
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

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 GO, 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

Lore ipsum en inglés.

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

Índice

Agradecimientos	VII
Resumen	IX
1. Introducción a Santesmases GO	1
1.1. Introducción	1
1.2. Museo de informática García-Santesmases	1
1.3. Objetivos y motivación	1
1.4. Antecedentes	2
Notas bibliográficas	2
2. Estado del Arte	3
2.1. La realidad aumentada	3
2.2. Alcance	4
2.2.1. Información interactiva	4
2.2.2. Entretenimiento	4
2.2.3. Ciencia y desarrollo	4
2.2.4. Otros	5
2.3. Realidad aumentada en videojuegos	5
2.3.1. Ejemplos de Realidad Aumentada en videojuegos . . .	5
2.4. Realidad aumentada en museos	7
2.4.1. El museo García-Santesmases	8
2.4.2. Otros museos	9
2.5. Herramientas de desarrollo	10
2.5.1. Vuforia	11
2.5.2. Cómo generar realidad aumentada	11
2.5.3. Plataformas de desarrollo	12
2.5.4. Unity3D	12
2.5.5. C#	13
2.5.6. Unity3D + Vuforia	13
Notas bibliográficas	14

3. Diseño del videojuego	15
3.1. Introducción	15
3.2. Hilo argumental	15
3.3. Conclusiones	15
Notas bibliográficas	15
En el próximo capítulo	16
4. Space Invaders	17
4.1. Historia	17
4.2. Nuestra versión	17
4.3. Implementación	18
4.3.1. Diseño	19
4.3.2. Desarrollo	20
4.3.3. Conclusiones	21
Notas bibliográficas	21
5. Arkanoid	23
5.1. Historia	23
5.2. Nuestra versión	23
5.3. Implementación	23
5.4. Conclusiones	23
Notas bibliográficas	23
6. Water Pipes	25
6.1. Historia	25
6.2. Nuestra versión	25
6.3. Implementación	25
6.4. Conclusiones	25
Notas bibliográficas	25
En el próximo capítulo	26
7. Evaluación con usuarios	27
7.1. Plan de evaluación	27
7.2. Descripción de la metodología del análisis de los datos	27
7.3. Primera evaluación con usuarios	27
Notas bibliográficas	27
En el próximo capítulo	28
8. Conclusiones y trabajo futuro	29
8.1. Conclusiones	29
8.2. Líneas futuras	29
Notas bibliográficas	29

En el próximo capítulo	29
9. Aportaciones individuales	31
9.1. Organización general del proyecto	31
9.2. Raúl Cobos	31
9.3. Álvar D. Soler	32
9.4. María Picado	33
Lista de acrónimos	35

Índice de figuras

1.1. Fotografía del museo	2
2.1. AR Cards para Nintendo3DS	6
2.2. Captura del juego Pokémon Go	6
2.3. Captura del juego Ingress	7
2.4. Captura del juego Paintball	7
2.5. Captura del juego Father.io	8
2.6. Captura del juego Night Terrors	8
2.7. Retrato de José García Santesmases por Eulogia Merle	9
2.8. Aplicación del Museo de Mataró	10
2.9. StreetMuseum	11
2.10. Crononautas	11
2.11. Captura de RACMA que muestra las zonas de influencia de las sociedades precolombinas.	12
4.1. Space Invaders original	18

Índice de Tablas

Capítulo 1

Introducción a Santesmases GO

1.1. Introducción

El proyecto que hemos desarrollado tiene como finalidad atraer al público al museo García Santesmases con una característica nueva y atractiva. Para ésto, 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 puede aportar a un museo. Todo esto dentro de una aplicación para móviles Android.

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.

1.2. Museo de informática García-Santesmases

El museo de informática García-Santesmases, se encuentra en los pasillos de las plantas 3º y 4º de la facultad de informática de la universidad Complutense de Madrid. Este museo hace un recorrido por las diferentes máquinas creadas por la Universidad Complutense de Madrid, así como de computadoras comerciales y equipos donados a la universidad.

1.3. Objetivos y motivación

Nuestro objetivo ha sido el desarrollo de una aplicación que mejorará la experiencia del usuario en un museo, en este caso, el museo de la Facultad de



Figura 1.1: Fotografía del museo

Informática García-Santesmases. Pero no solo esto, sino que nuestro objetivo era hacerlo a través de los videojuegos y utilizando la RA.

Esto lo logramos mediante una “gymkana”, guiando al visitante a que recorra el museo en busca de misiones que tendrá que ir completando mini-juegos 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. Por lo que enseguida comenzamos a investigar sobre aplicaciones creadas anteriormente y descubrimos que los museos se están haciendo cada vez más eco de los beneficios de aplicaciones como ésta para atraer al público. Esto nos motivó más, ya que nos ponía delante una opción real de desarrollo.

1.4. 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

...

...

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.

Con el continuo avance de los dispositivos móviles y la potencia, a un costo aceptable para la mayoría, de la que constan ahora mismo estos, permiten que cualquier usuario pueda hacer uso de la RA, ya que sus Smartphones cumplen todos los requisitos para ello.

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.

2.2.2. Entretenimiento

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.

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 y la construcción entre otras.

2.2.4. Otros

Además de los ya mencionados, existen todavía muchísimas áreas en las que tienen cabida en sus tecnologías la RA además de muchos otros usos que o bien están por descubrir o no se consta todavía de la tecnología suficiente como para integrarlo.

Algo que todavía está en una fase temprana de desarrollo pero que tiene un futuro prometedor es el uso de la RA para generación de terreno de forma que se pueda reconocer el mismo y en función de su forma generar un medio en relación un entorno que todavía no estaba registrado. Esta tecnología dota de un sinfín de posibilidades añadidas a las ya existentes para el uso de la RA en muchos otros campos.

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 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.



Figura 2.1: AR Cards para Nintendo3DS



Figura 2.2: Captura del juego Pokémon Go

- 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 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.



Figura 2.3: Captura del juego Ingress



Figura 2.4: Captura del juego Paintball

- 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.

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.



Figura 2.5: Captura del juego Father.io

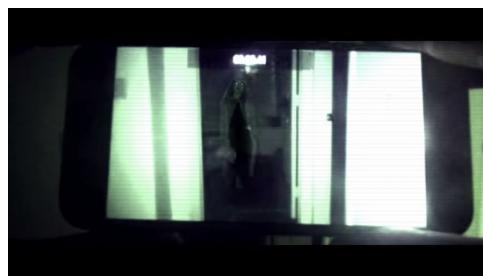


Figura 2.6: Captura del juego Night Terrors

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. El museo García-Santosmases

Inaugurado en noviembre del 2003, el museo debe su nombre al físico, profesor y precursor de la informática española José García Santosmases, 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

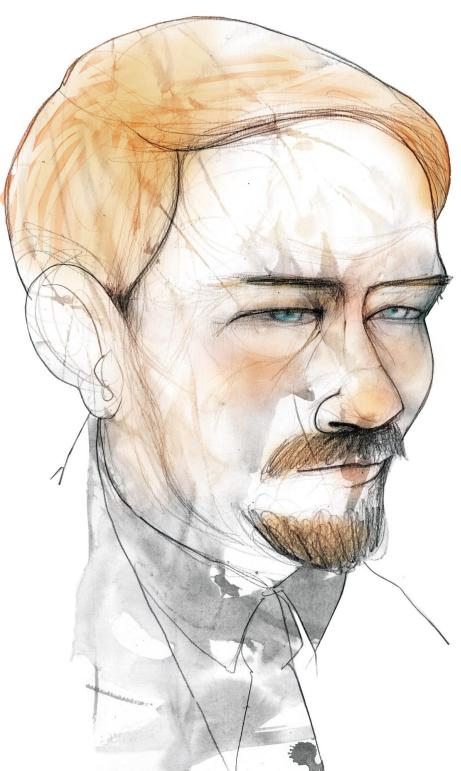


Figura 2.7: Retrato de José García Santesmases por Eulogia Merle

electrónico”, diseñado por García Santesmases y es el primer computador desarrollado en España.

2.4.2. Otros museos

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

- 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.

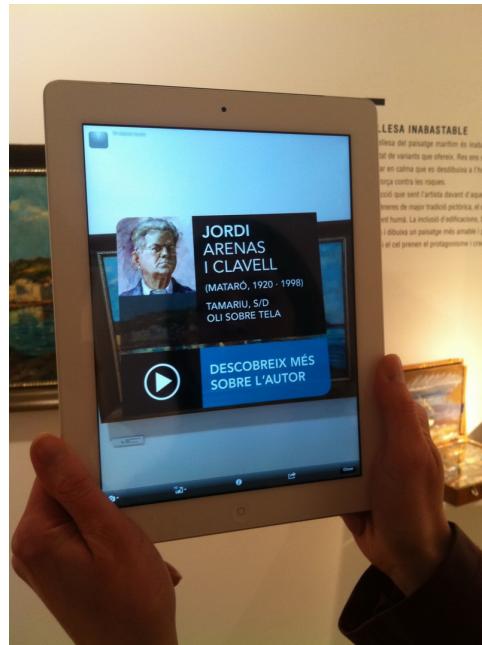


Figura 2.8: Aplicación del Museo de Mataró

- 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 hemos utilizado para la realización de este proyecto.



Figura 2.9: StreetMuseum



Figura 2.10: Crononautas

2.5.1. Vuforia

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...).

2.5.2. Cómo generar realidad aumentada

Básicamente, 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).



Figura 2.11: Captura de RACMA que muestra las zonas de influencia de las sociedades precolombinas.

2.5.3. Plataformas de desarrollo

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 porque los tres teníamos unos conocimientos básicos en desarrollo con Unity, además de que nos permite exportar después el proyecto al sistema operativo que quisieramos.

2.5.4. Unity3D

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.

2.5.5. C#

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:

- Hemos leído que 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.

1. 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.
2. Start (): se ejecuta después de Awake (), justo antes del primer Update () y después de que se active el objeto.
3. 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...
4. FixedUpdate (): el intervalo entre una ejecución y otra es consistente y siempre el mismo. Se utiliza para actualizaciones como ajustar objetos físicos.
5. 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.

2.5.6. Unity3D + 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 (lo explicaremos más adelante en profundidad). 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.
- 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.

Notas bibliográficas

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

Capítulo 3

Diseño del videojuego

...

...

RESUMEN: ...

3.1. Introducción

...

3.2. Hilo argumental

...

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 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 4

Space Invaders

4.1. Historia

Quizá uno de los juegos arcade clásicos más conocidos. La primera versión salió al mercado en 1978, hace casi cuarenta años. Uno de los precursores del género shoot em 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.

Éste juego 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, y cambiar el sistema. 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 realizar el disparo. Hemos pasado de manejar la nave defensora en tercera persona, a hacerlo 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

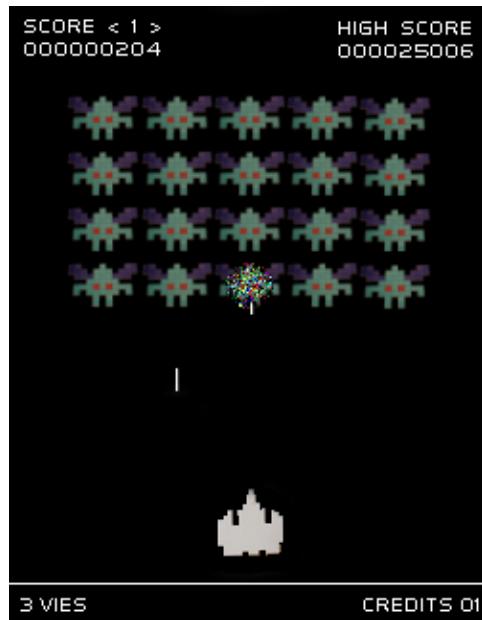


Figura 4.1: Space Invaders original

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.

4.3. Implementación

Durante el desarrollo de este juego hemos encontrado unos cuantos escollos que superar, algunos que nos han llevado quebraderos de cabeza. Comenzamos desarrollando el videojuego utilizando un código QR como marcador 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, si no únicamente FACULTAD DE INFORMÁTICA.

Para esto, seguimos los tutoriales de Vuforia, y, tras resolver algunas dudas, 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 Proyectil) 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 siempre 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 complicada. 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.

4.3.1. 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).

4.3.2. Desarrollo

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.
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.

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.
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.
5. Invader.cs: Lógica del invasor. Gestiona los disparos de los invasores, la muerte de éstos y el sonido que hacen al ser destruidos.
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.

4.3.3. Conclusiones

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

Evaluación con usuarios

...

RESUMEN: ...

7.1. Plan de evaluación

...

7.2. Descripción de la metodología del análisis de los datos

...

7.3. Primera evaluación con usuarios

...

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 8

Conclusiones y trabajo futuro

...

RESUMEN: ...

8.1. Conclusiones

...

8.2. 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 9

Aportaciones individuales

Trabajar en equipo divide el trabajo y multiplica los resultados

Anónimo

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

9.1. 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.

9.2. Raúl Cobos

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

¹Repositorio utilizado

- Realización de tutoriales en Unity3D para comprender en profundidad cómo funcionan sus escenas y sus mecánicas.
- Autoaprendizaje e investigación de nuevas tecnologías para mí 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.

9.3. 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 mí 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 minijuegos 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.

9.4. 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 las 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*

