

UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

I.T. TELECOMUNICACIÓN (SIST. DE TELECOMUNICACIÓN)



UNIVERSIDAD
POLITECNICA
DE VALENCIA



ESCUELA POLITECNICA
SUPERIOR DE GANDIA

**“Estudio de viabilidad del entorno de
programación Unity3D para la
Rehabilitación Cognitiva contra el
Alzheimer”**

TRABAJO FINAL DE CARRERA

Autor/es:

Pablo Soriano Maudos

Director/es:

D. José Francisco Montserrat del Río

GANDIA, 2012

ÍNDICE DE CONTENIDOS

1. Introducción	3
1.1. Motivación del proyecto.....	3
1.2. Relevancia del proyecto.....	5
1.3. Objetivos	6
1.4. Originalidad	7
1.5. Aplicabilidad	8
1.6. Estructura del proyecto	9
2. Unity3D - Conceptos básicos	10
2.1. Assets	10
2.2. Scenes	11
2.3. GameObjects	11
2.4. Components	13
2.5. Scripts	13
2.6. Prefabs	15
3. Unity3D - Aplicabilidad	16
3.1. Programación en Unity3D	17
3.2. Licencias	17
3.2.1. Licencias gratuitas	18
3.2.1.1. Web Player	18
3.2.1.2. PC y Mac	19
3.2.2. Licencias de pago	19
3.2.2.1. Flash Player	20
3.2.2.2. iOS	22

3.2.2.3. Android	23
4. Descripción de la aplicación implementada	25
4.1. Introducción a la aplicación	25
4.2. Menú principal	25
4.2.1. Introducción.....	25
4.2.2. Programación y desarrollo	26
4.2.3. Funcionamiento	30
4.3. Qué soy	32
4.3.1. Introducción.....	32
4.3.2. Programación y desarrollo	32
4.3.3. Funcionamiento	46
4.3.4. Aplicabilidad al estudio de la enfermedad del Alzheimer.....	48
4.4. Aventura gráfica	49
4.4.1. Introducción.....	49
4.4.2. Programación y desarrollo	49
4.4.3. Funcionamiento	58
4.4.4. Aplicabilidad al estudio de la enfermedad del Alzheimer.....	59
4.5. Demostración audiovisual	59
4.5.1. Introducción	59
4.5.2. Video explicativo	60
5. Conclusiones y líneas futuras	60
6. Bibliografía.....	63
7. Anexos	63

1. INTRODUCCIÓN

1.1. MOTIVACIÓN DEL PROYECTO

El Alzheimer no se puede considerar ni mucho menos una enfermedad minoritaria. De hecho, España se encuentra entre los 5 países europeos con más enfermos de Alzheimer. La media Europea se sitúa en torno al 1,27% de enfermos con respecto a la población. Según indica el "Anuario de Demencias 2006" el porcentaje de enfermos sería el siguiente.

	PAÍS	PORCENTAJE DE ENFERMOS	POBLACIÓN TOTAL (millones de hab.)
	ITALIA	1,55%	60,78
	SUECIA	1,54%	9,42
	ESPAÑA	1,36%	47,19
	ALEMANIA	1,36%	82,61
	FRANCIA	1,36%	66,01

Gráfico 1: Tabla de porcentaje de enfermos de Alzheimer en la UE.

Polonia (0,92%), Portugal (1,23%) y Eslovenia (1,09%) se encontrarían en la parte más baja de esta clasificación europea.

El 77% de los enfermos son mujeres y se estima que todavía existen unos 200.000 casos sin diagnosticar.

Además de tratarse de una enfermedad degenerativa, considerablemente extendida entre la tercera edad y sin cura, estamos hablando de una enfermedad que supone un altísimo coste para la población. Actualmente se estima que el gasto anual para el tratamiento del Alzheimer ronda los 24.000 millones de euros, afectando a 3,5

millones de personas, que comprenden a los enfermos y a sus cuidadores. Se estima que las cifras de gasto se dupliquen en 2 décadas alcanzando casi los 50.000 millones de euros.

Es evidente que lo más relevante de este hecho no es el gasto económico que conlleva la enfermedad, sino desgaste físico y mental que sufren tanto enfermos como cuidadores.

Es por ello que este proyecto puede resultar de gran interés para este sector tan afectado que supera el 7% de la población nacional.

La evolución tecnológica en estos últimos años ha sido evidente. Aquellas funciones que estaban pensadas para ser realizadas de forma manual e intuitiva, ahora han sido mecanizadas y automatizadas para realizarse mediante sistemas informáticos. El hecho de pensar que las nuevas tecnologías están reservadas únicamente para que las usen las generaciones más jóvenes es caer en un error.

Convencido de que las posibilidades de mejora son sustanciales y las opciones de evolución en muchos aspectos son evidentes, he decidido trabajar con Unity3D.

Unity3D es un motor 3D para el desarrollo de videojuegos. Se puede habilitar para su aplicación en distintas plataformas, como Tablet, Wii, iPhone o los sistemas operativos Windows y Mac OS X. Actualmente se trata de una aplicación novedosa y en plena expansión. Muy gráfica e intuitiva, Unity 3D ofrece notables facilidades para el diseño gráfico tridimensional.

Aplicado normalmente para diseños arquitectónicos y animaciones 3D, incluyendo los videojuegos, con la elaboración del proyecto se pretende darle una nueva funcionalidad a este motor. Aplicarlo con este objetivo puede ser un avance en el estudio de esta enfermedad en el ámbito informático con todas las mejoras que ello conlleva.

Todos estos factores unidos nos han llevado a crear la aplicación sobre la que se va a hablar en este texto y que se espera que tenga una buena acogida entre la gente que la vaya a utilizar en un futuro.

1.2. RELEVANCIA DEL PROYECTO

Unity 3D es una plataforma de desarrollo de videojuegos o cualquier otro contenido interactivo en tres dimensiones, como visualizaciones arquitectónicas o animaciones 3D en tiempo real.

Este programa es de reciente desarrollo y está en plena expansión. En el año 2009, Unity Technologies fue nombrada como una de las 5 mejores compañías de ese año. El trabajo se realiza sobre una de sus últimas versiones, la 3.5.2., lanzada al mercado en Abril de 2012.

Al tratarse de una aplicación de tan reciente aparición y en la que constantemente se están mejorando aspectos con nuevas versiones, todavía no se ha explorado suficientemente sus posibilidades de desarrollo. Se pretende en este trabajo final de carrera explorar todos los horizontes que se puedan alcanzar con las posibilidades que nos permite Unity3D en el ámbito de estudio del Alzheimer y la viabilidad de aplicación para la ralentización de la enfermedad.

Debe realizarse y resulta necesario hacer un estudio tecnológicos para ver sus capacidades, complejidad de implementación, portabilidad, etc. Este estudio, además, ayudará a las personas de la tercera edad mediante ejercicios de entrenamiento de la memoria.

1.3. OBJETIVOS

Vista la evidente necesidad de una evolución en el tratamiento de la enfermedad del Alzheimer y que, a día de hoy, todo con lo que trabajamos va adquiriendo un carácter tecnológico, se ha decidido trabajar en este proyecto.

El objetivo primordial de este proyecto es realizar un estudio de viabilidad de la tecnología Unity 3D de cara al tratamiento de la enfermedad de Alzheimer y la ralentización de su degeneración.

Este objetivo principal se puede dividir en los siguientes objetivos específicos:

1. - Ofrecer un ayuda para todos aquellos a los que, de forma directa o indirecta, les está afectando la enfermedad de Alzheimer y favorecer al retraso en la degeneración de la enfermedad.

Este grupo de beneficiarios comprenderá desde los propios enfermos hasta los familiares o colaboradores de organizaciones de enfermos de Alzheimer, que saldrán ampliamente beneficiados de todas las mejoras que les va a proporcionar el desarrollo de este proyecto de cara a una facilitación del trabajo a realizar de cara a la ralentización de la degeneración de la enfermedad del Alzheimer.



Gráfico 2: Paciente realizando juegos de asociación de formas.

2. - Implementar diversas actividades o juegos basados en el lenguaje utilizado en Unity3D. Estos juegos estarán basados en actividades que, hasta ahora, se realizaban mediante ejercicios sobre papel e implicarán razonamientos no excesivamente complejos.

Según referencias obtenidas a partir de la Asociación de Familiares de Enfermos de Alzheimer de Valencia (AFAV) estos juegos estarían destinados a los pacientes cuya enfermedad aún no ha degenerado demasiado, pues a los pacientes con graves deterioros mentales ya no les resulta útil este conjunto de ejercicios.

3. - Estudiar la viabilidad de aplicación que tienen las actividades o programas sobre algunos soportes como puedan ser Android, PC, ... Todo ello podrá facilitar el uso y mejorar el rendimiento de la aplicación que se va a implementar.

1.4. ORIGINALIDAD

La nueva aportación de la regeneración de las células nerviosas en adultos a través de la estimulación ambiental no farmacológicas abre nuevos caminos con insospechadas posibilidades terapéuticas. Desde los años 70 se han sugerido gran variedad de técnicas para modelar respuestas de orientación adecuadas, como la reorientación a la realidad, la re-motivación, la terapia de refuerzo de conocimientos, estimulación sensorial, rehabilitación cognitiva, ...

Son varios los métodos de trabajo que se han venido utilizando a lo largo de esta última década para la cura del Alzheimer. Entre ellos, destacan videojuegos tan conocidos como *Braintraining*. Aquello que ya es más que conocido es que la mejora de las aplicaciones ha favorecido a ralentizar la degeneración de esta enfermedad que a tantas personas afecta.

Tratar de elaborar una aplicación con esta finalidad sobre la base que establece Unity 3D es algo pionero. Anteriormente nunca se había intentado esto y se va a trabajar en darle otra aplicabilidad a Unity 3D, enfocado, en este caso, hacia la cura de la enfermedad de Alzheimer.

1.5. APLICABILIDAD

El producto resultante de este proyecto será mostrado inicialmente a la empresa SAI Wireless. SAI Wireless es una compañía valenciana especializada en la creación, el desarrollo y la integración de productos y servicios para el sector *e-Health* en movilidad. Esta empresa se dedica principalmente a la investigación e innovación de productos diseñados para ayudar a las personas de la tercera edad y a aquellas que no son capaces de desarrollar por sí mismas las actividades normales de la vida diaria.

Se pretende favorecer tecnológicamente a este sector de la población diseñando aplicaciones y aparatos cuyo uso sea simple y sencillo para ellos.



Gráfico 3: Ejemplo de móvil diseñado para la tercera edad.

SAI Wireless es la empresa hacia la que está pensada esta aplicación, pero se pretende que pueda ofrecerse a más empresas del sector y generar, así, empleo de alta cualificación en cuanto a la implementación, la mejora del programa y el diseño de versiones más novedosas y actualizaciones de forma periódica.

1.6. ESTRUCTURA DEL PROYECTO

La estructura de este TFC se divide en 5 grandes bloques. El primer punto que se aborda en la elaboración es la introducción al proyecto, donde se expresa la motivación del propio TFC y la repercusión del Alzheimer en la sociedad actual.

Es en el segundo bloque donde se explican algunos de los conceptos básicos que el lector debe comprender sobre Unity3D de cara a la correcta recepción de la información en los siguientes apartados.

En el tercer bloque se detallan algunos ejemplos de aplicación de Unity3D, en muchos casos centrados en el desarrollo de videojuegos de mero entretenimiento, y se profundiza en el conocimiento de las licencias de Unity3D para obtener mejor calidad de servicio y diferentes plataformas para poder desarrollarlo.

El cuarto bloque trata la aplicación en sí, detallando la programación que se ha utilizado, así como el diseño gráfico en el que se ha trabajado para la correcta configuración del modelo de programa.

El quinto y último bloque valora la aplicabilidad del producto en enfermos de Alzheimer. Con la colaboración de AFAV, la asociación de familiares de enfermos de Alzheimer de Valencia, se realizan pruebas de aplicabilidad y se analiza la reacción de los pacientes de cara a conocer si es o no viable la plataforma Unity3D de cara al estudio de la enfermedad de Alzheimer y su posible tratamiento. También se analizan las posibilidades de mejora, los fallos cometidos y una serie de cambios a realizar de cara a un futuro.

2. UNITY 3D - CONCEPTOS BÁSICOS

La mejora que proporciona Unity 3D con respecto al resto de plataformas de diseño de videojuegos y animaciones es la de trabajar con una tercera dimensión en el espacio. El uso del eje Z es algo novedoso con respecto a lo que ya conocemos.

En esta plataforma se han asentado una serie de conceptos que deben conocerse para poder trabajar a un nivel alto con ellas y que han debido ser tomadas como la base cognitiva para la elaboración de nuestro programa Alzh3D.

Se hablará de estos conceptos a lo largo de este apartado, en los que se explicará el porqué de cada uno de los comandos y qué función tienen.

2.1. ASSETS

Se trata sin duda de los componentes básicos de todos los proyectos de Unity3D. Desde texturas en forma de archivos de imagen, pasando por modelos 3D para mallas y los archivos de sonido destinados a ser usados en los efectos, unidos a los archivos que se van a utilizar para la elaboración del juego.

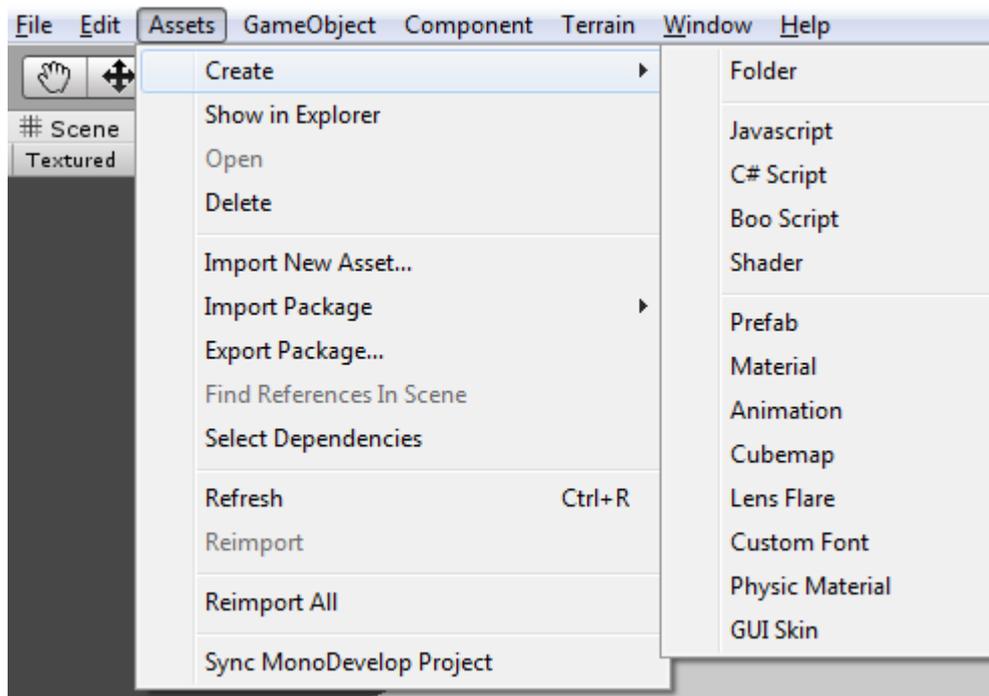


Gráfico 4: Captura del programa Unity3D con la pestaña Assets desplegada.

Los *Assets* o activos se almacenan en la misma carpeta y componen el total de los archivos utilizados a los que se les va a dar un uso de cara al correcto funcionamiento de la aplicación. La carpeta *Assets* está ubicada en la zona del panel del proyecto (Project panel) de la interfaz Unity3D.

2.2. SCENES

En Unity se deben plantear las *Scenes* o escenas como niveles individuales o áreas aisladas del juego, aunque es posible que, en algunos casos, la implementación del caso se pueda realizar en una sola escena, como pudiera ser en juegos de puzzle, desarrollando dinámicamente la carga de contenidos a través de código.

En la elaboración de una aplicación como de la que se trata en este trabajo, compuesta por varias escenas, se será capaz de poder distribuir los tiempos de carga (estos pueden demorar la velocidad del juego si se acumula demasiado contenido en una sola escena) a través de código de implementación.

Las escenas se tratan de forma individual y, como es lógico, no se puede interactuar en varias a la vez, sino de forma individual en cada una de ellas. Éstas están implementadas y construidas mediante el uso de la jerarquía, concepto de suma importancia en la elaboración de juegos en Unity3D.

2.3. GAMEOBJECTS

Se denomina *GameObject* u "objeto del juego" a cualquier objeto activo de la escena en la que se encuentre el juego en cada momento. Ciertos *Assets* extraídos del *Project Panel* pueden llegar a realizar la función de un *GameObject* dependiendo de la escena en la que se encuentre el programa, tales como pueda ser un *prefab* (modelo prefabricado).

Otros objetos como los generadores de partículas pueden situarse en la escena usando el botón *Create* en la jerarquía o mediante la pestaña *GameObject* en la parte superior del menú, tal como se muestra en el Gráfico 5.

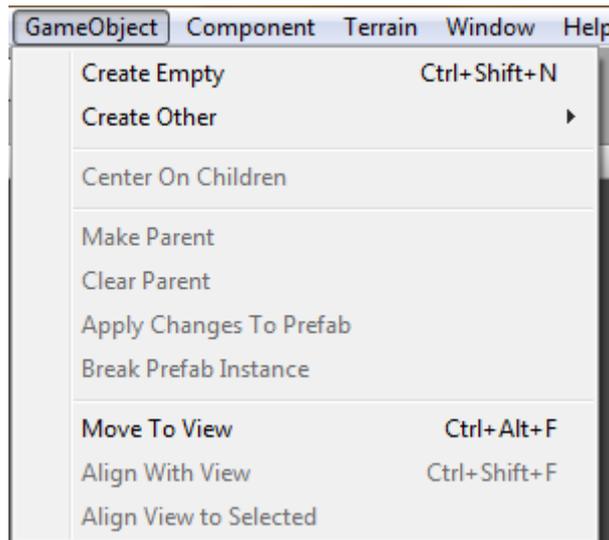


Gráfico 5: Captura del programa Unity3D con la pestaña GameObject desplegada.

Todos los *GameObject* deben contener al menos un componente para empezar a trabajar con ellos. Este componente es el de transformación (*Transform component*). Con transformación se contempla la rotación, posición y escalado en los ejes X-Y-Z del eje de coordenadas. A su vez, al *GameObject* se le pueden asignar una serie de comandos que puedan transformar tanto su rotación como su posición o su escala.

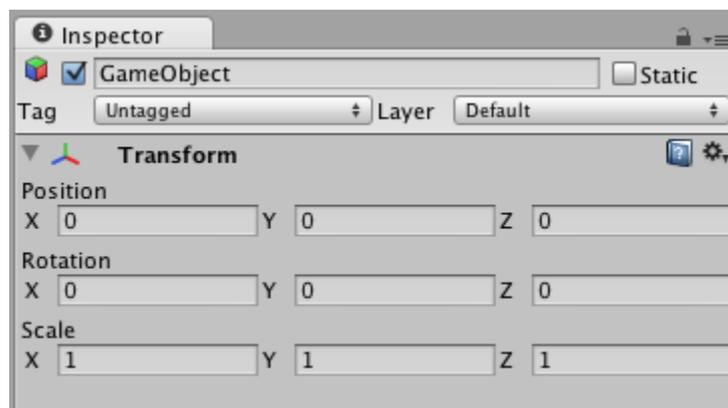


Gráfico 6: Captura del programa Unity3D del Inspector con las opciones de rotación, posición y escala del GameObject

Los *GameObjects* también son ordenados de forma jerárquica, y se puede dar el caso de la relación *parent-child*, en la que un *GameObject* pueda gobernar sobre otro y que éste segundo sea dependiente del primero.

2.4. COMPONENTS

Los *Components* o componentes pueden aparecer de diversas formas. Pueden utilizarse para la implementación de comportamientos, definir la apariencia e influir en otros aspectos de la función de un objeto en el juego. Al asociar los componentes a un objeto, se pueden aplicar directamente nuevas piezas del motor de juego a su objeto. Los componentes comunes de la dirección del juego vienen integrados por Unity, desde componentes más complejos, como el componente *Rigidbody*, hasta los más simples como luces, cámaras, emisores de partículas (*Particle Emitters*)...

De cara a una consolidación de los elementos interactivos será adecuada la implementación de *scripts*. Se pensará en un *script* como algo que extiende o modifica la funcionalidad existente de las disponibles en Unity.

2.5. SCRIPTS

Siendo considerada por Unity como un *component*, los *scripts* o secuencias de comandos son una parte esencial de la producción del juego, merecen ser mencionados como uno de los conceptos clave de la aplicación. Los dos lenguajes principales con los que se implementan los scripts suelen ser Java y C++, a pesar de poder utilizar el lenguaje Boo, de elaboración más compleja.

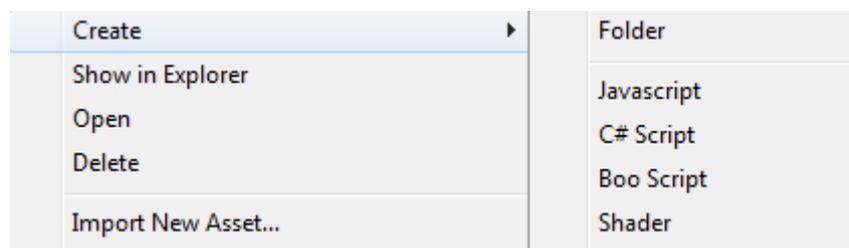


Gráfico 7: Pestaña Assets del programa Unity3D a la hora de crear un script

Unity no requiere tener el conocimiento exacto del funcionamiento del motor o la forma de modificarlo. pero sí que se utilizarán *scripts* en prácticamente cualquier escenario que se desarrolle en el juego. Supone una gran ventaja el escribir *scripts* en Unity, pues tras implementar unos pocos, el resto de secuencias de comandos resultan considerablemente asequibles para el programador. Esto es porque Unity tiene su propio conjunto de instrucciones en una secuencia de comandos llamados *Monobehaviour* al que el usuario puede recurrir en cualquier momento.

Para escribir una secuencia de comandos, se utilizará el editor de *script* del programa Unity3D llamado *Monodevelop*. Se trata de una aplicación asociada a Unity, pero de funcionamiento independiente. Se pondrá en marcha cuando se edite un *script* nuevo o uno ya existente.

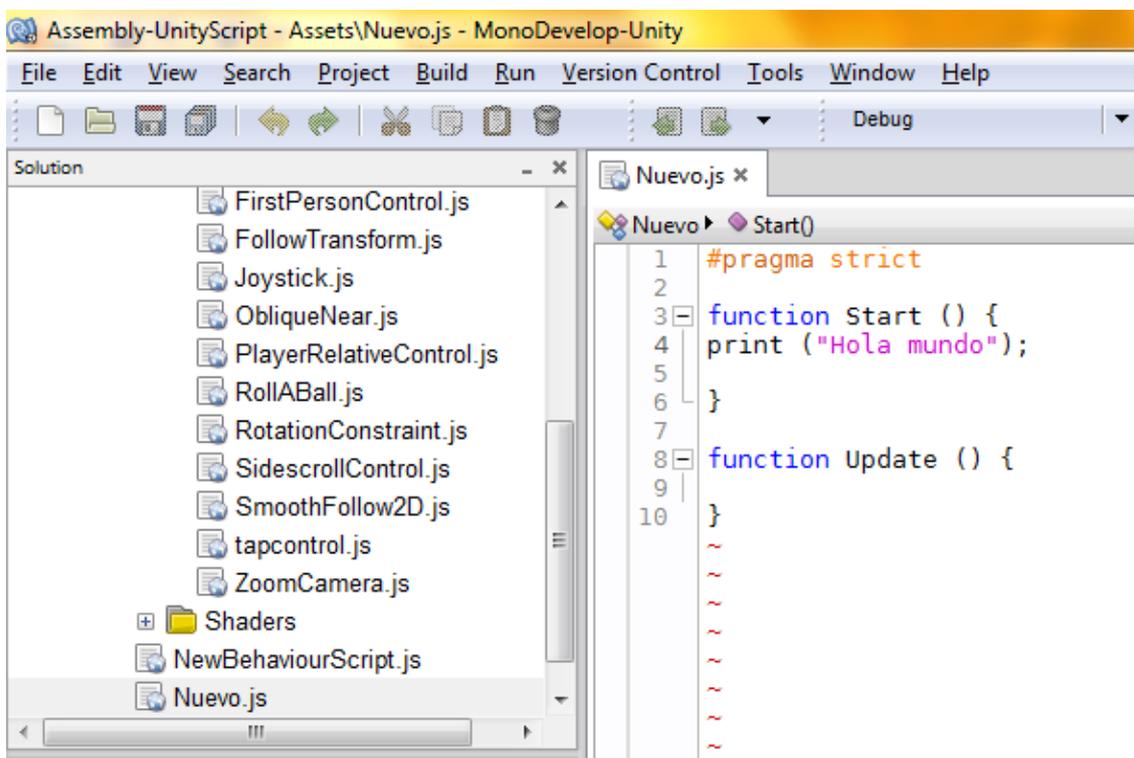


Gráfico 8: Programa Monodevelop abierto con código implementado

Para escribir una secuencia de comandos, se utilizará el editor de *script* del programa Unity3D llamado *Monodevelop*. Se trata de una aplicación asociada a Unity, pero de funcionamiento independiente. Se pondrá en marcha cuando se edite un *script* nuevo o uno ya existente.

Modificar y guardar secuencias de comandos en el editor de scripts actualizará inmediatamente el *script* de Unity tan pronto como se cierre el programa *Monodevelop* y se retorne al trabajo con Unity. *Monodevelop* es el editor más recomendado, ya que ofrece el auto-completado de código a medida que se escribe y es un programa diseñado y actualizado por Unity Technologies.

2.6. PREFABS

El enfoque del desarrollo de Unity gira en torno a la figura de los *GameObject*, pero también existe una forma de almacenar objetos activos para ser reutilizados en diferentes partes del juego que se vaya a desarrollar, pudiendo ser reproducidos o clonados. El ahorro de tiempo y código es notable, pues se pueden crear objetos complejos con varios componentes y que estos *prefab* sean utilizados en más de una escena sin necesidad de volver a implementar el código con el que lo hemos creado inicialmente.

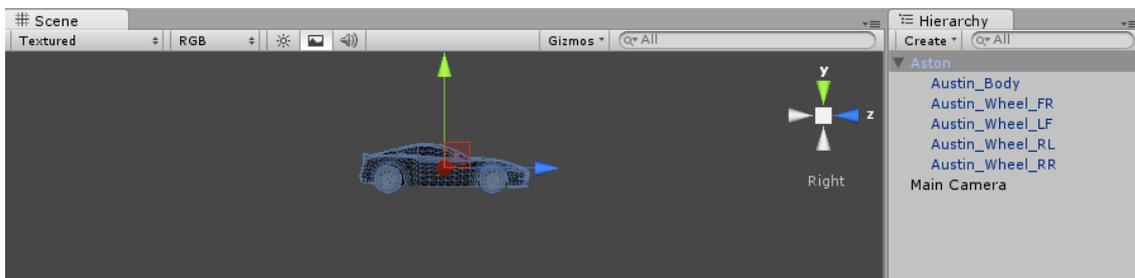


Gráfico 9: Prefab gratuito de un coche Aston Martin

El concepto de *prefab* podría ser comparable al de una caja vacía que ofrece la posibilidad de ser rellena con objetos para formar una plantilla que, posteriormente podría ser clonada o reproducida en otra escena o en la misma escena, tal como se muestra en el Gráfico 10.

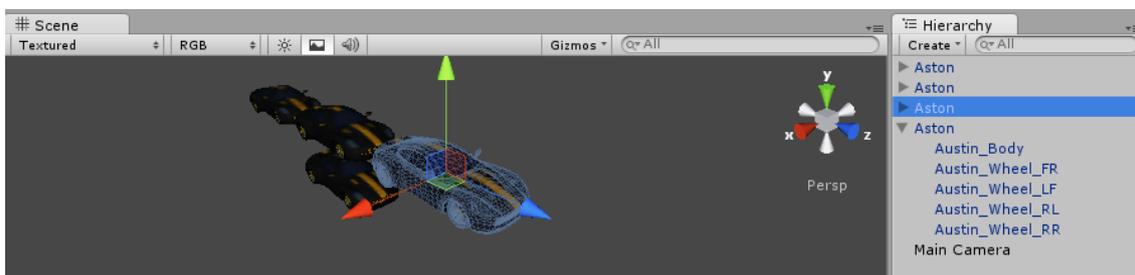


Gráfico 10: Prefab gratuito de un coche Aston Martin clonado en la misma escena

3. UNITY 3D - APLICABILIDAD

Lanzado al mercado en 2009, Unity 3D puede considerarse como una de las aplicaciones punteras de diseño de programas en 3 dimensiones, resulta inevitable la comparación de éste con otros productos similares, como pudiera ser el Adobe Flash.



Gráfico 11: Captura del programa "Cars" desarrollado sobre Unity3D.

Al contrario de lo que se pudiera conocer de Adobe Flash, Unity está enfocado más al entretenimiento del usuario y se encuentra dirigido hacia las aplicaciones en 3 dimensiones.

Posiblemente una de las grandes ventajas que tenga Unity es el amplio abanico de posibles destinatarios que pueda llegar a tener una misma aplicación desarrollada sobre la propia plataforma. Un mismo programa ya desarrollado puede ser configurado para poder usarse en Wii, iPhone, iPad, Android, Xbox, PlayStation o incluso cualquier navegador web si fuera necesario.

3.1. PROGRAMACIÓN EN UNITY 3D

Unity 3D es una aplicación en la que resulta imprescindible la programación con código, a pesar de tratarse de un programa que resulta simple al programador, al poder realizarse mucho trabajo de forma intuitiva y con continua repetición de muchos parámetros.

Además del aspecto visual que favorece a la facilidad de programación en Unity 3D existen una serie de aspectos que facilitan el trabajo al programador. Uno de ellos, y posiblemente el más importante, es el poder trabajar con 3 lenguajes distintos de programación. Los tres lenguajes permitidos son los siguientes:

- C++: Ideal para el mejor rendimiento del juego programado.
- Javascript: Facilita la implementación de scripts rápidos.
- Boo: Implementación del lenguaje Python (poco usado y no recomendado).

3.2. LICENCIAS

Dependiendo de la plataforma y de la empresa a la que esté destinado, en Unity 3D será necesario pagar por una licencia o no. En cuanto a los beneficios de la empresa, Unity sólo solicitará el pago de licencia si los ingresos de ésta superan los 100.000\$ anuales.

Considerando también hacia qué plataforma se dirija el proyecto deberemos pagar más o menos cantidad de dinero para obtener la versión Pro de Unity 3D, con mejores gráficos y una cantidad de assets imprescindibles para hacer un juego que roce la espectacularidad.

Mediante la utilización de la versión Free de Unity 3D, de licencia gratuita, se deberá pulsar el botón "Build&Run" alojado en la pestaña File para darle finalmente la plataforma al juego. Aparecerán entonces las siguientes opciones:

- Web Player
- PC and Mac Standalone
- Flash Player
- iOS
- Android

- Xbox 360
- PS3
- Wii

Se pasa a detallar las características de cada una de las plataformas y el precio de la licencia para la implementación de programas en Unity3D en cada una de ellas.

3.2.1. LICENCIAS GRATUITAS

Sólo existen dos formas de trabajar con Unity 3D e implementar programas de forma gratuita y legal. Existen 2 versiones de Unity 3D (*Unity 3D Free* y *Unity 3D Pro*), siendo la versión *Unity 3D Free* la versión gratuita. En esta versión, se podrá implementar los juegos para su ejecución en Web Player y en PC o MAC.

3.2.1.1. WEB PLAYER

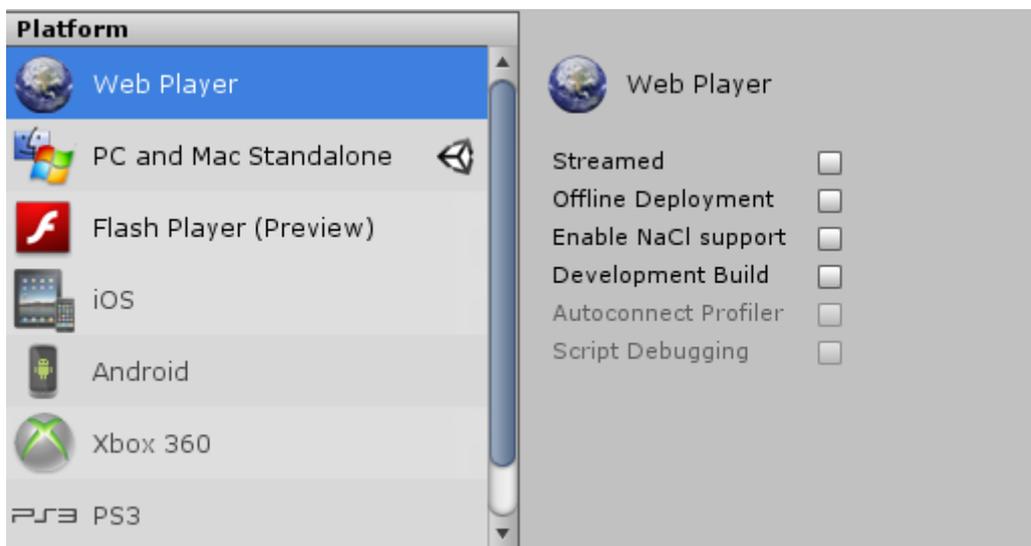


Gráfico 12: Captura del programa Unity3D al convertir el programa a Web Player.

El desarrollo en páginas web resulta gratuito en cuanto a licencias se refiere. Si utilizamos la versión Free, no tendremos que pagar nada a Unity para desarrollar una aplicación y alojarla en una página web de la que tengamos el control. Se deberán configurar también aspectos como si deseamos habilitar un "streaming" o si queremos que se pueda jugar también en modo offline.

3.2.1.2. PC Y MAC

También resulta gratuito el desarrollo de aplicaciones para los soportes de PC o MAC. Ello no conlleva gasto alguno y se requiere la especificación, dependiendo de si se va a querer que el juego sea aplicable en Windows, Windows 64-bit o MAC OS X.

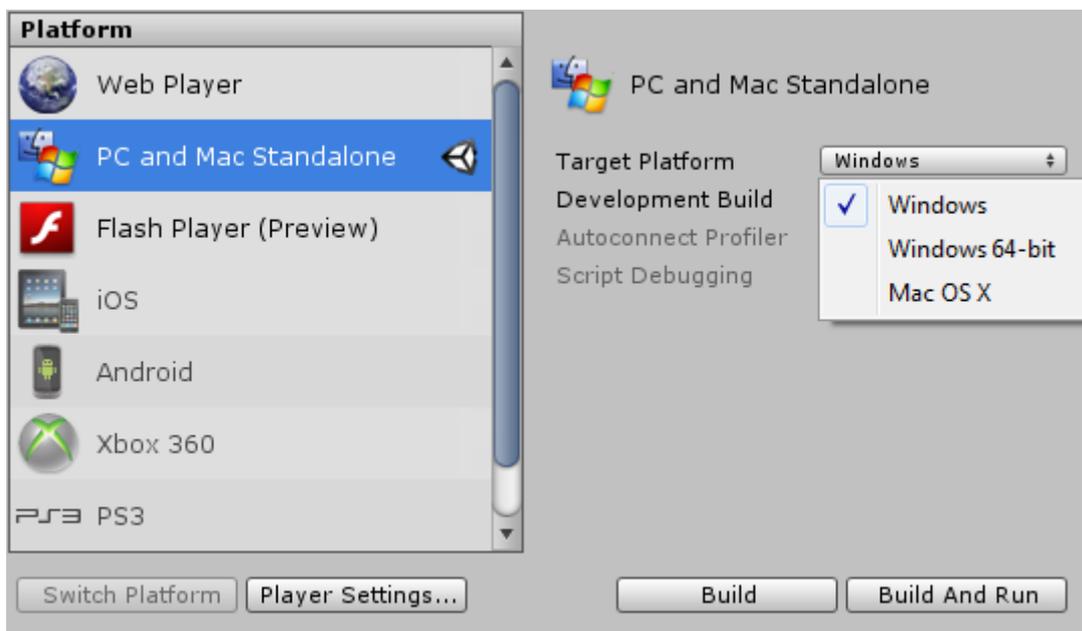


Gráfico 13: Captura del programa Unity3D al convertir el programa a PC and Mac Standalone.

3.2.2. LICENCIAS DE PAGO

Las licencias de pago son algo más complejas. Además de la compra de Unity 3D, se deberán pagar unas licencias extra según se pretenda implementar el juego en una plataforma o en otra.

Unity 3D Pro es la versión profesional de Unity 3D. Incluye una mayor calidad en los gráficos y una amplia gama de *Assets* con los cuales se puede trabajar. Combina la facilidad de uso de Unity 3D en su versión gratuita con el completo conjunto de potentes características de gama alta que requieren los desarrolladores profesionales de aplicaciones en tres dimensiones y los principales estudios. Es capaz de ayudar al desarrollo de juegos de mayor complejidad y visualmente más espectaculares. El precio de Unity 3D Pro es de 1.140€

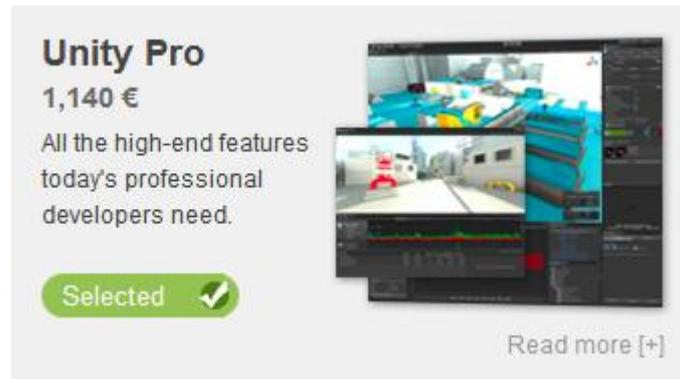


Gráfico 14: Precio de Unity 3D Pro obtenido desde la página web de Unity.

3.2.2.1. FLASH PLAYER

Adobe Flash Player es una aplicación instalada en más de mil millones de ordenadores a nivel mundial. Se trata de una aplicación de reproducción multimedia distribuido por la empresa Adobe Systems. El aprovechamiento de sus herramientas, líderes en el sector de la industria de la edición de archivos multimedia, tales como conductos de luz (*Light probe*), mapas de luz (*Lightmap*) o herramientas de selección de oclusión (*Occlusion Culling tools*) la convierten en una aplicación puntera en su ámbito. Además, cuenta con una gran complementariedad con Unity 3D, pues hace de ésta una perfecta herramienta de desarrollo de los motores impulsores de la creación de contenido 3D que posteriormente se reproducirá en Adobe Flash Player.

Como podremos observar, existen dos tipos de licencias, cuyos precios se detallan en los siguientes gráficos.

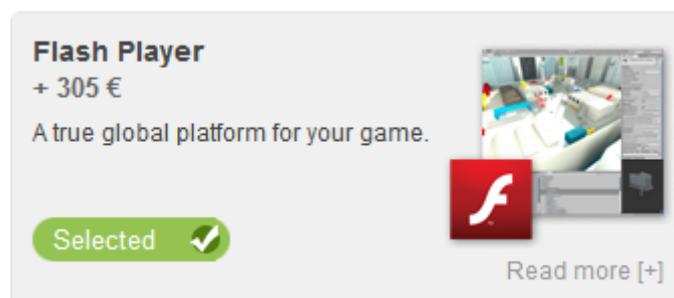


Gráfico 15: Precio de Adobe Flash Player obtenido desde la página web de Unity.

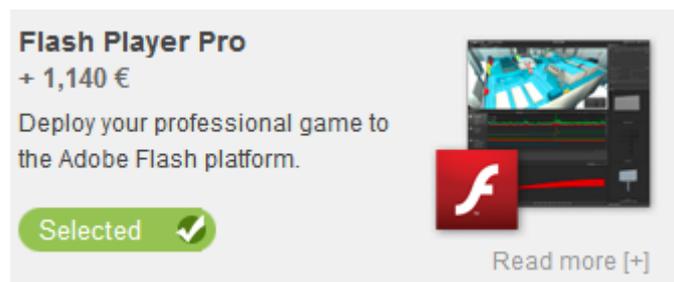


Gráfico 16: Precio de Adobe Flash Player Pro obtenido desde la página web de Unity.

Resulta evidente comentar que la adquisición de ambas licencias resulta incompatible, pues Flash Player Pro incluye lo mismo que Flash Player pero con mejoras. Así, se pasa a detallar cuáles son las mejoras más características que incluye Flash Player Pro con respecto a la otra versión de Flash Player.

Flash Player Pro gestiona mejor el rendimiento de la aplicación. Un hecho que no tiene gran importancia cuando la aplicación es pequeña, pero sí cuanto más grande se vaya haciendo. Maneja unas "mallas" con diferentes niveles de detalle según la proximidad que tenga el objeto a la cámara que se esté utilizando en ese momento. El conjunto de esas mallas se denomina en Unity LODGroups.

Otra de las características más importantes y relevantes que tiene Flash Player Pro respecto a Flash Player es la posibilidad de *streaming*. Es posible importar y reproducir videos desde la red. También pueden ubicarse sobre cualquier superficie 3D o mostrarlos en 2D en la parte superior de la pantalla del juego. Esto puede resultar útil para crear diversos efectos en el cielo de la pantalla.

Por supuesto, la velocidad de *streaming* para juegos *Multiplayer* es mucho mayor, pudiendo generarse texturas, modelos animados, escenas completas o parciales, misiones nuevas y ampliaciones y los propios personajes generados por los usuarios en un tiempo mucho menor que el de Flash Player.

3.2.2.2. iOS

iOS, anteriormente conocido como iPhone OS, es un sistema operativo destinado al sector de la telefonía móvil y desarrollado por Apple. Aunque la idea original era desarrollarlo únicamente para iPhone, este sistema ha sido utilizado para iPad y iPod, además de para consolas como Nintendo Wii o PS3. Miles de juegos para iPhone, iPod y iPad se publican y viven en la App Store, con lo que resulta de gran utilidad que Unity 3D pueda trabajar también con esta plataforma de cara a una correcta expansión del producto que se realice.

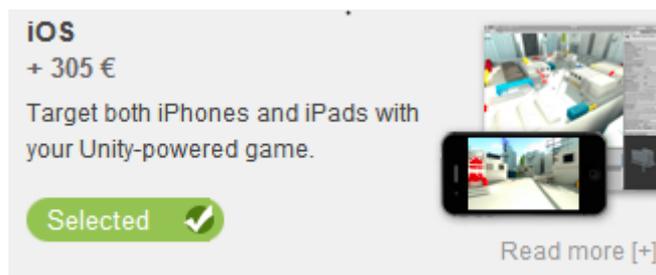


Gráfico 17: Precio de licencia iOS obtenido desde la página web de Unity.

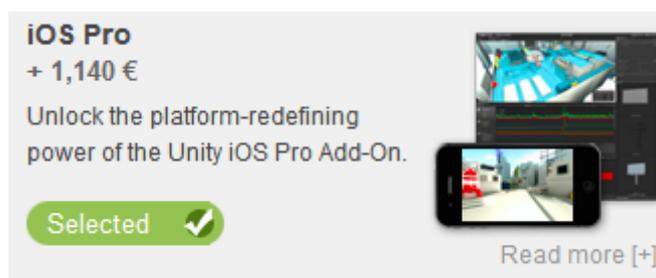


Gráfico 18: Precio de licencia iOS Pro obtenido desde la página web de Unity.

Igual que ocurre con Adobe Flash Player, iOS tiene una versión Standard y una versión Pro, entre las que existen una serie de diferencias, comenzando con la económica, que se ve reflejada en los 2 gráficos anteriores.

iOS Pro gestiona mejor el rendimiento de la aplicación. Igual que ocurre con Adobe Flash Player Pro, maneja unas "mallas" con diferentes niveles de detalle según la proximidad que tenga el objeto a la cámara que se esté utilizando en ese momento, los LODGroups anteriormente mencionados.

Otra de las mejoras que incluye iOS Pro con respecto a la versión Standard es la configuración de audio, muy mejorada, y que proporciona al usuario una sensación de inmersión total en el juego con un audio muy realista.

También se contempla la opción, ya mencionada en Adobe Flash Player Pro, de *streaming*, tanto sobre figuras en 2D como sobre figuras en tres dimensiones. Se tiene la posibilidad de difusión de video por Internet, lo que proporciona muchas mejoras al producto.

3.2.2.3. ANDROID

Android es un sistema operativo móvil basado en Linux, enfocado principalmente a aplicaciones móviles como teléfonos inteligentes, tabletas u otros dispositivos. Está desarrollado por la Open Handset Alliance, empresa liderada por Google. Tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. Se estima que se han sobrepasado ya las 600.000 aplicaciones disponibles en la tienda Google Play, de las cuales 400.000 son gratuitas. Así, Android ha demostrado ser la plataforma móvil de más rápido crecimiento y con más evoluciones. Unity 3D permite desarrollar aplicaciones para los más de 30 millones de dispositivos Android que hay.

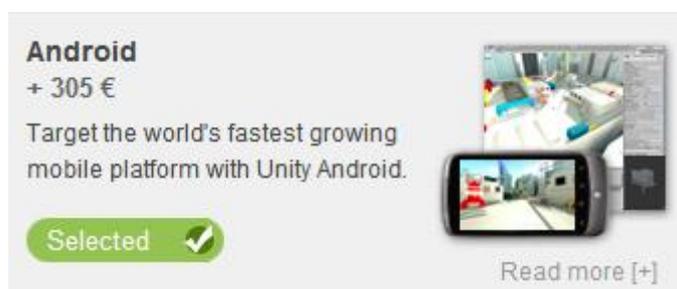


Gráfico 19: Precio de licencia Android obtenido desde la página web de Unity.

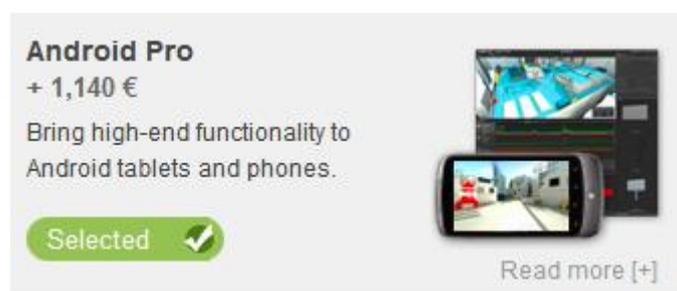


Gráfico 20: Precio de licencia Android Pro obtenido desde la página web de Unity.

Unity, al igual que en los casos anteriores, también oferta una licencia Android Pro y una licencia Android Standard. Las mejoras de la licencia Pro con respecto a la licencia Standard son muy similares a las mejoras que ofertan las licencias Pro de Adobe Flash Player o de iOS.

También Android Pro gestiona mejor el rendimiento de la aplicación. Los LODGroups agrupan en mallas los grupos de objetos según su cercanía y le dan un nitidez mayor o peor según se requiera. Ello mejora considerablemente la velocidad a la que trabajará la aplicación que se desarrolle.

Igual que ocurre en iOS Pro y en Adobe Flash Player Pro, se proporciona la opción de poder realizar *streaming*. Tanto reproducir video hacia el exterior como la recepción de video y la reproducción en el propio programa serán posibles con la licencia de Android Pro.

4. DESCRIPCIÓN DE LA APLICACIÓN IMPLEMENTADA

4.1. INTRODUCCIÓN A LA APLICACIÓN

En primer lugar, se debe tener en cuenta que el proyecto recibe el nombre de "Estudio de Viabilidad del Entorno de Programación Unity3D para la Rehabilitación Cognitiva contra el Alzheimer", con lo que la aplicación no trata de ser la más apropiada ni la más evolucionada para el entrenamiento de la memoria para enfermos de Alzheimer, sino de estudiar la viabilidad del entorno de programación Unity3D de cara al entrenamiento de la memoria en enfermos de Alzheimer. La aplicación que se muestra puede mejorarse según se requiera dependiendo de su funcionalidad final.

La aplicación que se ha programado consta de 3 partes destacadas. La primera parte que se procede a detallar es el Menú Principal. A continuación se hablará sobre un juego llamado "¿Qué soy?", que consta de preguntas y respuestas. Por último se explicará el funcionamiento y la elaboración de la parte de la "Aventura Gráfica".

De cada una de estas 3 fases del proyecto se van a detallar la elaboración y programación de la misma, el funcionamiento y la aplicabilidad al estudio de la enfermedad de Alzheimer (este último apartado se obviará en el menú principal).

4.2. MENÚ PRINCIPAL

4.2.1. INTRODUCCIÓN

El menú principal consta de dos *Levels* o niveles. Uno de los dos niveles será la primera pantalla que se vea cuando se arranque el programa y el siguiente será el que se ve cuando se opte por JUGAR en el menú anterior.

El primer menú consta de 2 botones, uno para JUGAR y el otro para SALIR DEL JUEGO. El menú secundario, al que se accederá si se decide JUGAR, consta de 3 botones. Los botones son los de AVENTURA, para acceder a la aventura gráfica, de QUÉ SOY, para acceder al juego de preguntas y respuestas, y de VOLVER A MENÚ para regresar al menú principal que ya se ha mencionado.

4.2.2. PROGRAMACIÓN Y DESARROLLO

A pesar de que Unity3D resulta una plataforma de trabajo muy intuitiva y apenas requiere programación, hay algunos momentos en los que resulta inevitable. En estos casos, de los 3 lenguajes de programación disponibles, se ha optado por utilizar Java en la programación de los *scripts* que se han implementado.

El ya conocido menú principal recibe el nombre de *menu* para ser identificado como nivel en Unity3D, mientras que el menú secundario, que nos dará paso a los distintos juegos y a la opción de volver al menú anterior recibe el nombre de *menu1*.

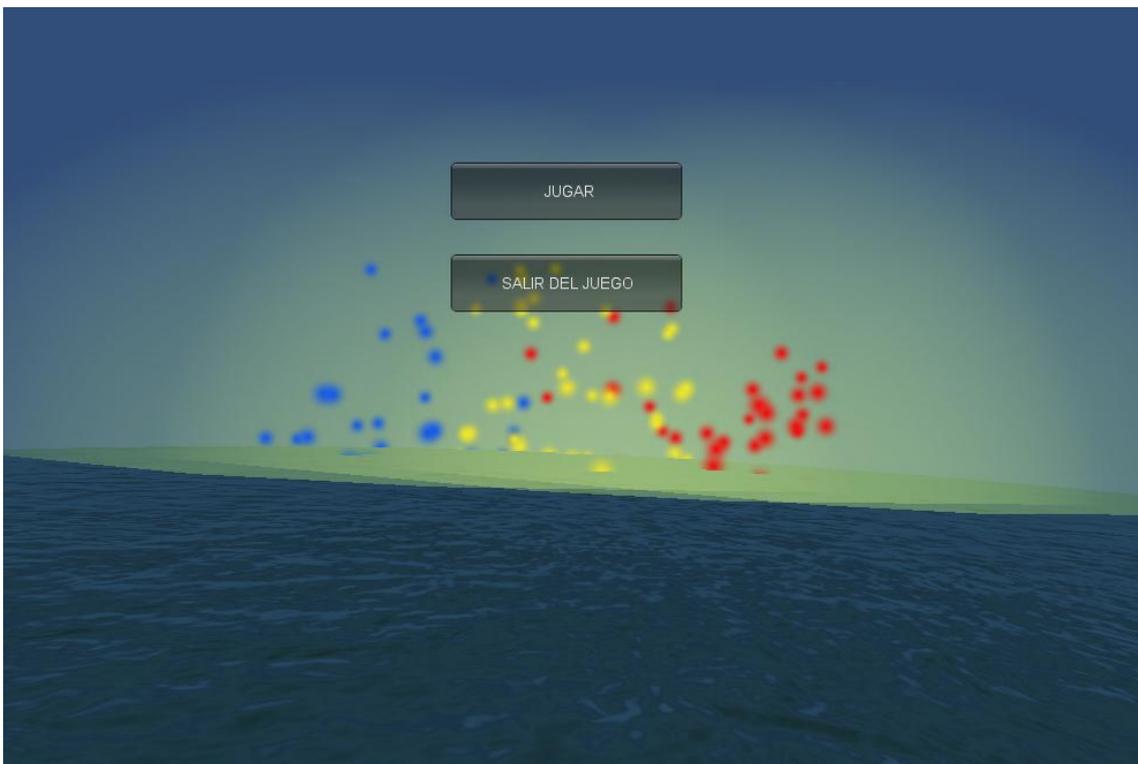


Gráfico 21: Level "menu" que corresponde al menú principal.

La música que corresponde al menú principal y al menú secundario es la misma. Se ha decidido optar por música libre de derechos, descargada gratuitamente desde Internet. En este caso corresponde con "King of the Word" de Sean Fournier. Esta música sonará cada vez que se inicie uno de los 2 menús y no concluirá hasta que no se salga de ellos o se acceda a otra pantalla.

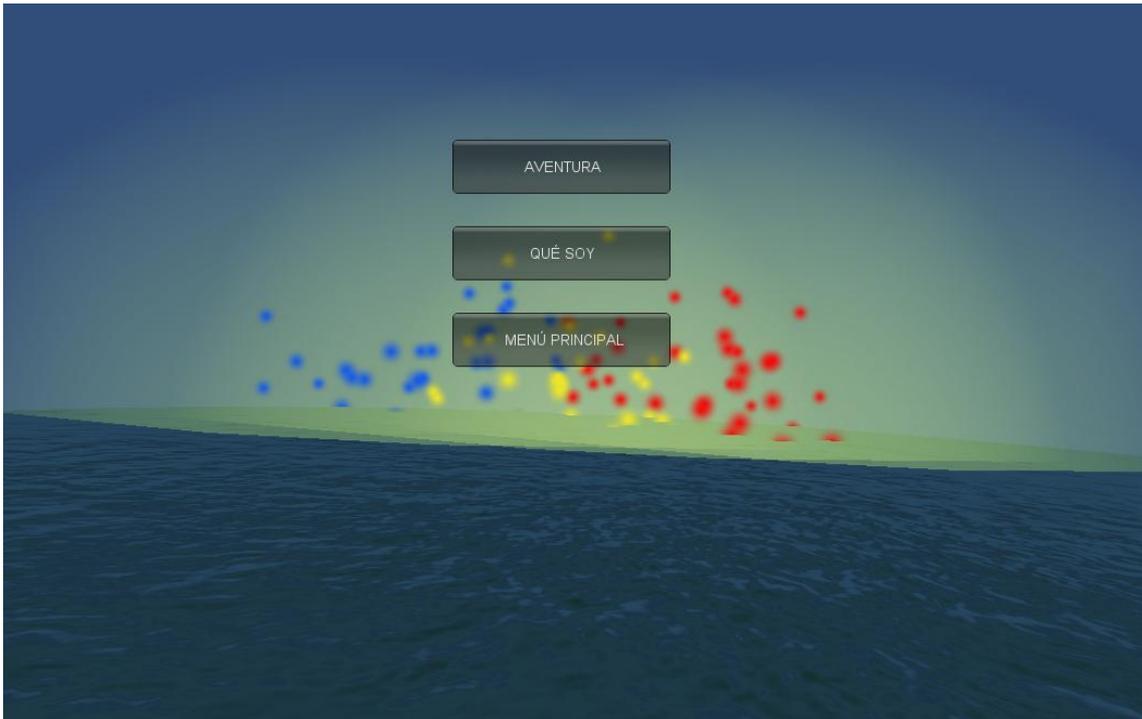


Gráfico 22: Level "menu" que corresponde al menú secundario.

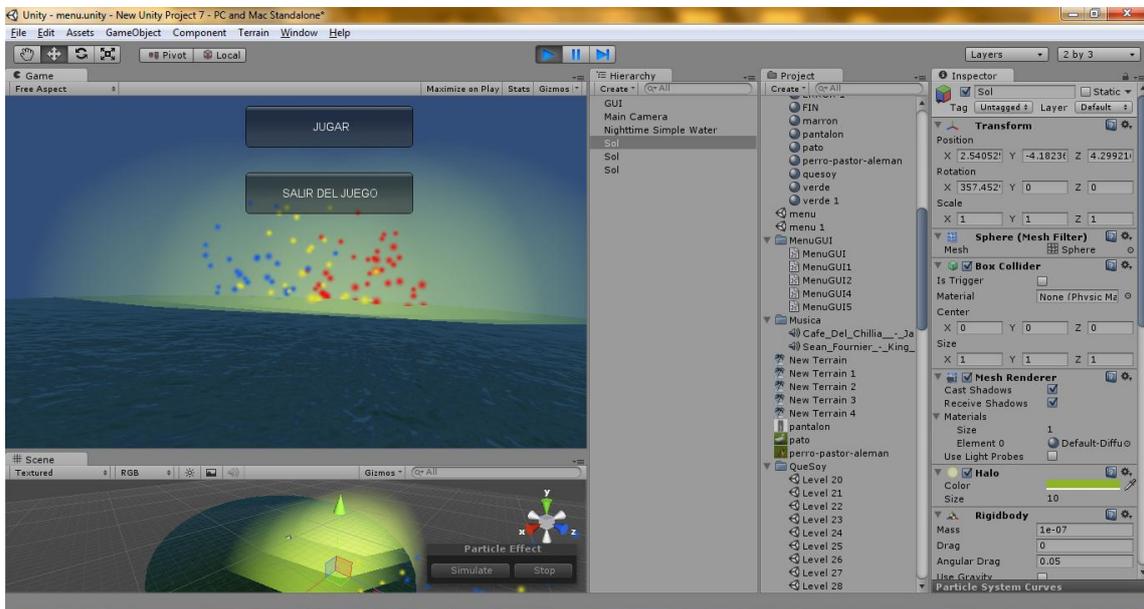


Gráfico 23: Pantalla del programa Unity3D con el menú principal activo

Tal y como se puede ver en el Gráfico 21, la parte gráfica del *menu* consta de varios objetos que se han añadido sin necesidad de programar. Todo el conjunto de objetos que se han añadido se pueden ver en Hierarchy (la columna a la derecha de las pantallas del juego).

Se ha llamado "Sol" a los 3 generadores de partículas que luego provocan un efecto de fondo parecido a globos saliendo de la nada. Además cada "Sol" genera un halo de luz que provoca un efecto similar al del atardecer. Los 3 elementos "Sol" están ocultos bajo el mar.

Se ha llamado "Nighttime Simple Water" a la capa de agua con marea incorporada que se obtuvo como software libre de Internet ya programada. Esta base es, tal y como se puede ver en la ventana escena (abajo izquierda), con lo que se considera un objeto más y no un terreno infinito.

Posiblemente el elemento más completo es el GameObject "GUI", que consta de un script en el que se generan los 2 botones que se pueden ver en el Gráfico 24. Se ha programado en Java la existencia de estos botones.

```
var x: float =200;
var y: float =150;
var Ancho : float=100;
var Alto: float=50;
var AjusteX: float= 60;
var AjusteY: float= 60;

function OnGUI (){
    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY,Ancho,Alto),"JUGAR")){
        Application.LoadLevel("menu 1");
    }

    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY+80,Ancho,Alto),"SALIR DEL JUEGO")){
        Application.Quit();
    }
}
```

Gráfico 24: Programación del MenuGUI para el menú principal

En el menú secundario reflejado en el Gráfico 23 se ha trabajado sobre las mismas bases. En este caso disponemos de 3 botones configurados en un archivo Java denominado MenuGUI4. El archivo es muy similar a MenuGUI con varias novedades. Estos cambios se pueden ver en el Gráfico 25.

```
var x: float =200;
var y: float =150;
var Ancho : float=100;
var Alto: float=50;
var AjusteX: float= 60;
var AjusteY: float= 60;

function OnGUI (){
    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY,Ancho,Alto),"AVENTURA")){
        Application.LoadLevel("Level 1");
    }
    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY+80,Ancho,Alto),"QUÉ SOY")){
        Application.LoadLevel("Level 20");
    }
    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY+160,Ancho,Alto),"MENÚ PRINCIPAL")){
        Application.LoadLevel("menu");
    }
}
```

Gráfico 25: Programación del MenuGUI4 para el menú secundario

Se puede observar cómo se cargará el *Level 1* si escogemos *AVENTURA*, el *Level 20* si escogemos *QUÉ SOY* y que volveremos al menú principal (*menu*) si presionamos el botón que se encuentra en la parte inferior. El resto del código es simplemente para ubicarlo en la pantalla según conviene, obteniendo el resultado del Gráfico 25.

Igual que ocurre en el primer documento Java, las variables *x*, *y*, *AjusteX*, *AjusteY*, *Alto* y *Ancho* determinan la posición y el tamaño de los botones teniendo en cuenta la anchura (*Screen.width*) y altura de la pantalla (*Screen.height*).

4.2.3. FUNCIONAMIENTO

El menú se divide en dos menús, uno principal, que será el que se vea cuando se arranque la aplicación, y otro secundario, al que se accederá pulsando el botón JUGAR. A continuación, se detalla el funcionamiento de los dos menús. Se ha pretendido ser lo más simple posible, pues estamos tratando con pacientes de avanzada edad que sabrán, o no, trabajar sin ayuda con un ordenador dependiendo de sus conocimientos, habilidades y estado mental en cada momento.

El menú principal tiene 2 botones, uno para JUGAR y otro para SALIR. Si presionamos sobre el botón JUGAR avanzaremos al menú secundario. Si presionamos sobre el botón SALIR abandonaremos la aplicación.

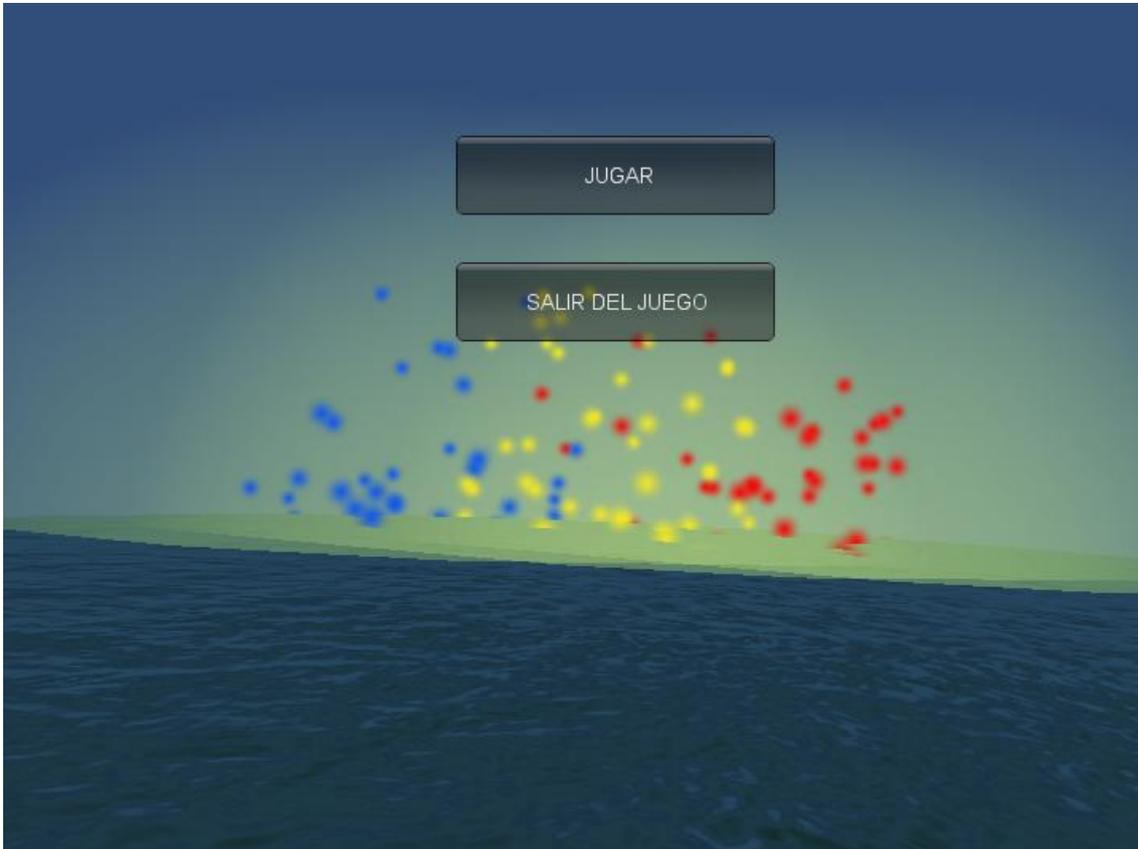


Gráfico 26: Pantalla del menú principal en pleno funcionamiento.

El menú secundario tiene 3 botones, uno para AVENTURA, otro para QUÉ SOY y un tercero para MENÚ PRINCIPAL. Presionando sobre el botón AVENTURA se accederá a la aventura gráfica. Presionando sobre el botón QUÉ SOY se accederá al juego de preguntas y respuestas con 4 opciones. Presionando sobre el botón MENÚ PRINCIPAL se volverá al menú anterior. Este menú no contempla la opción de salir directamente de la aplicación, pues no se ha creído necesario la implementación de ese botón al ser de rápido acceso volviendo al menú principal.

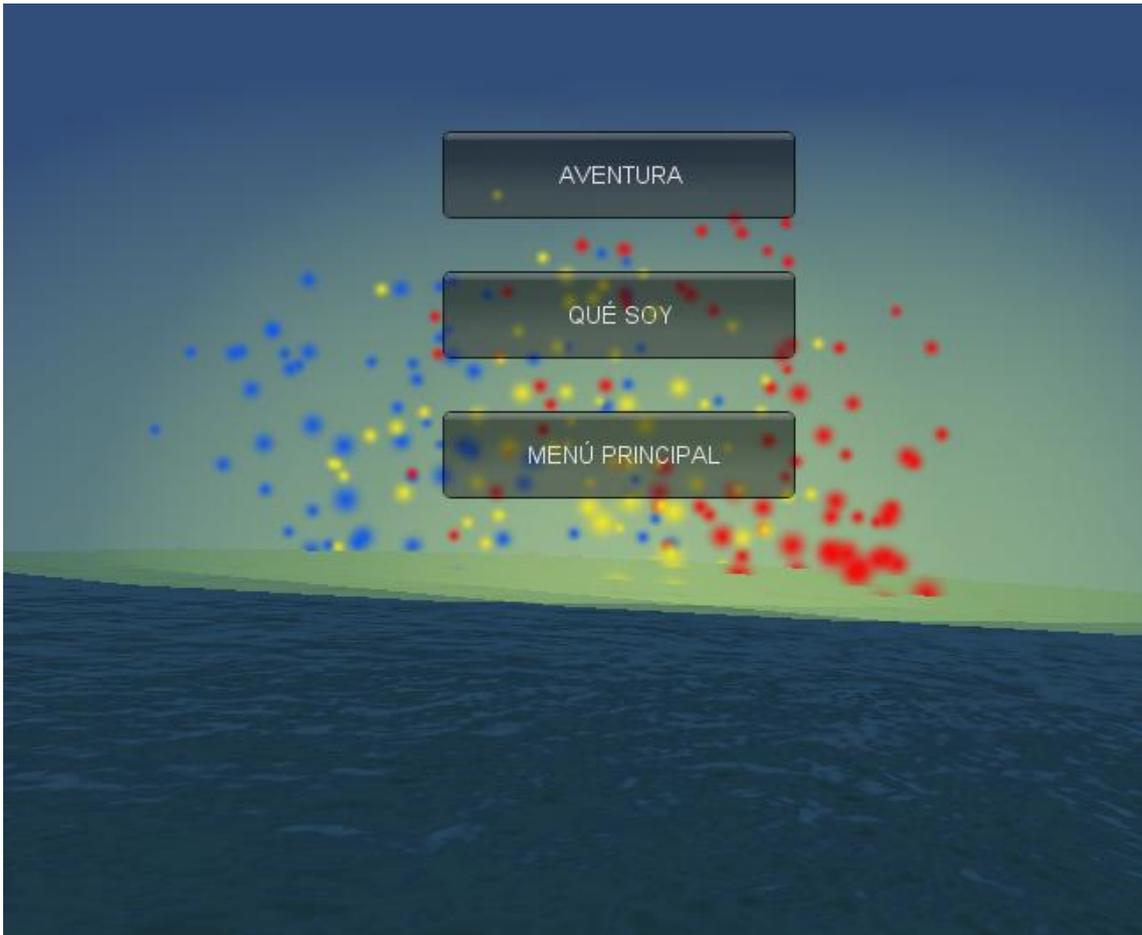


Gráfico 27: Pantalla del menú secundario en pleno funcionamiento.

4.3. QUÉ SOY

4.3.1. INTRODUCCIÓN

La primera aplicación que se va a explicar es la denominada QUÉ SOY, un programa de preguntas y respuestas que goza de la simplicidad requerida tanto en los temas como en su funcionamiento. El juego consta de 8 niveles de similar dificultad y el paciente deberá acertar de forma consecutiva y sin errores las preguntas de los 8 niveles para llegar al final de la aplicación.

4.3.2. PROGRAMACIÓN Y DESARROLLO

La primera aplicación que se va a explicar es la denominada QUÉ SOY, un programa de preguntas y respuestas que goza de la simplicidad requerida tanto en los temas como en su funcionamiento. El juego consta de 8 niveles de similar dificultad y el paciente deberá acertar de forma consecutiva y sin errores las preguntas de los 8 niveles para llegar al final de la aplicación.

La aplicación comienza con la pantalla de presentación, que se muestra en el Gráfico 28.

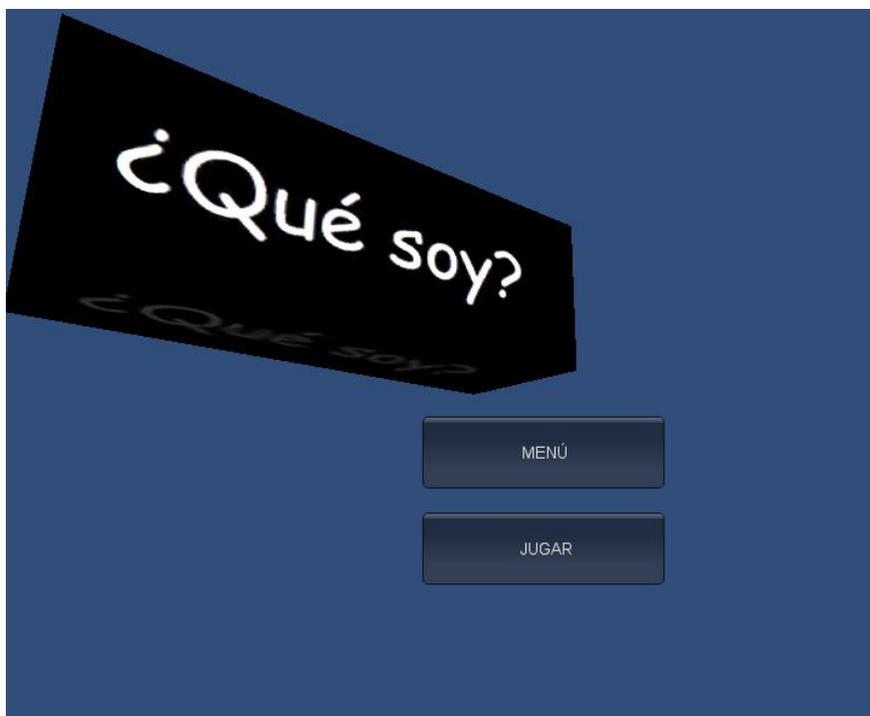


Gráfico 28: Pantalla del Level 20 (inicio de Qué soy) en funcionamiento.

Como se puede ver en el Gráfico 28, la aplicación consta de un paralelepípedo con una textura negra con el texto "Qué soy" impreso en blanco sobre ella. La creación de este paralelepípedo ha sido creando un *Cube* y variando sus medidas, añadiéndole posteriormente esta textura elaborada en Photoshop.

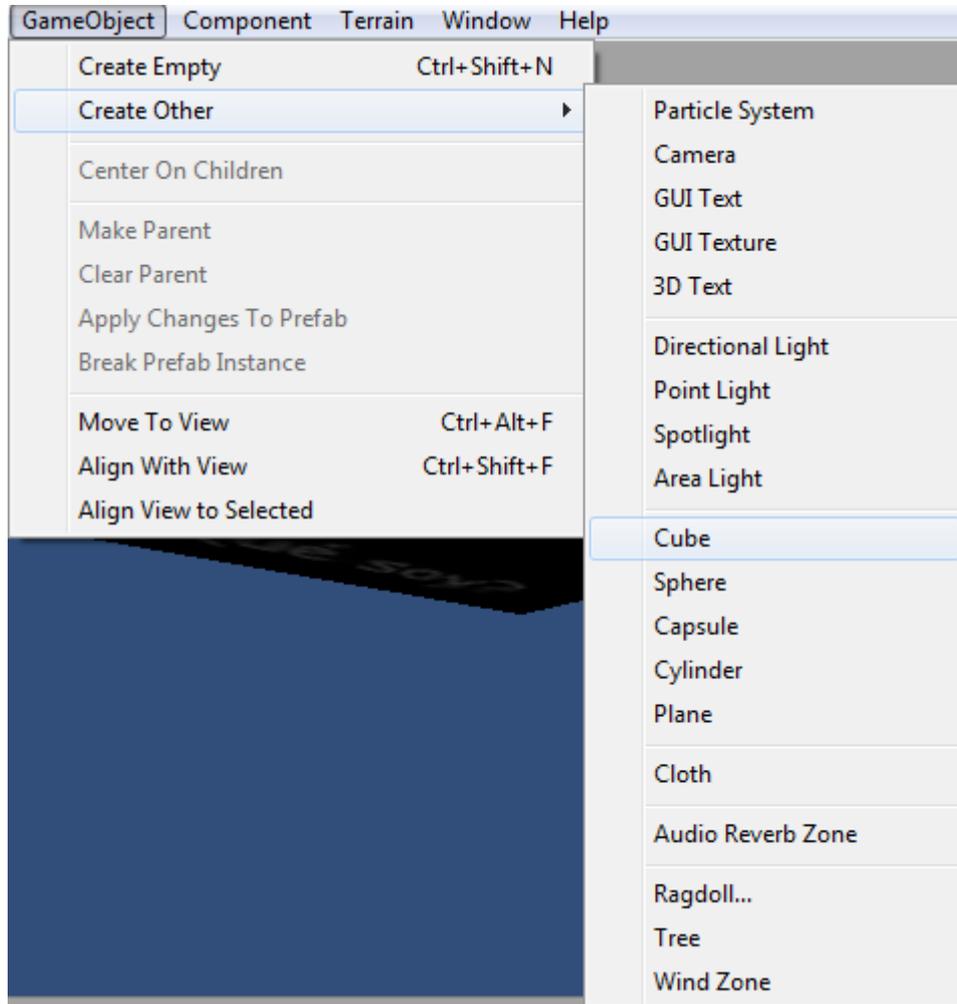


Gráfico 29: Inserción de un Cube

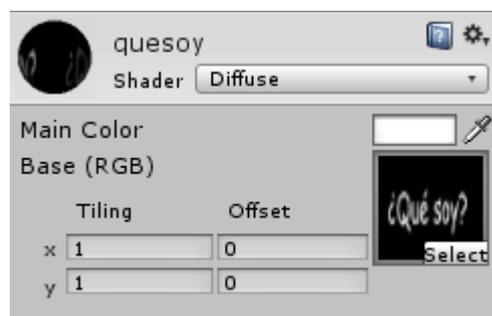


Gráfico 30: Inserción de textura en un Cube

Como ocurre en los menús principal y secundario se ha optado por programar en Java los botones que se ven en la aplicación. El botón MENÚ servirá para volver al menú principal, mientras que al pulsar el botón JUGAR pasaremos a la pantalla del *Level 21*, que corresponde con la primera pregunta del juego "¿Qué soy?".

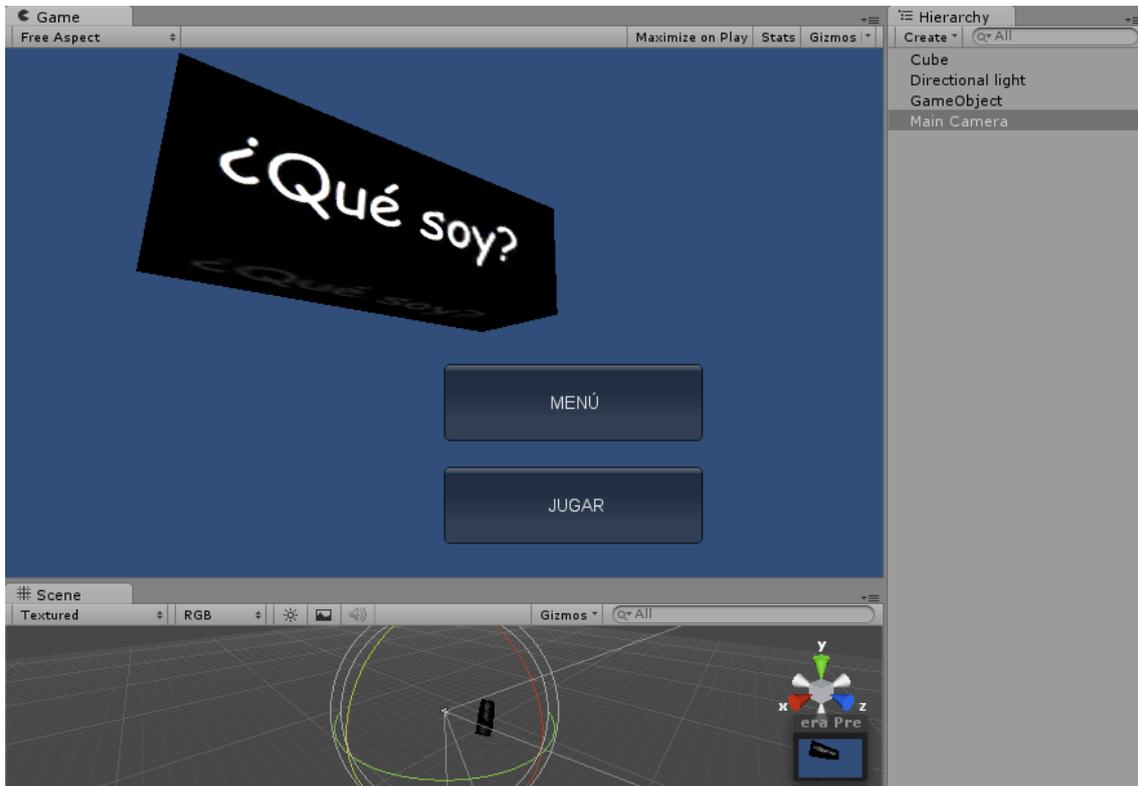


Gráfico 31: Pantalla de Unity3D con el Level 20 activado.

Se puede ver en el Gráfico 31 que en la *Hierarchy* (o jerarquía) tenemos activados los siguientes objetos:

- Cube: El paralelepípedo con la textura "¿Qué soy?".
- Directional Light: Una luz direccional que ayuda a la correcta visualización de la textura.
- GameObject: Contiene la programación en Java de los botones MENU y JUGAR.
- MainCamera: Cámara principal (la dirección de enfoque de la cámara se puede contemplar perfectamente en la pantalla Scene de la parte inferior del Gráfico 31).

El objeto más complejo de esta pantalla resulta ser sin duda el GameObject que contiene los botones MENÚ Y JUGAR. Al texto programado en Java se le ha llamado MenuGUI2 y es el mostrado en el Gráfico 32.

```
var x: float =200;
var y: float =150;
var Ancho : float=100;
var Alto: float=50;
var AjusteX: float= 60;
var AjusteY: float= 60;

function OnGUI (){
    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY,Ancho,Alto),"JUGAR")){
        Application.LoadLevel("Level 21");
    }

    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY-80,Ancho,Alto),"MENÚ")){
        Application.LoadLevel("menu");
    }
}
```

Gráfico 32: Programación en Java de MenuGUI2

Como en los casos anteriores, las variables que se declaran determinan la posición y el tamaño de los botones respecto al punto central de la pantalla. En el caso de pulsar el botón JUGAR la aplicación cargará el *Level 21*, mientras que si se presiona sobre el botón MENÚ la aplicación cargará el *menu* y se regresará al menú principal, la primera pantalla que aparece en el juego.

Los niveles creados con preguntas y respuestas han recibido los nombres de *Level 21*, *Level 22*, *Level 23*, *Level 24*, *Level 25*, *Level 26*, *Level 27* y *Level 28*. Los niveles son muy similares, variando únicamente las preguntas, el archivo a examinar (ya sea cápsula o foto) y las respuestas. En los Gráficos de X a X se pueden ver todos los niveles existentes y a continuación de las fotos se detalla la programación de todos ellos que, como ya se ha mencionado, resulta muy similar.

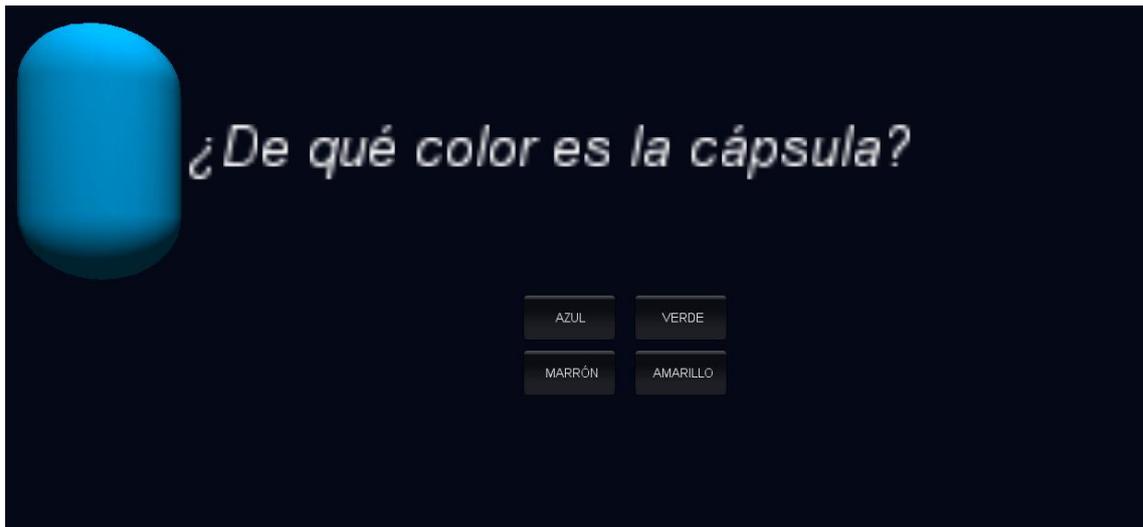


Gráfico 33: Level 21 activo

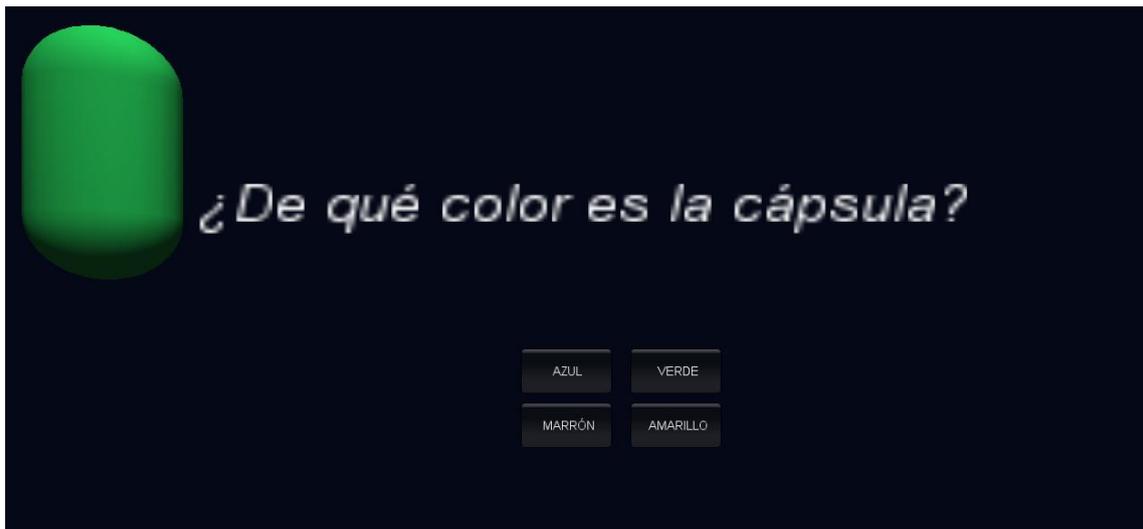


Gráfico 34: Level 22 activo

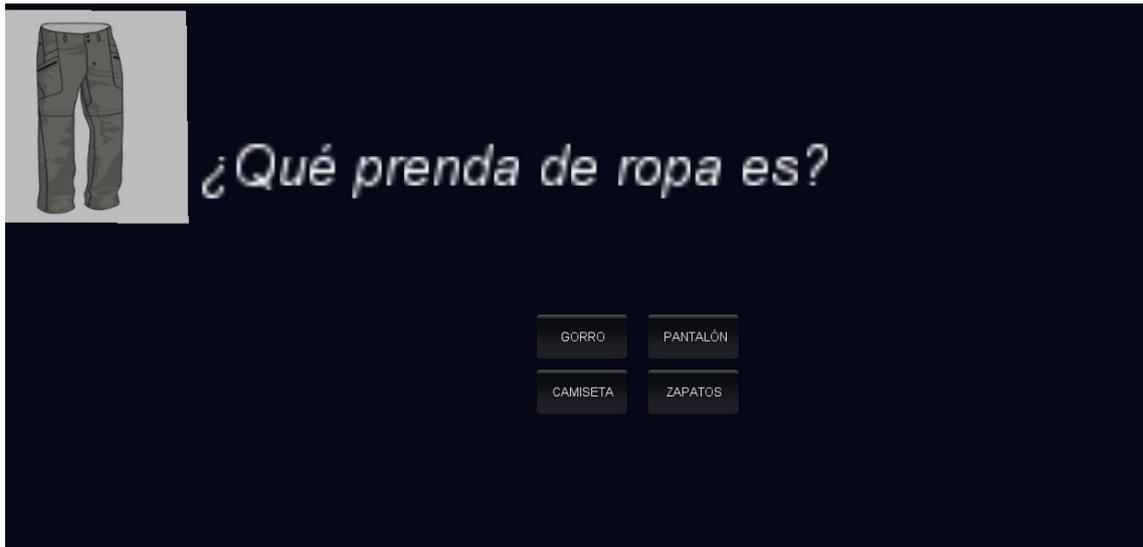


Gráfico 35: Level 23 activo

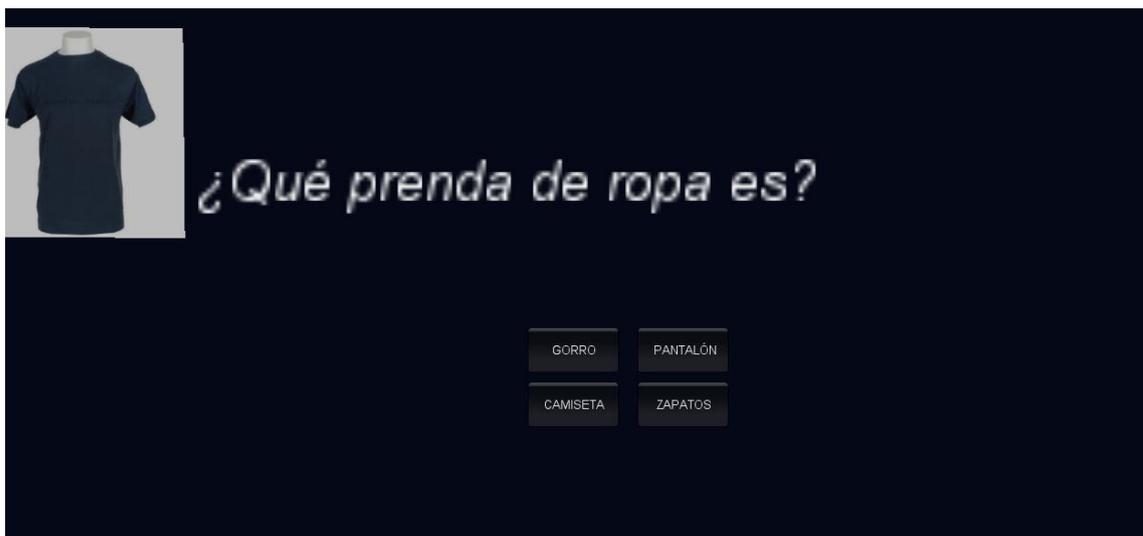


Gráfico 36: Level 24 activo

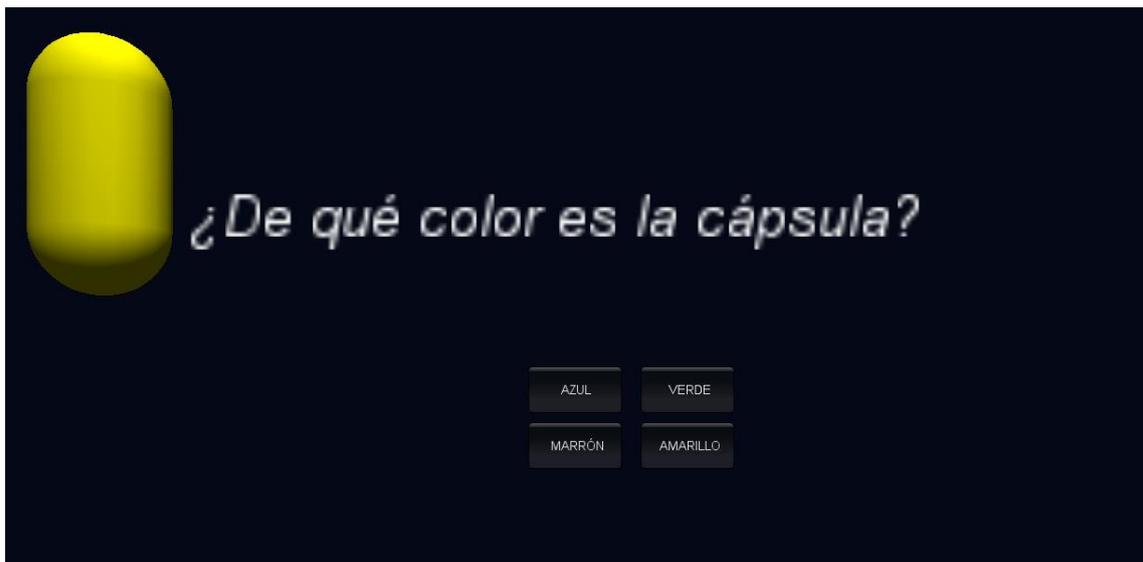


Gráfico 37: Level 25 activo

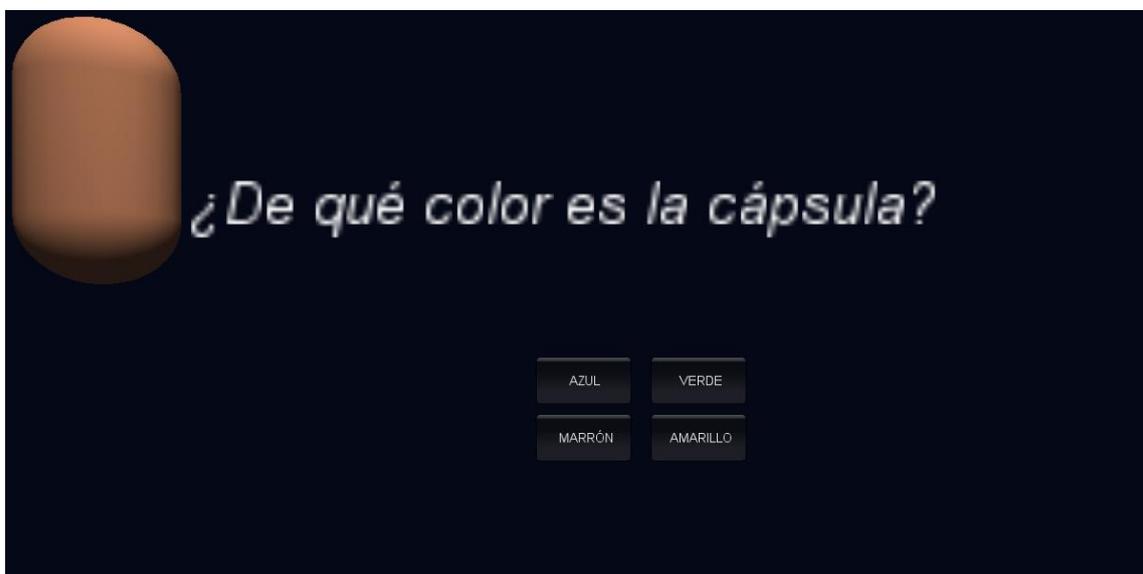


Gráfico 38: Level 26 activo

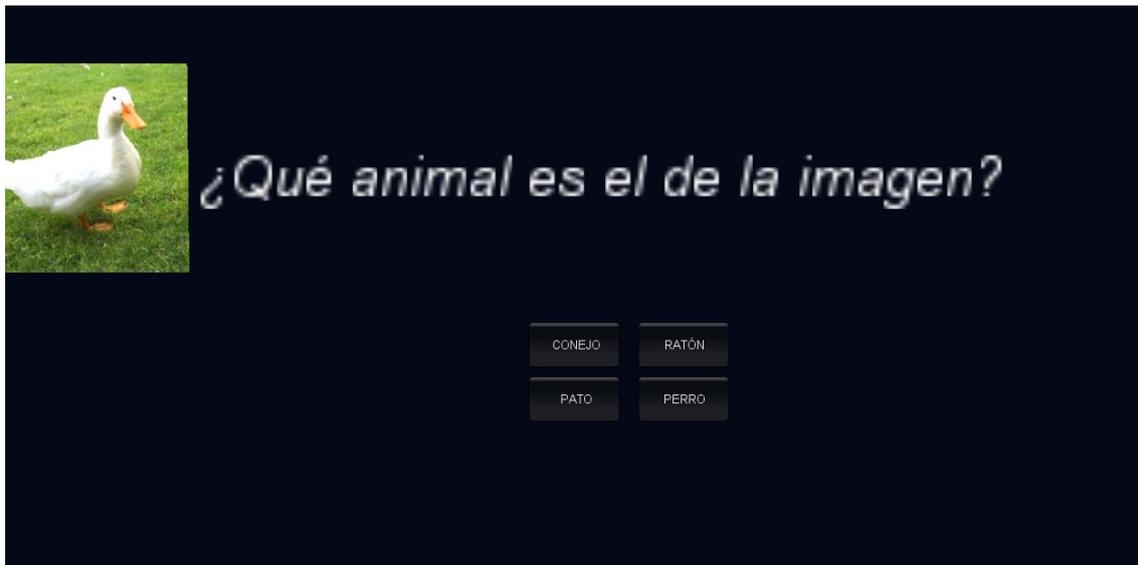


Gráfico 39: Level 27 activo

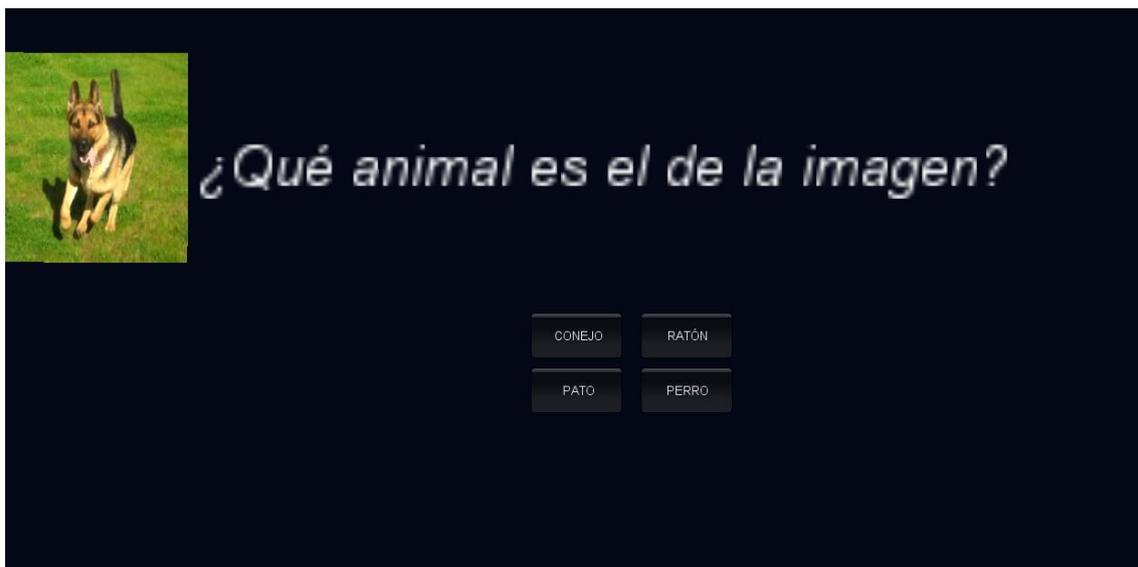


Gráfico 40: Level 28 activo

Los niveles creados con preguntas y respuestas *Level 21*, *Level 22*, *Level 23*, *Level 24*, *Level 25*, *Level 26*, *Level 27* y *Level 28*, como ya se ha mencionado, han sido configurados de forma muy similar. Por ello se ha escogido el *Level 21* como muestra para explicar qué se ha realizado en cada uno de ellos en su programación y desarrollo.

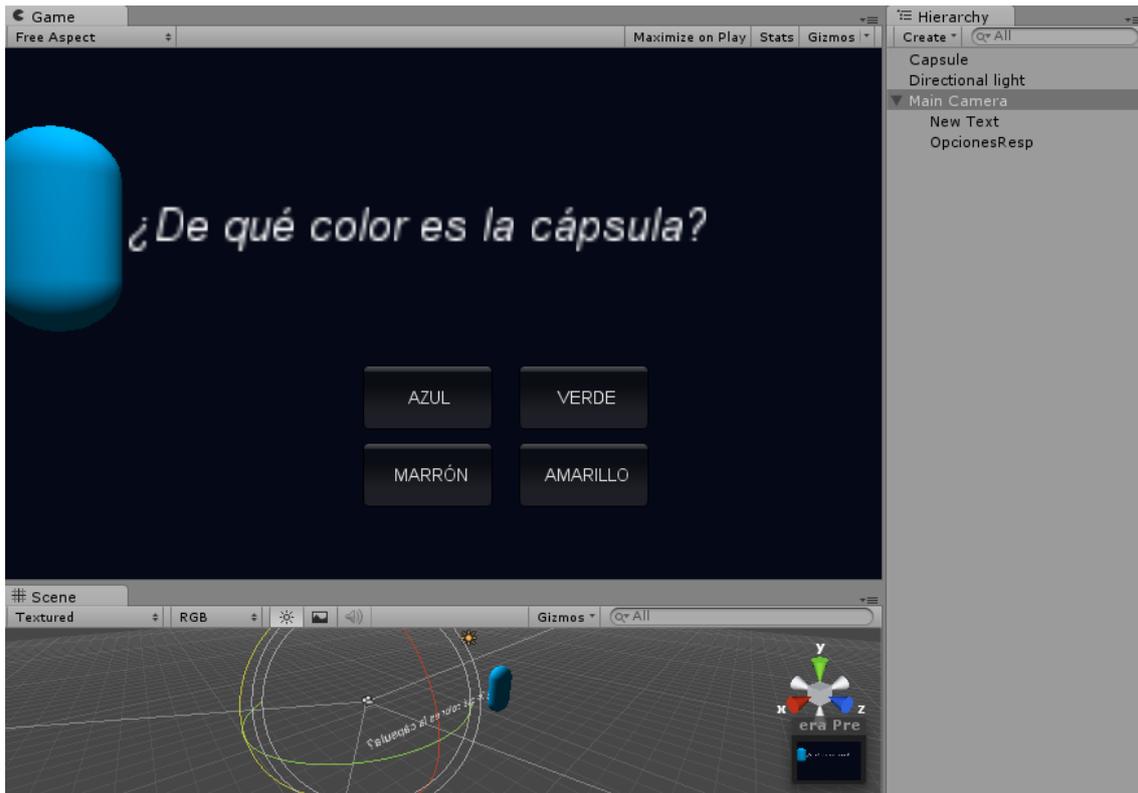


Gráfico 41: Pantalla de Unity3D con el Level 21 activo

En la *Hierarchy* se ha incluido una serie de objetos que se pasan a detallar:

- Capsule: Que corresponde con la cápsula, en este caso de color azul, que está situada en la parte izquierda de la imagen. En el caso de los *Level 23,24,27 y 28* la *Capsule* es un *Plane* o plano inclinado con una textura incorporada que muestra la foto que se debe reconocer.
- Directional Light: Imprescindible la presencia de una luz direccional para poder distinguir el color de la propia cápsula y leer el texto con claridad. Si no se hubiera incluido una luz, la imagen estaría tan oscura que sería irreconocible.
- Main Camera: Es la cámara principal. En la parte inferior del Gráfico 41 se puede ver hacia dónde está dirigida y la amplitud de imagen de la que goza.
- New Text: Es el texto que aparece al lado de la cápsula con la pregunta.
- OpcionesResp: Posiblemente la parte más compleja de estos niveles. Contiene un *script* en lenguaje Java compilado en el programa MonoDevelop denominado *Resp21*.

```
var x: float =200;
var y: float =150;
var Ancho : float=100;
var Alto: float=50;
var AjusteX: float= 60;
var AjusteY: float= 60;

function OnGUI (){
    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY,Ancho,Alto),"MARRÓN")){
        Application.LoadLevel("error");
    }

    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY-60,Ancho,Alto),"AZUL")){
        Application.LoadLevel("acierto1");
    }

    if (GUI.Button(Rect(Screen.width/2-AjusteX+120, Screen.height/2-AjusteY,Ancho,Alto),"AMARILLO")){
        Application.LoadLevel("error");
    }

    if (GUI.Button(Rect(Screen.width/2-AjusteX+120, Screen.height/2-AjusteY-60,Ancho,Alto),"VERDE")){
        Application.LoadLevel("error");
    }
}
```

Gráfico 42: Programación del archivo Resp21

Como ha ocurrido en el resto de casos, las variables declaradas al inicio determinan la posición de los botones con respecto a la posición central de la pantalla. En el *Level 21* en caso de contestar a alguna de las opciones falsas (véase MARRÓN, AMARILLO y VERDE) el programa cargará la pantalla de *error*, mientras que si se pulsa el botón con la opción correcta el programa cargará la pantalla de *acierto 1*. Cabe destacar que en el resto de niveles siempre se cargará la pantalla de *error* en caso de fallo, pero no siempre la de *acierto1* pues, cada nivel cargará, en caso de una respuesta correcta, el nivel *aciertoX* siendo X un número que se irá incrementando de 1 a 8. Esto ocurre porque cada nivel de acierto, aunque parezcan todos iguales, es distinto en cuanto a que sus botones llamados SIGUIENTE NIVEL deben cargar cada vez un nivel distinto, mientras que el nivel *error* siempre redirigirá al inicio de la aplicación.

Los niveles de acierto son *acierto1*, *acierto2*, *acierto3*, *acierto4*, *acierto5*, *acierto6*, *acierto7* y *acierto8*. Son todos muy similares con la salvedad de que *acierto1* da acceso al *Level 22*, *acierto2* da acceso al *Level 23*, *acierto3* al *Level 24* y así

sucesivamente, siendo el *Level 29* el nivel final que se detallará al final de este apartado.

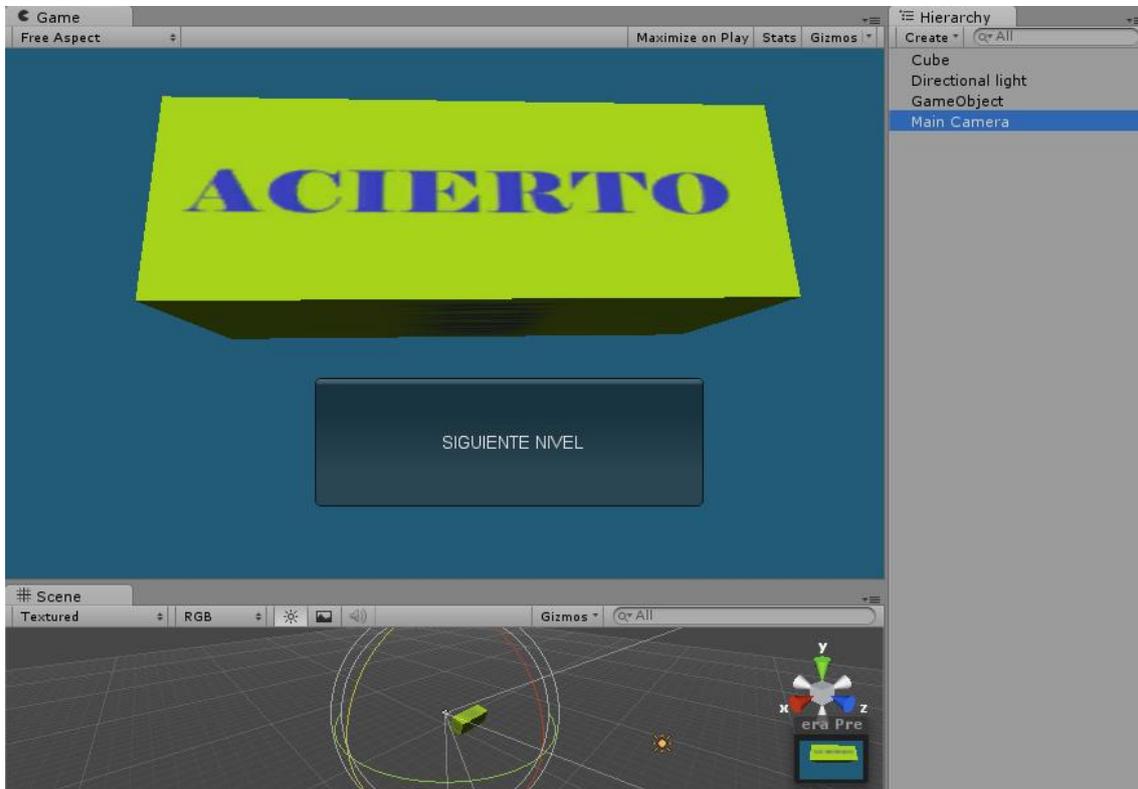


Gráfico 43: Pantalla de Unity3D con el Level *acierto1* activado

El nivel *acierto1* se compone de varios objetos que se proceden a detallar:

- Cube: Paralelepípedo que se encuentra en la parte superior de la pantalla con la textura verde y texto azul con ACIERTO escrito.
- Directional Light: Imprescindible la presencia de una luz direccional para poder distinguir el texto ACIERTO con claridad. En la parte inferior se puede ver el dibujo de un sol. Es ahí donde está ubicada la luz direccional y es por lo que la parte inferior del paralelepípedo está considerablemente más oscurecida.
- Main Camera: Cámara principal que, como hemos hecho en el resto de los gráficos, podemos ver en la parte de la *Scene* inferior del Gráfico 43.
- GameObject: Este *GameObject* contiene un *script* programado en lenguaje Java denominado *Siguiente1*.

El *script Siguiente1* es junto con el resto de *scripts* de tipo *SiguienteX* los más simples que se han programado en esta aplicación, pero resultan imprescindibles para el correcto funcionamiento de la misma.

```
var x: float =200;
var y: float =150;
var Ancho : float=100;
var Alto: float=50;
var AjusteX: float= 60;
var AjusteY: float= 60;

function OnGUI (){
    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY,Ancho,Alto),"SIGUIENTE NIVEL")){
        Application.LoadLevel("Level 22");
    }
}
```

Gráfico 44: Programación del *script Siguiente1*

El *script Siguiente1* sólo programa la existencia de un botón que ponga SIGUIENTE NIVEL y que, al presionarlo, la aplicación cargue el siguiente nivel. En el caso del Gráfico 44 se trata del *script Siguiente1*, con lo que cargará el *Level 22*, igual que en el caso del *script Siguiente2* cargará el *Level 23*.

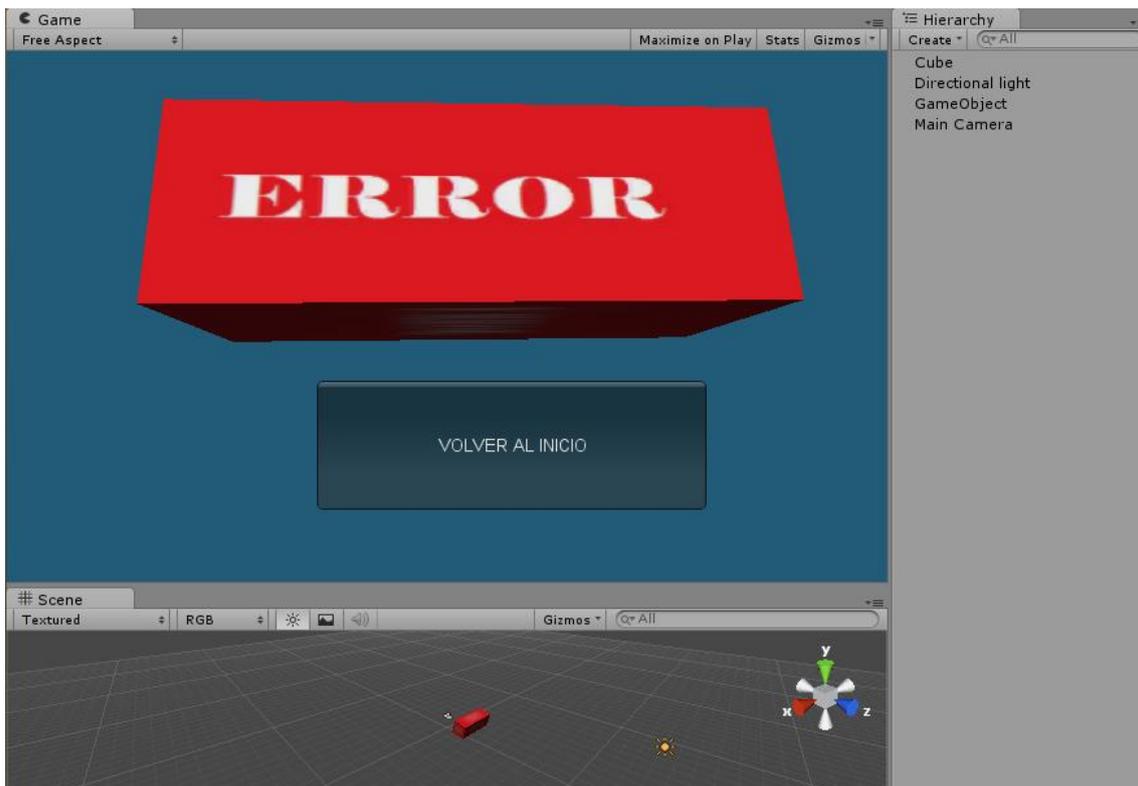


Gráfico 45: Pantalla de Unity3D con el *Level error* activado

Al contrario que con los niveles de tipo *acierto*, niveles de *error* sólo hay uno, que se muestra en el Gráfico 45. Posee las mismas características que el nivel tipo *acierto*, con 2 importantes novedades que se detallan a continuación:

- Cube: La textura es evidentemente distinta. En este caso se opta por una textura roja con letras blancas.
- GameObject: El botón no avanza de nivel sino que vuelve al inicio. Si antes se trataba del *script Siguiente1*, en este caso se trabaja con el *script VolverAMenu*.

```
var x: float =200;
var y: float =150;
var Ancho : float=100;
var Alto: float=50;
var AjusteX: float= 60;
var AjusteY: float= 60;

function OnGUI (){
    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY,Ancho,Alto),"VOLVER AL INICIO")){
        Application.LoadLevel("Level 20");
    }
}
```

Gráfico 46: Programación del script VolverAMenu

De simpleza similar a los *scripts* de tipo Siguiente, el *script VolverAMenu* sólo consta de un botón que dice VOLVER AL INICIO y que carga en la aplicación el *Level 20*, que es el primer nivel de todos, el de la presentación de esta aplicación.

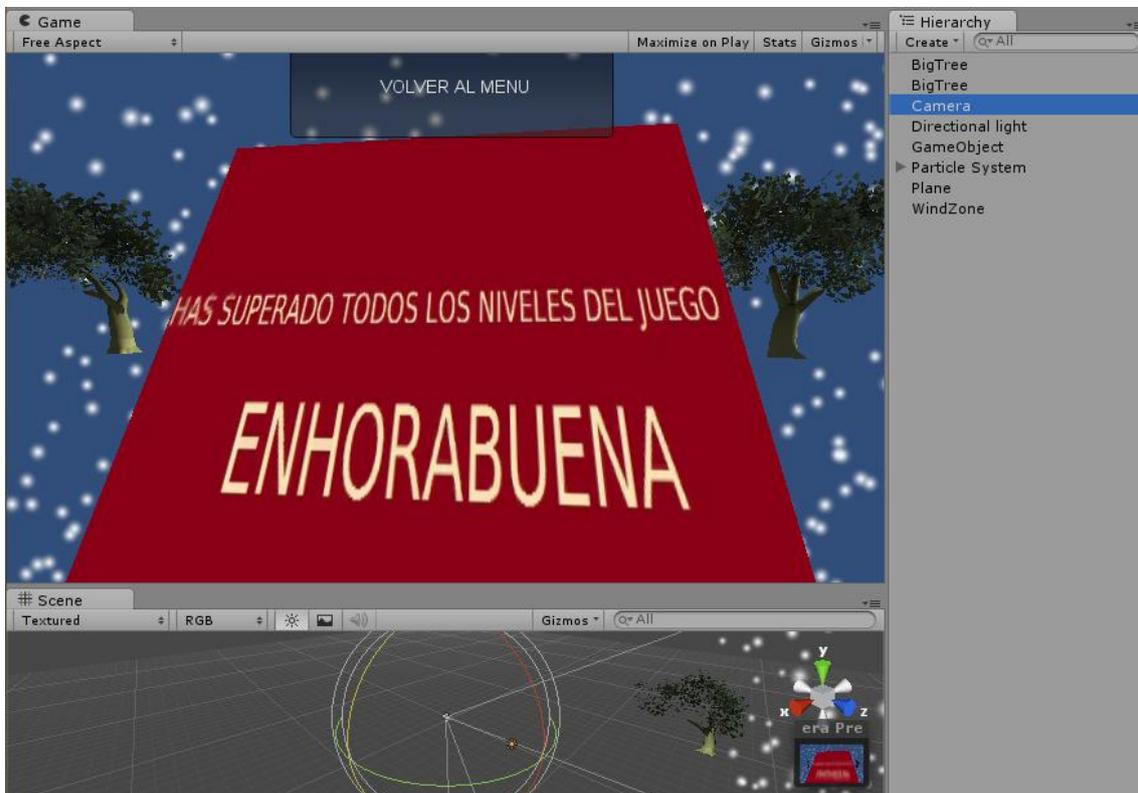


Gráfico 47: Pantalla de Unity3D con el nivel Level 29 activo

El *Level 29* es el último nivel de la aplicación y tiene la función de felicitar al paciente que ha logrado completar sin fallos y de forma consecutiva los 8 niveles de ésta. Aunque el botón VOLVER A MENÚ aparezca muy arriba, cuando la aplicación está activa y a pantalla completa el botón aparece justo encima del texto "HAS SUPERADO TODOS LOS NIVELES DEL JUEGO".

El *Level 29* se compone de varios objetos que se detallan a continuación:

- BigTree: Dos grandes árboles obtenidos de la propia página oficial de Unity3D.
- Camera: La cámara principal que enfoca a la escena.
- Directional Light: Luz direccional que hace que los objetos puedan verse con claridad.
- GameObject: Comprende un botón con la opción VOLVER A MENÚ y un script llamado *MenuGUI5* que está reflejado ya en el Gráfico 48. En este caso no se cargará el *Level 20* sino el *menu*, es decir, el menú principal.
- Particle System: Corresponde a los puntos blancos que va de abajo hacia arriba. No tienen ninguna utilidad real, tan sólo es decorativo.
- Plane: Es el plano inclinado con la textura "HAS SUPERADO TODOS LOS NIVELES DEL JUEGO, ENHORABUENA". Este plano aparece, por defecto, volcado. En este caso se ha girado algo menos de noventa grados para darle sensación de profundidad.
- Wind Zone: Genera una zona de aire para que las hojas del árbol se muevan. Como en el caso del Particle System, es meramente decorativo.

```
var x: float =200;
var y: float =150;
var Ancho : float=100;
var Alto: float=50;
var AjusteX: float= 60;
var AjusteY: float= 60;

function OnGUI (){
    if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY,Ancho,Alto),"VOLVER AL MENU")){
        Application.LoadLevel("menu");
    }
}
```

Gráfico 48: Programación del script *MenuGUI5*

4.3.3. FUNCIONAMIENTO

En el Gráfico 49 se encuentra detallada una estructura de los niveles y de las opciones que contempla cada uno de ellos. Es lógico afirmar que sólo se avanzará de nivel si la casilla que se presiona es la de tipo acierto, mientras que si se presiona en alguno de los errores se volverá al *Level 20* y se tendrá que volver a empezar a acertar desde el principio. Cuando se llegue al final, se habrá terminado con la aplicación y se deberá cargar, de nuevo, la pantalla del menú principal.

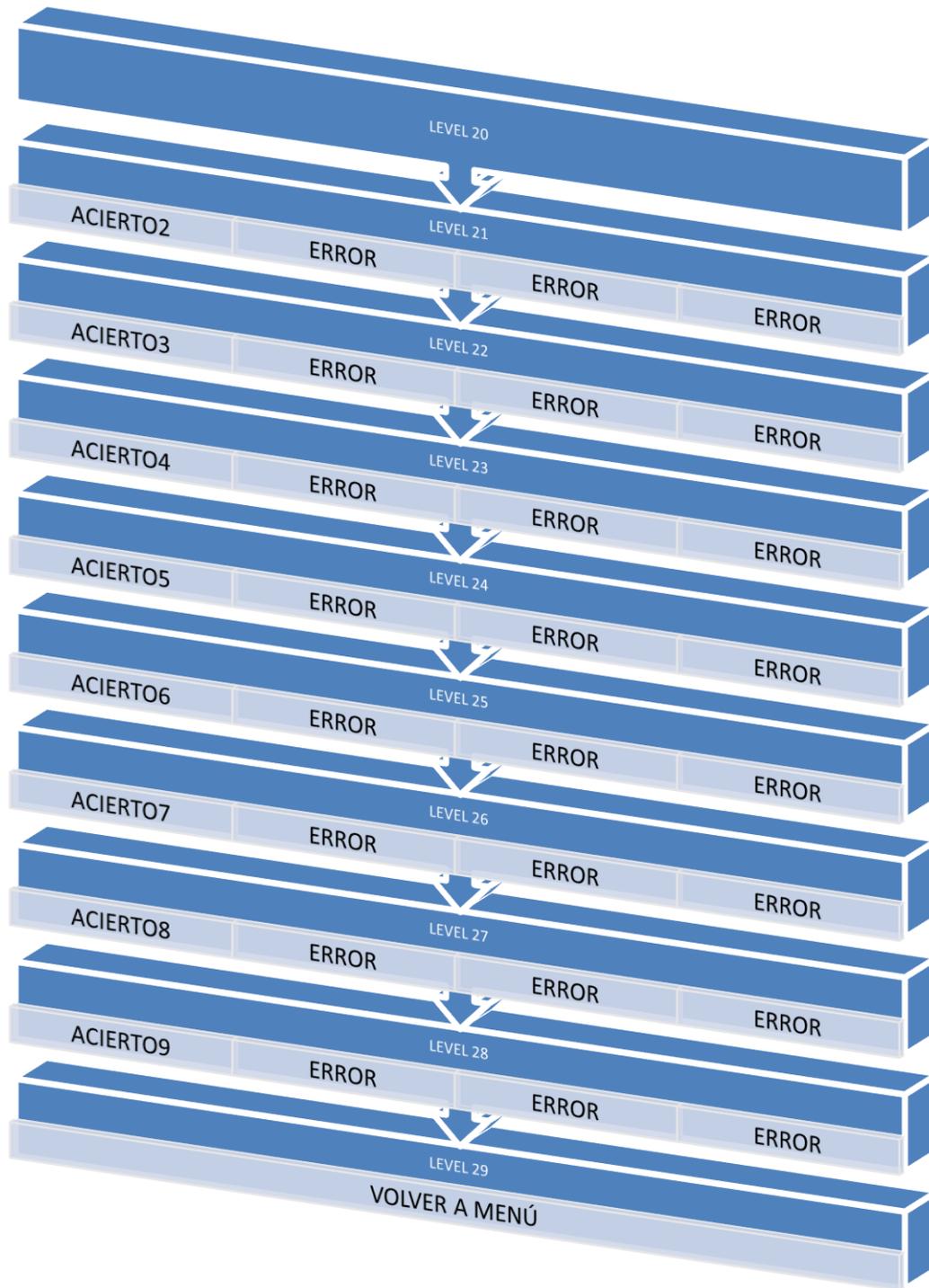


Gráfico 49: Estructura de niveles de "Qué soy"

4.3.4. APLICABILIDAD AL ESTUDIO DE LA ENFERMEDAD DE ALZHEIMER

En este trabajo final de carrera hay que tener siempre en cuenta que estamos tratando con personas, generalmente de avanzada edad, enfermas de Alzheimer. AFAV, la asociación de familiares de enfermos de Alzheimer de Valencia, nos transmitió, antes de elaborar esta aplicación, que ésta debía ser lo más simple posible y no disponer de demasiada complejidad.

Aplicaciones similares están siendo utilizadas para la mejora del tratamiento de la enfermedad de Alzheimer con pacientes que están en su primera fase de deterioro de memoria, pues con casos más avanzados la dificultad de trabajo aumenta y los ejercicios son muy distintos.

En el juego "Qué soy" se ha tratado de trabajar siempre con conceptos simples y cuestiones de la vida cotidiana. Se han establecido botones de un gran tamaño para favorecer el pulsarlos. Las preguntas que se hacen son sobre colores, animales u objetos de la vida cotidiana, favoreciendo así el repaso y la memorización de estos conceptos.

A pesar de estar escrito con letras grandes y visibles, se estableció el color verde para ACIERTO y el color rojo para ERROR. Son hechos simples y comúnmente poco valorados, pero gozan de gran importancia para la comprensión del juego en sí.

4.4. AVENTURA GRÁFICA

4.4.1. INTRODUCCIÓN

La segunda aplicación que se ha implementado es una aventura gráfica. En ella, el paciente en primera persona debe desplazarse por un terreno en tres dimensiones y realizar una serie de pruebas que ponen a prueba la orientación del mismo en un escenario desconocido.

Se intenta que el paciente siga un camino determinado por la posición de las bolas amarillas, existiendo además bolas de color rojo, azul, verde y árboles que pueden inducir a confusión.

Poco antes de que el paciente llegue a la última bola, que incluye una textura con el texto FIN, se habrá concluido la aventura y se felicitará al paciente volviendo al menú principal un par de segundos después.

4.3.2. PROGRAMACIÓN Y DESARROLLO

La parte de aventura consta de 4 pantallas distintas, teniendo especial importancia la tercera de éstas. La primera pantalla, visible en el Gráfico 50, tiene como función únicamente la lectura y comprensión de las instrucciones. En el Gráfico 51 aparece la que será la segunda pantalla, que explica los comandos de dirección del jugador. La tercera pantalla, que, como se ha mencionado, es la más importante, es la propia aventura gráfica, en la que el paciente deberá encontrar la bola amarilla siguiendo el camino de bolas amarillas que se encuentre a su paso. Se muestra una instantánea de esta pantalla en el Gráfico 52, mientras que el Gráfico 53 muestra la pantalla en la que se felicita al paciente por la culminación de la aventura.

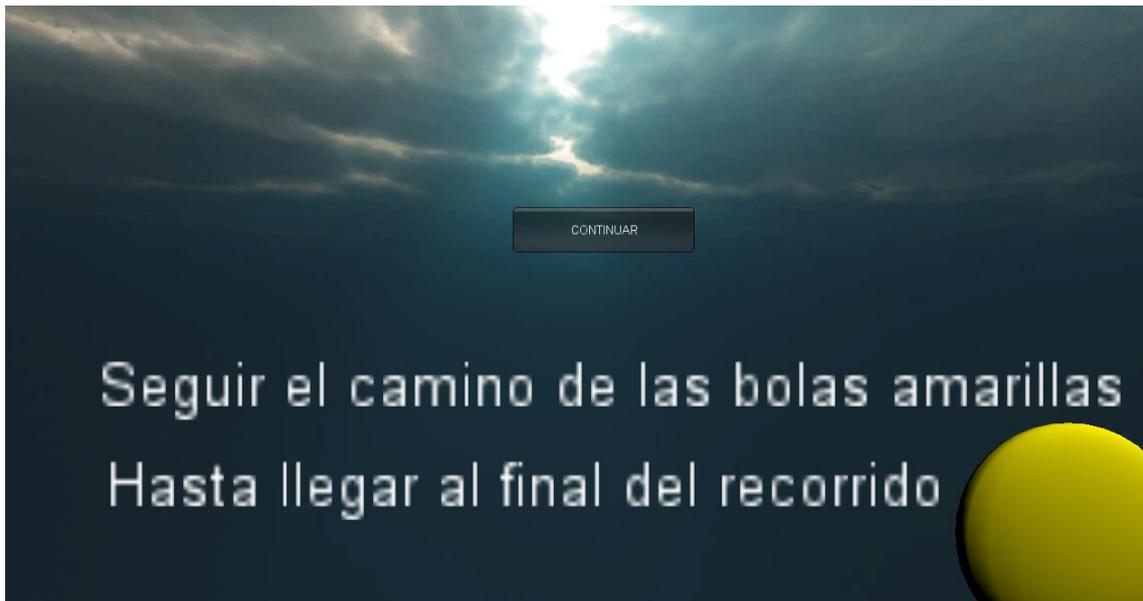


Gráfico 50: Primera pantalla de la aventura gráfica (Level 1)



Gráfico 51: Segunda pantalla de la aventura gráfica (Level 11)

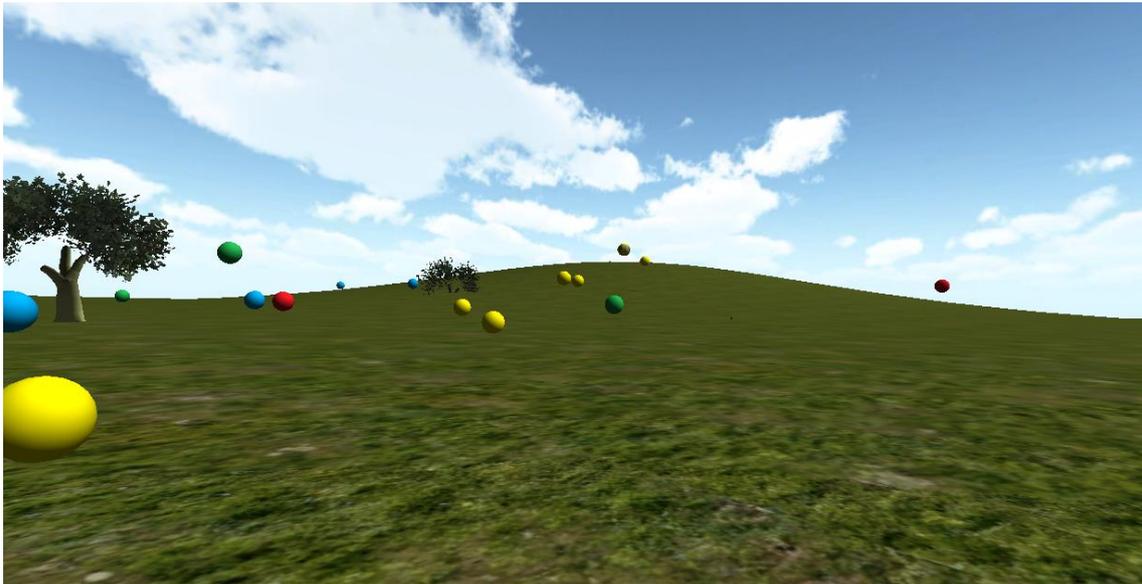


Gráfico 52: Instantánea de la Aventura gráfica (Level 12)



Gráfico 53: Conclusión de la aventura gráfica (Level 19)

El *Level 1* tiene como única función introducir al paciente al programa y darle las instrucciones a seguir para poder trabajar correctamente. Se ha intentado ser lo más claro posible y se ha añadido un botón de continuar, ya que es necesario esperar a que el paciente lo haya leído para proseguir, pues la información que se muestra es imprescindible.



Gráfico 54: Pantalla de Unity3D con el Level 1 activado

El *Level 1* muestra una serie de objetos añadidos en la *Hierarchy*. Se desarrollan a continuación estos objetos y la necesidad de su existencia:

- Directional Light: Importante la existencia de una luz direccional para la correcta lectura del texto y la visualización de la esfera amarilla.
- GameObject: Contiene el botón de texto CONTINUAR con un script al que se ha llamado *InstrucNext*, y que está detallado en el Gráfico 55. El script tiene la función de cargar el *Level 11* cuando se pulse el botón con el texto CONTINUAR.
- Main Camera: Es la cámara principal que enfoca hacia la esfera, el botón y el texto.

- Sphere: Es la esfera amarilla. Simplemente decorativa, sirve para que el paciente ya visualice el color amarillo y le sea más fácil identificar el camino a seguir.
- Texto 1 y Texto 2: Al no poder trabajar con un texto con salto de línea ha sido necesario escribir dos textos para completar las instrucciones del juego.

```
var Ancho : float=100;  
var Alto: float=50;  
var AjusteX: float= 60;  
var AjusteY: float= 60;  
  
function OnGUI () {  
  
if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY,Ancho,Alto),"CONTINUAR")){  
    Application.LoadLevel("Level 11");  
}  
}
```

Gráfico 55: Script InstrucNext



Gráfico 56: Pantalla de Unity3D con el Level 11 activado

El *Level 11* muestra una serie de objetos añadidos en la *Hierarchy*. Se desarrollan a continuación estos objetos y la necesidad de su existencia:

- Directional Light: Importante la existencia de una luz direccional para la correcta lectura visualización de la textura del teclado.
- GameObject: Contiene el botón de texto CONTINUAR con un script al que se ha llamado *TecladoNext*, y que está detallado en el Gráfico 57. El script tiene la función de cargar el *Level 12*, el nivel de la aventura gráfica, cuando se pulse el botón con el texto CONTINUAR. La ubicación del botón que aparece en el Gráfico 56 no es la correcta. Cuando se trabaja a pantalla completa ésta varía.
- Main Camera: Es la cámara principal que enfoca hacia el plano y el botón.
- Plane: Es el plano inclinado sobre el cual se ha dibujado la textura del teclado con los comandos indicados.

```
var Ancho : float=100;
var Alto: float=50;
var AjusteX: float= 60;
var AjusteY: float= 60;

function OnGUI () {

if (GUI.Button(Rect(Screen.width/2-AjusteX, Screen.height/2-AjusteY,Ancho,Alto), "CONTINUAR")){
    Application.LoadLevel("Level 12");
}
}
```

Gráfico 57: Script *TecladoNext*

De estas cuatro, la pantalla más importante, y lógicamente la más compleja, resulta ser la tercera, que es la aventura gráfica. Esta pantalla consta de un terreno de 2000x2000 unidades cuadradas, de las que se aprovecha considerablemente poco, pero no resulta necesario utilizar más. El Gráfico 58 muestra una imagen del recorrido que realiza el paciente hacia su destino, la bola más grande con el texto FIN sobre fondo amarillo.

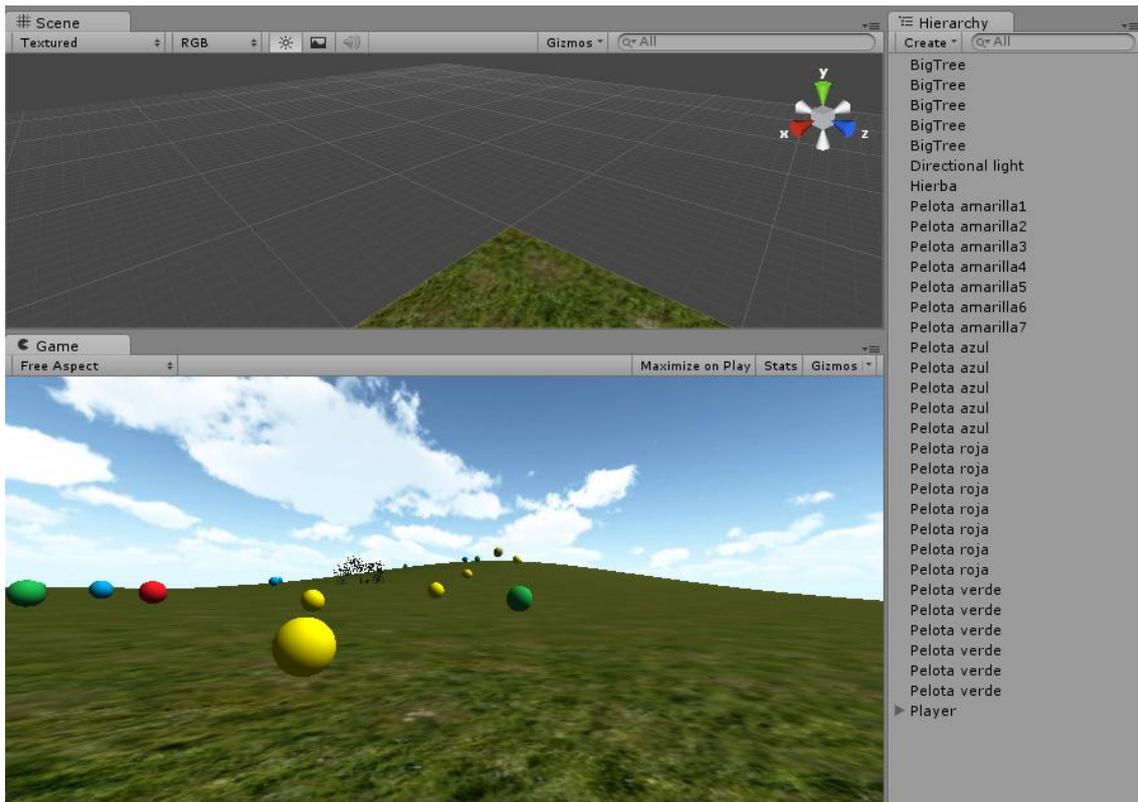


Gráfico 58: Pantalla de Unity3D con el Level 12 activado

La elaboración de esta pantalla consta de una gran cantidad de objetos, muchos de los cuales se encuentran incluso quintuplicados, como pueden ser los tipos de pelotas. Se procede a detallar la funcionalidad de cada uno de ellos y si han requerido alguna configuración:

- BigTree: Árboles que traía incorporados ya el programa. Son meramente decorativos.
- Directional Light: Luz direccional para darle visibilidad al terreno.
- Hierba: Es el nombre que se le ha dado al *Terrain*. Se le ha añadido una textura de hierba para darle un aspecto más bonito. En la configuración de la pantalla, aunque no aparezca en la *Hierarchy* se ha dado una textura al cielo.
- Pelota amarilla 1-7: Se han ubicado las pelotas amarillas siguiendo una dirección que no llega a ser en línea recta pero se asemeja. Para evitar cambios en la ubicación a las pelotas 5, 6 y 7 se les ha quitado la característica de la gravedad, con lo que permanecerán inmóviles y siempre estarán visibles y fijas.
- Pelota azul, Pelota verde, Pelota roja: Estas pelotas tienen todas gravedad y simplemente son para despistar al paciente, que deberá centrarse en seguir el camino de las pelotas amarillas olvidándose del resto.

- Player: Posiblemente el más complejo de todos los objetos que se muestran en esta pantalla. Dentro de *Player* se incluye la forma del jugador que, al no verse, resulta irrelevante que sea una cápsula, una cámara principal que hace que el paciente viva la aventura en primera persona y se le ha añadido al jugador la característica *First Person Controller* que proporciona Unity, una fuente de sonido con la canción de Sean Fournier "King of the World", libre de derechos, y un script al que hemos llamado *Finish* que da fin al juego.

El script *Finish* da finalidad al juego cuando el jugador alcanza la altura a la que se encuentra la última bola, que se ubica en una montaña creada sobre el terreno *Hierba* que ya se ha mencionado. Se adjunta la configuración del script *Finish* en el Gráfico 59.

```
function Update () {
    if (transform.position.y >=15.5){

        Dead();

    }

}

function Dead(){
    Application.LoadLevel("Level 19");
}
```

Gráfico 59: Script Finish

El script está basado en la función *Dead* o muerte que se utiliza en otros videojuegos y funciona cargando el *Level 19* (el nivel de felicitación al paciente) cuando la altura supera las 15.5 unidades, sabiendo que se parte desde la altura 0. Cuando esto ocurra, que será siempre poco antes de llegar a nuestro objetivo, se cargará el *Level 19*.

El *Level 19* es simplemente decorativo. Trata de felicitar al paciente por el buen trabajo realizado. Como gran novedad incluye un temporizador que, automáticamente, carga el menú secundario de la aplicación tras unos segundos de espera. A continuación se explica cómo se ha programado este temporizador y cuáles son el resto de objetos presentes en el *Level 19*, último de la aventura gráfica.

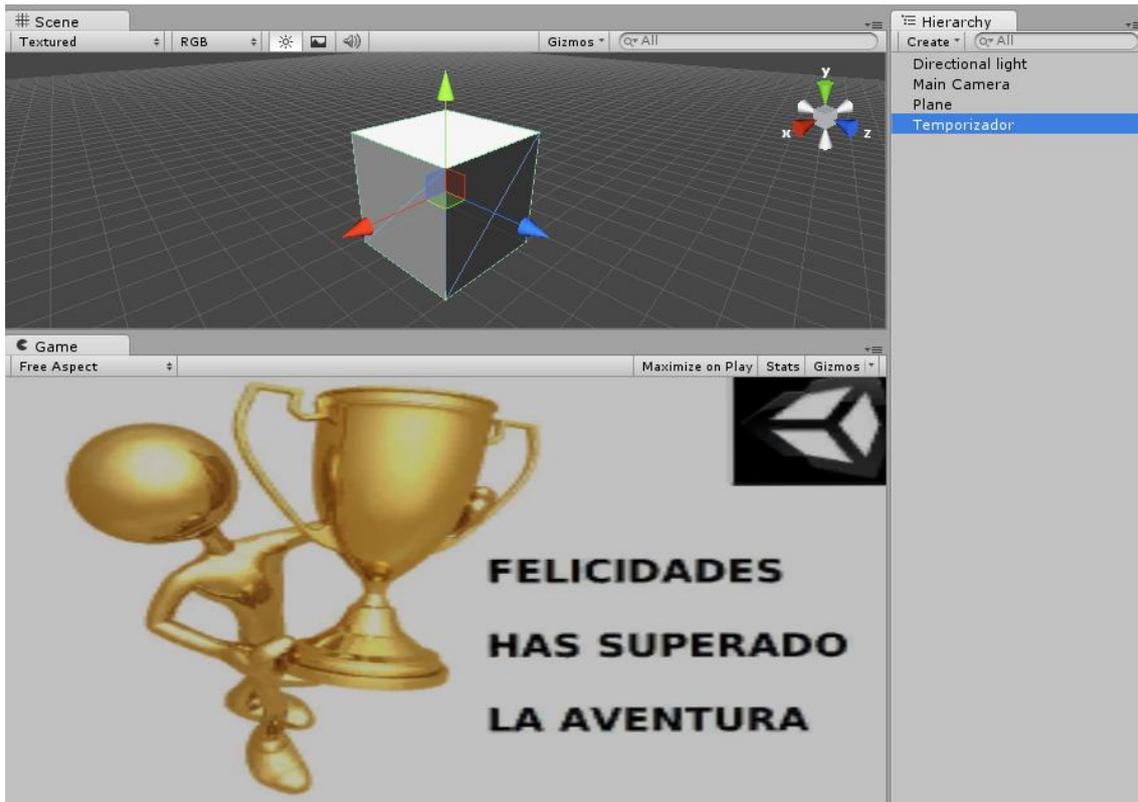


Gráfico 60: Pantalla de Unity3D con el Level 19 activado

En el Gráfico 60 podemos observar en la parte inferior la pantalla que se podrá contemplar cuando se haya completado la aventura gráfica. En la parte superior se puede ver un cubo que funciona como temporizador. Aplicando la gravedad a ese cubo y otorgándole un script de tipo "muerte" llamado también *Temporizador* cuando el cubo esté a una altura inferior a la establecida, la pantalla desaparecerá y se retornará al menú secundario de la aplicación. El resto de componentes han sido explicados anteriormente. Se adjunta en el Gráfico 61 el script *Temporizador*.

```
function Update () {  
    if (transform.position.y <=-50){  
  
        Dead();  
    }  
  
}  
  
function Dead(){  
    Application.LoadLevel("menu 1");  
}
```

Gráfico 61: Script Temporizador

4.4.3. FUNCIONAMIENTO

El paciente debe completar el circuito que se le propone siguiendo el camino que le marcan las bolas amarillas hacia la bola amarilla final con el texto FIN marcado en azul sobre fondo amarillo.

Los comandos son básicos: Dispone del cuadro de mandos para la dirección y del ratón para la orientación de la visión. También tenemos la opción de saltar con la barra espaciadora del teclado.

Cuando el paciente haya completado el trayecto que se le propone, se le felicitará por ello y se retornará al menú secundario para poder acceder, de nuevo, a esta aplicación o al juego "Qué soy" de preguntas y respuestas.

Las bolas del resto de colores, al igual que los árboles, están situadas con la intención de confundir al paciente. Cabe decir que las bolas amarillas gozan de un tamaño mayor y que, igual que las bolas del resto de colores pueden moverse, 3 de las bolas amarillas permanecen fijas en el sitio para facilitar la ubicación del final.

4.4.4. APLICABILIDAD AL ESTUDIO DE LA ENFERMEDAD DE ALZHEIMER

Según los estudios realizados, uno de las mayores dificultades que tiene un paciente enfermo de Alzheimer es la orientación y el seguimiento de trayectorias marcadas. Con esta finalidad, se ha diseñado un juego que permite al paciente entrenar el seguimiento de bolas del mismo color, marcándose ese objetivo y distrayéndose y obviando el resto de bolas de otros colores que no sean relevantes.

Con esto, se pretende que el paciente pueda adquirir habilidades para orientarse en su vida y ser capaz de seguir unas pautas marcadas para llegar a un objetivo final.

Los colores escogidos son conocidos por todos y el tramo que se debe completar es considerablemente corto, para no dificultar demasiado su correcta realización. Con este objetivo, se intentará favorecer al paciente y se espera que pueda realizarlo con éxito.

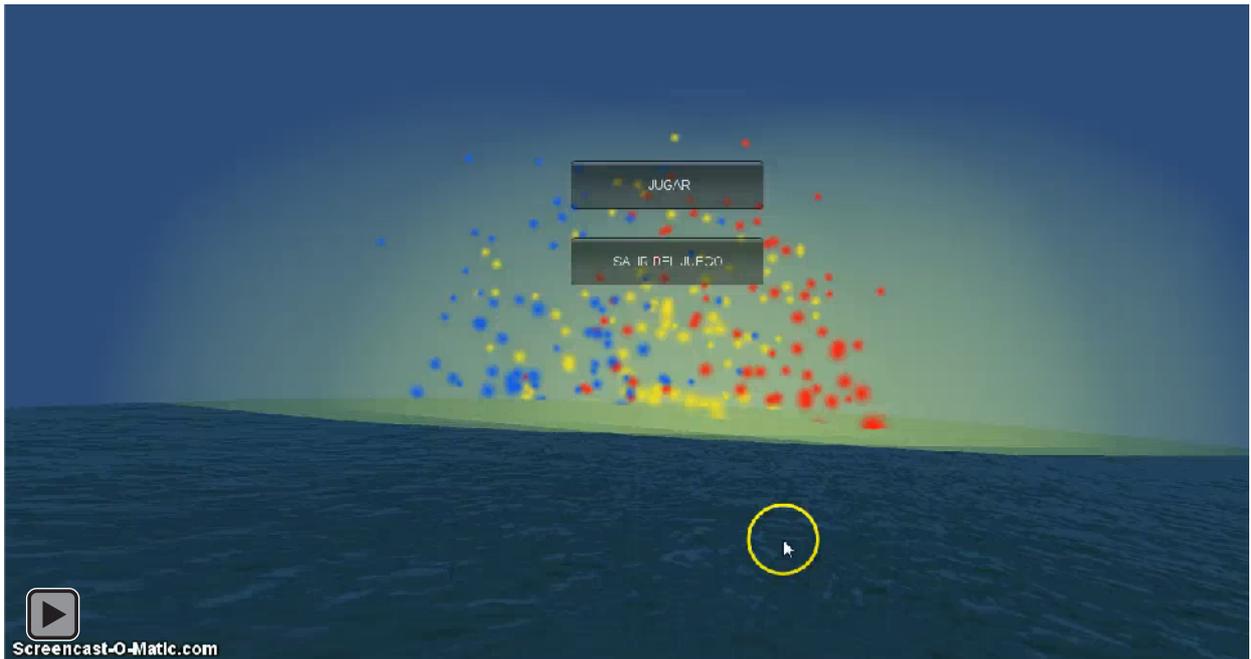
4.5. DEMOSTRACIÓN VISUAL DE LA APLICACIÓN

4.5.1. INTRODUCCIÓN

De cara a la correcta visualización de la aplicación que se ha realizado para estudiar la viabilidad de la plataforma Unity3D enfocada al trabajo que se pueda realizar con personas que padezcan problemas de Alzheimer, se ha optado por elaborar un *screencast* en que se pueda ver la evolución del programa, el salto entre niveles y poder contemplar cómo se juega a esta aplicación en cuestión.

La explicación la lleva a cabo el propio autor del proyecto, realizando todas las transiciones posibles del proyecto, exceptuando la salida del juego, que se realiza presionando SALIR DEL JUEGO en el menú principal.

4.5.2. VIDEO EXPLICATIVO



5. CONCLUSIONES Y LÍNEAS FUTURAS

Para realizar unas conclusiones válidas del proyecto se ha creído conveniente probar el producto con varios pacientes enfermos de Alzheimer en el primer escalón de la degeneración, que son todavía suficientemente conscientes de las cosas que hacen y que tienen todavía relativa facilidad para recordar.

Gracias a la AFAV (Asociación de Familiares de enfermos de Alzheimer de Valencia) se ha podido probar la aplicación con 4 pacientes que padecían la enfermedad. A pesar de que la cantidad de pruebas no pudo ser más amplia, se llegó a una serie de conclusiones:

- Los pacientes son de avanzada edad, con lo que el manejo del ordenador portátil que se les proporcionó no fue fácil.

- Las pruebas pueden tener mayor dificultad, pues para ellos resultó asequible superar los niveles del juego "Qué soy", añadiendo datos como "Puede ser una blusa o una chaqueta de caballero" cuando salió la foto de la camiseta o "Es un pastor alemán como el que yo tuve" cuando apareció el perro.

- A los 4 pacientes les pareció muy entretenido y divertido, además de visualmente atractivo, e incluso llegaron a ver las pruebas de la aplicación como un hecho relevante para contar a sus familias.

- Dos de los cuatro pacientes se pusieron muy nerviosos con la realización de las pruebas, a pesar de realizarlas con facilidad. Notaron, según me comentaron, que tenían una gran responsabilidad al ser los protagonistas y al enfrentarse ellos solos a algo tan novedoso como es una aplicación multimedia.

- Una pantalla táctil facilitaría en gran medida el trabajo que se realizó, pudiendo incluso subir el nivel de dificultad para algunos pacientes.

En definitiva fue una buena experiencia y con ello se demuestra que se puede utilizar la plataforma Unity3D para la realización de aplicaciones con el objetivo de entrenar la memoria para ralentizar el deterioro que esta tiene por culpa de la enfermedad de Alzheimer. Se muestran en los Gráficos 62 y 63 dos instantáneas que se tomaron el día de las pruebas en el centro de AFAV.

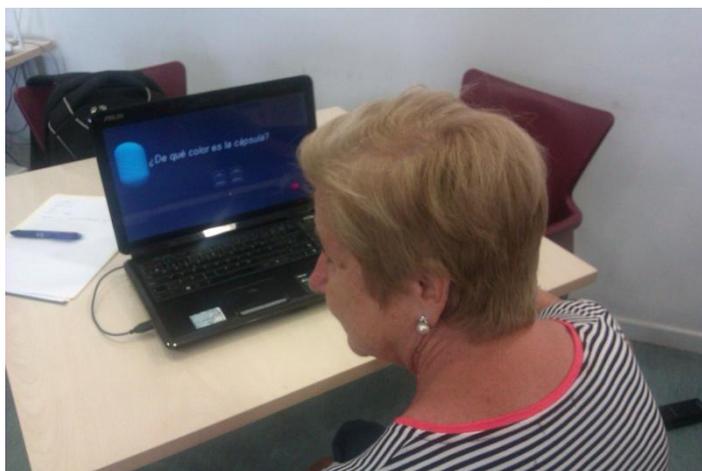


Gráfico 62: Paciente probando el juego (I)



Gráfico 63: Paciente probando el juego (II)

Se considera después de esta experiencia y tras la elaboración del TFC que sí es viable la implementación de juegos dedicados a entrenar la memoria de pacientes enfermos de Alzheimer y se añaden unas sugerencias de mejora del producto que puedan ser implementadas en un futuro.

Por desgracia, tal y como se muestra al principio de este TFC, la propagación de la enfermedad de Alzheimer es una realidad de este tiempo y es necesario seguir trabajando en el entrenamiento de la memoria en los pacientes que la padezcan.

La aplicación puede incrementar la dificultad en sus preguntas en el juego "Qué soy" y puede ampliarse el terreno de la "aventura gráfica". Debería considerarse la opción de adquirir la licencia necesaria para implementar esta aplicación sobre una Tablet, simplificando así las operaciones que el paciente debe realizar con los comandos que se muestran en esta aplicación, véase botones y desplazamientos en el terreno de la "aventura gráfica". Esto facilitaría la adaptación de los pacientes de avanzada edad y sin conocimientos informáticos a la aplicación sobre cualquier plataforma.

Tal y como se ha visto, la licencia de iOS tiene un coste de 305€, que se sumarían a los 1140€ que costaría la licencia de Unity3D Pro. Esta segunda licencia resultaría opcional, pero recomendable si se quisiera mejorar el nivel general de los gráficos que tiene la aplicación creada.

La Asociación de Familiares de enfermos de Alzheimer agradeció mucho el trabajo que se está realizando enfocado a la mejora de los recursos que ellos poseen y se postularon como posibles futuros clientes si este proyecto se llevaba a cabo con éxito.

Hasta ahora, todo el trabajo se ha realizado bajo la plataforma de PC por el simple hecho de ser gratuita, pero, tras las pruebas realizadas en el centro de la AFAV, resulta inevitable afirmar que es muy probable que sea la menos adecuada, resultando el resto de plataformas, todas ellas de pago, mucho más atractivas y adaptadas para el uso de pacientes del perfil de un enfermo de Alzheimer de avanzada edad, desconocedor de las nuevas tecnologías.

6. BIBLIOGRAFÍA

- [1] ANA MORÓN, "Alzheimer - Cuaderno del voluntario" Asociación de Familiares de Alzheimer de Valencia, 2003.
- [2] WILL GOLDSTONE, "Unity 3.x Game Development Essentials", Ed. Packt Publishing, 2011.
- [3] GRAHAM McALLISTER, "GUI Essentials"
- [4] GRAHAM McALLISTER, "Introduction to Scripting with Unity"
- [5] <http://www.unity3d.com>
- [6] <http://www.unityspain.com>
- [7] [http://en.wikipedia.org/wiki/Unity_\(game_engine\)](http://en.wikipedia.org/wiki/Unity_(game_engine))
- [8] <http://www.youtube.com/watch?v=dqtha6D3RWs>
- [9] http://www.youtube.com/user/mrjocyf?feature=results_main

7. ANEXOS

En este apartado se muestran los archivos que se adjuntan en el CD del TFC y que resultan necesarios para la comprensión del mismo. Se especifica el nombre del archivo anexo y su función.

ejemplo pfc.mp4 --> Video donde se trabaja con la aplicación final y se comprueba que funcionan las transiciones entre niveles

Alzh3D.exe --> Es la aplicación. Se recomienda ejecutarla a pantalla completa para su correcto funcionamiento.

Alzh3D_Data --> Es la carpeta adjunta a la aplicación.

Agradecimientos:

En primer lugar a mis padres, a mi hermana y al resto de familiares, por haber estado siempre a mi lado en estos años de carrera que han sido tan duros y difíciles.

A todos mis compañeros de la carrera y demás amistades que he hecho en la universidad por esas noches eternas estudiando exámenes que, al final, se aprobaron.

A Josetxo, por haber creído en mí desde el primer día que estuve en su despacho proponiéndole el proyecto hasta el día en que el TFC se presente oficialmente.

A la AFAV, por haberme abierto las puertas y haber colaborado conmigo en esto.

A Jocyf y a Fernando Matarrubia, por crear tutoriales de Unity, que me han ahorrado tantas horas de búsqueda y me han facilitado muchísimo el aprendizaje.

A todos los que he mencionado y a los que se me olvida nombrar, a todos los que han creído en mí y han pensado que esto podía salir adelante, MUCHÍSIMAS GRACIAS.

"Si buscas resultados distintos, no hagas siempre lo mismo"

Albert Einstein