
THE GREATEST MME

Diego Matilla De La Cierva

Desarrollo Y Diseño De Videjuegos

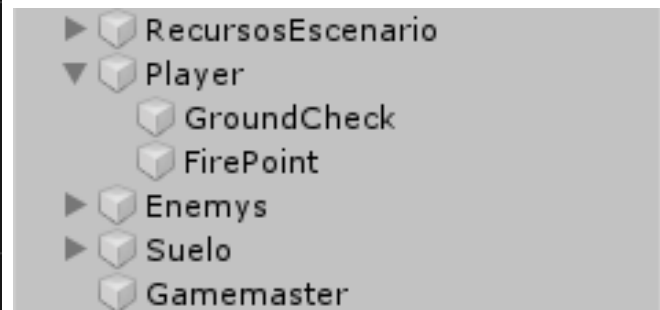
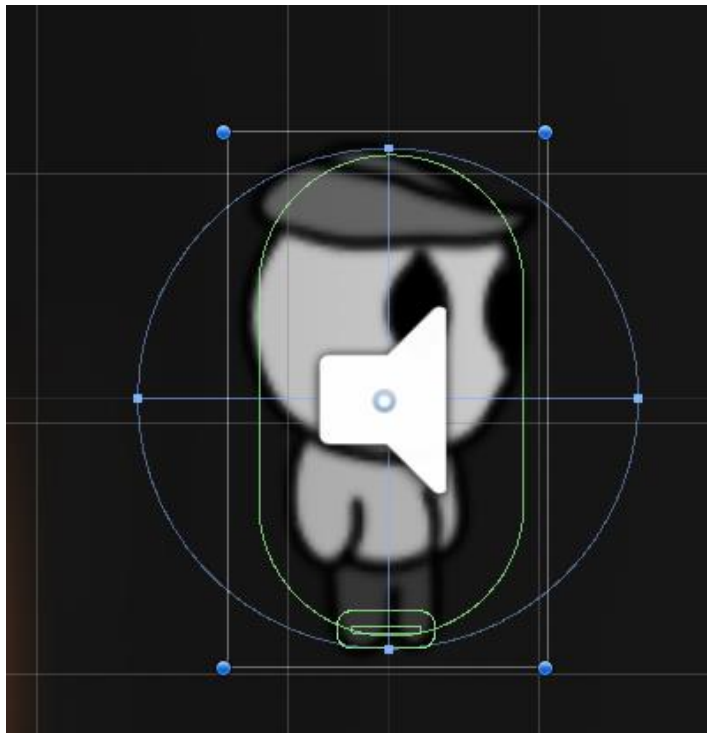
1.Introduccion, objetivos y motivación del Proyecto

Esta práctica consistirá en programar y realizar un pequeño juego de plataformas, este proyecto se centrará en la programación del personaje principal (todo menos sus animaciones), el cual contará con funciones como: Saltar, salto doble, hacer dash que se recarga cada vez que tocas el suelo, disparar y moverse. Este personaje esta dividido en tres partes: La parte principal del personaje, esta se ocupa de las funciones de la vida del personaje, su movimiento, gestión del salto... La parte GroundCheck que se ocupara de mandar la información de si el personaje toca el suelo y por ultimo la parte del punto del disparo del personaje.

Con este proyecto me propuse desde el inicio del curso a aprender a programar con la herramienta Unity. Ya que al iniciar este curso nunca había ejercido el roll de programador y mi conocimiento era muy básico me propuse a ser el lead del equipo de programación y programar cualquier tarea que propusiesen los diseñadores. Aparte, como equipo teníamos el objetivo de realizar un proyecto mas elaborado durante todo este año e intentar superarnos.

2 Descripción técnica de los componentes del proyecto.

Como he dicho antes, el proyecto se ha dividido en tres partes: La parte principal, el GroundCheck y el punto de disparo.



-Parte Principal:

Función:

Esta parte se ocupará de una gran parte de la gestión del personaje en sí. Este se ocupará de todo lo relacionado con el sistema de vida, la función del salto y el dash y el movimiento.



Descripción Técnica:

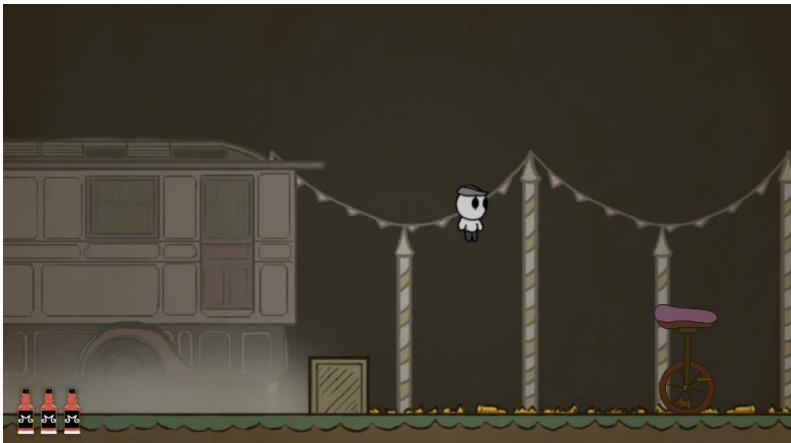
-Movimiento:

El movimiento del personaje es muy sencillo. Guardo en una variable float del Input del usuario y luego aplico este multiplicándolo por una velocidad a la velocidad del RigidBody.



-Salto Simple:

El salto simple del personaje funciona de tal manera que, si el usuario pulsa el botón de salto, el programa comprueba si el personaje toca el suelo (de esto se ocupará el GroundCheck) y si es esa la variable bool Jump se convierte en true, también me pone la variable bool DobleJump a true (en el doble salto la explicaré) y me instancia un GameObject vacío que pone un sonido de salto. Al ponerse la variable Jump a true, durante un rápido momento se aplica una velocidad del RigidBody en el eje y luego se vuelve a cambiar a false para repetir el proceso.



-Doble Salto:

Al activar el doble salto del salto simple, se da la posibilidad al jugador de que si no estás tocando el suelo, pero tienes la variable DobleJump true, el usuario podrá volver a saltar. Al producirse el segundo salto, se activará la variable Jump que se ocupará del salto, se desactiva la variable DobleJump para que el jugador no pueda volver a saltar y instanciara un GameObject que emitirá un sonido de doble salto. Aparte cuando el jugador toca el suelo se recarga la habilidad del doble salto permitiendo al jugador de si no salta y deja de tocar el suelo tenga ese doble salto.

-Dash:

En cuanto al sistema de dash es totalmente igual al sistema del salto. Solo que al apretar el botón del dash, aplica una gran velocidad hacia el lado en el que este mirando el personaje, aparte también se aplica un pequeño efecto de partículas y un pequeño sonido para dar un efecto más pulido.



-Sistema de cambio de orientación del personaje:

Al principio el personaje mira a la derecha y tiene una variable bool facingRight a true. Al presionar la tecla izquierda y el personaje mira a la derecha, se llama a una función llamada Flip que se encarga de cambiar la escala del personaje para que mire hacia el otro lado y me cambia la variable facingRight a false. Ocurriría lo mismo, pero al revés si al apretar la tecla derecha el personaje mira hacia la izquierda.

-Sistema de Vida:

El jugador parte con dos variables, una de la máxima vida que dispone este y otra que es la vida actual que posee, nada más iniciar el programa la vida actual se iguala con la vida máxima, esta vida se refleja en el juego mediante un canvas que cambiara el Sprite de una imagen según la vida del personaje. Luego el jugador si entra en colisiones con distintos enemigos se llamará a la función RecibeDaño, la cual restará una vida determinada al personaje, aparte se hará una animación de Shake en la cámara con una animación de un panel del canvas, un efecto de partículas y una instanciación de un Gameobject que emitirá sonido de daño para dar mas efecto al juego. Aparte el jugador podrá recuperar vida mediante pasar por distintos trigger los cuales sumaran vida al jugador, junto con un efecto de partículas y un sonido de cura. Si el jugador tiene mas vida de su máxima se iguala a esta y si se queda sin vida se llama a una función que elimina al jugador y se activa el menú de Game Over. Por último, si se recibe daño, el jugador dispone de un par de segundos de inmunidad mediante una variable bool Invencible que si esta activada al jugador no le afectara el daño.



-GroundCheck

Función:

Esta parte se ocupará de detectar de si el personaje está tocando el suelo mediante un collider que toque un tag determinado.

Descripción Técnica:

El collider del GroundCheck se encargará de modificar la variable bool IsGrounded del script de la parte principal del personaje del que es necesario para las funciones de salto y dash. Al tocar este collider los objetos con distintos tags este se pone true y al salir se pone en false. Aparte hay plataformas móviles que al colisionar el collider del GroundCheck con estas, el personaje se convierte en hijo de estas para poder moverse junto a estas, y al salirse se separan. Aparte hay un efecto que instancia un efecto de partículas cada vez que toca el suelo por primera vez para hacer un efecto de polvo en el suelo.

-Punto de Disparo:

Función:

Este Gameobject será el punto por el cual salgan las balas del personaje.

Descripción Técnica:

El personaje podrá efectuar el disparo mediante un Fire Rate mínimo que deberá de pasar. Cada vez que se apriete el botón de disparo se pondrá una animación de disparo y en esta se llamara a la función de disparar que efectuara el disparo mediante una instanciación de este objeto en el punto de disparo, aparte se instanciara un sonido de disparo. Estas balas al chocar con enemigos les quitaran vida (esto ya esta dentro de los scripts de los enemigos).

3. Diario de desarrollo y conclusiones: Obstáculos encontrados y enseñanzas aprendidas.

Empecé el proyecto con muy poca experiencia en Unity, por lo que al principio me costó bastante enterarme de cómo poder realizar el personaje. Al principio al no saber muy bien cómo hacerlo decidimos usar un código que encontramos de internet para que el personaje se moviera y saltara, pero este nos limitaba en muchos aspectos ya que al no entender el código de este no podíamos hacerle muchas modificaciones. Al ver que la anterior forma no iba a funcionar decidí crear mi propio código para el movimiento del personaje. Desarrollé en otro proyecto un personaje que fui mejorando y puse las funciones del dash y doble salto. Al realizar este pequeño proyecto entendí mucho mejor todo el sistema de Unity. Mientras tanto, mis compañeros se dedicaron al sistema de disparo y lo juntamos en este proyecto para tener nuestro personaje actual.

Durante el proyecto he tenido muchos problemas para entender Unity ya que era muy inexperto. Prácticamente he tenido problemas en cada parte del código ya que era la primera vez que me enfrentaba a desempeñar el rol de programador.

Durante este proyecto he aprendido una gran cantidad de cosas en cuanto a la programación de Unity. Me he centrado en intentar aprender lo máximo posible haciendo una gran cantidad de las tareas del equipo de programación y aparte antes de hacer cualquier cosa en el proyecto principal me hacía otro pequeño proyecto en el cual experimentaba otras distintas mecánicas y creaba pequeños juegos. Este trabajo ha sido muy útil para tener un conocimiento básico de Unity y para perderle el miedo a la programación.
