

UNIVERSIDAD TECNOLÓGICA DE SANTIAGO UTESA

Facultad de Ingeniería y
Arquitectura Carrera de
Informática



Asignatura:

Videojuegos

Tema:

Entrega Completa

Presentado a:

Iván Mendoza

Presentado por:

Marco Molina Disla 2-17-1509

INTRODUCCION

En el actual trabajo desarrollaremos la documentación de un videojuego del tipo Arcade donde habrá que superar distintos obstáculos y obtener monedas hasta ganarlas, de modo que se documentará todo el proceso de creación.

CAPITULO I: VIDEOJUEGO Y HERRAMIENTAS DE DESARROLLO

1.1 Descripción

El juego a mostrar llamado Laberinto es un juego del tipo arcade, el cual se trata de superar el laberinto tomando todas las monedas en el escenario y así mismo esquivando los diferentes obstáculos en movimiento.

1.2 Motivación

1.2.1 Originalidad

La idea surge de los distintos videojuegos de laberintos que existen en el mercado de forma gratuita, pero agregándole obstáculos y recoger todas las monedas del mapa. Estos tipos de videojuegos en su momento tuvieron una gran audiencia, pero con el tiempo pocas personas les interesaba, de forma que este se realizó para agregarle otros objetivos más interactivos.

1.2.2 Estado del arte

Este busca entretener a los usuarios con sus diferentes formas intuitivas a lo largo de la historia del videojuego.

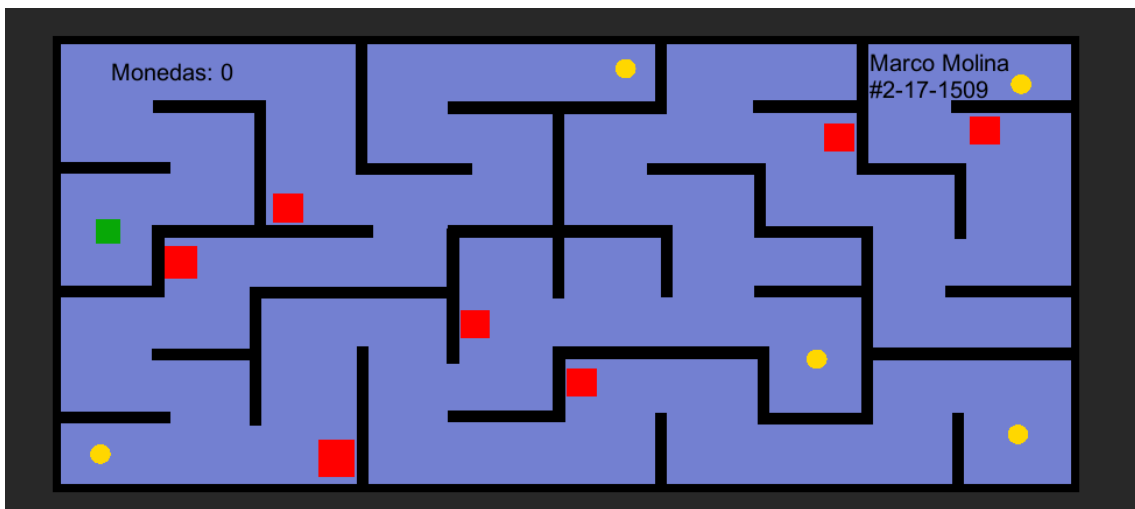
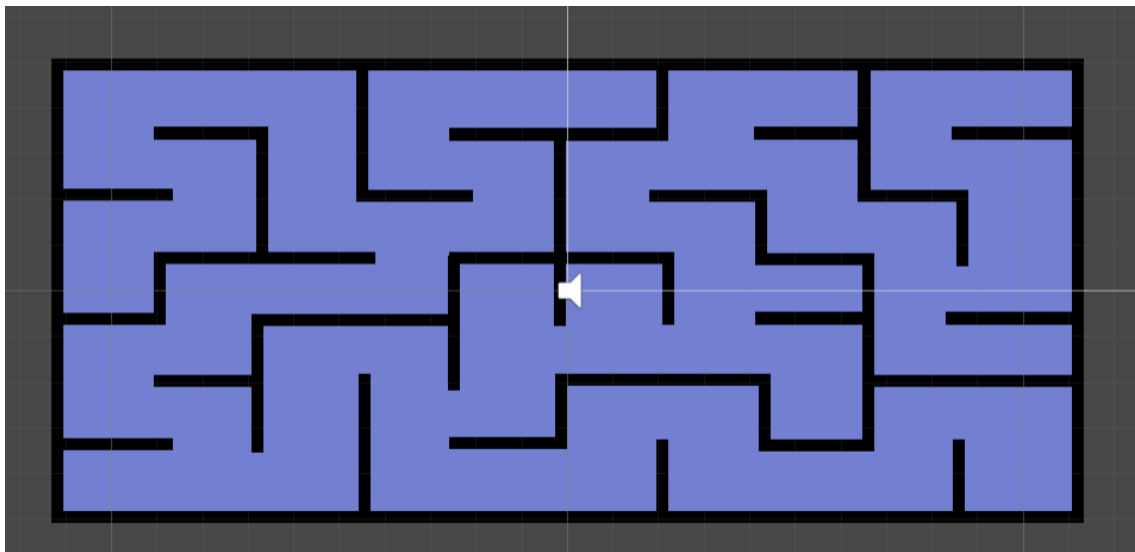
1.3 Objetivo general

El objetivo principal del juego es cruzar el laberinto obteniendo las monedas dispersas a lo largo del mapa, esquivando obstáculos para lograr la victoria.

1.4 Objetivos específicos

- Conseguir las monedas.
- Esquivar obstáculos.
- Superar el laberinto.

1.5 Escenario



1.6 Contenidos

Se presentará un videojuego 2d arcade de laberintos en el cual el jugador tendrá que superarlo obteniendo las monedas y evitándolos los distintos obstáculos que si los toca se reiniciara el nivel, de modo que si obtiene todas las monedas este ganara. Se usará una música desde el inicio del juego hasta el final.

1.7 Metodología

Cada moneda estará en un punto diferente del mapa, donde aumentara la dificultad en cada uno de ellas, el juego se iniciará desde que se cargue y finalizará al recoger todas las monedas del mapa.

1.8 Arquitectura de la aplicación

El juego va dirigido a las PC con un navegador compatible con unity, ya sea Google Chrome, Microsoft Edge, Mozilla Firefox, etc. El videojuego se baso en esta plataforma ya que es bastante fácil de desarrollar en este habiendo y en donde el rendimiento es mucho mayor, no se van a requerir ningún tipo de licencia ni suscripción y todo tipo de persona puede jugarlo si tiene uno de estos navegadores en su PC.

1.9 Herramientas de desarrollo

Las herramientas a utilizar para el desarrollo son: Unity y Visual Studio.

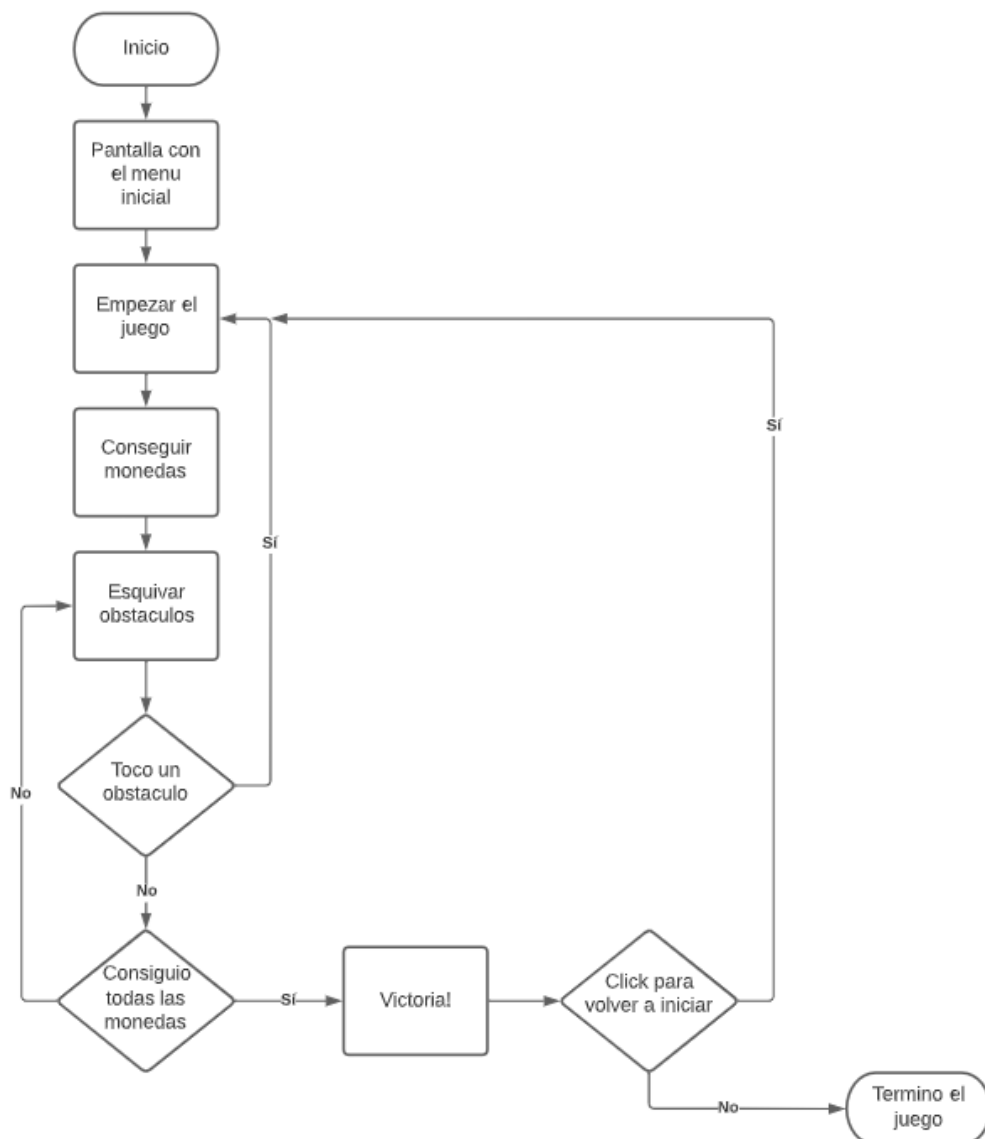
Capítulo II: Diseño e implementación.

2.1 Planificación (Diagrama de Gantt)

Proceso	ACTIVIDAD	INICIO	FINAL	ESTADO
Planificación del video juego	Creación de la plataforma.	28/03/2022	29/03/2022	
	Elección del género.	30/03/2022	30/03/2022	
	Clasificación del videojuego.	31/03/2022	31/03/2022	
	Tipo de animación a escoger.	31/03/2022	31/03/2022	
	Organización y definición de los equipos de trabajo.	01/04/2022	01/04/2022	
	Diseños	02/04/2022	05/04/2022	
Desarrollo de los escenarios y personajes	Creación del menu principal	06/04/2022	07/04/2022	
	Storyboard del contenido	08/04/2022	10/04/2022	
	Creación del personaje del jugador y la IA	11/04/2022	11/04/2022	
	Control del movimiento del personaje principal	11/04/2022	11/04/2022	
	Dirección de la cámara	11/04/2022	11/04/2022	
	Animaciones de la IA	12/04/2022	12/04/2022	
	Música del juego	12/04/2022	12/04/2022	
Documentación de los aspectos descriptivos de juego	Descripción de los requisitos del juego.	13/04/2022	13/04/2022	
	Instrucciones para el jugador.	13/04/2022	13/04/2022	
	Opciones para la gestión del juego.	14/04/2022	14/04/2022	
	Créditos.	14/04/2022	14/04/2022	
	Carteles a mostrar al finalizar el juego.	15/04/2022	15/04/2022	
	Contenido sobre las condiciones de uso del software	15/04/2022	15/04/2022	
	Versión.	16/04/2022	16/04/2022	

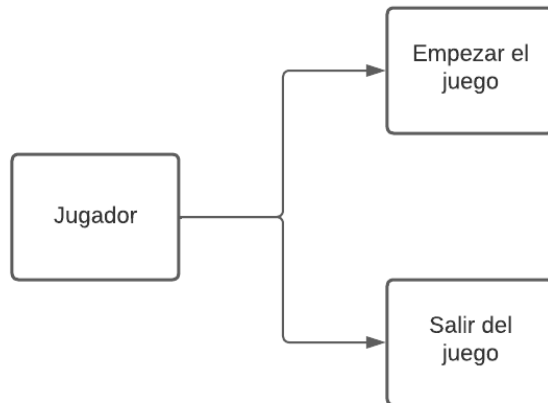
Pruebas y reportes	Pruebas de unidad	17/04/2022	19/04/2022	
	Pruebas de integración.	20/04/2022	23/04/2022	
	Fase ALPHA	24/04/2022	25/04/2022	
	Fase BETA	26/04/2022	26/04/2022	
	Documentación de los reportes con los resultados de las pruebas	26/04/2022	26/04/2022	
Lanzamiento y finalización del video juego	Lanzamientos a las diferentes tiendas de juegos.	27/04/2022	27/04/2022	
	Cierre del proyecto	28/04/2022	28/04/2022	

2.2 Diagramas y Casos de Uso

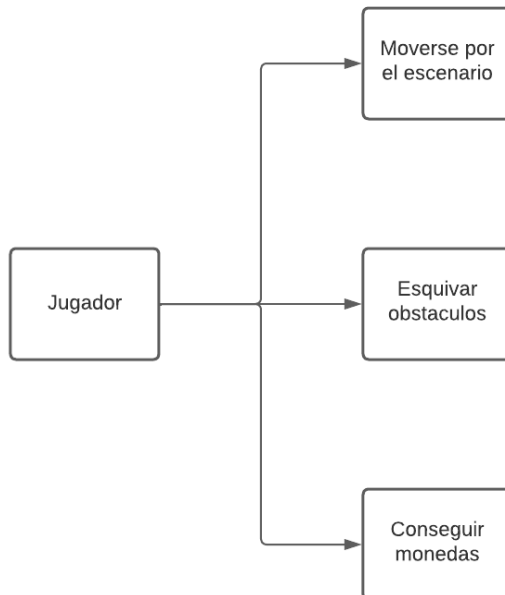


Casos de usos

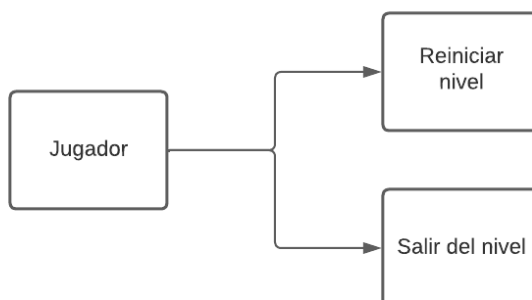
- Menu principal



- Dentro del juego



- Juego finalizado



2.3 Plataforma

El videojuego esta desarrollado para la plataforma de PC.

2.4 Género

El Genero del presente videojuego es Arcade.

2.5 Clasificación

Está clasificado para todo público(T).

2.6 Tipo de Animación

La animación utilizada es 2D.

2.7 Equipo de Trabajo

- Encargado del proyecto: Marco Molina
- Diseñador: Jose Peña.
- Programador: Ana Molina.
- Animador: Jose Eduardo.
- Audio: Yohanna Montan.

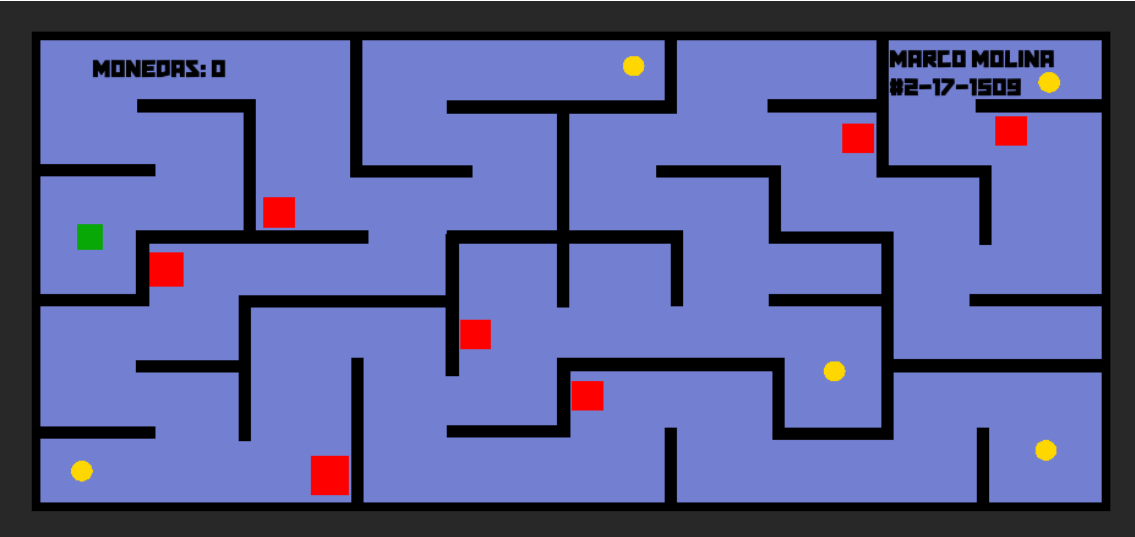
2.8 Historia

Nuestro personaje se encuentra atrapado en un laberinto en el cual debe conseguir todas las monedas dispersas en él, esquivando a todos los enemigos u obstáculos que estarán en su camino para que así al recolectar todas las monedas este logre superar el laberinto y así ganar.

2.9 Guion

A medida que se van recolectando monedas, nuestro personaje se encontrara con enemigos mucho más difíciles de superar, de modo que cuando las consiga todas ganara el nivel.

2.10 Storyboard



2.11 Personajes

Jugador principal



Enemigos



2.12 Niveles

El juego solo contiene un nivel.

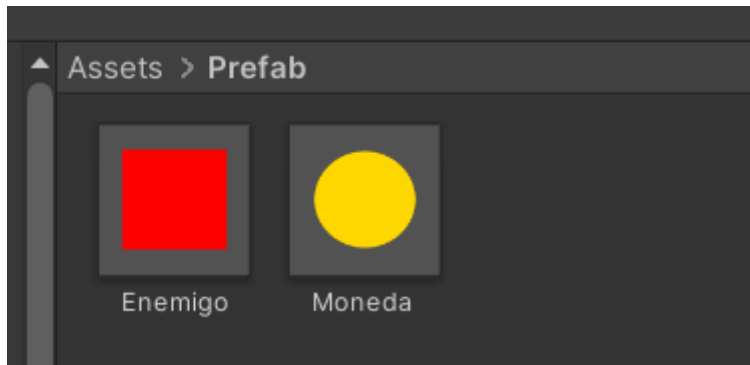
2.13 Mecánica del Juego

Superar a todos los enemigos y obstáculos para conseguir todas las monedas y ganar.

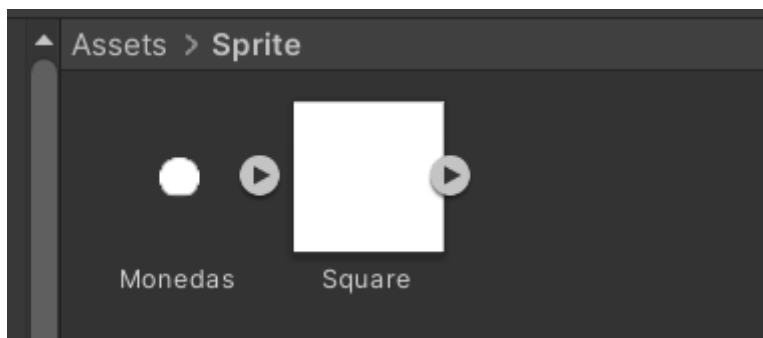
CAPÍTULO III: DESARROLLO

3.1 Capturas de la Aplicación (Documentación completa del desarrollo, Scripts, Sprites, Prefabs e imágenes)

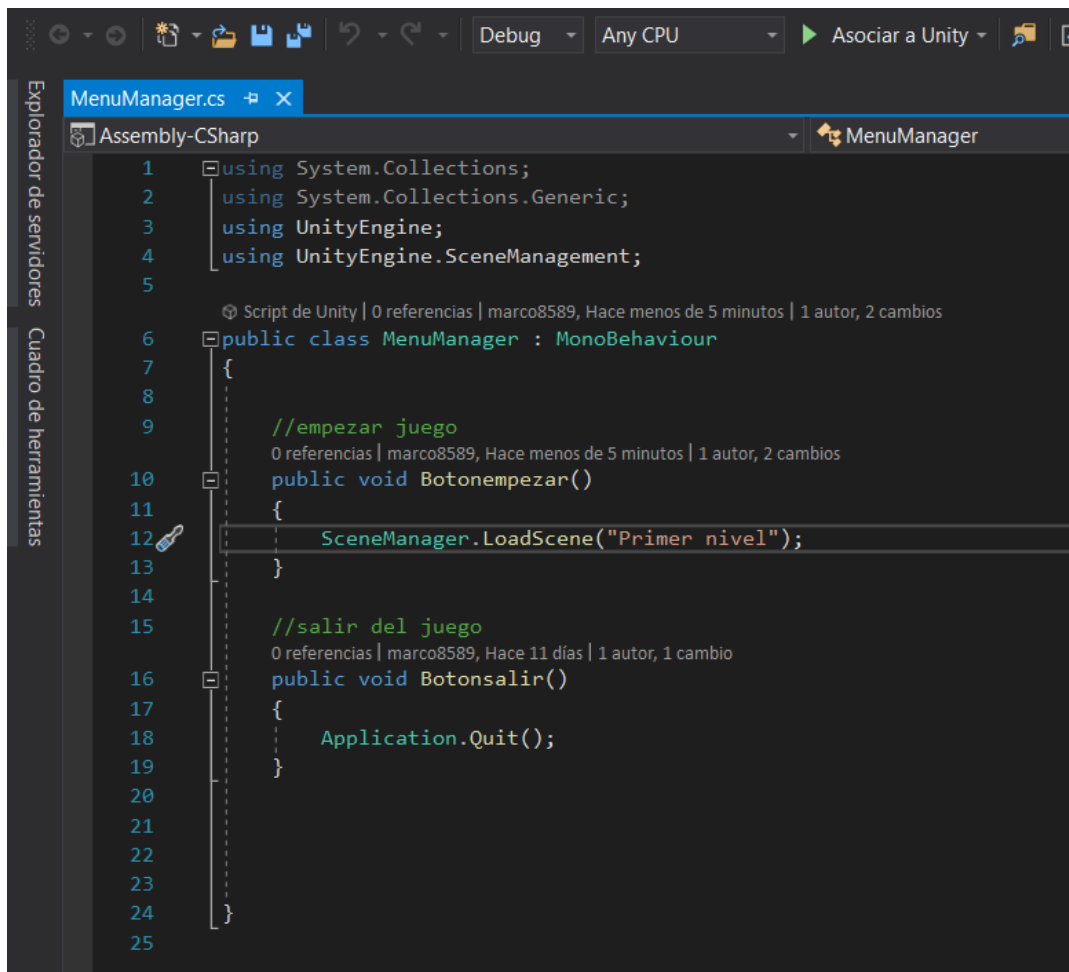
Prefab



Sprites



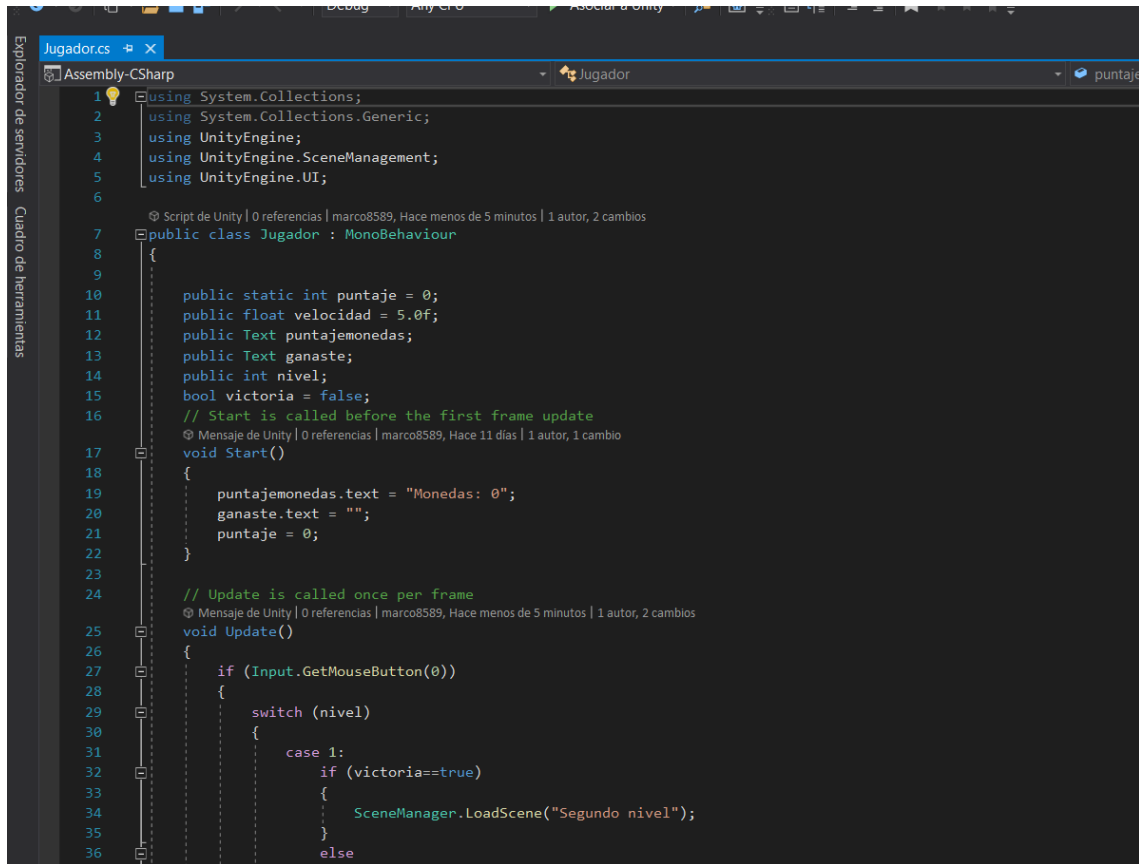
Scripts menumanager



The image shows a screenshot of the Visual Studio IDE with the Unity project 'MenuManager' open. The file 'MenuManager.cs' is selected in the 'Explorador de servidores' (Server Explorer) pane on the left. The code is written in C# and defines a 'MenuManager' class that inherits from 'MonoBehaviour'. The class contains two public methods: 'Botonempezar()' and 'Botonsalir()'. The 'Botonempezar()' method calls 'SceneManager.LoadScene("Primer nivel");' to load the first level. The 'Botonsalir()' method calls 'Application.Quit();' to quit the application. The code is as follows:

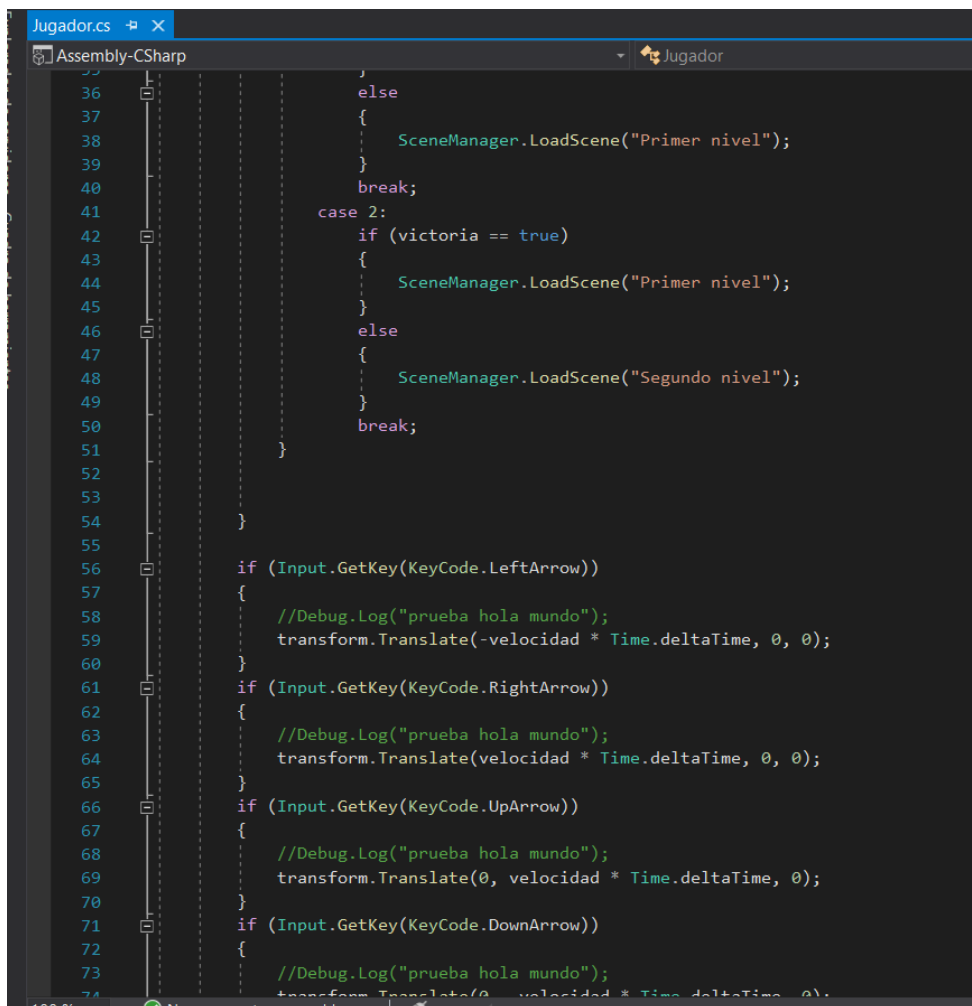
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class MenuManager : MonoBehaviour
7 {
8
9     //empezar juego
10    public void Botonempezar()
11    {
12        SceneManager.LoadScene("Primer nivel");
13    }
14
15    //salir del juego
16    public void Botonsalir()
17    {
18        Application.Quit();
19    }
20
21
22
23
24
25 }
```

Scripts Jugador



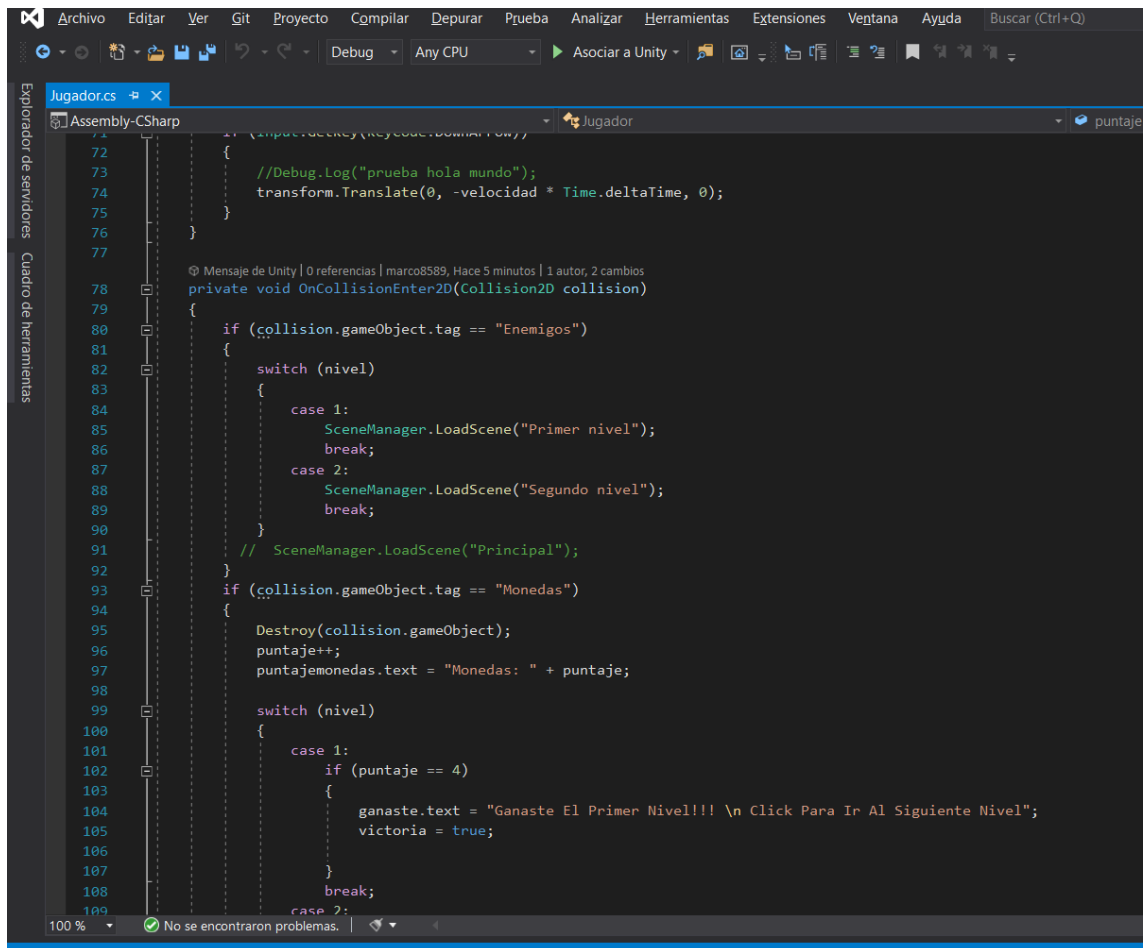
This screenshot shows the first part of the `Jugador.cs` script. It includes the necessary using statements for `System.Collections`, `System.Collections.Generic`, `UnityEngine`, `UnityEngine.SceneManagement`, and `UnityEngine.UI`. The `Jugador` class inherits from `MonoBehaviour`. It defines several public static and instance variables: `puntaje` (int), `velocidad` (float), `puntajemonedas` (Text), `ganaste` (Text), `nivel` (int), and `victoria` (bool). The `Start` method is annotated with a comment indicating it is called before the first frame update. It initializes `puntajemonedas.text` to "Monedas: 0", `ganaste.text` to an empty string, and `puntaje` to 0. The `Update` method is annotated with a comment indicating it is called once per frame. It begins with a check for the left mouse button being pressed.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5 using UnityEngine.UI;
6
7 public class Jugador : MonoBehaviour
8 {
9
10     public static int puntaje = 0;
11     public float velocidad = 5.0f;
12     public Text puntajemonedas;
13     public Text ganaste;
14     public int nivel;
15     bool victoria = false;
16     // Start is called before the first frame update
17     void Start()
18     {
19         puntajemonedas.text = "Monedas: 0";
20         ganaste.text = "";
21         puntaje = 0;
22     }
23
24     // Update is called once per frame
25     void Update()
26     {
27         if (Input.GetMouseButton(0))
28         {
29             switch (nivel)
30             {
31                 case 1:
32                     if (victoria == true)
33                     {
34                         SceneManager.LoadScene("Segundo nivel");
35                     }
36                     else
```

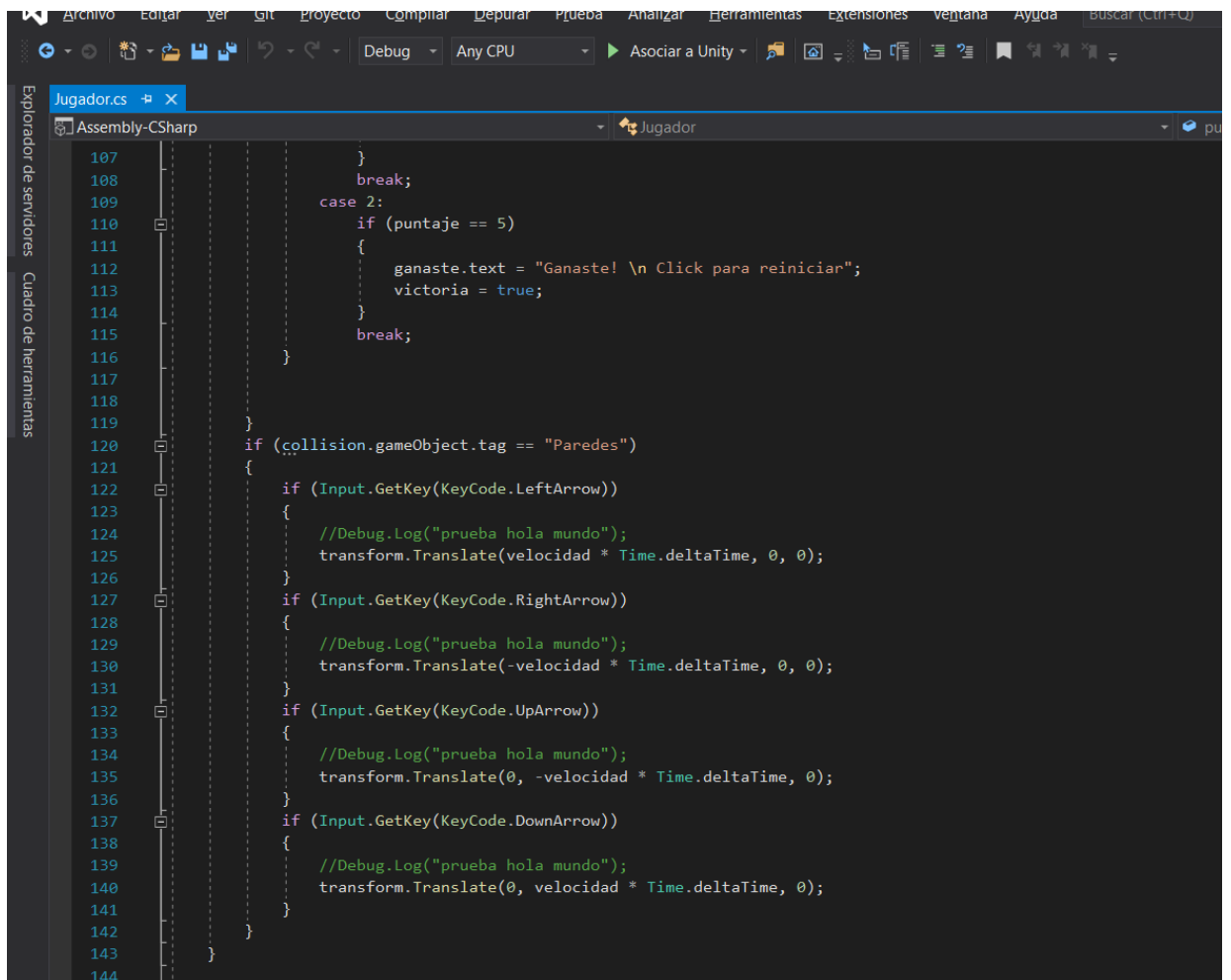


This screenshot shows the continuation of the `Jugador` class. It completes the switch statement for `Update` by adding `case 2`, which also loads "Primer nivel" if `victoria` is true. Following the switch, there are four if-statements that handle keyboard input for left, right, up, and down arrows. Each key press triggers a `Debug.Log` message and a movement of the `transform` along the corresponding axis, calculated as `velocidad * Time.deltaTime`.

```
37         else
38         {
39             SceneManager.LoadScene("Primer nivel");
40         }
41         break;
42     case 2:
43         if (victoria == true)
44         {
45             SceneManager.LoadScene("Primer nivel");
46         }
47         else
48         {
49             SceneManager.LoadScene("Segundo nivel");
50         }
51         break;
52     }
53
54     }
55
56     if (Input.GetKey(KeyCode.LeftArrow))
57     {
58         //Debug.Log("prueba hola mundo");
59         transform.Translate(-velocidad * Time.deltaTime, 0, 0);
60     }
61     if (Input.GetKey(KeyCode.RightArrow))
62     {
63         //Debug.Log("prueba hola mundo");
64         transform.Translate(velocidad * Time.deltaTime, 0, 0);
65     }
66     if (Input.GetKey(KeyCode.UpArrow))
67     {
68         //Debug.Log("prueba hola mundo");
69         transform.Translate(0, velocidad * Time.deltaTime, 0);
70     }
71     if (Input.GetKey(KeyCode.DownArrow))
72     {
73         //Debug.Log("prueba hola mundo");
74         transform.Translate(0, -velocidad * Time.deltaTime, 0);
75     }
76 }
```

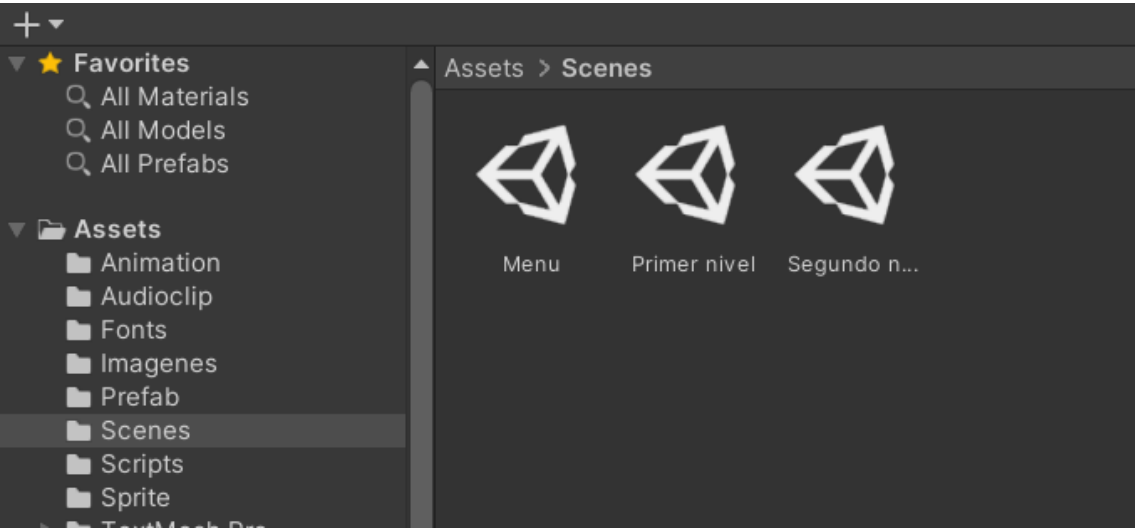


```
72 {
73     //Debug.Log("prueba hola mundo");
74     transform.Translate(0, -velocidad * Time.deltaTime, 0);
75 }
76
77
78 Mensaje de Unity | 0 referencias | marco8589, Hace 5 minutos | 1 autor, 2 cambios
79 private void OnCollisionEnter2D(Collision2D collision)
80 {
81     if (collision.gameObject.tag == "Enemigos")
82     {
83         switch (nivel)
84         {
85             case 1:
86                 SceneManager.LoadScene("Primer nivel");
87                 break;
88             case 2:
89                 SceneManager.LoadScene("Segundo nivel");
90                 break;
91             // SceneManager.LoadScene("Principal");
92         }
93     }
94     if (collision.gameObject.tag == "Monedas")
95     {
96         Destroy(collision.gameObject);
97         puntaje++;
98         puntajemonedas.text = "Monedas: " + puntaje;
99
100         switch (nivel)
101         {
102             case 1:
103                 if (puntaje == 4)
104                 {
105                     ganaste.text = "Ganaste El Primer Nivel!!! \n Click Para Ir Al Siguiente Nivel";
106                     victoria = true;
107                 }
108                 break;
109             case 2:
110                 if (puntaje == 5)
111                 {
112                     ganaste.text = "Ganaste! \n Click para reiniciar";
113                     victoria = true;
114                 }
115                 break;
116             }
117         }
118     }
119     if (collision.gameObject.tag == "Paredes")
120     {
121         if (Input.GetKey(KeyCode.LeftArrow))
122         {
123             //Debug.Log("prueba hola mundo");
124             transform.Translate(velocidad * Time.deltaTime, 0, 0);
125         }
126         if (Input.GetKey(KeyCode.RightArrow))
127         {
128             //Debug.Log("prueba hola mundo");
129             transform.Translate(-velocidad * Time.deltaTime, 0, 0);
130         }
131         if (Input.GetKey(KeyCode.UpArrow))
132         {
133             //Debug.Log("prueba hola mundo");
134             transform.Translate(0, -velocidad * Time.deltaTime, 0);
135         }
136         if (Input.GetKey(KeyCode.DownArrow))
137         {
138             //Debug.Log("prueba hola mundo");
139             transform.Translate(0, velocidad * Time.deltaTime, 0);
140         }
141     }
142 }
143
144
```

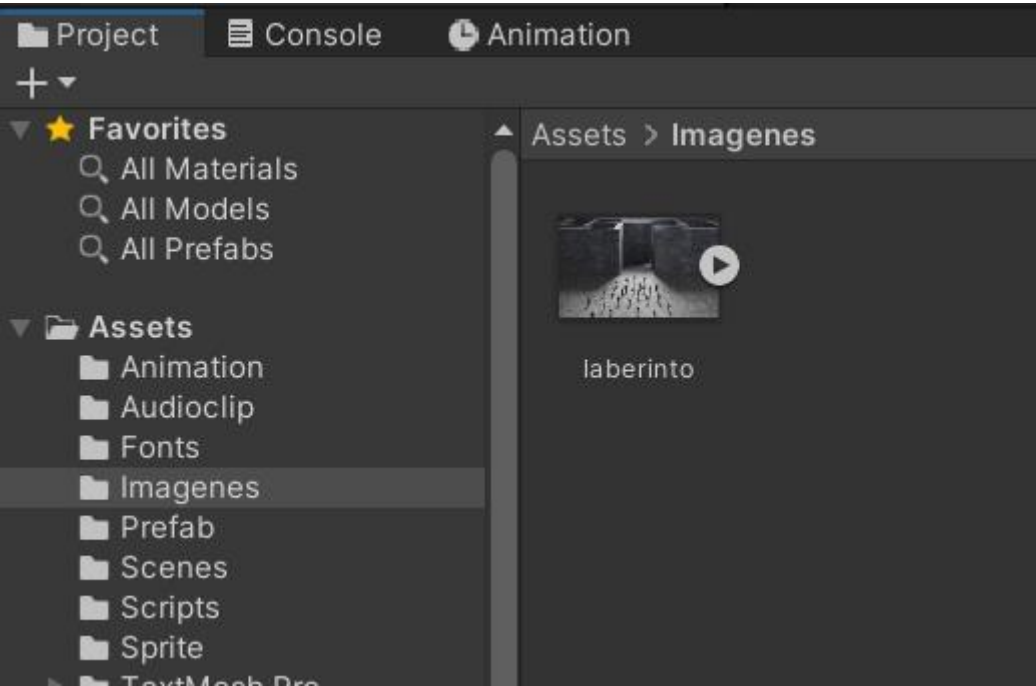


```
107     }
108     break;
109     case 2:
110         if (puntaje == 5)
111         {
112             ganaste.text = "Ganaste! \n Click para reiniciar";
113             victoria = true;
114         }
115         break;
116     }
117 }
118
119 if (collision.gameObject.tag == "Paredes")
120 {
121     if (Input.GetKey(KeyCode.LeftArrow))
122     {
123         //Debug.Log("prueba hola mundo");
124         transform.Translate(velocidad * Time.deltaTime, 0, 0);
125     }
126     if (Input.GetKey(KeyCode.RightArrow))
127     {
128         //Debug.Log("prueba hola mundo");
129         transform.Translate(-velocidad * Time.deltaTime, 0, 0);
130     }
131     if (Input.GetKey(KeyCode.UpArrow))
132     {
133         //Debug.Log("prueba hola mundo");
134         transform.Translate(0, -velocidad * Time.deltaTime, 0);
135     }
136     if (Input.GetKey(KeyCode.DownArrow))
137     {
138         //Debug.Log("prueba hola mundo");
139         transform.Translate(0, velocidad * Time.deltaTime, 0);
140     }
141 }
142 }
143
144
```

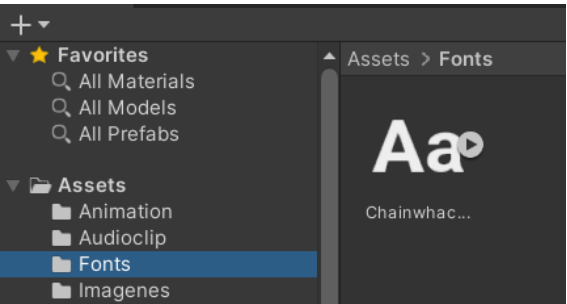
Scenes



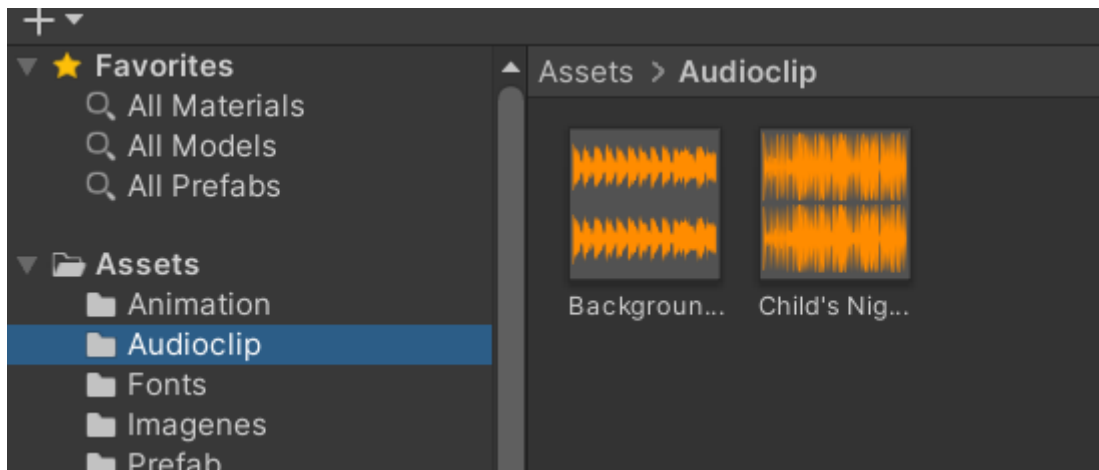
Imagenes



Fonts



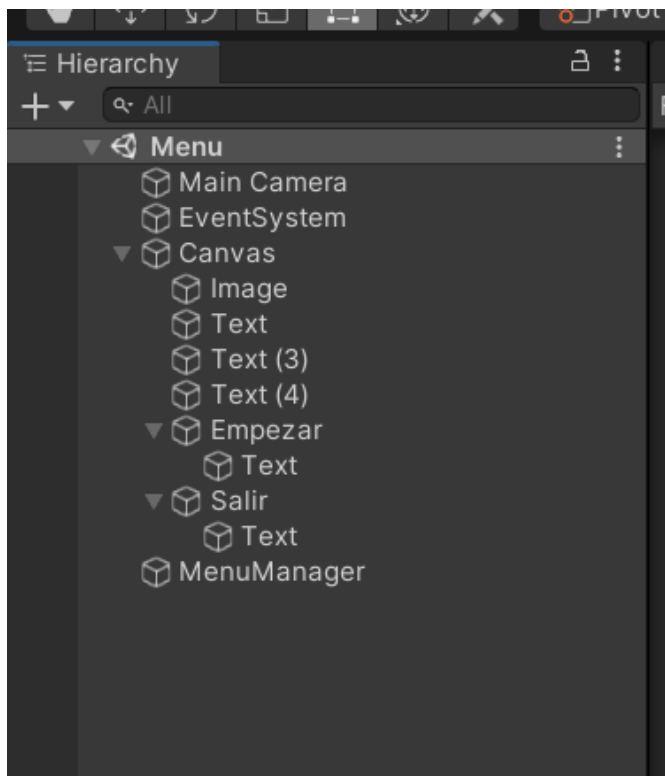
Audioclip



Animation

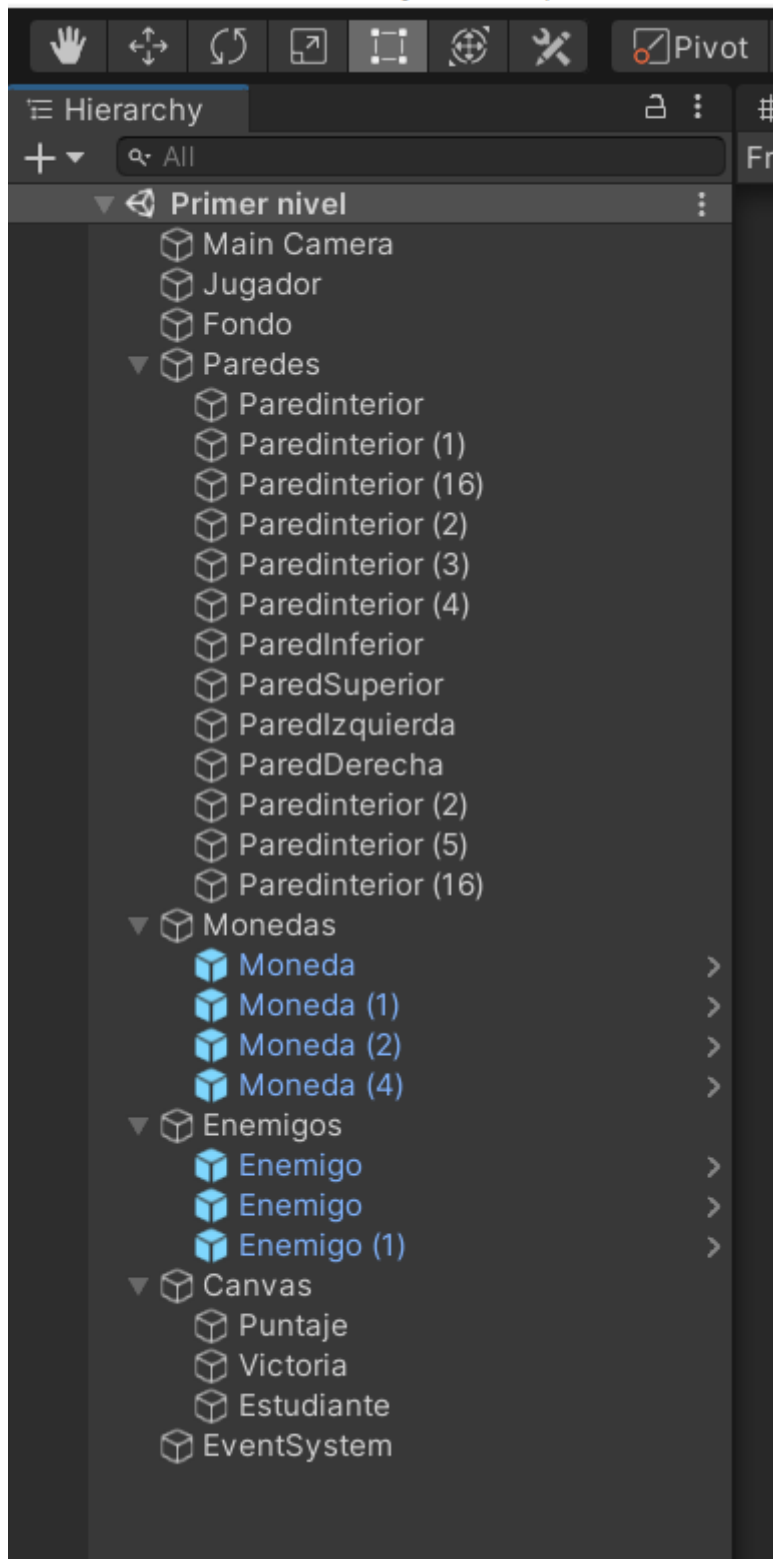


GameObjects de la escena Menu

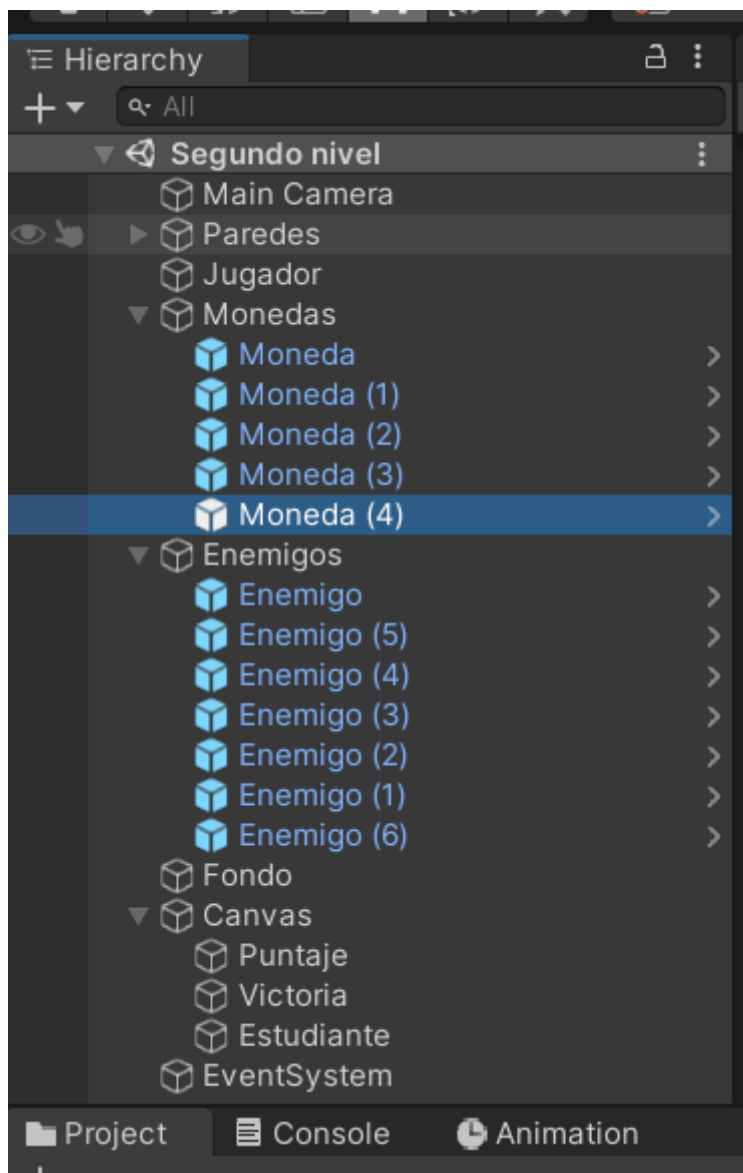


GameObjects de la escena Prime Nivel

File Edit Assets GameObject Component Window

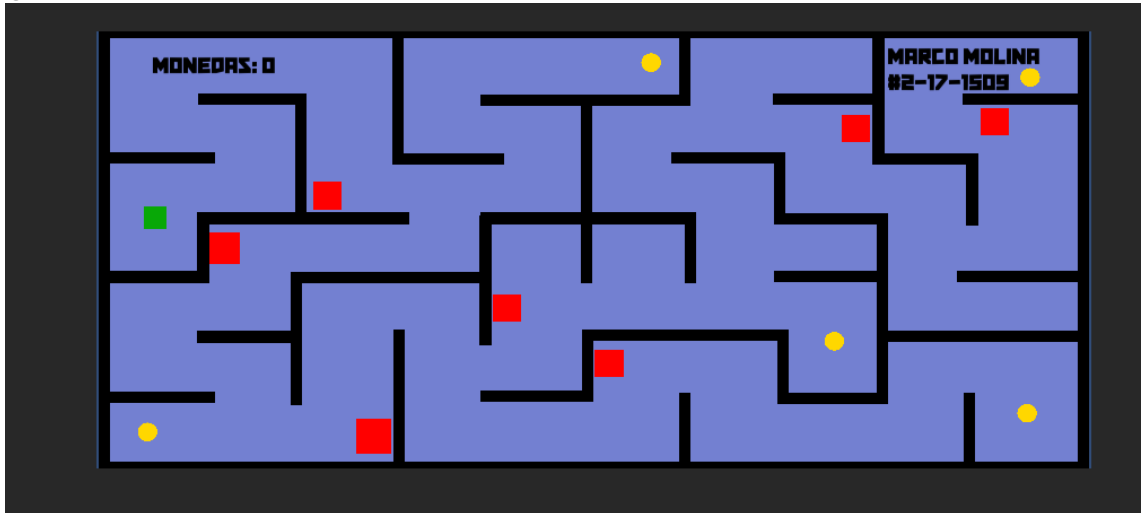


GameObjects de la escena Segundo Nivel



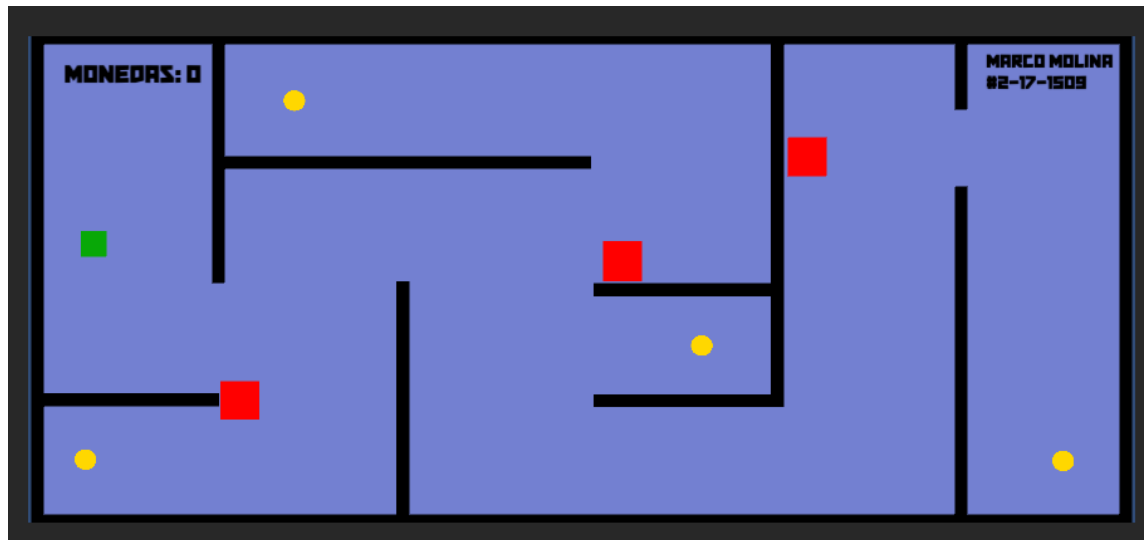
3.2 Prototipos

- El Primer prototipo creado hace un mes, en el cual se tenía un solo nivel sin un menú de selección, donde este contenía algunos errores de diseño y algunos bugs que se han arreglado con el transcurso de las pruebas realizadas. El único nivel que se tenía contaba con los obstáculos correspondiente y las monedas que se tienen que tomar para ganar con su música de fondo.



- El segundo prototipo y final, para este se completó el menú con algunas descripciones para los usuarios y realizo un nivel un poco más fácil para iniciar cómodamente esta experiencia.





3.3 Perfiles de usuarios

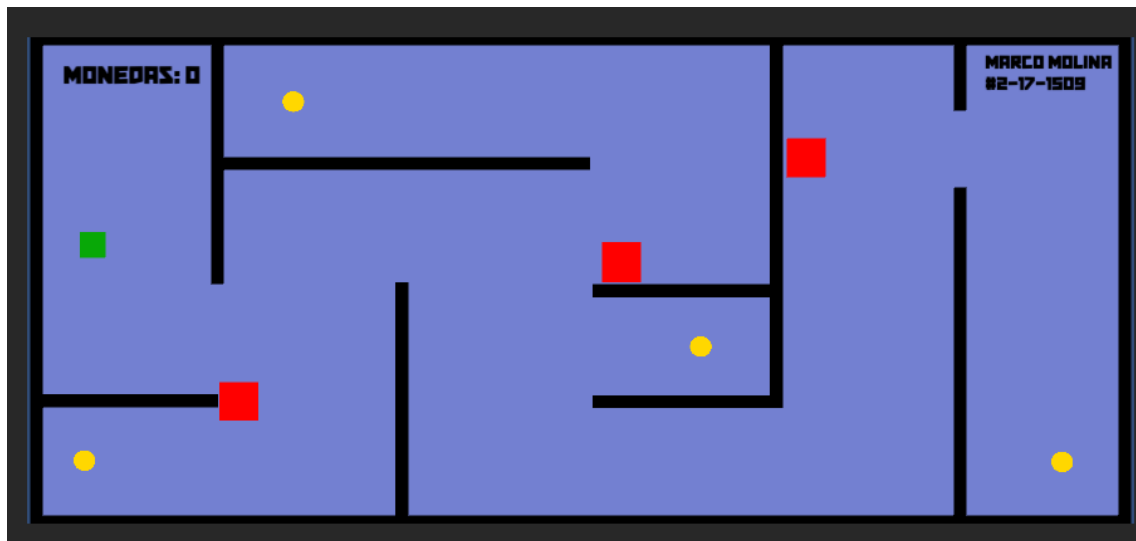
El juego va dirigido a todo tipo de personas, edades, con gustos particulares. Solo se tiene como requisito saber leer y una computadora en la cual instalar el juego.

3.4 Usabilidad

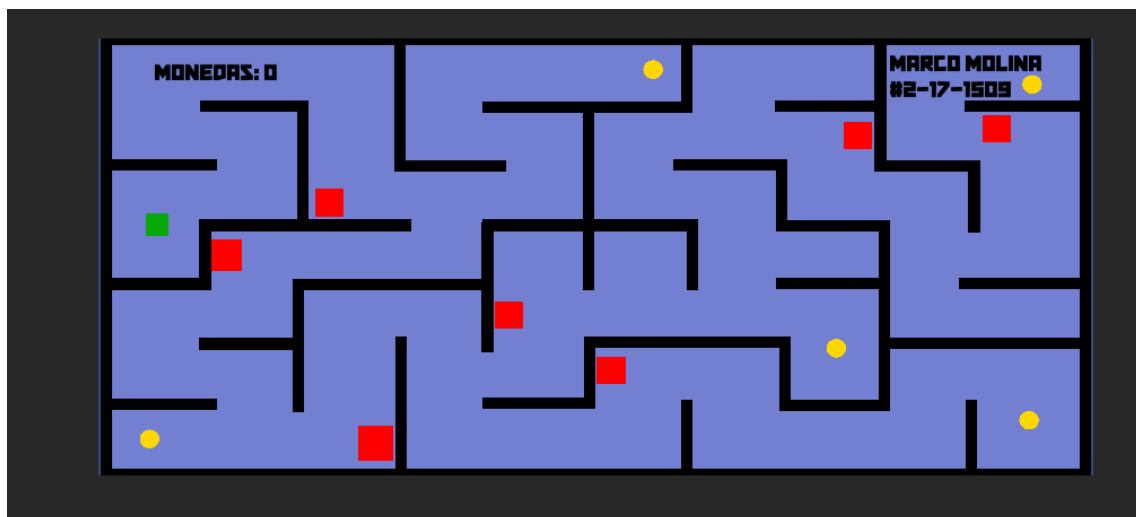
Al iniciar el juego tenemos un con las descripciones que el jugador necesita para tener una buena experiencia de juego, cabe destacar que los niveles se podrán avanzar conforme a se superan los niveles.



Al oprimir el botón empezar nos dirigiremos al primer nivel en el cual el jugador debe superar cada uno de los obstáculos para obtener la victoria y avanzar hacia los siguientes niveles.



Segundo nivel y ultimo.



3.5 Test

Se realizaron una serie de preguntas al publico en generar acerca del juego para medir la experiencia y otros aspectos importantes.

Entre estas tenemos:

- ¿Este videojuego es bueno para sentirse relajado?
- ¿Se comprenden las instrucciones de juego?
- ¿Es muy intuitivo?
- ¿Es divertido?
- ¿La velocidad de movimientos de todos los personajes es buena?
- ¿Es difícil?
- ¿Es buena la cantidad de niveles?

3.6 Versiones de la aplicación

La versión Alpha del videojuego era sin moneda solo superando un laberinto hasta salir.

La versión Beta del juego ya se le agregaron monedas y obstáculos donde ya no es superar un laberinto, es mas bien conseguir todas las monedas esquivando obstáculos a lo largo del nivel que le toque al jugador.

CAPÍTULO IV: PUBLICACIÓN

4.1 Requisitos de instalación

Sistema operativo: Windows XP / Vista / 7 /8.

Procesador: 1.0 Ghz o superior.

Memoria: 512 MB RAM.

DirectX: Version 9.0.

Disco duro: 67.5 MB de espacio libre.

Tarjeta de sonido: compatible con DirectX.

4.2 Instrucciones de uso

Para jugar se utilizan las flechas del teclado y para reiniciar el nivel se debe realizar un click en cualquier parte del juego.

4.3 BUGS

- Al momento de pasar por alguna pared demasiado cerca esta puede bloquear el camino del jugador como si fuera una pared invisible.
- Se tenia que en medio del mapa hubieses una pared invisible.
- A veces se traspasan paredes.
- Al cambiar la resolución de la computadora el juego puede perder visión de algunas partes.

4.4 Proyección a futuro

En el futuro se pretende agregar muchos más niveles y más intuitivos, con más obstáculos y situaciones que pongan a prueba al jugador. También se tiene contemplado agregar nuevos personajes para el jugador, en la tienda agregar la parte de donaciones para el apoyo del videojuego y poder ejecutarse en los diferentes dispositivos y computadoras del mercado.

4.5 Presupuesto

ROL	TIEMPO	COSTO
Diseñador Creativo	1 mes	RD\$20,000
Dibujador	2 meses	RD\$25,000
Animador	2 meses	RD\$25,000
Técnico de sonido	1 mes	RD\$20,000
Programador	3 meses	RD\$30,000
Tester	1 mes	RD\$15,000

El proyecto tendrá un costo total de RD\$135,000, con las cuales se esperan grandes ganancias al implementarlo.

4.6 Análisis del mercado

Con el pasar de los años los videojuegos van tomando mucha mas importancia en los mercados actuales, de modo que estos no son simplemente diversión sino una gran fuente de ingreso que a medida que se innovan en las diferentes tecnologías para desarrollar estos van incrementando su valor y generando una gran cantidad de ingresos.

En la actualidad este mercado a generado una gran cantidad de dinero y con grandes ideas que llegan cada año para la creación de otros más, de modo que es un negocio que se puede explotar a gran escala y crecer con el tiempo.

4.7 Viabilidad

El juego se podrá ejecutar en computadoras y sitios web, por la facilidad del uso del teclado. En la versión final de este se pondrá la opción de donación y luego de que el juego es te más completo se pondrán tiendas con micro pagos, ya sea para la compra de personajes o niveles.

Conclusiones

Con el desarrollo de este videojuego pude comprender lo que es la creación de un videojuego donde debemos aplicar una variedad de pasos desde el diseño conceptual hasta la realización de pruebas para la posterior subida a las diferentes tiendas del mercado, con lo cual también se debe realizar un análisis del mismo para verificar las competencias actuales que podrían no beneficiar al proyecto, es decir, es parecido a un proyecto de software. También pude aprender lo que es el Unity con todos sus aspectos para la creación de videojuegos como son los Sprite, prefabs y los diferentes componentes que se le pueden aplicar los Gameobjects que se tengan en el escenario a trabajar, las clasificaciones de los juegos, a quienes va dirigido, las pruebas que se realizan entre otros aspectos del desarrollo.

REFERENCIAS BIBLIOGRÁFICAS

- <https://www.muycomputer.com/2020/08/04/requisitos-de-juegos-en-pc/>
- <https://aleph.org.mx/que-es-un-bug-en-un-videojuego#:~:text=%C2%BFQu%C3%A9%20es%20un%20bug%20en%20los%20juegos%3F,bien%20una%20caracter%C3%ADstica%20no%20prevista.>
- <https://www.uco.es/ucopress/ojs/index.php/edmetic/article/view/11101>

Proyecto Final Documentación:

https://github.com/marco8589/Proyecto_final.git

Repositorio del juego:

<https://github.com/marco8589/juegolaberinto.git>

Juego publicado:

<https://marco8589.github.io/juegolaberinto/>

Códigos fuentes del juego:

<https://github.com/marco8589/codigolaberinto.git>