

Taller de Programación en C# para Unity: Creación de Personajes Jugables, Escenarios con Tiles, Cambio de Escenario con Triggers, y Objetos Coleccionables

Ingeniería en diseño de Entretenimiento Digital

Taller 2: Métodos de diseño

Objetivo: En este taller, crearás tres personajes jugables y tres escenarios diferentes utilizando el sistema de tiles en Unity. Los personajes tendrán variables y configuraciones básicas para moverse y saltar, y los escenarios tendrán plataformas que afectan las variables de los personajes. Implementarás objetos con Triggers para cambiar de escenario y objetos coleccionables que modifican variables específicas de los personajes.

Actividades:

1. Creación de los Personajes:

Personaje 1:

- **Movimiento:** Se mueve solo con las teclas A (izquierda) y D (derecha).
- **Salto:** Salta con la tecla Espacio.
- **Variables:** Tiene una variable “health” de tipo float.
- **Collider:** Usa un Box Collider 2D.
- **Physics Material:** Fricción 0.5, rebote 0.

Personaje 2:

- **Movimiento:** Se mueve solo con las flechas izquierda y derecha.
- **Salto:** Salta con la flecha hacia arriba.
- **Variables:** Tiene una variable “coins” de tipo int.
- **Collider:** Usa un Capsule Collider 2D.
- **Physics Material:** Fricción 0, rebote 0.3.

Personaje 3:

- **Movimiento:** Se mueve solo con dos teclas de elección libre, diferentes a los personajes anteriores.

- **Salto:** Salta con una tecla de elección libre diferente a las anteriores diferente a los personajes anteriores.
- **Variables:** Tiene una variable “evolution” de tipo bool.
- **Collider:** Usa un Polygon Collider 2D.
- **Physics Material:** Fricción 0.3, rebote 0.5.

Configuración Común:

- Todos los personajes deben tener un nuevo script llamado “PlayerController” más un diferenciador para cada personaje y las variables y configuración básica para moverse y saltar que se ha implementado en las clases, es decir, variables que modifiquen la velocidad y el salto del personaje, variables que se utilicen para detectar el tipo de Layer de una plataforma que se esta pisando y una variable que detecte si se puede saltar o no.

2. Creación de Prefabs:

- Configura cada personaje con el script correspondiente y asegúrate de ajustar las variables públicas en el Inspector de Unity según sea necesario.
- Crea un Prefab para cada uno de los tres personajes con los componentes y configuraciones descritos en una carpeta nueva llamada “Prefabs”.

3. Creación de Tilemaps con Tiles:

a. Preparación de los Tiles:

- Importa los tiles para el fondo, la decoración y las plataformas de los archivos que se encuentran en los materiales dados para el taller.
- Configura el Tile Palette en Unity para organizar tus tiles.

b. Diseño de los Tilemaps:

- Todos los Tilemaps estarán compuestos por 3 elementos: Fondo, Decoración y Plataformas; estos 3 elementos deben de estar separados

de forma de que la única capa con la que pueda interactuar el jugador sea las plataformas, que a su vez están divididas en dos clases:

- **Ground:** Plataformas que están asignadas a una Layer llamada “Ground” y que cuando el jugador este en contacto con estas su velocidad y salto tengan valores de 5.
- **Mud:** Plataformas que están asignadas a una Layer llamada “Mud” y que cuando el jugador este en contacto con estas su velocidad y salto tengan valores de 2.
- Todos los Tilemaps deben de ser diferentes entre ellos y cada uno debe de tener un objeto con un Trigger que va a permitir cambiar entre escenarios una vez toque al personaje. Tenga en cuenta que debe de mover la posición del jugador a una ubicación apta para el siguiente escenario ya que son diferentes y no hacer el cambio de posición podría generar errores.

4. Configuración de las Capas de Plataforma y Ajuste de Variables:

a. Creación de las Layers:

- Crea las Layers “Ground” y “Mud” en el proyecto. Asigna estas capas a las plataformas correspondientes en el escenario.

b. Implementación del Script de Ajuste de Variables:

- Implementa en el script “PlayerController” el código que permita ajustar las variables “speed” y “jumpForce” según la capa en la que el personaje esté tocando con OnCollisionEnter2D.

5. Generación de GameObjects con Trigger y Cambio de Tilemap:

a. Creación del Trigger:

- Crea un objeto en cada escenario que actuará como un Trigger. Este objeto debe tener un Collider2D configurado como Trigger.

- Añade un script llamado “TilemapChanger” a este objeto para manejar el cambio de escenario.

b. Implementación del Script para el Trigger:

- Implementa en el script “TilemapChanger” el cambio del escenario y mover al jugador cuando entra en el trigger. Se puede seguir el ejemplo de clase de cambio de Tilemap, que en este caso sería: Tilemap1 a Tilemap2, Tilemap2 a Tilemap3 y Tilemap3 a Tilemap1
- Asegúrate de que el código busque un PlayerController en el objeto con que colisiona para poder que el código solo se ejecute cuando el jugador sea el que entre en el trigger.

c. Manejo de la Posición de Spawn:

- En cada Tilemap crea un Empty llamado “PlayerSpawn” más el indicador de Tilemap correspondiente para manejar la posición de Spawn del jugador.

6. Creación de Objetos Coleccionables:

a. Creación del Pickup:

- Crea un objeto en cada escenario que actuará como un objeto coleccionable (pickup). Este objeto debe tener un Collider2D configurado como Trigger.
- Añade un script llamado “Pickup” a este objeto para manejar la lógica de recolección.

b. Implementación del Script para el Pickup:

- Implementa un script en C# llamado Pickup que modifique la variable específica del personaje según el tipo de personaje que lo recoja y después de esto se desactive en la escena. Estas son las condiciones para cada jugador:
 - **Personaje 1:** Posee una variable “health” y al tomar el pickup cambia su valor a 60.5

- **Personaje 2:** Posee una variable “coins” y al tomar el pickup cambia su valor a 15
- **Personaje 3:** Posee una variable “evolution” y al tomar el pickup cambia su valor a True. Tenga en cuenta que debe de realizar un código extra en la función Update que detecte si la variable “evolution” es verdadera, en caso de que si lo sea, el color de ese personaje del SpriteRenderer, cambia a verde; en caso de que “evolution” sea False, el color debe de ser blanco.

ENTREGABLE: Al finalizar, realizar una carpeta comprimida en .zip o .rar del proyecto usado para resolver el taller donde solo deben de estar las siguientes carpetas: Assets, Packages, ProjectSettings y UserSettings. Esta carpeta comprimida debe de ser enviada a la tarea correspondiente en el equipo de Teams antes finalizar la clase.