



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques
Universitat de Barcelona**

**JUEGOS FARMA UB
APLICACIÓ ANDROID
D'EDUCACIÓ PER A LA SALUT**

Ivan Alcoba Subirats

Director: Laura Igual Muñoz
Realitzat a: Departament de Matemàtica
Aplicada i Anàlisi. UB

Barcelona, 30 de juny de 2016

Abstract

The project which is documented in this paper under the name of “Juegos Farma UB: Aplicació d’educació per a la salut” consists in the development of an Android app comprising Games and Educational Health Guidebooks by using the Unreal Engine 4 game engine.

The whole process involved in the app development is detailed in this paper, from the previous design considerations to the app creation, and the following ultimate launching in Google Play Store.

Resumen

El proyecto que se documenta en la presente memoria con el nombre de «Juegos Farma UB: Aplicació d’educació per a la salut» consiste en el desarrollo de una aplicación Android de juegos y guías educativas para la salud mediante la utilización del motor de videojuegos Unreal Engine 4.

En dicha memoria se detalla todo el proceso implicado en el desarrollo de la aplicación, desde el estudio previo del diseño hasta la creación y la subsiguiente publicación final de la aplicación en la Play Store de Google.

Resum

El projecte documentat en aquesta memòria amb el nom de «Juegos Farma UB: Aplicació d’educació per a la salut» consisteix en el desenvolupament d’una aplicació Android de jocs i guies educatives per a la salut mitjançant la utilització del motor de videojocs Unreal Engine 4.

En aquesta memòria es detalla tot el procés implicat en el desenvolupament de l’aplicació, des de l’estudi previ del disseny fins la creació i la posterior publicació de l’aplicació final a la Play Store de Google.

Índex

| | |
|---|----|
| 1. Introducció..... | 6 |
| 1.1. Context..... | 6 |
| 1.2. Objectius | 7 |
| 1.3. Treball previ..... | 8 |
| 1.4. Contingut de la memòria | 8 |
| 2. Estudi..... | 10 |
| 2.1. Disseny de jocs mòbils | 10 |
| 2.2. Avaluació tecnològica | 10 |
| 2.3. Tecnologies utilitzades | 12 |
| 2.3.1. Unreal Engine | 12 |
| 2.3.2. Visual Studio | 13 |
| 2.3.3. Adobe Photoshop | 13 |
| 2.3.4. Github..... | 14 |
| 2.3.5. Audacity | 14 |
| 3. Desenvolupament..... | 15 |
| 3.1. Conceptes previs | 15 |
| 3.1.1. Level..... | 15 |
| 3.1.2. Actor | 16 |
| 3.1.3. Camera..... | 17 |
| 3.1.4. Lights..... | 18 |
| 3.1.5. AmbientSound..... | 18 |
| 3.1.7. Widget Blueprint | 19 |
| 3.1.8. SaveGameObject..... | 19 |
| 3.1.9. DataTable i Structure | 20 |
| 3.2. Interfície gràfica i menús | 21 |
| 3.2.1. Menú Principal | 21 |
| 3.2.2. Menú Opcions | 22 |
| 3.2.3. Pantalla de crèdits..... | 23 |
| 3.2.4. Pantalla de sobre nosaltres | 23 |
| 3.2.5. Pantalla de selecció de jocs | 24 |
| 3.2.6. Pantalla de guies..... | 25 |

| | |
|--|----|
| 3.2.7. Pantalla de fites | 26 |
| 3.3. Joc: Activitat Física Quiz..... | 27 |
| 3.3.1. Level | 27 |
| 3.3.2. GameDataObject | 27 |
| 3.3.3. DataTable & Structure | 28 |
| 3.3.4. Animations | 28 |
| 3.3.5. Preguntes | 29 |
| 3.3.6. Resultats | 29 |
| 3.4. Joc: Oca de l'Alcohol | 30 |
| 3.4.1. Level | 30 |
| 3.4.2. GameDataObject | 31 |
| 3.4.3. Actors | 31 |
| 3.4.4. Animations | 31 |
| 3.4.5. Lògica i resultats | 32 |
| 3.5. Joc: Farma Trivial..... | 33 |
| 3.5.1. Level | 33 |
| 3.5.2. GameDataObject | 33 |
| 3.5.3. Lògica & Preguntes | 33 |
| 3.5.4. Resultats | 34 |
| 3.6. Joc: Escales i Serps de les Vacunes | 35 |
| 3.6.1. Level | 35 |
| 3.6.2. GameDataObject | 35 |
| 3.6.3. Actors | 35 |
| 3.6.4. EscSerps_Engine | 36 |
| 3.6.5. Lògica i resultats | 36 |
| 3.7. Joc: Oca dels Polls | 37 |
| 3.7.1. Level | 37 |
| 3.7.2. GameDataObject | 37 |
| 3.7.3. Actors | 37 |
| 3.7.4. Lògica i resultats | 38 |
| 4. Proves i resultats..... | 39 |
| 4.1. Testing d'interfície gràfica i menús..... | 39 |
| 4.2. Testing Activitat Física Quiz..... | 39 |

| | |
|---|----|
| 4.3. Testing Oca de l'Alcohol..... | 40 |
| 4.4. Testing Farma Trivial | 40 |
| 4.5. Testing Escales i Serps de les Vacunes | 41 |
| 4.6. Testing Oca dels polls..... | 41 |
| 5. Conclusions | 42 |
| 5.1. Possibles ampliacions | 43 |
| 6. Bibliografia..... | 44 |
| 7. Annex | 46 |
| 7.1. Guia d'instal·lació de l'aplicació..... | 46 |

1. Introducció

Juegos Farma UB és una aplicació Android de jocs i guies educatives per a la salut desenvolupat com a Treball Final de Grau del Grau d'Enginyeria Informàtica de la Universitat de Barcelona utilitzant el motor de videojocs Unreal Engine 4.

El desenvolupament d'aquesta aplicació s'ha dut a terme mitjançant la transferència de coneixements entre la Facultat de Matemàtiques i la Facultat de Farmàcia de la Universitat de Barcelona com una iniciativa d'Aprenentatge i Servei.

El contingut didàctic de l'aplicació s'ha obtingut a partir del material d'educació al ciutadà de les pràctiques tutelades de la Facultat de Farmàcia de la UB.

1.1. Context

La Facultat de Farmàcia de la Universitat de Barcelona disposa d'un gran nombre de recursos educatius creats pels seus estudiants com a treballs de final de grau. Aquests recursos tracten sobre diversos aspectes de l'educació per a la salut i són utilitzats com a material educatiu en les diverses activitats socials (xerrades, presentacions, etc.) que realitzen per educar la ciutadania en diversos actes i llocs, com ara centres educatius.

Tot i tenir el material per poder realitzar aquestes activitats educatives, han observat que el material físic, concretament els jocs que utilitzen per reforçar l'aprenentatge, en la majoria dels casos no possibiliten la participació de tots els integrants de l'activitat a causa de la seva limitació d'ús. Aquesta limitació ha propiciat la temàtica d'aquest projecte per digitalitzar aquest material en forma d'aplicació per a dispositius mòbils Android.

Pel fet que l'educació farmacèutica comprèn moltes àrees i rangs d'edats a qui afecten, s'ha decidit enfocar l'aplicació a les categories temàtiques que afecten a la població jove, principalment en els rangs d'edat d'educació primària i secundària. Una de les raons primordials d'aquesta decisió ha estat que aquest col·lectiu de la població és el més familiaritzat amb la utilització de les noves tecnologies.

1.2. Objectius

L'objectiu principal d'aquest Treball de Fi de Grau és desenvolupar una aplicació de jocs educatius per a la plataforma Android mitjançant la utilització del motor de videojocs Unreal Engine 4. Aquesta aplicació ha de ser el més completa possible, ja que la finalitat és obtenir un producte final de software i, així, completar un cicle sencer de desenvolupament de software des de la primera fase d'anàlisi dels requisits fins l'obtenció del producte final.

Així doncs, els objectius principals que es pretén assolir amb la realització d'aquest projecte són els següents:

- Estudiar i dur a terme totes les fases involucrades en la creació de videojocs.
- Explorar i utilitzar les capacitats proporcionades pel motor de creació de videojocs Unreal Engine 4 enfocat a la plataforma Android.
- Implementar una interfície i sistema de menús atractius i intuïtius per a l'usuari.
- Disseny de l'aplicació multidioma. Mínim català-castellà, amb la possibilitat d'extensió a més idiomes com ara l'anglès.
- Presentar una secció amb la possibilitat de jugar a diversos jocs educatius breus amb diferents temàtiques d'educació per a la salut.
- Presentar una secció amb una sèrie de guies educatives sobre les temàtiques d'educació per a la salut tractades en els jocs.
- Presentar una secció de fites que l'usuari podrà aconseguir en completar diferents accions en els jocs de l'aplicació i poder tenir un seguiment del seu progrés.
- Desenvolupar una aplicació final i usable per a la plataforma Android.
- Pujar l'aplicació a la Play Store de Google per fer-la pública i accessible per a qualsevol usuari.

1.3. Treball previ

Un cop determinada la temàtica del projecte, com s'ha descrit en els apartats anteriors, s'ha realitzat una fase prèvia al procés de desenvolupament de l'aplicació per formar-se en els diversos aspectes que s'interrelacionaran durant la realització d'aquest projecte.

Tal com es detalla a l'apartat 2.2., s'ha utilitzat el motor de videojocs Unreal Engine 4 per desenvolupar l'aplicació. Així doncs, en aquesta fase, s'ha preparat l'entorn de treball necessari, s'ha instal·lat i configurat el motor i s'ha estudiat la documentació online d'aquest motor per profunditzar en les funcionalitats de l'editor i les configuracions pertinents per realitzar un projecte Android. També s'han realitzat una sèrie de vídeotutorials amb projectes d'exemple de la pàgina oficial d'Unreal Engine amb parts i funcionalitats semblants a les necessàries per al desenvolupament de la nostra aplicació.

1.4. Contingut de la memòria

Aquesta memòria recull tot el procés implicat en el desenvolupament de l'aplicació Juegos Farma UB de forma estructurada en les diverses fases del projecte, des de l'estudi previ de disseny fins la creació i publicació de l'aplicació final a la Play Store de Google.

Així doncs, a l'apartat 2 es realitza un estudi del context en el qual s'emmarca el disseny de jocs per plataformes mòbils, s'avaluen les possibilitats tecnològiques per desenvolupar una aplicació de jocs per a Android i es detallen les tecnologies emprades per a la realització d'aquest projecte.

A l'apartat 3, es mostra l'anàlisi realitzat amb anterioritat al desenvolupament de l'aplicació, el qual detalla el contingut de l'aplicació, els casos d'ús, els...

Tot seguit, a l'apartat 4, s'exposa el procés d'implementació de cadascuna de les parts en les quals es divideix l'aplicació, com són els menús i la interfície, i cadascun dels jocs de forma independent a la resta.

A l'apartat 5, s'expliquen i s'analitzen les proves realitzades per verificar el correcte funcionament de l'aplicació després de la realització de les fases comentades en els apartats anteriors.

A l'apartat 6, es mostra el procés realitzat per generar l'executable del projecte i per publicar l'aplicació a la Play Store.

A l'apartat 7, es plantegen les conclusions obtingudes del desenvolupament d'aquest projecte, així com les possibles ampliacions que es podrien dur a terme.

Per últim, a l'apartat 8, s'enumeren les fonts bibliogràfiques emprades durant la realització d'aquest projecte i, a l'apartat 9, es presenten una sèrie d'annexos on s'explica com instal·lar l'aplicació i l'estructura de fitxers del projecte.

2. Estudi

2.1. Disseny de jocs mòbils

Actualment, la quantitat de jocs per a plataformes mòbils presents al mercat és immensa i els gèneres que existeixen són extremadament variats. Evidentment, per poder crear tal quantitat de jocs és necessària l'existència de tecnologies que agilitzin el procés de desenvolupament dels jocs, ja que és un procés llarg. Aquestes tecnologies són els motors de videojocs, els IDE i altre programari destinat a l'optimització de les tasques implicades en aquest procés, com els emuladors de dispositius per testejar les aplicacions en diferents dispositius de forma virtual sense la necessitat de disposar físicament d'aquests.

Entre els motors de videojocs més utilitzats, accessibles i potents del mercat, actualment destaquen Unity3D i Unreal Engine 4. Altres motors com són Game Salad, Game Maker i Construct 2, entre molts altres, també aporten moltes funcionalitats però encara no han arribat al nivell d'especialització dels dos primers.

2.2. Avaluació tecnològica

Ja que el nostre projecte es basarà en jocs senzills per a la plataforma Android i s'utilitzaran tecnologies 2D/3D, que és un punt que ens interessa estudiar en aquest projecte, hem decidit comparar les dues millors tecnologies possibles per utilitzar. Aquestes tecnologies són Unity3D i Unreal Engine 4.

Per tal de decidir quina de les dues tecnologies escollir pel desenvolupament d'aquest projecte, s'ha realitzat un anàlisi dels avantatges d'una vers l'altra.

Els avantatges que presenta Unity3D sobre Unreal Engine 4 són els següents:

- Millor suport per a jocs 2D.
- S'obté el 100% dels beneficis proporcionats per un joc desenvolupat amb la versió Unity 3D Pro (de pagament). Amb Unreal Engine 4, a partir d'un determinat nombre d'ingressos, s'ha de pagar un 5% dels beneficis totals.

- Escala d'aprenentatge molt més ràpida.
- Desenvolupament de jocs més senzill i ràpid.
- Empaquetat de instal·ladors per a plataformes mòbils de menor mida.

D'altra banda, els avantatges que presenta Unreal Engine 4 sobre Unity3D són els següents:

- Millor suport per jocs 3D.
- Major nombre de funcionalitats pel desenvolupament de jocs.
- Disponible el 100% de les funcionalitats del motor en la versió gratuïta. Per disposar del 100% de les funcionalitats del Unity 3D amb les optimitzacions per plataformes mòbils costaria entre uns 2000-4000 euros.
- Millor qualitat en l'aspecte gràfic.
- Disponible la funcionalitat de crear models 3D per utilitzar en un joc/nivell amb un editor d'objectes 3D intern.

A partir de l'anàlisi realitzat, s'ha conclòs que la millor opció és utilitzar Unity3D Pro ja que els jocs que desenvoluparem a la nostra aplicació seran de caràcter senzill i breu, i aconseguiríem un instal·lador menys pesat (mida en bytes) i en menor temps de desenvolupament amb una qualitat gairebé idèntica que utilitzant Unreal Engine 4.

La segona millor opció seria Unreal Engine 4 degut que aconseguiríem el joc amb la millor qualitat possible i de forma gratuïta. Tot i això, l'aplicació seria bastant més pesada i amb més cost de temps de desenvolupament.

La tercera i última opció seria Unity3D Free que ens limitaria moltes funcionalitats d'optimitzacions per plataformes mòbils, ens obligaria a mostrar publicitat del motor a l'aplicació i la qualitat de l'aplicació final seria menor que en la resta d'opcions.

Si és té en compte el preu del programari, la millor opció seria utilitzar Unreal Engine 4, ja que no disposem de la versió Unity3D Pro de forma gratuïta i s'hauria de

comprar el programari per un valor d'entre 2.000-4.000 euros per aconseguir unes prestacions pràcticament idèntiques a Unreal Engine 4, que és gratuït i únicament se'ns cobraria un percentatge del 5% sobre els beneficis obtinguts amb la comercialització d'una aplicació desenvolupada amb aquest motor.

2.3. Tecnologies utilitzades

A l'hora de realitzar un projecte de software és indispensable escollir les tecnologies amb les quals es treballarà abans de començar amb el desenvolupament. A partir de l'avaluació tecnològica que s'ha realitzat a l'apartat 2.2., s'ha decidit treballar amb el motor de videojocs Unreal Engine juntament amb el IDE Microsoft Visual Studio. A més a més, en el desenvolupament d'un joc hi intervenen altres aspectes, a part del codi implementat, com són la música, les textures i els dissenys, entre altres, que necessiten d'altres tecnologies per treballar-los.

En els apartats següents es detallen les tecnologies emprades en el desenvolupament d'aquest projecte.

2.3.1. Unreal Engine

Unreal Engine és un motor de videojocs per a PC i videoconsoles creat per la companyia Epic Games l'any 1998. Inicialment, aquest motor únicament s'utilitzava en el desenvolupament de jocs en primera persona, però en l'actualitat s'utilitza per desenvolupar tot tipus de jocs.

Actualment, l'última versió d'aquest motor és Unreal Engine 4 que està dissenyat per suportar DirectX 11 i 12 de Microsoft (Windows Microsoft, XBOX One) i OpenGL (OS X, Linux, Playstation 4, iOS, Android), a més a més, també dóna suport a JavaScript/WebGL per cercador amb HTML 5.

Ja que Unreal Engine suporta el desenvolupament en els sistemes operatius de Microsoft i Mac, utilitzant Visual Studio o XCode respectivament, en la realització d'aquest projecte s'ha treballat amb la versió de Microsoft juntament amb Visual Studio.

2.3.1.2. Llenguatge de programació Unreal Engine

Unreal Engine proporciona la possibilitat de programar el codi de l'aplicació mitjançant Blueprints (vegeu *figura 1*) o codi C++. La diferència principal és que el primer mètode utilitza scripts visuals mitjançant nodes amb un editor gràfic i el segon línies de codi amb Visual Studio. Tot i això, en ambdós casos es poden obtenir els mateixos resultats, ja que es poden crear nodes propis programats amb C++ i utilitzar-los com Blueprints.

A la nostra aplicació s'ha utilitzat la configuració del projecte amb Blueprints, tot i que s'ha utilitzat Visual Studio per veure i modificar el codi dels nodes amb funcionalitats que ofereix Unreal Engine en les seves API.

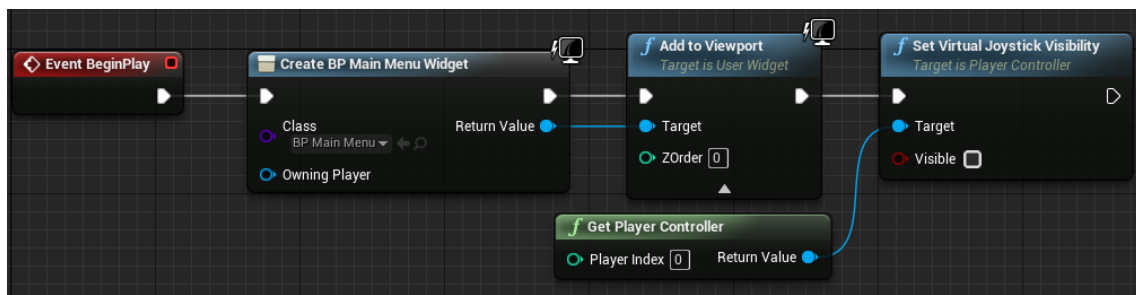


Figura 1. Exemple de Blueprint. Crea un Widget de tipus BP_MainMenu quan es llença el joc, el mostra en el viewport de la pantalla i oculta els joysticks virtuals

2.3.2. Visual Studio

Microsoft Visual Studio és un entorn IDE per sistemes operatius Windows. Suporta multitud de llenguatges de programació com C++, C#, Java, Python, etc. La versió que s'utilitza juntament amb Unreal Engine és Visual Studio 2013.

2.3.3. Adobe Photoshop

Adobe Photoshop és un programa creat per la companyia Adobe que s'utilitza per crear, editar, composar, retocar i transformar imatges. La seva funcionalitat de tractar i crear capes superposades sobre una imatge ens permet combinar diferents objectes i efectes sense necessitat d'alterar les imatges originals.

Durant la realització del projecte s'ha emprat aquest programa per dissenyar o retocar totes les textures i dissenys presents en l'aplicació.

2.3.4. Github

GitHub és una plataforma de desenvolupament de software que ens permet allotjar projectes al Web de forma gratuïta, mantenint una còpia de seguretat i un control de les versions del software amb la possibilitat de revertir canvis no desitjats.

S'ha utilitzat per mantenir una còpia de seguretat i un control de versions sobre aquest projecte.

2.3.5. Audacity

Audacity és un programa de gravació i edició d'àudios de codi lliure, multi plataforma i molt fàcil d'utilitzar.

S'ha utilitzat per quadrar el volum dels àudios utilitzats en el nostre projecte, així com ajustar l'inici i final del bucle de l'àudio de fons del menú principal.

3. Desenvolupament

3.1. Conceptes previs

Per tal d'entendre el procés de desenvolupament de les diferents parts que componen l'aplicació és necessari tenir clars alguns conceptes i elements relacionats amb el motor Unreal Engine que s'utilitzen en la creació de qualsevol joc o aplicació amb aquest software.

3.1.1. Level

En una aplicació o joc desenvolupat amb Unreal Engine, qualsevol objecte que es pugui veure o interactuar amb el món o escena en la que està contingut està ubicat en un Level. Per tant, es pot definir un Level com una col·lecció de volums, geometries, llums i Blueprints recollits en un objecte que permet que treballin junts per aportar una experiència d'usuari desitjada, com pot ser representar un terreny immens per donar la sensació que s'està enmig d'un extens món o un simple taulell amb unes fitxes, en el cas d'un mini joc.



Figura 2. Conjunt de Levels presents a Juegos Farma UB

Qualsevol aplicació o joc ha de començar amb l'execució d'un Level per poder executar la resta de l'aplicació en segon lloc, menús, geometries, events, animacions, efectes, etc.

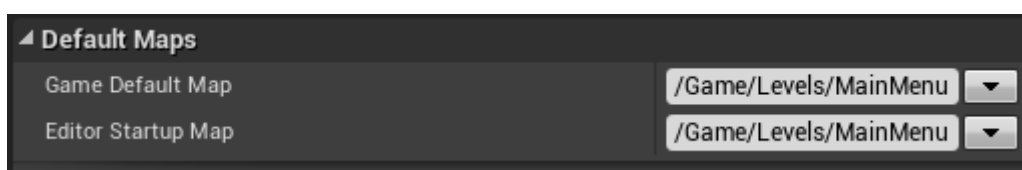


Figura 3. Assignem el Level MainMenu com a Level inicial que obrirà sempre l'aplicació

3.1.2. Actor

Un Actor és una classe que implementa qualsevol objecte que es pot posicionar en un Level. Suporta transformacions 3D com translacions, rotacions i escalats. Poden crear-se i destruir-se en temps d'execució. Existeixen diferents tipus d'Actors com el StaticMeshActor, CameraActor, PlayerStartActor, etc.

Els Actors disposen d'un Root Component que és un model geomètric mitjançant el qual es representa gràficament en un Level i que guarda les dades de les transformacions 3D que se li apliquin.

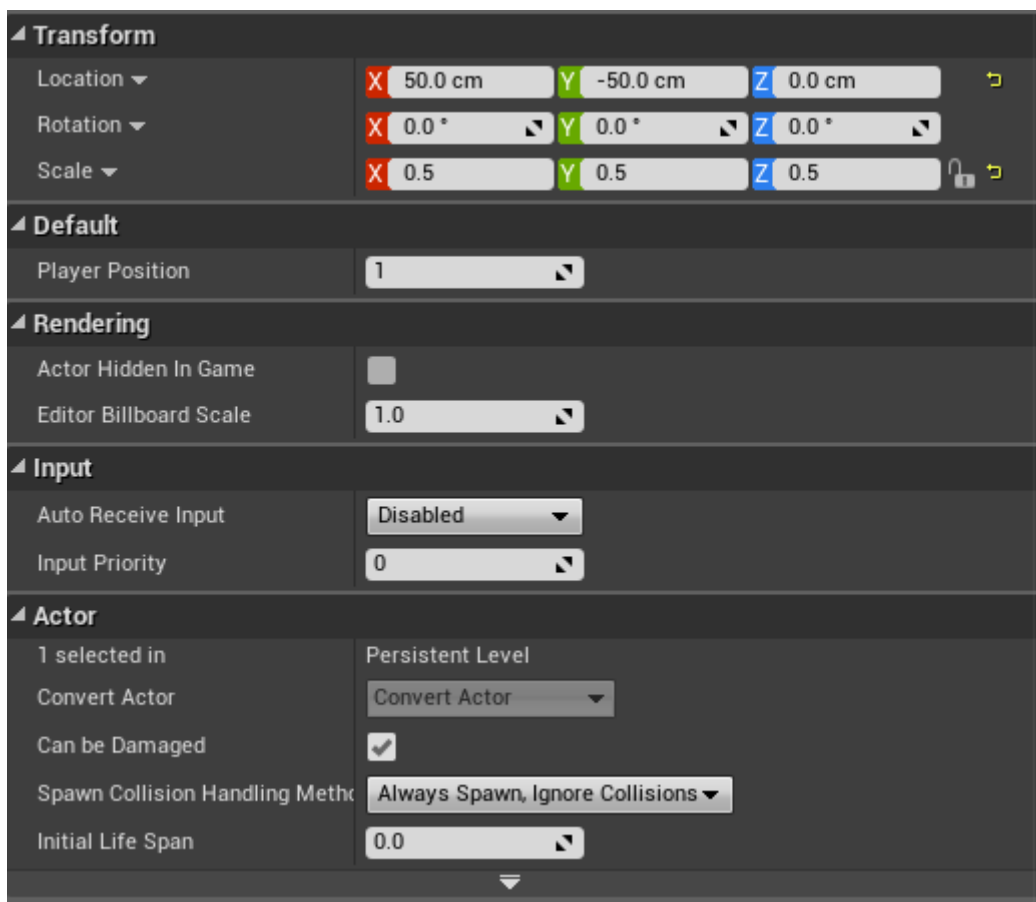


Figura 4. Propietats d'un objecte de tipus Actor

Com es mostra a la figura 4 un Actor disposa d'un component de tipus Transform que conté les dades de les transformacions 3D que se li apliquen i una sèrie de configuracions.

3.1.3. Camera

Una Camera és un objecte que hereta d'Actor i representa el punt de vista que té el jugador de l'escena, és a dir, de quina manera veu el jugador el món. Està composta pel CameraActor que és la classe pare i el CameraComponent que conté les dades, les propietats i el comportament assignat a la càmera (vegeu figura 5) del Level.

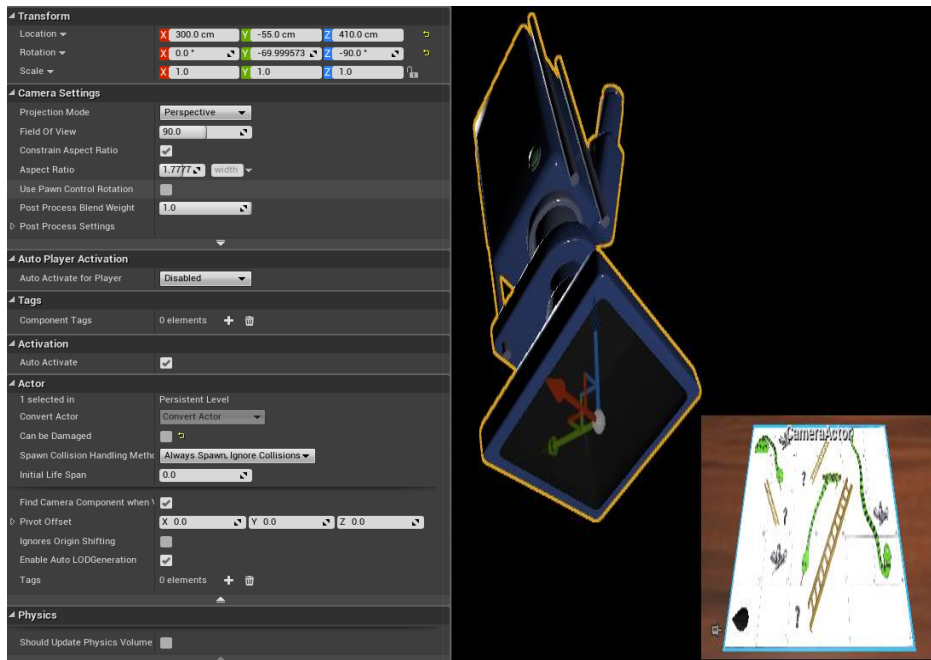


Figura 5. Propietats d'un objecte de tipus Camera

Únicament es pot tenir una Camera activa alhora en un Level, però es poden tenir tantes càmeres com es vulgui desactivades i activar-ne qualsevol en tot moment, sempre i quan la resta es desactivin.

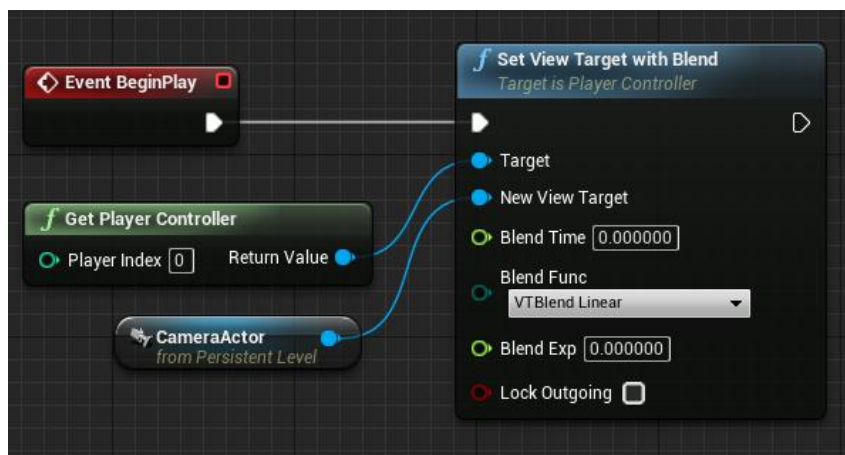


Figura 6. Codi Blueprint d'activació d'una càmera al carregar un Level

3.1.4. Lights

Les Lights són un tipus d'Actor que simula l'efecte de llum en una escena. Hi ha 4 tipus de Lights que podem utilitzar en un Level amb Unreal Engine. Aquests tipus són els següents: Directional Light, Point Light, Spot Light i Sky Light.

En aquest cas solament s'ha utilitzat el tipus de llum Directional Light perquè no apareixen escenes dinàmiques i només es requereix una llum constant a tota l'escena. Les Directional Light emeten una llum constant que es propaga fins l'infinit. Si es vol visualitzar l'escena de qualsevol Level on hi hagi una sèrie d'Actors i una Camera es necessitarà com a mínim una llum activa.

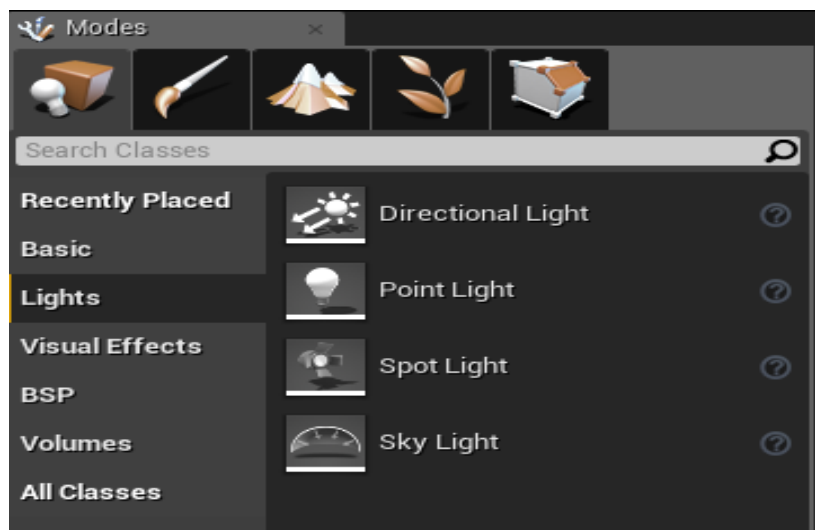


Figura 7. Panell de l'editor d'un Level per afegir Lights

3.1.5. AmbientSound

Un AmbientSound és un tipus d'Actor que es col·loca a l'escena i reproduïx la pista d'àudio que se li hagi assignat. Es pot configurar perquè l'àudio presenti propietats d'atenuació en funció de la distància de la posició del personatge del jugador, és a dir, a mida que s'allunyi de la font d'àudio se sentirà la pista més fluixa i al contrari quan s'hi approximi. També es pot configurar perquè l'àudio sigui constant a tota l'escena, independentment de la distància a la que s'estigui de la font. A més a més, es pot configurar perquè la pista es reproduïxi una única vegada o en mode bucle.

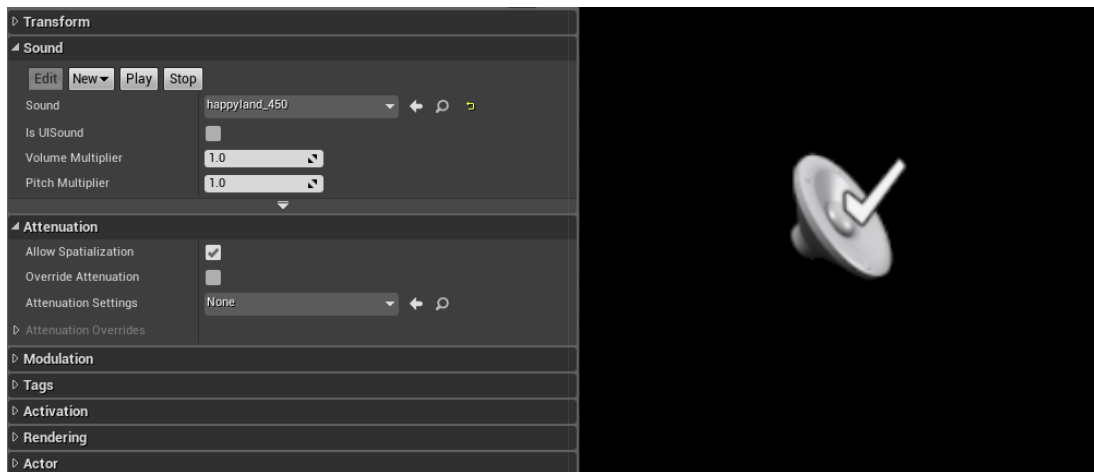


Figura 8. Propietats d'un objecte Ambient Sound

3.1.7. Widget Blueprint

Els Widget Blueprints són unes classes que aporten les funcionalitats necessàries per poder representar una interfície gràfica mitjançant elements com ara botons, menús, canvassos, imatges, etc. Aquests elements es pinten al viewport de la pantalla i presenten una sèrie de comportaments en funció d'una sèrie d'inputs com pot ser prémer un botó.

L'editor disposa de dos modes d'edició: el gràfic per poder afegir els elements visuals necessaris i el funcional on es poden programar les funcions, events i gestionar els inputs rebuts sobre els seus elements.

3.1.8. SaveGameObject

Unreal Engine proveeix una classe que implementa un objecte persistent que es guarda al directori de la instal·lació de l'aplicació amb un identificador únic.

A aquest SaveGameObject se li poden definir una sèrie d'estructures de dades on guardarem les dades i ens serveix com una base de dades petita. L'avantatge que proporciona vers una base de dades és que, en jocs on la informació que es vol guardar és mínima, com per exemple la millor puntuació del jugador en una partida o el nom del jugador, no ens cal utilitzar plugins d'accés i gestió de la base de dades a l'aplicació. A més, podem tenir tants SaveGameObject com vulguem amb identificadors únics i guardar la informació que es consideri pertinent.

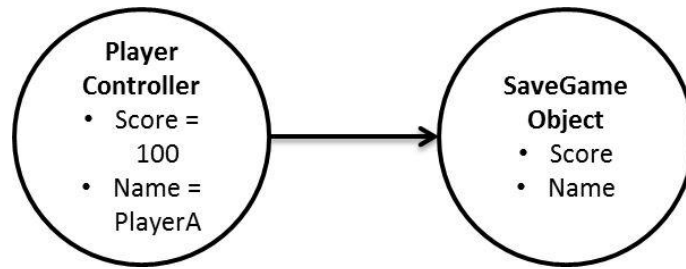


Figura 9. Esquema de l'ús d'un SaveGameObject

3.1.9. DataTable i Structure

El DataTable és una classe que conté un tipus d'estructura de dades que permet importar un arxiu .csv i guardar-lo com un objecte de tipus DataTable.

| | id | id_idioma | id_text | text |
|----|----|-----------|---------|---------------------------------------|
| 1 | 1 | 0 | 0 | Actividad Física Quiz |
| 2 | 2 | 1 | 0 | Activitat Física Quiz |
| 3 | 3 | 2 | 0 | Physical Activity Quiz |
| 4 | 4 | 0 | 1 | La Oca del Alcohol |
| 5 | 5 | 1 | 1 | L'Oca de l'Alcohol |
| 6 | 6 | 2 | 1 | Alcohol Goose Game |
| 7 | 7 | 0 | 2 | Farma Trivial |
| 8 | 8 | 1 | 2 | Farma Trivial |
| 9 | 9 | 2 | 2 | Pharma Trivial |
| 10 | 10 | 0 | 3 | Escaleras y Serpientes de las vacunas |
| 11 | 11 | 1 | 3 | Escales i Serps de les vacunes |
| 12 | 12 | 2 | 3 | Snakes and Ladders of Vaccines |
| 13 | 13 | 0 | 4 | La Oca de los Piojos |
| 14 | 14 | 1 | 4 | L'Oca dels Polls |
| 15 | 15 | 2 | 4 | Lice Goose Game |

Figura 10. A l'esquerra, la definició de l'Structure, a la dreta, la DataTable amb el contingut d'un fitxer .csv aplicant l'estructura de columnes de l'Structure de la dreta

L'Structure és un esquema de les columnes i del tipus de dades que conté cadascuna d'aquestes columnes. Per poder crear un DataTable, és necessari assignar-li un tipus de Structure i, posteriorment, importar el .csv amb les dades per emplenar el DataTable.

3.2. Interfície gràfica i menús

En aquest apartat es detallen tots els elements de software de l'aplicació que representen el sistema de navegabilitat entre les pantalles del joc, els menús i la interfície gràfica.

3.2.1. Menú Principal

El menú principal (vegeu *figura 11*) es mostra en la primera pantalla de l'aplicació i representa la pantalla base de la interfície gràfica.

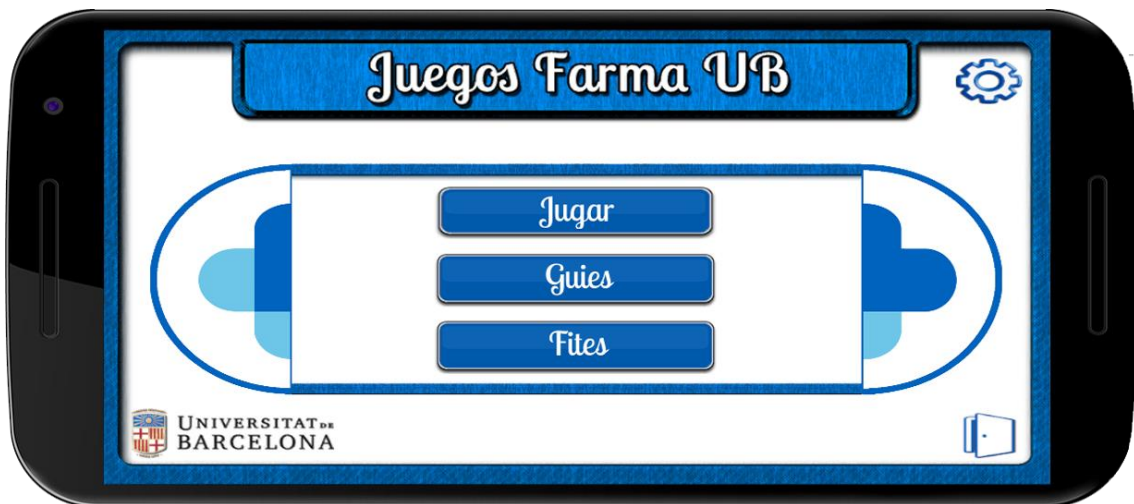


Figura 11. Pantalla del menú principal

Aquest menú principal s'implementa a la classe BP_MainMenu que hereta de la classe QWidget Blueprint. En obrir el Level MainMenu, es crea un objecte de tipus BP_MainMenu i es fixa al viewport perquè mostri el menú principal.

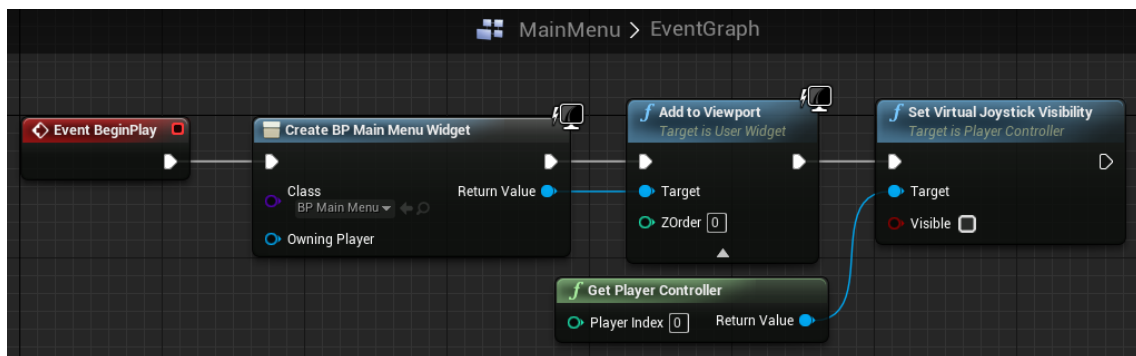


Figura 12. En iniciar el Level MainMenu, es crea i es mostra el menú principal

En obrir aquesta pantalla, s'executa una animació on la creu central gira i s'expandeix per mostrar els botons del menú.

3.2.2. Menú Opcions

El menú d'opcions (vegeu *figura 12*) es mostra en prémer el botó de la roda dentada del menú principal. Aquest menú es mostra sobreposat al menú principal ocupant la meitat de la pantalla i amb un fons transparent que cobreix la resta.

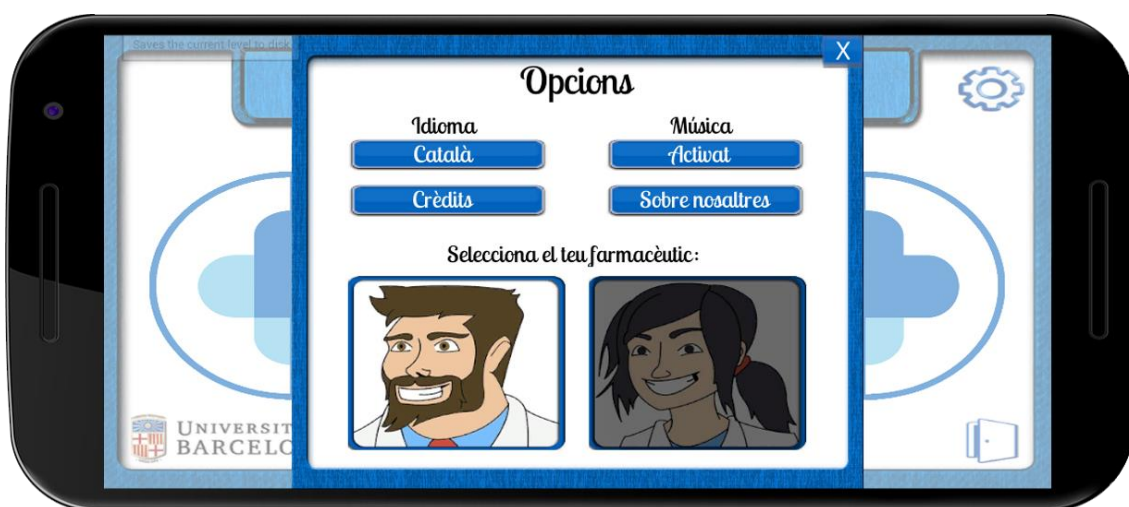


Figura 12. Pantalla d'opcions sobre el menú principal

Aquest menú s'implementa a la classe BP_Options_Panel que hereta de la classe QWidget Blueprint. Les funcionalitats que ofereix són les següents:

1. **Idioma:** permet canviar l'idioma de l'aplicació. Disposa dels idiomes català i castellà, tot i que està implementat per detectar més idiomes si es tenen les traduccions de tots els textos.
2. **Música:** permet activar o desactivar la música de fons del menú principal.
3. **Crèdits:** mostra la pantalla amb els crèdits del desenvolupament d'aquesta aplicació (vegeu l'*apartat 3.2.3*).
4. **Sobre nosaltres:** mostra la pantalla amb una breu explicació del projecte (vegeu l'*apartat 3.2.4*).

- 5. Selecciona el teu farmacèutic:** permet canviar el farmacèutic que anirà guiant i interactuant amb el jugador durant les partides.

Qualsevol canvi que es realitzi en les opcions d'aquest menú es guarda automàticament en el GameDataObject de l'aplicació per tal de conservar els canvis després de tancar l'aplicació.

3.2.3. Pantalla de crèdits

La pantalla de crèdits (vegeu *figura 13*) mostra els col·laboradors del projecte. Aquesta pantalla s'implementa a la classe BP_Options_Panel com un component de tipus Canvas Panel que es mostra a sobre d'aquest.

Està formada per una sèrie de components de tipus Text Block que contenen el text amb la informació i d'un botó que tanca aquesta pantalla i retorna el control a la pantalla d'opcions.

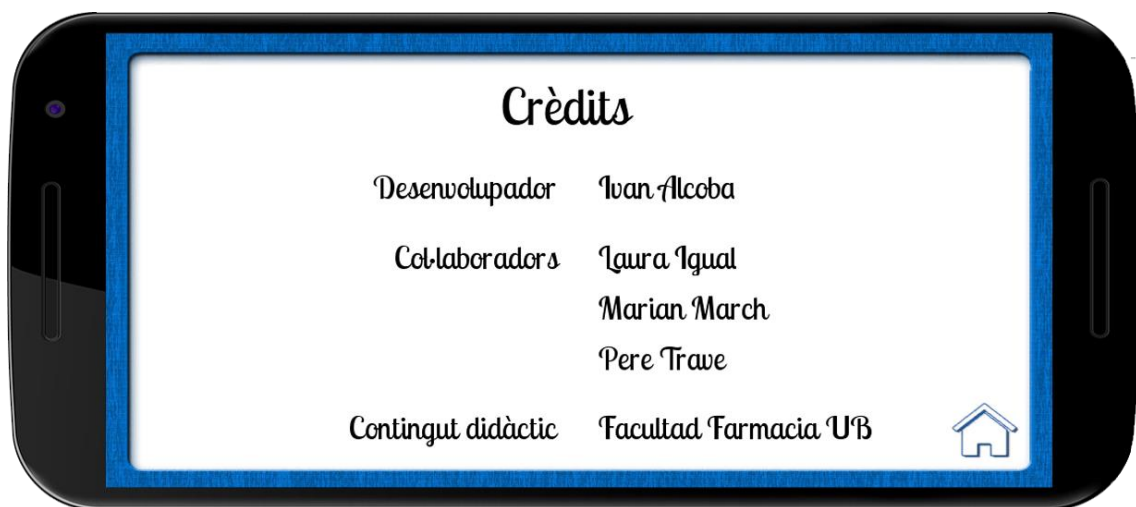


Figura 13. Pantalla de crèdits

3.2.4. Pantalla de sobre nosaltres

La pantalla «Sobre nosaltres» (vegeu *figura 14*) mostra un text que descriu l'aplicació i el projecte d'aquest TFG. Aquesta pantalla s'implementa a la classe BP_Options_Panel com un component de tipus Canvas Panel que es mostra a sobre d'aquest.

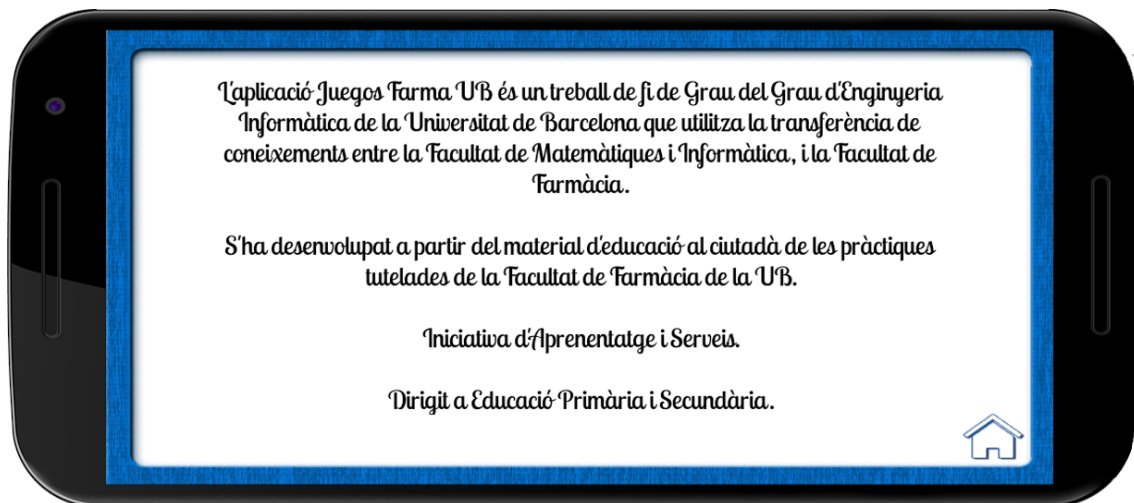


Figura 14. Pantalla «Sobre nosaltres»

3.2.5. Pantalla de selecció de jocs

La pantalla de selecció de jocs (vegeu *figura 15*) permet escollir a quin dels cinc jocs de l'aplicació es vol jugar. Aquesta pantalla s'implementa a la classe BP_GamesSelection que hereta de la classe QWidget Blueprint. Presenta dos botons amb la icona d'una fletxa per passar d'un joc a l'altre, un botó amb la icona d'una casa per tornar al menú principal i els botons amb les portades dels jocs per començar una nova partida.

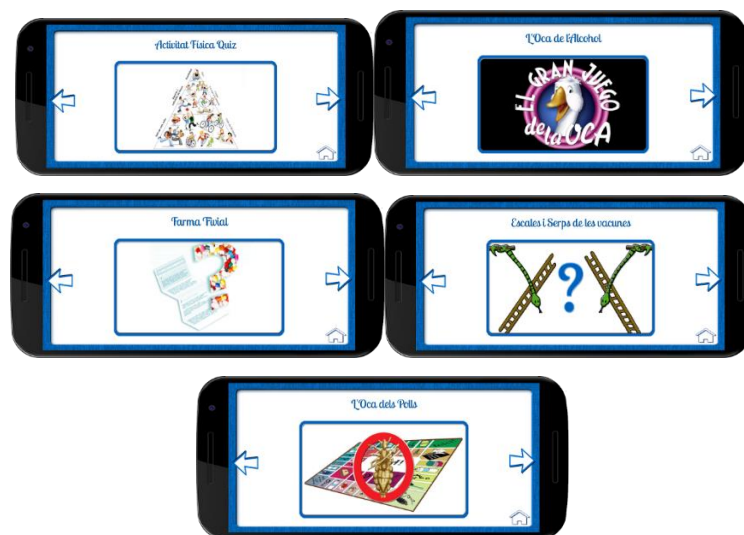


Figura 15. Pantalla de selecció de jocs. Es mostren les 5 transicions amb la selecció de cadascun dels jocs

3.2.6. Pantalla de guies

La pantalla de guies (vegeu *figura 16*) permet accedir a tres de les guies que conté l'aplicació. Aquesta pantalla s'implementa a la classe BP_GuidesSelection que hereta de la classe QWidget Blueprint.

El primer botó mostra la pantalla de la guia d'Activitat Física (*figura 17*), el segon la pantalla de la guia del Consum d'Alcohol (*figura 18*) i el tercer la pantalla de la guia de les Vacunes (*figura 19*). Per últim, el botó amb la icona d'una casa ens retorna al menú principal.

Dins de les pantalles de les guies hi ha dos botons que permeten navegar a la pàgina anterior o a la posterior, respectivament.



Figura 16. Pantalla de selecció de guies

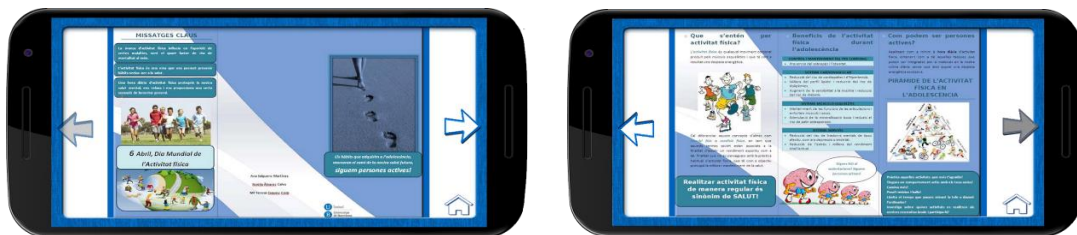


Figura 17. Pantalles de la guia d'Activitat Física

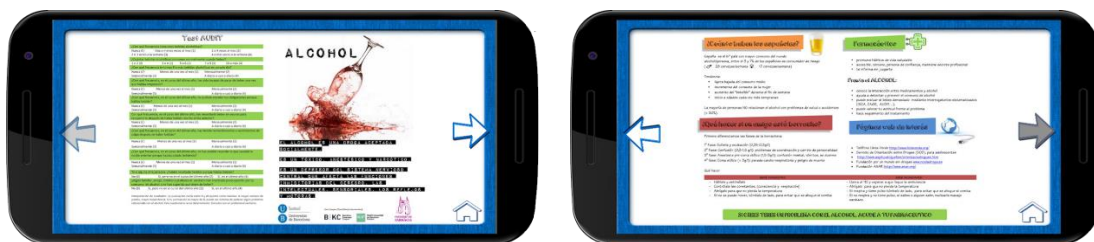


Figura 18. Pantalles de la guia del Consum d'Alcohol

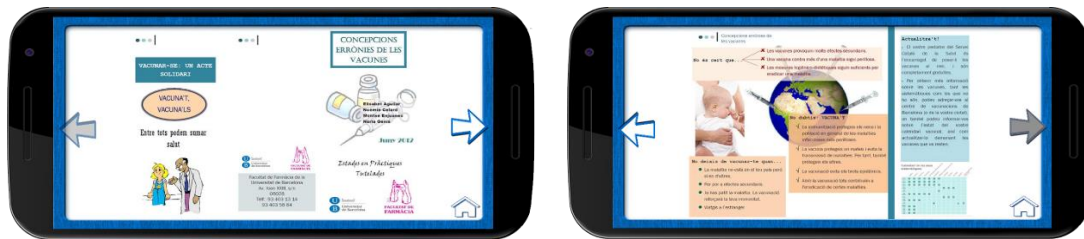


Figura 19. Pantalles de la guia de les Vacunes

3.2.7. Pantalla de fites

La pantalla de fites (vegeu figura 20) mostra les fites aconseguides en els diferents jocs de l'aplicació. Aquesta pantalla s'implementa a la classe BP_Logros que hereta de la classe QWidget Blueprint.

Està formada per un botó amb la icona d'una casa per tornar al menú principal i d'un component Scroll Box amb els textos de les fites. Cadascuna d'aquestes fites es marca amb tres estrelles grogues o grises, en funció de si s'han aconseguït o no. A més a més, estan organitzades per jocs, ja que cada joc té les seves fites pròpies.



Figura 20. Pantalla de les fites aconseguïdes pel jugador

3.3. Joc: Activitat Física Quiz

El joc Activitat Física Quiz consisteix en un joc de tipus qüestionari amb un total de preguntes per obtenir el perfil d'activitat física en funció dels hàbits de l'usuari.



Figura 21. Una pregunta durant una partida del joc Activitat Física Quiz

3.3.1. Level

El Level on s'executa el joc de l'Activitat Física Quiz és el AFQuiz_Game. En el Level Blueprint es crea un objecte de tipus BP_AFQuiz_Game que implementa la interfície i la lògica del joc (vegeu figura 22).

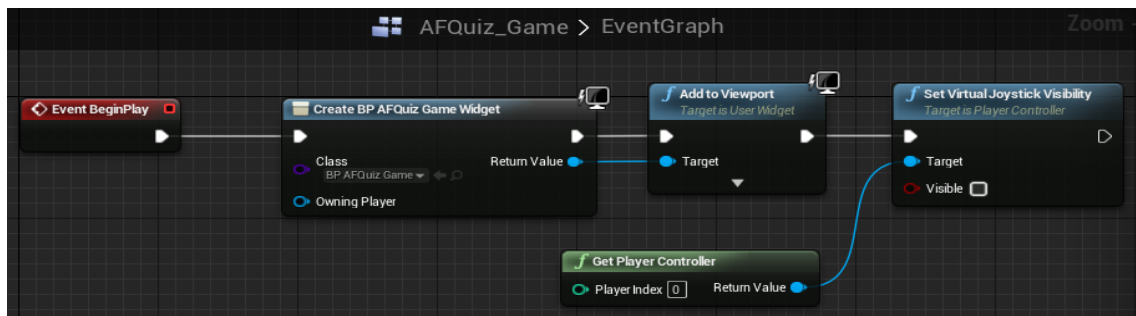


Figura 22. AFQuiz_Game Level Blueprint

3.3.2. GameDataObject

En primer lloc carreguem el GameDataObject (vegeu figura 23) que conté les dades de configuració de l'aplicació. En aquest cas, ens interessa l'idioma, el farmacèutic i l'estat de les fites relacionades amb aquest joc.

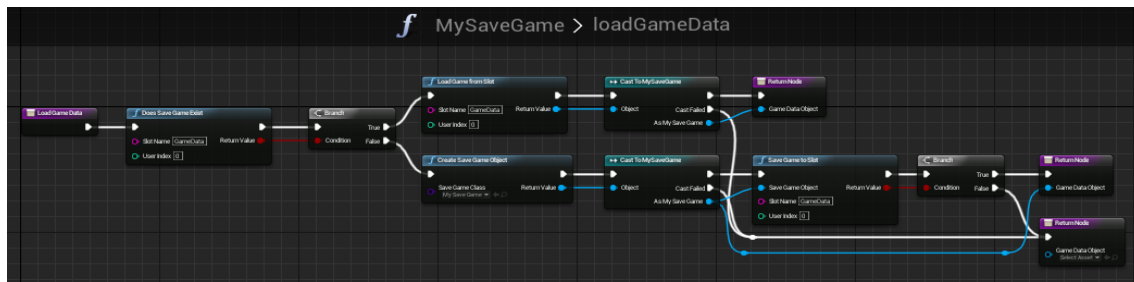


Figura 23. Càrrega del GameDataObject on guardem les dades del joc

3.3.3. DataTable & Structure

Els textos en multi idioma de les preguntes i de la interfície del joc Activitat Física Quiz estan organitzats a les DataTables AFQuiz_Preguntes_T i AFQuiz_Text_T, respectivament. A partir d'aquestes DataTables, assignem els textos dels Text Blocks de la interfície en funció de l'idioma que obtenim del GameDataObject.

3.3.4. Animations

Un cop assignats els textos en l'idioma corresponent, executem la animació d'introducció al joc en funció del farmacèutic que tinguem marcat a les opcions.

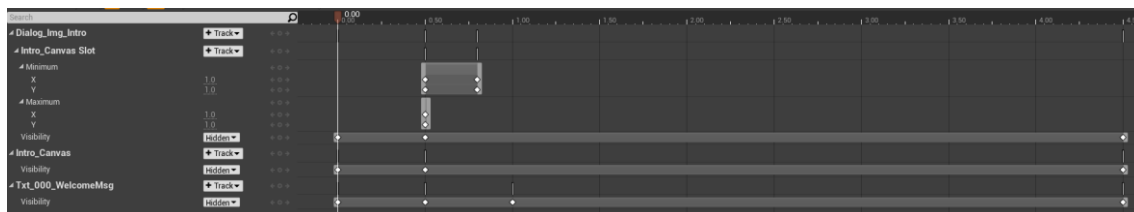


Figura 24. Intro_Anim. Animation que mostra el farmacèutic amb un diàleg donant la benvinguda al joc

Aquestes Animations es creen a partir d'una línia del temps, assignant fotogrames claus a instants de temps concrets on es modifiquen les propietats dels components del Widget que intervenen en l'animació.

En aquest joc disposem d'una animació introductòria on s'informa al jugador de que tracta el joc (vegeu figura 25) i d'una animació en bucle que reproduïx l'efecte de parpelleig del farmacèutic durant tota la partida per donar-li major realisme.

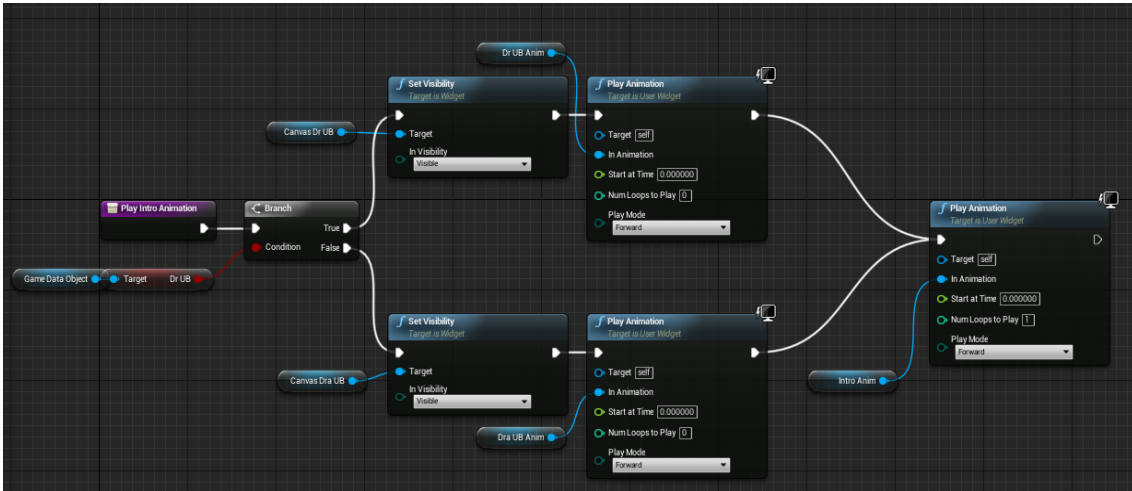


Figura 25. Funció PlayIntroAnimation() que inicia l'execució de l'animació assignant el farmacèutic corresponent

3.3.5. Preguntes

Després de l'animació, executem el codi que va mostrant les preguntes (vegeu figura 21) a mesura que l'usuari les contesta, fins un total d'onze preguntes. Cadascuna d'aquestes preguntes, solament pot ser contestada de forma afirmativa o negativa. En funció de la pregunta i la resposta, es sumen o es resten punts.

3.3.6. Resultats

Un cop contestades totes les preguntes, obtenim una puntuació total, si no superem el primer llindar obtenim la qualificació més baixa de 'Inactiu', si el superem n'obtenim una intermitja de 'Poc Actiu' i si superem el segon llindar obtenim la més alta de 'Moderadament Actiu'. Quan s'obté per primera vegada la qualificació de 'Moderadament Actiu' es guanya la fita del joc i es guarda al GameDataObject.



Figura 26. Resultat d'una partida del joc Activitat Física Quiz

3.4. Joc: Oca de l'Alcohol

El joc de l'Oca de l'Alcohol és un joc de taula educatiu sobre el consum d'alcohol on es mou una fitxa mitjançant un dau per intentar arribar des de la casella inicial a la final superant una sèrie d'obstacles i proves, abans de que ho aconsegueixi el rival.



Figura 27. Partida de l'Oca de l'Alcohol

3.4.1. Level

El Level on s'executa el joc de l'Oca de l'Alcohol és el OcaAlcohol_Game. En el Level Blueprint es crea un objecte de tipus BP_OcaAlcohol_Game que hereta de QWidget Blueprint i que implementa la interfície i la lògica del joc. Al Level Blueprint també s'ajusta l'Aspect Ratio de la càmera (vegeu figura 28) per a que s'ajusti a la mida de qualsevol dispositiu mòbil.

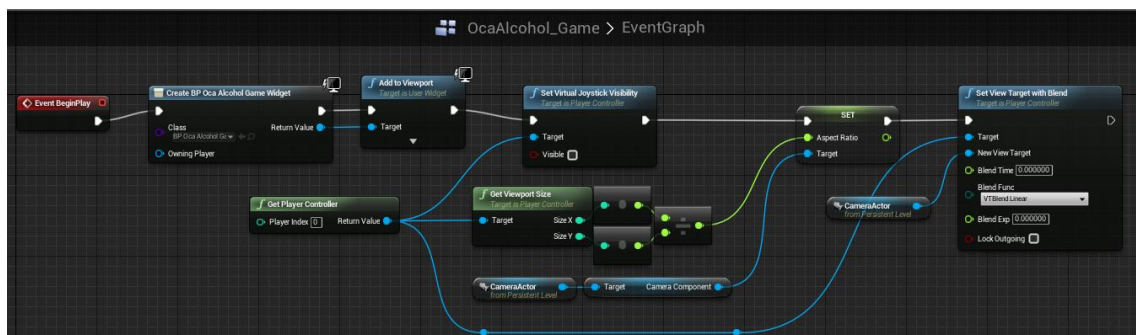


Figura 28. OcaAlcohol_Game Level Blueprint. Es crea el QWidget amb la interfície del joc i s'ajusta l'aspect ratio de la càmera en funció de la pantalla del dispositiu.

En aquest joc, a més a més, d'utilitzar els QWidgets per la interfície, també s'utilitzen Actors amb geometries 3D que representen el taulell, les caselles i les fitxes. Així com, també, es disposa d'una càmera i llums.

3.4.2. GameDataObject

Com en la resta de jocs, el primer que fem es carregar les dades del joc (vegeu *figura 23*) per assignar els textos en funció de l'idioma i poder gestionar les fites aconseguides durant les partides.

3.4.3. Actors

En aquest Level s'utilitzen tres Actors. El primer el TableroCasillas (taulell i caselles) que és un Actor amb una component StaticMeshActor que representa una geometria 3D no mòbil en la escena. Els segon i tercer són el Ficha i FichaAI Actors (vegeu *figura 29*) que són del tipus Actor Blueprint i són movibles per l'escena.

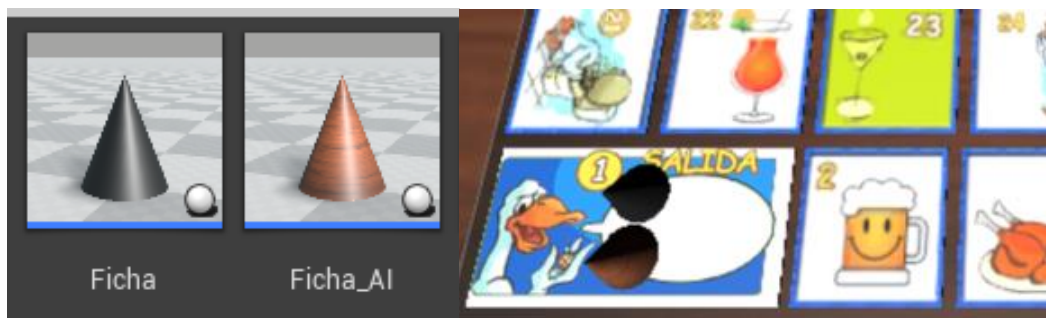


Figura 29. Ficha i FichaAI Actors Blueprints.

3.4.4. Animations

L'animació de moviment de les fitxes és realitza mitjançant un Custom Event amb un Timer By Event que realitza l'animació d'avanç de casella en casella en funció del valor del dau obtingut en el torn del jugador o de la IA.

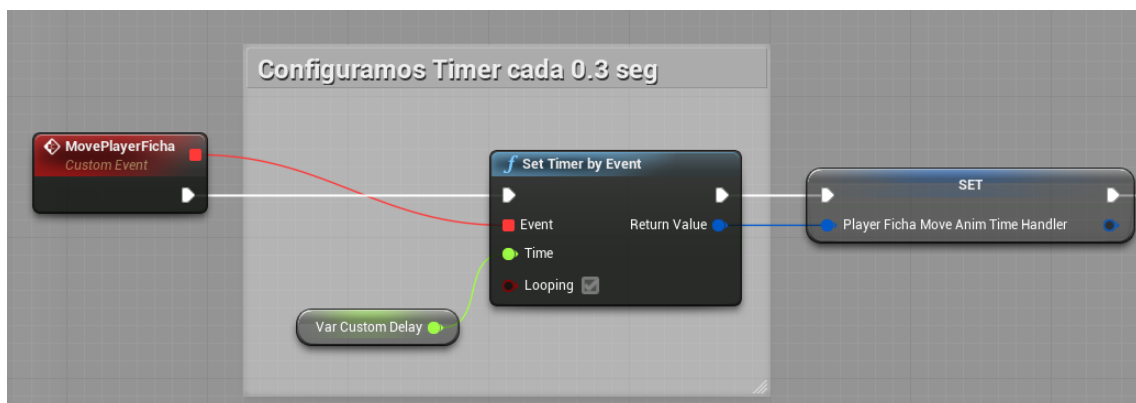


Figura 30. MovePlayerFicha Event. Custom Event que realitza l'animació de moviment de la fitxa pel taulell.

3.4.5. Lògica i resultats

Una partida de l'Oca de l'Alcohol sempre comença amb el jugador tirant el primer torn. En funció de la tirada, avança el nombre de caselles amb l'animació detallada a l'apartat 3.4.4. Un cop arriba a la casella final del torn, s'avalua si és una casella especial. Si ho és, es mostra el missatge de la casella en qüestió (vegeu *figura 31*) i es realitza l'acció que provoca haver caigut en aquella casella com pot ser tornar a la casella de sortida, retrocedir d'oca en oca, quedar bloquejat durant un torn, etc.



Figura 31. Missatge que es mostra al caure en una casella d'oca.

Un cop arribi un jugador a la casella final es para el joc i es mostra el resultat, guanya el jugador o perd, mostrant un missatge, un botó per sortir del joc i un altre per tornar a jugar. En cas que guanyi el jugador, s'incrementa el nombre de partides guanyades en una unitat i es guarda al GameDataObject per l'enregistrament de les fites.

3.5. Joc: Farma Trivial

El joc Farma Trivial és un joc amb diferents preguntes de diverses temàtiques educatives per a la salut que té com a objectiu intentar encertar el màxim de les preguntes que es plantegen.

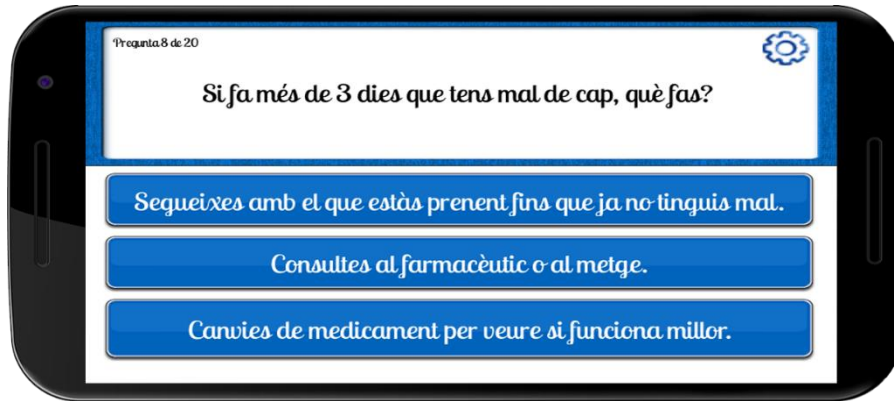


Figura 32. Una pregunta durant una partida a 20 preguntes del joc Farma Trivial

3.5.1. Level

El Level on s'executa el joc del Farma Trivial és PharmaTrivial_Game. En el Level Blueprint es crea un objecte de tipus BP_PharmaTrivial que hereta de QWidget Blueprint i implementa la lògica i interfície del joc.

3.5.2. GameDataObject

Com en la resta de jocs, el primer que fem es carregar les dades del joc (vegeu figura 23) per assignar els textos en funció de l'idioma i poder gestionar les fites aconseguides durant les partides.

3.5.3. Lògica & Preguntes

Quan es juga una nova partida, en primer lloc s'ha de seleccionar el tipus de partida que es vol jugar. Una partida de 10 o de 20 preguntes. Un cop seleccionat aquest paràmetre es mostra la primera pregunta amb les possibles opcions de la resposta (vegeu figura 32). Les respostes poden ser de 2 o 3 opcions. En prémer alguna de les respostes

es procedeix a informar de la correctesa de la resposta i es passa a la següent pregunta fins que es realitzen el número de preguntes seleccionades al començar la partida.

3.5.4. Resultats

Un cop contestades totes les preguntes, es mostra la puntuació obtinguda (vegeu *figura 33*) i s'enregistren les respostes contestades correctament per les fites del joc.



Figura 33. Resultat d'una partida de 20 preguntes.

3.6. Joc: Escales i Serps de les Vacunes

El joc de les Escales i Serps de les Vacunes és un joc de taula educatiu sobre les concepcions errònies de les vacunes en què es mou una fitxa mitjançant un dau per intentar arribar des de la casella inicial a la final superant una sèrie d'obstacles i proves abans de quedar-se sense vides.

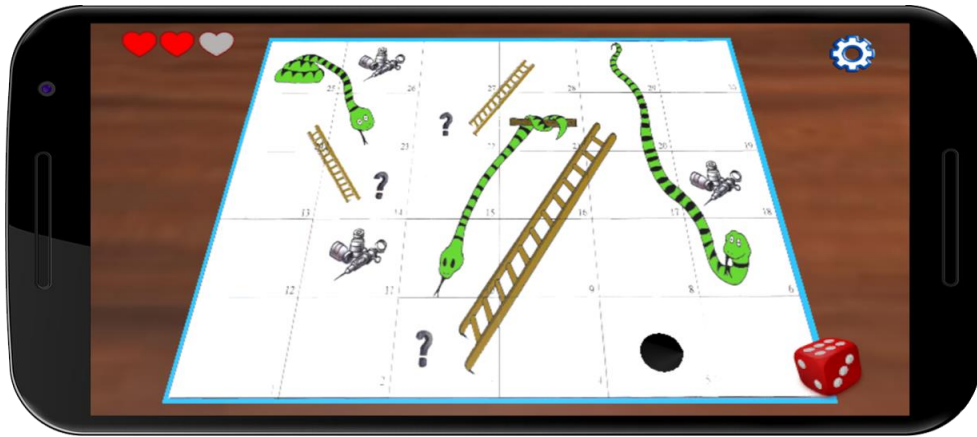


Figura 34. Partida del Joc Escales i Serps de les Vacunes

3.6.1. Level

El Level on s'executa el joc de les Escales i Serps de les Vacunes és el `EscaSerps_Game`. Al Level Blueprint s'ajusta l'Aspect Ratio de la càmera per a que s'ajusti a la mida de qualsevol dispositiu mòbil.

3.6.2. GameDataObject

Com en la resta de jocs, el primer que fem es carregar les dades del joc (vegeu *figura 23*) per assignar els textos en funció de l'idioma i poder gestionar les fites aconseguides durant les partides.

3.6.3. Actors

En aquest Level s'utilitzen quatre Actors. El primer el `EscSerps_Tablero` (taulell i caselles) que és un Actor amb una component `StaticMeshActor` que representa una geometria 3D no mòbil en la escena. Els segon és el `Ficha Actor` (vegeu *figura*) que és del tipus `Actor Blueprint` i és pot moure per l'escena. El tercer és el `EscSerps_Engine` que és un Actor sense geometria que implementa la lògica del joc i s'encarrega de gestionar la interfície del joc. Per últim, tenim el `GUI_EscSerps_Game` que conté la interfície i els mètodes per mostrar les preguntes, missatges i resultats de la partida.

3.6.4. EscSerps_Engine

Aquest Actor implementa la lògica del joc i gestiona la interfície a partir d'un objecte de tipus GUI_EscSerps_Game. S'encarrega d'enviar peticions a la interfície per mostrar els missatges de casella, les preguntes, així com els resultats. També s'encarrega de gestionar el moviment de la fitxa i les posicions del taulell.

3.6.5. Lògica i resultats

Quan es juga una nova partida, primer es mostra un missatge explicatiu del joc. Tot seguit es visualitza el taulell amb la interfície. Quan es prem el botó del dau el EscSerps_Engine s'encarrega de generar una tirada i enviar a la GUI la petició perquè mostri el missatge amb el valor de la tirada. Tot seguit, executa l'animació de moviment de la fitxa pel taulell.

Un cop finalitzada l'animació de moviment, es gestiona la casella final on es troba la fitxa i es formula una pregunta. En cas de ser una casella especial i contestar correctament es guanyen una sèrie d'avantatges com una vida extra, un avanç notable de casella, etc. En cas de no ser una casella especial i contestar incorrectament es perd una vida. Quan et quedes sense vides es perd la partida. Tot i que si s'arriba a la casella final es guanya la partida i es mostren els resultats. Tot seguit es guarda el nombre de partides guanyades al GameDataObject per enregistrar el seguiment de les fites.

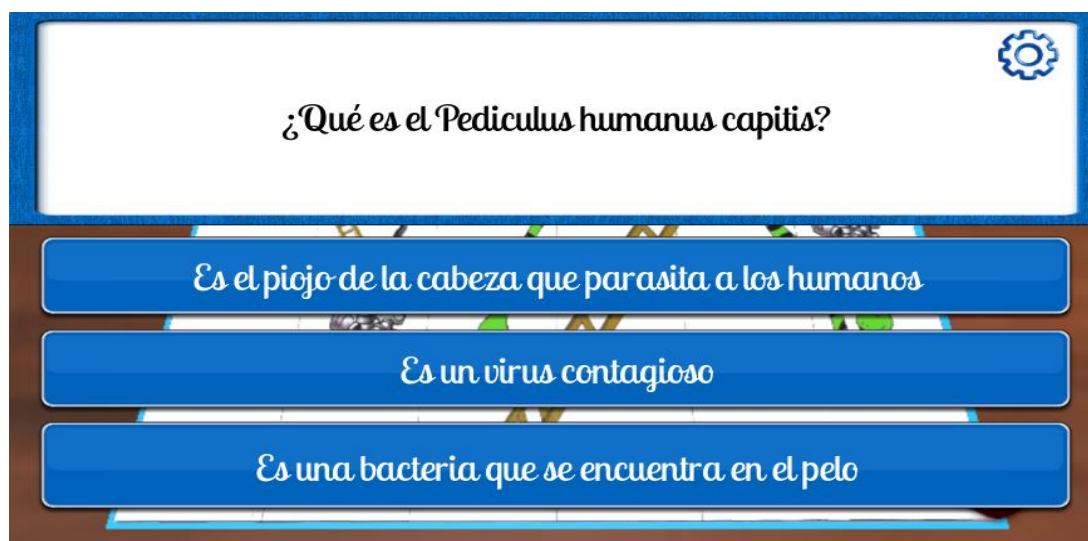


Figura 35. Pregunta d'una casella

3.7. Joc: Oca dels Polls

El joc de l'Oca dels Polls és un joc de taula educatiu sobre el contagi i el tractament dels polls, on es mou una fitxa mitjançant un dau per intentar arribar des de la casella inicial a la final superant una sèrie d'obstacles i proves abans de quedar-se sense vides.



Figura 36. Joc de l'Oca dels Polls

3.7.1. Level

El Level on s'executa el joc de l'Oca dels Polls és el OcaPiojos_Game. Al Level Blueprint s'ajusta l'Aspect Ratio de la càmera per a que s'ajusti a la mida de qualsevol dispositiu mòbil.

3.7.2. GameDataObject

Com en la resta de jocs, el primer que fem es carregar les dades del joc (vegeu figura 23) per assignar els textos en funció de l'idioma i poder gestionar les fites aconseguides durant les partides.

3.7.3. Actors

En aquest Level s'utilitzen quatre Actors. El primer el OcaPiojos_Tablero (taulell i caselles) que és un Actor amb una component StaticMeshActor que representa una geometria 3D no mòbil en la escena. Els segon és el Ficha Actor (vegeu figura) que és

del tipus Actor Blueprint i és pot moure per l'escena. El tercer és el OcaPiojos_Engine que és un Actor sense geometria que implementa la lògica del joc i s'encarrega de gestionar la interfície del joc. Per últim, tenim el GUI_OcaPiojos_Game que conté la interfície i els mètodes per mostrar les preguntes, missatges i resultats de la partida.

3.7.4. Lògica i resultats

Quan es juga una nova partida, primer es mostra un missatge explicatiu del joc. Tot seguit es visualitza el taulell amb la interfície. Quan es prem el botó del dau el OcaPiojos_Engine s'encarrega de generar una tirada i enviar a la GUI la petició perquè mostri el missatge amb el valor de la tirada. Tot seguit, executa l'animació de moviment de la fitxa pel taulell.

Un cop finalitzada l'animació de moviment, es gestiona la casella final on es troba la fitxa i es formula una pregunta. En cas de ser una casella especial i contestar correctament es guanyen una sèrie d'avantatges com una vida extra, un avanç notable de casella, etc. En cas de no ser una casella especial i contestar incorrectament es perd una vida. Quan et quedes sense vides es perd la partida. Tot i que si s'arriba a la casella final es guanya la partida i es mostren els resultats. Tot seguit es guarda el nombre de partides guanyades al GameDataObject per enregistrar el seguiment de les fites.

4. Proves i resultats

En aquest apartat es detallen les proves realitzades a l'aplicació obtinguda en el desenvolupament d'aquest projecte i els resultats obtinguts.

4.1. Testing d'interfície gràfica i menús

En el conjunt de pantalles que conformen la interfície gràfica i els menús (descrits a l'apartat 3.2) de l'aplicació s'han realitzat les proves i analitzat els resultats que es recullen a la taula següent:

| Prova | Resultat |
|--|-----------------|
| Correcta navegació entre pantalles utilitzant els botons implementats | Correcte |
| Els textos apareixen en l'idioma seleccionat | Correcte |
| Activació/desactivació de la música de fons | Correcte |
| Les fites aconseguides es registren i es mostren correctament | Correcte |
| Les opcions es guarden correctament al tancar l'aplicació | Correcte |
| En minimitzar l'aplicació i tornar-la a obrir, es mostra la pantalla on estàvem de forma correcta. | Correcte |
| L'animació del menú principal s'executa correctament | Correcte |

4.2. Testing Activitat Física Quiz

En el joc de l'Activitat Física Quiz s'han realitzat les proves i analitzat els resultats que es recullen a la taula següent:

| Prova | Resultat |
|--|-----------------|
| Acabar una partida i obtenir una valoració del perfil físic | Correcte |
| Obtenció de la fita «Valoració de moderadament actiu» en completar una partida i obtenir aquesta valoració final | Correcte |
| El farmacèutic que pregunta és el que s'ha seleccionat a les opcions | Correcte |
| Les preguntes apareixen en l'idioma seleccionat | Correcte |
| Els textos de la interfície apareixen en l'idioma seleccionat | Correcte |

4.3. Testing Oca de l'Alcohol

En el joc de l'Oca de l'Alcohol s'han realitzat les proves i analitzat els resultats que es recullen a la taula següent:

| Prova | Resultat |
|---|-----------------|
| Guanyar una partida i obtenir la fita «Guanya 1 partida» | Correcte |
| Perdre una partida per comprovar els missatges de resultat | Correcte |
| Es mostren els missatges a les caselles especials | Correcte |
| Es realitzen les penalitzacions/bonificacions de les caselles especials | Correcte |
| Durant el transcurs de la partida no es bloqueja | Correcte |
| Moviment de les fitxes correcte en funció de la tirada obtinguda | Correcte |
| Els textos de la interfície apareixen en l'idioma seleccionat | Correcte |

4.4. Testing Farma Trivial

En el joc del Farma Trivial s'han realitzat les proves i analitzat els resultats que es recullen a la taula següent:

| Prova | Resultat |
|---|-----------------|
| Completar les fites en guanyar partides | Correcte |
| Completar la fita de preguntes encertades | Correcte |
| Comprovar que les respostes siguin les pertinents a la pregunta | Correcte |
| Comprovar que es marquin les respostes correctes/incorrectes | Correcte |
| Jugar una partida a 10 preguntes | Correcte |
| Jugar una partida a 20 preguntes | Correcte |
| Els textos de les preguntes i respostes apareixen en l'idioma seleccionat | Correcte |
| Els textos de la interfície apareixen en l'idioma seleccionat | Correcte |
| Es mostra el farmacèutic seleccionat a les opcions | Correcte |

4.5. Testing Escales i Serps de les Vacunes

En el joc de les Escales i Serps de les Vacunes s'han realitzat les proves i analitzat els resultats que es recullen a la taula següent:

| Prova | Resultat |
|---|-----------------|
| Guanyar una partida i obtenir la fita «Guanya 1 partida» | Correcte |
| Perdre una partida per comprovar els missatges de resultat | Correcte |
| Es mostren els missatges a les caselles especials | Correcte |
| Es realitzen les penalitzacions/bonificacions de les caselles especials | Correcte |
| Durant el transcurs de la partida no es bloqueja | Correcte |
| Moviment de les fitxes correcte en funció de la tirada obtinguda | Correcte |
| Els textos de la interfície apareixen en l'idioma seleccionat | Correcte |
| Comprovar que les respostes siguin les pertinents a la pregunta | Correcte |
| Comprovar que es marquin les respostes correctes/incorrectes | Correcte |

4.6. Testing Oca dels polls

En el joc de l'Oca dels Polls s'han realitzat les proves i analitzat els resultats que es recullen en la taula següent:

| Prova | Resultat |
|---|-----------------|
| Guanyar una partida i obtenir la fita «Guanya 1 partida» | Correcte |
| Perdre una partida per comprovar els missatges de resultat | Correcte |
| Es mostren els missatges a les caselles especials | Correcte |
| Es realitzen les penalitzacions/bonificacions de les caselles especials | Correcte |
| Durant el transcurs de la partida no es bloqueja | Correcte |
| Moviment de les fitxes correcte en funció de la tirada obtinguda | Correcte |
| Els textos de la interfície apareixen en l'idioma seleccionat | Correcte |
| Comprovar que les respostes siguin les pertinents a la pregunta | Correcte |
| Comprovar que es marquin les respostes correctes/incorrectes | Correcte |

5. Conclusions

Un cop finalitzat el procés de desenvolupament de l'aplicació es podria afirmar que s'han aconseguit els objectius que es pretenien assolir, ja que s'havia plantejat el projecte com el repte de crear una aplicació Android amb jocs i guies educatives mitjançant l'ús d'Unreal Engine 4 que pogués ser publicat a la Play Store de Google com una aplicació final i accessible als usuaris.

Personalment, crec que el resultat obtingut és satisfactori, ja que vaig començar aquest projecte sense haver treballat mai amb Unreal Engine, tot i que sí que ho havia fet amb altres motors de videojocs, en concret Unity3D, i durant el desenvolupament d'aquest projecte he pogut descobrir que desenvolupar una aplicació Android amb Unreal Engine és una tasca amb bastanta complexitat, fins i tot per a professionals del sector. Això es deu a què, fins fa relativament poc, Unreal Engine únicament se centrava en desenvolupar funcionalitats orientades a la creació de jocs per a PC, destacant per una qualitat molt superior als motors de la competència en aquest sector. No obstant, actualment, s'han començat a centrar en el desenvolupament per a plataformes mòbils, fet que provoca que encara hi hagi aspectes pendents de millorar, com per exemple la mida dels APK generats a partir d'un nou projecte que inclogui el contingut bàsic del motor, que gairebé arriba al límit d'espai permès per Google, la dificultat a l'hora d'utilitzar plugins en projectes per plataformes mòbils o la complexitat a l'hora de debugar l'aplicació en un dispositiu mòbil.

El procés de realització d'aquest projecte ha estat costós i han sorgit molts imprevistos pel fet de no tenir experiència prèvia, però, en aquest punt, després de tots els obstacles superats, em veig capacitat per continuar utilitzant aquesta tecnologia de manera molt més òptima i amb projectes de major complexitat.

Tot i que el contingut actual del projecte ja estigui finalitzat, la meua intenció és continuar treballant en l'aplicació per afegir nou contingut amb la col·laboració de la Facultat de Farmàcia i amb l'objectiu d'adquirir més coneixements sobre Unreal Engine, així com aconseguir desenvolupar una aplicació més completa.

5.1. Possibles ampliacions

En una aplicació de jocs, existeixen moltes possibles ampliacions o millores a realitzar. Aquest projecte com aplicació educativa té molt potencial per ser ampliat. Algunes de les ampliacions que es podrien dur a terme són les següents:

- Ampliar el conjunt de preguntes per al joc del Farma Trivial.
- Augmentar el nombre de fites per joc per aconseguir un major nombre de reptes per l'usuari.
- Afegir nous jocs amb temàtiques diferents per ampliar el rang de temes educatius per a la salut que tracta l'aplicació.
- Implementar un sistema de Data Mining a partir dels resultats obtinguts a les preguntes del Farma Trivial per analitzar els coneixements que té la població sobre certs aspectes i temàtiques sobre la salut.
- Implementar un mode multi jugador per poder crear partides online de l'Oca de l'Alcohol o del Farma Trivial. Sobretot entre els participants de les xerrades o els actes educatius.
- Afegir nous jocs més interactius i dinàmics. Com per exemple un joc de conducció on es mostrin els efectes de l'alcohol al volant dificultant el control, la visió o la velocitat mentre es condueix.

6. Bibliografia

BEN ORMSTAD. (2014). *Unreal Engine 4 Tutorial - Main Menu with Start / Load Level + Exit Game*. Recuperat de https://www.youtube.com/watch?v=sqyTJ_xXwkI

CGSTUDIO. (2016). *Textures Free 3D Models*. Recuperat de <http://tf3dm.com/>

CHRIS COBB. (2014). *Unreal Engine 4 C++ TriggerVolume Tutorial*. Recuperat de <https://www.youtube.com/watch?v=dC1TteDPy60>

EPIC GAMES, INC. (2016). *Introduction to C++ Programming in UE4*. Recuperat de <https://docs.unrealengine.com/latest/INT/Programming/Introduction/index.html>

EPIC GAMES, INC. (2015). *First Person Shooter C++ Tutorial Shooter*. Recuperat de [https://wiki.unrealengine.com/First_Person_Shooter_\(Tutorial\)](https://wiki.unrealengine.com/First_Person_Shooter_(Tutorial))

EPIC GAMES, INC. (2015). *List of Key/Gamepad Input Names*. Recuperat de https://wiki.unrealengine.com/List_of_Key/Gamepad_Input_Names

EPIC GAMES, INC. (2016). *Unreal Engine 4 For Unity Developers*, Recuperat de <https://docs.unrealengine.com/latest/INT/GettingStarted/FromUnity/>

EPIC GAMES, INC. (2016). *Content Examples*. Recuperat de <https://docs.unrealengine.com/latest/INT/Resources/ContentExamples/index.html>

EPIC GAMES, INC. (2015). *Create A Custom Weapon - How Firing Works*. Recuperat de https://wiki.unrealengine.com/Create_A_Custom_Weapon_-_How_Firing_Works

EPIC GAMES, INC. (2015). *Event handling*. Recuperat de https://wiki.unrealengine.com/Event_handling

EPIC GAMES, INC. (2015). *Iterators: Object & Actor Iterators, Optional Class Scope For Faster Search*. Recuperat de https://wiki.unrealengine.com/Iterators:_Object_%26_Actor_Iterators,_Optional_Class_Scope_For_Faster_Search

EPIC GAMES, INC. (2016). *3rd Person Power-Up Game with C++*. Recuperat de https://wiki.unrealengine.com/Videos/Player?series=PLZlv_N0_O1gb5xvsc7VM7pfoRAKLuIcFi

EPIC GAMES, INC. (2016). *3rd Person Game with Blueprints*. Recuperat de https://wiki.unrealengine.com/Videos/Player?series=PLZlv_N0_O1gZS5HylO_368myr-Kg2ZLwb&video=GWbd0Wowwnk

EPIC GAMES, INC. (2016). *Ambient Sound Actor User Guide*. Recuperat de <https://docs.unrealengine.com/latest/INT/Engine/Audio/SoundActors/index.html>

EPIC GAMES, INC. (2015). *Create An AI Bot in Blueprint*. Recuperat de https://wiki.unrealengine.com/Create_An_AI_Bot_in_Blueprint

TOM LOOMAN. (2014). *UsableActor System in C++*. Recuperat de <http://www.tomlooman.com/tutorial-usableactor-system-in-c/>

UNREAL DEVS. (2015). *Unreal developers*. Recuperat de <https://www.youtube.com/user/unrealdevs/playlists>

7. Annex

En aquest apartat es detallen una sèrie de conceptes sobre com instal·lar i executar l'aplicació en un dispositiu Android (*apartat 7.1*).

7.1. Guia d'instal·lació de l'aplicació

Per instal·lar l'aplicació a un dispositiu mòbil Android s'han de seguir els passos següents:

1. Localitza i copia l'arxiu 'JuegosFarmaUB-Android-Shipping-armv7-es2.apk' que es troba al directori src del codi font lliurat juntament amb aquesta memòria.
2. Connecta el teu dispositiu Android a un PC i enganxa l'arxiu del punt 1 a alguna carpeta del teu dispositiu Android.
3. Un cop es tingui l'arxiu al dispositiu Android, amb un explorador de fitxer (per exemple, l'App ES File Explorer) s'ha de localitzar l'arxiu i polsar-lo.
4. Es mostrarà un missatge amb els permisos que necessita l'aplicació i es donarà la opció per instal·lar l'aplicació. S'ha de seleccionar l'opció d'instal·lar.
5. Un cop instal·lada l'aplicació, solament cal buscar la icona a les aplicacions del dispositiu i obrir-la.

D'altra banda, l'aplicació es troba penjada a la Play Store en versió beta pública. Accedint al link que es mostra a continuació des d'un dispositiu Android, es pot descarregar l'aplicació i participar com a tester:

<https://play.google.com/apps/testing/com.IvanASGames.FarmaJuegosUB>