

UNIVERSIDAD TECNOLÓGICA DE SANTIAGO UTESA

Facultad de Ingeniería y
Arquitectura Carrera de
Informática



Asignatura:

Videojuegos

Tema:

Capítulo 3

Presentado a:

Iván Mendoza

Presentado por:

Marco Molina Disla 2-17-1509

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/>

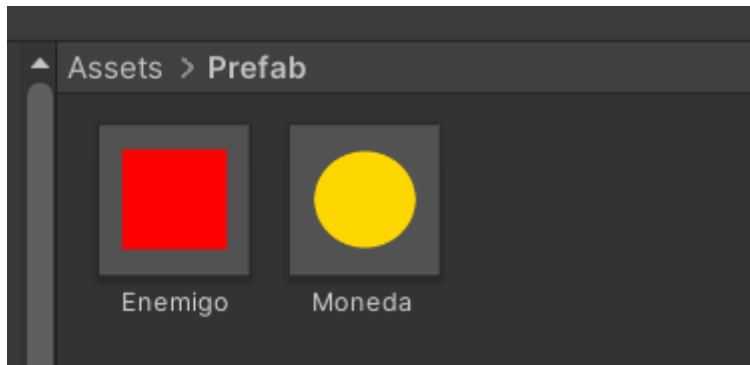
Códigos fuentes del juego:

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

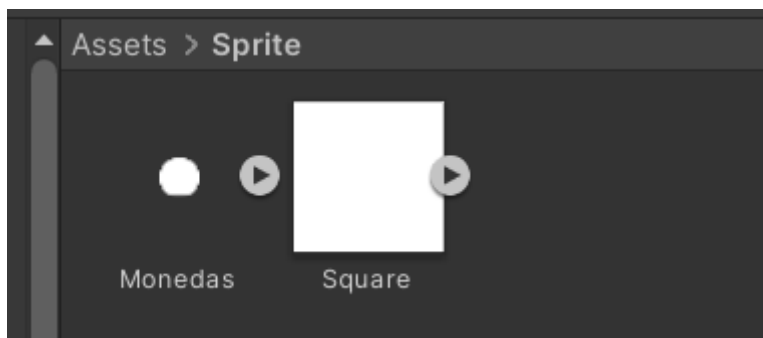
CAPÍTULO III: DESARROLLO

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

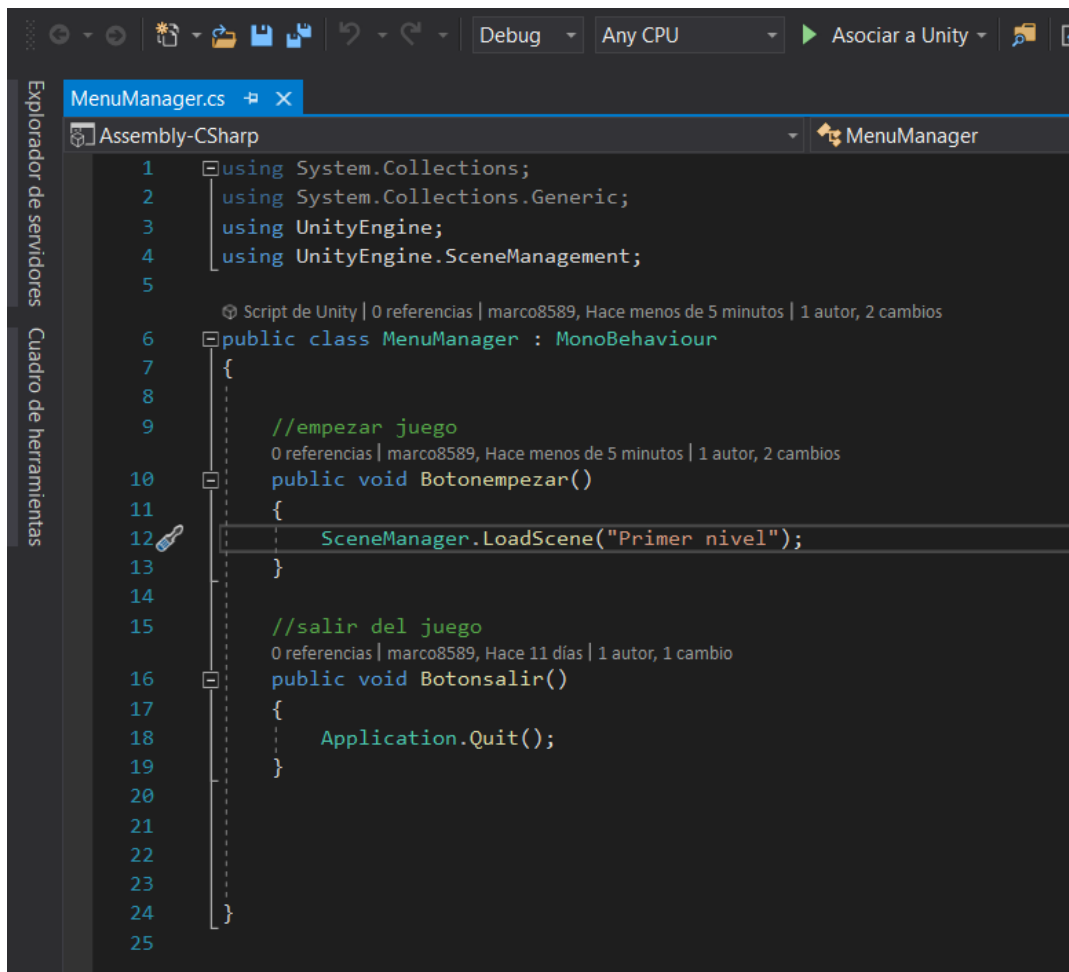
Prefab



Sprites



Scripts menumanager

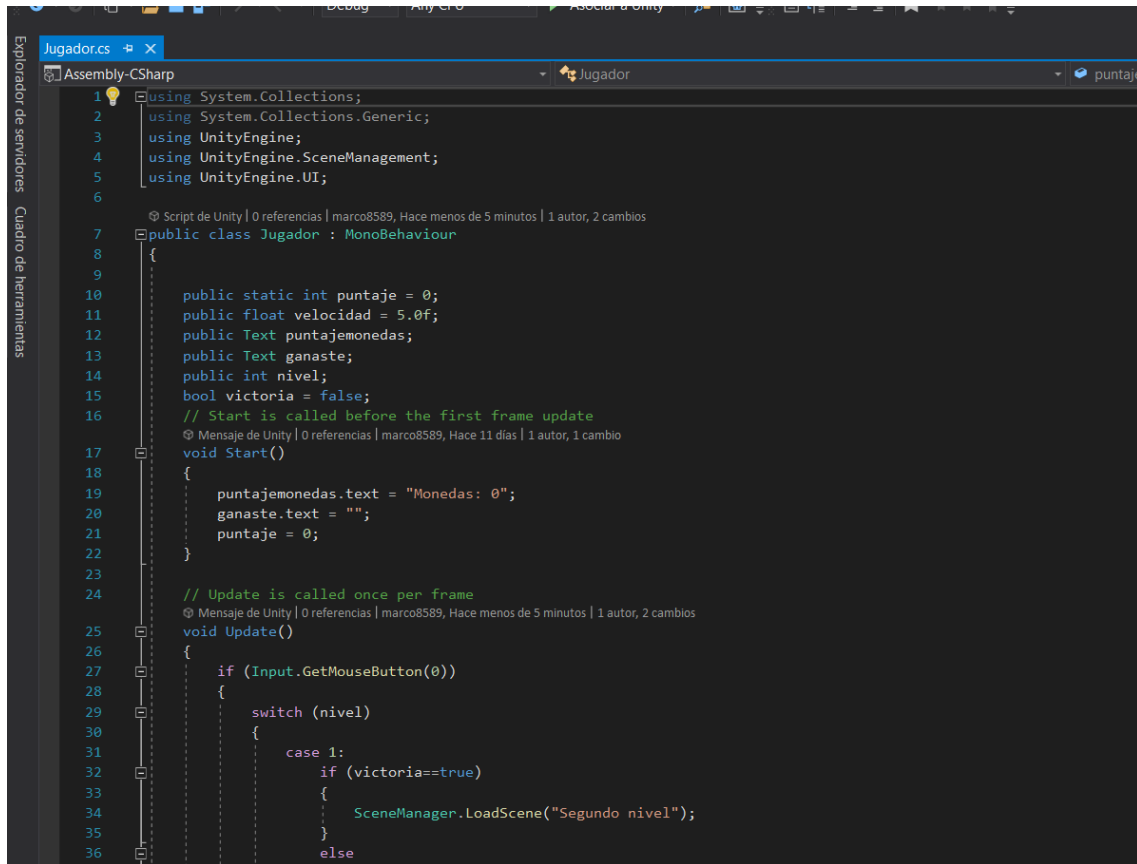


The image shows a screenshot of the Visual Studio IDE with the Unity project open. The file explorer on the left shows the 'MenuManager.cs' file. The code editor displays the following C# code:

```
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 }
```

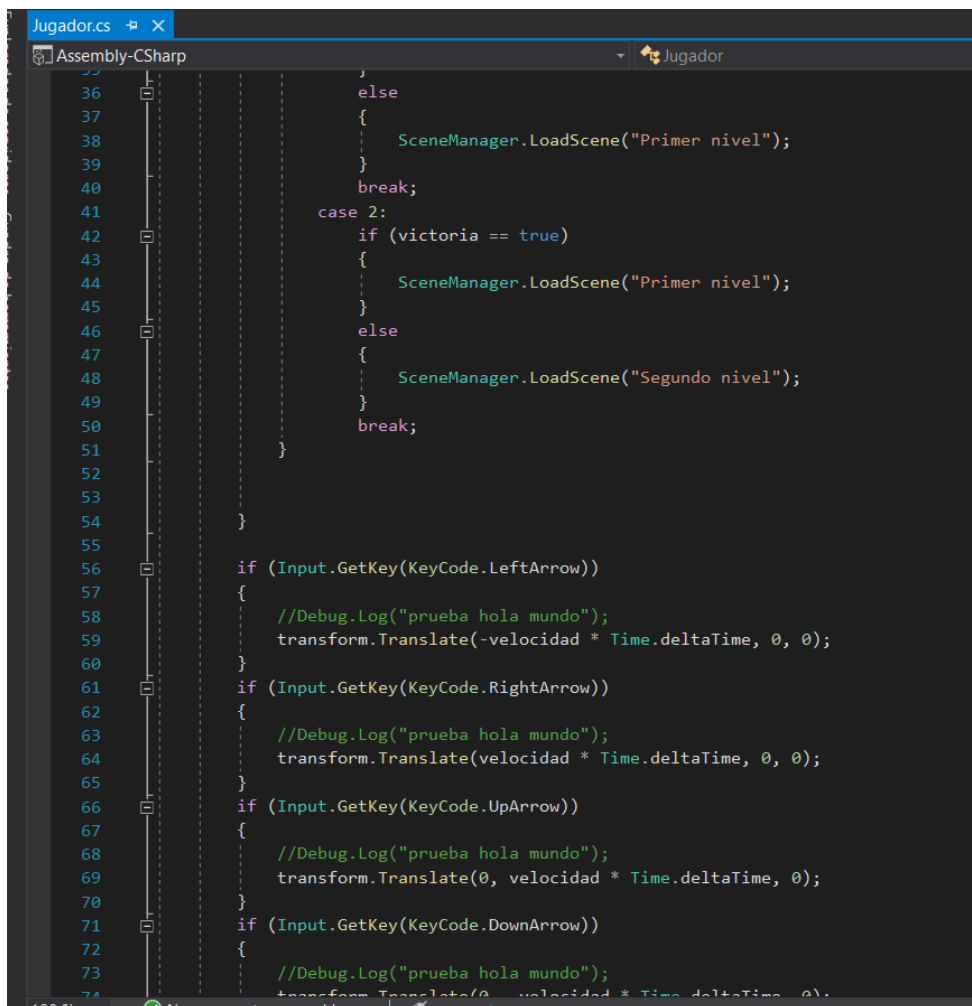
The code defines a `MenuManager` class that inherits from `MonoBehaviour`. It contains two public methods: `Botonempezar()` which loads the scene "Primer nivel", and `Botonsalir()` which quits the application. The code is written in C# and uses Unity's `SceneManager` and `Application` classes.

Scripts Jugador



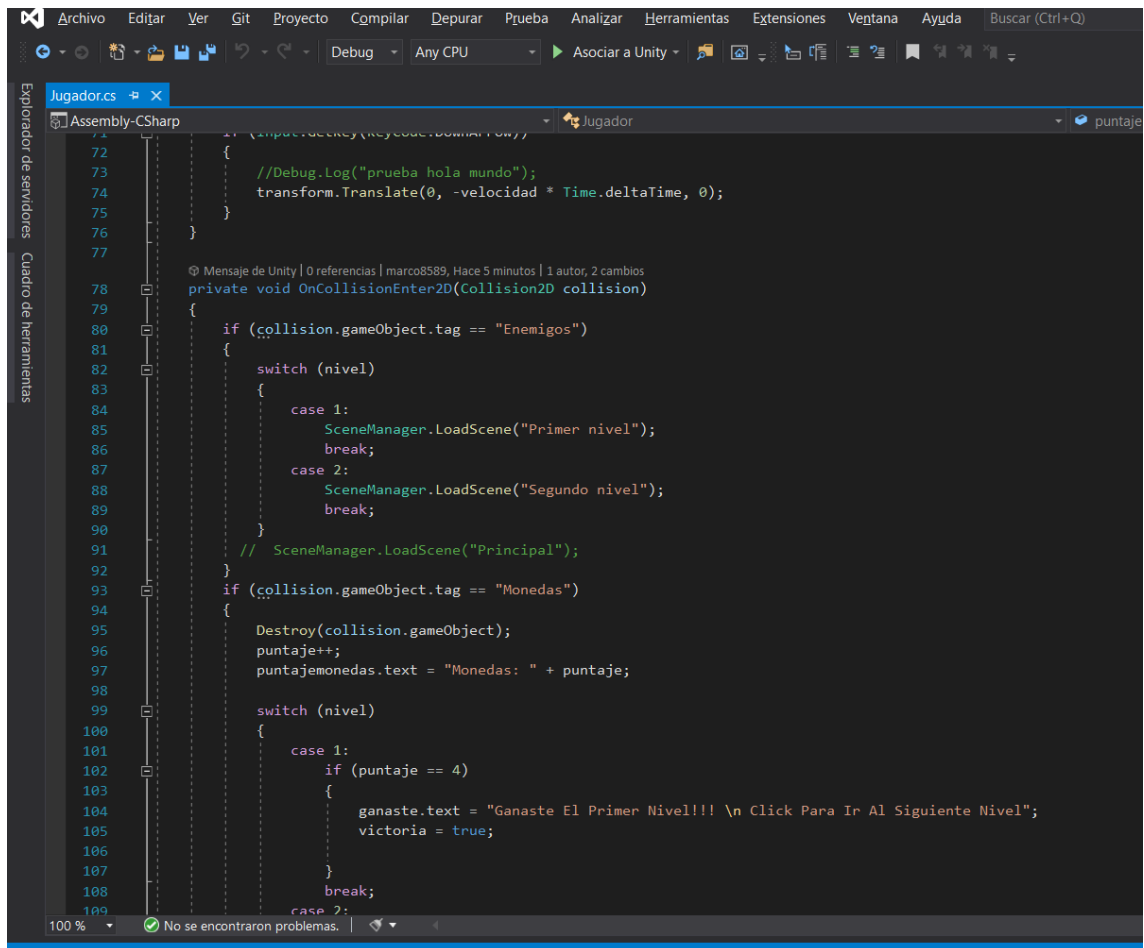
This screenshot shows the first part of the `Jugador.cs` script. It includes the necessary using statements for `System.Collections`, `System.Collections.Generic`, `UnityEngine`, `UnityEngine.SceneManagement`, and `UnityEngine.UI`. The `Jugador` class inherits from `MonoBehaviour`. It defines several 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
```

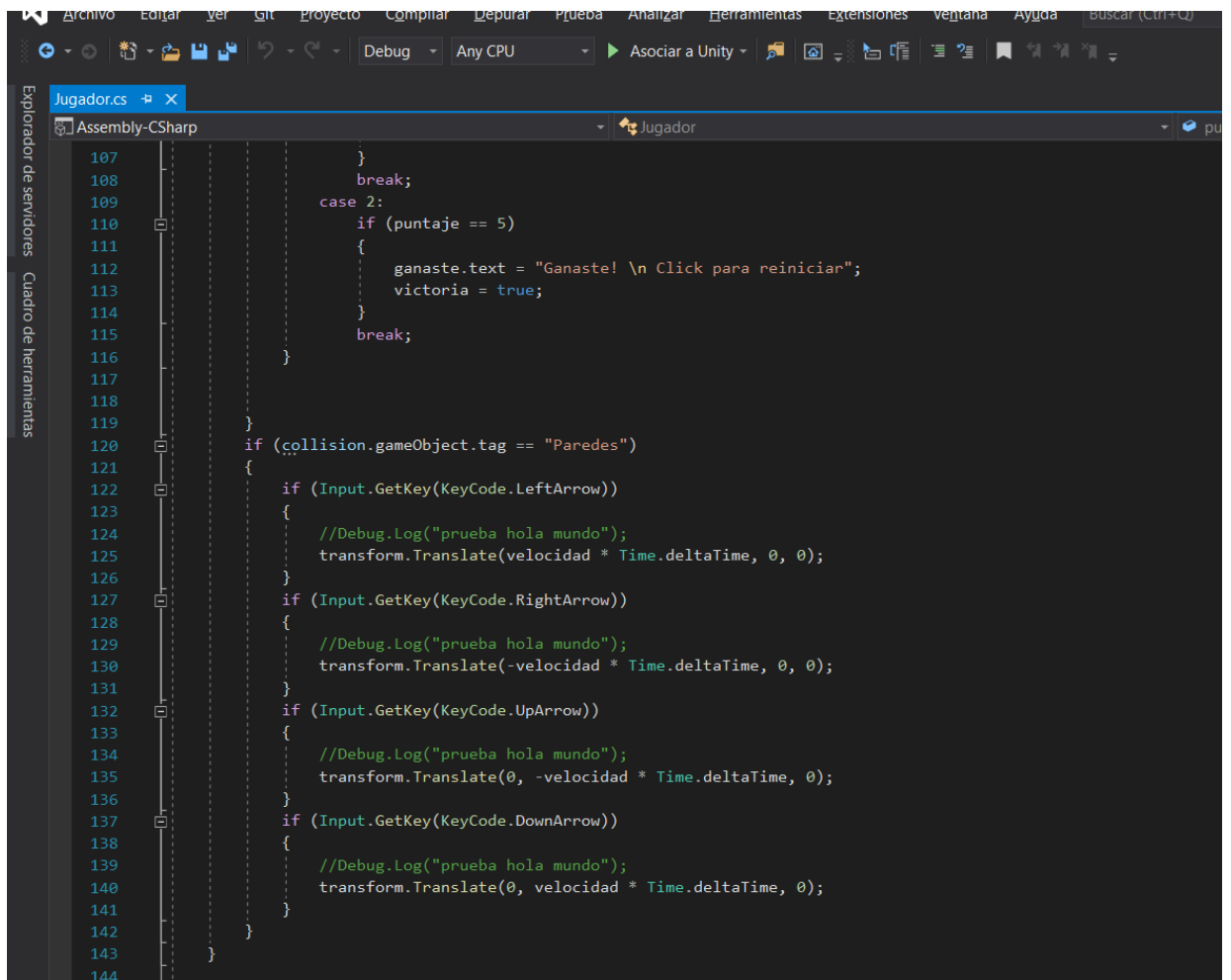


This screenshot shows the continuation of the `Jugador` class. It completes the switch statement for `case 1` and adds `case 2`. Both cases check for `victoria == true` and call `SceneManager.LoadScene("Primer nivel")` if true, or `SceneManager.LoadScene("Segundo nivel")` if false, before breaking. Following the switch statement, there are four if-statements for keyboard input: Left Arrow, Right Arrow, Up Arrow, and Down Arrow. Each key press triggers a `Debug.Log` statement (commented out) and a `transform.Translate` call to move the player. The movement is calculated as `velocidad * Time.deltaTime` in the appropriate direction.

```
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     if (Input.GetKey(KeyCode.LeftArrow))
55     {
56         //Debug.Log("prueba hola mundo");
57         transform.Translate(-velocidad * Time.deltaTime, 0, 0);
58     }
59     if (Input.GetKey(KeyCode.RightArrow))
60     {
61         //Debug.Log("prueba hola mundo");
62         transform.Translate(velocidad * Time.deltaTime, 0, 0);
63     }
64     if (Input.GetKey(KeyCode.UpArrow))
65     {
66         //Debug.Log("prueba hola mundo");
67         transform.Translate(0, velocidad * Time.deltaTime, 0);
68     }
69     if (Input.GetKey(KeyCode.DownArrow))
70     {
71         //Debug.Log("prueba hola mundo");
72         transform.Translate(0, -velocidad * Time.deltaTime, 0);
73     }
74 }
```

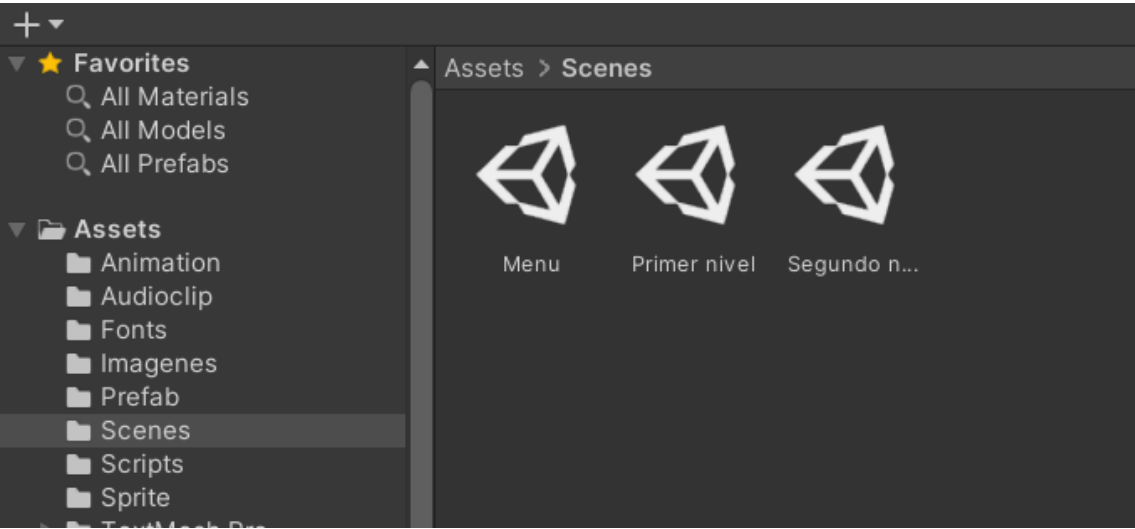


```
72 {
73     //Debug.Log("prueba hola mundo");
74     transform.Translate(0, -velocidad * Time.deltaTime, 0);
75 }
76
77
78 Mensaje de Unity | 0 referencias | marco8589, Hace 5 minutos | 1 autor, 2 cambios
79 private void OnCollisionEnter2D(Collision2D collision)
80 {
81     if (collision.gameObject.tag == "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:
```

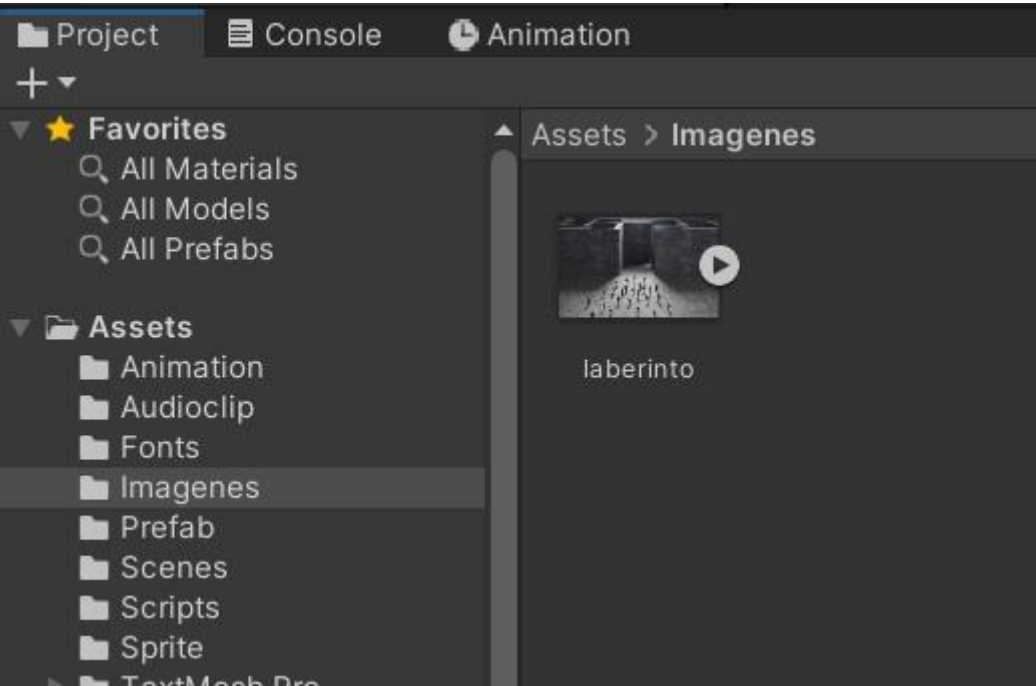


```
107     }
108     break;
109     case 2:
110         if (puntaje == 5)
111         {
112             ganaste.text = "Ganaste! \n Click para reiniciar";
113             victoria = true;
114         }
115         break;
116     }
117
118
119 }
120 if (collision.gameObject.tag == "Paredes")
121 {
122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

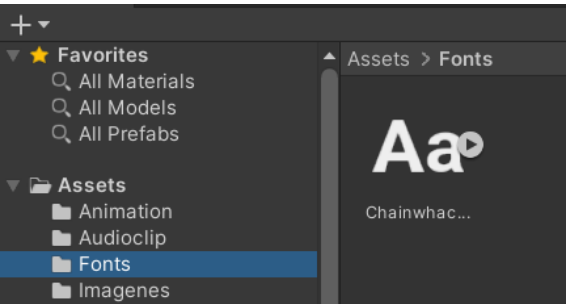
Scenes



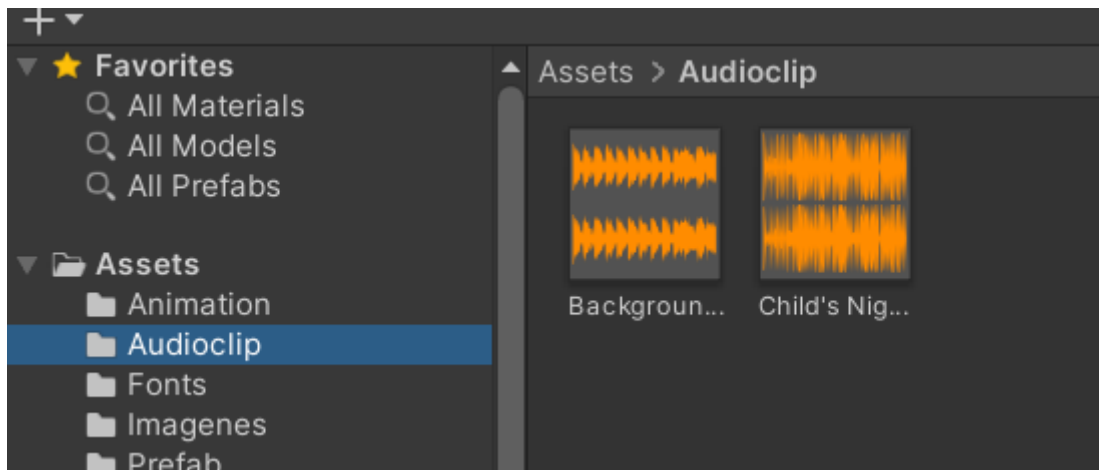
Imagenes



Fonts



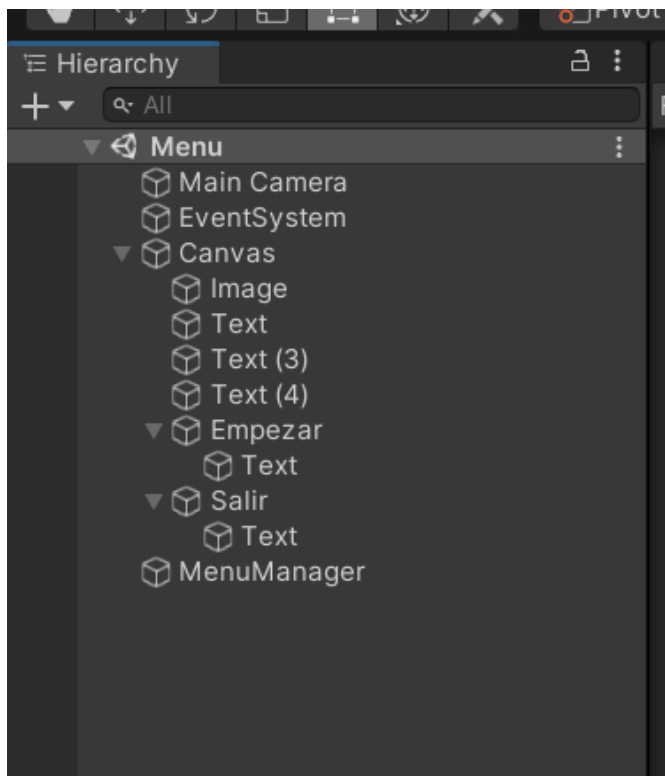
Audioclip



Animation



GameObjects de la escena Menu

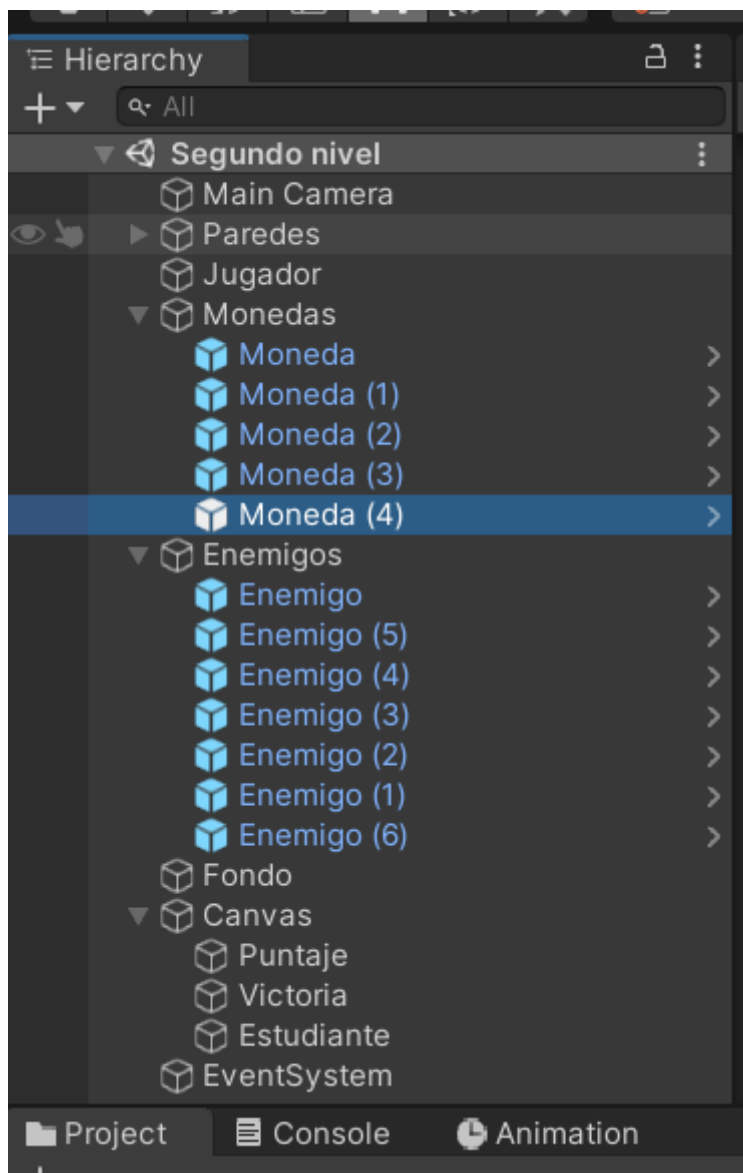


GameObjects de la escena Prime Nivel

File Edit Assets GameObject Component Window

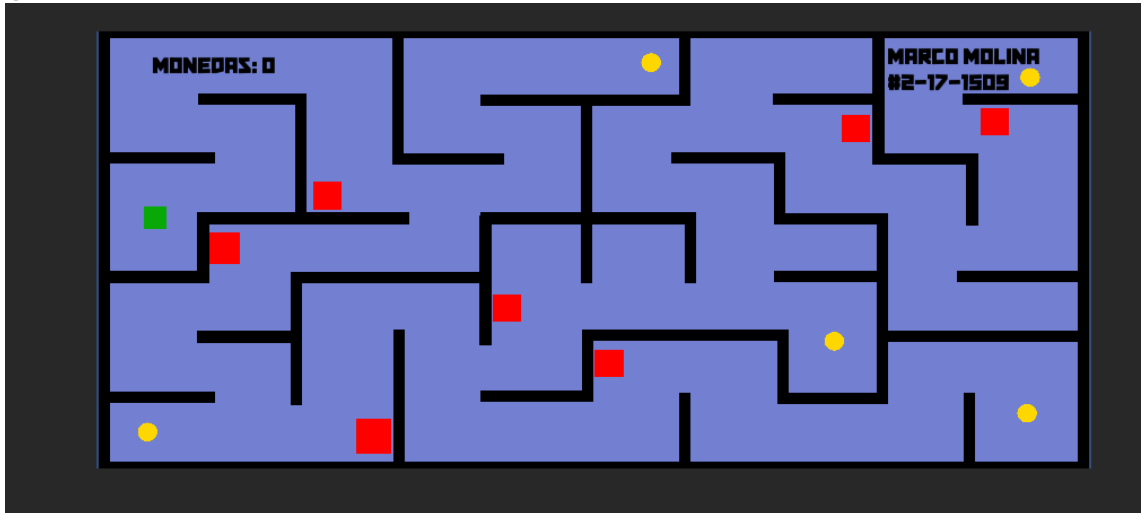


GameObjects de la escena Segundo Nivel



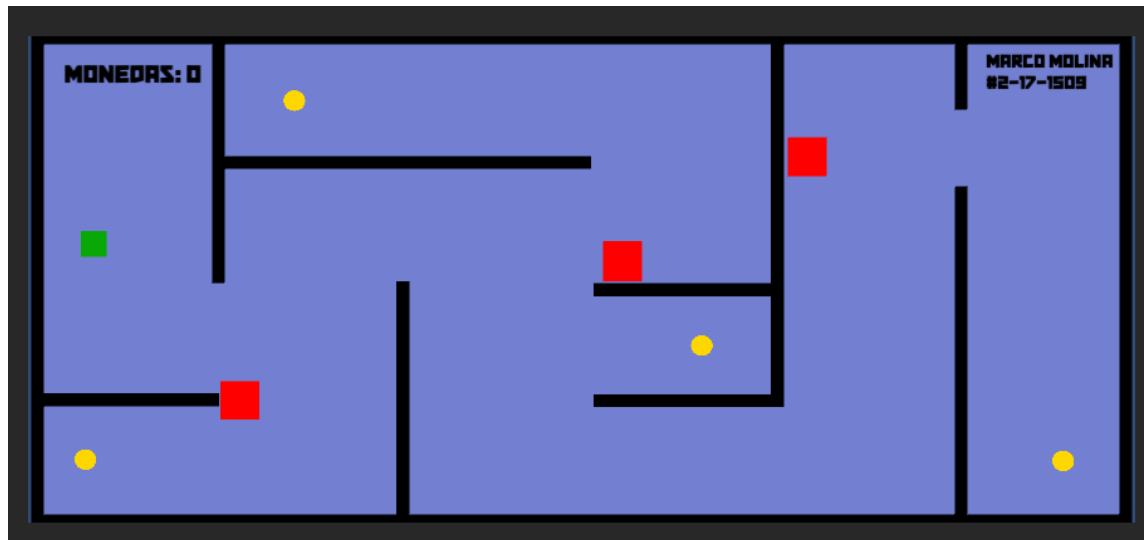
3.2 Prototipos

- 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

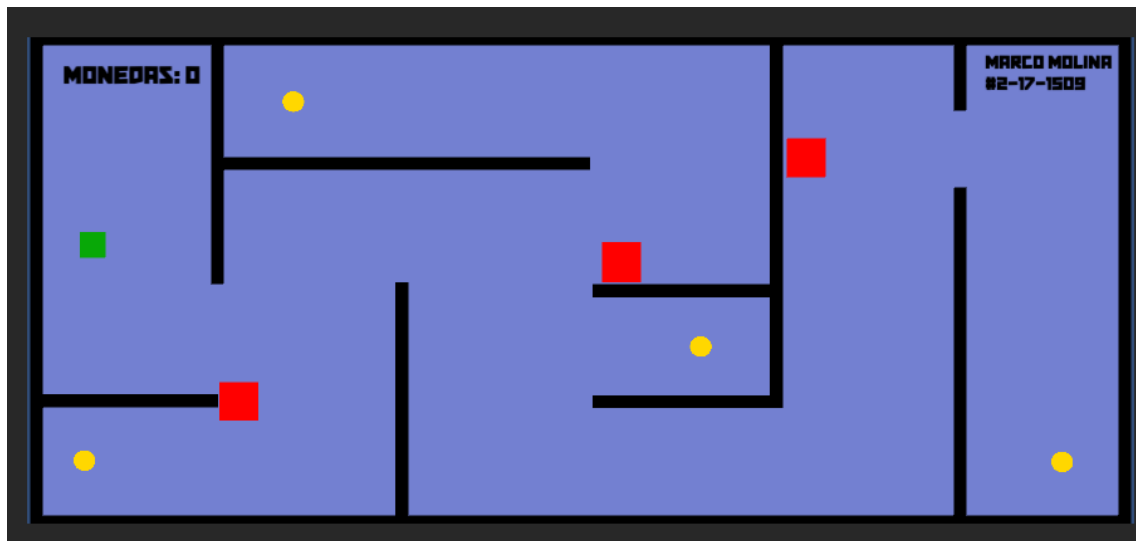
El juego va dirigido a todo tipo de personas, edades, con gustos particulares. Solo se tiene como requisito saber leer y una computadora en la cual instalar el juego.

3.4 Usabilidad

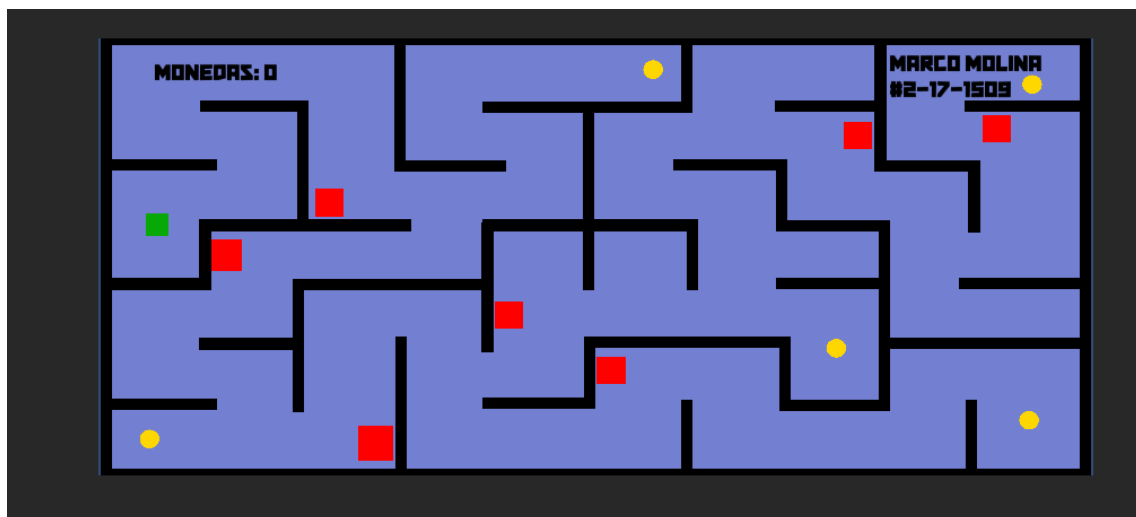
Al iniciar el juego tenemos un con las descripciones que el jugador necesita para tener una buena experiencia de juego, cabe destacar que los niveles se podrán avanzar conforme a se superan los niveles.



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.