



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

Juegos serios con Unity

Serious games with Unity

Cristian Luis Hdez.

La Laguna, 31 de agosto de 2015

D. **Carina Soledad González González**, con N.I.F. 54.064.251-Z profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y Sistemas de la Universidad de La Laguna, como tutor.

D. **Yeray del Cristo Barrios Fleitas**, con N.I.F. 54.106.627-R profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y Sistemas de la Universidad de La Laguna, como cotutor.

CERTIFICAN

Que la presente memoria titulada:

“Juegos serios con Unity.”

Ha sido realizada bajo su dirección por D. **Cristian Luis Hernández**, con N.I.F. 43.379.212-Q.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 4 de marzo de 2015.

Agradecimientos

En primer lugar, me gustaría darle las gracias a D. Carina Soledad González González y a D. Yeray del Cristo Barrios Fleitas, por haberme guiado y apoyado durante estos largos meses.

Además, me gustaría darle a todos aquellos profesores que me han permitido llegar hasta aquí, en especial a Concha Cabrera, quien consiguió que me interesase por la informática.

También a todos los profesores que se han preocupado por mí durante estos cuatro años. No hubiera llegado tan lejos de no ser por ellos.

Quisiera también darle las gracias a mi familia, por apoyarme siempre y tener fe en mí. Sobre todo a mis padres, que siempre han estado a mi lado en los peores momentos.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

Resumen

EDHospi es un proyecto colaborativo entre aulas hospitalarias para la mejora del aprendizaje del alumnado hospitalizado o convaleciente mediante el uso de la tecnología.

En las aulas hospitalarias, la inserción de un nuevo paciente suele conllevar miedo y/o ansiedad por parte del mismo. Siguiendo la ideología de EDHospi, este trabajo final de grado propone emplear la gamificación para invitar al alumno a investigar su nuevo entorno. De este modo pretendemos combatir el miedo convirtiendo el cambio en una aventura.

Se plantea, por consiguiente, la creación de una aplicación destinada a los nuevos pacientes que les guíe a través de la zona designada en el hospital, así como la implementación de un editor de niveles destinado a los docentes, en el cual podrán adaptar la aplicación a cada institución.

Palabras clave: EDHospi, Gamificación, Aula hospitalaria, Editor de niveles.

Abstract

EDHospi is a collaborative project between hospital classrooms to improve how students hospitalized or convalescing learn by the use of technology.

The new patient at hospital classrooms can suffer stress or anxiety. The student is involved with a new world which is unknown and frightening. Using the ideology of EDHospi, this project propose create a serious game for solve the situation. The plan is use the gamification for foment the investigation of the new student ambient.

Definitely, this project is about the creation of two applications which can solve the problem presented previously. On one hand, create an application for the students, which attenuate the bad emotion in this change in the student's life. On the other hand, implement an application for the teacher, which allow the teacher adapt the application to the facilities.

Keywords: EDHospi, Gamification, Hospital classroom, Level editor.

Índice General

Capítulo 1. Introducción	4
I.1.1 EDHospi.....	4
I.1.2 Problema a resolver	4
Capítulo 2. Objetivos y decisiones.	6
I.2.1 Objetivo del proyecto.	6
I.2.2 El prototipo.....	6
I.2.2.1 Requisitos funcionales.....	6
I.2.2.2 Requisitos no funcionales.....	7
I.2.2.3 Unity y Vuforia.....	7
Capítulo 3. Desarrollo y Fases.	9
I.3.1 Diseño del prototipo.	11
I.3.1.1 Idea Base.....	11
I.3.1.2 Almacenamiento de datos e interacción app-editor.	12
I.3.2 Implementación de la aplicación Android.	13
I.3.2.1 Menú Inicial.....	13
I.3.2.2 Personalización.....	14
I.3.2.3 Ventana de pistas.....	15
I.3.2.4 Ventana de búsqueda.....	17
I.3.3 Implementación del editor de niveles.	19
I.3.3.1 Funcionamiento del editor.....	19
I.3.3.2 Salida del editor.....	21
Capítulo 4. Estado del proyecto.	22
Capítulo 5. Conclusiones y líneas futuras	23
Capítulo 6. Summary and Conclusions	24
Capítulo 7. Presupuesto	25
Capítulo 8. Bibliografía	26

Índice de figuras

Figura 3.1 Cronograma	10
Figura 3.1.1. Boceto lápiz y papel	12
Figura 3.2.1. Menú inicial.....	13
Figura 3.2.2.1. Personalización 1	14
Figura 3.2.2.2. Personalización 2	14
Figura 3.2.2.3. Clase: Personaje	15
Figura 3.2.3.1. Interfaz: Pistas 1	16
Figura 3.2.3.2. Interfaz: Pistas 2	16
Figura 3.2.3.2. Mapa UV.....	17
Figura 3.2.4.1. Código QR no detectado.....	17
Figura 3.2.4.2. Código QR detectado	17
Figura 3.2.4.3. Código: Cambio de estado.....	18
Figura 3.2.4.4. Ejemplo de Realidad Aumentada.....	19
Figura 3.3.1. Editor: Selección de ID	20
Figura 3.3.2 Editor: Introducción de pistas	20
Figura 3.3.3 Editor: Buscador de archivos	21

Índice de tablas

Tabla 2.3.1. Terminología Unity	8
Tabla 7.1. Presupuesto.....	25

Capítulo 1.

Introducción

I.1.1 EDHospi

Las aulas hospitalarias son un lugar en centros hospitalarios que las comunidades sanitarias y educativas ponen a disposición de los alumnos hospitalizados. EDHospi es un proyecto colaborativo entre aulas hospitalarias que nace de la idea de mejorar la enseñanza que se lleva a cabo actualmente en hospitales por medio de la introducción de las nuevas tecnologías.

La clave del sistema que se emplea en las aulas hospitalarias está en la adaptabilidad y flexibilidad a las condiciones especiales de cada alumno. Durante su estancia en las instalaciones, el principal objetivo de educadores, enfermeros, doctores, etc. consiste en minimizar los aspectos negativos de la condición de sus pacientes. Por un lado, se procura garantizar la continuidad del aprendizaje del alumno, con el fin de que tras su recuperación, pueda continuar sus estudios sin grandes impedimentos. Por el otro lado, este programa también se preocupa de aminorar las posibles consecuencias emocionales que puede llevar la estancia hospitalaria, como pueden ser el aislamiento, la ansiedad o la sensación de vulnerabilidad.

I.1.2 Problema a resolver

La experiencia de estar hospitalizado puede ser considerablemente amarga, y cuando tenemos en cuenta las edades que rondan los pacientes que pasan por las aulas hospitalarias, es más fácil comprender las repercusiones emocionales que pueden ocasionar los primeros días en los alumnos. Desde el punto de vista del paciente, se ha visto obligado a ingresar en un lugar desconocido a causa de una situación que muchas veces es incapaz de comprender.

El miedo y la ansiedad tienden a estar presentes en los primeros días del alumno en su nuevo entorno. Este TFG busca crear un conjunto de herramientas que le permita a los educadores atenuar estas emociones negativas a través de la gamificación.

Usando como base el concepto de los juegos serios y la idea del proyecto EDHospi de emplear la tecnología para mejorar la situación de los alumnos durante su estancia hospitalaria, se pretende desarrollar un videojuego que

aliente al nuevo paciente a explorar su entorno. De este modo, su intenta convertir un lugar extraño y temible, en un sitio que explorar y descubrir.

Capítulo 2.

Objetivos y decisiones.

I.2.1 Objetivo del proyecto.

El principal objetivo de este TFG es generar un prototipo que se capaz de cumplir con la tarea que se nos ha encomendado, es decir, desarrollar una aplicación que facilite la adaptación de los nuevos alumnos a su nuevo entorno en las aulas hospitalarias. Lo que se espera obtener es una herramienta capaz de ayudar a esos jóvenes que atraviesan un aciago momento, pero también facilitar el duro trabajo de los docentes que prestan su ayuda en las aulas hospitalarias. Utilizar el juego como una herramienta para enseñar y a la vez unir a las personas es lo que se espera obtener a través del proyecto EDHospi y ese pensamiento es el que se pretende reflejar en este trabajo.

I.2.2 El prototipo.

Con el fin de cumplir el objetivo de este proyecto se han de analizar detenidamente las necesidades que deberá satisfacer el producto final. Con este propósito se definirán los requisitos funcionales y no funcionales del prototipo.

I.2.2.1 Requisitos funcionales.

En primer lugar, la aplicación a desarrollar debe poder emplearse para explorar el entorno y funcionar como un guía para el alumno. Por este motivo se trabajará enfocado a dispositivos móviles que permitan el desplazamiento por el entorno del aula hospitalaria.

A continuación, se necesitará un sistema de balizas que le indique que ruta tomar al alumno. Aquí se presenta el primer problema a afrontar. Una primera idea fue emplear el GPS para marcar los puntos a seguir, pero esto presenta un claro problema: los sistemas GPS son bastantes problemáticos cuando se trabaja en interiores. Por este motivo, se enfocó el sistema de balizas de otra forma, en vez de emplear la geolocalización se emplearía un

sistema de marcadores, el cual, más adelante se combinaría con realidad aumentada para dar lugar al prototipo final.

El último requisito para que el prototipo sea funcional consiste en la posibilidad de adaptarlo al centro hospitalario. Cada hospital constará con un aula hospitalaria diferente, por lo que, si el prototipo está enfocado a la exploración, este debe adaptarse a las condiciones de cada centro. Este es el motivo por el cual se incluirá el diseño y desarrollo de un editor de niveles, el cual permita a los docentes, de manera sencilla, adaptar la aplicación a las necesidades del paciente y el centro.

1.2.2.2 Requisitos no funcionales.

Hay un último punto que se debe tener en cuenta antes de finalizar el análisis de requisitos de este prototipo. Se ha mencionado anteriormente la implementación de un editor de niveles para la adaptación de la aplicación a la zona en la que se utiliza. Ahora bien, este editor de niveles está destinado a los docentes, es decir, usuarios que no tienen por qué tener conocimientos de informática, de modo que la interfaz y el sistema de generación de niveles deben ser sencillo y fácil de usar.

1.2.2.3 Unity y Vuforia.

Para el desarrollo de este TFG se ha empleado Unity como entorno de desarrollo. Aunque la opción óptima hubiera sido el trabajar directamente en Android nativo, eso hubiera supuesto un incremento desmesurado en el tiempo de desarrollo. Por otro lado, Unity presenta una buena compatibilidad con Android, además de ser un entorno de desarrollo gratuito.

El principal problema que presenta Unity como entorno de desarrollo es la baja calidad de sus físicas frente a sus competidores, sin embargo, es una desventaja que no influirá en este proyecto, dado que no se emplearan físicas.

Finalmente destacar la utilización de la librería Vuforia para el uso de la realidad aumentada. Esta librería ofrece un gran juego de marcadores, así como un pequeño conjunto de herramientas para trabajar con ellos, las cual han facilitado la implementación del prototipo.

1.2.2.3.1 Terminología en Unity.

Al emplear Unity como entorno de trabajo a lo largo de esta memoria se emplearán muchos términos referentes al mismo. Para simplificar la explicación de los mismos en caso de alguna duda se incluirá una tabla a continuación con los elementos más importantes.

Término	Definición
Asset	Un objeto básico de un proyecto Unity. Es un término genérico que referencia imágenes, modelos, texturas, etc.
Escena	Una escena corresponde a un nivel o area en un juego, como puede ser, por ejemplo, un menú. El funcionamiento de Unity se basa en emplear escenas para los distintos escenarios que tendrán lugar a lo largo del programa.
GameObjects	Cualquier elemento que se incluya en la escena de contar con esta clase, la cual permite situar el elemento en el espacio.
Elemento prefabricado	Este nombre hace referencia a un objeto reutilizable, es decir, que puede emplearse en una o más escena a partir de un modelo almacenado, simplemente creando una instancia del mismo. Su principal característica consiste en que al modificar el elemento inicial el resto de elementos del programa también se modifican.

Tabla 2.3.1: Terminología en Unity.

Capítulo 3.

Desarrollo y Fases.

El desarrollo de este proyecto se ha llevado a cabo siguiendo una metodología basada en un Sprint de tres iteraciones. Esto significa que el desarrollo se dividirá en tres fases diferenciadas. El cronograma que se ha planteado seguir es el definido en la figura 3.1.

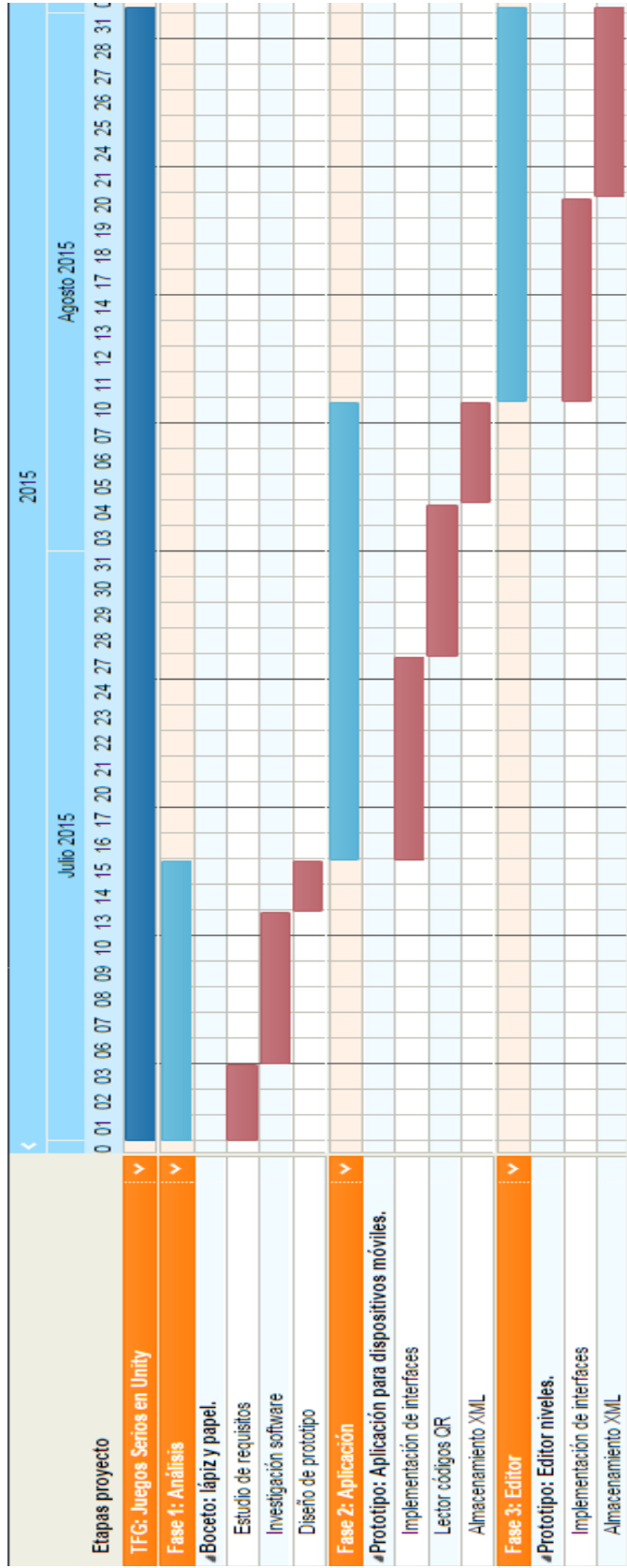


Figura 3.1 Cronograma.

I.3.1 Diseño del prototipo.

El primer punto a destacar del prototipo diseñado es la jugabilidad que presenta. Dentro de los juegos serios, la jugabilidad es uno de los puntos más cruciales, pues debe ser amena a la vez que cumple su objetivo. En este caso, la aplicación está dirigida a guiar al alumno a través de las instalaciones que formaran parte de su nuevo día a día. En definitiva, la jugabilidad elegida se define a través de los siguientes aspectos:

1. El alumno se guiará a través de pistas, empleando un método muy similar al de una búsqueda del tesoro o una yincana.
2. Las pistas con las que contará el alumno se dividen en dos tipos. Por un lado, una pista visual, una imagen que permita identificar el punto exacto al que debe dirigirse. Y por el otro lado, una pista escrita, con el fin de comunicar todo aquello que no fuera posible con una imagen.
3. Para completar el juego, es necesario ir encontrando las balizas, en orden y leyéndolas (códigos QR) con la cámara del dispositivo en el que se emplee la aplicación.

Uno de los motivos para diseñar este tipo de jugabilidad es la posibilidad de aprovechar la situación en la que se encuentra el usuario. Si el personaje del usuario se encuentra en su misma situación, es mucho más fácil que se identifique con él. Asimismo, se incluirá un sistema de personalización de personaje, muy sencillo, con el fin favorecer aún más la inmersión del usuario en el juego.

I.3.1.1 Idea Base.

En este punto del proyecto ya se han fijado las principales ideas que se llevarán a cabo durante el desarrollo; sin embargo, esto no es suficiente para comenzar la implementación. Antes es necesario realizar un boceto sencillo, en este caso se ha realizado a lápiz y papel, que se empleará como guía durante el desarrollo.

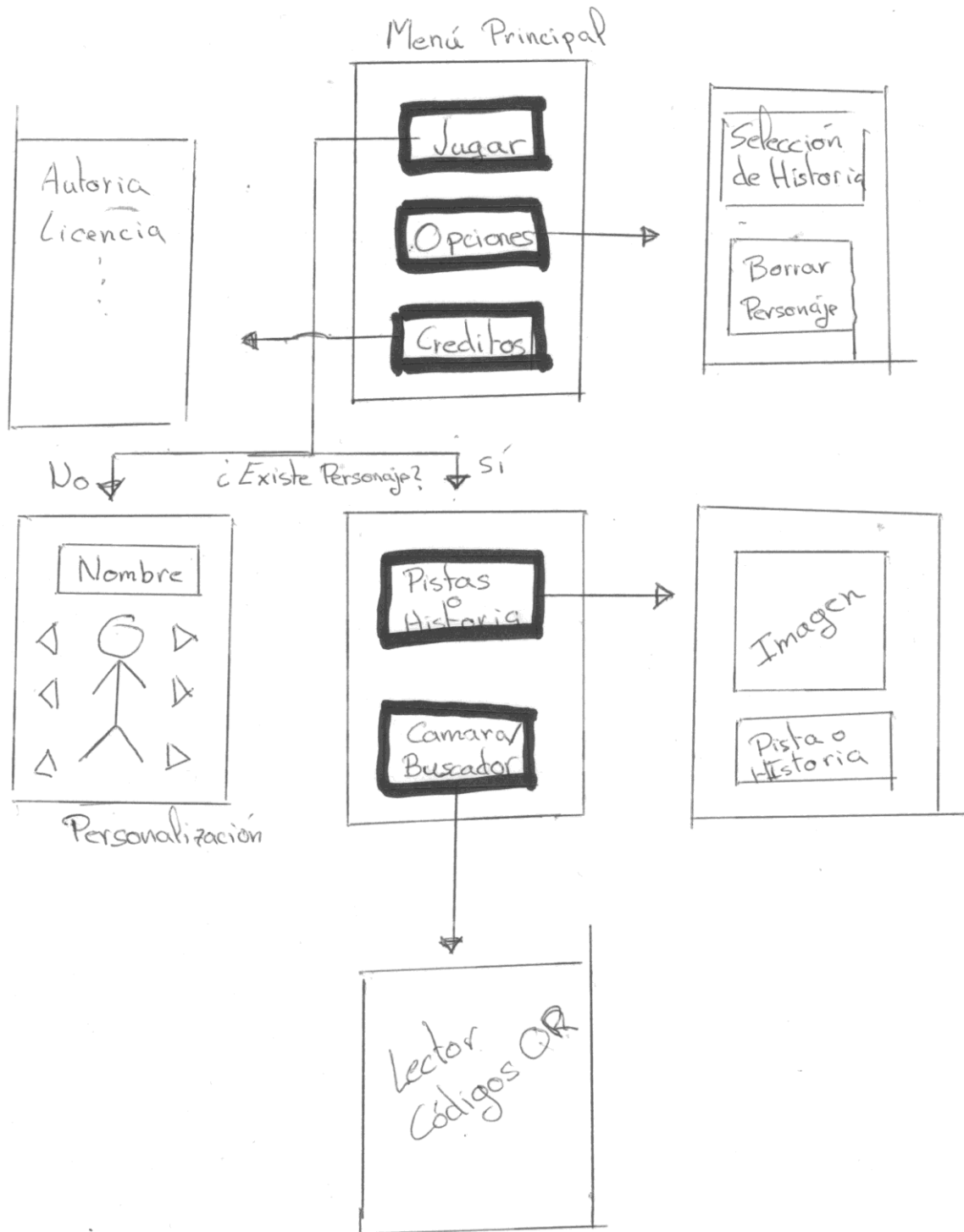


Figura 3.1.1 Boceto lápiz y papel.

I.3.1.2 Almacenamiento de datos e interacción app-editor.

Como se ha mencionado anteriormente, el desarrollo de este proyecto constará de dos aplicaciones que funcionarán colaborativamente para cumplir con el objetivo final. Sin embargo, para la comunicación entre las dos

plataformas se ha de definir un medio y, en esta ocasión, se ha optado por emplear archivos XML a modo de base de datos.

Son dos los motivos por los cuales se ha decidido emplear este sistema. En primer lugar, las aplicaciones a desarrollar están enfocadas a distintas plataformas, el *juego* para dispositivos móviles y el editor de niveles para ordenador, lo que podría desencadenar, en algunos casos, problemas de compatibilidad. El segundo motivo viene dado por el entorno de desarrollo, ya que Unity consta de una amplia gama de herramientas que permiten trabajar con los ficheros XML de manera eficaz.

1.3.2 Implementación de la aplicación Android.

A partir del boceto inicial se retocaron y pulieron las interfaces a implementar dando como resultado un conjunto de escenas o estados, cuyo funcionamiento y detalles de mayor relevancia se explicarán a continuación.

1.3.2.1 Menú Inicial.

El menú inicial es lo primero que ve el usuario al iniciar la aplicación, por lo tanto debe ser claro y sencillo. Por este motivo se ha optado por emplear sólo tres botones.

Lo más destacable de esta pantalla es el evento que se lanza al iniciar el juego, ya que se debe tener en cuenta de que la partida puede estar iniciada o no. En el caso de que el usuario entre por primera vez, es necesario que se cargue el creador de personaje. No obstante, si el usuario ya ha creado su personaje debería transcurrir directamente al juego. Para esto se ha empleado un fichero de configuración XML.



Figura 3.2.1: Menú inicial.

Este fichero de configuración, nombrado como “config.xml”, se crea a partir de una clase definida en C# empleando métodos de Unity. Este proceso es

conocido como serialización y permite traducir una estructura de datos sencilla a un fichero XML.

I.3.2.2 Personalización.

Uno de los puntos vitales para la inmersión en el *juego*, como mencionamos anteriormente, es la posibilidad de que el usuario cree su propio personaje. Con este fin se han implementado un sistema de personalización basado en dos pantallas. En primer lugar, el usuario seleccionará su nombre, para a continuación modificar el que corresponderá a su personaje durante la partida.



Figura 3.2.2.1: Personalización.



Figura 3.2.2.2: Personalización.

La información de personalización se almacena también en un fichero XML a través de la serialización. El proceso es el siguiente: En primer lugar, se ha definido la clase "Personaje" con los atributos necesarios para almacenar la información necesaria. A continuación, se ha especificado con etiquetas ([XmlAttribute]), las cuales indican como deben ser interpretados los datos al pasar a fichero XML. Finalmente, se declara un objeto XmlSrializer, el cual se encargará de cambiar el formato de la información, y se almacena en un fichero.

```

public class Personaje
{
    /// <summary>
    /// Constructor por defecto, elimina errores en la serializacion.
    /// </summary>
    public Personaje()
    {
        Name = "NoName";
        Head = 1;
        Body = 1;
        Legs = 1;
    }
    /// <summary>
    /// Constructor de asignacion directa. Se utilizará preferiblemente este constructor.
    /// </summary>
    public Personaje(string _name, int _head, int _body, int _legs)
    {
        Name = _name;
        Head = _head;
        Body = _body;
        Legs = _legs;
    }

    [XmlAttribute]
    public string Name { get; set; }
    [XmlAttribute]
    public int Head { get; set; }
    [XmlAttribute]
    public int Body { get; set; }
    [XmlAttribute]
    public int Legs { get; set; }
}

```

Figura 3.2.2.3: Clase Personaje.

1.3.2.3 Ventana de pistas.

Esta ventana cuenta con dos tipos de pistas. En primer lugar, una pista visual, es decir, una imagen de un punto concreto. La función de este tipo de pista es permitir que el usuario pueda identificar donde se encuentra su objetivo, es decir, el lugar al que debe dirigirse. La principal importancia de esta pista se basa en la premisa de que el usuario se encuentra en un lugar nuevo y desconocido. En segundo lugar, esta ventana presenta también una pista escrita, la cual será necesaria para introducir indicaciones, narrativa u otro tipo de información que el docente crea necesaria.

En esta ventana se ha añadido un pequeño minijuego, con el fin de mejorar la experiencia de exploración y búsqueda. La idea consiste en dividir la imagen en una cuadrícula 3 x 3, al pulsar una de las casillas se invierte tanto esta como todas las adyacentes.

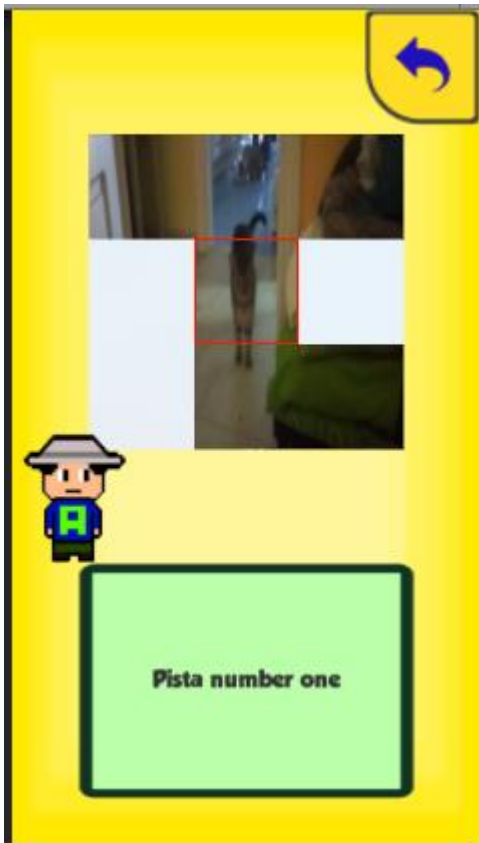


Figura 3.2.3.1: Interfaz: Pistas 1.

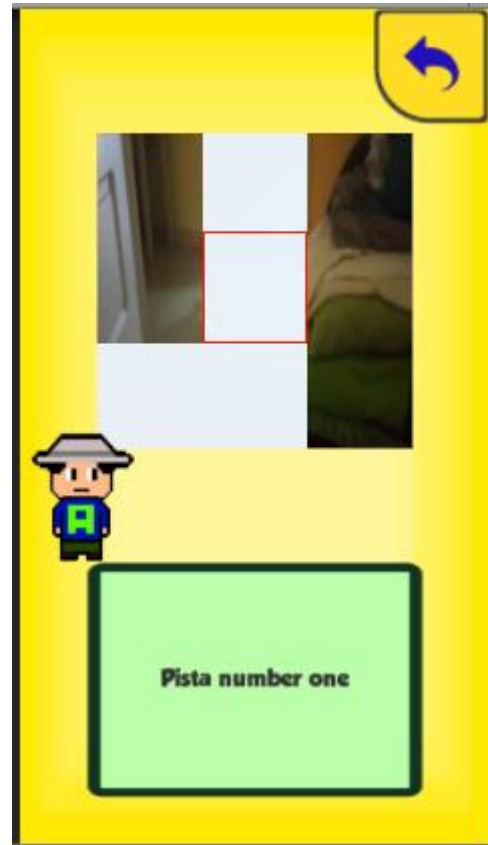


Figura 3.2.3.2: Interfaz: Pistas 2.

La subdivisión de imágenes se ha llevado a cabo a través de mapeado por UVs sobre paneles. Un mapa UV hace referencia a un plano cuadrado de 1x1, donde los vectores \vec{i} y \vec{j} se les renombra como \vec{u} y \vec{v} respectivamente. Esto se traduce como un mapa, tal y como muestra la figura 3.2.3.3 el cual permite referenciar cualquier punto de una textura.

Un objeto tridimensional en Unity se define a partir de un conjunto de puntos en las tres dimensiones. Estos puntos forman pequeños triángulos y son, estos triángulos, los que crean el cuerpo tridimensional.

Para aplicar una textura sobre un objeto, basta con indicar que punto de la textura se corresponde con cada vértice de la figura. A partir de estos puntos de referencia, el motor gráfico extiende la textura sobre los triángulos de la superficie del objeto.

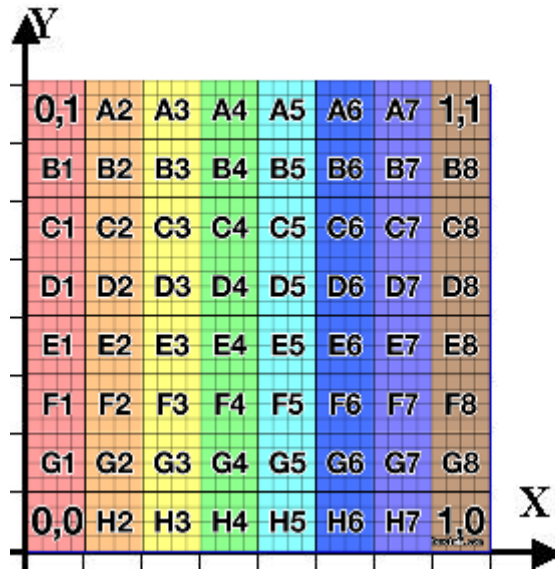


Figura 3.2.3.3: Mapa UV.

I.3.2.4 Ventana de búsqueda.

La última ventana que queda por destacar se trata de la ventana de búsqueda. En esta ventana se captura una imagen en tiempo real a través de la cámara y se procesa con el fin de identificar un código QR. El procesado de la imagen se realiza a través de una librería auxiliar denominada Vuforia. Se ha añadido un evento a la función de reconocimiento proporcionada por esta librería con el fin de lanzar un mensaje en pantalla si se ha detectado el código QR que se buscaba.

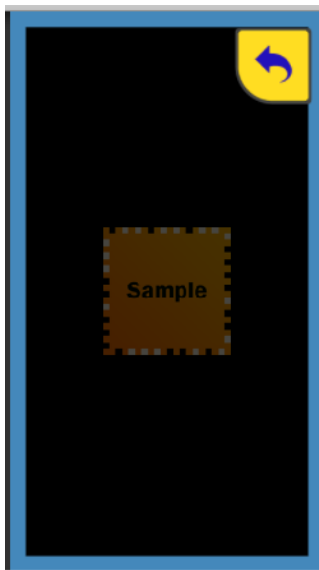


Figura 3.2.4.1: Código QR no detectado.



Figura 3.2.4.2: Código QR detectado.

Vuforia mantiene un estado que determina si se está visualizando o no el código QR. En cada frame del video recogido por la cámara del dispositivo móvil, se lanza la función de detección. En el caso de que la función detecte un cambio de estado, es decir, empieza o deja de detectar el código QR en el frame, se lanza la función encargada de controlar este evento, `OnTrackableStateChanged`. Esta función es la encargada de determinar si los elementos pertenecientes a la realidad aumentada son visibles o no. Aprovechando este comportamiento, se han incluido dos funciones encargadas de lanzar los eventos correspondientes, con el fin de actualizar la interfaz de la ventana, o lo que es lo mismo, mostrar u ocultar el mensaje de encontrado.

```
public void OnTrackableStateChanged(  
    TrackableBehaviour.Status previousStatus,  
    TrackableBehaviour.Status newStatus)  
{  
    if (newStatus == TrackableBehaviour.Status.DETECTED ||  
        newStatus == TrackableBehaviour.Status.TRACKED ||  
        newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)  
    {  
        OnTrackingFound();  
        LanzadorEventos.MarcadorEncontrado();  
    }  
    else  
    {  
        OnTrackingLost();  
        LanzadorEventos.MarcadorPerdido();  
    }  
}
```

Figura 3.2.4.3: Código: Cambio de estado.

El funcionamiento de Vuforia, al igual que muchas otras extensiones de Unity, se basa en la utilización de elementos prefabricados. En este caso proporciona un elemento denominado Marcador, el cual permite la utilización de realidad aumentada de manera sencilla. Basta con añadir al Marcador los elementos que se buscan que aparezcan al detectar el código QR.

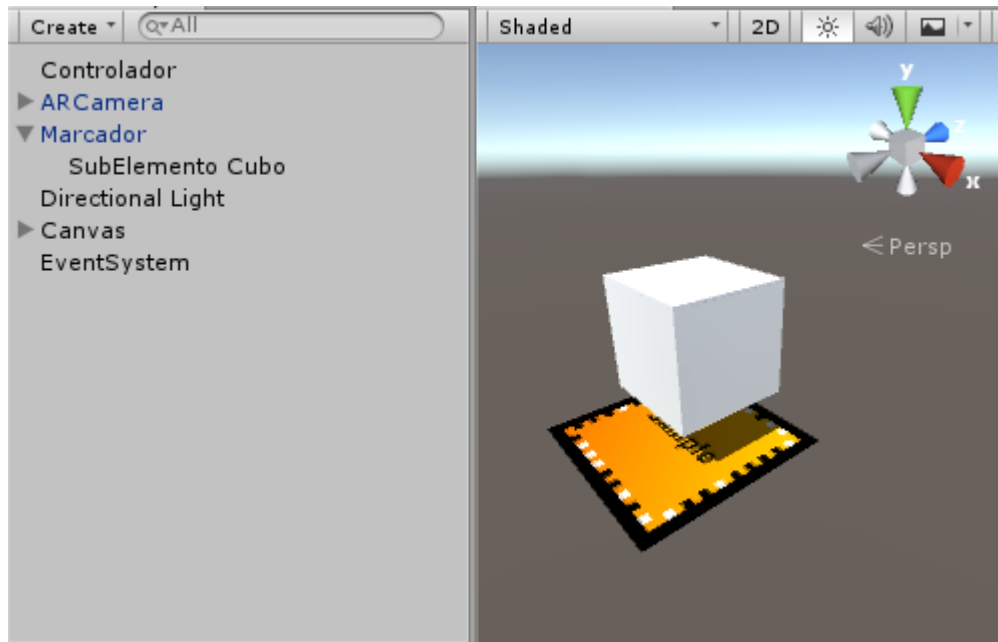


Figura 3.2.4.4: Ejemplo Realidad Aumentada.

I.3.3 Implementación del editor de niveles.

El editor de niveles será una herramienta a disposición de los docentes para poder modificar la aplicación a las necesidades de las instalaciones y el alumno. Uno de los principales puntos a tener en cuenta es que el docente no tiene por qué poseer ningún conocimiento previo sobre informática, por lo que será imprescindible una interfaz sencilla y simple. Partiendo de esta base se pasará a explicar el funcionamiento del mismo.

I.3.3.1 Funcionamiento del editor.

La principal función del editor de niveles es general el conjunto de archivos necesarios para que al pasar dicha información a la aplicación, esta pueda funcionar adecuándose al ambiente de uso. Los pasos a seguir por el usuario son muy sencillos. En primer lugar, ha de identificar el nivel que desea crear con un nombre, el cual funcionará como identificador en la aplicación.

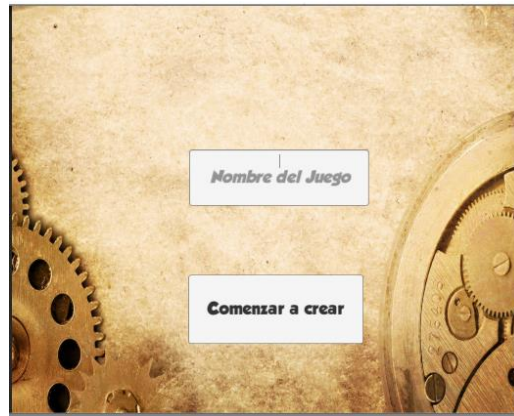


Figura 3.3.1: Editor: Selección de id.

Una vez decidido el nombre que identificará al nivel, es el momento para el usuario de comenzar a crear la historia. Cada fase contendrá dos elementos. Por un lado, una pista escrita, que se empleará para dirigir la historia o guiar al alumno.

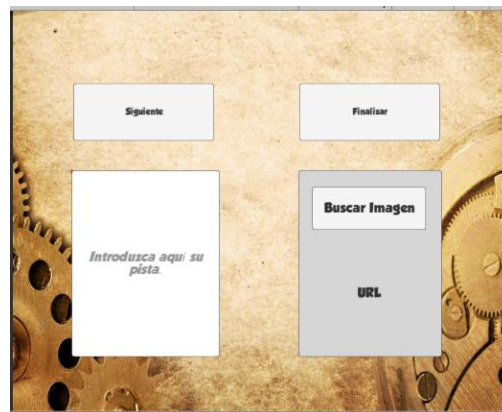


Figura 3.3.2: Editor: Introducción de pistas.

A continuación, el usuario deberá incluir una imagen, que cumpla la función de guiar al alumno hasta el donde se encuentre el código QR. Para realizar esta acción el usuario simplemente deberá presionar sobre el botón de buscar imagen, el cual desplegará un buscador de archivos en el que deberá seleccionar la imagen que crea adecuada.



Figura 3.3.1: Editor: Buscador de archivos.

Para añadir un nuevo nivel, el usuario simplemente presionará “Siguiente” y para finalizarlo, simplemente presionará “Finalizar”.

1.3.3.2 Salida del editor.

Durante la creación del nivel, el programa se encarga de ir almacenando toda la información necesaria para el mismo, ubicación de imágenes, id, número de fases, etc. El resultado al finalizar la creación del nivel será el siguiente.

En primer lugar, toda la información respectiva al nivel se almacenará en una carpeta cuyo identificador corresponderá al nombre del nivel. Dentro de esta carpeta encontraremos dos tipos de archivos. En primer lugar, encontraremos una copia del conjunto de imágenes que se seleccionaron, las cuales se emplearán durante la ejecución de la aplicación. El otro tipo de archivo consistirá en un documento XML, el cual contendrá la información relativa al nivel, como son el número de fases o las pistas a usar.

Finalmente, se ha de mencionar un pequeño arreglo que se ha llevado a cabo a la hora de almacenar las pistas en el documento XML. Para almacenar el conjunto de pista se emplea una lista de cadenas de caracteres. Esto no desencadena ningún problema al almacenar la información en un fichero XML, sin embargo si los ocasiona al tratar de recuperarla. Esto se debe a que los espacios en un documento XML se interpretan como el final del elemento de una lista, lo que supone que un texto que debía ser un único elemento se trasforme en un conjunto de elementos para la lista. Para paliar este problema, antes de almacenar la lista de cadenas, se realiza un remplazo de los espacios, de modo que la información se pueda recuperar correctamente.

Capítulo 4.

Estado del proyecto.

En este punto del proyecto ya está disponible un prototipo funcional, tanto en lo que consta a la aplicación destinada a los nuevos estudiantes de aulas hospitalarias, como el editor de niveles destinado a los docentes. Sin embargo, en este punto aún se hacen presentes carencias importantes para poder pasar del prototipo desarrollado hasta el momento a un producto final.

En primer lugar, el prototipo actual se ha probado en dispositivos móviles con Android como sistema operativo. Sin embargo, al no disponer de ningún dispositivo con IOS y las dificultades que presenta la exportación de aplicaciones a este sistema operativo desde Windows, ha sido imposible comprobar su funcionalidad en estos sistemas.

También cabe destacar el apartado artístico del juego. Aunque durante el desarrollo del proyecto se ha contado con el apoyo de un graduado en bellas artes, dada la naturaleza del proyecto, sus aportaciones han sido meramente como asesor. Esto ha culminado en un apartado gráfico muy modesto. Siendo los gráficos uno de los apartados más importantes en este tipo de aplicaciones, este campo debería ser uno de los que más se tengan en cuenta a mejorar en versiones posteriores.

Por último, se hará mención a uno de los campos más importantes a la hora de desarrollar un videojuego, pero que no se ha podido incluir en este TFG. Este campo es el sonido. Una banda sonora facilita la inmersión del usuario y hace más rica la experiencia. Sin embargo, dado los escasos conocimientos en este campo se ha tomado la decisión de dejar este campo como una línea futura de desarrollo.

Capítulo 5.

Conclusiones y líneas futuras

Desde mi punto de vista son dos los puntos más interesantes de este proyecto como trabajo de final de carrera. En primer lugar, me ha brindado la posibilidad de trabajar con tecnología que no se ve durante el transcurso del grado. Los motores gráficos han sido objetivo de mi curiosidad, por lo que aprender a trabajar en Unity ha resultado un factor muy motivante durante el transcurso de este trabajo. También mencionar lo interesante que ha resultado el trabajar con códigos QR, que aunque este tipo de tecnología sí que ha estado más presente en la carrera, el haber desarrollado este trabajo me ha permitido añadirle profundidad a lo que había aprendido sobre este asunto.

Asimismo, el otro gran punto al que me gustaría hacer mención sobre este proyecto tiene que ver con la naturaleza del mismo. Este trabajo se me ha presentado como un problema real a resolver. El no disponer de un guion predefinido para solucionar un problema es una situación que sólo he encontrado durante el transcurso de las prácticas externas, pero que estará presente durante toda mi carrera laboral. Por este motivo, agradezco en gran medida el haber podido probarme a través de este trabajo, que soy capaz de afrontar este tipo de situaciones.

Por último, se indicará, en caso de continuar con este proyecto, cuál sería la siguiente línea de trabajo. En este momento, el prototipo cuenta con dos aplicaciones que se basan en la colaboración para cumplir su objetivo. No obstante, no existe una unión directa entre las mismas. Por lo tanto, el siguiente punto de desarrollo lógico sería la creación de un servidor, un nexo entre las dos aplicaciones.

Capítulo 6.

Summary and Conclusions

This project include the use of gamification for resolve the emotional problems which can suffer the students that are hospitalized. The hospitalization is a sudden change in the student's life which may have negative effects. To solve this problem, it has been developed the necessary software to motivate the students to learn and explore their new environment as well as it attenuates the aftermath of this changes.

In my opinion, this project has been two very positives aspects in my formation. First, I needed learn about how graphic engine works, which isn't seen in the grade. The other positive aspect is the possibility to work with a real problem. This type of experience I lived only in the external practices. However, I think that this type of experience is very important in our formation.

Capítulo 7.

Presupuesto

En cuanto al coste del proyecto, aunque no se trata de un trabajo muy grande, sí que ha requerido un gran gasto de tiempo. Esto se ha debido principalmente al desconocimiento a priori de muchas de las herramientas que se han empleado. Además, mencionar que todo el software empleado en el desarrollo de este proyecto es totalmente gratuito en lo referente al entorno educativo, por lo que no se contemplan costes en lo referente a este aspecto.

El cálculo del presupuesto se ha realizado en torno al cronograma de la figura 3.1, a partir del cual se han estimado las horas de trabajo y los costes correspondientes según la tabla 7.

Fase	Tipo de tarea	Coste/Hora	Horas	Coste
Análisis y diseño	Análisis y preparación	10€/h	40h	400€
	Implementación	20€/h	5h	100€
	Investigación del software a usar	10€/h	15h	150€
	Subtotal	-	60h	650€
Aplicación para dispositivos móviles	Análisis y preparación	10€/h	10h	100€
	Implementación	20€/h	70h	1400€
	Investigación del software a usar	10€/h	35h	350€
	Subtotal	-	115h	1850€
Investigación sobre el software a emplear.	Análisis y preparación	10€/h	10h	100€
	Implementación	20€/h	70h	1400€
	Investigación del software a usar	10€/h	15h	150€
	Subtotal	-	95h	1650€
Total	-	-	270h	4150€

Tabla 7. Presupuesto.

Capítulo 8. Bibliografía

- [1] Proyecto EDHospi. <http://www.edhospi.esy.es/>.
- [2] Unity 3D. <https://unity3d.com/es>
- [3] Vuforia. <https://developer.vuforia.com/>
- [4] Foros Unity: <https://unity3d.com/es/community>
- [5] Conceptos generales: <https://es.wikipedia.org/wiki/Wikipedia:Portada>
- [6] SDK de Android: <https://developer.android.com/sdk/index.html>
- [7] Apartado gráfico: <http://www.pixelsmil.com/>
- [8] Programación C#: <https://msdn.microsoft.com>
- [9] Modelaje 3D: <https://www.blender.org/>