



Universitat de les  
Illes Balears



Treball Final de Grau

GRAU D'ENGINYERIA INFORMÀTICA

Aplicació de la metodologia dels videojocs  
seriosos a entorns educatius i  
rehabilitadors

MARC CAMPINS

**Tutor**

Paco Perales

Escola Politècnica Superior  
Universitat de les Illes Balears  
Palma, juny de 2017



Aquest projecte de fi de grau ha estat parcialment finançat per el  
TIN2015-67149-C3-2-R.

Agraïments especials a totes les persones que m'han ajudat en el desenvolupament del projecte i amb els que he tingut el plaer de treballar. Especialment a en Miquel Mascaró Oliver per la base que em va donar als laberints, a en Joan Florit per la seva feina prèvia en el GeniusUp, a na Silvia Ramis per la seva ajuda i col·laboració tant en el GeniusUp com en Laberint, a na Milica Vujovic per tota la seva feina, ajuda i col·laboració durant el projecte del laberint, i a en Paco Perales per els continus ànims, suport i motivació per a fer el millor treball de fi de grau possible. Finalment agrair a la meva família per el suport que em donen cada dia i als amics i companys de l'universitat per fer d'aquests quatre anys una molt bona experiència.



# SUMARI

<b>Sumari</b>	<b>iii</b>
<b>Índex de figures</b>	<b>vii</b>
<b>Acrònims</b>	<b>ix</b>
<b>Resum</b>	<b>xi</b>
<b>1 Introducció</b>	<b>1</b>
1.1 Context . . . . .	1
1.2 Projecte GeniusUp . . . . .	2
1.3 Projecte Juguemos . . . . .	2
1.4 Objectius . . . . .	3
1.5 Estructura del document . . . . .	4
<b>2 Gamificació i Serious Games</b>	<b>5</b>
2.1 Estat actual . . . . .	5
2.1.1 Origen . . . . .	5
2.1.2 Definició . . . . .	5
2.1.3 Nivells d'abstracció . . . . .	6
2.1.4 Serious Games . . . . .	6
2.2 Categories . . . . .	7
2.2.1 Jocs corporatius . . . . .	7
2.2.2 Jocs educatius . . . . .	8
2.2.3 Jocs per a la salut . . . . .	8
2.2.4 Jocs socials . . . . .	9
2.2.5 Jocs persuasius . . . . .	9
2.3 Aplicació al projecte GeniusUp . . . . .	10
2.4 Aplicació al projecte Juguemos . . . . .	10
2.5 Futur pròxim . . . . .	11
<b>3 Unity</b>	<b>13</b>
3.1 Motors de videojocs . . . . .	13
3.2 Arquitectura . . . . .	14
3.2.1 Hardware . . . . .	14
3.2.2 Sistema operatiu . . . . .	15
3.2.3 Capa d'independència de plataforma . . . . .	16

3.2.4	Llibreries externes . . . . .	16
3.2.5	Nucli del motor . . . . .	16
3.2.6	Xarxa . . . . .	17
3.2.7	Gestor de recursos . . . . .	18
3.2.8	Dispositius d'entrada . . . . .	18
3.2.9	Àudio . . . . .	18
3.2.10	Gràfics i rendering . . . . .	18
3.2.11	Interfície d'usuari . . . . .	19
3.2.12	Gameplay i scripting . . . . .	19
3.3	Característiques . . . . .	19
<b>4</b>	<b>GeniusUp</b>	<b>21</b>
4.1	Estat del projecte al moment de la meva incorporació . . . . .	21
4.2	Objectius . . . . .	22
4.3	Motiu per el que és un serious game . . . . .	24
4.4	Implementació amb Unity . . . . .	26
4.4.1	Estructura general . . . . .	26
4.4.2	Splashscreen . . . . .	27
4.4.3	Menú principal . . . . .	27
4.4.4	Opcions . . . . .	29
4.4.5	Crèdits . . . . .	30
4.4.6	Menú de nivells . . . . .	30
4.4.7	Registre . . . . .	34
4.4.8	Joc . . . . .	34
4.5	Accessibilitat . . . . .	41
4.5.1	Definició . . . . .	41
4.5.2	Barreres . . . . .	41
4.5.3	Idioma . . . . .	41
4.5.4	Color . . . . .	42
4.6	Verificació . . . . .	43
4.7	Estat final i futur del projecte . . . . .	43
<b>5</b>	<b>Laberint</b>	<b>47</b>
5.1	Estat del projecte al moment de la meva incorporació . . . . .	47
5.2	Objectius . . . . .	48
5.3	Motiu per el que és un serious game . . . . .	50
5.4	Implementació amb Unity . . . . .	51
5.4.1	Model comú del laberint . . . . .	51
5.4.2	Lògica de l'aplicació de recollida de dades . . . . .	55
5.4.3	Lògica de l'aplicació de presa de fotos . . . . .	60
5.5	Verificació amb grups de control i anàlisi de dades . . . . .	60
5.6	Estat final i futur del projecte . . . . .	61
<b>6</b>	<b>Conclusions</b>	<b>67</b>
6.1	Assoliment d'objectius . . . . .	67
6.2	Experiència personal . . . . .	68

*SUMARI*

v

<b>7 Annexos</b>	<b>69</b>
7.1 A.1 Repercussió mediàtica . . . . .	69
7.2 A.2 Google Play Store i App Store . . . . .	69
7.3 A.3 Papers projecte Jugamos . . . . .	69
<b>Bibliografia</b>	<b>71</b>





## ÍNDIX DE FIGURES

2.1	Competències dels serious games [1]	8
3.1	Arquitectura del motor de videojocs Unity [1]	15
3.2	Comparació de Unity amb altres motors de videojocs [1]	20
4.1	Resum d'iniciatives del projecte POTU	22
4.2	Marc de desenvolupament de Serious Educational Game (SEG) (figura original de [2])	25
4.3	Relació entre escenes de l'aplicació GeniusUp	27
4.4	Escena Splashscreen	28
4.5	Escena Menú principal	28
4.6	Escena Opcions	29
4.7	Escena Crèdits	30
4.8	Jerarquia dins Unity dels nivells	31
4.9	Nivells de la categoria de biologia	32
4.10	Nivells de la categoria de química	32
4.11	Estructura de dades d'una combinació de química	33
4.12	Estructura de dades d'una combinació de biologia	33
4.13	Estructura de dades de la part gràfica d'un nivell	34
4.14	Estructura de dades de la part lògica d'un nivell	35
4.15	Escena Registre	36
4.16	Escena Joc amb elements químics	36
4.17	Escena Joc amb elements biològics	37
4.18	Missatge d'ajuda per a l'usuari	37
4.19	Missatge de joc finalitzat	39
4.20	Missatge de nivell completat amb èxit	39
4.21	Missatge de nivell fallit	40
4.22	Comparació del mode protanopia	42
4.23	Comparació del mode deuteranopia	43
4.24	Comparació del mode tritanopia	43
4.25	Valoracions Play Store	44
5.1	Taula amb l'estructura dels laberints segons nivell	49
5.2	Dispositius de interacció	51
5.3	Estructura d'escenes del laberintl	51
5.4	Escena Menú principal	52
5.5	Escena Introducció de dades	53

5.6	Escena Laberint, primera categoria de dificultat . . . . .	53
5.7	Escena Laberint, segona categoria de dificultat . . . . .	54
5.8	Escena Laberint, tercera categoria de dificultat . . . . .	54
5.9	Escena Laberint, aplicació de recollida de fotos . . . . .	55
5.10	Missatge de finalització de nivell . . . . .	55
5.11	Missatge de finalització de ronda de nivells . . . . .	56
5.12	Missatge d'ajuda inicial . . . . .	56
5.13	Missatge d'inici de fase de calibratge . . . . .	57
5.14	Missatge de fi de fase de calibratge . . . . .	57
5.15	Missatge de victòria . . . . .	58
5.16	Taula amb els punts crítics i les emocions a capturar . . . . .	63
5.17	Diagrama de la metodologia de l'estudi . . . . .	64
5.18	Resultats de cinc participants de nou anys . . . . .	65
5.19	Comparació del temps entre la mitjana i una nena amb paràlisi cerebral . . . . .	65

## ACRÒNIMS

**TFG** Treball Final de Grau

**UIB** Universitat de les Illes Balears

**POTU** Programa d'Orientació i Transició a la Universitat

**UGIVIA** Unitat de Gràfics i Visió per Computador i Intel·ligència Artificial de la Universitat de les Illes Balears

**GPU** Graphics Processing Unit

**QR** Quick Response

**ASPACE** Asociación de Parálisis Cerebral de España

**GASIG** Game Accessibility Special Interest Group

**IGDA** International Game Developers Association

**SEG** Serious Educational Game



## RESUM

El meu treball de fi de grau es basa en l'aplicació de les metodologies de la gamificació i els serious games per tal de desenvolupar videojocs que tinguin un pròposit més enllà del pur entreteniment. S'han desenvolupat tres videojocs distints. El primer és un joc educatiu per a mòbils, en el qual el jugador ha de realitzar combinacions d'elements per tal de sumar punts. Aquestes combinacions representen conceptes d'una de les quatre branques científiques que compren el joc—química, biologia, física i matemàtiques—, amb el que el jugador va aprenent nous conceptes de la ciència a mesura que juga.

El segon videojoc es tractava d'un laberint, el qual llavors es fa servir per a varis propòsits. Els propòsits realitzats fins a aquesta data són dos. El primer es basa en enregistrar una sèrie de variables durant les partides del jugador, per a què llavors nens i nenes amb autisme i dèficit d'atenció juguin al laberint i, mitjançant l'estudi d'aquestes variables obtingudes, es pugui determinar la seva atenció i quins estímuls la afecten més, tan positiva com negativament. El segon es basa en el mateix laberint que l'anterior, però enlloc d'enregistrar les variables es fan una sèrie de fotos a certs punts crítics per a així llavors poder estudiar les expressions dels jugadors a distints moments de la partida. Altres variants i estudis s'han proposat per al mateix entorn del laberint, el que denota la seva versatilitat, però s'han deixat com a feina per al futur.

Finalment, s'ha desenvolupat una tercera aplicació que seveix com a eina d'entrenament en l'expressió d'emocions per a persones amb autisme, mitjançant l'ús d'emoticons i una webcam. Els jugadors s'han de fer fotos intentant imitar l'expressió que veuen a la icona. Les primeres fotos es fan amb els jugadors veient-se a ells mateixos a través de la webcam, les fotos finals es fan sense referència visual.

Totes les aplicacions s'han desenvolupat amb el motor de jocs Unity, i per tant s'han aplicat les mateixes metodologies a l'hora de crear les aplicacions de cada respectiu projecte.

Aquest document seguirà una estructura habitual, primer s'especificaran breument el context i objectius del projecte; després s'exposaran les característiques d'aquests tipus de projectes i la seva raó de ser; a continuació es justificarà l'elecció del programa Unity, així com les particularitats i implementacions de cada joc dins el motor; llavors s'explicaran en detall els dos projectes i el seu desenvolupament; i finalment s'enunciaran les conclusions a les que s'ha arribat i les línies futures o possibles continuacions dels projectes.



## INTRODUCCIÓ

### 1.1 Context

Ens els darrers anys la indústria del videojoc ha crescut notablament, degut a l'augment de les capacitats de processament dels ordinadors, cada cop més potents, la possibilitat de creació d'imatges més realistes o amb un estil més original i artístic, i finalment l'estreta relació entre les pel·lícules i sèries—que també han crescut en popularitat últimament—i els videojocs, els quals cada cop volen crear una major immersió i contar històries més personals i emotives. A més, els videojocs han expandit els seus dominis fora de les consoles tradicionals, amb una explosió similar en els videojocs dirigits cap a les plataformes mòbils. [3]

Aquest ascenens tan meteòric durant aquesta darrera dècada, ha fet créixer les pràctiques i metodologies de la indústria més enllà de la pura creació de productes d'entreteniment. L'èxit de dites tècniques i processos ha dut a terme una expansió cap a altres àmbits laborals, els quals les han volgut adoptar en els seus propis procediments. D'interès per aquest treball de fi de grau són dues d'aquestes expansions dels videojocs cap a territoris desconeguts, l'anomenada *ludificació* o *gamificació* i la branca de *videojocs seriosos* o *serious games*.

La ludificació es basa en intentar aplicar els elements de disseny i els principis dels videojocs a altres contextos no basats únicament en l'entreteniment. D'aquesta manera es pretén millorar la productivitat, la implicació i motivació, la facilitat d'interacció i ús, i incrementar altres factors com la fluïdesa, l'avaluació o l'aprenentatge d'una tasca en concret.

El serious games són videojocs que han estat dissenyats amb un propòsit distint del pur entreteniment. És a dir, videojocs que a la pràctica es fan servir en altres indústries, sigui l'educació, la recerca científica o la medicina. Pot partir de la idea d'oferir una simulació més o menys real per tal de transmetre, ensenyar o treballar una sèrie de conceptes o accions al jugador, o es pot basar en usar el valor pedagògic de la diversió i competició inherents als videojocs.

Tant el projecte *GeniusUp* com el projecte *Juguemos* parteixen d'aquesta nova

manera de pensar, i pretenen usar l'atractiva interfície d'un videojoc per tal d'educar a joves en la ciència—en el cas de *GeniusUp*—, o avaluar, monitoritzar i millorar l'atenció i l'expressió d'emocions de nens i nenes amb autisme o dèficit d'atenció—en el cas de *Juguemos*.

### 1.2 Projecte GeniusUp

El projecte *GeniusUp* era una iteració més del que inicialment havia nascut com a projecte *PotuGame*. I el projecte *PotuGame* era un avanç més del programa Programa d'Orientació i Transició a la Universitat (POTU), així que quan em vaig incorporar el projecte estava més que avançat. El problema és que per falta de finançament, entre altres dificultats, el projecte de l'aplicació educativa encara no s'havia pogut dur a terme.

Dins del marc del POTU ja s'havien creat unes cartes similars al joc *Uno*, on cada un dels quatre colors representava una branca de la ciència—en concret les de matemàtiques, física, biologia i química—. Les cartes eren una manera d'educar sobre la ciència de manera entretinguda, divertida, on els estudiants estassin motivats.

A part de les cartes però, el projecte de crear un joc educatiu dins del marc POTU no es va poder finalitzar, i quan jo em vaig incorporar s'havia transformat en el projecte *GeniusUp*, el qual volia assolir el mateix fi que el *PotuGame*, educar als joves i interessar-los en la ciència mitjançant un videojoc.

### 1.3 Projecte Juguemos

El projecte *Juguemos* era un altre dels projectes de la Unitat de Gràfics i Visió per Computador i Intel·ligència Artificial de la Universitat de les Illes Balears (UGIVIA) de la Universitat de les Illes Balears (UIB), i tenia un abast molt ampli. Així a grans trets el projecte pretenia definir un marc conceptual, una metodologia de disseny i un conjunt de eines software que permetessin el disseny i desenvolupament de jocs multimodals.

És evident que un projecte de tal magnitud tindrà distints subprojectes en marxa, i un d'ells era el desenvolupament d'un joc tipus laberint, el qual forma un entorn en el que llavors es realitzen dos estudis. El primer es basa en enregistrar el temps, les col·lisions i la longitud del camí amb els que el jugador ha arribat a la solució, per llavors tenir una base de dades amb la qual estudiar l'atenció dels jugadors, així com també comparar l'efecte en el seu rendiment de distints estímuls, siguin sons o visuals. El segon estudi es centrava en el reconeixement d'expressions en nens i nenes amb autisme, mitjançant unes fotografies en certs punts crítics del laberint, sobretot aquells on el jugador serà més expressiu—quan per exemple veu que ja arriba al final del laberint o que ha entrat a un carreró sense sortida.

Finalment un dels altres projectes que sorgiren dins el marc *Juguemos* és la realització d'una aplicació amb webcam que pretén millorar el reconeixement i expressió d'emocions. L'aplicació consisteix en una sèrie d'imatges d'emocions bàsiques que van sortint aleatòriament a l'esquerra de la pantalla. Els nens i nenes amb autisme haurien de reproduir les emocions i fer-se una foto via webcam. Primer tindrien la webcam de referència, però al cap d'una sèrie de fotos ja no, així llavors es podran comparar els resultats. Aquesta darrera a l'hora de realització del treball encara està en una fase



primerenca, i no es pot definir exactament com a un serious game, ja que l'aplicació de moment no aporta cap element característic dels jocs. Ara bé, el que s'ha desenvolupat fins ara és la base, la qual està pendent de verificar per experts en el tema. Un cop s'hagi verificat, al ser bàsicament un entrenador d'emocions, conté molts dels trets dels *serious games*—fomenta la millora o l'esperit de superació—, així que es podrien anar afegint nous elements que apropin l'aplicació cada cop més a l'àmbit dels *serious games*. Això s'ha deixat però com a treball per al futur.

### 1.4 Objectius

Ambdós projectes tenen una cosa en comú: l'ús dels videojocs en un àmbit fora de l'entreteniment. En el cas del *GeniusUp* ens acostem a un àmbit més educatiu, és a dir, pretén animar als estudiants a aprendre ciència, fent-ho d'una manera que els motivi, els enganxi, i sobretot que els hi sigui més propera que l'educació tradicional. Això es pot aconseguir gràcies a les virtuts dels videojocs, els quals inciten la competició i l'autosuperació. A més, en aquest cas en concret, es fa ús dels dispositius mòbils, els quals han esdevingut imprescindibles en la vida dels joves d'avui en dia.

Més concretament:

- Fomentar la ciència entra la gent jove.
- Aconseguir una educació de la ciència fluïda i natural, allunyar-la dels prejudicis habituals que tenen els estudiants.
- Obtenir una aplicació funcional, penjada a la *Google Play Store*, amb el major nombre de descàrregues i valoracions positives possibles.
- Crear una videojoc modular, amb un sol model poder educar en més d'una branca científica.
- Desenvolupar una aplicació accessible, que compleixi els estàndars d'accessibilitat de videojocs.

El projecte *Juguemos* s'acosta més a un àmbit de recerca i investigació, però de nou fa servir els videojocs com a mitjà i mètode d'obtenció de dades. En aquest cas es tracten temes de caire més psicològic, com l'atenció o l'expressió, però un altre cop es fan servir les virtuts dels videojocs per a aconseguir que l'experiència dels joves que realitzen l'estudi sigui més agradable, interactiva i divertida. Així es poden obtenir els mateixos resultats que obtindria una recerca tradicional però d'una manera que enganxa i anima molt més als participants.

Més concretament:

- Definir un marc conceptual, una metodologia de disseny i un conjunt d'eines de software que permetin el disseny de jocs multimodals.
- Crear un joc modular, on en un mateix entorn—el laberint—es puguin aconseguir objectius i propòsits distints.

- Establir una metodologia de creació de laberints, per tal d'aconseguir laberints consistents.
- Obtenir dades sobre el rendiment dels jugadors durant les partides.
- Obtenir fotos dels jugadors durant les partides, per tal d'estudiar les seves expressions.
- Entrenar l'expressió d'emocions de nens i nenes amb autisme.

### 1.5 Estructura del document

El document es divideix en els següents capítols:

- *Introducció*: s'estableix el context del projecte, es descriuen breument les aplicacions realitzades i s'enumeren els objectius.
- *Gamificació i Serious Games*: es defineix la gamificació i els serious games, així com les seves característiques i metodologia de creació. S'analitzen els tipus de serious games que existeixen i es comenten les seves particularitats.
- *Unity*: s'ofereix una visió global dels motors de videojocs i una visió concreta i detallada del motor de videojocs Unity.
- *GeniusUp*: es mostra una visió completa del videojoc GeniusUp, explicant objectius, estructura, implementació, verificació i conclusions.
- *Laberint*: es mostra una visió completa del videojoc del laberint, explicant objectius, estructura, implementació, verificació i conclusions.
- *Conclusions*: es recullen les conclusions extretes al llarg de tot el projecte, comentant els objectius assolits i l'experiència personal.

## GAMIFICACIÓ I SERIOUS GAMES

### 2.1 Estat actual

#### 2.1.1 Origen

Els darrers anys han vist una proliferació molt ràpida de software destinat a la consumició en massa que s'inspira en els videojocs. Aquesta metodologia es sol definir com a *gamificació*, una manera de treballar que connecta el gran nombre de conceptes i recerca ja feta en la interacció entre els humans i els ordinadors en el camp dels videojocs i l'aplica a altres àmbits.

Ara bé, és un concepte tan relativament modern que la seva definició concreta encara no és única, i els experts en el tema difereixen lleugerament en com enfoquen la *gamificació* i el seu ús. La definició comuna i acceptada és que la *gamificació* és simplement la introducció i ús d'elements de jocs a altres serveis. Però el concepte inicial data del 2008, i la seva adopció com a terme comú no va ser fins el 2010. Encara avui en dia es fan servir altres conceptes similars per a referir-nos al mateix fenomen, el que denota la novetat del concepte i la falta d'estudi i consens entre experts. [4]

#### 2.1.2 Definició

La idea de gamificació sorgeix del fet que els videojocs es poden resumir a una col·lecció de *múltiples condicions*. Cap d'aquestes condicions individualment pot constituir un joc, sinó que és la combinació de totes que forma i defineix el joc. Les definicions específiques d'aquestes condicions varien segons els autors, però sempre es basen en dos pilars, una *component sistemàtica* que descriu com el joc s'ha construït i una *component experimental* que descriu com s'involucra l'humà en el joc. Amb aquestes dues condicions es pot definir clarament un videojoc i els seus elements, i per tant llavors aquests es poden extrapolar a altres sistemes.

### 2.1.3 Nivells d'abstracció

Una manera resumida de classificar aquestes condicions és per nivells d'abstracció. El primer nivell d'abstracció, el més bàsic, engloba tots les definicions de videojoc. Defineix els videojocs com a un sistema format per conjunts de mecanismes i actors que interactuen entre ells—el que seria la component sistemàtica del sistema—i per com a mínim la participació activa d'un jugador—la component experimental.

El segon nivell d'abstracció ja inclou característiques més específiques, per tant ja no són presents a tots els videojocs. La component sistemàtica d'aquest nivell d'abstracció està composta per elements com les regles del joc, els objectius, els quals poden ser conflictius o no, i els resultats incerts. Aquests sistemes ja aporten en la seva component experimental experiències hedòniques al jugador—ja que hi ha un certs objectius a obtenir, els quals aporten satisfacció al jugador—i experiències més negatives com el suspens i la tensió—ja que el resultat i, per tant, l'èxit del jugador depèn del seu rendiment i actuació.

El tercer nivell d'abstracció encara està poc definit ja que, segons els experts, engloba aquells elements que s'usen única i exclusivament en els videojocs. Aquest últim nivell és una font de conflictes perquè, segons alguns autors, existeixen components experimentals úniques dels videojocs, i segons altres no. Els que estan a favor argumenten que sense aquestes condicions úniques no es podria reconèixer un videojoc. Raonen que si un joc està format per un conjunt de condicions, però cap d'elles es única dels videojocs, no es pot saber amb certesa que el sistema és tracta d'un videojoc i no un altre sistema amb una estructura similar. Amb la informació i coneixements actuals aquest darrer nivell d'abstracció queda en un estat neutre, on encara no s'ha arribat a un consens entre els experts que unifiqui i descrigui clarament les seves característiques. [5]

### 2.1.4 Serious Games

Una de les branques de la gamificació, encara que a cops les dues branques no es relacionen juntes, és el desenvolupament dels anomenats *serious games*. L'explosió dels videojocs, com ja hem comentat, va obrir noves portes a la resta de disciplines. No només eren els avanços tecnològics els que feien els videojocs atractius per a perseguir objectius distints de l'entreteniment pur, sinó que les noves metodologies que s'estaven creant per al desenvolupament de videojocs també eren dignes d'atenció.

Els dissenyadors de videojocs van adquirir habilitats per a involucrar emocionalment als jugadors i van identificar, gràcies a la recerca, nous factors importants per a gaudir un videojoc, com la presentació estètica dels continguts i l'ús de la narrativa. Això va demostrar que els videojocs són intrínsecament motivadors i que són capaços de crear en els jugadors un estat mental de fluïdesa, on es senten completament absorbits i immersos en l'activitat.

És evident que tots aquests factors són desitjables en la recerca, la investigació i l'aprenentatge. Altres mitjans ja ho havien provat. La televisió, amb programes infantils com Barrio Sesamo, pretenia transformar l'experiència de veure la televisió en un aprenentatge per a infants. Els esports també havien passat de ser un simple passatemps a ser recomanats per altres aspectes com la millora de la salut i la forma física. Per tant, si

altres disciplines ja assolien propòsits distints de l'entreteniment pur, perquè no els videojocs? [1]

A partir d'aquí va sorgir el concepte dels *serious games*, aquells videojocs creats amb la intenció d'entretenir i aconseguir al menys un objectiu addicional, sigui educatiu o de salut. Els objectius addicionals s'anomenen objectius característics. Ara bé, també és un concepte relativament jove, com la gamificació, i per tant encara no s'ha definit clarament, no s'ha arribat a un consens. Altres definicions caracteritzen els *serious games* no per la intenció amb la que han sigut desenvolupats, sinó amb la intenció del jugador. Els defensors d'aquesta definició creuen que qualsevol videojoc pot convertir-se en un *serious games* si el jugador no té com a única prioritat l'entreteniment, sinó que a més pretén millorar la seva velocitat de reacció o les seves habilitats motores, per exemple. [6]

La diferència entre la gamificació i els *serious games* és que conceptualment la gamificació és l'aplicació de les metodologies o elements dels videojocs a altres aplicacions, un exemple del qual serien els programes de lleialtat de les companyies que fan servir metes, medalles i recompenses per a motivar i involucrar els seus clients. Per tant la gamificació no és sempre necessàriament un joc, en canvi els *serious games* sí. Un cop feta aquesta introducció d'ambdós conceptes, en aquest treball ens centrarem principalment en els *serious games*.

## 2.2 Categories

Els *serious games* s'han definit com a jocs que tenen objectius característics més enllà del pur entreteniment. Aquests objectius poden pertànyer a varies competències, resumides en la figura 2.1.

Cada un d'aquests objectius s'igualava a una competència, sigui cognició, percepció, emoció, etc. Aquesta classificació dels objectius addicionals ens permet catalogar els *serious games* per categories. Una altra manera de agrupar-los, tot i que menys comuna, és segons el grup al que van dirigits.

Un altre factor important a l'hora de destriar els *serious games*, a part dels objectius addicionals i el grup al que van dirigits, és la seva àrea d'aplicació. Alguns exemples d'àrees d'aplicació serien jocs corporatius, jocs educatius, jocs de salut, jocs socials, entre moltes d'altres. [1]

### 2.2.1 Jocs corporatius

És la categoria de *serious games* més gran i més rellevant econòmicament. Alguns exemples d'aquesta categoria en són simulacions i programes d'entrenament de forces militars o policials, entorns d'entrenament virtual per a empleats en diversos sectors—conductors de tren, de bus, pilots, venedors, caixers de supermercat—, i jocs de gestió de negoci. El propòsit d'aquests jocs pot anar fins i tot més enllà del simple entrenament i arribar a ser una eina de màrqueting o reclutament.

Alguns d'aquests jocs han quedat només en l'àmbit privat de l'empresa o companyia que els utilitza, però en altres casos s'han comercialitzat en el mercat públic. Els simuladors de vol o els simuladors d'una granja en són exemples que s'han venut als entusiastes i fans d'aquest gènere.

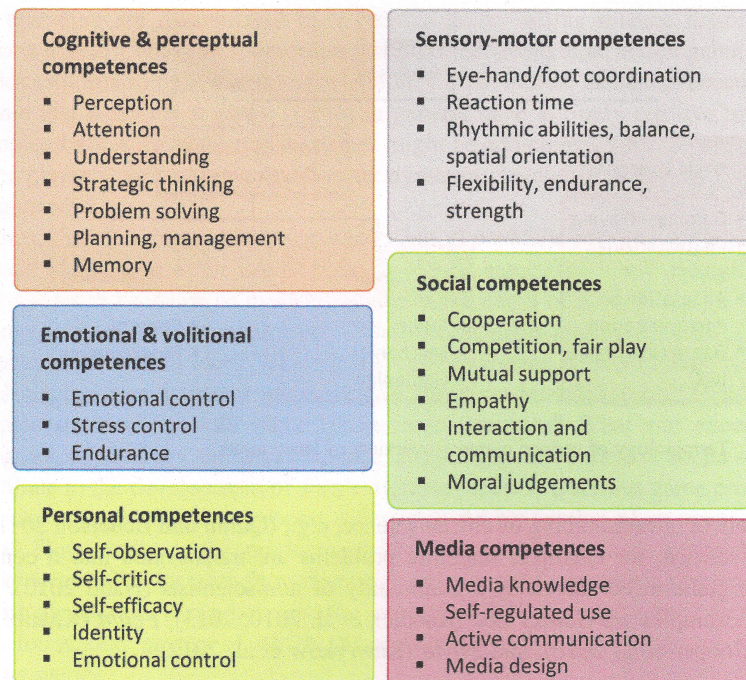


Figura 2.1: Competències dels serious games [1]

### 2.2.2 Jocs educatius

Els jocs educatius es semblen molt als d'entrenament i simulació, ja que el seu objectiu característic és el d'adquirir i transmetre coneixements, habilitats i competències als jugadors. La gran diferència resideix en el segon factor de classificació: el grup al que van destinats. Els jocs educatius es destinen molt més a grups d'infants, joves estudiants, famílies i gent major, enlloc de treballadors d'empreses, soldats o oficials de policia.

Molts d'aquests videojocs s'originen en projectes de recerca, els quals donen lloc tant a productes més *innovadors* centrats en un grup i unes competències a treballar específiques, com a altres més *comercials*, que busquen l'acceptació del públic general.

### 2.2.3 Jocs per a la salut

La tercera gran categoria dels *serious games* és la dels jocs aplicats a la salut, tant preventiva com rehabilitadora, amb exercicis físics que treballen l'equilibri, la coordinació o la força. En general, qualsevol joc que pretengui millorar la salut i l'estil de vida del jugador.

Una altra de les raons per la que han sorgit amb força és que actualment hi ha certes edats en què l'esport ja no és realitza tan sovint. L'estil de vida de moltes persones és molt més sedentari, amb menys activitat física, i el sistema sanitari és cada vegada més costós. Això no ha fet més que impulsar aquestes noves alternatives i iniciatives sanitàries.

De nou, com en el cas dels jocs educatius, hi ha molts jocs per a la salut, tant

aquells que provenen de la recerca com aquells d'àmbit més comercial. Aquests jocs més comercials no es centren en una característica o necessitat concreta, sinó que pretenen arribar al major nombre de persones, amb un èmfasi major a l'entreteniment i la diversió involucrades en l'esport.

Els jocs per a la salut és caracteritzen per tenir un pressupost de desenvolupament molt elevat comparat amb la resta de *serious games*, sigui provinent d'un contracte privat o del pressupost d'un estudi o recerca, així com un major nombre de projectes d'investigació i articles que en validen la seva utilitat.

### 2.2.4 Jocs socials

Els jocs socials adopten el seu nom del fet que es caracteritzen per centrar-se en problemes socials i públics rellevants, d'interès per a tota la societat. Alguns dels temes principals dels jocs socials són la política, la història, la religió, el sexisme, el racisme, el canvi climàtic, etc. Aquests projectes solen ser finançats o bé per institucions públiques, o bé per entitats sense ànim de lucre. En alguns casos empreses privades s'han involucrat econòmicament en un d'aquests projectes, però sempre com a campanya de neteja o millora de la seva imatge.

Comparteixen llavors un objectiu característic molt similar al dels jocs educatius, ja que pretenen informar als jugadors de temes socials rellevants, i fer-ho de la manera més divertida i motivant possible, però es diferencien en què els temes són d'interès molt més general i, per tant, van dirigits a tota la població, no a un grup de persones en concret.

### 2.2.5 Jocs persuasius

La idea darrera dels jocs persuasius és integrar els conceptes i metodologies dels jocs en els processos i activitats de cada dia. Per tant, és el que s'apropa més a la definició pura de gamificació. Així i tot, segueix sent un cas específic de la gamificació, ja que utilitza els conceptes i metodologies per a crear un nou joc que es pugui usar en la vida diària dels usuaris.

Per tant, per assolir aquest objectiu, és una àrea d'aplicació que fa ús de la tecnologia mòbil—la posició, l'àudio, la càmera, l'acceleròmetre, el giroscopi, etc.—. Llavors es combinen la informació que tots aquests dispositius aporten de l'usuari per tal de confeccionar un joc.

La vessant més habitual té un caire molt semblant als jocs per a la salut, ja que aporta als usuaris informació a temps real i mòbil sobre el seu estat. Però altres vessants també interessants pretenen revalorar el patrimoni cultural de ciutats o països. Aquests jocs més culturals permetrien als jugadors explorar ciutats, edificis o monuments de manera interactiva, aprofitant les possibilitats dels jocs persuasius. Una aplicació més comercial d'aquest darrer model és en l'àmbit turístic, on mitjançant un joc els turistes reben més informació sobre els monuments històrics que estan visitant.

Finalment els jocs persuasius no només transmeten un missatge cap als jugadors, com sí ho fan els jocs educatius o per a la salut, sinó que també s'utilitzen per a rebre informació dels jugadors. Així s'obtenen dades, tant geomètriques—dels llocs que visita l'usuari—com personals—els sentiments de l'usuari—. Aquesta informació pot

ser usada tan per projectes de recerca i investigació, com per projectes comercials de recollecció i mineria de dades. [7]

### 2.3 Aplicació al projecte GeniusUp

El projecte *GeniusUp* es basava en crear un videojoc que permetés als jugadors aprendre ciència de manera motivadora i entretinguda. Per tant, de les categories explicades, és evident que es tracta d'un *serious game* educatiu. No és corporatiu perquè el joc va destinat a joves adolescents, estudiants, no a treballadors adults.

En aquest cas en concret l'objectiu principal era treballar les competències relacionades amb la ciència, transformar els prejudicis i estereotips que molts joves tenen de què la ciència és avorrida i complicada, i incrementar l'interès i l'accessibilitat a una formació educativa més científica.

Així i tot, el joc s'ha desenvolupat com a aplicació mòbil, per tant, tot i que l'objectiu principal és la promoció de la ciència entre joves, l'interès comercial també hi és, ja que s'ha triat la plataforma més usada i accessible actualment—avui en dia gairebé tothom posseeix un smartphone.

Les branques científiques triades han estat les matemàtiques, la química, la física i la biologia. Cada branca té el seu joc, però per a fer el joc més proper als joves s'ha seguit l'estructura del joc per a mòbils extremadament popular *Candy Crush*, i per tant tot i que hi ha un joc per branca, tots es basen en el mateix model. El joc consisteix en un tauler amb certs elements, que varien segons la branca en concret, els quals s'han de combinar tant verticalment com horitzontalment per a fer combinacions conegudes.

En el cas de la química, el jugador ha de combinar elements químics per tal de realitzar composts. D'aquesta manera està aprenent formulació química dins un entorn competitiu, en el que s'encoratja l'autosuperació i s'anima al jugador a contínuament millorar els seus registres.

En el cas de la branca de biologia, l'altra branca implementada de moment, el jugador combina distints animals per tal d'agrupar-los per grups taxonòmics. De nou, el jugador està aprenent els grups taxonòmics en què es classifiquen els animals així com els seus integrants. Com que el format té el mateix format que la resta, el jugador també es troba dins un entorn competitiu que l'anima a millorar i superar-se a si mateix.

Per tant, el videojoc final és una manera accessible i fàcil de fer servir—al tractar-se d'un joc en plataforma mòbil—per els joves d'aprendre ciència de manera entretinguda i motivadora—les característiques del joc, restricció de temps, puntuació, nivells i rècord, entre d'altres, creen un entorn competitiu en el que el jugador està incentivat a superar-se constantment.

### 2.4 Aplicació al projecte Juguemos

El projecte *Juguemos* en canvi tenia una perspectiva diferent. El seu objectiu principal era la investigació, la recerca. Més concretament, la recerca en nens i nenes amb *paràlisi cerebral*, *autisme* i *dèficit d'atenció*. Com ja hem comentat el projecte englobava varis estudis, però molts es realitzaven sobre el mateix entorn, un *serious game* en forma de laberint. L'altre estudi en el que vaig participar desenvolupava una aplicació distinta, i



com ja hem comentat encara no es pot definir exactament com a *serious game*, però s'ha desenvolupat amb el mateix *motor de videojocs* que el laberint, amb previsió de què en un futur pròxim sí que se li hagin d'afegir elements més pròxims als videojocs.

Es tracten doncs de *serious games* per a la salut, ja que tot i no tenir una aplicació directa, els resultats que s'extreguin d'aquests estudis sí que tindran un ús en el camp de la salut i milloraran la qualitat de vida dels grups afectats. En el cas del laberint, a més, en segons quina versió del joc, el moviment de la bolla que ha d'escapar del laberint no es realitzarà mitjançant mètodes d'interacció tradicionals com el teclat, sinó que amb dispositius d'interacció adaptats als grups participants, els quals treballaran així les seves capacitats físiques i psíquiques.

Aquests *serious games* però ja no tenen la perspectiva comercial, com en el cas del projecte *GeniusUp*. Estan centrats completament en una perspectiva d'estudi d'investigació, on els participants són seleccionats d'acord a uns criteris i jugant al joc aporten dades per al posterior estudi. El videojoc en aquest cas no pretén enganxar als jugadors, mantenir-los jugant, sinó que utilitza els avantatges que aporten els videojocs per a realitzar un estudi, el qual d'altra banda podria resultar molt pesat i avorrit per als participants.

Llavors, no només pertany a una categoria diferent al projecte *GeniusUp*, sinó que la seva perspectiva també és completament diferent. El que comparteixen són els valors dels *serious games*, és a dir, ambdós utilitzen els valors de diversió i competició dels videojocs per a tal d'aconseguir el seu propòsit.

## 2.5 Futur pròxim

Els *serious games* són relativament recents, així que actualment encara no s'han definit les fronteres de les seves capacitats. El que sí sabem és que cada cop s'intenten usar en més disciplines, ja que s'ha demostrat que l'ús de metodologies de disseny, estructura narrativa, arts visuals i tècniques d'interacció presents en els videojocs en aplicacions que no tenen com a objectiu principal o únic l'entreteniment és un mètode efectiu de millorar els resultats i la motivació dels jugadors. [1]

Així i tot, els *serious games* encara no han explorat moltes de les característiques que els videojocs tradicionals usen per a fomentar l'entreteniment, per tant encara poden seguir millorant i desenvolupant noves idees que augmentin encara més l'entreteniment i la motivació dels seus jugadors, com ja ho han fet els videojocs que busquen el pur entreteniment.

Llavors, una de les línies i direccions que haurien de seguir els *serious games* en un futur pròxim es basa en seguir identificant quines característiques o elements dels videojocs són els que enganxen als jugadors, mesurar l'impacte d'aquests aspectes més rellevants i, finalment, trobar la manera més efectiva d'implementar-los en el seu propi àmbit, per així continuar la seva evolució, augmentar l'ús, reduir costos i millorar la seva efectivitat.



### 3.1 Motors de videojocs

Un *game engine* o *motor de videojoc*, és un software de treball que està dissenyat per a la creació i desenvolupament de videojocs. Aquests jocs poden anar destinats tant a *consoles*, com a *ordinadors*, com a *dispositius mòbils*. L'ús principal dels *motors de videojocs* és facilitar la creació al unificar moltes de les parts involucrades en la creació d'un videojoc en un mateix entorn, però sobretot el seu gran avantatge és la reducció de costos que comporta, ja que es pot usar el mateix *game engine* per a crear distints videojocs. Això permet reutilitzar parts d'altres jocs per a realitzar-ne de nous. En aquest projecte aquesta part ha estat molt important, ja que tant en el cas del *GeniusUp* com del laberint del projecte *Juguemos*, a partir d'una base simple s'han desenvolupat variants del mateix joc orientades a objectius completament diferents.

En la majoria de casos, els *motors de videojocs* posseeixen una sèrie de eines de desenvolupament, les quals solen ser el més gràfiques i simples possibles, per tal d'oferir al creador uns recursos que li permetin el desenvolupament ràpid del videojoc. Aquestes eines a més són reutilitzables i estan totes en un mateix entorn integrat. D'aquesta manera la majoria de *game engines* ja venen proveïts de funcionalitats d'elements bàsics en els videojocs com els gràfics, l'àudio, la física entre elements del joc o la intel·ligència artificial. Aquesta característica dels *motors de videojocs* és la que els categoritza més com a *programa intermediari*, ja que són un entorn de software senzill, flexible i reutilitzable, amb totes les funcionalitats bàsiques necessàries per a crear un videojoc ja implementades. Per tant, els *motors de videojocs* permeten reduir costos i complexitat en la creació d'un videojoc, reduint llavors el temps que tarda a sortir el producte al mercat—un factor importantíssim en una indústria tan competitiva com la del videojoc.

A cops la línia de separació entre els *motors de videojocs* i els videojocs en si no és molt clara. Hi ha motors que sí que estableixen una distinció molt clara, altres no tant. El que defineix aquest límit entre motors i videojocs és l'existència de regles lògiques o de joc codificades de forma rígida, és a dir, amb casos especials de codi per a objectes

específics del joc. En aquests casos és molt difícil reutilitzar el software per a crear un nou videojoc distint. Com ja s'ha comentat un *motor de videojocs* fa referència a aquell software que és extensible, que es pot fer servir com a base per a la creació de videojocs distints sense haver de modificar excessivament el nucli del motor.

Idealment un *motor de videojocs* podria executar i crear qualsevol tipus de joc. En la realitat la gran majoria de *motors de videojocs* estan elaborats per a crear i desenvolupar videojocs d'una plataforma específica. Com més general és el motor, menys òptim serà a l'hora d'executar jocs concrets, ja que al no tenir un software i hardware específic en ment, no es pot optimitzar el codi del motor. Així i tot, amb la contínua millora dels processadors i Graphics Processing Unit (GPU) dels ordinadors i la refinació dels algorismes de representació—els quals són cada cop més eficients i estructurats—, aquesta diferència entre *motors de videojocs* orientats a distints gèneres i plataformes s'està suavitzant. De totes maneres, de moment, el millor motor per a un videojoc serà aquell que estigui preparat i creat específicament per a satisfer les necessitats d'aquest videojoc en concret.

## 3.2 Arquitectura

L'arquitectura dels motors de videojocs no és única, ja que no existeix una sola base a partir de la qual crear qualsevol videojoc. La primera diferència principal que categoritza les arquitectures dels motors de videojocs és si aquesta està destinada a un gènere de videojocs o plataforma específica—el que s'anomena una *arquitectura estàtica*—, o si del contrari l'arquitectura és independent del gènere i plataforma—el que es coneix com a *arquitectura dinàmica*.

La gran majoria de *motors de videojocs* actuals es basen en una arquitectura modular, on existeixen distintes parts anomenades mòduls les quals interactuen entre ells. Aquesta modularització permet que certs components puguin ser fàcilment canviats per altres sense afectar a la resta de mòduls. Un clar exemple d'aquest concepte és l'ús de les APIs gràfiques, ja que un bon motor de videojocs permetrà canviar l'API gràfica de *OpenGL* a *DirectX* per exemple, o viceversa. Alternativament la portabilitat multiplataforma també seria un exemple de modularització. [8]

La figura 3.1 il·lustra genèricament la arquitectura dels motors de videojocs

### 3.2.1 Hardware

La capa de *hardware* és la més inferior, i descriu la màquina en la que s'executa el joc. Bàsicament es pot classificar en quatre grups:

- *Dispositius portables*: Nintendo Switch, Nintendo 3DS, Playstation Portable, etc.
- *Smartphones*: engloba tant els mòbils com les tablets amb iOS o Android.
- *Consoles*: Playstation 4, Xbox One, Wii U, etc.
- *Ordinadors*: engloba tant els de sobretaula com els portàtils

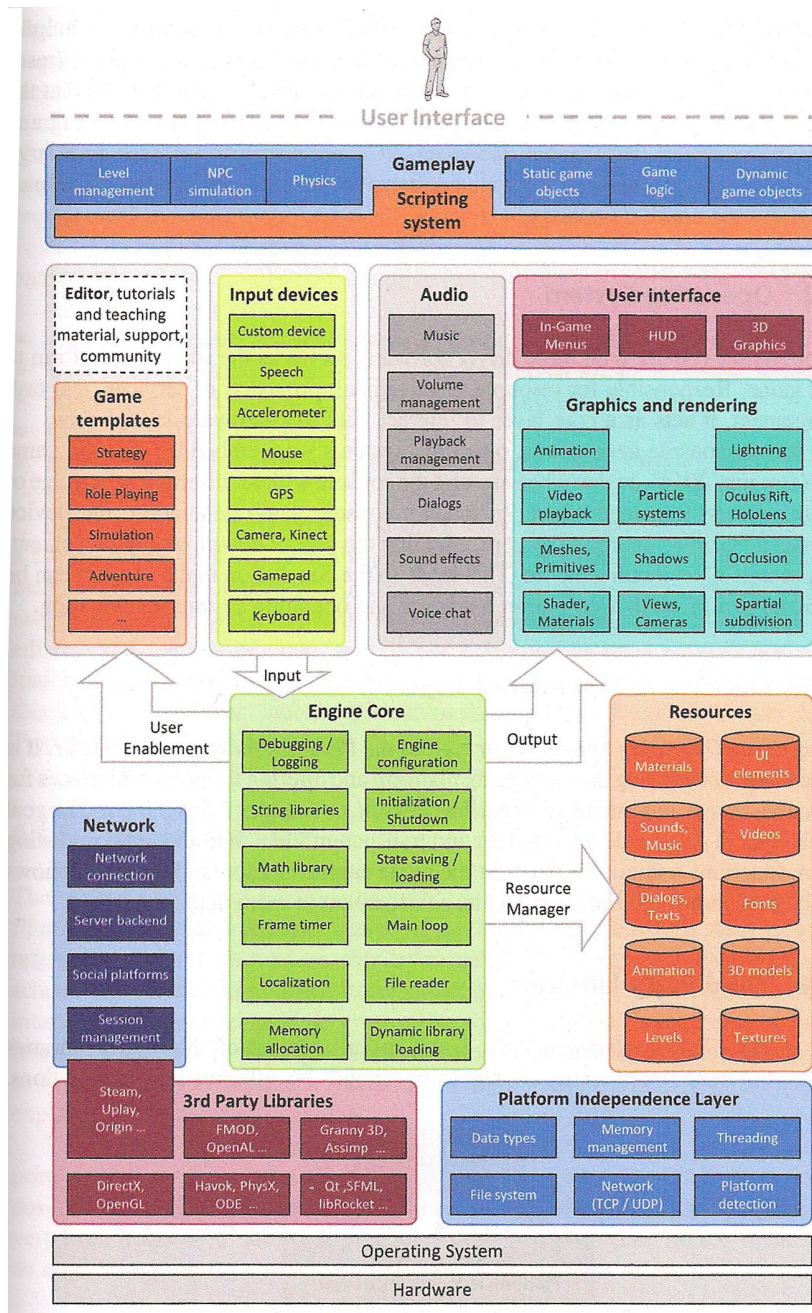


Figura 3.1: Arquitectura del motor de videojocs Unity [1]

### 3.2.2 Sistema operatiu

El *sistema operatiu* és la plataforma de software en la que s'executa el joc. És el responsable de gestionar els recursos, els serveis i les tasques del joc, a més de ser la capa que interacciona amb el *hardware* de la capa inferior. Avui en dia fins i tot les consoles tenen el seu *sistema operatiu* propi, el qual permet a l'usuari baixar jocs, escoltar música o veure

pel·lícules. Per tant, avui en dia, les consoles poden ser comparades amb els ordinadors des de la perspectiva d'aquestes dues primeres capes, ja que ambdós dispositius tenen un determinat *hardware* i sistema operatiu que interactuen entre ells.

#### 3.2.3 Capa d'independència de plataforma

Aquesta capa és just a sobre de la capa del *sistema operatiu*, i és la responsable de detectar la plataforma en la que s'executa el joc i oferir les interfícies adequades per a accedir a fitxers, comunicar-se a través de la xarxa o gestionar la memòria. Aquesta capa pretén aportar una vista consistent i transparent a les funcions principals del sistema sense que el *motor del videojoc* hagi d'intervenir.

#### 3.2.4 Llibreries externes

Al mateix nivell que la *capa d'independència de plataforma*, la capa de *llibreries externes* és la que dona suport a la funcionalitat del *motor de videojoc* mitjançant funcions, algorismes i marcs de referència. Un exemple de *llibreria externa* seria la llibreria PhysX, la qual aporta al *motor de videojoc* funcions per a simular fluids, articulacions, col·lisions, entre d'altres. Les *llibreries externes* contenen aquelles funcionalitats que són independents del joc i del motor amb el que s'ha creat. Evidentment aquestes llibreries hauran de ser adaptables a la funcionalitat que es necessiti.

Alguns altres exemples en són:

- *Àudio i multimèdia*: FMOD, OpenAL.
- *Interfície d'usuari*: Qt, libRocket, SFML.
- *Animació i modelització*: Granny 3D, Assimp.
- *Gràfics API*: OpenGL, DirectX.
- *Distribució digital*: Steam, Uplay, Origin.

#### 3.2.5 Nucli del motor

L'element central de qualsevol *motor de videojocs*, basat en la *capa d'independència de plataforma* i les diverses *llibreries externes* que s'usin. El *nucli* manté l'estat del motor, s'ocupa de l'assignació i manteniment dels diversos fils del motor, i gestiona totes les interaccions amb els mòduls o capes superiors—com per exemple els *gràfics* o els *dispositius d'entrada*.

Alguns dels exemples dels mòduls del nucli del motor són:

- *Debugging i logging*: permet imprimir l'estat del sistema, per tal de poder detectar errors o veure com funcionen certs elements dins del joc. Això és degut a que el bucle principal del joc s'executa a una freqüència molt elevada—30 frames per segon com a mínim, 60 o més seria l'ideal—, i per tant per a poder saber què està fallant a un moment determinat convé enviar un missatge o ser capaç d'anar instrucció a instrucció.

- *Llibreria matemàtica*: conté les operacions bàsiques com la suma, la resta, la multiplicació o la divisió, però a més és capaç de calcular aquestes mateixes operacions en vectors i matrius—elements bàsics de la computació gràfica—. En segons quins casos fins i tot permet l'ús d'elements més complexos com els quaternions o splines.
- *Mòdul d'assignació de memòria*: és el responsable de carregar els models, les textures i els sons a memòria, per a que mentre el jugador jugui no hi hagi cap tipus de retard o baixada de frames degut a què algun element no s'ha carregat. Implementa un recol·lector de fems, el qual detecta si un element que ocupa memòria es tornarà a fer servir o no. D'aquesta manera es poden anar carregant només les parts a les quals el jugador pot accedir, assegurant que no hi hagi temps de càrrega mentre aquest està jugant.

### 3.2.6 Xarxa

Els jocs es poden classificar en locals, d'un sol jugador o jocs en xarxa. La capa de xarxa evidentment només existeix en els jocs *en xarxa* o *multijugador*. Consisteix en tots aquells procediments que permeten establir, gestionar i monitoritzar les connexions entre el *client* o clients del joc i el *servidor*. Per tant ha de determinar quin protocol es fa servir, gestionar la sessió d'usuari, la connexió al servidor, etc.

Les tasques es divideixen entre les que es realitzen al servidor i la que es realitzen al client. Al servidor es fan aquelles tasques que afecten a l'estat global del joc. És a dir, aquelles tasques més lògiques, de l'estat del joc i situació dels jugadors, no de la representació visual. Alguns exemples en són:

- Actualitzar la posició dels objectes en el món virtual.
- Comprovar col·lisions entre elements.
- Mantenir l'estat global del joc (puntuació, recursos, habilitats adquirides per el jugador, etc.).
- Verificar que els clients estan connectats.

A la part de client es realitzen més aquelles tasques que no afecten a l'estat global, sinó només a la representació visual al dispositiu del jugador. Per tant alguns exemples de tasques en són:

- Animar els models.
- Renderitzar els objectes.
- Crear els efectes de partícules.
- Generar sons i música.

#### 3.2.7 Gestor de recursos

Els recursos dels videojocs són el que formen la visualització gràfica del joc, i són essencials per a la seva lògica—materials, elements de la interfície d'usuari, sons, música, vídeos, texts, models, esquelets d'animació, nivells, textures, etc.—. Aquests recursos es combinen per a generar la immersió del jugador en el món virtual.

Amb els avanços de la tecnologia els recursos han anat orientant-se cap a una perspectiva més artística, sigui per crear models o textures ultrarealistes, com per replicar estils o paletes de colors més artístiques—8-bit, tipus dibuixos animats, manga, colors pastel, blanc-i-negre, etc.—. A més, els recursos ja no només es separen segons el seu tipus, sinó també segons la plataforma a la qual van dirigida. No és necessari el mateix tipus de detall per a un element que es veurà a una pantalla amb resolució alta, que per a un que es veurà a la pantalla d'un mòbil.

#### 3.2.8 Dispositius d'entrada

Gairebé el 90% dels videojocs tradicionals estan controlats amb el *ratolí*, el *teclat*, o un *controlador*—sigui tipus joystick o un ja específic com el de *Playstation 4* o el de *Xbox One*—. Així i tot, en els darrers anys s'ha començat a experimentar amb altres tipus de dispositius d'entrada. La *Wii* o la *Kinect* en són exemples més comercials, el *Leap Motion* o el *Virtuix Omni* en són exemples més innovadors però que encara estan en una fase inicial de desenvolupament.

En el cas dels *serious games* la necessitat o, més ben dit, la utilitat d'altres dispositius d'entrada és més patent que en els videojocs tradicionals. Simuladors de vol, entrenadors mèdics o jocs rehabilitadors necessitaran tots un dispositiu d'entrada específic a la necessitat o objectiu que cobreixen.

#### 3.2.9 Àudio

L'àudio ha augmentat molt la seva importància recentment. Fa uns anys únicament necessitava poder reproduir *efectes sonors* i *música* en seccions o situacions determinades del joc. Avui en dia els jocs moderns en fan un ús molt més complet. Han aparegut els *diàlegs* entre personatges que són cada cop més fluids, realistes i naturals, i *xats de veu* en jocs multijugador que permeten al jugador comunicar-se i crear estratègies amb la resta de jugadors. A més el so s'ha refinat per tal de què s'adapti a l'espai virtual 3D on es produeix. D'aquesta manera el jugador, gràcies als auriculars, pot sentir el so provenir de l'esquerra o la dreta, el pot sentir més fort o més fluix segons la distància a la que es troba el seu personatge de l'objecte que emet el so, i fins i tot pot sentir efectes com l'eco.

#### 3.2.10 Gràfics i rendering

El càlcul i la renderització de gràfics és una de les tasques més complexes a l'hora de dissenyar un motor de videojocs. Actualment s'usa el que s'anomena *pipeline rendering*, és a dir, el procés de rendering es realitza a través de diverses passes. Les passes són les següents:

- Calcular les transformacions geomètriques.



- Calcular la il·luminació.
- Convertir la imatge a mapa de bits.
- Aplicar textures.
- Aplicar textures.
- Aplicar shading.
- Aplicar anti-aliasing.
- Calcular oclusions.

Aquests càlculs són molt costosos computacionalment i per tant es realitzen a la GPU. A més del *rendering*, la capa de gràfics d'un *motor de videojocs* també realitza altres tasques com l'animació dels models o la detecció de col·lisions. Com que la gran majoria de jocs han de fer tots aquests càlculs gràfics en temps real, el *motor de videojocs* té varies optimitzacions per a fer el procés més ràpid.

#### 3.2.11 Interfície d'usuari

La interfície d'usuari sol treballar per sobre del *mòdul de gràfics*, i s'encarrega de mostrar i gestionar els elements que formen la interfície d'usuari, com per exemple els botons o els texts. Aquests elements o bé aporten informació al jugador del seu estat i de les interaccions que realitza durant el joc, o bé serveixen per a crear menús amb els que el jugador pot interactuar per a guardar el seu joc, carregar-ne un de ja guardat, començar-ne un de nou, canviar les opcions o directament sortir del joc.

#### 3.2.12 Gameplay i scripting

Aquest darrer mòdul és el que s'encarrega de gestionar com funciona el joc. Per això, es fa servir un llenguatge de programació d'alt nivell. Els més usats són *Lua* o *C++*, però actualment molts altres com *Python*, *Javascript* o *C#* també es poden usar. Aquests llenguatges es poden comunicar amb el *nucli del motor* i poden ser part del debugging per tal de poder veure l'estat del joc en un moment determinat del codi. Els *scripts*-*fitxers* del llenguatge de programació en qüestió—determinen com actua el joc i els seus elements, per exemple, el comportament dels personatges que no controla el jugador normalment s'especifica mitjançant un *script*, ja que així se'ls pot fer reaccionar dinàmicament a situacions específiques durant el joc.

### 3.3 Característiques

Per a la realització de tots els projectes, es va decidir fer servir el motor de videojocs *Unity*. El gran avantatge que tenia era que la seva versió gratuïta tenia tot el que es podia necessitar per a realitzar els projectes, a més de ser un motor amb molts de manuals educatius fàcilment accessibles. Un altre avantatge era el que *Unity* anomena la *Asset Store*, una botiga on es poden adquirir distints elements que llavors es poden inserir fàcilment a dins l'entorn del joc.

### 3. UNITY

Amb la realització del videojoc *GeniusUp*, el laberint de *Juguemos*, i l'entrenador d'emocions—també del projecte *Juguemos*—mitjançant *Unity*, queda clar un dels punts forts dels motors de videojocs: la versatilitat. Seguint un model simple de desenvolupament de videojocs, s'ha pogut crear de manera molt similar dins el mateix *motor de videojocs* tres aplicacions distintes, tant en el seu aspecte, com en la seva funcionalitat, com en el seu objectiu.

A la figura 3.2 es poden veure les principals característiques de *Unity* comparades amb els dos altres motors de videojocs més populars.

	Unity 5	CryENGINE 3	Unreal Engine 4
Supported platforms	Windows PC, Mac OS X, Linux, SteamOS, iOS, Android, Windows Phone 8, Blackberry 10, Tizen, WebGL, WebPlayer (Plugin), PlayStation 3, PlayStation 4, PS Vita, Xbox One, Xbox 360, Wii U	Windows PC, iOS, Android, Xbox One, Playstation 4, Wii U	Windows PC, Mac OS X, Linux, SteamOS, iOS, Android, HTML5, PlayStation 4, Xbox One
Development environment	Windows XP SP2, Windows 7, Windows 8, Mac OS X >10.8	Windows Vista SP1, Windows 7, Windows 8	Windows 7 64 bit, Mac OS X > 10.9.2
Engine source code access	Yes (not free)	Yes (not free, full license required)	Yes
License costs	Free (personal edition), \$75 per month (professional edition), no royalties	\$9.90 USD per month, no royalties (standard subscription)	Free to use, 5 % royalty on gross product revenue after the first \$3000 USD per game per calendar quarter
Limited to a specific genre	No	No	No
Programming languages	C#, UnityScript, Boo	C++	C++

Figura 3.2: Comparació de Unity amb altres motors de videojocs [1]

Una altra raó per triar *Unity*, a més de les ja comentades, és el seu suport per a *multiplataforma*. En qualsevol joc és desitjat que com més gent el pugui jugar millor, ja que més jugadors representen més ingressos i reconeixement per als creadors. En el cas dels *serious games* això no és sempre així, en segons quins casos el suport per a *multiplataforma* no és necessari.

Aquest és el cas de les dues aplicacions del projecte *Juguemos*. Al estar dirigides directament a un grup específic de persones, i tenint en compte que l'entorn en què es jugarà el joc serà controlat i conegut, no fa falta que els jocs es puguin executar en més d'una plataforma.

En canvi, el videojoc del *GeniusUp* pretén arribar com a més joves millor, i per tant el suport *multiplataforma* era desitjat. Amb *Unity* donar aquest suport és molt senzill, i això a permès crear el mateix joc tant per a mòbils *iOS*, com mòbils *Android*, com fins i tot executables per a *ordinadors*.

## GENIUSUP

### 4.1 Estat del projecte al moment de la meva incorporació

El projecte *GeniusUp*, com s'ha comentat en la introducció, va néixer originalment amb el nom de *PotuGame*, però al moment en què m'hi vaig incorporar ja s'havia fet el canvi cap a *GeniusUp*. Del projecte POTU s'havien creat unes cartes similars a les conegudes cartes d'*Uno* que, d'acord amb els objectius bàsics del projecte, fonamentaven l'aprenentatge de la ciència mitjançant l'entreteniment que proporciona un joc. Les cartes estaven dividides per colors segons les branques de *matemàtiques, física, química i biologia*, i cada carta en particular contenia informació sobre un científic determinat d'aquella branca científica en concret. A més, les cartes afegien un codi Quick Response (QR) amb el qual obtenir més informació del científic en qüestió i també s'estava creant una aplicació que permetés la interacció amb les cartes via realitat augmentada.

L'objectiu del *PotuGame* és avançar una passa més en les estratègies ja desenvolupades en les anteriors edicions del projecte POTU, afegint una component nova i multidisciplinària que permeti als estudiants de secundària, professors i investigadors de la UIB compartir coneixements fent servir un llenguatge i món comú. Per això es considera que les tecnologies relacionades amb els videojocs són les més adequades per a transmetre aquests coneixements de forma interactiva i lúdica, incrementant la participació massiva de tots els components de l'àmbit docent a tots els nivells. [9]

Dintre del mateix marc del projecte POTU expressat a la figura 4.1, s'havia començat a desenvolupar una primera versió de l'aplicació per a mòbils *GeniusUp*. La idea era seguir el mateix principi que les cartes, dividir el joc en quatre mòduls—un per a cada branca—similars, els quals constituïssin un joc en ells mateixos però alhora tinguessin unes certes característiques en comú. Al moment de la meua incorporació però, l'única versió que s'havia començat a implementar era la química, però encara estava molt enfora de poder ser considerada funcional. L'estructura del joc estava definida, així com gran part de la seva lògica. Ara bé, hi havia molts de falls que no estaven detectats, altres que sí que ho estaven però no s'havien arreglat, i sobretot mancaven molts elements tant de la interfície gràfica d'usuari com del propi món virtual.

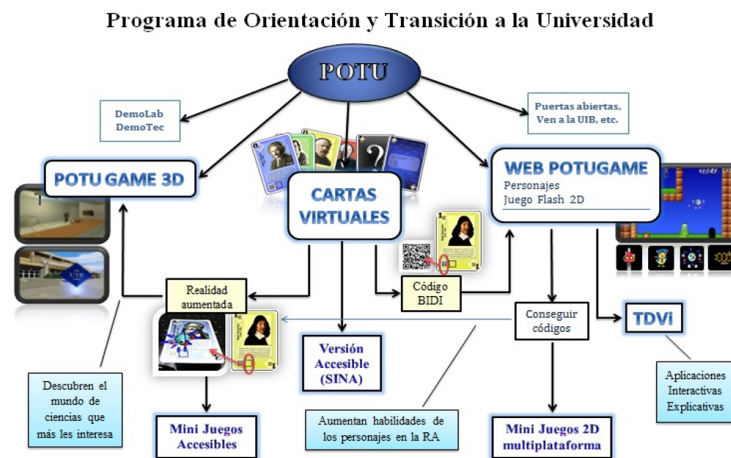


Figura 4.1: Resum d’iniciatives del projecte POTU

La meva feina era convertir aquell inici d’aplicació en una funcional, validada per científics i que es pogués distribuir al públic general.

## 4.2 Objectius

Els objectius de l’aplicació *GeniusUp* es poden dividir en dues categories:

- *Objectius generals*: tots aquells objectius heretats del projecte original Potu.
- *Objectius específics*: tots aquells objectius nous i concrets de l’aplicació mòbil.

L’objectiu general principal del projecte *GeniusUp* és fomentar la ciència entre la gent jove. Per a fer-ho s’ha decidit emprar un mètode de transmissió de coneixements proper als estudiants, un mètode que els sigui conegut, que els motivi a millorar i a voler aprendre. El mètode de transmissió que s’ajusta millor a aquest objectiu són els *serious games*. Si l’aplicació es tracta com un joc, els joves estaran aprenent ciència gairebé sense ni adonar-se’n, simplement jugant al joc.

A més, un dels altres objectius generals és que aquesta educació científica sigui el més natural i fluïda possible, que s’allunyi dels valors que habitualment s’associen a les assignatures científiques—tals com que són molt complicades o avorrides—. Com que el videojoc té puntuacions, rècords i límits de temps—entre molts d’altres elements—, està incentivant constantment als jugadors a superar les seves puntuacions anteriors, a passar-se el següent nivell, a millorar el registre del seu amic. D’aquesta manera s’incrementaria el temps jugat, però d’una manera autònoma, on no se li força res al jugador, sinó que ell mateix vol i decideix seguir jugant. Aquest augment en el temps jugat no fa més que incrementar els coneixements científics apresos pels jugadors a través de l’aplicació.

Finalment el darrer objectiu general del projecte és que la plataforma amb la que es fomenta la ciència sigui coneguda i pròxima als joves. Per això s'ha triat un videojoc, ja que és un dels elements més presents en la vida dels joves adolescents. Ara bé, per a fer-ho encara més accessible i proper, s'ha decidit crear el videojoc per a dispositius mòbils. Els mòbils formen encara més part del dia a dia dels joves, han tingut una explosió brutal aquests darrers anys amb l'aparició de les noves aplicacions de xarxes socials, missatgeria o jocs. A més, les possibilitats dels *smartphones* han augmentat exponencialment, avui en dia no hi ha pràcticament restriccions a l'hora de crear noves aplicacions. Per això els dispositius *Android* i *iOS* són les plataformes més properes als joves, que assegurin una interacció senzilla i natural, sense necessitat de cap explicació sobre el funcionament de l'aplicació, i que a més permeten crear-la tal i com es desitja, sense restriccions creatives importants.

D'altra banda els objectius específics eren aquells que, dintre del marc definit pels objectius generals, descriuen què és vol assolir amb l'aplicació o amb alguna part o mòdul d'aquesta en concret. Per tant seran molt més concrets, objectius que potser no varen sorgir a l'inici del projecte però que a una etapa posterior van cobrar sentit. Ja no es centren en definir el marc i l'abast del projecte, sinó que defineixen l'aplicació en si, la direcció que pren el seu desenvolupament i gairebé es pot dir que justifiquen certes decisions preses.

El primer objectiu específic que crida l'atenció és purament comercial. El projecte *GeniusUp* provenia del projecte *PotuGame*, com ja s'ha comentat diverses vegades. La falta de pressupost a aquest projecte va posposar futures línies de treball que es tenien previstes, com l'anomenada millora de les cartes *Potu* mitjançant la realitat augmentada. El mercat de les aplicacions mòbils és molt competitiu, però a l'hora molt lucratiu. En l'època actual on la informació està a l'abast de tothom que té una connexió a internet, basta una mica de reconeixement per a què una aplicació exploti i les seves descàrregues comencin a pujar. Al realitzar l'aplicació per a dispositius mòbils un dels objectius que es busca assolir és obtenir aquest reconeixement, el qual donaria una molt bona imatge del projecte i ajudaria a l'hora de demanar noves ajudes o obtenir finançament.

El segon objectiu específic era replicar el sistema i les regles del famós joc per a mòbils *Candy Crush*. Tot i que ja s'ha justificat la decisió de la plataforma, el joc en si podia ser relativament de qualsevol gènere. Un dels punts forts dels *serious games* és la seva versatilitat, per tal d'aconseguir un objectiu determinat no s'ha de limitar el gènere del joc a un en concret. Però d'acord amb l'objectiu general de crear un videojoc proper als joves, s'ha decidit seguir un model de joc que funciona—molt conegut a més—i adaptar l'objectiu a assolir al model, i no viceversa. A més, així s'ha tingut una referència final, el que ha facilitat la concepció del videojoc i el seu desenvolupament—en tot moment podíem comparar l'estat del nostre joc amb el model final que volíem assolir—. Finalment, el *Candy Crush* té un avantatge molt clar i és que no té fi. Al funcionar per nivells sempre se'n poden afegir de nous. D'aquesta manera, al imitar el seu model, si el joc agrada i funciona, podem seguir afegint nous nivells i, per tant, nous conceptes a aprendre.

Un altre objectiu específic era modularitzar l'aplicació. Com hem dit anteriorment el joc es dividia en les quatre branques de la ciència. Però per evitar favoritismes d'una branca sobre la resta, un objectiu va esdevenir crear els quatre mòduls a partir d'un mateix model de joc. Així no només equilibrem els quatre models de joc sinó que a més

facilem a l'usuari l'ús de l'aplicació, ja que no haurà d'aprendre el funcionament d'un nou joc al passar d'una branca a l'altra. Aquest objectiu és força important ja que limita l'estructura del joc, i per tant afecta a tot el procés de creació. Això té avantatges i inconvenients, els quals comentarem més endavant, però de nou demostra la versatilitat dels motors de videojocs, ja que a partir d'un model de joc se n'han creat quatre, cadascun amb els seus elements gràfics propis.

Tornant a la idea de intentar fer arribar l'aplicació a com més persones millor, un dels objectius específics que va sorgir durant la creació va ser que el videojoc tingués múltiples idiomes. S'han implementat tres idiomes de moment: català, castellà i anglès. Els dos primers tenen una justificació d'àmbit més local. Pretenen assegurar que els jugadors aprenen els conceptes en els dos idiomes de la comunitat autònoma. A més, el català permet que l'aplicació es pugui fer servir a comunitats com *Catalunya* o *València*, i el castellà a la resta de l'estat espanyol i fins i tot a països de l'*Amèrica del Sud*. L'anglès té una justificació més global, pretén assegurar que l'aplicació es pot jugar en un idioma que es pugui entendre a la resta del món.

Com que el videojoc vol educar als joves en la ciència, tots els conceptes científics que apareixen dins el joc han d'haver estat validats per experts. És per això que les propostes de joc de cada branca, així com el contingut de cada nivell, s'han validat amb un grup d'experts de la UIB per tal d'assegurar la seva validesa i adequació al nivell educatiu dels jugadors.

Finalment el darrer objectiu específic important era que, d'acord amb la normativa d'accessibilitat actual, el joc s'adaptés al màxim nombre d'usuaris possibles. Evidentment aconseguir l'accessibilitat total és molt costós. Així que decidirem descartar nous mètodes de interacció—que no fossin la pantalla tàctil del mòbil—i centrar-nos en l'àmbit visual. Implementarem un mode *colorblind*, per tal de què jugadors amb *protanopia*, *deuteranopia* o *tritanopia* poguessin tots distingir els distints elements del joc.

Evidentment hi hagueren altres objectius durant tot el procés de desenvolupament, però aquests eren els de més alt nivell. La resta serviren per a guiar certes decisions, o per a decidir com s'anava actualitzant el joc, però no són suficientment importants com per a ser comentats en aquest treball.

### 4.3 Motiu per el que és un serious game

L'aplicació *GeniusUp* és primàriament un *serious game* educatiu, ja que el seu objectiu pretén formar i educar als joves en les quatre branques principals de la ciència. Altres objectius com el de transformar la manera com els joves veuen i entenen la ciència—passar de ser un camp avorrit i complex a un divertit i interessant—podria servir com a argument per a classificar el videojoc com a un *serious game* social. Però el joc s'ha orientat més a la perspectiva dels joves, no de tota la societat, per tant crec que és més adequat classificar-lo com a educatiu.

Com ja hem explicat abans els *serious games* educatius pretenen transferir coneixement i habilitats de la manera més natural i entretinguda possible. En el nostre cas l'aplicació transmet coneixements bàsics sobre la biologia, la química, la física i les matemàtiques.

Un *serious game* educatiu es pot estructurar en un espai de coneixement i un espai de continguts del joc. Aquesta estructuració té l'avantatge d'aportar molt més control a la fase de generació i creació del joc. La separació entre els coneixements i els continguts del joc permet que, d'una banda, es puguin decidir i anotar els continguts que es volen ensenyar i definir l'estratègia amb la que es presentaran als jugadors, i d'altra banda, es pugui categoritzar el contingut dels jocs d'acord a nivells de dificultat i decidir com agrupar i relacionar el contingut de cada nivell. Finalment les dues parts s'han de posar en comú per tal de crear el joc en si. (figura 4.2)

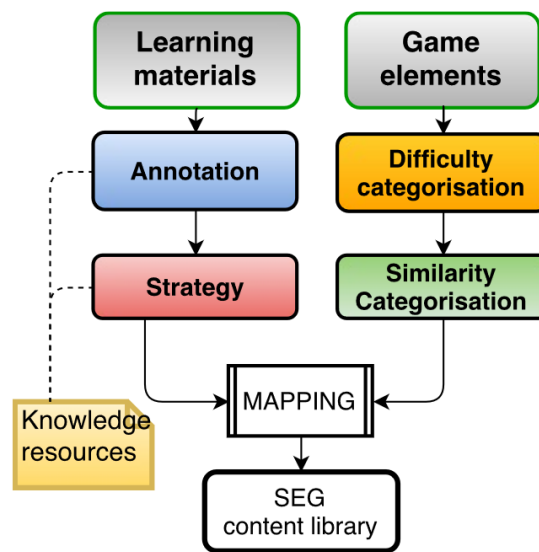


Figura 4.2: Marc de desenvolupament de SEG (figura original de [2])

Per a anotar i decidir els continguts s'ha de tenir en compte el seu ús i utilitat—no és el mateix educar nou vocabulari que noves fórmules matemàtiques—i la seva representació dins del joc. L'estratègia permet estructurar la progressió educativa del jugador, la manera en què aprén els nous conceptes. La categorització en nivells de dificultat permet separar els jugadors segons la seva habilitat, i permetre que jugadors amb distintes habilitats puguin jugar al mateix joc. Finalment, la selecció de continguts de cada nivell—així com les relacions entre ells—permet adaptar els continguts educatius al gènere i tipus de joc que es vulgui desenvolupar. [2]

Concretament, en el GeniusUp, a l'hora de decidir els conceptes i coneixements educatius a transmetre, les quatre branques de la ciència triades són tan amples que s'havia de definir què es volia fer ensenyar exactament, ja que era impossible fer un joc que ho expliqués tot.

Primer es va decidir que tot els conceptes transmesos per l'aplicació serien relativament genèrics, no s'entraria en detalls específics. El joc volia educar els joves en la ciència, interessar-los en el que aquesta podia oferir. Per tant no podíem decidir que un camp de les matemàtiques, l'àlgebra per exemple, era més interessant que un altre com la geometria. Llavors els conceptes de cada branca havien de ser genèrics, conceptes

que qualsevol jove estudiant hagués de saber.

L'altre punt a decidir era el nivell del coneixement a transmetre. És va decidir que com que el projecte era de la UIB, es voldria arribar als joves que pròximament hi volguessin anar. Per tant el nivell acadèmic dels conceptes que s'aprenen a través de l'aplicació són d'entre 4t d'ESO i 2n de Batxillerat. Al moment de realització d'aquest treball només s'havien definit el tipus de joc per a les branques de química i biologia. En la de química es va triar el tema de la formulació química i en el de biologia la classificació taxonòmica dels animals.

Un cop triat el coneixement que el videojoc ha d'ajudar a aprendre, s'ha de definir els continguts i estructura del joc en si. Això és el que defineix com s'aconsegueix que aquesta educació a través del joc sigui entretinguda i divertida. El model de joc ja s'ha explicat que era similar al del popular joc *Candy Crush*. En aquest el jugador té una graella plena de caramels de diferents colors. Pot intercanviar un caramel per un que està immediatament a sobre, a sota, a l'esquerra o a la dreta. Si amb aquest intercanvi aconseguix ajuntar tres o més caramels del mateix color aquella combinació desapareix de la graella i atorga punts al jugador. El jugador realitza totes les combinacions que pot en un temps determinat per el nivell i, quan aquest temps s'acaba, si ha fet suficients punts passa al següent nivell, sinó l'ha de repetir.

La gràcia que té aquest model de videojoc és que el jugador està limitat per el temps, però no per el nombre de combinacions que pot fer durant aquest límit de temps. Per tant l'usuari sempre pot intentar superar la seva puntuació en un nivell determinat, ja que no hi ha màxim. A més, al tenir un límit de temps, el jugador està contínuament sota la pressió competitiva d'aconseguir la puntuació necessària per superar el nivell. Si aquestes dues variables s'ajusten bé, el que s'aconsegueix és que el jugador s'enganxi al joc, ja que sempre està intentant o bé superar el seu rècord en un dels nivells que ja s'ha passat o bé aconseguir passar-se el darrer nivell al que ha arribat. La dificultat de nivells anirà definida per la complexitat de les combinacions a realitzar, així com del temps que es dona per fer-les i la puntuació mínima a assolir.

## 4.4 Implementació amb Unity

### 4.4.1 Estructura general

En el motor de videojocs Unity es creen escenes, on cada escena té una sèrie d'elements tan *gràfics*—botons, imatges, textures, etc.—com *lògics*—scripts de programació que guien el comportament del joc—. Els elements gràfics són una sèrie del que *Unity* anomena *GameObjects*, objectes genèrics als quals se li poden assignar components. Aquestes components varien tant l'aspecte visual com el comportament del *GameObject*. Els components gràfics com *Text*, *Image* o *Transform* canvien l'aspecte o la posició dins el món virtual d'acord a uns paràmetres especificats per l'usuari. Els components lògics com *Script*, inicialitzen l'objecte—mitjançant la funció *Start()*—, canvien el comportament de l'objecte durant la seva vida—mitjançant la funció *Update()*—i finalment responen a esdeveniments com el clic de un botó o el canvi d'una variable.

Les escenes creades per al *GeniusUP* i les relacions entre elles es mostra a la figura 4.3.

El flux del joc comença a l'escena *Splashscreen*—s'anomena així en anglès aquella pantalla amb el logotip inicial de la UIB—, la qual ens porta al menú principal. D'allí



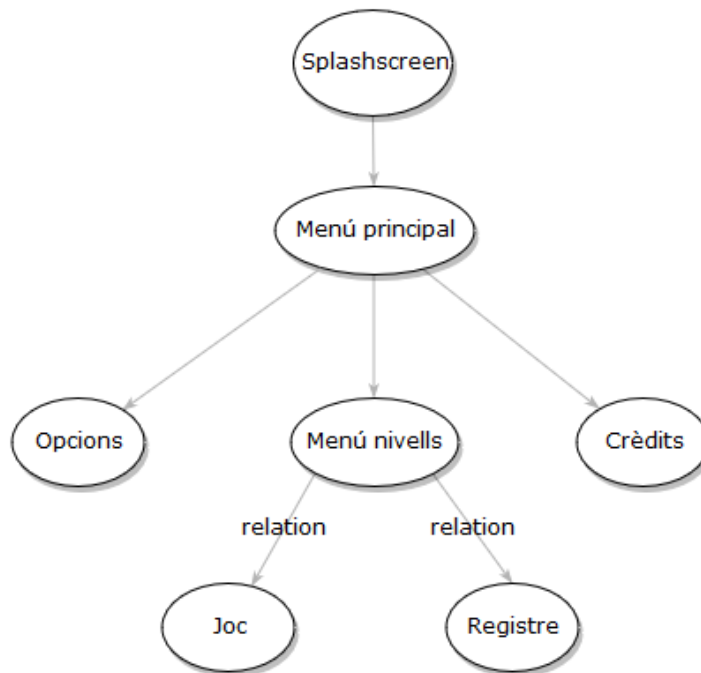


Figura 4.3: Relació entre escenes de l'aplicació GeniusUp

podem anar a les opcions, als crèdits, o seleccionar una de les quatre branques científiques i anar a la selecció de nivell. Finalment d'allà podem anar a un registre on es guarden totes les combinacions a fer, o a un nivell en concret en el que vulguem jugar.

#### 4.4.2 Splashscreen

Aquesta pantalla es pot interpretar des de dues perspectives. Des de la perspectiva gràfica i visual, es mostra a l'usuari el logotip de la UIB, per tal d'indicar qui ha creat l'aplicació (figura 4.4). Des del punt de vista lògic però, es carreguen les opcions bàsiques de tot el joc. És carreguen els sons—tant la *música* que sona contínuament com els *efectes sonors* per indicar que una combinació s'ha realitzat o que un nou rècord s'ha aconseguit, per exemple—, els colors per a cada un dels modes *colorblind*, i finalment el gestor d'idiomes. D'aquesta manera la resta d'escenes ja no perdran temps carregant cap d'aquests elements, per el que l'experiència de l'usuari serà molt més fluïda—no hi haurà cap retràs degut a que l'aplicació estigui carregant alguna textura o audio.

#### 4.4.3 Menú principal

El menú principal és la pàgina principal de l'aplicació. Com hem vist al flux del joc des del menú principal es pot navegar a tres escenes distintes: les opcions, els crèdits i el menú de nivells. Addicionalment hi ha un botó per sortir del joc, tot i que aquest al



Figura 4.4: Escena Splashscreen

ser per a mòbils també es pot tancar d'altres maneres—pitjant el botó de Home en els mòbils *iOS*, o el botó d'Enrere en mòbils *Android* (figura 5.4).

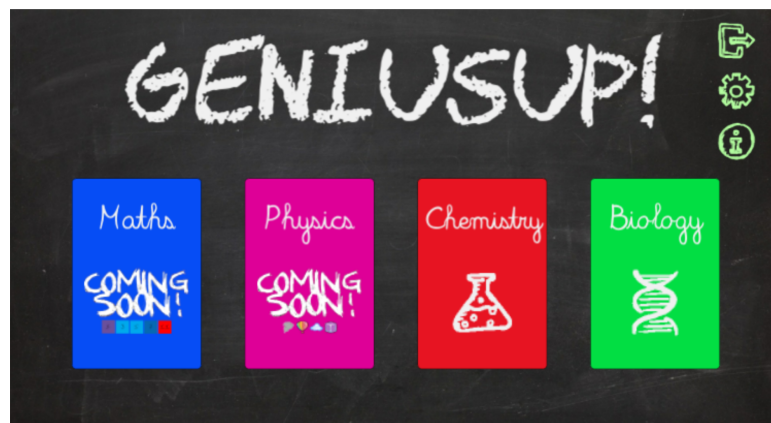


Figura 4.5: Escena Menú principal

Els quatre botons principals indiquen les categories possibles del joc. Només dos d'ells són funcionals, química i biologia, els de matemàtiques i física encara no s'han definit. Al pitjar en una de les categories que encara no estan definides un missatge informa a l'usuari que encara no s'hi pot jugar. Si es pitja al botó de química o de biologia es porta al jugador a la selecció de nivell.

Els tres botons de la part superior dreta són el de sortir de l'aplicació, el de les opcions i els dels crèdits, en aquest ordre. Es va optar per icones per eficiència d'espai. De moment la verificació amb usuaris reals no ha portat cap queixa, però en cas de confusió per part dels jugadors respecte la funció de cada botó es podria canviar la icona o afegir-hi una etiqueta.

Tota la part lògica d'aquesta escena es gestiona amb un sol *script*. Aquest té com a variables els quatre botons i els dos missatges que informen al jugador que la categoria

encara no està disponible per jugar.

En el procediment *Start()*, el qual es crida quan es crea l'objecte—és a dir, quan es crea l'escena del menú principal—, simplement es pinta cada botó del color corresponent. Aquests colors recordem que ja han estat carregats a l'escena *Splashscreen*. Així si tornem de l'escena de les opcions havent canviat el mode *colorblind*, els colors dels botons es repintaran amb el nou color.

D'altra banda, el *script* respon als esdeveniments de quan es pitja un botó carregant l'escena corresponent—sigui el menú de nivells d'una branca en concret, les opcions, o els crèdits—. Si la categoria pitjada encara no està disponible, posa visible el missatge d'informació. Si l'usuari pitja el botó de sortir o, alternativament, pitja el botó *Enrere*, *Home* o la tecla *Escape*, l'aplicació es tanca.

#### 4.4.4 Opcions

En l'escena de les opcions l'usuari pot canviar quatre aspectes. Pot apagar o encendre la *música* de l'aplicació—la qual sona contínuament en bucle—, pot activar o desactivar els *efectes de so*—els quals sonen al realitzar una nova combinació, al quedar poc temps o al seleccionar certs elements—, pot canviar el *mode de visualització de colors*—pot triar entre quatre modes: *normal*, *protanopia*, *deuteranopia* o *tritanopia*—i pot triar quin dels tres *idiomes* vol (figura 4.6). Els modes de visualització de colors seran comentats amb més detall al punt d'accessibilitat 4.5.

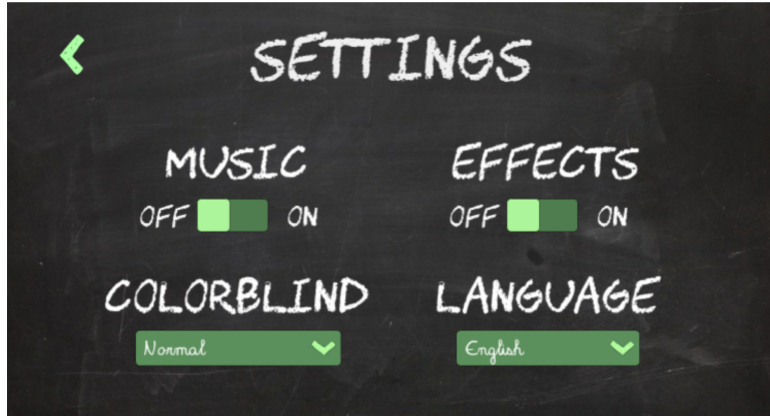


Figura 4.6: Escena Opcions

Des de la perspectiva lògica aquesta escena també té un sol *script* que gestiona tots els esdeveniments i el comportament dels elements. Igual que amb l'escena del menú principal, aquest *script* conté de variables els quatre elements: dos botons tipus commutador—sols tenen dos valors, activat o no—per als botons de música i efectes sonors, i dues llistes desplegable per a la selecció de mode de color i idioma.

A l'inici, quan el *GameObject* es crea, la funció *Start()* els inicialitza als seus valors per defecte. Tant els dos botons commutadors com les llistes s'inicialitzen al darrer valor que van guardar. És a dir, si es tanca l'aplicació amb l'idioma espanyol, al tornar-la a obrir l'espanyol serà el que estarà seleccionat.

Aquest script no necessita funció *Update()*, ja que no ha de fer cap canvi de manera contínua, tot funciona per esdeveniments. Si es pitja un dels quatre botons simplement s'actualitza el seu valor i es guarda aquest valor a l'*script* de les opcions del joc, visible per tota la resta d'escenes—aquest *script* és el que fa servir per exemple l'escena *Splashscreen* per a carregar les opcions al principi del joc.

##### 4.4.5 Crèdits

L'escena més simple de tot el videojoc. Consisteix gairebé només d'elements gràfics. L'única component lògica és la que gestiona el bucle amb el que es mostren els crèdits. A la dreta hi ha els logotips de les entitats i departaments involucrats en la producció del *GeniusUp*, i al mig hi ha un scroll vertical en el qual els crèdits es van desplaçant verticalment (figura 4.7).



Figura 4.7: Escena Crèdits

La component lògica és un simple *script*, el qual té com a única variable el *scroll*. Aquest és una classe pròpia de *Unity*, amb els seus mètodes i funcions. Una d'elles retorna la posició del *scroll*—és a dir, al principi, a dalt de tot, aquesta funció retorna el valor 1, i al final, a baix de tot, retorna un 0—. El que fa aquest *script* és anar disminuint el valor del *scroll*, per així simular l'efecte de què el text està movent-se verticalment. Al arribar a sota, retorna al principi. Per a evitar que el text fes un bot que l'usuari pogués veure al passar de final a principi, es van introduir una sèrie de línies en blanc, per tal de que la transició fos neta i fluïda.

##### 4.4.6 Menú de nivells

El menú de nivells és on l'usuari selecciona a quin nivell vol jugar, així com també veure quins nivells ja ha superat i amb quina puntuació. Primer de tot cal comentar que aquesta escena és la primera realment modular. Les quatre escenes anteriors serien idèntiques si només s'hagués implementat el joc amb una sola branca científica.

El que fa que aquesta escena en particular sigui modular és que està adaptada per, al rebre del menú principal la categoria seleccionada, només mostrar la informació dels nivells d'aquella categoria particular. Al haver d'estar tot en una sola escena, això

implica crear quatre *GameObjects* iguals, però cada un amb la seva informació. Cada un d'aquests *GameObjects*—els quals anomenarem panells—conté quatre elements més. En concret, dos texts, un per a indicar a l'usuari la categoria i un per accedir al registre de combinacions possibles—el qual s'explicarà a continuació—, un altre objecte que contendrà la llista de combinacions d'aquella categoria—en el cas de química seran composts químics, en el cas de biologia grups taxonòmics—, i finalment una llista dels nivells, amb tota la seva informació—tant paràmetres com el temps per a completar, la puntuació necessària a assolir o les combinacions a realitzar, com elements gràfics com imatges, textures. (figura 4.8)

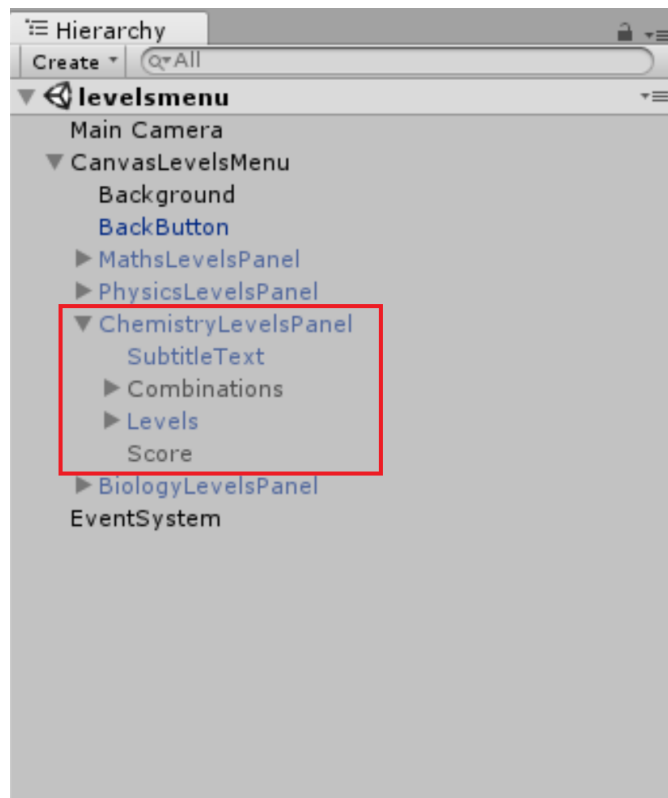


Figura 4.8: Jerarquia dins Unity dels nivells

Per acabar l'apartat gràfic, comentar que visualment apareixeran els deu nivells que s'han implementat fins al moment, amb els que estiguin ja superats pintats del color de la categoria i amb des de zero a tres estrelles—segons la puntuació que l'usuari ha obtingut—, i a dalt a la dreta l'etiqueta de text que servirà com a botó per anar al registre. Per tal d'il·lustrar les diferències entre dues categories es mostren a les figures 4.9 i 4.10 dues imatges fetes al mateix usuari, el qual ja s'havia passat el joc de biologia però no havia començat a jugar al de química.

Aquesta escena també és la primera en la que la part gràfica és minúscula comparada amb la part lògica que la fa funcionar. En les escenes anteriors, els *scripts* simplement carregaven alguna informació o responien a certs esdeveniments. Aquestes dues parts encara hi són, al principi de l'aplicació—recordem, al *script Start()*—els quatre panells



Figura 4.9: Nivells de la categoria de biologia

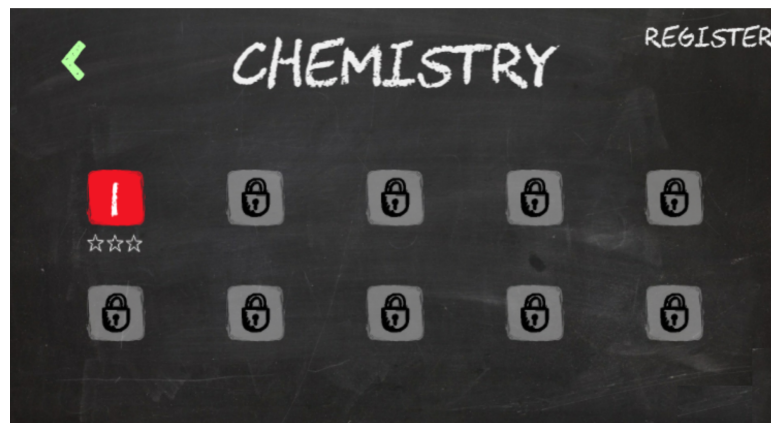


Figura 4.10: Nivells de la categoria de química

están inactius, és a dir, no visibles. El *script* *Start()* s'encarrega de fer visible el panell de la categoria en què es troba l'usuari—informació que haurà guardat el menú principal al seleccionar categoria.

Cada panell té un *script* associat, el qual quan detecta que el panell s'ha fet visible carrega els nivells. Cada nivell també té una part d'interfície d'usuari—el botó quadrat, amb el nombre de nivell i les estrelles—i una part lògica—el nombre de nivell, les files i columnes, el temps, les puntuacions a aconseguir, etc.—. El *script* del panell inicialitza ambdues parts, carrega la informació i, segons el seu valor, pinta el nivell desbloquejat amb un cert nombre d'estrelles o el pinta bloquejat amb un pany.

La llista de combinacions conté una llista de *GameObjects* amb un *script* associat. De nou, com que hi ha combinacions de quatre tipus, un per a cada categoria, lògicament això s'ha representat mitjançant una classe base *Combinació* i llavors quatre classes específiques de cada branca filles de *Combinació*. D'aquesta manera totes les classes són iguals, però cada una té la seva pròpia llista d'elements que poden formar una combinació—la classe de *CombinacióQuímica* conté elements químics, amb els

quals pot formar composts químics, la classe *CombinacióBiologia* conté animals, amb els quals pot formar grups taxonòmics.

A les figures 4.11 i 4.12 es mostren dos exemples de combinacions, una de cada branca implementada. Cada una indica la seva *longitud*, els *elements* que la formen, el *nom* de la combinació, el *pes* que té dins el nivell—a major pes més cops es podrà fer—i finalment un *valor booleà* que indica si la combinació té un ordre únic o no.

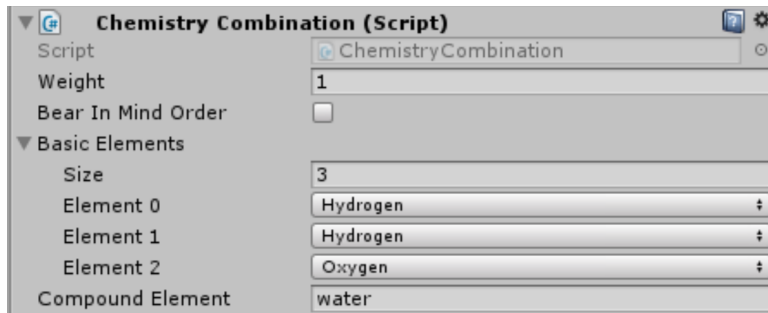


Figura 4.11: Estructura de dades d'una combinació de química

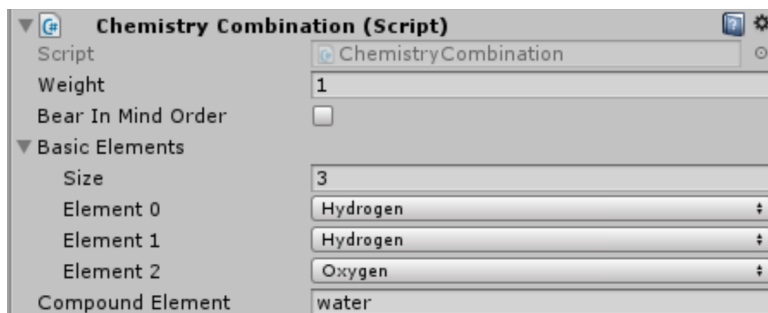


Figura 4.12: Estructura de dades d'una combinació de biologia

Finalment a la part gràfica hi ha la llista de nivells del panell. Com hem comentat cada nivell té una part gràfica i una part lògica. La part gràfica conté la informació necessària per a representar correctament el nivell dins el món virtual. És a dir, el nombre del nivell, el botó associat a ell, el rècord, la seva imatge d'estrelles, etc. (figura 4.13)

En canvi la part lògica del nivell al complet—és a dir, cada element gràfic té una part lògica que forma la seva estructura dins el codi, però aquí ens referim a la part lògica que afecta a tota l'escena en general—conté la informació necessària per a definir el seu comportament. Bàsicament conté el *nombre del nivell*, les seves *columnes* i *files*, el *temps* que es té per a completar-lo, la *puntuació mínima* a obtenir, el *nivell anterior* i *posterior*, les *combinacions* que s'han de realitzar, i finalment les seves *excepcions*, aquells elements que no han d'aparèixer per evitar conflictes—si una combinació és mamífers i està formada per un cérvol, un ase i un hipopòtam, no pot aparèixer al nivell

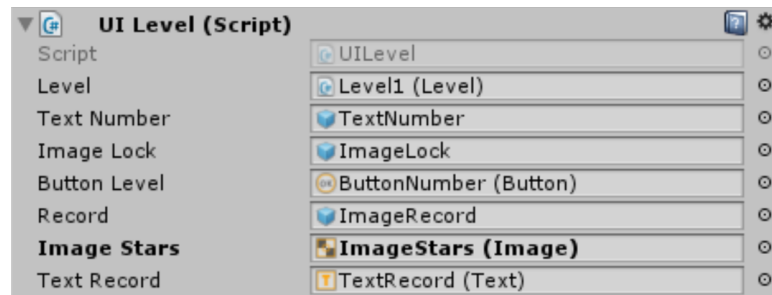


Figura 4.13: Estructura de dades de la part gràfica d'un nivell

un lleó, ja que també seria un mamífer però no contaria per a la combinació. (figura 4.14)

#### 4.4.7 Registre

L'escena registre és aquella que mostra totes les combinacions possibles d'una categoria determinada. En un futur es vol que també indiqui quines combinacions s'han realitzat i quines no, o algun tipus d'informació que informi a l'usuari de quines combinacions fa més i quines menys, per tal de que pugui saber què li queda per aprendre o reforçar, però de moment, així com està estructurat el codi, s'ha deixat com a feina futura—el canvi implicaria una reestructuració important—. Aquesta escena també és *modular*, ja que una sola escena ha de contenir els registres de totes les categories.

Gràficament l'escena és de les més simples del joc. Conté només dos elements, un *títol* i una *llista amb scroll* que conté les combinacions. La llista s'ha fet en *scroll* ja que el nombre de combinacions és molt alt per a mostrar-les totes en la pantalla de cop. Aquesta però no és automàtica com la de l'escena dels crèdits, sinó que la mou l'usuari amb el dit—o el ratolí en el cas de l'aplicació per ordinador. (figura 4.15)

La part lògica consisteix bàsicament de tres elements: un *script* que gestiona el *scroll*, i llavors dos tipus de *l·listes*. La primera conté totes les combinacions d'una categoria en particular en forma de *GameObjects*—idèntics a la llista del menú de nivells—. L'altra és una llista d'aquests *GameObjects*. És a dir, conceptualment les dues llistes representen la mateixa idea: les combinacions d'una categoria en particular. Però en la primera cada combinació individual està formada mitjançant un *GameObject*. En canvi en la segona la llista és un *GameObject* en si que conté totes les combinacions—sent cada combinació un dels *GameObjects* anteriors—. Finalment el *scroll* agafa com a contingut aquest segon *GameObject* amb les combinacions de la categoria en la que es trobi l'usuari.

#### 4.4.8 Joc

La darrera escena és la més complexa de tot el videojoc, tan gràfica com lògicament. És la que conté major nombre d'elements gràfics, tan visuals com sons, i és la que té l'estructura de scripts més gran i complexa.

Començarem analitzant l'aspecte gràfic. Visualment la pantalla es separa en tres *zones*. La primera conté la informació del nivell en particular. A dalt a l'esquerra apa-



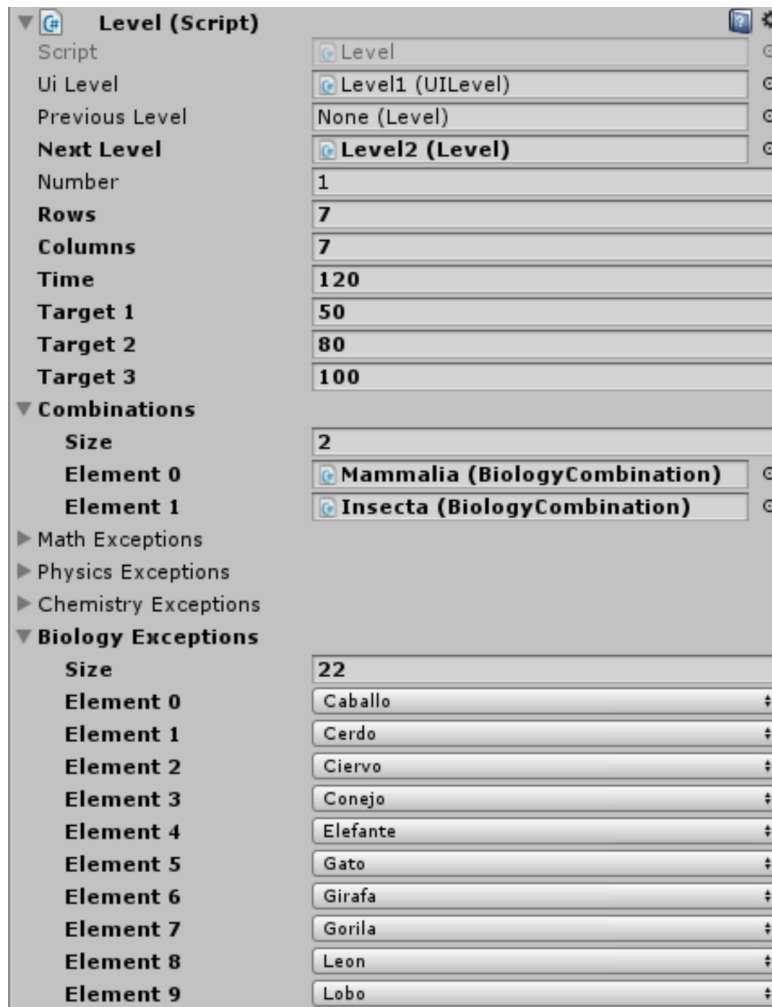


Figura 4.14: Estructura de dades de la part lògica d'un nivell

reix el nombre del nivell en què es troba l'usuari, seguit de quatre icones. Les icones representen, d'esquerra a dreta i de dalt a baix, el *temps restant*, la *puntuació* obtinguda respecte la que s'ha d'aconseguir, el *nombre de moviments*, i el *rècord del nivell*. Així l'usuari pot saber en tot moment quant de temps li queda per acabar, quants de punts més ha de fer, i si està a prop o no del rècord. Els moviments de moment només serveixen per a desempatar dues puntuacions idèntiques, però en un futur podrien entrar com a una nova restricció més—s'ha d'aconseguir la puntuació amb un màxim de moviments per a passar el nivell.

La segona zona conté el tauler en si. La seva dimensió ve determinada per la informació lògica del nivell. Gràficament el tauler conté els elements del nivell en particular. Les imatges que es mostren estan guardades en el projecte, i com veurem d'aquí un moment la part lògica de l'escena carrega la que correspongui segons la categoria triada per l'usuari. Els elements gràfics evidentment depenen de la categoria en la que es juga, com il·lustren les figures 4.16 i 4.17.



Figura 4.15: Escena Registre

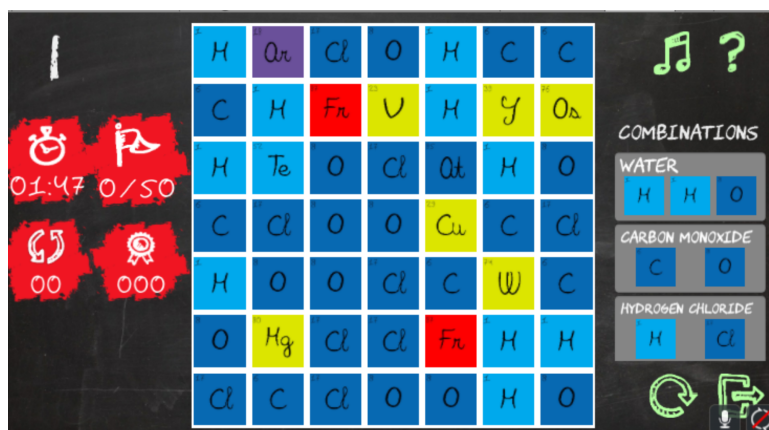


Figura 4.16: Escena Joc amb elements químics

La tercera i darrera zona es pot al seu torn dividir en dues parts. La primera seria la llista amb les combinacions a realitzar, la qual informa constantment a l'usuari de quines combinacions ha d'intentar fer. Aquesta llista inicialment no hi era, es va implementar més endavant perquè sinó era difícil per a l'usuari recordar quines combinacions exactament eren les que demanava el nivell.

L'altra part són els quatre botons. De nou, d'esquerra a dreta i de dalt a baix, el primer serveix per a activar i desactivar la música. Al clicar-lo la imatge canvia per a indicar l'estat del botó. El segon botó, l'interrogant, obre un missatge que explica tant per text com amb una petita animació com es juga al joc. (figura 4.18)

El tercer botó permet reiniciar el nivell. Això implica tornar a reiniciar les variables de temps, puntuació i moviments, així com recarregar el tauler. Aquest botó s'ha posat per a que si un jugador ja veu que la partida no ha sortit com ell esperava, no hagi d'esperar a que acabi el temps per a tornar a intentar el nivell. Finalment el quart botó permet sortir del nivell i tornar al menú de selecció de nivell.

La part lògica es pot també dividir en dues parts, la que controla tot el joc i el seu



Figura 4.17: Escena Joc amb elements biològics

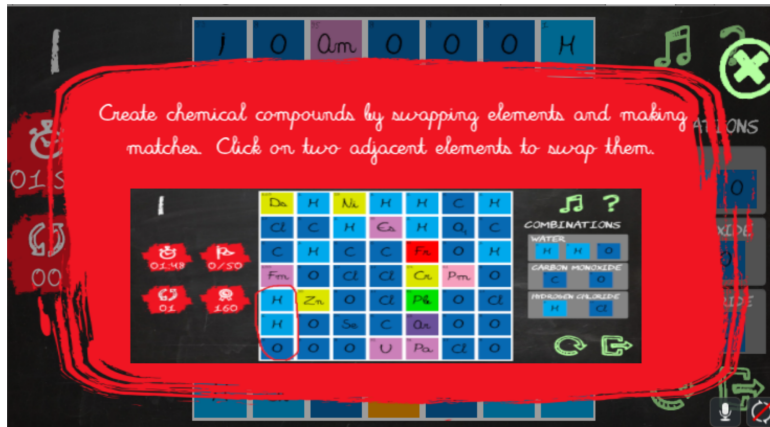


Figura 4.18: Missatge d'ajuda per a l'usuari

comportament, centrada al voltant del script *GameLogic*, i la que respon als nombrosos esdeveniments que poden ocórrer.

Primer explicarem la lògica global. Evidentment no es pot explicar tota amb detall ja que s'hauria de comentar el codi. Però es pot explicar el bucle de control principal. Primer de tot, el script *GameLogic* té una funció *Start()*, la qual inicialitza les variables i estructures de dades internes. Es col·loca el temps restant, els moviments i els punts a zero, i es carrega el rècord de nivell en particular. A més, s'omple la llista de combinacions i el tauler.

De tots aquests processos l'únic que val la pena comentar és el de omplir el tauler. Primer de tot es genera el tauler lògic, és a dir, s'omplen les estructures de dades que donen suport i formen el tauler. Llavors es traslladen els valors d'aquesta estructura de dades al tauler gràfic que veu l'usuari.

Les estructures de dades es carreguen seguint el següent procediment:

- Es crea un tauler temporal, on a cada casella se li assigna una casella. La casella

que s'assigna es decideix de la següent manera:

- S'agafa un nombre aleatori entre 0 i 1.
  - Si aquest és menor que la probabilitat d'elements útils del nivell—la qual indica, també entre 0 i 1, la proporció d'elements del tauler que serveixen per a fer una combinació—llavors s'agafa un element aleatori d'una de les combinacions del nivell.
  - Si no és menor, s'agafa un element aleatori que no sigui útil ni que tampoc formi part de les excepcions.
- D'aquesta manera el tauler temporal queda ple d'elements, i aquests estan distribuïts en útils i no útils en la proporció indicada per la probabilitat d'elements útils del nivell—la qual és un paràmetre que pot modificar l'usuari per a fer més fàcil o més difícil el nivell.
  - Un cop es té el tauler temporal omplert es comprova si s'ha creat amb algunes combinacions ja fetes. Mitjançant un bucle amb condició es van eliminant totes les combinacions inicials que haguessin sorgit.
  - Finalment s'assegura que el tauler tingui com a mínim una combinació.

Al final d'aquest procediment tenim les estructures de dades que formen el tauler plenes d'elements, i sabem que no hi ha cap combinació inicial ja feta—elements d'una combinació ja en fila o columna un vora l'altre—i que a més l'usuari tindrà al menys un moviment possible.

L'altre part involucrada en la creació del tauler és passar d'aquesta estructura lògica a una gràfica que pugui veure l'usuari. Això es fa simplement iterant en bucle per totes les cel·les del tauler i agafant la imatge corresponent a l'element de cada una de les cel·les.

Un cop es tenen inicialitzades totes les estructures i variables comença el bucle de joc. El bucle es basa en la funció *Update()*, la qual es crida a cada frame. Això implica entre unes 30 i 60 vegades per segon. El que fa aquesta funció és que, mentre l'estat del joc no sigui finalitzat, va disminuint el temps restant i actualitzant el text de la icona corresponent. Un cop el temps restant arriba a 0, assigna a l'estat del joc l'estat finalitzat.

Un cop l'estat és el de joc finalitzat, el programa mira:

- Si l'usuari ha superat el nivell i aquest era el darrer de tots, es calcula la puntuació total sumant les puntuacions de tots els nivells i es mostra un missatge de victòria amb la puntuació total aconseguida. (figura 4.19)
- Sinó es mira si el jugador ha fet suficient punts per a passar-se el nivell. En cas afirmatiu es mostra un missatge d'enhorabona (figura 4.20), en cas contrari s'anima a què l'usuari ho torni a provar. (figura 4.21)
- Un cop s'ha tancat el missatge que hagi sortit a l'usuari, el joc torna a la pàgina de selecció de nivell.

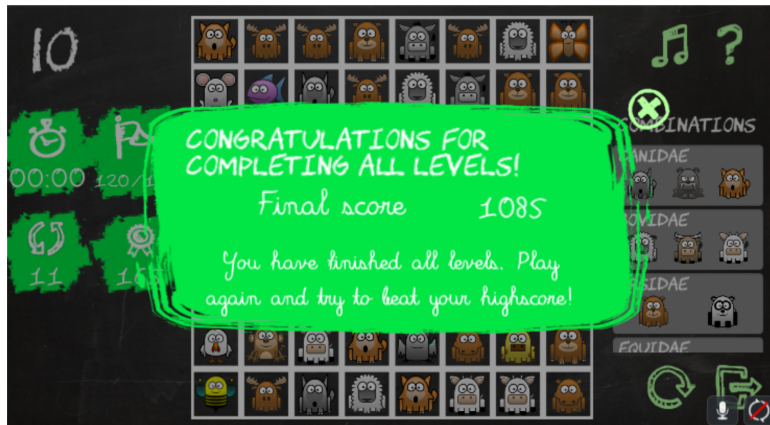


Figura 4.19: Missatge de joc finalitzat



Figura 4.20: Missatge de nivell completat amb èxit

Aquesta és la lògica global del joc. Ens queda per explicar com es tracten els esdeveniments. Els esdeveniments al pitjar un botó s'obviaran perquè funcionen de manera molt similar a l'explicada en anteriors escenes, al rebre un clic realitzen una acció en concret. El que cal comentar és com es poden realitzar combinacions.

Una combinació es fa al intercanviar la posició d'una cel·la amb la del seu costat, tan vertical com horitzontalment. Inicialment només s'havia pensat un moviment d'arrossegar, però com que aquest no acabava de ser del tot intuïtiu, vaig decidir afegir-ne un altre. Ara, a part d'arrossegar una cel·la a una contigua per tal d'intercanviar els seus valors, també es podia clicar dues cel·les contigües per a realitzar l'intercanvi.

El moviment d'arrossegar funciona de la següent manera:

- Es detecta que s'ha fet un clic prolongat, sense aixecar el punter—sigui aquest el dit o el ratolí.
- És la cel·la afectada la que aixeca l'esdeveniment, per tant al rebre aquesta cridada ja sabem quina cel·la estem movent.



Figura 4.21: Missatge de nivell fallit

- Anem calculant en tot moment la seva nova posició al anar-la arrossegant. Un cop s'ha mogut una certa distància es limita el seu moviment a un sol eix. Si s'havia arrossegat més distància horitzontalment que verticalment es determina que l'usuari volia fer un moviment horitzontal, i viceversa per al moviment vertical. La cel·la llavors ja només es mourà en dita direcció.
- Un cop s'amolla el punter, es calcula a quina cel·la s'ha amollat. Si aquesta és contigua a la inicial s'intercanvien les imatges dels elements i es calcula si això ha donat una combinació. En cas contrari simplement es retorna la cel·la a la seva posició inicial.

El moviment de clic funciona de manera similiar, tot i que és més simple ja que no s'ha de tenir en compte l'animació del moviment de cel·la. La idea bàsica és la següent:

- Al rebre un clic es mira si és el primer o no, és a dir, si ja hi havia una cel·la clicada.
- Si no hi havia cap cel·la clicada abans es guarda la cel·la actual com a primer clic. Si ja n'hi havia una, es calcula si aquesta nova cel·la és contigua a la del primer clic.
- Si són contigües s'intercanvien les cel·les i es calcula si l'intercanvi resulta en una combinació. Si no són contigües no es calcula res. En ambdós casos es reinicien les variables per a nous clics.

Finalment el càlcul per a saber si un intercanvi ha resultat en una combinació és el següent:

- Es canvien els valors a l'estructura de dades de les cel·les intercanviades.
- Es mira si alguna de les dues forma combinació amb les seves noves cel·les contigües.

- Si una de les dues—o ambdues en segons quins casos—formen una combinació amb alguna de les seves cel·les contigües, es realitza una petita animació per a informar a l'usuari de què la combinació era vàlida, es sumen els punts, i es donen nous valors a les cel·les que han quedat buides després de la combinació. Finalment, no es retorna al flux de programa principal fins que el tauler resultant d'afegir les noves cel·les té combinacions disponibles, ja que sinó l'usuari podria quedar sense moviments vàlids. Si no queden combinacions vàlides i s'ha de refer el tauler, s'informa a l'usuari que es reiniciarà el tauler mitjançant un missatge.
- Si cap de les dues ha resultat en combinació, es retornen tant els valors de les cel·les en l'estructura de dades com les seves imatges als originals.

## 4.5 Accessibilitat

### 4.5.1 Definició

El Game Accessibility Special Interest Group (GASIG) de la International Game Developers Association (IGDA) defineix l'accessibilitat en videojocs com a l'habilitat de jugar un joc fins i tot sota condicions restrictives, ja siguin limitacions funcionals o discapacitats, per exemple, sensorials o motores. Es per tant una definició molt àmplia que inclou no només l'accessibilitat per a persones amb diversitat funcional, sinó també per als jugadors més joves i d'edat avançada, així com jugadors ocasionals i jugadors nous i poc experimentats. Si un joc és poc accessible, els jugadors amb diversitat funcional experimentaran dificultats o no podran jugar, per el que sentiran frustració al ser excloses d'una de les formes d'entreteniment més populars d'avui en dia. [10]

### 4.5.2 Barreres

Les principals barreres d'accessibilitat en els videojocs són: [10]

- El jugador no pot rebre estímuls, ja siguin visuals, auditius o tàctils.
- El jugador no pot determinar quina és la resposta adequada per a realitzar una acció concreta necessària per a avançar en el joc.
- El jugador no pot proporcionar input al joc al no poder manipular el dispositiu d'interacció entre el jugador i el videojoc, sigui aquest un teclat, un ratolí, etc.

### 4.5.3 Idioma

El primer aspecte d'accessibilitat important de tots és la barrera de l'idioma. Amb l'explosió de les noves tecnologies en la segona meitat del segle XX i la seva instauració en el dia a dia de la societat actual, la manera com ens comunicam i la manera d'entendre l'oci han canviat radicalment. En aquest nou escenari, els videojocs han cobrat una gran rellevància, superant altres sectors com el cine o la música. Amb aquest increment d'ús dels videojocs, ha augmentat el que s'anomena *casual gaming*, és a dir, persones que juguen de manera poc habitual i relaxada. Això ha fet que el perfil d'usuari dels videojocs s'hagi difuminat enormement, i per tant els videojocs actuals van destinats a un espectre molt més gran i representatiu de la societat. [10]

Al augmentar el nombre de persones a les que van destinats els videojocs, els esforços en accessibilitat han de ser majors també, ja que s'han d'adaptar a la multiculturalitat i multilingüisme dels jugadors. Es per això que s'ha de tenir en compte la barrera de l'idioma en tot videojoc, així com l'ús de referències culturals ofensives o difícils de comprendre. En el nostre cas s'han seguit les línies d'actuació de traduir i oferir el joc en tres idiomes diferents, les dues llengües oficials del territori on s'ha creat—català i castellà—, i la llengua forta o prevalent del món actualment—anglès.

### 4.5.4 Color

L'altre aspecte tractat ha estat el color. El color és un aspecte molt important de qualsevol objecte, cada color transmet unes sensacions i cada persona rep aquestes sensacions d'una manera subjectiva. L'ús per tant d'un pigment o un altre afectarà al missatge que es vulgui transmetre. Degut a l'existència de certs tipus de discapacitats visuals les quals afecten la manera en la que una persona visualitza un color en concret, si l'aplicació no està preparada per a aquestes persones llavors els colors triats no seran els que veuran els jugadors, i per tant el seu ús no serà el desitjat.

En el cas del *GeniusUp* el color servia com a distinció entre les distintes categories. L'usuari associava cada color amb una categoria i així no s'havia de recordar constantment en quina categoria es trobava. Per exemple, els nivells estaven pintats amb el color de la categoria en què jugava l'usuari en un moment donat—il·lustrat amb les figures 4.9 i 4.10—. Els colors triats van ser verd per a biologia, blau per a matemàtiques, vermell per a química i rosa per a física 5.4. S'ha triat rosa perquè per a algú sense discapacitat visual el color és suficientment diferent, i per a persones amb discapacitat visual, si no s'ha canviat el mode de color, és més distint que si s'hagüés triat un altre color més típic, com per exemple el groc.

Els colors han de ser suficientment diferents entre ells, i alhora han de tenir un contrast suficient amb el text i el fons, per tal de què tots els elements siguin visibles. Per tal d'assegurar-ho s'ha utilitzat l'eina Vischeck [11], la qual ens permet veure com es veurien les nostres escenes si tinguéssim protanopia, deuteranopia—dèficit en la visió de verds i vermells, relativament comuns—o tritanopia—dèficit en la visió de blaus i grocs, molt rar—. Els resultats es poden visualitzar en les figures 4.22, 4.23 i 4.24.



Figura 4.22: Comparació del mode protanopia





Figura 4.23: Comparació del mode deuteranopia



Figura 4.24: Comparació del mode tritanopia

## 4.6 Verificació

Al tractar-se d'un *serious game* educatiu, els conceptes que es volen transmetre han de ser verificats per experts d'alguna manera, per tal d'assegurar que l'educació que ofereix el *GeniusUp* és vàlida, útil, i sobretot, correcta. La fase de verificació només s'ha pogut implementar al joc de química. La versió amb la branca de biologia també és funcional, però encara no s'ha validat.

La primera part de la verificació la van fer els responsables del projecte, abans de què jo hi comencés a fer feina, al parlar amb químics de la UIB i comentar amb ells la idea del joc. Un cop és va veure que era una idea factible, amb l'aprovació d'experts en el tema, es van demanar a químics de la UIB una llista de composts adequats per al joc, ordenats per nivell. D'aquesta manera és va validar que els composts químics apresos eren vàlids i útils per als estudiants, i a més que la progressió de la dificultat dels nivells augmentava d'acord amb l'augment de dificultat dels composts.

Finalment, un cop l'aplicació funcional amb la versió de química va estar acabada, es va realitzar un qüestionari per tal de veure les opinions d'alumnes de la facultat de química a l'hora de jugar al joc. Aquesta validació tenia una part més específica sobre els conceptes químics del joc i una part més general sobre el disseny i el funcionament del joc que es podrà reutilitzar més endavant per a futurs usuaris no experts.

## 4.7 Estat final i futur del projecte

Al final de la meua contribució ja s'han implementat dues de les quatre branques, totalment funcionals, una d'elles verificada per experts i usuaris genèrics. La versió

#### 4. GENIUSUP

amb la branca de química s'ha penjat tant a l'*AppStore* de *iOS* com a la *Google Play Store* d'*Android*, amb entre 500 i 1000 descàrregues combinades. Les puntuacions rebudes han estat molt bones—com mostra la figura 4.25—, així com el nombre de descàrregues.

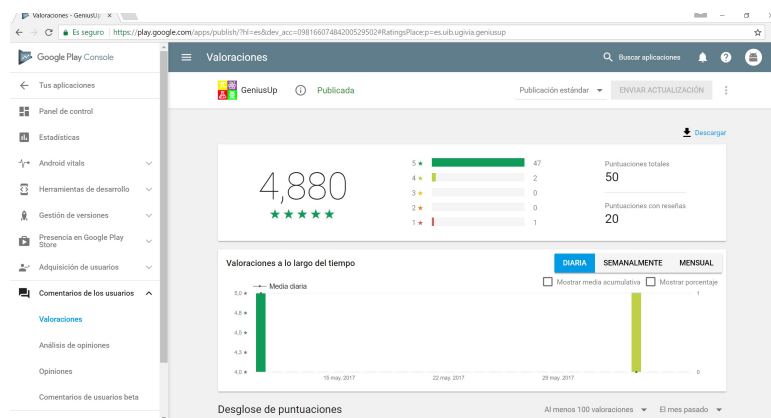


Figura 4.25: Valoracions Play Store

De cara a un futur immediat caldria acabar de polir alguns aspectes gràfics de la versió de biologia i pujar aquesta nova versió a les botigues de mòbils. Així es tindria un nou reclam per a atreure encara més usuaris. Ja de cara a un futur no tan immediat, s'haurien de definir els jocs de matemàtiques i física, implementar-los i verificar-los, per tal de tenir l'aplicació completa.

Respecte al funcionament en si de l'aplicació, una possible llista de millores que farien l'aplicació més fluida i interessant podria ser:

- Millorar el control d'arrossegat, fer-lo més natural.
- Introduir noves animacions per a avisar a l'usuari de què una combinació s'ha realitzada, per així millorar el feedback visual—punts aconseguits per exemple, o un simple missatge d'ànims.
- Afegir nous nivells, amb noves combinacions a fer.
- Introduir nous tipus de nivell, que no tots siguin limitats per temps i puntuació—per exemple limitar el nombre de moviments, enlloc de puntuació mínima forçar a realitzar una combinació determinada un mínim de cops, introduir nous elements especials amb algun efecte, etc.
- Donar més informació a l'escena del registre sobre les combinacions fetes—distingir combinacions fetes de no fetes i aportar algun tipus d'estadístiques, com per exemple el nombre total de vegades que s'ha fet cada combinació.
- Crear un sistema de ranking, tant de cada nivell com total. Primer de tot podria ser local, de cada dispositiu, però més endavant es podria augmentar a un ranking basat en servidor, on tots els usuaris poguessin veure les puntuacions d'altres.

- Crear un sistema d'usuaris, sigui propi com a través de tercers com *Facebook*, *Gmail* o *Google Play*. Afegir recompenses especials en un cert nivell, així com funcionalitats extres als usuaris—compartir puntuacions, nivells passats, recompenses aconseguides, etc.

La gràcia del projecte *GeniusUp* és que un cop s'ha vist la idea no té gairebé límits de fins on es pot arribar. Com més millores s'afegeixin, més motivant i interessant serà el joc. Això implicarà que els joves hi jugaran més i, per tant, aprendran més i veuran la ciència d'una manera diferent a la que estan acostumats, que al cap i a la fi és l'objectiu principal del projecte.



## LABERINT

### 5.1 Estat del projecte al moment de la meva incorporació

L'aplicació del laberint, a diferència de la *GeniusUp*, encara no estava creada. Hi havia una base establerta amb la qual, a través de *Unity* i alguns dels assets que ofereix la seva botiga, s'havia creat un laberint. Quan em vaig afegir al projecte se'm va passar un projecte de *Unity* en el que s'havien fet les proves, i un laberint creat amb una bolla que es movia d'acord a l'*input* que li venia per *teclat* o *joystick*.

Ara bé, el projecte *Juguemos* volia arribar molt més enfora. La idea era crear un *serious game* mitjançant aquest laberint, el qual fos el més versàtil possible i pogués enfocar-se a vàries perspectives diferents. Així com el joc *GeniusUp* ja tenia un focus clar, el laberint es volia que fos una base a través de la qual llavors es poguessin treballar diferents àrees i realitzar distints estudis.

A partir d'aquesta proposta inicial s'han desenvolupat aplicacions per a dos fins distints. Cal remarcar però que ambdues aplicacions estan basades en el mateix videojoc. La primera volia obtenir el *temps* que tardava el jugador a sortir del laberint, el *nombre de col·lisions* amb les parets que feia i la *longitud de camí* que seguia, per tal de llavors poder estudiar aquestes dades i treure conclusions. L'altre volia realitzar un cert nombre de *fotografies* del jugador durant determinats punts crítics del laberint, per tal d'estudiar les seves expressions.

Tot i que tenien un enfocament distint, les dues aplicacions anaven dirigides a grups específics de persones. El primer model del *serious game* del laberint volia estudiar l'atenció de nens i nenes amb *paràlisi cerebral*, amb *autisme* i amb *dèficit d'atenció* a partir de les dades de temps, col·lisions i longitud. El segon anava enfocat només a nens i nenes amb *autisme*, per tal de poder estudiar les seves expressions segons la situació en què es trobessin dins del laberint.

Aquest projecte doncs pretenia desenvolupar un *serious game* diferent al *GeniusUp*, més orientat a l'estudi i a la recerca. A més a més, tot i que el *GeniusUp* tenia una forta component modular per tal de poder cobrir les quatre branques científiques, en el projecte del laberint aquesta component modular era encara més forta, ja que realment

es volia definir un sol videojoc base, el qual llavors es pogués adaptar a l'estudi o recerca concreta que es volgués fer.

### 5.2 Objectius

Igual que amb el projecte *GeniusUp* dividirem els objectius en generals i específics. En aquest cas els generals provenien del projecte *Juguemos*, i com hem explicat abans defineixen l'abast del projecte al complet. Els específics descriuen i limiten en certa manera el disseny i funcionament de l'aplicació en concret.

L'objectiu general del projecte *Juguemos* és, com s'ha explicat a la introducció<sup>1</sup>, definir un *marc conceptual*, una *metodologia de disseny* i un conjunt d'*eines software* que permetessin el disseny i desenvolupament de jocs multimodals. El projecte del laberint es centra més en la creació d'un videojoc multimodal particular, el qual ha de seguir tant el marc conceptual com la metodologia de disseny definides dins el projecte. Llavors en aquest cas el projecte més que tenir un objectiu general que assolir, es podria dir que té un objectiu general a complir, a respectar, però que ja s'ha definit externament, fora del projecte en si.

Per tant el projecte del laberint es caracteritza més bé per els seus objectius específics. Cal comentar que al tenir dues aplicacions amb propòsits distints, cada una tindrà els seus propis objectius. Primer però comentarem els comuns. El primer és la modularitat del videojoc. Un dels objectius específics principals és assegurar que un sol model de joc serveix múltiples propòsits. És a dir, que amb un sol disseny del videojoc és poden assolir distints objectius, fer distints estudis o recerques. És per això que el format de *serious game* fet amb un motor de videojocs és tan efectiu, ja que com ja s'ha explicat els motors de videojocs permeten i animen a la reutilització d'eines i recursos fets servir en altres jocs<sup>3</sup>.

El segon objectiu específic important que encara afecta a ambdues aplicacions és definir un marc per a la creació de laberints. És a dir, els laberints han de seguir algun tipus de regla a l'hora de ser creats, i com que el joc tindrà més d'un nivell, també s'ha de definir la progressió de dificultat per tal d'assegurar que els laberints són consistents en dificultat a mesura que s'avança en el joc. Inicialment s'ha decidit crear 10 nivells, dividits en 3 categories de dificultat. Els laberints de cada una d'aquestes categoria es caracteritza per tenir una *dimensió*, *nombre de sortides* i *nombre de carrerons sense sortida* determinats. Per tal d'assegurar la proporcionalitat entre laberints de distintes categories s'han decidit les següents dimensions: 5x4, 10x8, 15x12. El nombre de sortides i de carrerons sense sortida es va decidir inicialment segons la taula il·lustrada per la figura 5.1.

El primer nivell de cada categoria estableix el nombre base de sortides i carrerons sense sortida, i llavors cada nou nivell afegeix una nova sortida o un nou carreró sense sortida, alternativament. El problema va sorgir més endavant quan, en una de les dues aplicacions del projecte, al avaluar durant la partida d'un jugador el temps que tarda en sortir, les col·lisions que fa amb les parets, i la longitud del camí seguit, si un nivell tenia dues o més sortides llavors les tres variables triades ja no podien definir el rendiment de tots els jugadors per igual. Amb més d'una sortida apareix una component afegida que afecta a les variables: l'*elecció de sortida*. Com que les sortides no estaven a la mateixa distància de l'inici, dos jugadors que juguessin de la mateixa manera però

	Exits	Dead ends	Formula
Level			$X + [\text{exit}] + [\text{dead end}]$
1.1	1	0	n (basic)
1.2	2	0	$n + 1 + 0$
2.1	1	1	m (basic)
2.2	1	2	$m + 0 + 1$
2.3	2	2	$n + 1 + 1$
3.1	1	1	p (basic)
3.2	1	2	$p + 0 + 1$
3.3	2	2	$p + 1 + 1$
3.4	2	3	$p + 1 + 2$
3.5	3	3	$p + 2 + 2$

Figura 5.1: Taula amb l'estructura dels laberints segons nivell

triessin distintes sortides, tindrien resultats distintos. Per tant, s'ha decidit deixar tots els laberints amb una sola sortida.

També respecte a l'aplicació del laberint que registra dades per a ser analitzades, un altre objectiu és realitzar un *calibratge previ*, per tal de saber fins a quin nivell pot un cert jugador arribar. Com que una de les variables enregistrades és el temps que es tarda a finalitzar un laberint, si un nivell és massa complex per a un jugador aquest tardarà molt més temps que la resta de participants, afectant així a la resta de dades.

A més a més, per tal d'obtenir dades fiables i consistents, cada participant iterarà sobre tots els laberints un cert nombre de vegades—inicialment es va pensar 10, però a l'hora de finalització d'aquest treball s'estava esperant l'opinió dels experts, ja que 10 semblava un nombre massa alt—. Sense el calibratge inicial tots els jugadors haurien de jugar moltes partides, i algunes en nivells que no serien adequats per les seves capacitats, per tant els resultats obtinguts no serien verídics—es veurien afectats per variables com el cansament.

En el cas de l'altre aplicació del laberint—la que pren fotos del jugador a determinats moments—, un dels seus objectius propis és obtenir les expressions del jugador durant la partida. Com que gravar la partida al complet era molt costós, tant en recursos com llavors en el posterior anàlisi, s'han definit uns *punts clau* dins cada laberint.

En aquesta versió del videojoc, a diferència de l'anterior, no hi ha fase de calibratge, sinó que els jugadors iteren tots els nivells un sol cop. A més, hi ha un *límit de temps* a cada nivell. Si no acaben el laberint en menys temps que el límit llavors han de tornar-lo a intentar. Aquest afegit introdueix la *derrota* dins el joc—en l'anterior un cop fet la fase de calibratge el jugador no perdia mai, tardés el temps que tardés—, el que permet fer una foto al moment en què el jugador ha perdut per tal d'intentar capturar *tristesia*, *frustració* o *molèstia*. Altres punts claus són al finalitzar el laberint, per a capturar *alegria*, i carrerons sense sortida, on s'espera capturar *frustració* o *enuig*.

Finalment, tot i no usar el videojoc del laberint, el projecte *Juguemos* inclou encara una altra aplicació més. És el ja explicat *entrenador d'emocions*. Com s'ha justificat en la introducció1, en el moment d'escriptura de la memòria l'entrenador d'emocions

no es pot considerar encara com a un *serious game* en si. Manca molts dels elements típics del joc, tot i que sí que en certa manera anima als participants a entrenar les emocions d'una manera més visual i entretinguda. Ara bé, està desenvolupat a través del motor de videojocs *Unity*, com la resta d'aplicacions d'aquest treball de fi de grau, per el que en un futur es contempla afegir-li noves utilitats o elements més pròpies dels videojocs que no faran més que—mantenint el mateix objectiu d'aconseguir entrenar les emocions en nens i nenes amb *autisme*—apropar encara més l'aplicació a l'àmbit dels *serious games*. Degut a la simplicitat de l'aplicació, la falta d'elements típics dels videojocs, i la fase primerenca en què es troba el projecte, l'entrenador d'emocions no s'analitzarà amb detall, només se'n farà un petit incís més al punt final de línies futures d'aquest capítol.

### 5.3 Motiu per el que és un serious game

Com acabem de comentar l'objectiu principal d'ambdues aplicacions era la *investigació*, la *recerca*. En les dues aplicacions la recerca es fa amb nens i nenes amb *dèficit d'atenció*, *autisme* i *paràlisi cerebral*. Com ja hem comentat en el capítol 2, ambdós videojocs es poden classificar doncs com *serious games* per a la salut.

En el cas de la primera aplicació, el laberint que enregistra dades, la recerca es caracteritza per dos factors. El primer es basa en estudiar quins factors afecten a l'*atenció* dels participants, quins la milloren i quins l'empitjoren. El procés de recerca és el següent:

- Obtenir una base de dades prèvia amb un grup de control. En aquest grup els usuaris són de fora dels grups estudiats. El grup de control realitza només els 10 nivells, no fa fase de calibratge.
- A partir dels seus resultats, extreure els límits de temps per a cada nivell.
- Obtenir la base de dades amb els nens i nenes amb *dèficit d'atenció*, *autisme* i *paràlisi cerebral*—aquí sí que ja es faria servir el procés complet, amb fase de calibratge.
- Un cop es té la base de dades, començar a realitzar noves versions del joc, amb nous elements—sons, música, colors, missatges, etc.—i veure com afecten els elements als nous resultats respecte els obtinguts a la base de dades inicials—completament lliure d'elements.

El segon factor es basava no tant en els resultats, sinó en la interacció entre el jugador i el laberint. El moviment de la bolla s'ha realitzat mitjançant dos mètodes d'interacció (figura 5.2). El primer, el més estàndard, funciona o bé amb el *teclat* o bé amb un *joystick convencional*. El segon és més innovador i fa servir la polsera *Myo*[12]—una polsera amb sensors de mesurament d'inèrcia i electromiografia, com el giroscopi—per a guiar la bolla a través del laberint. Per tant també hi ha una component d'aquest primer *serious game* que pretén rehabilitar i enfortir els moviments dels participants, al forçar-los a usar distints mètodes d'interacció, més físics que els convencionals.

La segona aplicació encara no té els objectius de la recerca tan ben especificats. Al moment de finalització del treball el projecte estava més en un estat de prova. La idea i l'objectiu del joc de capturar les emocions dels participants durant les seves partides



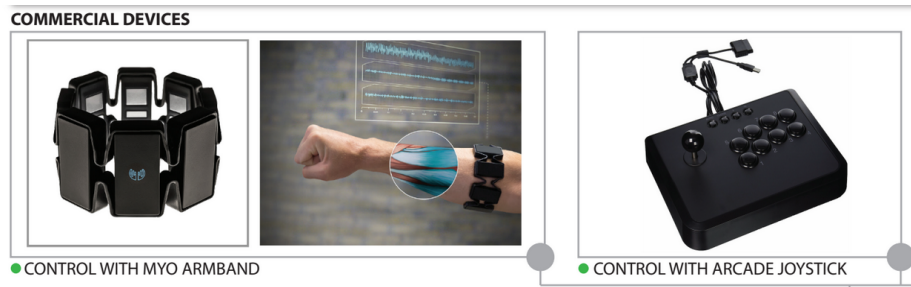


Figura 5.2: Dispositius de interacció

era definitiva, però el mètode amb el que es capturarien aquestes emocions encara no. S'ha optat per fer fotografies per les raons explicades anteriorment, però encara no s'ha decidit si aquest és el millor mètode. Per tant la recerca està en un punt de gairebé investigació, on les fotografies que s'obtinguin determinaran si el mètode usat és fiable o no per a futures iteracions del projecte.

## 5.4 Implementació amb Unity

Aquest projecte és el que millor il·lustra la versatilitat de *Unity* com a motor de videojocs. Primer de tot s'explicarà la base comuna que ambdues aplicacions fan servir, i llavors ja s'entrarà més en els detalls particulars de cada una.

### 5.4.1 Model comú del laberint

Les dues aplicacions tenen la mateixa estructura d'escenes. El joc és més simple que el *GeniusUp* en aquest aspecte. L'estructura es mostra a la figura 5.3.

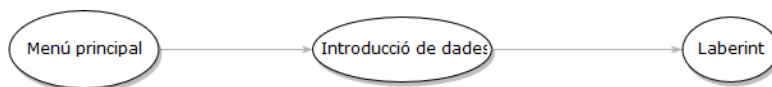


Figura 5.3: Estructura d'escenes del laberintl

La primera escena és la del *Menú Principal*, la qual permet o bé sortir de l'aplicació o bé començar a jugar. Al començar a jugar es porta a l'usuari a l'escena *Introducció de Dades*, en la qual el jugador introdueix el seu nom, edat i sexe. Finalment, un cop les dades s'han enregistrat, el jugador comença a jugar fins que l'objectiu particular s'hagi aconseguit.

L'escena *Menú Principal* i l'escena *Introducció de Dades* són molt simples, tan gràficament com lògicament. El *Menú Principal* gràficament consisteix només de dos botons i una imatge de fons. La lògica d'aquesta escena és purament d'esdeveniments, al clicar-se un dels dos botons es fa una acció determinada—al clicar el de sortir es tanca l'aplicació, al clicar el de jugar es carrega la següent escena. (figura 5.4)



Figura 5.4: Escena Menú principall

L'escena *Introducció de Dades* és molt similar en l'apartat gràfic. Té dos camps per a introduir text—nom i edat—i dues icones per escollir sexe. Finalment té el botó de començar a jugar. (figura 5.5)

La part lògica ja és una mica més interessant. Els botons de selecció de sexe i el de començar partida són purament programació per esdeveniments, al ser clicats es fa una acció. La part més interessant és com es guarden les dades de l'usuari.

*Unity* ens proporciona un mètode de mantenir objectes a través d'escenes. Normalment quan es carrega una nova escena, tots els elements que formaven l'anterior es perden. Per tant, si no es guardessin les dades en un fitxer o base de dades externa, es perdrien al carregar la següent escena. Com que les dades de l'usuari serveixen per a crear la carpeta on es guarden tant les variables de temps, col·lisions o longitud de camí com les fotografies—per així poder llavors classificar les dades obtingudes per sexe, edat, o jugadors individuals—, es necessari que l'escena *Laberint* les tingui disponibles.

La solució que hem usat per a guardar les dades entre escenes és la funció de *Unity*  `DontDestroyOnLoad()`, la qual mantindrà l'objecte des de que es crea fins a que es tanca l'aplicació. Per tant s'ha creat una classe de C# la qual contindrà tres propietats, el *nom del jugador*, l'edat, i el *gènere*. Així l'escena d'*Introducció de Dades* usa la funció `set` de cada propietat per guardar les dades de l'usuari, i l'escena *Laberint* usa la funció `get` per a accedir a elles.

Finalment ens queda comentar l'última escena, la més complexa, l'escena *Laberint*. Aquesta escena conté tots els laberints dissenyats. Gràficament l'escena és gairebé igual en totes les aplicacions del projecte. Inicialment hi ha visibles quatre elements: dues etiquetes, una bolla i un laberint. Les dues etiquetes indiquen el nivell en què es troba l'usuari i el seu nom, respectivament. La bolla es troba sempre a dins el laberint i és l'usuari el que la mou per tal de fer-la sortir. Finalment hi ha el laberint en si, el qual sempre ocupa el mateix espai en pantalla tot i tenir dimensions diferents segons el nivell—això s'ha aconseguit creant tots els laberints a partir de la mateixa proporció. (figures 5.6, 5.7 i 5.8)

Cal remarcar que en una versió interna del laberint apareixien a l'esquerra una sèrie de etiquetes que indicaven el temps, les col·lisions i la longitud del camí en temps

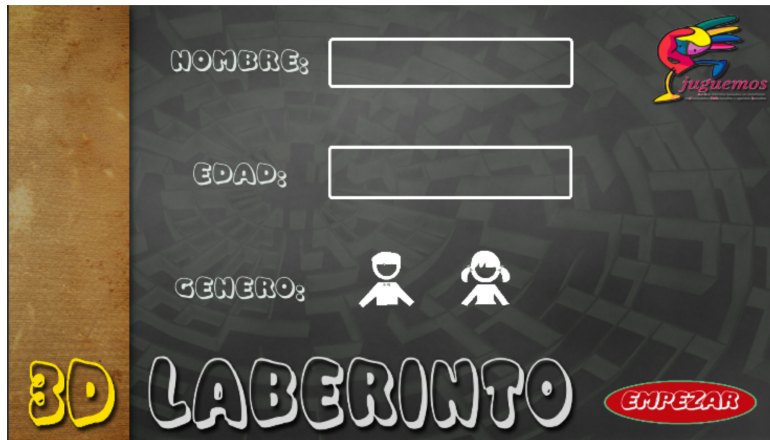


Figura 5.5: Escena Introducció de dades

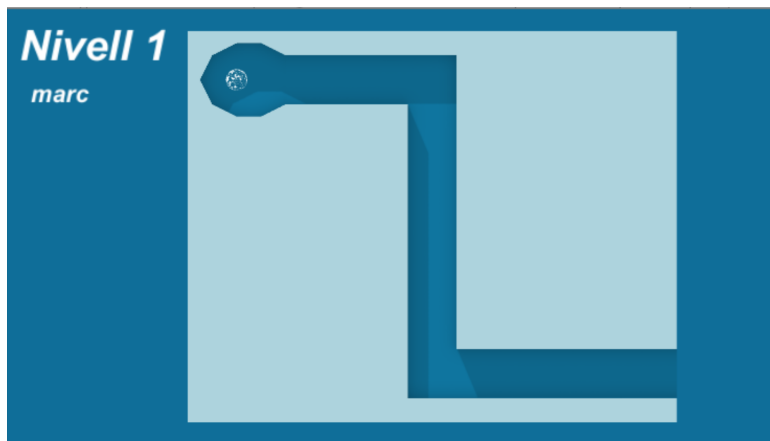


Figura 5.6: Escena Laberint, primera categoria de dificultat

real, per tal de poder assegurar el seu bon funcionament. En l'aplicació de recollida de dades s'han amagat per tal de no distreure el jugador. Ara bé, en l'aplicació de recollida de fotos, com s'ha comentat anteriorment, l'usuari té un límit de temps. Per tant, a part dels quatre elements en comú amb els laberints mostrats anteriorment apareix una etiqueta de temps restant a l'esquerra. (figura 5.9)

A més d'aquests quatre elements sempre visibles, existeixen una sèrie de missatges que van informant a l'usuari del seu progrés en el joc. Com que els missatges que es mostren també depenen del tipus d'aplicació—recollida de dades o recollida de fotos—s'explicaran per separat.

En l'aplicació de recollida de dades existeixen sis missatges:

- Finalització d'un nivell en particular. (figura 5.10)
- Finalització de ronda de laberints (des del nivell 1 fins al nivell que hagi assignat la fase de calibratge). (figura 5.11)

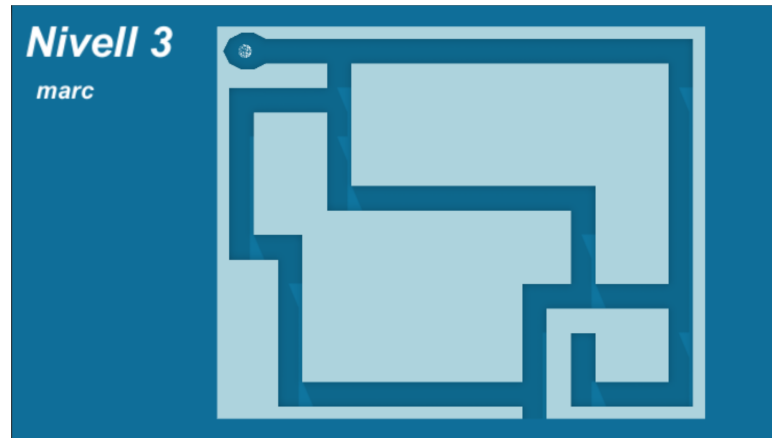


Figura 5.7: Escena Laberint, segona categoria de dificultat

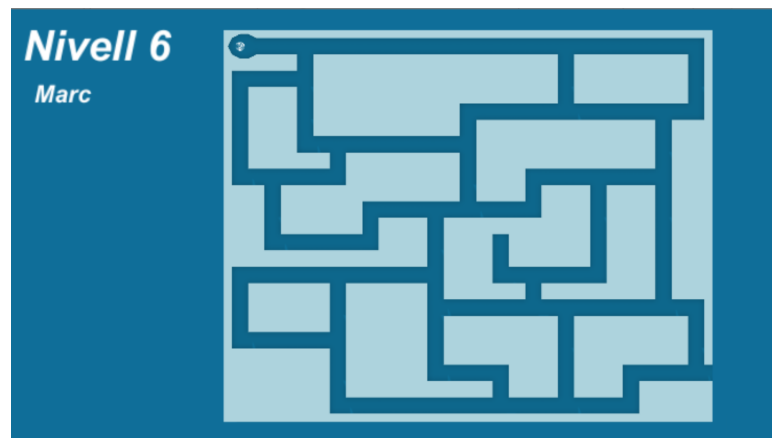


Figura 5.8: Escena Laberint, tercera categoria de dificultat

- Missatge d'ajuda inicial. (figura 5.12)
- Missatge d'inici de fase de calibratge. (figura 5.13)
- Missatge de final de fase de calibratge. (figura 5.14)
- Missatge de victòria (visible al final de totes les iteracions, quan l'usuari ha acabat tots els seus nivells). (figura 5.15)

En l'aplicació de recollida de fotos existeixen només quatre dels sis missatges anteriors—degut a què no té fase de calibratge—. El missatge de finalització d'un nivell, d'ajuda inicial i de victòria són idèntics als anteriors, i a més afegeix un nou missatge de *Game Over*, el qual apareix quan el temps s'ha acabat.

Finalment de la part gràfica cal comentar que tots els menús i missatges estan en castellà només, s'ha deixat com a feina per al futur afegir una selecció d'idioma.

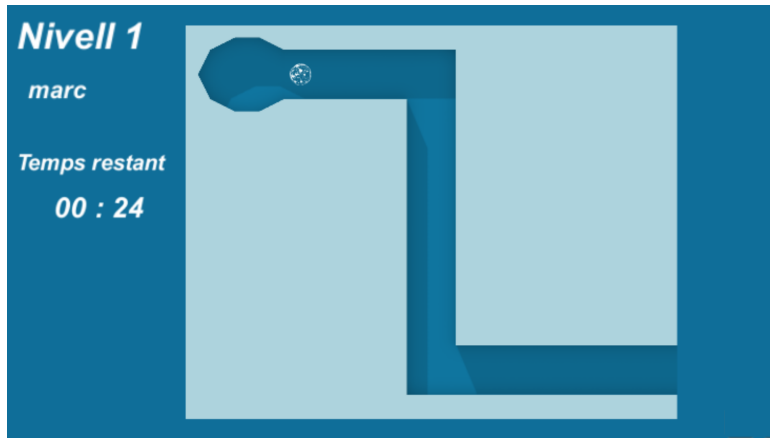


Figura 5.9: Escena Laberint, aplicació de recollida de fotos

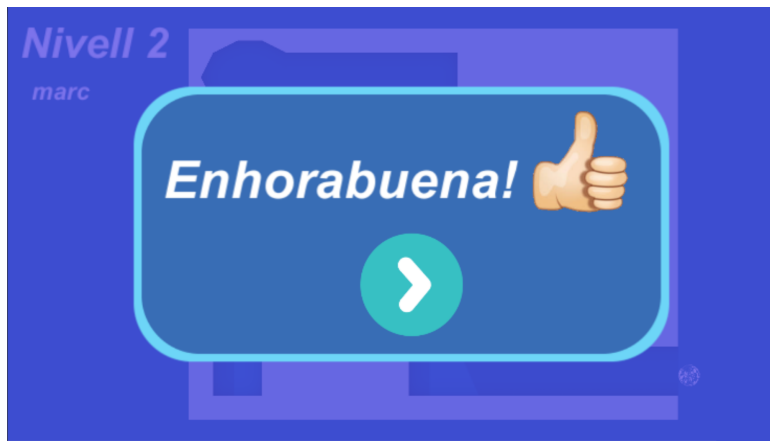


Figura 5.10: Missatge de finalització de nivell

### 5.4.2 Lògica de l'aplicació de recollida de dades

Un cop explicada tota la part gràfica, entrem dins la lògica. Ambdues aplicacions tenen un bucle de joc semblant, la funció *Update()*—la qual com ja s'ha explicat es crida varies vegades per segon—està contínuament comprovant la situació actual del jugador i actualitzant l'escena d'acord a aquesta situació.

Abans d'entrar en el contingut de la funció *Update()* però, explicarem primer les dues classes que permeten el funcionament del laberint. La primera s'ha anomenat *LevelManager* i és l'encarregada de carregar els nivells i mostrar els missatges corresponents. Al crear-se inicialitza el primer nivell, i ho fa a més en fase de calibratge.

L'altra classe s'ha anomenat *gameController*, i és l'encarregada de moure la bolla i controlar l'estat actual de jugador dins un nivell particular, així com guardar la informació que s'obté al final del nivell—temps, col·lisions i longitud del camí—en un fitxer. Al crear-se, al igual que la classe *LevelManager* inicialitza totes les variables internes, per

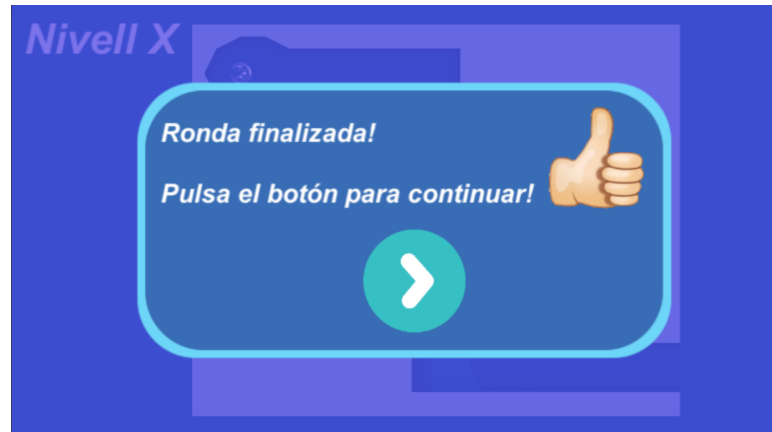


Figura 5.11: Missatge de finalització de ronda de nivells

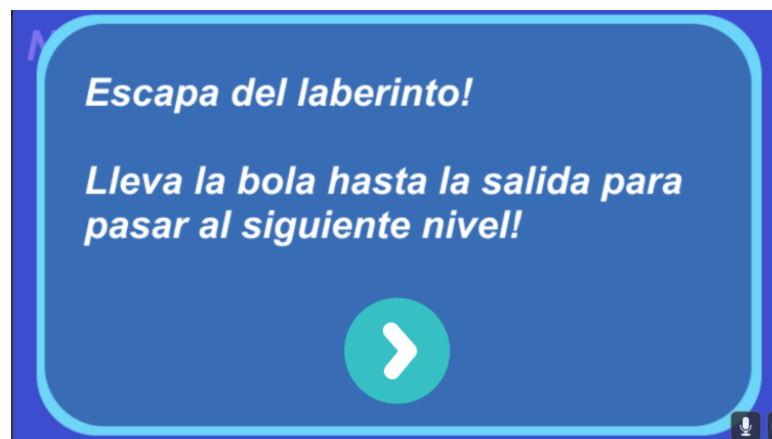


Figura 5.12: Missatge d'ajuda inicial

tal de començar de zero.

Llavors, un cop carregat el primer nivell en fase de calibratge, l'única classe que es va actualitzant via la funció *Update()* és la de *gameController*. La de *LevelManager* esperarà a rebre una notificació per part de *gameController* de què ha de carregar un nou nivell per a tornar a actuar.

La funció *Update()* consta d'una estructura molt clara. Primer de tot es mira si es vol sortir del joc mitjançant la tecla *Escape* o si el nivell s'ha acabat. Si no s'ha produït cap de les dues situacions, llavors realitza les següents accions:

- Actualitza el moviment de la bola i comprova col·lisions.
- Actualitza la longitud del camí.
- Comprova si la bola ha sortit del laberint.

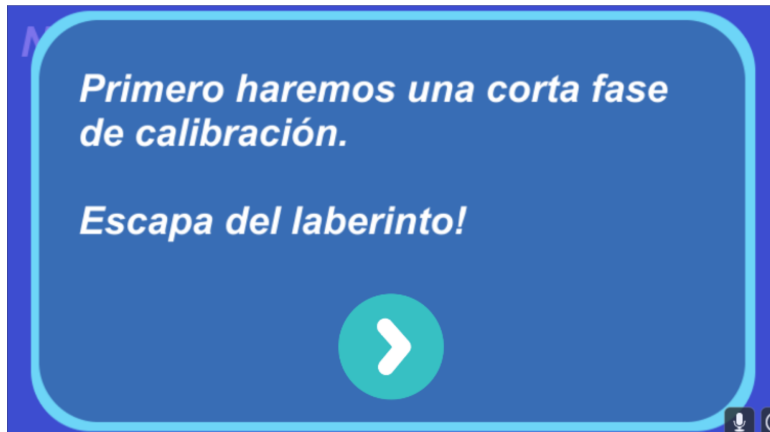


Figura 5.13: Missatge d'inici de fase de calibratge

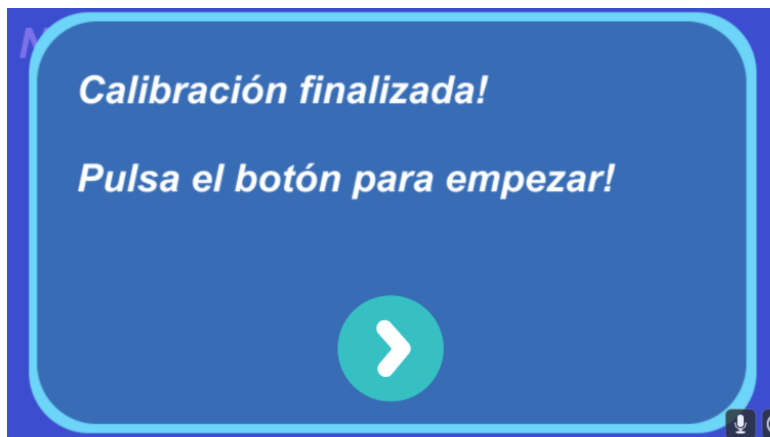


Figura 5.14: Missatge de fi de fase de calibratge

Inicialment el moviment de la bola està restringit només cap a la dreta—totes les sortides parteixen inicialment cap a la dreta—. Llavors, un cop s'ha començat el moviment, es mira quina direcció té clicada el mètode d'interacció, sigui un *joystick* o un *teclat*. Després, es realitza una simple comprovació per tal de saber si la bola es pot moure en aquella direcció. Es llança un raig de llum mitjançant la funció de *Unity Physics.Raycast* en la direcció de moviment, i si aquest raig rebota a una distància inferior a un límit establert, es determina que hi ha una paret en aquella direcció i que per tant la bola no pot avançar. Aquest límit establert s'ha obtingut a través de diverses proves per tal de veure quin límit permetia acostar-se més a les parets sense que es detectés una col·lisió.

Evidentment quan el raig de llum rebota a una distància inferior al límit establert, la bola entra en col·lisió. Ara bé, no es pot sumar una unitat al total de col·lisions cada cop que això passa, perquè cal recordar que la funció *Update()* es crida moltes vegades per segon. Això faria que el nombre final de col·lisions fos altíssim. Per a arreglar-ho



Figura 5.15: Missatge de victòria

s'ha creat una variable global que indica si la bolla està en un estat de col·lisió o no. Si és així, les col·lisions totals no augmenten.

Ara bé, aquest mètode tampoc era perfecte, ja que tot i estar en col·lisió en una direcció, la bolla es pot moure en una altra direcció simultàniament—si la bolla xoca amb una paret a la dreta encara es pot moure cap a dalt, per exemple—. Per tant, com que l'estat de col·lisió es llevava quan la bolla es movia en una direcció, el que passava en aquests casos era que es contava una col·lisió a la dreta, però al avançar cap a baix es llevava l'estat de col·lisió. A la següent iteració de la funció *Update()* la bolla ja no estava en col·lisió—el moviment cap a baix l'hi havia canviat l'estat—, i llavors es tornava a contar col·lisió a la dreta, augmentant de nou les col·lisions totals fins a arribar un altre cop a nombres altíssims.

La solució final, a més de l'estat de col·lisió o no, guarda també la direcció en què ha ocorregut la darrera col·lisió. Així, només es lleva l'estat de col·lisió si el moviment que es fa és en direcció contrària a la darrera direcció de col·lisió. Seguint l'exemple anterior, al xocar a la dreta es posarà la bolla en estat de col·lisió. Però així com abans al moure la bolla cap a baix aquest estat es llevava, ara ja no. Només es llevarà quan ens moguem cap a l'esquerra. D'aquesta manera, si avancem aferrats a una paret no se'ns comptaran moltes col·lisions, només una, la primera.

Finalment el darrer aspecte a comentar del moviment és que és dependent del nivell en què es troba l'usuari. A nivells més alts, al ser laberints més grans en dimensió però que ocupen el mateix espai físic, la bolla s'ha de moure més lenta. Per això hi ha tres valors enters que divideixen la velocitat dels nivells de cada una de les tres categories per tal de regular la velocitat amb la que es mou la bolla.

La segona part de la funció *Update()* es actualitzar la longitud del camí. Per a fer-ho es calcula la posició actual i es mira si ha canviat respecte a la posició guardada de la iteració anterior. Si és així es determina que s'ha avançat una unitat dins el laberint. Això es pot fer així perquè el laberint està format per fitxes quadrades amb distintes formes. Per tant avançar una fitxa implicarà augmentar la longitud del camí en una unitat.

L'única particularitat important és com calcular la posició en forma de  $(x, y)$ , on  $x$



és la fila de fitxes i  $y$  la columna. La bolla inicialment es troba al centre de la primera fitxa. Però en coordenades del món aquesta posició és la  $(0,0)$ . El  $(0,0)$  real hauria de ser el cantó superior de la primera fitxa, no el centre, és per això que aquesta posició  $(0,0)$  hauria de ser realment  $(\frac{l}{2}, \frac{l}{2})$ , sent  $l$  la longitud del costat de cada fitxa. Per tant es suma aquest valor  $\frac{l}{2}$  a la posició actual de la bolla en coordenades del món i es calcula el mòdul respecte  $l$ —aquesta longitud de fitxa dependrà del nivell, a nivells majors, amb més fitxes, la longitud serà menor—. Així es sabrà mitjançant un nombre enter la posició (fila, columna) de la nostra bolla. Llavors s'augmentarà la longitud en una unitat quan aquesta posició hagi canviat respecte la iteració anterior.

La darrera comprovació a realitzar dins l'*Update()* és mirar si el jugador ha arribat al final. Tot i no ser una manera ideal, com que els laberints tots ocupen el mateix espai en la pantalla, per a determinar si la bolla ha sortit o no del laberint es mira la seva posició en coordenades del món virtual. Si aquesta posició és fora del requadre del laberint, es determina que el nivell s'ha finalitzat. Un cop el nivell s'ha finalitzat la funció *Update()* ja no realitza més comprovacions, sinó que espera a que la classe *LevelManager* carregui el següent nivell per a continuar.

Quan la funció *Update()* està aturada perquè el nivell ha finalitzat, el procediment a seguir és diferent segons si el joc està en fase de calibratge o no. Si el joc està en calibratge, es mira el límit de temps que hi havia establert per aquell nivell en particular. Si l'usuari ha tardat més, s'informa amb un missatge que s'ha acabat la fase de calibratge, i se li assigna a l'usuari el nivell anterior—és a dir, el darrer nivell que s'ha passat dintre dels límits de temps—. Sinó, es mostra el missatge de finalització de nivell i es deixa avançar a l'usuari al següent nivell. Si el missatge que veu l'usuari és el de calibratge finalitzat, llavors al clicar la fletxa de continuar la classe *LevelManager* tornarà a carregar el primer nivell, però aquest cop fora de la fase de calibratge. Si el missatge que clica l'usuari és el de nivell superat, *LevelManager* carregarà el següent nivell encara en fase de calibratge.

Si el joc ja no està en fase de calibratge, sinó en fase de joc real, en la qual es prenen dades, mentre no s'ha arribat al nivell de l'usuari—el que hagi obtingut a través de la fase de calibratge—es guarden les dades en una carpeta—personalitzada amb l'edat, sexe i nom del jugador—i es mostra el missatge de nivell superat, el qual al ser clicat farà que el *LevelManager* carregui el següent nivell. Si l'usuari ja ha arribat al seu nivell màxim, es mira les iteracions total que ha realitzat. Si ja les ha realitzat totes es mostra el missatge de victòria, sinó s'avisa de què ha acabat una ronda, s'augmenta en una unitat el nombre d'iteracions realitzades, i quan l'usuari clica la fletxa la classe *LevelManager* torna a carregar el primer nivell de tots. Del missatge de victòria només es pot anar al menú principal.

Per tant, per a concloure la part lògica de l'aplicació, la classe *LevelManager* carrega els nivells quan una de les fletxes dels missatges s'ha clicat, per tal d'així diferenciar quin nivell ha de carregar i amb quins paràmetres. Els deu nivells són deu *GameObjects* de *Unity* diferents, tots no visibles inicialment. Llavors carregar un nivell vol dir fer visible l'objecte corresponent al nivell a carregar i, si ja n'hi havia un de carregat, fer-lo no visible.

### 5.4.3 Lògica de l'aplicació de presa de fotos

La lògica de l'aplicació de presa de fotos és gairebé idèntica. Implementa les tres parts explicades en el punt anterior. L'única que no seria estrictament necessària seria el còmput de col·lisions, però com que encara no s'ha decidit del tot si es vol prendre dades mentre es fan fotos, s'ha deixat implementada.

La nova part lògica és la necessària per a fer les fotos. Primer de tot hi ha una classe que s'anomena *Webcam* la qual fa les fotos i les guarda a la carpeta que se li indiqui, segons l'emoció que es vol capturar. Consta de només una funció la qual crea la carpeta d'usuari segons la seva edat, nom i sexe—de manera idèntica a l'aplicació anterior—i llavors, dins aquesta carpeta, crea una nova carpeta amb el nom de l'emoció a capturar. En aquesta versió inicial només es capturen tres emocions, Neutral, Enfado i Alegría, a més de dues carpetes extres per a quan es guanya la partida i per a quan es perd—*Win* i *GameOver* respectivament.

L'altra part necessària per a fer les fotos és una nova funció que s'ha afegit dins l'actualització de la longitud del camí. Les fotos es fan en una sèrie de punts, i aquests punts s'expressen en coordenades (fila, columna). És per això que aquesta nova funció es fa dins la comprovació de longitud de camí, ja que és allí dins on es converteixen les coordenades del món virtual en coordenades (fila, columna). La funció nova anomenada *checkPhoto()* simplement mira si les coordenades en què es troba la bolla corresponen a un dels punts crítics. Si és així, es crida a la funció per a fer foto amb l'emoció que correspon a aquell punt crític en concret. Els punts crítics es mostren a la figura 5.16.

A més de la part que pren fotos aquesta aplicació també té un límit de temps a cada nivell. Així com amb la part gràfica només implicava afegir una etiqueta, dins la part lògica la implementació del límit de temps també és molt simple. Dins la funció *Update()* s'ha afegit un condicional, el qual comprova si el temps restant s'ha acabat. Si és així es posa l'estat del nivell com a finalitzat i apareix el missatge de Game Over. Es fa una foto i quan es clica la fletxa la classe *LevelManager* recarrega el mateix nivell.

En aquesta aplicació en concret no hi ha iteracions ni calibratge, simplement el jugador va avançant nivells si se'ls passa amb suficient temps fins que es cansa o arriba al final. Així i tot, tot i aquestes diferències lògiques, aquestes dues aplicacions demostren la versatilitat dels motors de videojocs, ja que a partir d'un mateix model s'han creat dos videojocs similars però que persegueixen objectius diferents.

## 5.5 Verificació amb grups de control i anàlisi de dades

A la data de finalització d'aquest Treball Final de Grau (TFG) les dues aplicacions s'han implementat als centres Asociación de Parálisis Cerebral de España (ASPACE) i *Gaspar Hauser*, on nens i nenes amb *autisme*, *dèficit d'atenció* i *paràlisi cerebral* han jugat als laberints, aportant uns primers resultats.

Com hem comentat abans l'aplicació que feia fotos de moment no està del tot definida, encara s'ha de veure si les fotos són adequades i concretar la metodologia d'estudi, així que les fotos obtingudes encara no s'han analitzat. En el cas de l'aplicació que recull dades però sí que l'estudi està ben definit, per el que s'ha pogut fer un anàlisi preliminar. [13] (figura 5.17)

Amb aquesta primera onada de resultats, les variables dependents—temps, col·lisions i longitud del camí—permeten inferir la velocitat amb la que s'ha mogut la bolla, els dubtes que ha tingut el jugador, els girs que ha pres, etc. Totes aquestes observacions s'han fet a partir de l'anàlisi de parelles de variables. Per exemple, si un jugador té un temps bastant major a la longitud del camí, es pot concloure que no estava massa atent. Alternativament, si la longitud és significativament menor que el nombre de col·lisions, es pot concloure que l'usuari no té suficient control sobre la bolla.

Els resultats obtinguts mostren que els nens amb necessitats especials poden finalitzar els laberints, però els valors de les variables dependents varien respecte al grup de control—nens i nenes agafats a l'atzar durant la fira de la ciència de la UIB—. Aquest fet confirma que el joc és una eina vàlida per a fer feina amb nens i nenes amb necessitats especials, ja que es veu gràficament les diferències entre grups, i per tant es pot inferir les capacitats tan físiques com psíquiques a treballar. (figura 5.18)

Entrant més en detall alguns dels participants amb *paràlisi cerebral* han mostrat una capacitat extremadament alta en les habilitats mentals necessàries per a solucionar el laberint, mentre que en canvi les seves habilitats motores no els han permès moure la bolla ràpida i fluidament (figura 5.19). Els nens i nenes amb *autisme* demostren una atenció molt variable, finalitzant alguns nivells molt ràpidament i altres molt lentament, independentment de la seva dificultat.

La diferència principal que s'extreu d'aquests primers resultats és que els nens i nenes amb necessitats especials no aconsegueixen veure el laberint al complet, tenen una visió més túnel i els hi costa imaginar la imatge al complet. La causa principal és la *falta d'atenció* i la falta de *capacitat de concentració* durant un període llarg de temps. Això fa que s'equivoquin més en el camí triat i que facin més col·lisions—tant perquè no estan atents i xoquen, o perquè no tenen suficient control—. Finalment, cal comentar que els resultats han estat una mica sorprenents, ja que hi ha hagut molta més varietat entre els nens i nenes de cada grup de la que s'esperava. És per això que la feina futura en l'aplicació haurà de ser intensiva per tal d'arribar a satisfer les necessitats de tots els usuaris.

Finalment comentar que totes les dades obtingues dels nens i nenes que han participat han seguit les regles de protecció de dades de l'estat espanyol.

## 5.6 Estat final i futur del projecte

L'estat final del projecte s'ha de comentar aplicació per aplicació, ja que tot i tenir el model del laberint en comú cada una d'elles ha acabat a una etapa diferent. L'aplicació que pren dades s'ha vist que està en una fase més avançada, ja ha obtingut uns primers resultats inicials. Aquests resultats s'han obtingut amb un entorn sense cap tipus d'estímul extra, és a dir, un entorn neutral. Llavors l'estat futur d'aquesta aplicació es basa en anar afegint nous estímuls—música, efectes sonors, nous colors, imatges, etc.—al llarg de distintes iteracions i veure com afecten aquests als resultats, obtenint així una manera de veure com afecta cada estímul a l'atenció, la concentració o el rendiment dels jugadors. Una altra possibilitat de cara el futur es la implementació del laberint en 3D. Cal comentar que el model del laberint creat ja és en tres dimensions, però es presenta amb una vista zenital. Per tant canviar la vista a una 3D no seria gens complicat, i una feina per al futur canviar una mica el paradigma de joc.

L'aplicació de fotografies encara està en un estat més prematur, per el que la feina futura serà més gran. Principalment primer de tot s'ha d'establir si els punts crítics triats són adequats, és a dir, si les fotos mostren les emocions del jugador durant la partida. Si no, s'haurà de o bé triar nous punts o bé replantejar el mètode d'obtenció d'imatges. Llavors ja es podrà tenir una base de dades d'imatges fiables que representen les emocions dels jugadors durant distints moments claus de la partida, informació que després es podrà usar en recerques i estudis variats.

Cal comentar que tot i que els aspectes d'accessibilitats estan menys avançats que en el *GenisuUp*, ja s'ha creat un mode d'alt contrast, on els colors de fons, parets del laberint i bolla són suficientment diferents com per a ser visibles per a qualesvol persona. A més, com ja s'ha explicat, té un mètode d'interacció extra—la *Myo band*—, el qual permet a persones amb limitacions físiques poder jugar igualment.

Un punt a part que també cal tornar a comentar és la tercera aplicació que s'ha realitzat dintre del marc del projecte *Juguemos*. No s'hi ha entrat en detalls per raons explicades anteriorment, però en un futur pròxim s'espera també que es pugui catalogar com a *serious game*. De moment, cal que els experts en *autisme* verifiquin la seva validesa. Un cop fet, s'haurà d'iniciar un procés de feedback entre experts i desenvolupadors del projecte per tal de veure quin camí segueix l'aplicació, quines noves funcionalitats s'hi afegeixin i, en definitiva, quin objectiu i utilitat té.

## 5.6. Estat final i futur del projecte

	Neutral	Happy	Angry	Fear	Disgust	Surprise	Sad
<b>Lab. 1.1</b>	3A	4D Win	G.O.		G.O.	Win	G.O.
<b>Lab. 1.2</b>	3A, 3B	4D Win	1C G.O.		G.O.	Win	G.O.
<b>Lab. 2.1</b>	3B, 8F	6H, 7H, 7G Win	10G, 10H, 9H, 8H, 8G G.O.		G.O.	Win	G.O.
<b>Lab. 2.2</b>	3D, E3, E2	2G, 2H, 3H Win	10G, 10H, 9H, 8H, 8G, 4E, 5E, F5, F4. G.O.		G.O.	Win	G.O.
<b>Lab. 2.3</b>	3D, E3, E2, 8F, 9F, 10F, 2G, 2H, 3H.	9H, 10H, 10G. Win	4E, 5E, F5, F4. C7, C8, C9, B9. G.O.		G.O.	Win	G.O.
<b>Lab. 3.1</b>	14C, 8F, 7F, 9L.	J15, K15, K14. Win	9H, 9G. G.O.		G.O.	Win	G.O.
<b>Lab. 3.2</b>	9L, 9K, 11A, 3A, 4H, 4I, 4J.	H14, H15. Win	C5, B5, B6. G9, H9. L2, K2, J2. G.O.		G.O.	Win	G.O.
<b>Lab. 3.3</b>	9L, 9K, 11A, 3A, 4D, 5D, 14I. H14, H15.	L2, K2, J2. Win	C5, B5, B6. G9, H9. G.O.		G.O.	Win	G.O.
<b>Lab. 3.4</b>	A3, B3, A11, C11, D14, 4H	H14, H15. Win	C5, B5, B6. G9, H9. E5, F5. L2, K2, K3, L3. G.O.		G.O.	Win	G.O.
<b>Lab. 3.5</b>	A3, B3, A11, D14, 7H, I1, H1, I2.	H14, H15. Win	C5, B5, B6. E6, E5, F5, F4. I6, I5, J5, K5. L2, K2, K3, L3. G.O.		G.O.	Win	G.O.

**G.O.:** Game Over.

**Win:** Cuando termina y pasa de nivel.

Figura 5.16: Taula amb els punts crítics i les emocions a capturar

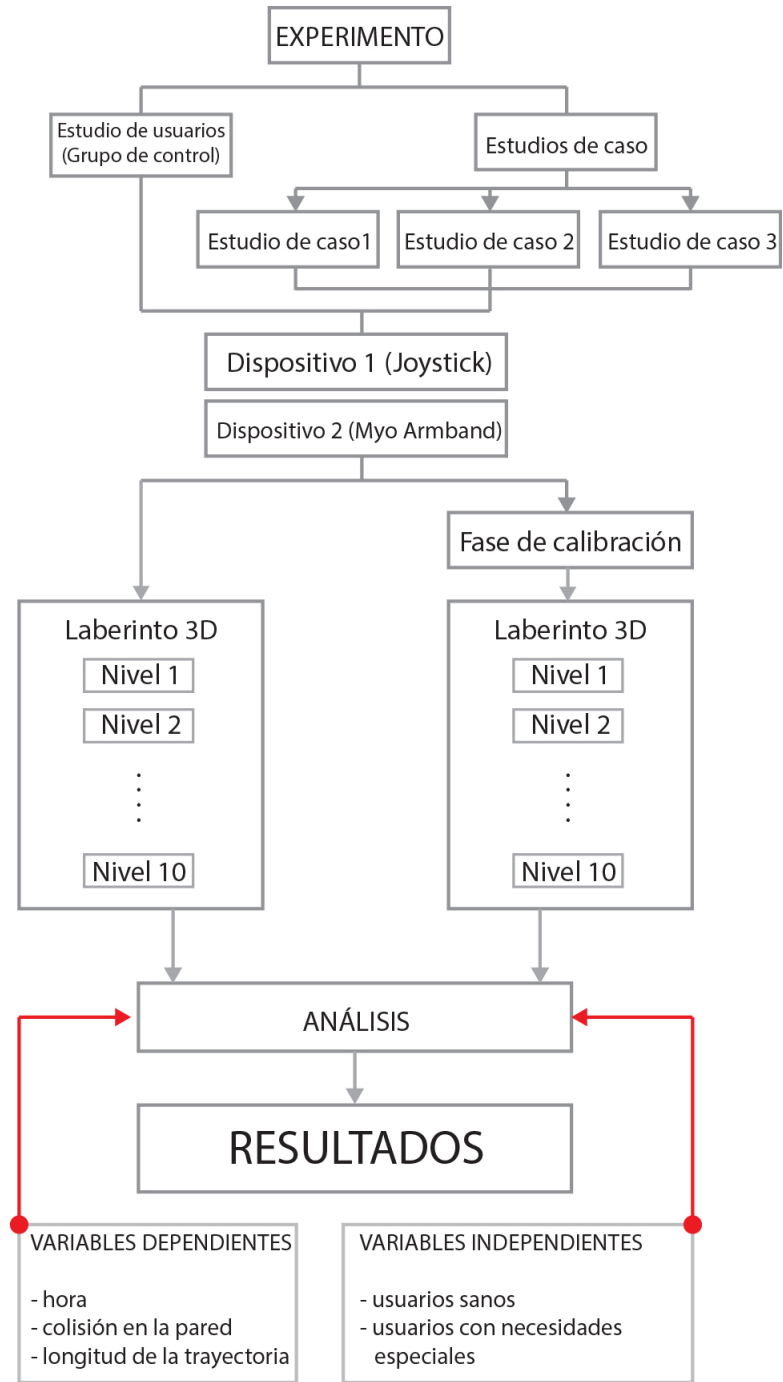


Figura 5.17: Diagrama de la metodología de l'estudi

## 5.6. Estat final i futur del projecte

9 years old children (5 participants)

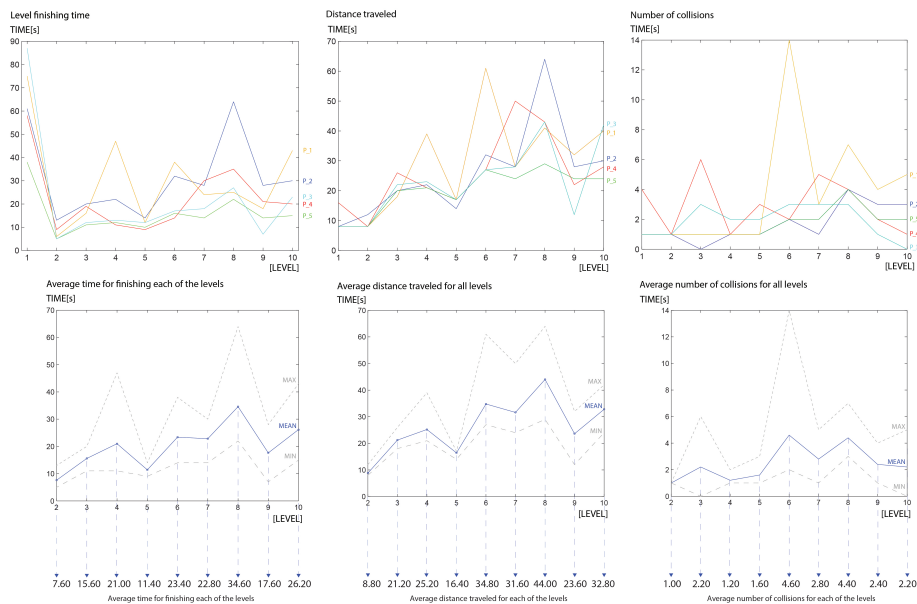


Figura 5.18: Resultats de cinc participants de nou anys

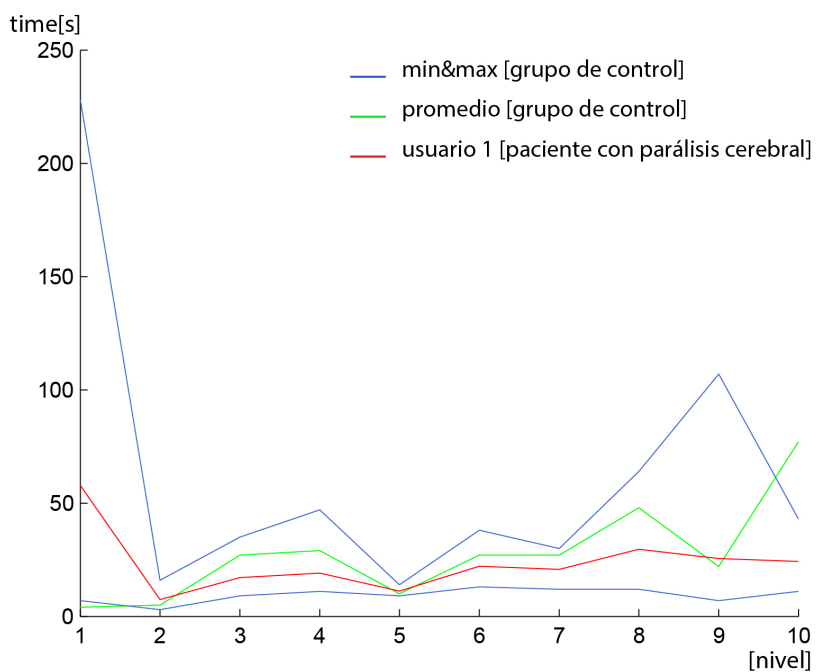


Figura 5.19: Comparació del temps entre la mitjana i una nena amb paràlisi cerebral





## CONCLUSIONS

### 6.1 Assoliment d'objectius

Tots els projectes i videojocs realitzats tenien el mateix objectiu general: aconseguir aplicar la filosofia dels serious games i crear videojocs dins un marc definit els quals contribuïssin tant a l'educació i promoció de la UIB com la investigació i recerca. Els objectius majoritàriament s'han realitzat. Evidenment hi ha espai per a millora, com s'ha explicat en els punts de estat futur de cada un dels projectes, però en general els resultats finals són molt satisfactoris.

En el *GeniusUp* s'ha aconseguit:

- Crear una aplicació funcional amb deu nivells, tant de la branca de biologia com la de química
- Verificar els coneixements que transmet el joc mitjançant experts en el tema.
- Adaptar tot el contingut a tres idiomes diferents.
- Guardar puntuacions i rècords per a motivar als jugadors.
- Crear l'aplicació modularment, preparada per a futures expansions.
- Desenvolupar un videojoc que compleix els principis bàsics d'accessibilitat.
- Penjar l'aplicació a les botigues de mòbil més importants, rebent molt bones valoracions i descàrregues.

En el *laberint* s'ha aconseguit:

- Crear un entorn comú funcional i extensible, preparat per a realitzar diferents estudis.
- Definir un marc de creació de nivells, per a tal de poder crear laberints consistents.

- Obtenir dades inicials de grups de control, per tal de formar una base de dades inicial.
- Realitzar l'estudi a distints centres amb els grups d'usuaris als que va destinat el laberint.
- Desenvolupar dos mètodes d'interacció distints, funcionals en tots els laberints.

Finalment cal comentar que el *GeniusUp* va tenir repercussió mediàtica al ser publicat, sortint la notícia a varis diaris, i que de la feina realitzada n'han sorgit dos papers dels que som co-autor ja publicats—referents al projecte Juguemos—, i un tercer que s'està acabant de redactar per a ser enviat—referent al GeniusUp. 7

### 6.2 Experiència personal

Personalment ha estat una molt bona experiència, que m'ha obert portes a noves oportunitats les quals al principi del treball no havia ni pensat poder fer. Inicialment volia fer alguna cosa relacionada amb els gràfics, però amb el pas del temps he acabat fent molt més.

He pogut dissenyar algunes icones i elements gràfics, com jo volia inicialment, però a més he acabat dissenyant també nivells i videojocs sencers. He après una nova eina de treball, Unity, la qual dins el món dels videojocs és un plus important. He treballat amb altres persones, siguin professors o investigadors, i he vist també el que és treballar amb la pressió d'unes dates límit. He rebut crítiques tan positives com negatives, així com validacions i valoracions d'experts, i m'he hagut d'adaptar als seus comentaris per tal de millorar i adequar els videojocs.

Finalment, gràcies als contactes i amb les capacitats adquirides durant l'inici del meu treball de fi de grau, he aconseguit una feina remunerada dins la pròpia universitat, afegint més hores dedicades a la feina i més experiència laboral.

En resum, ha estat una experiència molt bona que ha resultat en una feina de la qual estic molt orgullós i que a més ha millorat tant el meu currículum i les meves habilitats, com el meu caràcter i experiència personal.



## ANNEXOS

### 7.1 A.1 Repercussió mediàtica

UltimaHora, <https://ultimahora.es/noticias/local/2017/02/22/250284/geniusup-videojuego-made-uib-fomenta-interes-por-ciencia.html>

EuropaPress, <http://www.europapress.es/illes-balears/noticia-geniusup-videojuego-hecho-uib-fomentar-interes-ciencia-20170222140232.html>

LaVanguardia, <http://www.lavanguardia.com/vida/20170222/42225137829/un-equipo-de-la-uib-lanza-el-videojuego-geniusup-para-fomentar-el-interes-por-la-ciencia.html>

### 7.2 A.2 Google Play Store i App Store

GeniusUp, Google Play Store, <https://play.google.com/store/apps/details?id=es.uib.ugivia.geniusup&hl=es>

GeniusUp, App Store, <https://itunes.apple.com/us/app/geniusup/id1209063623?mt=8>

### 7.3 A.3 Papers projecte Juguemos

*Serious games: tool for human computer interaction for children with special needs*, Milica Vujovic and Francisco J. Perales and Marc Campins and Markos Milos and Miquel Mascaró, IcETRAN Conference, June, 2017, Kladovo, Serbia.

*Un videojuego serio para el estudio de expresiones faciales en personas con Autismo*, Silvia Ramis, Francisco J. Perales, Marc Campins, UIB, España.



## BIBLIOGRAFIA

- [1] R. Dörner, S. Göbel, W. Effelsberg, and J. Wiemeyer, *Serious Games: Foundations, Concepts and Practice*, 1st ed. Springer International, 2016. (document), 2.1.4, 2.2, 2.1, 2.5, 3.1, 3.2
- [2] H. A. Rosyid, M. Palmerlee, and K. Chen, “Deploying learning materials to game content for serious education game development: A case study,” 2016. (document), 4.2, 4.3
- [3] F. J. O. Quetglas, “Motor de videojuegos para plataforma android,” 2014. 1.1
- [4] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, “From game design elements to gamefulness: Defining “gamification”,” in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, Tampere, Finland, September 2011, pp. 9–15. 2.1.1
- [5] K. Huotari and J. Hamari, “Defining gamification: a service marketing perspective,” in *Proceeding of the 16th International Academic MindTrek Conference*, Tampere, Finland, October 2012, pp. 17–22. 2.1.3
- [6] P. Rego, P. M. Moreira, and L. P. Reis, “Serious games for rehabilitation: A survey and a classification towards a taxonomy,” in *Information Systems and Technologies, 5th Iberian Conference*, Santiago de Compostela, Spain, June 2010. 2.1.4
- [7] V. Kasapakis and D. Gavalas, “Pervasive gaming: Status, trends and design principles,” *Journal of Network and Computer Applications*, vol. 55, pp. 213–236, 2015. 2.2.5
- [8] J. D. Craighead, J. Burke, and R. R. Murphy, “Using the unity game engine to develop sarge: A case study,” January 2007. 3.2
- [9] A. G. y Francisco J. Perales y Antoni Jaume y Jose Maria Buades, “Potugame. un videojuego para conocer la universidad,” *VAR*, vol. 4, 2013. 4.1
- [10] L. P.-C. Alvarez, *Buenas prácticas de accesibilidad en videojuegos*, 1st ed. Ministerio de Sanidad, Servicios Sociales e Igualdad, 2012. 4.5.1, 4.5.2, 4.5.3
- [11] B. Dougherty and A. Wade. Vischeck. [Online]. Available: <http://www.vischeck.com/> 4.5.4
- [12] T. Labs. Myo. [Online]. Available: <https://www.myo.com/> 5.3

## BIBLIOGRAFIA

---

- [13] M. Vujovic, F. J. Perales, M. Campins, M. Milos, and M. Mascaró, "Serious games: tool for human computer interaction for children with special needs," in *IcETRAN Conference*, Kladovo, Serbia, June 2017. 5.5