

Juego en 3D para la evaluación de la memoria de las personas

3D game for people's memory evaluation

FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID



GRADO EN INGENIERÍA INFORMÁTICA
TRABAJO DE FIN DE GRADO

AUTORES: Yiteng Wang , Yifei Liu
DIRECTOR: Iván García-Magariño García

CURSO ACADÉMICO: 2019/2020
CONVOCATORIA: Junio de 2020

Resumen

Como motor de videojuegos profesional, Unity3D, ha sido muy popular en este sector de motores gráficos en los últimos años. Debido a su alta flexibilidad y la sencillez de aprendizaje, muchos principiantes lo eligen como su primer motor gráfico para desarrollar su juego.

Este proyecto tiene como objetivo desarrollar un juego en 3D utilizando Unity para evaluar la memoria del usuario, de manera que pueda conocer su capacidad de memoria a través del juego y ejercitarla. El proyecto dispone de tres niveles y hemos escogido escenas que acercan más a la vida cotidiana, con el fin de dar una sensación de familiaridad al usuario para que pueda hacer el test relajado. El juego, a través de la memorización de las ubicaciones de los objetos, evalúa la memoria visual a corto plazo y la memoria espacial del usuario.

En el final de esta memoria se presentan los resultados preliminares del proyecto, los cuales tras realizar una comparación de resultados de la medida con otro test de memoria validado se encuentra una cierta correlación.

Palabras clave

- Juego de memoria
- Memoria de corto plazo
- Unity3D
- Evaluación

Abstract

As a professional video game engine, Unity3D has been very popular in this graphics engine sector in recent years. Due to its high flexibility and simplicity of learning many beginners choose it as their first graphics engine to develop their game.

This project aims to develop a 3D game using Unity to assess the user's memory, so that they can know their memory capacity through the game and exercise it. The project has three levels and we have chosen scenes that are closer to everyday life, in order to give the user a feeling of familiarity so that they can do the relaxed test. The game, by memorizing the locations of the objects, evaluates the short-term visual memory and the user's spatial memory.

At the end of this paper, the preliminary results of the project are presented, which after comparing the measurement results with another validated memory test, finds a certain correlation.

Keywords

- Memory game
- Short-term memory
- Unity3D
- Evaluation

Índice

Resumen	1
Palabras clave	1
Abstract	2
Keywords	2
Introducción	5
1.1 Objetivo	5
1.2 Plan de trabajo	5
Introduction	8
1.1 Purpose	8
1.2 Work plan	8
Estado del arte	10
2.1 Estudio de memoria	10
2.1.2 Experimento científico	12
2.1.3 Trabajo relacionado	13
2.2 Juegos de memoria	14
Videojuego	19
3.1 Análisis	19
3.1.1 Diagrama de casos de uso	19
3.1.2 Diagrama de secuencia	21
3.2 Creación de videojuego en Unity 3D	24
3.3 Mecánicas del juego	31
3.4 Herramientas utilizadas	33
3.4.1 Unity	33
3.4.2 C#	34
3.4.3 Visual Studio	34
Evaluación de usuarios	35
4.1 FeedBack	35
4.2 Comparación	38
Contribuciones individuales	40
5.1 Yi Teng Wang	40
5.2 Yifei Liu	41
Conclusiones y trabajo futuro	43
6.1 Conclusión	43

6.2 Trabajo futuro	43
Conclusions and future work	47
6.1 Conclusions	47
6.2 Future work	47
Bibliografía	50
Vistas del juego	52

Capítulo 1

Introducción

La memoria es una capacidad importante que tenemos ya que interviene en la mayoría de nuestras actividades diarias, es como una caja que nos permite guardar las cosas que hemos aprendido, vivido o visto. Cada uno tenemos nuestra caja con tamaño diferente, que puede guardar informaciones limitadas dependiendo de la capacidad de memorización que tenemos cada persona.

Sin embargo, hay personas que sufren por la pérdida de memoria, sea por enfermedad, como Alzheimer, o por envejecimiento. Pero la memoria puede ser entrenada para conseguir una mayor capacidad de memorización y poder guardar más informaciones. Para ello existen varios tests de memoria para evaluar y entrenar la memoria que se puede encontrar con facilidad por internet u otras vías.

1.1 Objetivo

El proyecto tiene como objetivo diseñar un juego simple para que el usuario pueda evaluar y entrenar la memoria en un entorno al aire libre. El juego consiste en que el usuario tiene que recoger un número de objetos en el juego y recordar dónde estaba puesto, luego se pedirá al usuario que devuelva los objetos a su sitio, de esta forma se evalúa la memoria a corto plazo del usuario contando los objetos que ha conseguido colocar en su lugar correcto.

1.2 Plan de trabajo

Para desarrollar el proyecto hemos aplicado el modelo de desarrollo iterativo con las fases de investigación, diseño del juego, implementación del código y test del juego.

1. Investigación

En esta fase tratamos de buscar información sobre la plataforma Unity y conocer sus funcionalidades, dado que esta herramienta es completamente nueva para nosotras, hemos practicado también con pequeños tutoriales antes de empezar a crear nuestro juego.

2. Diseño

Una vez obtenido suficientes conocimientos para desarrollar el juego, comenzamos con el diseño, como la dinámica del juego, las reglas y la parte gráfica. En esta fase hemos creado un guión que define los escenarios, objetos y otros aspectos del juego.

3. Implementación

Con la dinámica y reglas de juego decidida, nos ponemos con la implementación del juego en código.

4. Test

El objetivo de esta fase es encontrar los “bugs” y aumentar la calidad del juego, por tanto se realizará varias pruebas para sacar versiones mejores.

Finalmente después de acabar el proyecto, el resto del tiempo lo hemos dedicado a la elaboración de esta memoria y la revisión de la misma. En el siguiente diagrama se refleja una estimación de tiempo para cada fase.

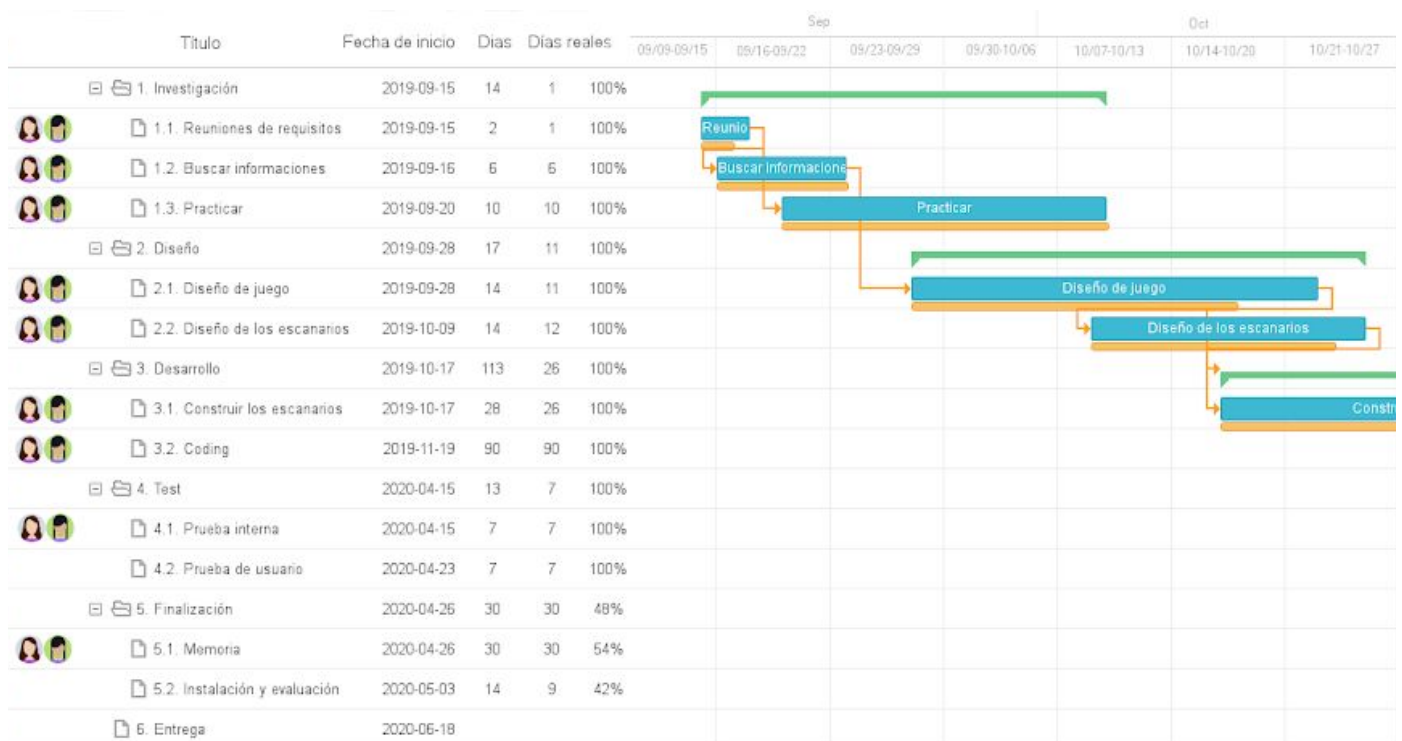


Figura 1.1 : Diagrama de gantt

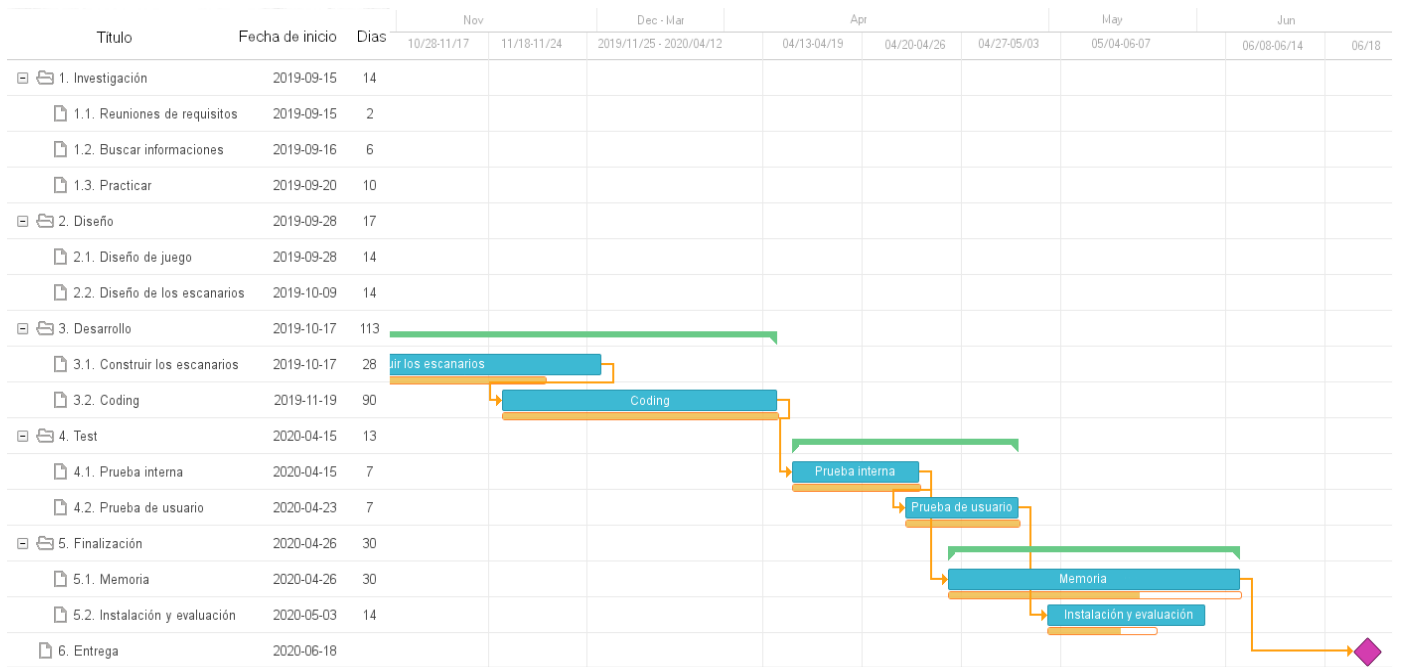


Figura 1.2 : Diagrama de gantt

Introduction

Memory is an important capacity that we have since it intervenes in most of our daily activities, it is like a box that allows us to keep the things that we have learned, lived or seen. We each have our box with a different size, which can store limited information depending on the memory capacity that each person has.

However, there are people who suffer from memory loss, either due to disease, such as Alzheimer's or aging. But the memory can be trained to achieve a greater memory capacity and to save more information. For this there are several memory tests to evaluate and train memory that can be easily found on the internet or other means.

1.1 Purpose

The project aims to design a simple game so that the user can evaluate and train memory in an outdoor environment. The game is that the user has to collect a number of objects in the game and remember where it was placed, then the user will be asked to return these objects to the original place, in this way the user's short-term memory is evaluated by counting the objects that you have managed to put in their correct place.

1.2 Work plan

To develop the project we have applied the iterative development model with the investigation, game design, code implementation and game test phases.

1. Investigation

In this phase we try to find information about the Unity platform and know its functionalities, since this tool is completely new for us, we have also practiced with simple tutorials before starting to create our game.

2. Design

Once we have obtained enough knowledge to develop the game, we start with the design, such as the dynamics of the game, the rules and the graphic part. In this phase we have created a script that defines the scenarios, objects and other aspects of the game.

3. Implementation

With the dynamics and rules of the game decided, we get down to the implementation of the game in code.

4. Test

The objective of this phase is to find the bugs and increase the quality of the game, therefore several tests will be carried out to get better versions.

Finally, after finishing the project, the rest of the time we have dedicated to preparing this report and reviewing it.

Capítulo 2

Estado del arte

2.1 Estudio de memoria

La memoria, explicada en términos de la segunda morfología de la vida, se refiere a la impresión de los cinco canales de información de visión, oído, tacto, olfato y gusto en el cerebro después de enfocarse en cosas objetivas.

Se puede clasificar la memoria en[4] :

1. Según el canal de uso, se puede dividir en:
 - Memoria visual: Se basa principalmente en la memoria de impresión de forma y color.
 - Memoria auditiva: Después de que se presenta la información de voz, se registra en la memoria auditiva de inmediato y se retiene un corto tiempo en la memoria.
2. Según el contenido de la memoria, se puede dividir en:
 - Memoria representativa: Las cosas percibidas no se reproducen delante de los ojos sino en la mente.
 - Memoria lógica: Se refiere al proceso de memoria en el que las personas forman la comprensión o conclusión de las cosas a través del proceso de pensamiento lógico abstracto de concepto, razonamiento, análisis y juicio después de recibir información externa, y lo dejan en la memoria.
 - Memoria emocional: Es un recuerdo de las emociones y sentimientos que he experimentado.
 - Memoria procedimental: Es cuando el sujeto no sabe conscientemente que tiene este tipo de memoria, solo puede expresarse naturalmente en la operación de una tarea específica. La operación de esta tarea no depende de la recuperación consciente del sujeto de la experiencia previa.

3. Según la duración del tiempo de almacenamiento de información de la memoria, la memoria se divide en:
 - Memoria a largo plazo: mantener unos días o años de memoria.
 - Memoria a corto plazo: solo mantener unos segundos a unas pocas horas

2.1.1 Método de loci (Memory Palace)

Memory Palace existió desde la antigua Roma, es uno de los métodos más poderosos hasta la fecha y también es el método que más se utiliza en el entrenamiento para El Campeonato Mundial de Memoria. Este método provenía de un poeta griego Simónides de Ceos. Una vez Simónides fue invitado a cenar en la casa de un hombre rico llamado Scopas, el cual salió de casa después de componer un poema, y en el momento que se fue, la casa se derrumbó. Todos los invitados murieron en el interior, los cuerpos estaban muy dañados y para los familiares eran difíciles de distinguir, pero Simónides reconoció los cuerpos uno por uno según los asientos de los invitados.

Memory Palace es como una caja donde puedes poner la información que deseas memorizar en esa caja del cerebro, para que pueda sacar rápidamente las informaciones por la caja cuando quieras recordar.

Construir un palacio de memoria[5] consiste en 4 pasos:

1. Elegir un palacio y definir una ruta

Primero, debe elegir un lugar con el que esté muy familiarizado, por ejemplo, una mejor opción inicial podría ser su casa. Cuanto más fiel sea la reproducción de los detalles de este lugar, hay más posibilidad de recordarlo.

Segundo, determine una ruta especial en su palacio, en lugar de una escena fija, haga un recorrido detallado de su hogar en lugar de visualizarlo simplemente.

Algunos ejemplos de otras rutas:

- Paisaje. Imagina una ruta al caminar o trotar.
- Escuela. Imagina la ruta del aula a la cafetería, la biblioteca o el laboratorio.
- Camino a la ciudad. El orden de las rutas o calles para ir a trabajar.

2. Lista de características obvias

Preste atención a las características obvias del lugar. Por ejemplo, cuando elige visitar su casa, la puerta debe ser la primera característica que atraiga la atención. Después de entrar por la puerta, preste atención a lo que hay en la primera habitación. Descomponga la sala (se puede determinar una secuencia, como de izquierda a derecha), y se determina la siguiente característica que llama la atención. Recuerde que mientras camina, cada una de ellas se convertirá en una "caja", que luego puede usarse para almacenar información específica.

3. Recuerda este palacio

Para que este método sea efectivo, el lugar o la ruta debe estar 100% impreso en la mente.

4. Asociación

El Palacio de la Memoria funciona a través de una asociación visual. Proceso: seleccione una imagen conocida (enlace de memoria) y combine los elementos que desea recordar. El gancho de memoria es una característica obvia en nuestro palacio de la memoria. La forma correcta de visualizar la asociación es: loca, divertida, desagradable, inusual, etc. La regla única que crea la escena: lo tedioso está mal.

2.1.2 Experimento científico

Dresler Martin y su equipo de la Universidad de Radboud en los Países Bajos reclutaron a 23 de los 50 mejores atletas de memoria en la competencia mundial de memoria, y al mismo tiempo seleccionaron un grupo de control que nunca han sido entrenados y que coincidan con la edad, el sexo y el coeficiente intelectual de estos jugadores de clase mundial.[\[6\]](#)

En el experimento, los investigadores primero escanearon los cerebros de 17 atletas de memoria y el grupo de control en un estado de relajación mental. Después, hicieron una prueba de memoria con 72 palabras mientras eran escaneados. Después de eso, 51 personas del grupo de control se dividieron aleatoriamente en tres grupos. Un grupo realizó un entrenamiento de método de loci durante las siguientes 6 semanas, el otro grupo utilizó el mismo tiempo para el entrenamiento de la memoria de trabajo, y el último grupo no hizo nada. Regresaron al laboratorio después de 6 semanas para hacer un escáner cerebral y una prueba de memoria, y registraron los nuevos datos después de 4 meses.

En la primera parte del experimento se descubrió que después de 20 minutos de memorizar la lista de palabras, el atleta de memoria pudo recordar 70 palabras de promedio, a comparación de solo 40 en el grupo de control.

Esta investigación no solo descubrió que los cerebros de los atletas de memoria y las personas comunes son diferentes, sino que también mostró que las personas comunes también pueden convertirse en personas de súper memoria a través del entrenamiento. Efectivamente a través del entrenamiento de alta intensidad, los participantes del experimento ordinario también pueden recordar la misma cantidad de palabras que los atletas de memoria. Al mismo tiempo, los datos del escáner cerebral mostraron que después del entrenamiento, la conectividad funcional cerebral de estas personas también está más cerca a la de los jugadores de memoria.

Métodos de entrenamiento para atletas de memoria:

1. Necesitan almacenar algunas imágenes particularmente vívidas en la memoria a largo plazo del cerebro, y luego, cuando se necesita la información de la memoria en el juego, esta información temporal se conecta a las imágenes que ya están en el cerebro para ayudarlos a recordar mucha información.

2. Se almacena una ruta muy familiar en la memoria a largo plazo del cerebro. Esta ruta tiene una posición fija. Durante la competencia, el jugador coloca las imágenes temporales que tiene que recordar en estas posiciones de la ruta y relacionar las imágenes con esas posiciones familiares. Cuando necesite recordar, simplemente "camine" a estas posiciones en su imaginación y observe qué imágenes específicas están colocadas.

2.1.3 Trabajo relacionado

En el artículo de *Smart Cupboard for Assessing Memory in Home Environment* [7], el equipo desarrolló un gabinete inteligente(SC) para evaluar la memoria del usuario. Este SC tiene 3 compartimentos del mismo tamaño, contiene de 5 a 10 elementos sin superposición en cada compartimento y dispone de sensores en cada puerta. El equipo reclutó a 23 voluntarios con una edad promedio de 36.17 años.

El experimento consiste en dos fases. En la primera fase, los usuarios deben observar el interior del SC e intentar recordar todos los objetos de cada compartimento (15 objetos en cada uno), en un tiempo de 30s para cada compartimento. Luego en la fase recuperación, el experimentador seleccionará 10 objetos al azar y se pedirá al usuario que los encuentre. El experimento ha seleccionado 30 objetos comunes en armarios de cocina, como tazas, huevos, sal, leche condensada, etc.

El equipo realizó varios análisis considerando las mediciones de memoria en los diferentes métodos, como el tiempo de reacción y la edad de los participantes.

Hicieron también una comparación de resultado del SC test con *A test of face–name pairs*[8] (ya que este tipo de prueba ha sido validada para evaluar la memoria), y los resultados muestran que el test tuvo éxito.

Nuestro proyecto al igual que el test de gabinete inteligente evalúa la memoria visual-espacial a corto plazo del usuario mediante observación - memorización - recuperación de objetos, también hemos seleccionado escenas y objetos acerca de la vida cotidiana para que el usuario tenga una sensación de familiaridad y poder estar más relajado durante el test. No obstante, para realizar el test necesitan un local suficientemente amplio y un personal que controla el avance del test, y con el juego se puede ahorrar todos estos trabajos, además ofrece la posibilidad de evaluar la memoria cuando quiera y donde quiera siempre que haya un PC al lado.

En nuestro juego el usuario se encuentra en una habitación, en la primera fase va a tener que recoger un número determinado de objetos y recordar bien donde estaban, aquí le damos al usuario la libertad de elegir los objetos que quiera recordar. En la fase de “recuperación” el usuario tiene que recolocar los objetos que había cogido anteriormente, y notamos su tasa de acierto y fallo que se refleja en la sección de score de la pantalla, tanto acierto como fallo se tendrá en cuenta a la hora de evaluar. Dado que ahora el usuario tiene más movilidad y más espacio para observar, hemos dejado más tiempo tanto en la fase de adquisición como recuperación.

2.2 Juegos de memoria

1. Brain Workshop[9]

El entrenamiento *Dual N Back* es un proyecto de actividad psicológica que la comunidad científica ha demostrado que mejora la memoria de trabajo y la inteligencia fluida de las personas. Un proyecto de investigación de memoria llamado *dobles n-back* publicado en PNAS promueve la investigación de la memoria a corto plazo y la inteligencia fluida.

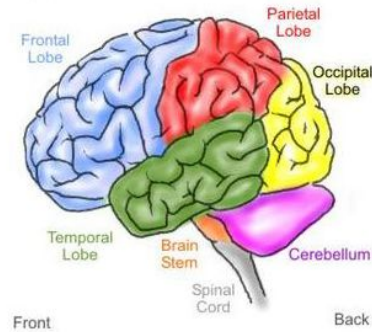
Dual significa información de doble canal durante el entrenamiento. *N Back* significa juzgar si la información de los dos canales es la misma que la información de los *N* canales anteriores.

Brain Workshop es un software gratuito que puede proporcionar capacitación Dual N Back. Las características principales de este software son: admite múltiples usuarios, puede registrar resultados de entrenamiento, puede ajustar automáticamente la dificultad de acuerdo con el rendimiento, etc.

Brain Workshop

Version 4.8.4

Regions of the Human Brain



- C: Choose Game Mode**
- S: Choose Sounds**
- I: Choose Images**
- U: Choose User**
- G: Daily Progress Graph**
- H: Help / Tutorial**
- D: Donate**
- F: Go to Forum / Mailing List**
- O: Edit configuration file**

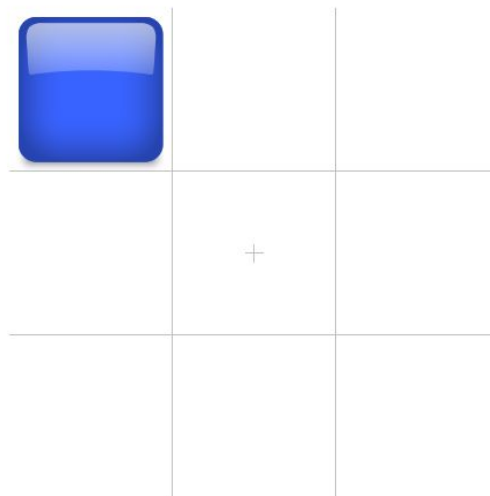
Press SPACE to enter the Workshop

Figura 2.1: Brain Workshop

- P: Pause / Unpause
- F8: Hide / Reveal Text
- ESC: Cancel Session

Dual 1-Back

17 remaining



Left-click or A: position match

Right-click or L: audio match

Figura 2.2: Vista de juego

G: Return to Main Screen

N: Next Game Type

Dual N-Back

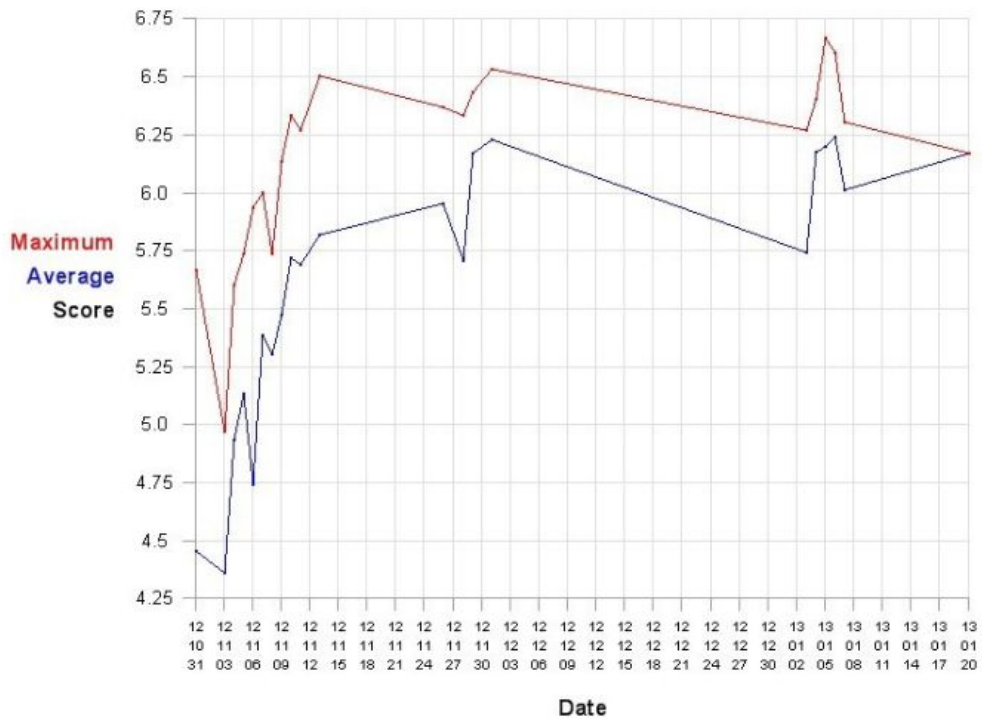


Figura 2.3: Vista de los datos

2. Rules

La jugabilidad básica es que el primer nivel te da una regla, y tienes que elegir las imágenes de acuerdo con ella. Para cada nivel posterior se dará una nueva regla. Después de cumplir esta regla, se pedirá que vuelvas a elegir las imágenes siguiendo la regla del nivel anterior, es decir, la sección de reglas es recursiva, lo que significa que hay una regla más que debe recordarse después de cada nivel.

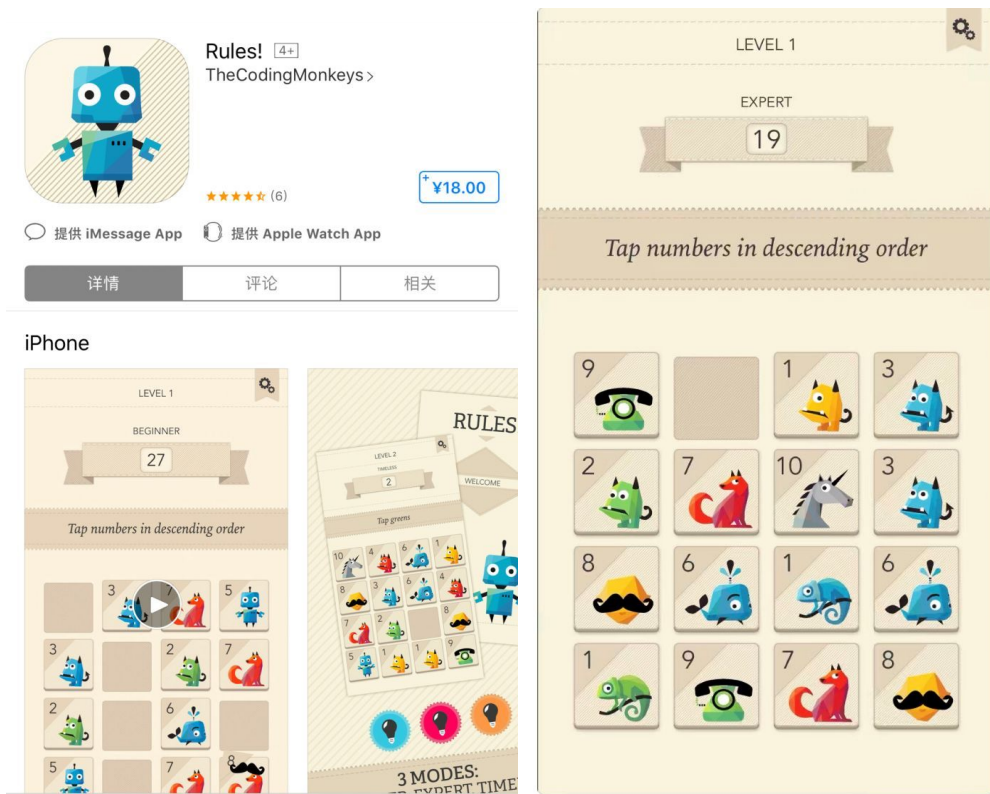


Figura 2.4: App Rules!

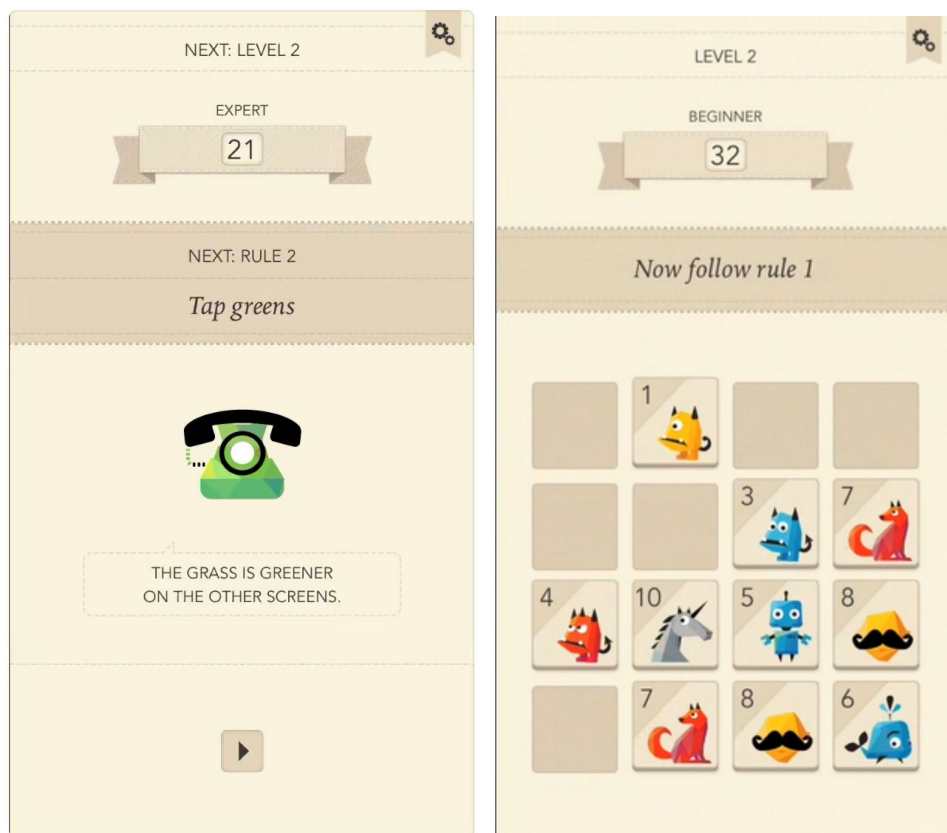


Figura 2.5: App Rules!

3. MEMORY[10]

Este es un juego web que se trata de un juego de memoria de flop de 6 niveles. Requiere que los jugadores volteen las cartas y unan las fichas en parejas, mientras se registra el número de acciones.

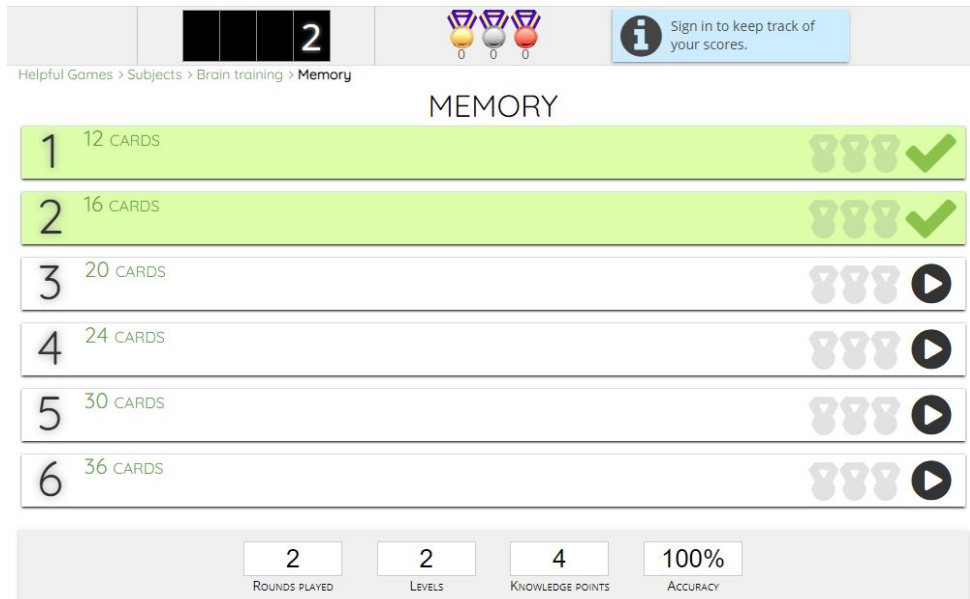


Figura 2.6: Vista principal

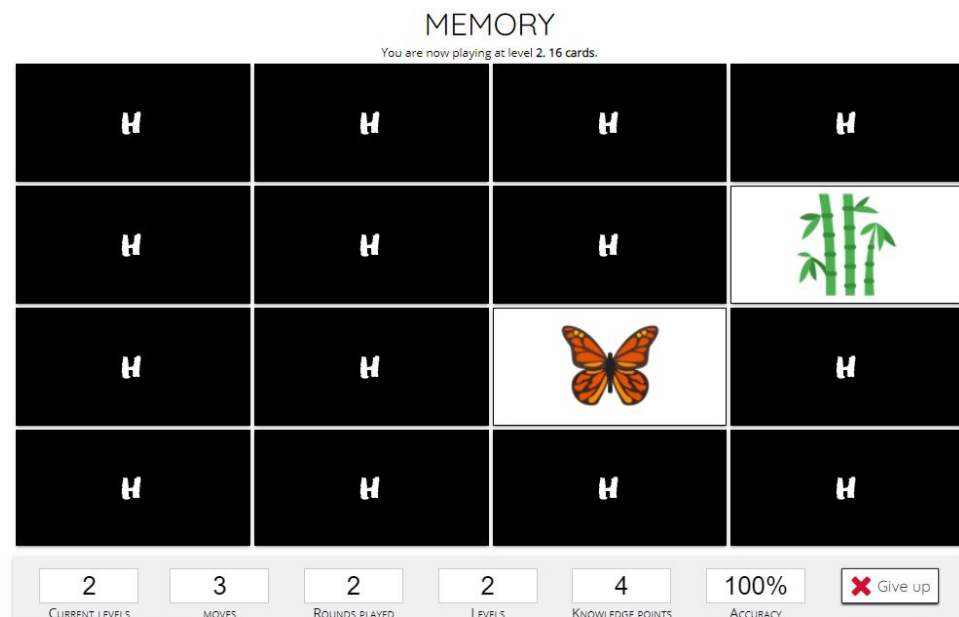


Figura 2.7: Vista de juego

Capítulo 3

Videojuego

En este capítulo explicaremos detalladamente cómo hemos creado el videojuego con el motor de juego Unity. Posteriormente en el apartado de análisis presentaremos los diagramas de casos de uso y de secuencia que definen las relaciones que hay entre el jugador y el sistema. En el apartado de mecánicas del juego describiremos la dinámica de nuestro juego y por último las herramientas que hemos utilizado para el proyecto.

3.1 Análisis

A continuación vamos a ver las especificaciones del comportamiento y las funcionalidades del sistema mediante los diagramas UML.

3.1.1 Diagrama de casos de uso

El diagrama de casos de uso se refiere a la vista utilizada para describir las funciones del sistema, que se compone de actores, casos de uso, límites y las relaciones entre ellos. En la Figura 3.1 muestra los casos de uso del videojuego, así como las funciones principales del sistema.

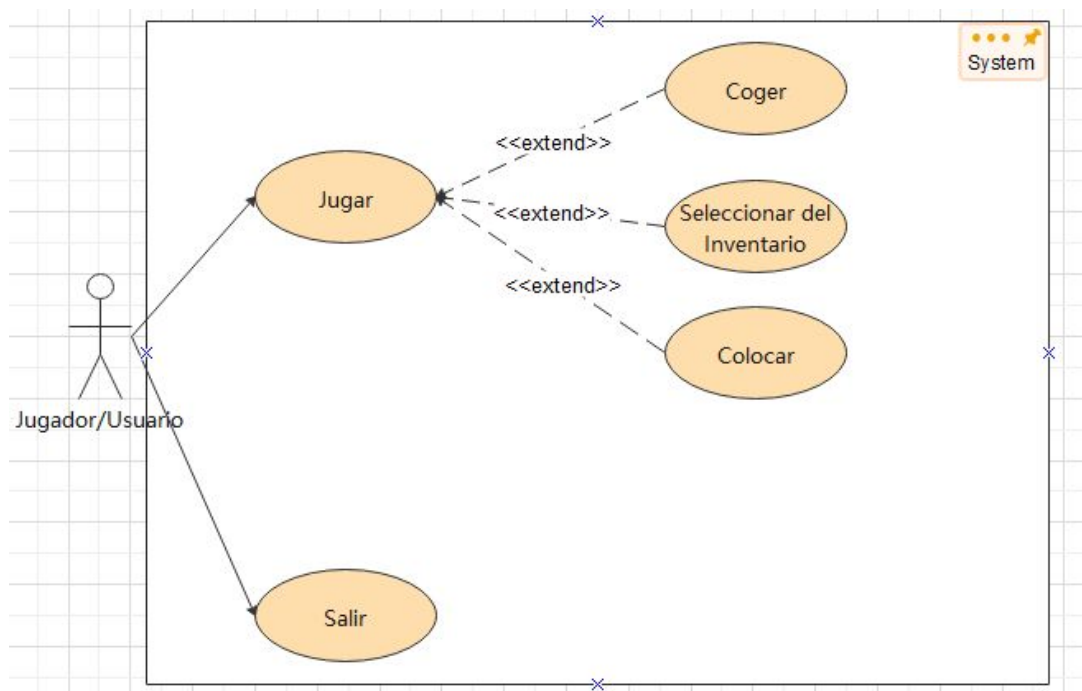


Figura 3.1: Diagrama de Casos de Uso

- Especificación de casos de uso

Nombre de caso de uso	Jugar
Identificador	CU-01
Descripción	El jugador hace clic en el botón "Play" para iniciar el juego.
Actores	Jugador/Usuario
Prioridad	Alta
Precondiciones	-
Postcondiciones	-

Tabla 3.1: CU-01 Jugar

Nombre de caso de uso	Coger
Identificador	CU-02
Descripción	El jugador recoge objetos en la escena para llenar su inventario
Actores	Jugador/Usuario
Prioridad	Alta
Precondiciones	-
Postcondiciones	-

Tabla 3.2: CU-02 Coger

Nombre de caso de uso	Seleccionar del inventario
Identificador	CU-03
Descripción	Selecciona el objeto del inventario que quiera colocarse en su sitio
Actores	Jugador/Usuario
Prioridad	Alta
Precondiciones	-

Postcondiciones	-
------------------------	---

Tabla 3.3: CU-03 Seleccionar del inventario

Nombre de caso de uso	Colocar
Identificador	CU-04
Descripción	Colocar objeto en su sitio
Actores	Jugador/Usuario
Prioridad	Alta
Precondiciones	CU-03
Postcondiciones	-

Tabla 3.4: CU-04 Colocar

Nombre de caso de uso	Salir
Identificador	CU-05
Descripción	El jugador hace clic en el botón "Exit" para salir del juego.
Actores	Jugador/Usuario
Prioridad	Alta
Precondiciones	-
Postcondiciones	-

Tabla 3.5: CU-05 Salir

3.1.2 Diagrama de secuencia

El diagrama de secuencia nos sirve para especificar las interacciones de los distintos objetos del sistema a través del tiempo y se modela para cada caso de uso.

- **Jugar / Salir**

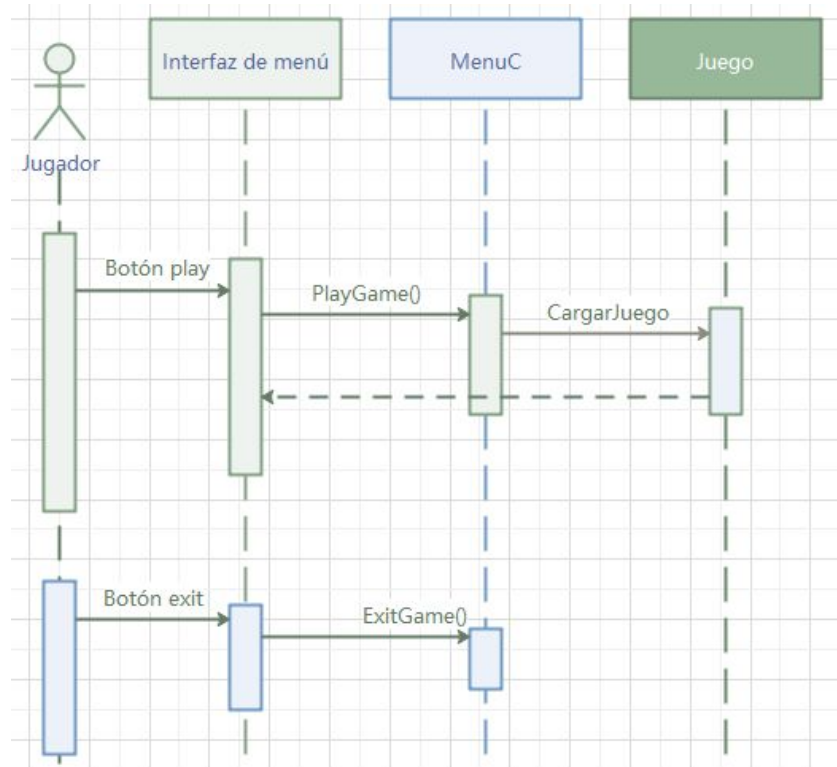


Figura 3.2: Diagrama de secuencia del acto *jugar / salir*

- **Coger**

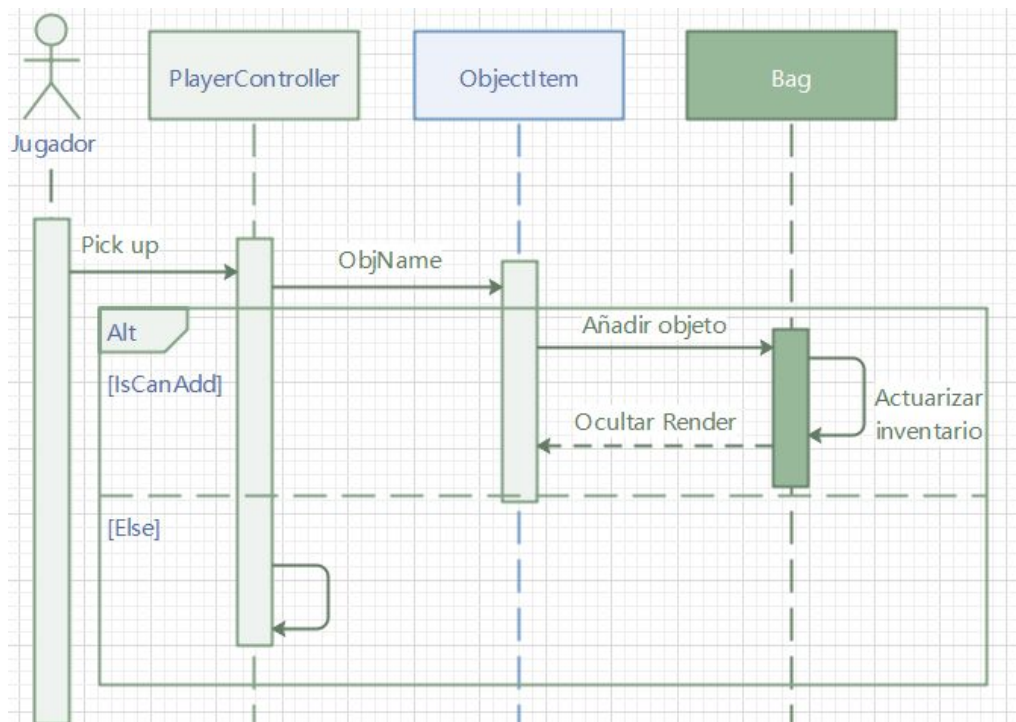


Figura 3.3: Diagrama de secuencia del acto *coger objeto*

- **Seleccionar del inventario**

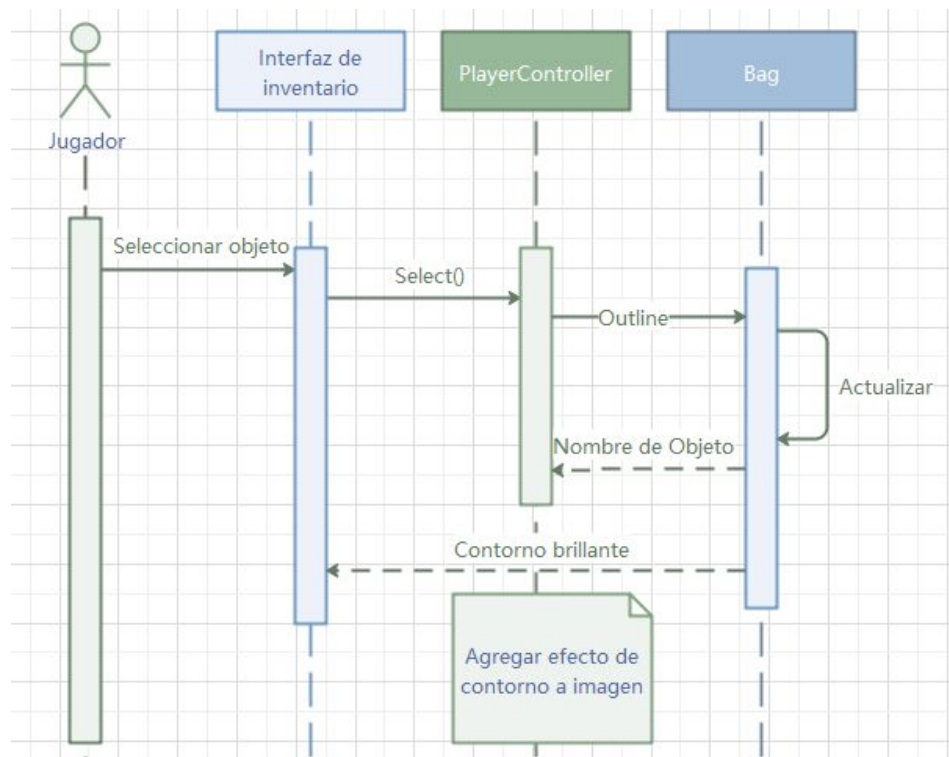


Figura 3.4: Diagrama de secuencia del acto *seleccionar objeto del inventario*

- **Colocar**

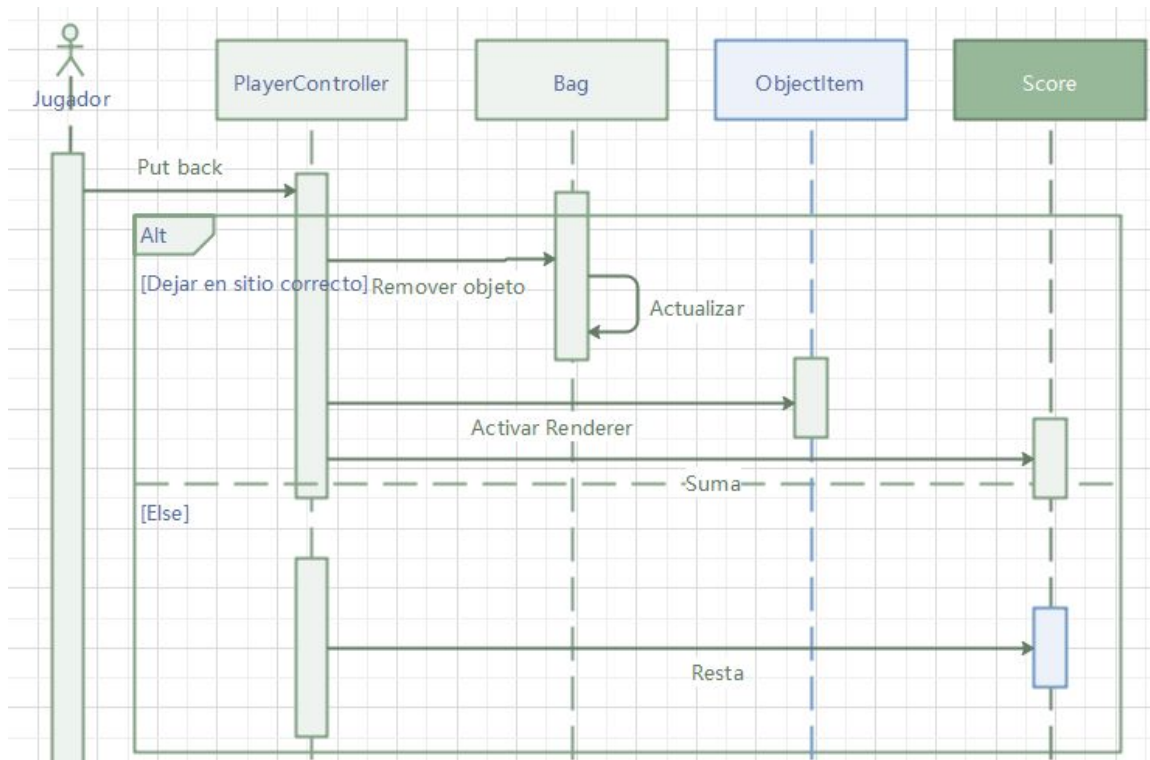


Figura 3.5: Diagrama de secuencia del acto *colocar objeto*

3.2 Creación de videojuego en Unity 3D

Antes de empezar a hacer un juego en Unity, hay que pasar previamente por el proceso de diseño de juego, es el proceso donde se diseña el contenido, las reglas y el sistema del juego.

Unos de los conceptos principales del proceso que hay que tener en cuenta al diseñar el juego son:

- *Tema*: elegir tema del juego es el primer paso y es importante ya que aquí definimos el tipo y la dinámica del juego.
- *Jugabilidad*: la jugabilidad define cómo puede ser el comportamiento del jugador en el juego, las interacciones que puede haber. Se puede decir que es el alma del juego ya que el primer objetivo de un juego es entretener al jugador, por tanto encontrar una jugabilidad buena es la clave.
- *Presentación*: la presentación se trata de mostrar el tema del juego al público así que hay que buscar un estilo gráfico que concuerda con el tema del juego (3D, 2D, pixel art...).

Una vez conseguido todos los anteriores, ya podemos comenzar la creación del juego con Unity, para eso hemos seguido los siguientes pasos hasta conseguir la primera versión jugable :

1. Escenas

Lo primero de todo es crear un proyecto en Unity que a partir del cual podremos establecer diferentes escenas y niveles. El proyecto puede contener diversas escenas, desde la pantalla del menú inicial del juego, la de game over, hasta la construcción de diferentes niveles que irán activando según el avance del juego.

Cuando pruebes el juego en Unity, la vista de escena se actualizará con el estado actual del juego, e incluso puedes agregar nuevos elementos a la escena al instante. Cuando detenga el juego, la vista de escena volverá a su estado original. Cualquier cambio que realice en la vista de escena mientras juegas se perderá cuando se detenga.

En este paso como hemos diseñado 3 niveles de juego por tanto hemos creado 3 escenas (*Figura 3.6 - 3.8*) diferentes para cada nivel. En la sección de asset store disponen de varios modelos que se pueden descargar y utilizar en tu proyecto, en nuestro caso hemos encontrado algunos paquetes de modelos gratis para construir nuestras escenas.



Figura 3.6: Escena 1. Bedroom



Figura 3.7: Escena 2. Kitchen

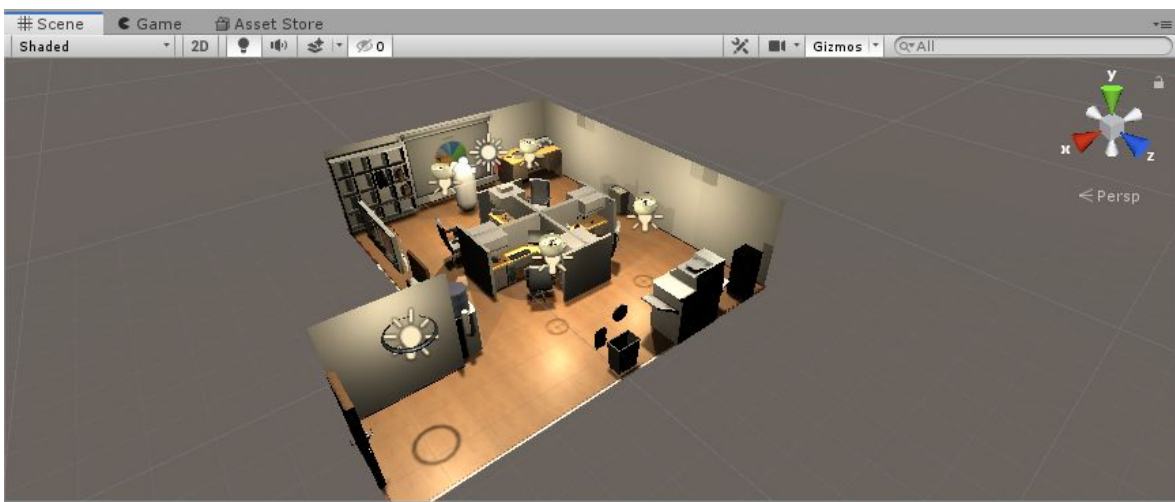


Figura 3.8: Escena 3. Office

2. Colliders

Cuando el personaje del juego se mueve, si encuentra obstáculos como árboles, casas y paredes de roca, en teoría no debería poder avanzar para cumplir con las leyes físicas del mundo real. Por tanto, después de construir las escenas la primera tarea que hay que hacer es establecer colliders (colisionador) en la escena.

Un colisionador es un elemento físico que tiene muchas funciones importantes en el juego, se puede definir su forma pero se recomienda que se ajuste más a la forma del objeto. Cuando agregamos un elemento colisionador al objeto, el personaje controlado por el jugador no podrá pasar a través del objeto y generarán colisión con el objeto. Podemos configurar el collider como un disparador que nos ayuda a determinar cuándo se activa el evento establecido y de esta forma llevaremos a cabo la gran parte de la lógica de interacción del jugador con el entorno que le rodea.

Durante el proyecto la primera dificultad que hemos encontrado es cómo determinar si el jugador ha dejado el objeto en el lugar correcto. La primera idea que se nos ocurrió era guardar las coordenadas del punto central del objeto, y en base a este punto central marcamos un rango de distancia de manera que si el jugador hace clic dentro del rango consideramos que ha colocado bien el objeto en su sitio. Sin embargo hemos encontrado una forma más fácil usando el collider de los objetos con la función de **RaycastHit**. La función de **RaycastHit** se trata de que cuando el usuario hace clic sobre la pantalla se dispara una línea recta infinita desde este punto, durante la trayectoria una vez que la línea colisiona con el primer collider, se dejará de emitir y almacena la información de la colisión. Utilizando esta información podemos detectar el objeto a través de su collider, cuando el jugador recoge el objeto desactivamos el renderizado pero el collider seguirá existiendo, como se muestra en la Figura 3.9, que nos servirá luego para determinar si el jugador está dejando el objeto en su lugar correspondiente.

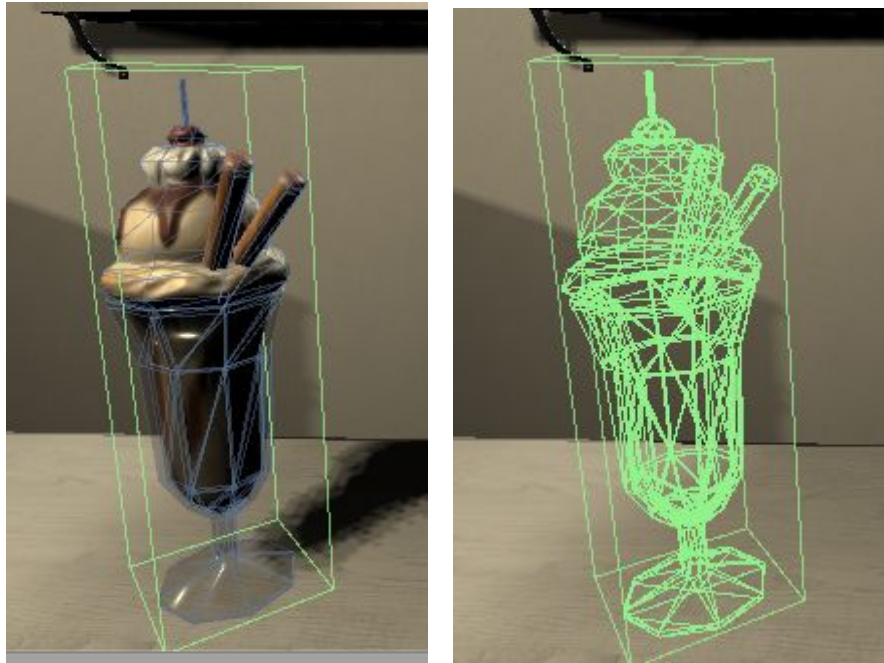


Figura 3.9: Collider con/sin renderer

3. Interfaz de usuario.

La interfaz de usuario es una parte indispensable del juego, siempre se muestra en la parte frontal de la pantalla y no se ve afectada por deformación, colisión o iluminación. La UI puede servir para proporcionar una guía para el juego o informaciones importantes sobre el contenido del juego, y también es un medio importante para embellecer el juego.

Para diseñar la interfaz se requiere canvas como un contenedor para colocar los componentes que se desea mostrar por pantalla. En nuestro juego el jugador tiene que recoger y colocar los objetos dentro de un tiempo determinado mientras evaluamos su tasa de acierto y fallo, por tanto la información básica que se debe mostrar por pantalla incluye un inventario de objetos, un temporizador y la puntuación que se actualiza por cada intento, tal y como se muestra en la Figura 3.10.



Figura 3.10: Interfaz de usuario

4. Scripts

Todos los juegos necesitan script para responder a las acciones de los jugadores y activar los eventos del juego en el momento adecuado. Nos permiten establecer la lógica del funcionamiento asociada tanto a las colisiones que se producen, como para crear efectos gráficos, controlar el comportamiento físico de los objetos e incluso codificar sistemas de inteligencia artificial para los personajes de los juegos.

Este paso es el que llevaría la mayor parte del tiempo ya que básicamente vamos a codificar el juego entero. En Unity3D todos los scripts deben heredar de la clase `MonoBehaviour`, es la clase base de la que deriva cada script de Unity. A continuación vamos a ver las diagramas de clases agrupados por funcionalidades.

- Controladores del juego

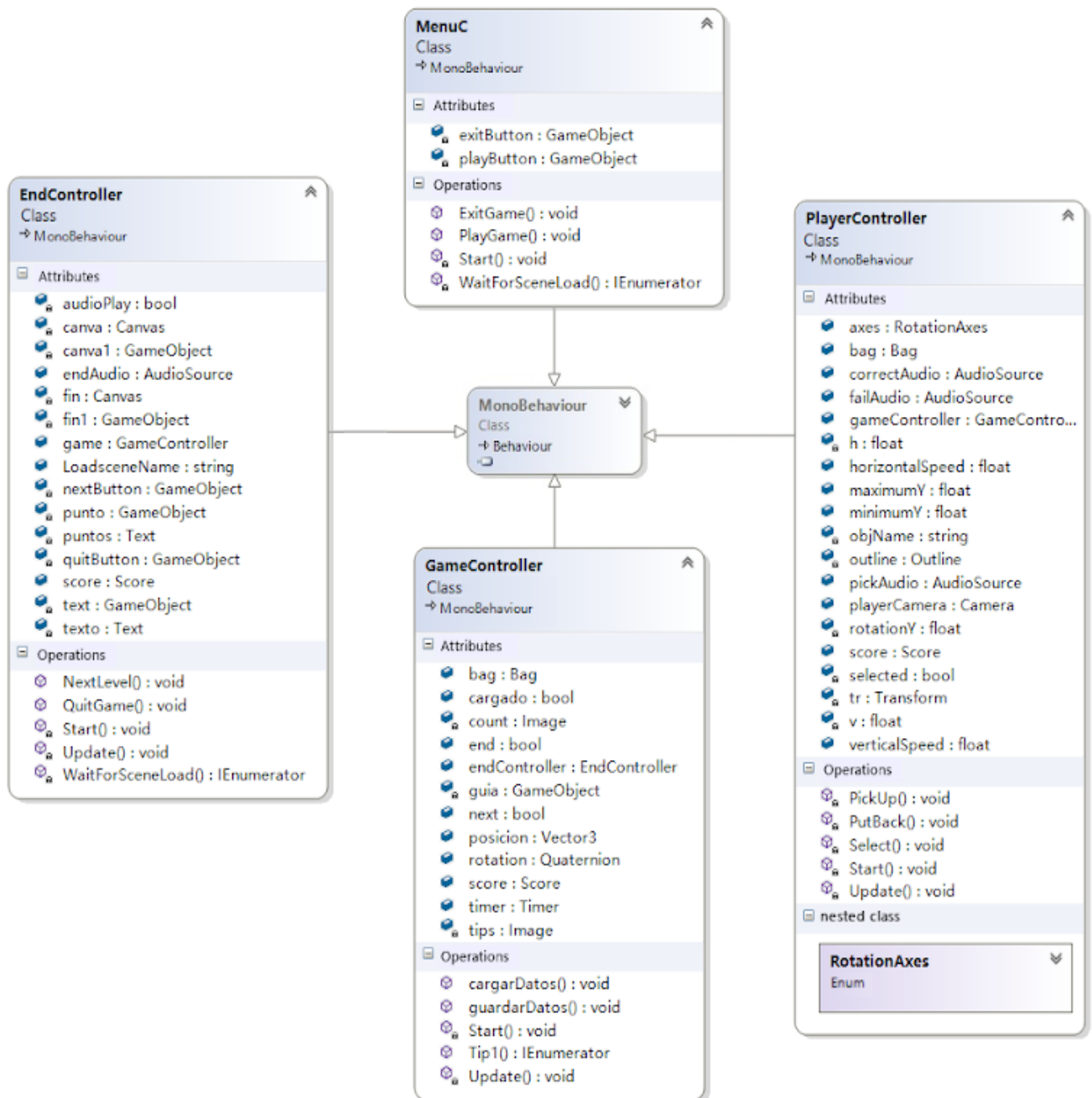


Figura 3.11: Diagrama de clases de los controladores del juego

- Componentes de interfaz

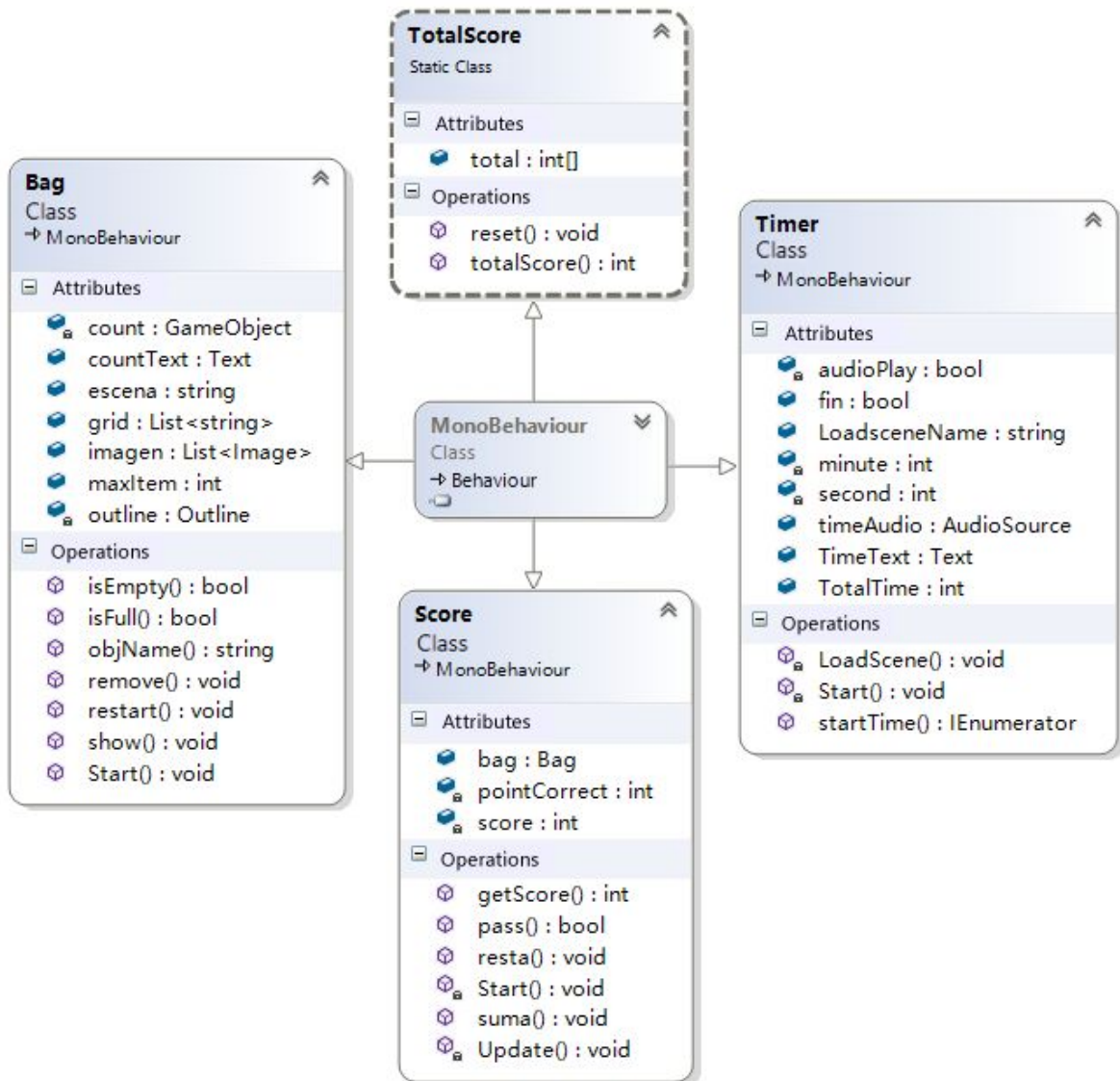


Figura 3.12: Diagrama de clases de los componentes del juego

- Objetos del juego

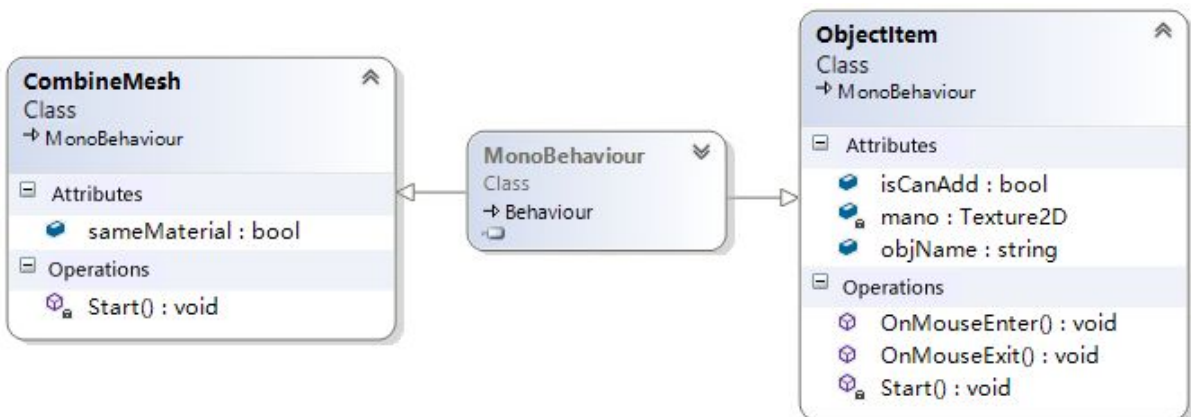


Figura 3.13: Diagrama de clases de los objetos

5. Sonidos

Llegamos al último paso para finalizar el juego, el sonido. El sonido de los videojuegos no solo es para proporcionar un fondo del juego, sino que también afectan el estado de ánimo del jugador, por tanto hay que elegir un estilo de sonido que encaje con las diferentes escenas del juego.

En primer lugar, antes de seleccionar los efectos del sonido del juego, debemos determinar claramente el estilo de juego y las emociones ambientales. En segundo lugar, cuando la música se integra con el juego, debemos prestar atención a la optimización de la música y a la adaptabilidad.

Teniendo en cuenta todo lo anterior, a parte de la música del fondo hemos asignado audios para cada interacción, como pulsar un botón, recoger objetos, cometer un fallo al colocar un objeto, etc.

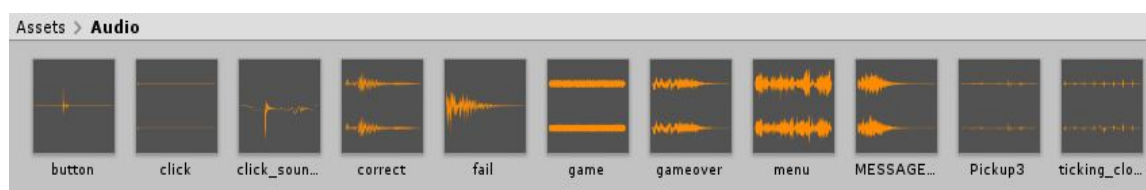


Figura 3.14: Audios del juego.

Hasta aquí hemos conseguido una primera versión jugable de nuestro videojuego, que luego se irán repitiendo algunos pasos para mejorarlo y sacar una versión final.

3.3 Mecánicas del juego

En este apartado vamos a comentar todas las partes sobre la jugabilidad y los mecanismos de gamificación que hemos empleado en el juego My Memory.


- Jugabilidad

My Memory es un juego 3D con el objetivo de evaluar la memoria del usuario. Al inicio del juego el jugador se encuentra en una habitación cerrada mientras que en la pantalla se mostrará la instrucción, como la Figura 3.10, que indica al usuario el objetivo del momento.



Figura 3.15: Instrucción del juego.

La partida consta de dos fases con diferentes objetivos, la primera fase (*Pick Up*) consiste en coger un número determinado de objetos en la habitación y recordar el lugar donde estaban puestos. La segunda fase (*Put Back*) consiste en colocar esos objetos en su sitio. Durante la primera fase el usuario solo puede coger aquel

objeto en el cual aparezca una mano  cuando el cursor pasa por encima, una vez que ha recogido la cantidad de objetos que se le pide, el jugador volverá a la posición inicial y es cuando empieza la segunda fase. En esta fase el usuario tendrá que dejar los objetos en el sitio según lo que recuerda y cada intento se tiene en cuenta para la puntuación.

El juego consta de 3 habitaciones que representan 3 niveles de dificultad, cada nivel pide una cantidad distinta de objetos que tiene que coger y recordar, si en un nivel el usuario no ha llegado a una determinada tasa de acierto ($\frac{1}{3}$ de total de objetos) no podrá pasar al siguiente nivel.

- Mecanismos de gamificación

Para el juego hemos aplicado 3 mecanismos que son puntos, niveles y clasificación:

1. Puntos: los puntos es una de las formas más simple para tener un feedback inmediato de las cosas que hacemos. En el juego, en cada nivel se obtiene una puntuación que se calcula con la fórmula de score y al final del juego se calculará una puntuación final con la que se evaluará la memoria del usuario.

$$score = numCorrect \times \left(\frac{100}{n}\right) - numError \times \left(\frac{100}{2n}\right)$$

numCorrect : número de intentos exitoso (ha colocado el objeto en lugar correcto)

numError : número de intentos fallido

n : número total de objetos

$$totalScore = \sum_{i=1}^k \frac{score_i}{i} \quad 1 \leq k \leq 3$$

k : el nivel que ha llegado el jugador

2. Niveles: los niveles nos permiten establecer distintos retos para el jugador, en este caso hemos establecido una dificultad distinta para cada nivel. La dificultad se refleja en el número de objetos (5 en el nivel 1, 10 en el nivel 2, 15 en el nivel 3), cada vez que pasa al siguiente nivel el jugador tiene que recoger más objetos y acordarse, lo cual cada vez se pide una mayor capacidad de memorizar.
3. Clasificación: con sólo las puntuaciones no podemos extraer una conclusión concreta y clara, por lo que necesitamos una forma de medir el esfuerzo, por ejemplo clasificar las puntuaciones en bajo, medio, alto, etc.
Para ello hemos dividido las puntuaciones en diferentes rangos y una conclusión de evaluación para cada rango.

0 - 30 → "Oops... You've got some work to do... "

31 - 70 → "You have ordinary memorization skills like the rest of us :)"

71 - 99 → "Awesome!! You have very good memorization skills"

100 → "Wooooow!! Incredible... You are the Grand Master of Memory!!"

3.4 Herramientas utilizadas

3.4.1 Unity

Unity3D es una herramienta de desarrollo de juegos multiplataforma creado por Unity Technologies que permite a los jugadores crear fácilmente contenido interactivo como videojuegos 3D, visualizaciones arquitectónicas, animaciones 3D en tiempo real, etc. Es un motor gráfico potente con un entorno de desarrollo amigable para el usuario, suficientemente fácil para el principiante y suficientemente potente para el experto.



Figura 3.16: Logo de Unity

3.4.2 C#

C# es el lenguaje de referencia en Unity, es un lenguaje de programación orientado a objetos derivado de C y C++ lanzado por Microsoft Corporation, un lenguaje de programación de alto nivel que se ejecuta en .NET Framework y .NET Core (código abierto, multiplataforma). Hereda las poderosas funciones de C y C++ mientras elimina algunas de sus características complejas (como no tener macros y no se permite la herencia múltiple).



Figura 3.17: Logo de C#

C# se ve muy similar a Java; incluye procesos como herencia única, interfaces, y tiene casi la misma sintaxis que Java. Pero también existen varias diferencias, como el entorno de desarrollo, aunque C# también es un lenguaje multiplataforma, pero se utiliza principalmente para propio sistema de Microsoft.

3.4.3 Visual Studio



Visual Studio es un entorno de desarrollo lanzado por Microsoft. Es un conjunto de herramientas de desarrollo básico y completo, que incluye la mayoría de las herramientas necesarias a lo largo del ciclo de vida del software, como herramientas UML, herramientas de control de código, entorno de desarrollo integrado, etc.

Figura 3.18: Logo de Visual Studio

Capítulo 4

Evaluación de usuarios

4.1 FeedBack

Para comprobar si el proyecto tuvo éxito la forma más directa es a través del feedback de los usuarios una vez que prueben el juego, ya que la experiencia de usuario es uno de los aspectos más importantes que hay que tener en cuenta a la hora de hacer un videojuego.

- Qiu

Resultados obtenidos del juego:

- 30 (2 niveles completado)
- 36 (3 niveles completado)
- 68 (3 niveles completado)

El diseño gráfico es aceptable, los controles de movimiento son un poco incómodos y lleva tiempo para acostumbrarse. El juego no ofrece un tutorial y empieza el conteo de temporizador desde el momento de la entrada, por lo que los jugadores pueden sentirse difíciles de adaptar cuando empiezan a jugar, y por consecuencia genera una gran brecha entre la experiencia y el puntaje antes y después de familiarizarse con el juego.

Recomendaciones para el juego:

1. Agregar un tutorial simple para principiantes, por ejemplo, enseñar al jugador a avanzar una corta distancia y luego girar el ángulo de visión. Un ejemplo con un libro de la mesa, coloque el mouse sobre el libro y mostrar 'haga clic para recoger', y luego solicite, 'vuelva a hacer clic en el escritorio para recolocar el libro en su lugar '. Así para familiarizarse con el control y la jugabilidad.
2. Cuando el puntaje es bajo, se recomienda agregar una opción para repetir el tutorial.
3. Después de tomar una cantidad suficiente de artículos, cambiar el orden de los artículos en el inventario.
4. Para los objetos con los que se puede interactuar, agregar un efecto de luz cuando pasa por encima el ratón, o que muestre el nombre del objeto para indicar que se puede coger.

5. Optimización adecuada de la determinación de la ubicación de objetos pequeños.

- Wu

Resultados obtenidos del juego:

- 33(2 niveles completado)
- 66(3 niveles completado)

Creo que este juego puede ejercitar mi memoria y mejorar mi concentración en mi tiempo libre. Puede reflejar la capacidad de nuestra memoria hasta cierto punto. A parte de eso, creo que hay margen de mejora en los gráficos del juego, y también se puede añadir más niveles.

- Michelle Guerra

Resultados obtenidos del juego:

- 67 (3 niveles completado)
- 50 (3 niveles completado)
- 84 (3 niveles completado)

Es un juego entretenido, desafía tu capacidad de memoria, la dificultad es adecuada al nivel, e incluso es el propio jugador quien aumenta esa dificultad al juego eligiendo objetos pequeños más difícil de recordar y colocar, por lo que además es un autodesafío. La jugabilidad es buena en cuanto a movilidad por el espacio disponible y tiempo necesario para realizar las acciones que caracterizan el objetivo del juego.

En resumen, está bien implementado, es interesante y adaptable a diferentes jugadores. Para terminar, me gustaría añadir como mejoras que por ejemplo se elimine el cursor de coger objeto (la mano señalando) cuando se están recolocando porque produce un poco de confusión, disminuir quizás un poco el tiempo para recolocar los objetos y por último que haya una opción de salir y no tener que esperar a que se agote el tiempo.

- Javier Ricaldi

Resultados obtenidos del juego:

- 93 (3 niveles completado)
- 88 (3 niveles completado)
- 93 (3 niveles completado)

My memory en cuanto a diseño gráfico está muy bien pensado, niveles con dificultad incremental basados en entornos cotidianos lo cual da una sensación de familiaridad.

En cuanto a la jugabilidad se puede ver que difiere un poco del tipo de control común con respecto a la cámara del jugador, la cual, también tiene un zoom considerable que en ocasiones puede entorpecer el avance en algunos niveles un poco angostos.

Pero en general es fácil acostumbrarse y no hay ningún problema para terminar las distintas pruebas.

En mi opinión My memory cumple bastante bien su función, permitiendo al jugador poder evaluar su capacidad y agilidad de memoria, sin embargo, al ser los niveles estáticos en los cuales los objetos interaccionables siempre aparecen el mismo lugar, provoca que sea solo la primera vez que se prueba el juego cuando se puede dar validez a los resultados.

Como mejora se podría modificar lo anteriormente mencionado añadiendo pseudoaleatoriedad al spawner de los objetos para que cada partida sea distinta, de manera que cada jugada sea válida para extraer nuestros resultados.

- Ying

- + Introducción de juego en el menú.
- + Hay cambio de cursor para los elementos que se puede recoger (pero espero desactivar este punto después entrar al modo de recolocación).
- + Colocación inclinada de bolígrafos y otros artículos pequeños para aumentar el área y mejorar la tolerancia a fallos.
- + La música de fondo es satisfactoria y los efectos de sonido son aceptables.

- El efecto de iluminación aún necesita ajustes.
- Espero que el icono del temporizador se pueda añadir un efecto dinámico.
- Espero que puedas escribir al comienzo de cada nivel la cantidad de artículos que hay que coger, por ejemplo, "in this level you need to pick up 15 objects and then put them back".
- Se recomienda ajustar ligeramente la velocidad de la rotación de vista.
- Añadir un límite al desplegable del inventario de tercer nivel, a veces se deja en blanco la mayor parte al arrastrar hasta fondo.

En cuanto a la memoria:

Debido a la gran cantidad de información recibida por el ojo humano y el poder de procesamiento y la memoria limitados del cerebro, muchas veces descubriremos que nuestro cerebro aplica un modo selectivo con las informaciones, olvidando aquellas que no se consideran importantes: por ejemplo, recordar la cara de un conocido pero no la longitud de su cabello, ver un coche pero al pasar ya se nos olvida su número de matrícula, etc.

My Memory es una prueba de la capacidad de la persona para recordar información de ubicación específica. Por ejemplo, en el tercer nivel que es la habitación que más me impresiona, en la pequeña mesa cerca de la ventana hay macetas, bolígrafos, papeles y teléfonos. Puedo recordar que esta secuencia comienza desde la ventana e inconscientemente puse la maceta en la derecha del todo, aunque en realidad su posición original es un poco más a la derecha del centro. Incluso al tomar los elementos en orden y volver a ponerlos sin tener que memorizar su orden original, me resulta difícil recordar la ubicación exacta de cada objeto (también por la razón de que después de quitar todos los objetos de escritorio, pierdo la referencia).

4.2 Comparación

A parte de recibir el feedback de los usuarios también hemos buscado algunos tests validados de memoria para tomarlos como referencia y hacer una comparación de resultados, con el objetivo de ver si el resultado de la evaluación del juego aporta algo de validez. Para ello hemos reclutado 15 voluntarios con un edad promedio de 22.73 años que, a parte de nuestro juego, realicen también un test de imagen que se basa en el test de retención visual de Benton[12] y test de memory Malingering (TOMM[13]) pero en versión simplificada que se puede realizar solo, ya que por la situación que estamos ahora no podemos realizar en persona estos test completos.

El test consta de 10 imágenes, cada imagen se mostrará en la pantalla durante 2 segundos. En este tiempo se debe recordar todas las características de la imagen. Al desaparecer la imagen se mostrará por pantalla durante 5 segundos 4 opciones que entre las cuales hay que elegir la correcta, tal y como se muestra en la Figura 4.1.

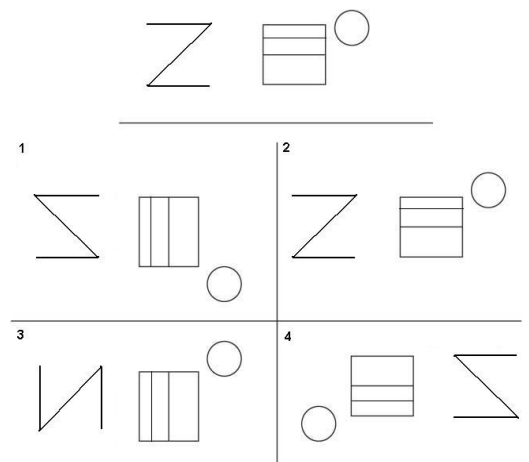


Figura 4.1: Test de imagen



Figura 4.2: Comparación de los resultados

Respecto a la gráfica de la Figura 4.2 podemos observar que en ciertos puntos se asemeja la tendencia, pero no es suficiente para decir que nuestro juego evalúa la memoria al 100%, dado que nos falta realizar más análisis para obtener un método adecuado de medir la memoria, como por ejemplo tener en cuenta también la edad, el tiempo de reacción, etc. Además, la característica 3D del juego aumenta en cierta parte la dificultad en el juego, por ejemplo a la hora de recolocar los objetos de tamaño pequeño, se pide una mayor precisión al jugador, lo que provoca un posible aumento en la tasa de fallos.

Capítulo 5

Contribuciones individuales

- 5.1 Yi Teng Wang

Como hemos explicado anteriormente en la planificación del trabajo, la primera fase del proyecto se trató de investigar y practicar. Tanto las informaciones sobre el tema de nuestro proyecto como las herramientas que íbamos a utilizar.

En la primera reunión con nuestro tutor *Iván García*, nos comentó la idea principal del proyecto y nos aconsejó que empezáramos familiarizándonos con la herramienta Unity 3D. De ahí mi compañera y yo estuvimos buscando informaciones y juegos relacionados con el tema de evaluación de memoria y mientras practicamos con los tutoriales cada una por su cuenta hasta tener una base de conocimiento suficiente para poder empezar el desarrollo del proyecto. Hasta aquí completamos nuestra primera fase.

Tras tener el visto bueno por parte de nuestro tutor sobre nuestro plan de diseño del proyecto después de unas reuniones, empezamos a montar las escenas del juego. Para eso hemos iniciado las dos una búsqueda de assets que sirvieran para las escenas hasta terminar la construcción de escenas principales. Este proceso la mayor parte del tiempo lo hemos llevado a cabo con los ordenadores de facultad ya que Unity 3D consume muchos recursos y con nuestro portátil no va muy fluido.

A continuación llega la parte de scripts, en un principio hemos repartido unas tareas, pero dado que es algo nuevo para ambas y con el fin de aprender más, las dificultades que hemos encontrado a lo largo las hemos resuelto juntas así que ambas tareas han sido completadas con la ayuda de la otra en la facultad. Posteriormente por el tema de la Covid-19 hemos tenido que acabar el proyecto en casa, una vez descubierto que la interfaz gráfica del juego se abre distinto en el portátil de cada una (no hemos encontrado el motivo), decidimos realizar el resto de tareas en mi ordenador. De manera que nos reunimos por Skype para hablar de las ideas, mi compañera se encargaba de buscar los recursos para el juego y yo me encargué de añadirlos y escribir el código a la vez.

Una vez terminado el juego comenzamos con la fase del test, durante esta fase he sido la encargada de reportar los “*bugs*” del juego y mi compañera ha sido la encargada de corregir esos “*bugs*” hasta comprobar el funcionamiento correcto del juego.

Finalmente junto a mi compañera hemos elaborado esta memoria repartiendo los apartados y la revisión de la misma.

- 5.2 Yifei Liu

Antes de empezar este proyecto, apenas sabíamos cómo usar Unity para desarrollar juegos, así que en la etapa inicial del proyecto, hemos descargado Unity y hemos seguido los tutoriales de la web oficial para practicar algunas de las características que podemos desarrollar más adelante. Después de completar algunos juegos básicos, fuimos a buscar informaciones de memoria, pruebas o minijuegos en la biblioteca de la universidad y, en base a estos, nos surgió 2-3 ideas y un plan inicial del proyecto. Finalmente hemos determinado el plan de desarrollo con nuestro tutor en la reunión.

En la segunda etapa, comenzamos a buscar modelos adecuados para nuestro juego en Unity asset store. Para hacer este juego más cercano a la vida diaria, la configuración inicial de las escenas fue: dormitorio, cocina y supermercado. Más tarde, al construir el dormitorio, descubrimos que al tener diferentes versiones de Unity provocaba que el proyecto no se iniciase con normalidad, por lo tanto tuvimos que actualizar el proyecto a una versión anterior para poder iniciarlo. El proceso de construcción llevará mucho tiempo, por tanto decidí construir el dormitorio en mi portátil y luego llevar el proyecto al laboratorio, donde se actualizó la versión a Unity 2019.2.1f1 (versión unificada) y también hemos construido un modelo de cocina con mi compañera en el laboratorio. Cuando nos estábamos preparando para construir un supermercado, no había suficiente número de objetos y había demasiada duplicación con la cocina, por lo que en el último escenario hemos cambiado a una oficina y hemos agregado varios objetos similares para aumentar la dificultad.

La siguiente parte es principalmente escribir script. Lo primero es agregar la clase `ObjectItem` para los objetos que se pueden coger, esta parte también la hemos terminado en cooperación, para que cada una de nosotras tenga la oportunidad de familiarizarse con Unity y nuestro proyecto. En el laboratorio de la universidad, mi compañera y yo hemos completado la transformación

de cursor del jugador, con su movimiento agregado y hemos terminado el componente de inventario de la interfaz de usuario y el temporizador.

Posteriormente por tema de la Covid-19, tuvimos que hacer las reuniones y terminar el proyecto a través de Skype, y luego descubrimos que múltiples transmisiones del proyecto causan el cambio de los componentes del juego en el panel del usuario. Así que decidimos unificar el código que afectaba al panel en el ordenador de Yi Teng.

Después de realizar la mayoría de las funciones, debido a que faltan colisionadores algunos elementos descargados del asset store, me he encargado de crear CombineMesh para combinar mesh y el material de dos o más elementos.

Al mismo tiempo, hemos terminado de hacer que se muestre el contorno del objeto cuando lo elegimos en el inventario. Y por último hemos tomado capturas de pantalla a nuestra habitación y la hemos editado como fondo. Después de eso, hemos estado buscando y compartiendo recursos de sonido libre de derechos de autor. Hasta aquí básicamente está terminado el proyecto, por lo que el siguiente paso que hemos hecho es realizar pruebas para encontrar bugs...

Finalmente, hemos empaquetado el proyecto y hemos generado el archivo .exe, y hemos completado la memoria a través de Google Drive.

Capítulo 6

Conclusiones y trabajo futuro

6.1 Conclusión

Una vez finalizado el proyecto llega la hora de hacer una valoración del trabajo que hemos conseguido y de lo que hemos aprendido.

El objetivo principal de este proyecto era desarrollar un juego con Unity 3D que evalúa la memoria del usuario, lo cual creemos que está cumplido puesto que, por un lado, sí requiere suficientes características para ser un juego, y por otro lado, los resultados obtenidos del juego si que aportan cierta validez que se puede observar mediante la comparación que hicimos con otro test de memoria válido.

Este proyecto nos resultó un reto por la inexperiencia que teníamos en este campo y la herramienta Unity, que a pesar de que hemos dedicado tiempo para aprender usar Unity al principio del cuatrimestre, la falta de conocimientos nos condujo a cometer varios errores durante el desarrollo que nos llevó tiempo en buscar soluciones y corregirlos. Pero igual estamos satisfechas ya que el aprendizaje es un proceso de prueba y error, y de los errores se aprenden más.

En resumen, aunque hemos encontrado muchas dificultades con el proyecto, principalmente con la herramienta, también nos ha ofrecido una oportunidad de poner en práctica lo que hemos aprendido en la carrera y adquirir nuevos conocimientos.

6.2 Trabajo futuro

Hasta ahora, nuestro juego se ha completado básicamente, pero algunas funciones no son perfectas o pueden mejorarse. También queda espacio para añadir más cosas al juego y llevarlo a otro nivel. A continuación vamos a explicar unos problemas actuales que existe en el proyecto y unas ideas de trabajo futuro, así como unas herramientas que se podría utilizar para llevarlo a cabo.

- **Problemas actuales**

1. Hay algunos pequeños problemas en el empalme de la pared prefabricada en la escena en el que se puede ver un ligero destello de luz. El jugador puede caerse de la habitación cuando camina por cierto ángulo o dirección. Dado que la probabilidad de este evento no es alta

e implica cambios en la posición de muchos elementos en la escena, no hemos solucionado del todo este problema en esta etapa.

2. El juego se ejecuta en modo ventana y tiene un tamaño fijo. Durante las pruebas del usuario, descubrimos que en algunas computadoras la ventana de visualización es demasiado pequeña o la interfaz del juego no se muestra completamente. Así que una parte por mejorar en un trabajo futuro es hacer que la ventana del juego se pueda adaptar automáticamente a la computadora del jugador.
3. Hace falta hacer más análisis en cuanto a la medición de memoria, por ejemplo, la edad de las personas que hicieron la prueba en la etapa actual está relativamente concentrada, lo que no es suficiente para reflejar el 100% de validez de la evaluación de memoria de nuestro juego.

- **Jugabilidad**

Añadir la aleatoriedad a la posición de los objetos para que cada partida sea distinta. Los objetos en el inventario que no se muestran en orden de recogida, se pueden ordenar por orden alfabético para quitar la posibilidad de que los jugadores memoricen con la ayuda del orden (para aumentar dificultad).

- **Versión móvil del juego**

Teniendo en cuenta la conveniencia y la frecuencia de uso de los teléfonos móviles, se podría sacar una versión móvil del juego, que habrá que hacer unos cambios como el control de movimiento, que pasa de teclado a la pantalla táctil. Hay dos herramientas que pueden servir para realizar este trabajo:

- El software **UnityRemote**[\[14\]](#) permite conectar el dispositivo con Unity mientras ejecutas tu proyecto, a través de esta herramienta puedes saber cómo se ve realmente tu juego y cómo se maneja en el dispositivo.

UnityRemote se instala en el móvil, luego hay que conectar el dispositivo con el PC, una vez conectado se ejecuta UnityRemote. Tras activar el software en el PC hay que iniciar Unity para ejecutar el juego, la salida visual se enviará a la pantalla del dispositivo, y las entradas en vivo se enviarán de vuelta al proyecto en ejecución en Unity, de esta manera se podrá jugar en el dispositivo aunque el juego no esté instalado en el móvil.



Figura 6.1: Unity Remote 5

- Unity tiene una herramienta llamada **Unity Asset Server**[15], es un sistema de control de versiones muy conveniente para el desarrollo de proyecto grupal. Por ejemplo, en un equipo pequeño, puedo usar mi propia computadora como servidor y crear un nuevo proyecto vacío en el servidor, cargar mi proyecto actual en el servidor y luego configurar los permisos de usuario para que puedan acceder al proyecto. Y junto con los otros miembros del equipo podremos desarrollar el proyecto ya sea, trabajando en el sitio o de forma remota.

- **Almacenamiento de datos**

El almacenamiento local de los datos de desarrollo de Unity también es una función básica importante. Para los juegos de un solo jugador, los datos deben almacenarse localmente. Los juegos online pueden almacenarse en el servidor, pero algunos datos deben almacenarse localmente, como la cuenta de los jugadores.

 1. Unity proporciona una solución de almacenamiento de datos persistentes: PlayerPrefs, que puede satisfacer las necesidades de nuestro almacenamiento básico de cuentas, pero su defecto también es obvio, el tipo de datos almacenados es limitado, solo se pueden almacenar algunos tipos de datos muy básicos.
 2. La base de datos SQLite requiere complementos externos. Habría que importar mono.data.sqlite.dll, System.data.dll, sqlite3.dll en la carpeta Assets/Plugins
 3. EasySafe 3: Más rápido que PlayerPrefs, puede guardar casi cualquier tipo de dato, no hay que preocuparse por la serialización, su capacidad de multiplataforma es muy potente y solo necesita un complemento, lo

que evita problemas de desarrollo y compatibilidad redundantes por el reemplazo de la plataforma.



Figura 6.2: Base de datos

Conclusions and future work

6.1 Conclusions

Once the project is finished, it is time to make an assessment of the work that we have achieved and what we have learned.

The main objective of this project was to develop a game with Unity 3D that assesses the user's memory, which we believe is fulfilled since, on the one hand, it does require enough features to be a game, and on the other hand, the results obtained of the game provide a certain validity that can be observed through the comparison we made with another valid memory test.

This project was a challenge for us due to the inexperience we had in this field and the Unity tool, which despite the fact that we have spent time learning to use Unity at the beginning of the term, the lack of knowledge led us to make several mistakes during the development that it took us time to find solutions and correct them. But we are still satisfied because learning is a process of trial and error, and we can learn more from our mistakes.

In summary, although we have encountered many difficulties with the project, mainly with the tool, but it has also offered us an opportunity to put into practice what we have learned in the college and acquire new knowledge.

6.2 Future work

So far, our game is basically complete, but some features are not perfect or can be improved. There's also room to add more to the game and take it to another level. Next we will explain some current problems that exist in the project and some ideas for future work, as well as some tools that could be used to carry it out.

- **Actual problems**

1. There are some small problems in the precast wall splice in the scene where you can see a slight flash of light. The player may fall out of the room when walking at a certain angle or direction. Since the probability of this event is not high and involves changes in the position of many elements in the scene, we have not fully solved this problem at this stage.

2. The game runs in windowed mode and has a fixed size. During user testing, we discovered that on some computers the display window is too small or the game interface is not fully displayed. So a part to improve in a future job is to make the game window automatically adapt to the player's computer.
3. More analysis is needed in terms of memory measurement, for example, the age of people who took the test at the current stage is relatively concentrated, which is not enough to reflect 100% validity of the memory assessment of our game.

- **Gameplay**

Add randomness to the position of the objects so that each game is different. Items in inventory that are not displayed in pickup order can be sorted alphabetically to remove the ability for players to memorize with the help of order (to increase difficulty).

- **Mobile version of the game**

Considering the convenience and frequency of use of mobile phones, a mobile version of the game could be released, which will have to be changed such as the movement control, which goes from the keyboard to the touch screen. There are two tools that can be used to carry out this work:

- The UnityRemote software allows you to connect the device with Unity while executing your project, through this tool you can know how your game really looks and how it is handled on the device.

UnityRemote is installed on the mobile, then the device must be connected to the PC, once connected UnityRemote runs. After activating the software on the PC, Unity must be started to run the game, the visual output will be sent to the device screen, and the live inputs will be sent back to the project running in Unity, so it can be played in the device even if the game is not installed on the mobile.

- Unity has a tool called Unity Asset Server, it is a very convenient version control system for group project development. For example, on a small computer, I can use my own computer as a server and create a new empty project on the server, upload my current project to the server, and then configure user permissions so that they can access the project. And together with the other team members we can develop the project either by working on site or remotely.

- **Data storage**

Local storage of Unity development data is also an important basic function. For single player games, the data must be stored locally. Online games can be stored on the server, but some data must be locally from the players, like the players account.

1. Unity provides a persistent data storage solution: PlayerPrefs, which can meet the needs of our basic account storage, but its flaw is also obvious, the type of data stored is limited, only some very basic data types can be stored.
2. The SQLite database requires external plugins. It would have to import mono.data.sqlite.dll, System.data.dll, sqlite3.dll in the Assets / Plugins folder.
3. EasySafe 3: Faster than PlayerPrefs, can save almost any type of data, no need to worry about serialization, you only need a plug-in and its cross-platform capacity is very powerful, which avoids development problems and redundant platform compatibility

Bibliografía

- [1] Cowan, N. (2008). What are the differences between long-term, short-term, and working memory?. *Progress in brain research*, 169, 323-338.
- [2] Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological review*, 102(2), 211.
- [3] Michael C. Levin. (2019). *Memory Loss - Brain, Spinal Cord, and Nerve Disorders* - Medicine.com.
- [4] 朱磊, & 杨治良. (2003). 多种记忆分类之研究 (Doctoral dissertation).
- [5] Fassbender, E., & Heiden, W. (2006). The virtual memory palace. *Journal of Computational Information Systems*, 2(1), 457-464.
- [6] Dresler, M., Shirer, W. R., Konrad, B. N., Müller, N. C., Wagner, I. C., Fernández, G., ... & Greicius, M. D. (2017). Mnemonic training reshapes brain networks to support superior memory. *Neuron*, 93(5), 1227-1235.
- [7] González-Landero, F., García-Magariño, I., Amariglio, R., & Lacuesta, R. (2019). Smart cupboard for assessing memory in home environment. *Sensors*, 19(11), 2552.
- [8] Hampstead, B. M., Sathian, K., Moore, A. B., Nalisnick, C., & Stringer, A. Y. (2008). Explicit memory training leads to improved memory for face–name pairs in patients with mild cognitive impairment: Results of a pilot investigation. *Journal of the International Neuropsychological Society*, 14(5), 883-889.
- [9] Hoskinson, P. (2018). *Brain workshop-a dual n-back game*.
- [10] Games, H., & training, B. (2020). *Memory: Free online game (Brain training) - Helpful Games*.
- [11] Creación de videojuego en Unity 3D (I): estructura y elementos – Academia Android. (2020). Retrieved 31 May 2020, from <https://academiaandroid.com/creacion-videojuego-unity3d-estructura-y-elementos/>
- [12] Benton, A., & Benton Sivan, A. (1992). *Benton visual retention test*. San Antonio [etc]: The Psychological Corporation.

- [13] Tombaugh, T. N. (1997). The Test of Memory Malingering (TOMM): Normative data from cognitively intact and cognitively impaired individuals. *Psychological Assessment*, 9(3), 260.
- [14] Kelly, S. (2016). Unity Remote 5 Setup and First Test (Multi-Touch). Viitattu 14.03. 2018.
- [15] Technologies, U. (2020). Unity - Manual: Asset Server (Team License). Retrieved 31 May 2020, from <https://docs.unity3d.com/Manual/AssetServer.html>
- [16] Felicia, P. (2013). Getting started with Unity. Packt Publishing Ltd.

Apéndice A

Vistas del juego

- Menú

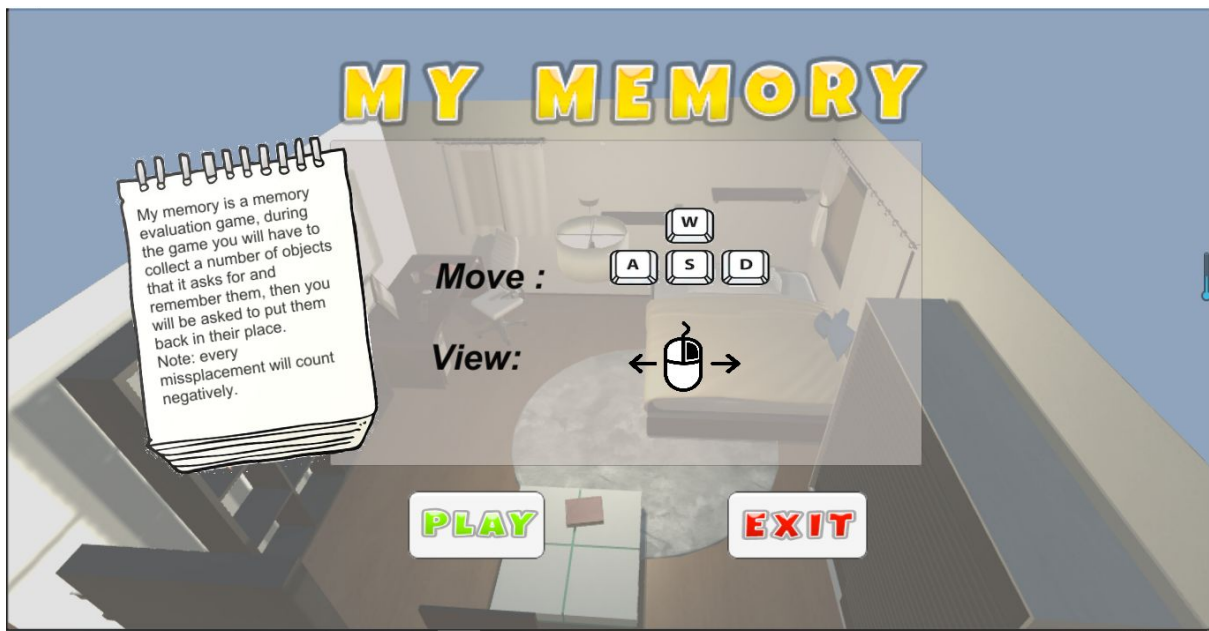


Figura 1: Menú del juego

- Nivel 1. Bedroom



Figura 2: Instrucción en fase 1 (Pick Up)



Figura 3: Instrucción en fase 2 (Put Back)

- Segunda habitación (nivel 2)



Figura 4: Vista de la habitación de nivel 2



Figura 5: Vista de la habitación de nivel 2

- Nivel 3.Office



Figura 6: Vista de la habitación de nivel 3



Figura 7: Vista de la habitación de nivel 3

- Pantalla de finalización

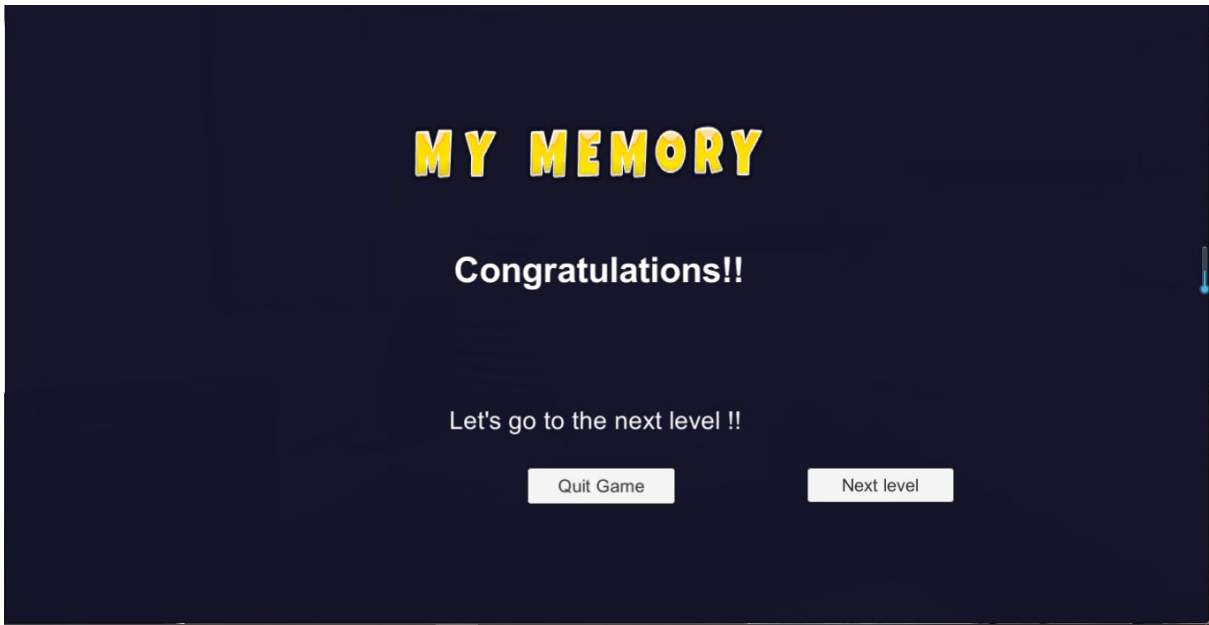


Figura 8: Vista de nivel completado

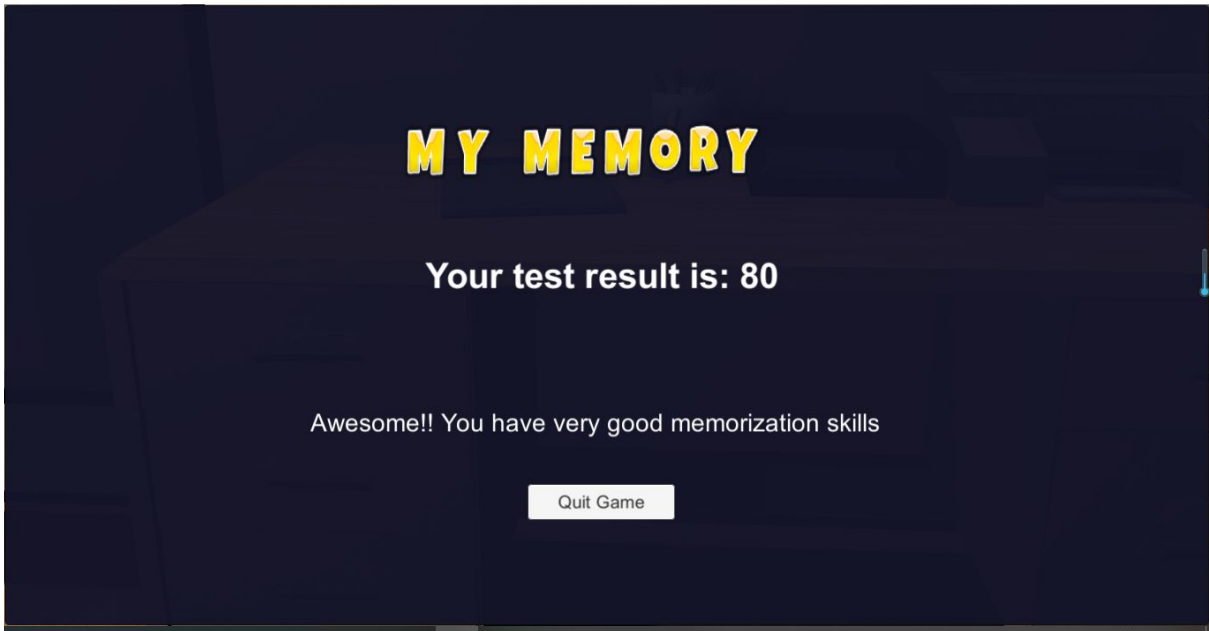


Figura 8: *Vista de resultado final*