unreal. Aquest sistema consisteix en dues grans parts. La primera és un controlador, que és el component encarregat de dirigir i gestionar cada personatge enemic. Té funcions d'intel·ligència artificial per calcular la navegació pel nivell i sensors com la vista o el mal per detectar el jugador i poder atacar-lo. La segona part és el *Behaviour Tree*. Els Behaviour Trees són esquemes amb múltiples branques que permeten organitzar com ha d'actuar un enemic del joc. Dins de les branques, podem executar tasques on creem la lògica que ha de seguir l'enemic quan compleix els requisits específics, com ara si ha detectat el jugador. Aquest sistema de fer les intel·ligències artificials del joc és l'estàndard actual a la indústria i encara mai ho havíem provat.

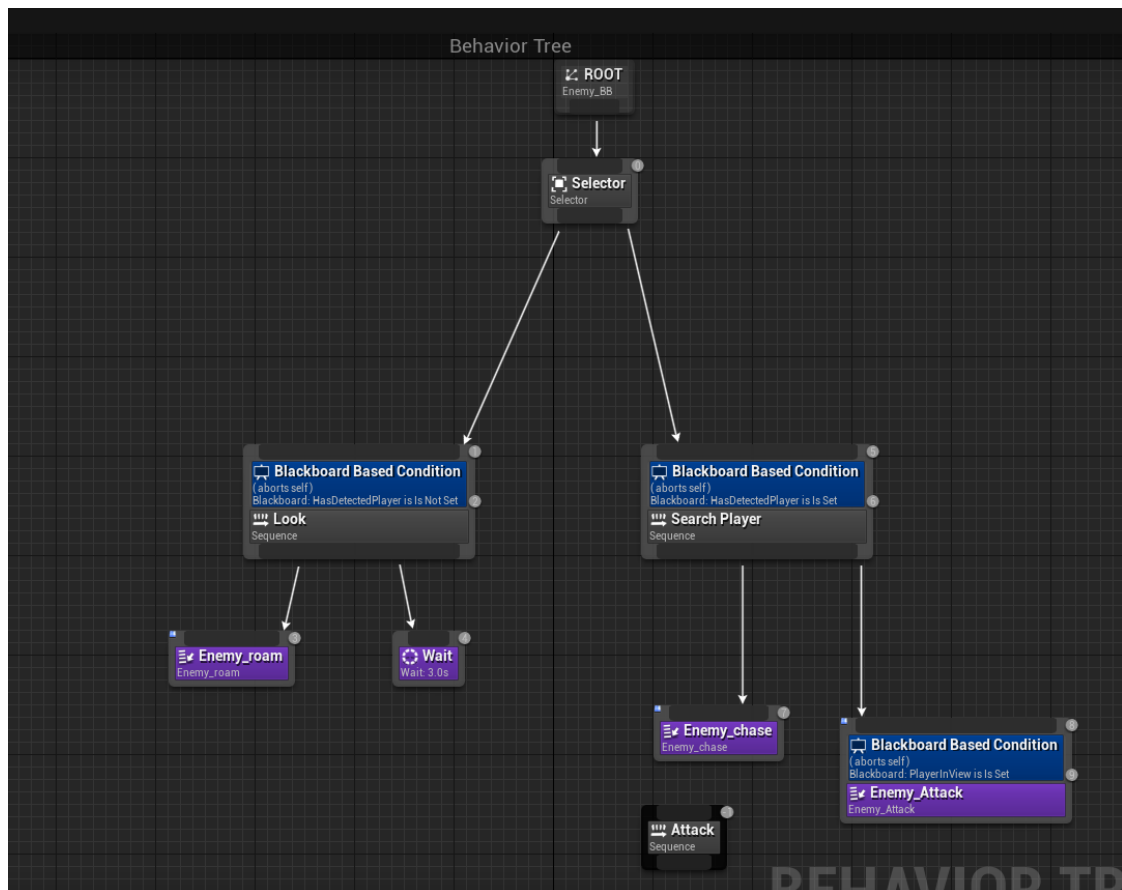


Figura 42. Behavior Tree enemig

La nostra IA té dos comportaments principals en el *Behaviour Tree*, mostrat a la Figura 42: El comportament de patrulla, que s'executa quan l'enemic no ha detectat el jugador i, per tant, va desplaçant-se imitant a una ronda de reconeixement, i el mode de combat, quan l'enemic detecta visualment al jugador. Quan s'activa aquest mode, l'objectiu es transforma en eliminar el jugador. En cas que el perdi de vista, anirà a l'última ubicació i, quan el torni a veure, tornarà a atacar-lo, sempre que estigui a una distància suficient.

Cada tipus d'enemic té un comportament diferent. En alguns casos, quan localitzin al jugador, aniran cap a ell, aproximant-se el màxim perquè la seva arma és de curta distància. En canvi, en d'altres, preferiran mantenir una distància i atacar de lluny.

Vàrem crear tres enemics diferents, cadascun amb la seva pròpia lògica tot i les seves similituds.

L'enemic base mostrat a la Figura 43, que és el més comú, simplement té el mode patrulla i el mode combat. Quan veu el jugador, el dispara amb un rifle des de la distància. Té una vida més limitada que els altres enemics.



Figura 43. Enemy base



Figura 44. Enemy llançafomes

L'altre enemic és l'enemic llançafomes, mostrat a la Figura 44. Com el seu nom indica, té un llançafomes, que crema el jugador i fa molt dany. Actua de manera semblant a l'enemic base, però s'acosta molt més al jugador perquè l'arma té un rang limitat. Disposa també de més vida, ja que si no, és molt fàcil matar-lo abans que s'acosti i pugui atacar.

Finalment, tenim el *boss*, mostrat a la Figura 45, l'enemic més difícil del joc. Vàrem voler fer-lo especial i difícil de derrotar. El *boss* et detecta directament, ja que només te'l trobes a l'última sala de cara. El seu comportament és de combat directament, però la manera d'atacar és lleugerament diferent. Té una escopeta que dispara dos cops seguits i fa molt mal, però hi ha un petit retard entre que apunta i dispara que permet esquivar els tirs si el jugador és molt hàbil. Un cop fa cinc seqüències de tir, llença un coet cap al jugador. Aquest coet és molt perillós, però pot ser esquivat amb més facilitat, i a més un jugador hàbil pot dirigir el coet cap a altres enemics. En ser el *boss* final, té moltíssima vida. Visualment, té unes dimensions més grans que els altres enemics per diferenciar-lo encara més.



Figura 45. Boss

4.5 Proves / Validació

Per obtenir *feedback* del joc, ens vàrem plantejar realitzar un *playtesting*. En el nostre cas, deixàvem un temps per provar el joc i, quan s'acabava el temps o finalitzaven el joc, els hi fèiem omplir un formulari. Durant una sessió de joc, per no influir en les respostes, els creadors no vam ajudar el jugador de cap manera.

Dins d'aquest formulari, hi consten un seguit de preguntes on donem l'opció de valorar-les de l'1 al 5, sent 1 la més baixa i 5 la més alta. En el nostre cas, volíem rebre una valoració i actuar en conseqüència. Per aquest motiu, dins del formulari, demanàvem l'opinió dels jugadors en els apartats que més ens interessaven.

Per la creació d'aquest formulari, ens vàrem basar en la informació recopilada de la pàgina web *asociacionludo* [3], però en el nostre cas, vàrem modificar els que ja existien per adaptar-los a les nostres necessitats i que preguntés pels apartats que consideràvem més interessants i rellevants.

Per facilitar que els jugadors responguessin el formulari i que a nosaltres ens fos més ràpid analitzar les respostes i organitzar-les, vàrem optar per fer un formulari en línia utilitzant forms [4] de Microsoft.

A continuació, un exemple del formulari que ha omplert cada jugador:

Valora cada aspecte del joc de l'1 al 5, sent 1 la puntuació més baixa i 5 la puntuació més alta:

Taula 1 Formulari Play Testing

Experiència de Joc:	
El joc és emocionant i entretingut?	
La jugabilitat és fluida i satisfactòria?	
L'experiència general ha estat immersiva?	
La duració del joc és apropiada?	
Gràfics i Disseny:	
Els gràfics són atractius i de qualitat?	
Els entorns i personatges estan ben dissenyats?	
L'estil visual del joc és coherent i agradable?	
So i Banda Sonora:	
La música i els efectes de so complementen bé l'acció del joc?	
La qualitat del so és bona i realista?	
La banda sonora és adequada i millora l'experiència de joc?	
Intel·ligència Artificial (IA):	
Els enemics presenten una dificultat adequada?	
L'IA és realista i respon adequadament?	
Armes i Equilibri:	
Les armes són suficientment variades i satisfactòries d'utilitzar?	
L'equilibri entre les armes és adequat?	
Has sentit que les armes s'ajustaven bé a les diferents situacions del joc?	
Comentaris i Suggeriments:	

En el nostre cas, vàrem deixar provar el joc i fer omplir el qüestionari a gent propera a nosaltres, principalment familiars i amics, amb un total de 5 persones. El perfil d'edat rondava els 20-50 anys, predominant les edats no superiors a 30. La majoria dels enquestats no eren jugadors experts, però sí que tenien nocions de videojocs i sobretot de *shooters*. Això ens va anar molt bé, ja que sabíem que respondrien contrastant amb experiències prèvies.

En l'annex 2 es mostra el resultat obtingut de les enquestes realitzades.

A partir de les sessions de prova i el qüestionari, hem realitzat alguns canvis al disseny de nivell i a la dificultat. Principalment, per millorar la navegació pel nivell, ja que alguns usuaris es perdien o no entenien la part del puzzle per continuar endavant. També hem afegit una quantitat substancial de *pickups* de vida a la part del *boss* final per fer-la més fàcil.

4.6 Cost del projecte

El cost del projecte el podem dividir en dues grans seccions: el preu material pels programes i elements utilitzats, i el temps invertit per cada un dels integrants del grup.

Cost Material

En aquest apartat mostrarem el preu dels diferents programes que hem utilitzat.

Llicències i pagaments de software i serveis:

- **Adobe:** hem utilitzat programes d'Adobe amb la llicència de la universitat, així doncs, no han tingut cap cost. En cas que n'haguéssim adquirit una, el preu seria d'uns 62,99€ al mes. Contant la duració del projecte, donaria un total de 491,04€.
- **Amazon Web Services:** la prova d'utilitzar Perforce amb AWS ens ha costat 0,09€.
- **Plastic SCM:** el programa de versions que hem utilitzat, també té una versió gratuïta limitada a 5GB. El total del projecte no ha superat aquest pes, per tant, no hem hagut de fer el pagament per ampliar-ho. Si no, el cost per GB és de 0,14€.

Hardware:

El nostre hardware utilitzat han estat dos portàtils *gamers*, un d'un preu aproximat de 1000€, i l'altre d'uns 500€. En aquest cas, no ha estat necessari adquirir-los expressament per la realització del projecte, sinó que eren els personals de cada integrant del grup.

Conjuntament amb els portàtils, s'hi afegeix un comandament de Xbox d'un valor de 20€, aquest també adquirit abans de començar el projecte.

Cost en personal

L'equip per desenvolupar aquest projecte estava format per dos programadors del mateix nivell d'experiència. Partint d'aquesta base, podem estimar un preu total que seria el sou d'aquests treballadors.

Segons un article de Jobted [5] que parla sobre el sou dels desenvolupadors de videojocs, podem crear una hipòtesi i aproximar-nos al cost real.

Cuánto Gana un Programador de Videojuegos según de la Experiencia

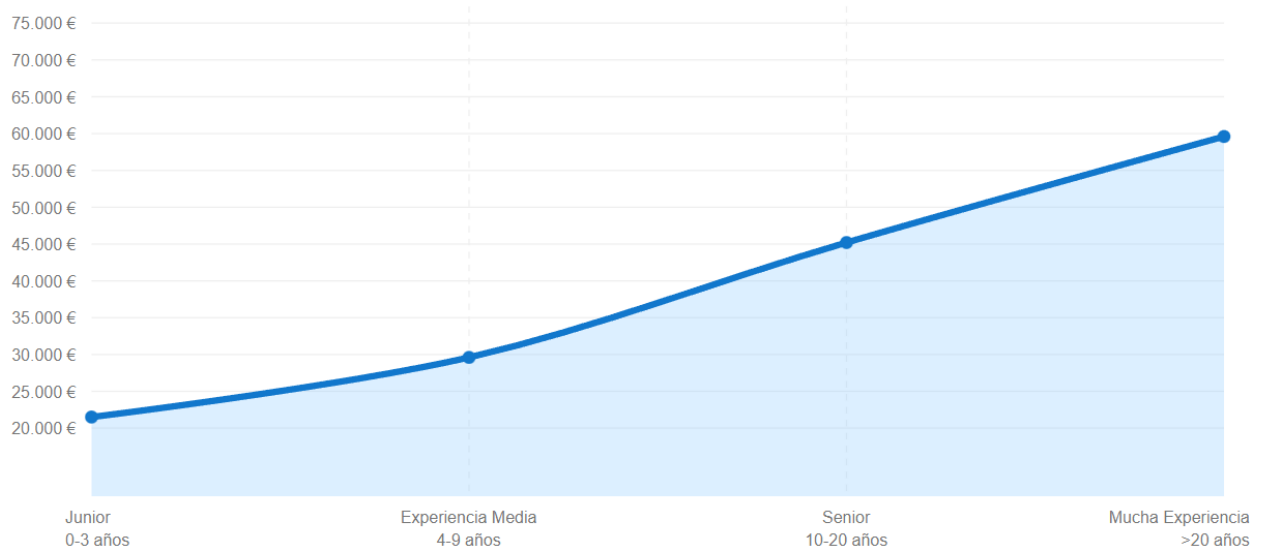


Figura 46. Sou programador de videojocs

Com que la nostra experiència en aquest sector com a professionals és nul·la, comencem des del punt més baix. Segons aquest article, i mirant el gràfic mostrat a la Figura 46, el nostre sou anual pujaria a uns 21.500€. Si ens basem en el nostre projecte, d'una duració aproximada de 300 hores per cada integrant, en total suma uns 4 mesos. Això equivaldria a uns 7.000€ per treballador.

Per tant, si ho sumem tot, contant que és un equip de dues persones, donaria un total aproximat de 14.000€ en el sou dels treballadors.

Total

Taula 2 Cost projecte

Tipus	Preu
Software <ul style="list-style-type: none"> • Adobe • AWS • Plastic SCM 	491,04€ <ul style="list-style-type: none"> • 62,99€/mes • 0,09€ • Gratis
Salari <ul style="list-style-type: none"> • Programador 	14.000€ <ul style="list-style-type: none"> • 7.000€
Hardware <ul style="list-style-type: none"> • Portàtil 1 • Portàtil 2 • Comandament Xbox 	1520€ <ul style="list-style-type: none"> • 1000€ • 500€ • 20€
Total	16.011,04€

El preu total resultat és orientatiu, ja que depèn de diversos factors, però ens ajuda a tenir una idea molt més clara de quant costaria fer un projecte d'aquestes magnituds.

5 Resultats

A continuació es mostra el resultat final obtingut amb imatges. Es poden veure els diferents exemples de les pantalles que conformen el videojoc. Adjunt al TFG, hem entregat un vídeo on es pot veure el resultat final del projecte.



Figura 47. Menú inicial

En primer lloc, tenim el menú principal, des d'on podem accedir al joc o tancar-lo (veure Figura 47).

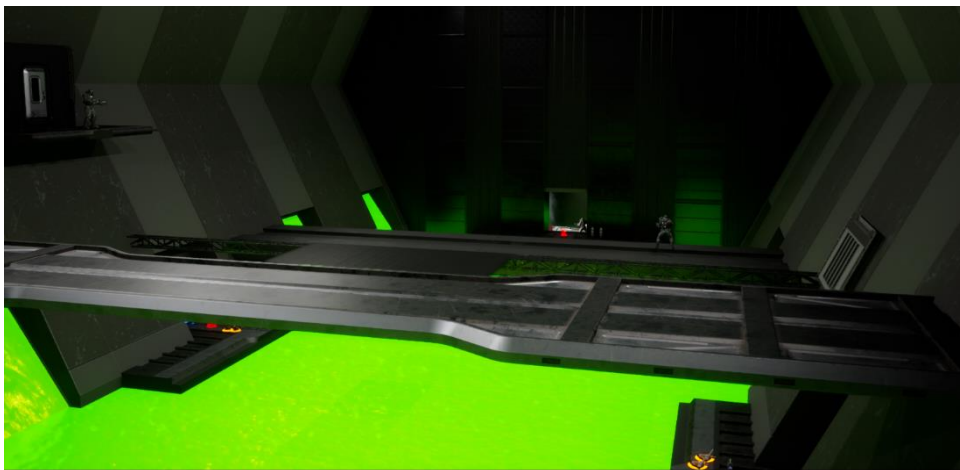


Figura 48. Cinemàtica inicial

Un cop iniciem una partida, ens trobem amb la cinemàtica inicial, que mostra diverses escenes del nivell mentre sona una narració amb veu robot, que parla de la revolució robot i els objectius del jugador (veure Figura 48).



Figura 49. Spawn del jugador

L'inici del joc. El jugador comença en aquesta petita sala amb una porta que s'obre automàticament. D'entrada, el jugador comença amb 100 punts de vida i 30 bales de munició de rifle, la primera arma. També té desbloquejades les habilitats de la puntada robot i el *dash* (veure Figura 49).

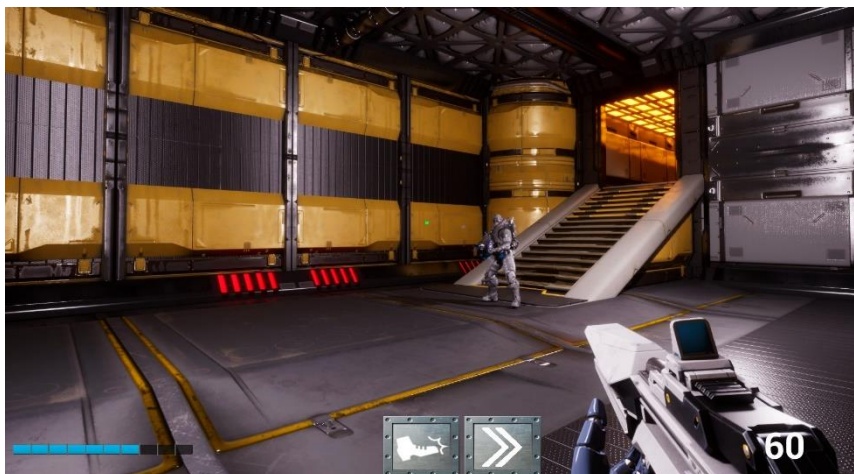


Figura 50. Primera sala

La primera sala principal del joc. Aquí podem trobar un enemic patrullant, o situat a dalt de les escales (veure Figura 50).

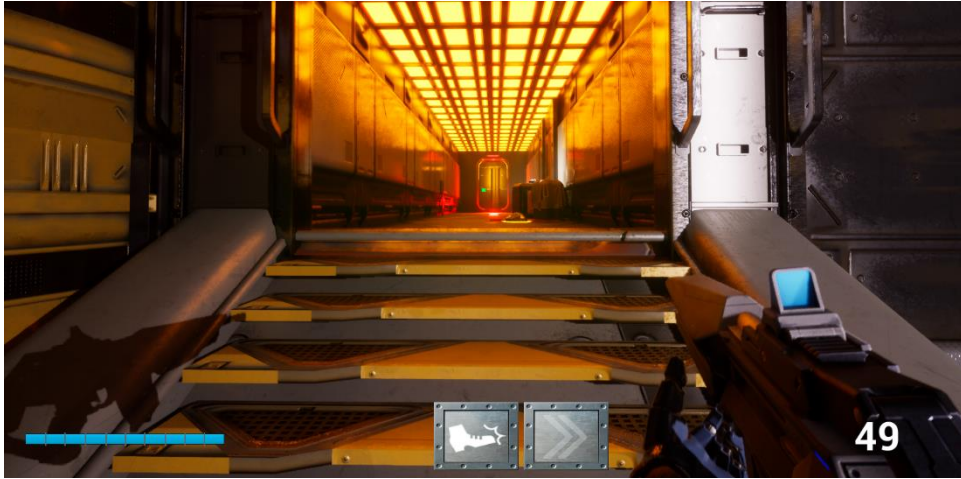


Figura 51. Passadís de la primera sala

En aquest passadís ens podem trobar un enemic i munició del rifle (veure Figura 51).



Figura 52. Segona sala

La segona sala del nivell està il·luminada per llums d'emergència i hi podem trobar una escopeta, la segona arma del joc, darrere les caixes de la dreta (veure Figura 52).



Figura 53. Tercera sala

La tercera sala conté una piscina d'àcid que és mortal pels enemics orgànics i pot danyar el nostre jugador robot. Aquí es troben múltiples *pickups* de munició, de vida, i finalment, l'arma llançacoets (veure Figura 53).



Figura 54. Sala central

La sala més central del nivell. A part de múltiples enemics i *pickups*, podem trobar aquí l'accés a la sala del *boss*, però està bloquejada i necessitem primer superar un puzzle (veure Figura 54).



Figura 55. Parkour

La sala del *parkour* està situada a la dreta de la sala central. Està dotada amb una gran piscina d'àcid i múltiples plataformes que es mouen. El jugador ha de saltar i arribar a l'última plataforma alçada per activar la primera part del puzle. La sala té múltiples enemics que disparen el jugador des de posicions avantatjoses. També trobarà aquí *pickups* amagats. Un jugador intel·ligent pot saltar directament al final del *parkour* si fa servir el llançacoets per impulsar-se (veure Figura 55).



Figura 56. Sala fosca

L'altra sala del puzle és la sala fosca. Poc il·luminada, excepte quan t'ataca un dels enemics amb llançaflames. Al final dels passadissos trobarem l'altre punt del puzle, i així es desbloqueja la sala final del *boss* (veure Figura 56).



Figura 57. Passadís del boss

Abans de la sala final del joc, trobem un passadís (veure Figura 57).



Figura 58. Sala del boss

En la part final del joc, trobem una sala més gran que combina elements de les altres anteriors. Hi trobem tota mena d'enemics, que es combinen per atacar el jugador. A la dreta podem veure un enemic més gran, que és el *boss* final. Aquesta sala està dotada de molts *pickups* per facilitar la lluita (veure Figura 58).

6 Conclusions

Aquesta memòria detalla els processos utilitzats en la creació del videojoc per PC RoboRising. Aquest projecte ha estat creat com a Treball de Final del Grau en Multimèdia. Aplicacions i Videojocs de la Universitat de Vic (UVic-UCC).

Considerem que els objectius principals que ens vàrem proposar a l'inici del projecte s'han assolit satisfactòriament. En primer lloc, tot i reduir la duració del joc per augmentar la qualitat, hem pogut crear una experiència completa, amb un clar inici i final, i amb un element de narrativa per tal d'ajuntar-ho tot. També hem pogut crear una experiència entretinguda, dinàmica i divertida pel jugador.

La creació de RoboRising ens ha aportat molts coneixements, tant per la creació de videojocs en els apartats de disseny i programació, com per la distribució de feina en un projecte real. En especial, estem molt contents per haver pogut dur a terme el projecte amb una eina com Unreal Engine 5, que havíem vist molt poc, de la qual hem après moltíssim i ens agradaria continuar utilitzant en el futur.

Després d'uns tres mesos de desenvolupament, hem aconseguit crear un videojoc per PC completament funcional i, si fos el cas, preparat per a la seva distribució com a una petita *demo*.

En un *shooter*, el que més es valora és el dinamisme del *gameplay* i, per aquest motiu, és la part que hem cuidat més. Hem procurat donar les màximes opcions al jugador per evitar la monotonia, ampliant la capacitat de moviment amb un doble salt i un *dash*, que ajuda a augmentar la velocitat del joc. També, hem creat armes amb funcionalitats diferents per poder encarar tots els reptes amb les eines adequades.

Havent finalitzat la creació del joc RoboRising, podem dir que estem extremadament orgullosos i satisfets de la feina feta. Tot i tenir el temps en contra, hem aconseguit crear un videojoc complet, dividint les tasques i ajudant-nos mútuament a millorar el nostre aprenentatge.

6.1 Limitacions

La limitació principal, i potser la més habitual en aquests projectes, ha estat el temps. En un principi, les idees que teníem eren molt més ambiciosos, reflectides en el document de disseny. Més endavant, es va canviar l'abast del projecte per adaptar-se al temps disponible i poder oferir un producte finalitzat de qualitat.

L'altra limitació important, és el fet d'utilitzar un motor de joc del qual havíem tocat poc. Tot i haver-nos documentat abans de començar el projecte, era un terreny desconegut per nosaltres. En alguns casos, tocava fer prova i error o investigar possibles solucions als problemes que anaven sorgint. Tot i que vàrem poder trobar molts tutorials, en fer el nostre propi joc, havíem d'entendre conceptes tècnics nous i adaptar-los al funcionament del projecte. A la part final del desenvolupament, ja vam aconseguir més rapidesa gràcies a l'experiència.

6.2 Treball futur

A continuació, esmentarem uns quants punts que en cas de continuar el projecte tindriem en compte per millorar, afegir o modificar.

- **Experiència de joc:** revisar i millorar l'equilibri de dificultat perquè sigui adequada i satisfactòria pels jugadors. Es podria oferir un selector de dificultat per adaptar l'experiència a jugadors de tots nivells.
- **Gràfics i estètica:** explorar la possibilitat de millorar els detalls visuals, models, efectes i les textures per millorar encara més els gràfics del joc. Fins i tot, avaluar la possibilitat de crear-ne de propis en un futur.
- **Contingut i nivells:** ampliar la varietat de contingut disponible, incloent-hi nous nivells, reptes i més elements per mantenir l'interès dels jugadors. Assegurar-se que cada nivell tingui una progressió interessant i ofereixi reptes únics per evitar una experiència repetitiva.
- **Narrativa i història:** explorar la possibilitat d'aprofundir en la narrativa i desenvolupar una història més rica i complexa. Afegir diàlegs, escenes i altres elements de narrativa que puguin donar més profunditat al món del joc.
- **Multijugador:** considerar la creació de modes de joc multijugador, sigui en línia o en mode local, per poder gaudir de l'experiència amb altres jugadors.
- **Mecànica especial:** en un principi havíem planejat crear una mecànica especial, on el jugador adquiriria habilitats de les peces de robot que obtenia durant la partida. Però s'ha hagut de modificar i reduir. Ens agradaria poder explorar aquesta idea per afegir-la en un futur.
- **Accessibilitat:** si volguéssim publicar aquest projecte, hauríem d'afegir opcions d'accessibilitat per garantir que el màxim nombre de persones pugui gaudir de l'experiència.

7 Webgrafia

[1] What's the Fornite Count in 2022?. [online]. [Accedit l'1 de maig de 2023]. Disponible a:

<<https://www.gamesradar.com/how-many-people-play-fortnite/>>

[2] Superhot passes two million lifetime sales. [online] Youtube.com. [Accedit el 8 de maig de 2023]. Disponible a:

<<https://www.gamesindustry.biz/superhot-passes-two-million-lifetime-sales>>

[3] Metodología de Playtesting profesional. [online] [Accedit el 15 de maig de 2023]. Disponible a:

<<https://asociacionludo.com/storage/guides/Tutorial%20-%20Metodolog%C3%ADa%20de%20Playtesting%20profesional.pdf>>

[4] Formulari RoboRising. [online] [Accedit el 18 de maig de 2023]. Disponible a:

<<https://forms.office.com/e/3jFq6amZPV>>

[5] ¿Cuánto cobra un programador de videojuegos?. [online] [Accedit el 26 de maig de 2023]. Disponible a:

<<https://www.jobted.es/salario/programador-videojuegos#:~:text=El%20salario%20medio%20de%20un,salario%20medio%20anual%20en%20Espa%C3%B1a>>

Annex A

Assets utilitzats en el projecte

- Animation Starter Pack <<https://www.unrealengine.com/marketplace/en-US/product/animation-starter-pack>>
- Basic Pickups VFX Set <<https://www.unrealengine.com/marketplace/en-US/product/basic-pickups-vfx-set-niagara>>
- Paragon: Murdock <<https://www.unrealengine.com/marketplace/en-US/product/paragon-murdock>>
- Realistic Starter VFX Pack Vol 2 <<https://www.unrealengine.com/marketplace/en-US/product/realistic-starter-vfx-pack-vol>>
- Sci Fi Weapons Dark <<https://www.unrealengine.com/marketplace/en-US/product/sci-fi-weapons-dark>>
- Spaceship Interior Environment Set <<https://www.unrealengine.com/marketplace/en-US/product/spaceship-interior-environment-set>>
- Modular Scifi Season 2 Starter Bundle <<https://www.unrealengine.com/marketplace/en-US/product/modular-scifi-season-2-starter-bundle>>
- Big Star Station LUMEN <<https://www.unrealengine.com/marketplace/en-US/product/big-star-station>>

Llistat de Figures - Annex B

Figura 1. Respostes de la pregunta 1 del play testing.....	iii
Figura 2. Respostes de la pregunta 2 del play testing.....	iii
Figura 3. Respostes de la pregunta 3 del play testing.....	iiiv
Figura 4. Respostes de la pregunta 4 del play testing.....	iiiv
Figura 5. Respostes de la pregunta 5 del play testing.....	v
Figura 6. Respostes de la pregunta 6 del play testing.....	v
Figura 7. Respostes de la pregunta 7 del play testing.....	vi
Figura 8. Respostes de la pregunta 8 del play testing.....	vi
Figura 9. Respostes de la pregunta 9 del play testing.....	vii
Figura 10. Respostes de la pregunta 10 del play testing.....	vii
Figura 11. Respostes de la pregunta 11 del play testing.....	viii
Figura 12. Respostes de la pregunta 12 del play testing.....	viii
Figura 13. Respostes de la pregunta 13 del play testing.....	iiix
Figura 14. Respostes de la pregunta 14 del play testing.....	iiix
Figura 15. Respostes de la pregunta 15 del play testing.....	x

Annex B

Fotos dels resultats obtinguts de les enquestes

1. El joc és emocionant i entretingut? (0 punt)

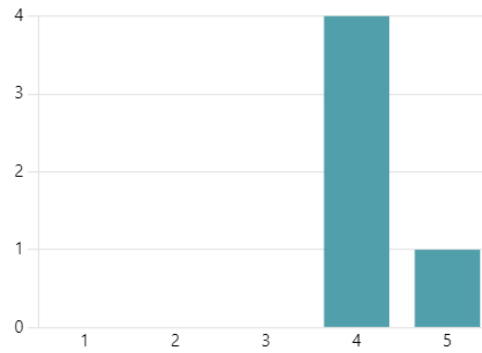
[Més dades](#)4.20
Valoració mitjana

Figura 1. Respostes de la pregunta 1 del play testing

2. La jugabilitat és fluida i satisfactòria? (0 punt)

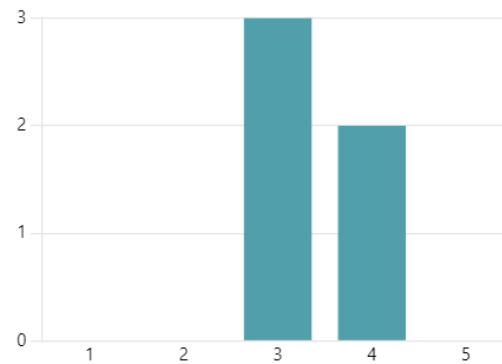
[Més dades](#)3.40
Valoració mitjana

Figura 2. Respostes de la pregunta 2 del play testing

3. L'experiència general ha estat immersiva? (0 punt)

[Més dades](#)

4.00
Valoració mitjana

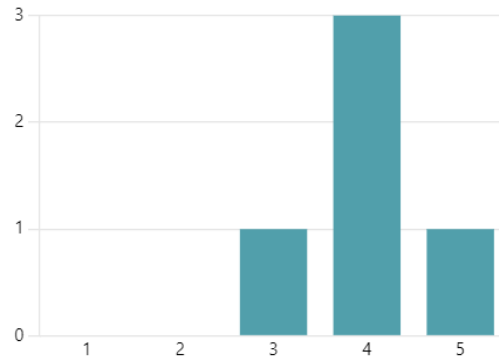


Figura 3. Respostes de la pregunta 3 del play testing

4. La duració del joc és apropiada? (0 punt)

[Més dades](#)

3.00
Valoració mitjana

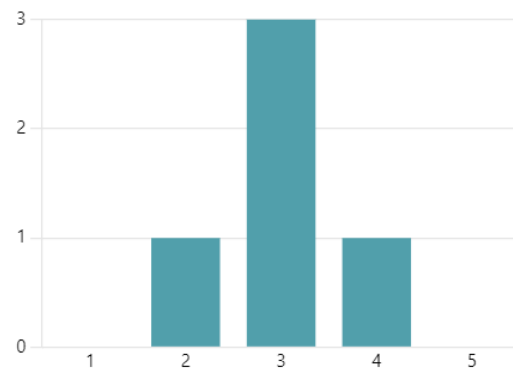


Figura 4. Respostes de la pregunta 4 del play testing

5. Els gràfics són atractius i de qualitat? (0 punt)

[Més dades](#)

4.60
Valoració mitjana

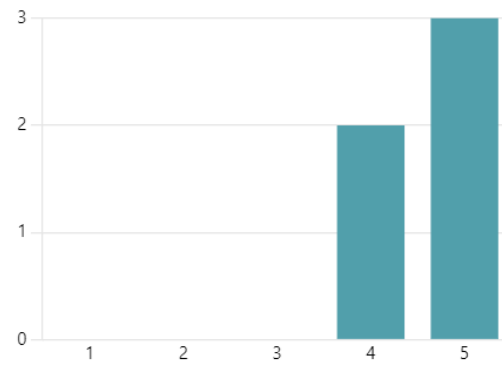


Figura 5. Respostes de la pregunta 5 del play testing

6. Els entorns i personatges estan ben dissenyats? (0 punt)

[Més dades](#)

4.40
Valoració mitjana

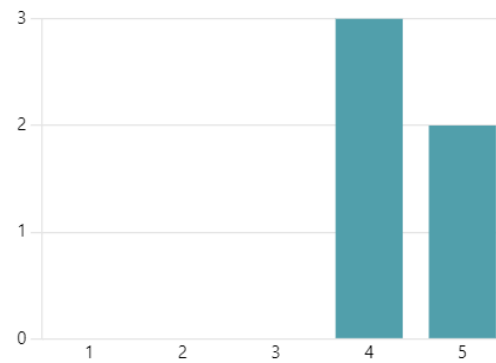


Figura 6. Respostes de la pregunta 6 del play testing

7. L'estil visual del joc és coherent i agradable? (0 punt)

[Més dades](#)

4.80
Valoració mitjana

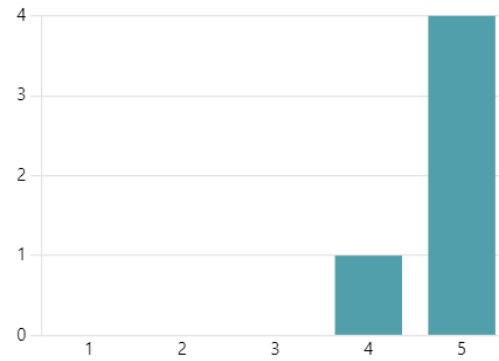


Figura 7. Respostes de la pregunta 7 del play testing

8. La música i els efectes de so complementen bé l'acció del joc? (0 punt)

[Més dades](#)

4.40
Valoració mitjana

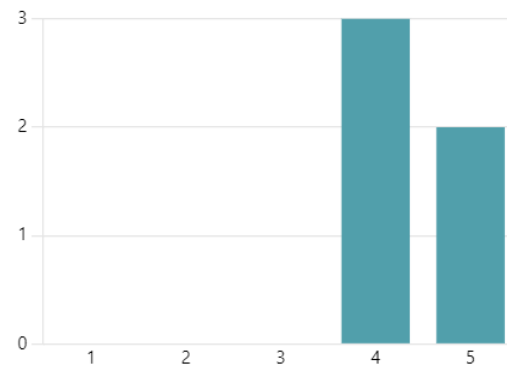


Figura 8. Respostes de la pregunta 8 del play testing

9. La qualitat del so és bona i realista? (0 punt)

[Més dades](#)

4.60
Valoració mitjana

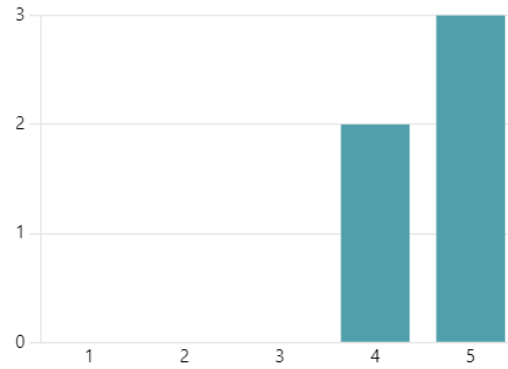


Figura 9. Respostes de la pregunta 9 del play testing

10. La banda sonora és adequada i millora l'experiència de joc? (0 punt)

[Més dades](#)

4.60
Valoració mitjana

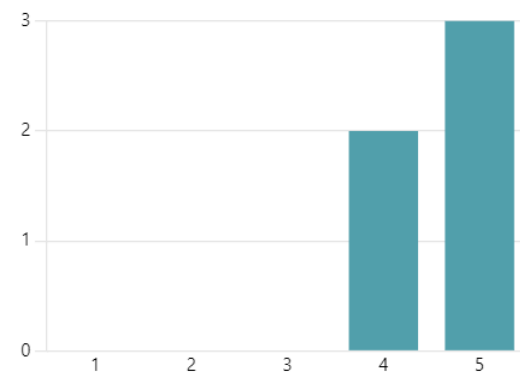


Figura 10. Respostes de la pregunta 10 del play testing

11. Els enemics presenten una dificultat adequada? (0 punt)

[Més dades](#)

2.80
Valoració mitjana

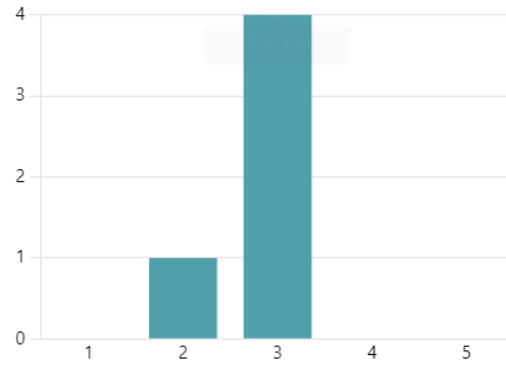


Figura 11. Respostes de la pregunta 11 del play testing

12. L'IA és realista i respon adequadament? (0 punt)

[Més dades](#)

3.40
Valoració mitjana

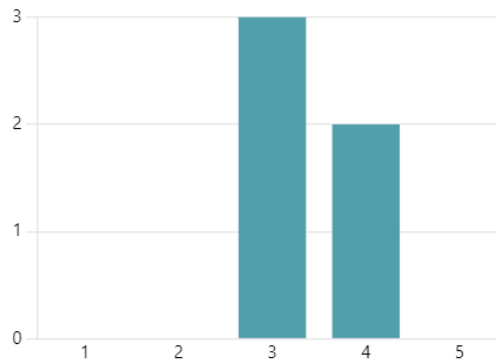


Figura 12. Respostes de la pregunta 12 del play testing

13. Les armes són suficientment variades i satisfactòries d'utilitzar? (0 punt)

[Més dades](#)

3.80
Valoració mitjana

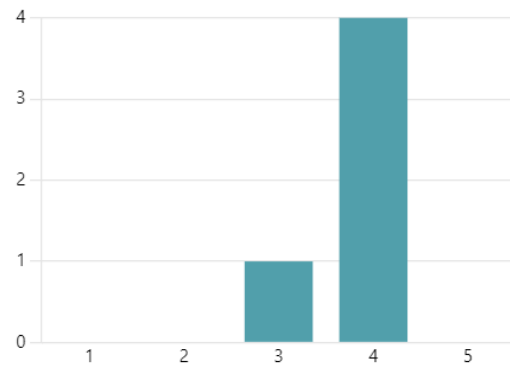


Figura 13. Respostes de la pregunta 13 del play testing

14. L'equilibri entre les armes és adequat? (0 punt)

[Més dades](#)

4.20
Valoració mitjana

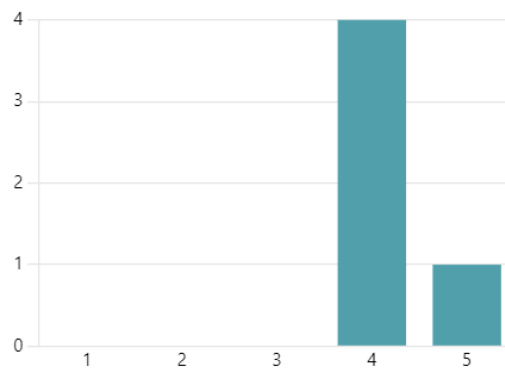


Figura 14. Respostes de la pregunta 14 del play testing

15. Has sentirt que les armes s'ajustaven bé a les diferents situacions del joc? (0 punt)

[Més dades](#)

3.60
Valoració mitjana

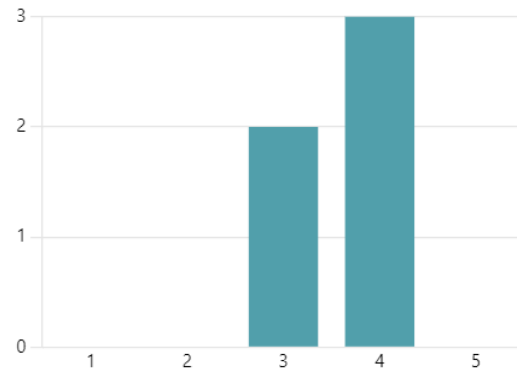


Figura 15. Respostes de la pregunta 15 del play testing