**UNIVERSIDAD TECNOLOGICA DE SANTIAGO**

**(UTESA)**

**SISTEMA CORPORATIVO**

**FACULTAD DE ARQUITECTURA E INGENIERÍA**



**Asignatura:**

Programación de Video Juegos

**Tarea Capítulo III**

**Facilitador:**

Iván Mendoza

**Realizado por:**

Yamilka Gómez Morán 2-18-0529

Wayddy Ashley Grullón 2-18-0269

Jean Ureña 2-17-1955

17 de agosto, 2022

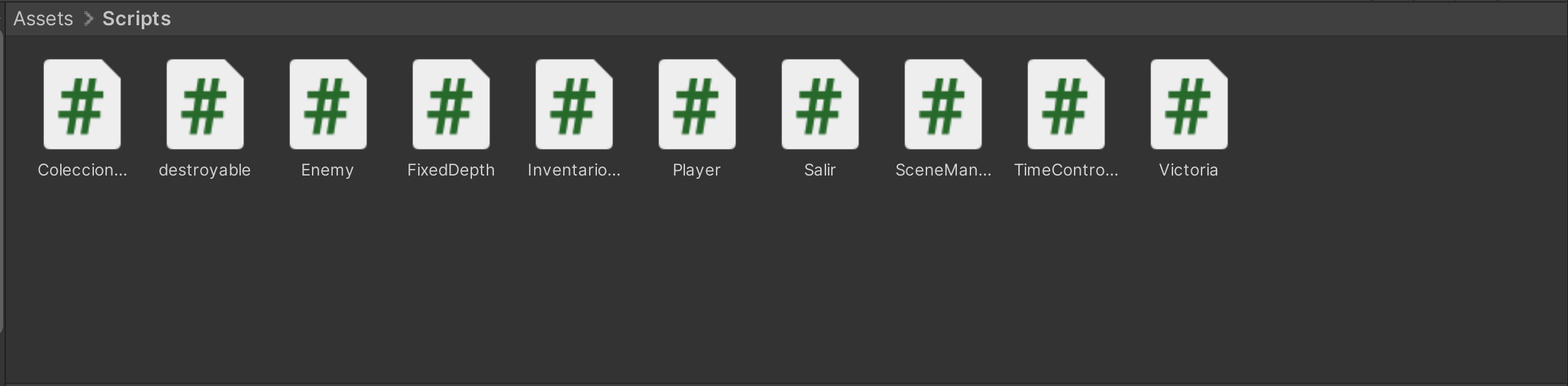
Santiago de los Caballeros

Rep. Dom.

**CAPÍTULO III: DESARROLLO**

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

**Scripts**



**Coleccionables**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class Coleccionables : MonoBehaviour

{

    private int contador, numero;

    public Text contadorDestroy;

    AudioSource fuenteDeAudio;

    public AudioClip bling;

    public float timeForDisable;

    public Renderer rend;

    void Start(){

        rend=GetComponent<Renderer>();

        rend.enabled=true;

        //fuenteDeAudio=GetComponent<AudioSource>();

    }

    void OnTriggerEnter2D(Collider2D other)

    {

        if(other.tag=="Player"){

                //other.enabled= false;

                fuenteDeAudio=GetComponent<AudioSource>();

                fuenteDeAudio.clip = bling;

                fuenteDeAudio.Play();

                rend.enabled=false;

                Destroy(gameObject,fuenteDeAudio.clip.length);

                int.TryParse(contadorDestroy.text, out numero);

                Debug.Log(numero);

                numero = numero + 1;

                setTextoContador(numero);

        }

    }

    void destruir(){

    }

    void setTextoContador(int num){

        contador= numero;

        contadorDestroy.text = contador.ToString();

    }

}

**Destroyables(Para la destrucción de objetos)**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class destroyable : MonoBehaviour

{

    // Start is called before the first frame update

    //public InventarioDestruidos inventario;

    public string destroyState;

    public float timeForDisable;

    private int contador, numero;

    public Text contadorDestroy;

    AudioSource fuenteDeAudio;

    public AudioClip crash;

    Animator anim;

    void Start()

    {

        anim=GetComponent<Animator>();

        fuenteDeAudio=GetComponent<AudioSource>();

       // inventario = GameObject.FindGameObjectWithTag("Player").GetComponent<InventarioDestruidos>();

    }

    IEnumerator OnTriggerEnter2D(Collider2D col){

        if(col.tag=="Attack"){

            anim.Play(destroyState);

            yield return new WaitForSeconds(timeForDisable);

            foreach(Collider2D c in GetComponents<Collider2D>()){

                c.enabled=false;

                fuenteDeAudio.clip = crash;

                fuenteDeAudio.Play();

                //inventario.cantidad = inventario.cantidad + 1;

                int.TryParse(contadorDestroy.text, out numero);

                Debug.Log(numero);

                numero = numero + 1;

                setTextoContador(numero);

            }

        }

    }

    void setTextoContador(int num){

        contador= numero;

        contadorDestroy.text = contador.ToString();

    }

    // Update is called once per frame

    void Update()

    {

        AnimatorStateInfo stateInfo= anim.GetCurrentAnimatorStateInfo(0);

        if(stateInfo.IsName(destroyState)&& stateInfo.normalizedTime>=1){

            Destroy(gameObject);

        }

    }

}

**Fixed Depth –Profundidad al personaje**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class FixedDepth : MonoBehaviour

{

    // Start is called before the first frame update

     public bool fixEveryFrame;

     SpriteRenderer spr;

    void Start()

    {

        spr=GetComponent<SpriteRenderer>();

        spr.sortingLayerName= "character\_0";

        spr.sortingOrder= Mathf.RoundToInt(-transform.position.y \*100);

    }

    // Update is called once per frame

    void Update()

    {

            if(fixEveryFrame){

                spr.sortingOrder=Mathf.RoundToInt(-transform.position.y \*100);

            }

    }

}

**Player**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Player : MonoBehaviour

{

    public float speed=4f;

    Animator anim;

    Rigidbody2D rb2d;

    Vector2 mov;

   //new

   CircleCollider2D attackCollider;

    // Start is called before the first frame update

    void Start()

    {

        anim=GetComponent <Animator>();

        rb2d=GetComponent<Rigidbody2D>();

        //new

        attackCollider=transform.GetChild(0).GetComponent<CircleCollider2D>();

        attackCollider.enabled=false;

    }

    // Update is called once per frame

    void Update()

    {

        /\*Vector3 mov=new Vector3(

            Input.GetAxisRaw("Horizontal"),

            Input.GetAxisRaw("Vertical"),

            0

        );\*/

        mov = new Vector2(

            Input.GetAxisRaw("Horizontal"),

            Input.GetAxisRaw("Vertical")

        );

        /\*transform.position = Vector3.MoveTowards(

            transform.position,

            transform.position + mov,

            speed \* Time.deltaTime

        ); \*/

        if( mov !=Vector2.zero){

            anim.SetFloat("movX",mov.x);

            anim.SetFloat("movY",mov.y);

            anim.SetBool("walking",true);

        }else{

            anim.SetBool("walking",false);

        }

        //detectar ataque

        AnimatorStateInfo stateInfo=anim.GetCurrentAnimatorStateInfo(0); //new

        bool attacking= stateInfo.IsName("player\_attack");

        if(Input.GetKeyDown("space") && !attacking){

            anim.SetTrigger("attacking");

        }

        //new

        if(mov !=Vector2.zero) attackCollider.offset=new Vector2(mov.x/2, mov.y/2);

        //new

        if(attacking){

            float playbackTime= stateInfo.normalizedTime;

            if(playbackTime >0.33 && playbackTime <0.66) attackCollider.enabled=true;

            else attackCollider.enabled=false;

        }

    }

    void FixedUpdate(){

            rb2d.MovePosition(rb2d.position+mov\*speed\*Time.deltaTime);

        }

}

**Salir**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Salir : MonoBehaviour

{

    public void Exit(){

        Application.Quit();

    }

}

**Scene Management**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class SceneManagement : MonoBehaviour

{

    public void LoadScene(string SceneName){

        SceneManager.LoadScene(SceneName);

    }

}

**Time Controller**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SceneManagement;

public class TimeController : MonoBehaviour

{

    [SerializeField] int min,seg;

    [SerializeField] Text tiempo;

    private float restante;

    public bool enMarcha;

    private void Awake()

    {

        restante = (min \* 60) + seg;

        enMarcha=true;

    }

    // Update is called once per frame

    void Update()

    {

        if(enMarcha)

        {

            restante -= Time.deltaTime;

            if(restante < 1)

            {

                enMarcha=false;

                SceneManager.LoadScene("Perder");

                //iniciar el juego

            }

            int tempMin = Mathf.FloorToInt(restante / 60);

            int tempSeg = Mathf.FloorToInt(restante % 60);

            tiempo.text = string.Format("{00:00}:{01:00}",tempMin,tempSeg);

        }

    }

}

**Victoria**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SceneManagement;

public class Victoria : MonoBehaviour

{

    public Text contdestroy,contcollect, victory, message;

    AudioSource fuenteDeAudio;

    public AudioClip win, error;

    public int numd, numc;

    public Renderer rend;

    // Start is called before the first frame update

    void Start()

    {

        rend=GetComponent<Renderer>();

        rend.enabled=true;

        //fuenteDeAudio= GetComponent <AudioSource>();

        message.text="";

        victory.text="";

    }

     void OnTriggerEnter2D(Collider2D other)

    {

        if(other.tag=="Player"){

            int.TryParse(contdestroy.text, out numd);

            Debug.Log(numd);

            int.TryParse(contcollect.text, out numc);

            Debug.Log(numc);

            if(numc == 25 && numd == 25 ){

                //other.enabled= false;

                fuenteDeAudio=GetComponent<AudioSource>();

                fuenteDeAudio.clip = win;

                fuenteDeAudio.Play();

                rend.enabled=false;

                Destroy(gameObject,fuenteDeAudio.clip.length);

                SceneManager.LoadScene("Victoria");

            }

            else {

                message.text="NIVEL INCOMPLETO";

                fuenteDeAudio=GetComponent<AudioSource>();

                fuenteDeAudio.clip = error;

                fuenteDeAudio.Play();

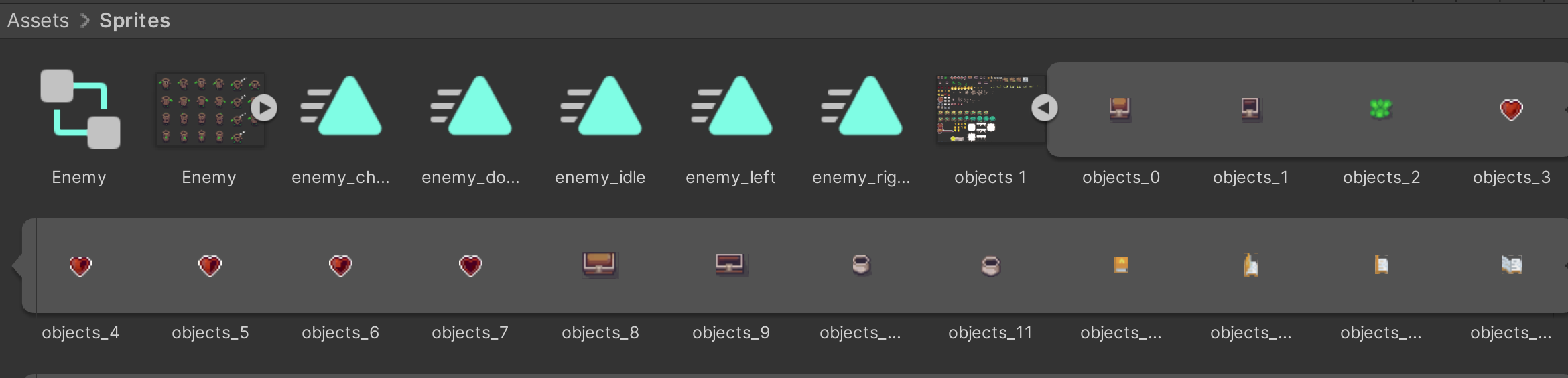
            }

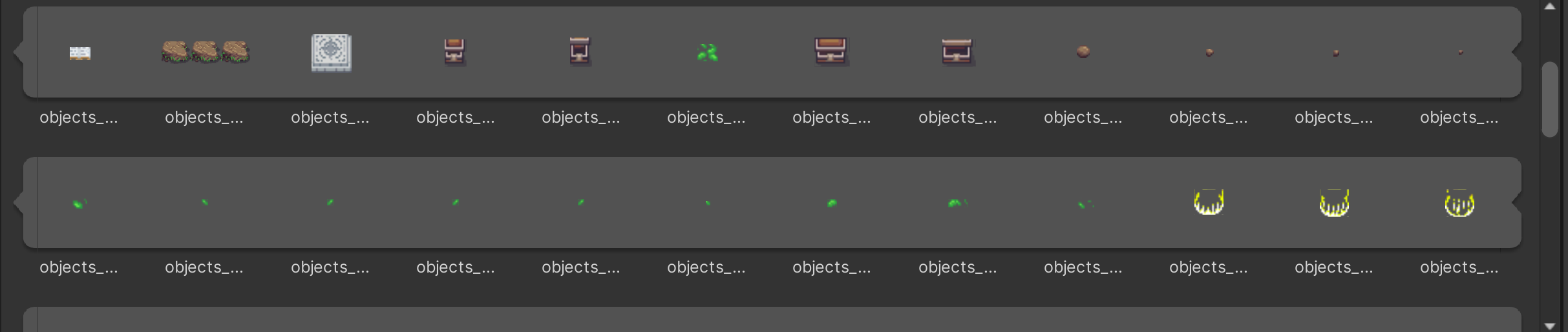
        }

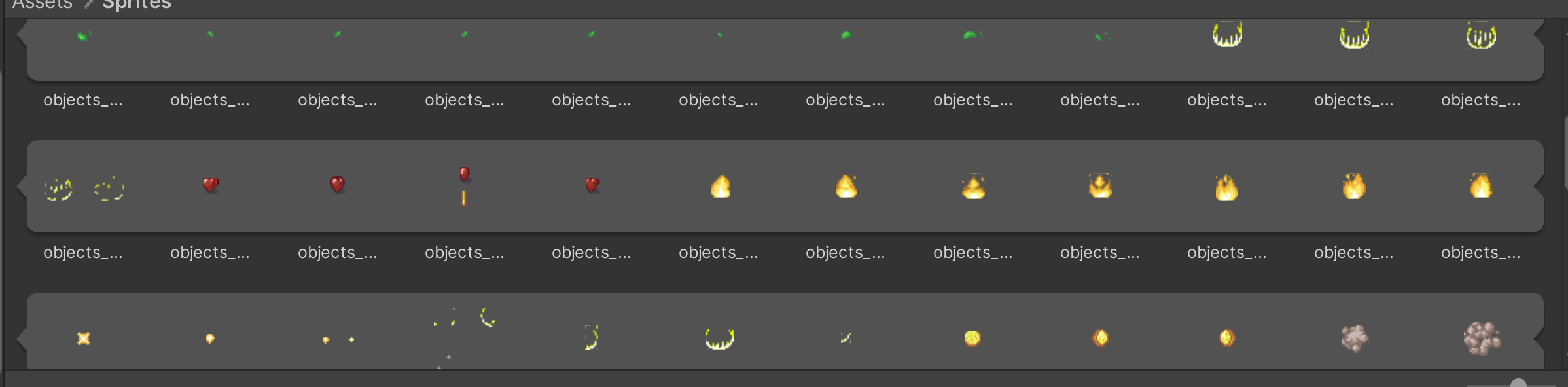
    }

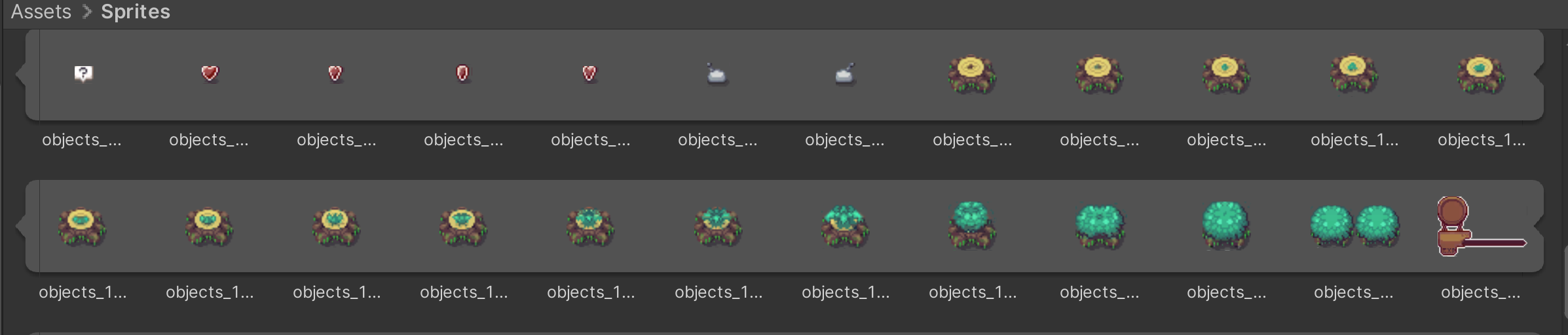
}

**Sprites**

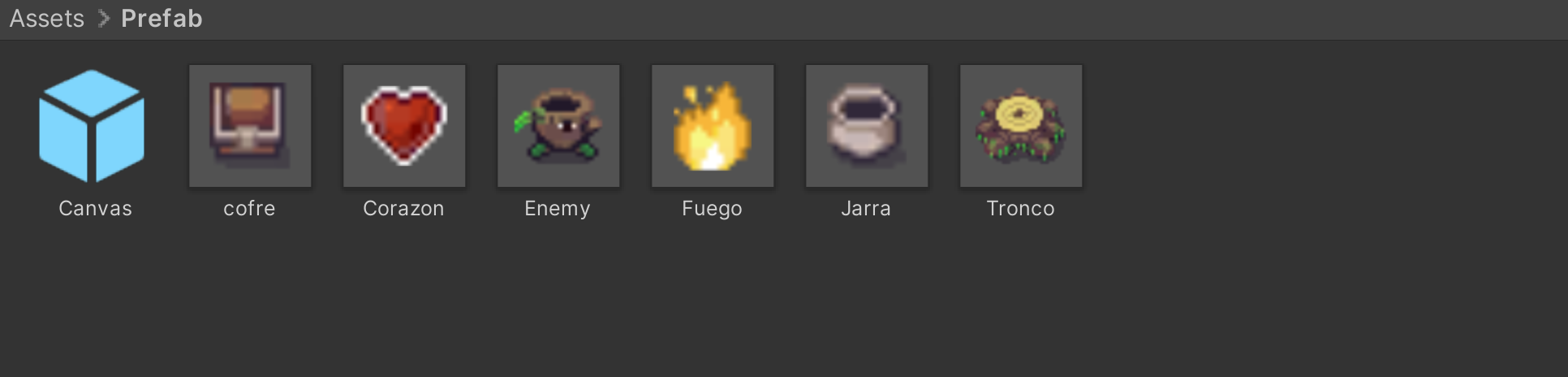




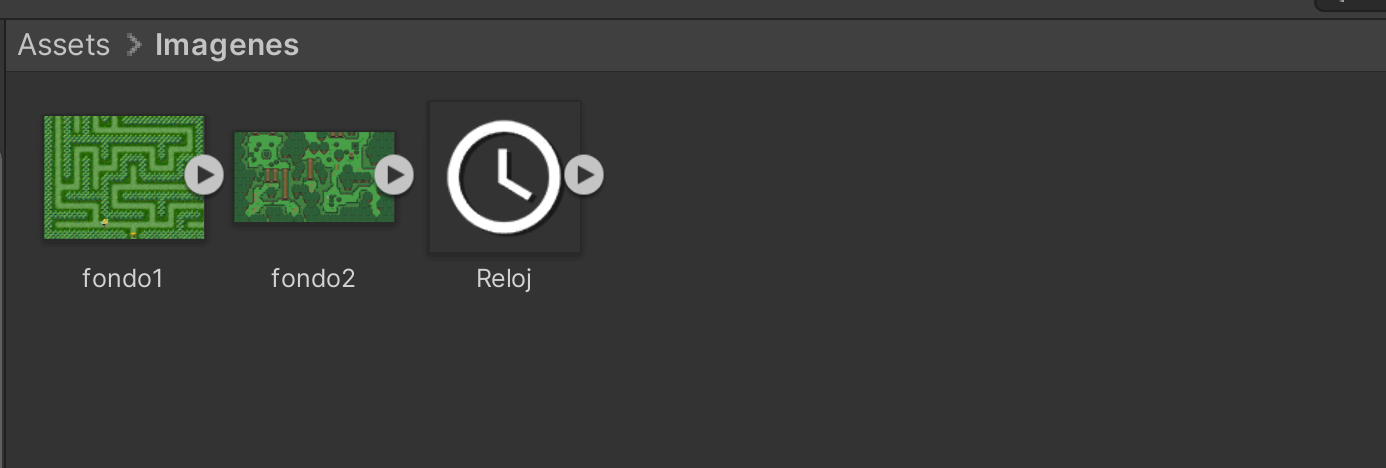




**Prefabs**



**Imágenes**



**3.2 Prototipos**

Lo-Fi

Contiene el nivel principal con los movimientos del personaje y los objetos

<https://yamigomez14.itch.io/lo-fi-stranger-maze-v1>

**3.3 Perfiles de Usuarios**

El público para el cual se diseña este videojuego es para:

* Personas mayores de 10 años.
* Personas con poco conocimiento de videojuegos.
* Personas con buena agilidad y sentido de la orientación.

**3.4 Usabilidad**

Con respecto a la usabilidad de este videojuego, hasta el momento por parte de los diseñadores, la misma ha sido exitosa. Se ha tratado de hacer de este lo más explicito y comprensible en cuanto a la interfaz de usuario y mecánica, para que así incluso los controles sean los que usualmente se utilizan y sea más fácil de aprender.

**3.5 Test**

**Test 1:**

|  |  |
| --- | --- |
| **Género** | Masculino |
| **Edad** | 14 |
| **Nivel de estudios** | Básico |

Individuo 1: Alberto

Resultados

|  |  |
| --- | --- |
| **Puntos a evaluar** | **Puntuación(en base a 5)** |
| Jugabilidad | 4 |
| Dificultad | 1 |
| Control de personaje | 5 |
| Guía de usuario | 3 |
| Información proporcionada por el juego | 2 |
| Diseño visual | 4 |
| Coherencia | 4 |

**Test 2:**

|  |  |
| --- | --- |
| **Género** | Masculino |
| **Edad** | 16 |
| **Nivel de estudios** | Bachiller |

Individuo 1: Anthony

Resultados

|  |  |
| --- | --- |
| **Puntos a evaluar** | **Puntuación(en base a 5)** |
| Jugabilidad | 4.5 |
| Dificultad | 1 |
| Control de personaje | 5 |
| Guía de usuario | 3 |
| Información proporcionada por el juego | 2 |
| Diseño visual | 4 |
| Coherencia | 4 |

Test 3:

|  |  |
| --- | --- |
| **Género** | Femenino |
| **Edad** | 19 |
| **Nivel de estudios** | Bachillerato |

Individuo 1: Karla

Resultados

|  |  |
| --- | --- |
| **Puntos a evaluar** | **Puntuación(en base a 5)** |
| Jugabilidad | 4.5 |
| Dificultad | 1 |
| Control de personaje | 5 |
| Guía de usuario | 4 |
| Información proporcionada por el juego | 1 |
| Diseño visual | 3 |
| Coherencia | 4 |

**3.6 Versiones de la Aplicación**

**V1. N.1** [**https://yamigomez14.itch.io/stranger-maze-v1**](https://yamigomez14.itch.io/stranger-maze-v1)