



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

Tecnologías de realidad virtual para el tratamiento de ambliopía en adultos

*Virtual reality technologies the treatment of amblyopia for
adults*

Néstor Viña León

La Laguna, 8 de septiembre de 2015

D. **Isabel Sánchez Berriel**, con N.I.F. 12.345.678-X profesora Colaboradora adscrita al Departamento de Ingeniería Informática y Sistemas de la Universidad de La Laguna, como tutora

D. **Rafael Melián Villalobos**, con N.I.F. 12.345.678-X como cotutor.

C E R T I F I C A (N)

Que la presente memoria titulada:

“Tecnologías de realidad virtual para el tratamiento de ambliopía en adultos”

ha sido realizada bajo su dirección por D. **Néstor Viña León**,
con N.I.F. 54.114.111-X.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 6 de septiembre de 2015

Para mi abuela Matilde, porque nosotros siempre
fuimos su proyecto.

Licencia



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

Resumen

Con este trabajo se ha tratado de explorar las posibilidades de la realidad virtual como tratamiento para la ambliopía en adultos, algo comúnmente imposible hasta ahora.

Hasta hace poco tiempo se creía que la ambliopía (también llamada ojo vago) era incurable para las personas adultas que la padecían. Algunos estudios tratan de exponer lo contrario, han obtenido resultados usando ejercicios parecidos a videojuegos. Pero aún todos estos estudios no han desembocado en un tratamiento. Elegimos como hardware para el tratamiento cascos de realidad virtual porque creemos firmemente que esto supone una ventaja para el propósito planteado.

De ahí nace este trabajo, de la necesidad de explorar nuevas maneras de crear tratamientos para la ambliopía y que sirva como referencia para la creación de futuros tratamientos.

Palabras clave: Realidad virtual, ambliopía, tratamiento

Abstract

This work has tried to explore the possibilities of virtual reality as a treatment for amblyopia in adults, something usually impossible until now.

Until recently it was believed that amblyopia (also called lazy eye) was incurable for adults that suffered. Some studies try to explain otherwise have obtained similar results using video games exercises. But even these studies have not led to a treatment. We choose as hardware for the treatment virtual reality helmets because we firmly believe that this is an advantage for treatment.

Thus it was born this work, the need to explore new ways to create treatments for amblyopia and serve as a reference for the creation of future treatments.

Keywords: Virtual Reality, Amblyopia, Treatment

Índice general

Capítulo 1 Introducción.....	11
1.1 Antecedentes.....	11
Capítulo 2 Elección de herramientas	20
2.2 Mono Develop.....	21
2.3 Cardboard SDK for Unity.....	21
2.4 Plugin para reproducir video en Android.....	22
2.5 Lenguaje de programación C#.....	22
2.6 Shaders GLSL.....	22
Capítulo 3 Desarrollo Técnico.....	25
3.1.1 Desarrollo del visor de memes.....	26
3.1.2 Desarrollo del efecto para la ambliopía.....	30
3.1.3 Desarrollo del visor de vídeos.....	32
3.1.4 Desarrollo del Breakout.....	33
3.1.5 Desarrollo del Menú.....	35
Capítulo 4 Conclusiones y líneas futuras.....	37
Capítulo 6 Presupuesto	39

Índice de Figuras

Figura 1.1: Ejemplo “Captura de pantalla de una de las aplicaciones que tratan de mejorar la ambliopía con el sistema anaglífico”.....	12
Figura 1.2: “Google Cardboard, con un Nexus 5 abierto”.....	14
Figura 1.3: Ejemplo “Gear VR, el producto de Samsung para la realidad virtual”.....	15
Figura 1.4: Ejemplo “Estimaciones de Business Insider acerca del futuro crecimiento de la realidad virtual”.....	16
Figura 1.5: Ejemplo “Imagen en side-by-side estereoscópica”.....	17
Figura 2.1: Ejemplo “Entorno del editor de Unity3D”.....	20
Figura 2.2: Ejemplo “Logo de Mono Develop”.....	21
Figura 2.3: Ejemplo “Shaders funcionando en tiempo real con Directx”.....	23
Figura 2.4: Ejemplo “Demo de OpenGL ES usando shaders en dispositivos móviles”.....	24
Figura 2.5: Ejemplo “Shader de terciopelo funcionando con Directx 11”....	24
Figura 3.1: Ejemplo “Ejemplo de un meme cualquiera de internet”.....	26
Figura 3.2: Ejemplo “Esquema de la relación y funcionamiento entre las webs de memes y el módulo de gestión de memes de la aplicación”.....	29
Figura 3.3: Ejemplo “Primera propuesta del prototipo”.....	30
Figura 3.4: Ejemplo “Segunda propuesta”.....	31
Figura 3.5: Ejemplo “Reproductor de memes funcionando con el efecto visual aplicado”.....	32
Figura 3.6: Ejemplo “Reproductor de vídeo funcionando con el efecto visual aplicado”.....	33
Figura 3.7: Ejemplo “Captura de pantalla en el nivel 1 del Breakout”.....	34
Figura 3.8: Ejemplo “Captura de pantalla en el nivel 3 del Breakout”.....	35
Figura 3.9: Ejemplo “Panel de configuración que permite seleccionar el ojo	

afectado”.....36

Figura 3.10: Ejemplo “Panel para reproducir un vídeo”.....36

Índice de tablas

Tabla 6.1: Presupuesto.....	39
-----------------------------	----

Capítulo 1

Introducción

Es apropiado comenzar describiendo a qué está orientado el tratamiento, qué es la ambliopía: Hasta hace poco tiempo se creía que la ambliopía (también llamada ojo vago) era incurable para las personas adultas que la padecían.

“La ambliopía (del griego ἀμβλύς [amblýs], ‘débil’, y ὄψ [ops], ‘ojo’) es una disminución de la agudeza visual sin que exista ninguna lesión orgánica que la justifique.¹ Puede existir algún defecto en el ojo (por ejemplo, una miopía), pero éste no justifica la pérdida de visión. Generalmente la afectación es unilateral y se produce como consecuencia de falta de estimulación visual adecuada durante el período crítico de desarrollo visual, lo que afecta a los mecanismos neuronales encargados de la visión. Está presente en un 4 por ciento de la población. “ - Wikipedia [consulta 19 de septiembre de 2014]

A partir de los 8 años de edad se consideraba que era irreversible pero algunos experimentos han demostrado que con estímulos visuales se pueden conseguir resultados con mejoría. A pesar de que estos experimentos han sido bastante exitosos, esta idea aún no se ha concebido más allá del experimento.

Este proyecto trata de crear una aplicación que actúe como tratamiento del ojo ambliope y que no solo funcione en niños sino también en adultos. Para ello se usarán cascos de realidad virtual y smartphones.

1.1 Antecedentes

Existen varios experimentos que afirman que hay una mejora en el ojo vago al someterlo bajo un entrenamiento con ejercicios visuales [1]. En los ejercicios se usaban videojuegos convencionales durante un número determinado de horas semanales y ocluyendo con un parche uno de los ojos de los sujetos de prueba. Después de varios meses presentaban síntomas de mejoría.

También se pueden encontrar referentes de aplicación que han sido un

intento de conseguir lo que persigue este proyecto pero de una forma menos efectiva. Se trata de aplicaciones que manipulan la información que percibe cada ojo para así desarrollar un ejercicio concreto. De esta forma puede eliminar información del ojo sano y mostrarla en el ojo afectado y así forzarlo a tratar de percibir la imagen binocular. El problema de esta aplicación es que para enviar la información a cada ojo utiliza el sistema anaglifo con las gafas rojo/cyan y extrae de una imagen la información para los dos ojos. Ésto es una barrera que impide realizar todos los ejercicios, solo se puede eliminar información de uno de los dos ojos. Con un casco de realidad virtual podemos controlar totalmente que imagen se visualiza con cada ojo y realizar actividades más complejas.

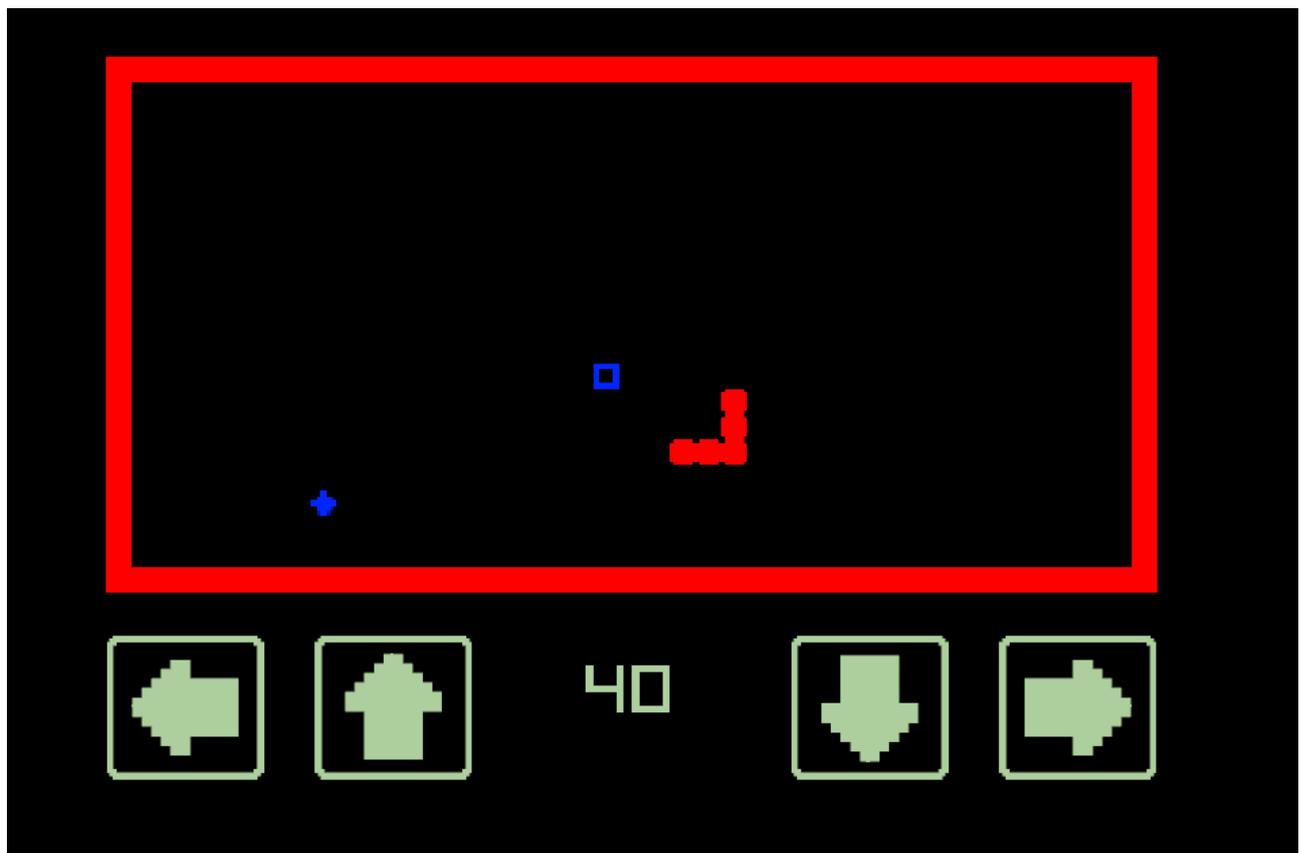


Figura 1.1: Ejemplo "Captura de pantalla de una de las aplicaciones que tratan de mejorar la ambliopía con el sistema anaglífico"

En la imagen superior se muestra una captura de pantalla de una de las aplicaciones mencionadas previamente. Hay dos objetivos azules y la serpiente en rojo. A través de la lente roja solo veremos los objetivos y a través de la lente cyan solo veremos la serpiente, lo que obliga al usuario a usar los dos ojos para poder continuar jugando. Cuando lo pruebas percibes anomalías en los colores y puede resultar incómodo. Se producen efectos como el ghosting, que sucede porque al percibir la información de una misma imagen a veces se percibe un color que está destinado a un ojo en el otro. Tampoco se percibirán nunca los colores como exactamente deberían ser

(<http://nzphoto.tripod.com/sterea/anaglyphs.htm>).

1.2 Definición de de objetivos

Los objetivos planteados son:

- Desarrollar una aplicación con las siguientes características:
 - Compatible con cascos de realidad virtual
 - Eficiente para que pueda funcionar en la máxima cantidad de smartphones
 - Eficaz como tratamiento de la ambliopía
 - Que no aburra al usarla, y permita así su uso prolongado
- Investigar en colaboración con el co-tutor oftalmólogo:
 - Los ejercicios visuales que permiten la mejoría de la ambliopía
 - Cómo medir la efectividad del tratamiento
 - Cómo crear el tratamiento evitando agravar otros defectos de la vista
- Aprender
 - Cómo desarrollar shaders GLSL
 - El uso del software Unity3D
 - Trabajar con otras materias como la oftalmología
 - C#
- Ayudar a mucha gente

1.3 Introducción: VR para smartphones

Para entender cuál es el alcance se debe entender como funciona la realidad virtual y que dificultades técnicas nos presenta.

La realidad virtual para smartphone es una tecnología de reciente creación que permite al usuario tener la sensación de estar en un lugar ficticio o vivir experiencias ficticias.

El funcionamiento de la mayoría de cascos que utilizan esta tecnología es muy sencillo. Existen una carcasa o casco con dos lentes y una estructura para sujetar el smartphone. En el smartphone se proyecta una imagen que las

lentes enfocan a esa distancia respecto al ojo. Cada lente enfoca la mitad de la pantalla del smartphone, que está procesando la imagen a mostrar. A través del acelerómetro, giroscopio y brújula se puede determinar la dirección a la que está orientado el smartphone y así replicar los movimientos de la cabeza en la cámara virtual que está renderizando lo que se ve en la pantalla. Con esto conseguimos que movimientos de nuestra cabeza tengan efecto en el entorno virtual como si fuera real.



Figura 1.2: “Google Cardboard, con un Nexus 5 abierto”

En versiones más avanzadas nos podemos encontrar con nuevas características que aumentan la calidad de la experiencia. En cuanto a la óptica algunos cascos vienen con unas lentes que permiten un campo de visión superior y por lo tanto aumentan la capacidad inmersiva. También vienen con mecanismos para regular el enfoque y la distancia interpupilar. En algunos se han desarrollado botones e incluso paneles táctiles para mejorar el input como en el caso del Samsung Gear VR.



Figura 1.3: Ejemplo “Gear VR, el producto de Samsung para la realidad virtual”

Entre una de las características que más interesan al proyecto está la capacidad de controlar la información que se envía a cada ojo.

También es interesante su reducido coste y amplia gama de calidades, lo que permite que las aplicaciones desarrolladas para VR sean más accesibles. Aunque esto es una situación pasajera, todo apunta a que la realidad virtual se va a generalizar tanto como lo han hecho los smartphones, y ahora solo estamos en el comienzo. Business Insider apunta que el crecimiento de venta de cascos será muy elevado en los próximos años lo que hará más accesible a la aplicación.

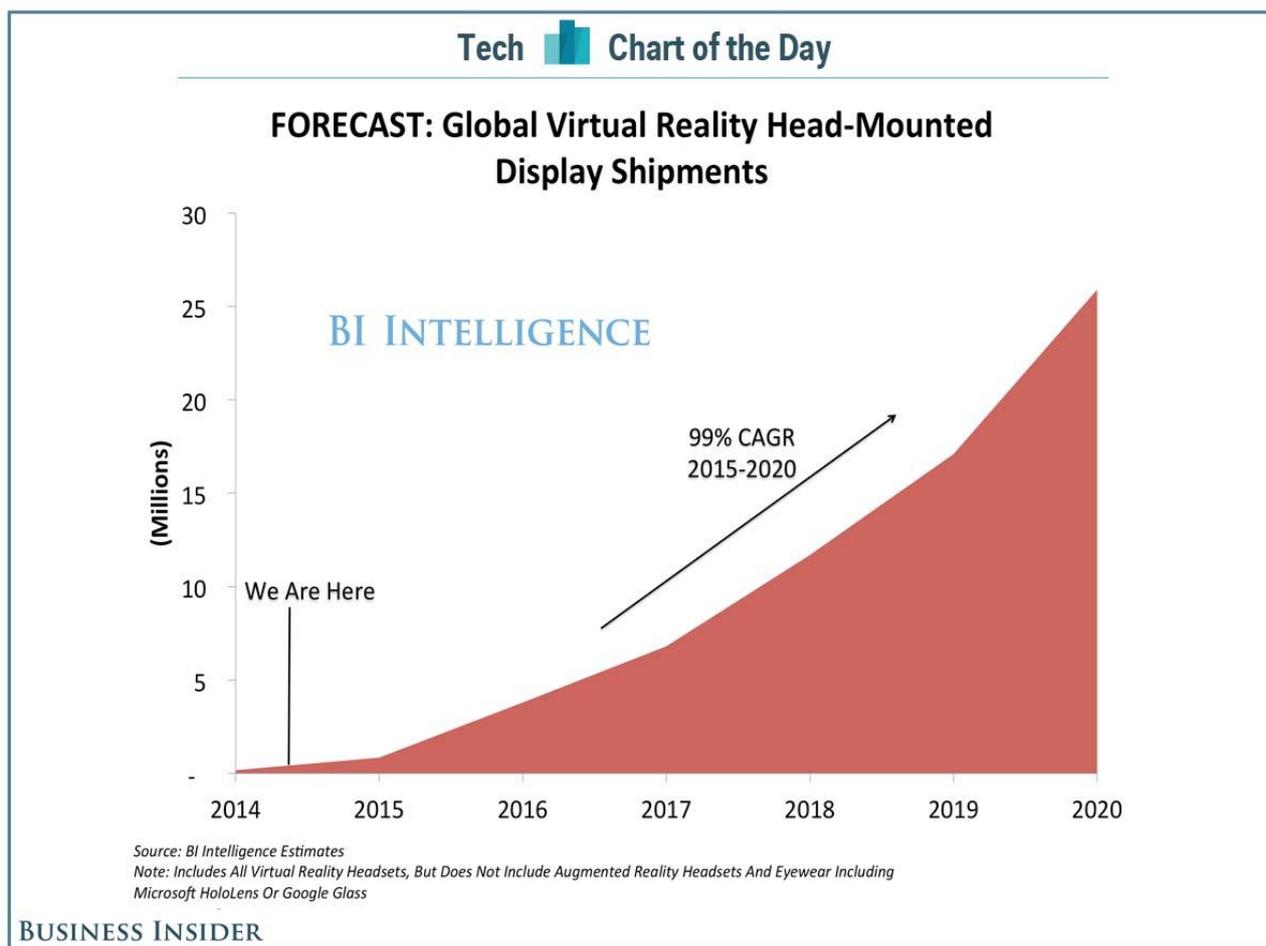


Figura 1.4: Ejemplo “Estimaciones de Business Insider acerca del futuro crecimiento de la realidad virtual”

1.4 Planteamiento de requisitos

El tratamiento será una aplicación que funcionará en smartphones Android, y deberá ser compatible con los cascos de realidad virtual orientados a smartphone. El reducido precio de los cascos para realidad virtual permite que la aplicación pueda ser usada por más usuarios.

Los ejercicios son visuales, se hace necesario el desarrollo de gráficos 2D y 3D en tiempo real para recrear las distintas actividades. El uso del casco también requiere que estas actividades estén en formato side-by-side (SBS).

El formato side-by-side es una técnica estándar de visualización de gráficos que permite visualizar contenido en cascos de realidad virtual y almacenar información estereoscópica (más conocido como visión 3D). Para ello se coloca la información de cada ojo a cada lado de una imagen, video o render. Al visualizar esto con un casco de realidad virtual se percibe en estéreo con profundidad.

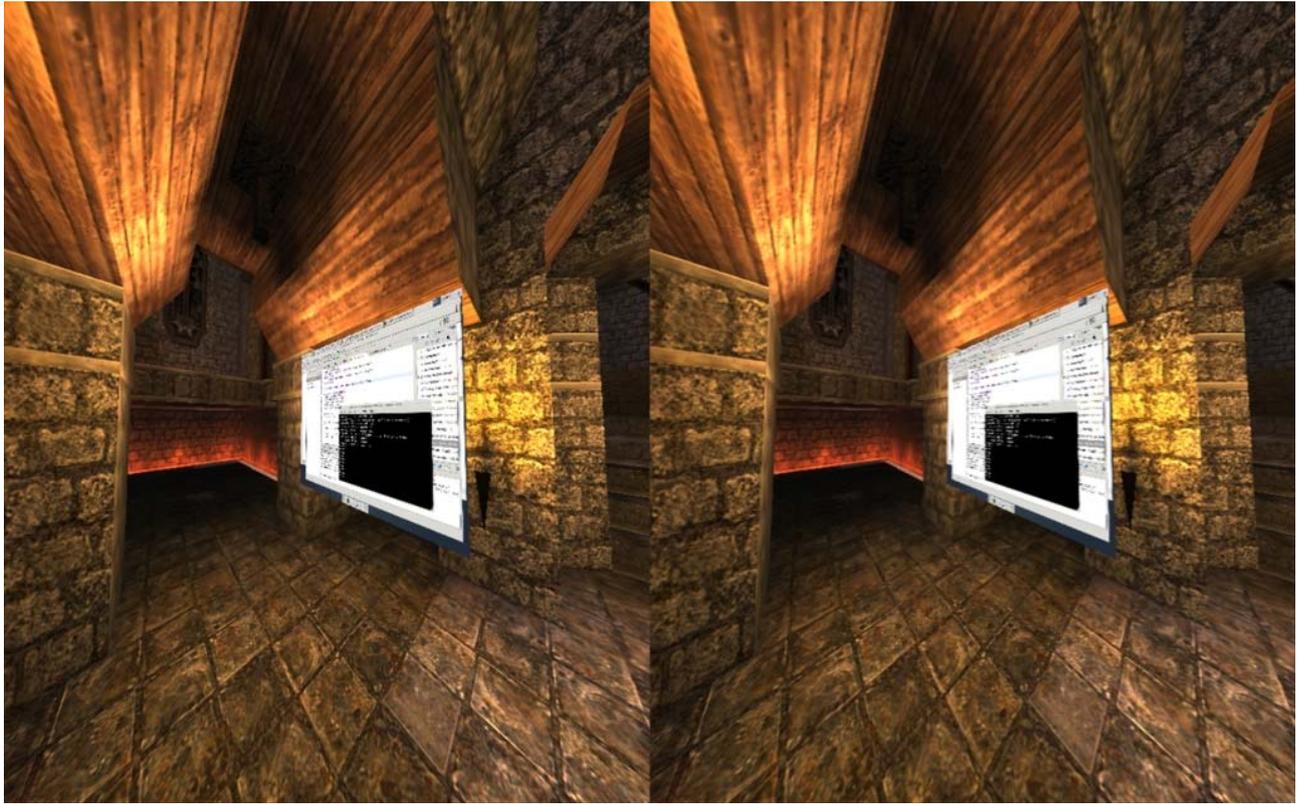


Figura 1.5: Ejemplo “Imagen en side-by-side estereoscópica”

Como dispositivos de entrada disponemos de los propios sensores del smartphone y los gamepads externos. Con ellos debemos de crear menús y eventos que nos permitan controlar las actividades.

Para el funcionamiento correcto de la aplicación-tratamiento el usuario-paciente tiene que cumplir con su uso. Por ello se acordaron varios ejercicios que simplemente fueran actividades comunes realizadas de forma que fueran el propio ejercicio:

- Breakout adaptado para la ambliopía: el Breakout es un juego que todo el mundo conoce. Además los elementos en movimiento en el juego son propicios para el ejercicio. Se tendrá que adaptar para su uso en SBS y también para que no sea aburrido y se pueda jugar durante el periodo del tratamiento.
- Visor de memes: actualmente los “memes” tienen muchos seguidores jóvenes en España que pueden pasarse horas viendo imágenes. Se aprovecharán estas imágenes y la información de internet para conseguir un flujo continuo de información que el paciente quiera ver, y así adaptarla al tratamiento.
- Visor de vídeo para ambliopía: prácticamente todo el mundo disfruta de algún tipo de contenido audiovisual: series, películas, etc.. el visor

permitirá disfrutarlos mientras se lleva a cabo el tratamiento.

1.5 Alcance

El alcance del proyecto comprende los siguientes puntos:

- Menu en VR que permite:
 - Indicar el ojo afectado por la ambliopía
 - Seleccionar el video que queremos visualizar con el reproductor de vídeos
 - Seleccionar cada actividad
 - Visión en SBS
- Breakout
 - Mecánicas básicas del breakout
 - Física irreal de la bola del breakout
 - Dificultad incremental aumentando el número de ladrillos
 - Interfaz señalando número de vidas y nivel actual
 - Compatibilidad con gamepads android
 - Visión en SBS
 - Escalado de todos los elementos visuales del juego en función del tamaño de pantalla.
 - Modificaciones para ejercitar el ojo vago
- Visor de memes
 - Compatibilidad con gamepads android
 - Obtención de imágenes a partir de webs usando expresiones regulares
 - Sistema de carga y descarga de imágenes activo que se ocupa de minimizar los tiempos de carga que pueda provocar su uso así como regular el uso de memoria.
 - Capacidad de navegación por el meme.
 - Modificaciones para ejercitar el ojo vago
 - Escalado de las imágenes en función de su tamaño y el tamaño de pantalla

- Visión en SBS
- Reproductor de videos
 - Capacidad de reproducir todos los vídeos mp4 y 3gp situados en /sdcard/Movies/
 - Escalado del vídeo en función del tamaño de pantalla.
 - Visión en SBS
 - Uso de shader GLSL para crear el efecto necesario para ejercitar el ojo vago en el vídeo.
 - Reproducción en múltiples geometrías para cada ojo.

1.6 Destinatarios

Como destinatario tenemos a las personas afectadas por el ojo ambliope. Con seguridad será efectiva hasta los 10 años y a partir de esa edad hay que probar con los pacientes y determinar si lo ha sido.

Capítulo 2

Elección de herramientas

Las herramientas han sido seleccionadas para cumplir con los requisitos.

2.1 Unity3D

Unity3D es un software con licencia gratuita que permite el desarrollo de aplicaciones de gráficos interactivos 2D o 3D en tiempo real (videojuegos, simulación, etc...) en múltiples plataformas (windows, linux, Mac, Android, iOS). Permitirá desarrollar para Android la aplicación y tenerla preparada para otras futuras plataformas con mucho trabajo ya realizado. Además está tan extendido que la mayoría de plugins o SDK tienen compatibilidad con él. También está muy estandarizado su uso para realidad virtual.

Tiene herramientas para: diseñar mapas, crear/aplicar materiales, importar modelos 3D, debuggear la aplicación, profiler para medir el rendimiento, crear iluminación, físicas 2D/3D, etc...

El uso de Unity3D, sus herramientas y su API acelerarán el desarrollo del proyecto y cumplirá con las necesidades básicas.



Figura 2.1: Ejemplo "Entorno del editor de Unity3D"

2.2 Mono Develop

Mono es un IDE multiplataforma que usaremos para el desarrollo del código de la aplicación y la gestión del mismo.

Soporta completado de código automático, plantillas, plegado de código, refactorización. Tiene herramientas para diversas tareas útiles: un debugger de código, makefiles, unit testing, etc.



Figura 2.2: Ejemplo “Logo de Mono Develop”

2.3 Cardboard SDK for Unity

Es un SDK para la creación de aplicaciones de realidad virtual para smartphones. Permite crear aplicaciones, adaptar las existentes o hacer una aplicación que funcione en el smartphone con o sin la compatibilidad con la realidad virtual. Su principal objetivo es la compatibilidad con la gama “Google Cardboard”.

Provee de características útiles para el proyecto:

- **Head tracking:** a través de los sensores del smartphone es capaz de saber la rotación de la cabeza del usuario para así mostrarle lo que hay en esa dirección en la escena virtual.
- **Side-by-side stereo rendering:** permite visualizar la escena 3D en

estéreo y con cascos de realidad virtual.

- Corrección de distorsiones de las lentes
- Corrección de giro de los sensores.

2.4 Plugin para reproducir video en Android

Permite la reproducción de video en texturas de Unity, algo necesario para el desarrollo de la actividad que funciona con video. Permite aplicar distintos materiales al video y aplicarlos a cualquier geometría 3D, fundamental para manipular la imagen. También nos permite las acciones típicas de un vídeo a través de código como pausar, reproducir, ir al minuto y segundo X.

2.5 Lenguaje de programación C#

C# es un lenguaje de programación moderno, flexible, con un alto nivel de abstracción y con un buen rendimiento. Es la evolución de C y C++. Sus características principales como lenguaje:es de tipado fuerte, declarativo, imperativo, funcional, genérico y orientado a objetos. Dispone de un “Garbage Collector” que se encarga de seleccionar toda la memoria que ya no esté siendo usada por el software y la libera, esto es más cómodo desde el punto de vista del desarrollador pero afecta al rendimiento si lo comparamos con c++ o una gestión propia de la memoria. En comparación a c++ que permite herencia múltiple, C# está limitado a herencia simple aunque dispone de interfaces.

En este proyecto se usa C# como lenguaje de programación y la API de Unity3D principalmente. El compilador de Mono genera un bytecode multiplataforma.

2.6 Shaders GLSL

Los shaders son un tipo de programa muy específico que nace en el ámbito de la computación de gráficos para crear sombreados realistas en aplicaciones gráficas en tiempo real, de ahí su nombre. Una vez superado el problema de los sombreados se empezó a usar para efectos visuales diferentes dado que eran muy eficientes y flexibles. Los shaders fueron creados para su ejecución dentro de la unidad de procesamiento gráfico(GPU) principalmente y a la hora de desarrollarlos hay que seguir una serie de consideraciones respecto al

hardware que se usa. Son ampliamente usados en la industria del cine, de la infografía y los videojuegos.

Con ellos se pueden conseguir acabados hiperrealistas de materiales como piedra, madera, porcelana, cristal. También podemos conseguir efectos más irreales



Figura 2.3: Ejemplo "Shaders funcionando en tiempo real con DirectX"

Hay distintos lenguajes con los que podemos desarrollar un shader como GLSL, HLSL, CG y FX. Dependiendo de la plataforma usaremos uno u otro. En el caso de este proyecto, está orientado a Android y éste usa OpenGL ES que tiene como lenguaje de shaders GLSL.



Figura 2.4: Ejemplo “Demo de OpenGL ES usando shaders en dispositivos móviles”



Figura 2.5: Ejemplo “Shader de terciopelo funcionando con DirectX 11”

Capítulo 3

Desarrollo Técnico

En base a los requisitos, objetivos y alcance, este capítulo describe el proceso y los detalles de la ejecución técnica del desarrollo del proyecto. En él figuran las fases de desarrollo, los problemas que han aparecido y cómo se han solventado.

Comenzaremos con el análisis y la visión general inicial que se hizo del proyecto antes de la creación de la primera línea de código.

3.1 Análisis inicial

Lo primero que se aprecia es que el proyecto a un alto nivel está dividido en cuatro partes de las cuales tres son las actividades independientes y la restante (el menú) actuará como vía de acceso a las tres actividades. Estas actividades son el visor de memes, Breakout y el reproductor de vídeo, que junto con el menú suponen 4 grandes bloques de desarrollo.

Tienen funcionalidades con conexión entre sí, lo que nos permite reutilizar código y pone en valor identificar bien con qué código podemos hacerlo dado que facilita el desarrollo si lo tenemos en cuenta desde el principio:

- La necesidad de que el contenido a visualizar sea redimensionado en función del tamaño de pantalla es parecida en el visor de memes y en el reproductor de vídeo. La diferencia es que el visor de memes permite que sobresalgan verticalmente las imágenes para navegar a través de ellas posteriormente y el visor de vídeos no. Aunque no se pudo adaptar un código común debido a las necesidades particulares de cada redimensión, sí que se pudo reutilizar parte de el código del visor de vídeos modificado en el visor de memes.
- Reconocimiento de comandos comunes. Todas las aplicaciones tienen un botón de acción y un botón de regreso al menú común.

- Shader GLSL: usar un shader nos permite cambiar cualquier textura y modificar cómo se visualiza para que se realice el ejercicio visual, de esta manera todo lo que podamos representar en una imagen podríamos convertirlo en un ejercicio con un mismo código. Los shaders son una tecnología muy eficiente que permite hacer efectos imposibles de hacer eficientemente de otra forma en móviles. El visor de memes es al fin y al cabo un visualizador de imágenes y el reproductor de vídeo es un reproductor de una secuencia de imágenes y audio. Si les aplicamos este shader a esas imágenes podremos adaptarlo fácilmente para que sean un ejercicio. También sirve de base para futuras aplicaciones o un SDK para la ambliopía.

3.1 Fases del desarrollo

3.1.1 Desarrollo del visor de memes

Decidí empezar con uno de los ejercicios, el visor de memes, y terminarlo. De esta forma me permitiría encontrarme con muchas dificultades técnicas que una vez aprendidas sería más fácil superar en el resto de actividades, y en el mejor de los casos podría reutilizar código como comentaba anteriormente.



Figura 3.1: Ejemplo “Ejemplo de un meme cualquiera de internet”

¿Qué es un meme?

Un meme es un fenómeno de internet, es una imagen, video, concepto, producto o suceso que se populariza en internet. Dentro de los memes, hay una gran parte que son imágenes que tratan de producir humor, crítica o reflexión. Hay páginas en las que diariamente son publicados por los usuarios decenas de memes que miles de jóvenes siguen, invirtiendo incluso horas diarias. Este tipo de memes, las páginas webs y su público me hicieron considerarlo como una gran opción para que los usuarios de la aplicación la usaran diariamente. De hecho conozco un caso muy cercano de ambliopía que diariamente invierte una hora en ver memes y como ese caso hay muchos más. Esa persona y todas las de su mismo perfil (que son bastantes entre los 14-30 años) con esta actividad podrán seguir haciendo lo mismo que hacen siempre pero de una manera que les ayude a mejorar el problema de su visión.

¿En qué consiste el visor de memes exactamente?

En él podrás acceder a los memes de las dos mejores páginas de memes de España con un visor de realidad virtual para smartphones. Al mismo tiempo la forma en que se visualiza la imagen está modificada para que al mirarla realices un ejercicio visual beneficioso para la ambliopía. Con el gamepad para android externo puedes pasar al siguiente meme, al anterior y navegar en vertical por el meme dado que la mayoría están en un formato “portaretrato” en el que la altura de la imagen es superior al ancho.

Desarrollo

El desarrollo comenzó con el módulo que se encargaba de la gestión de los memes desde esas páginas. Consiste en un sistema para la descarga, el almacenamiento, asignación de identificadores a cada meme y descarga de los memes más lejanos en la lista de memes.

Para obtener los memes, se accede a los enlaces de las imágenes a partir del código HTML de la página. Para ello se usaron expresiones regulares. El código HTML de las dos páginas de memes que tenía previsto integrar seguían una estructura muy similar. Los memes venían de cinco en cinco divididos en páginas a las cuales se podía acceder insertando a continuación del nombre del dominio la terminación “/p/<número de página web>”. En base a esto podemos concluir que para acceder a la imagen de meme nº 17 tendríamos que realizar estas operaciones:

$$n^{\circ} \text{ de página web} = (int)(17/5)$$

$$\text{posición del meme dentro de la página} = 17 - (n^{\circ} \text{ de página web} * 5)$$

Hay una estructura común de etiquetas div que contenían una etiqueta img que accedía a un subdominio de cada página dedicado a alojar las imágenes. En base a estas consideraciones se crea la expresión regular que extrae el “src” de la etiqueta “div”.

Fue necesario el estudio de la librería expresiones regulares de C#. Una vez terminada y depurada la parte de expresión regular, usando la característica de ejecución dentro del editor de Unity3D lo compilé y exporté para Android y no funcionaba la extracción de imágenes. Se estaba descargando correctamente el código HTML pero la expresión regular no estaba detectando nada. El problema estaba en que el código HTML que se descargaba en el móvil y en el PC de la misma página eran diferentes. Esto se debió a que desde el móvil la llamada http estaba indicando que estaba siendo realizada a través de un smartphone Android y automáticamente la web devolvía una versión responsive adaptada para móviles. Se realizó la adaptación de la expresión regular para la versión mobile de la web, lo que corrigió los problemas en la obtención de imágenes de la web.

La gestión del almacenamiento de los memes en la memoria pretendía que el usuario pudiera visualizar los memes con total fluidez en el móvil teniendo como único cuello de botella la tasa de transferencia a internet del usuario. Está claro que esta actividad estaba pensada usando la WIFI debido al alto consumo que puede tener en una red móvil. Respecto a la carga en la memoria, las imágenes son almacenadas en una tabla hash de tipo diccionario (“*new Dictionary<int, Texture2D>()*”) que como clave tenía un entero que lo identifica según el orden en la web y como valor una imagen que se almacena en el tipo Texture2D específico de Unity3D para ello. Para saber qué es lo que hay que cargar o descargar siempre se almacena un índice con el valor del puesto del meme actual según la página, y en función de su posición tratamos de tener cargados siempre un número páginas de memes “n” anterior, posterior y la página en la que se encuentra el índice localizado. Hay que tener en cuenta que cada página contiene 5 memes y se cargan de cinco en cinco, de ahí que la carga y la descarga se haga de cinco en cinco también. Por ejemplo, si consideráramos ese número n=“1 página” de margen por cada lado y el meme actual fuera el número 11 entonces quedaría algo como en la siguiente tabla:

...	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	...
-----	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

El azul claro serían los memes que pertenecen a esa página de margen que ha de tener cargada de forma contigua a la página actual siempre. En amarillo están los memes cargados por pertenecer a la página actual en la que se encuentra el meme actual (en verde). En blanco quedan los memes que no están cargados y que se descargan en cuanto se cambia de página actual.

Podríamos resumir el ciclo de carga de memes en el siguiente esquema:

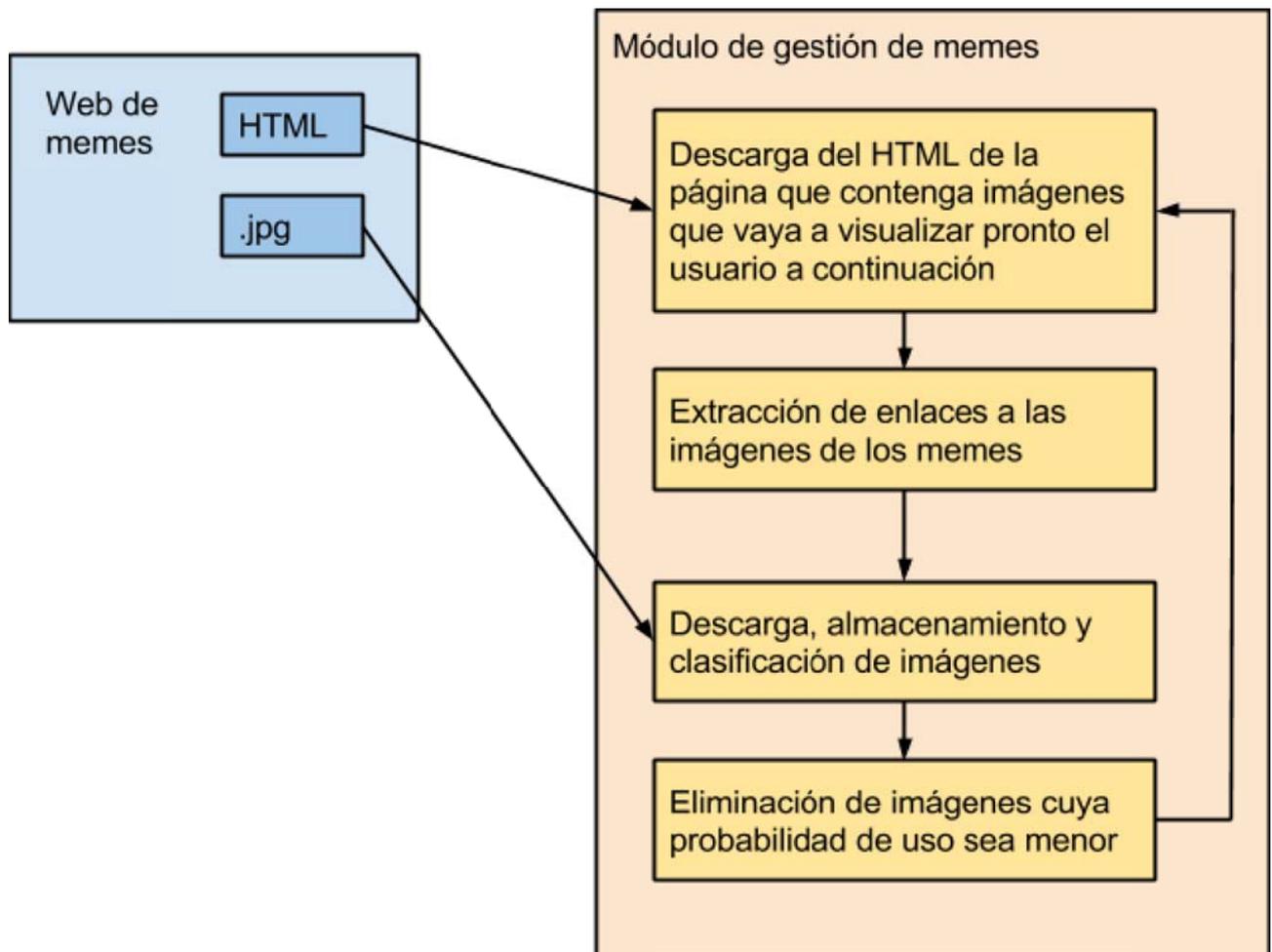


Figura 3.2: Ejemplo “Esquema de la relación y funcionamiento entre las webs de memes y el módulo de gestión de memes de la aplicación”

A continuación se trabajó en la visualización de los memes. Con el sistema de almacenamiento descrito previamente el visualizador solo tiene que pedir el meme actual y éste resuelve la operación automáticamente. La visualización de los memes comprendía tres problemas:

- Visualización side-by-side
- Ajuste dinámico de una imagen a la pantalla (y en SBS)
- Control de la imagen
- Efecto del ejercicio visual

Al tratarse de una perspectiva ortográfica totalmente bidimensional, para crear el efecto side-by-side lo único que se hizo fue crear dos cámaras (ortográficas) y ponerlas en la misma posición siempre. Cada una tiene restringida la mitad de la pantalla para renderizar.

Una vez solucionado el side-by-side se investigó cuáles eran los cálculos necesarios para adaptar las dimensiones de un plano 3D a los bordes en ancho de lo que la cámara captaba en ortográfico (contando con que era en SBS). Las imágenes de memes tienen comúnmente una disposición vertical con una altura superior al ancho, de ahí que se ajuste al ancho.

Se incorporaron los controles para mover verticalmente la imagen y navegar por ella además de pasar al siguiente y anterior meme.

3.1.2 Desarrollo del efecto para la ambliopía

Llegado a este punto del proyecto, se empezó con una de las tareas más importantes del mismo. Investigar como hacer el ejercicio visual. Para empezar se generó una cuadrícula que hacía que en un ojo se viera una parte de la imagen y se tuviera que usar el otro para completarla. Era una primera propuesta que se decidió con el especialista en oftalmología. Éste fue el resultado:

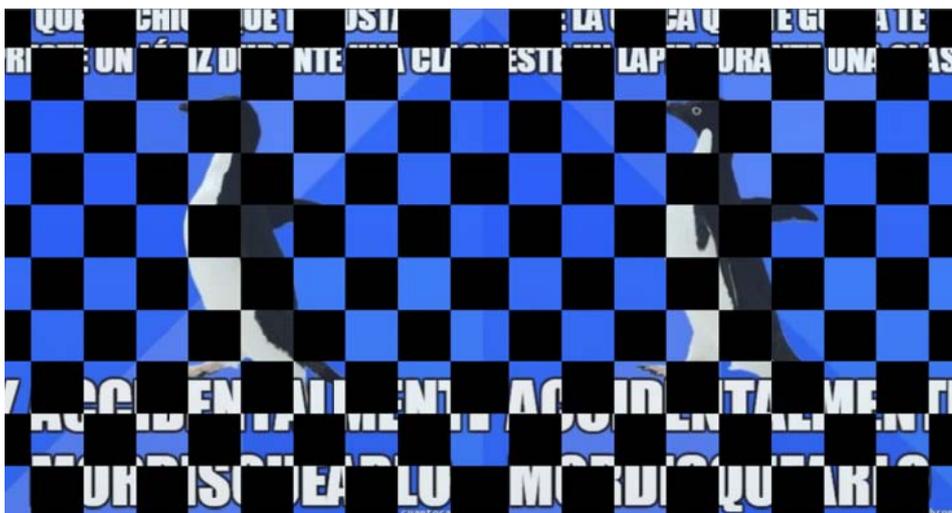


Figura 3.3: Ejemplo “Primera propuesta del prototipo”

El primer problema detectado fue que la visión con los cascos no abarcaba toda la pantalla, lo que hizo que se le añadiera posteriormente un margen lateral que permitiera ver el meme en un tamaño visible. Lo de los cuadros fue inviable, era demasiado aguda la diferencia entre los dos ojos, y no se

llegaba a fusionar la imagen. Se hizo una siguiente prueba solo con un ojo con la cuadrícula aplicada y mejoró pero no lo suficiente, seguían siendo muy bruscos los cuadros a la hora de visualizar la imagen. Se propuso en vez de poner cuadros negros, hacerlos en otro color, pero dependiendo de la imagen funcionaba o no, porque cada imagen tenía su color en el que se visualizaba de forma más cómoda y aún así, en una propia imagen un color podía ser “cómodo” para el ojo en una parte y no serlo en otra.

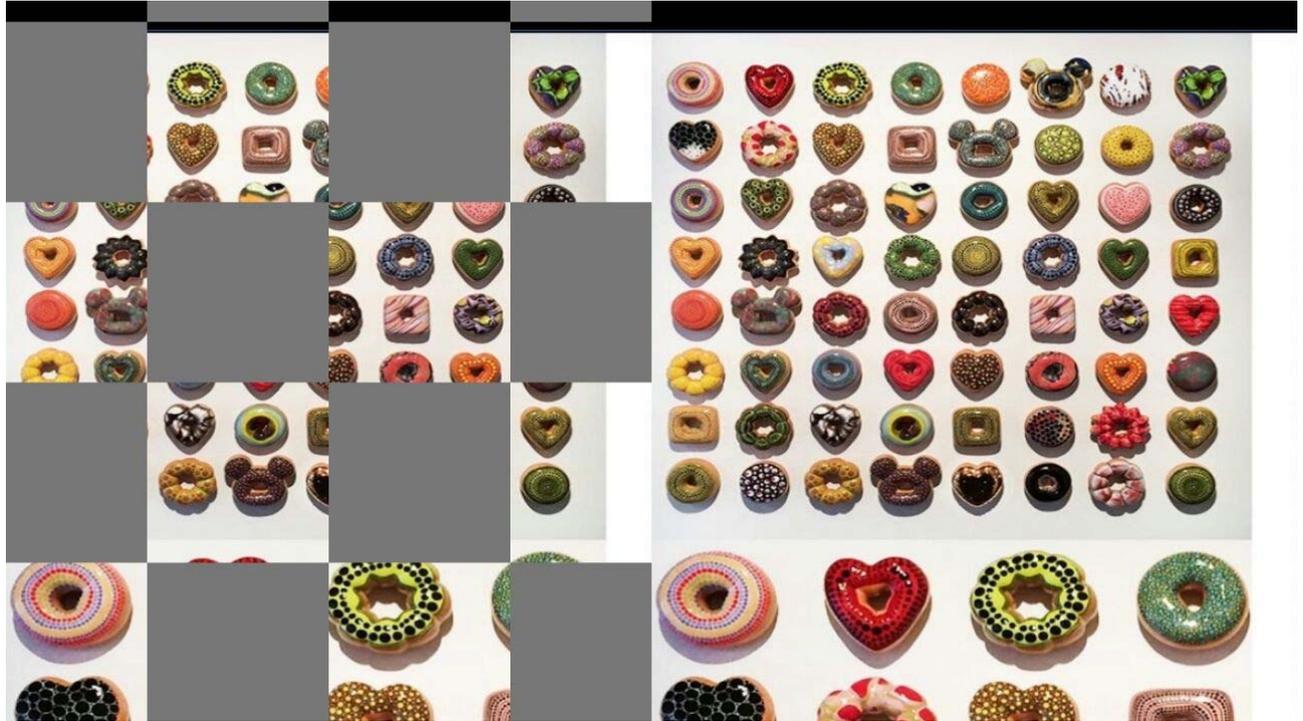


Figura 3.4: Ejemplo “Segunda propuesta”

La opción que solucionó finalmente el problema fue difuminar en cada cuadrado lo que ocluía en la imagen. Era un efecto mucho más complicado de desarrollar que los anteriores, se desarrolló un shader en GLSL para OpenGL ES de “blur” selectivo que actuaba solo en los cuadrados y en el resto renderizaba la imagen de forma convencional. También tiene parámetros que permiten controlar el tamaño de los cuadros.



Figura 3.5: Ejemplo “Reproductor de memes funcionando con el efecto visual aplicado”

3.1.3 Desarrollo del visor de vídeos

El primer paso fue estudiar el funcionamiento del plugin que permite reproducir videos. Lo que hace es transmitir a una textura los fotogramas del video y reproducir el audio incorporando todas las funciones como play, pause, etc a través de código. Esto nos permitía aplicar el shader que habíamos desarrollado previamente para crear el ejercicio visual. Hubo que hacer adaptaciones del código del plugin para reproducirlo en dos planos (uno para cada ojo).

Se reutilizó el código que dimensionaba en pantalla los memes para el vídeo, modificando parte de su comportamiento debido a las necesidades de este visor en concreto.

Posteriormente se añadieron los controles para el visor que permitían acceder a todas las características convencionales de un reproductor de video.

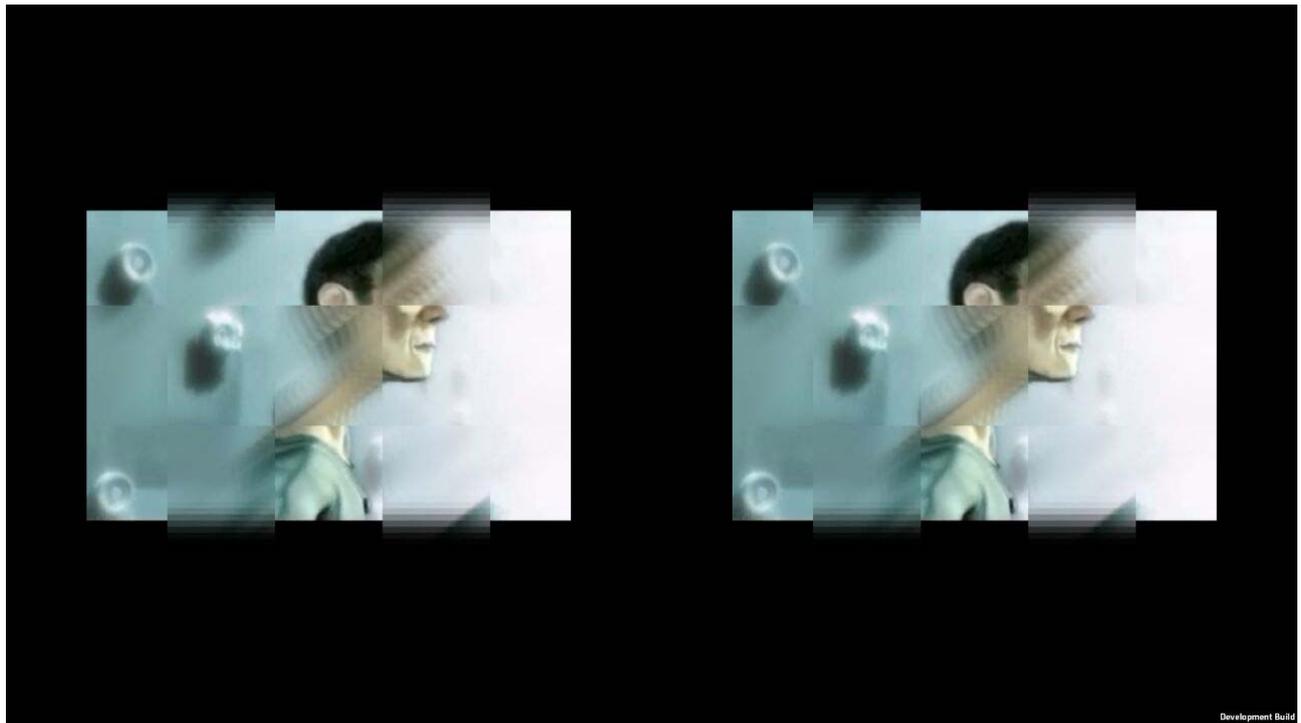


Figura 3.6: Ejemplo “Reproductor de vídeo funcionando con el efecto visual aplicado”

3.1.4 Desarrollo del Breakout

El breakout es un juego muy conocido en el que hay que tratar que una bola rompa todos los ladrillos rebotando con una plataforma que controlamos sin que se salga fuera de los límites del mapa.

Para el desarrollo de las físicas se utilizó el motor de físicas 2D de Unity. Se empezó por delimitar los bordes del mapa y establecer los tamaños de los objetos del juego (la bola, la plataforma y los ladrillos) en función del tamaño de pantalla en SBS. Se consiguió obteniendo un factor de escala común que se pudiera aplicar a todas las escalas y posiciones al comenzar la actividad.

Lo más complejo fue conseguir que la física de la bola fuera la correcta. La bola del breakout no sigue un comportamiento físico normal, siempre tiene una velocidad constante, al rebotar no pierde energía ni sufre fricción, los ángulos que toma son exactos, hay que evitar que la bola entre en un bucle rebotando eternamente, etc... Para conseguir esto hubo que ajustar muchos parámetros y controlar a través de código todas las situaciones.

Se desarrolló un gestor del gameplay que se encargaba de posicionar los bloques, gestionar el número de bolas del que disponía el player y el nivel en el que se encontraba. Para que el juego fuera mínimamente divertido se hizo que hubieran varios niveles de dificultad. Comienza en el nivel uno, el más facil, que tiene 2x2 ladrillos, el dos tiene 3x3 ladrillos, el tres 4x4 ladrillos,... y así continúa la progresión. Se va aumentando el número de ladrillos con cada nivel y así su dificultad también. Cada vez que una bola se pierde o se

aumenta un nivel de dificultad hay una interfaz que es controlada por este gestor y que muestra esa información al usuario.

Respecto al ejercicio visual, en este decidimos hacer uno diferente para realizar pruebas. El ejercicio consiste en que la bola solo se ve a través del ojo afectado.

Se integró soporte para gamepads Android de forma que con él se puede controlar la plataforma, moverla horizontalmente y lanzar la bola al inicio.

A continuación podemos ver el resultado, con las interfaces en el nivel 1 y el 3 ajustando como ojo amblíope el ojo derecho:

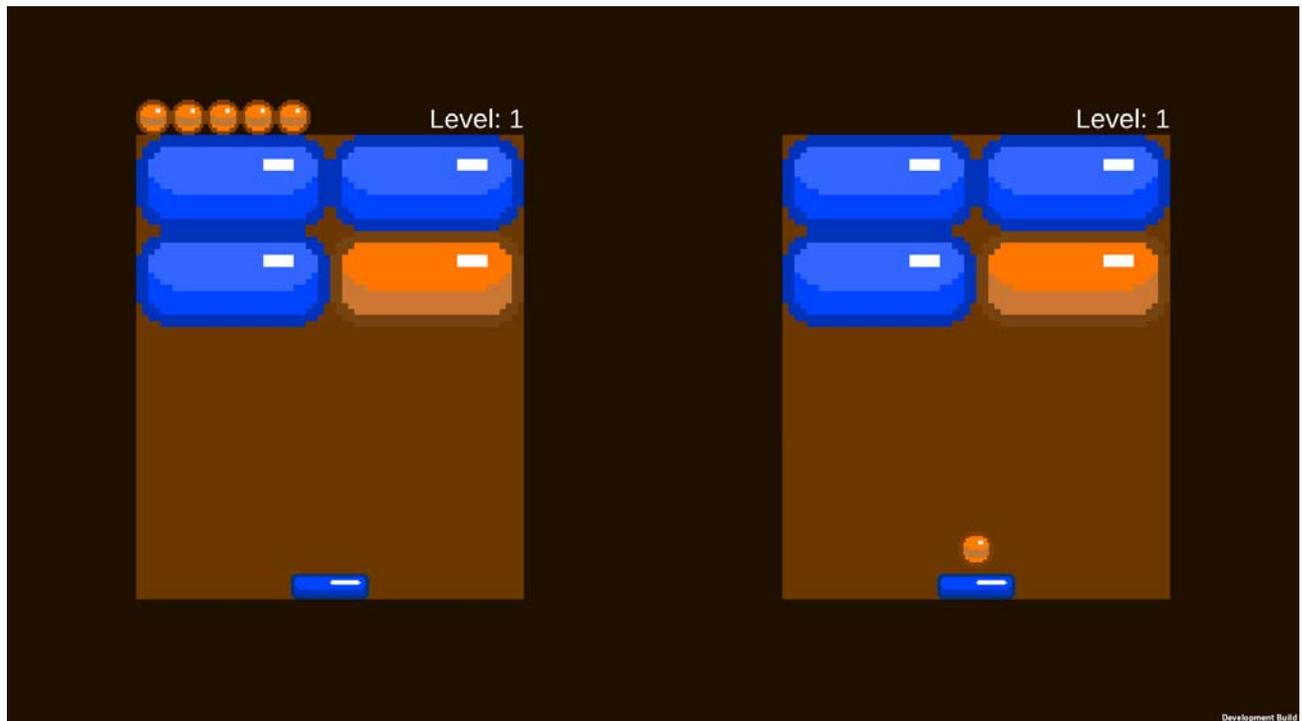


Figura 3.7: Ejemplo “Captura de pantalla en el nivel 1 del Breakout”

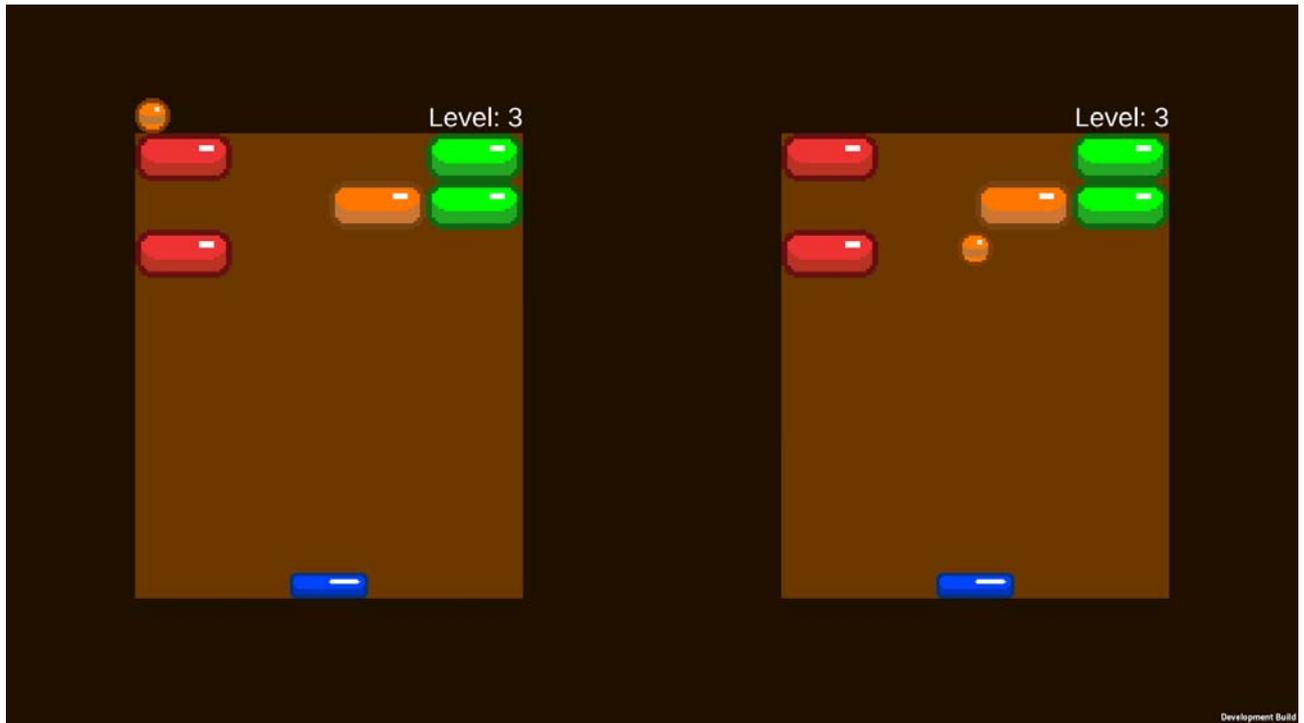


Figura 3.8: Ejemplo “Captura de pantalla en el nivel 3 del Breakout”

3.1.5 Desarrollo del Menú

Por último era necesario tener un menú con el cual poder acceder a todas las actividades. Existen 4 paneles en el menú:

- Breakout
- Meme viewer
- Video player
- Config

Para crear los botones de los paneles, se usa la librería de cardboard SDK que provee de funciones para pulsar los botones en el entorno virtual.

El panel del breakout y el meme viewer solo tienen un botón cada uno que ejecuta cada una de las actividades. En el panel config hay dos botones para indicar que ojo es el afectado y que los ejercicios se adapten en función de ello. Se cambia una variable estática que cada actividad reconoce posteriormente para modificar su manera de visualizar para cada ojo.

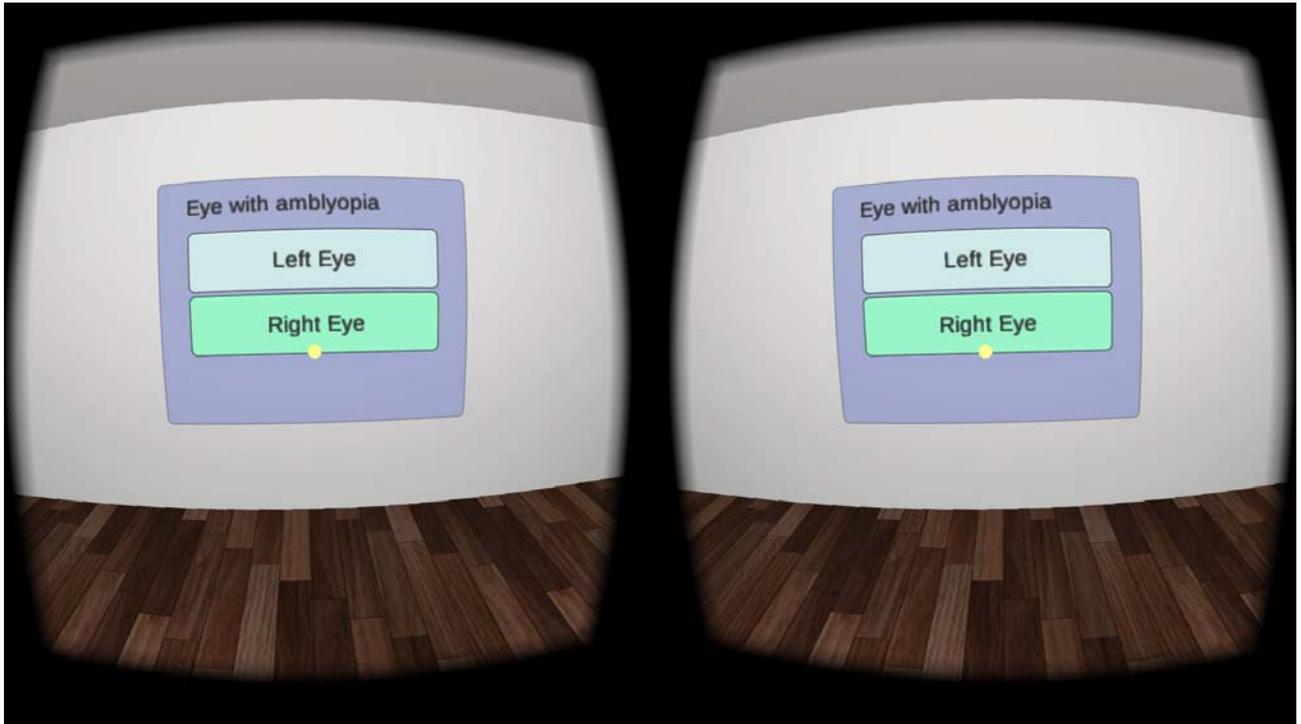


Figura 3.9: Ejemplo “Panel de configuración que permite seleccionar el ojo afectado”

Para la reproducción de video era necesario hacer un panel en el que poder seleccionar el video. Para ello se ha creado un panel como este:



Figura 3.10: Ejemplo “Panel para reproducir un vídeo”

El menú reconoce los vídeos que están en la ruta “/sdcard/Movies” del teléfono. Tenemos botones para pasar al video siguiente y anterior de esa carpeta y el botón “Play” para reproducirlo.

Capítulo 4

Conclusiones y líneas futuras

El desarrollo de este proyecto ha sido muy gratificante dado que he cubierto todos los objetivos que me propuse al hacerlo. He aprendido muchas cosas, he realizado un proyecto que tiene parte de investigación con toda la aventura que ello conlleva, he colaborado con otros campos ajenos a la ingeniería informática, he terminado el desarrollo técnico con éxito y lo más importante: existe un tratamiento accesible que permitirá a muchas personas con problemas visuales mejorar.

Todavía queda una cuestión muy importante que descubriremos con el tiempo: ¿Qué efectividad tiene el tratamiento? Hay que realizar un estudio con personas que estén dispuestas a comprometerse con las actividades y dar información de la evolución. Ese estudio puede llevar muchos meses pero tarde o temprano sabremos los resultados.

Capítulo 5

Summary and Conclusions

The development of this project has been very rewarding because I have covered all the goals I set out to do. I have learned many things, I have done a project with research part with all the adventure that entails, I have collaborated with other fields outside computer engineering, I have finished the technical development successfully and most importantly: there is an accessible treatment and it will allow many people with visual problems improve.

There is still a very important issue that we will discover with time: How effective is the treatment? A study must be carried out with people who are willing to commit to the activities and provide information of evolution. That study may take many months, but sooner or later we'll know the results.

Capítulo 6

Presupuesto

Tipos	Descripción	Importe
Casco de realidad de smartphones para VR gama media	Casco necesario para desarrollar y visualizar la aplicación.	100€
Plugin para reproducir video	Plugin que permite reproducir vídeo en Android con Unity 3D	30€

Tabla 6.1: Presupuesto

Bibliografía

- [1] <http://www.ncbi.nlm.nih.gov/pubmed/25131694>
- [2] <http://www.elmundo.es/elmundosalud/2008/03/04/medicina/1204619448.html>
- [3] <http://www.fayerwayer.com/2013/04/jugar-al-tetris-un-nuevo-tratamiento-para-corregir-el-ojo-vago-en-adultos/>
- [4] [http://www.news-medical.net/health/New-hope-for-adults-with-amblyopia-\(Spanish\).aspx](http://www.news-medical.net/health/New-hope-for-adults-with-amblyopia-(Spanish).aspx)
- [5] <http://www.elmundo.es/elmundosalud/2008/03/04/medicina/1204619448.html>
- [6] <http://www.fayerwayer.com/2013/04/jugar-al-tetris-un-nuevo-tratamiento-para-corregir-el-ojo-vago-en-adultos/>
- [7] [http://www.news-medical.net/health/New-hope-for-adults-with-amblyopia-\(Spanish\).aspx](http://www.news-medical.net/health/New-hope-for-adults-with-amblyopia-(Spanish).aspx)