
Realidad aumentada para el Museo García-Santesmases



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 T_EX_S v.1.0+.

Este documento está preparado para ser imprimido a doble cara.

Realidad aumentada para el Museo García-Santesmases

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	1
Notas bibliográficas	2
En el próximo capítulo	2
2. Estado del Arte	3
2.1. La realidad aumentada	3
2.2. Alcance	3
2.3. Herramientas de desarrollo	3
Notas bibliográficas	3
En el próximo capítulo	4
3. Realidad aumentada en museos	5
3.1. Introducción	5
3.2. El museo García-Santesmases	5
3.3. Otros museos	5
Notas bibliográficas	5
En el próximo capítulo	6
4. Diseño del videojuego	7
4.1. Introducción	7
4.2. Plan de trabajo	7
4.3. Otros aspectos	7
4.4. Conclusiones	7
Notas bibliográficas	7

En el próximo capítulo	8
5. Space Invaders	9
5.1. Historia	9
5.2. Nuestra versión	9
5.3. Implementación	10
5.3.1. Diseño	11
5.3.2. Desarrollo	12
5.3.3. Conclusiones	13
Notas bibliográficas	13
6. Arkanoid	15
6.1. Historia	15
6.2. Nuestra versión	15
6.3. Implementación	15
6.4. Conclusiones	15
Notas bibliográficas	15
En el próximo capítulo	16
7. Water Pipes	17
7.1. Historia	17
7.2. Nuestra versión	17
7.3. Implementación	17
7.4. Conclusiones	17
Notas bibliográficas	17
En el próximo capítulo	18
8. Evaluación con usuarios	19
8.1. Plan de evaluación	19
8.2. Descripción de la metodología del análisis de los datos	19
8.3. Primera evaluación con usuarios	19
Notas bibliográficas	19
En el próximo capítulo	20
9. Conclusiones y trabajo futuro	21
9.1. Conclusiones	21
9.2. Líneas futuras	21
Notas bibliográficas	21
En el próximo capítulo	21
10. Aportaciones individuales	23
10.1. Organización general del proyecto	23

10.2. Raúl Cobos	23
10.3. Álvar D. Soler	23
10.4. María Picado	23
Notas bibliográficas	23
En el próximo capítulo	24
I Apéndices	25
A. Así se hizo...	27
A.1. Introducción	27
Lista de acrónimos	29

Índice de figuras

5.1. Space Invaders original	10
--	----

Índice de Tablas

Capítulo 1

Introducción a Santesmases GO

...

...

RESUMEN: ...

1.1. Introducción

...

1.2. Museo de informática García-Santesmases

...

1.3. Objetivos y motivación

...

1.4. Antecedentes

...

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 defines la constante `\acronimosEnRelease` (en `config.tex`).

En el próximo capítulo...

...

Capítulo 2

Estado del Arte

...

...

RESUMEN: ...

2.1. La realidad aumentada

...

2.2. Alcance

...

2.3. Herramientas de desarrollo

...

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 3

Realidad aumentada en museos

...

...

RESUMEN: ...

3.1. Introducción

...

3.2. El museo García-Santesmases

...

3.3. Otros museos

...

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

Diseño del videojuego

...

RESUMEN: ...

4.1. Introducción

...

4.2. Plan de trabajo

...

4.3. Otros aspectos

...

4.4. 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 5

Space Invaders

5.1. 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 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.

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.

5.2. 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

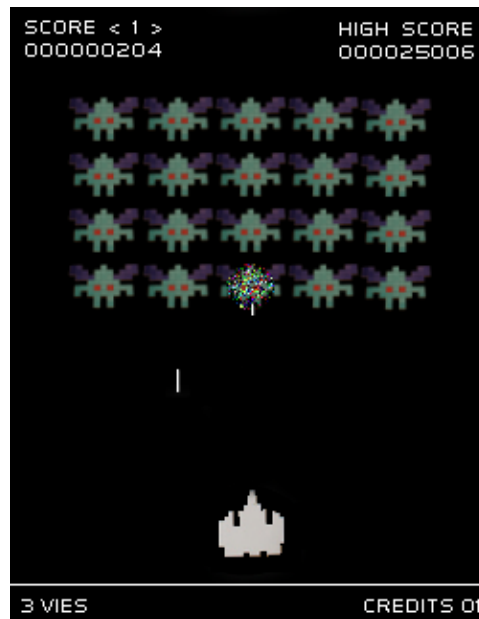


Figura 5.1: Space Invaders original

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.

5.3. 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 ImageTarget, 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.

5.3.1. 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).

5.3.2. 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.

5.3.3. 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... (?)

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

Arkanoid

...

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

Water Pipes

...

RESUMEN: ...

7.1. Historia

...

7.2. Nuestra versión

...

7.3. Implementación

...

7.4. 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 8

Evaluación con usuarios

...

RESUMEN: ...

8.1. Plan de evaluación

...

8.2. Descripción de la metodología del análisis de los datos

...

8.3. 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 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

...

RESUMEN: ...

10.1. Organización general del proyecto

...

10.2. Raúl Cobos

...

10.3. Álvaro D. Soler

...

10.4. María Picado

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...

...

Parte I

Apéndices

Apéndice A

Así se hizo...

...

...

RESUMEN: ...

A.1. Introducción

...

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 –; fírmela 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*

