



# **Exploració i anàlisi de la psicologia del jugador aplicada a un platformer**

Treball Final de Grau  
Grau en Disseny i Desenvolupament de Videojocs

Cognoms: Clotet Rodríguez

Nom: Gerard

Curs: 2021-2022

Director: Fábregas Ruesgas, Juan José

# Índex

<b>Resum</b>	<b>4</b>
<b>Paraules Clau</b>	<b>5</b>
<b>Enllaços</b>	<b>5</b>
<b>Índex de taules</b>	<b>6</b>
<b>Índex de figures</b>	<b>7</b>
<b>Glossari</b>	<b>9</b>
<b>1. Introducció</b>	<b>10</b>
1.1 Motivació	11
1.2 Formulació del problema	11
1.3 Objectius generals	11
1.4 Objectius específics	12
1.5 Abast del projecte	12
<b>2. Estat del art</b>	<b>13</b>
2.1 La psicologia del jugador i les seves taxonomies	13
2.1.1 Aplicació actual als videojocs:	17
2.1.2 Aplicació al projecte:	18
2.2 Extracció de dades a Unity	19
2.2.1 Tipo de format	19
2.2.2 Implementació al codi	20
2.2.3 Bases de dades	22
<b>3. Planificació / Gestió del projecte</b>	<b>23</b>
3.1 Eines per a la gestió del projecte	23
3.1.1 GANTT	23
3.1.2 Trello	25
3.1.3 GitHub repositori en xarxes, Git eines de control de versions	26
3.1.4 Playfab	27
3.1.5 Estat de la planificació	27
Rúbrica 2	27
Rúbrica 3	28
3.2 Eines de validació	29
3.3 DAFO	30
3.4 Riscos i pla de contingències	30
3.5 Anàlisi Inicial de Costos	31
<b>4. Metodologia</b>	<b>33</b>
4.1 Tipus de Metodologia	33
4.2 Estat de la metodologia	34

<b>5. Desenvolupament</b>	<b>36</b>
5.1 Disseny i Exploració	36
5.1.1 Taxonomia de Jugadors	36
5.1.2 Mètriques per diferenciar als jugadors	39
5.1.3 Elements per acomodar l'usuari al seu perfil de jugador	39
5.2 Desenvolupament del Videojoc Base	40
5.2.1 Sistema de Input	40
5.2.2 Mecàniques Bàsiques del joc	41
Moviment	44
Salt	44
Salt per les parets	45
Dash	45
5.2.3 Cinemachine	45
5.2.4 Game Feel	46
Animacions	47
Audio	47
5.2.5 Elements del mapa	48
Plataformes	48
Objectes recol·lectables	48
Morts	49
5.2.6 Nivells	49
5.3 Monitoratge i Extracció de dades	50
5.3.1 Canvi de plantejament	53
5.3.2 Còmput de dades	54
5.4 Connexió amb base de dades	55
5.4.1 Problemes Inicials	55
5.4.2 Funcionament	55
5.5 Tecnologies Externes a Unity	57
<b>6. Conclusions i treballs futurs</b>	<b>58</b>
6.1 Tests	58
6.2 Objectius generals	61
6.3 Objectius específics	62
6.4 Treballs futurs	62
<b>7. Bibliografia</b>	<b>63</b>

## Resum

Aquest document descriu el procediment, el desenvolupament i l'exploració d'una taxonomia del perfil de jugador aplicat a un videojoc del gènere de plataformes.

L'objectiu principal d'aquest projecte és veure si existeix alguna relació amb la taxonomia establerta i l'experiència dels usuaris mitjançant tests. Entendre si l'experiència de joc ha millorat.

Per aconseguir-ho s'ha desenvolupat un prototip d'un videojoc de plataformes centrat a recollir dades i modificar el comportament del joc segons la taxonomia establerta. S'ha agafat una mostra amb diferents usuaris i s'ha comprovat si realment la taxonomia es compleix.

En el document s'exposen les tècniques i els procediments utilitzats d'acord amb les decisions preses per l'estudiant des de la concepció de la taxonomia fins als tests finals. Passant per les dificultats del desenvolupament.

Gerard Clotet

Exploració i anàlisi de la psicologia del jugador aplicada a un platfomer



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Centre de la Imatge i la Tecnologia Multimèdia

## Paraules Clau

Taxonomia, Unity, Videojocs, Bartle

## Enllaços

### Repositori del projecte

<https://github.com/GerardClotet/TFG>

### Release del prototip

<https://github.com/GerardClotet/TFG/releases/tag/1.0>

## Índex de taules

<b>Taula 2.1:</b> Model de Yee sobre la motivació en els MMORPGS	Pag. 16
<b>Taula 3.1:</b> Estimació total d'hores del projecte	Pag. 25
<b>Taula 3.3:</b> DAFO	Pag. 30
<b>Taula 3.4:</b> Riscos i pla de contingències	Pag. 30
<b>Taula 3.5:</b> Anàlisi inicial de costos	Pag. 31

## Índex de figures

<b>Figura 2.1.1:</b> Taxonomia de Bartle	Pag. 13
<b>Figura 2.1.2:</b> Comparació de taxonomies	Pag. 15
<b>Figura 2.2.1:</b> Esquema de subscriptors del patró observer	Pag. 20
<b>Figura 2.2.2:</b> Esquema del patró observer	Pag. 21
<b>Figura 2.2.3:</b> Esquema de controls del input system	Pag. 22
<b>Figura 3.1.1.1:</b> Gant Inicial	Pag. 23
<b>Figura 3.1.1.2:</b> Gant Rúbrica 2	Pag. 24
<b>Figura 3.1.2:</b> Trello	Pag. 26
<b>Figura 3.1.3:</b> Esquema del funcionament de les branques a Github	Pag. 27
<b>Figura 5.1.1.1:</b> Tipologia de jugadors per a platformers	Pag. 36
<b>Figura 5.1.1.2:</b> Connexions entre rols principals i secundaris	Pag. 38
<b>Figura 5.2.1.1:</b> Mapa d'accions del jugador	Pag. 40
<b>Figura 5.2.1.2:</b> Mapa d'accions de l'interfície d'usuari	Pag. 41
<b>Figura 5.2.2.1:</b> Loop general del jugador	Pag. 41
<b>Figura 5.2.2.2:</b> Diferents estats del jugador	Pag. 42
<b>Figura 5.2.2.3:</b> Flux dels diferents estats del jugador	Pag. 43
<b>Figura 5.2.2.4:</b> Diferents estats del jugador	Pag. 44
<b>Figura 5.2.2.5:</b> Alçada del salt segons el frame	Pag. 45
<b>Figura 5.2.3:</b> Sistema d'habitacions de l'escena	Pag. 46
<b>Figura 5.2.4:</b> Valors del càmera shake	Pag. 47
<b>Figura 5.3.1:</b> Fases del monitoratge i l'extracció de dades	Pag. 50
<b>Figura 5.3.2:</b> Format JSON del qüestionari	Pag. 51

<b>Figura 5.3.3:</b> Preguntes i respostes generades de forma dinàmica	Pag. 52
<b>Figura 5.3.4:</b> Pantalla de controlador desconnectat	Pag. 53
<b>Figura 5.3.1.1:</b> Dades extretes del jugador a PlayFab	Pag 54
<b>Figura 5.4.2.1:</b> Login de Playfab	Pag. 55
<b>Figura 5.4.2.2:</b> Pujar les dades a Playfab	Pag. 56
<b>Figura 5.4.2.3:</b> Usuaris resitrats de forma anònima a Playfab	Pag. 56
<b>Figura 5.4.2.4:</b> Informació dels usuaris	Pag. 57
<b>Figura 6.1:</b> Mitjana de percentages dels rols principals	Pag 58
<b>Figura 6.2:</b> Mitjana de morts per nivell	Pag 59
<b>Figura 6.3:</b> Percentage de rol d'Achiever	Pag 60
<b>Figura 6.4:</b> Percentatge de rol d'Explorer	Pag 61



## Glossari

**Taxonomia de Bartle:** és una classificació de jugadors de videojoc que va ser realitzada el 1996 per Richard Bartle, Classifica el perfil de jugador segons la personalitat i comportament que mostren en jocs MUD (multi-user dungeons).

**Motor Gràfic:** Software amb totes les eines necessàries per crear un videojoc.

**Unity:** Motor gràfic 3D multiplataforma en temps real publicat per Unity Technologies al 2005.

**Github:** servei de hosting de repositoris Git.

**Playfab:** plataforma de desenvolupament d'aplicacions multimèdia llançada el 2014 i per microsoft el 2018.

**Patró Observer:** Patró de disseny de comportament de programació.

**Platformer:** Gènere de videojoc que es caracteritzen per haver de caminar, córrer, saltar o escalar sobre una sèrie de plataformes i penya-segats.

**Interfície d'Usuari(UI):** elements visuals que permeten a l'usuari interactiu de manera efectiva i intuïtiva amb un sistema.

**Package:** paquet que conté tecnologia per a ser importada i utilitzada a un projecte de Unity.

**Serialitzar/Deserialitzar:** codificar o decodificar dades en un format concret.

**Script:** seqüència de comandes que llegeix l'aplicació en temps d'execució.

**Input:** senyals d'entrada a l'aplicació per part de l'usuari.

**Components:** elements funcionals que s'adhereixen als objectes de la escena de Unity per modificar el seu comportament.

**Event:** funcionalitat en programació on un objecte envia una senyal quan passa una acció als receptors

**Collider:** component de Unity que serveix per detectar col·lisions entre objectes.

## 1.Introducció

Els estudis de videojocs busquen que els seus usuaris passin el temps més gran possible dintre de la seva aplicació, sobretot als jocs que tenen pagaments in-game. A la indústria del videojoc la majoria de companyies s'enfoquen en treure continguts nous, fer rebaixes, esdeveniments dins del joc i un llarg etcètera.

El que no s'ha vist dintre de la indústria és que el joc s'acomodi a l'usuari. Que el target del joc sigui modular o modificable i diferents tipus d'usuari hi tinguin cabuda. De fet, passa el contrari que només es té un tipus de públic objectiu i s'aparta a la resta de jugadors.

La finalitat d'aquest treball és una exploració. A través d'un videojoc extreure informació dels usuaris mentre juguen [\[5.2.2\]](#) i a través de un qüestionari final, catalogar els usuaris segons un model propi [\[5.2.1\]](#) que parteix de la taxonomia de Bartle però enfocat a un Platformer, gènere del joc que es desenvolupa en aquest treball.

Un cop els usuaris estiguin classificats es modifica el joc per al seu perfil de jugador d'acord amb el seu comportament amb l'objectiu de comprovar si les diferents modificacions del joc aconseguen mantenir els usuaris més temps jugant.

Tot això es verifica amb un qüestionari final i fent que els usuaris juguin les altres modificacions del joc amb el seu corresponent qüestionari així es verifica si la classificació inicial que s'ha fet de l'usuari era la més adient per mantenir-lo més temps jugant.

### 1.1 Motivació

El disseny de videojocs és l'aspecte dels videojocs que més m'apassiona. Sempre que he dissenyat qualsevol aspecte o mecànica d'un joc hi ha la dualitat hi ha la dualitat de pensar a quin grup de jugadors estic atraient i a quin n'estic fent fora.

Un dels propòsits d'aquest projecte és que la dualitat del disseny quedi més diluïda, ja que el videojoc es pot enfocar en diferents tipus de jugadors alhora. Amb aquest treball es poden explorar diferents tipus de disseny que és el que més m'apassiona amb el fet de veure com els usuaris reaccionen als diferents tipus de disseny plantejats.

No es pretén reinventar la roda ni fer un experiment, sinó comprovar el grau d'encert i d'impacte de modificar certs aspectes d'un joc per a cada individu.

## 1.2 Formulació del problema

El que es planteja en aquest treball és una exploració. Veure com afecta catalogar diferents jugadors segons mètriques pròpies o basades en algun marc ja existent com podria ser *La Taxonomia de Bartle* en la seva experiència de joc.

Aquesta exploració neix de la necessitat de resoldre un problema que sorgeix a l'hora de dissenyar videojocs, com saber quines mecàniques són més adients per al públic que es vol atreure o si fins i tot si diferents perfils de jugadors poden coexistir dintre del mateix joc, és a dir ampliar el públic a qui va dirigit el joc.

És exactament això el que s'explora: veure si el perfil amb el qual hem catalogat a aquell jugador i la seva posterior experiència de joc concorden o no. Extreure dades de si diferents jugadors comparteixen elements similars per arribar a crear un perfil d'elements o peculiaritats constants on es pugui categoritzar un tipus de jugador.

## 1.3 Objectius generals

- Definir els paràmetres per catalogar els diferents tipus de jugadors.
- Crear videojoc on fer l'exploració.
- Contrastar els resultats obtinguts amb els esperats.

## 1.4 Objectius específics

En relació al primer punt:

- Definir els 4 tipus de jugadors partint com a base de la taxonomia de Bartle però enfocats en un joc platformer.
- Obtenir les dades dels usuaris a través de diferents elements de monitoratge dins de Unity i exportar-los en diferents formats (CSV i JSON).

En relació al segon punt:

- Desenvolupament de un videojoc amb unity
  - Desenvolupar i dissenyar les 4 modalitats diferents del joc
  - Desenvolupar i dissenyar el sistema d'extracció de dades
- Integar l'extracció de dades a Unity

En relació al tercer punt:

- Fer tests a usuaris amb qüestionaris.
- Comprovar la precisió de les prediccions.

## 1.5 Abast del projecte

Aquest projecte està dirigit als desenvolupadors de videojocs, que sabent quines mecàniques poden modificar al seu joc poden ampliar la base de jugadors i, per tant, arribar a crear un producte més competitiu de cara al mercat. Ja sigui empreses Indie com a empreses grans, el funcionament és el mateix.

S'ha de tindre en compte que el cost del producte augmenta, ja que la fase de desenvolupament s'allarga i empresa ha de fer un anàlisi de mercat, veure si el públic extra que poden atraure compensa els costos de desenvolupament.

Indirectament, es beneficien els jugadors, ja que en augmentar el públic objectiu dels jocs és més probable que un jugador se senti atret per a més jocs. Per una altra banda, en modificar el joc segons el perfil psicològic de l'usuari la qualitat de l'entreteniment augmenta.

## 2. Estat del art

L'estat de l'art està enfocat en dos aspectes molt concrets que són els quals el projecte pivota. El primer és l'obtenció de dades i monitorització amb Unity, veure quins mètodes i packages s'utilitzen actualment. L'altre aspecte en el qual s'enfoca l'estat de l'art és en quin punt està la psicologia del jugador aplicada als jocs.

### 2.1 La psicologia del jugador i les seves taxonomies

La taxonomia de Bartle<sup>[1]</sup> és el referent utilitzat arreu del món quan es vol classificar la psicologia dels jugadors. (Fig 2.1) **(més que el referent és el primer que va classificar els jugadors en diferents tipos)** En aquest cas eren per jocs MUD (Multi-User-Dungeons) jocs de text primitius a el que avui serien els MMORPG com el WOW (World of Warcraft) com a màxim exponent. Hi ha diferents tests per determinar quin tipus de jugador és l'usuari, per exemple el de Matthew Barr[2] que va migrar el test quan la pàgina GamerDNA, la qual guardava les dades de tots els tests va deixar de ser operativa.

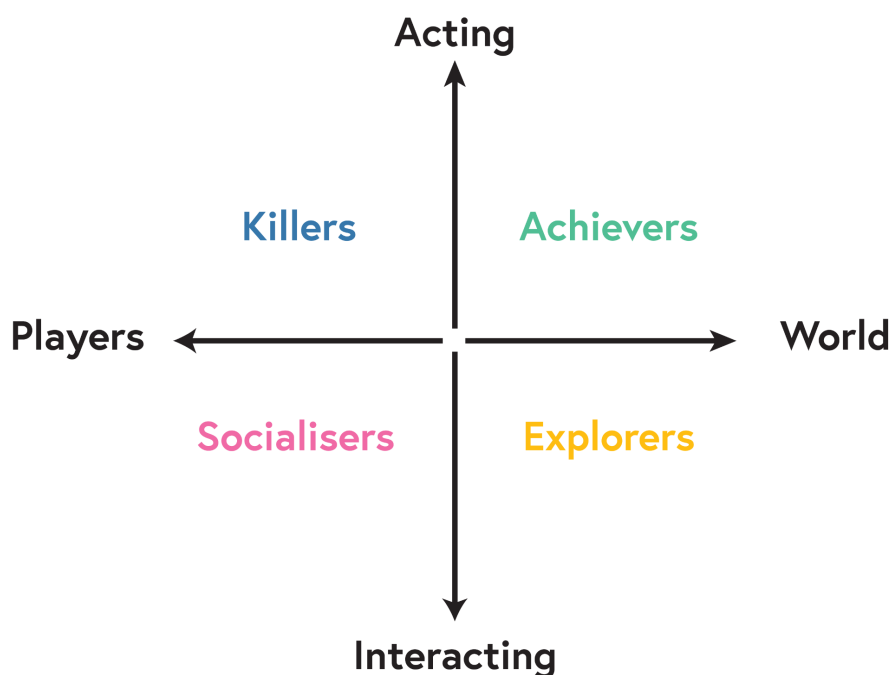


Figura 2.1.1: Taxonomia de Bartle

Els models teòrics de jugadors *Killer*, *Achiever*, *Socialiser* i *Explorer* que es poden traduir com a Competició, Repte, Germanor i Exploració. Aquest cas és per un tipus de jocs concrets, on la diversió del joc es pot obtenir d'aquestes quatre formes segons Bartle.

Pot ser que per la forma en què un joc estigui dissenyat o les característiques que tingui hi hagi algun d'aquests elements que no hi sigui present per assolir la diversió que en essència és l'objectiu de tots els jocs.

Hunicke, Leblanc i Zubek<sup>[3]</sup> expandeixen o creen 8 termes sobre quines categories característiques del disseny poden influir en el que un usuari gaudeix dels videojocs

- Fantasia: imaginar mons i personatges ficticis.
- Narrativa: desenvolupament dramàtic dels events.
- Expressió: customitzar i personalitzar els objectes dels jocs (com el propi jugador).
- Submissió: suspensió de la realitat.
- Sensació: activació dels sentits, en el cas dels videojocs seria el tacte, la vista i l'oïda.
- Repte: resolució de problemes.
- Confraternitat: amistat, cooperació i comunitat.
- Descobrimet: buscar i trobar coses noves.

Per una altre banda John M. Quick i Robert K. Atkinson<sup>[4]</sup> a través d'un qüestionari on es planteja algunes característiques del joc com que és en un món en 3D, el joc permet buscar objectes amagats i que està ambientat en un món de fantasia. Extreuen sis tipus de jugadors diferents per aquest tipus de joc:

- Fantasia: el gaudir de les característiques de un món de fantasia y jugar amb el rol de diferents espècies.(13%)
- Explorció: el plaer de buscar coses ocultes, col·leccionar objectes i explorar llocs desconeguts(10%)
- Companyerisme: el gaudir dels jocs multijugador i jugar amb amics(9%)
- Realisme: disfrutar dels gràfics realistes i els escenaris del món real (8%)
- Repte: el plaer de dominar els jocs difícils, superar obstacles i aconseguir puntuacions altes.

Aquesta taxonomia es compara amb taxonomies del disseny de videojocs anteriors.

Heeter<sup>[5]</sup> expandeix el número de tipus de jugadors als MMO a 13 tipus de jugadors diferents que els agrupa en quatre categories: intrínsec, extrínsec, antisocial i prosocial.

Els jugadors intrínsecs estan fortament relacionats amb les dimensions de la fantasia, l'exploració i el repte (challenge). Els jugadors extrínsecs estan lligats a les categories de competició, repte i exploració. Els prosocials estan lligats al companyerisme i els antisocials a la competició.

En el mateix article es comparen 10 taxonomies diferents per extreure punts comuns.

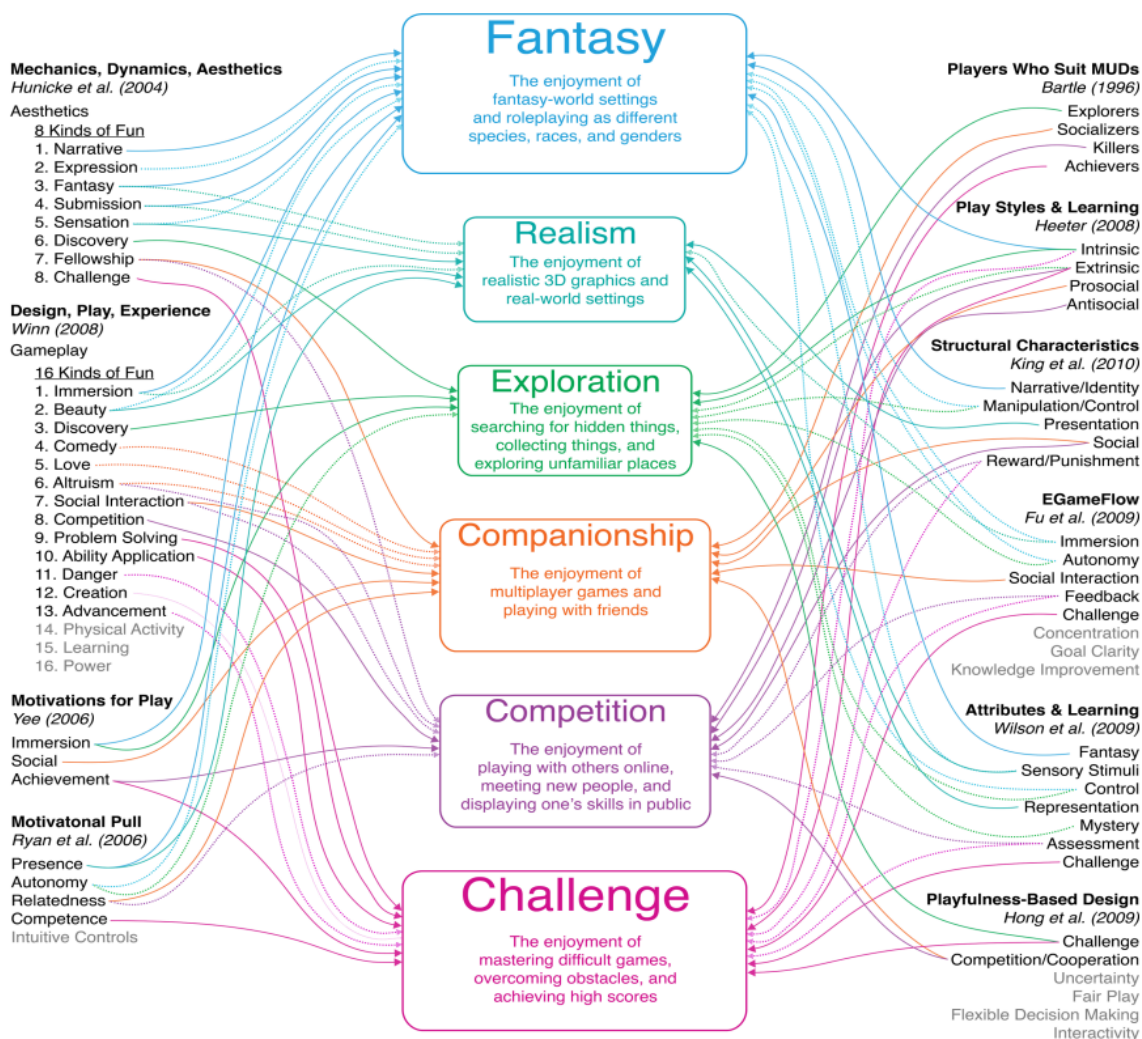


Figura 2.1.2: Comparació de deu taxonomies diferents respecte a la de Undergraduate Student

Yee<sup>[6]</sup> respecte als MMORPGs i la motivació, proposa una taxonomia de tres components: Immersió, Social i Achievement a diferència de Bartle (que l'utilitza com a punt de partida) els seus components sí que estan relacionats entre ells. En la taula 2.1 es veu en més profunditat els components de Yee. John M. Quick i Robert K. Atkinson agafen els tres components i els relacionen amb la seva taxonomia com es pot veure a la figura 2.

Achievement	Social	Immersion
Progrés, estatus	Socialitzar	Descobrir
Mecàniques	Relacionar-se	Role-Playing
Competició	Treball en equip	Personalitzar
-	-	Evadir-se

Taula 2.1: Model de Yee sobre la motivació en els MMORPGS<sup>[7]</sup>

Ryan (2006)<sup>[8]</sup> explora el rol de la teoria de l'autodeterminació 'STD'<sup>[9]</sup> en quant a la diversió dels videojocs. Segons l'estudi gaudir d'un videojoc està relacionat amb el tipus de feedback utilitzat cap a l'usuari, les regles establertes pel joc i els seus elements socials.

Extreu cinc dimensions diferents per als jugadors:

- Presència: el jugador té la certesa que està dintre del món del videojoc, també anomenat suspensió de la realitat.
- Autonomia: sensació de voler fer les tasques que s'encomanen a l'usuari dintre del joc. El jugador ha de tindre la sensació que té el control i pot escollir dintre del joc.
- Relació: quan el jugador se sent connectat amb altres jugadors o NPC's
- Competència: necessitat per tindre reptes i que el jugador noti que les seves accions impacten en el joc. Per exemple oportunitats per obtenir noves habilitats o rebre feedback positiu.
- Controls Intuïtius: la satisfacció al joc és el grau en el qual els controls són intuïtius; tenen sentit, són fàcils de masteritzar i no interfereixen amb la sensació d'estar dintre del joc.

Hi ha una infinitat de taxonomies, el que comparteixen totes elles, són diferents elements intrínsecs a l'ésser humà, segons diferents articles com el de Science of Fun<sup>[10]</sup> són elements que produeixen dopamina a l'assolir-los i serotonina en recordar-los i que estan fortament lligats al tipus de joc, és a dir al disseny del joc. Per això no es pot determinar una taxonomia com a universal per a tots els jocs. Si no que cada estudi escull o determina un marc amb els elements més adients a les seves mecàniques i dinàmiques.



Els estudis de videojocs tenen en compte diferents models teòrics de la psicologia del jugador segons les característiques dels seus jocs. No obstant això, no les potencien, és a dir, no exploten els elements que generen més dopamina al jugador, sinó que el joc bombardeja el jugador a parts iguals amb totes les seves mecàniques diferents per atraure a segments diferents de jugadors i és el mateix jugador el que ha d'escollir l'element li agradi més, amb el perill que al ser bombardejat amb la resta de mecàniques/dinàmiques, el jugador acaba avorrint el joc en si.

En aquest treball el que es planteja és una perspectiva diferent i complementaria: determinar prenent com a referència un marc propi o bé un marc ja creat, com per exemple el de Bartle, quins elements s'han de maximitzar i quins s'han d'amagar o directament suprimir en el videojoc per adaptar-lo al jugador, amb la fi que aquest passi el temps més gran possible utilitzant les mecàniques o les dinàmiques del joc que produeixen més dopamina al jugador, és a dir les més satisfactòries per a ell.

### 2.1.1 Aplicació actual als videojocs:

Segons *Dave Rickey*<sup>[11]</sup>, les taxonomies s'utilitzen per als dissenyadors per ajudar a definir els requeriments d'un videojoc per a atraure a un tipus de jugadors. És a dir que sempre s'usa la taxonomia com a plantejament inicial des de el qual partir a l'hora de dissenyar el videojoc.

Segons *Atharva Kulkarni*<sup>[12]</sup> la fàcil accessibilitat als videojocs de mòbil per a l'usuari ha aconseguit que aquests tinguin una base molt ampla de jugadors i també molt variada, per poder monetitzar a tots els jugadors se'ls ha de dividir en diferents parts. En aquest cas posa d'exemple la monetització a través de la taxonomia de Bartle. Per això s'ha d'explicar el concepte de les quatre C's de Will Luton al seu llibre *Free To Play*<sup>[13]</sup>:

- **Content:** el nucli del joc, la part de la qual tot gira, el que mou l'usuari a jugar, monetitzar el contingut és complex ja que se suposa que el joc és gratuït.
- **Convenience:** és el pa de cada dia quan es tracta de monetitzar el joc. El temps o el progrés es poden comprar directament i quan s'ofereix en el moment adequat i en la quantitat correcta suposen un incentiu considerable perquè els jugadors es gastin diners.
- **Customization:** està orientada principalment als socializers i als achievers. Skins, marcs de perfil, icones, etc. Són col·leccionables per als achievers i incrementen la interacció per als socializers.

- **Competitive Advantage:** està dirigida principalment als killers, jugadors que volen guanyar als altres de la forma que sigui. L'avantatge competitiu a vegades pot ser difícil de monetitzar, ja que un petit desequilibri al joc pot desincentivar la motivació dels jugadors a continuar utilitzant el joc.

Les interaccions entre diferents tipus de jugadors poden ajustar-se per afegir noves opcions de compra. Proporcionar opcions de compra diferents entre els jugadors és clau per fidelitzar-los i treure rendiment econòmic.

### 2.1.2 Aplicació al projecte:

En aquest projecte el que es busca és definir un tipus de taxonomia per aplicar-la a un videojoc on a diferència de la indústria, s'utilitza la taxonomia per millorar l'experiència de joc de tots els possibles jugadors que es defineixen dins del gènere del joc.

En ser un procés exploratori a través del test i la iteració, els tipus de jugadors que s'inclouen a la taxonomia es defineixen en base de les mateixes mecàniques del joc.

La taxonomia que s'usa en el videojoc que es crea al projecte parteix de la base de la Taxonomia de Bartle, la més referenciada a Google. Els seus models de jugadors són bastant universals i aplicables a molts gèneres diferents. Tot i això, la taxonomia de Bartle presenta certes limitacions respecte al platforming d'un jugador, ja que Bartle crea la seva taxonomia enfocada al multijugador; així i tot, en dividir els jugadors en anhels intrínsecs a l'ésser humà; com el d'explorar, el de competir o el d'obtenir-ho tot. Aquests elements bàsics es poden extrapolar a un joc de plataformes i les seves mecàniques, com saltar, aconseguir puntuacions altes, objectes o zones ocultes. Com també es pot extrapolar el test de Bartle<sup>[2]</sup> per poder definir quin tipus de jugador és l'usuari a través d'un qüestionari online.

## 2.2 Extracció de dades a Unity

Una de les finalitats d'aquest projecte és comprar els resultats obtinguts als tests amb les mètriques resultants a l'hora de jugar. Per això és imprescindible tenir un bon sistema d'extracció de dades.

### 2.2.1 Tipo de format

Es defineixen per defecte tres tipus de formats<sup>[14]</sup>:

**CSV(*comma separated values*):** Aquest format de dades és bàsicament una llista d'elements separats per comes.

- **Pros:** és el format més compacte dels tres, sol ser la meitat que els formats XML i JSON, cosa que ajuda a reduir l'amplada de banda, cosa que en aquest projecte no és un problema.
- **Cons:** és el format menys versàtil dels tres, requereix un parsing propi per serialitzar i deserialitzar les dades, per tant sí l'estructura es modifica s'ha de tenir en compte a l'hora de deserialitzar. Però el problema més gran respecte a aquest projecte és que no suporta jerarquies, imprescindible a l'hora de fer test i tindre tota la informació del jugador encapsulada.

**XML(*extensible markup language*):** creat sobretot per representar millor formats de dades amb una estructura jeràrquica.

- **Pros:** suporta dades amb estructura jeràrquica i funciona molt bé fins i tot quant les dades són complexes. Alhora és fàcil de llegir de cara a l'usuari amb les dades crues del fitxer.
- **Cons:** aquest tipus de format pesa molt i si s'ha d'enviar a un servidor o a través de la xarxa pot ser un inconvenient.

**JSON(*Javascript Object Notation*):** és una estructura de dades jeràrquica, al contrari que l'XML representa les dades jeràrquiques amb comes.

- **Pros:** suporta un format jeràrquic de dades, és molt llegible de cara a l'usuari. La serialització i deserialització de dades és relativament més senzill. Ajunta el millor dels dos formats anteriors, és simple i compacte com el CSV i suporta jerarquies. La mida dels fitxers JSON només són el doble de llargs que els de CSV.
- **Cons:** si es necessita amplada de banda, la millor opció seria CSV, ja que pesa menys.

## Conclusió

JSON és el millor format per guardar i transferir informació avui en dia, s'utilitza en moltes webs, buscadors i videojocs. És lleuger i versàtil. El CSV només s'hauria d'usar quan s'envien grans quantitats de dades i si l'amplada de banda és un problema. Sobre l'XML ja no és necessari emprar-lo, només en documents Markups.

En aquest treball es generen dos tipus de fitxers, CSV per estudiar dades a gran escala. Com la quantitat de salts que s'han fet, les morts que hi ha hagut i un fitxer JSON per estudiar cas per cas cada usuari.

### 2.2.2 Implementació al codi

S'utilitza el patró Observer<sup>[15]</sup> per extreure dades dels events. Consisteix en el fet que diferents objectes se subscriuen a un objecte, per exemple quan el jugador aconsegueix un *PowerUp* i quan a l'objecte on s'està subscript canvia d'estat, els objectes subscriptors reben una notificació del seu canvi d'estat.

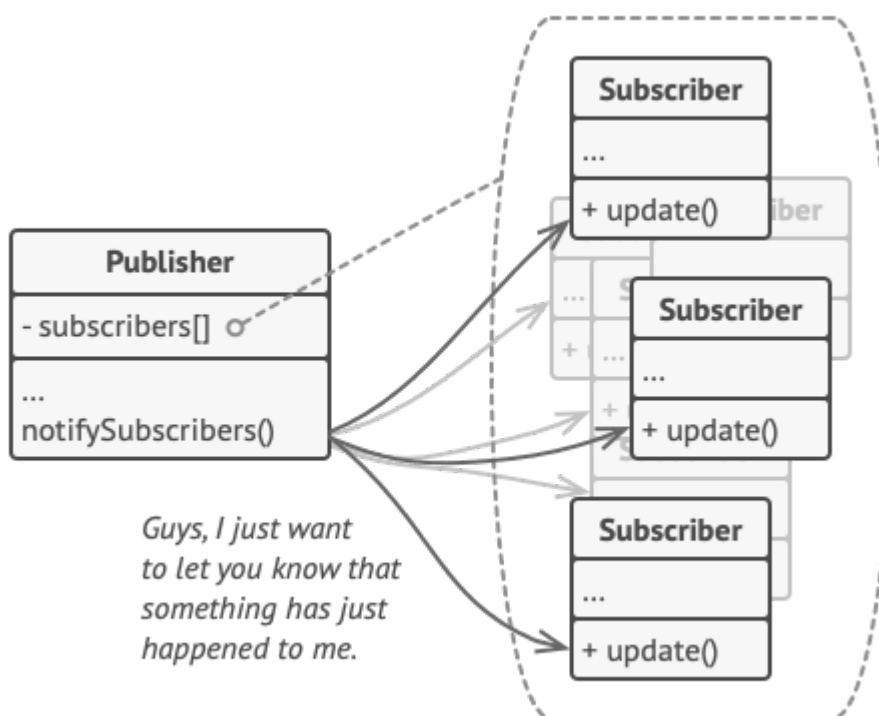


Figura 2.2.1: Esquema de subscriptors del patró observer

Un exemple és que quan el jugador agafa el PowerUp, s'actualitza el fitxer on es recullen les dades del jugador, per una banda, i per un altra banda a l'objecte del jugador s'activen els efectes del PowerUp.

L'avantatge d'aquesta forma és que s'aconsegueix que els objectes subscriptors no estiguin sempre preguntant per l'estat de l'objecte notificador, sinó que és el notificador qui avisa. Al notificador li és igual el nombre de subscriptors que tingui, i es pot passar el mateix objecte notificador com a paràmetre perquè cada subscriptor agafi els elements que vulgui del notificador.

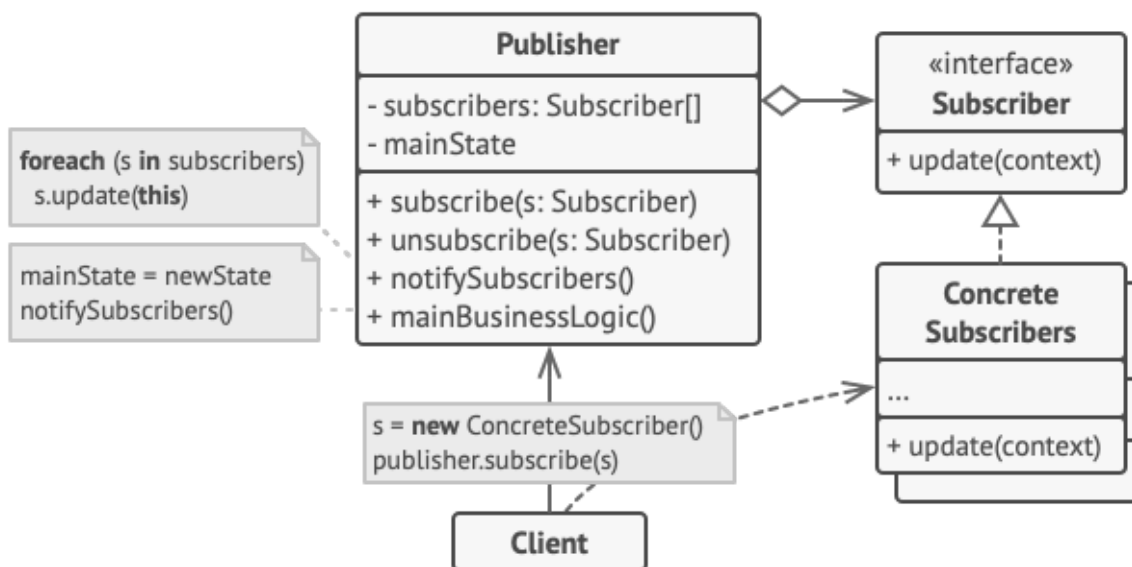


Figura 2.2.2: Estructura del patró observer

A Unity ja hi ha una classe preparada per serialitzar i deserialitzar fitxers JSON anomenada `JsonUtility`<sup>[16]</sup> lo que agilitza molt el procés de treballar amb aquest tipus de fitxers. Per altre banda la serialització de CSV també<sup>[17]</sup> és senzilla i ràpida amb el package de Unity.

## Sistema de Input

S'utilitza un package creat per al propi Unity anomenat `InputSystem`<sup>[18]</sup>, la principal raó és que és més fàcil i accessible de modificar i permet crear un sistema d'events al voltant de l'input de manera molt ràpida. Permet crear i gestionar un esquema de controls de forma visual i generar un script amb el qual treballar després.

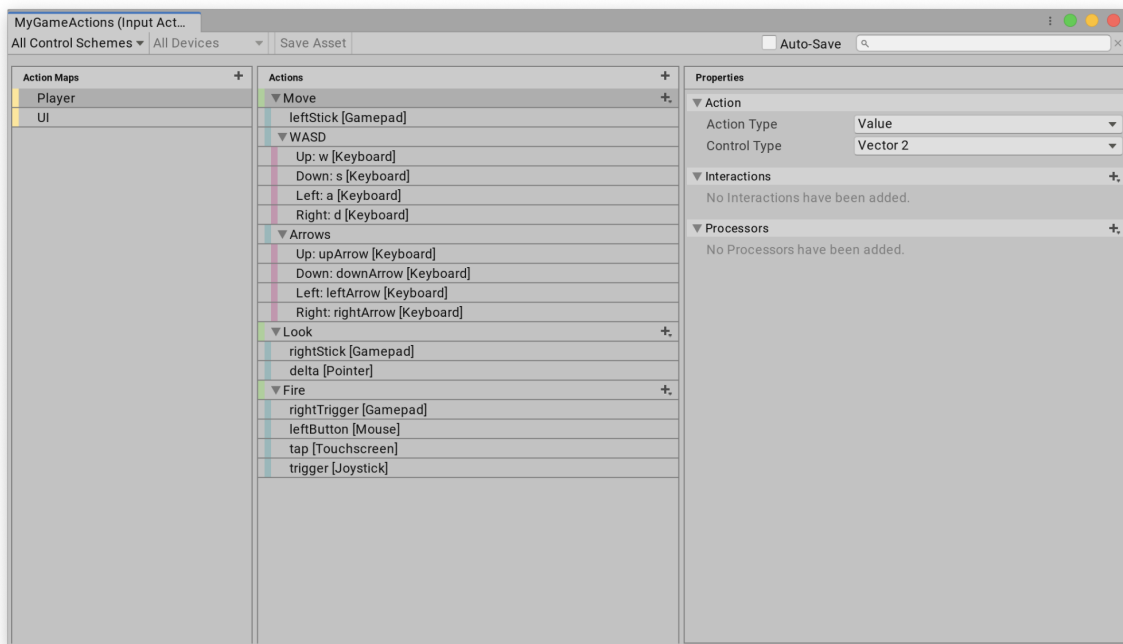


Figura 2.2.3: Esquema de controls del Input System

Amb aquest sistema d'events creat s'agilitza molt el procés de monitoritzar el jugador sobre els aspectes referents a les seves accions com per exemple quantes vegades salta.

### 2.2.3 Bases de dades

**Firebase:** proporciona una base de dades en temps real, organitzada en forma d'arbre JSON. Aquest servei proporciona una API que permet emmagatzemar i sincronitzar informació de les aplicacions en el núvol de Firebase, la implementació amb Unity es senzilla. Tot i ser la més utilitzada actualment no està pensada per ser utilitzada en aplicacions d'escriptori, proporciona emmagatzemament de fitxers.

**Playfab:** també proporciona una base de dades en temps real, es pot organitzar en format JSON, contempla un sistema de analítiques en temps real és de franc i la implementació amb Unity és senzilla i funciona amb aplicacions de ordinador.

**AWS:** Amazon Web Services o AWS és una sèrie de serveis de computació en el núvol públic que conjuntament formen un servei d'emmagatzematge en el núvol. Actualment competeix directament. Més complicada d'utilitzar que Playfab o Firebase.

## 3. Planificació / Gestió del projecte

### 3.1 Eines per a la gestió del projecte

- **GitHub:** per a guardar el repositori del codi a la xarxa i ordenar-lo millor, a sobre poder treballar des de qualsevol lloc.
- **Trello:** per aplicar la metodologia Kanban a la gestió del projecte.
- **Visual Studio:** aplicació per programar el joc i l'extracció de dades.
- **Unity:** El motor on es munta el joc i s'extreuen les dades.
- **Playfab:** sistema on s'emmagatzema les dades extretes de cada partida.
- **TeamGantt:** software online on es pot planificar de manera àgil les tasques i la seva duració, amb el diagrama de Gantt.

#### 3.1.1 GANTT

El diagrama de Gantt és una eina gràfica que serveix per mostrar el temps previst per diferents tasques, diferenciades amb els seus subgrups, en un període de temps.

Aquest interval comprèn tot el desenvolupament d'aquest projecte.

Les tasques de cada fase són tasques generals, és a dir, dins de cada tasca hi haurà múltiples subtasques per assolir l'objectiu que es proposa.

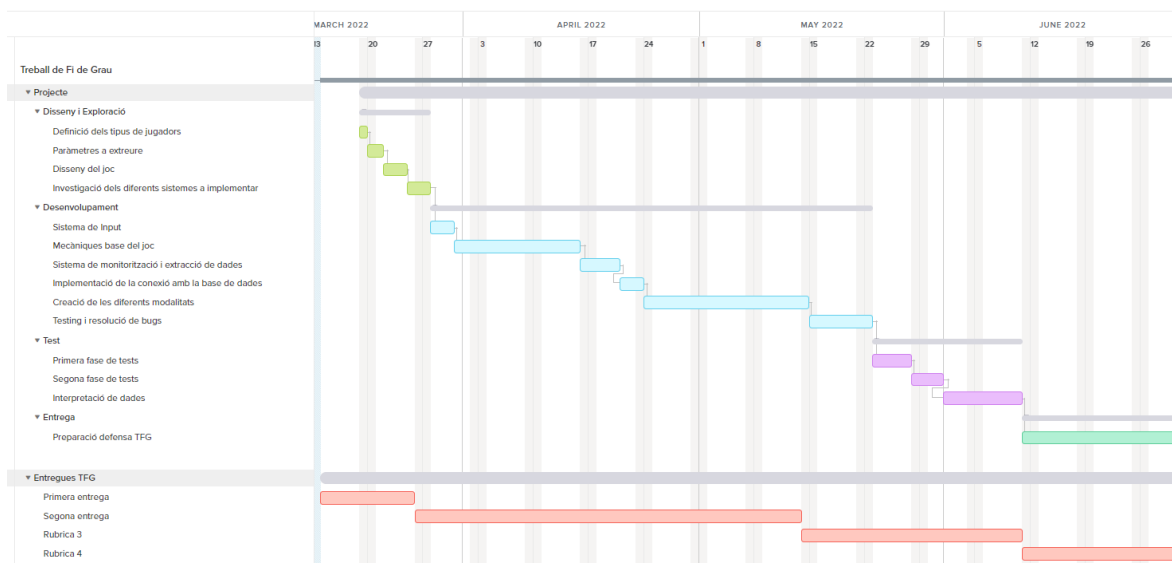
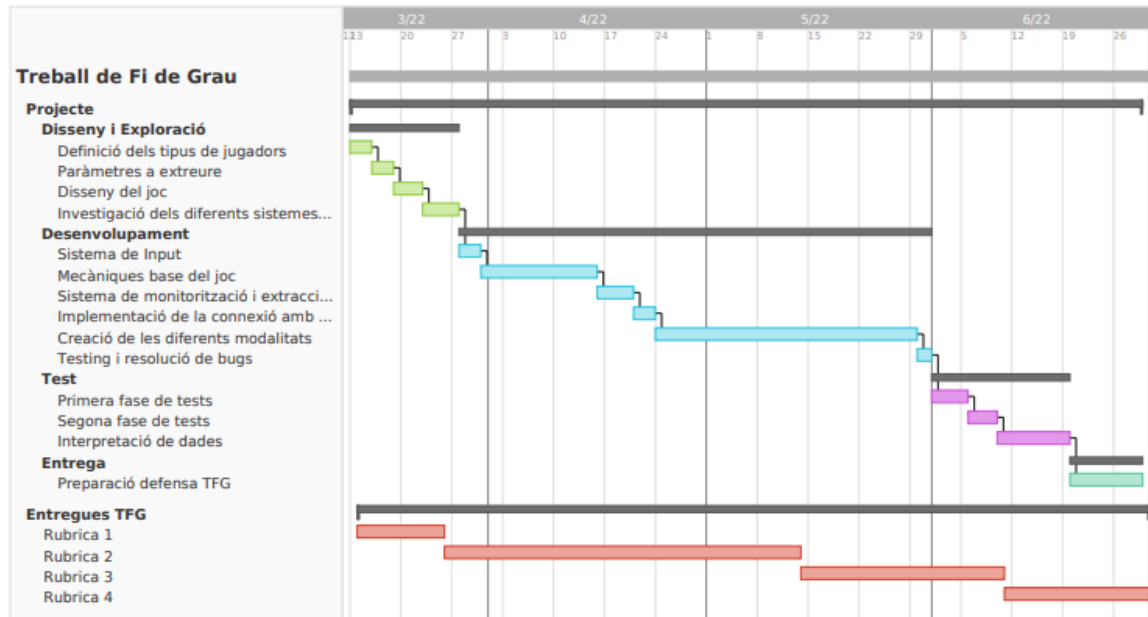


Figura 3.1.1.1: GANTT inicial

Link per veure la imatge en alta resolució:

<https://drive.google.com/file/d/1oEDCOKtJOjSIG4t8ZlIgxZpwC93bwVll/view?usp=sharing>



**Figura 3.1.1.2: GANTT Rúbrica 2**

Link per veure la imatge en alta resolució:

<https://drive.google.com/file/d/1kvFmXagk804xz4B1O-cUUQ1Fw3FeV6do/view?usp=sharing>

El desenvolupament, com es pot veure en la **figura 3.1.1**, és un desenvolupament en cascada. També es mostren les diferents fases del desenvolupament i el temps destinat a cada una.

- **Disseny i Exploració:**
  - Definició dels tipus de jugadors, del 13 al 15 de març.
  - Paràmetres a extreure, del 16 al 18 de març.
  - Disseny del joc, del 19 al 22 de març.
  - Investigació dels diferents sistemes a implementar, del 23 al 27 de març.
- **Desenvolupament:**
  - Sistema de Input, del 28 al 30 de març.
  - Mecàniques base del joc, del 31 de març al 15 d'abril.
  - Monitorització i extracció de dades, del 16 al 20 d'abril.
  - Connexió amb la base de dades, del 21 al 23 d'abril.
  - Creació de les diferents modalitats del joc, del 24 d'abril al 14 de maig.
  - Testing i resolució de bugs, del 15 al 22 de maig.
- **Test:**
  - Primera fase de tests, del 23 al 27 de maig
  - Segona fase de tests, del 28 al 31 de maig.
  - Interpretació de dades del 1 al 10 de juny.



- **Entrega:** (la següent fase esta enfocada a preparar la defensa del tfg pero si es necessita més temps s'agafarà d'aquesta fase)
  - Preparació defensa del TFG del 11 de juny al 30 de juny.

Com a recurs addicional al Gantt s'ha fet una estimació en el qual els mostren els dies necessaris per a cada fase i les hores dedicades.

	Dies	Hores diàries	Hores totals
Disseny i Exploració	14	2	28
Desenvolupament	55	4	220
Test	18	3	54
Entrega	19	1.5	28,5
<b>Total</b>	<b>106</b>	<b>-</b>	<b>330.5</b>

**Taula 3.1:** Estimació total d'hores del projecte

### 3.1.2 Trello

El Trello s'utilitza per a visualitzar el flux del treball<sup>[19]</sup> a més de gestor de tasques., fent servir la metodologia KANBAN durant el projecte. També per veure la capacitat de treball de l'equip, en aquest cas una sola persona, alhora permet mesurar el temps de cada tasca per fer el còmput global a la gràfica de GANTT. Addicionalment veure quan s'ha tardat per realitzar cada fase (Disseny i exploració, Desenvolupament, etc.)

Es distribueix en 4 llistes:

- **OPEN:** on estan totes les tasques creades que encara no s'han començat.
- **IN PROGRESS:** les tasques que estan en desenvolupament.
- **REVIEW:** les tasques que s'han acabat pero s'han de revisar, ja sigui fer una tutoria per veure l'estat de la tasca o fe QA Test si es tracta de codi.
- **DONE:** quan les tasques ja s'han finalitzat i revisat.

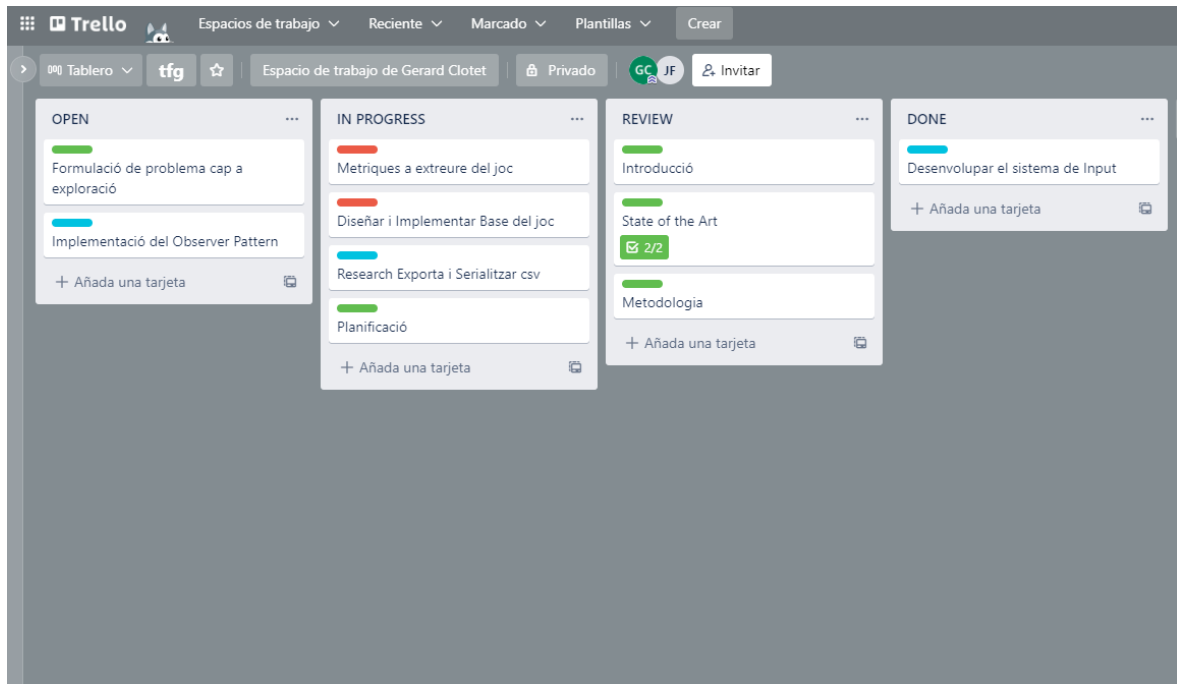


Figura 3.1.2: Trello

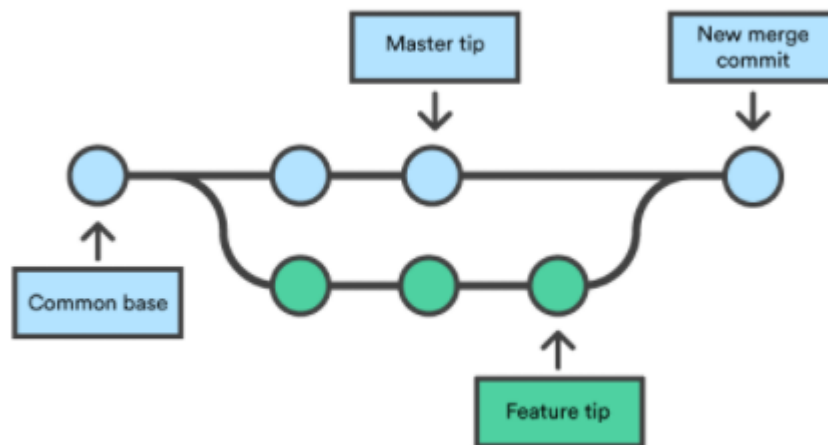
### 3.1.3 GitHub repositori en xarxes, Git eines de control de versions

Link al repositori: <https://github.com/GerardClotet/TFG>

Pel control de versió s'utilitza GitHub on es treballa en diferents branques i permet després unir-les amb seguretat que no es perdin els canvis de cap branca.

A sobre ens permet recuperar canvis anteriors i està integrat amb Unity el que agilitza el procés de desenvolupament.

La funcionalitat serà la següent, es partirà de la branca màster i a mesura que avanci el desenvolupament se n'aniran creant noves branques per a cada funcionalitat. Un cop el desenvolupament de la funcionalitat hagi finalitzat la branca s'aboca a màster. Aquest sistema de branques ajuda a organitzar millor el codi i a reduir els possibles conflictes.



**Figura 3.1.3:** Esquema del funcionament de les branques a Github

### 3.1.4 Playfab

Per a extreure les dades dels usuaris s'utilitza Playfab. Cada usuari puja unes mètriques extretes i el qüestionari posterior. Un cop tots els usuaris hagin fet el PlayTesting l'administrador de la base de dades descarregarà tota la informació extreta per a analitzar-la.

### 3.1.5 Estat de la planificació

En aquesta secció es descriu en detall l'estat del projecte en funció de la seva planificació original i els canvis o modificacions realitzats en cada etapa.

#### Rúbrica 2

Respecte a la primera entrega de desenvolupament prevista el dia 13/05/22 vinculada amb l'entrega de la segona rúbrica de la memòria el desenvolupament del projecte es troba endarrerit respecte a la proposta inicial, a causa de diferents factors.

Respecte a les tasques previstes s'ha assolit crear el sistema d'input sense cap problema. Les mecàniques base del joc, on s'ha invertit la major part del temps de desenvolupament, ja que era necessari iterar i resoldre bugs abans de començar la tasca la creació de les variants del joc base a més durant el desenvolupament es va veure la necessitat de crear un menú per posar en pausa el joc i de fer un sistema de desconnexió de mando, perquè si es desconnecta, pot distorsionar les mètriques que s'extreuen del jugador.

S'ha creat el sistema de monitoratge i extracció de dades on hi ha hagut una ampliació respecte al plantejament inicial, ja que el qüestionari que es fa als usuaris inicialment es conduïa en una altra aplicació externa, però per millorar l'eficiència dels tests i que l'usuari pugui jugar al joc i fer el qüestionari seguidament es decideix implementar un sistema de test dintre de l'aplicació.

Per últim, la connexió amb la base de dades és la segona tasca on s'ha invertit més temps, ja que la previsió i la recerca inicial eren errònies i el servei de base de dades Firebase no funciona de manera correcta amb aplicacions de Unity per a ordinador, si per a mòbil. Es van iterar diferents bases de dades com Unity Cloud Services i finalment s'utilitza Playfab que té una integració molt senzilla amb Unity i funciona correctament amb aplicacions per ordinador.

Estava previst haver finalitzat la tasca de crear les diferents variants del joc base per a cada perfil psicològic de jugador, la qual correspon amb la tasca final d'aquesta entrega.

A causa de diferents variacions i ampliacions de la planificació inicial la data prevista per iniciar aquesta tasca s'ha anat dilatant en el temps i queda posposada per la següent entrega.

Es pot veure a la nova planificació a la figura 3.1.1.2 per a la rúbrica 3.

### Rúbrica 3

En haver-se fusionat la rúbrica 3 i la rúbrica 4 els terminis s'han dilatat.

Respecte a les tasques inicialment plantejades La tasca de creació de diferents modalitats i testing i resolució de bugs s'han fusionat, ja que els bugs s'anaven resolent mentre es feia la tasca de les diferents modalitats. Estava prevista que s'acabessin el dia 30 de maig, però finalment s'ha finalitzat el dia 20 de juny. Aquesta desviació de vint dies es deu a diferents factors.

Es va haver de modificar la forma en la qual el jugador tenia en compte que tocava el terra o les parets, ja que a vegades fallava i no detectava les col·lisions correctament. Aquest va ser un bug on es va invertir bastant de temps.

Una altra tasca on es va invertir un temps considerable va ser a modificar el sistema d'extracció de dades [5.3.1] i a crear el còmput de dades [5.3.2].

Per una altra banda, el disseny dels nivells d'acord amb les diferents modalitats dels jugadors va ocupar un gran temps en l'entrega [5.2.6] a que s'hi havia d'adequar a cada perfil de jugador.

La creació de les diferents modalitats i integrar-les als nivells, elements que no es van tindre en compte com crear i aplicar les animacions a diferents elements del mapa o l'àudio al joc també va suposar una gran part del temps.

Es va crear un sistema de plataformes per donar més dinamisme al joc, cosa que no s'havia fet prèviament

Per altra banda, els tests s'han fet en un espai de temps més reduït, des del dia 21 de juny al 29.

## 3.2 Eines de validació

En ser una investigació exploratòria les eines de validació estan enfocades en extreure els resultats de les partides que facin els usuaris. Les dades estan al Playfab per ser examinades i als usuaris els hi faran tests per poder determinar millor els resultats extrets.

Depenent dels objectius es validen de diferents formes, s'han de validar tres objectius diferents:

### **Definir els paràmetres per catalogar els jugadors:**

- Aquest objectiu és el més arbitrari i subjectiu, la forma per validar-ho és fent Playtesting i qüestionaris a diferents usuaris per veure si les mètriques pensades per catalogar els jugadors són prou diferents entre elles per captar diferències segons l'estil de joc de l'usuari.

### **Desenvolupar el videojoc on fer l'exploració**

- En aquest cas la forma de validar es basarà a fer controls de qualitat, siguin persones externes al projecte o la persona que estigui desenvolupant-lo, a cada implementació nova es farà un QA Test per assegurar el funcionament correcte.

### **Contrastar els Resultats obtinguts amb els esperats**

- Es fa playtesting amb diferents usuaris i s'extreuen les mètriques dissenyades prèviament. A partir d'aquí se'ls hi pregunta als usuaris quins aspectes del joc els hi han agradat més o menys, en forma de test. Un cop tinguem les dues dades les creuem per veure si coincideix el que mostra les mètriques extretes amb el que els usuaris han respost.
- Segons el resultat de l'encreuament de dades, els usuaris jugaran diferents modalitats del joc, on es repeteix el procediment anterior: se'ls hi fa un test per veure el grau de satisfacció respecte al que han jugat, els usuaris juguen la resta de les modalitats del joc per veure si s'ha errat en la predicció inicial i es torna a fer un test per comprovar la satisfacció de l'usuari.
- Un cop tinguem tots els tests dels usuaris guardats a la base de dades, es busca alguna correlació entre diferents usuaris per veure si realment existeix alguna connexió entre les dades extretes i les preferències dels usuaris.

### 3.3 DAFO

	Positius	Negatius
Origen Intern	<b>Fortaleses</b> - Coneixement de programació - Interès en el tema	<b>Debilitats</b> - No ni ha informació sobre el tema. - Poc temps degut a feina de jornada completa (40 hores/setmana) - No es domina la part de la psicologia del jugador
Origen Extern	<b>Oportunitats</b> - Segons el que s'ha pogut veure ningú ha explorat el tema, no hi ha competència	<b>Amenaces</b> - Treball de una sola persona, un grup amb més recursos pot fer una millor exploració i validació

Taula 3.3: DAFO

### 3.4 Riscos i pla de contingències

Risc	Solució
Bugs i errors	Deixar un marge de dues setmanes abans de cada entregar per buscar errors i arreglar-los.
Falta de temps	Prioritzar les funcionalitats principals, fragmentar tasques per tenir una idea més clara del temps necessari. Òptima gestió del projecte
Caiguda dels servidors de Firebase	Migrar a una nova base de dades
Falta de coneixement alhora d'implementar diferents sistemes com el firebase o la psicologia del jugador	Buscar estudiar i analitzar la informació sobre el tema

Taula 3.4: Riscos i pla de contingències

### 3.5 Anàlisi Inicial de Costos

Per poder calcular el cost que suposaria un projecte d'aquest tipus s'han tingut en compte diferents factors:

- El projecte està desenvolupat per una sola persona a mitja jornada i el sou brut s'ha estimat a part del sou base d'un programador junior a Catalunya[20].
- En la depreciació de l'equipament s'ha de tenir en compte que són 4 mesos de desenvolupament.
- Els usuaris als quals se'ls fa el test són voluntaris, per tant, de franc.
- El rol d'aquest treball es representa com a programador

Com es pot veure a la taula de sota el cost estimat total del projecte arriba a 3.096.00 €.

El salari es calcula a partir de les hores estimades al apartat 3.1.1, en el qual s'imposa un salari de 8 € l'hora.

	Març	Abril	Maig	Juny	Total
<b>Costos Directes</b>					
<b>Personal</b>	352.00€	960.00€	896.00€	480.00€	<b>2.661.00€</b>
Programador	352.00€	960.00€	896.00€	480.00€	<b>2.661.00€</b>
<b>Software</b>	-€	-€	-€	-€	<b>-€</b>
Visual Studio	-€	-€	-€	-€	<b>-€</b>
Unity	-€	-€	-€	-€	<b>-€</b>
Firebase	-€	-€	-€	-€	<b>-€</b>
<b>Depreciació equipament</b>	32.00€	32.00€	32.00€	32.00€	<b>128.00€</b>
Ordinador	22.00€	22.00€	22.00€	22.00€	<b>88.00€</b>
Pantalla + Ratolí + teclat	5.00€	5.00€	5.00€	5.00€	<b>20.00€</b>
Taula + Cadira	5.00€	5.00€	5.00€	5.00€	<b>20.00€</b>
<b>Total Costos directes</b>	<b>384.00€</b>	<b>992.00€</b>	<b>928.00€</b>	<b>512.00€</b>	<b>2.789.00€</b>
<b>Costos Indirectes</b>					
Electricitat	40.00€	40.00€	40.00€	40.00€	<b>160.00€</b>

Internet	30.00€	30.00€	30.00€	30.00€	<b>120.00€</b>
Total Costos indirectes	70.00€	70.00€	70.00€	70.00€	<b>160.00€</b>
Cost Total	454.00€	1.062.00€	998.00€	582.00€	<b>3.096.00€</b>

**Taula 3.5:** Anàlisi inicial de costos



## 4. Metodologia

El projecte s'enfoca com a una investigació exploratòria, on a partir d'un cert desconeixement de la matèria en qüestió, la psicologia del jugador, només es pretén conèixer les característiques més rellevants dels problemes a desenvolupar.

### 4.1 Tipus de Metodologia

El projecte consta de varies fases i cada fase té una metodologia diferent.

#### **Disseny i Exploració:**

On es dona forma al projecte, creant el document que consta de les següents parts:

- Descriure l'exploració que es portarà a terme en relació amb l'usuari
- Dissenyar el joc i les dades que s'utilitzen per classificar a l'usuari
- Determinar quins tipus de jugadors es comprenen segons les mètriques.
- Explicar les tecnologies que es faran servir per dur-ho a terme.
- Fonamentar els tests per validar i treure conclusions dels resultats posteriors

Segons Juan Ortiz<sup>[19]</sup> la investigació exploratòria defineix el "mètode secundari" on en aquest cas l'investigador se serveix de treballs anteriors en el mateix àmbit per ampliar el seu coneixement i context d'una matèria que no domina.

En aquest projecte s'aplica a la psicologia del jugador, però més que per expandir el coneixement és per orientar la pròpia taxonomia que es pretén explorar, ja que parteix de la de Bartle.

#### **Desenvolupament:**

Es desenvoluparà el videojoc base i les seves variants en funció del dissenyat a la primera fase i implementant les tecnologies pertinents.

L'objectiu d'aquesta fase és tindre el joc finalitzat amb els sistemes per extreure dades i enviar-les al Playfab.

Un cop el joc base estigui complet, el desenvolupament de les ramificacions dels jocs es farà amb un desenvolupament iteratiu incremental on segons el feedback rebut a cada entrega i en tests a usuaris es modifica, millora i poleix diferents aspectes del joc, principalment l'optimització i bugs varis que sorgeixen i les diferents mecàniques del joc perquè sigui més atractiu de cara als usuaris que el jugaran a la següent fase.

Per al seguiment del desenvolupament s'utilitzarà Trello per veure el flux del treball i TeamGantt per fer un seguiment més a gran escala.

El contenidor tant del codi com dels fitxers del projecte està a GitHub, així es podrà controlar les diferents versions del projecte i el seu desenvolupament pas a pas.

### Test:

En aquesta fase s'utilitza el mètode primari de la investigació exploratòria<sup>[21]</sup> on es busquen proves a través de l'observació, en aquest cas els tests que se'ls hi fa als usuaris i les mètriques que s'extreuen.

#### Procediment de l'exploració

1. Els usuaris juguen al tutorial i es recullen les dades sobre com han jugat
2. En finalitzar es fa un breu qüestionari als usuaris sobre la seva experiència i els seus gustos, partint del qüestionari de Bartle, però acomodat al gènere Platfomer
3. Segons el resultat del test juguen la modalitat que més s'adeqüi al seu tipus de jugador, és un procés automatitzat
4. Mentre juguen preguntar a veure si la seva experiència de joc ha augmentat respecte al tutorial
5. Que els jugadors juguin a les altres modalitats del joc.
6. Passar un test final per veure quina modalitat els hi ha agradat més.

Un cop s'hagi fet el procés es passa a l'etapa de validació i l'anàlisi dels resultats obtinguts amb els esperats per treure una conclusió de si efectivament hi ha alguna relació entre el tipus de jugador el qual hem assignat als usuaris i la millora amb l'experiència de joc segons el tipus de modalitat que han jugat.

## 4.2 Estat de la metodologia

En aquesta secció es descriu en detall l'estat de la metodologia en funció de la proposta inicial i el possibles canvis o modificacions realitzats en cada etapa o entrega.

### Rúbrica 2

La segona rúbrica comprèn les fases de disseny i exploració i desenvolupament.

En quant a la fase de disseny i exploració la metodologia de investigació exploratòria s'ha mantingut constant.

Per una altra banda, la fase de desenvolupament ha seguit el mètode de desenvolupament iteratiu incremental. S'han patit retards però, no ha modificat el mètode de treball de fer tasques segons l'ordre de prioritats i un cop finalitzades comprovar que el resultat final és satisfactori. Es pot donar el cas que la tasca s'hagi de reobrir ja que hi hagi hagut alguna desviació segons la planificació inicial, tot i això el flux de treball no es veu alterat.

### **Rúbrica 3**

La tercera rúbrica comprèn les fases de desenvolupament i test.

Quant a la fase de desenvolupament s'ha mantingut igual que a la rúbrica anterior, desenvolupament iteratiu incremental. S'han patit variacions, però no han modificat el mètode de desenvolupament.

A la fase de test s'ha fet exactament igual, amb una única variació. En comptes d'haver-hi un qüestionari al final de primer nivell i un qüestionari al final del joc, s'ha fet un petit qüestionari al final de cada nivell per saber en calent les seves reaccions sobre la modalitat que han jugat, obtenint d'aquesta forma més informació.

## 5. Desenvolupament

Com s'ha comentat en els anteriors apartats, el desenvolupament del projecte està dividit en diferents fases. Els quals són descrits a la secció [\[3.1.1\]](#) on es pot veure visualment al diagrama de Gantt i s'explica amb detall.

### 5.1 Disseny i Exploració

En aquest apartat es crea una taxonomia de jugadors de plataformes a partir d'una exploració subjectiva, és a dir a través d'experiències pròpies. L'objectiu final és comprovar si aquesta experiència personal és aplicable a més jugadors.

#### 5.1.1 Taxonomia de Jugadors

Partint de la Taxonomia de Bartle es defineixen quatre tipus de perfils de comportament del jugador aplicats a un videojoc de plataformes.

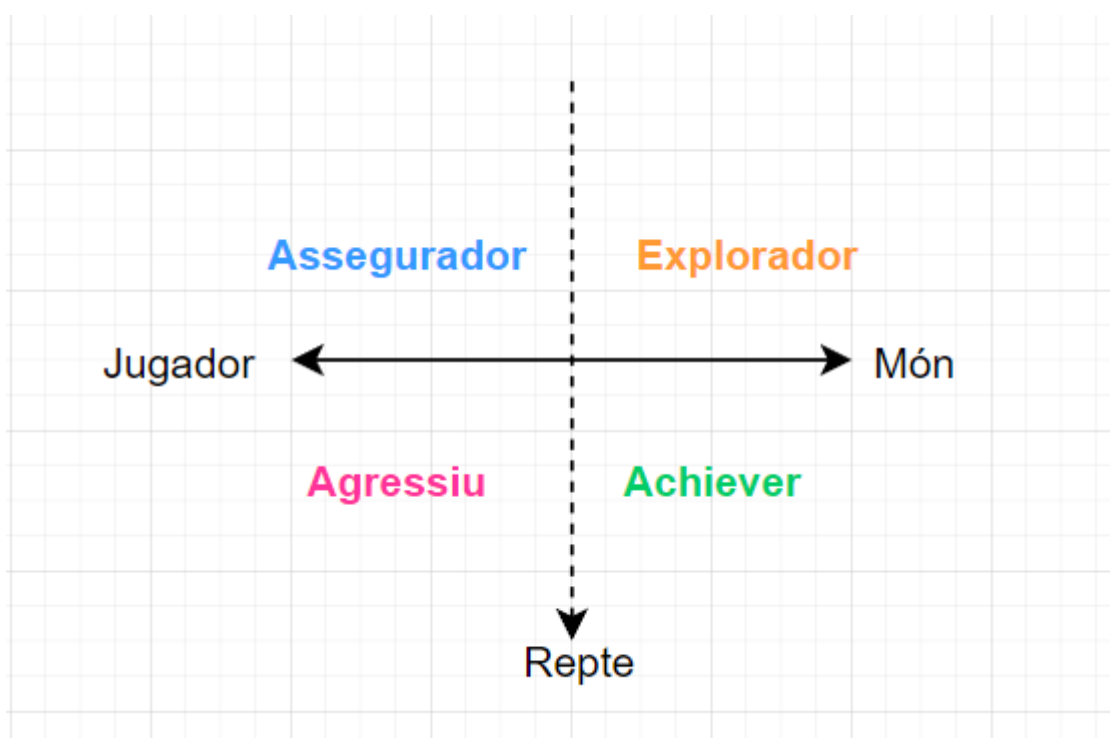


Figura 5.1.1.1: Tipologia de jugadors per a platformers

Respecte a Bartle sorgeixen dos perfils de jugadors diferents, Assegurador i Agressiu. Al contrari que la tipologia de Bartle, és impossible que sorgeixi el tipus de jugador social, ja que no hi ha cap forma d'interactuar amb ningú. Per les mecàniques donades es canvien el

tipus de jugadors *Killer* i *Socializer* relacionats amb elements multijugador per la forma en què l'usuari afronta els reptes del joc.

Definició dels 4 tipus de jugadors:

**Agressiu:** es caracteritza per voler avançar en el nivell com més ràpid possible, el motiva el repte de superació i la repetició de nivells. Aconseguir els elements secundaris que presenta el joc també ho visualitza com a repte, tendeix a ser *achiever*.

**Assegurador:** contrari a l'agressiu, planteja els nivells com a puzles per resoldre. Estudia els possibles camins i només quan està segur de quin és el camí òptim per superar el nivell es comença a moure. Tendeix cap a l'exploració

**Explorador:** vol coneixer tot els secrets del joc, del món, les històries, etc. El goig sorgeix d'endinsar-se en l'univers del joc.

**Achiever:** vol assolir la màxima puntuació, obtenir col·leccionables difícils d'obtenir. El goig sorgeix completar els objectius opcionals.

En aquesta taxonomia es diferencien dos tipus de rols: els principals i els secundaris.

Els rols principals Assegurador i Agressiu són mútuament excloents. Un Assegurador no pot ser alhora un Agressiu. Es pot donar el cas durant el transcurs del joc o en un altre joc que l'usuari faci una transició de rol principal. Per exemple degut a la masterització de les mecàniques del joc, on visualitzar el nivell com a un puzle a resoldre perd atractiu, ja que masteritzant les mecàniques del joc el jugador pot permetre's corregir errors enmig de l'execució del nivell.

Els rols secundaris, explorador i *achiever* es poden manifestar d'una forma més o menys intensa en els jugadors. No són mútuament excloents, un jugador pot tenir característiques dels dos perfils. Els *killers* tendeixen a aconseguir els col·leccionables que estan al camí principal, ho veuen com un repte extra. En canvi, els asseguradors tendeixen més a explorar, ja que és un element intrínsec a la seva forma de superar els nivells, en intentar trobar la ruta òptima és més probable que trobin camins secundaris o secrets ocults.

A la figura 5.1.1.1 es veuen 3 vectors que defineixen cada jugador:

- **Món:** l'espai on es desenvolupa el joc que està més lligat amb l'achiever i l'explorador.
- **Jugador:** determina els rols principals segons el goig de superar els reptes que planteja el joc.
- **Repte:** la dificultat autoimposada pel mateix usuari. Com més gran vol que sigui el repte a superar intenta aconseguir els col·leccionables i juga d'una forma més agressiva.

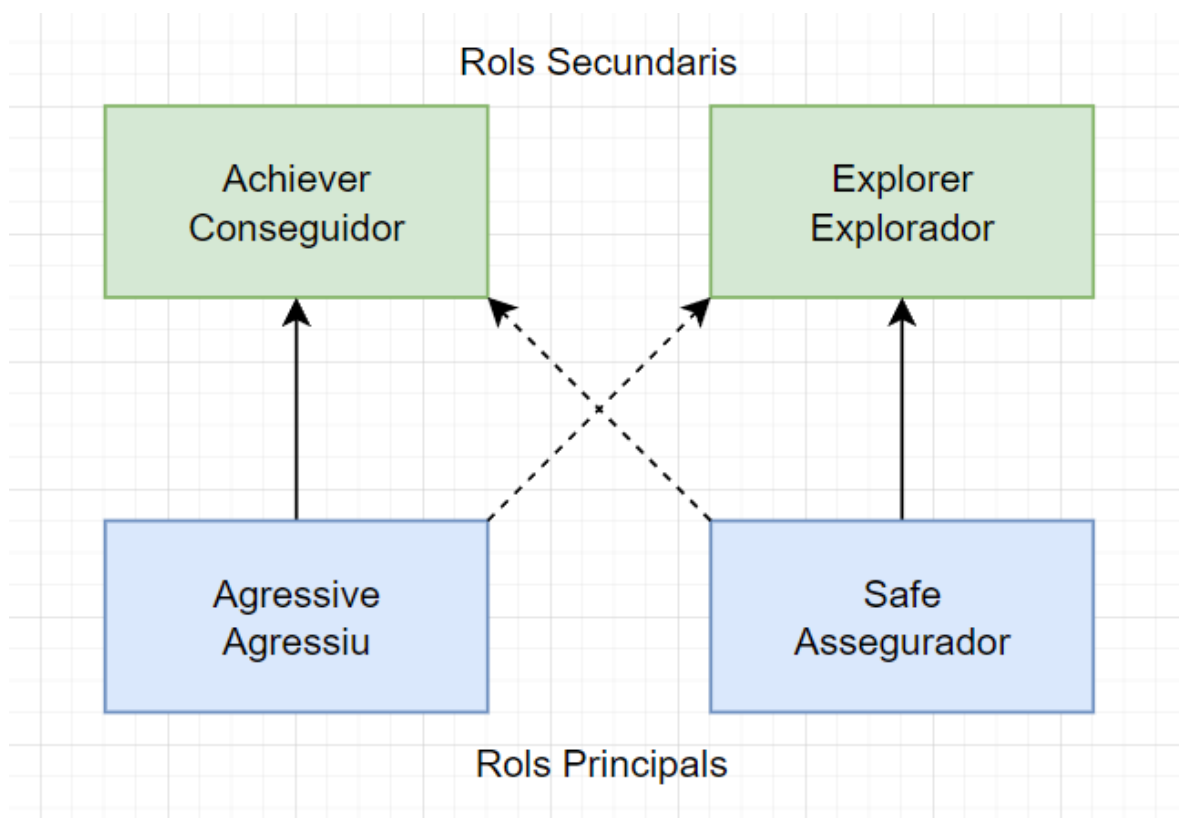


Figura 5.1.1.2.: Connexions entre rols principals i secundaris

## 5.1.2 Mètriques per diferenciar als jugadors

Cada perfil de jugador té unes característiques pròpies que el fan reconeixible a un nivell estadístic, necessari conèixer per poder analitzar el playtesting a les fases finals del projecte.

**Agressiu:** mor moltes vegades, des que reapareix fins que completa el nivell passa poc temps comparat amb l'assegurador. Es passa el nivell saltant menys vegades o utilitzen menys habilitats, però si es fa una suma total de cada vegada que ha fet accions com saltar fins que ha aconseguit passar-se el nivell supera l'assegurador.

**Assegurador:** tarda més temps des de que reapareix fins que acaba el nivell, fa servir més habilitats i salts per cada vida. Sol tornar cap enrere...

**Achiever:** tendeix a obtenir tots els col·leccionables, increment de morts, és a dir d'intents fallits en les zones on hi ha col·leccionables.

**Explorador:** en aquest cas l'explorador depèn de la magnitud amb la qual el desenvolupador generi un univers propi prou ric. Es pot saber quan el jugador explora àrees opcionals sent la recompensa més coneixement sobre el món que envolta el joc.

## 5.1.3 Elements per acomodar l'usuari al seu perfil de jugador

Aquesta secció és una part central del projecte, quins elements acomoden el joc als diferents tipus de jugadors.

**Mecàniques core:** mecàniques que al ser canviades modifiquem per complet l'essència del joc o com es sent. Aquestes han de ser constants i no poden ser modificades o eliminades.

**Mecàniques circumstancials:** mecàniques que en ser modificades no canvien com se sent o el gènere del joc per exemple, treure o posar multiplicadors.

**Agressiu:** es necessiten elements que empenyin el jugador a anar més de pressa, a augmentar el repte. Fer que la càmera es mogui constantment en direcció al final del nivell, si l'extrem oposat a la direcció on es mou la càmera toca el jugador es mor.

**Assegurador:** Crear més camins diferents per resoldre el nivell, l'increment de diferents possibilitats per superar el nivell provocarà una reacció positiva cap al jugador.  
augmentar el nombre de plataformes.

**Achiever:** augmentar el número de col·leccionables que aconseguir i la dificultat per obtenir'ls.

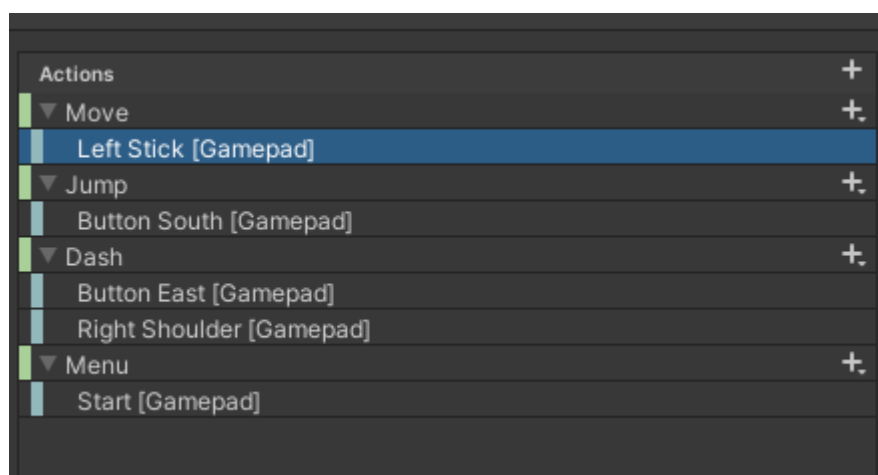
**Explorador:** crear més zones secundàries, afegir text. Per falta de temps segurament aquest tipus de perfil de jugador no sorgirà, ja que requereix un mínim de worldbuilding per part del desenvolupador.

## 5.2 Desenvolupament del Videojoc Base

El projecte es desenvolupa amb el motor gràfic Unity per dues raons principals. És una eina molt estandarditzada, hi ha molta informació i és simple d'utilitzar l'altre motiu és la comoditat, ja que en projectes anteriors ja s'ha fet servir i la familiaritat agilitza els temps de desenvolupament.

### 5.2.1 Sistema de Input

Com s'ha comentat a l'Estat de l'Art s'utilitza el Sistema d'Input nou de Unity<sup>[18]</sup>. Amb aquest sistema es genera un script a través de l'editor d'accions<sup>[22]</sup> que ens ofereix el sistema d'Input. Es creen dos mapes d'accions, un que recull les accions que pot fer l'usuari durant el joc com moure's, saltar, obrir el menú, etc. A cada acció se li assigna un botó del controlador, veure figura 5.2.1.1



**Figura 5.2.1.1:** Mapa d'accions del jugador

Es crea un segon mapa d'acció enfocat a què l'usuari pugui navegar i moure's per la UI (Interfície d'Usuari), tant amb el teclat i ratolí com amb el controlador que utilitza per jugar.



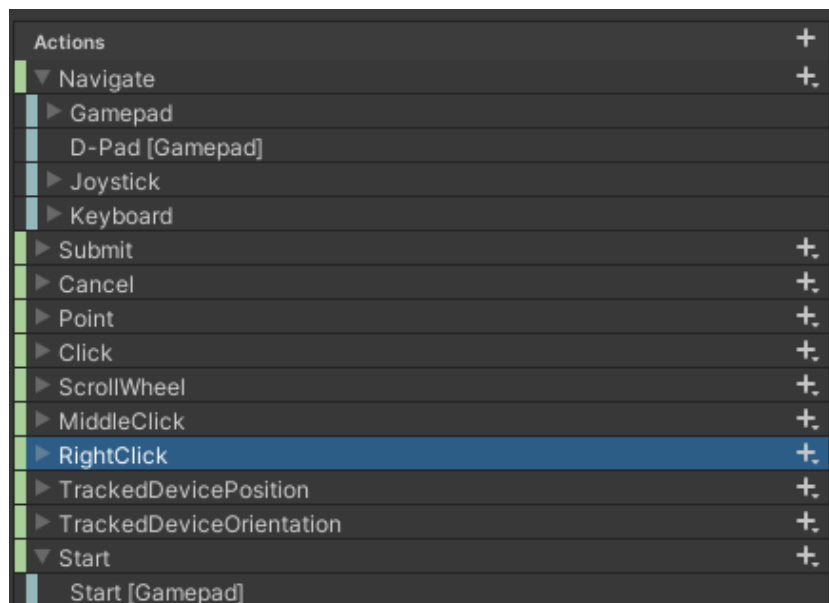


Figura 5.2.1.2: Mapa d'accions de l'interfície d'usuari

Els dos mapes generen automàticament un script amb diferents events<sup>[23]</sup> segons el mapa d'accions. Els events es criden quan l'usuari prem un botó al mando que estigui registrat al mapa d'events.

El sistema d'input també permet habilitar i deshabilitar els diferents mapes d'acció, per exemple quan l'usuari finalitza el joc i passa a fer el test dintre de l'aplicació. Les accions que mouen al jugador com saltar es deshabiliten automàticament.

## 5.2.2 Mecàniques Bàsiques del joc

El loop bàsic del jugador es divideix principalment en tres parts:

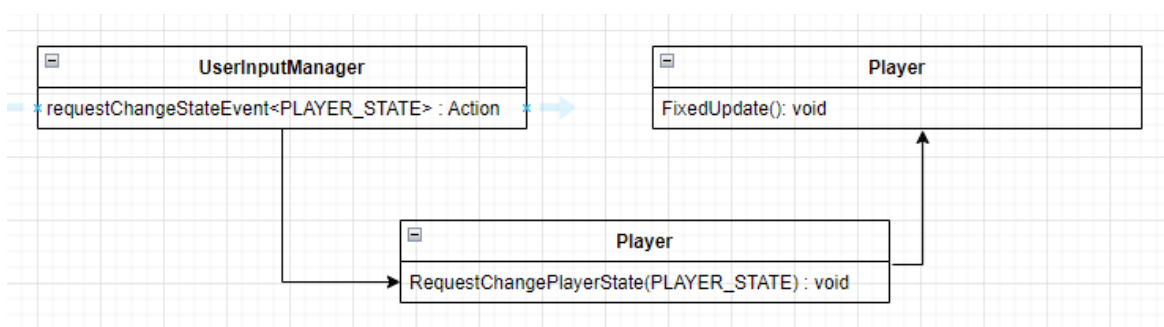


Figura 5.2.2.1: Loop general del jugador

L'input de l'usuari es gestiona a part en el script **UserInputManager** per poder utilitzar de forma més eficient i neta l'*Observer Pattern*<sup>[15]</sup> a l'hora d'extreure dades del jugador, com s'explica a l'apartat [5.4].

Al *UserInputManager* quasi totes les accions del jugador invoquen el mateix event, ***requestChangeEvent***, segons l'acció que l'usuari faci es passa un paràmetre diferent. Per exemple si el jugador salta el paràmetre que es passa a la funció serà el de *PLAYER\_STATE.JUMP* en la figura 5.2.2.2 es poden veure tots els estats que pot tindre el jugador.

```
public enum PLAYER_STATE
{
    IDLE,
    MOVE,
    JUMP,
    HOLD_DASH,
    DASH,
    ON_AIR_DASH,
    ON_AIR,
    GRAB_WALL,
    BOUNCE,
    BOUNCE_AIR
}
```

**Figura 5.2.2.2:** Diferents estats del jugador

Un cop s'ha invocat l'event, aquest crida a la funció ***RequestChangePlayerState***, comprova si és possible o no que el jugador amb el seu estat actual pugui canviar a l'estat que està rebent des de l'*InputManager*.

Per una altra banda a la funció ***FixedUpdate***, una funció pròpia de Unity que està lligada al sistema de físiques del motor. S'ha d'emprar quan s'utilitzen components del sistema de físiques de Unity, ja que la seva freqüència d'execució és diferent de la resta de la lògica del joc.

En aquesta funció s'aplica diferents forces o velocitats al jugador segon el seu estat actual, s'explica amb més detall als següents apartats.

A la figura 5.2.2.3 es pot veure un esquema del sistema d'estats del jugador.

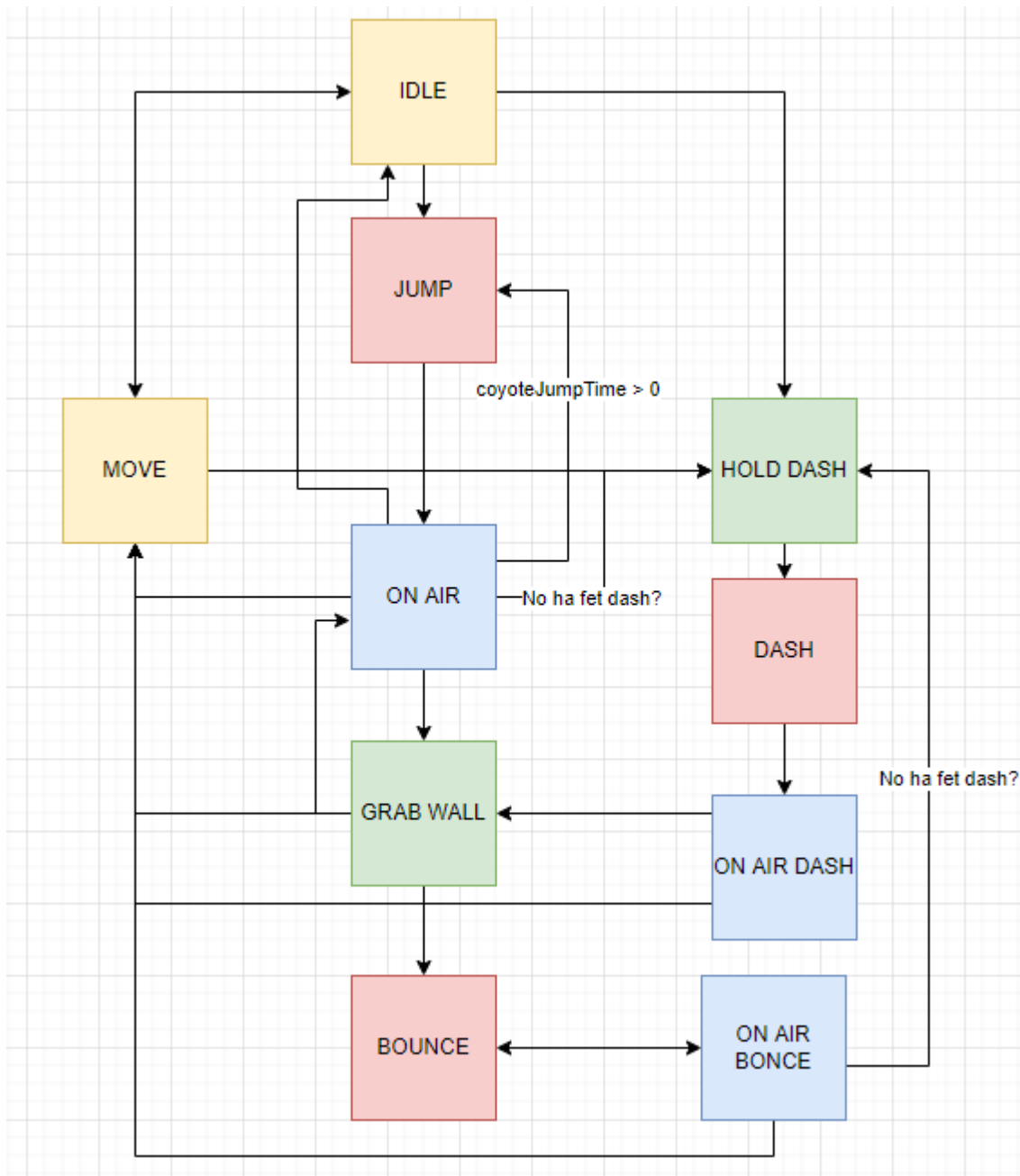


Figura 5.2.2.3: Flux dels diferents estats del jugador

## Moviment

El moviment del jugador ve determinat per la direcció del joystick. Des de el mapa d'accions s'invoca un event on es passa la direcció del joystick en un vector de dues dimensions.

```
2 referencias
public void OnMoveInput(float x, float y)
{
    leftStick.x = x;
    leftStick.y = y;
}
```

**Figura 5.2.2.4:** funció que rep el valor del joystick

El valor `leftStick.x` es el valor de la posició horitzontal del joystick respecte al seu centre. Determina en quina direcció (esquerra o dreta) es desplaça el personatge que controla el jugador.

El valor `leftStick.y` és el valor de la posició vertical del joystick respecte al centre. Aquest valor només es té en compte a l'hora de fer un dash, s'explica a l'apartat de dash.

Les funcions de moviment del jugador es criden quan el jugador té l'estat de MOVE o un dels tres estats quan està a l'aire: ON\_AIR, ON\_BOUNCE\_AIR o ON\_AIR\_DASH.

En comptes de crear només un estat per a quan el jugador està al aire se'n creen tres de diferents, ja que així es pot aconseguir crear diferents efectes segons l'acció feta per al jugador, com modificar la fricció del aire o que perdi més velocitat en un eix en concret.

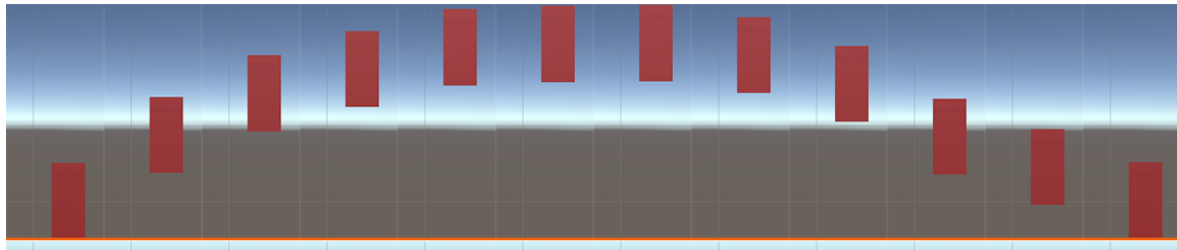
## Salt

Funcionament del salt, depèn del temps que el jugador premi el botó de salt el personatge salta més o menys, així té una sensació de control en tot moment.

Un cop ha deixat de prémer el botó o ha passat  $x$  temps hi ha una força contrària al salt per suavitzar la paràbola que es fa<sup>[24]</sup>.

Disseccionant el salt es veu que:

- El salt té una duració de 30 fotogrames,
- Al fotograma 17 arriba a la alçada màxima
- Al frame 20 comença a baixar.
- L'alçada del salt és 3 vegades l'alçada del jugador.



**Figura 5.2.2.5:** Alçada del salt segons el frame

Quan el jugador cau de una plataforma, té una fracció de segon per poder saltar mentres està al aire, a aquest tipus de salt se l'anomena *coyote jump*. D'aquesta forma no penalitza tant als jugadors que el personatge tingui un moviment molt ràpid.

### Salt per les parets

Quan el jugador està en contacte amb una paret, si prem el botó de saltar en comptes de fer un salt normal, rebotarà de la paret en direcció contrària. El jugador no pot decidir ni l'angle ni la força del rebot, però sí que pot controlar la velocitat horitzontal del jugador un cop ha saltat.

Aquesta mecànica és simple d'implementar i ofereix molta verticalitat a l'hora de dissenyar un nivell.

### Dash

El dash és un impuls que pot fer el jugador en qualsevol direcció on apunti amb el joystick, ja sigui saltant o desde el terra. Fins que el jugador no torni a tocar el terra no podrà tornar a utilitzar-lo.

El dash es divideix en 2 fases:

- Quan el jugador prem el botó perd tota inèrcia i gravetat durant una fracció de segon, durant aquest temps el jugador apunta a la direcció on vol anar.
- El jugador surt disparat en aquella direcció i va perdent velocitat gradualment.

## 5.2.3 Cinemachine

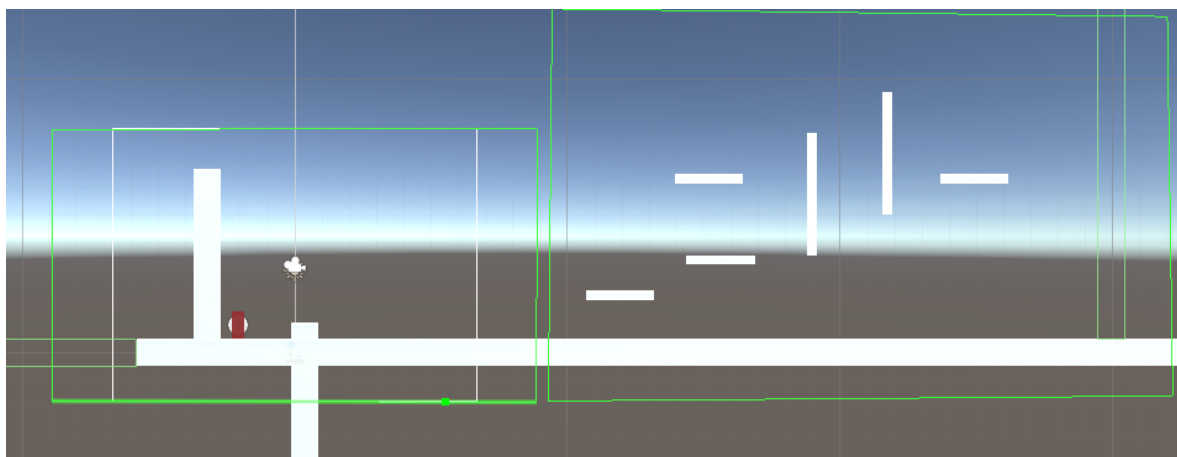
S'utilitza el package Cinemachine<sup>[25]</sup>, és un editor de càmeres de Unity que permet dissenyar diferents tipus de càmeres per al joc i editar-les de manera ràpida i senzilla. En el projecte es fa servir el cinemachine per dues funcionalitats principals.

La primera és el component de càmeres virtuals<sup>[26]</sup>: són varies càmeres amb diferents configuracions repartides per l'escena del joc que estan desactivades o en repòs, fins que

l'aplicació les activa. Quan una càmera és activada el que veu l'usuari a l'aplicació és el que enfoca aquella càmera.

En aquest joc hi ha una càmera virtual per cada habitació. Cada nivell pot tindre x habitacions, que consisteixen en espais on la càmera segueix al jugador fins al límit d'aquella habitació. Quan el jugador canvia d'habitació hi ha un script que detecta quan marxa o entra en una habitació i el que fa és canviar la càmera virtual activa. Provocant una transició de les càmeres.

A la figura 5.2.3 es veu la representació de dues habitacions delimitades pels rectangles verds. Quan el jugador toca el rectangle verd s'activa o es desactiva la càmera virtual d'aquella habitació.



**Figura 5.2.3:** Sistema d'habitacions de l'escena

La segona funcionalitat s'explica a l'apartat de Game Feel.

### 5.2.4 Game Feel

El game feel<sup>[27]</sup> són les diferents respostes sensorials que ajuden al jugador a transportar-se al món del videojoc, a fer-lo més creïble.

La primera resposta sensorial és a través del Cinemachine. S'utilitza la vibració de la càmera o el camera shake. Les càmeres virtuals tenen un apartat anomenat Noise on a la càmera se li dona una freqüència i amplitud de translació i rotació. S'usa per simular que les accions del jugador tenen un impacte al món. En aquest cas la càmera vibra quan el jugador fa un dash.

A la figura 5.2.4 es poden veure els valors que permeten fer vibrar la càmera de diferents formes, en aquest cas es fan servir dos elements l'amplitud i la freqüència. La freqüència fa

referència a la velocitat amb la qual vibra la càmera i l'amplitud la distància de translació i rotació de la càmera per simular la vibració.



**Figura 5.2.4:** valors del camera shake

La funció que s'ha creat és que durant un temps donat l'amplitud passi d'un valor X a zero.

La segona forma d'apropar el jugador al joc és a través del tacte, de la vibració del controlador.

El sistema d'input que s'utilitza té una funció específica per fer vibrar els dos motors dels controladors a una freqüència desitjada.

S'han creat dues funcions diferents, una per fer vibrar el controlador a la freqüència desitjada de forma constant durant un cert temps. Emprada per a quan el jugador toca el terra.

L'altra funció la freqüència de vibració varia de forma lineal durant X temps entre dos valors donats. S'usa quan el jugador fa un dash, segons Steve Swink<sup>[27]</sup> aconseguint així que el món digital tingui un pes físic.

## Animacions

S'han animat diferents elements del mapa com col·leccionables o els diferents moviments del jugador. Com per exemple quan toca el terra, quan salta, fa un dash, també s'han animat diferents objectes per ressaltar-los de cara al jugador.

Aconseguint així que el jugador noti el pes de les accions en moure's per l'entorn.

## Audio

L'àudio és essencial a l'hora d'entendre els moviments que fa el jugador i rebre feedback, a través de l'àudio l'usuari és molt més proper a entendre que està passant en pantalla. Per això cada moviment del jugador i acció possible té un soroll diferent assignat

## 5.2.5 Elements del mapa

### Plataformes

Per donar més dinamisme als nivells s'ha creat un sistema de plataformes que se subdivideix en tres categories:

#### **Plataformes de moviment perpetu**

Quan el jugador entra dintre de la room del nivell comencen a anar i tornar, l'usuari pot decidir en quina direcció és molt si vertical o horitzontal, la seva posició d'origen i de destí i la velocitat amb la qual es mou.

#### **Plataformes de moviment amb contacte**

Aquestes plataformes tenen la mateixa funcionalitat que les altres, però només es mouen quan el jugador entra amb contacte amb elles, fan el seu recorregut d'anada i tornada i es tornen a activar quan el jugador les torna a tocar.

Aquests dos tipus de plataforma tenen una cosa més en comú i és que quan el jugador salta estant les plataformes en moviment el jugador surt disparat segons la velocitat que les plataformes tinguin, donant més dinamisme i possibilitats al nivell.

#### **Plataformes trencables**

Són les plataformes més diferents de la resta, les seves característiques són que si el jugador les toca al cap de  $x$  segons es trenquen i tarden uns altres temps donat en tenir una estructura sòlida. Perquè el jugador s'assabenti del funcionament d'aquest tipus de plataforma, quan el jugador està a sobre comença a vibrar i a dependre partícules, simulant així que s'està a punt de trencar

Amb aquest tipus de plataformes es pot forçar al jugador a tindre una actitud més agressiva.

### Objectes recol·lectables

Els nivells tenen diferents tipus d'objectes els quals es poden agafar.

#### **Col·leccionables**

Són objectes que representen un repte d'aconseguir pel jugador. La seva funcionalitat és simple: estan en algun lloc de la room i el jugador els ha d'agafar i portar-los fins al final de l'habitació. Si el jugador mor mentre té el col·leccionable aquest torna al seu lloc d'origen.

#### **Reset del Dash**

Són objectes estàtics repartits pel mapa que quan el jugador les toques permeten al jugador fer un altre dash a l'aire sense necessitat de tocar el terra. Aquests tenen un temps  $X$  de cooldown durant el qual si el jugador les toques no tindran cap efecte. Visualment, l'objecte



canvia de color segons l'estat i emet partícules mentre està en cooldown, perquè el jugador ho pugui diferenciar.

### **Claus**

Hi ha un tercer tipus d'objecte recol·lectable que funciona igual que els col·leccionables. El jugador l'agafa i si mor torna al seu lloc inicial. Quan el jugador surt de l'habitació la clau obre una porta que impedeix el pas del jugador pel nivell.

### **Morts**

S'ha creat un sistema que si el jugador és tocat per un collider de tipus mort el nivell es reinicia i el jugador torna al punt de partida. Quan el jugador toca un d'aquest colliders es deshabilita l'input de l'usuari i s'atura el joc, fa un fos a negre i el jugador fa un traveling cap a la posició inicial.

En el cas que el jugador tingui el perfil agressiu, no es fa el fos a negre sinó que directament apareix al punt de partida, donant més dinamisme al joc, element que busca un perfil de jugador agressiu.

### **Level Eater**

Al nivell agressiu per donar més urgència al jugador hi ha un rectangle negre que ocupa tota l'habitació i persegueix el jugador fins al final del nivell amb una velocitat donada per l'usuari.

### **Colliders de mort**

Hi ha punxes repartides pel mapa, si el jugador els toca mort i es reinicia el nivell.

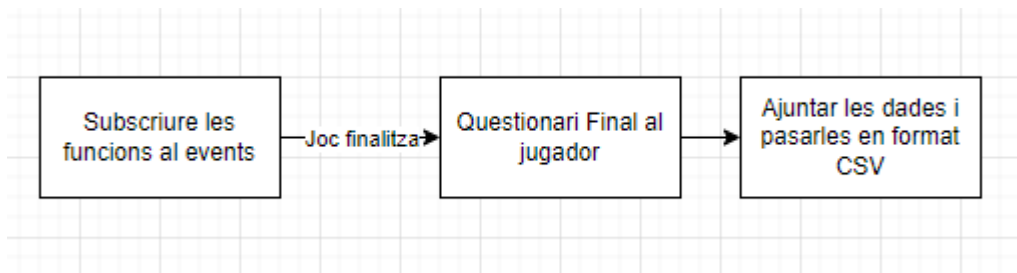
## **5.2.6 Nivells**

El joc es compon de tres nivells. L'inicial que és un nivell mixt on es busca determinar quin perfil de jugador té l'usuari i els altres són per confirmar si la predicció feta al còmput de dades [5.3.2]. Si la predicció feta coincideix amb el tipus de jugador, es veu reflectit en les dades que s'extreuen i confirmat al qüestionari al final de cada nivell.

## 5.3 Monitoratge i Extracció de dades

A l'aplicació el monitoratge i l'extracció de dades consta de tres fases.

Durant el joc de forma indirecta s'extreuen dades relacionades amb les accions de l'usuari. Quan el joc finalitza s'extreuen dades directament a través d'un test al jugador. La fase final consisteix a ajuntar totes les dades i passar-les en format CSV per posteriorment enviar-les a la base de dades.



**Figura 5.3.1:** fases del monitoratge i l'extracció de dades

Indirectament, s'extreuen les dades de l'usuari mentre juga, quan ell no n'és conscient. A través del patró observer el qual s'explica a l'apartat [2.2.2] des del script ReportGatherer se subscriuen diferents funcions a events que passen al joc, com el número de vegades que el jugador mor, salta o fa un dash. Aquestes funcions augmenten un comptador que en finalitzar el joc es posen en format de text o string.

Per a la segona fase de monitoratge de dades s'ha utilitzat una tecnologia anomenada LeenTwean<sup>[28]</sup>, utilitzada per animar la interfície d'usuari de manera més eficient i fàcil d'utilitzar a l'hora de fer el qüestionari.

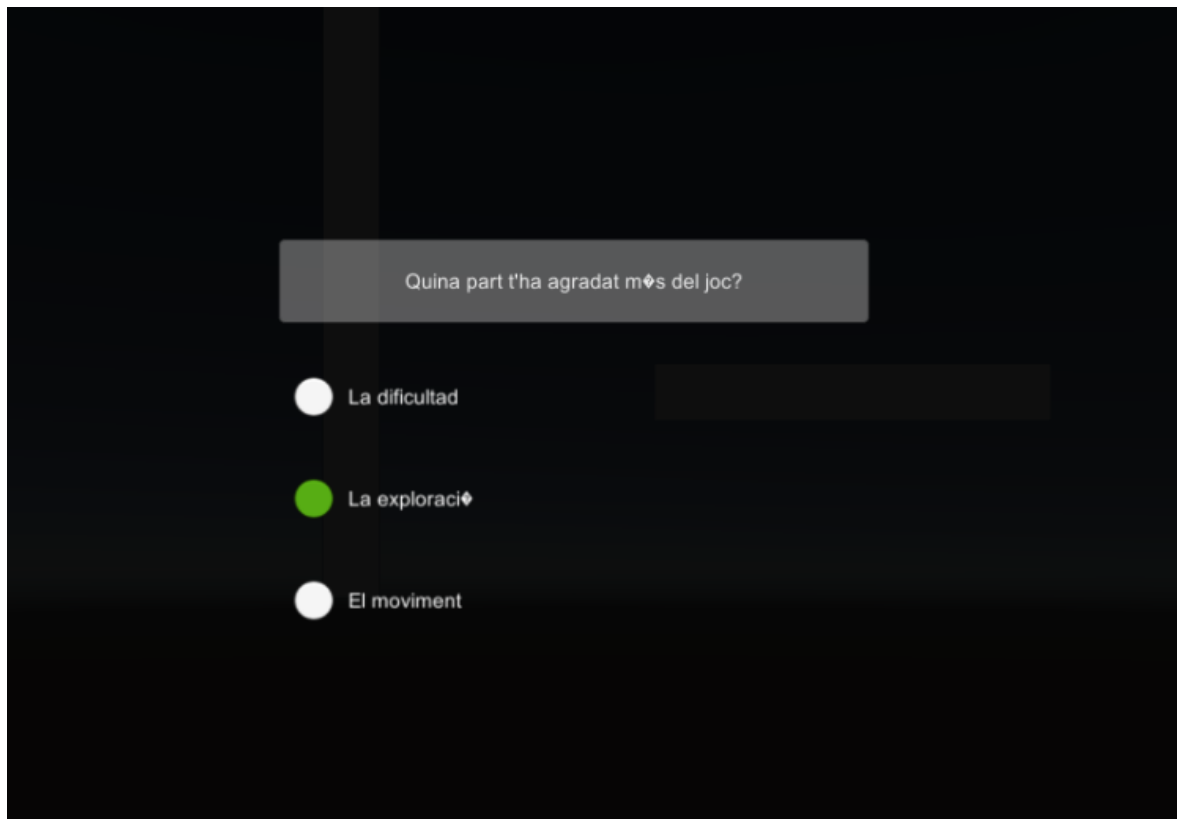
L'aplicació demana el nom a l'usuari, important per poder classificar el jugador a la base de dades.

Es deserialitza el fitxer JSON (5.3.2) i cada apartat es converteix a una classe on hi ha una pregunta i varies respostes segons el fitxer JSON que s'ha deserialitzat.

```
"questions": [  
  {  
    "question": "Quina part t'ha agradat més del joc?",  
    "answers": [  
      "La dificultat",  
      "La exploració",  
      "El moviment"  
    ]  
  },  
  {  
    "question": "Que t'ha frustrat més?",  
    "answers": [  
      "La duració del joc",  
      "Les vegades que he mort",  
      "La música"  
    ]  
  },  
  {  
    "question": "El joc anava massa ràpid?",  
    "answers": [  
      "Si",  
      "No"  
    ]  
  }  
]
```

Figura 5.3.2: format JSON del qüestionari

De manera dinàmica es creen les preguntes i respostes segons el fitxer JSON deserialitzat. En forma d'interfície d'usuari on el jugador selecciona la resposta més convenient 5.3.3. Aquesta resposta es guarda al script ReportGatherer per ser enviada al final del test a la base de dades.

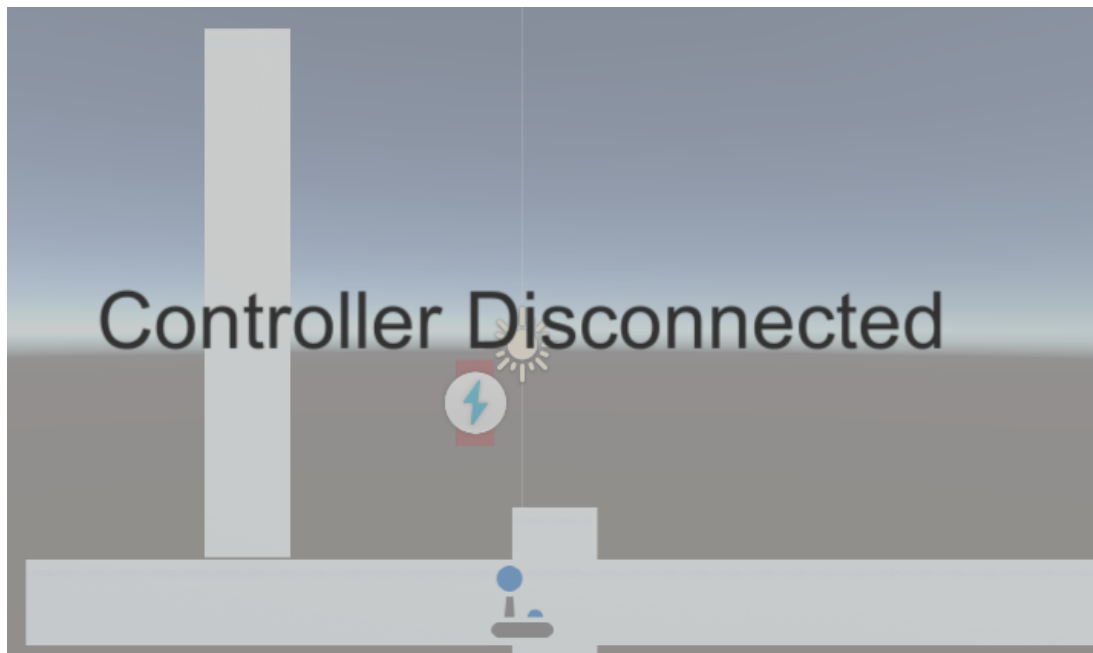


**Figura 5.3.3:** Preguntes i respostes generades de forma dinàmica

Un cop l'usuari finalitza el test tota la informació que s'ha monitorat es passa a format CSV, és a dir una línia de text on se separen els diferents camps per comes.

Per una altra banda s'ha creat un sistema de desconnexió del controlador per evitar que si el controlador es desconnecta de l'aplicació les mètriques que s'estan extraient quedin distorsionades.

El funcionament és simple, quan l'aplicació detecta que el controlador s'ha desconnectat el joc queda pausat fins que es torni a connectar.



**Figura 5.3.4:** Pantalla de controlador desconnectat.

### 5.3.1 Canvi de plantejament

A causa de la quantitat de dades que s'han d'extreure per cada jugador és molt més fàcil de llegir les dades per una persona en format JSON en comptes de CSV.

A la figura X es poden veure les classes que s'utilitzen per extreure les dades del jugador i enviar-les a PlayFab.

Les dades que s'extreuen estan compartimentades en rooms, i agrupades per nivells. S'extreu quantes accions de cada tipus has fet per habitació i quantes n'ha necessitat la vegada que s'ha passat el nivell. Així es pot començar a intuir el patró de comportament del jugador. També s'extreu el número de vegades que ha mort i el temps que ha necessitat per passar-se el nivell.

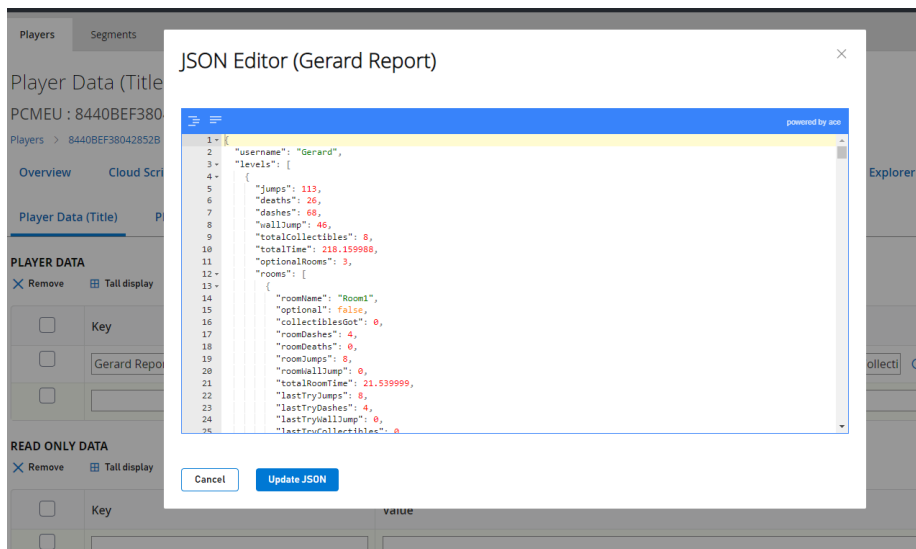
Per una altra banda, a través d'uns certs colliders s'extreu quin camí ha recorregut per passar-se el nivell, si el fàcil o el difícil. I el número de vegades que ha recorregut el tipus de camí fins que s'ha passat el nivell.

Pel que fa al rol secundari dels jugadors s'extreuen diferents dades.

Per a l'Achiever es mira quants col·leccionables hi ha pel nivell i quants n'ha agafat o quantes vegades els ha intentat agafar i després del còmput que s'explica al següent apartat es treu quin percentatge d'aquest perfil és.

Per al perfil de l'Explorer és molt similar, es calculen quantes rooms opcionals hi ha i si el jugador ha entrat i d'això se'n treu un percentatge.

I finalment també es recullen les preguntes i les respostes que ha escollit l'usuari.



**Figura 5.3.1.1:** Dades extreptes del jugador a playfab

### 5.3.2 Càlcul de dades

Un cop el nivell s'ha finalitzat, s'han extret totes les dades i el jugador ha fet el qüestionari, és el programa internament analitza les dades que ha extret per determinar quin perfil de jugador té l'usuari. Segons el rol principal es carrega una escena o una altra, és a dir un nivell dissenyat per a un jugador de tipus Agressiu o un nivell per un jugador de tipus Assegurador. I per als rols secundaris s'afegeixen o se suprimeixen elements a l'escena. Com que els dos rols secundaris poden coexistir en el mateix perfil de jugador es pot donar el cas que un jugador sigui Achiever i Explorer alhora.

Per definir si un jugador és Achiever en aquell nivell que ha jugat es calcula quin percentatge de col·leccionables respecte del total ha agafat l'usuari i si supera el 50%, se li adjudica el perfil Achiever.

Pel cas del perfil Explorer és el mateix, es calcula el percentatge d'habitacions opcionals visitades pel jugador respecte el total i si supera o iguala el 50% se li adjudica el perfil Explorer.

Per decidir si el jugador és Agressiu o Assegurador es calcula de manera diferent. Es llegeixen unes dades que determinen a partir de quantes accions un jugador es pot passar el nivell de manera agressiva, si l'usuari supera aquest nombre d'accions, se'l considera Assegurador. Això es fa per a cada tipus d'acció per a cada habitació i també es treu una mitjana dividint les accions totals per habitació entre el nombre de morts. Si es considera que es Assegurador a la variable se li suma un punt, si es considera que és Agressiu se li resta. Al

final s'acaba amb un valor que representa l'eix vertical a la Figura 5.1.1.1. Com més negatiu sigui el valor més Agressiu és el jugador i com més positiu més Assegurador.

## 5.4 Connexió amb base de dades

En aquesta secció s'explica els problemes inicials per connectar amb la base de dades i la posterior resolució.

### 5.4.1 Problemes Inicials

Al principi es va fer servir Firebase com a base de dades. Però es va veure que està orientat a aplicacions mòbils, és a dir d'Android i IOS, no a aplicacions d'ordinador. Tot i que sigui el servei més potent, en el cas d'aquest projecte no és útil, ja que la seva integració és molt complicada per a ordinador.

Després es va provar amb Unity Cloud Services plataforma d'anàlisi i emmagatzematge de dades desenvolupada per Unity, similar a Firebase. El problema d'aquest servei és que nou és prou madur, falten funcionalitats. L'administrador no pot veure les dades dels usuaris si no coneix l'ID de l'usuari la qual és anònima<sup>[29]</sup>, fet que impossibilita l'anàlisi de dades individuals, element troncal per al desenvolupament del projecte.

Finalment es va trobar Playfab, plataforma semblant a Firebase però desenvolupada per Microsoft, compta amb una integració dedicada amb Unity. L'avantatge de Playfab sobre Firebase en aquest cas és que també està creada per aplicacions d'ordinador.

### 5.4.2 Funcionament

L'aplicació registra automàticament l'usuari amb un ID únic, el qual està lligat a cada ordinador quan fa l'inici de sessió.

```
void Login()
{
    LoginWithCustomIDRequest request = new LoginWithCustomIDRequest
    {
        CustomId = SystemInfo.deviceUniqueIdentifier,
        CreateAccount = true
    };
    PlayFabClientAPI.LoginWithCustomID(request, OnSuccess, OnError);
}
```




Figura 5.4.2.1: Login de Playfab

En finalitzar la sessió de joc l'aplicació envia una línia de text separada per comes com a informació d'aquell jugador.

```
public void UploadDataCSV(Dictionary<string,string> dict)
{
    // UpdateUserDataRequest
    var request = new UpdateUserDataRequest
    {
        Data = dict
    };
    PlayFabClientAPI.UpdateUserData(request, OnDataSend,OnError);
}
```

**Figura 5.4.2.2:** Pujar les dades a Playfab

Des de l'aplicació web de Playfab es poden visualitzar tots els usuaris registrats, encara que sigui de forma anònima.

ID	Last login	Created
 8440BEF38042852B	9:18 AM	29 days ago
 84C926020D08E2FE PC Feina (Global)	May 5, 2022 3:18 PM	6 days ago
 7ABB2043AD05C042	Apr 9, 2022 2:34 PM	29 days ago

**Figura 5.4.2.3:** Usuaris registrats de forma anònima a Playfab

L'administrador de la base de dades pot veure les dades que els usuaris han pujat i recollir la informació per a analitzar-la.



## PLAYER DATA

 Remove  Full display

<input type="checkbox"/>	Key	Value	Permissions
<input type="checkbox"/>	Gerard Report	2,0,3,0,0,0,07/05/2022 19:06:16	Private ▾
<input type="checkbox"/>	Report	juan,4,1,1,6,09/04/2022 15:25:46	Private ▾

Figura 5.4.2.4: Informació dels usuaris

## 5.5 Tecnologies Externes a Unity

Recull de tecnologies utilitzades que no venen per defecte a Unity tot i que algunes han estat comprades per Unity com l'input system.

**LeanTween:** és un motor d'animacions molt eficient i ràpid d'utilitzar, compte amb moltes customitzacions i un sistema de callbacks que fa molt ràpid de fer la concatenació d'animacions. En aquest projecte s'usa per animar la interfície d'Usuari.

**InputSystem:** sistema de captació de les senyals d'entrada a l'aplicació per part de l'usuari "inputs". Explicada en detall a l'apartat [\[2.2.2\]](#).

**NewtonSoft:** serialitzador de fitxers JSON molt eficient i potent, fàcil d'utilitzar i el més popular.

**Cinemachine:** editor de càmeres per a Unity, fàcil d'utilitzar.

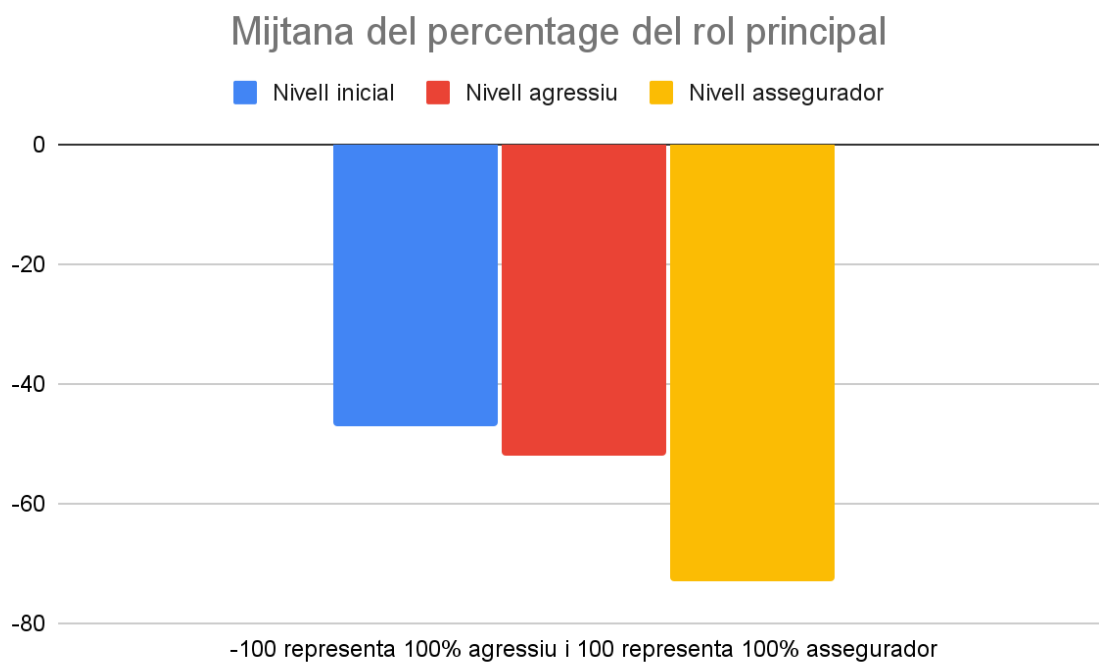
**Playfab:** base de dades de Microsoft, fàcil d'utilitzar.

## 6. Conclusions i treballs futurs

### 6.1 Tests

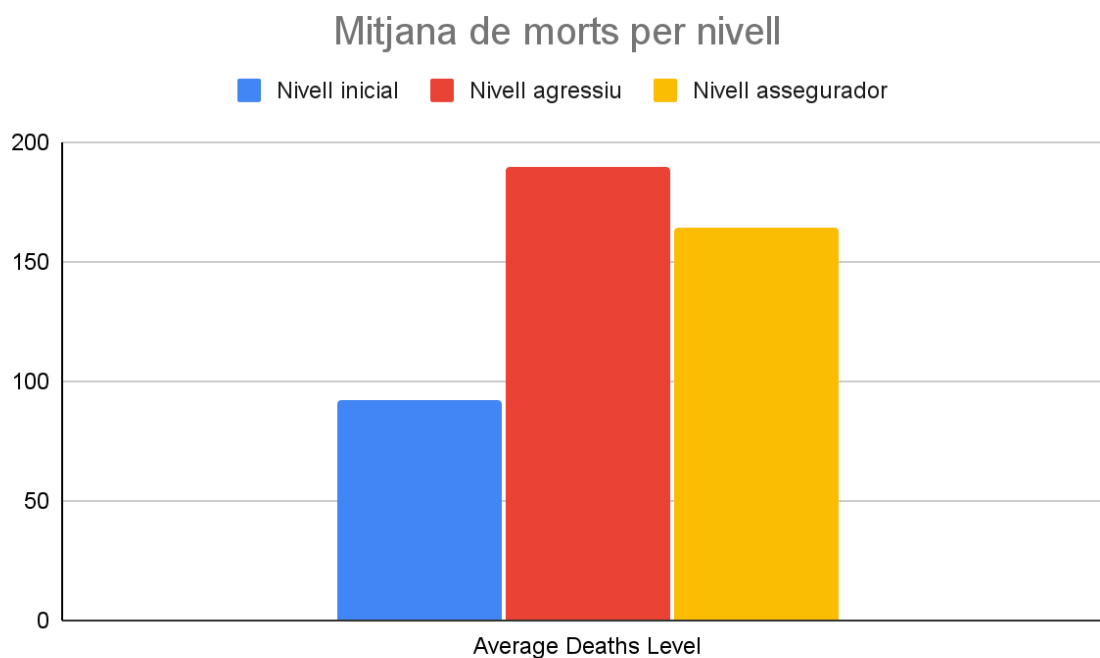
En aquest apartat s'explica el resultat dels tests realitzats i la concordança amb la taxonomia descrita a l'apartat [5.1.1].

La majoria de jugadors ha optat per un tipus de manera de jugar agressiva figura 6.1.



**Figura 6.1:** Mitjana de percentatges dels rols principals

Pot ser degut a diferents factors. El primer és que la forma de comptabilitzar la tendència de l'usuari. Si el nombre de morts és molt elevat, repercuteix molt significativament en tindre en compte a aquell usuari com a agressiu en comptes d'assegurador, ja que es fa una mitjana entre el nombre d'accions fetes i entre el nombre de morts per nivell, com s'explica a l'apartat [5.3.2]. Un altre element que pot maquillar el tipus de rol principal de l'usuari és l'elevada dificultat de l'aplicació, als qüestionaris tots els usuaris concorden en què el joc és molt difícil de passar, es pot veure a la figura 6.2.



**Figura 6.2:** Mitjana de morts per nivell

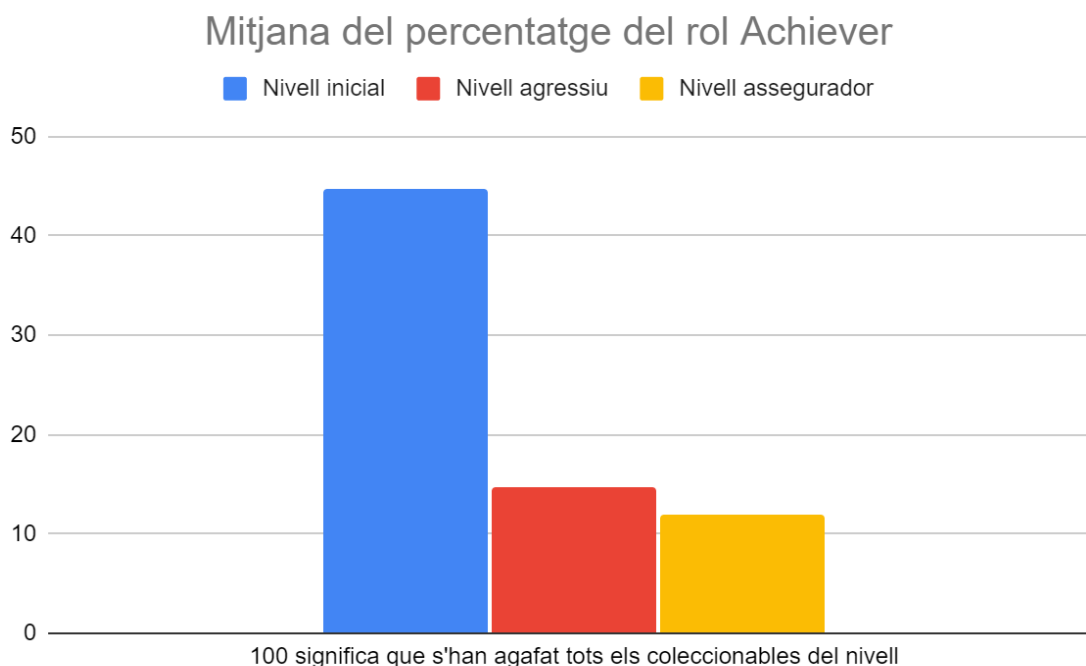
Per una altra banda, la majoria de jugadors independentment de si el seu perfil ha resultat com a Agressiu o Assegurador escullen el camí més difícil dels diferents camins possibles, segons la taxonomia un element típic dels jugadors agressius, pot ser degut al fet que en la majoria de casos el camí difícil és el que condueix cap al col·leccionable o el camí que està més a la vista. Al contrari que el fàcil que s'ha de rodejar més.

Al nivell agressiu tots els usuaris han respost que és el nivell amb més dificultat i el més estressant i ràpid, a causa de la mecànica única del nivell [[LevelEater](#)].

Al nivell Assegurador, on hi ha més possibles camins per solucionar cada habitació o room els jugadors han notat menys dificultat, així i tot, no s'ha aconseguit establir una relació directa entre la taxonomia [[5.1.1](#)], les modificacions del joc i l'experiència de l'usuari.

Quant als rols secundaris (Achiever i Explorer) sí que s'ha encertat més. A la gràfica 6.3 es pot veure que al primer nivell la mitjana ronda el 50% de perfil Achiever, en ser el nivell més senzill els jugadors agafen col·leccionables més fàcilment. A mesura que augmenta la dificultat com es pot veure el percentatge baixa dràsticament, o bé perquè des d'un inici els jugadors que no tenen interès a agafar col·leccionables quan arriben als següents nivells ja no en tenen, llavors fa baixar la mitjana o perquè l'augment de la dificultat dissuadeix als usuaris d'intentar agafar els col·leccionables i prefereixen passar-se el nivell només.

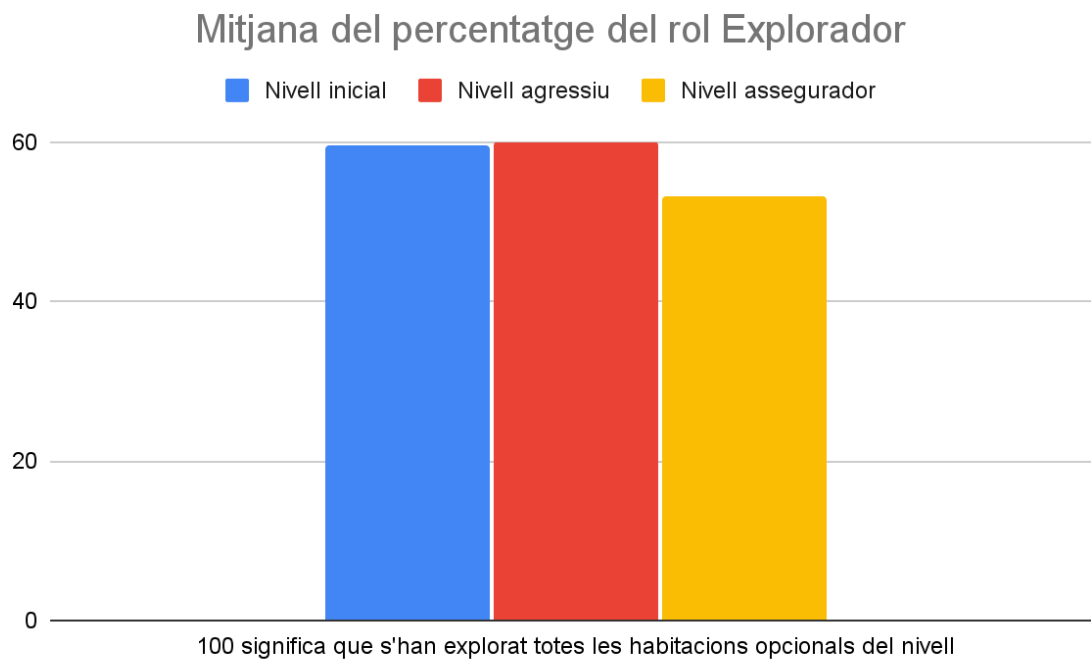
En canvi, els jugadors que superen el 50% de col·leccionables obtinguts per nivell sí que mostren a les respostes dels tests que és una mecànica que els hi agrada.



**Figura 6.3:** Percentatge de rol Achiever

Amb el rol d'Explorador els percentatges augmenten respecte a l'Achiever, un dels principals motius és degut al fet que la dificultat per entrar en una zona opcional és molt més reduïda que per aconseguir un col·leccionable.

A les respostes s'observa que els jugadors que més exploren troben el nivell lineal, deixa entreveure una necessitat de crear més zones opcionals i augmentar la mida del mapa per arribar complir amb les seves expectatives.



**Figura 6.4:** Percentage de rol Explorer

## 6.2 Objectius generals

En termes generals es pot considerar que s'han complert amb èxit els tres punts generals [\[1.3\]](#).

**Definir els paràmetres per catalogar els diferents tipus de jugadors:** s'ha aconseguit crear una taxonomia a partir de la proposta de Bartle però enfocada als videojocs de plataformes.

**Crear videojoc on fer l'exploració:** S'ha desenvolupat un prototip amb Unity, d'un videojoc de plataformes on poder extreure les dades dels jugadors i modificar el joc segons els criteris del primer punt.

**Contrastar els resultats obtinguts amb els esperats:** el temps ha sigut molt limitat, però s'han pogut fer playtesting a diferents usuaris, extreure dades i comparar-les entre elles i veure si guarden relació amb la taxonomia.

### 6.3 Objectius específics

En relació amb els punts específics no tots s'han complert totalment degut a la desviació de la planificació original [3.1.5].

**Definir els 4 tipus de jugadors partint com a base de la taxonomia de Bartle però enfocats en un joc platformer:** al punt [5.1.1] s'explica la taxonomia creada per a un platformer tot i que com s'ha explicat a l'apartat anterior de test no es pot donar per vàlida al complet, ja que el rol d'Assegurador no ha pogut ser validat perquè quasi tota la mostra d'usuaris té o dona el perfil Agressiu.

**Obtenir les dades dels usuaris a través de diferents elements de monitoratge dins de Unity i exportar-los en diferents formats (CSV i JSON):** S'ha complert amb èxit, es monitoritzen totes les accions del jugador a cada secció del joc i s'envien a la base de dades [5.3].

**Desenvolupament de un videojoc amb unity:** al llarg del punt 5 s'ha explicat el desenvolupament del videojoc i les dificultats que han sorgit. S'ha complert amb èxit, igual que la taxonomia és un element fonamental per seguir amb els següents punts.

**Integrar l'extracció de dades a Unity:** s'ha complert amb èxit després d'iterar amb diferents serveis i provocar una desviació de la planificació [5.4].

**Fer tests a usuaris amb qüestionaris:** aquest ha sigut el punt més complicat, ja que per falta de temps no s'han pogut fer tants qüestionaris com estava planificat, tot i això han sigut suficients per extreure conclusions.

**Comprovar la precisió de les prediccions:** no s'ha complert amb èxit, ja que el temps ha sigut mínim, i no s'ha pogut comprovar de manera fiable si la taxonomia realment és efectiva amb tots els perfils de jugador [6.1].

### 6.4 Treballs futurs

El problema proposat en aquest treball encara és vigent, tot i això, s'ha pogut demostrar que sí que els jugadors tenen certes tendències que poden ser explotades per part dels desenvolupadors de videojocs i millorar l'experiència de joc.

La idea és molt transversal i pot ser aplicada en qualsevol gènere i àmbit dels videojocs. El que ha faltat en aquest treball ha sigut trobar una correlació prou forta per determinar que modificant certs aspectes del joc l'experiència d'un cert tipus de jugador millora.

## 7. Bibliografia

1. Richard Bartle. Abril 1996.  
**HEARTS, CLUBS, DIAMONDS, SPADES: PLAYERS WHO SUIT MUDDS**  
<https://mud.co.uk/richard/hcnds.htm>
2. MattherBarr. 19 Gener 2013.  
**The Bartle Test of Gamer Psychology**  
<https://matthewbarr.co.uk/bartle>
3. Hunicke, R., LeBlanc, M., & Zubek, R. 2004.  
**MDA: A formal approach to game design and game research.**
4. John M. Quick, Robert K. Atkinson.  
**A Data-Driven Taxonomy of Undergraduate Student Videogame Enjoyment.**  
<https://dl.acm.org/doi/pdf/10.5555/2206376.2206399>
5. Heeter, C. 2008.  
**Play styles and learning.** In R. Ferdig (Ed.), *Handbook of Research on Effective Electronic Gaming in Education* (pp. 826-846).
6. Yee,N. 2007.  
**Motivations for Play in online games.** *CyberPsychology & Behavior.*  
[http://www.nickyee.com/pubs/Yee%20-%20Motivations%20\(2007\).pdf](http://www.nickyee.com/pubs/Yee%20-%20Motivations%20(2007).pdf)
7. Frederiek de Vette.  
**Engaging Elderly People in Telemedicine Through Gamification.**  
[https://www.researchgate.net/publication/287217318\\_Engaging\\_Elderly\\_People\\_in\\_Telemedicine\\_Through\\_Gamification#pf7](https://www.researchgate.net/publication/287217318_Engaging_Elderly_People_in_Telemedicine_Through_Gamification#pf7)
8. Richard M. Ryan · C. Scott Rigby · Andrew Przybylski. 2006.  
**The Motivational Pull of Video Games: A Self-Determination Theory Approach**  
<https://link.springer.com/article/10.1007/s11031-006-9051-8>
9. Richard M. Ryan i Edward L. Deci.  
**Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being.**  
[https://selfdeterminationtheory.org/SDT/documents/2000\\_RyanDeci\\_SDT.pdf](https://selfdeterminationtheory.org/SDT/documents/2000_RyanDeci_SDT.pdf)
10. 1 Març 2018.  
**Your Brain on Video Games.**

<https://sciencemadefun.net/blog/your-brain-on-video-games/>

11. Dave Rickey. 2003.  
**Engines of Creation #1**  
<http://www.skotos.net/articles/engines01.shtml>
12. Atharva Kulkarni (9 Setembre 2021)  
**Mobile Games and the Bartle Taxonomy of Player Types.**  
<https://medium.com/@atharvakulkarniamk/mobile-games-and-the-bartle-taxonomy-of-player-types-9af7a6579814>
13. Will Luton. 12 Maig 2013.  
**Free-to-Play: Making Money From Games You Give Away.**  
[https://books.google.es/books?id=QIXNquWvB2oC&pg=PA76&lpg=PA76&dq=Will+Luton+the+four+c%27s&source=bl&ots=Wm1lF2jVKt&sig=ACfU3U0\\_P0LQkO2gm1BKnQ6qw89rD9Wr3Q&hl=es&sa=X&ved=2ahUKewjVxY\\_F1dL2AhXGt6QKHZ15DwUQ6AF6BAgCEAM#v=onepage&q=Will%20Luton%20the%20four%20c's&f=false](https://books.google.es/books?id=QIXNquWvB2oC&pg=PA76&lpg=PA76&dq=Will+Luton+the+four+c%27s&source=bl&ots=Wm1lF2jVKt&sig=ACfU3U0_P0LQkO2gm1BKnQ6qw89rD9Wr3Q&hl=es&sa=X&ved=2ahUKewjVxY_F1dL2AhXGt6QKHZ15DwUQ6AF6BAgCEAM#v=onepage&q=Will%20Luton%20the%20four%20c's&f=false)
14. Susanne Morris. 8 Abril 2021.  
**Data Storage: JSON vs. CSV**  
<https://coresignal.com/blog/json-vs-csv/>
15. *Alexander Shvets.*  
**Patrón Observer.**  
<https://refactoring.guru/es/design-patterns/observer>
16. JSON Utility Unity.  
<https://docs.unity3d.com/es/530/Manual/JSONSerialization.html>
17. CSV Package  
<https://docs.unity3d.com/Packages/com.unity.localization@0.9/manual/CSV.html>
18. Soda Rocket Studios. 6 Setembre 2020.  
**Unity Input System Deep Dive**  
<https://www.youtube.com/watch?v=o9wauUU7nmg>
19. **The Official Guide to The Kanban Method.** 2022, March 14.  
<https://resources.kanban.university/kanban-guide/#1614205710105-243aa9a9-c795>
20. **Estimació del salari brut** realitzat a partir de les dades extretes d'indeed.es  
(Consultat el 13/03/2022)



<https://es.indeed.com/career/programador-junior/salaries>

21. Ortiz, Juan. 2 de desembre de 2019.  
**Investigación exploratoria: tipos, metodología y ejemplos.** Lifeder.  
<https://www.lifeder.com/investigacion-exploratoria/>
22. Unity Manual 2019.  
**Actions.** Input System ver 1.3.0. 2019  
<https://docs.unity3d.com/Packages/com.unity.inputsystem@1.3/manual/Actions.html>
23. **C# - Events.** Tutorials teacher  
<https://www.tutorialsteacher.com/csharp/csharp-event>
24. Strachan, David.  
**Gamedevplatformer.**  
<http://www.davetech.co.uk/gamedevplatformer>
25. Unity.  
**Cinemachine**  
<https://unity.com/es/unity/features/editor/art-and-design/cinemachine>
26. Unity Manual  
**Class CinemachineVirtualCamera.**  
<https://docs.unity3d.com/Packages/com.unity.cinemachine@2.1/api/Cinemachine.VirtualCamera.html>
27. Swink, Steve. 19 novembre de 2008.  
**Game Feel: A Game Designer's Guide to Virtual Sensation.**  
<https://gamifique.files.wordpress.com/2011/11/2-game-feel.pdf>
28. Dented Pixel  
**LeanTween Class**  
<http://dentedpixel.com/LeanTweenDocumentation/classes/LeanTween.html>
29. Philip Herlitz. 24 octubre 2021.  
**Will Unity's New Cloud Save Work for Me?**  
<https://www.youtube.com/watch?v=DSXpl-yi6vg>

