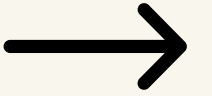

INFORME

PROGRAMACIÓN 3 - VIDEOJUEGO

Universidad de Belgrano
Diseño y Animación Digital
Profesora: Patricia Elena Daitter
Alumna: Maia Pantelides

2025



IDEA Y ASSETS

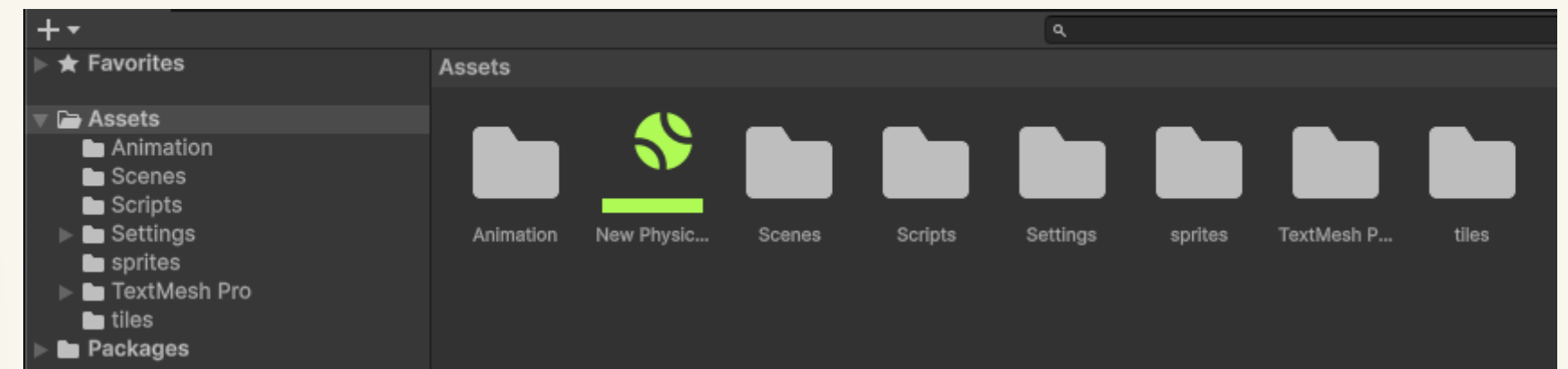
Para mi juego me inspiré en juegos de plataforma vertical en donde el personaje principal **debe escapar de agua que emerge del piso mientras recoge comidas.**

Decidí usar los assets de **Kirby** (un personaje existente) y para las plataformas importé un **tilemap**.

CONTROLES:

Flechas izquierda y derecha: mover a Kirby.
Barra espaciadora: saltar.

Organicé así mis carpetas:



Animation: todas las animaciones que creé

Scenes: el menú y el juego

Scripts: todos los códigos

sprites: todos los diseños del juego

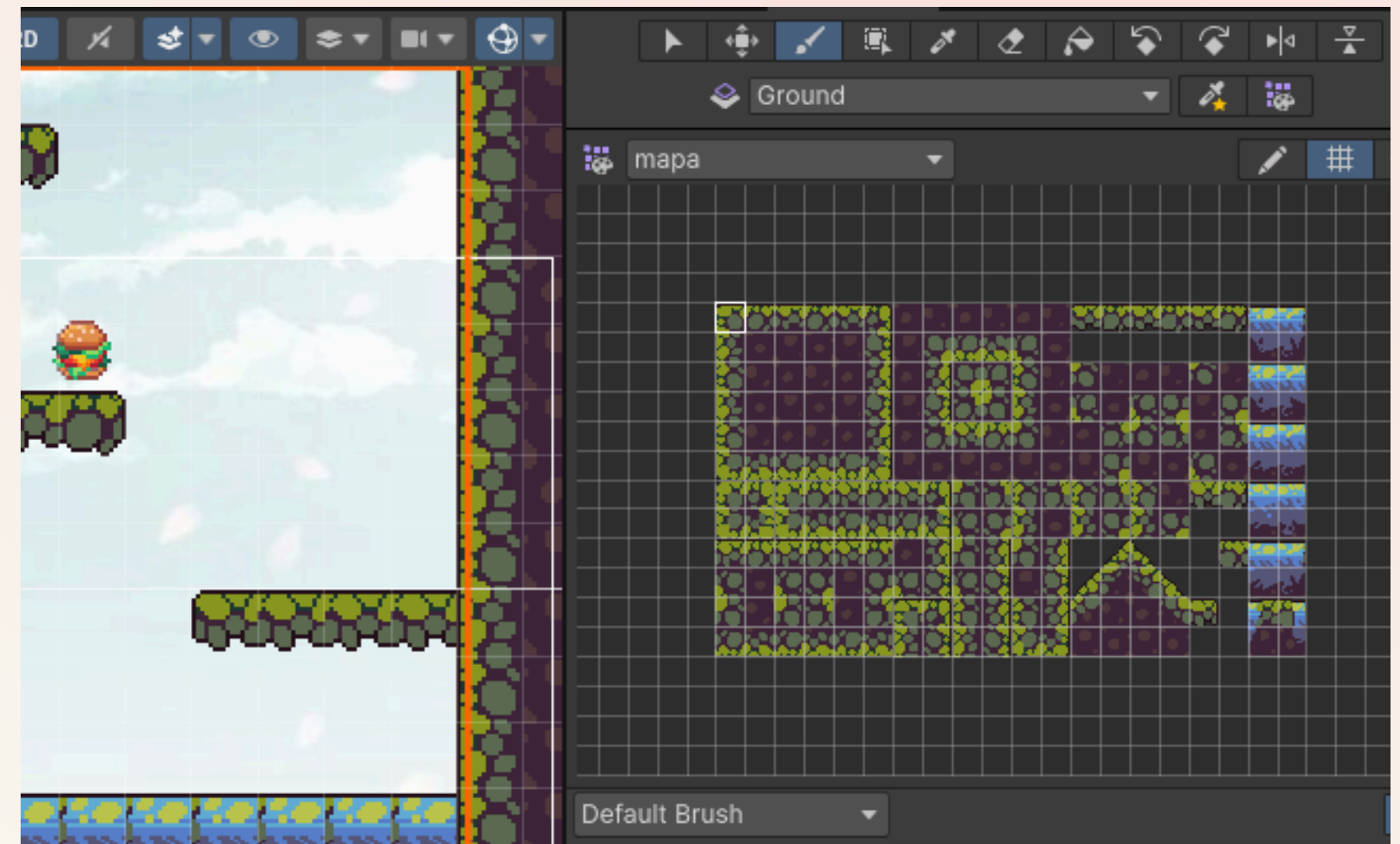
tiles: donde importé el Tileset

SOFTWARE Y LIENZO

En mi videojuego decidí usar **Unity** ya que es uno de los mejores motores actuales. Al empezar creé la carpeta “**sprites**” dentro de Assets, en esta importé todos los diseños del juego (el agua que sube, el cielo, la comida, Kirby, el punto de llegada y el Tilemap).

Usé el tileset llamado **Free Swamp 2D tileset pixel art** ya que tenía la estética que quería conseguir. Recorté los tiles dentro del sprite editor, en la jerarquía creé un Tilemap y luego una TilePalette. Arrastré mis tiles ya recortados a la paleta y con eso ya pude comenzar a “pintar” sobre mi lienzo las plataformas.

Con estas herramientas fui pintando las plataformas



PLAYER Y MOVIMIENTO

Primero importé los sprites de Kirby a la carpeta de Sprites, en la jerarquía creé un objeto2D cuadrado y enlacé la pose inicial de Kirby con el objeto. Le agregué **Rigidbody2D** al Player para que tenga gravedad (la cual cambié a 2).

Para que el Player colisione con las plataformas les agregué **Tilemap Collider2D** a las plataformas y BoxCollider2D al Player.

Dentro de mi carpeta de Scripts creé un **C# script** el cual arrastré dentro del Player.

En principio mi código se veía así, pero el personaje podía saltar sin haber vuelto a tocar el piso así que lo modifiqué.

```
using UnityEngine;
public class Player : MonoBehaviour {
    public float moveSpeed = 105f;
    public float jumpForce = 110f;
    private Rigidbody2D rb;

    void Start() {
        rb = GetComponent<Rigidbody2D>();
    }

    void Update()
    {
        float moveInput = Input.GetAxis("Horizontal");
        rb.linearVelocity = new Vector2(moveInput * moveSpeed,
        rb.linearVelocity.y);

        if (Input.GetKeyDown(KeyCode.Space))
        {
            rb.linearVelocity = new Vector2(rb.linearVelocity.x, jumpForce);
        }
    }
}
```

PLAYER Y MOVIMIENTO

Agregué los siguientes public y private y modifiqué que solo pueda volver a saltar si está tocando el piso.

```
public Transform groundCheck;  
public float groundCheckRadius = 0.2f;  
public LayerMask groundLayer;  
private bool isGrounded;
```

```
void Update()  
{  
    if (Input.GetKeyDown(KeyCode.Space) && isGrounded)  
    {  
        rb.linearVelocity = new Vector2(rb.linearVelocity.x, jumpForce);  
    }  
}
```

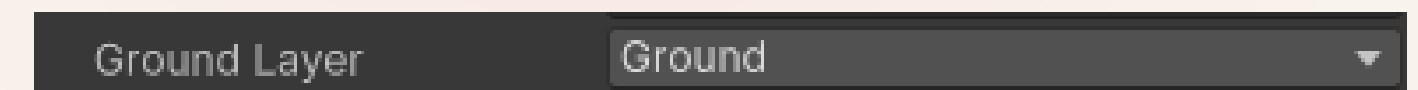
```
private void FixedUpdate()  
{  
    isGrounded = Physics2D.OverlapCircle(groundCheck.position,  
groundCheckRadius, groundLayer);  
}
```

Luego en la jerarquía, dentro del Player, creé un Empty llamado GroundCheck y lo puse a los pies del personaje.

Dentro del Player arrastré el GroundCheck a la casilla correspondiente



Creé una etiqueta llamada Ground y se la asigne al Ground creado anteriormente con el Tilemap

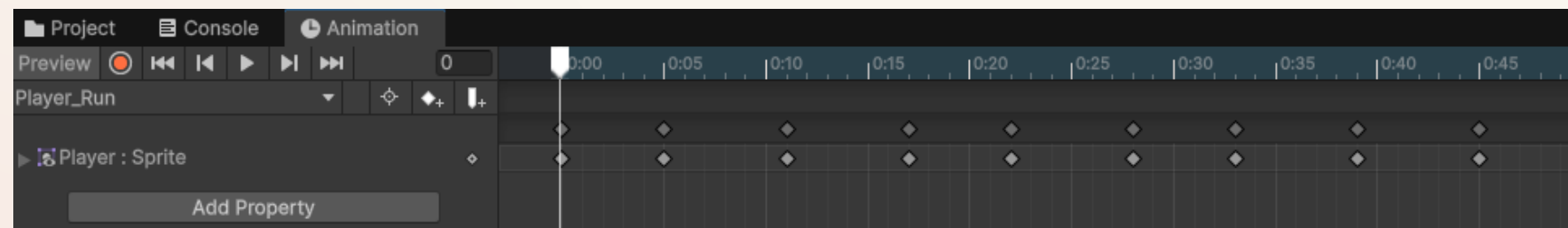


También para que el personaje no rote congelé la rotación en Z dentro del Rigidbody2D del player.

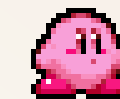
ANIMACIONES

Abrí el Animation y creé una carpeta dentro de Assets llamada Animation.

Empecé creando la animación que aparece cuando el personaje está quieto (Player_Idle), luego hice cuando camina (Player_Run), cuando come (Player_Eat), cuando salta (Player_Jump) y cuando cae (Player_Fall).



Quieto



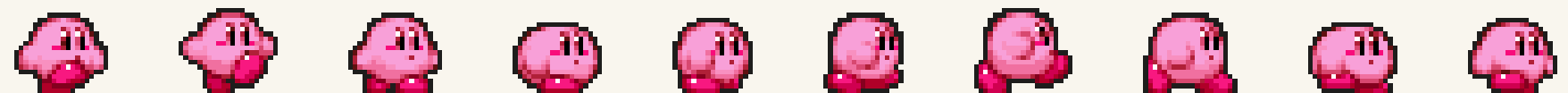
Saltar y caer:



Comer:



Caminar:



ANIMACIONES Y CÁMARA

```
private Animator animator;
void Start()
{
    animator = GetComponent<Animator>();
}
void Update()
{ SetAnimation(moveInput);
  if (moveInput > 0)
    spriteRenderer.flipX = false;
  else if (moveInput < 0)
    spriteRenderer.flipX = true;
}
private void SetAnimation(float moveInput)
{
  if (isGrounded)
  {
    if (moveInput == 0)
      {animator.Play("Player_Idle"); }
    else
      {animator.Play("Player_Run"); }
  }
}
```

→ el Player mira a la derecha

→ el Player mira a la izquierda

reproduce la animación de idle/ run

```
else
{ if(rb.linearVelocityY > 0)
  {animator.Play("Player_Jump");}
  else
    {animator.Play("Player_Fall"); }
}
```

reproduce la animación de salto/ caída

Para hacer el movimiento de cámara me instalé el Cinemachine y creé un CinemachineCamera. Le cambié la posición a -10 y arrastré al Player al Target de la cámara. Para que siga al personaje cambié el Position Control a follow.

Procedural Components

Position Control	Follow
Rotation Control	None
Noise	None

PLATAFORMAS MOVIBLES

Primero empecé arrastrando los tiles que iba a usar al proyecto, creé un Empty object y le puse de nombre MovingPlatform, dentro de este creé otro y le puse Container.

Le agregué Rigidbody2D y lo cambié a Kinematic. También le agregué un BoxCollider2D y le asigné la etiqueta de Ground.

Metí los tiles dentro del contenedor y los acomodé.

Creé otro empty dentro de la plataforma y lo llame Point 1, lo moví a la posición inicial de la plataforma, después repetí el proceso con el Point 2 solo que esta vez puse la posición final.

```
using UnityEngine;

public class NewMonoBehaviourScript : MonoBehaviour
{
    public float speed = 2f;
    public Transform[] points;

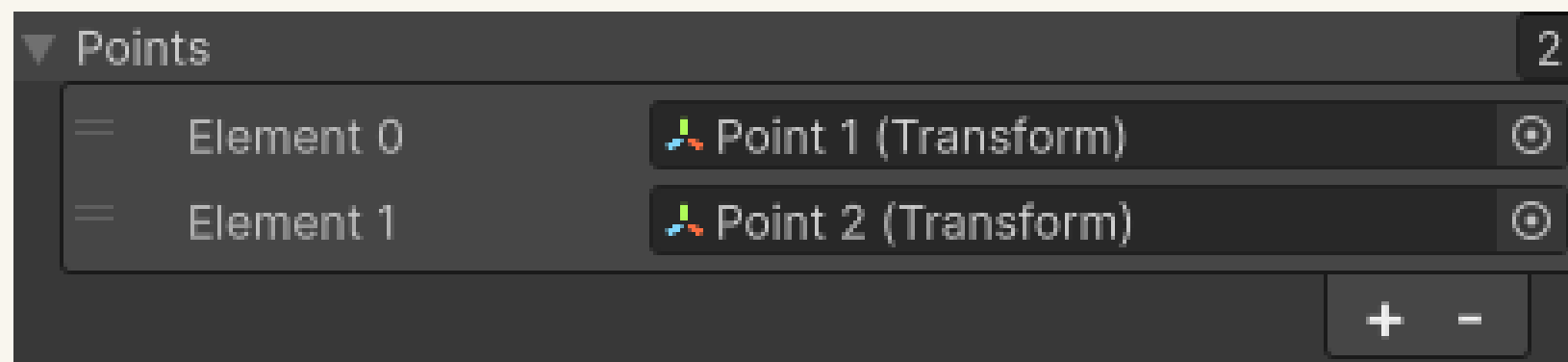
    private int i;
    void Start()
    {
        transform.position = points[0].position;
    }

    void Update()
    {
        if(Vector2.Distance(transform.position, points[i].position) < 0.01f)
        {
            i++;
            if(i == points.Length)
            {
                i = 0;
            }
        }

        transform.position = Vector2.MoveTowards(transform.position, points[i].position, speed * Time.deltaTime);
    }
}
```


PLATAFORMAS MOVIBLES

Al terminar el script lo arrastré al contenedor y dentro arrastré cada punto al lugar correspondiente



También le asigné la etiqueta Player al Player

Agregué estas líneas para que cuando el Player esté encima de la plataforma que la siga (el primer if los vincula y el segundo los desvincula)

```
private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.tag == "Player")
    {
        collision.transform.SetParent(transform);
    }
}

private void OnCollisionExit2D(Collision2D collision)
{
    if (collision.gameObject.tag == "Player")
    {
        collision.transform.SetParent(null);
    }
}
```

DAÑO Y PLATAFORMA INVISIBLE

Dentro de la Grid creé otro piso al que llamé Damage y le asigné una nueva etiqueta llamada Damage. Agregué la siguientes líneas de código.

```
public int health = 100;
private void OnCollisionEnter2D(Collision2D collision)
{
    if(collision.gameObject.tag == "Damage")
    {
        health -=100;
        if(health <= 0)
        {
            Die();
        }
    }
}
```

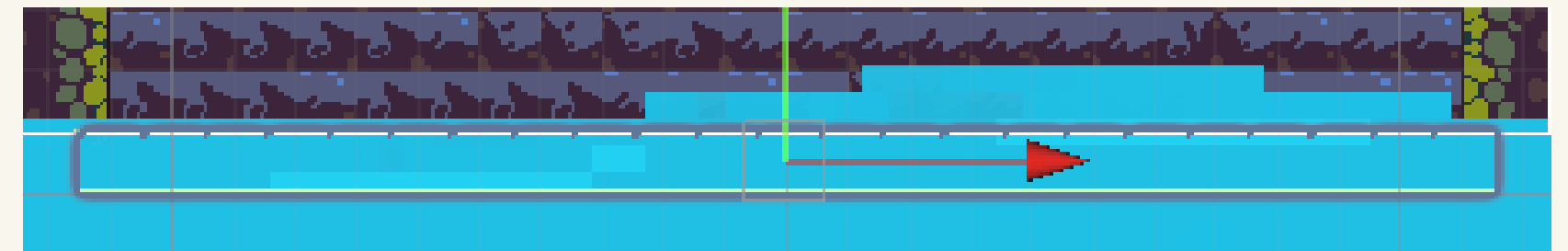
quería hacer que al tocar el agua el personaje muriera inmediatamente, así que hice que cuando el player colisiona con un game object con la etiqueta Damage pierda toda su vida.

```
private void Die()
{
    UnityEngine.SceneManagement.SceneManager.LoadScene
    ("GameScene");
}
```

con este código, al morir, la escena del juego se reinicia

Para crear el agua que sube repetí el código de las plataformas movibles solo que de manera vertical y con una imagen. También le agregué una pequeña animación al agua con el animator (son dos sprites espejados para que parezca que el agua tiene olas)

Para que el personaje muera cuando toca el agua creé otra plataforma movible que llame InvisiblePlatform y le asigné la etiqueta Damage. Esta plataforma se mueve al mismo tiempo que el agua por debajo de esta (no se ve en el juego)



COMIDAS

Empecé arrastrando el sprite de la hamburguesa a la jerarquía y le hice una animación donde le varié la posición de arriba a abajo. Creé un nuevo script donde escribí lo siguiente:

```
private void OnTriggerEnter2D(Collider2D collision)
{
    if(collision.gameObject.tag == "Player")
    {
        Player player = collision.gameObject.GetComponent<Player>();
        player.food += 1;
        coinText.text = player.food.ToString();
        Destroy(gameObject);
    }
}

public int food;      ➡ lo agregué en el script del Player
```

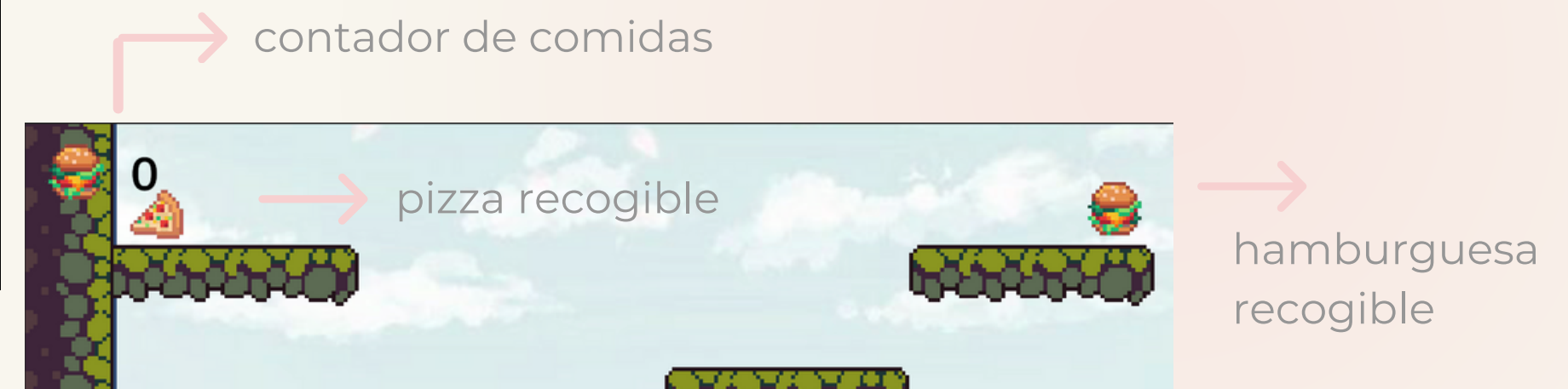
Arrastré el script a mi hamburguesa y le agregué BoxCollider2D con IsTrigger activado. También le asigné la etiqueta food.

Después creé un Canvas y dentro de este hice un UI image. Puse de imagen la hamburguesa y edité el tamaño a mi gusto. Dentro de la imagen creé un Text donde escribí 0. Le agregué esto al código:

```
using TMPro;
private TextMeshProUGUI coinText;

private void Start()
{
    coinText =
    GameObject.FindWithTag("CoinText").GetComponent<TextMeshProUGUI>();
}
```

Por último repetí el proceso con las otras comidas (usando el mismo script y tag pero con diferentes sprites y animaciones)



PLAYER_EAT

Una de las animaciones que quería agregar era la de Player_Eat, la cual se reproduciría al entrar en contacto con la comida.

Primero intenté usar OnCollision y animator.Play("Player_Eat") pero no funcionó, después intenté con un OnTriggerer pero tampoco se reproducía la animación.

Me di cuenta que las animaciones se estaban superponiendo así que para que se reprodujera Player_Eat tenía que pausar las otras. Para hacer esto escribí el siguiente código.

```
private bool isEating = false;
```

```
public void AccionDespuesDeComer()
{
    isEating = false;
}


private void OnTriggerEnter2D(Collider2D other)
{
    if (other.CompareTag("Food"))
    {
        Debug.Log("Comida recogida");
        isEating = true;
        animator.Play("Player_Eat");
        //isEating = false;
        //animator.SetTrigger("Player_Eat");
    }
}
```

También agregué un if (!isEating) en el private void SetAnimation(float moveInput) antes del if (isGrounded) para que las animaciones solo se reproduzcan si el Player no está comiendo

PANTALLA DEL FINAL

Primero diseñé la bandera del final, la metí en mi carpeta de sprites. y la puse en el final del mapa. Le agregué BoxCollider2D, le activé el IsTrigger y le creé un script donde escribí lo siguiente

```
using UnityEngine;
public class Flag : MonoBehaviour
{
    public GameObject winUI;

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if(collision.gameObject.tag == "Player")
        {
            Time.timeScale = 0;
            winUI.SetActive(true);  pausa el juego
        }
    }
}
```

Después dentro del Canvas creé una UI Image llamada WinUI e hice que ocupara toda la pantalla.

Agregué un Text Mesh Pro en el UI Image que dice “Ganaste!” y también agregué un UI Button en el cual escribí volver a jugar

Le creé un script al botón de reinicio en el que escribí esto:

```
using UnityEngine;
public class RestartGame : MonoBehaviour
{
    public void LoadCurrentScene()
    {
        UnityEngine.SceneManagement.SceneManager.LoadScene("GameScene");
        Time.timeScale = 1;
    }
}
```

Por último arrastré el script al componente del botón y le asigné la función RestartGame LoadCurrentScene(). Deshabilité el WinUI y lo arrastré como referencia dentro de la bandera.

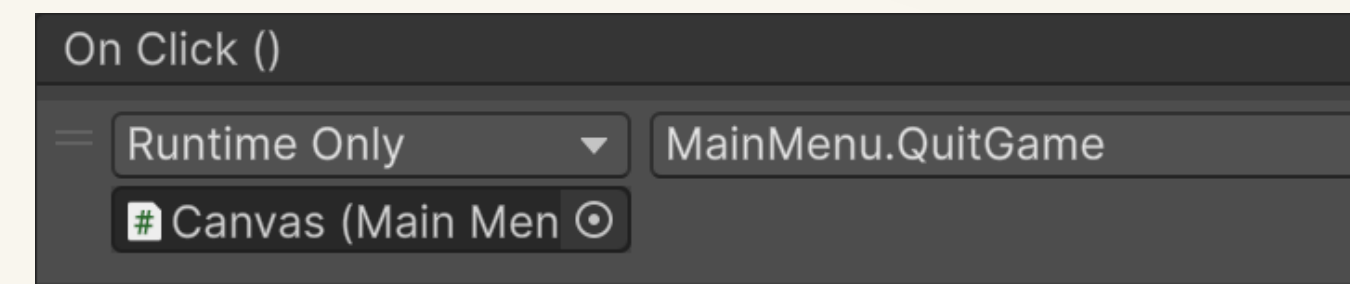
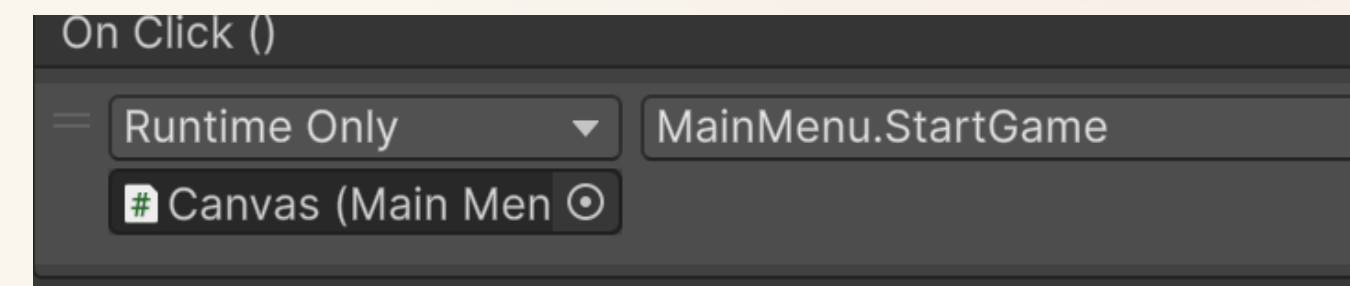
MENÚ INICIAL

Para crear el menú inicial creé una nueva escena y un canvas. Dentro del canvas creé un botón para empezar a jugar y otro para salir del juego.

Para que los botones funcionen creé el siguiente código

```
using UnityEngine;
using UnityEngine.SceneManagement;
public class MainMenu : MonoBehaviour
{
    public void StartGame()
    {
        SceneManager.LoadScene("GameScene");
    }
    public void QuitGame()
    {
        Application.Quit();
    }
}
```

En los componentes del botón de comenzar le asigné la función de MainMenu StartGame() y en los de salir del juego le asigné la función MainMenu QuitGame()



Tuve varios problemas para la imagen del fondo en el menú, ya que yo quería poner un video pero cuando lo exporté para WebGL me apareció un aviso que decía que no aceptaba este formato. Tristemente no pude poner el video de forma que lo aceptara así que quedó sin ningún fondo.

SUBIR A PÁGINA WEB

Primero tuve que cambiar los Build Sttings a WebGL y asegurarme de que aparecieran todas las escenas de mi juego en orden (MainMenu y GameScene). También cambié unas opciones de compresión del Player.

Creé una carpeta donde guardé mi Build para web, lo comprimí y lo subí a Itch.io

Para subirlo elegí que fuera jugable en el ordenador, tipo de archivo HTML y subi el zip. También subí una foto de cover del juego.

Para meterlo dentro de mi página web primero creé una nueva sección llamada juegos, después entré a mi proyecto en itch.io, fui a Distriute, luego a Embed Game y copié el código que me ofreció la página dentro de mi nueva sección.

Title

Juego Final Programación 3

Project URL

https://maia-pantelides.itch.io/juego-final-programacion-3

Short description or tagline

Shown when we link to your project. Avoid duplicating your project's title

Optional

Classification

What are you uploading?

Games — A piece of software you can play

Kind of project

HTML — You have a ZIP or HTML file that will be played in the browser


TIP

You can add additional downloadable files for any of the types above

Release status

Released — Project is complete, but might receive some updates

Pricing



The cover image is used whenever itch.io wants to link to your project from another part of the site. Required (Minimum: 315x250, Recommended: 630x500)

Gameplay video or trailer

Provide a link to YouTube or Vimeo.

eg. https://www.youtube.com/watch?v=5JEaA47sPJQ

Screenshots

Optional but highly recommended. Upload 3 to 5 for best results.

Add screenshots

Embed game

If you have a browser playable game on itch.io then you can generate embed code to host your project on other websites.

Select a file

JuegoWeb.zip

Width

640

Height

380

Color

#333333

Preview

```
<iframe frameborder="0" src="https://itch.io/embed-upload/15561516?color=333333" allowfullscreen="" width="640" height="380"><a href="https://maia-pantelides.itch.io/juego-final-programacion-3">Play Juego Final Programación 3 on
```

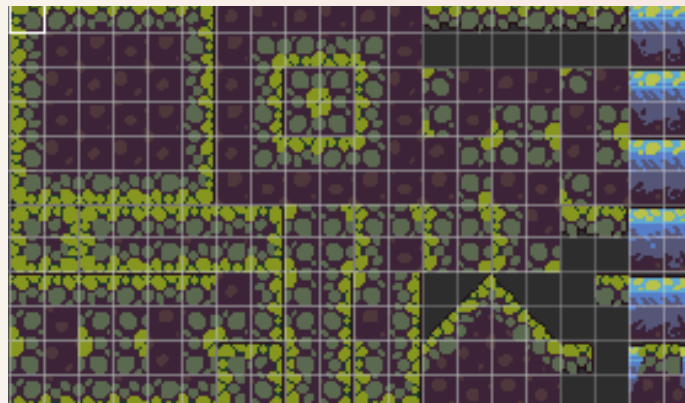
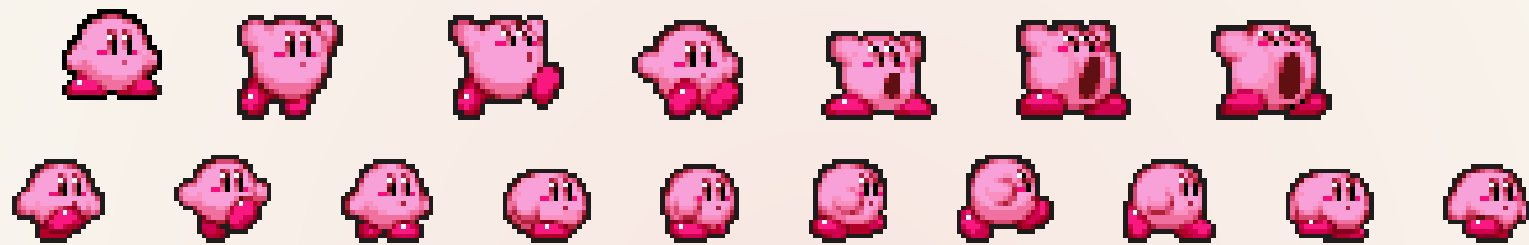
Directions Copy and paste this HTML into your website where you want the game to appear.

Direct link? Use this link to go directly to your game:

https://itch.io/embed-upload/15561516?color=333333

RECURSOS

Sacado de Internet



Hechos por mi



CONCLUSIONES

En un principio estaba asustada porque pensé que iba a ser muchísimo más difícil pero una vez que me puse a hacerlo me divertí mucho. Hay varias cosas que siento que podría mejorar o me gustaría haber hecho diferente pero para ser mi primera vez haciendo un juego con Unity y completamente sola estoy muy orgullosa.

Me gustaría algún día cuando tenga más tiempo poder hacerlo todo yo sin utilizar sprites de Internet. También me gustaría aprender como arreglar el video de la pantalla de inicio para poder agregarlo.

Muchas cosas que se me complicaron las pude arreglar (como el Player_Eat) así que estoy muy feliz..

Este proyecto me enseñó muchas cosas, y aunque al principio fue un reto, me di cuenta de que se pueden hacer cosas muy buenas. Estoy súper emocionada por crear nuevos juegos en un futuro.

GRACIAS

PROGRAMACIÓN 3 - VIDEOJUEGO