

Tarea 5: Mi primer videojuego

JESÚS MARTÍN ROBLES

DAM a distancia Curso 23-24

ÍNDICE:

Detalle de la tarea	2
Capturas de Pantalla	4
Dificultades	15

Detalle de la tarea

Funcionamiento básico del juego:

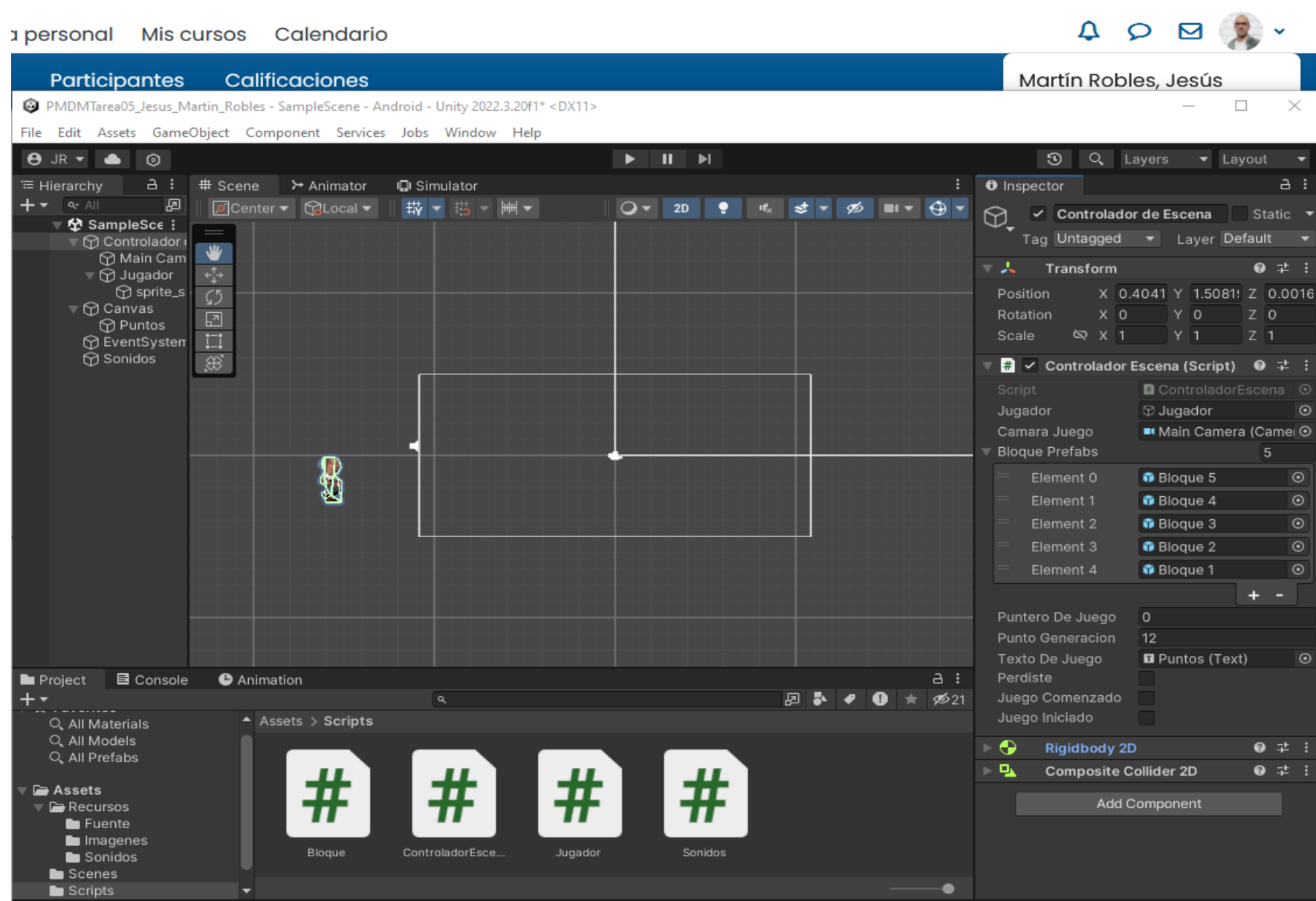
- El juego consiste en un avatar que se mueve automáticamente hacia la derecha a través de un desierto continuo, mientras el jugador lo controla para saltar obstáculos como cactus y piedras. El objetivo es evitar colisionar con los obstáculos para mantenerse en juego el mayor tiempo posible. El fondo del desierto se desplaza de derecha a izquierda para simular movimiento y el juego incluye efectos de sonido para los saltos y una melodía de fondo además de un marcador con la puntuación. Si el avatar choca con un obstáculo, el juego termina mostrando un mensaje de "GAME OVER" y ofrece la opción de reiniciar.

Clases utilizadas en su desarrollo:

- La clase **ControladorEscena** es el controlador principal del juego. En el método `Start`, establece la posición inicial del jugador y algunas variables de estado del juego. En el método `Update`, que se llama una vez por frame, maneja la entrada del usuario para iniciar el juego y reiniciar después de perder. También mueve la cámara para seguir al jugador y actualiza el texto del juego para mostrar los puntos del jugador o el mensaje de "Game Over". Además, en cada frame, si el juego ha comenzado, genera nuevos bloques en la escena del juego basándose en la posición del jugador y un punto de generación predefinido. Los bloques se instancian a partir de un conjunto de prefabs y se colocan en la escena en una posición calculada.
- La clase **Jugador** controla el comportamiento del jugador en el juego. En el método Start, establece la posición inicial del jugador, su velocidad inicial y obtiene referencias al controlador de la escena y al animator del jugador. En el método Update, que se llama una vez por frame, controla el movimiento del jugador. Si el juego ha comenzado, el jugador se mueve hacia adelante a una velocidad constante. Si el jugador está en el suelo y se presiona el botón izquierdo del mouse, el jugador salta. Además, se controla la animación del jugador en función de si el juego ha comenzado o no. Finalmente, los métodos OnCollisionEnter2D y OnTriggerEnter2D manejan las colisiones y los contactos del jugador con otros objetos. Si el jugador colisiona o entra en contacto con un objeto con la etiqueta "Obstaculo", el jugador es destruido.

- La clase **Sonidos** se encarga de la gestión de los sonidos en el juego. En el método ``Awake``, se implementa un patrón singleton para asegurar que solo exista una instancia de ``Sonidos`` en el juego. Si ya existe una instancia, se destruye el objeto actual. La clase proporciona dos métodos públicos, ``PlaySalto`` y ``PlayGolpe``, que se utilizan para reproducir los sonidos de salto y golpe respectivamente. Estos métodos establecen el clip de audio correspondiente en el ``AudioSource`` y luego lo reproducen. Finalmente, en el método ``OnDestroy``, se asegura de que la referencia estática a la instancia se establezca en null cuando el objeto se destruya.
- La clase **Bloque** representa un bloque en el juego. Actualmente, solo tiene una propiedad pública `Tamaño` que representa el tamaño del bloque. Los métodos `Start` y `Update` están presentes pero vacíos, lo que indica que no se realiza ninguna acción específica al inicio o en cada frame para los objetos de esta clase.

Capturas de Pantalla



Unity 2022.3.20f1* <DX11>

File Edit Assets GameObject Component Services Jobs Window Help

Hierarchy | Scene | Animator | Simulator

SampleScene

- Controlador
- Main Cam
- Jugador
- sprite_s
- Canvas
- Puntos
- EventSystem
- Sonidos

Inspector


Main Camera

- Tag: MainCamera
- Layer: Default

Transform

- Position: X -0.404 Y -1.508 Z -10.00
- Rotation: X 0 Y 0 Z 0
- Scale: X 1 Y 1 Z 1

Camera

- Clear Flags: Solid Color
- Background: 
- Culling Mask: Everything
- Projection: Orthographic
- Size: 5
- Clipping Planes: Near 0.3 Far 1000
- Viewport Rect: X 0 Y 0 W 1 H 1
- Depth: -1
- Rendering Path: Use Graphics Settings
- Target Texture: None (Render Texture)
- Occlusion Culling: ☐
- HDR: Use Graphics Settings
- MSAA: Off
- Allow Dynamic Resol: ☐
- Target Display: Display 1
- Target Eye: None (Main Display)

Audio Listener

- Add Component

Project | Console | Animation

Assets > Scripts

- Bloque
- ControladorEsce...
- Jugador
- Sonidos

Participantes

Calificaciones

Martín Robles, Jesús

PMDMTarea05_Jesus_Martin_Robles - SampleScene - Android - Unity 2022.3.20f1* <DX11>

File Edit Assets GameObject Component Services Jobs Window Help

JR

☁️

⚙️

▶️ ⏸️ ⏹️

🔄

🔍

Layers

Layout

Hierarchy

Scene

Animator

Simulator

Center

Local

2D

Light

🔊

👁️

📐

🔄

SampleSc...

Controlador...

Main Cam

Jugador

sprite_s

Canvas

Puntos

EventSystem

Sonidos

👤

🔄

📐

🔍

🔄

Position

X

-8.95

Y

-4.49

Z

-0.001

Rotation

X

0

Y

0

Z

0

Scale

X

1

Y

1

Z

1

Rigidbody 2D

Body Type

Dynamic

Material

None (Physics Mater...

Simulated

☑️

Use Auto Mass

☐

Mass

2

Linear Drag

0

Angular Drag

0.05

Gravity Scale

4.5

Collision Detection

Discrete

Sleeping Mode

Start Awake

Interpolate

None

Constraints

Freeze Position

☐ X ☐ Y

Freeze Rotation

☑️ Z

Layer Overrides

Include Layers

Nothing

Exclude Layers

Nothing

Info

Jugador (Script)

Script

Jugador

Fuerza Salto

2100

Velocidad Movimient

8

En Suelo

☐

Controlador Escena

None (Controlador E...

Project

Console

Animation

All Materials

All Models

All Prefabs

Assets

Recursos

Fuente

Imagenes

Sonidos

Scenes

Scripts

Assets > Scripts

Bloque

ControladorEsce...

Jugador

Sonidos

Participantes

Calificaciones

Martín Robles, Jesús

PMDMTarea05_Jesus_Martin_Robles - SampleScene - Android - Unity 2022.3.20f1* <DX11>

File Edit Assets GameObject Component Services Jobs Window Help

JR

☁️

🏠

▶️

⏸️

⏩

🕒

🔍

Layers

Layout

Hierarchy

+

🔍 All

📁

SampleScene

Controlador

Main Cam

Jugador

sprite_s

Canvas

Puntos

EventSystem

Sonidos

Scene

Animator

Simulator

Center

Local

2D

Light

🔊

👁️

🔍

📏

👤

🔄

📏

🔍

📏

🔍

📏

🔍

Inspector

📦

sprite_sheet_jesus_0

Static

Tag

Untagged

Layer

Default

Transform

Position

X -7.18 Y 1.48 Z 0

Rotation

X 0 Y 0 Z 0

Scale

X 2.3077 Y 2.2508 Z 1

Sprite Renderer

Sprite

sprite_sheet_jesus_0

Color

Flip

X Y

Draw Mode

Simple

Mask Interaction

None

Sprite Sort Point

Center

Material

Sprites-Default

Additional Settings

Sorting Layer

Default

Order in Layer

0

Animator

Material

None (Physics Mater)

Is Trigger

Used By Effector

Used By Composite

Auto Tiling

Offset

X 0 Y 0

Use Delaunay Mesh

Points

Layer Overrides

Info

Project

Console

Animation

+

🔍

📁

🔍

🔍

🔍

Assets

Scripts

📄

Bloque

📄

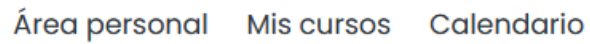
ControladorEsce...

📄

Jugador

📄

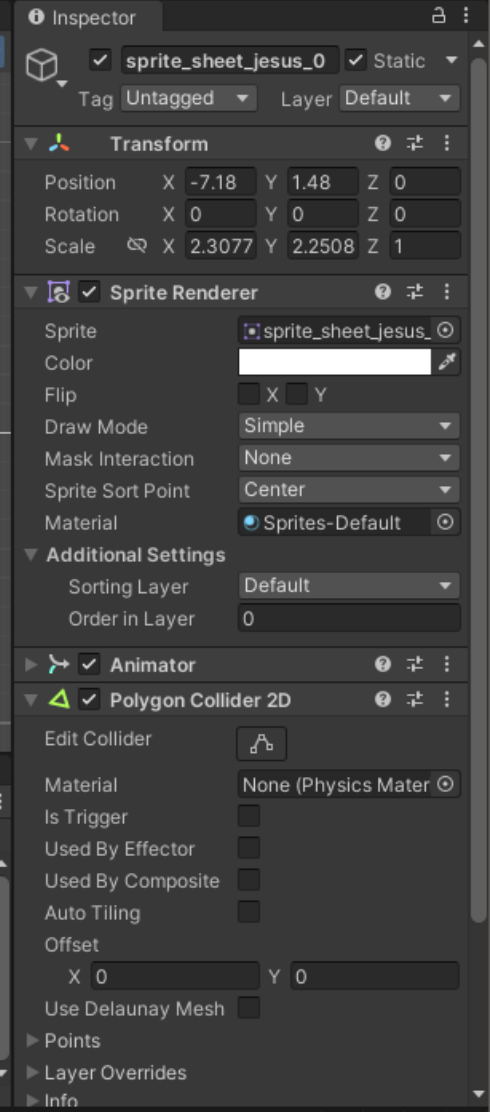
Sonidos



Curso	Participantes	Calificaciones
-------	---------------	----------------

PMDMTarea05_Jesus_Martin_Robles - SampleScene - Android - Unity 2022.3.20f1 <DX11>

File Edit Assets GameObject Component Services Jobs Window Help



— □ ×

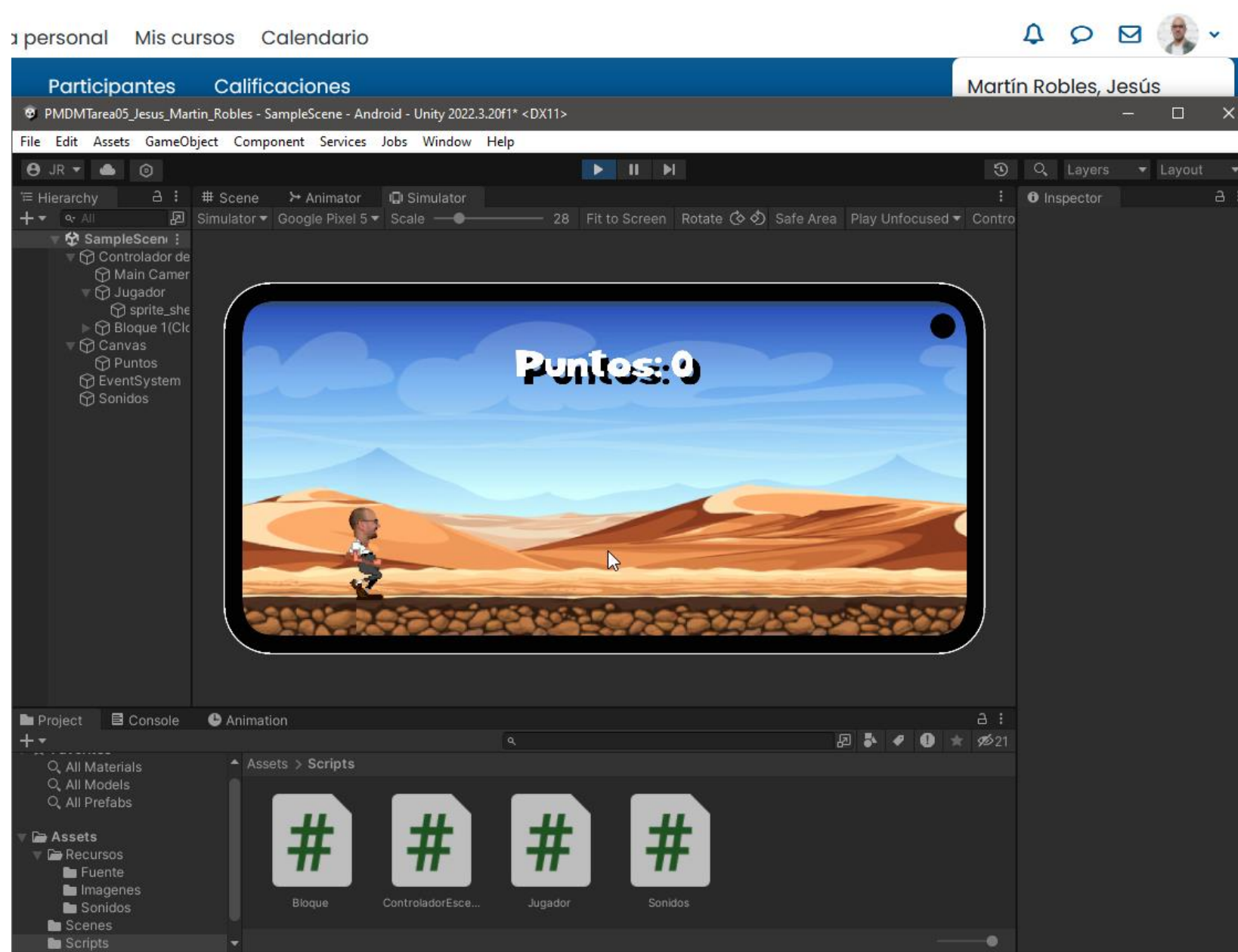
 JR

Layers
 Layout

The screenshot displays the Unity 2020.3.10f1 interface. The Hierarchy panel on the left lists the scene's structure: SampleScene (Controlador, Main Cam, Jugador, sprite_s, Canvas, Puntos, EventSystem, Sonidos). The Inspector panel on the right shows the properties of the 'Sonidos' object, including Transform, Sonidos (Script), and Audio Source.

- En los diferentes objetos se han usado diferentes componentes, tales como:
 - **Transform:** Determina la posición, rotación y escala de un objeto en la escena.
 - **Script:** Contiene el código que controla el comportamiento de un objeto.
 - **Rigidbody2D:** Aplica físicas 2D a un objeto, permitiendo movimientos y colisiones.
 - **Composite Collider 2D:** Combina múltiples colisionadores en uno para optimizar las colisiones.
 - **Camera:** Captura y muestra una vista de la escena al jugador.
 - **Audio Listener:** Recibe el audio de los Audio Source y lo reproduce para el jugador.
 - **Animator:** Controla las animaciones de un objeto.
 - **Sprite Renderer:** Dibuja un sprite para un objeto en la escena.
 - **Polygon Collider 2D:** Define una forma para las colisiones 2D de un objeto.
 - **Canvas Scaler:** Ajusta el tamaño y la escala de un Canvas.
 - **Graphic Raycaster:** Permite la interacción con la interfaz de usuario mediante rayos.
 - **Canvas:** Es el área donde se dibujan los elementos de la interfaz de usuario.
 - **Canvas Renderer:** Se encarga de dibujar los elementos de la interfaz de usuario en un Canvas.
 - **Text:** Muestra texto en la interfaz de usuario.
 - **Audio Source:** Reproduce sonidos en la escena.

- A continuación, se muestran capturas de pantalla del juego en funcionamiento:





Participantes

Calificaciones

Martín Robles, Jesús

PMDMTarea05_Jesus_Martin_Robles - SampleScene - Android - Unity 2022.3.20f1* <DX11>

File Edit Assets GameObject Component Services Jobs Window Help

JR ☁️ ⚙️

▶️ ⏸️ ⏩


🔄 🔍 Layers ▾ Layout ▾

Hierarchy 🔒

Scene ▶️ Animator 📄 Simulator

🔍 All 📄 Simulator ▾ Google Pixel 5 ▾ Scale ————— 28 Fit to Screen Rotate 🔄🌀 Safe Area Play Unfocused ▾ Contro

▶️ SampleScene :



Project Console Animation 🔒

🔍

📄 🗑️ 🔄 ⓘ ⭐ ⚙️ 21

Assets

Assets > Scripts

Bloque

ControladorEsce...

Jugador

Sonidos

Participantes

Calificaciones

Martín Robles, Jesús

PMDMTarea05_Jesus_Martin_Robles - SampleScene - Android - Unity 2022.3.20f1* <DX11>

File Edit Assets GameObject Component Services Jobs Window Help

JR ☁️ 🏠


▶️ ⏸️ ⏭️

🕒 🔍 Layers ▾ Layout ▾

Hierarchy 🔍 All 📄 Simulator

Scene Animator Simulator Google Pixel 5 Scale 28 Fit to Screen Rotate ⚙️ ⚙️ Safe Area Play Unfocused Contro

▶️ SampleSceni



Project Console Animation

Assets > Scripts

Bloque

ControladorEsce...

Jugador

Sonidos

Dificultades

Durante el proceso de desarrollo del juego me enfrenté a varios desafíos técnicos que requirieron de ajustes específicos y aprendizaje continuo. A continuación, describo algunas de las dificultades más destacadas, aunque no las únicas, y las soluciones que apliqué:

1. **Ajuste de Collider:** Inicialmente, utilicé un BoxCollider2D para el personaje, lo cual no se adaptaba correctamente al contorno del mismo. Esto provocaba que el personaje pareciera flotar sobre el suelo. Tras experimentar con diferentes configuraciones, opté por cambiar a un PolygonCollider2D, lo cual permitió un mejor ajuste a la forma del avatar y resolvió el problema de la flotación.
2. **Configuración del Salto:** El salto del personaje era inicialmente demasiado alto y lento, lo que afectaba la jugabilidad. Mediante pruebas iterativas, ajusté la fuerza de salto, la masa y la gravedad del personaje hasta conseguir una dinámica de salto que resultó satisfactoria y realista dentro del contexto del juego.
3. **Fondo Infinito:** Crear un efecto de fondo continuo e infinito fue particularmente desafiante. La solución llegó mediante la implementación de una serie de bloques que se suceden de manera aleatoria y horizontal, un método que aprendí gracias a un tutorial en YouTube. Este enfoque logró dar la impresión de un desierto interminable, vital para la inmersión en el juego.
4. **Control de Inicio y Posición del Personaje:** Garantizar que el personaje permaneciera en el lado izquierdo de la pantalla y no comenzara a moverse hasta que el usuario tocara la pantalla inicialmente supuso un reto. Este comportamiento era esencial para la mecánica de inicio del juego, y su implementación correcta fue crucial para la experiencia del usuario.
5. **Estabilidad del Personaje:** Al principio, el personaje giraba y volcaba después de algunos saltos, lo que resultaba en una postura acostada que comprometía la estética y funcionalidad del juego. Resolví este problema activando la opción de Freeze Rotation en el componente Rigidbody del personaje, lo cual estabilizó su orientación y mejoró significativamente la jugabilidad.

Cada uno de estos problemas fue una oportunidad de aprendizaje y me ayudó a profundizar mi comprensión de Unity y el desarrollo de juegos en general. Estas experiencias subrayan la importancia de la experimentación y la adaptabilidad en el proceso de desarrollo de software.