



Universidad Nacional Autónoma de México



Facultad de Estudios Superiores Aragón

Proyecto 1. RPG

Materia: Programación de videojuegos 1

Grupo: 2011 (2023 - II)

Profesor: M.I. José Francisco Salgado Rodríguez

Integrantes:

Espinosa Castro Heili Yamilit

Jimenez Almazan Alejandro

Ortiz Tellez Itzel

Fecha: 23 de abril de 2023

Documentación

Se creó un videojuego 2D RPG en el cuál el jugador tiene que explorar 5 escenarios donde encontrará NPCs que le dirán lo que tiene que hacer para completar el juego. En el juego hay objetos coleccionables a lo largo de todos los escenarios contando con cerezas como comida, monedas y gemas que van al inventario del jugador, sin embargo, de acuerdo a la historia contada por los NPCs, el jugador tiene que coleccionar un total de 10 bloques de oro y llevarlos a la casa del zorro para que el juego sea completado.

Además el jugador cuenta con una barra de vida y una de mana que le ayudarán a poder completar el juego, ya que en los escenarios hay enemigos que dificultan el camino para el usuario, para derrotarlos este podrá disparar a distancia o atacarlos cuerpo a cuerpo.

Código implementado:

Para programar el movimiento del player:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MovPlayer : MonoBehaviour
{
    private Vector2 dirMov;
    public float velMov;
    public Rigidbody2D rb;
    public Animator anim;

    private bool PlayerMoviendose = false;
    private float ultimoMovX, ultimoMovY;

    public static int dirAtaque = 0; //1-Front, 2-Back, 3-Left, 4-Right

    void FixedUpdate()
    {
        Movimiento();
        if (CCC.atacando == false && CAD.disparando == false){
            AnimacionesPlayer();
        }
    }

    private void Movimiento(){
        float movX = Input.GetAxisRaw("Horizontal");
        float movY = Input.GetAxisRaw("Vertical");
        dirMov = new Vector2(movX,movY).normalized;
        rb.velocity = new Vector2(dirMov.x * velMov, dirMov.y * velMov);

        if (movX == -1) {
            dirAtaque = 3;
        }
        if (movX == 1) {
```

```

        dirAtaque = 4;
    }
    if (movY == -1) {
        dirAtaque = 1;
    }
    if (movY == 1) {
        dirAtaque = 2;
    }

    if(movX == 0 && movY == 0) {
        PlayerMoviendose = false;
    } else{
        PlayerMoviendose = true;
        ultimoMovX = movX;
        ultimoMovY = movY;
    }
    ActualizaCapa();
}

private void AnimacionesPlayer(){
    anim.SetFloat("movX", ultimoMovX);
    anim.SetFloat("movY", ultimoMovY);
}

private void ActualizaCapa(){
    if(CCC.atacando == false && CAD.disparando == false){
        if(PlayerMoviendose){
            activaCapa("Caminar");
        } else{
            activaCapa("Idle");
        }
    } else {
        activaCapa("Ataque");
    }
}

private void activaCapa(string nombre){
    for(int i=0; i < anim.layerCount; i++){
        anim.SetLayerWeight(i,0); // Ambos layers con peso a 0
    }
    anim.SetLayerWeight(anim.GetLayerIndex(nombre),1);
}
}

```

Movimiento de cámara:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MovCamara : MonoBehaviour
{
    public Camera camara;

    protected void OnTriggerEnter2D(Collider2D obj)
    {
        //Portales escenario 1
        if(obj.gameObject.tag == "portal1"){ // Escenario 1 a 2
            Vector3 posicionCamara = new Vector3(22,0,-10);
            camara.transform.position = posicionCamara;

            Vector3 posicionPlayer = new Vector3(14.43f,2.54f,0);
            this.transform.position = posicionPlayer;
        }
    }
}

```

```

}

if(obj.gameObject.tag == "portal2"){ //De Escenario 1 a 3
    Vector3 posicionCamara = new Vector3(44,0,-10);
    camara.transform.position = posicionCamara;

    Vector3 posicionPlayer = new Vector3(37.15f, -1.31f, 0);
    this.transform.position = posicionPlayer;
}

if(obj.gameObject.tag == "portal3"){ //Escenario 1 a Escenario 3 (bosques perdidos)
    Vector3 posicionCamara = new Vector3(44,0, -10);
    camara.transform.position = posicionCamara;
    Vector3 posicionPlayer = new Vector3(44, 3, 0);
    this.transform.position = posicionPlayer;
}
//Portales escenario 2
if(obj.gameObject.tag == "portal4"){ // Escenario 2 a inicio
    Vector3 posicionCamara = new Vector3(0,0, -10);
    camara.transform.position = posicionCamara;
    Vector3 posicionPlayer = new Vector3(7.6f, 2.5f, 0);
    this.transform.position = posicionPlayer;
}

if(obj.gameObject.tag == "portal5"){ //Escenario 2 a Escenario 3 bosques perdidos)
    Vector3 posicionCamara = new Vector3(44,0, -10);
    camara.transform.position = posicionCamara;
    Vector3 posicionPlayer = new Vector3(37, -1.4f, 0);
    this.transform.position = posicionPlayer;
}

//Portales escenario 3
if(obj.gameObject.tag == "portal6"){ //Escenario 3 a 2
    Vector3 posicionCamara = new Vector3(22,0, -10);
    camara.transform.position = posicionCamara;
    Vector3 posicionPlayer = new Vector3(29.5f, -1.5f, 0);
    this.transform.position = posicionPlayer;
}

if(obj.gameObject.tag == "portal7"){ //de Dungeon a inicio
    Vector3 posicionCamara = new Vector3(69,0, -10);
    camara.transform.position = posicionCamara;
    Vector3 posicionPlayer = new Vector3(64.5f,-1.2f,0);
    this.transform.position = posicionPlayer;
}

//Portales escenario 4
if(obj.gameObject.tag == "portal8"){ //de Dungeon a inicio
    Vector3 posicionCamara = new Vector3(44,0, -10);
    camara.transform.position = posicionCamara;
    Vector3 posicionPlayer = new Vector3(51.50f, -1.83f, 0);
    this.transform.position = posicionPlayer;
}

if(obj.gameObject.tag == "portal10"){ //de (bosques perdidos)4 a 5
    Vector3 posicionCamara = new Vector3(94,0, -10);
    camara.transform.position = posicionCamara;
    Vector3 posicionPlayer = new Vector3(92.56f, 3.61f, 0);
    this.transform.position = posicionPlayer;
}

//Portales escenario 5
if(obj.gameObject.tag == "portal9"){ //de final Escenario 4
    Vector3 posicionCamara = new Vector3(69,0, -10);

```

```

        camara.transform.position = posicionCamara;
        Vector3 posicionPlayer = new Vector3(73.51f,-1.31f, 0);
        this.transform.position = posicionPlayer;

    }

}

```

Control de vidas del player:

```

using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class VidasPlayer : MonoBehaviour
{
    public Image vidaPlayer;
    private float anchoVidasPlayer;
    public static int vida;
    private bool haMuerto;
    public GameObject gameOver;
    private const int vidasINI = 5;
    public static int puedePerderVida = 1;
    public AudioSource reproductor;
    public AudioClip sndGameOver, sndDaño;

    void Start()
    {
        anchoVidasPlayer = vidaPlayer.GetComponent<RectTransform>().sizeDelta.x;
        haMuerto = false;
        vida = vidasINI;
        gameOver.SetActive(false);
    }

    public void TomarDaño(int daño)
    {
        if (vida > 0 && puedePerderVida == 1){
            puedePerderVida = 0;
            vida -= daño;
            DibujaVida(vida);
            reproductor.clip = sndDaño;
            reproductor.Play();
        }
        if (vida <= 0 && !haMuerto){
            haMuerto = true;
            StartCoroutine(EjecutaMuerte());
        }
    }

    public void DibujaVida(int vida){
        if(vida <= vidasINI){
            RectTransform transformalmagen = vidaPlayer.GetComponent<RectTransform>();
            transformalmagen.sizeDelta = new Vector2(anchoVidasPlayer * (float)vida
/(float)vidasINI,transformalmagen.sizeDelta.y);
        }
    }

    IEnumerator EjecutaMuerte()
    {
        yield return new WaitForSeconds(1.2f);
    }
}

```

```

        gameOver.SetActive(true);
        reproductor = GetComponent();
        reproductor.clip = sndGameOver;
        reproductor.Play();
    }
}

```

Coleccionables del player:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class ColeccionablesPlayer : MonoBehaviour
{
    private GameObject player;
    public static string objetoAColeccionar;
    private Inventario inventario; //Referencia a instancia a la clase inventario
    public AudioSource reproductor;
    public AudioClip sndMoneda, sndVida, sndMana, sndGema, sndComida, sndOro, sndGemaVerde,
    sndGemaAzul;

    public TextMeshProUGUI txtPuntos;

    public static int puntos = 0;

    void Start(){
        player = GameObject.Find("player");
        objetoAColeccionar = "";
        inventario = FindObjectOfType<Inventario>();
        EscribePuntos();
        reproductor = GetComponent<AudioSource>();
    }

    protected void OnTriggerEnter2D(Collider2D obj){
        if(obj.tag == "vida"){
            //Debug.Log("Aumenta vida");
            VidasPlayer.vida++;
            player.GetComponent<VidasPlayer>().DibujaVida(VidasPlayer.vida);
            Destroy(obj.gameObject);
            reproductor.clip = sndVida;
            reproductor.Play();
        }
        if(obj.tag == "mana"){
            //AumentarMana
            Destroy(obj.gameObject);
            reproductor.clip = sndMana;
            reproductor.Play();
        }
        if(obj.tag == "moneda"){
            AplicaCambios(obj);
            reproductor.clip = sndMoneda;
            reproductor.Play();
        }
        if(obj.tag == "gema"){
            AplicaCambios(obj);
            reproductor.clip = sndGema;
            reproductor.Play();
        }
        if(obj.tag == "comida"){
            AplicaCambios(obj);
        }
    }
}

```

```

        reproductor.clip = sndComida;
        reproductor.Play();
    }
    if(obj.tag == "gemaVerde"){
        AplicaCambios(obj);
        reproductor.clip = sndGemaVerde;
        reproductor.Play();
    }
    if(obj.tag == "gemaAzul"){
        AplicaCambios(obj);
        reproductor.clip = sndGemaAzul;
        reproductor.Play();
    }
    if(obj.tag == "oro"){
        Destroy(obj.gameObject);
        puntos++;
        EscribePuntos();
        reproductor.clip = sndOro;
        reproductor.Play();
    }
}

private void EscribePuntos(){
    txtPuntos.text = puntos.ToString();
}

private void AplicaCambios(Collider2D obj){
    objetoAColeccionar = obj.tag;
    inventario.EscribeArreglo();
    Destroy(obj.gameObject);
}
}

```

Inventario:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Inventario : MonoBehaviour
{
    private bool muestralInventario;
    public GameObject golInventario; //mostrar y ocultar
    [SerializeField] private string[] valoresInventario; //Arreglo de Strings (valores en el inventario) //"
    -Sin elemento, string elemento
    private int numMonedas, numGemas, numComida, numGemasVerdes, numGemasAzules,
    numOro; //aumentar cantidad al colisionar
    Button boton; //grabar la información
    public Sprite moneda, gema, comida, gemaVerde, gemaAzul, oro, contenedor;

    void Start(){
        muestralInventario = false;
        BorraArreglo();
        numMonedas = 0;
        numGemas = 0;
        numComida = 0;
        ColeccionablesPlayer.puntos = 0;
        numGemasVerdes = 0;
        numGemasAzules = 0;
    }
}

```

```

public void StatusInventario(){
    if(muestralInventario){
        muestralInventario = false;
        golInventario.SetActive(false);
        Time.timeScale = 1; //El juego se reactiva
    }else{
        muestralInventario = true;
        golInventario.SetActive(true);
        Time.timeScale = 0; //tiempo se pone en 0 (el juego se detiene)
    }
}

public void EscribeArreglo() {
    if(VerificaEnArreglo() == -1) { //No está en el inventario
        for(int i=0; i< valoresInventario.Length; i++){
            if (valoresInventario[i] == ""){ //Si encuentra una cadena vacía
                valoresInventario[i] = ColeccionablesPlayer.objetoAColeccionar;
                DibujaElementos(i); //Dibuja el elemento
                break;
            }
        }
    } else { //Está en el inventario
        DibujaElementos(VerificaEnArreglo());
    }
}

private int VerificaEnArreglo(){
    int pos = -1;
    for (int i = 0; i < valoresInventario.Length; i++){
        if(valoresInventario[i] == ColeccionablesPlayer.objetoAColeccionar){ //Determina la
posición del objeto con el que se colisionó
            pos = i;
            break;
        }
    }
    return pos; //Regresa a posicion inicial
}

public void DibujaElementos(int pos){
    StatusInventario(); //Mostrar el inventario
    boton = GameObject.Find("elemento (" + pos + ")").GetComponent<Button>(); //Encuentra el
elemento llamado elemento en su posición
    switch (ColeccionablesPlayer.objetoAColeccionar){
        case "moneda":
            contenedor = moneda;
            numMonedas++;
            boton.GetComponentInChildren<Text>().text = "x" +
numMonedas.ToString(); //Muestra num de objetos coleccionados
            break;
        case "gema":
            contenedor = gema;
            numGemas++;
            boton.GetComponentInChildren<Text>().text = "x" +
numGemas.ToString(); //Muestra num de objetos coleccionados
            break;
        case "gemaVerde":
            contenedor = gemaVerde;
            numGemasVerdes++;
            boton.GetComponentInChildren<Text>().text = "x" +
numGemasVerdes.ToString(); //Muestra num de objetos coleccionados
            break;
        case "gemaAzul":

```



```

                                contenedor = gemaAzul;
                                numGemasAzules++;
                                boton.GetComponentInChildren<Text>().text = "x" +
numGemasVerdes.ToString();      //Muestra num de objetos coleccionados
                                break;
                                case "comida":
                                    contenedor = comida;
                                    numComida++;
                                    boton.GetComponentInChildren<Text>().text = "x" +
numComida.ToString();          //Muestra num de objetos coleccionados
                                    break;
                                }
                                boton.GetComponent<Image>().sprite = contenedor;
                            }

                            private void BorraArreglo(){
                                for(int i = 0; i< valoresInventario.Length; i++){//desde 0 a la cantidad del array
                                    valoresInventario[i] = "";
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Disparo a distancia:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CAD : MonoBehaviour
{
    [SerializeField] private GameObject proyectil;
    public float tiempoEntreAtaques;
    public float tiempoSigAtaque;
    public Transform puntoEmision;
    private Animator anim;
    public static int dirDisparo = 0;
    public static bool disparando = false;
    public AudioSource reproductor;
    public AudioClip sndDispara;

    void Start()
    {
        anim = GetComponent<Animator>();
        reproductor = GetComponent<AudioSource>();
    }

    void Update()
    {
        if(tiempoSigAtaque < 0.05f && tiempoEntreAtaques > 0){
            disparando = false;
        }
        if(tiempoSigAtaque > 0) {
            tiempoSigAtaque -= Time.deltaTime;
        }
        if (Input.GetButtonDown("Fire2") && tiempoSigAtaque <= 0){
            disparando = true;
            activaCapa("Ataque");
            Dispara();
            tiempoSigAtaque = tiempoEntreAtaques;
        }
    }
}

```

```

private void Dispara() {
    if (MovPlayer.dirAtaque == 1){
        anim.SetTrigger("disparoFront");
    } else if (MovPlayer.dirAtaque == 2){
        anim.SetTrigger("disparoBack");
    } else if (MovPlayer.dirAtaque == 3){
        anim.SetTrigger("disparolzq");
    } else if (MovPlayer.dirAtaque == 4){
        anim.SetTrigger("disparoDer");
    }
}

private void EmiteProyectil() {
    dirDisparo = MovPlayer.dirAtaque;
    Instantiate(proyectil, puntoEmision.position, transform.rotation);
    reproductor.clip = sndDispara;
    reproductor.Play();
}

private void activaCapa(string nombre){
    for (int i=0; i < anim.layerCount; i++){
        anim.SetLayerWeight(i,0); // Ambos layers con peso a 0
    }
    anim.SetLayerWeight(anim.GetLayerIndex(nombre),1);
}
}

```

Dispara proyectil:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DisparaProyectil : MonoBehaviour
{
    [SerializeField] private float velocidad = 8.0f;

    void FixedUpdate(){
        if (CAD.dirDisparo == 1){
            transform.position += new Vector3(0,-1,0) * Time.deltaTime * velocidad;
        }
        if (CAD.dirDisparo == 2){
            transform.position += new Vector3(0,1,0) * Time.deltaTime * velocidad;
        }
        if (CAD.dirDisparo == 3){
            transform.position += new Vector3(-1,0,0) * Time.deltaTime * velocidad;
        }
        if (CAD.dirDisparo == 4){
            transform.position += new Vector3(1,0,0) * Time.deltaTime * velocidad;
        }
    }

    private void OnTriggerEnter2D(Collider2D collision){
        if(collision.gameObject.tag == "limites"){
            Destroy(this.gameObject);
        }
        if(collision.gameObject.tag == "enemigo"){
            collision.transform.GetComponent<Enemigo>().TomarDaño(1);
            Destroy(this.gameObject);
        }
    }
}

```

```
}
```

Ataque cuerpo a cuerpo:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CCC : MonoBehaviour
{
    public Transform controladorGolpe;
    public float radioGolpe;
    public int dañoGolpe;
    public float tiempoEntreAtaques;
    public float tiempoSigAtaque;
    private Animator anim;
    public AudioSource reproductor;
    public AudioClip sndGolpe;

    public static bool atacando;

    void Start()
    {
        anim = GetComponent<Animator>();
        reproductor = GetComponent<AudioSource>();
    }

    void Update()
    {
        if(tiempoSigAtaque < 0.05f && tiempoEntreAtaques > 0){
            atacando = false;
        }
        if(tiempoSigAtaque > 0) {
            tiempoSigAtaque -= Time.deltaTime;
        }
        if (Input.GetButtonDown("Fire1") && tiempoSigAtaque <= 0){
            atacando = true;
            activaCapa("Ataque");
            Golpe();
            tiempoSigAtaque = tiempoEntreAtaques;
        }
    }

    private void Golpe() {
        if (MovPlayer.dirAtaque == 1){
            anim.SetTrigger("ataqueFront");
        } else if (MovPlayer.dirAtaque == 2){
            anim.SetTrigger("ataqueBack");
        } else if (MovPlayer.dirAtaque == 3){
            anim.SetTrigger("ataqueIzq");
        } else if (MovPlayer.dirAtaque == 4){
            anim.SetTrigger("ataqueDer");
        }
    }

    private void VerificaGolpe(){
        Collider2D[] objs = Physics2D.OverlapCircleAll(controladorGolpe.position, radioGolpe);
        foreach (Collider2D colisionador in objs){
            if(colisionador.CompareTag("enemigo")){
                colisionador.transform.GetComponent<Enemigo>().TomarDaño(dañoGolpe);
                reproductor.clip = sndGolpe;
            }
        }
    }
}
```

```

        reproductor.Play();
    }
}

private void OnDrawGizmos(){
    Gizmos.color = Color.blue;
    Gizmos.DrawWireSphere(controladorGolpe.position, radioGolpe);
}

private void activaCapa(string nombre){
    for(int i = 0; i < anim.layerCount; i++){
        anim.SetLayerWeight(i, 0);
    }
    anim.SetLayerWeight(anim.GetLayerIndex(nombre),1);
}
}

```

Enemigo:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.AI;

public class Enemigo : MonoBehaviour
{
    public static int vidasEnemigo = 1;
    private float frecAtaque = 2.5f, tiempoSiAtaque = 0, iniciaConteo;

    public Transform personaje;
    private NavMeshAgent agente;
    public Transform[] puntosRuta;
    private int indiceRuta = 0;
    private bool playerEnRango = false;
    [SerializeField] private float distanciaDeteccionPlayer;
    private SpriteRenderer spriteEnemigo;
    private Transform mirarHacia;

    private void Awake(){
        agente = GetComponent<NavMeshAgent>();
        spriteEnemigo = GetComponent<SpriteRenderer>();
    }

    void Start()
    {
        vidasEnemigo = 1;
        agente.updateRotation = false;
        agente.updateUpAxis = false;
    }

    void Update(){
        this.transform.position = new Vector3(transform.position.x, transform.position.y, 0);
        float distancia = Vector3.Distance(personaje.position, this.transform.position);

        if(this.transform.position == puntosRuta[indiceRuta].position){
            if(indiceRuta < puntosRuta.Length - 1){
                indiceRuta++;
            }else if(indiceRuta == puntosRuta.Length - 1){
                indiceRuta = 0;
            }
        }

        if(distancia < distanciaDeteccionPlayer){

```

```

        playerEnRango = true;
    }else{
        playerEnRango = false;
    }

    if (tiempoSiAtaque > 0){
        tiempoSiAtaque = frecAtaque + iniciaConteo - Time.time;
    } else {
        tiempoSiAtaque = 0;
        VidasPlayer.puedePerderVida = 1;
        SigueAIPlayer(playerEnRango);
        RotaEnemigo();
    }
}

private void SigueAIPlayer(bool playerEnRango){
    if(playerEnRango){
        agente.SetDestination(personaje.position);
        mirarHacia = personaje;
    } else {
        agente.SetDestination(puntosRuta[indiceRuta].position);
        mirarHacia = puntosRuta[indiceRuta];
    }
}

private void RotaEnemigo(){
    if(this.transform.position.x > mirarHacia.position.x){
        spriteEnemigo.flipX = true;
        Debug.Log("FlipX");
    }else {
        spriteEnemigo.flipX = false;
        Debug.Log("Sin FlipX");
    }
}

private void OnTriggerEnter2D(Collider2D obj){
    if(obj.tag == "Player"){
        tiempoSiAtaque = frecAtaque;
        iniciaConteo = Time.time;
        obj.transform.GetComponentInChildren<VidasPlayer>().TomarDaño(1);
    }
}

public void TomarDaño(int daño){
    vidasEnemigo -= daño;
    if (vidasEnemigo <= 0 ){
        Destroy(gameObject);
    }
}
}

```

NPC:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class NPC : MonoBehaviour
{
    public GameObject txtDialogo;
    public int numVisitas;

    public Sprite txt1, txt2;
}

```

```

void Start()
{
    txtDialogo.SetActive(false);
    numVisitas = 0;
}

private void OnTriggerEnter2D(Collider2D obj)
{
    if(obj.tag == "Player") {
        txtDialogo.SetActive(true);
        EscribeDialogo();
        numVisitas++;
    }
}

private void EscribeDialogo(){
    switch(numVisitas){
        case 0:
            txtDialogo.GetComponent<SpriteRenderer>().sprite = txt1;
            break;
    }
}
}

```

Diálogos NPC:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DialogoNPC : MonoBehaviour
{
    public GameObject txtDialogo;

    private void OnMouseDown()
    {
        this.gameObject.SetActive(false);
    }
}

```

Para hacer el menú inicial:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class menuInicial : MonoBehaviour
{
    public void Jugar(){
        SceneManager.LoadScene("SampleScene");
    }
}

```

```

        public void Creditos(){
            SceneManager.LoadScene("Creditos");
        }

        public void Salir(){
            Debug.Log("Salir");
            Application.Quit();
        }
    }

```

Para mostrar los créditos:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class creditos : MonoBehaviour
{
    public void Menu(){
        SceneManager.LoadScene("menuInicial");
    }
}

```

Cuando se completa correctamente el juego:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class Victoria : MonoBehaviour
{
    // Nombre de la escena a la que se desea cambiar
    public string nombreDeLaEscena;

    private void OnTriggerEnter2D(Collider2D other)
    {
        // Si el objeto que choca con el NPC tiene la etiqueta "Player"
        if (other.CompareTag("Player") && ColeccionablesPlayer.puntos == 10)
        {
            // Carga la escena con el nombre especificado
            SceneManager.LoadScene("final");
        }
    }
}

```

Introducción del juego:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

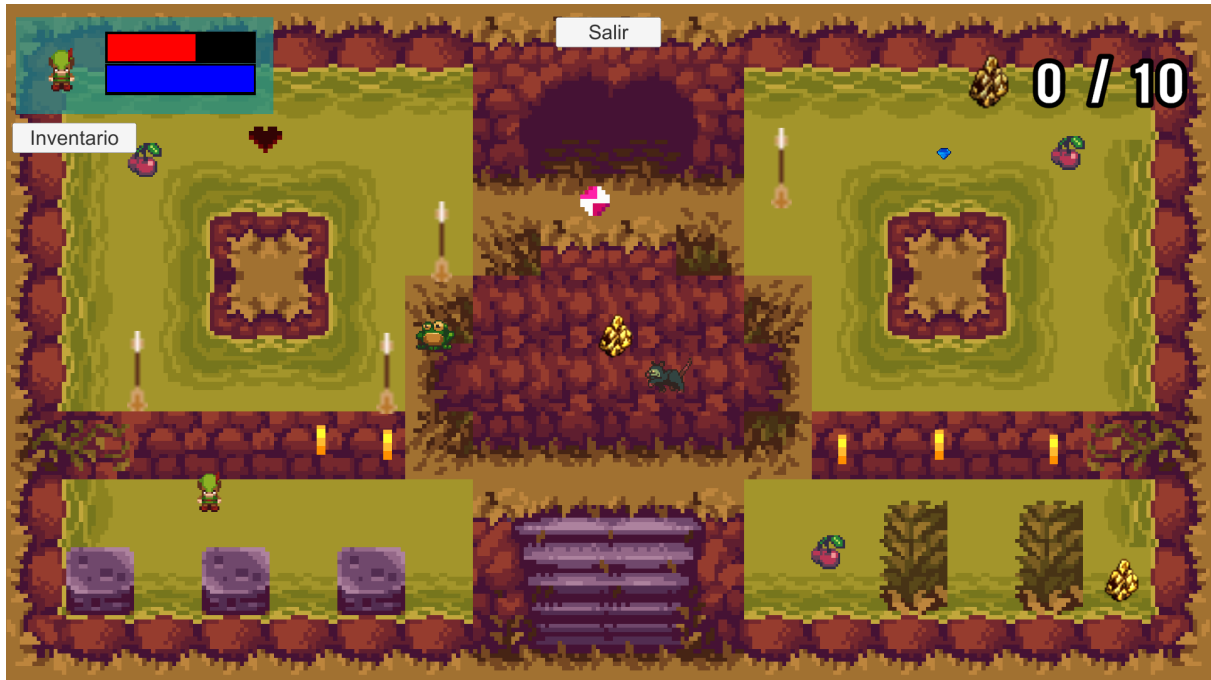
```

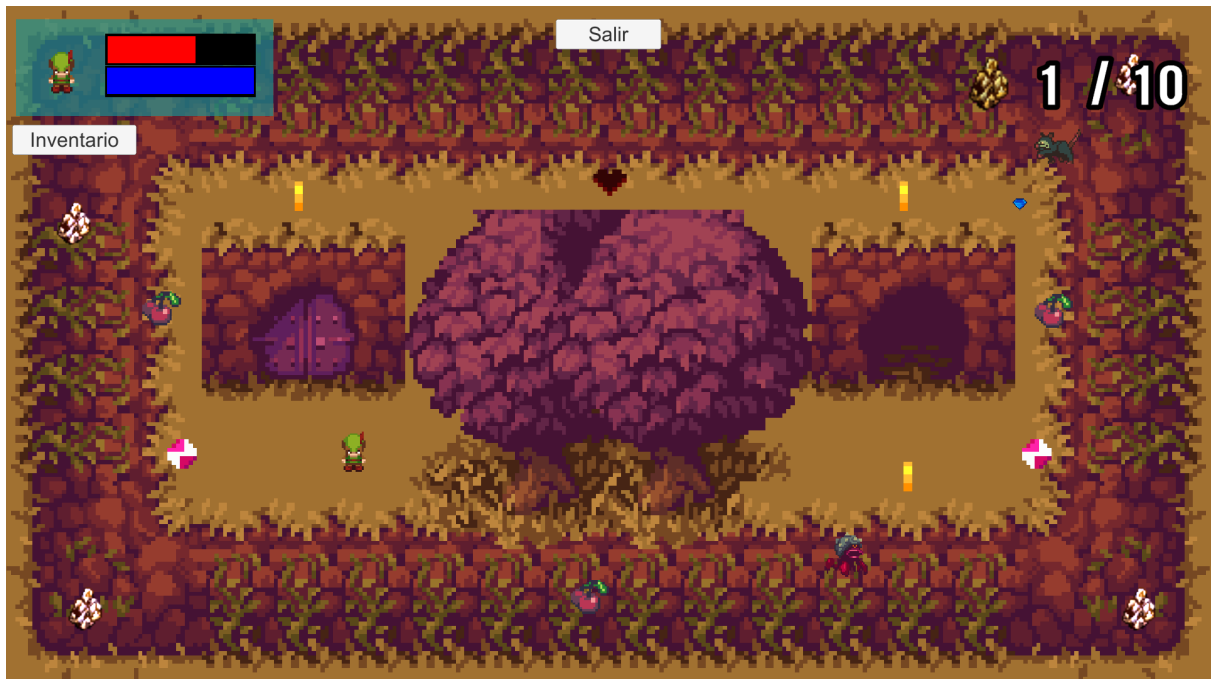
```
using UnityEngine.SceneManagement;

public class Introduccion : MonoBehaviour
{
    public void Continuar1(){
        SceneManager.LoadScene("introduccion2");
    }
    public void Continuar2(){
        SceneManager.LoadScene("SampleScene");
    }
}
```

Escenarios del juego:







Menú de inicio:

ARCHEST FOREST

JUGAR

CREDITOS

SALIR

*PULSA CLICK IZQUIERDO
PARA GOLPEAR*

*PULSA CLICK DERECHO
PARA DISPARAR*



Pantalla Game Over:

Game Over

Menú

Pantalla de Fin del Juego:



!FELICIDADES!



!HAZ CONCLUIDO EL JUEGO!



MENU

