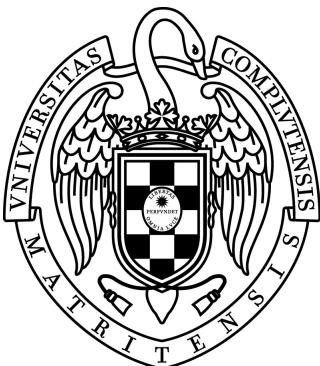

Realidad aumentada para el Museo García-Santosmases



Trabajo de Fin de Grado en Ingeniería del Software

Raúl Cobos Hernando
María Picado Álvarez
Álvar D. Soler Rus

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Junio 2016

Documento maquetado con TEXIS v.1.0+.

Este documento está preparado para ser imprimido a doble cara.

Realidad aumentada para el Museo García-Santosmases

Trabajo de Fin de Grado en Ingeniería del Software

Raúl Cobos Hernando

María Picado Álvarez

Álvar D. Soler Rus

Dirigida por el Doctor

Dr. Guillermo Jiménez Díaz

**Departamento de Ingeniería del Software e Inteligencia
Artificial**

Facultad de Informática

Universidad Complutense de Madrid

Junio 2016

Copyright ©

Contenido: Raúl Cobos Hernando, María Picado Álvarez y Álvar D. Soler Rus

Plantilla : Marco Antonio y Pedro Pablo Gómez Martín

ISBN 978-84-692-7109-4

A nuestras familias

Autorización

Nosotros, Raúl Cobos Hernando, María Picado Álvarez y Álvar D. Soler Rus, alumnos matriculados en la asignatura Trabajo de Fin de Grado (TFG) en la Facultad de Informática de la Universidad complutense de Madrid durante el curso 2015/2016, dirigidos por Guillermo Jiménez Díaz, autorizamos la difusión y utilización con fines académicos, no comerciales, y mencionando expresamente a sus autores del contenido de esta memoria, el código, la documentación adicional y el prototipo desarrollado.

Raúl Cobos Hernando, María Picado Álvarez y Álvar D. Soler Rus

Agradecimientos

*I don't care if it works on your machine!
We are not shipping your machine!*

Vidiu Platon.

A quien proceda.

Resumen

La Realidad Aumentada es una tecnología que combina imágenes reales con la superposición de imágenes virtuales. En esta memoria se detalla el trabajo hecho con esta tecnología en la creación de varios minijuegos para dotar al Museo García Santesmases de un atractivo añadido al de los objetos físicos ya expuestos. Veremos cómo ha sido el proceso de desarrollo de la aplicación, la toma de decisiones y los problemas que hemos encontrado. El nombre de la aplicación es Santesmases RA, y está disponible en Play Store para que cualquiera que visite el museo la pueda descargar y usar.

Palabras clave: Realidad Aumentada, Museos, Unity3D, Vuforia, Android, Videojuegos

Abstract

Augmented Reality is a technology that combines real images with overlapping virtual images. In this report we detail the work done with this technology in the creation of various mini-games to provide to the Museo García Santesmases an attractive addition to the exposed physical objects. We will see how was the application's development process, decision-making and the problems we have encountered. The application's name is Santesmases RA, and it is available in Play Store so anyone who visits the museum can download and use.

Keywords: Augmented Reality, Museums, Unity3D, Vuforia, Android, Videogames

Índice

Agradecimientos	VII
Resumen	IX
1. Introducción a Santesmases RA	1
1.1. Museo de informática García-Santesmases	1
1.2. Objetivos y motivación	2
1.3. Antecedentes	3
Notas bibliográficas	3
2. Estado del Arte	5
2.1. La realidad aumentada	5
2.2. Alcance	6
2.2.1. Información interactiva	6
2.2.2. Entretenimiento y educación	7
2.2.3. Ciencia y desarrollo	8
2.2.4. Otros	8
2.3. Realidad aumentada en videojuegos	9
2.3.1. Ejemplos de Realidad Aumentada en videojuegos . . .	10
2.4. Realidad aumentada en museos	12
2.4.1. Otros museos	12
2.5. Herramientas de desarrollo	13
Notas bibliográficas	15
3. Diseño del videojuego	17
3.1. Introducción	17
3.2. Hilo argumental	18
3.3. Mecánicas	19
3.4. Conclusiones	20
Notas bibliográficas	20
4. Space Invaders	21

4.1. Historia	21
4.2. Nuestra versión	21
4.2.1. Diseño	22
4.2.2. Implementación	23
4.2.3. Conclusiones	25
Notas bibliográficas	27
5. Arkanoid	29
5.1. Historia	29
5.2. Nuestra versión	29
5.3. Implementación	29
5.4. Conclusiones	29
Notas bibliográficas	29
6. Water Pipes	31
6.1. Historia	31
6.2. Nuestra versión	31
6.3. Implementación	31
6.4. Conclusiones	31
Notas bibliográficas	31
En el próximo capítulo	32
7. Detalles de implementación	33
7.1. Realidad aumentada	33
7.1.1. Vuforia	33
7.1.2. Unity	34
7.1.3. Unity + Vuforia	35
7.2. Escenas intermedias	36
7.2.1. SALSA with Random Eyes	36
7.2.2. VozMe MP3 generator	37
7.3. Persistencia de puntuaciones	37
7.3.1. Symfony	37
Notas bibliográficas	38
8. Evaluación con usuarios	39
8.1. Plan de evaluación	39
8.1.1. Objetivos de la evaluación	39
8.1.2. Métricas	40
8.2. Descripción de la metodología del análisis de los datos	40
8.2.1. Primera fase	40
8.2.2. Segunda fase	41
8.2.3. Tercera fase	41

8.3. Primera evaluación con usuarios	41
8.3.1. Informe de resultados	42
8.3.2. Acciones	44
8.3.3. Conclusiones	44
Notas bibliográficas	44
En el próximo capítulo	44
9. Conclusiones y trabajo futuro	45
9.1. Conclusiones	45
9.2. Líneas futuras	45
Notas bibliográficas	45
En el próximo capítulo	45
10. Aportaciones individuales	47
10.1. Organización general del proyecto	47
10.2. Raúl Cobos	47
10.3. Álvar D. Soler	48
10.4. María Picado	49
Lista de acrónimos	51

Índice de figuras

1.1. Retrato de José García Santesmases por Eulogia Merle	2
2.1. Cartel publicitario de la empresa CamOnApp	7
2.2. Captura del entorno de desarrollo de la aplicación Aumentaty	8
2.3. Realidad aumentada en medicina	8
2.4. Realidad aumentada en medicina	9
2.5. Aplicación de RA aplicada a la arquitectura	9
2.6. AR Cards para Nintendo3DS	10
2.7. Captura del juego Pokémon Go	11
2.8. Captura del juego Ingress	11
2.9. Captura del juego Paintball	12
2.10. Captura del juego Father.io	12
2.11. Captura del juego Night Terrors	13
2.12. Aplicación del Museo de Mataró	14
2.13. StreetMuseum	15
2.14. Crononautas	15
2.15. Captura de RACMA que muestra las zonas de influencia de las sociedades precolombinas.	16
4.1. Space Invaders original	22
4.2. Diagrama de los GameObjects	24
4.3. Captura durante el proceso de implementación	25
7.1. Ejemplo de aplicación con Smart Terrain	36
8.1. Usuario realizando un prueba del Space Invaders	41

Índice de Tablas

Capítulo 1

Introducción a Santesmases RA

El proyecto que hemos desarrollado tiene como finalidad atraer al público al museo García Santesmases con una característica nueva y atractiva. Para esto, hemos utilizado la Realidad Aumentada (en adelante RA), y con ella, diseñado tres pequeños minijuegos que requieren poco tiempo para ser jugados y dan una visión nueva de lo que la RA y los videojuegos pueden aportar a un museo. Todo esto dentro de una aplicación para móviles Android.

1.1. Museo de informática García-Santesmases

Inaugurado en noviembre del 2003, el museo debe su nombre al físico, profesor y precursor de la informática española José García Santesmases, el cual fue catedrático de la Universidad Complutense. En él, se exponen máquinas desarrolladas por la UCM entre 1970 y 1950. Además, se exponen las computadoras comerciales del Centro de Cálculo de la UCM, aportaciones de particulares, los propios departamentos de la Universidad, etcétera.

Además de computadoras, hay paneles explicativos que muestran información sobre éstas y sobre historia de la informática en general y cuenta también con gran cantidad de bibliografía presente en la biblioteca de la facultad.

El museo cuenta con dos plantas situadas en la 3^a y 4^a planta de la Facultad de Informática y su pieza más significativa es el “Analizador diferencial electrónico”, diseñado por García Santesmases y es el primer computador desarrollado en España.

Las visitas al museo son libres y cualquiera puede acercarse a la Facultad de Informática y recorrer sus pasillos, pero si se prefiere ir en grupo, se puede concertar una cita a través del email del museo. Además, a lo largo del curso se organizan visitas guiadas por el museo.

Toda la información necesaria sobre el museo se puede encontrar en la web: <http://www.fdi.ucm.es/migs/>.

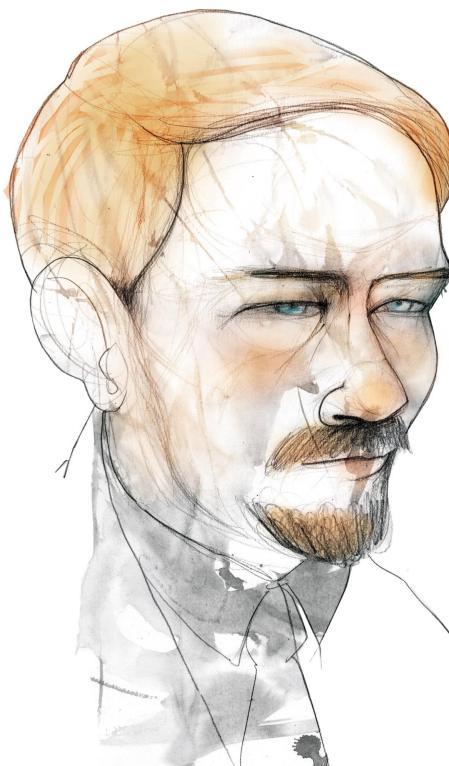


Figura 1.1: Retrato de José García Santesmases por Eulogia Merle

1.2. Objetivos y motivación

Nuestro objetivo es el desarrollo de una aplicación que mejorará la experiencia del usuario en un museo, en este caso, el museo de la Facultad de Informática “García-Santesmases”. Pero no solo esto, sino que nuestro objetivo es hacerlo a través de los videojuegos y utilizando la RA.

Esto lo logramos mediante una “yincana”, guiando al visitante a que recorra el museo en busca de misiones que tendrá que ir completando minijuegos para poder pasar a la siguiente misión. Así, al finalizar la visita, el jugador habrá recorrido el museo de una forma amena y divertida.

Nuestra motivación principal en la realización de este proyecto, fue profundizar en el desarrollo de videojuegos con una herramienta tan innovadora como lo es la RA. Para poder cumplir con los objetivos citados anteriormente, debemos antes centrarnos en investigar y recopilar información sobre distintos temas.

- Para la realización de este proyecto, es fundamental conocer los trabajos ya existentes en museos con RA y más específicamente los que añaden a esto los videojuegos en su proyecto. Esta diferencia es impor-

tante, debido a que los museos están apostando por la RA de muchas formas diferentes. Pero al estar nuestro trabajo enfocado a los videojuegos, creemos importante centrarnos en este punto más detenidamente.

- Existen diversas formas de incluir los videojuegos en un museo para dinamizar la visita al usuario. Como en nuestro proyecto decidimos hacer una yimcana, otro objetivo que debemos cumplir es volver a investigar, pero esta vez, para recopilar la mayor información posible sobre cómo crear una yimcana en un museo, cuál puede ser la mejor manera de ir guiando al visitante por el museo y cómo decidir qué partes del museo son las mejores para hacer captar la atención del visitante en ese lugar en concreto.
- Otra decisión importante que tenemos que tomar es que minijuegos van a componer nuestra yimcana. No podemos solo pensar en qué minijuegos queremos implementar, sino también debemos tener en cuenta los puntos anteriores para pensar en esto. Visitar el museo para ver en qué partes de éste se pueden incluir los juegos nos puede ayudar mucho para saber cuales son los juegos que queremos adaptar a la RA e incluir en nuestra yimcana.

1.3. Antecedentes

El proyecto que hemos realizado para este TFG, es un proyecto nuevo y que ha sido diseñado e implementado desde el principio por nosotros. En años anteriores se realizaron trabajos de fin de grado dedicados a la RA en museos, como el realizado el año pasado (2014/2015) para el Museo de América. Nuestro proyecto guarda muchas similitudes con el citado anteriormente, como el uso de RA en museos para mejorar la experiencia del visitante. Por tanto, este trabajo puede que sea nuestro antecedente, aunque el concepto de proyecto sea distinto, ya que ellos utilizaban la RA como medio de información, mientras que nosotros añadimos los videojuegos en RA como medio de entretenimiento en la visita.

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (?)

Capítulo 2

Estado del Arte

La Realidad Aumentada es el término que se usa para definir una visión a través de un dispositivo tecnológico, directa o indirecta, de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real.

Este tipo de tecnología se está utilizando cada vez más en distintos museos, para dinamizar su visita.

2.1. La realidad aumentada

Según Ronald Azuma, desarrollador y líder de proyecto en New Media, Intel Corporation, y uno de los pioneros en el campo de la realidad aumentada, dice que la RA consta de las siguientes características:

- Combina elementos reales y virtuales.
- Es interactiva en tiempo real.
- Está registrada en 3D.

A su vez, consta de unos requisitos mínimos para poder ser emulada, los cuales son:

1. Una pantalla, donde mostrar la combinación de los elementos reales captados por algún dispositivo y los elementos virtuales generados por un software.
2. Un conjunto de dispositivos que capturen los elementos del entorno y nuestra situación como son una cámara, acelerómetro, giroscopio... de tal forma que permita al software tener referencias de cómo y dónde debe mostrar sus elementos virtuales.

3. Un hardware relativamente potente, para poder realizar los cálculos necesarios para mostrar el entorno que captura la cámara y ser capaz de hacer frente al software que genera los elementos virtuales y combinarlos en la pantalla con los reales.
4. Un software capaz de reconocer el entorno y calcular donde y como debe representar los elementos virtuales combinados con los reales para conseguir una visión de RA.

La evolución de los dispositivos móviles junto al aumento de su uso en la sociedad, han permitido que casi cualquier persona pueda tener acceso a la realidad aumentada.

2.2. Alcance

La RA es una tecnología que, aunque ya lleva muchos años, no se conocía a nivel usuario. Pero el incremento tan alto del uso de los Smartphones en los últimos años en la sociedad ha permitido crear más aplicaciones que todo el mundo pueda utilizar, ya que hoy en día casi cualquier persona de una edad comprendida entre los 16 y 55 años tiene un dispositivo móvil que le permite ejecutar una aplicación de RA en cualquier lugar y momento.

Ahora mismo, tiene diferentes aplicaciones en diversos campos como son:

2.2.1. Información interactiva

En este caso se utiliza para dar a conocer de forma interactiva información acerca de un elemento cercano al usuario, de tal forma que se puede mostrar un tipo de información u otra en función de la interacción entre el usuario y dicho elemento.

Ahora mismo este área está muy presente en museos como forma dinámica de conseguir una inmersión del usuario con lo que está viendo y hacer de su visita una experiencia más atractiva e incluso hasta más productiva.

Dentro de este campo también se hace uso de la RA en herramientas utilizadas para el intercambio de información en proyectos profesionales o con fines comerciales como el de mostrar catálogos de productos de una forma interactiva.

Un buen ejemplo de esto puede ser la empresa CamOnApp, que ha creado una aplicación con el mismo nombre, que crea contenidos interactivos para otras empresas. La aplicación se encarga de escanear los carteles, y te mostrará la información correspondiente. Pero todas ellas tienen el mismo elemento en común, la RA como plataforma para llegar al público.



Figura 2.1: Cartel publicitario de la empresa CamOnApp

2.2.2. Entretenimiento y educación

En la actualidad, aunque no está todavía muy popularizado su uso, existen una gran cantidad de videojuegos que hacen uso de esta tecnología que, junto con diferentes mecánicas que se adaptan a ella, generan una novedosa experiencia para entretener el usuario.

Aunque en secciones posteriores, desarrollaremos más detenidamente este punto mostrando algunos ejemplos del uso de la RA en videojuegos, creemos interesante poner un ejemplo de este uso, pero más centrado en el entretenimiento y su aplicación en el ámbito educativo. Los profesionales de este sector, se están interesando en la RA, con el fin de hacer los contenidos académicos que deben de impartir a sus alumnos, más entretenidos y atractivos. Con esto consiguen que la motivación de sus alumnos aumente, y con ella la capacidad de aprender y así el resultado sea que se aprenda lo mismo pero de una manera más amena y divertida.

Existen diversas aplicaciones que permiten al profesor subir contenido y crear apps en realidad aumentada para llevar a cabo los aspectos anteriores, por parte de los alumnos y sobre todo de los profesores.

El proyecto Aumentaty es una idea del grupo de investigación LabHuman de la Universidad Politécnica de Valencia. Y lo que busca es potenciar el uso de los contenidos interactivos y con ello la RA, en las aulas de los colegios. Para esto, pone a disposición de los alumnos y los profesores, software gratuito que permite la creación de aplicaciones en RA de una forma sencilla.



Figura 2.2: Captura del entorno de desarrollo de la aplicación Aumentaty

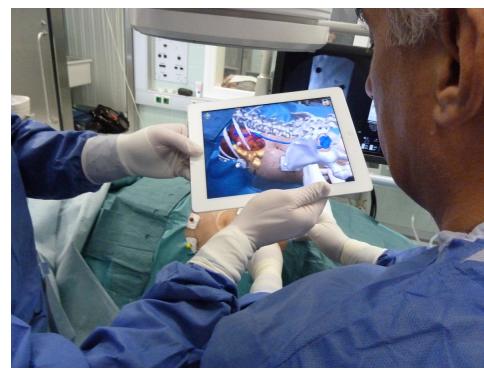


Figura 2.3: Realidad aumentada en medicina

2.2.3. Ciencia y desarrollo

Aunque todavía no está normalizado el uso de la RA en este campo, si están naciendo numerosos proyectos con el fin de desarrollar esta tecnología para utilizarse en áreas como la medicina.

Especialmente en el campo de la medicina, la inclusión de la RA puede suponer un gran avance. La unión del mundo real con elementos virtuales, puede convertirse y de hecho lo está haciendo, en una buena fórmula para facilitar la labor al especialista médico. Esto se consigue, debido a que la RA se convierte en una forma de ver la medicina nunca vista antes. Imágenes con esta tecnología, pueden ayudar al médico a tomar la mejor solución posible.

A continuación mostramos algunas de estas imágenes, que nos sirven de ejemplo de lo citado en las líneas anteriores

2.2.4. Otros

Además de los ya mencionados, existen todavía muchísimas áreas en las que tienen cabida en sus tecnologías la RA.

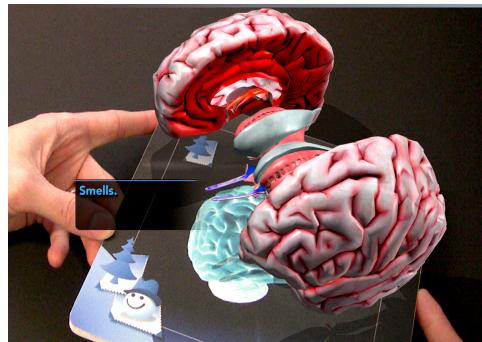


Figura 2.4: Realidad aumentada en medicina



Figura 2.5: Aplicación de RA aplicada a la arquitectura

El marketing y la publicidad están cada vez más incluyendo la RA en sus proyectos, esto se debe a que la RA es algo que suele llamar mucho la atención de las personas, lo que es en definitiva el objetivo principal de este sector.

Otros como la arquitectura, se benefician de esta tecnología al poder crear maquetas de edificios o realizar proyectos en espacios abiertos, todavía no construidos.

2.3. Realidad aumentada en videojuegos

La RA está siendo un gran descubrimiento para distintos ámbitos en el mundo de la informática y la tecnología. Pero, el sector que más interesado está en integrar la RA en sus desarrollos, para hacerla llegar a gran parte de la sociedad, es el de los videojuegos.

En los últimos años, han aparecido nuevas formas de jugar, y todas ellas intentan mejorar la experiencia del jugador, ya sea evadiéndolo de la realidad inventando un mundo y un espacio completamente nuevo, como hace La Realidad Virtual, o como es en nuestro caso, añadiendo información al mundo



Figura 2.6: AR Cards para Nintendo3DS

real, con la RA. Esta última consigue que el jugador maximice la interacción con el juego, este motivo es lo que la hace tan atractiva para la industria del videojuego.

2.3.1. Ejemplos de Realidad Aumentada en videojuegos

Hoy en día, la industria del videojuego está permanente creciendo, y la competencia es tan grande que siempre se están buscando nuevas formas de llamar la atención de los jugadores. Las nuevas tecnologías como la Realidad Virtual o la Realidad Aumentada son de las más utilizadas por las empresas para esto.

A continuación, se muestran algunos ejemplos de videojuegos con RA.

- Tarjetas RA Nintendo 3DS: Nintendo integra, desde su lanzamiento en 2011, los juegos en RA en su nueva consola. Puedes jugar a distintos minijuegos, utilizando las 6 tarjetas AR Cards. El jugador tiene también la posibilidad de desbloquear nuevos minijuegos. A parte de estas aplicaciones preinstaladas en la consola, Nintendo ha ido sacando distintos juegos que también utilizan las AR Cards.
- Pokémon Go: En esta versión, los jugadores podrán salir a explorar su ciudad e ir en busca de nuevos pokémon que se esconden por escenarios reales. Los encargados del desarrollo de este juego es la empresa Niantic Labs.
- Ingress: Este juego es otro ejemplo de desarrollo de Niantic Labs. Pero en esta ocasión, se trata de un juego de rol donde el jugador debe decidir en qué bando se encuentra, y capturar los portales.
- Pintball: Este juego creado por Mambo Studios, utiliza la realidad aumentada para simular una partida del juego con el mismo nombre, pero



Figura 2.7: Captura del juego Pokémon Go



Figura 2.8: Captura del juego Ingress

con la gran ventaja de poder jugarlo en cualquier lugar, sin necesidad de ninguna equipación especial.

- Father.io: es un juego clasificado como shooter multijugador. Donde el objetivo del jugador es luchar en las calles, para conseguir capturar edificios e ir recolectando recursos para su equipo. Lo único necesario para poder jugar es un smartphone, pero se tiene la posibilidad de adquirir un Inceptor, un dispositivo que se adhiere al smartphone y que lo convierte en un arma láser.
- Nighth Terrors: Es un juego Survival Horror al que se le añade la RA, lo que da aumenta la sensación de miedo en el jugador. Para poder jugar, solo necesitas una casa a oscuras y un smartphone, con esto el juego se encarga de ir reconociendo los elementos de la casa o el lugar en el que te encuentres, para integrarlos al juego. El objetivo del jugador es salvar a una niña y salir con vida.

Como hemos podido observar, existen una gran variedad de juegos que utilizan RA. y aunque pueda parecer una tecnología muy joven, su crecimiento ha sido muy significativo.



Figura 2.9: Captura del juego Paintball



Figura 2.10: Captura del juego Father.io

2.4. Realidad aumentada en museos

La RA ha supuesto un avance muy importante en los museos ya que aporta nuevas experiencias para los visitantes. Se usan diferentes formas de usar la RA en el museo, desde aportar información adicional de un objeto expuesto (enlace a un vídeo, descarga de contenido extra, más información en texto, audios...) hasta búsquedas del tesoro, yincanas como la desarrollada en este trabajo... todo ello son experiencias añadidas y nuevas para la mayoría de museos.

A parte de los beneficios para el público, para el museo son formas de añadir contenido muy baratas en relación con lo que puede costar hacer obras para añadir salas, expositores... Además, se puede añadir contenido adicional a las aplicaciones, consiguiendo que el público vuelva. Sí se puede contemplar la posibilidad de tener smartphones y/o tabletas para el público visitante, una red WiFi abierta para facilitar la descarga de la aplicación.

2.4.1. Otros museos

Vamos a ver algunos de los museos que aplican la RA:



Figura 2.11: Captura del juego Night Terrors

- Centro de artes Ca l'Árenas del Museo de Mataró: para la exposición Mar de Fons el museo dispone de una aplicación que, al enfocar a cualquiera de los cuadros, se muestra información extra.
- Street Museum del Museo de Londres: una de las más famosas. Añade contenidos en el exterior del museo. Básicamente, nos permite ver fotografías antiguas de los sitios donde estamos, dándonos una visión de cómo era mientras vemos cómo es ahora ayudándose del posicionamiento GPS del dispositivo.
- Crononautas del Museo Thyssen-Bornemisza de Madrid: esta aplicación pone al usuario como protagonista de una aventura con toques de ciencia ficción donde el usuario debe tomar decisiones que afectarán a la aventura. Además de servir de hilo conductor para la visita en el museo, aporta contenido adicional sobre las obras.
- RACMA del Museo de América de Madrid: aplicación desarrollada por compañeros de esta misma facultad el curso pasado. En ella se ofrece información a los usuarios sobre distintas sociedades precolombinas. Además, nos muestra sobre un gran mapa mudo que se encuentra en el museo el lugar de influencia de varias de esas tribus y se puede interactuar con los personajes.

2.5. Herramientas de desarrollo

En este apartado explicaremos las herramientas que existen para poder implementar RA.

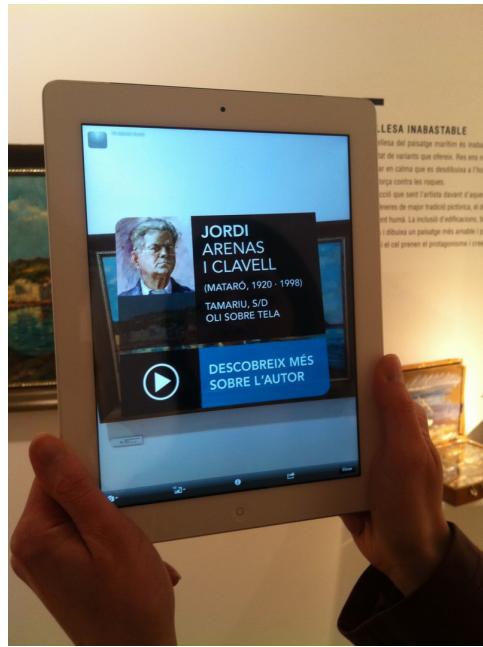


Figura 2.12: Aplicación del Museo de Mataró

- Vuforia: es el framework que hemos utilizado para hacer toda la parte de RA. Es gratuito y tiene una gran comunidad, así como buenos ejemplos y tutoriales. Facilita mucho el trabajo, y se puede utilizar con diferentes SDK (Android, iOS, Unity 3D, ahora con gafas de realidad virtual también...).
- Layar: Proporciona un SDK con soporte para Android e IOS que permite integrar la RA en nuestras aplicaciones. Es de pago aunque tiene una versión de prueba de 30 días. Aunque no es una de las más potentes, si que es una de las más sencillas ya que contienen un montón de utilidades dentro del propio SDK. No está orientada a los videjuegos sino a la muestra de información interactiva basándose en la RA.
- ARTool Kit: Es de código abierto y es gratuita. Tiene soporte para múltiples SDK como son Mac OS X, Linux, Android e IOS, a través del plug-in ARToolkit para Unity3D. Tiene una gran comunidad detrás y la documentación está bastante completa donde incluyen tutoriales, APIs detalladas y multitud de ejemplos.¹

¹<http://www.hitl.washington.edu/artoolkit/>



Figura 2.13: StreetMuseum



Figura 2.14: Crononautas

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (?)



Figura 2.15: Captura de RACMA que muestra las zonas de influencia de las sociedades precolombinas.

Capítulo 3

Diseño del videojuego

El proyecto que se ha realizado es una yimcana, con videojuegos en RA, por el museo García-Santesmases, de la Facultad de Informática de la Universidad Complutense. Ésta se compone de varios minijuegos en RA y un hilo argumental. Las funciones de este último son las de ir dirigiendo al visitante por el museo, y ayudarle a ir completando las diferentes misiones que se le van a ir presentando para poder finalizar con éxito la yimcana.

3.1. Introducción

La idea principal del proyecto es la diseñar e implementar estos minijuegos para que podamos adaptarlos a la RA, y hacer del museo no solo un espacio que aporte información interesante al usuario sobre la historia de la informática, sino que haga de éste un lugar más interactivo y entretenido para sus visitantes.

En este caso, hemos tenido la suerte de realizar un proyecto sobre videojuegos en RA, en el museo García-Santesmases situado en la facultad de Informática. Esto no brinda la oportunidad de poder entretener al visitante mediante los videojuegos pero es importante destacar que a la vez estamos enseñando en primera persona historia de los videojuegos. Esto lo conseguimos al proponer al visitante que juegue a distintos videojuegos míticos como lo son el Space Invader, el Arkanoid o el Pipe Manía, juegos que hemos elegido para implementar en este trabajo.

Con el fin de hacer uso de la tecnología actual que se nos brinda para el desarrollo de la RA hemos implementado estos tres juegos citados anteriormente. Pero se tiene que tener en cuenta que los 3 son juegos que sus primeras versiones cuentan con más de 20 años de antigüedad, por lo que hemos tenido que adaptarlos para que fuera viable jugarlos con RA, aunque siempre manteniendo su esencia original.

3.2. Hilo argumental

El videojuego nos traslada a una nave espacial que está siendo atacada por unos alienígenas. El piloto de la nave nos pondrá en situación y nos irá pidiendo que realicemos diferentes tareas para que la nave pueda despegar. De ésta forma conseguimos que el jugador se mueva por diferentes sitios del museo, ya que a cada minijuego solo se puede jugar en un punto del museo. Así añadimos dinamismo a la aplicación. Además, hay un sistema de puntuación diferente para cada minijuego y solo una oportunidad para cada minijuego.

Al principio nos pedirá que destruyamos a los alienígenas que nos están atacando. Para ello, debemos salir al exterior de la nave (la puerta de entrada de la facultad), donde comienza el primer minijuego al enfocar el cartel de la Facultad de Informática. Así, comenzamos a jugar al Space Invaders. Conseguiremos puntos cada vez que destruyamos un invasor, y perderemos puntos si no conseguimos destruir todos antes de que destruyan el escudo de la nave.

A continuación, el piloto nos pedirá que limpiemos los restos de las naves invasoras. Para ésto debemos ir a los mandos del robot de reciclaje (la máquina arcade que hay en la tercera planta del museo) y así comenzará el segundo minijuego, el Arkanoid. En este conseguimos puntos cuando la bola choca contra la “chatarra espacial”.

Por último, libres de enemigos y con el camino despejado, solo nos queda un escollo para poder escapar. Los tubos del sistema de refrigeración de la nave no están bien alineados y tenemos que rehacer el camino desde el depósito hasta el motor. Para ésto debemos ir al puesto de mando donde está el panel informativo del sistema de combustión (cartel “Simulador Básico de Centrales térmicas” del otro pasillo de la tercera planta) y reajustarlos. Aquí comienza el tercer y último minijuego, WaterPipes. Éste minijuego va por tiempo, y debemos encontrar un camino para el combustible antes de que se agote. La puntuación, con este minijuego, solo puede disminuir, pero en nuestra mano está que el camino sea lo más corto posible y así se nos reste el mínimo posible de puntos.

Al finalizar, nuestra tarea habrá acabado. Se nos pedirá que introduzcamos nuestro nombre, y este se añadirá a un “Hall de la fama” de todos los usuarios del juego, ordenados por puntuación. Con ésto, hemos querido intentar que el juego sea rejutable por los usuarios, que intentarán conseguir más puntos afinando su puntería o su velocidad para quedar más alto del “Hall de la fama”.

3.3. Mecánicas

Uno de los principales problemas de la RA son las mecánicas. Es complicado adaptar un juego haciendo uso de la RA y elegir las mecánicas apropiadas para que este no se vea perjudicado por ella.

En este caso, se implementan juegos “clásicos”, por lo que un usuario que más o menos haya sido jugador, debiera conocer las mecánicas de estos, por lo que la comprensión inicial de estas no debiera ser el principal problema. Ahora bien, hay que hacerle comprender al jugador como debe ejecutarlas y que estas le resulten cómodas, porque el hecho de utilizar la RA limita las mecánicas considerablemente. En primer lugar, el tener que estar enfocando a un “target” dificulta la acción; el usuario deberá estar apuntando a este y eso restringe aquellas mecánicas que puedan propiciar movimientos bruscos. Otra dificultad reside en la ergonomía; el hecho de tener que apuntar con el smartphone, obliga a coger este de una determinada forma que deja apenas dos dedos libres, los cuales tendrán un área limitada donde poder pulsar. Además, el hecho de que sea un juego para móviles conlleva también limitaciones a la hora de recrear juegos que se jugasen con un mando, o que requieran un amplio campo de visión.

El planteamiento teniendo en cuenta las limitaciones de la RA, sumadas a nuestra inexperiencia en el desarrollo de videojuegos ha sido el de implementar tres videojuegos con no mucha complejidad de desarrollo y con mecánicas muy simples que encajen con la RA.

En primer lugar, un menú principal para permitir al jugador comenzar cuando este lo deseé y consultar las puntuaciones; algo importante como hemos mencionado anteriormente que utiliza una interfaz muy simple con botones grandes. A continuación, un hilo principal compuestos por escenas que le dicen al jugador a dónde tiene que ir para comenzar el juego. Tras estas, cada uno de los juegos donde antes de comenzar cada uno de estos se muestra la información que describe las mecánicas y que le dice a el jugador cuál será el “target”al que deberán apuntar. Y en último lugar el juego en sí, que comenzará al reconocer el “target” y que contienen tres elementos fundamentales:

- El tiempo, que supone un obstáculo para el jugador.
- Control. El jugador interactúa con el juego en el caso del Space Invaders apuntando con el propio dispositivo y pulsando sobre la pantalla; en el caso del Arkanoid deslizando el dedo sobre esta para desplazar la plataforma, y el Water Pipes, donde se pulsa sobre las tuberías para intercambiarlas.
- Puntuación, lo que motiva al jugador no solo a pasarse el juego sino a hacerlo de la mejor forma posible.

3.4. Conclusiones

En general hemos quedado bastante satisfechos con el resultado final. Sí nos hemos dado cuenta que podríamos haber implementado más minijuegos, pero los primeros meses fueron bastante lentos en cuanto a desarrollo.

Creemos que la yimcana es una buena manera para que el usuario del museo se mueva por éste. Y que el interactuar con los elementos del museo mejorarán su experiencia. Incluso es probable que pudiese atraer visitas, que motivadas por el hecho de probar esta tecnología relativamente “novedosa”, visiten el museo.

Por otra parte, se considera un acierto el sistema de puntuaciones, ya que anima al usuario a volver a jugar, y cuando lo hace con la mecánicas aprendidad, uno es capaz de percibir más los detalles de las escenas, los cuales pueden contener referencias del museo lo cual refuerza el objetivo del videojuego.

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (?)

Capítulo 4

Space Invaders

El primero de los juegos de la yimcana. Inspirado en el famoso Space Invaders, el jugador, desde una perspectiva en primera persona, tendrá que disparar a un enjambre de Invaders que pretenden destruir las defensas de la nave donde se desarrolla la historia.

4.1. Historia

Space Invaders es uno de los juegos arcade clásicos más conocidos. La primera versión salió al mercado en 1978, hace casi cuarenta años¹. Uno de los precursores del género shoot ém up. El jugador controla una nave espacial que se mueve horizontalmente y debe hacer frente a hordas de alienígenas enemigos que atacan al jugador disparándole proyectiles. Además, a veces el jugador cuenta con pequeñas construcciones que hacen la labor de búnker donde ponerse a cubierto de los disparos, aunque éstos se van destruyendo.

Space Invaders está ampliamente extendido en la cultura popular ya que es uno de los grandes clásicos, por eso hemos considerado acertado incluirlo en nuestro proyecto.

4.2. Nuestra versión

Nosotros hemos decidido darle un cambio a la jugabilidad del juego. En nuestra versión utilizamos la RA para que la experiencia sea completamente diferente. Nuestro juego arrancará al detectar el cartel de FACULTAD DE INFORMÁTICA (Text Recognition), mostrándonos unos invasores alienígenas sobre el cartel, y unas defensas bajo éste.

Para destruir a los invasores, lo que tenemos que hacer es mover nuestro Smartphone para mover nuestra cámara y pulsar en la pantalla para disparar. Hemos pasado de manejar la nave defensora en tercera persona, a hacerlo

¹,

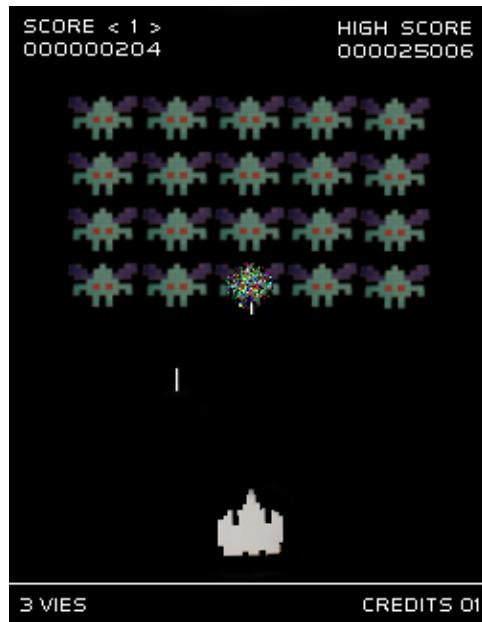


Figura 4.1: Space Invaders original

en primera persona, con un punto de mira en el centro de la pantalla que nos marca en qué dirección irán los láseres de nuestra torreta de defensa, convirtiendo el juego en un First Person Shooter, y tendremos que hacerlo antes de que los enemigos consigan destruir el escudo de nuestra nave espacial (cuando pasa del verde al rojo). Iremos obteniendo puntos según destruyamos naves enemigas, y perderemos puntos al recibir impactos en el escudo, por lo que cuanto más rápidos seamos, más puntos obtendremos.

Con estos cambios, hemos conseguido (a nuestro juicio), transformar un clásico de los arcade en un nuevo juego que utiliza la RA dando una experiencia diferente.

4.2.1. Diseño

El juego se compone de una única escena que contiene todo el juego. Básicamente se compone de:

1. La cámara de Vuforia, que a su vez tiene los siguientes hijos:
 - a) El canvas con la Interfaz de usuario (puntos y mensajes de inicio y fin del juego).
 - b) El punto de mira que utilizamos para apuntar al disparar, que también está hecho con un canvas.

- c) El Cannon (Cañón de disparo) que representa nuestra arma. Básicamente dibuja una línea hacia el infinito para que de la sensación de un puntero láser para apuntar, además, desde su posición se lanza el raycast que calcula las colisiones con los posibles enemigos.
- 2. Un GameObject vacío llamado SpaceInvadersGame que contiene la clase singleton que gestiona el juego y la información mostrada por la interfaz.
- 3. El TextRecognition que sirve para cargar la detección de textos. A éste le hemos añadido un diccionario propio de palabras para poder leer texto en castellano. El diccionario contiene únicamente dos palabras, facultad e informática. Hemos configurado el TextRecognition de manera que sólo busque las palabras que están en su white list (lista blanca), que son las dos antes mencionadas, así las operaciones son más ligeras ya que no tiene que comprobar las miles de palabras.
- 4. Word representa a una palabra detectada por Vuforia. Se puede configurar para que represente cualquier palabra detectada o alguna en particular. Nosotros lo utilizamos para representar en particular la palabra INFORMÁTICA. Éste es el GameObject que sustituye al ImageTarget que utilizábamos en el pasado. Al detectar la palabra “INFORMÁTICA” en la cámara de RA, activa sus hijos y “avisa” al gestor del juego de que debe empezar a ejecutarse.
- 5. Un Enjambre, que contiene la lógica para crear varios Invasores y posicionarlos a cada uno en su sitio, así como para moverlos todos juntos.
- 6. Las copias de los invasores, las cuales disparan a veces a las defensas.
- 7. Las defensas, un objeto en tres dimensiones que representa a las defensas del jugador. Van cambiando de color, desde el verde al rojo según van recibiendo impactos de los invasores (o del propio jugador que apunta mal, para ser algo más realista).

4.2.2. Implementación

Pasamos a explicar qué clases componen el juego y para qué las utilizamos.

- 1. Defense.cs: Gestiona las defensas del usuario. Marca el color de inicio y el de final que debe tener la defensa para calcular los colores intermedios. Además gestiona las colisiones. Va añadido al GameObject de las defensas.

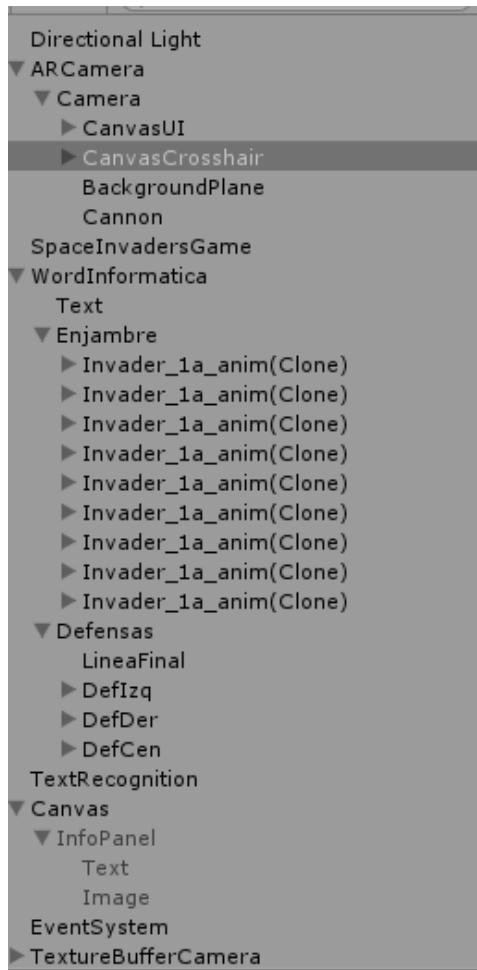


Figura 4.2: Diagrama de los GameObjects

2. Projectile.cs: Muy simple. Va asociada a los proyectiles y los destruye al pasar unos segundos en escena. Es para que los proyectiles que no impacten con nada, no se queden siempre en la escena. En la versión final, los proyectiles con este Script solo son lanzados por los enemigos.
3. Enjambre.cs: Se encarga de gestionar la inicialización del Enjambre y de sus invasores (colocándolos en la posición que les corresponda en función de cuántos sean y cuántas filas queremos que haya) y el movimiento del Enjambre (del que “cuelgan” los invasores), así como la escala de los invasores. Además contiene la información para saber si se han eliminado a todos los invasores o no. Va añadido a un GameObject vacío llamado Enjambre, el cual cuelga del detector de texto y del cual cuelgan los invasores.

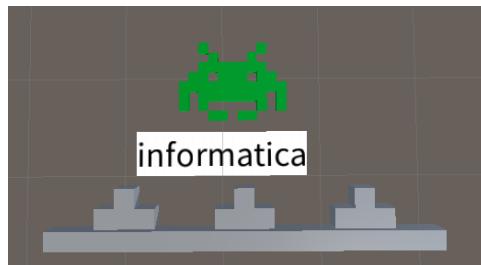


Figura 4.3: Captura durante el proceso de implementación

4. GameManager.cs: Es la clase que gestiona el juego en sí. Es un singleton y se le llama desde la mayoría de los otros scripts. Gestiona la interfaz de usuario, mostrando mensajes y los puntos cuando empieza el juego, además de cuando se puede disparar, etcétera. Es componente del GameManager, un GameObject vacío también.
5. Invader.cs: Lógica del invasor. Gestiona los disparos de los invasores, la muerte de éstos y el sonido que hacen al ser destruidos. Va añadido al GameObject Invader.
6. TextInformaticaTrackableEventHandler.cs: Implementación propia de la clase ITrackableEventHandler de Vuforia. Va asociada al Text de RA. Al “encontrarse” y si no está instanciado ya (es decir, que no se ha “encontrado” varias veces), le dice al GameManager que comience el juego, indicándole dónde están el Enjambre y las Defensas en relación al Text.
7. TextTimer.cs: De manera muy sencilla destruye el texto (de la interfaz gráfica) al que está asociado al pasar un tiempo dado una vez se ha habilitado. Lo utilizamos para mostrar los mensajes de texto de información previa, avisando al jugador cuando debe comenzar a disparar. Va asociado a GameObjects de texto.

4.2.3. Conclusiones

Al detectar el texto de FACULTAD DE INFORMÁTICA, el GameManager es avisado y comienza el juego. El GameManager instancia el Enjambre, que a su vez instancia todos los Invaders. Una vez hecho esto, se instancian las defensas desde el GameManager, se activa la información de la interfaz de usuario (puntuación, punto de mira . . .). A partir de este momento, el jugador puede disparar el láser.

Ahora, en cada actualización, el Enjambre mueve su posición, moviendo a todos los Invaders con él. Por su parte, cada Invader tiene un tanto porciento

de posibilidades de disparar. Si se da esa posibilidad, instanciará un proyectil en la dirección de las defensas.

Durante el desarrollo de este juego hemos encontrado unos cuantos es-
collos que superar, algunos que nos han llevado quebraderos de cabeza. Co-
menzamos desarrollando el videojuego utilizando un código QR como mar-
cador de la RA, pensando que después pasar a utilizar un texto no tendría
complicaciones. Una vez teníamos el juego desarrollado con un código QR
(ImageTarget), probamos a detectar texto propio, ya que Vuforia nos da
un diccionario con miles de palabras en inglés, pero nosotros no queríamos
detectar esas palabras, sino únicamente FACULTAD DE INFORMÁTICA.

Para esto, seguimos los tutoriales de Vuforia, y, tras resolver algunas du-
das, implementamos una escena sencilla en la que se mostraba una esfera
sobre el texto. Después, intentamos transferir lo desarrollado con el Image-
Target, al texto.

Entonces surgieron los problemas. El primero que vimos, era que las
proporciones de nuestros Invasores y las Defensas se quedaron muy pequeñas,
haciendo imposible jugar cómodamente. La mejor manera que conseguimos
para que se ajustaran los elementos del juego a un tamaño aceptable fue
mediante un Script de C# que aumentaba la escala local de los Invasores y
las Defensas (local scale).

El siguiente problema que encontramos fue al insertar nuestro Enjambre
de Invasores. Por un lado, una vez aparecían los enemigos, si movíamos
la cámara, éstos se quedaban en la misma posición respecto a la cámara, es
decir, si cuando salían por primera vez estaban en la parte superior izquierda
(por ejemplo) de la pantalla, y nos movíamos, seguían ahí, en vez de ajustar
su posición con respecto al texto detectado. Tuvimos que cambiar la forma en
la que los insertábamos en la escena, antes de manera dinámica y creando una
instancia con un Script, y ahora como hijos del GameObject que representa
al texto detectado. Éste tipo de problema (de la posición respecto a los
objetos de RA) nos volvería a salir más adelante, pero con los proyectiles
que lanzábamos para acabar con los enemigos.

En las primeras versiones del juego, lanzábamos un prisma (nuestro Pro-
yectil, similar al que lanzan los enemigos contra las defensas) contra los
enemigos. Según lanzábamos, el proyectil se iba dirigiendo en la dirección
que tenía la cámara respecto al ImageTarget al realizar el disparo. Al pasar
a utilizar el texto, esto cambió. Vimos que nuestro disparo se mantenía siem-
pre en el vector de dirección de la cámara, y cambiaba con éste. Es decir,
nuestro proyectil siempre estaba en el centro de la pantalla, con lo cual se
perdía toda la gracia al juego y su jugabilidad pasaba a ser bastante compli-
cada. Éste problema nos dejó bastante confusos, ya que con cambiar el objeto
de la RA (ImageTarget o TextRecognition) cambiaba el comportamiento del
proyectil.

Para resolver esto, cambiamos nuestro proyectil por un láser. Ahora al

tocar la pantalla no se lanza un proyectil, si no que se dispara un láser que destruirá los enemigos que estén en el vector de dirección de la cámara. Así hemos conseguido solventar este extraño comportamiento.

El juego en general ha quedado sencillo pero con todos los aspectos que debe tener un juego completo. Sonidos, animaciones, efectos visuales y jugabilidad aceptable. Creemos que sirve como una buena aproximación a otros juegos de este tipo, y que se podría ir escalando para desarrollar un juego más ambicioso.

La complejidad no es grande, aunque sí es recomendable que se tenga cierta habilidad para apuntar con el móvil.

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (?)

Capítulo 5

Arkanoid

...

...

...

RESUMEN: ...

5.1. Historia

...

5.2. Nuestra versión

...

5.3. Implementación

...

5.4. Conclusiones

...

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (?)

Y también ponemos el acrónimo **CVS!** para que no cruja.

Ten en cuenta que si no quieres acrónimos (o no quieres que te falle la compilación en “release” mientras no tengas ninguno) basta con que no definas la constante `\acronimosEnRelease` (en `config.tex`).

Capítulo 6

Water Pipes

...

...

RESUMEN: ...

6.1. Historia

...

6.2. Nuestra versión

...

6.3. Implementación

...

6.4. Conclusiones

...

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (?)

Y también ponemos el acrónimo **CVS!** para que no cruja.

Ten en cuenta que si no quieres acrónimos (o no quieres que te falle la compilación en “release” mientras no tengas ninguno) basta con que no definas la constante `\acronimosEnRelease` (en `config.tex`).

En el próximo capítulo...

...

Capítulo 7

Detalles de implementación

En esta sección se van a exponer detalles técnicos del desarrollo que sirva para completar todos aquellos puntos que no se tratan en los capítulos de los otros juegos, pero que han sido necesarios para el desarrollo del proyecto.

7.1. Realidad aumentada

Vamos a ver los diferentes aspectos y herramientas de la realidad aumentada que hemos utilizado.

7.1.1. Vuforia

Vuforia es el framework que hemos utilizado para hacer toda la parte de RA. Este tiene una magnífica integración con Unity 3D y facilita muchísimo su desarrollo.

Para generar RA, Vuforia superpone a la imagen tomada por la cámara de, en este caso, nuestro Smartphone, cualquier modelo en tres dimensiones que queramos sobre la posición de un Image Target (u otro marcador) que le hayamos indicado. De esta manera, tenemos un “fondo” con la imagen tomada por la cámara, con modelos en tres dimensiones “por encima”. Además, nos mantiene siempre los objetos de tres dimensiones en el mismo punto del espacio, por lo que si movemos nuestra cámara, cambiará la perspectiva desde donde vemos el objeto, pudiendo girar alrededor de éste. El comportamiento puede ser diferente, dependiendo de cómo lo hayamos configurado (podemos hacer que el objeto persista aun que perdamos de vista el detector).

Vuforia proporciona paquetes para trabajar directamente con el SDK de Android o el de iOS, así como para Unity3D. Utilizando Unity3D podemos exportarlo después a una aplicación de Android o iOS también, aunque no quedaría de una manera tan “pulida” como desarrollándola directamente con el SDK del sistema operativo deseado. Nosotros hemos decidido utilizar el paquete para Unity3D, ya que es un motor de videojuegos y los tres tení-

mos unos conocimientos básicos en desarrollo con Unity, además de que nos permite exportar después el proyecto al sistema operativo que quisiéramos..

7.1.2. Unity

Unity funciona con algo a lo que han llamado escenas, que son diferentes situaciones o niveles del juego. En toda escena hay una jerarquía de objetos que la componen, y de cada objeto pueden colgar otros objetos, además de que se pueden añadir (por medio de código programable) otros objetos a esa jerarquía de manera dinámica. Todos los objetos de Unity tienen una serie de componentes, el más básico sería el de su situación en las tres dimensiones (o dos), su escala y su rotación con respecto a los tres planos. Estos componentes permiten configurar los objetos de manera sencilla, encapsulando funcionalidades. Esta forma de “componer” los objetos no es casual: es la más utilizada en programación de videojuegos.

Además, Unity cuenta con una extensísima comunidad de desarrolladores, así como tutoriales, guías, dudas resueltas... solo con los tutoriales que proporciona la propia gente de Unity podemos hacer un sencillo juego casi de cada uno de los tipos más comunes de juegos.

Unity nos proporciona por defecto el cálculo de colisiones entre objetos, gravedad, eventos de teclado o ratón... en pocos minutos podemos hacer cosas sencillas pero que con otras herramientas, o programándolo directamente a mano con un lenguaje de programación cualquiera como podría ser Java o C++, nos llevarían bastante más tiempo.

Los scripts los podemos escribir en C, Boo o un lenguaje “parecido” a JavaScript. Nosotros hemos decidido utilizar C#, ya que era la opción que más nos convencía por varias razones:

- Es más eficiente. Los tres teníamos conocimientos previos de Java, y C# es muy similar a Java en cuanto a sintaxis.
- Es el más usado por la comunidad.

Todos los Scripts utilizados en Unity heredan de la clase MonoBehaviour, la cual permite a estos scripts integrarse con la ejecución interna de Unity. Toda clase que herede de MonoBehaviour tiene los métodos Start (), Awake (), Update (), FixedUpdate () y OnGUI (), entre otros.

Éstos se ejecutan en diferentes momentos del juego.

- Awake (): el primer método al que se llama, antes incluso de que el objeto asociado esté habilitado en la escena. Se utiliza para inicializaciones o referencias entre scripts.
- Start (): se ejecuta después de Awake (), justo antes del primer Update () y después de que se active el objeto.

- Update (): se ejecuta en cada frame. Esto hace que dependa del procesador y del equipo donde se ejecuta. Se usa para actualizaciones comunes como mover objetos no físicos, recoger entrada del usuario...
- FixedUpdate (): el intervalo entre una ejecución y otra es consistente y siempre el mismo. Se utiliza para actualizaciones como ajustar objetos físicos.
- OnGUI (): se utiliza para gestionar y renderizar eventos de la Interfaz Gráfica de Usuario (Graphic User Interface, GUI). Sólo es llamada si el objeto está habilitado.

7.1.3. Unity + Vuforia

Vuforia nos proporciona un paquete de extensión de Unity 3D el cual debemos importar para trabajar. Éste paquete contiene diferentes prefabs (objetos ya construidos) que nos harán la tarea muy sencilla.

Lo que debe tener toda aplicación de RA hecha con Vuforia y Unity 3D es una ARCamera (cámara de RA). A ésta hay que indicarle el product key que nos da Vuforia desde su portal para desarrolladores, además de ésto, se le indicará el paquete de targets propios . Es la unidad mínima de desarrollo de RA con Vuforia.

Una vez hecho esto, tendremos diferentes opciones para lanzar los objetos de RA, que deben colgar en la jerarquía de Unity de cualquiera de los siguientes prefabs:

- Frame Markers: Son marcadores muy sencillos que son proporcionados por la gente de Vuforia en su paquete. Se pueden utilizar para calibrar la cámara, pero no tienen una gran calidad a la hora de ser detectados. Son lo más sencillo para comenzar una aplicación de prueba.
- Image Targets: Imágenes propias del desarrollador. Funcionan como los Frame Markers, pero éstas deben ser importadas desde un paquete generado por el portal de desarrolladores de Vuforia, el cual nos indicará la calidad de esa imagen para ser detectada.
- Multi-Targets: Son varios ImageTargets que representan las diferentes caras de un prisma en tres dimensiones.
- Cylinder Targets: ImageTarget que envuelve un cilindro, para representar, por ejemplo, una botella u otro objeto similar.
- Text Recognition: Nos permite detectar textos, ya sean del diccionario proporcionado por Vuforia de palabras en inglés (más de 100.000 palabras diferentes) o de uno creado por nosotros mismos.



Figura 7.1: Ejemplo de aplicación con Smart Terrain

- Object Recognition: Sirve para configurar un objeto en tres dimensiones que no sea ninguno de los anteriores.
- Smart Terrain: Permite reconstruir el entorno del usuario de la aplicación en tres dimensiones.

Con cualquiera de estos objetos, la funcionalidad por defecto (que podemos modificar creando nuestras propias clases que hereden de las que nos da Vuforia) es que al detectarse (ya sea un ImageTarget, un Text Recognition, etcétera) se comenzarán a mostrar todos los objetos que cuelguen de él en la jerarquía de Unity.

7.2. Escenas intermedias

Entre cada escena; o juego del desarrollo, se ha establecido una intermedia donde un humanoide nos pone en situación antes de cada misión y nos sitúa acerca de cómo se ha llegado a dicha situación y a dónde debemos ir para resolverla.

7.2.1. SALSA with Random Eyes

Gracias al paquete de Unity de animación por voz, SALSA lip sync, el proceso de sincronizar el audio con la animación del discurso del modelo ha sido algo muy sencillo. Dicha animación consta de tres componentes:

- El modelo, que en este caso era un prefab que venía configurado por defecto para soportar el componente de audio y de enfoque aleatoria.
- Componente de animación de los músculos faciales, al cual se le asigna un conjunto de audios de tal forma que la cara del modelo se articula de forma sincronizada con el audio proporcionado. El componente viene configurado para que el audio que le proporcionamos al modelo pueda ser interpretado por este con más o menos énfasis o con diferentes estados de ánimo.
- Random eyes es un componente que, asignado al modelo, articula sus ojos de tal forma que definiendo unos puntos objetivo, alterna y gesti-

cula mirando a los diferentes objetivos a lo largo de la animación del objeto.

7.2.2. VozMe MP3 generator

Además del modelo utilizado, también hubo que crear los audios que narran el hilo argumental de nuestro juego. Para esta tarea, se generarán los audio con la herramienta vozme.com¹ que convierte a mp3, con una voz un “robótica”, un texto dado.

7.3. Persistencia de puntuaciones

Al finalizar el juego, el usuario puede almacenar su puntuación, y ver el ranking de estas. Este sistema se hizo para enlazar el juego con lo que es un servicio web, ya que nos parecía una práctica interesante el hacer uso de la parte cliente que nos ofrece Unity para comunicarnos con servicios web. Se han implementado ambas partes en este caso, tanto el lado cliente que consume los servicios en Unity, como el lado del servidor, en el que se ha implementado un sistema de gestión de usuarios con una sencilla api REST en PHP, haciendo uso del framework Symfony 2.

7.3.1. Symfony

Para el desarrollo con symfony de una API REST se tuvieron en cuenta los siguientes puntos:

- Base de datos: Se genera una base de datos SQL con una sola tabla, la de usuarios y los campos que deseábamos guardar: id, nombre y puntuación.
- Operaciones: Para la parte REST; los servicios, solo pueden realizar las operaciones de leer y crear nuevos usuarios. En el panel de administración, se pueden realizar las CRUD, crear un nuevo usuario, leer los usuarios y ver el usuario en detalle, actualizar y eliminar.
- REST: las operaciones de la API son, de tipo GET /api/users.[json | xml] para obtener un listado de los usuarios y sus puntuaciones en uno de los formatos especificados. Y para guardar la puntuación de un usuario, de tipo POST /api/users dónde se les manda, en el body de la petición, la puntuación del usuario y el nombre de este.
- Seguridad: De lado del servidor, Symfony nos provee de un sistema de seguridad basado en tres factores; la dirección a la que se accede, la

¹<http://vozme.com/index.php?lang=es>

autenticación para acceder a ella y una vez autenticado, el rol que tiene dicho usuario para poder acceder. En nuestro caso hay dos usuarios: “admin”, con el rol de administrador, que le permite entrar tanto en el panel de administración, como hacer uso de la API. Y “api_user”, un usuario cuyo rol solo le permite usar los servicios web. El tipo de autenticación es a través de Basic auth. Consiste en añadir un campo en el header con la clave “Authorization” y como valor, la palabra “Basic” concatenada y separados por un espacio con la combinación de “username:password” codificada en Base64.

La implementación del sistema de persistencia se compone de dos partes, la parte del cliente, dónde se consumen los servicios web; y la parte del servidor, donde se configura el sistema.

El lenguaje utilizado para implementar la parte del lado del Servidor es PHP. Sobre este lenguaje se utiliza un framework de desarrollo llamado Symfony, que facilita mucho la tarea de construir este tipo de sistemas. Symfony hace uso de trabajar basada en el Modelo Vista Controlador y el flujo de trabajo sobre este framework consiste en simplificarte, bajo esta arquitectura, un montón de procedimientos y líneas de código que se repiten constantemente en este tipo de software.

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (?)

Capítulo 8

Evaluación con usuarios

Una vez hemos tenido una versión terminada de todos los juegos y unidos ellos entre sí mediante una pequeña historia, damos por finalizada la primera versión estable de nuestro juego. Con esta versión, decidimos realizar pruebas con usuarios para mejorar los posibles errores o mejorar la usabilidad de los juegos de cara a una segunda iteración.

8.1. Plan de evaluación

8.1.1. Objetivos de la evaluación

Nuestros objetivos a la hora de realizar la evaluación con distintos usuarios son: mejorar la usabilidad de cara al usuario, detectar posibles bugs, comprobar si las mecánicas son comprensibles para los usuarios del museo y ver si estas son entretenidas y coherentes de cara a la RA.

Cuando se desarrolla un juego, el programador realiza distintas pruebas durante toda la implementación. Y una vez que ha terminado el juego, intenta jugar de varias formas posibles con la única intención de encontrar el mayor número de errores posibles para luego solventarlos. Pero esto no suele ser suficiente debido a que si realiza las pruebas la misma persona que lo ha implementado, pasará por alto muchas cosas. Esto es debido a que él sabe el funcionamiento del juego e inconscientemente jugará bien y le resultará muy difícil darse cuenta de cierto tipo de errores. Un buen ejemplo de esto es, que si el juego es intuitivo, es decir, si aunque no sepas como funciona, esté lo suficientemente bien explicado y diseñado para que el usuario no tenga problemas de ese tipo.

Otro motivo por el cual pensamos que la realización de pruebas a distintos usuarios es importante, es para tener la información suficiente, como para poder evaluar los errores en y clasificarlos por prioridad. Después de realizar las suficientes pruebas, se van a presentar varios errores, pero no tendrán la misma complejidad a la hora de solucionarlo y sobre todo no suponen lo

mismo para la jugabilidad de la aplicación.

Creemos, que la evaluación con usuarios es una fase fundamental en cualquier aplicación. Cuantas más pruebas se realicen, la aplicación será mejor y la experiencia del usuario con el juego será más satisfactoria, lo que seguramente hará que ese usuario quiera repetir el juego y se descargue tu aplicación

8.1.2. Métricas

Al ser una aplicación, decidimos utilizar el cuestionario SUS para realizar estas pruebas. Pero este cuestionario está en un principio dirigido a evaluar una aplicación web, por lo que tuvimos que modificar un poco las preguntas para adaptarlo a un juego. Además, aparte de este, mientras el usuario juega, se toma nota de todo aquello que se puede observar importante, ya sean bugs o posibles mejoras y aquello que el usuario menciona mientras está jugando: Cosas que no comprende, observaciones que aprecia acerca de la dificultad de los juegos o de sus mecánicas.

La siguiente tabla muestra las preguntas finales, con las respuestas necesarias para sacar el 100 % de la puntuación.

[c]@lll@ N	1	2
Preguntas	Me gustaría volver a	El juego tiene una
Puntuación	jugar	complejidad
	5	innecesaria
		1

Una de las ventajas de este cuestionario, es que las preguntas son sencillas y fáciles de entender. Pero también, es muy importante que sea muy sencillo de contestar, ya que el jugador, solo tendrá que evaluar cada pregunta en un intervalo del 1 al 5; siendo 1 la respuesta elegida si está muy en desacuerdo con lo que le preguntan, o 5 en el caso de que esté muy de acuerdo.

8.2. Descripción de la metodología del análisis de los datos

Dividimos en varias fases este proceso, con la intención de simplificar cada una de ellas y así tener muy claro lo que debíamos hacer en cada momento para que no se nos escapará ningún detalle, obteniendo así, no el mejor resultado en los test, sino lo más importante datos e información relevante para poder mejorar nuestra aplicación

8.2.1. Primera fase

Lo primero que hicimos, una vez adaptamos las preguntas del cuestionario a las de un juego, fue decidir sobre qué nos deberíamos fijar más detenida-



Figura 8.1: Usuario realizando un prueba del Space Invaders

mente mientras realizamos los test y así poder tomar notas sobre ello para más tarde poder añadirlo a la evaluación final, ya que cuanta más información se obtenga de los test a los usuarios, más mejoras podremos realizar en nuestro juego.

Cuando tuvimos claro todos estos puntos, empezamos a realizar test a la mayor cantidad de usuarios posibles, sin importarnos su edad, su habilidad a la hora de utilizar un smartphone o si habían probado previamente los juegos que hemos implementado.

8.2.2. Segunda fase

El siguiente paso es la evaluación directa con los usuarios. Nuestro procedimiento a seguir en esta fase, es la de que el usuario juegue e intentando no explicarle nada del juego previamente analizar mientras juega y decirle que comente en voz alta sus observaciones. Además, mientras el usuario está jugando, nosotros tomamos notas que nos sirven de ayuda. Y una vez el usuario ha finalizado, se le pasa el cuestionario para que responda a las preguntas y anote su edad.

8.2.3. Tercera fase

Analizamos los resultados, tanto los obtenidos a través del cuestionario SUS, como de las notas recogidas durante la prueba del juego.

8.3. Primera evaluación con usuarios

En esta primera evaluación, efectuamos un total de 14 pruebas a los usuarios, de las cuales, 7 se realizaron en la facultad y el resto desde casa.

Tras realizar la evaluación nos encontramos con que la mayoría de los usuarios tenían problemas similares:

- El space invaders, aunque era el más intuitivo, se acababa muy rápido y era demasiado fácil de jugar.
- En el Arkanoid, la bola iba demasiado rápida, el dedo que controlaba la barra tapaba la escena y no les permitía ver bien, el control era poco preciso y el fondo no permitía distinguir bien cuáles eran los objetivos que había que destruir.
- En el Water Pipes, la gente no comprendía la mecánica del juego al iniciar. Pulsaban sobre una pieza y no comprendían si tenían que intercambiar las piezas. Además, les costaba comprender cuál era el comienzo y cuál la meta.

8.3.1. Informe de resultados

Cuestionario SUS

El análisis de resultados de la primera evaluación con usuarios fue muy revelador con respecto a los controles del usuario con los distintos juegos. Tanto como los resultados del cuestionario SUS, como los comentarios de los usuarios unidos con nuestras notas tomadas mientras jugaban, nos revelaron que en los dos últimos minijuegos de la aplicación era necesario realizar alguna modificación para que el usuario pudiera manejar los controles del juego de manera más efectiva.

Es importante saber que el cuestionario SUS coloca el resultado obtenido en un intervalo de puntuación que determina la usabilidad de la aplicación y en nuestro caso de los mini juegos.

Este intervalo es el siguiente:

- Puntuación $>80.3 \rightarrow$ Buena usabilidad
- $80.3 \geq \text{Puntuación} > 68 \rightarrow$ Bien, pero se puede mejorar
- Puntuación $\leq 51 \rightarrow$ La usabilidad no es buena

A continuación mostramos el promedio de los resultados obtenidos de los cuestionarios realizados en este primer análisis.

Simplemente con estos datos, se puede ver a primera vista lo comentando anteriormente. El primer mini juego, según la escala SUS, no obtiene la puntuación necesaria para obtener una buena usabilidad, pero se acerca mucho, lo que nos indica que aunque existen ciertas cosas se pueden mejorar, el usuario piensa que el juego tiene lo necesario para ser jugado sin problemas y a demás que el juego le resulta entretenido.

En cambio, la puntuación de los otros dos juegos se encuentran en el caso opuesto. Aunque no llegan a obtener el 51. límite para que la usabilidad del juego no sea buena, se aproximan demasiado.

Información recogida de los usuarios

Los resultados que nos ha proporcionado el cuestionario SUS, nos resultan muy útiles para obtener una visión clara de la usabilidad de cada juego, y aunque si analizamos las respuestas más detenidamente, podríamos hacernos una idea de los fallos y los deseos de los usuarios. Nos es imprescindible también tomar nota, tanto de los comentarios de los usuarios antes, durante y después de haber probado el juego, como de las observaciones que nosotros mismos realizamos.

En la primera evaluación, esta información es muy relevante, debido también a que se repite las mismas opiniones o muy parecidas, en la mayoría de los usuarios evaluados.

En las siguientes líneas mostramos las opiniones más frecuentes y relevantes de los usuarios, por cada uno de los juegos en esta primera evaluación.

- Space Invaders
 - 1. El juego sale muy pequeño con respecto al Image Target.
 - 2. La dificultad del juego es demasiado sencilla.
- Arkanoid
 - 1. El control del juego es demasiado complicado. La mayoría de los usuarios, no han sido capaces de controlar la barra para mantener la pelota a salvo.
 - 2. La velocidad de la pelota es demasiado rápida, lo que hace que el juego sea demasiado difícil de completar con éxito.
 - 3. La escena es demasiado grande, lo que obliga al usuario a separarse mucho del QR.
- Water Pipes
 - 1. La mayoría de los usuarios no saben cómo controlar el juego.
 - 2. El tiempo de comienzo antes de que el agua empiece a pasar por las tuberías es demasiado corto, por lo que los usuarios pierden antes de poder comprender el funcionamiento del juego.
 - 3. Las casillas de Start y End, se confunden por lo que el usuario no sabe dónde tiene que empezar a colocar las tuberías antes de que el tiempo acabe.

8.3.2. Acciones

Las acciones a emprender atendiendo al análisis de los cuestionarios son las siguientes:

- Space Invaders
 - 1. Aumentar el tamaño de los invaders.
 - 2. Aumentar el daño de los enemigos sobre la plataforma, aumentar la velocidad de desplazamientos de estos y aumentar la separación.
- Arkanoid
 - 1. Informar del control en la pantalla de información del principio e intentar mejorar la precisión del control de la plataforma.
 - 2. Si se es capaz de mejorar en el apartado uno quizá el reducir la velocidad de la pelota facilita mucho el juego, pero si no, se reducirá la velocidad o se ampliará el tamaño de la plataforma.
 - 3. Disminuir el tamaño del modelo del televisor.
- Water Pipes
 - 1. Especificar mejor el control y el objetivo del juego en el panel de información inicial.
 - 2. Si se mejora en el apartado uno quizás el aumentar el tiempo en este caso facilita en exceso el juego, pero sino, se aumentará el tiempo inicial.
 - 3. Resaltar de forma más significativa la diferencia entre ambos extremos.

8.3.3. Conclusiones

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (?)

Y también ponemos el acrónimo **CVS!** para que no cruja.

Ten en cuenta que si no quieres acrónimos (o no quieras que te falle la compilación en “release” mientras no tengas ninguno) basta con que no definas la constante \acronimosEnRelease (en config.tex).

En el próximo capítulo...

...

Capítulo 9

Conclusiones y trabajo futuro

...

...

...

RESUMEN: ...

9.1. Conclusiones

...

9.2. Líneas futuras

...

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (?)

Y también ponemos el acrónimo **CVS!** para que no cruja.

Ten en cuenta que si no quieres acrónimos (o no quieres que te falle la compilación en “release” mientras no tengas ninguno) basta con que no definas la constante `\acronimosEnRelease` (en `config.tex`).

En el próximo capítulo...

...

Capítulo 10

Aportaciones individuales

Trabajar en equipo divide el trabajo y multiplica los resultados

Anónimo

Lista de aportaciones individuales de cada uno de los miembros del equipo.

10.1. Organización general del proyecto

En general nos hemos organizado de manera independiente. Cada uno de los miembros ha realizado uno de los minijuegos, aunque luego hemos desarrollado algunas funcionalidades juntos y otras, a parte del minijuego de cada uno, también de manera individual. Aun siendo cada uno de los minijuegos responsabilidad de uno de los miembros del equipo, nos hemos apoyado cuando teníamos problemas en el desarrollo individual de cada uno.

Para mejorar el trabajo en equipo hemos hecho uso del Sistema de control de versiones Git a través de la plataforma de GitHub¹. Se trabajaba en una rama de desarrollo sobre la que íbamos integrando aquellos cambios que generábamos sobre el proyecto y otra de producción, donde se registraban y etiquetaban, aquellas versiones finales del juego de cada fase del desarrollo.

10.2. Raúl Cobos

El videojuego que he desarrollado es el Arkanoid además de las siguientes aportaciones.

- Realización de tutoriales en Unity3D para comprender en profundidad cómo funcionan sus escenas y sus mecánicas.

¹Repositorio utilizado

- Autoaprendizaje e investigación de nuevas tecnologías para mi como era Vuforia.
- Realización de prototipos de Unity3D y Vuforia.
- Testeo con Álvar de diferentes maneras de interactuar con la RA en los primeros momentos.
- Comunicación y reuniones con nuestro tutor Guillermo a través de reuniones presenciales y correos electrónicos.
- Evaluación con usuarios e interpretación de éstas.
- Escritura de todos los apartados de la memoria (menos los de los minijuegos de mis compañeros) y revisión de los contenidos del a misma.
- Toma de decisiones con el resto de mis compañeros.
- Construcción del entorno de persistencia de puntuaciones.
- Adaptación del cuestionario SUS.
- Integración de la extensión SALSA With RandomEyes para la animación del personaje de las escenas intermedias.
- Adaptación responsiva de las fuentes de los textos del juego.

10.3. Álvar D. Soler

El minijuego que yo he desarrollado ha sido el Space Invader. Además de todo este minijuego, he llevado a cabo las siguientes tareas:

- Realización de tutoriales en Unity3D para comprender en profundidad cómo funcionan sus escenas y sus mecánicas.
- Autoaprendizaje e investigación de nuevas tecnologías para mi como era Vuforia.
- Realización de prototipos de Unity3D y Vuforia.
- Puesta en práctica de reconocimiento de texto propio en castellano.
- Testeo con Raúl de diferentes maneras de interactuar con la RA en los primeros momentos.
- Pruebas con Virtual Buttons para ver la viabilidad de utilizarlos en los minijuegos.
- Animaciones de los Invaders.

- Búsqueda de sonidos.
- Realización de fotografías en el museo para poder orientarnos cuando trabajamos en nuestras casas.
- Maquetación en LaTeX utilizando la plantilla de TeXiS.
- Comunicación y reuniones con nuestro tutor Guillermo a través de reuniones presenciales y correos electrónicos.
- Testeo para calibrar bien el tamaño de los Invaders y las Defensas con el cartel del exterior de la Facultad.
- Evaluación con usuarios e interpretación de éstas.
- Búsqueda de imágenes que fueran fáciles de reconocer por la cámara de Vuforia.
- Escritura de todos los apartados de la memoria (menos los de los mijuegos de mis compañeros) y revisión de los contenidos del a misma.
- Toma de decisiones con el resto de mis compañeros.
- Traducción al inglés de los apartados de la memoria que están traducidos.
- Testeo con smartphones Moto G 2013 y Orange Hi4G.
- Mantenimiento del repositorio con la memoria en LaTeX.

10.4. María Picado

En mi caso, mi trabajo ha estado dividido en varias etapas, en cada una de las cuales me he dedicado a diferentes tareas.

- La primera de ellas, fue junto a mis compañeros la realización de tutoriales en Unity3D para comprender y afianzar los conocimientos sobre el funcionamiento de esta herramienta.
- Como ya dijimos anteriormente, aunque si conocíamos Unity, Vuforia era completamente desconocida para nosotros, con lo que mi siguiente tarea fue la de investigación y autoaprendizaje para conocer el funcionamiento de Vuforia. Y más adelante comencé con la creación de pequeños prototipos en Unity3d y Vuforia.
- Durante todo el proceso de creación del proyecto, hemos mantenido los tres reuniones periódicas con nuestro director del TFG para ir mostrándole los avances y ponernos nuevos objetivos de cara a la siguiente reunión.

- Una vez adquirimos los suficientes conocimientos para poder desenvolvernos tanto con Unity como con Vuforia, decidimos reunirnos los tres para diseñar el videojuego y decidir qué minijuegos implementaremos.
- En esa reunión, una de las decisiones que tomamos fue la de que juegos íbamos a implementar cada uno, y a partir de ese momento me centré en la realización del minijuego Water Pipes. Lo primero que hice fue documentarme del juego original para ver cómo sería la mejor manera de adaptarlo a la RA.
- Una de las cosas más importantes para que el juego se pudiera adaptar a la RA, era obtener la forma de que el usuario pudiera manipular las tuberías. Al ser una aplicación móvil, lo primero que intenté fue el ?DRAG AND DROP?, para que el jugador pulsara una tubería y la arrastrarse hasta donde quisiera cambiarla y al soltarla se cambiará automáticamente una tubería por la otra. Pero esta opción me dio bastantes problemas y entonces decidí que la forma en que se fueran colocando las tuberías fuera el intercambio entre ellas, pulsando sobre una e intercambiando por la siguiente en ser pulsada.
- Una vez implementada la funcionalidad para colocar las tuberías, comencé a implementar la parte más importante del juego; el flujo del agua. Como existen 6 opciones de tuberías distintas y cada una de ellas tiene otras dos direcciones posibles por las que puede circular el agua. Tenía que buscar una forma de encapsular esa información para que lo único que nos preocupase fuera la salida de la tubería actual y la entrada de la siguiente. Por eso decidí usar el patrón de diseño Command.
- Al acabar la implementación del juego, el siguiente paso fue realizar el mayor número de test posibles a los usuarios. Los test los realizamos del juego completo, pero pedimos a los usuarios que puntuaron los juegos de manera independiente. Por eso, un vez que obtuvimos los primeros resultados, me dispuse a modificar el juego para implementar las mejoras que me aconsejaron los usuarios.
- Durante todo el proyecto he colaborado en el desarrollo de esta memoria.

Lista de acrónimos

*-¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.*

*Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

*-Buena está – dijo Sancho –; firmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.*

*Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

