



GENERALITAT  
VALENCIANA  
Presidència



GOBIERNO  
DE ESPAÑA

MINISTERIO  
DE EDUCACIÓN



# MAP ADVENTURE

Proyecto de videojuegos

**AUTOR:**  
**ALBERTO ZAMORA LANDETE**

**TUTOR/A:**  
**ALEJANDRO ASENSI FORÉS**

Junio 2022

les Font de San Lluís

# Índice:

1.	Descripción del proyecto.	Página 4.
1.1.	Carácter del proyecto.	
1.2.	Explicación de pertinencia.	
1.3.	GDD.	
2.	Recursos necesarios.	Página 5.
2.1.	Recursos.	
2.2.	Software	
3.	Presupuesto.	
4.	Temporalización.	Página 6.
4.1.	Idea.	
4.2.	Diseño.	
4.3.	Mejorando ideas.	
4.4.	Programación.	
4.5.	Niveles.	
5.	Financiación y comercialización.	Página 7.
6.	Proyecto finalizado.	
6.1.	Menú.	
6.2.	Selector de niveles.	Página 8.
6.3.	Créditos.	
6.4.	Niveles.	Página 9.
6.4.1.	Tutorial.	
6.4.2.	Nivel 1.	Página 10
6.4.3.	Nivel 2.	Página 11
6.4.4.	Nivel 3.	Página 13
6.5.	Animaciones.	Página 14
6.6.	Unity.	Página 15
6.6.1.	Unity.	
6.6.2.	Animator.	Página 16
6.6.3.	Scripts.	Página 17
7.	Anexos.	Página 22
8.	Bibliografía.	Página 25
8.1.	Tutoriales.	
8.2.	Sonidos (SFX).	
8.3.	Música	
9.	Memoria.	Página 27

## **Resumen:**

Este proyecto se ha creado con la finalidad de hacer un juego para poder tener más experiencia y aumentar el portfolio, y así poder trabajar en el sector de los videojuegos que es lo que me interesa.

Para el juego empecé a crear un juego de plataformas 2D, el cual fue uno de los proyectos que ya vimos en clase y así me resultaba más familiar.

En este juego manejas a Nappy, un ser que siempre está cansado y desea ir a la cama a dormir para poder descansar un rato, durante el viaje encontrarás distintos enemigos que atacarán al jugador si lo golpean, además cogerá caramelos para poder estar más activo durante la aventura.

Para la creación de los diseños utilicé Paint 3D. Y todos los diseños están creados por mí, salvo el del árbol y las nubes del fondo de los niveles.

Para el desarrollo del juego utilicé Unity y Visual Studio para la parte de programación del juego.

# 1: Descripción del proyecto.

1. Título: Nap Adventure

1.1. Carácter del proyecto: Es un proyecto de videojuego.

1.2. Explicación de pertinencia: Relacionado con el módulo de Desarrollo de entornos interactivos multidispositivo, proyecto de videojuegos y diseño de personajes.

## 1.3. Descripción del proyecto (GDD):

**Género del proyecto:** Proyecto de videojuegos de plataformas.

El proyecto es un juego creado para que sirva como portfolio, y además para mis compañeros de clase y amigos cercanos, para que me puedan dar un feedback para mejorar el juego y escuchar sus consejos por si algún día lo subo a algún lado.

El estilo del juego es pixelado porque es un estilo que considero que es bastante bonito y el cual me gusta mucho hacer.

Objetivos a conseguir:

1. Conseguir que el videojuego sea jugable al 100% quitando que sea más corto o más largo.
2. Realizar un videojuego básico con el que aprender un poco más de lo que ya sabía para poder dedicarme a ello algún día en el futuro.
3. Mejorar mis conocimientos de programación y diseño de videojuegos, tanto de personajes como de escenarios.

## **2. Recursos necesarios.**

2.1. - Para la creación del videojuego he utilizado:

Unity - Es gratis.

Paint 3D - Es gratis.

Y fue desarrollado por mí solo, y no seguí ningún tipo de inspiración, simplemente hice a los personajes y los escenarios según mi creatividad.

2.2. - Ordenador: LENOVO

Sistema: 81DC

BIOS: 8RCN44WW

Procesador: Intel(R) Core(TM) i3-6006U CPU @ 2.00Ghz (4 CPUs), ~2.0Ghz.

Memoria: 8192MB RAM

Versión de DirectX: 12.

## **3. Presupuesto.**

Recurso	€/h	Horas invertidas	Total
Ordenador	400€		400€
Diseños	10€	10	100€
Programación	11,5€	35	402,50€
Total			902,50€

## 4. Temporalización.

Proyecto final	Inicio	finalización	Semana	1	2	3	4	5	6	7
Días	En días	En días								
Idea		1		1						
Diseños		1		4						
Mejorar ideas		5		6						
Programación		1		7						
Niveles		4		7						

Esa era la estimación de cómo iba a desarrollarse el proyecto, a sabiendas de mis conocimientos y los distintos tutoriales que ya conocía. Quise ponerme como reto hacerlo en una semana completa.

4.1. Idea: Pensar en la idea del juego

4.2. Diseños: Creación de sprites, tanto de personajes, como de objetos, como de escenario.

4.3. Mejorando idea: Mejorando ideas principales o quitándolas según veía que no encajaban con el juego final.

4.4. Programación: Programar el juego.

4.5. Niveles: Creación de niveles del juego.

## **5. Financiación y comercialización.**

El juego está creado por mí y está hecho con el único propósito de servir como portfolio y de pasarlo bien si me apetece jugarlo alguna vez o ampliar sus niveles o mecánicas, aunque en el plan de “comercialización”, había pensado en subirlo a youtube, por ejemplo, y con un link de descarga en la descripción que te mande al drive, por ejemplo, para poder hacer más fácil que la gente cercana pueda jugarlo si quieren, así solo haría falta pasar el link del vídeo y el resto ya es fácil.

## **6. Proyecto finalizado.**

6.1 - Menú:



En el menú está, junto a un fondo simple: el título del juego con una fuente que he creado yo mismo y cuatro botones, el del play que te manda al nivel del tutorial a aprender a cómo jugar, el del selector de niveles, que veremos más adelante, el de créditos en la esquina derecha inferior, que también veremos más adelante y por último en la esquina izquierda inferior el botón para salir del juego.

## 6.2 - Level select:



En esta pantalla hay tantos botones como niveles que hay que te manda a cada uno de los niveles y un botón para volver al menú principal en la esquina izquierda inferior, al igual que el de salir del menú. Posiblemente ampliable seg\xfcrea seguir creando el juego despu\xe9s de la entrega del proyecto.

## 6.3 - Créditos:



En esta pantalla est\xfan los 3 personajes del juego, Nappy, el personaje principal, Limo, el enemigo terrestre, y Birdie, el enemigo volador, junto al nombre de Azul\_Art, que es mi nombre art\xedstico. Y con el bot\xf3n de volver, una vez m\xfas.

## 6.4 - Niveles:

### 6.4.1 - Tutorial:

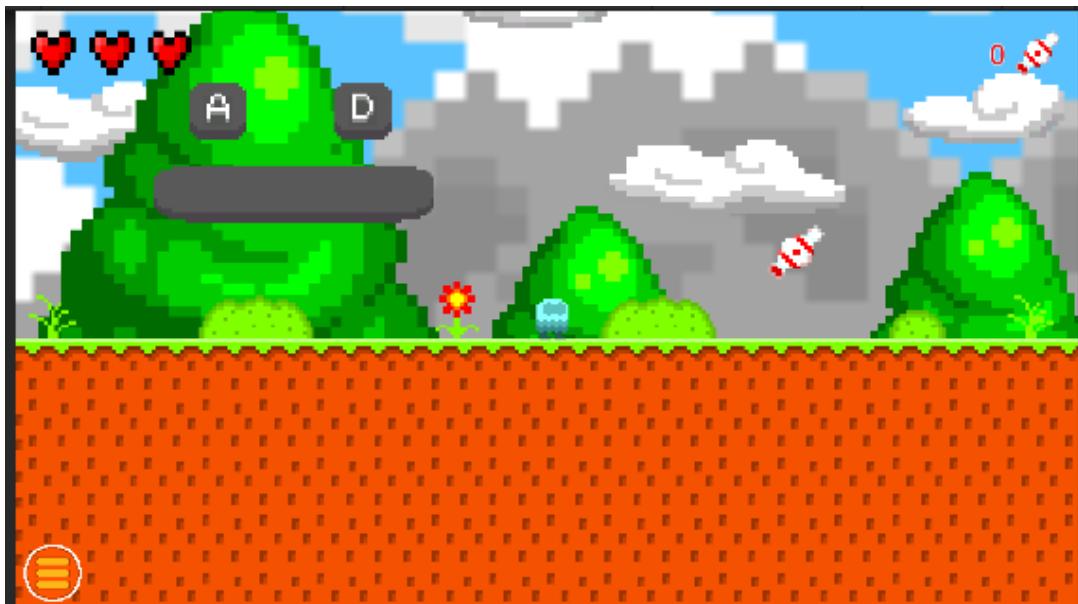


Imagen 1. Comienzo

Al comienzo del nivel del tutorial hay 3 corazones flotantes que te indican los botones que puedes pulsar para avanzar y saltar.

Junto a la interfaz de los niveles, que tiene los corazones de vida en la esquina superior izquierda, el conteo de caramelos en la esquina opuesta y por último el botón de selector de niveles abajo por si quieres salir del nivel a elegir otro diferente.

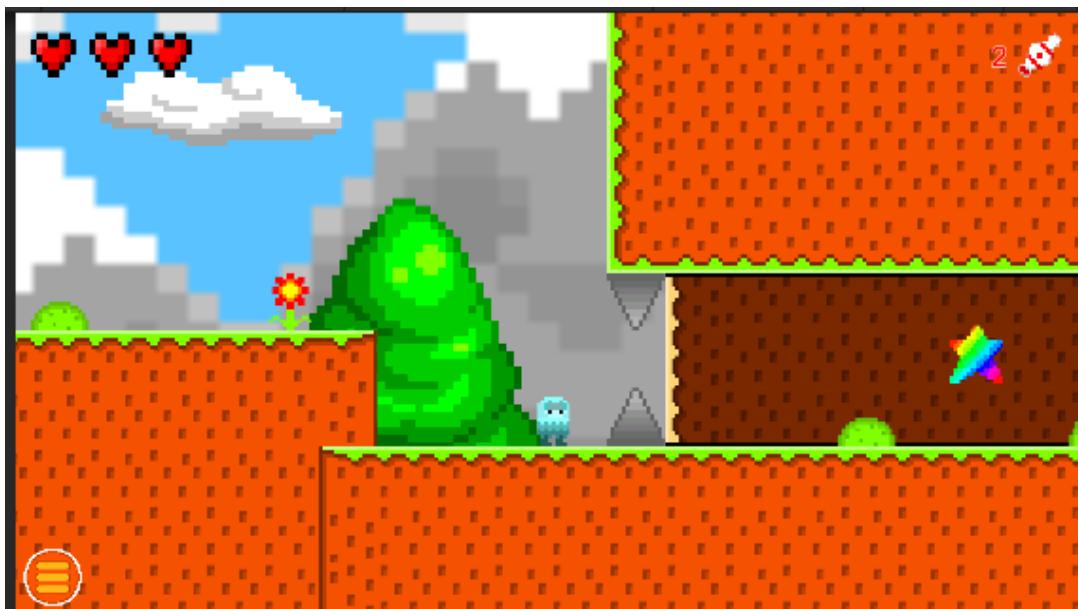


Imagen 2. Pasillo mortal.

Más adelante unos pinchos inesquivables y justo después una estrella que te recupera vida, para que veas que las estrellas recuperan corazones, sin necesidad de explicarlo por texto o por imágenes, como antes.



Imagen 3. Enemigo.

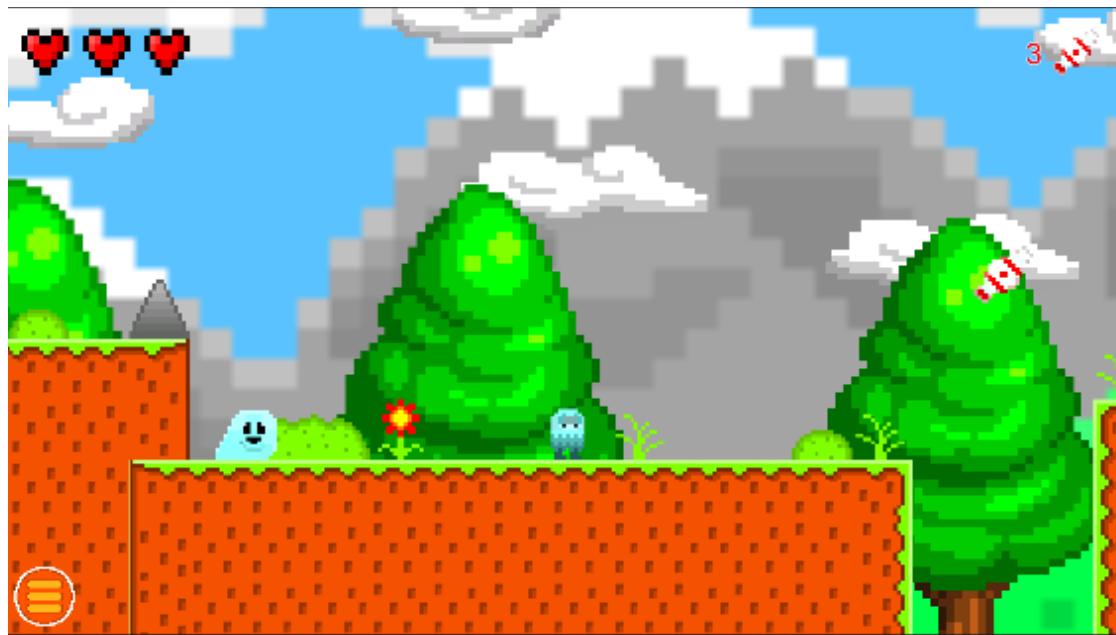
Más adelante un enemigo encerrado en un foso con un caramelito encima que te indica que puedes saltarle encima.



Imagen 4. Final del nivel.

Y por último la cama que es la meta de los niveles.

#### 6.4.2 - Nivel 1:



El nivel uno es un plataformeo más complejo simplemente, con pinchos que ya son esquivables, junto a los enemigos que ya están libres con un movimiento más amplio para poder esquivarlos en caso de no querer matarlos.

#### 6.4.2 - Nivel 2:



Imagen 5. Nuevo enemigo.

En el segundo nivel hay enemigos voladores que entorpecen algunos saltos y con los que tendrás que lidiar para poder hacer los saltos a tu gusto, o con

los que tendrás que impulsar para llegar a caramelos inalcanzables normalmente.



Imagen 6. Cueva

También hay zonas más curradas, y con plataformeo más complejo que en los niveles anteriores.



Imagen 7. Salto infame.

Y por último, unas zonas en las que tendrás que elegir como pasar si quieres continuar, y en las que habrán unos mini checkpoints para no empezar desde el principio, pero cuidado si se te acaban las vidas, porque entonces si que empezará el nivel desde el principio.

#### 6.4.3 - Nivel 3:



Imagen 8. Nuevo ambiente

El último nivel tiene una ambientación distinta al resto, siendo un nivel de nieve y con las decoraciones adaptadas a esta ambientación. Y los pinchos siendo sustituidos por carámbanos para adaptarse también a esta.

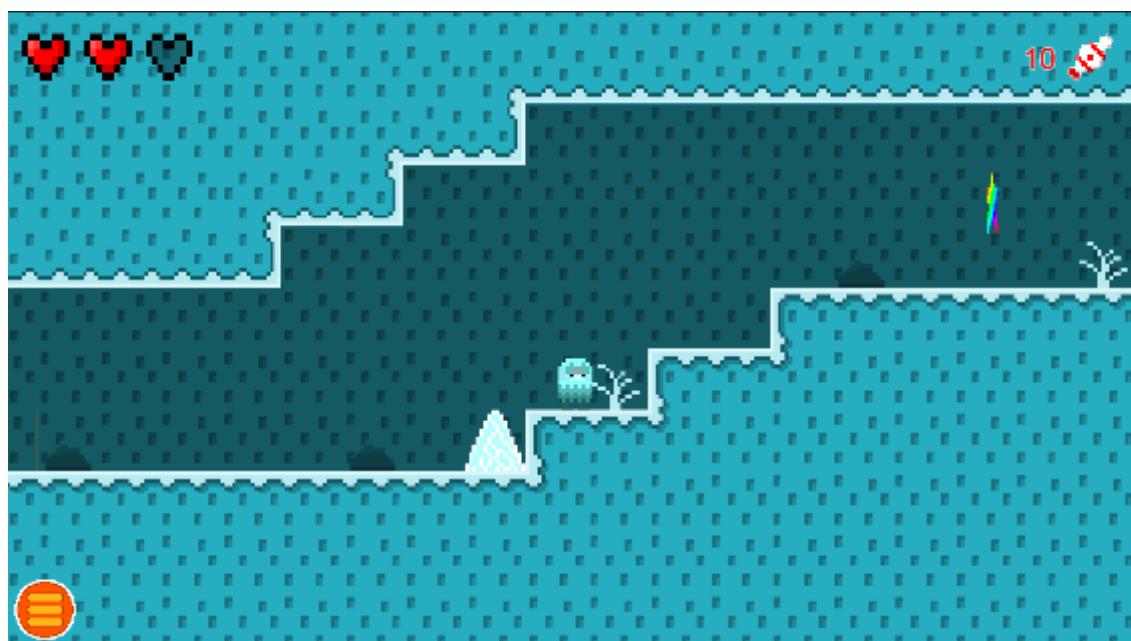


Imagen 8. Cueva

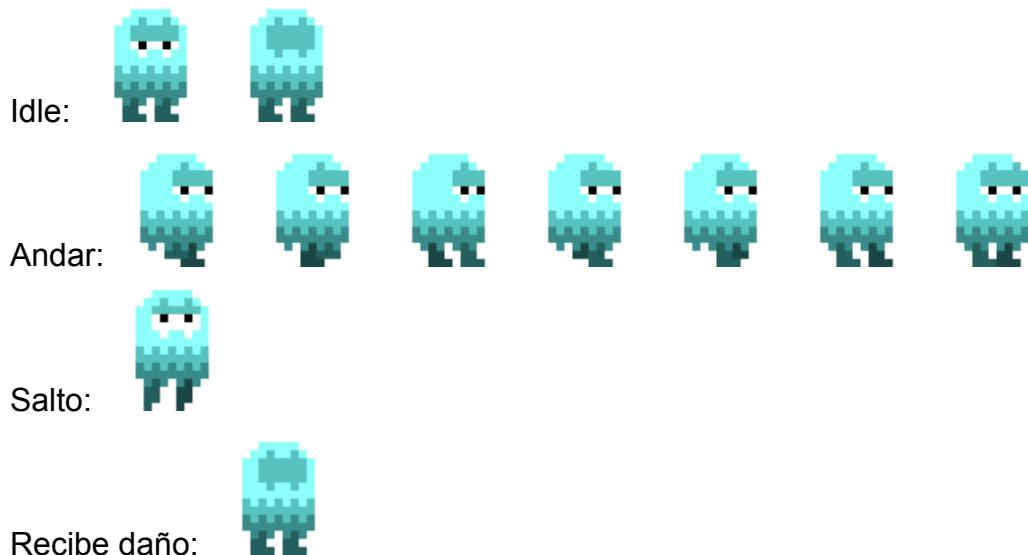
Con algunas zonas cubiertas en las que hay rocas que indican que es una mini cueva en comparación a la zona abierta de antes y que indica el final de la primera parte del nivel.



Imagen 10. Zona oculta.

Y por último algunos plataformeos más complejos y con alguna zona “oculta” con recompensas en ellas.

6.5 - El personaje principal tiene animaciones para el idle, el movimiento, el salto y para el daño:



Los enemigos están animados también:





Birdie:

Además del efecto de muerte del cual ya hablé antes.

## 6.6. - Fotos varias:

### 6.6.1 - Unity.

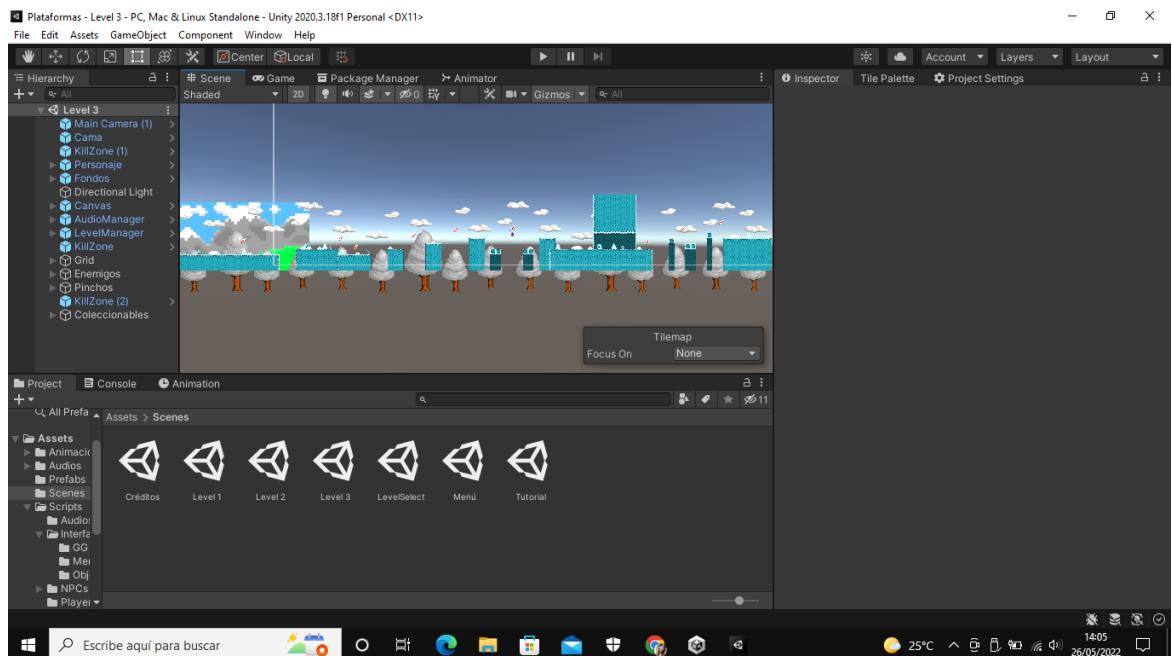


Imagen 11. Escenas de Unity.

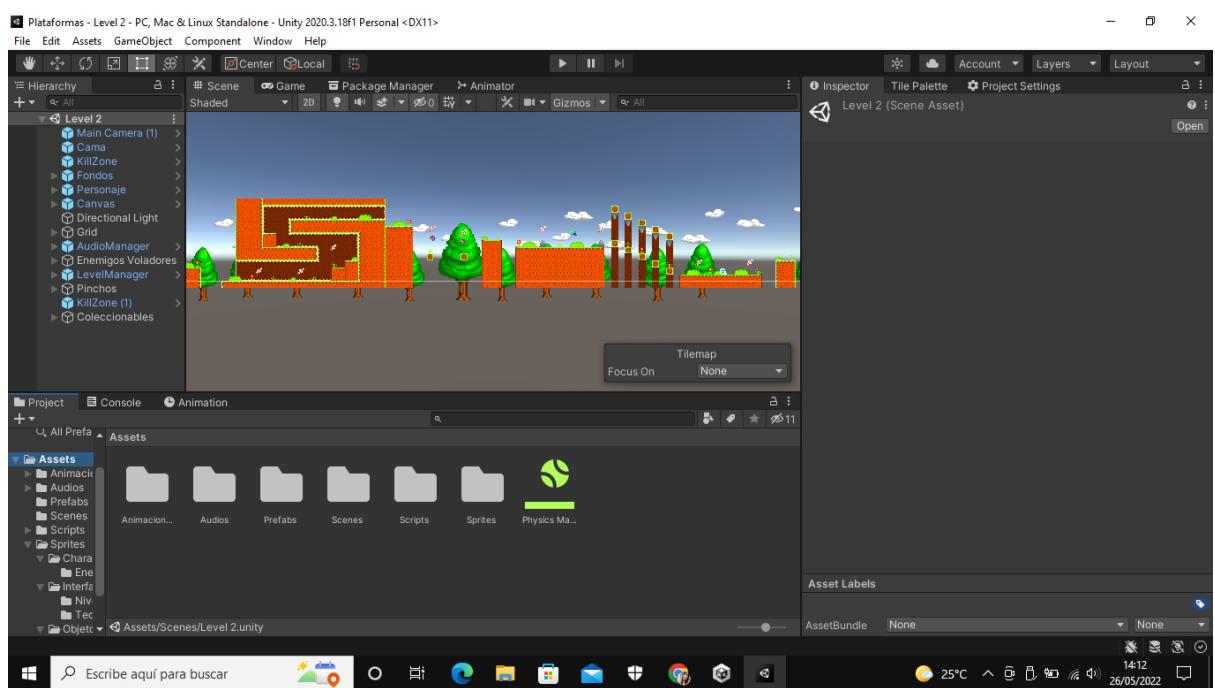


Imagen 12. Carpetas de Unity.

### 6.6.2 - Animator.

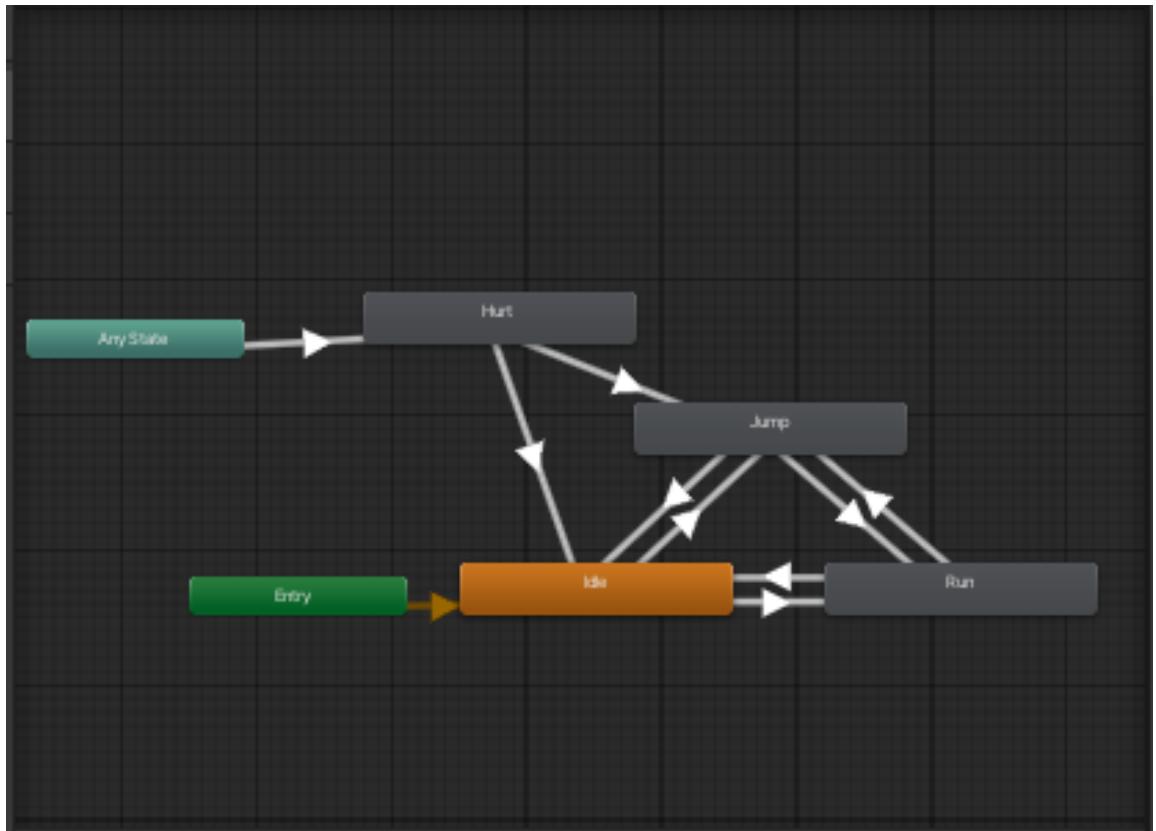


Imagen 13. Animación del jugador.

Este es el animator del personaje que hace un triángulo entre el idle, run y jump. Y que si recibe daño solo vuelve al estado del idle o de jump.

### 6.6.3 - Scripts.

#### Player:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerController : MonoBehaviour
6  {
7      public static PlayerController instance;
8      [Header("Movimiento")]
9      public float moveSpeed;
10     public float jumpForce;
11     public float rebote;
12
13     private bool direccion = true;
14     private float movimiento;
15
16     [Header("Componentes")]
17     public Rigidbody2D RB;
18
19     [Header("Animaciones")]
20     public Animator anim;
21
22     [Header("Tierra")]
23     public bool isGrounded;
24     public Transform groundChekpoint;
25     public LayerMask WhatisGround;
26
27     [Header("Knockback")]
28
29     public float kBLength, kBForce;
30     private float kBCounter;
31
32     void Awake()
33     {
34         instance = this;
35     }

```

Imagen 14. Script Player - Cabecera.

```
--  
37     void Start()  
38     {  
39         anim = GetComponent<Animator>();  
40         direccion = GetComponent<SpriteRenderer>();  
41     }  
42  
43     void Update()  
44     {  
45         movimiento = Input.GetAxis("Horizontal");  
46         RB.velocity = new Vector2(moveSpeed * movimiento, RB.velocity.y);  
47  
48         isGrounded = Physics2D.OverlapCircle(groundChekpoint.position, .2f, WhatisGround);  
49  
50         if (kBCounter <= 0)  
51         {  
52             if (Input.GetButtonDown("Jump"))  
53             {  
54                 if (isGrounded)  
55                 {  
56                     RB.velocity = new Vector2(RB.velocity.x, jumpForce);  
57                     AudioManager.instance.PlaySFX(4);  
58                 }  
59             }  
60  
61             if (direccion == true && movimiento > 0)  
62             {  
63                 flip();  
64             }  
65             else if (direccion == false && movimiento < 0)  
66             {  
67                 flip();  
68             }  
69         }  
}
```

Imagen 15. Script Player - Movimiento

```

70    }
71    else
72    {
73        kBCounter -= Time.deltaTime;
74        if (!direccion)
75        {
76            RB.velocity = new Vector2(-kBForce, RB.velocity.y);
77        }
78        else
79        {
80            RB.velocity = new Vector2(kBForce, RB.velocity.y);
81        }
82    }
83
84    anim.SetFloat("MoveSpeed", Mathf.Abs(RB.velocity.x));
85    anim.SetBool("isGrounded", isGrounded);
86}
87
88 void flip()
89{
90    direccion = !direccion;
91    Vector3 Scaler = transform.localScale;
92    Scaler.x *= -1;
93    transform.localScale = Scaler;
94}
95
96 public void KnockBack()
97{
98    kBCounter = kBLength;
99    RB.velocity = new Vector2(0f, kBForce);
100}
101
102 public void Bounce()
103{
104    RB.velocity = new Vector2(RB.velocity.x, rebote);
105}

```

Imagen 16. Script Player - daño y rebote.

## Controlador de la interfaz:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class UIController : MonoBehaviour
7  {
8      public static UIController instance;
9
10     [Header("Corazones")]
11
12     public Image cor1, cor2, cor3;
13
14     public Sprite corFull, corEmpty;
15
16     [Header("Caramelos")]
17
18     public Text caramelosTx;
19
20     void Awake()
21     {
22         instance = this;
23     }
24
25     // Start is called before the first frame update
26     void Start()
27     {
28         CuentaCaramelos();
29     }
30
31     // Update is called once per frame
32     void Update()
33     {
34     }
35 }
```

Imagen 17. Script UI - Cabecera

```
37     public void UpdateHealthDisplay()
38     {
39         switch (VidaPlayer.instance.vida)
40         {
41             case 3:
42                 cor1.sprite = corFull;
43                 cor2.sprite = corFull;
44                 cor3.sprite = corFull;
45
46                 break;
47
48             case 2:
49                 cor1.sprite = corFull;
50                 cor2.sprite = corFull;
51                 cor3.sprite = corEmpty;
52
53                 break;
54
55             case 1:
56                 cor1.sprite = corFull;
57                 cor2.sprite = corEmpty;
58                 cor3.sprite = corEmpty;
59
60                 break;
61
62             case 0:
63                 cor1.sprite = corEmpty;
64                 cor2.sprite = corEmpty;
65                 cor3.sprite = corEmpty;
66
67                 break;
68
69             default:
70                 cor1.sprite = corEmpty;
71                 cor2.sprite = corEmpty;
72                 cor3.sprite = corEmpty;
73         }
74     }
75
76     public void CuentaCaramelos()
77     {
78         caramelosTx.text = Caramelos.instance.cantCaramelos.ToString();
79     }
80 }
```

Imagen 18. Script UI - Cambio de vidas.

Imagen 19. Script UI - Caramelos.

## Respawn:

```
5  public class Respawnnear : MonoBehaviour
6  {
7      public static Respawnnear instance;
8
9      public GameObject Reaparecer, Reaparecer2, Player;
10
11     public float timeToRespawn;
12
13     // Start is called before the first frame update
14     void Start()
15     {
16         instance = this;
17     }
18
19     // Update is called once per frame
20     void Update()
21     {
22     }
23
24
25     public void RespawnPlayer()
26     {
27         Player.transform.position = Reaparecer.transform.position;
28
29         VidaPlayer.instance.vida = VidaPlayer.instance.maxVida;
30     }
31
32     public void Caida()
33     {
34         Player.transform.position = Reaparecer.transform.position;
35     }
36
37     public void Caida2()
38     {
39         Player.transform.position = Reaparecer2.transform.position;
40     }

```

Imagen 20. Script Respawn - Reapariciones.

He elegido estos scripts, porque considero que son los más importantes del juego. El del player es el que controla el movimiento del jugador con la imagen 15, y la 16 controla el tema de los golpes al jugador.

El de la interfaz controla las vidas del jugador (Imagen 18), los caramelos recolectados (Imagen 19).

El del respawn controla la reaparición del personaje en los niveles (Imagen 20).

## 7. Anexos.



Imagen 21. Tileset Super Mario World.

Para el Tileset me inspiré en el del “overworld” de Super Mario World.



Imagen 22. Enemigo Mario.

También me “inspiré” en este personaje de Mario & Luigi Dream Team Bros para crear al protagonista. (Capnap en inglés y Modorrón en castellano). Simplemente pensé en un personaje de este estilo y como soy fanático de Mario se me vino este personaje a la cabeza y me gustó como quedó el diseño final.



Imagen 23. Fondos.

Este árbol y las nubes los saqué de un juego para móviles llamado “Pixel Fun”.



Imagen 24. Recolor.

Y el árbol nevado lo recoloreé del original.



Imagen 25. Mundo del Super Mario World.

Para el tilemap, aunque no lo usé al final, me inspiré en el mapa del mundo del Super Mario Wolrd, del que en realidad cogí bastantes referencias, como la estética 16x16 pixeles para los diseños de los personajes y tilesets.



Imagen 26. Pincho.

Para los pinchos del juego pensé en los de Sonic, aunque al final me inspiré en los de Geometry Dash para su diseño y función, los cuales actúan como obstáculo para el protagonista y no como algo natural del nivel.



Imagen 27. Saltador (pad).

Para el pad descartado del juego, me inspiré en el pad normal del Geometry Dash. El cual haría la misma función que este pad de Geometry Dash.

## 8. Bibliografía:

### Links:

#### 8.1. Tutoriales:

Bendux (2 junio 2021). *How to make 2D Teleporters in Unity* - (<https://youtu.be/0JXVT28KCIg>)

BlackThornProd (26 febrero 2018). *2D Double/Triple Jump Plataformer Controller - Easy Unity Tutorial* - (<https://youtu.be/QGDeafTx5ug>)

BlackThornProd (28 febrero 2019). *Making Run, Idle & jump 2D Game animations - Unity Tutorial* - (<https://youtu.be/FTxQKHG5WCA>)

Solo Unity (23 agosto 2021). *Crear Juego de Plataformas 2D con Unity - Curso Completo* - (<https://youtu.be/JMT-tgtTKK8>)

#### 8.2. Audios (SFX):

Nintendo - Super Mario World - Jump ( JOSE DAVID 2002 (12 marzo 2021), *Super Mario World - Jump Sound Effect* - (<https://youtu.be/bC4w4s5-IHl>))

Nintendo - Super Mario World - Spin (JOSE DAVID 2002 (12 marzo 2021), *Super Mario World - Spin Sound Effect* - (<https://youtu.be/oklyuMbDZE8>))

Tobey Fox - Undertale - Finishing Puzzle (Dest (6 abril 2016), *Undertale Sound Effect - Finishing Puzzle* - (<https://youtu.be/IgGPNMQhoYM>))

Tobey Fox - Undertale - Restoring Health (Dest (27 marzo 2016), *Undertale Sound Effect - Restoring Health* - (<https://youtu.be/kYfaFrXEeiU>))

Tobey Fox - Undertale - Taking Damage (Dest (19 marzo 2016), *Undertale Sound Effect - Taking Damage* - ([https://youtu.be/Dzwdr\\_RdFa0](https://youtu.be/Dzwdr_RdFa0)))

#### 8.3. Música:

AlphaDream - Mario & Luigi Bowser's inside story - To a New Adventure! -  
(DeoxysPrime (17 febrero 2015), *To a New Adventure! - Mario & Luigi: Bowser's Inside Story [OST]* - (<https://youtu.be/bBFvBHgX0gs>))

AlphaDream - Mario & Luigi Bowser's inside story - The Wind that blows over cavi cape (DeoxysPrime (17 febrero 2015), *The Wind That Blows Over Cavi Cape - Mario & Luigi: Bowser's Inside Story [OST]* - (<https://youtu.be/JmfcLUhsmjc>))

AlphaDream - Mario & Luigi Bowser's inside story - The Wind that blows over cavi cape (DeoxysPrime (17 febrero 2015), *The Wind That Blows Over Cavi Cape (Inside Bowser) - Mario & Luigi: Bowser's Inside Story [OST]* - (<https://youtu.be/-0zOCGBtKJE>))

Nintendo - Super Mario World - World Map (Mooglecito (4 noviembre 2010), *Música de Super Mario World - Mapa del Mundo* - (<https://youtu.be/A-7bOwJ9eZo>))

## 9. Memoria:

Planificación de la idea.

La idea era hacer un plataformas 2D.

No se me ocurrían maneras de crear personajes, que fue lo que más estuve pensando, pero a base de ir haciendo varios diseños, me quedé con este:..



Su nombre es Nappy, que viene de Nap, siesta en inglés, y hecho como si fuera un diminutivo.

Posteriormente hice a los enemigos básicos:



El primero es Limo, es un slime que camina por las plataformas sin llegar a caerse.

Y el segundo es Birdie, como el término de golf, que es un enemigo que vuela de izquierda a derecha y que obstaculiza algunas zonas en las que tendrás que calcular como pasar por él.

También creé los tilesets para empezar a tener escenarios con los que poder crear pruebas y los niveles futuros y los objetos.

El tileset básico es de césped y tierra, con decoraciones de hierbas, arbustos y flores, el segundo tileset era de nieve, por lo que tenía un tono azul y blanco para dar ambiente de nieve y hielo, con hierbas congeladas, nieve y rocas para las zonas parecidas a cuevas heladas.

(La zona más oscura es una pared de fondo).



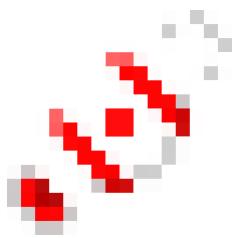
Junto al Tileset creé los pinchos que dañarían al personaje si los toca.



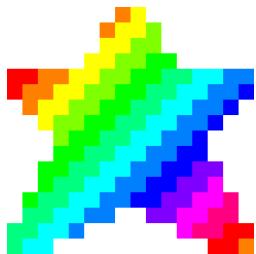
Posteriormente creé el carámbano, el cual era el pincho de la zona de nieve, para que estuviese más acorde a la temática de nieve y hielo.



Pensé en varios objetos para ser colecciónables, pero al final me quedé con los caramelos, porque le sirven a nuestro personaje como método de activación, pues siempre está somnoliento.



Y también hice la estrella, la cual sirve para curar tus heridas.

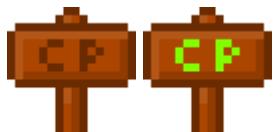


Después de eso creé 2 efectos, uno para la muerte de los enemigos y otro para la recolección de caramelos.



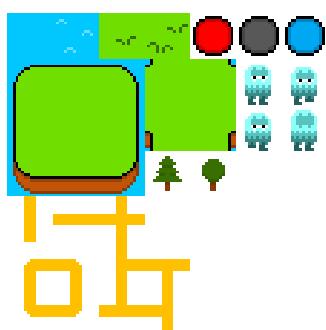
El primero para los enemigos y el segundo para los caramelos.

También iba a haber un sistema de checkpoints, pero al final no me gustó la idea y dejé esa mecánica en que si caes al vacío halla una especie de checkpoint, pero si mueres perdiendo corazones empiezas el nivel completo.



(Estos iban a ser los carteles que activarían los checkpoints, el primero es el cartel normal y el segundo el cartel con el checkpoint activado)

Hay una pantalla de selector de niveles donde escogerás entre los distintos niveles que hay en el juego, sin restricciones, que al principio se iba a construir como un mapa con niveles a lo “Súper Mario Bros” con este tileset que creé por si lo usaba al final:



Y una pantalla de créditos del juego, para que la gente sepa los nombres oficiales de los personajes y donde está mi nickname para que sepan quien ha creado el juego.



Creé este fondo también para tener un fondo en los niveles que dieran vida a los distintos niveles del juego, que en realidad es una imagen reutilizada del proyecto anterior, en el que hay unas montañas que parecen tener caras junto a un terreno enfrente de un acantilado.

Además también reciclé los botones del “play” y de “atrás”:



A partir de estos hice el de selector de niveles:



Y junto a este el de los distintos niveles a elegir:



**Diseño de niveles** = Los niveles no fueron muy complicados de hacer, hice un nivel que funciona como tutorial, muy simple, donde te enseñan las mecánicas básicas del juego y poco más.

El primer nivel era un nivel fácil con algún plataformeo un poco más complejo que en el tutorial, pero fácil todavía. Además enemigos que tienen mucho rango de caminata para que puedas calcular tus saltos mejor y para que no te obstaculice demasiado a la hora de esquivarlos.

El segundo nivel tiene un plataformeo más complejo y con nuevos enemigos voladores que molestan un poco a la hora de calcular los saltos pero sin que sea injusto acabar con ellos, además con un poco más de decoración.

El tercer y último nivel creado, tiene una dificultad más elevada y con los enemigos y carámbanos molestando más o teniendo que calcular cuándo saltar y cómo hacerlo. Además de una ambientación de nieve, para contrastar más con los anteriores niveles.

**Creando el juego** = Estuve siguiendo un tutorial general donde explicaban como hacer un juego de plataformas2D, y con los problemas que tuve en algunas partes, seguí otros tutoriales que ya habíamos visto en clase que me sirvieron bastante para poder acabar de pulir los detalles que con el otro no me funcionaban o que prefería hacerlos más simples y no tan complicado con los que conseguí crear el juego. Una vez tenía creado los scripts necesarios para moverse en y por los niveles, la interfaz, el conteo de vidas y caramelos, los enemigos y el respawn de los niveles, el siguiente paso era muy sencillo, hacer lo necesario un prefab y luego en los siguientes niveles poner los prefabs para la creación de los distintos niveles del juego.

Hay un selector de niveles, para poder disfrutar del nivel que quieras en cada momento, o un botón de “Start” para comenzar desde el tutorial directamente.

Al principio iban a haber más mecánicas que no acabaron en el proyecto final, por tiempo, dificultades técnicas o porque no me gustó la idea de ponerlo al final. Como el mapa de niveles, el sistema de checkpoints, persecución de enemigos o objetos potenciadores.

### **Ideas descartadas:**

A lo largo del proyecto fueron descartadas varias ideas, por falta de tiempo o porque no me gustaban como quedaban en el juego, a continuación varias ideas:

El checkpoint y el mapa de niveles ya han sido explicados anteriormente, con lo que no voy a explicarlos de nuevo.

### Plataformas que se mueven:

Estas plataformas iban a ser lo mismo que los pájaros pero en vez de moverse en vertical, en horizontal. Al final no fue introducida porque no me gustaba mucho la mecánica, y además no sabía en qué nivel ponerlo y de qué manera introducirlo para que funcionase en todo el nivel y no solo una o dos veces.



### Saltadores automáticos:

Estos iban a ser unos pads que iban a estar en el suelo y que iban a impulsar al personaje con un salto más alto de lo normal, pero no me gustó mucho la mecánica después de verlo en acción, aunque no hubiese venido mal introducirlo de todos modos aunque sea por variar el gameplay, tal vez en una actualización.



#### Seguimiento de enemigos:

Al principio pensé en que los enemigos voladores te persiguiesen al verte, pero luego pensé que no vendría bien, porque la idea de estos era que estuviesen a su royo y molesten sin necesidad de perseguirte.

#### Doble salto:

Pensé en hacer que el personaje pudiera hacer un doble salto, pero no me gustó porque quería hacer que el juego fuese “realista” en ese aspecto, además que nunca me gustó demasiado la idea de hacer dobles saltos.

#### Temporalización:

Al final el reto duró un día menos de lo que me había propuesto, los diseños del juego al final duró un día más

A lo largo del tiempo he tenido varias dificultades, que son las siguientes:

#### Checkpoints:

Al principio quería poner checkpoints para no empezar de 0 cada vez que morías, pero con problemas con el código al final descarté la idea y decidí hacer niveles no muy largos para que no sea frustrante empezar de cero si mueres. Aunque si que creé una subzona en la que si caías al vacío en los niveles 2 y 3 apareces en la mitad del nivel y no al principio, aunque si mueres si que empiezas desde cero, pues ahí si que mueres por falta de corazones.

Un mapa: Quería hacer una especie de mapa por el que caminarías para entrar en los distintos niveles, pero por pensar que no me daría tiempo o fallos que podrían salir no lo terminé haciendo, aunque sí que creé el tileset del mapa.

Audios: Al principio había algunos audios que no sonaban cuando tocaban, aunque cuando entendí que era lo que fallaba fue muy sencillo solucionarlo.

Canvas hecho prefab: Al final del proyecto, creé la escena de créditos, metí el prefab del canvas para la interfaz y olvidé desvincularlo del prefab, con lo que borré casi todo y eso actualizó todas las interfaces del juego, lo guardé sin darme cuenta y cuando me puse a jugar niveles me di cuenta de que la había liado. Por suerte fue fácil de arreglar, pues el script que tenía con las interfaces no se habían borrado,

solo tuve que rehacer la sección de corazones y la de caramelos y vincularlos al script. Antes de arreglarlo, desvinculé el canvas de los créditos para que no me molestara de nuevo.

Respawn: Cuando el personaje respawnaba, lo hacía encogiéndose o agrandándose, cuando entendí que lo que había pasado era que se había movido en el eje “Z”, solo tuve que actualizar su posición a la del nivel al igual que la del respawn.

Diseños descartados o que no llegaron a tener funciones en el juego: Hubieron varios diseños que creé que al final no utilicé, por no usar su mecánica al final o por no gustarme el diseño y crear otro, como el del protagonista o otras ideas descartadas.

Colliders de Limo: Al principio tenía un problema con los colliders de enemigo “Limo” el cual aunque le saltaras y lo mataras, dañaba al personaje. Por lo que después de estar retocando los colliders decidí que el collider para matar al enemigo estuviese más alejado de el del personaje, por lo que ahora es menos común que pase ese bug, aunque si caes mal puedes sufrir daño igualmente, solo que si lo alejo más parece antinatural.