

Future Punk

Miquel Sanuy Trullàs
Grau en Enginyeria Informàtica

Joel Servitja Feu
Joan Arnedo Moreno

06/06/2021



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

Llicències alternatives (triar alguna de les següents i substituir la de la pàgina anterior)

A) Creative Commons:



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-SenseObraDerivada 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-CompartirIgual 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement 3.0 Espanya de Creative Commons](#)

B) GNU Free Documentation License (GNU FDL)

Copyright © ANY EL-TEU-NOM.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free

Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (l'autor/a)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Future Punk</i>
Nom de l'autor:	<i>Miquel Sanuy Trullàs</i>
Nom del consultor/a:	<i>Joel Servitja Feu</i>
Nom del PRA:	<i>Joan Arnedo Moreno</i>
Data de lliurament (mm/aaaa):	<i>06/2021</i>
Titulació o programa:	<i>Grau en Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Videojocs</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Videojoc, Unity, FPS</i>
<p>Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i></p>	
<p>El treball explica el desenvolupament i l'estat del projecte de videojoc anomenat Future Punk. Se'n detalla l'arquitectura i el disseny de nivells, així com les mecàniques generals i el disseny de les armes i els enemics.</p> <p>El joc en si és un First Person Shooter o FPS d'estètica futurista, amb un únic nivell ambientat en un entorn industrial. L'objectiu d'aquest és derrotar al Boss, mentre que els enemics intenten derrotar al jugador.</p> <p>També es tracta el projecte. S'explica la seva metodologia i com ha funcionat i es detalla la planificació inicial i els canvis que ha patit.</p> <p>Finalment es desenvolupen un seguit de conclusions del treball i del projecte.</p>	
<p>Abstract (in English, 250 words or less):</p>	
<p>The work explains the development and the state of the video game called Future Punk. Its architecture and level design are detailed as well as general mechanics and enemies and weapons design.</p> <p>The game is a First Person Shooter or FPS with a futuristic aesthetic and a single level set in an industrial environment. The objective of said level is defeating the Boss while enemies try to defeat the player.</p> <p>The work also explains the project. Methodology and progress are discussed while initial planning and the changes it has undergone are explained. In the end there are drawn a set of conclusions to the work and the project itself.</p>	

Índex

1. Introducció.....	1
1.1. Context i justificació del Treball	1
1.2. Objectius del Treball.....	1
1.3. Enfocament i mètode seguit.....	2
1.4. Planificació del Treball	2
1.5. Breu sumari de productes obtinguts.....	3
1.6. Breu descripció dels altres capítols de la memòria.....	3
2. Estat de l'art.....	4
2.1. Unity.....	4
2.2. El gènere dels FPS.....	4
2.3. Referents estètics.....	5
3. Definició del joc.....	6
3.1. Objectius i mecàniques.....	6
3.2 Armes.....	6
3.3 Enemics.....	7
3.3 Ambientació i història.....	8
4. Procés de desenvolupament.....	9
4.1. Plantejament inicial.....	9
4.2 Ús del tutorial FPS Microgame.....	9
4.3 Desenvolupament del projecte.....	10
5. Disseny tècnic.....	11
5.1. Arquitectura del projecte.....	11
5.2. Disseny del nivell.....	18
5.3. Assets utilitzats.....	19
6. Manual d'usuari.....	21
6.1. Manual del joc.....	21
6.2. Requisits tècnics.....	21
7. Conclusions.....	22
8. Glossari.....	24
9. Bibliografia.....	25

Llista de figures

Figura 1: Calendari de la planificació inicial.....	2
Figura 2: Captura de Doom(2016).....	4
Figura 3: Fotograma de Ghost in the shell (1995).....	5
Figura 4: Pistola.....	6
Figura 5: Escopeta.....	6
Figura 6: Rifle.....	7
Figura 7: Soldier.....	7
Figura 8: Drone.....	7
Figura 9: Paladin.....	7
Figura 10: Boss.....	8
Figura 11: Vista del nivell.....	18
Figura 12: Executable del joc.....	21

1. Introducció

1.1. Context i justificació del Treball

Els videojocs són un mitjà que fa anys que està en constant creixement. Aquest s'ha accentuat encara més amb la pandèmia, arribant a moure quantitats de diners més grans que la indústria cinematogràfica [1].

En els seus inicis els videojocs eren principalment això: jocs. Normalment creats per companyies especialitzades dedicant-hi gran quantitat de temps i recursos. Amb el temps i gràcies a la innovació, s'han diversificat en molts aspectes.

Avui en dia trobem jocs de tota mena, que semblen pel·lícules o que es juguen com jocs de taula. Els gèneres s'han expandit i molts desenvolupadors han creat projectes experimentals.

Un dels avenços més importants que justifica i permet que aquest projecte sigui possible és l'escena dels videojocs "indie". La definició estricta a vegades ha esdevingut un debat, però el consens general és que són jocs fets per equips reduïts amb no molts recursos. Avui en dia no és necessari ser una gran empresa per desenvolupar videojocs, en part degut a l'existència de motors de tercers i de plataformes on qualsevol pot publicar el seu producte.

1.2. Objectius del Treball

Els principals objectius del treball han sigut els següents:

-Aprentatge de la plataforma de Unity: utilitzant els recursos oficials o qualsevol altra eina a l'abast. Assolir el coneixement i les habilitats necessàries per poder desenvolupar el projecte plantejat.

-Planificació del projecte: traçar un pla factible i sensible al context i les necessitats. Saber detectar quan cal fer modificacions i quines també és important.

-Desenvolupament del producte: el joc a presentar. Sent prioritària la seva funcionalitat. La idea és un producte senzill però complet.

1.3. Enfocament i mètode seguit

Per desenvolupar el projecte he adoptat una metodologia àgil. La funció d'aquesta ha sigut permetre reaccionar i fer canvis de ser necessari, valorant que el meu coneixement a l'inici era molt limitat i confiant en el feedback i en la perspectiva un cop assolís més coneixements.

El desenvolupament ha sigut incremental i això m'ha permès anar afegint o modificant elements del projecte a mesura que aprenia el seu funcionament. A més, permetia una revisió constant de l'estat general del projecte.

Tot i això, no he utilitzat cap mètode formalitzat, ja que la majoria incorporen molts elements destinats a la comunicació i la coordinació entre persones i equips. En el meu cas, sent una sola persona, no tenia gaire sentit fer reunions o generar molta documentació. De totes maneres sí que he creat alguns documents per mantenir registrat el progrés del projecte i poder seguir el calendari de l'assignatura.

Així, he organitzat la feina en setmanes. Cada setmana he plantejat un seguit d'objectius que tenien en compte el progrés necessari respecte al plantejament i les entregues de les PAC. Al final de cada setmana valorava la feina feta i encarava la següent, tenint en compte les tasques inacabades si n'hi havia.

1.4. Planificació del Treball

Al principi del treball vaig plantejar un calendari que estructurava la feina a fer a partir de les diferents PACs i de l'entrega final. A més, també incloïa els principals tutorials de Unity.

Calendari																	
Mes	març					abril				maig					juny		
Setmana	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21
Activitat	PAC 2					PAC 3				PAC Final					Defensa		
Aprentatge Unity	Tutorial "Junior Programmer"					Aprentatge complementari*											
			Tutorial: FPS microgame														
Desenvolupament projecte	-	Prototip bàsic				Desenvolupament Principal				Part final			Testeig i correccions		Preparació de la defensa		

Figura 1: Calendari de la planificació inicial

Els objectius de les diferents PAC eren els següents:

- PAC 2: mecàniques principals i nivell bàsic:
- PAC 3: la resta de característiques i l'apartat artístic.
- Entrega final: correcció, feina pendent i noves característiques en cas de tindre temps.

També vaig plantejar les característiques i objectius del producte final. Algunes d'aquestes han canviat durant el transcurs del projecte per diversos motius: recomanacions, falta de temps, imprevistos, noves idees, etc.

1.5. Breu sumari de productes obtinguts

El principal producte és el prototip del joc: Future Punk. Es tracta d'una versió senzilla amb un sol nivell, i per això es consideraria més una demo que un joc complet en cas de sortir al mercat. Tot i això, en el nivell s'inclouen tots els elements generats.

A part del joc i la memòria, s'han generat altres elements per a l'entrega:

-El tràiler del joc: <https://youtu.be/y3ajhpa4zec>

-El vídeo de la presentació final: <https://youtu.be/CCgfQTIP2W4>

Es pot accedir al Drive amb tots els enllaços i arxius d'interès amb el següent enllaç:

<https://drive.google.com/drive/folders/1Dc9ZNRymKz-0cKloA7EYlpBcp9mDnN8H?usp=sharing>

1.6. Breu descripció dels altres capítols de la memòria

2. Estat de l'art: explica l'estat actual de diversos afers relacionats amb el projecte més enllà d'aquest. Serveix per donar context al treball.

3. Definició del joc: explica, a grans trets, quines són les característiques del joc sense entrar en com estan implementades o en cap altre aspecte tècnic.

4. Procés de desenvolupament: explica com ha sigut aquest, així com les decisions que s'han pres.

5. Disseny tècnic: explica les característiques més tècniques del producte, especialment la seva arquitectura.

6. Manual d'usuari: ensenya el funcionament del producte i mostra les característiques de l'entorn amb què s'ha provat com a referència.

7. Conclusions: explica les conclusions de l'aprenentatge realitzat, tant del producte mateix com del treball.

2. Estat de l'art

2.1. Unity

Unity és un motor de videojocs amb més de quinze anys que segueix sent actualitzat avui en dia. Compta amb una plataforma de desenvolupament que permet desenvolupar productes per a diferents plataformes objectiu.

És un dels motors més utilitzats a causa dels diferents avantatges [2] que ofereix, principalment el fet de ser gratuït (almenys fins a obtenir ingressos considerables), el suport a diferents plataformes, la facilitat per desenvolupar en 2D, 3D o una combinació de les dues o la gran quantitat de recursos per al seu aprenentatge.

En el meu cas el que més ha determinat l'elecció és la facilitat per aprendre'n l'ús. La mateixa empresa ofereix tutorials gratuïts de qualitat, la documentació es troba sempre accessible i actualitzada i el llenguatge de programació que utilitzen els scripts és C#, que ja havia utilitzat anteriorment.

2.2. El gènere dels FPS

Els first person shooters o FPS són un dels gèneres més antics i, a la vegada, més exitosos al llarg del temps [3]. La definició del gènere només dicta que han de ser en primera persona i el jugador ha de ser capaç de disparar.

Durant molts anys la majoria de títols han sigut similars i es podria dir que els FPS s'han estancat en diverses ocasions, però també s'han vist etapes de molta innovació amb títols més experimentals.

Així, avui en dia trobem diversos estils i subgèneres dins els FPS [4]. Tenim jocs similars als antics arcade com el Doom de 2016: frenètic, violent i extremadament entretingut. També hi ha la saga Call of Duty, que ha sigut l'estàndard durant els últims anys, o títols experimentals com Superhot.



Figura 2: Captura de Doom(2016)

Actualment els multijugador dominen el gènere i no és estrany veure jocs que n'inclouen perquè es dona per suposat que ho faran. Altres títols estan basats directament a jugar en línia, com Counter Strike o Destiny. Finalment trobem Overwatch, que és una celebració dels FPS que agafa algunes mecàniques més típiques d'altres gèneres de videojocs.

En el cas que ens ocupa les principals referències són alguns shooters moderns amb elements similars a d'altres més antics. Principalment hi ha dos títols que destacaria: Doom (2016) i Devil Daggers.

El primer es tracta d'una mena de reboot de la franquícia amb la tecnologia del moment. El resultat és un joc ràpid i entretingut, que en les dificultats més altes força al jugador a canviar d'arma en cada situació i no aturar-se en cap moment.

En el cas de Devils Daggers trobem un joc més simple, però més difícil, on una gran quantitat d'enemics que omplen la pantalla i assetgen constantment al jugador, que lluita per allargar tant com sigui possible la partida.

2.3. Referents estètics

L'estètica del joc té uns referents bastant diferents dels de la jugabilitat. En aquest cas la principal referència és el concepte del cyberpunk, que dona nom al joc.

El cyberpunk [5] no és només una estètica. De fet, es tracta d'un subgènere de la ciència-ficció, tot i que hi ha qui ho defineix com una cultura. Una frase sovint utilitzada per resumir el concepte és "High-tech. Low life", ja que aquest gènere tracta sobre distopies futuristes on el progrés científic i tecnològic no comporta una millora de les condicions de vida de les persones, més aviat el contrari.

Estèticament, trobem referents en tota mena de mitjans, però en qualsevol cas ens interessa de cara a crear un joc, i per això ens fixarem sobretot en altres jocs i en el cinema.

Un dels exemples recents més interessants és Cyberpunk 2077, un joc de rol ambientat en una ciutat colossal envoltada per entorns industrials i suburbis i, més enllà, el desert.

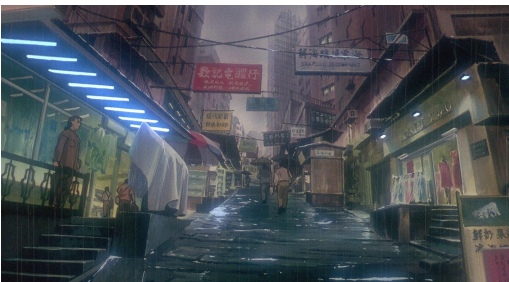


Figura 3: Fotograma de Ghost in the shell (1995)

En el cinema destaca Blade Runner, que va ser una de les pel·lícules que va popularitzar el gènere.

Finalment cal mencionar Ghost in the Shell, una franquícia de manga adaptada tant al cinema com en format de sèrie.

3. Definició del joc

3.1. Objectius i mecàniques

El joc s'anomena Future Punk i és un shooter on el jugador s'enfronta a diferents tipus d'enemic. L'objectiu principal del nivell implementat és derrotar al Boss, però hi ha diversos obstacles que bloquegen el camí.

Per derrotar als enemics el jugador té diferents armes, però aquestes no són l'única manera de progressar. Es pot utilitzar l'escenari per a guanyar avantatge sobre els enemics o inclús per evitar lluitar amb alguns d'ells.

El jugador també pot ser derrotat quan la seva vida cau a 0. Per evitar-ho sovint necessitarà curar-se i ho podrà fer "robant" la vida als enemics, és a dir, atacant-los. Aquesta característica i la mobilitat superior permeten al jugador enfrontar-se als enemics en inferioritat numèrica.

El jugador té 500 punts de vida i recupera el 35% del mal que fa en atacar als soldats i al Boss. És el personatge més ràpid de tots amb una velocitat màxima de 15.

3.2 Armes

El jugador compta amb tres armes i cadascuna utilitza un tipus de munició diferent:



Figura 4: Pistola

-Pistola: arma simple amb poca capacitat de bales i que fa poc mal. Té poc temps de recàrrega i la munició que utilitza és abundant. Resulta útil com a arma de propòsit general quan hi ha pocs enemics o com a arma secundària quan caldria recarregar la resta.

Arma	Dany	Cadència	Capacitat	Temps de recàrrega
Pistola	35	3.33	11 bales	1s



Figura 5: Escopeta

-Escopeta: arma devastadora a curta distància que perd eficàcia de lluny degut a la dispersió dels projectils que dispara. La munició és escassa. La seva utilitat és en els espais tancats, contra enemics resistents o contra grups d'enemics molt junts, però en qualsevol dels casos a curta distància.

Arma	Dany	Cadència	Capacitat	Temps de recàrrega
Escopeta	12x18*	2	8 cartutxos	3.4s

*Dispara 12 projectils que fan 18 punts de dany



Figura 6: Rifle

-Rifle: arma de propòsit general amb alta cadència i capacitat. Preferible a llargues distàncies. És útil en la majoria de situacions, però només destaca quan hi ha molts enemics.

Arma	Dany	Cadència	Capacitat	Temps de recàrrega
Rifle	40	11.11	30 bales	2.2s

3.3 Enemics

El joc compta amb tres tipus d'enemic bàsic i el Boss final:



Figura 7: Soldier

-Soldier: l'enemic més abundant. Utilitza un rifle similar al del jugador i esdevé perillós gràcies al seu rang d'atac, especialment en grups.

Compta amb 200 punts de vida, però es duplica el mal si es dispara al cap. Ataca en ràfegues de 5 trets que fan 12 punts de mal cadascun.



Figura 8: Drone

-Drone: un enemic més dèbil que ataca a distàncies curtes. No és gaire perillós, però fàcil de subestimar, ja que no permet que el jugador es curi en atacar-lo.

Compta amb 100 punts de vida i ataca al jugador cada 0.8 s amb projectils que fan 9 punts de mal.

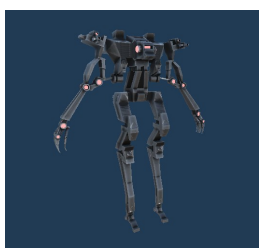


Figura 9: Paladin

-Paladin: un enemic resistent i perillós que pot eliminar fàcilment al jugador. Tot i això, és fàcil esquivar els seus atacs.

Compta amb 900 punts de vida i ataca al jugador constantment amb dos làsers idèntics que fan 35 punts de dany en impactar.

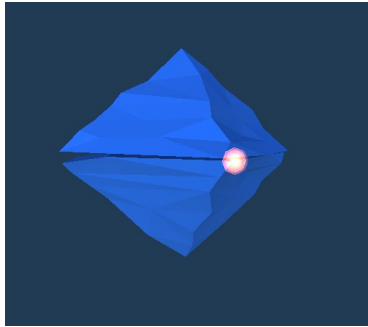


Figura 10: Boss

-Boss: l'enemic final del nivell. Compta amb dos tipus d'atac: un fàcil d'esquivar, però difícil cobrir-se i un que és just el contrari.

Té 2500 punts de vida i només rep el 70% del mal (dificultant al jugador curar-se). Els projectils de l'arma principal fan 40 punts de mal i són ràpids. Els de l'arma alternativa fan 75 punts en una àrea de 7 m (reduint el mal proporcionalment a la distància de l'impacte).

3.3 Ambientació i història

El joc utilitza una ambientació "cyberpunk" en un futur relativament proper. Armes, enemics i edificis reflecteixen això.

Trobem armes convencionals però també làser. Els edificis utilitzen una arquitectura bastant similar a l'actual, però contenen elements, com les portes automàtiques, més aviat futuristes.

La resta d'elements intenten reforçar l'aspecte futurista del joc: molts dels objectes, la interfície i la il·luminació.

Pel que fa a la història, és senzilla i no té molt pes, sinó que serveix únicament per situar al jugador en el context del joc. El personatge protagonista és una víctima d'experiments poc ètics d'una gran corporació i busca venjança contra aquesta. Per aconseguir-ho, s'infiltra en un dels seus complexos per destruir un dels experiments més valuosos.

4. Procés de desenvolupament

4.1. Plantejament inicial

Des d'un primer moment vaig tenir clar que volia desenvolupar el projecte de manera incremental i que una de les prioritats havia de ser l'aprenentatge de l'entorn, a causa de la meua inexperiència.

No tenia clares moltes de les característiques, especialment l'estètica o l'ambientació i algunes de les que creia que tenia clares han canviat amb el temps.

Tot i això sí que sabia quines eren les prioritats: la senzillesa, la varietat tant d'eines com d'obstacles i la cerca d'un combat satisfactori per al jugador.

Al principi em vaig centrar a fer els tutorials oficials i absorbir tant coneixement com em fos possible, a la vegada que experimentava amb el motor. També vaig voler estructurar el desenvolupament i portar un registre més o menys documentat del progrés.

4.2 Ús del tutorial FPS Microgame

Una de les tasques que havia plantejat era fer un dels tutorials oficials de Unity anomenat FPS microgame. Aquest comptava amb alguns elements bàsics: moviment i armes del jugador, alguns enemics, etc.

Finalment vaig acabar utilitzant els recursos d'aquest tutorial com a base per al meu projecte. Això va aportar tant avantatges com entrebancs, però el motiu principal de l'elecció era poder proporcionar una primera versió per a l'entrega de la PAC 2. També em permetia dedicar més temps als tutorials i menys a programar les funcionalitats més bàsiques com per exemple el moviment del jugador.

Tot i això, també m'ha dificultat la feina en algunes ocasions. Principalment a l'hora d'afegir contingut, ja que molts dels elements del tutorial que aparenten ser genèrics i escalables no estan pensats per ser modificats o funcionar en combinació amb nous elements.

Això m'ha portat a modificar molt codi i descartar molts elements que considerava útils, però de totes maneres els elements del tutorial m'han permès reutilitzar codi i, sobretot, comptar amb bons exemples de funcionament de diversos elements del joc.

4.3 Desenvolupament del projecte

Al principi del projecte la meua experiència en el desenvolupament de videojocs era nul·la. Coneixia la indústria com a aficionat i tenia coneixements de programació, però la resta m'era bastant aliè. Per això vaig centrar les dues primeres setmanes en l'aprenentatge de Unity.

Durant les dues setmanes següents vaig treballar en el projecte a la vegada que seguia amb els tutorials, però vaig veure que afegir elements requeria bastant de temps. També vaig cometre l'error de no tindre prou en compte la resta d'elements de l'entrega de la PAC 2 i, tot i acabar-los, això em va deixar menys temps per al projecte.

Aquí vaig fer un replantejament amb el feedback de la PAC 2 i el que havia après en aquest mes. Vaig canviar diversos elements de la planificació i em vaig centrar en els nous elements (sistema de munició i enemics). Una de les limitacions era que no tenia el temps i les habilitats per a crear molts dels assets necessaris per a un joc d'aquest tipus, així que hauria de buscar tot el que em servís a la botiga de Unity.

Després d'això em vaig centrar en el disseny del nivell i també en configurar la interfície. Com que volia enemics senzills calia que el nivell fos més interessant. També va ser aquí quan vaig introduir el Boss, primer com a idea i finalment dins del joc.

De cara a la PAC 3 vaig planificar millor el document i el vídeo de l'entrega i no em va faltar tant temps. A més, contenia un nivell funcional (tot i que calia refinar-lo).

Durant les últimes setmanes la prioritat ha sigut la correcció i millora del joc, sobretot en l'apartat estètic.

També he fet proves enviant una copia del joc a diversos amics per testejar el nivell. Els demanava que em notifiquessin, sobretot, qualsevol error que trobessin i, en segon lloc, idees sobre com millorar el joc o coses que no els agradessin.

5. Disseny tècnic

5.1. Arquitectura del projecte

El codi del projecte s'estructura sobre el que hi havia del tutorial del FPS microgame. Si bé la major part de classes han sigut modificades, l'estructura subjacent és pràcticament la mateixa. També hi ha parts del codi que no s'utilitzen i que, per tant, no es mencionen. Aquestes s'han mantingut com a exemple o perquè existia l'opció d'utilitzar-les més endavant.

Així, només menciono les modificacions importants (les que canvien significativament el comportament del codi) i no les que simplement permeten adaptar les noves classes al conjunt.

A continuació separo el codi en diferents apartats segons la seva funcionalitat i utilitzo el subratllat per denotar elements del codi (classes, funcions, atributs, etc.)

5.1.1. Compartits

Hi ha diversos documents de codi que utilitzen diversos agents, principalment els jugadors i els enemics. Aquests regulen la vida, l'ús de les armes o altres aspectes generals.

En el cas de la vida afegim Health a l'objecte pare (el jugador, un enemic, etc.) i utilitzem Damageable per a les diferents "HitBox", que en rebre projectils s'encarregaran d'actualitzar la vida.

Actor:

Determina l'objecte com a agent. En registra l'afiliació a un dels grups, el que permet determinar quins agents són enemics de quins. El paràmetre més important és l'AimPoint, que determina el punt a on la resta d'enemics apuntaran en atacar l'actor.

Health:

Regula diversos aspectes relacionats amb la vida d'un agent, principalment la quantitat màxima (amb l'atribut MaxHealth). L'atribut Invincible regula si l'agent pot rebre mal i steal si se li pot drenar la vida. lifeSteal determina el percentatge de vida que el propietari drena en atacar a tercers, sempre que aquest tingui l'opció activada.

La classe té funcions per rebre mal (i curar l'atacant si cal) i per curar-se. Quan la vida arriba a 0 es duen a terme les accions determinades per onDie si n'hi ha (normalment reproduir VFXs o animacions) o altrament només s'elimina l'objecte propietari.

Damageable:

Afegeix la propietat de rebre mal quan impacta un projectil o una AoE. Permet regular la proporció que rep del mal total. Això permet fer enemics més resistents (com el Boss) o enemics amb punts dèbils (com el Soldier o el Paladin), ja que podem assignar diferents objectes amb Damageable a un objecte pare amb una única Health.

WeaponController:

Determina la majoria de característiques d'una arma.

La majoria d'armes tenen un funcionament similar: disparen projectils repetidament a petició del seu propietari (jugador o no) fins que no queden bales al "clip" (el carregador). Aleshores poden recarregar, ja sigui utilitzant el mecanisme de munició infinita o el sistema de munició del jugador. Els paràmetres determinen la velocitat d'atac o la recàrrega, però les propietats del projectil (com la velocitat o el mal) les determina la classe ProjectileStandard.

Amb WeaponRoot i WeaponMuzzle es controla la posició de l'arma dins del joc, vinculant-la a un model o a un enemic.

Els atributs que regulen el comportament al disparar són els següents:

- ShootType determina si el mode és manual (cada clic correspon a un tret) o automàtic (es dispara continuament si es manté premut el clic).

- DelayBetweenShots dicta el temps entre atacs, independentment del mode en què es dispara.

- BulletsPerShot, BulletSpreadAngle i ProjectilePrefab determinen, respectivament, la quantitat de projectils, l'angle de dispersió i el tipus de projectil que es disparen en cada tret. Així, l'escopeta dispara un gran nombre de projectils que es distribueixen en un angle, el que fa que a més distància sigui difícil que impactin un mateix objectiu. La pistola, el rifle i els enemics disparen un sol projectil sense dispersió.

Els següents paràmetres regulen la munició:

- AmmoType determina el tipus de munició que utilitza l'arma, sent Infi quan no s'utilitza munició, ja que es considera infinita, com en el cas de les armes dels enemics. La resta d'armes (les del jugador) utilitzen diferents tipus de munició de la reserva especificada a la classe Ammo.

- reloadTime és el temps que a triga en recarregar l'arma sempre que es tingui munició disponible.

-ClipSize determina el nombre de bales que es poden disparar abans de recarregar.

-El paràmetre hasPhysicalBullets determina si s'han d'instanciar beines al disparar. En cas afirmatiu els paràmetres ShellCasing, EjectionPort, ShellCasingEjectionForce i ShellPoolSize serveixen per determinar el prefab a instanciar, la posició inicial, la força que reben i el nombre de beines que poden existir simultàniament.

-Els paràmetres AmmoReloadDelay i AmmoReloadRate serveixen per recuperar munició amb les armes de munició infinita. El primer determina el temps que es triga des que es deixa de disparar fins a començar a recuperar i el segon quanta munició es recupera per segon.

La resta de paràmetres regulen aspectes visuals i sonors: animacions (si s'escau), sons, VFX en disparar i l'aspecte de la mira.

ProjectileStandard:

Determina el comportament dels projectils: aspecte, velocitat, mal, mida, temps de vida. Utilitza ProjectileBase, que recull algunes propietats inicials del projectil. Els atributs més importants són:

-Radius, Root i Tip, que determinen la forma del projectil pel que fa a la detecció de col·lisions

-Speed i MaxLifeTime, que determinen la velocitat a què viatja i durant quanta estona ho pot fer abans de desaparèixer.

Les principals modificacions a les classes d'aquest apartat han sigut a Health per donar suport al sistema de robar vida i a WeaponController per donar suport a la recàrrega i el nou sistema de munició.

5.1.2. Enemies

El joc compta amb 4 tipus d'enemic, incloent-hi el Boss. Tot i les diferències entre ells, funcionen de manera similar. Tots utilitzen el component NavMeshAgent per gestionar el moviment i el "pathfinding".

El Drone utilitza EnemyMobile i EnemyController, mentre que la resta utilitzen versions modificades: SoldierMobile i SoldierController per al Soldier i el Paladin i BossMobile i BossController per al Boss. Els documents amb Mobile al nom gestionen la IA i les transicions entre els seus estats, així com les animacions (si s'escau). Els Controllers defineixen l'apuntat, els atacs i altres accions com els drops.

També utilitzen la mateixa IA amb tres estats diferents: Patrol, Follow i Attack. El primer és l'estat per defecte en què l'enemic recorre el camí de patrulla que té assignat o està estàtic si no en té cap. Si detecta un objectiu passa a l'estat de Follow. En aquest estat, persegueix l'objectiu fins que es troba dins del seu rang d'atac i passa a Attack o fins que passa un temps determinat i abandona la cerca. En l'estat d'atac es dispara a l'objectiu fins que aquest ja no es troba dins del seu rang (passant a l'estat de follow) o fins que s'elimina.

A l'hora de detectar al jugador els enemics compten amb dos mecanismes: o bé el jugador entra dins del seu rang de detecció (establert a DetectionModule) o bé el jugador els ataca.

És a l'hora d'atacar que els enemics funcionen de manera diferent, ja que compten amb diferents armes.

En el cas del Drone es tracta d'un enemic del tutorial pràcticament inalterat. Té un munt de paràmetres per modificar-ne l'aspecte visual. La principal diferència amb la resta d'enemics és la manera d'apuntar, ja que mou tot el cos per enfocar-lo a l'objectiu.

La resta d'enemics, en canvi, només roten horitzontalment. Si cal apuntar més amunt o avall només roten verticalment el canó de l'arma (en comptes de tot el cos).

També hi ha diferències en les HitBox. El Soldier i el Paladin n'utilitzen diverses i una d'elles és més sensible al mal dels projectils, el que crea un punt dèbil.

El Boss funciona totalment diferent: no es pot desplaçar sinó que només rota sobre ell mateix. Com que és bastant gran resulta fàcil encertar-lo en atacar, però les dues HitBox redueixen el mal que rebria normalment. El resultat és un enemic no només resistent, sinó que tot i permetre curar-se ho fa en menys proporció que el Soldier.

El Boss també té dues armes que fa servir: quan exhauereix la munició del carregador d'una d'elles passa a la següent. Això fa que intercali ràfegues d'atacs ràpids amb altres de més lents.

A continuació detallo els paràmetres importants de SoldierController i BossController (la majoria també presents a EnemyController) i n'explico la seva utilitat:

-PathReachingRadius: determina la distància a la qual es considera que s'ha arribat a l'objectiu.

-RotationSpeed: defineix la velocitat amb què un enemic pot girar sobre si mateix. Resulta determinant a l'hora d'apuntar al jugador.

-AimSpeed: velocitat amb què es corregeix el factor vertical en apuntar.

-LootPrefab, DropRate i les seves versions secundàries: gestionen la probabilitat de què es creï un objecte en morir i quin tipus d'objecte es crea (normalment un Pickup de munició).

La resta d'atributs gestionen principalment diversos sons i VFX que els enemics utilitzen en ser atacats, en morir o en detectar al jugador.

5.1.3. Jugador

El jugador, en l'àmbit de codi, comparteix moltes similituds amb els enemics. Principalment utilitza la mateixa vida, conta com a Actor i utilitza armes similars. A banda de les similituds, també compta amb algunes diferències, que principalment permeten la interacció amb l'usuari, però també altres aspectes com la gestió de la munició o les diferents armes.

PlayerInputHandler:

Gestiona els inputs de l'usuari ja siguin del teclat o del comandament si aquest està configurat. Ho fa utilitzant GameConstants, que recull les diferents tecles o botons assignades a cada acció. També utilitza alguns atributs per ajustar la sensibilitat o invertir els eixos.

PlayerCharacterController:

Permet controlar el personatge jugador. Gestiona el moviment i les seves restriccions, la capacitat d'ajupir-se, etc.

El jugador té capacitat per moure's horitzontalment, saltar, ajupir-se i córrer. Pot combinar aquestes accions, per exemple corrent abans de saltar per arribar més lluny.

De cara al moviment, es poden modificar diferents paràmetres. Un és la velocitat màxima (MaxSpeedOnGround), la velocitat de rotació (RotationSpeed) i un paràmetre per gestionar la facilitat amb què el jugador accelera i frena (MovementSharpnessOnGround). També hi ha paràmetres que modifiquen aquests tres proporcionalment en ajupir-se, córrer o saltar.

El salt funciona aplicant una força vertical definida a JumpForce. Un cop el jugador salta (o bé si està caient per qualsevol motiu), s'actualitzarà isGrounded com a negatiu. Mentre cau, es tenen en compte diversos factors com la gravetat (modificable a GravityDownForce) o la velocitat i acceleració màximes. En qualsevol cas el salt comporta restriccions al jugador, que no pot tornar a saltar o canviar de direcció fins que toqui el terra un altre cop.

Finalment trobem paràmetres per ajustar el model del jugador: la seva alçada (de peu i ajupit), la posició de la càmera, etc.

PlayerWeaponsManager:

S'encarrega de gestionar la llista d'armes del jugador i controla alguns paràmetres visuals com el retrocés o el camp de visió.

Per a gestionar la llista d'armes (`m_WeaponSlots`) compta amb les armes inicials (`StartingWeapons`). El codi s'encarrega d'actualitzar la llista, afegint o eliminant armes segons calgui. Conté un mètode per obtenir l'arma que està sent utilitzada (`GetActiveWeapon()`).

Gestiona el camp de visió, el moviment de l'arma o el retrocés en funció de si s'està disparant, apuntant o cap de les dues, fent zoom o aplicant transformacions al model de l'arma. Això es pot ajustar amb diferents paràmetres.

Ammo:

Gestiona la munició del jugador. Es pot configurar la munició inicial i la quantitat màxima dels quatre tipus, tot i que només se n'utilitzen tres. Hi ha funcions per afegir munició o per retirar-ne, així com per comprovar quantitats.

En aquest apartat trobem la modificació de PlayerWeaponsManager per donar suport a la recàrrega i la creació d'Ammo.

5.1.4. Gestió

Aquest apartat recull el codi que gestiona el progrés del nivell i els diferents elements d'aquest.

EventManager:

Gestiona el sistema d'esdeveniments. Aquest pot afegir i eliminar listeners, que estan pendents dels diferents esdeveniments. Quan es dona un esdeveniment, es fa broadcast i els listeners reben la notificació, actuant segons convingui.

GameFlowManager:

Controla si el joc continua o està acabant, ja sigui perquè el jugador ha sigut derrotat o perquè ha guanyat. El jugador guanya quan compleix tots els objectius. Hi ha diversos paràmetres per configurar què passa quan s'acaba el joc: WinSceneName i LoseSceneName determinen les escenes que es carregaran en cada cas, mentre que la resta de paràmetres gestionen els missatges que es mostren o el temps que es triga.

Amb GameIsEnding la resta de classes també poden veure si s'està acabant la partida i actuar en conseqüència. Per exemple, bloquejant les accions del jugador.

EnemyManager:

Permet portar un registre dels enemics, afegint-ne o eliminant-ne segons convingui.

ObjectivesManager:

Gestiona tots els objectius del joc registrant-los o complint-los quan calgui. Si tots els objectius no opcionals es compleixen, s'avisava amb el sistema d'esdeveniments per poder acabar la partida.

Objective:

La classe d'objectiu genèric, que compta amb un títol i una descripció (Title i Description). També permet marcar un objectiu amb IsOptional, convertint-lo en innecessari per acabar el nivell. Compta amb les funcions per actualitzar i complir l'objectiu.

ObjectiveDestroy:

Objectiu específic que es compleix quan s'elimina un objecte determinat. Aquest objecte pot ser qualsevol cosa del nivell i no només un enemic.

ToggleOnReach i ToggleOnDestroy:

Permeten activar o desactivar objectes del nivell en arribar a una zona o destruir un objecte concret, respectivament. S'utilitzen per bloquejar el camí al jugador i fer aparèixer enemics nous.

Els canvis importants són la creació de ObjectiveDestroy, ToggleOnReach i ToggleOnDestroy.

5.1.5. Interfície

La interfície compta amb diversos elements, principalment el HUD que veu el jugador durant la partida. Dins d'aquest trobem:

-**La vida:** utilitzant PlayerHealthBar i PlayerHealthPercentage es mostra la vida tant en forma de barra com en un percentatge. Aquesta s'actualitza accedint al component Health del jugador.

-**Les armes:** utilitzant WeaponHUDManager i AmmoCounter es mostra, de cada arma, la icona i la munició (accedint als atributs de PlayerWeaponsManager). Pel que fa a la munició es mostra la que hi ha al carregador (accedint a WeaponController) i la de l'inventari del jugador (accedint a Ammo).

-**Objectius i notificacions:** simplement es mostra el text d'aquestes (títol, descripció, etc.). En el cas dels objectius, s'eliminen un cop complets. Les classes que ho fan són ObjectiveHUDManager i NotificationHUDManager.

-**Brúixola:** mostra tots els elements rellevants (enemics o objectius). Amb Compass es manté el registre de qualsevol element i amb CompassElement es configura cadascun d'aquests.

-**Menú de pausa:** conté unes poques opcions i permet fer captures de pantalla. També permet veure els controls.

-**Menús del joc:** tots els menús utilitzen botons senzills que permeten canviar d'escena (LoadSceneButton).

En aquest cas s'ha modificat la interfície de les armes per adaptar-la al nou sistema de munició. També s'ha modificat la vida per mostrar un percentatge i s'ha actualitzat l'esquema de controls del menú de pausa.

5.1.6 Altres

Pickups:

Troblem la versió genèrica a Pickup. Existeix el HealthPickup que no s'utilitza i s'ha creat l'AmmoPickup, que permet recuperar munició dels enemics.

5.2. Disseny del nivell

El nivell, en ser només un, està dissenyat per mostrar totes les armes i els enemics. Aquests s'introdueixen progressivament i la dificultat es va incrementant quan s'avança.

L'ambientació és en un espai industrial on trobem fàbriques i magatzems i, al final, un laboratori.

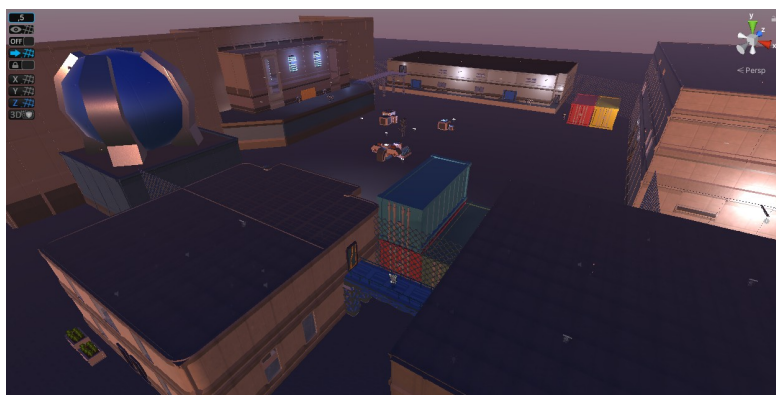


Figura 11: Vista del nivell

En el disseny inicial ja tenia clares moltes de les idees del nivell. En primer lloc, volia repartir els diferents edificis al voltant d'un pati central on hi hauria l'enemic dur. Tot i això la idea era que el jugador, si bé podria prendre decisions sobre el camí exacte que utilitzava, hauria de passar per tots els edificis. També volia afegir verticalitat que el jugador pogués utilitzar a favor seu.

5.3. Assets utilitzats

Per al projecte he utilitzat molts assets diferents. Tots els models, animacions, sons i textures del joc són, directament o indirectament, producte d'aquests assets. Tots estan disponibles a la Unity Store de manera gratuïta:

-Cyberpunk Material Pack - SIX High Quality Materials:

Asset utilitzat per a les textures dels edificis.

<https://assetstore.unity.com/packages/2d/textures-materials/cyberpunk-material-pack-six-high-quality-materials-188067>

-Free Cyberpunk Music Loops:

Conté la música de fons del menú i del nivell.

<https://assetstore.unity.com/packages/audio/music/electronic/free-cyberpunk-music-loops-95045>

-FREE Skybox Extended Shader:

Utilitzat en la Skybox del nivell 1.

<https://assetstore.unity.com/packages/vfx/shaders/free-skybox-extended-shader-107400>

-Free Sound Effects Pack:

Sons variats que utilitzen les armes i els efectes dels projectils, entre d'altres.

<https://assetstore.unity.com/packages/audio/sound-fx/free-sound-effects-pack-155776>

-Hornet rifle:

Model del rifle del jugador.

<https://assetstore.unity.com/packages/3d/props/guns/hornet-rifle-37309>

-Low Poly FPS Pack - Free (Sample):

Model de la pistola del jugador.

<https://assetstore.unity.com/packages/3d/props/weapons/low-poly-fps-pack-free-sample-144839>

-Low Poly Office Props - LITE:

Diversos objectes del nivell.

<https://assetstore.unity.com/packages/3d/environments/low-poly-office-props-lite-131438>

-Lowpoly Sci-Fi buildings set:

Part dels edificis i els objectes del nivell.

<https://assetstore.unity.com/packages/3d/environments/sci-fi/lowpoly-sci-fi-buildings-set-66885>

-Low Poly Sci Fi Set:

Part dels edificis i els objectes del nivell.

<https://assetstore.unity.com/packages/3d/environments/sci-fi/low-poly-sci-fi-set-110775>

-Low Poly Storage Pack:

Alguns dels objectes del nivell.

<https://assetstore.unity.com/packages/3d/environments/urban/low-poly-storage-pack-101732>

-Robot Metallic humanoid:

L'enemic Paladin.

<https://assetstore.unity.com/packages/3d/characters/robots/robot-metallic-humanoid-171295>

-Sci-Fi Styled Modular Pack:

Part dels edificis i els objectes del nivell.

<https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-styled-modular-pack-82913>

-Sci Fi Warrior PBR HP Polyart:

L'enemic Soldier.

<https://assetstore.unity.com/packages/3d/characters/robots/sci-fi-warrior-pbr-hp-polyart-106154>

-Shotgun Sci-Fi Weapon:

Model de l'escopeta del jugador.

<https://assetstore.unity.com/packages/3d/props/guns/shotgun-sci-fi-weapon-100194>

-Snaps Prototype | Sci-Fi / Industrial:

Diversos dels objectes del nivell.

<https://assetstore.unity.com/packages/3d/environments/sci-fi/snaps-prototype-sci-fi-industrial-136759>

6. Manual d'usuari

6.1. Manual del joc

Per jugar no cal instal·lar el joc, n'hi ha prou amb fer clic en l'executable Future Punk.

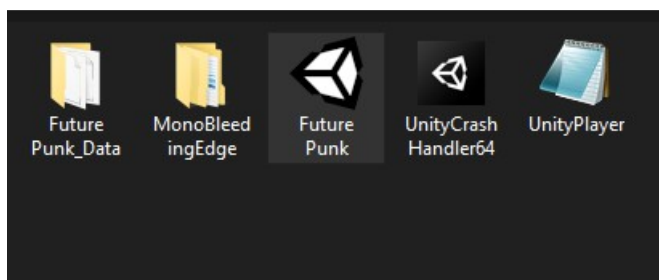


Figura 12: Executable del joc

Un cop dins podem accedir al menú de nivells i d'aquests al nivell 1.

Els controls del jugador són els següents:

Moviment - WASD o fletxes

Disparar - Clic esquerre

Apuntar - Clic dret

Saltar - Espai

Ajupir-se - C

Córrer - Shift

Recàrrega - R

Pausa - Tabulador o P

6.2. Requisits tècnics

El joc s'ha desenvolupat i executat en un equip amb les següents especificacions:

Sistema operatiu	Windows 10 Pro (64 bits)
Processador	Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz 3.50 GHz
RAM	16,0 GB
Targeta gràfica	NVIDIA GeForce GTX 970
Espai en el disc	466 MB

7. Conclusions

Aquest apartat recull diferents conclusions a l'aprenentatge realitzat. Les he separat en diferents tipus:

-Sobre l'aprenentatge del desenvolupament de videojocs:

-He descobert la gran varietat de disciplines implicades en la indústria, així com les bases del seu funcionament. Tot i que coneixia la majoria d'aquestes, he vist de primera mà la seva importància i he entès millor el seu paper.

-Moltes de les assignatures que he cursat m'han aportat coneixements útils. No només han sigut aquelles relacionades amb la programació o l'enginyeria del programari, també he aplicat coneixements de la gestió de projectes o producció de textos.

-Considero que encara tinc molt per aprendre en aquest àmbit i que, si bé no aspiro a dominar totes les habilitats relacionades amb la creació de jocs, m'agradaria aprofundir més en alguns aspectes.

-Sobre Unity i el FPS Microgame:

-Valoro molt l'aprenentatge de Unity i ho considero profitós. No només és una eina molt popular, sinó que facilita les coses a petits desenvolupadors i persones en procés d'aprenentatge.

-L'ús del FPS Microgame m'ha proporcionat una base sobre la qual podia treballar i m'ha estalviat temps en desenvolupar les parts més bàsiques del joc. També m'ha proporcionat diversos exemples de codi que m'han donat idees sobre com implementar mecàniques noves.

-També considero positiu haver treballat sobre una base per aprendre a treballar en un projecte de codi que no és només meu.

-Tot i això, també ha tingut conseqüències negatives, dificultant-me la implementació d'algunes característiques per la poca escalabilitat.

-Sobre els objectius del treball:

-Sobre l'aprenentatge el valoro molt positivament. Considero que he après no només respecte a la plataforma de Unity sinó sobre el desenvolupament en general i el que hi ha més enllà del codi.

-Sobre el producte, estic content amb la feina feta, però tinc molt clar que no esdevindria un joc en l'estat actual o que, com a mínim, jo no el jugaria. Tinc molt clares les millores que necessitaria i com les implementaria.

-Sobre la metodologia i la planificació:

-La metodologia ha sigut útil i, generalment, s'ha adaptat bé a les meves circumstàncies. Tot i això considero que es deu majoritàriament al fet que treballava sol i que el desenvolupament en un equip és un repte més gran.

-Sobre la planificació, valoro que la podria millorar molt. He hagut de modificar molts aspectes i he gestionat malament el temps. Havia sentit a parlar sobre la dificultat de planificar en el desenvolupament de videojocs i ara sóc conscient d'aquesta. Tot i això crec que l'experiència m'ha servit i que ara podria fer una planificació molt millor.

-Sobre com expandiria i seguiria treballant en el projecte:

-En primer lloc, deixaria d'utilitzar la base del FPS Microgame. Si bé considero que ha sigut útil, acabaria fent més nosa que servei. Així doncs, conservaria aquelles característiques que considerés interessants al nou projecte i descartaria la resta.

-Em plantejaria aprendre més sobre algunes disciplines com les animacions o els models per poder desenvolupar alguns dels meus assets.

-Tinc clares algunes de les millores en les mecàniques que m'agradaria implementar: millorar la IA, afegir "spawns" d'enemics, crear més nivells, crear un "Endless mode" o nous tipus d'enemic serien les meves prioritats.

-També consideraria aprendre altres entorns, destacant el nou Unreal Engine 5 que resulta bastant prometedor.

-Una de les maneres d'aprendre que he sentit i m'ha semblat més interessant és intentar implementar mecàniques de jocs que he jugat.

8. Glossari

-FPS: acrònim de First Person Shooter, fa referència als videojocs d'acció en primera persona.

-UI: acrònim de User Interface, és a dir, la interfície d'usuari.

-HUD: acrònim de Heads Up Display, és la part de la interfície que el jugador té disponible mentre està jugant.

-AoE: acrònim de Area of Effect, s'utilitza per denominar efectes que es donen en una zona i poden afectar tots els agents de dins.

-pathfinding: és la tècnica per calcular el moviment d'un personatge en funció de la seva destinació.

-drop: literalment vol dir "deixar anar", es refereix als objectes que els enemics deixen anar en morir.

-pickup: un objecte que el jugador recull obtenint-ne alguna cosa (una millora temporal, un objecte, un missatge, etc.)

-clip: el carregador d'una arma.

-demo: és la versió de demostració d'un programa informàtic.

9. Bibliografía

- 1: Video Game Industry Statistics, Trends and Data in 2021, 2021 (visitat el 19/05/2021) <https://www.wepc.com/news/video-game-statistics/>
- 2: Los indies tenían razón: Unity y los motores de terceros le han ganado la partida a los motores propios a la hora de crear juegos, 2021 (visitat el 31/05/2021) <https://www.xataka.com/videojuegos/indies-tenian-razon-unity-motores-terceros-le-han-ganado-partida-a-motores-propios-a-hora-crear-juegos-1>
- 3: The history of the first person shooter, 2017 (visitat el 30/05/2021) <https://www.pcgamer.com/the-history-of-the-first-person-shooter/>
- 4: De Doom a Overwatch: Presente, pasado y futuro del First Person Shooter, 2017 (visitat el 19/05/2021) <https://www.eurogamer.es/articulos/fun-and-serious-entrevista-jeff-kaplan-john-romero>
- 5: What is cyberpunk, 2018 (visitat el 02/06/21) <https://www.polygon.com/features/2018/8/30/17796680/cyberpunk-2077-history-blade-runner-neuromancer>