Ejercicio 2.3.1 Jonatan Valle Corrales

1. **Probar los movimientos izquierda y derecha. Tiene que funcionar igual que antes. Si no es así, revisar el código a ver qué sucede.**

* Los movimientos todos me funcionan perfectamente.

1. **Probar el salto (Barra espaciadora). Veremos que el Player salta una vez y luego el juego se bloquea. ¿A qué se debe? Mirar el código y explicar razonadamente porque no va el salto y tampoco los desplazamientos horizontales. ( El objetivo es acabar de coger familiaridad a la estructura del código ).**

* Cuando aún no he saltado el monigote aún se me mueve de lado a lado pero una vez que hago el salto el monigote se me queda bloqueado y no se me mueve.

Después de tanto mirar y mirar porque debería ser, al final me di cuenta que debe de ser porque no se reinicia el estado del monigote. Es decir que a la hora de saltar entra en el caso de jump y salta, pero luego al aterrizar no reinicio los estados para poder moverme de nuevo o saltar.

Primeramente, para que el monigote salte tendremos que programar en el onStateChange el caso jump, con el siguiente código:

case PlayerStateController.playerStates.jump:

Debug.Log(newState);

if (playerHasLanded)

{

// jumpDirection determina si el salt es a la dreta, esquerra o vertical

float jumpDirection = 0.0f;

if (currentState == PlayerStateController.playerStates.left)

jumpDirection = -1.

else if (currentState == PlayerStateController.playerStates.right)

jumpDirection = 1.0f;

else jumpDirection = 0.0f;

// aplicar la força per fer el salt

GetComponent<Rigidbody2D>().AddForce(new Vector2(jumpDirection \* playerJumpForceHorizontal, playerJumpForceVertical));

//indicar que el Player esta saltant en l'aire

playerHasLanded = false;

}

Si te fijas podremos ver que al final del código el playerHasLanded pasa a ser false que es el aterrizaje.

Que pasa que si es falso en el código no puede volver a entrar en la condición del if(playerHasLanded) entonces el personaje se bloquea y no puede moverse ni saltar.

Para solucionar esto en el onStateCycle que es un método que se está ejecutando continuamente en el caso del jump llamamos al estado del aterrizaje para a si luego en caso del landing del onStateChange poder reiniciar el valor a true;

case PlayerStateController.playerStates.jump:

onStateChange(PlayerStateController.playerStates.landing);

break;

Ahora volvemos al onStateChange y reiniciamos el valor del playerHasLanded conforme el monigote ya a aterrizado y puede volver a moverse o saltar.

case PlayerStateController.playerStates.landing:

playerHasLanded = true;

break;