**UNIVERSIDAD TECNOLOGICA DE SANTIAGO**

**(UTESA)**

**SISTEMA CORPORATIVO**

**FACULTAD DE ARQUITECTURA E INGENIERÍA**



**Asignatura:**

Programación de Video Juegos

**Entrega Final**

**Facilitador:**

Iván Mendoza

**Realizado por:**

Yamilka Gómez Morán 2-18-0529

Wayddy Ashley Grullón 2-18-0269

Jean Ureña 2-17-1955

25 de agosto, 2022

Santiago de los Caballeros,

Rep. Dom.

[**CAPÍTULO I: VIDEOJUEGO Y HERRAMIENTAS DE DESARROLLO** 3](#_Toc112267250)

[**1.1** **Descripción:** 3](#_Toc112267251)

[**1.2** **Motivación** 4](#_Toc112267252)

[**1.2.1** **Originalidad de la idea** 4](#_Toc112267253)

[**1.2.2** **Estado del Arte** 4](#_Toc112267254)

[**1.3** **Objetivo general** 4](#_Toc112267255)

[**1.4** **Objetivos específicos** 4](#_Toc112267256)

[**1.5 Escenario** 4](#_Toc112267257)

[**1.6 Contenidos** 4](#_Toc112267258)

[**1.7 Metodología** 5](#_Toc112267259)

[**1.8 Arquitectura de la aplicación** 5](#_Toc112267260)

[**1.9 Herramientas de desarrollo** 5](#_Toc112267261)

[**CAPÍTULO II: DISEÑO E IMPLEMENTACIÓN** 6](#_Toc112267262)

[**2.1 Planificación** 6](#_Toc112267263)

[**Documentación Capítulo 1** 6](#_Toc112267264)

[**Documentación Capítulo 2** 6](#_Toc112267265)

[**Inicio de codificación** 6](#_Toc112267266)

[**Integración de los componentes del videojuego** 6](#_Toc112267267)

[**Finalización de la codificación** 6](#_Toc112267268)

[**Documentación Capítulo 4** 6](#_Toc112267269)

[**Publicación** 6](#_Toc112267270)

[**2.2 Diagramas y Casos de Uso** 6](#_Toc112267271)

[**2.3 Plataforma** 7](#_Toc112267272)

[**2.4 Género** 7](#_Toc112267273)

[**2.5 Clasificación** 7](#_Toc112267274)

[**2.6 Tipo de Animación** 7](#_Toc112267275)

[**2.7 Equipo de Trabajo** 7](#_Toc112267276)

[**2.8 Historia** 8](#_Toc112267277)

[**2.9 Guión** 8](#_Toc112267278)

[**2.10 Storyboard** 9](#_Toc112267279)

[**2.11 Personajes** 9](#_Toc112267280)

[**2.13 Mecánica del Juego** 12](#_Toc112267281)

[**CAPÍTULO III: DESARROLLO** 13](#_Toc112267282)

[**3.1 Capturas de la Aplicación (Documentación completa del desarrollo, Scripts, Sprites, Prefabs e imágenes)** 13](#_Toc112267283)

[**3.2 Prototipos** 23](#_Toc112267284)

[**3.3 Perfiles de Usuarios** 23](#_Toc112267285)

[**3.4 Usabilidad** 23](#_Toc112267286)

[**3.5 Test** 23](#_Toc112267287)

[**3.6 Versiones de la Aplicación** 25](#_Toc112267288)

[**CAPÍTULO IV: PUBLICACIÓN** 25](#_Toc112267289)

[**4.1 Requisitos de la instalación** 25](#_Toc112267290)

[**4.2 Instrucciones de Uso** 25](#_Toc112267291)

[**4.3 Bugs** 25](#_Toc112267292)

[**4.4 Proyección a Futuro** 25](#_Toc112267293)

[**4.5 Presupuesto** 25](#_Toc112267294)

[**4.6 Análisis de Mercado** 26](#_Toc112267295)

[**4.7 Viabilidad** 26](#_Toc112267296)

[**CONCLUSIONES** 26](#_Toc112267297)

[**REFERENCIAS BIBLIOGRÁFICAS** 28](#_Toc112267298)

**INTRODUCCIÓN**

# **CAPÍTULO I: VIDEOJUEGO Y HERRAMIENTAS DE DESARROLLO**

# **Descripción:**

Stranger Maze es un videojuego que puede categorizarse como un RPG estratégico donde el personaje principal vive en un bosque y debe salir a buscar suplementos al exterior que es un laberinto, este deberá superar los obstáculos del bosque como ríos, rocas, ramas y recolectar los objetos requeridos hasta poder salir para poder superar el nivel.

# **Motivación**

# **Originalidad de la idea**

El motivo de creación de este videojuego se debe a la popular de los RPGS y también al interés de fusionar diversos géneros en un solo juego ya que lo que se busca es que el jugador pueda hacer cualquier cosa en el juego pero que deba pensar antes de realizar las acciones y que requiera tener habilidades lógicas, a su vez que este se divierta recolectando tesoros a lo largo del trayecto.

# **Estado del Arte**

Pretende y busca agradar a los jugadores a través de diseños intuitivos, fáciles y rápidos en cuanto a recolección de objetos.

# **Objetivo general**

Capturar la atención del jugador mediante el establecimiento de misiones para encontrar objetos específicos dentro de un laberinto localizado en un bosque.

# **Objetivos específicos**

* Recorrer el laberinto hasta el final.
* Encontrar los objetos necesarios para pasar de nivel.
* Crear una experiencia única e intrigante para el jugador.

# **1.5 Escenario**

Se establecerá un laberinto por nivel que estará localizado dentro de un bosque.

# **1.6 Contenidos**

Stranger Maze contiene los siguientes objetos:

Troncos

Rocas

Linternas

Pozos de agua

Cofres

# **1.7 Metodología**

El personaje representado por un chico, ubicado en el bosque, debe de buscar la manera de sobrevivir con los alimentos y objetos que brinda la naturaleza, ya que, el suministro que contiene está a punto de acabarse. Para sobrevivir debe:

* Recolectar frutos.
* Recolectar objetos para generar fuego y luz.

# **1.8 Arquitectura de la aplicación**

Las condiciones que se deben cumplir para el desarrollo del videojuego, son:

* El juego debe de tener un objetivo, que define el final de un nivel.
* Los escenarios deberán tener cierto grado de realismo.
* El jugador tendrá un tiempo límite para recolectar objetos del juego, de lo contrario pierde la partida.
* Versiones de demostración para que los jugadores puedan presenciar las funcionalidades del juego.

# **1.9 Herramientas de desarrollo**

El entorno de desarrollo está compuesto por la siguientes herramientas:

Unity, utilizado para crear los objetos de juegos, colisiones, movimientos del personaje, diálogos, entre otros.

C Sharp, para ser utilizado en conjunto con Unity, para ofrecerle un funcionamiento a cada elemento, crear temporizadores, y realizar cambios de escena.

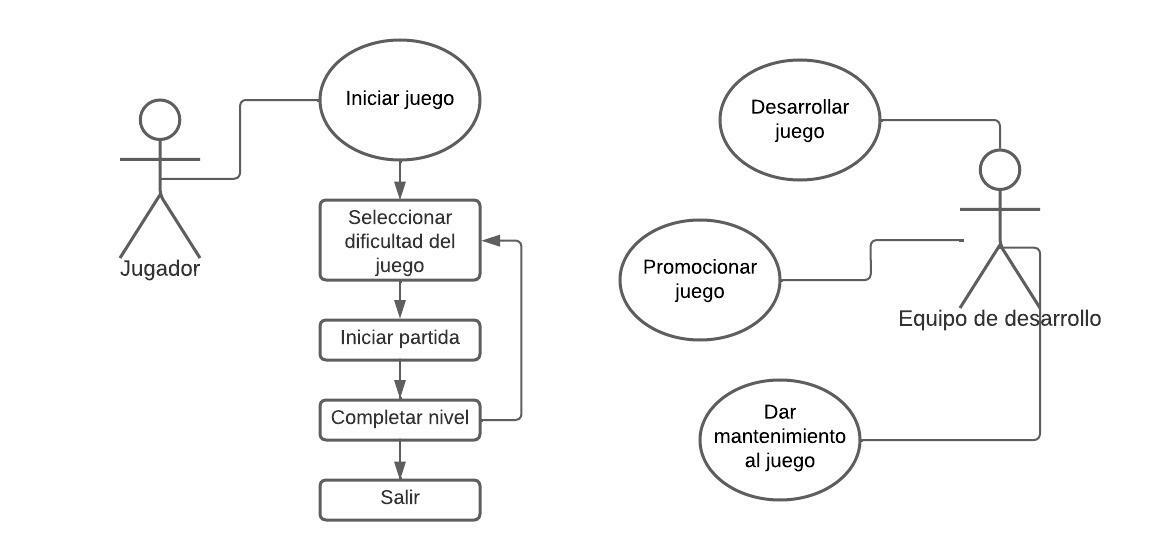
Asset Store, tienda de unity, para descargar escenarios en tres dimensiones, el cual reduce el tiempo de desarrollo y ofrece entornos interactivos.

# **CAPÍTULO II: DISEÑO E IMPLEMENTACIÓN**

# **2.1 Planificación**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 07/08/2022 | | 11/08/2022 | 16/08/2022 | 22/08/2022 | 24/08/2022 |
| **Documentación Capítulo 1** |  |  | |  |  |  |
| **Documentación Capítulo 2** |  |  | |  |  |  |
| **Inicio de codificación** |  |  | |  |  |  |
| **Integración de los componentes del videojuego** |  |  | |  |  |  |
| **Finalización de la codificación** |  |  | |  |  |  |
| **Documentación Capítulo 4** |  |  | |  |  |  |
| **Publicación** |  |  | |  |  |  |

# **2.2 Diagramas y Casos de Uso**



# **2.3 Plataforma**

El juego será desarrollado para Windows, posteriormente se evaluará la posibilidad de adaptarlo para dispositivos móviles.

# **2.4 Género**

El género del videojuego es arcade.

# **2.5 Clasificación**

Clasificación Everyone (Todos)



# **2.6 Tipo de Animación**

En dos dimensiones (2D).

# **2.7 Equipo de Trabajo**

* **Ingeniero de audio:** Wayddy Ashley.
* **Diseñadores:** Yamilka Guzmán.
* **Ilustradores:** Wayddy Ashley.
* **Programadores:** Jean Ureña.
* **Animadores:** Yamilka Guzmán / Jean Ureña

# 

# **2.8 Historia**

El chico Jesús el cual se encuentra atrapado en una cabaña en el bosque, donde pretendía pasar una temporada del año, debe intentar sobrevivir en esa localidad con los productos y herramientas propias de la naturaleza. El bosque aokigahara comprende un terreno repleto de laberintos, donde el personaje principal deberá desplazarse por el mismo, en busca de mapas, cofres, herramientas de supervivencias y alimentos, los cuales tienen como finalidad asegurar la vida del personaje y ayudarle afrontar los retos o obstáculos que se originarán a lo largo de su estadía en el bosque. El personaje deberá administrar sus recursos, de manera que pueda superar los obstáculos que se presenten en el trayecto con los medios que contiene al momento de enfrentar el evento, esto es, porque el juego contiene atributos, tales como, temporizadores, barra de energía y herramientas de seguridad, las cuales irán limitando las funcionalidades del jugador a lo largo del juego, donde mantener cierto control ayudará a sobrepasar ciertos espacios que están estructurados como misiones.

# **2.9 Guión**

Jesús, cansado de gestionar proyectos, tratar con clientes y problemas rutinarios, decide tomar una temporada de descanso, y se dirige a un bosque alejado de la civilización, con la finalidad de descansar, disfrutar de la naturaleza y olvidar frustraciones.

Su destino fue una cabaña ubicada en el bosque Aikogahara. Un día, empieza a caminar por el bosque, luego de caminar por varias horas y decide regresar a casa, descubre que se encuentra perdido en un laberinto, que impone una traba para regresar a casa, porque cada trayecto que escoge lo lleva a otro lugar desconocido. Cuando al fin logra llegar a la cabaña, inicia su incertidumbre de salir del bosque ya que el mismo puede ser un lugar peligroso, solitario y que impone un reto salir del mismo.

Cada día, al salir el sol empieza una caminata con la finalidad de entender lo que parece tener una estructura de crucigrama, en sus caminatas diarias descubre objetos fuera de lo normal, alimentos que le brindan vitalidad, objetos de supervivencias y mapas, los cuales serán de utilidad para cumplir los retos y salir del laberinto.

# **2.10 Storyboard**

Como tal no se cuenta con un storyboard, básicamente se está utilizando un personaje en 2D que posteriormente va a ser modificado por otro sprite, es una versión de prueba.

# **2.11 Personajes**

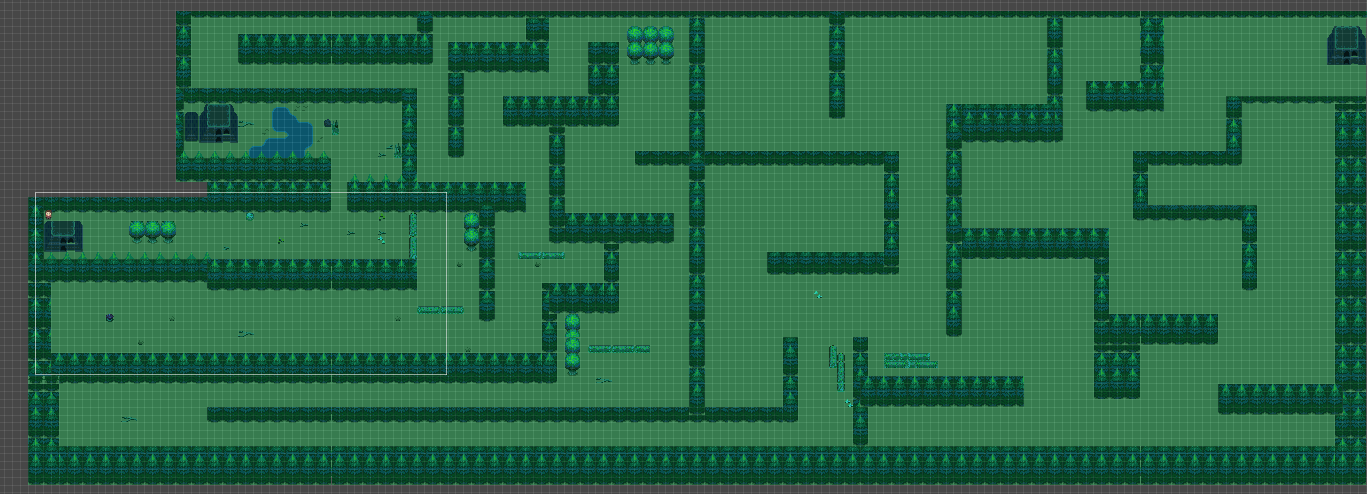
Como personaje tenemos a Jesús que es un joven de contextura física delgada, cabello marrón, tamaño pequeño y el mismo es representado en 2D.

 **2.12 Niveles**

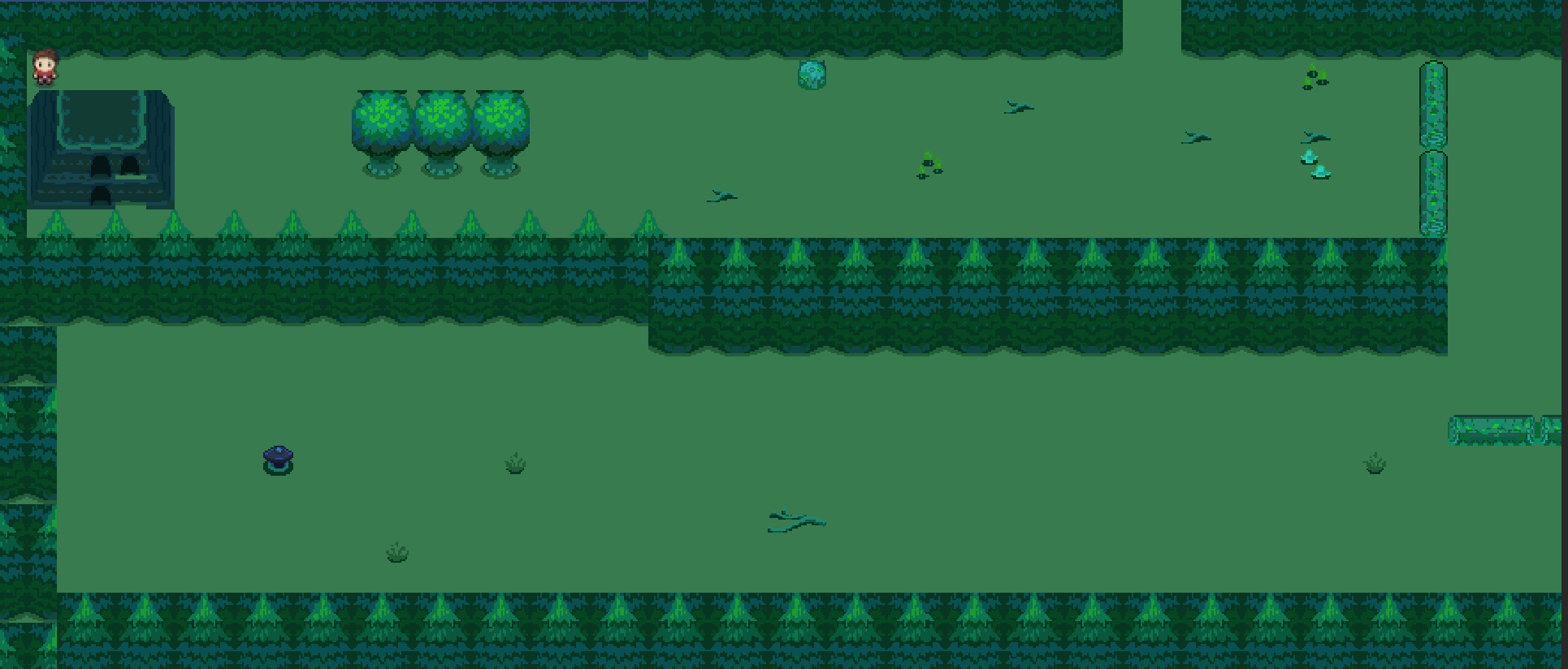
El juego cuenta con 3 niveles, con un grado de dificultad distinta

* **Grado 1 Genio Superior**

**Escena**

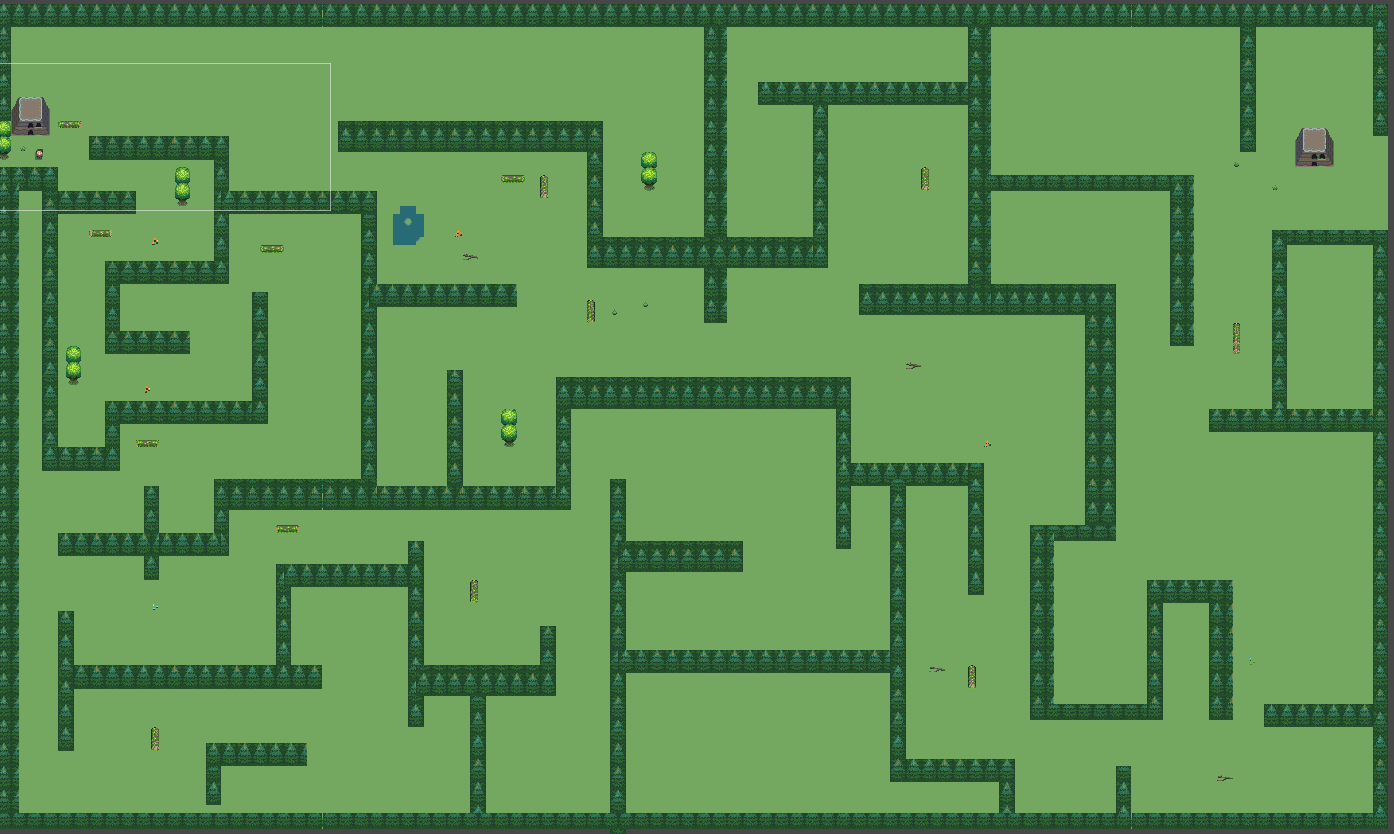


**Inicio del Nivel**

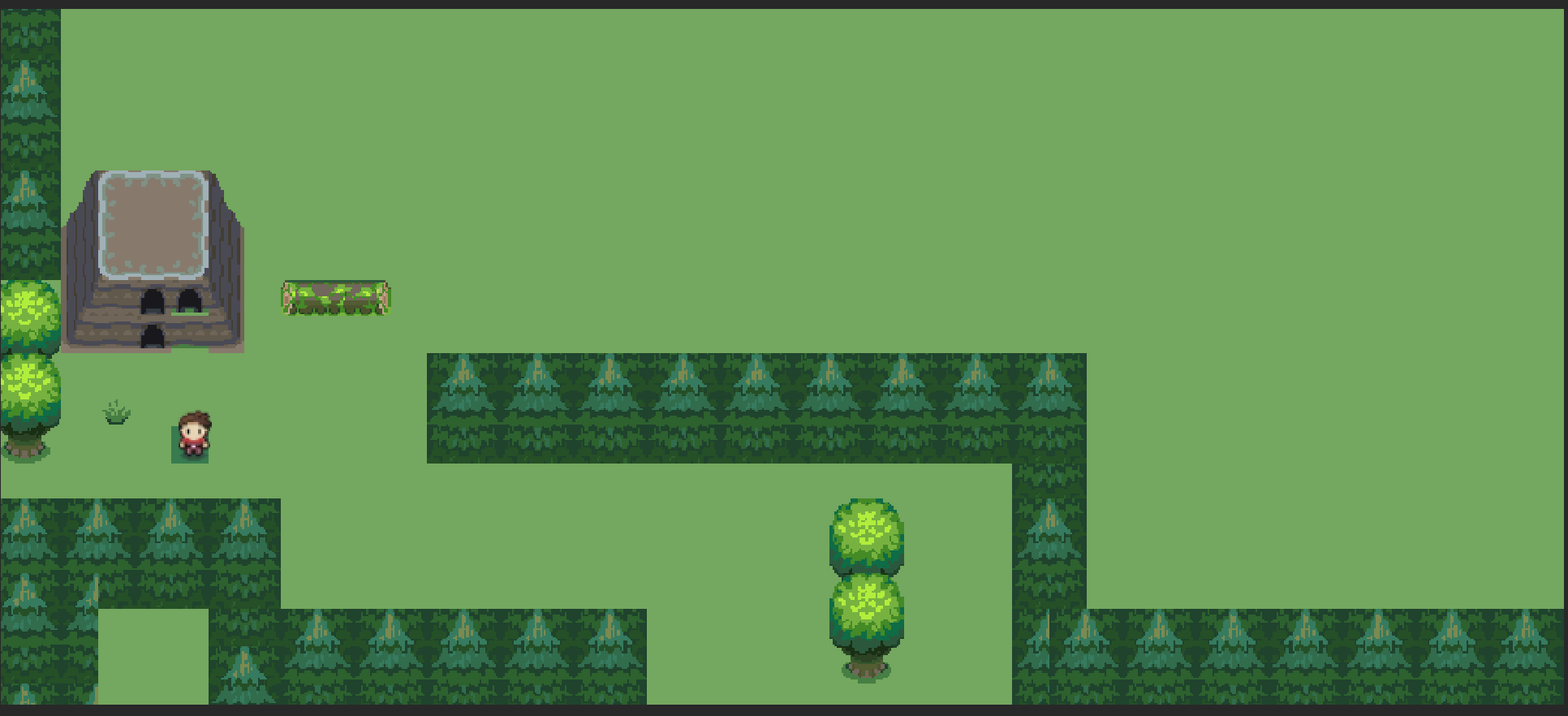


* **Grado 2 Intermedio**

**Imagen del diseño de la escena:**

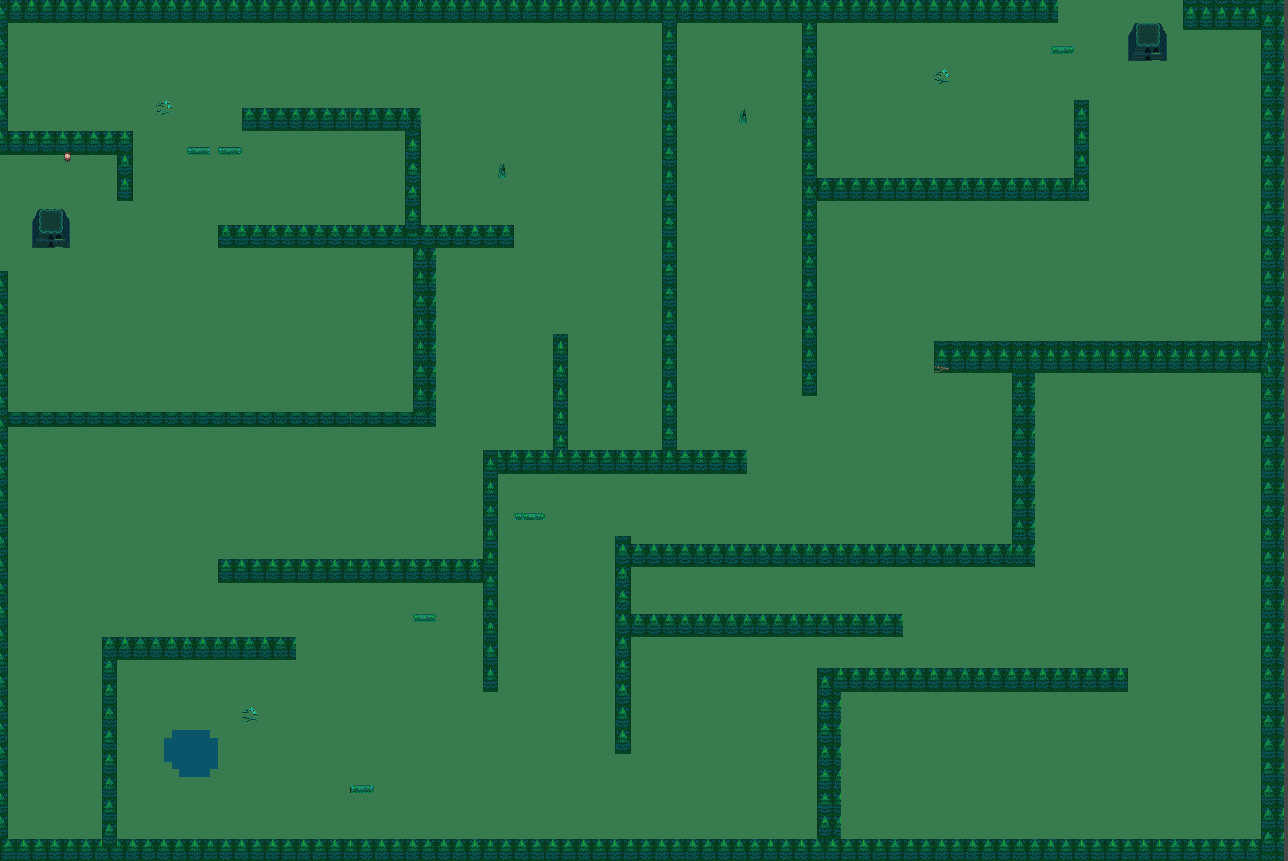


**Inicio del nivel 2:**



* **Grado 3 Fácil**

**Escena**



**Inicio del Nivel 3**



# **2.13 Mecánica del Juego**

Se utilizarán las teclas:

Arriba para saltar

Abajo para agacharse

Lado para desplazarse

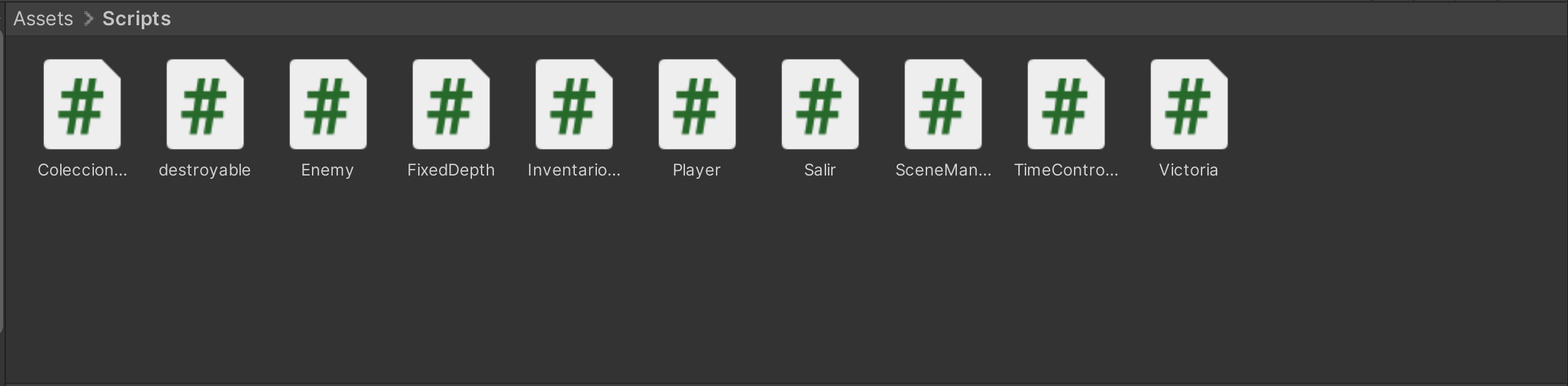
Espacio para usar espada

Será un laberinto libre, donde el personaje deberá atravesar las trampas en el mismo para salir.

# **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();

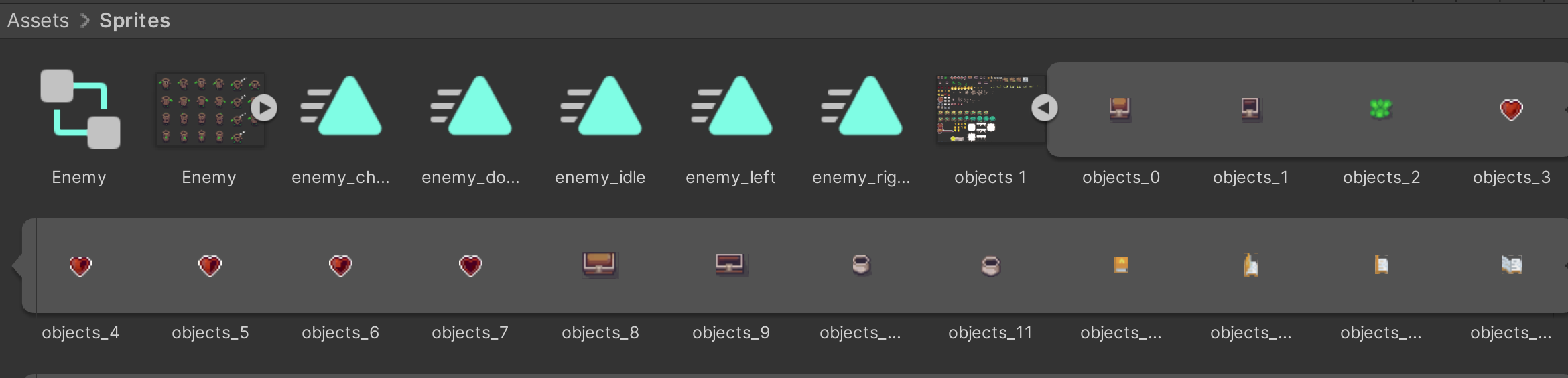
            }

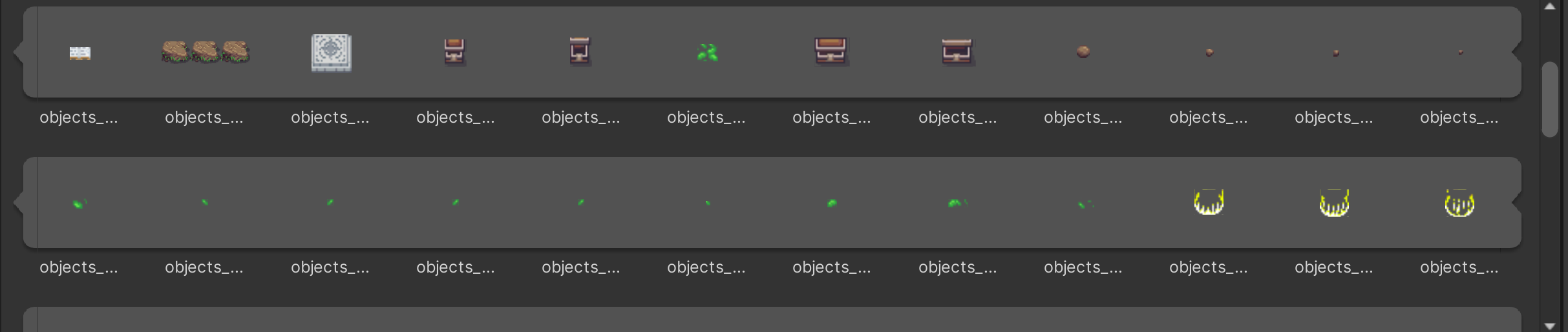
        }

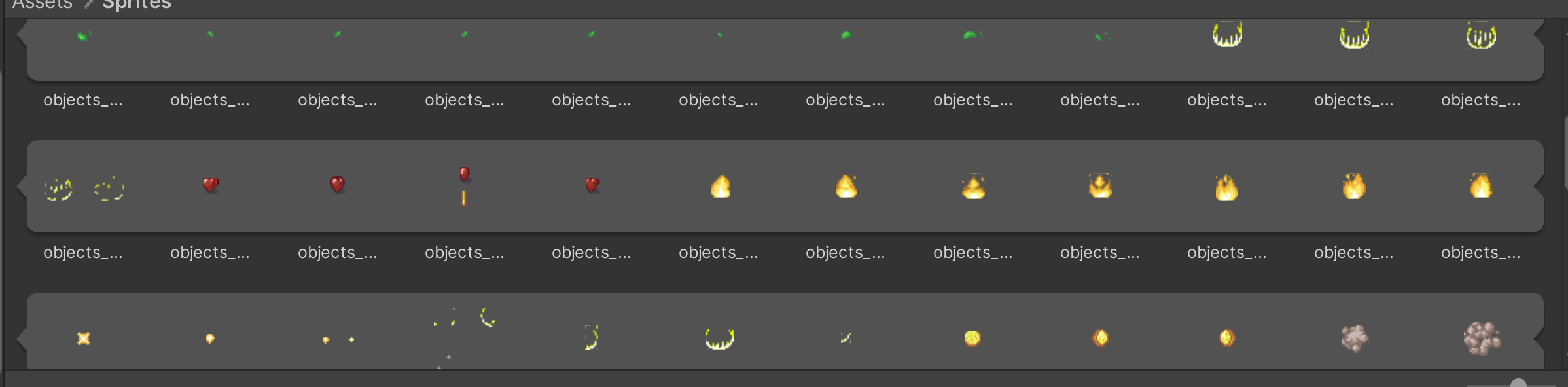
    }

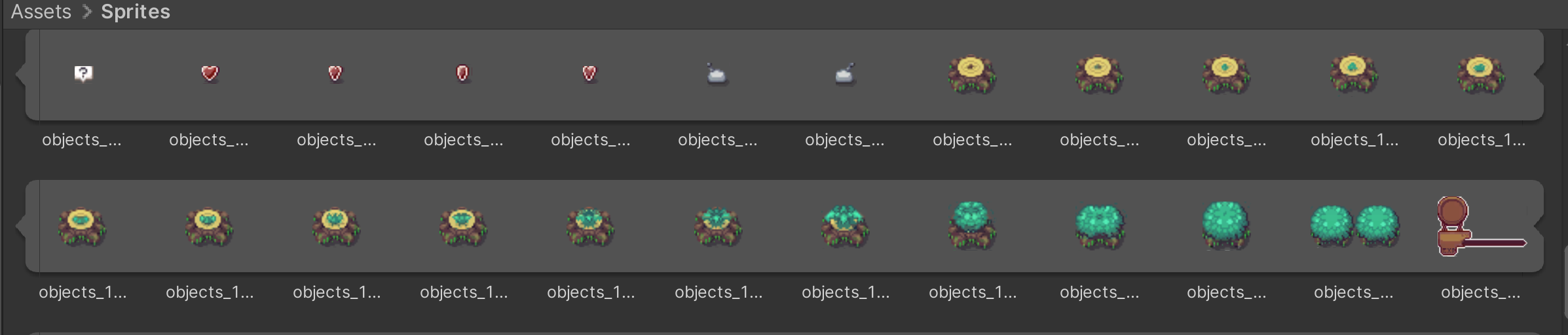
}

**Sprites**

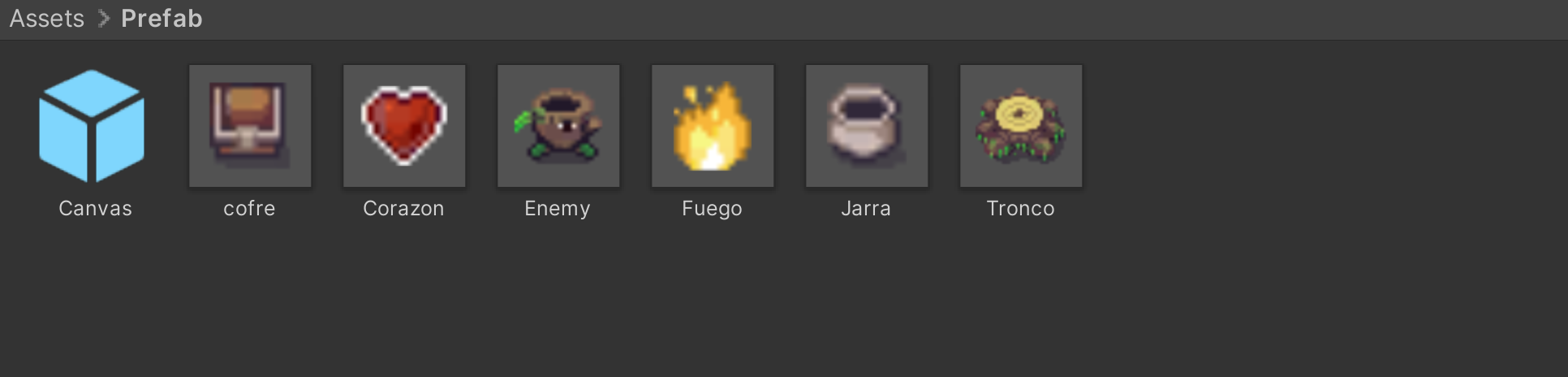




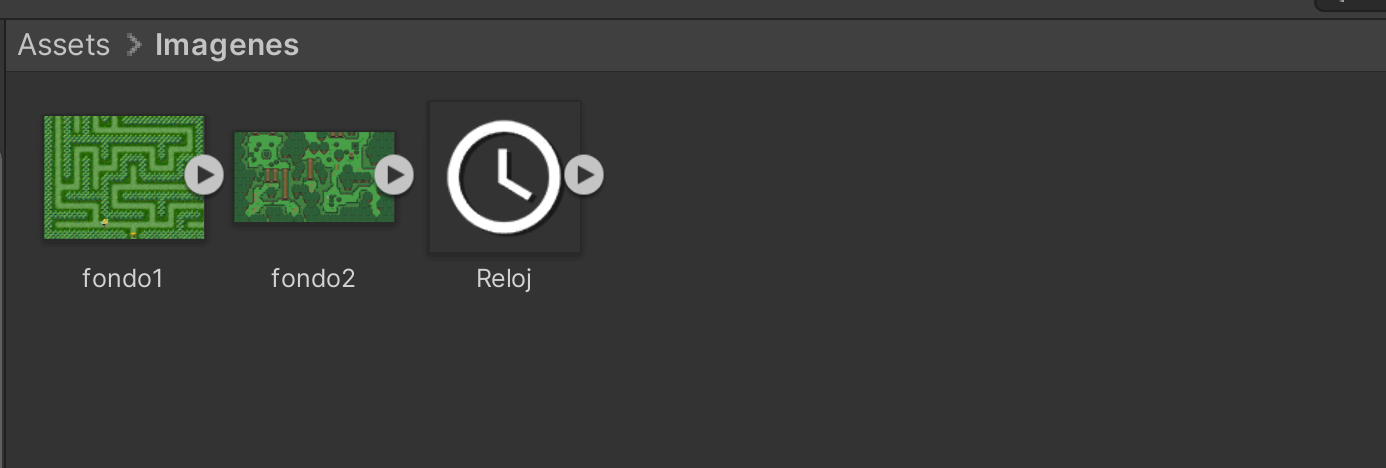




**Prefabs**



**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)

# **CAPÍTULO IV: PUBLICACIÓN**

# 

# **4.1 Requisitos de la instalación**

* RAM: 4GB (mínimo).
* Espacio en disco duro: 2GB (como mínimo, es lo recomendable).
* Tarjeta gráfica: con una integrada es suficiente.
* CPU: Intel i3 (mínimo).

# **4.2 Instrucciones de Uso**

En cuanto a los controles, tenemos los siguientes:

* Las flechas del teclado para mover al personaje.
* Barra de espacio para golpear.
* Barra de espacio para sacar espada

# **4.3 Bugs**

* Algunas detecciones de colisiones pueden fallar.
* Como la cámara sigue al personaje por todo el trayecto del laberinto puede que se torne un poco lenta con la rapidez del mismo.
* Al interactuar con algunos objetos puede que haya un bug de conteo.

# **4.4 Proyección a Futuro**

Como equipo se tiene a futuro modificar los posibles bugs que surgen y además modificar las funcionalidades de los coleccionables, añadir enemigos que hagan más difícil el trayecto de supervivencia por lo que al tener todo listo, quizás haya posibilidad de venta.

# **4.5 Presupuesto**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Costo de producción del videojuego**  Unity personal es posible utilizarlo si genera menos de 100,000 dólares al año. | **Tiempo que le tomará a cada encargado terminar su parte** | **Usuarios tests** | **Perfiles de los profesionales** | **Costo de cada perfil profesional** |
|  | Yamilka Gómez– 10 días | Alberto | Diseñador creativo | RD$ 20,000 |
|  | Jean Ureña – 15 días | Anthony | Programador | RD$ 35,000 |
|  | Wayddy Grullón – 10 días | Karla | Usuarios tests | RD$ 2,000 |
|  |  | Marcos | Ingeniero de sonido | RD$ 10,000 |
|  |  | Elvis | Animadores | RD$ 20,000 |
|  |  |  |  |  |

# **4.6 Análisis de Mercado**

Como es conocido por muchos, los juegos RPG y de supervivencia son bastante entretenidos, por lo que en el futuro se pretende adaptar a un modo online donde los jugadores podrán interactuar entre sí, tener zonas, hacer misiones y adaptarse más a los juegos que actualmente están en el mercado, debido a que hay mucha competitividad en cuanto a grandes juegos.

Del mismo modo lo que se espera es que los juegos de este estilo crezcan en un 9.5% lo cual es algo bastante grande.

# **4.7 Viabilidad**

Por ahora, el juego solo está disponible para plataforma PC, más adelante y que también es una de las proyecciones que tenemos a futuro, desarrollarlo para dispositivos móviles.

# **CONCLUSIONES**

* **Yamilka Gómez**:

En relación a Stranger Maze, se puede decir que para mi persona a pesar de que es un juego muy simple fue un desafío, debido a que básicamente no sabía nada de programación de videojuegos, ni del entorno de desarrollado usado que fue Unity.

Puedo concluir que el proyecto fue un medio de aprendizaje y de entrega, donde no solo se obtuvo un juego como resultado sino, capacidad de trabajo en equipo, análisis, pensamiento lógico y básicamente mucha constancia y persistencia.

Realizar los proyectos y este juego, requiere de habilidades y puedo decir que programar n videojuego no es fácil y se necesita tiempo de dedicación, pero es posible llevarlo a cabo.

Fue una muy buena experiencia a nivel general.

* **Jean Ureña**

Los videojuegos abarcan una amplía área de la ingeniería de software, debido a que esta se debe apoyar de los fundamentos de este campo de estudio, porque gran parte del desarrollo del videojuego es similar a la creación de un sistema empresarial, ya que debe seguir un plan de proyecto, recopilar requisitos, diseñar diagramas de procesos, desarrollar el código fuente del programa, hacer revisiones, etc. La rama de los videojuegos representa una alta demanda en la población actual, existen distintas herramientas tecnológicas ayudan a obtener los resultados que esperamos de un videojuego, tales como, la interactividad, la usabilidad, la animación, entre otras características que aumenta el interés de los usuarios por el videojuego, algunas de estas herramientas son Unity, C#, Assets Store, la misma han sido utilizadas para desarrollar nuestro videojuego Stranger Maze.

El conocimiento obtenido durante el período del curso nos ha servido para desarrollar nuestro propio videojuego, el cual cumple con una serie de condiciones, tales como, animaciones, efectos de sonido, retos, laberintos, temporizadores y objetos de juego, las cuales permiten que el jugador se sienta atraído al momento de ejecutar el juego.

Referente a la asignatura, esta nos ha proporcionado interesantes conocimientos, permitiéndonos conocer la estructura que compone un videojuego, además aumentando nuestra creatividad para desarrollar historias, guiones, escenarios, misiones, de manera que pudimos desarrollar un videojuego que cumple con el objetivo de captar el interés del usuario.

* **Wayddy Ashley Grullón**

El desarrollo de videojuegos es una de las industrias más importantes y lucrativas de la actualidad debido a su diversidad, en el caso de este juego ofrecemos una experiencia única que te lleva a poner en funcionamiento distintas habilidades como son la lógica, la atención al detalle y el manejo del tiempo.

Se cumplió con los objetivos planteados inicialmente y personalmente aprendí mucho sobre el manejo de escenas, su diseño y su programación, además de los efectos sonoros que son los que dictan las emociones del juego.

# **REFERENCIAS BIBLIOGRÁFICAS**

**ProCamera2D**: <http://www.procamera2d.com/>

**Librería de audio**: <https://pixabay.com/es/music/>

<http://www.sonidosmp3gratis.com/>

**Link de la Build (Windows y Web):** [**https://yamigomez14.itch.io/stranger-maze-v1**](https://yamigomez14.itch.io/stranger-maze-v1)

**Link del proyecto Git Hub:** [**https://github.com/YamiGomez14/Stranger-Maze-Final-Version-1**](https://github.com/YamiGomez14/Stranger-Maze-Final-Version-1)