

1. Problemas al pasar del estado PlayerIdleState a PlayerOnAirState. Detecta el suelo mientras salta y lo para a mitad.

Solucionado sustituyendo `Input.GetButtonDown("Jump")` por `Input.GetButton("Jump")`. No soluciona el problema pero añade más fotogramas para que se detecte la acción de saltar del jugador.

2. Al iniciar la escena el jugador aparece en el aire y al caer se incrusta en el suelo. Esto solo pasa al aparecer el jugador, no durante el resto de saltos.

Solucionado por ahora moviendo al Player a una posición pegada al suelo.

3. En el estado PlayerMovingState no se detecta correctamente si el jugador está pulsando LeftArrow y no realiza correctamente el movimiento a la izquierda. El movimiento a la derecha funciona perfectamente. Falla con `Input.GetKey(KeyCode.LeftArrow)` y con `Input.GetAxis("Horizontal")`.

Causa error en la programación. Para decidir si se pasa del estado PlayerMovingState a PlayerStopingState se comprueba si la dirección que sigue el jugador es menor que 0.001 (cercana a cero). Esto hace que si la dirección es negativa, es decir, se está moviendo hacia la izquierda, la dirección es también menor que 0.001 y pasa al estado PlayerStopingState. Esto provoca que ir hacia la derecha provoque una oscilación continua entre PlayerMovingState y PlayerStopingState, resultando en un movimiento estático.

Solucionado sustituyendo:

```
if (Math.Abs(direction) < 0.001f)
```

por:

```
if (Math.Abs(direction) < 0.001f)
```

4. Durante la animación de movimiento del Player, al moverse hacia la izquierda se ejecuta la animación de movimiento hacia la derecha.

Solucionado con el objeto SpriteRenderer asociado al Player. El SpriteRenderer tiene un atributo `flipX` que invierte el Sprite si está a verdadero. Si el Player va a la derecha `flipX` estará a falso y se mantendrá la animación de movimiento a la derecha. Si el Player va a la izquierda `flipX` estará a verdadero y se mantendrá la animación de movimiento a la izquierda.

5. Al hacer cambios bruscos de dirección durante PlayerMovingState(), a veces no detecta el cambio de dirección y se queda enganchado en yendo en una dirección.

El problema yace en que a pesar de que se está usando la función `GetAxis()` si se cambia la dirección del movimiento "bruscamente" (por ejemplo pulsar la tecla de flecha derecha, pasar a tocar las flechas de tecla derecha y tecla izquierda y luego soltar la tecla derecha provoca que se siga desplazando a la derecha y no que cambie la dirección a la izquierda) se mantiene la dirección inicial sin cambiarse a la nueva.

6. **Al reaparecer tras morir el jugador no aparece en el punto exacto de reaparición, sino un poco al sureste del punto de reaparición. Se sospecha que este error tiene que ver con el error 2.**

Solución temporal moviendo el punto de reaparición a una posición que no genera conflicto.

7. **Poner el nombre "collider" a una variable Collider2D de KinematicObject se sospecha que generaba algún error de convenio de nombre que hace que el acceso al Collider2D de la clase KinematicObject funcionase de forma inestable.**

Solucionado cambiando el nombre de la variable de "collider" a "mycollider".

8. **El Player al entrar en contacto con los enemigos no sufría daños, no se detecta correctamente la colisión entre el enemigo y el Player.**

El problema está en que el atributo "Is Trigger" del Player estaba a true. Poniéndolo a false se soluciona el problema, sin embargo es posible que este cambio genere problemas en el futuro.

9. **Cuando el Player salta sobre un enemigo se supone que tendría que salir revotando hacia arriba una pequeña distancia, sin embargo no lo hace.**

El impulso aplicado era tan pequeño que parecía no se aplicaba ningún impulso. Con aumentar la fuerza del impulso se ha solucionado el problema.

10. **Problemas para implementar la colisión con las paredes.**

Colisionar con las paredes supone dos problemas:

El primero es que como no se ha implementado una forma de impedir el movimiento en una dirección al actualizarse el movimiento, cuando se esta en una pared, el KinematicObject se sigue moviendo a pesar de estar chocando con una pared, eso sí, a menor velocidad.

El segundo es como decidir hacia donde impedir el movimiento del KinematicObject cuando colisiona con la pared.

Solucionado cambiando el objeto KinematicObject añadiéndole atributos que limiten el movimiento en una dirección. Cuando el KinematicObject collisiona con el muro se impide el movimiento en la dirección del movimiento.