

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Grado

Prototipo de Videojuego (Ocio o No Ocio – Serious Game)

Alumno: Samuel Martínez de Lamo

Tutor: Prof. D. Francisco Ramón Feito Higuera
Prof. D. Juan Roberto Jiménez Pérez

Dpto: Informática

Septiembre, 2016



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Francisco Ramón Feito Higuera y Don Juan Roberto Jiménez Pérez, tutores del Proyecto Fin de Carrera titulado: Prototipo de Videojuego (Ocio o No Ocio – Serious Game), que presenta Samuel Martínez de Lamo, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2016

El alumno:

Los tutores:

Samuel Martínez de Lamo

Francisco Ramón Feito Higuera
Juan Roberto Jiménez Pérez

Agradecimientos

A mis tutores, Francisco Ramón Feito Higuera, por darme valiosos consejos que han hecho posible este trabajo, y Juan Roberto Jiménez Pérez, por despertar en mí la pasión por el desarrollo de videojuegos con su labor docente y la atención dedicada.

A mis profesores de la Universidad de Jaén, por enseñarme y hacerme crecer en todos los aspectos.

A mis amigos de la Universidad de Jaén y a mis compañeros de piso, que han hecho posible unos años de alegrías y que siempre han hecho lo posible por ayudarme cuando lo he necesitado.

A mis amigos de siempre, en especial a José Francisco Arenas Fernández y Óscar Sánchez Molina, pues son responsables de muchos de los valores que hoy tengo y han demostrado ser amigos de verdad.

A mis padres, a mi hermana, y a mis abuelos, por entregarme todos los medios posibles para construirme un futuro que siempre he deseado y apoyarme bajo cualquier circunstancia, y al resto de mi familia, por ser siempre un soporte para seguir adelante.

A mi familia polaca, por toda la entrega, el cariño y el apoyo.

Finalmente, a Karola Krzesińska, por saber darme la motivación, fuerza, dedicación y ánimo durante cada día sin excepción, por aligerar mi peso en mi camino y demostrarme que todo se puede conseguir si se quiere. El fin de mis estudios y este proyecto presente no hubieran sido posibles sin ti.

Gracias a todos.

Índice

1.	INTRODUCCIÓN	7
1.1.	Motivación	9
1.2.	Objetivo.....	10
1.3.	Estructura del documento	12
2.	EL ESTADO DEL ARTE DEL VIDEOJUEGO	14
2.1.	Introducción.....	14
2.2.	El concepto de videojuego.....	17
2.3.	Serious Games frente a los videojuegos tradicionales.	19
2.4.	Componentes de un videojuego	21
3.	EL DESARROLLO DEL VIDEOJUEGO	24
3.1.	Introducción.....	24
3.2.	Fases de desarrollo de un videojuego.	25
4.	EL MOTOR DE JUEGOS	29
4.1.	Introducción.....	29
4.2.	El motor de juegos. Características principales.	30
4.3.	Estudio sobre motores de juegos actuales.	32
4.3.1.	CryEngine 3 SDK	34
4.3.2.	Unreal Engine 5.....	37
4.3.3.	Unity 5.....	40
4.4.	Elección del motor de juego.	44
5.	DESARROLLO DEL PROTOTIPO	46
5.1.	Introducción.....	46
5.2.	Metodología de trabajo	47
5.3.	Planificación	50
5.4.	Diseño	51
5.4.1.	Concepción y diseño	51
5.4.2.	Género	54
5.4.3.	Jugabilidad	54
5.5.	Implementación	55
5.5.1.	Diseño técnico	55
5.5.1.1.	Diagrama de secuencia.....	55
5.5.1.2.	Herramientas software y hardware.....	56

5.5.2.	Diseño del juego	57
5.5.3.	Diseño Mecánico.....	60
5.5.4.	Diseño artístico	65
5.5.5.	Implementación.....	66
5.6.	Pruebas	68
6.	CONCLUSIONES	71
7.	LÍNEA DE FUTURO	73
	Bibliografía	74
	Anexo I: Manual de instalación.....	76
11.	Descarga e instalación de Unity 5.....	76
12.	Descarga e instalación del paquete Vuforia.....	78
13.	Configuración de Vuforia en Unity.	81
14.	Creación de bases de datos de imágenes objetivo.	82
15.	Instalación del SDK de Android y generación de la aplicación.....	83
	Anexo II: Manual de usuario.	86

Índice de figuras

Figura 1: Assassin's Creed en el cine y en los videojuegos.....	14
Figura 2: Asociación de Minecraft y arquitectura	15
Figura 3: Evolución de los videojuegos para consola (2013)	15
Figura 4: Shigeru Miyamoto junto a su creación más famosa	16
Figura 5: Visión esquematizada del videojuego	17
Figura 6: Microsoft Flight Simulator (Serious Game)	19
Figura 7: Serious Game	20
Figura 8: Etapas del desarrollo del videojuego	28
Figura 9: Game Engine Architecture	32
Figura 10: Listado de motores de videojuegos	33
Figura 11: CryEngine3.....	34
Figura 12: Blueprints en Unreal Engine.....	37
Figura 13: ARToolkit en Unreal Engine.....	40
Figura 14: SDK de Unity.....	43
Figura 15: Vuforia sobre Unity	43
Figura 16: Diagrama de Gantt	50
Figura 17: Concepto GUI	52
Figura 18: Storyboard del juego (1).....	53
Figura 19: Storyboard del juego (2).....	53
Figura 20: Logotipo de la gymkhana	53
Figura 21: Diagrama de jugabilidad.....	54
Figura 22: Diagrama de secuencia del juego.....	55

Figura 23: Campo de bases de datos en la cámara AR de Vuforia.....	57
Figura 24: Campo de licencia en la cámara AR de Vuforia.....	58
Figura 25: Canvas de la escena con sus elementos correspondientes	58
Figura 26: Imagen objetivo de una escena	59
Figura 27: Elementos pertenecientes a la prueba	59
Figura 28: Tabla de ruta propuesta para el videojuego	64
Figura 29: Recorrido a través de la ciudad de Jaén.....	65
Figura 30: Paleta de colores del logotipo.....	66
Figura 31: Paleta de colores de los elementos (puede contener transparencia)	66
Figura 32: Escena completa del Segundo nivel.....	68
Figura I 1: Tienda para descargar el motor Unity 5.....	76
Figura I 2: Instalador de Unity	77
Figura I 3: Selección de arquitectura.....	77
Figura I 4: Formulario de registro en Vuforia	78
Figura I 5: Descarga del paquete Vuforia	78
Figura I 6: Pantalla de creación de nuevo proyecto	79
Figura I 7: Proyecto sin elementos	79
Figura I 8: Menú de importación de paquetes personalizados.....	80
Figura I 9: Importar el paquete Vuforia.....	80
Figura I 10: Creación de licencia de Vuforia	81
Figura I 11: Licencia de Vuforia	81
Figura I 12: Configurando Vuforia sobre Unity	82
Figura I 13: Añadiendo un nuevo objetivo	83
Figura I 14: Activando la base de datos en Unity	83
Figura I 15: Menú de descargas del JDK de Java	84
Figura I 16: Descarga del instalador del SDK de Android	84
Figura I 17: Android SDK Manager	85
Figura I 18: Exportando el juego a Android.....	85
Figura II 1: Pantalla de bienvenida con las instrucciones del juego	86
Figura II 2: Pista con información sobre el siguiente objetivo	87
Figura II 3: Ejercicio oculto en la primera etapa.....	88
Figura II 4: Pista del nuevo destino tras la prueba de la etapa	89
Figura II 5: Destino mostrado de forma directa tras un período de tiempo.....	90

1. INTRODUCCIÓN

A medida que avanzan los tiempos, tanto la tecnología como el conocimiento que tenemos de ella aumentan. En estos tiempos, vivimos en una sociedad en la que la información está en un punto álgido gracias a la aparición de la Economía del Conocimiento y de medios de comunicación de masas, que generan necesidad de información en la mayoría de las personas. Además, la tendencia actual es disponer de una buena cantidad y calidad de información de forma rápida y sencilla. Así, los dispositivos que tenemos se adaptan a esta época con numerosas aplicaciones y widgets que surgen día a día. No sólo eso, sino que el hardware también avanza a grandes pasos para ofrecernos nuevas experiencias conforme a los nuevos tiempos. A fin de cuentas, vivimos en la Era de la Información.

En los últimos años empezamos a oír hablar de nuevas tecnologías que permiten darnos nuevas experiencias conforme a lo que demanda la sociedad actual. Es entonces cuando aparece la Realidad Virtual y, por así decirlo, su hermana, la Realidad Aumentada. La primera de ellas pretende trasladarnos a un mundo completamente virtual recreado para que la persona se sienta parte del mismo. La segunda por otra parte, la cual es objeto de estudio en este documento, pretende utilizar el mundo que nos rodea sumergiéndonos en él de una manera diferente, haciendo que nuestra percepción y concepción del mundo que nos rodea se altere de forma parcial.

Por la importancia que supone el uso de la Realidad Aumentada en este proyecto, es necesario recurrir a una definición formal de la misma para entender qué es lo que tenemos ante nosotros. Según la Fundación Telefónica (Realidad Aumentada: una nueva lente para ver el mundo, 2011) la Realidad Aumentada es un conjunto de tecnologías que permiten la superposición, en tiempo real, de imágenes, marcadores o información generados virtualmente, sobre imágenes del mundo real, creando un entorno en el que la información y los objetos virtuales se fusionan con los objetos reales, ofreciendo una experiencia tal para el usuario que puede llegar a pensar que forma parte de su realidad cotidiana, olvidando incluso la tecnología que le da soporte.

Para que la Realidad Aumentada funcione, se necesitan cuatro componentes esenciales:

- Un elemento de captura de realidad, como por ejemplo una cámara, que extraiga la realidad que están viendo los usuarios.
- Un elemento de proyección sobre la cual se plasman las imágenes reales y las imágenes sintetizadas entremezcladas, por ejemplo una pantalla.
- Un elemento de procesamiento, o varios de ellos, que interpreten la información del mundo real, generen la información virtual y mezclen ambas de forma adecuada.
- Un elemento activador de realidad aumentada, elementos de localización, etiquetas o marcadores que suministren información equivalente a la que proporcionaría lo que ve el usuario, de igual forma que trabajaría un sensor.

En base a lo anteriormente descrito, los primeros pasos de la Realidad Aumentada en la sociedad fueron precisamente con el objetivo de que la información formase parte de nuestro día a día y además estuviese integrada con el mundo que nos rodea. Los conceptos eran sencillos pero potentes a la vez, pues pretendían que el usuario recibiera información a través de lo que veía o de donde se encontrase en ese momento. Surgieron los primeros hardware que pretendían acoplar la información a nuestra vida, dispositivos que se asimilaban a unas gafas en las que se mostraba los lugares o sitios emblemáticos que el usuario estaba viendo, o permitía ver quién llamaba al teléfono del usuario sin hacer uso del terminal móvil.

A partir de este momento se empieza a desarrollar la Realidad Virtual en otros ámbitos, que van desde el marketing hasta el mundo de los videojuegos. Tan sólo tenemos que observar el impacto social que ha supuesto un juego de Realidad Aumentada como es el reciente Pokémon GO, el cual con una sencilla idea ha conseguido movilizar a millones de personas para que jueguen y utilicen la aplicación.

En resumen, esta tecnología poco a poco se va introduciendo cada vez más en nuestras vidas de formas que hace unos años eran totalmente impensables para las personas. Están pensadas para ayudarnos y mejorar nuestras experiencias, pero también para divertirnos y hacernos disfrutar del mundo que nos rodea de una forma que de otra manera sería prácticamente imposible.

1.1. Motivación

Debido al impacto que produce en la sociedad la combinación de la educación con la Realidad Aumentada, como ya hemos introducido anteriormente, nuestro proyecto va a partir de la asociación de dichos conceptos para desarrollar una aplicación que permita educar y a la vez divertir al usuario. Nos atrae mucho la idea de aproximarnos a desarrollar un videojuego que sea capaz de transmitir conocimiento mediante el entorno que te rodea.

Creemos que la combinación de un dispositivo móvil con las técnicas de Realidad Aumentada son un punto a favor para este tipo de juegos que pretenden sumergir al jugador en un entorno realista y que éste sea consciente del entorno que le rodea y sea capaz de apreciar y extraer información del mundo real.

Nos interesa aproximarnos al mundo de los videojuegos y dar a conocer un poco más acerca del mismo, y sobre todo profundizar en las herramientas que ofrecen las tecnologías actuales en el mundo de los videojuegos. Con todo esto, queremos abordar gracias al conocimiento adquirido un proyecto de desarrollo que nos parece interesante y, además, nos complace la idea de estar entregando algo al público que puede aportar cierto conocimiento a aquellas personas que jueguen de una forma transparente, sencilla y divertida. Este conocimiento a transmitir puede abarcar un mundo muy grande gracias al aspecto que queremos fomentar, que no es otro que el turismo en la ciudad de Jaén, lo que nos permite una gran libertad a la hora de transmitir aspectos que no son tan conocidos o que creemos que son de cultura general.

A modo personal, haber cursado en la Universidad de Jaén una asignatura específica de videojuegos y poder entrar en este mundo, a la vez tan complejo y tan fascinante, ha sido un elemento decisivo a la hora de escoger este proyecto. Repetir la experiencia y poder trabajar en un proyecto a nivel individual de estas características es una oportunidad única que seguro nos va a satisfacer.

1.2. Objetivo

Nuestro objetivo principal es aproximarnos al concepto de juego y de educación, ya que creemos que ésta es un aspecto fundamental de la sociedad en la que vivimos. La cultura forma parte indispensable del mundo actual, pero no muchas veces la sociedad cree necesario adquirir nuevos conocimientos sobre ciertas materias.

Por otra parte, la tecnología también florece actualmente y nos llama la atención entrar en este mundo en el que las aplicaciones y, en concreto, los videojuegos, están a la orden del día. Creemos por tanto que ofrecer una alternativa en plataforma móvil que a la vez que divierte, de una forma sencilla, nos proporcione nuevos conocimientos y dé a conocer al mundo ciertos aspectos de nuestra cultura es una propuesta interesante y que tiene cabida en la sociedad actual.

Ofrecemos a través un videojuego serio o videojuego formativo una alternativa al clásico juego de la gymkhana gracias a la Realidad Aumentada, que sea capaz de fomentar el turismo de la ciudad, concretamente de la ciudad de Jaén, y cuyo objetivo es que el usuario participe en un videojuego en el que además pueda aumentar sus conocimientos sobre la historia y sobre la ciudad misma de una forma ágil y divertida.

La idea que subyace tras el desarrollo de la aplicación se enlaza con los conceptos clásicos de los videojuegos, es decir, un punto de partida y un objetivo que alcanzar a través de unas acciones determinadas. Así, la gymkhana partirá de un punto inicial y, a través de un recorrido con diferentes puntos de control,

los participantes llegarán a un punto final en el que se indica el éxito o no éxito a la hora de cumplir el objetivo.

La Realidad Aumentada se mezcla con el videojuego en la propia ciudad. La idea es que ésta proporcione la información, que el participante necesita en forma de pistas y juegos de tal forma que se le facilite avanzar hacia el objetivo de una forma u otra dependiendo del éxito que tenga cumpliendo el objetivo.

Además, dicha información que proporciona la ciudad es sobre ella misma, de tal forma que se dé a conocer la historia, la sociedad, la gastronomía y otros conceptos de interés general, siendo ésta la forma que proponemos para fomentar el turismo tanto para la gente local como para cualquier visitante que quiera conocer más sobre la cultura andaluza y jienense.

He aquí una lista con los objetivos que se permiten con el desarrollo de este proyecto:

- Acercarnos brevemente a las nuevas tecnologías, en concreto, dar ciertas nociones sobre la Realidad Aumentada.
- Estudiar el estado del arte de los videojuegos.
- Definir el concepto de videojuego y de videojuego serio.
- Estudiar los elementos característicos de un videojuego.
- Estudiar las fases de desarrollo de un videojuego.
- Definir un motor de videojuegos y sus componentes.
- Realizar un estudio y comparativa de los diferentes motores de videojuegos actuales.
- Seleccionar un motor de videojuegos para el desarrollo de un prototipo.
- Adaptar las fases del desarrollo de un videojuego a nuestro propio desarrollo personal.
- Crear un prototipo de videojuego para una plataforma móvil.
- Incluir el uso de tecnología de Realidad Aumentada en el videojuego.
- Transmitir al jugador información educativa a través del videojuego.

1.3. Estructura del documento

Este documento ha sido estructurado en siete capítulos en los cuales avanzaremos en líneas generales en nuestro conocimiento sobre el mundo de los videojuegos y uno de sus componentes más importantes, el motor gráfico, para luego profundizar y aplicar los conocimientos adquiridos en un proyecto real desarrollando un prototipo de videojuego.

El orden de los capítulos es conforme a lo descrito anteriormente, empezando por una aproximación del tema que abordamos, realizando un análisis de términos, componentes y herramientas y aplicando los conocimientos extraídos en un proyecto de software de desarrollo de videojuegos.

Capítulo 1 – Introducción:

En primer lugar introduciremos el marco contextual del proyecto que realizamos, hablaremos sobre los motivos que nos llevan a realizarlo y enumeraremos los objetivos que queremos alcanzar con este proyecto.

Capítulo 2 – El estado del arte del videojuego:

En el segundo capítulo nos centraremos en el estado del arte del videojuego. Intentaremos dar una definición lo más ajustada posible del concepto de videojuego, así como analizaremos los elementos que componen un videojuego. También intentaremos encontrar las diferencias existentes entre un videojuego tradicional y un videojuego educativo o serio.

Capítulo 3 – El desarrollo del videojuego:

Aquí nos centraremos fundamentalmente en la etapa de desarrollo del videojuego. Veremos las diferentes fases que se producen en el desarrollo de este tipo de aplicaciones y las analizaremos brevemente para entender de qué se ocupa cada fase en específico.

Capítulo 4 – El motor de juegos:

El capítulo cuarto comienza con una descripción de los motores de videojuegos y sus componentes. Queremos intentar definir de la mejor manera posible tanto el término como dichos elementos que lo conforman. A continuación, realizaremos un análisis de los diferentes motores de videojuegos actuales, mostraremos sus puntos fuertes y débiles en cuanto al desarrollo de videojuegos se refiere y sobre todo en cuanto al soporte que ofrecen a la Realidad Aumentada.

Capítulo 5 – Desarrollo del prototipo:

En el capítulo quinto queremos reunir todos los elementos descritos con anterioridad y aplicarlos a un proyecto de desarrollo software, en concreto, un prototipo de videojuego serio con Realidad Aumentada. Desglosaremos cada fase de nuestro videojuego y analizaremos con detalle el sistema, los elementos del videojuego, el estilo visual y el funcionamiento e interacción con el usuario. Para finalizar, realizaremos algunas pruebas con nuestro prototipo y mostraremos los resultados obtenidos.

Capítulo 6 – Conclusiones:

En el penúltimo capítulo haremos una recopilación global de la trayectoria seguida en este documento, analizaremos los objetivos que nos planteamos y el progreso que hemos conseguido en los mismos, llegando a ciertas conclusiones sobre el proyecto que tratamos.

Capítulo 7 – Líneas de futuro:

El último capítulo intentará dar una visión global sobre el futuro de la aplicación, indicando posibles mejoras y sistemas que incluir en futuras versiones del prototipo hasta poder llegar a una versión final completamente funcional.

2. EL ESTADO DEL ARTE DEL VIDEOJUEGO

2.1. Introducción.

A través de los años, mucha gente ha catalogado a los videojuegos como una herramienta de ocio inservible en cualquier otro contexto que no sea el del puro entretenimiento. Han sido números los detractores de este tipo de expresión artística, pero poco a poco la sociedad empieza a entender lo que supone crear un elemento de estas características.

El videojuego, a día de hoy, está presente en muchos ámbitos de nuestra vida cotidiana. Se ha instaurado en nuestra sociedad como un elemento potente de distracción, pero también se ha hecho hueco en el mundo educativo. Y ha ido aún más allá, pues en los últimos años, éstos han inspirado obras tanto en el mundo del cine como en el mundo de la literatura, y viceversa, pues también beben de estos elementos para conformar referencias y obras diferentes, alternativas o recreaciones de las mismas.



Figura 1: Assassin's Creed en el cine y en los videojuegos. Fuente: Gamespot.

Poniendo un ejemplo reciente, uno de los últimos títulos desarrollados, conocido como Minecraft, y que es un juego de mundo abierto donde, en palabras de los desarrolladores, “el cielo es el límite”, ha desarrollado una versión educativa para colegios y escuelas, pues se descubrió que el título original que estaba pensado para fomentar la construcción libre, desarrolla habilidades en el ámbito de la arquitectura, la mecánica, la física, la ingeniería, la creatividad y la visión espacial en aquellos que lo juegan.

Es más, este título es capaz de incluir diferentes modificaciones de la comunidad que convierten el juego en un mundo relativamente ilimitado. Esto junto con que su público objetivo es niños y adolescentes de corta edad, supone una combinación excelente para el aprendizaje. Además, las empresas han reconocido el producto como elemento fundamental de un modelo de negocio que genera beneficios por valor de miles de millones de dólares anuales en todas las plataformas disponibles.



Figura 2: Asociación de Minecraft y arquitectura. Fuente: lineshapespace.com

EVOLUCIÓN SW CONOSOLAS (VALOR Y UNIDADES)

SOFTWARE	AÑO 2013			
	UNIDADES	12/13	VALOR	12/13
	En miles	Unidades	Millones €	Valor
VIDEOJUEGOS PARA CONSOLA	9.837	-12%	381	-4%
PS4	240		15	
PS3	4.050	3%	180	5%
PS2	92	-68%	1	-64%
PSP	594	-55%	8	-57%
PSVITA	372	21%	13	2%
XBox360	1.253	-13%	47	-16%
Xbox One	60		4	
Wii	1.397	-29%	50	-30%
Wii U	211	306%	10	262%
DS	302	-70%	7	-75%
3DS	1.267	45%	46	42%

SEGMENTACIÓN EN VALOR 2013

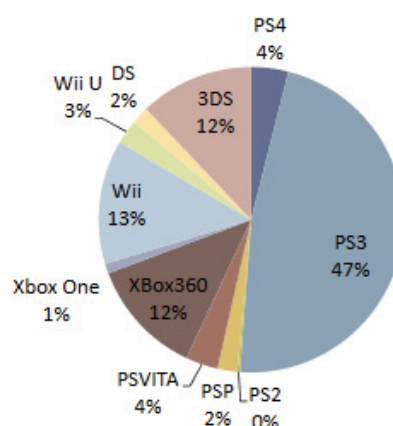


Figura 3: Evolución de los videojuegos para consola (2013). Fuente: ADESE

El arte con mayúsculas, el que canon de estar en un museo, empieza ahora a envidiar a su hermano pequeño, el arte del mundo de los videojuegos. Y es que este tipo de obras, que precisan de una recreación artística al nivel de muchas obras en lienzo, por no decir que algunas incluso las superan, es a día de hoy el sector artístico que genera mayores beneficios. El arte y la creatividad se desatan en este tipo de obras, no sólo en las recreaciones artísticas a nivel de diseño (niveles, personajes, expresiones, etc.), sino a nivel histórico o incluso a nivel imaginativo a la hora de programar o enlazar diferentes componentes.

En este marco, destaca la expresión personal individual, que ha creado figuras sumamente influyentes en este mundo, personas que han sido reconocidas internacionalmente por la comunidad y que ha recibido premios tales como el Premio Príncipe de Asturias de Comunicación y Humanidades. Por citar algunos de ellos, Hideo Kojima o Shigeru Miyamoto, creador de uno de los personajes e iconos más potentes y representativos de los últimos años, Mario.



Figura 4: Shigeru Miyamoto junto a su creación más famosa. Fuente: relax.mx.

Observamos como un programa informático que se ideó para el puro entretenimiento empieza a expandir sus raíces entre las diferentes disciplinas y ámbitos de la sociedad moderna, se consolida y crece como una fuente inagotable de transmisión de elementos recreativos y educativos proporcionando un medio de comunicación audiovisual que a día de hoy no tiene competidor.

2.2. El concepto de videojuego.

Definir un videojuego puede ser una tarea complicada, debido a que ciertos aspectos del mismo son subjetivos. No se puede tener, a priori, una idea clara de la cantidad de elementos que debe contener ni tampoco de la forma de incluirlos en éste, pero sí que hay ciertos aspectos que son claramente identificables y que nos permiten tener no sólo una visión global de un juego, sino acercarnos de manera más o menos precisa al concepto.

Según Jesper Juul, un videojuego es un “*sistema basado en reglas con una respuesta variable y cuantificable en el que se le asignan diferentes valores a los diferentes efectos, en el que el jugador realiza un esfuerzo por influir en estos efectos, tiene un apego emocional a los efectos y las consecuencias de dicha actividad son opcionales y negociables.*” (Juul, 2003).

En otras palabras, un videojuego es un sistema con reglas y objetivos con el que interacciona un jugador. Esta definición es amplia, pues como ya hemos mencionado, definir este concepto no es algo sencillo. La simple idea de acercarse a conocer a qué efectos y esfuerzos se refiere es muy subjetivo.

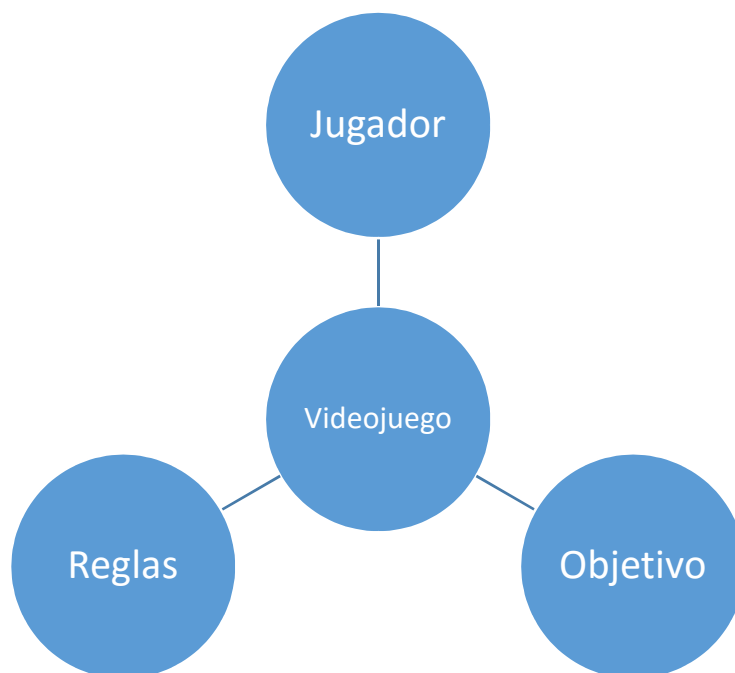


Figura 5: Visión esquematizada del videojuego. Fuente: elaboración propia.

Frank Lantz y Eric Zimmerman afirman de los juegos que *“para los antropólogos, los juegos son artefactos enredados entre la materia prima de una cultura particular. Para los psicólogos, los juegos son instrumentos posibilitadores en la narración del desarrollo personal. Entre los teóricos matemáticos, los juegos emergen como evidencia de las intrincadas estrategias y los espinosos dilemas de la teoría de juegos, mientras que los científicos usan modelos similares a los juegos para aproximarse a sistemas del mundo real. Ni que decir tiene que el lenguaje de los juegos proporciona muchas metáforas posmodernas de elección.”* (Lantz; Zimmerman, 1999).

Esta definición de juego puede extrapolarse claramente al mundo de los videojuegos, pues qué otra cosa es sino un juego virtual. Como vemos otra vez, la subjetividad está presente en el término.

Una definición algo más sencilla la podemos encontrar de la mano de David Ballejo y Cleto Martín: *“un videojuego es una aplicación gráfica en tiempo real en la que existe una interacción explícita mediante el usuario y el propio videojuego.”*(Vallejo; Martín, 2012).

Por otra parte, Jesse Scheel indica que es *“la combinación de cuatro elementos relacionados entre sí que conforman la experiencia del usuario con el juego y que son las mecánicas, la historia, la estética y la tecnología.”* (Scheel, 2008).

Insistimos en que no hay una definición precisa del concepto de videojuego, pero si observamos a estas alturas que todas las definiciones tienen elementos en común. El videojuego es un sistema o aplicación que está diseñada para interactuar con un jugador (aunque no debemos limitar la interacción con una única persona, pues pueden ser varias) mediante ciertos elementos que además están relacionados entre sí.

En este caso, Jesse Scheel se aproxima más al concepto generalizado de videojuego porque define cuáles son esos elementos que, en mayor o menor medida, deben estar presentes en el sistema.

Dejando de lado la tecnología, pues creemos que todos los videojuegos contienen los mismos elementos independientemente de la tecnología, los otros tres elementos son los pilares básicos del videojuego, y son éstos los que vamos a detallar a continuación.

2.3. Serious Games frente a los videojuegos tradicionales.

Para finalizar este capítulo, hay que hablar sobre una pequeña línea que divide al mundo de los videojuegos. Cuando hablamos de ellos, nos surgen dudas sobre cuál es el propósito del mismo, qué objetivo quiere cumplir sobre las personas que lo juegan. ¿Es acaso un videojuego un sistema puramente enfocado al ocio, o intenta transmitir una información menos trivial? ¿Intentan los videojuegos que aprendamos? He aquí, gracias a estas preguntas, donde surge la división.



Figura 6: Microsoft Flight Simulator (Serious Game). Fuente: news.softpedia.com

Un videojuego, como ya hemos comentado, no tiene por qué intentar transmitir una información que sirva a modo de historia, aunque sí se sitúa en un contexto determinado. Razonando de la misma manera, un videojuego no tiene por qué estar puramente orientado al ocio. Si bien es cierto que los juegos están catalogados como una herramienta puramente de diversión, diversos estudios demuestran que el aprendizaje se refuerza a la hora de jugar a un videojuego.

Entonces, ¿qué ocurre cuando mezclamos los conceptos de juego y aprendizaje? La respuesta es los juegos serios o “serious games”.

El término ya fue acuñado en 1970 por Clark Albert, quien afirma que “estos juegos están explícita y cuidadosamente pensados para fines educativos y no están destinados a ser jugados principalmente por diversión.” (Albert, 1970).



Figura 7: Serious Game. Fuente: Jordane Pâquet (2010)

Hay que precisar en esta afirmación que ocio (específicamente, su componente recreativo) y serious game no son excluyentes. Se puede realizar un videojuego de ocio cuyo trasfondo sea enseñar algo a través del juego, es decir, se puede aprender divirtiéndose. En este contexto, aparece una definición que se ajusta mucho más al objetivo de este proyecto:

“Una prueba mental, de acuerdo con unas reglas específicas, que usa la diversión como modo de formación gubernamental o corporativo, con objetivos en el ámbito de la educación, sanidad, política pública y comunicación estratégica.”

(Zyda, 2005)

Como vemos, Zyda propone un modelo de juego en el que la diversión es el canal para transmitir cierto tipo de formación o, más ampliamente dicho, de conocimiento en diferentes ámbitos.

Nuestra opinión va un paso más, pues somos de la creencia de que un serious game (y a grandes rasgos un videojuego) no solo transmite, sino que potencia habilidades importantes como la creatividad y refuerza otros aspectos psicológicos como la autoestima y la motivación.

Las cuestiones anteriores ahora quedan reducidas a un simple problema de enfoque, es decir, hacia dónde queremos conducir nuestro videojuego, o mejor dicho, qué queremos transmitir con él. Ahora sabemos que tenemos la posibilidad de crear un sistema que refuerce el conocimiento de quién lo juega de forma ágil y divertida, que enmascara el proceso de aprendizaje detrás de una aplicación que a priori está enfocada a fines recreativos.

2.4. Componentes de un videojuego

En el anterior subcapítulo hemos comprobado que son muchos los aspectos a tener en cuenta en un videojuego, pero básicamente podemos definir tres aspectos primordiales incluidos en todos ellos. En un juego nos encontramos una información que quiere ser transmitida, que puede ser una historia o no, unos objetos con cierta apariencia que se presentan en una escena concreta, y una manera de interacción entre los diferentes elementos y entre el usuario y los elementos del videojuego.

Así, podríamos agrupar la mayoría de componentes de un videojuego en tres bloques:

- **Historia:** Este bloque englobaría la información que presenta el juego al usuario. La historia puede ser una historia plana o con profundidad y su objetivo puede ser sencillo, como ponernos en una situación, entregarnos un problema y ayudarnos a resolverlo, o podría ser algo muy complejo en lo que se intenta transmitir un mensaje profundo o

presentar, propiamente dicho, una historia que podría compararse a las que encontramos en cualquier novela. Aquí podríamos englobar todos los elementos orientados a transmitir información y la forma en la que se transmite la misma y aquellos componentes que, en cierta manera, forman parte del ambiente que se construye alrededor de un videojuego.

- **Arte:** En este bloque se encuadran la gran mayoría de aspectos del juego, incluyendo los conceptos de qué se ve y cómo se ve. Escenas, personajes, texturas, luces, efectos de partículas, y un largo etcétera tienen cabida en este componente. Dicho de otra manera, todos los aspectos que de alguna manera interactúan con nuestros sentidos de forma directa. Y mencionando esto, es obvio que la parte de sonido también se encontraría aquí, ya sea la música que acompaña a la escena o el sonido ambiente que rellena y construye el mundo alrededor de nosotros. El arte se relaciona con la historia, pues permite de alguna manera que el jugador reciba información de forma indirecta y que completa los aspectos que no son fundamentales de la historia pero que aportan hasta cierto punto una cantidad de información necesaria.
- **Mecánicas y jugabilidad:** Este último componente abarca aquellos aspectos relacionados con los procedimientos del juego y sus reglas. Definen acciones tanto de personajes y objetos como de la propia escena, incluyendo interacciones entre elementos, así como aspectos de qué es posible e imposible en el juego, objetivos, condiciones de éxito y fracaso y, en general, qué nos permite hacer el juego y qué no permite. Y no sólo queda ahí, pues además de definir qué puede realizar el usuario con el título, también define cómo puede interactuar con éste, teniendo en cuenta aspectos de controles y periféricos. Esta parte está relacionada ampliamente con la parte de arte y, por consiguiente, con la de historia, pues podemos considerar la inmersión como un aspecto fundamental de la jugabilidad y que depende enormemente de la parte artística en tanto a cómo se integra el jugador en el mundo.

En una aproximación general, podemos destacar y englobar las partes fundamentales de un videojuego en los bloques mencionados anteriormente sin dejarnos demasiadas cosas fuera de contexto. La historia nos pone en situación en un juego, nos plantea un objetivo y nos guía a través del mismo, el arte nos hace visualizar la historia, nos completa la información de forma indirecta y nos sumerge en el mundo que se quiere construir, y la jugabilidad nos permite explorar e interactuar con dicho mundo en base a unas ciertas reglas para alcanzar nuestro objetivo.

3. EL DESARROLLO DEL VIDEOJUEGO

3.1. Introducción

Videojuego es un término muy amplio en todos los sentidos, e incluso va más allá, pues éste supone una forma de expresión de conceptos e ideas por parte de una o varias personas y una forma de ilustrarlas, generalmente de forma visual, para el resto. Cada elección es sumamente importante y afectará en mayor o menor medida al resultado final, pero son los elementos estructurales básicos los que definen al propio videojuego, es decir, aquellos elementos que sientan las bases del mismo. Hablamos de elecciones como la historia, el arte conceptual o las mecánicas de los diferentes elementos que lo componen y que permiten la interacción con el usuario final, sin olvidarnos de la elección de la plataforma o el público objetivo al que dirigir el producto.

Además, desarrollar un videojuego es una tarea muy complicada a pesar de lo que parece. Abordar un proyecto de desarrollo de un videojuego es abordar en toda regla un proyecto de desarrollo software en el que a veces hay que incluir conceptos abstractos que a día de hoy no están del todo definidos y de los cuales la forma de recrearlos y representarlos no tiene por qué ser única. Sólo a través de diferentes fases de desarrollo, bien definidas, se puede alcanzar un producto final que puede definirse como videojuego, siempre y cuando se tengan claros los fundamentos del mismo, es decir, los elementos básicos que lo componen. No vamos a entrar en el impacto que puede tener el videojuego tanto a nivel técnico como social, pues son varios los factores subjetivos que influyen en estos casos, pero sí que debemos tener en cuenta que a la hora de desarrollarlo hay ciertos aspectos que son fundamentales, que están perfectamente definidos y que indican exactamente qué es o lo qué debe contener un videojuego, así como la forma de conseguir implementar dichos aspectos.

Aparte de esta estructura fundamental del videojuego, hay otros conceptos, como ya hemos indicado anteriormente, que son variables y que dotan al mismo de un punto de diferenciación del resto. Son ideas que intentan que el producto final sea algo único, que exprese de forma adecuada lo que el autor o autores

quieren mostrar (ya hemos indicado que, a fin de cuentas, el videojuego es en sí una pieza de arte) y, por último, que atraiga al público objetivo hacia el producto.

Vamos a realizar una visión general a cada uno de estos aspectos y analizaremos sus componentes, sus etapas de desarrollo y las diferentes elecciones que puedan afectar en mayor medida al resultado final.

3.2. Fases de desarrollo de un videojuego.

En el desarrollo de un videojuego, el proceso de producción del mismo consta de una serie de fases claramente identificables y definidas y que abarcan todo el proceso completo, desde la concepción de la idea hasta la obtención del sistema final. Vamos a tratar de abordar cada una de ellas haciendo un pequeño análisis y describiendo qué elementos engloba o qué elementos se desarrollan en éstas.

Para empezar, podemos realizar una aproximación global en la que se pueden identificar tres grandes bloques estrechamente relacionados entre sí. El primero de ellos es un bloque de preproducción que tiene como idea sentar las bases del videojuego para, en base a las mismas, construir el sistema deseado.

En esta primera fase, por describirlo de manera sencilla, se construyen las bases del videojuego como si construyésemos los cimientos de un edificio. Para ello, se parte de una idea o concepto sencillo y se intenta dar forma a estas bases en función de ella, a saber:

- La jugabilidad: “directrices que indican cómo implementar los elementos necesarios (como las reglas) para dar nacimiento a una especie de tiempo de juego o un entretenimiento social deseado.” (Järvinen; Heliö; Mäyrä, 2002). La jugabilidad es, por tanto, un conjunto de elementos que se ofrecen al usuario para facilitar su uso y que describen la experiencia del mismo en nuestro sistema. Aquí se puede definir qué acciones puede realizar el usuario, cómo se transforman los elementos en función de determinadas acciones, cómo recibe el jugador la información, etc.

- Concepción y diseño: En esta parte se elabora un sencillo storyboard con aquellos elementos que deben formar parte de la historia (o de lo que se pretende transmitir al jugador), así como se realizan dibujos preliminares de las diferentes escenas, personajes, objetos y otros elementos, se conforma superficialmente la música y voces del título a desarrollar y se definen otros aspectos que tengan características similares y se puedan enmarcar en esta fase, como la ambientación del juego.
- Género del juego: Hay que definir también el tipo de juego a realizar, es decir, en qué lugar se categoriza nuestro título dentro del mundo de los videojuegos, ya que este aspecto afecta en gran medida al modo en que interactúan las entidades dentro del mismo, es decir, que afectan a la mecánica del juego (al conjunto de reglas para alcanzar los objetivos que hemos descrito con anterioridad), y por tanto, a la jugabilidad.

Con estas tres fases descritas queda cerrado el primero de los bloques que conforman el desarrollo del videojuego y que sientan las bases de éste. El segundo bloque es más extenso que el anterior y que, si bien el primero de ellos sentaba los cimientos del sistema, éste tiene como objetivo construir el videojuego como tal.

Siguiendo con la analogía anterior, digamos que en esta fase vamos a levantar nuestro sistema como si de un edificio se tratase, aunque esto no quiere decir que una vez finalizada la fase nuestro sistema sea el producto final, pues el objetivo es construir una versión inicial o prototipo a mejorar en el futuro. Ana María Manrubia define en este bloque las siguientes fases:

- Diseño del juego: Se especifica claramente el diseño de los elementos del juego al equipo de desarrollo, añadiendo profundidad a los conceptos anteriormente descritos.
- Diseño artístico: Se conforman los elementos visuales y ambientales, así como la historia del juego. Se desarrollan las historias de los personajes y se crea el hilo argumental, así como se

diseñan los elementos sonoros del videojuego, se recrea la interfaz y la interacción del usuario con el juego (GUI y HUD), y se diseñan los elementos gráficos del sistema, que dependiendo del tipo de juego que sea, serán elementos 2D o 3D.

- **Diseño mecánico:** Se marcan las pautas de interacción con el juego, se definen las reglas del sistema, se construye la inteligencia artificial de los elementos controlados por el videojuego y se les añaden comportamientos y animaciones.
- **Motor de juego:** Se debe escoger un sistema de rutinas que permita representar todos los elementos. Este componente permite definir cómo se interactúa y cómo se representan dichos elementos.
- **Diseño técnico:** Aquí el videojuego se acerca al mundo informático y permite que el sistema se trate como un verdadero proyecto de software. Aquí se generan los flujos que describen el funcionamiento del videojuego y los diferentes estados que atraviesa.
- **Implementación:** Ahora se reúnen todos los elementos descritos con anterioridad y se encajan como si se tratase de un sistema de engranajes que beben del motor gráfico y se lleva a cabo una etapa de perfilado y corrección de errores crucial de cara al resultado final. A partir de aquí empieza lo que se conoce como fase de ajuste fino.

Así, nos encontramos ya en la última fase del desarrollo del videojuego, y no por ello la menos importante, pues cabría afirmar que ésta es posiblemente una de las fases que más influyen en el producto final. Este bloque es conocido como bloque de ajuste fino o fase de pruebas.

Con todos los elementos funcionando correctamente, se realizan pruebas para identificar errores y mejorar características de jugabilidad que no se han contemplado con anterioridad o no se han conseguido. Estas pruebas se realizan primero por miembros del equipo que ha participado en alguna fase del desarrollo del videojuego y luego por miembros externos al proyecto, y cuando finalizan el juego no debe contener ningún error crítico.

A partir de este momento, se realizan los últimos ajustes para intentar alcanzar el mejor resultado posible y obtener un producto final completo de acuerdo a las especificaciones de la fase de preproducción, finalizando el desarrollo del videojuego, que queda listo para su salida al mercado.

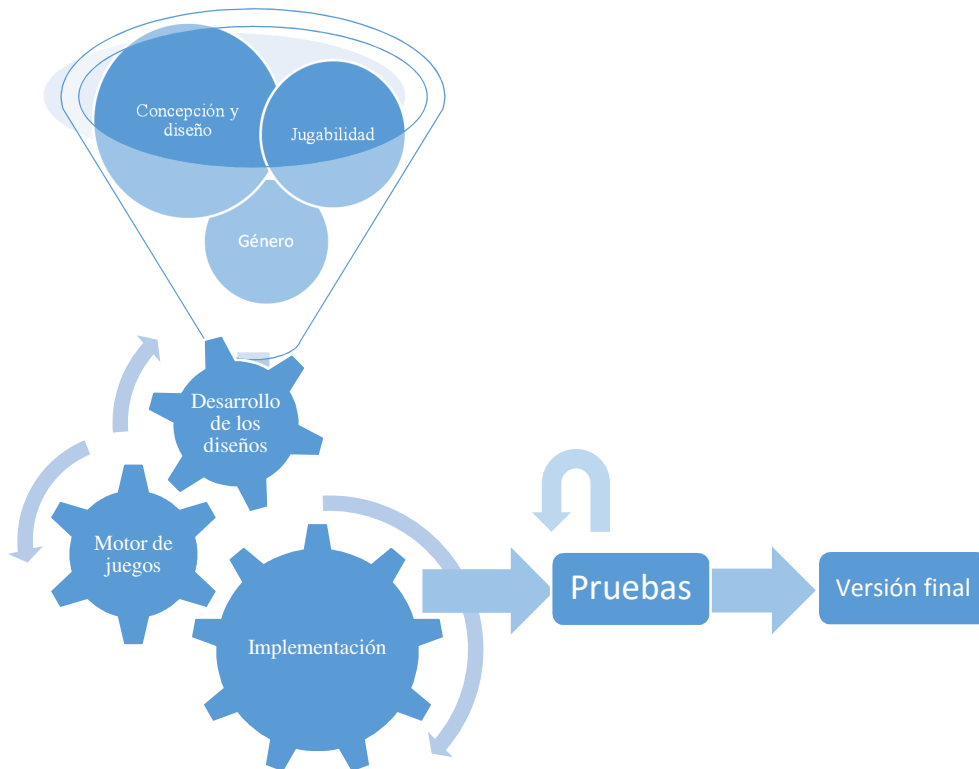


Figura 8: Etapas del desarrollo del videojuego. Fuente: elaboración propia.

4. EL MOTOR DE JUEGOS

4.1. Introducción.

La elección de un motor de juegos a la hora de desarrollar un videojuego es crucial cuanto menos. De éste depende el tipo de reglas que podemos establecer, los objetivos a realizar, los elementos a representar y, en general, todo aquello que interviene en el juego y cómo interaccionan estos elementos entre sí. Y, por supuesto, también define la forma en la que el usuario interacciona con el sistema, le muestra información y lo guía a través del proceso de resolución de las diferentes pruebas propuestas. En definitiva, el videojuego no existe sin el motor de juegos.

La elección es subjetiva, pues los motores de juego hoy en día ofrecen una enorme cantidad de soluciones y elementos para el desarrollo del videojuego. Debemos intentar ajustarnos lo mejor posible a las características que necesitamos implementar en nuestro sistema y buscar aquel motor que nos ofrezca las mejores soluciones.

Tampoco hay que dejar de lado el apartado artístico, pues como ya hemos visto, es una pieza fundamental del juego. El motor de videojuegos debe proporcionarnos un buen soporte para el audio y elementos gráficos que recrearán el mundo que queremos construir.

Es importante entonces realizar un pequeño estudio sobre los motores de videojuegos y las características que ofrecen para poder tomar una buena decisión de cara al prototipo que queremos desarrollar. En este caso, nos centraremos en buena medida en analizar qué tipo de soporte dan estos motores a la Realidad Aumentada, que es el pilar básico sobre el que se mueve nuestro sistema, pero sin dejar de lado otros aspectos importantes que formen parte del videojuego.

4.2. El motor de juegos. Características principales.

Ya hemos indicado que el motor de juego es una de las piezas fundamentales a la hora de desarrollar videojuegos y sin ella, simplemente no sería posible crear un sistema como tal. El motor de videojuegos es una colección de un motor de gráficos, un motor de sonido, un motor gestor de inteligencia artificial y un motor de físicas, además de otro tipo de gestores que se consideren necesarios.

Los motores de juego inicialmente tenían código preparado para ejecutar un único juego, pero luego evolucionaron para ejecutar partes en común de diferentes juegos, aumentando su flexibilidad hasta la actualidad. Solían ser desarrollados por las propias compañías de videojuegos y, además, los motores de juego solían ser cerrados, esto es, que no estaban pensados para que gente externa a la compañía los utilizase. Con el tiempo, los motores adquirieron un propósito general, reduciendo su código y volviéndose genérico y reutilizable, y empiezan a ser vistos como un sistema que permite desarrollar varios tipos de videojuegos.

Cualquiera sea el caso, Jason Gregory describió los motores de juego y propuso en su tiempo una arquitectura por capas que puede seguirse como referencia, aunque no es estricta, y que a día de hoy sigue siendo válida. Una visión general a los componentes del motor de juegos sería la siguiente:

- En la capa más baja se encuentra el hardware, que define un catálogo de plataformas de forma transparente para poder adaptar el juego a nuestras necesidades.
- Los drivers son la siguiente capa y actúan entre la capa del sistema operativo y del hardware.
- Encima del sistema operativo tenemos la capa de 3rd parties SDK, que interactúan a bajo nivel en ciertas características y que pueden servir como base para elementos de capas superiores.
- La siguiente, Platform Independence Layer, es una de las más importantes. Permite que todo lo que viene de capas superiores y que son cosas fundamentales se abstraigan. Además, también

abstrae las capas inferiores, como el tema de librerías de gráficos. Así se permite una gran portabilidad de los juegos. Esto no quiere decir que el juego funcione igual en todas las plataformas, pero simplifica mucho las cosas.

- Por encima encontramos los sistemas básicos para debugging, aspectos de gestión matemática o de memoria, etc. Forma nuestros sistemas básicos que utilizaremos a continuación, siempre basados en la plataforma independiente.
- Seguidamente encontramos la capa de recursos, capaz de gestionar diferentes activos del juego, bien sean propiedades, modelos, esqueletos, etc. Se distingue entre recursos que dispones en el juego y recursos activos en la escena. Esta parte consume buena parte de la memoria.
- Sobre los recursos se encuentran varios elementos que definen aspectos como la parte gráfica (también segmentada por capas, incluyendo partes de rendering, grafos de escena, efectos visuales y la parte del front-end), la parte de físicas y colisiones (que engloba todos los aspectos de propiedades físicas de los objetos de la escena), la parte de animación, interacción y dispositivos de entrada, la parte de audio, y el grupo de fundamentos de gameplay, que engloba aspectos de la jugabilidad como las mecánicas, objetivos de éxito y fracaso y, en general, aspectos que determinan el juego.
- En la parte alta encontramos la capa de subsistemas específicos de juego, que cuenta con aspectos como la IA, cámaras de juego, mecánicas, etc.

Como vemos, el motor es un sistema software que dispone de una colección de herramientas de edición. Algunos de ellos permiten crear juegos sin necesidad de programar, mientras que otros tienen su propio lenguaje de programación y refuerzan la creación de contenidos a base de scripts. A día de hoy, un motor de videojuegos se basa en la clasificación que hizo Jason Gregory e implementa estas capas en mayor o menor medida para simplificar el proceso de creación de videojuegos, haciendo que nos centremos en los aspectos fundamentales e interactuemos sobre todo con las capas más específicas.

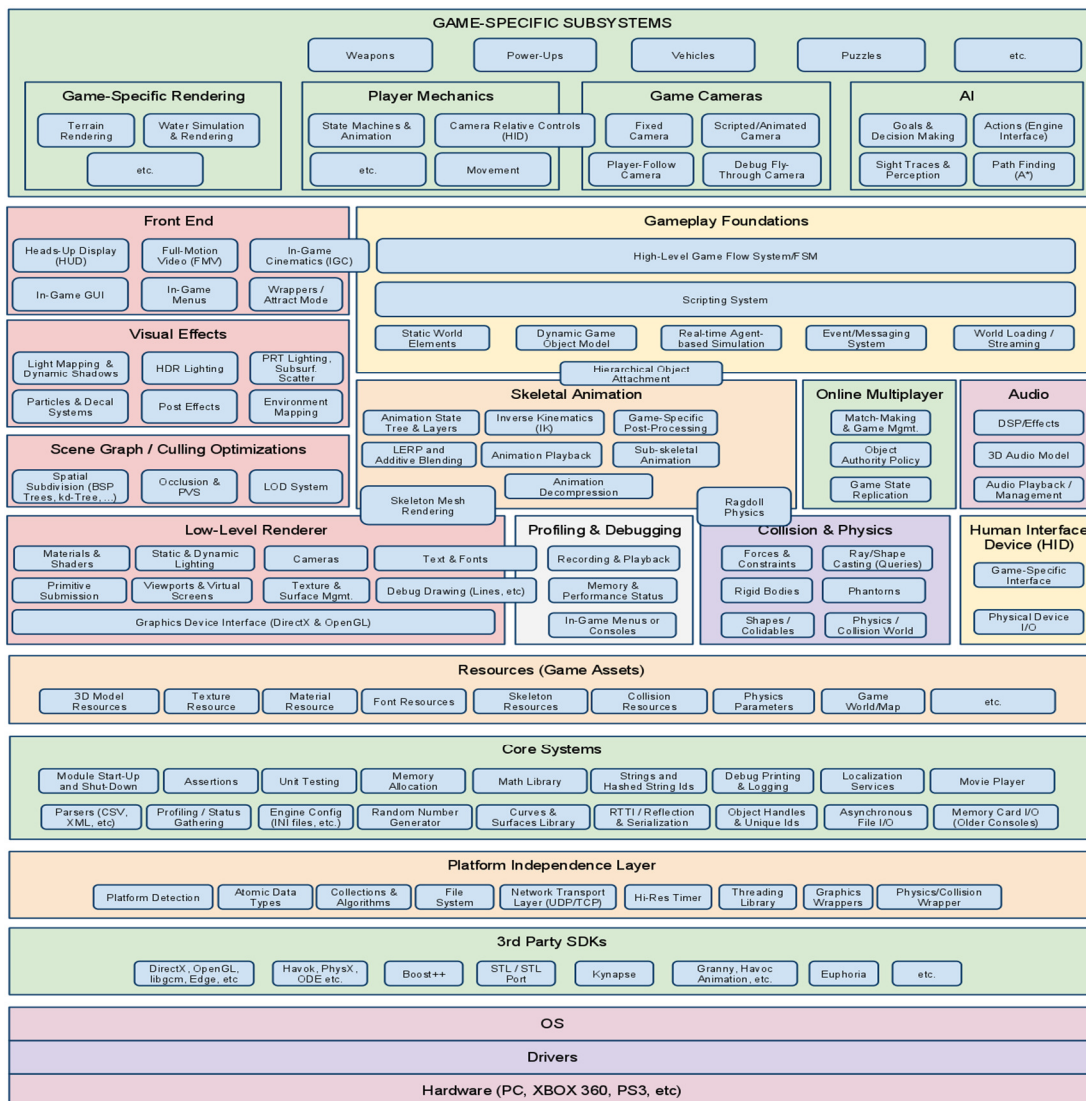


Figura 9: Game Engine Architecture. Fuente: Gregory, J. (2009): Game Engine Architecture.

4.3. Estudio sobre motores de juegos actuales.

Las ventajas de los motores de juegos son tangibles y realmente útiles. Un motor bien construido potencia aspectos como la reusabilidad, para utilizar el mismo software en diferentes sistemas; la portabilidad, que da soporte a la multiplataforma y permite una abstracción del hardware; o menores costes de desarrollo, facilitando el trabajo y consiguiendo que el producto final llegue al mercado de forma rápida. Un buen motor, como mínimo, debe presentar estas ventajas en mayor o menor medida.

No obstante, como ya se ha comentado anteriormente, los motores no tienen por qué ofrecer los mismos componentes y herramientas. Es digno de

estudio entonces el entorno de los motores de videojuegos, qué características presentan y que prestaciones tienen con respecto a otras posibles soluciones y, en base a este análisis, poder tomar una decisión consecuente con el sistema que queremos desarrollar. Debemos intentar ajustarnos a los requisitos de nuestro videojuego de la mejor forma posible, pues si no tomamos una buena decisión, el resultado es una mayor cantidad de tiempo de desarrollo (bien sea por tratar de entender una gran cantidad de componentes y herramientas en el entorno de trabajo, bien sea por la necesidad de buscar herramientas externas que realicen tareas que no puede realizar el motor elegido), una mayor cantidad de dinero invertido, o un problema de rendimiento del juego a la hora de ejecutarlo. De esto deducimos que, generalmente, no hay un motor que sea mejor y otro que sea peor, simplemente un motor que se adapta mejor a nuestro problema y otros que no se adaptan tan bien.

Dicho todo esto, vamos a realizar un pequeño estudio sobre los diferentes tipos de motores gráficos que existen a día de hoy. Queda fuera de nuestra competencia analizar motores de videojuegos pasados que no ofrecen soluciones factibles a día de hoy, ya que nuestro objetivo final es poder escoger uno que nos permita desarrollar nuestro sistema. También vamos a poner especial interés en comprobar que soluciones ofrecen al problema de la Realidad Aumentada, pues nuestro videojuego utilizará dicha tecnología como pilar fundamental para el desarrollo del mismo, y es importante que el motor que escojamos le dé soporte.

V·T·E		Game engines (list)	[hide]
Source port · First-person shooter engine (list) · Tile engine · Game engine recreation (list) · Game creation system			
Free and open-source	2D	Adventure Game Studio · Beats of Rage · Cocos2d · Flixel · Jogle · KiriKiri · libGDX · Moai · OpenFL · ORX · Pygame · RenPy · Stratagus · Thousand Parsec · VASSAL · Xconq	
	2.5D	Aleph One · Cube	
	3D	Away3D · Blender Game · Cafu · Crystal Space · Cube 2 · Delta3D · Dim3 · Goo Create · GLScene · Horde3D · HPL 1 · Irrlicht · id Tech 3 · id Tech 4 · JMonkey · OGRE · Open Wonderland · Panda3D · Papervision3D · Platinum Arts Sandbox Free 3D Game Maker · PlayCanvas · PLIB · Quake · Quake II · RealmForge · Retribution · Torque 3D	
	Mix	Allegro · Construct Classic · Godot · Lightweight Java Game Library · Spring · Wintertime Engine	
Proprietary	2D	Construct 2 · Corona · Clickteam Fusion · GameMaker: Studio · GameSalad · M.U.G.E.N · NScripter · RPG Maker · Southpaw · Stencyl · UbiArt Framework · Vicious · Virtual Theatre · VPlay · Zillions of Games	
	3D	4A · Amazon Lumberyard · Anvil · Bork3D · C4 · Chrome · Clausewitz Engine · Creation · CryEngine · Crystal Tools · Diesel · Dunia · EAGL · EGO · ElfFlight · Enforce · Enigma · Essence · Flare3D · Fox · Frostbite · Geo-Mod · GoldSrc · HeroEngine · HydroEngine · HPL 2-3 · id Tech 5 · id Tech 6 · Ignite · IW · Jade · Kinetic · LS3D · Leadwerks · LithTech · Luminous Studio · LyN · Marmalade · Mizuchi · MT Framework · Outerra · Panta Rhei · PhyreEngine · Q · Real Virtuality · REDengine · Refractor · Riot · RAGE · SAGE · Serious · Shark 3D · ShiVa · Silent Storm · Source · Titan · TOSHI · Truevision3D · Unigine · Unity · Unreal · Vision · Visual3D · XnGine · X-Ray · YETI · Zero	
	Mix	Gamebryo · Hybrid Graphics · Kaneva Game Platform · Metismo	
Historical interest		BRender · Build · Dark · Doom · Game-Maker · GameMaker · Garry Kitchen's GameMaker · Genesis3D · Genie · Filmmation · Freescape · INSANE · Jedi · MADE · Pie in the Sky · RenderWare · Sim RPG Maker · Sith · Voxel Space · Wolfenstein 3D	
Proprietary middleware (list)		Euphoria · Gameware · GameWorks · Havok · iMUSE · Kynapse · Quazal · SpeedTree · Xaitment · FaceGen	

Figura 10: Listado de motores de videojuegos. Fuente: www.wikipedia.org

4.3.1. CryEngine 3 SDK

La primera solución de motor de videojuegos llega de la mano de CryTek. CryEngine 3 no es ni el primero ni el último de la familia, pero es la versión libre más reciente hasta la fecha. Éste es un motor principalmente orientado a la educación y dispone de una versión no comercial para el aprendizaje por parte de escuelas y universidades, aunque cuenta con una licencia de desarrollo independiente en la que se paga un 20% de los beneficios del título en derechos de autor. Otro de los objetivos que tiene es el de crear una comunidad de desarrollo.

La fama de este motor de videojuegos viene de la mano de la potencia gráfica. CryEngine y, en general, los motores de CryTek suelen ser contar con motores gráficos muy potentes, con aspectos muy desarrollados como las físicas, los sistemas de animación de modelos y la capacidad de iluminar escenas en tiempo real. No hay muchos motores capaces de igualar la potencia de los motores de CryTek, aunque en concreto la versión 3 de CryEngine no es tan superior como la de su hermana mayor, el CryEngine4. Históricamente, sólo Unreal Engine ha sido capaz de mostrar una potencia en el motor gráfico similar.

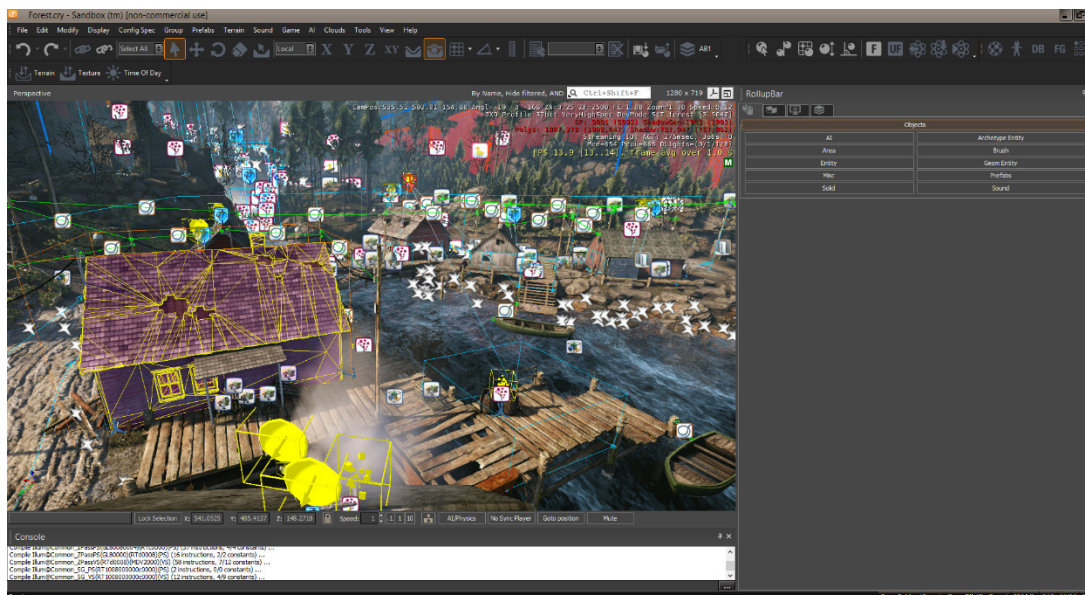


Figura 11: CryEngine3. Fuente: ce3sdk.wikia.com/wiki/CryEngine_3_SDK

En concreto, CryEngine 3 destaca principalmente por su capacidad de generar el mundo del videojuego, sobre todo en videojuegos no lineales (conocidos por su término inglés “sandbox”). El editor de este motor cuenta con una suite de herramientas y sub-editores enfocados específicamente al desarrollo de sandbox. Se puede pensar en el editor de CryEngine 3 como una herramienta de composición que actúa de forma similar a una herramienta de edición de audio o video.

Entre las características a destacar de CryEngine 3 se encuentran las de la siguiente lista:

- “What you see is what you play”, sistema con la característica de reproducir todos los elementos y características del motor de forma inmediata sin excepción y de forma simultánea en todas las plataformas.
- Auto-actualización multiplataforma, que permite actualizar y comprimir los recursos disponibles desarrollados para una plataforma cuando se cambia a otra diferente y de forma automática.
- Gráficos de nueva generación tanto para ambientes interiores como exteriores, con elementos generados en tiempo casi real o real para la mayoría de ellos.
- Grafos de flujo para el control de eventos y la lógica del juego.
- Sistema de generación de terrenos y vegetación integrada, con comportamientos naturales y fluidos y en el que se incluye un sistema de destrucción de terrenos, sistemas de partículas con detección de colisiones.
- Herramientas de recreación de caminos y canales de forma automática.
- Herramientas de creación de vehículos con controles intuitivos y físicas realistas.
- Soporte multi-núcleo con compatibilidad con los nuevos procesadores del mercado.
- Iluminación global dinámica e iluminación diferida en tiempo real.
- Iluminación natural, niebla volumétrica y sombras suaves dinámicas.
- Soporte de mapas de normales y mapas de oclusión de paralaje.

- “Screen Space Ambient Occlusion” que proporciona una simulación real de oclusión de luz ambiental.
- Tecnología uber shader con capacidad de incluir shaders específicos y combinarlos con otros para obtener diferentes resultados y con capacidad para soportar iluminación por píxel.
- Sistema de personajes realista, que cuenta con un sistema de animación e individualización de personajes y esqueletos, así como un editor facial dedicado.
- Sistema de inteligencia artificial modular avanzado con capacidad para editar IA amistosa, búsqueda de ruta dinámica y generación de mallas de navegación automáticas.
- Sistema de físicas de alto rendimiento capaz de generar flujos de luz volumétricos, fluidos en 3D y ambientación natural realista.
- Sistema de físicas de deformación de objetos debido a otros componentes.
- Herramientas de análisis de rendimiento.
- Sistema de capas para el diseño.
- Compilador de recursos.
- Creador, mezclador y editor de parámetros de música y audio.
- Sonidos dinámicos y música interactiva, que permite que las fuentes de audio se modifiquen en tiempo real en función de la distancia o de otros factores como atenuación o ruido.
- Scripting en C++, C# y LUA.

Como vemos, la mayoría de elementos de CryEngine están enfocados al motor gráfico, dando un peso aún mayor a la creación de terrenos y de elementos del mismo. Cuenta con una gran cantidad de editores y parámetros para diferentes aspectos gráficos no solo de elementos naturales, sino también de otros elementos como personajes. Además dispone de editores de audio que ayudan a recrear la experiencia de juego.

La parte negativa de este motor de videojuegos es que el soporte para Realidad Aumentada y Realidad Virtual por parte de CryTek aún está en una fase de desarrollo muy temprana. Ya se han empezado a ver las primeras

aplicaciones que utilizan este tipo de tecnologías en algunas ferias, como por ejemplo, un juego de disparos utilizando Realidad Aumentada. No obstante, aún no se han liberado herramientas de creación de contenido para este tipo de aplicaciones. Otro aspecto negativo es su curva de aprendizaje de dificultad elevada. Si no se ha tomado mucho contacto con los motores gráficos, puede suponer un problema elegir CryEngine3 como primer motor a utilizar.

4.3.2. Unreal Engine 5

Unreal Engine 5 es posiblemente uno de los motores favoritos de los desarrolladores para trabajar. Viene de la mano de Epic Games y se diseñó exclusivamente para el título de la compañía Unreal Tournament. No obstante, debido a que es un motor genérico que además dispone de unas altas prestaciones en cuanto a calidad y rendimiento, suele ser la primera o segunda opción a elegir. A diferencia del anterior, este no está destinado a fines educativos, sino que cuenta con una versión con licencia (y suscripción mensual) y una tasa por derechos de autor del 5%. No obstante, el motor es gratuito para estudiantes y universidades, lo que lo convierte en una opción más que considerable para utilizarse a la hora de aprender a manejar un software de estas características.

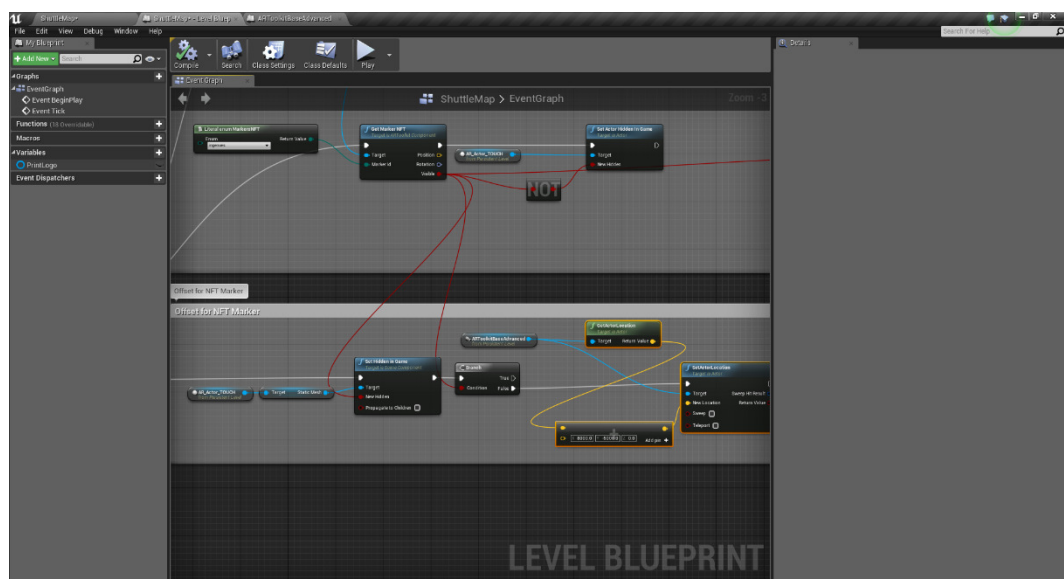


Figura 12: Blueprints en Unreal Engine. Fuente: <http://safefromrobots.com/?p=149>

Unreal Engine supera con creces la capacidad gráfica de la mayoría de motores de videojuegos del mercado y, como ya hemos comentado, está a la par con CryEngine en su última versión. Sin embargo, en cuanto al desarrollo independiente, ha perdido bastante terreno cuando el motor Unity ha entrado en juego.

Los puntos fuertes de Unreal Engine son los siguientes:

- Soporte multiplataforma.
- Acceso al código fuente, escrito en C++.
- Uso de diferentes librerías externas, como PhysX de Aegia / NVidia para el cálculo de las físicas.
- Sistema que pretende incluir todas las herramientas posibles dentro del motor.
- Gráficos de nueva generación en tiempo real a 30 fotogramas por segundo.
- Gestor de contenidos para los diferentes elementos de la escena y clasificación en diferentes carpetas
- Software SpeedTree de procesamiento de vegetación y ayuda de procesamiento de la flora y con renderizado a dos caras.
- Soporte al trabajo simultáneo en diferentes mallas de terreno.
- Iluminación global dinámica e iluminación diferida de gran rendimiento para un procesamiento rápido.
- Sistema de partículas de gran capacidad que permite manejar hasta un millón de partículas a la vez en tiempo real gracias al módulo Cascade.
- Sistema de raycasting para agilizar el cálculo de reflejos y transparencias y la generación de sombras suaves.
- Módulo Papers2D para la gestión y creación de contenido 2D, incluyendo un sistema de físicas, colisiones y segregación por capas, que además es compatible con desarrollo de juegos 2D-3D.
- Situación de elementos y materiales de forma procedural.
- Materiales PRB con propiedades físicas y matemáticas.
- Herramientas de diseño y edición de niveles sin utilizar scripts.

- Sistema Persona para creación, edición y animación de personajes.
- Sistema Matinee para creación y edición de cinemáticas.
- Sistema Blueprints de programación visual para la creación de contenido rápido y sencillo de cualquier tipo y sin utilizar scripts.
- Soporte para scripting en C++.
- Creador, mezclador y editor de parámetros de música y audio.

Como podemos observar, Unreal Engine no dispone a priori de tantas características gráficas como CryEngine, pero realiza otras muchas características que lo convierten en un motor de juegos muy potente. Principalmente podemos señalar su sistema de blueprints para la programación visual, es decir, un sistema diseñado para crear un videojuego sin tener que escribir código. Y cuando nos referimos a crear un videojuego, nos referimos a que con este sistema se puede controlar prácticamente cualquier aspecto, desde las reglas a las mecánicas o la jugabilidad, pasando por eventos y disparadores. Otro punto a favor es su sistema gestor de contenidos, que presenta una curva de dificultad más asequible que el anterior.

Otro punto a favor es la documentación que proporciona Unreal Engine, de gran cantidad y que a día de hoy aún sigue creciendo, sobre todo gracias a la puesta en marcha de una comunidad para el desarrollo con el motor, que permite a los usuarios no sólo intercambiar información, sino aportar ideas a proyectos en desarrollo y tener acceso a ellos para seguir el proceso de forma cercana. Además, cuenta con una librería de activos digitales (conocidos por su término en inglés “asset”) decente que cuenta con mucho material aportado por la comunidad y que se puede incorporar a otros proyectos.

Ya hemos señalado que nos importa especialmente si estos motores presentan soporte a nuevas tecnologías como la Realidad Aumentada. En este caso, Unreal Engine tiene compatibilidad con el dispositivo de Realidad Virtual Oculus y además dispone un módulo de Realidad Aumentada conocido como ARToolkit, que cuenta con licencia LGPL v3.0, es de código abierto, presenta compatibilidad con los principales sistemas operativos (Windows, MAC OS X,

Linux, iOS y Android), e incluye OpenGL e integración con sistemas de navegación y utilidades de calibrado de cámara.

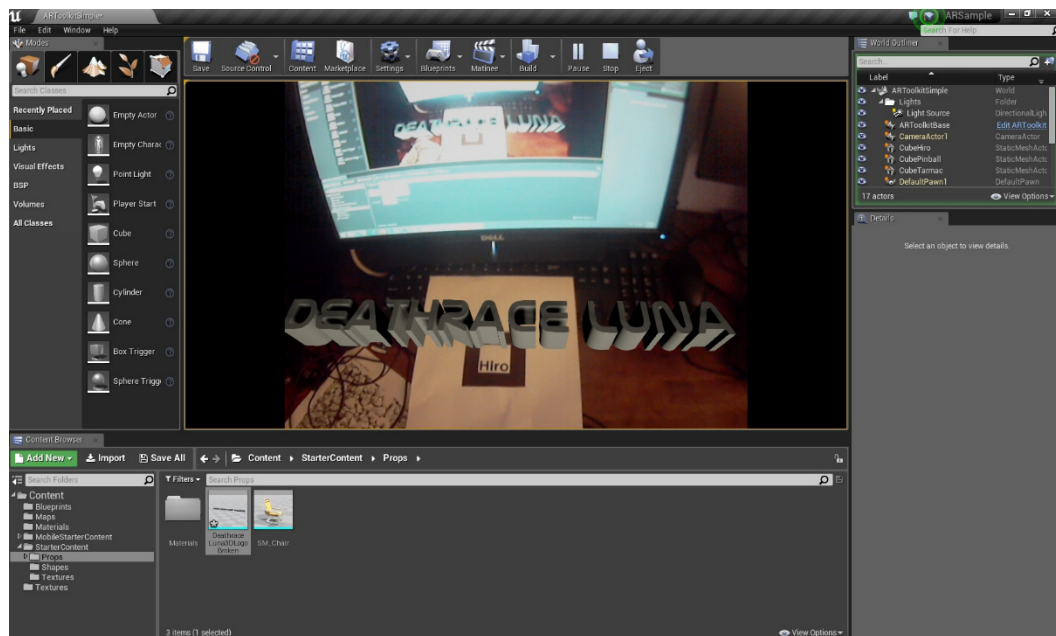


Figura 13: ARToolkit en Unreal Engine. Fuente: <http://www.wrenar.com/#!Unreal-Engine-4-AR-Test/c2011/568a9c7e0cf232a0de2119ad>

4.3.3. Unity 5

La idea que subyace tras el tercer motor de juegos que vamos a analizar es muy simple pero a la vez detalla con precisión el objetivo de los motores, y ésta no es otra que ofrecer una solución que los programadores, diseñadores y artistas de pequeñas y medianas empresas puedan utilizar de forma amigable y que su trabajo no se vea limitado por la plataforma de desarrollo. Creado por Unity Technologies primeramente para MAC OS, el auge de este tipo de tecnologías y, en concreto, la fama que se ganó este motor gráfico, lo hicieron avanzar hasta otras plataformas, incluidas aquellas móviles y de consolas, y poco a poco fue incrementando su funcionalidad y su rendimiento.

Hablar de Unity es hablar de un motor gráfico que permite el desarrollo de videojuegos tanto en 2D como en 3D y que cuenta con una versión gratuita y libre de derechos de autor, que además permite comercializar juegos siempre y cuando las ganancias sean inferiores a 100.000 \$ anuales. Aparte, ofrece otro

tipo de licencias de pago para aquellos desarrolladores con proyectos más ambiciosos y que pretenden generar un volumen de ventas superior al citado.

Sin embargo, respecto al resto de competidores, Unity pierde potencia en el motor gráfico aunque en general presenta buenos gráficos, pero de calidad inferior a motores como Unreal Engine 4 o CryEngine. Pero lo que este sistema pierde en cuanto a potencia gráfica lo gana en sencillez, presentando una curva de aprendizaje muy asequible incluso para aquellos que nunca han trabajado con un entorno similar.

Por citar algunos de los puntos fuertes de Unity, estos serían:

- Soporte multiplataforma.
- Iluminación global en tiempo real.
- Sistema de partículas que reacciona ante fuerzas externas.
- Sistema de reflejos HDR.
- Sistema PhysX de Aegia / NVidia para el cálculo de las físicas.
- Obstáculos dinámicos y prioridad de evasión.
- Flujos de trabajo en la ventana del proyecto.
- Animaciones por comportamiento basado en máquinas de estado.
- Sistema de transiciones de máquina de estado de alto nivel.
- Soporte para árboles de mezcla (conocidos por su término en inglés “blend trees”).
- Movimiento raíz para un mejor desplazamiento de los componentes del modelo.
- AudioMixer para crear y editar fuentes de audio como sonidos y música.
- Soporte para atenuación de fuentes de audio por elementos del juego tales como otras fuentes de audio, objetos o ruido.
- Control de versiones incluido en el propio motor.
- Compatibilidad con diferentes shaders básicos y posibilidad de modificarlos.
- Soporte para scripting en C#.
- Asistente de descarga de assets en el propio motor.

- Sistema Mecanim de animación de personajes de forma fluida a través de una interfaz sencilla.

Analizando la anterior lista ya vemos claramente como Unity es inferior (teniendo cuidado con el sentido que le damos a la palabra inferior, pues como ya hemos dicho, es cuestión de necesidades a cubrir ante un problema de desarrollo software) a sus competidores ya estudiados con anterioridad. Sin embargo, nos hemos dejado algunas cosas que merecen una descripción algo más extensa y que añaden valor fundamental a este motor.

Lo primero que hay que destacar es que Unity está envuelto por una comunidad inmensa. Dispone de una tienda de assets, algunos de ellos gratuitos, que ofrecen soluciones muy buenas a ciertas carencias del motor, por no hablar de una cantidad enorme de material para incluir en nuestro proyecto, que va desde modelos hasta materiales, pasando por fuentes de audio o incluso elementos de terreno, e incluso soluciones a problemas concretos en forma de paquetes que se pueden importar. Gracias a esta comunidad, la documentación de Unity es extensa, pero ya de por sí la propia web oficial de Unity ofrece un manual detallado con muchos ejemplos de implementación en cada sección, lo que dota a esta documentación de un valor importante. De forma no oficial, la comunidad se extiende por foros y sitios web que donde, en general, puedes recibir respuesta y soporte de forma rápida y útil.

Otro de los aspectos a destacar es que Unity trae soporte a desarrollo de aplicaciones 2D, por lo que es una plataforma excelente para crear este tipo de videojuegos, y lo hace una opción muy viable para proyectos que están orientados al entorno móvil.

Por último, y no menos importante, Unity dispone de un plugin conocido como Vuforia que da soporte a la Realidad Aumentada y también es compatible con Oculus. El paquete Vuforia dispone de diferentes tipos de licencias, pero una de ellas es gratuita y permite empezar a trabajar con él de forma inmediata. Vuforia cuenta con un sistema de rastreo de imagen y texto robusto y detección rápida de objetivos, que además puede ser simultáneo, y es compatible con la mayoría de plataformas, al igual que ARToolkit de Unreal Engine 4.

En resumen, Unity es un motor de videojuegos potente pero sencillo. Ofrece soluciones aceptables para el desarrollo en plataformas exigentes y se adapta muy bien al desarrollo en plataformas móviles. Este hecho combina perfectamente con la Realidad Aumentada, que generalmente está enfocada a dispositivos móviles, por lo que la combinación de este motor y el plugin Vuforia constituyen una solución perfecta para este tipo de aplicaciones.

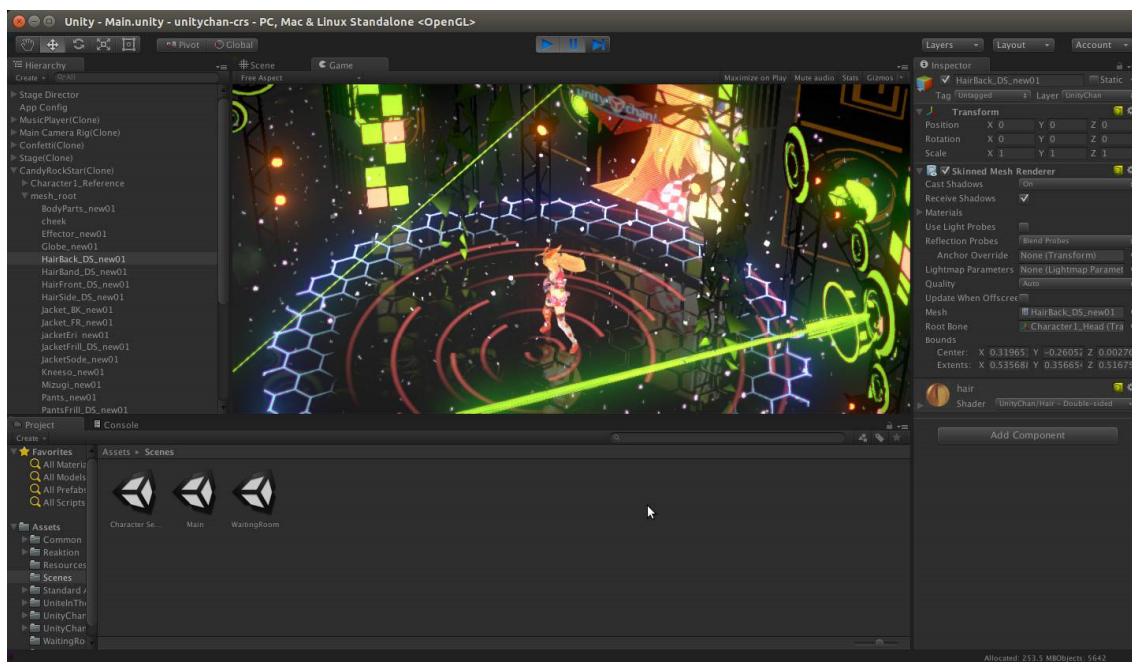


Figura 14: SDK de Unity. Fuente: <http://www.muylinux.com/2015/08/26/editor-unity-lanzado-nativa-gnu-linux>

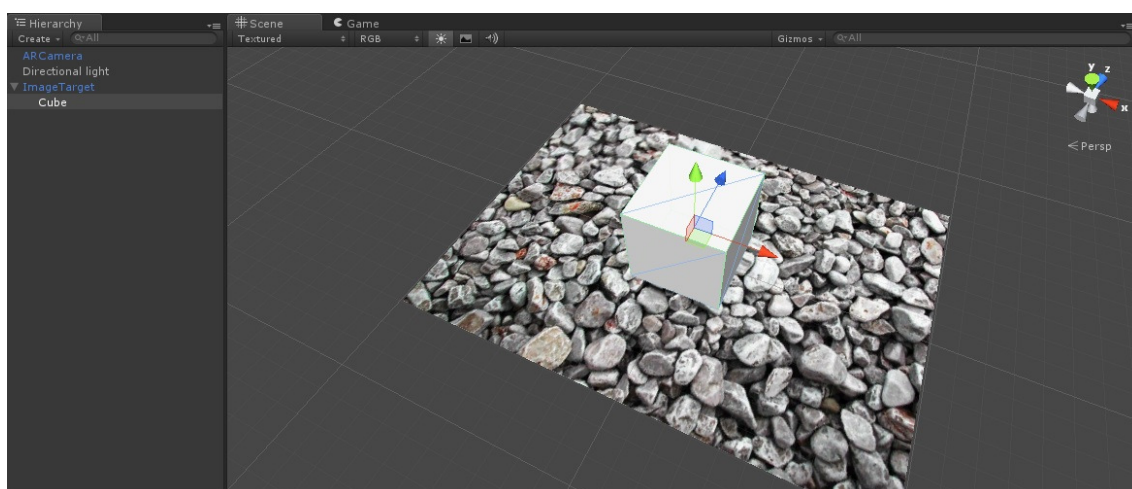


Figura 15: Vuforia sobre Unity. Fuente: <https://library.vuforia.com/articles/Solution/Compiling-a-Simple-Unity-Project>

4.4. Elección del motor de juego.

Ya hemos citado con anterioridad en este documento que la elección del motor de videojuegos por un lado es subjetiva y por otro lado es de vital importancia. Hemos analizado las características de los motores que ofrecen soluciones actualizadas a los videojuegos de hoy en día, y en base a ellas, debemos tomar una decisión de cara al desarrollo de nuestro proyecto, que recordamos, se basa en el uso de la tecnología de Realidad Aumentada y que está orientado al uso de dispositivos móviles.

Comencemos por algo sencillo, y es elegir los motores que no se adaptan a nuestras circunstancias. El primero de ellos es CryEngine, que directamente no soporta proyectos de Realidad Aumentada. Además, no necesitamos una solución gráfica potente, ni nos vemos en la necesidad de desarrollar una aplicación en un motor cuya curva de aprendizaje es dura.

Dicho esto, nos quedan dos posibles soluciones. Una de ellas es Unreal Engine 4, y la otra es Unity 5. Ambos se adaptan bastante bien a nuestras necesidades, sin entrar en detalles sobre cuál de ellos ofrece una mejor solución a la Realidad Aumentada.

Hay un hecho importante en nuestro proyecto que nos hace decantarnos por uno de los dos, y éste es que nuestro videojuego se orienta a dispositivos móviles. Esto quiere decir que no necesitamos a priori una potencia gráfica superior y, dado que Unity ensalza la sencillez en detrimento de los gráficos y de otras herramientas, parece ser que éste último va a adaptarse mejor a nuestro proyecto finalmente.

En cuanto a las herramientas, en principio no necesitamos desarrollar cosas demasiado complejas y, en cualquier caso, disponemos de soporte para scripts en lenguaje C#. Además, Unity tiene posiblemente la curva de aprendizaje más asequible, por lo que será más rápido y fácil hacerse al entorno.

Son razones pequeñas, y posiblemente elegir Unreal Engine es también tomar una buena decisión, pero creemos conveniente elegir Unity como motor de nuestra aplicación porque creemos que estas pequeñas razones marcarán

diferencias significativas a la hora de abordar el proyecto, sobre todo en temas de asequibilidad y facilidad de uso. Además, al elegir Unity, contamos con una gran comunidad que nos respalda y que hasta cierto punto puede ser de gran ayuda para solucionar problemas que se presenten demasiado complejos.

5. DESARROLLO DEL PROTOTIPO

5.1. Introducción.

Echando la vista atrás, en este documento nos hemos encargado de ponernos en el contexto asociado al desarrollo de un videojuego. Hemos dado una visión general sobre el estado del arte en los videojuegos, nos hemos adentrado en los conceptos principales para luego dar el salto a las etapas de desarrollo que debe seguir un proyecto de estas características. Tras ello, hemos realizado un análisis de motores gráficos que nos ha permitido tomar una decisión adecuada a nuestro sistema en cuanto a características y dificultad se refiere. Con todo ello, consideramos estar preparados para afrontar la última parte de este documento, la correspondiente al desarrollo del prototipo.

En términos globales, hemos planteado un videojuego que se adapte a plataformas móviles, concretamente a la plataforma Android, debido fundamentalmente a dos factores.

El primero de ellos es el uso de la Realidad Aumentada, que es el pilar fundamental de nuestro proyecto. No tiene mucho sentido escoger una plataforma diferente ante un problema de tal calibre, pues necesitamos un dispositivo portable y manejable con el que movernos para encontrar los diferentes objetivos que permitirán mostrar elementos solapados a la imagen de una cámara.

El segundo de ellos es que el juego está planteado como una gymkhana que definitivamente lleva asociado un componente de movilidad por parte del usuario y que debe satisfacer la plataforma en la que desarrollamos.

Para introducirnos un poco en el desarrollo de nuestro trabajo, indicamos de forma superficial los requisitos de los que partimos. Primeramente, necesitamos una aplicación que sea sencilla de manejar y que sea comprensible de forma inmediata a pesar de las ayudas que reciba el jugador por parte del sistema.

Consideramos que el público objetivo de esta aplicación debe ser adolescente, aunque pueda participar cualquier persona si lo desea, siendo éste el motivo que justifica la decisión anterior. Cabe destacar que la sencillez no sólo debe estar presente en conceptos como la interfaz o la información que se muestra, sino también debe aparecer en los juegos que se realizan. Éstos deben tener una dificultad bien ajustada para que puedan ser resueltos de forma ágil por diferentes tipos de personas.

Además, dado que nuestro planteamiento es un videojuego cuyo objetivo es realizar un recorrido completo, debemos considerar ciertos aspectos que si suceden no deben interrumpir la experiencia de juego, sino que debemos proporcionar un sistema continuo que permita avanzar a todo el mundo hasta el objetivo final.

Dicho todo esto, vamos a desglosar nuestro proyecto siguiendo las fases de desarrollo que hemos descrito con anterioridad.

5.2. Metodología de trabajo

Escoger una metodología de trabajo adecuada es esencial para desarrollar un proyecto software de forma eficaz y eficiente. La Ingeniería del Software pone a nuestra disposición diferentes paradigmas de ingeniería, una serie de pasos que abarca métodos, herramientas y procedimientos del software que se desarrolla.

En el contexto de este trabajo, hemos elegido como paradigma el modelo de prototipos por las siguientes razones:

- Conocemos un conjunto de objetivos generales que debe cumplir el software, pero no conocemos con seguridad cómo se va a comportar el sistema ante diferentes situaciones ni estamos seguros de la eficacia del paquete Vuforia en diferentes ámbitos (por cuestiones de iluminación o condiciones climáticas adversas).

- Debemos realizar un prototipo con el que acercarnos poco a poco a la solución final, comprendiendo poco a poco el motor de juegos y el paquete de Realidad Aumentada asociado. La idea principal es realizar un primer prototipo de pruebas para observar el comportamiento del software y desde este momento, empezar a desarrollar las diferentes versiones del producto añadiendo funcionalidades nuevas (por ejemplo nuevas etapas del recorrido) poco a poco y sólo cuando nos aseguremos de que los componentes implementados funcionan correctamente.
- Por la propia naturaleza del proyecto, debemos implementar un prototipo que funcione y que implemente algunos subconjuntos de funcionalidad del software final y con características que puedan mejorarse en el futuro.

En este modelo, en la fase de recolección de requisitos, se han considerado los siguientes:

- El sistema debe utilizar un motor de juegos con soporte a plataformas móviles (Android) y que disponga de tecnología de Realidad Aumentada.
- El sistema debe estar segmentado en diferentes escenas que representen cada uno de los tramos del recorrido.
- El sistema debe contener una cámara AR en común para todas las escenas.
- El sistema debe contar con una interfaz principal en común para todas las escenas, que utilice la paleta de colores adecuada y que muestre en tiempo real tanto el tiempo de duración del ejercicio como el marcador con la puntuación del jugador.
- Cada escena debe contar con un elemento informador que sea capaz de enviar al usuario información en forma de pistas en cualquier formato disponible y que debe estar asociado a la cámara AR para un visionado estático. Estos elementos deben ocupar el centro de la escena.
- Cada escena debe contar con un elemento informador que sea capaz de enviar al usuario información sobre las pruebas a desarrollar y compruebe el estado resolutorio de las mismas.

- Cada escena debe contar con dos temporizadores que permitan al usuario avanzar hasta el objetivo final. Uno de ellos ocultará las pistas y mostrará el nombre del destino a alcanzar, y el otro ocultará los juegos y mostrará las pistas del nuevo destino a alcanzar.
- Cada escena debe contener un elemento objetivo con una imagen asociada en referencia al lugar al que debe dirigirse el jugador en ese tramo del recorrido.

En cuanto a la fase de diseño rápido, se va a construir una escena básica con elementos sencillos, tales como una cámara AR, un elemento objetivo y un elemento a mostrar para las diferentes pruebas. Esto dará una pequeña idea de cómo se muestran los objetos en pantalla y permitirá organizar los elementos de forma adecuada, así como proporcionará una idea global de la implementación necesaria para que el sistema funcione correctamente.

En la fase de prototipo, se va a construir un sistema con funcionalidad parcial. El objetivo es diseñar dos escenas completamente funcionales en las que se proporcionen las pistas y los ejercicios de forma adecuada y en las que tanto el temporizador como el marcador funcionen correctamente y permitan el trasvase de valores de una escena a otra.

Se implementarán ejercicios sencillos y se reducirán los tiempos de los temporizadores de eventos (temporizadores para continuar hacia el objetivo final) con el objetivo de poder comprobar de forma ágil el funcionamiento del sistema.

La última fase se dedicará a pruebas y análisis para posibles mejoras futuras y para solventar fallos de diseño o corrección de errores. A través de ellas, podremos ir mejorando e incrementando la funcionalidad del sistema hasta obtener la versión final.

5.3. Planificación

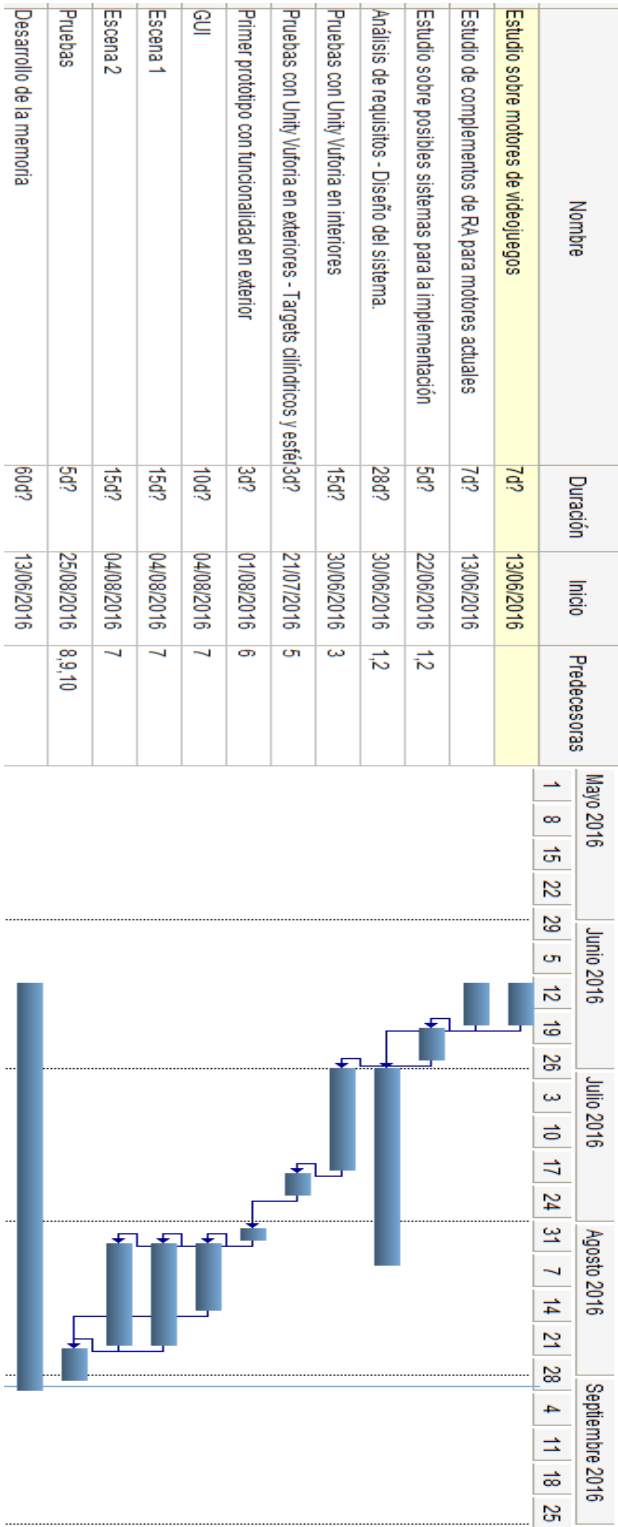


Figura 16: Diagrama de Gantt. Fuente: elaboración propia.

5.4. Diseño

5.4.1. Concepción y diseño

Nuestro juego nace con la idea de ser un camino guiado a través de una serie de pruebas. El planteamiento es recibir, de forma auditiva o visual (a través de textos o imágenes) información sobre un lugar de la ciudad al que dirigirse. Una vez que el jugador acierta con el lugar de destino, debe dirigirse a éste y posicionarse en una zona concreta para utilizar su terminal. En este momento, gracias a la Realidad Aumentada, se muestra por pantalla en una zona del lugar en el que se encuentra el jugador una prueba que se debe realizar. Cuando se supera, se da por finalizado el tramo correspondiente al recorrido y se repite el proceso con un nuevo tramo.

En cuanto a las pistas y pruebas, éstas son las encargadas de transmitir información cultural al jugador. Las pistas revelarán o formarán parte de la cultura de la ciudad, mientras que las pruebas permitirán al jugador recibir conocimiento a través de la resolución de pequeños ejercicios relacionados con la historia o el concepto del lugar que están visitando, de la ciudad o de la sociedad.

El objetivo del juego es terminar el recorrido en el menor tiempo posible, por lo que debemos medir la duración de la prueba de alguna manera. No obstante, también queremos potenciar la competitividad en el juego, por lo que nos interesa incluir un sistema de puntuación en cada una de las pruebas que hay que superar. Con esto, pretendemos también solventar algunos problemas que plantea la idea del juego. Uno de ellos es asegurarnos de que los jugadores realizan las pruebas y por tanto, reciben conocimiento sobre los diferentes lugares del juego, y evitar que algún jugador se limite exclusivamente a realizar el mismo recorrido que están realizando otros competidores. Entonces, podemos redefinir el objetivo del juego como recibir la mayor cantidad de puntos posibles y realizar la prueba en el menor tiempo posible.

En cuanto a la ambientación del juego, ésta se describe por sí sola gracias a la Realidad Aumentada. No necesitamos incluir demasiados elementos en nuestra aplicación porque el mundo que nos rodea se encargará de mezclar la información y presentar una información adecuada, sobre todo teniendo en

cuenta que nuestra intención es visitar lugares emblemáticos de la ciudad. Frente a esto, nos encontramos con el apartado de diseño.

Queremos realizar una aplicación sencilla, que muestre información necesaria, pero que ésta no sea demasiado abundante, ya que nuestro principal objetivo es el de fomentar el aprendizaje, por lo que conceptos que no son muy complejos generalmente funcionarán mejor a la hora de extraer la información importante del juego. Queremos diseñar algo fácil y rápidamente entendible, que sea capaz de guiar al jugador sin desviar demasiado su atención del mundo que le rodea, que a fin de cuentas también es quien transmite información educativa.

Reuniendo todos los factores anteriores, queremos realizar una aplicación que contenga un marcador y un temporizador como información permanente para el usuario. Luego, en cada sección del recorrido, queremos mostrar las pistas y los ejercicios a realizar de forma sencilla, sin que el usuario tenga que preocuparse de utilizar el sistema (por ejemplo, a través del uso de botones) para cambiar entre ellas o para poder realizar el siguiente tramo.



Figura 17: Concepto GUI. Fuente: elaboración propia

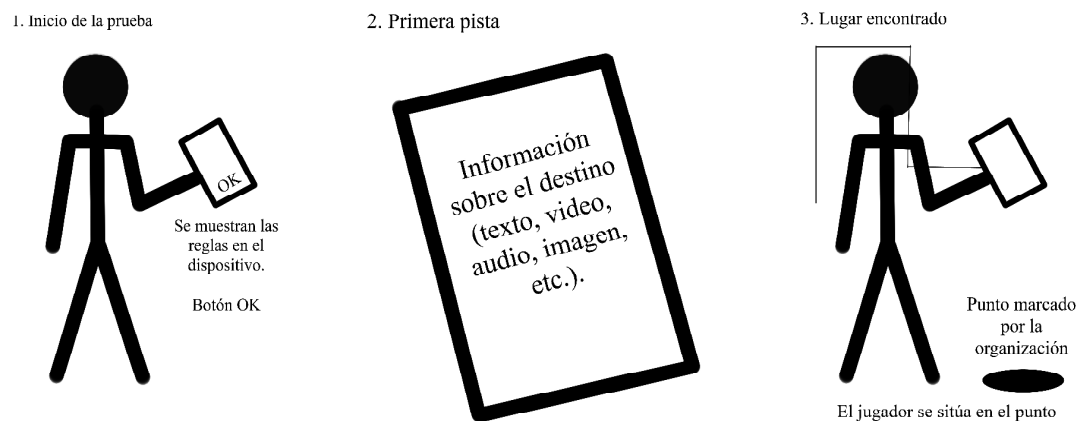


Figura 18: Storyboard del juego (1). Fuente: elaboración propia.

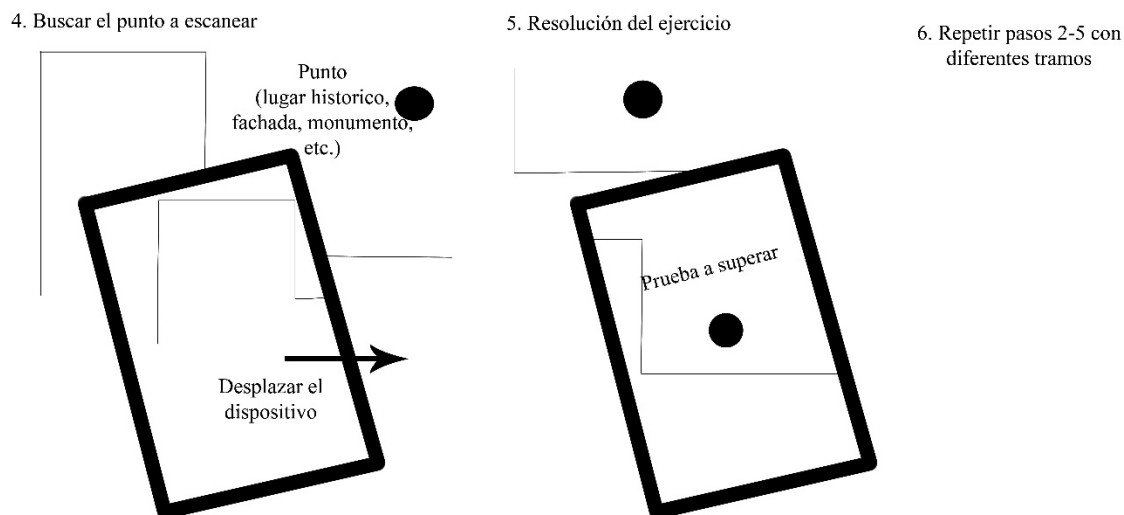


Figura 19: Storyboard del juego (2). Fuente: elaboración propia.

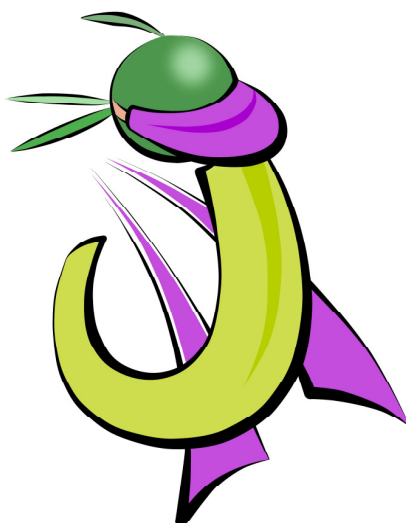


Figura 20: Logotipo de la gymkhana. Fuente: elaboración propia.

5.4.2. Género

Nuestro videojuego no está pensado en ser utilizado por un jugador de forma repetitiva y continuada, por lo que debemos catalogar nuestro juego como uno de género casual. Además, nuestra aplicación pretende transmitir conocimiento al jugador, el cual se ha de enfrentar a diferentes retos de memoria, resolución lógica y otro tipo de problemas que pueden asignar las etiquetas de educativo y puzle al género de nuestro videojuego.

5.4.3. Jugabilidad

Queremos presentar una aplicación en la que el usuario reciba información de forma automática y no tenga que interactuar demasiado con el sistema, guardando las distancias con la resolución de ejercicios, centrando la atención tanto en el mundo que le rodea como en los elementos importantes, las pistas y los ejercicios.

Permitiremos que éste controle los aspectos relacionados con la búsqueda del lugar histórico a través de la cámara de Realidad Aumentada y que interactúe con el terminal cuando haya que resolver ejercicios a través de la pantalla táctil del dispositivo. El resto de información se recibe de manera automática, esto es, cuando el jugador se encuentra en la fase de búsqueda del lugar, se muestra la información relacionada con dicho lugar (pista), y cuando éste sea capaz de encontrar el punto concreto, el sistema automáticamente ocultará dicha información y mostrará los elementos asociados a la prueba que se debe resolver.

Cuando el jugador es capaz de resolver una de las pruebas, es capaz de avanzar al siguiente tramo del juego, por lo que la mecánica general es que el jugador vaya resolviendo ejercicios, reciba nuevas pistas y pueda llegar al objetivo final.

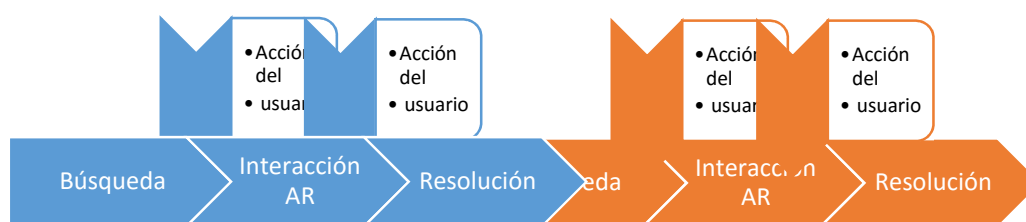


Figura 21: Diagrama de jugabilidad. Fuente: elaboración propia

5.5. Implementación

5.5.1. Diseño técnico

5.5.1.1. Diagrama de secuencia

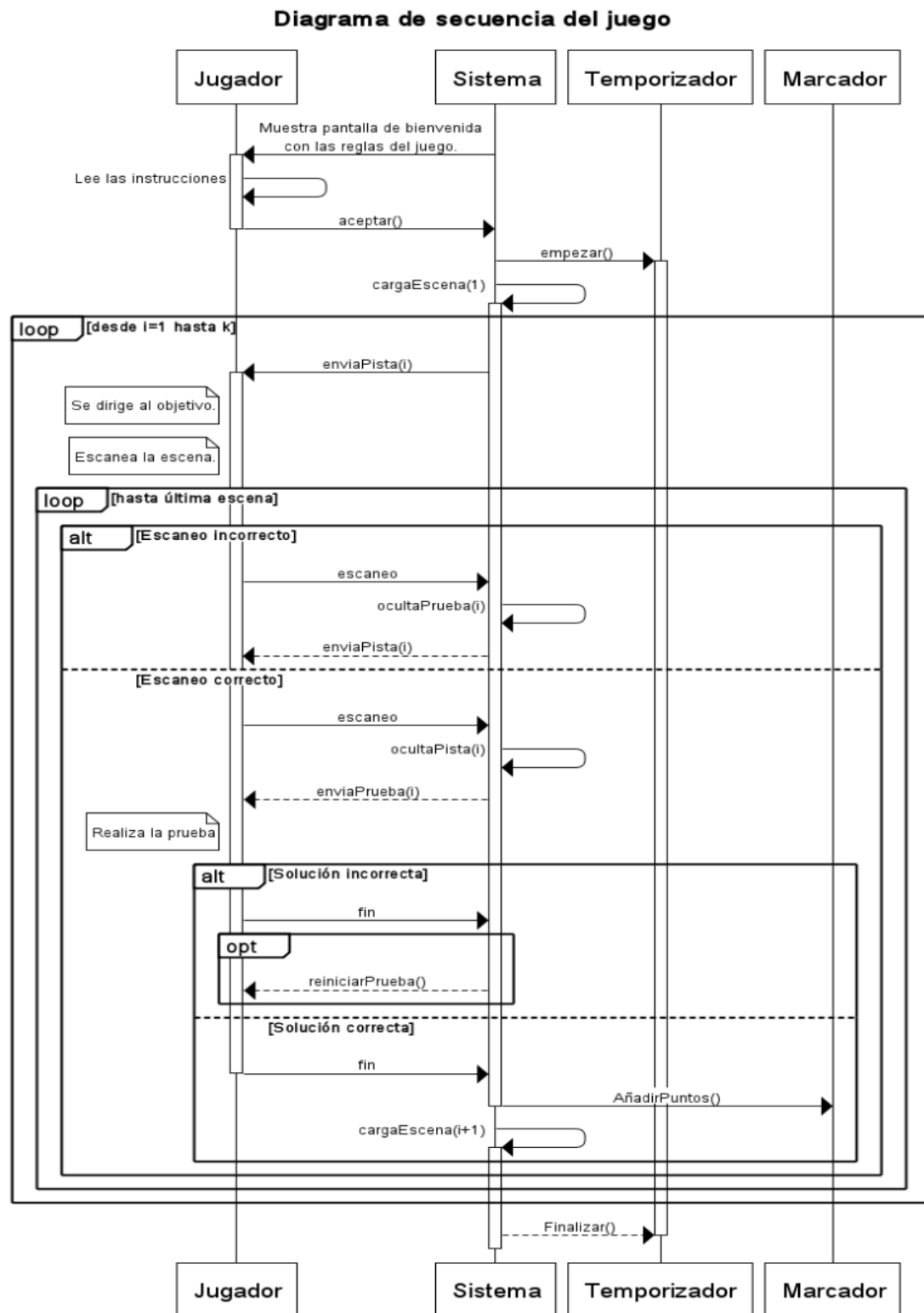


Figura 22: Diagrama de secuencia del juego. Fuente: elaboración propia.

5.5.1.2. Herramientas software y hardware.

Las herramientas de apoyo software que vamos a utilizar en nuestro proyecto son tres. La primera de ellas es el motor de videojuegos, que ya hemos comentado que se trata de Unity. La segunda sería un SDK de desarrollo para Android. La última de ellas es un complemento para el motor llamado Vuforia, y que da soporte a la Realidad Aumentada. Antes de continuar, creemos necesario comentar ciertos aspectos de estas herramientas.

Vamos a hacer uso de Unity en su versión 5 y, para ello, hemos obtenido una licencia gratuita de entre las diferentes opciones que ofrece. Así mismo, para desarrollar nuestro proyecto y que Unity pueda exportar la aplicación a nuestro terminal, debemos instalar el SDK de Android, del cual nosotros hemos elegido la versión 6.0 Marshmallow.

Por otro lado, Vuforia es un paquete que se importa a cualquier proyecto de Unity y que consta principalmente de una cámara de Realidad Aumentada (a partir de ahora cámara AR), una o varias bases de datos donde se encuentran almacenadas las imágenes objetivo que previamente hemos añadido y un elemento objetivo que se puede asociar a una de las imágenes de las diferentes bases de datos que contiene el proyecto.

La cámara AR cuenta con varios scripts desarrollados por el equipo de Vuforia que permite extraer los componentes de la escena en base a la posición relativa de una imagen objetivo. Por ejemplo, si en la escena colocamos una imagen objetivo cualquiera y sobre ésta colocamos un objeto en la esquina superior izquierda, cuando escaneemos la imagen con nuestro terminal y el script la detecte, mostrará el mismo objeto de la escena en la pantalla del dispositivo en la misma posición en la que se encontraba, es decir, la esquina superior izquierda.

Otro de los aspectos importantes del paquete Vuforia es que para que funcione, se debe obtener una licencia de las disponibles a través de su sitio web. La licencia es una secuencia de caracteres que deben colocarse en uno de los parámetros de la cámara para que ésta funcione. En nuestro caso, hemos

utilizado una licencia gratuita que limita el número de imágenes y escaneos mensuales a 1000 y que coloca una marca de agua en la aplicación.

5.5.2. Diseño del juego

Nuestro juego debe estar diseñado en diferentes escenas con ayuda del motor Unity. Cada una de ellas representa un tramo del recorrido total de la gymkhana y, además, en cada escena se presentan elementos similares y condiciones similares que se deben cumplir para pasar a la siguiente escena.

También debemos prestar especial atención a aquellos elementos que deben mostrarse continuamente y aquellos que no dependen de la posición de la cámara AR, es decir, aquellos elementos dependientes de la cámara AR y que están expresados en coordenadas de la cámara serán aquellos que se muestren en cualquier caso siempre y cuando estén dentro del frustum de visión de la misma, mientras que el resto de elementos cuyas coordenadas estén expresadas en coordenadas del mundo no siempre se mostrarán, pues dependerán del punto de visión de la cámara AR.

Debemos crear un proyecto en Unity al que se le importa el paquete Vuforia y las bases de datos de objetivos correspondientes, y crear diferentes escenas según la cantidad de niveles que queremos incluir en nuestro juego final. Cada una de estas escenas contiene diferentes elementos que detallaremos a continuación:

- Un objeto cámara virtual AR posicionado en algún punto de la escena y que contenga en el campo de licencia una válida y que tenga activas las diferentes bases de datos de imágenes objetivo. Este elemento forma parte del paquete Vuforia y puede encontrarse en los assets del proyecto, en el directorio correspondiente.

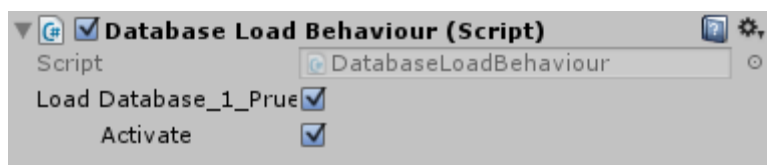


Figura 23: Campo de bases de datos en la cámara AR de Vuforia. Fuente: elaboración propia.

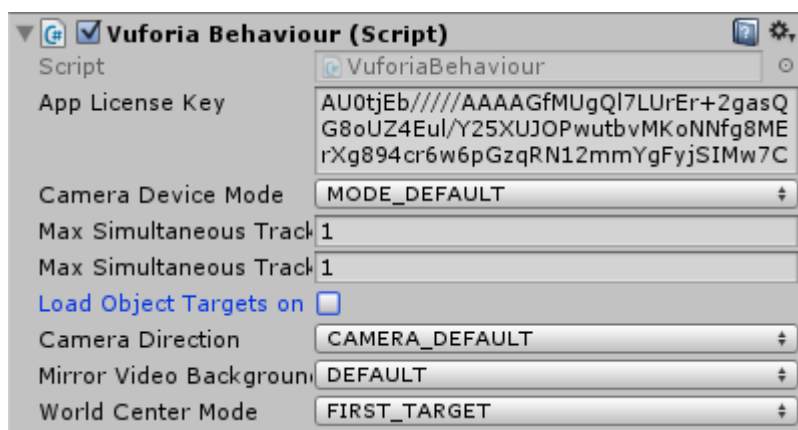


Figura 24: Campo de licencia en la cámara AR de Vuforia. Fuente: elaboración propia.

- Dos objetos de tipo canvas cuyos modos de renderizado están en coordenadas de la cámara AR descrita anteriormente. El primer canvas contiene dos elementos de tipo texto en el que se muestran tanto el temporizador como el marcador, controlados con scripts asociados a los mismos o al propio canvas, y un elemento imagen en el que se muestra el logotipo de la aplicación. Además, ambos elementos de texto hacen uso de la variable “PlayerPrefs” de Unity para guardar la información entre las diferentes escenas. El segundo canvas contiene uno o varios elementos de audio y/o imagen y al menos un elemento de texto correspondientes a la fase de pistas de cada sección del recorrido. En ellos, se muestra información que orienta al jugador sobre cuál es el lugar al que dirigirse.

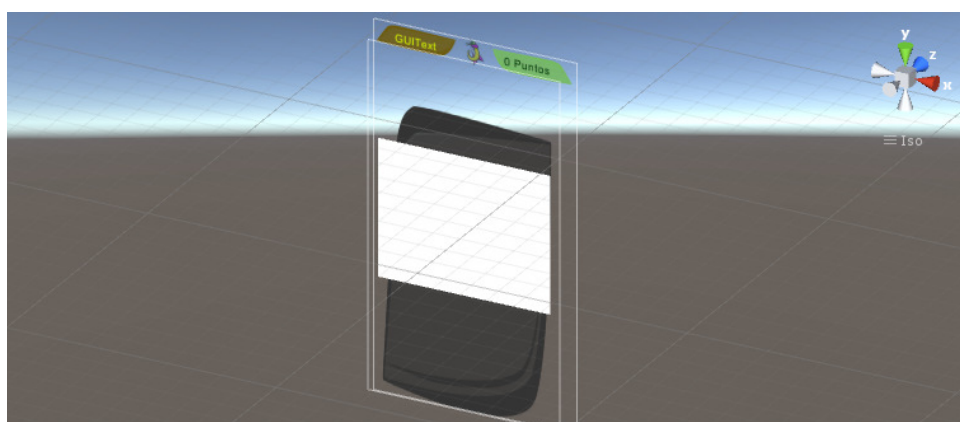


Figura 25: Canvas de la escena con sus elementos correspondientes. Fuente: elaboración propia.

- Un elemento objetivo (conocido por su palabra en inglés target) que tiene asociada una imagen de las diferentes bases de datos activadas por la cámara AR y que actuará como elemento identificador a la hora de realizar el escaneo con el terminal.

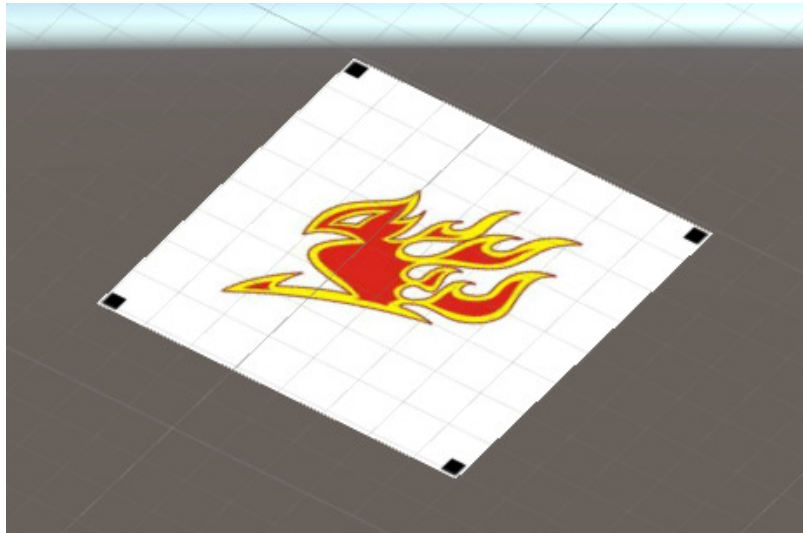


Figura 26: Imagen objetivo de una escena. Fuente: elaboración propia.

- Uno o varios elementos que representan a la prueba que se debe realizar con sus correspondientes scripts de funcionamiento, un elemento de texto con un temporizador para realizar la prueba, y un script asociado a todo el conjunto que compruebe si se ha realizado satisfactoriamente la prueba y se carga la siguiente escena o si se permanece en la misma y se reinicia la prueba en algunos casos.

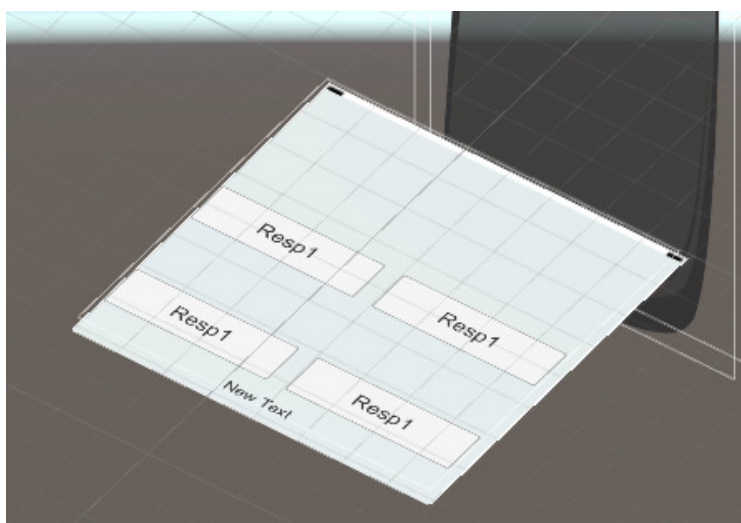


Figura 27: Elementos pertenecientes a la prueba. Fuente: elaboración propia.

5.5.3. Diseño Mecánico

Las reglas del juego son sencillas. El jugador dispone de un dispositivo de realidad virtual que le guía hacia ciertos lugares de la ciudad, en los cuales debe utilizarlo para escanear la zona con la cámara del mismo y realizar un sencillo ejercicio una vez esté enfocando el lugar correcto. Tras realizar el ejercicio, el jugador supera el nivel y se le presenta uno nuevo en el que recibe información sobre otro destino concreto y debe realizar el mismo procedimiento.

Ya que la interacción del jugador con el sistema se limita exclusivamente a los ejercicios, debemos controlar nosotros el resto de aspectos, la manera en la que se muestra la información y la forma en la que los elementos cambian o se transforman. En un principio, el sistema se diseñó para que el usuario pudiese ocluir los elementos que mostraban información de lugares mediante un botón, pero hemos considerado que la información debe ser persistente a lo largo de todo el recorrido hasta que el jugador resuelva el ejercicio. Esto supone un problema en cuanto el usuario encuentra el punto de detección del nivel, pues la información de la prueba y de la pista puede solaparse. Se ha optado por una solución que involucra al elemento objetivo mediante un script que detecta cuándo se está escaneando la imagen objetivo de la fase y, en caso afirmativo, se oculta la información de la pista y se muestra la información del juego. En caso negativo, se sigue mostrando la información de la pista y el juego permanece oculto. Este hecho supone otra ventaja, y es que obligamos al jugador a que deba realizar las pruebas en un sitio concreto, pues si éste no está apuntando directamente con su terminal al objetivo, el juego se oculta y es imposible realizarlo.

Cuando un jugador no es capaz de encontrar el lugar al que debe dirigirse, se interrumpe el flujo de juego. Hemos considerado entonces una solución que se basa en un disparador por tiempo, que cambia el texto de la pista por el nombre del lugar concreto al que el jugador debe dirigirse. El objetivo es fomentar que todo el mundo participe independientemente de si un segmento del recorrido no se ha realizado satisfactoriamente, pues nuestro objetivo es hacer llegar la mayor cantidad de conocimiento a través de nuestro juego. Por otra parte, este mismo problema puede surgir de la mano de las pruebas de cada fase y

podemos encontrar jugadores que no sean capaces de resolver un problema en un corto período de tiempo. En este caso, la solución adoptada es la misma, un disparador por tiempo que te redirige a la siguiente fase del recorrido, pero esta vez sin que el jugador pueda acumular puntos por resolver el ejercicio.

Ya que estamos con el sistema de puntuación, éste se actualiza de forma automática antes de pasar a la siguiente fase. Necesitamos persistencia de datos en este caso y es Unity el que nos proporciona de manera sencilla una solución. Este motor dispone de una función para scripts, `PlayerPrefs`, que permite almacenar y recuperar valores de una variable, para poder mantener el contenido de éstas en diferentes sesiones o escenas. Con el temporizador podemos optar por una solución similar y utilizar `PlayerPrefs` para controlar el tiempo total del recorrido entre las diferentes escenas.

En cuanto a los objetivos, hemos considerado dos posibles opciones. La primera de ellas es usar una imagen estática que está almacenada en la base de datos de Vuforia y que se debe situar en los lugares emblemáticos a los que queremos dirigir al jugador. La otra solución fue tomar una fotografía del lugar y añadirla a la base de datos. Desde puntos cercanos a los que se tomó la fotografía, el módulo Vuforia es capaz de reconocer los patrones de la foto directamente escaneando el objeto real. Ambas soluciones tienen sus ventajas e inconvenientes. La primera de ellas presenta una ventaja principal con respecto a la otra, y es que la segunda opción puede verse afectada por temas de iluminación (o directamente no funcionar si el ejercicio se realiza con muy poca luz). Esto no quiere decir que la primera no se vea igualmente afectada, pero es mucho más sencillo iluminar un objeto o imagen pequeño que uno grande. Como inconveniente, la primera solución puede hacer fallar el sistema de juego si, por alguna circunstancia, se pierde la imagen objetivo o se daña de alguna manera, bien sea por condiciones climáticas adversas, interacciones de la gente ajena al juego, etc. Con todo esto, consideramos mejor opción la segunda de ellas, pues la sensación de inmersión es mucho mayor en este caso, y las desventajas no llegan a suponer un punto decisivo para no optar por esta solución.

Ahora hablaremos de los juegos implementados en nuestro prototipo, que son dos hasta la fecha. El primero de ellos es una batería de preguntas que se

realiza en un lugar emblemático de la ciudad, la Catedral de Jaén. Para llegar a este sitio, mostramos un pequeño poema que la describe sin mencionarla. Cuando el jugador llega y encuentra el punto objetivo, se muestra el ejercicio. El jugador entonces debe responder a diferentes preguntas sobre la arquitectura y la historia de la Catedral eligiendo una de entre cuatro posibles respuestas. Si se produce un fallo, el juego se reinicia y se debe responder a todas las preguntas de nuevo. Además, cada vez que se muestra una de las preguntas, las respuestas están ordenadas de forma aleatoria, por lo que en principio ningún jugador en ninguno de sus intentos tendrá las respuestas colocadas en el mismo orden. Con este juego pretendemos que el jugador aprecie la historia y la cultura y amplíe su conocimiento sobre uno de los lugares más emblemáticos de la ciudad.

El segundo juego es un pequeño puzzle que se basa en el intercambio de piezas. La pista que guía hasta el objetivo es una secuencia de imágenes históricas de un lugar concreto, la Plaza de la Constitución. Cuando se alcanza el punto y se escanea el objetivo, se muestra una imagen dividida en pequeños rectángulos desordenados. El jugador debe seleccionar una pieza y a continuación otra para que éstas intercambien posiciones. Con este sistema, el usuario debe ser capaz de colocar la imagen en orden de forma que la imagen del puzzle coincida con la vista a través de la cámara AR. Esta imagen es una fotografía del Palacio de los Vilches. Al igual que en el ejercicio anterior, la mezcla de las piezas del puzzle es aleatoria cada vez que se ejecuta el juego. El objetivo es poder apreciar la evolución histórica del lugar, desde sus orígenes como rotonda hasta los días actuales como plaza, y poder observar con detalle la arquitectura del Palacio realizando el puzzle.

Hay que considerar que para que todo el juego funcione se necesita de un factor humano que, por una parte, controle los tiempos de salida de los jugadores y que se asegure de que éstos han iniciado la aplicación correctamente, y por otra parte se ocupe de marcar los puntos destino en la ciudad con el logotipo de la aplicación para que éstos sean claramente identificables a corta distancia. Sería interesante también que un personal asignado a la prueba se disponga en los diferentes lugares del recorrido por si algún jugador necesita consultar o

resolver alguna duda. Así mismo, si el jugador es menor, éste debe estar acompañado por un adulto responsable del mismo.

Para finalizar, debemos trazar un recorrido por la ciudad que nos permita abarcar la mayor cantidad de lugares emblemáticos, a ser posible que sean diferentes entre ellos, pero sin llegar a crear una ruta demasiado larga o complicada para el jugador, pues hay que tener en cuenta el factor físico y mental de los jugadores, que recordamos, deberían ser gente de corta edad.

Con todo esto, planteamos la siguiente ruta para el juego:

Inicio: Oficina de turismo de Jaén	
Etapa 1	
Lugar:	Catedral de Jaén.
Distancia al punto anterior:	Aproximadamente 200 metros.
Enfoque del turismo:	Conocimiento histórico y arquitectónico de la ciudad (Catedral).
Etapa 2	
Lugar	Plaza de la Constitución
Distancia al punto anterior	Aproximadamente 450 metros.
Enfoque del turismo	Conocimiento histórico y arquitectónico de la Ciudad (Plaza de la Constitución).
Etapa 3	
Lugar	Baños árabes.
Distancia al punto anterior	Aproximadamente 1000 metros.
Enfoque del turismo	Conocimiento cultural y tradición histórica (Influencias árabes).
Etapa 4	
Lugar	Casa de la Cultura, Biblioteca Provincial.
Distancia al punto anterior	Aproximadamente 1000 metros.
Enfoque del turismo	Arte y literatura local.
Etapa 5	
Lugar	Tienda oficial del Real Jaén C.F.

Distancia al punto anterior	Aproximadamente 850 metros.
Enfoque del turismo	Deporte local.
Etapa 6	
Lugar	Mercado de abastos (Plaza Federico G ^a Lorca).
Distancia al punto anterior	Aproximadamente 1600 metros.
Enfoque del turismo	Gastronomía andaluza.
Etapa 7	
Lugar	Parque del Bulevar
Distancia al punto anterior	Aproximadamente 1600 metros.
Enfoque del turismo	Actividades recreativas locales.
Etapa 8	
Lugar	Lagarto de Jaén
Distancia al punto anterior	Aproximadamente 1500 metros.
Enfoque del turismo	Leyendas populares locales.
Etapa 9	
Lugar	Universidad de Jaén
Distancia al punto anterior	Aproximadamente 700 metros.
Enfoque del turismo	Educación y cultura. La Universidad como institución educativa.
Etapa 10	
Lugar	Ronda de Marroquíes. Entrada a Jaén.
Distancia al punto anterior	Aproximadamente 400 metros.
Enfoque del turismo	El olivo como símbolo andaluz y comercio internacional.
Final	
Distancia recorrida	Aproximadamente 9300 metros.
Duración estimada	2 horas.

Figura 28: Tabla de ruta propuesta para el videojuego. Fuente: elaboración propia.

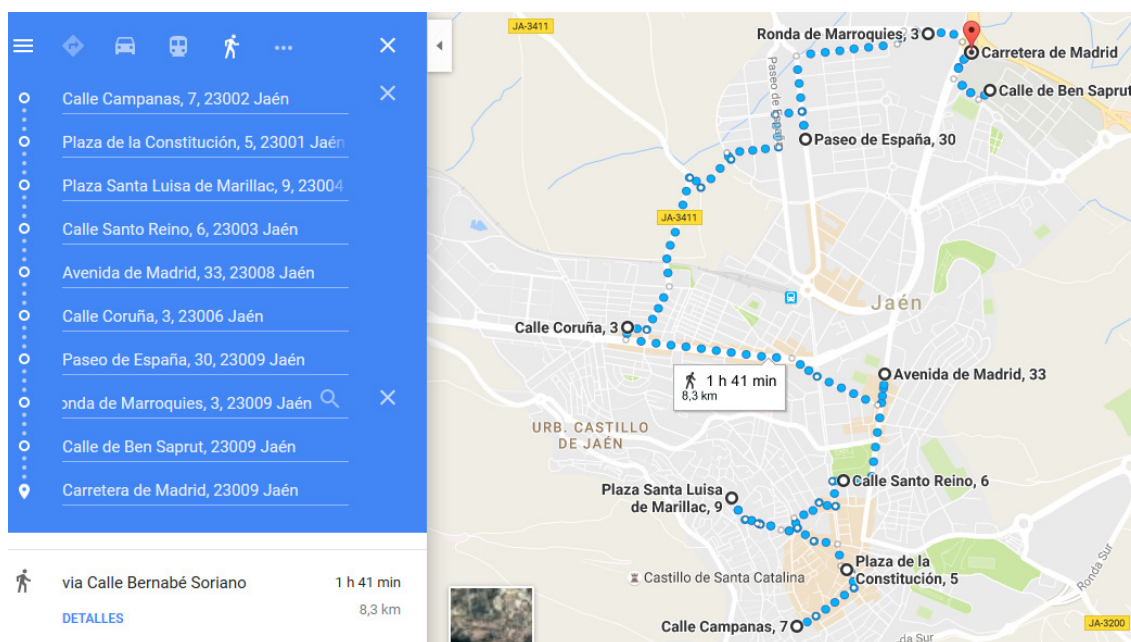


Figura 29: Recorrido a través de la ciudad de Jaén. Fuente: Google Maps.

5.5.4. Diseño artístico

El logotipo de la carrera representa a un jugador realizando el recorrido a través de una metáfora en forma de letra J que deriva de la palabra Jaén. Los colores se han elegido en base a los colores del escudo de la ciudad (amarillo y morado) y a su símbolo más característico, la oliva (color verde).

Los elementos se disponen en la pantalla en diferentes proporciones y posiciones. Dejamos el grueso central de ésta para mostrar en esta zona tanto las pistas como los ejercicios a realizar. Queremos conseguir con esta decisión centrar toda la atención del jugador en el lugar al que se dirige o en el ejercicio que debe resolver, dejando en un plano secundario el resto de información que no es necesaria para llegar al objetivo final del recorrido.

En la parte superior encontramos el logotipo de la carrera como elemento central, pues es un icono que se debe recordar para encontrar los diferentes puntos en las localizaciones a las que se dirigen los jugadores, por tanto, tomándolo como elemento central, conseguimos dirigir la vista hacia este elemento. En ambos lados de la parte superior se encuentran el marcador y el temporizador con colores vivos para contrastar con el resto de la escena, y bajo estos elementos se han colocado imágenes semitransparentes que oscurecen

la escena para mejorar su visualización sin llegar a ocultar del todo la salida de la cámara AR.

A su vez, los elementos de texto se realzan con un color llamativo sobre un fondo oscuro semitransparente que sirve para contrastar el texto mostrado con el resto de la escena, que puede adoptar colores muy claros o vivos en función de donde esté apuntando el jugador con la cámara AR.

Los elementos de juegos se realizarán en la medida de lo posible con fondo blanco y texto o elementos en colores oscuros o negro para reforzar la sensación de contraste citada con anterioridad. La posición de los mismos dependerá de la posición de la cámara AR con respecto al objetivo.



Figura 30: Paleta de colores del logotipo. Fuente: elaboración propia.



Figura 31: Paleta de colores de los elementos (puede contener transparencia). Fuente: elaboración propia.

5.5.5. Implementación

Ya hemos dicho que la implementación es el momento de juntar las piezas del mecanismo para que trabajen a la vez y lleven adelante el funcionamiento del sistema. Nuestro juego se basa en la cámara AR como uno de los elementos principales, por lo que hemos colocado una de ellas en nuestra escena. No necesitamos más cámaras porque el usuario dispone de una única pantalla en su dispositivo.

Lo primero que hemos hecho con ella ha sido agregar la licencia y la base de datos que contiene nuestros objetivos captura. Además hemos fijado el proyecto a una resolución concreta pero escalable, esto es, hemos bloqueado la opción de rotación gracias al giroscopio del terminal para mantener siempre la misma proporción, pero hemos diseñado los elementos escalables para diferentes resoluciones.

Hemos creado el canvas principal que contiene el marcador, el temporizador y el logotipo de la aplicación y lo hemos obligado a mantenerse en coordenadas de la cámara para que las imágenes objetivo no afecten a los elementos. Con esto conseguimos mostrarlos siempre en la misma posición independientemente de los movimientos de la cámara.

De igual forma, hemos creado un canvas secundario con coordenadas fijadas en la cámara en el que se pueden mostrar las pistas de los diferentes lugares. Hemos incluido en ambos casos diferentes scripts que controlan el funcionamiento de los diferentes elementos, tal y como hemos comentado antes (avance del tiempo, adición de puntos al marcador, ocultación del canvas de pistas cuando se escanea un objetivo, etc.).

Otro de los elementos principales son las imágenes objetivo. Hemos colocado una de ellas en la escena con la fotografía del lugar que corresponde a una sección del recorrido y sobre la misma hemos colocado los diferentes objetos que gracias a sus scripts asociados permiten la ejecución de la prueba del nivel. Queríamos dar especial atención a que el ejercicio sólo pueda realizarse cuando se está escaneando el objetivo, por lo que hemos decidido dejar estos elementos en coordenadas del mundo y no de la cámara AR.

Gracias a que el paquete Vuforia relaciona directamente a la cámara AR con las imágenes objetivo, y que a su vez los diferentes elementos de nuestra escena están relacionados bien con uno u otro, obtenemos un sistema que funciona correctamente y que permite llevar una secuencia ordenada y nos permite avanzar en el juego de forma satisfactoria.

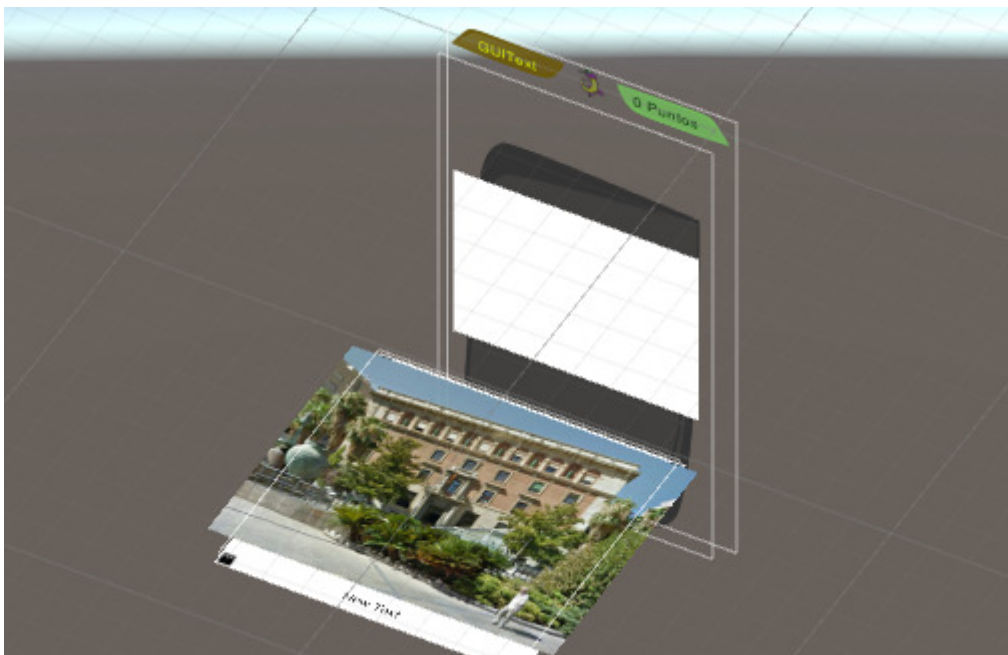


Figura 32: Escena completa del Segundo nivel. Fuente: elaboración propia.

5.6. Pruebas

Test 1: Identificación de objetivos impresos en papel coincidentes con la imagen objetivo en la escena de Unity.	
Resultado esperado	Resultado obtenido
Los elementos de la escena se muestran correctamente a través de la cámara AR en el dispositivo móvil.	Los elementos de la escena se han mostrado correctamente en el dispositivo móvil.

Test 2: Identificación de objetivos a través del reconocimiento de escenarios del mundo exterior.	
Resultado esperado	Resultado obtenido
Los elementos de la escena se muestran correctamente a través de la cámara AR en el dispositivo móvil.	Los elementos de la escena se han mostrado correctamente en el dispositivo móvil.

Test 3: Identificación de objetivos a través del reconocimiento de escenarios del mundo exterior con baja iluminación.

Resultado esperado	Resultado obtenido
Los elementos de la escena se muestran correctamente a través de la cámara AR en el dispositivo móvil.	Los elementos de la escena se muestran con artefactos y vibraciones dependiendo de la calidad de la imagen objetivo.

Test 4: Identificación objetivos a través del reconocimiento de escenarios del mundo real con alta iluminación y sombras definidas.

Resultado esperado	Resultado obtenido
Los elementos de la escena se muestran correctamente a través de la cámara AR en el dispositivo móvil.	Los elementos de la escena se muestran con artefactos y vibraciones dependiendo de la calidad de la imagen objetivo, pero en menor cantidad que durante el Test 3.

Test 5: Identificación objetivos a través del reconocimiento de escenarios del mundo con variaciones en la distancia entre el punto en que se tomó la imagen y el objetivo.

Resultado esperado	Resultado obtenido
Los elementos de la escena se muestran correctamente a través de la cámara AR en el dispositivo móvil.	Los elementos de la escena se muestran con artefactos y vibraciones y se produce oclusión por frustum en distancias superiores a la recomendada. La imagen objetivo no es detectada y los elementos no se muestran en distancias muy cortas a la recomendada.

Test 6: Comprobación de reconocimiento de imagen objetivo para implementación de mecánicas.

Resultado esperado	Resultado obtenido
El sistema es capaz de detectar e informar a otros componentes cuando la cámara AR está detectando una imagen objetivo.	El sistema detecta e informa cuando se produce la detección de imágenes objetivo a través de la cámara AR.

Test 7: Adaptación de la información desplegada en función de la detección de imágenes objetivo.

Resultado esperado	Resultado obtenido
El sistema es de intercalar la información de las pruebas y de los ejercicios dependiendo de si se está escaneando una imagen objetivo o no.	El sistema intercambia la información de las pruebas y de los ejercicios dependiendo de si se está escaneando una imagen objetivo o no.

Test 8: Los elementos con información persistente envían datos a través de las diferentes escenas.

Resultado esperado	Resultado obtenido
El sistema es capaz de recolectar información de variables entre escenas.	El sistema recoge información a través de PlayerPrefs entre escenas.

Test 9: El sistema de disparadores por tiempo responde correctamente.

Resultado esperado	Resultado obtenido
El sistema permite al usuario avanzar en el juego gracias a disparadores por tiempo cuando éste no consigue resolver una pista o un ejercicio.	El sistema avanza entre pruebas y escenas permitiendo alcanzar el objetivo final.

Test 10: El sistema es capaz de cumplir con los requisitos de diseño y funcionamiento especificados.	
Resultado esperado	Resultado obtenido
El sistema presenta las pistas y muestra al usuario las pruebas cuando alcanza su destino, calcula correctamente el tiempo de juego, añade puntos al marcador si éstas se superan y avanza entre escenas en cualquier situación.	El sistema funciona en cuanto a los requisitos de diseño y funcionamiento, cumpliendo las expectativas.

6. CONCLUSIONES

Al final de este recorrido, queremos hacer un pequeño resumen sobre los aspectos tratados en este documento y realizar un balance de todos los aspectos, tanto positivos como negativos, que nos hemos encontrado durante el desarrollo del proyecto.

Para empezar, hemos abordado el tema de la Realidad Aumentada para situarnos en el contexto de este proyecto. Hemos intentado definir los componentes de una aplicación de Realidad Aumentada y cómo funcionan a modo de introducción para el desarrollo del documento.

Seguidamente, indicar que nos hemos aproximado en gran medida al concepto de videojuego, hemos dado una definición que podemos considerar como global y hemos estudiado qué elementos lo conforman y cuál es la importancia de cada uno de ellos.

Hemos podido comprobar de primera mano lo difícil que es abordar el tema de los videojuegos, ya que es un mundo artístico lleno de subjetividad y contraste de opiniones.

El análisis de los motores de videojuegos nos ha permitido obtener un amplio conocimiento sobre este tipo de herramientas, que son fundamentales en el desarrollo de un videojuego, y nos han llevado a tomar una decisión en cuanto a la elección de un motor de juegos adecuado a nuestro proyecto.

Con el siguiente tema hemos podido aplicar una metodología de trabajo y los conocimientos adquiridos a nuestro proyecto de desarrollo del prototipo. Hemos trabajado con el motor de videojuegos y con el paquete de Realidad Aumentada Vuforia con resultados satisfactorios, a pesar de algunos inconvenientes que presenta este tipo de tecnología, como la iluminación o la susceptibilidad a cambios de las imágenes objeto.

Hemos concluido que tanto Unity como Vuforia aún tienen camino que recorrer, pero ofrecen una solución sólida en líneas generales para nuestro propósito, y gracias a estas herramientas software hemos podido construir un primer prototipo de videojuego que funciona en los términos planteados. El sistema se comporta de acuerdo a los requerimientos, aunque es necesario escoger bien las imágenes objetivo, teniendo en cuenta varios factores como la iluminación y las sombras, la distancia al punto de detección, las condiciones climáticas, etc.

Para finalizar, a nivel personal este trabajo ha supuesto una gran experiencia para nosotros. Utilizar un motor de videojuegos y comprobar la utilidad de esta herramienta es un valor necesario en estos tiempos en los que la tecnología avanza a pasos agigantados.

Desarrollar un videojuego para una plataforma móvil que cada día tiene más usuarios ha sido una decisión correcta que nos ha permitido una toma de contacto con este mundo y que nos motiva a seguir en la misma línea.

Y, sobre todo, adentrarnos en el mundo menos conocido de los videojuegos, aquel destinado a fines educativos, supone abarcar un tema no tan común que nos genera una gran fascinación y satisfacción cuando nuestra aplicación funciona en éstos términos.

7. LÍNEA DE FUTURO

Como planteamiento para el futuro, nos gustaría detallar qué nuevas incorporaciones y mejoras deberá contener el sistema.

- Escenas que correspondan al resto de fases de la gymkhana descrita en la fase de desarrollo.
- Ejercicios que cubran los diferentes aspectos del turismo en cada una de las escenas descritos en la fase de desarrollo.
- Mejoras en la detección de imágenes objetivo para evitar artefactos y oclusiones.
- Agilizar el proceso de cambio entre escenas.
- Añadir nuevas mecánicas que refuercen el aprendizaje, por ejemplo, inclusión de textos o vídeos después de cada prueba, antes de mostrar las pistas al siguiente objetivo, y realizar una ronda final para obtener puntos en base a los conocimientos adquiridos.
- Añadir nuevos sistemas que completen la información, tales como mapas o brújulas digitales.

Bibliografía

- [1] Fundación Telefónica (2011): Realidad Aumentada: una nueva lente para ver el mundo.
- [2] Manrubia, A. M. (2014): "El proceso productivo del videojuego", *Historia y Comunicación Social*, Volumen 19 Nº Esp. Marzo, 791-805
- [3] Juul, J. (2003): "The game, the Player, the World: Looking for a Heart of Gameness", *Digital Games Research Conference Proceedings*, 30-45.
- [4] Lantz, F; Zimmerman, E. (1999): "Checkmate: Rules, Play and Culture!", *Merge Magazine* Nº5, 41-43.
- [5] Ballejo Fernández, D; Martín Angelina, C. (2012): *Desarrollo de Videojuegos 1: Arquitectura del Motor de Videojuegos*, Universidad de Castilla-La Mancha.
- [6] Scheel, J (2008): *The art of Game Design. A Book of Lenses*, CRC Press.
- [7] Järvinen, A.; Heliö, S.; Mäyrä, F. (2002): *Communication and Community in Digital Entertainment Services: Prestudy Research Report*, University of Tampere.
- [8] Gregory, J. (2009): *Game Engine Architecture*, CRC Press.
- [9] Abt, C. (1970): *Serious Games*, Viking.
- [10] Zyda, M. (2005): *From visual simulation to virtual reality to games*, IEEE Computer.
- [11] Pâquet, J. (2010): <http://www.jordanepaquet.fr/>
- [12] Lundgren, F; Pearce-Authers, R. (2013): *CryENGINE Game Programming with C++, C# and Lua*, Packt.
- [13] Tracy, S; Reindell, P (2012): *Cryengine 3 Game Development: Beginner's Guide*, Packt.
- [14] CryTek: <http://www.crytek.com/cryengine/>
- [15] Unreal Engine: <https://docs.unrealengine.com/latest/INT/>
- [16] Coronado, C. (2014): *Unreal Engine 4, Herramienta versátil de desarrollo*, <http://www.zehngames.com/developers/unreal-engine-4-herramienta-versatil-de-desarrollo/>
- [17] Ariel (2014): *Unreal Engine 4 Vs Unity 4*, <http://epicnerdstadium.com/2014/10/11/unreal-engine-4-vs-unity-4/>

- [18] Ariel (2016): *Unreal Engine 4: Blueprints Vs C++*,
<http://epicnerdstadium.com/2016/07/04/unreal-engine-4-blueprints-vs-c/>
- [19] Cookson, A.; DowlingSoka, R.; Crumpler, C.; Johnson, T. (2016): *Unreal Engine 4 Game Development in 24 Hours*, Sams Publishing.
- [20] Candil, D. (2014): *Unity, el motor de desarrollo capaz de partir la historia de los videojuegos en dos*, <http://www.vidaextra.com/industria/unity-el-motor-de-desarrollo-capaz-de-partir-la-historia-de-los-videojuegos-en-dos>
- [21] Unity: <https://unity3d.com/es/unity/whats-new/unity-5.0>
- [22] Wikipedia: [https://es.wikipedia.org/wiki/Unity_\(software\)](https://es.wikipedia.org/wiki/Unity_(software))
- [23] Wikipedia: https://en.wikipedia.org/wiki/List_of_game_engines

Anexo I: Manual de instalación.

En este anexo se va desarrollar el manual de instalación mediante el cual queremos exponer y explicar la manera de instalar todas las herramientas necesarias para nuestro proyecto.

I1. Descarga e instalación de Unity 5.

Paso 1: Acceder al sitio web de la tienda de Unity a través de <https://store.unity.com> y seleccionar una de los paquetes disponibles. Nosotros hemos descargado la versión gratuita de Unity 5. Tras elegirlo, nos dirigirá a un nuevo sitio web en el que podremos descargar el instalador del motor gráfico.

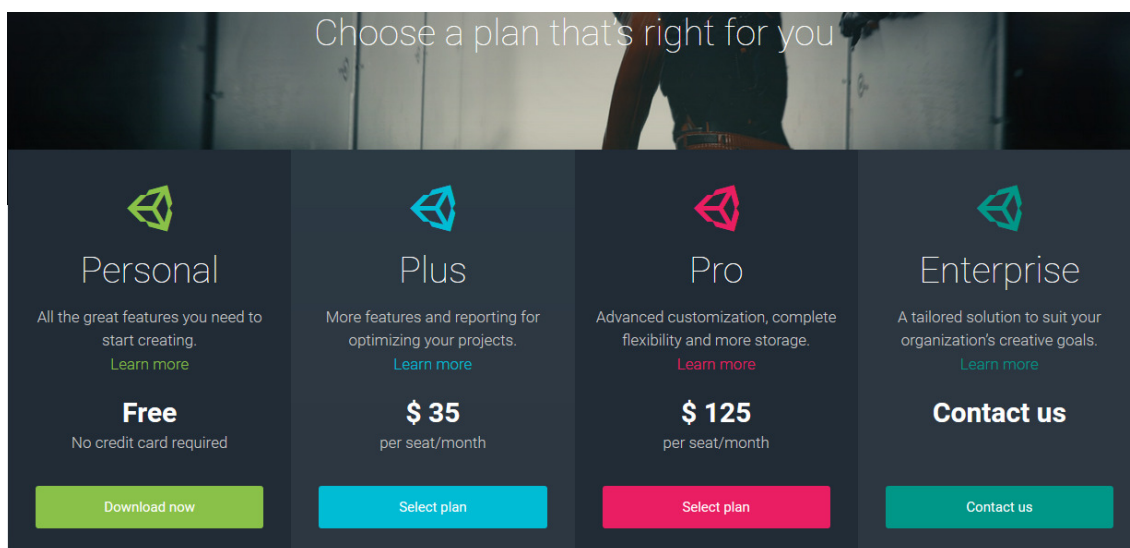


Figura I 1: Tienda para descargar el motor Unity 5. Fuente: elaboración propia.

Paso 2: Ejecutar el instalador descargado y seguir los pasos de instalación. En el momento correspondiente, seleccionar la arquitectura de 32 bits, ya que Vuforia aún no es compatible con la versión de 64 bits y puede generar algunos problemas. Tras esto, seleccionar los componentes deseados (se puede realizar la instalación con los elementos que vienen seleccionados por defecto) y elegir el directorio de instalación del programa que queramos.

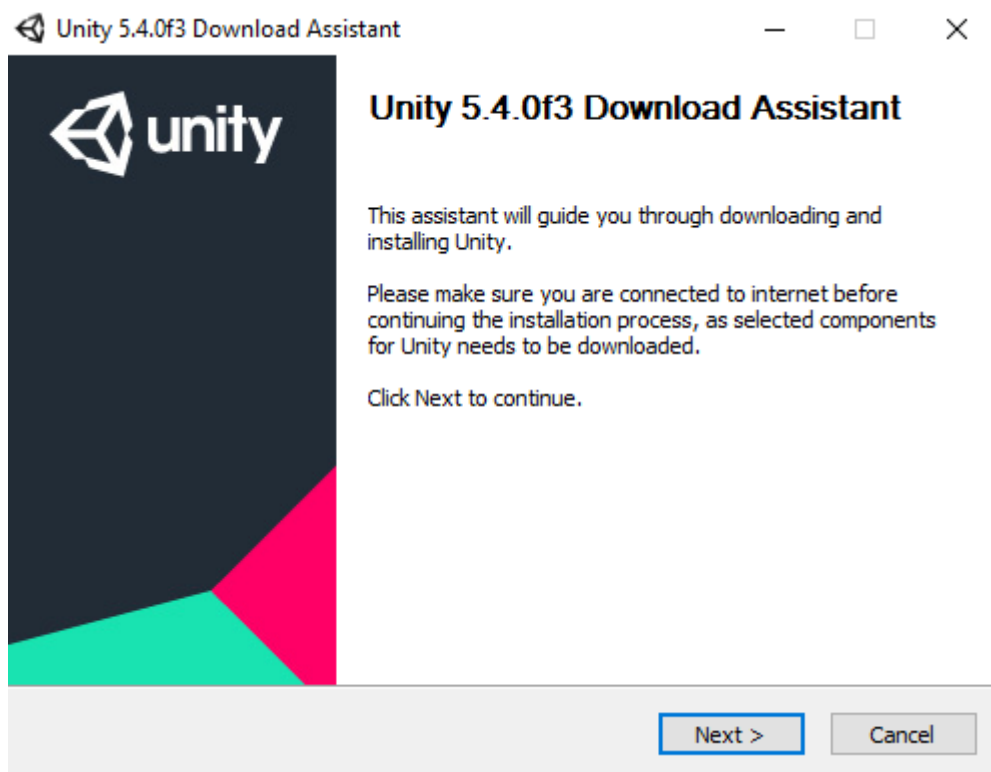


Figura I 2: Instalador de Unity. Fuente: elaboración propia.

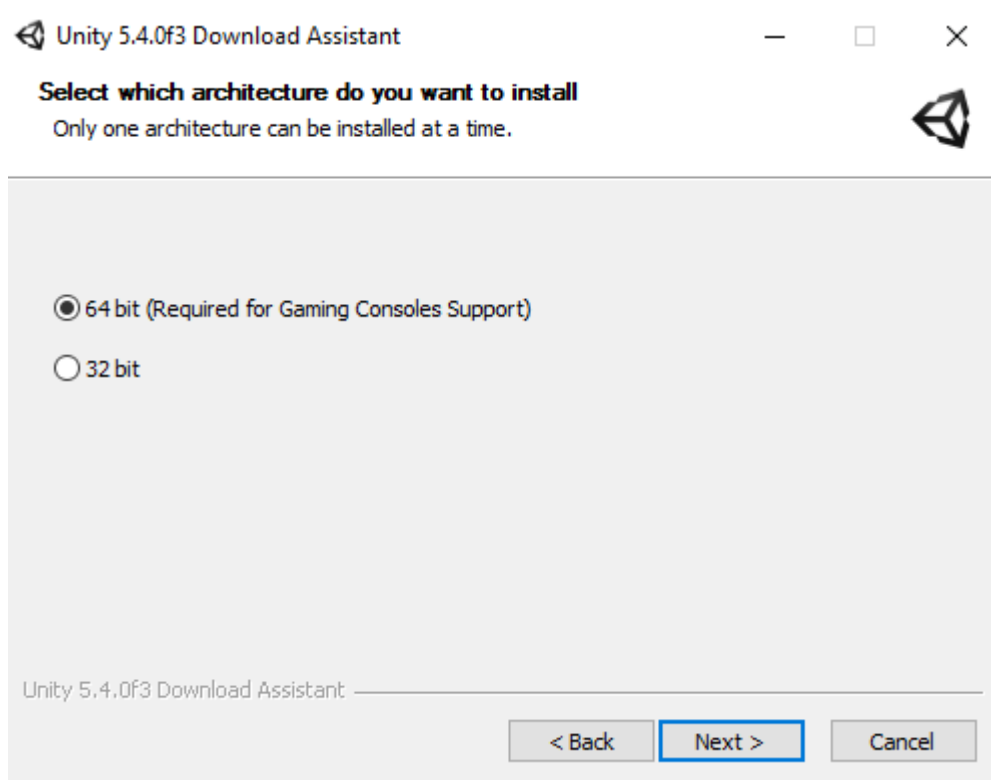
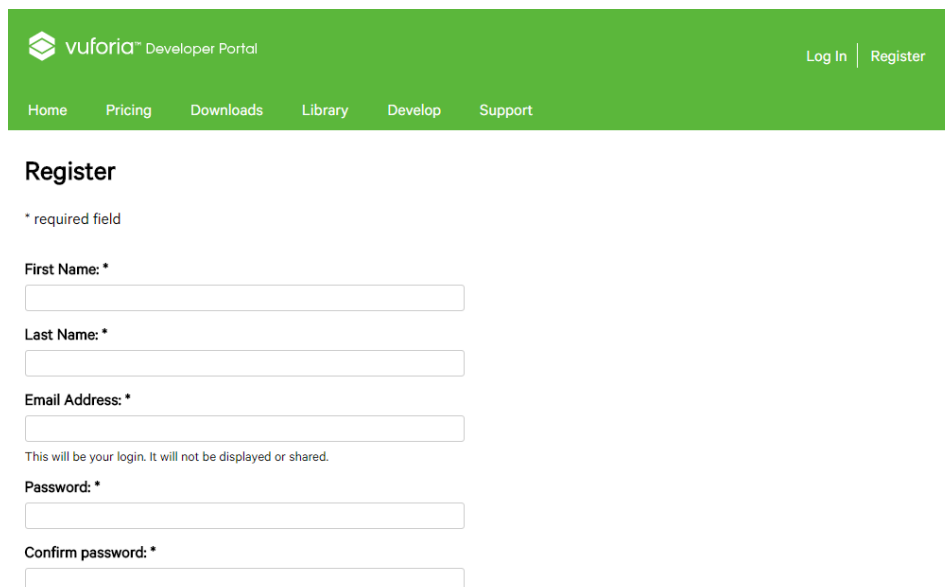


Figura I 3: Selección de arquitectura. Fuente: elaboración propia.

I2. Descarga e instalación del paquete Vuforia.

Paso 1: Acceder al sitio web de Vuforia a través de <https://developer.vuforia.com> y crear una cuenta de forma gratuita.



The image shows the 'Register' page of the Vuforia Developer Portal. The header is green with the Vuforia logo and 'Developer Portal' text. Navigation links include Home, Pricing, Downloads, Library, Develop, and Support. The 'Log In' and 'Register' links are on the right. The registration form includes fields for First Name, Last Name, Email Address, Password, and Confirm password, each marked with an asterisk to indicate they are required. A note states: 'This will be your login. It will not be displayed or shared.'

Figura I 4: Formulario de registro en Vuforia. Fuente: Elaboración propia.

Paso 2: Acceder al sitio web de descargas de Vuforia a través de <https://developer.vuforia.com/downloads/sdk>, aceptar los términos de licencia y descargar el paquete con el SDK para Unity.

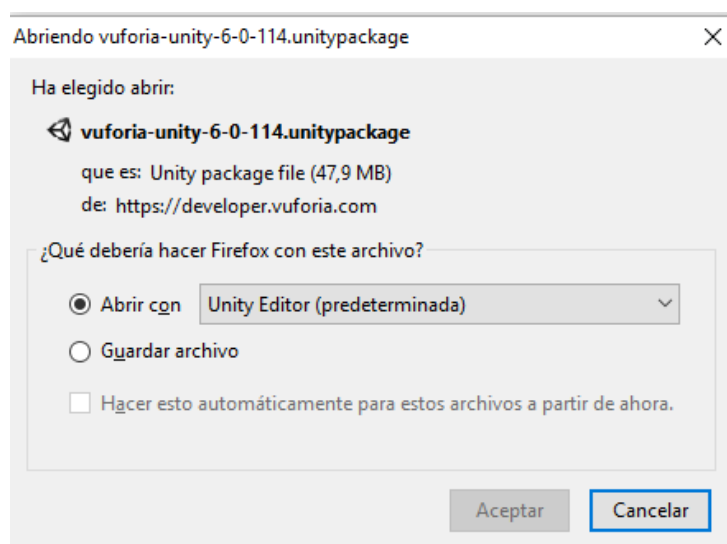


Figura I 5: Descarga del paquete Vuforia. Fuente: elaboración propia.

Paso 3: Crear un nuevo proyecto 3D en Unity y eliminar los elementos generados automáticamente.

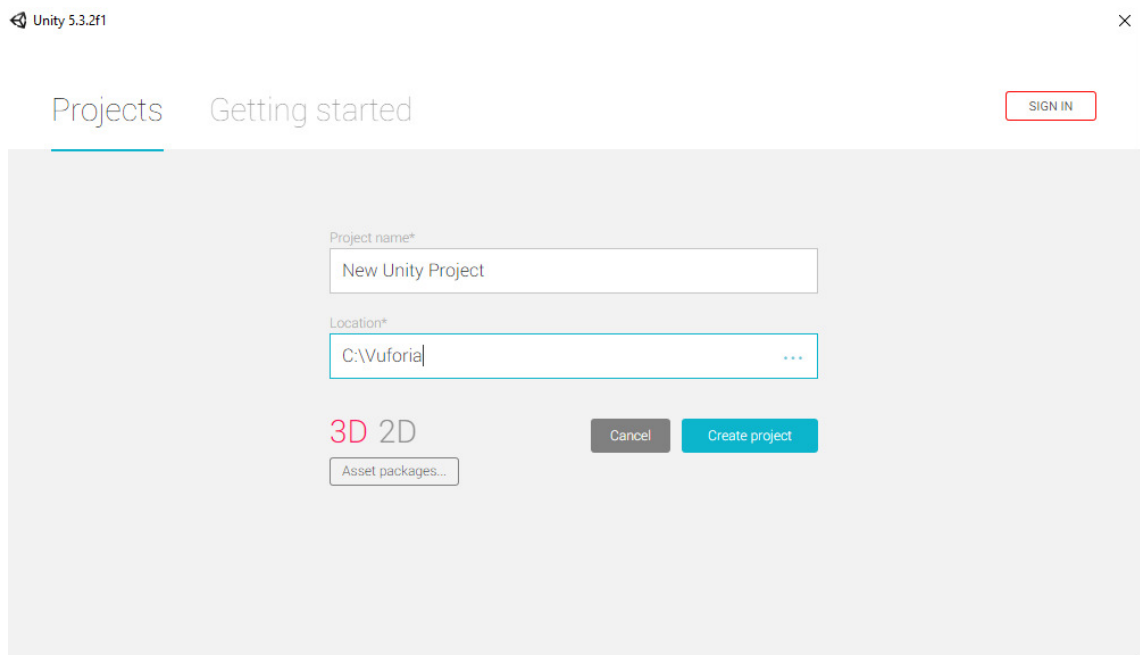


Figura I 6: Pantalla de creación de nuevo proyecto. Fuente: elaboración propia.

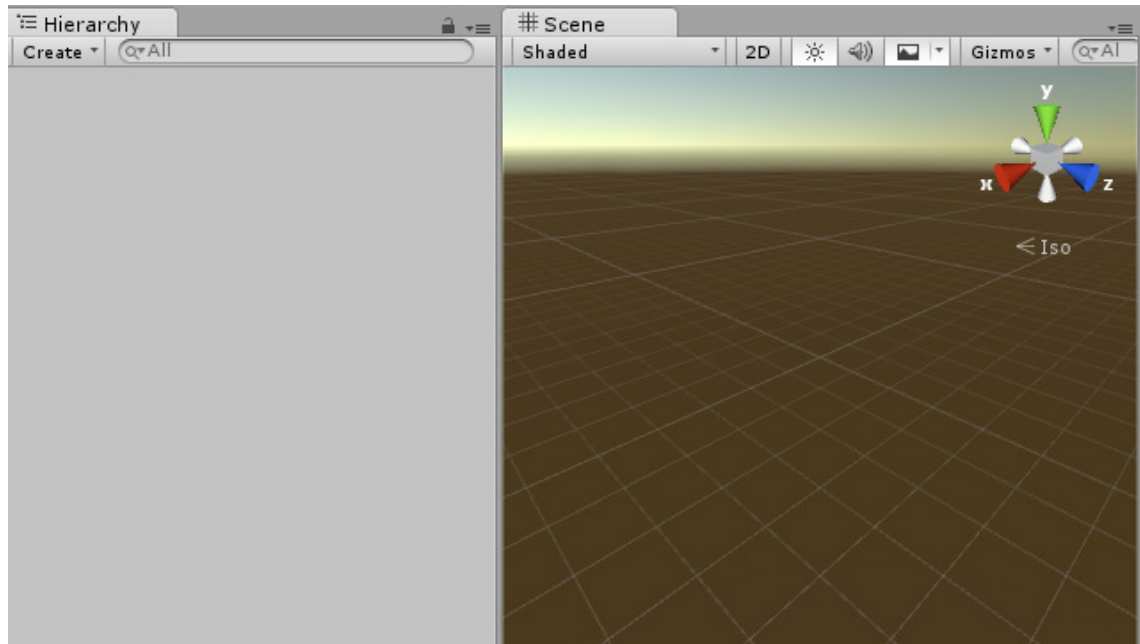


Figura I 7: Proyecto sin elementos. Fuente: elaboración propia.

Paso 4. Importar un nuevo paquete personalizado desde el motor de Unity y seleccionar el paquete de Vuforia descargado. Una vez que el paquete se haya importado, seleccionar todos los componentes en la nueva ventana y pulsar sobre el botón importar.

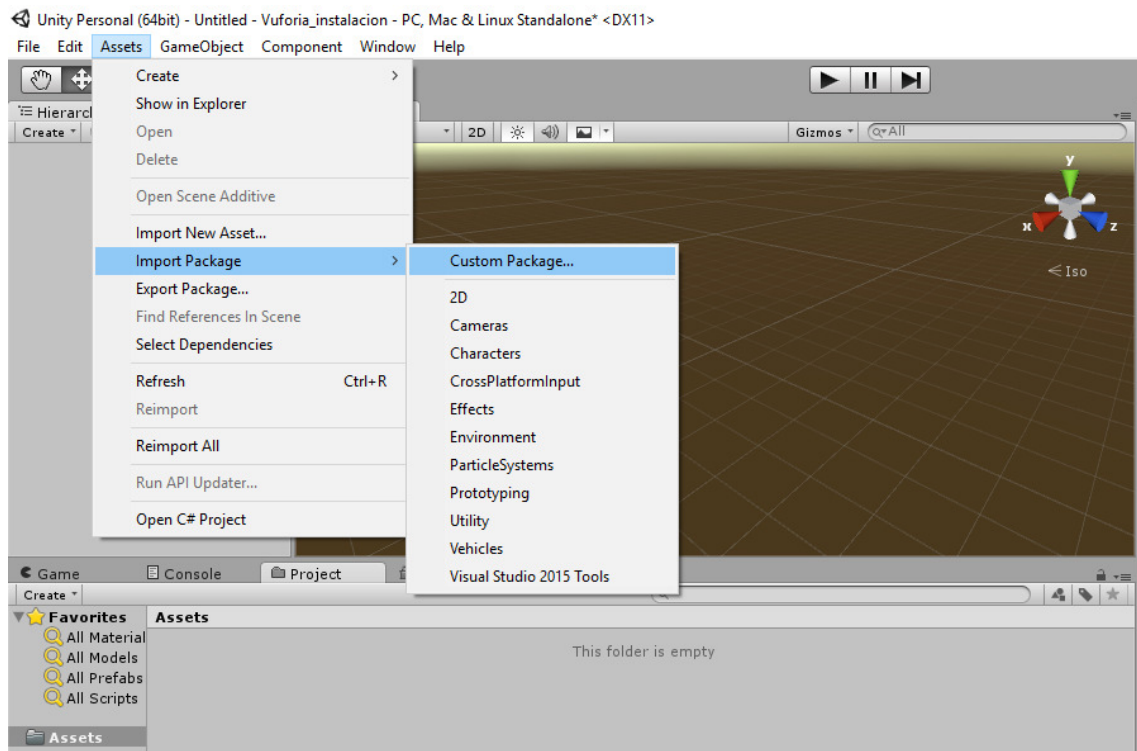


Figura I 8: Menú de importación de paquetes personalizados. Fuente: elaboración propia.

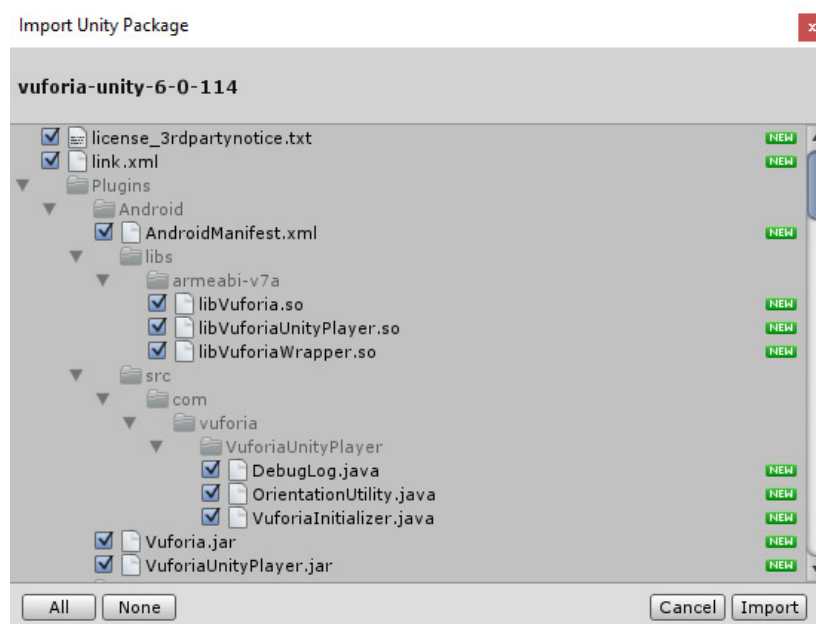


Figura I 9: Importar el paquete Vuforia. Fuente: elaboración propia.

I3. Configuración de Vuforia en Unity.

Paso 1: Acceder a la sección develop del sitio web de Vuforia habiéndote identificado previamente con tu cuenta. Una vez dentro, ingresa en la sección license manager y crea una nueva licencia, nosotros elegimos la de desarrollo.

Figura I 10: Creación de licencia de Vuforia. Fuente: elaboración propia.

Paso 2: De vuelta en la sección license manager, pulsar sobre la licencia que se acaba de crear y copiar el código que se muestra en una ventana nueva. Éste código debe ir en la cámara de Realidad Aumentada dentro del motor de Unity.

Figura I 11: Licencia de Vuforia. Fuente: elaboración propia.

Paso 3: Desde el motor Unity, crear un objeto nuevo de tipo ARCamera, posicionarla en la escena (pulsar y arrastrar hacia la escena con el ratón), y pegar el código de licencia en la parte del inspector, en el campo App License Key.

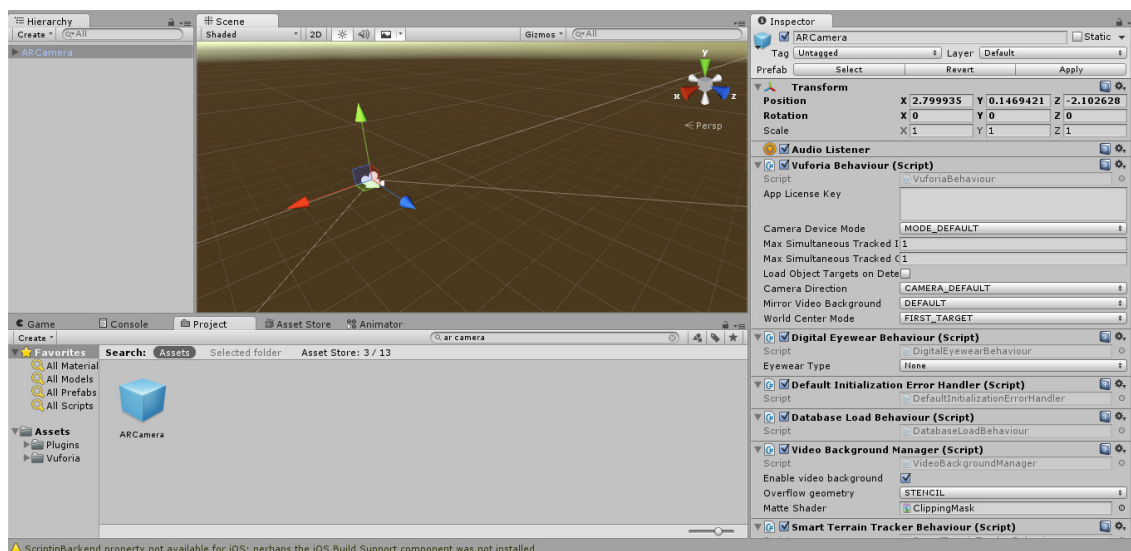


Figura I 12: Configurando Vuforia sobre Unity. Fuente: elaboración propia.

14. Creación de bases de datos de imágenes objetivo.

Paso 1: De forma similar a la creación de una licencia, acceder a la sección develop del sitio web de Vuforia habiéndote identificado previamente con tu cuenta. Una vez dentro, ingresa en la sección target manager y crea una nueva base de datos de tipo dispositivo (device).

Paso 2: Vuelve a la sección target manager, selecciona la base de datos creada y pulsa sobre add target. Selecciona el tipo de objetivo (en nuestro caso single image), examina el archivo con la imagen que deseas, agrega el nombre y pulsa sobre añadir.

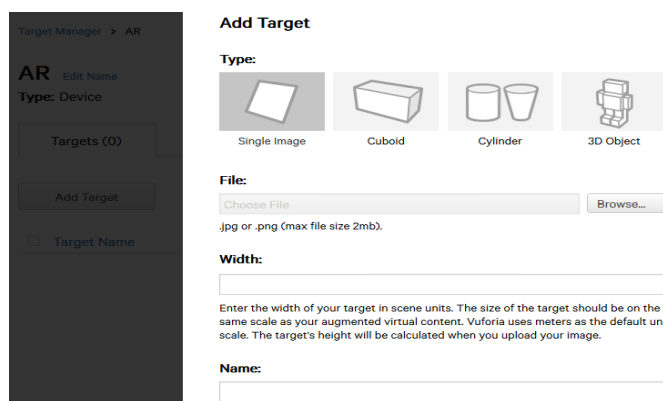


Figura I 13: Añadiendo un nuevo objetivo. Fuente: elaboración propia.

Paso 3: Vuelve a la sección target manager, accede de nuevo a la base de datos que has creado y pulsa sobre el botón Download Database (All). Se iniciará la descarga de un nuevo paquete para Unity con la base de datos que se importa de la misma manera que el paquete Vuforia, descrito en el subcapítulo I2 de este anexo.

Paso 4: Selecciona el componente ARCamera en Unity, en el inspector despliega el script Database Load Behaviour y marca la base de datos que has importado y actívala.

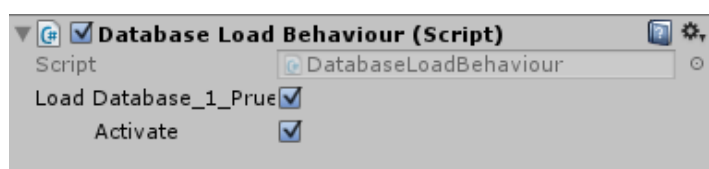


Figura I 14: Activando la base de datos en Unity. Fuente: elaboración propia.

I5. Instalación del SDK de Android y generación de la aplicación.

Paso 1: El SDK de Android tiene dependencias del JDK de Java, por lo que primero debemos instalar el JDK (en su última versión a ser posible) accediendo al sitio web de Java a través de <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> y descargando la versión correspondiente a nuestro sistema operativo. La instalación del JDK es sencilla y tan sólo hay que seguir los pasos del instalador.

Java SE Development Kit 8u101		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
☐ Accept License Agreement ☒ Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe

Figura I 15: Menú de descargas del JDK de Java. Fuente: elaboración propia.

Paso 2: acceder al sitio web de desarrollo de Android a través de <https://developer.android.com/studio/index.html> y descargar el paquete recomendado. Una vez descargado, la instalación es sencilla y únicamente hay que seguir los pasos del instalador.



Figura I 16: Descarga del instalador del SDK de Android. Fuente: elaboración propia.

Paso 4: Buscamos en nuestro equipo el programa Android SDK Tools, y una vez dentro, seleccionaremos la opción SDK Manager. Cuando esté abierto, debemos seleccionar uno o varios de los repositorios disponibles y pulsamos sobre Install packages.

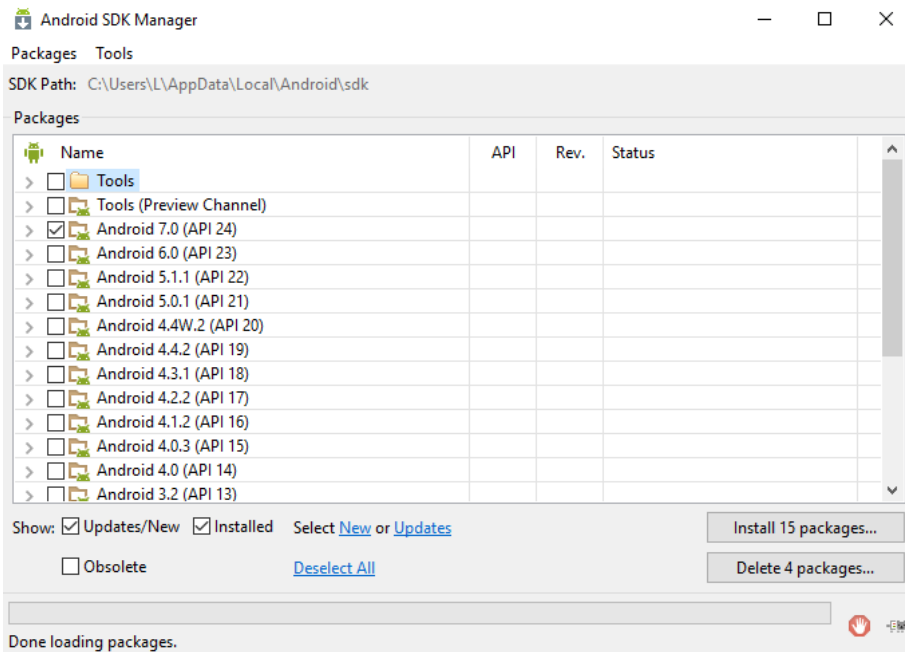


Figura I 17: Android SDK Manager. Fuente: elaboración propia.

Paso 5: En Unity, nos dirigimos al menú File, Build Settings y seleccionamos la plataforma Android. Tras ello, pulsamos sobre el botón Add Open Scenes y luego en el botón Build para generar la aplicación. Se puede configurar los parámetros de la aplicación pulsando en esta misma ventana sobre Player settings.

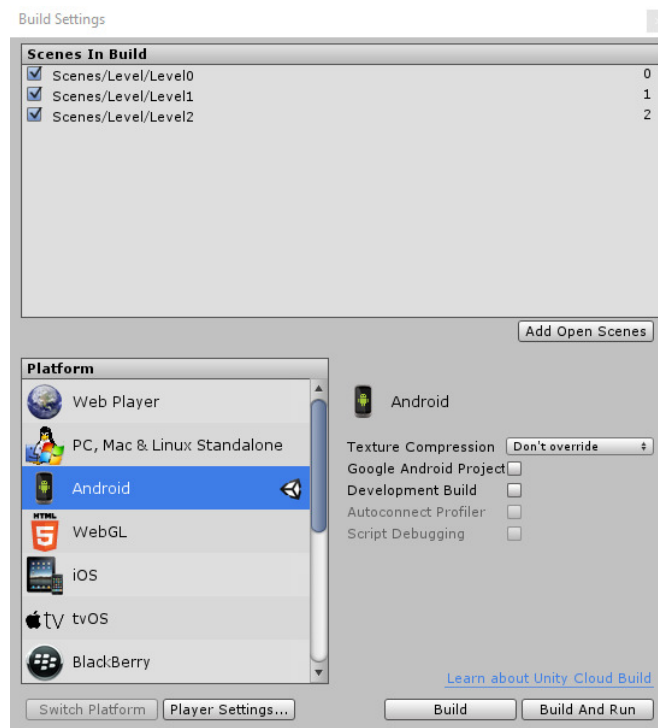


Figura I 18: Exportando el juego a Android. Fuente: elaboración propia.

Anexo II: Manual de usuario.

En este anexo se va desarrollar el manual de usuario donde se pretende realizar un esquema de uso del prototipo.

1. Al iniciar la aplicación nos encontramos con una ventana en la que se muestran las instrucciones del juego y un botón de empezar para comenzar la carrera. El temporizador no se inicia hasta que se pulsa sobre el botón. Cuando el jugador haya entendido todas las reglas correctamente puede empezar el juego.

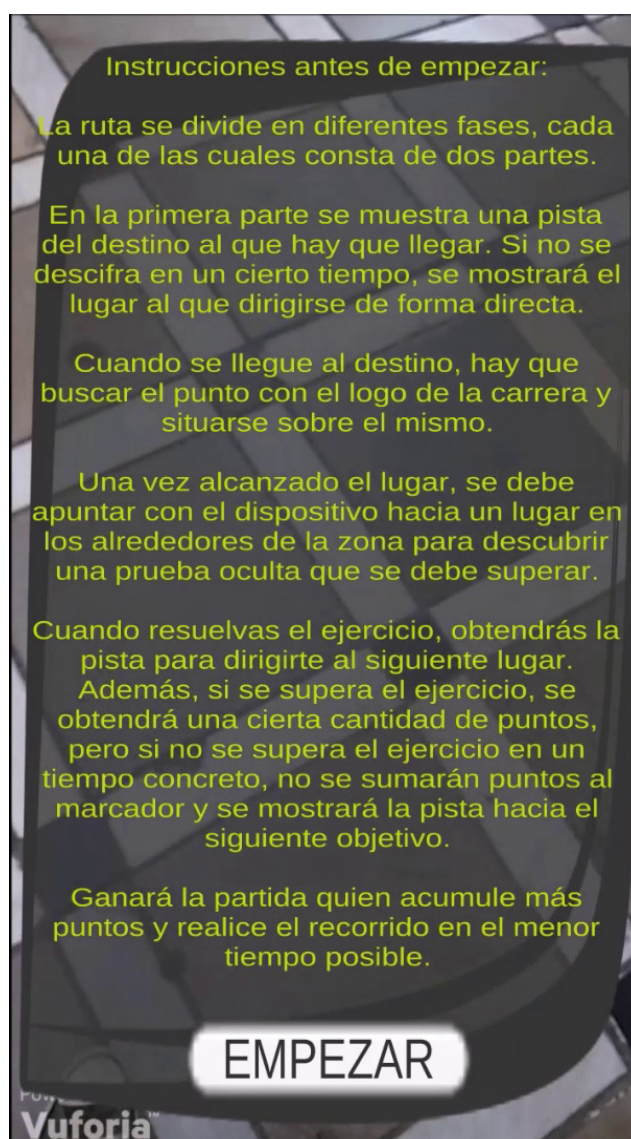


Figura II 1: Pantalla de bienvenida con las instrucciones del juego. Fuente: elaboración propia.

2. El jugador recibe una pista sobre un punto al que debe dirigirse. Esta pista está relacionada con un aspecto del turismo del lugar. El jugador debe adivinar a qué lugar dirigirse y caminar hasta encontrar el símbolo de la carrera.

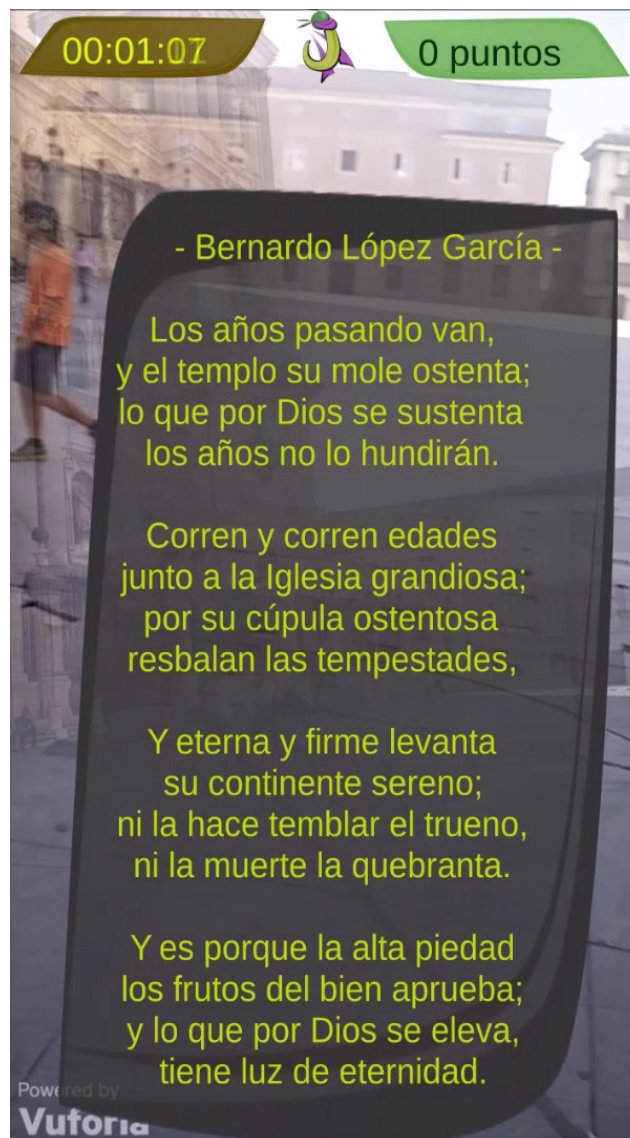


Figura II 2: Pista con información sobre el siguiente objetivo. Fuente: elaboración propia.

3. Cuando el jugador se encuentra sobre el logo de la carrera, debe utilizar su terminal para escanear el entorno en busca de un ejercicio oculto que, si resuelve, le indicará una nueva pista y le hará ganar puntos en la competición.

Si no lo resuelve, podrá seguir jugando, pero no sumará puntos en este tramo del recorrido.

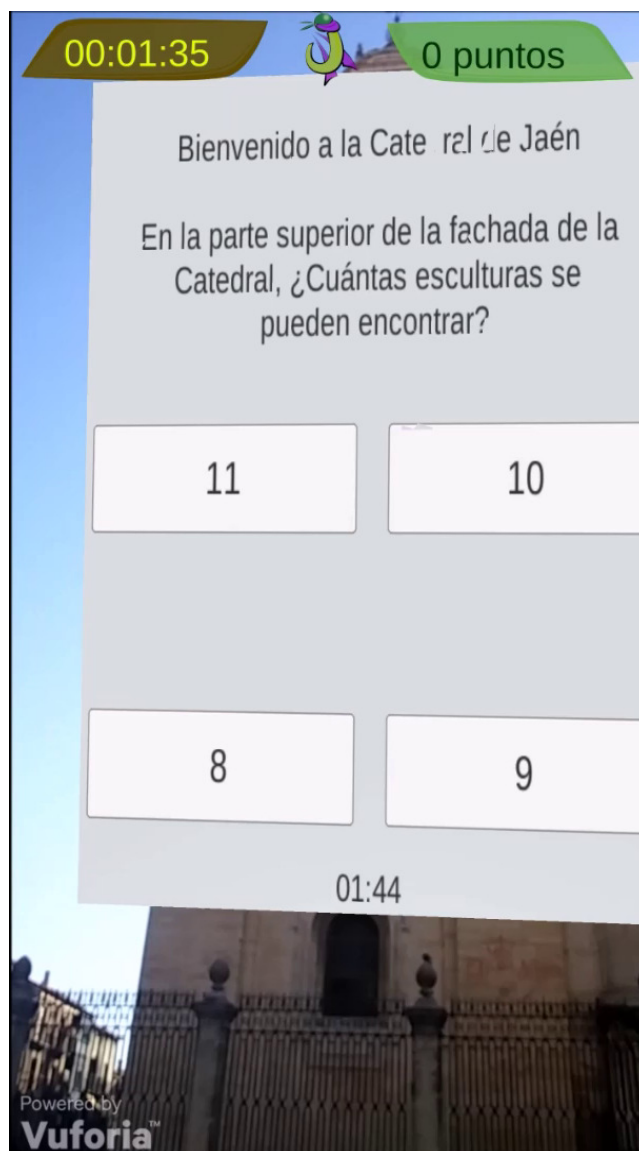


Figura II 3: Ejercicio oculto en la primera etapa. Fuente: elaboración propia.

4. El jugador recibirá una nueva pista sobre un nuevo destino al que dirigirse. El proceder de esta nueva etapa es similar a las fases anteriores. Así, el jugador deberá descifrar a través de las pistas o esperando un tiempo determinado cuál es el siguiente destino, alcanzarlo, utilizar su terminal para escanear el entorno y realizar un nuevo ejercicio que le permitirá continuar.



Figura II 4: Pista del nuevo destino tras la prueba de la etapa. Fuente: elaboración propia.



Figura II 5: Destino mostrado de forma directa tras un período de tiempo.
Fuente: elaboración propia.