



VNiVERSIDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Práctica optativa del motor Unity

Jaime Gómez García

Introducción	3
Creación del mapa de prueba	3
Creación de físicas y animaciones del personaje	3
Creación de enemigos	6
Creación del mapa final	9
Sonidos	9
Items	10
Creación de la escena final	10
Bibliografía	11

Introducción

Para la realización de esta práctica he decidido hacer un juego 2D de plataformas utilizando diferentes assets de unity, he empezado a desarrollarlo desde el proyecto de la práctica del tema 9.

Creación del mapa de prueba

Primero de todo he decidido empezar realizando el mapa, ya que así voy a poder ir interactuando y comprobando las diferentes físicas que voy añadiendo. He usado Sunny Land de unity, un conocido asset para no tener que comenzar desde cero en la creación del mundo.

Para empezar lo primero que he hecho es crear un objeto 2D de tipo Tilemap, para poder dibujar el mapa, una vez hecho esto abro la ventana de Tile Palette y añado el entorno que viene en este asset a una nueva paleta. Hecho esto me pongo a dibujar un pequeño mapa de plataformas. Para que este mapa tenga colisiones añado el componente Tilemap Collider 2D al objeto Tilemap.

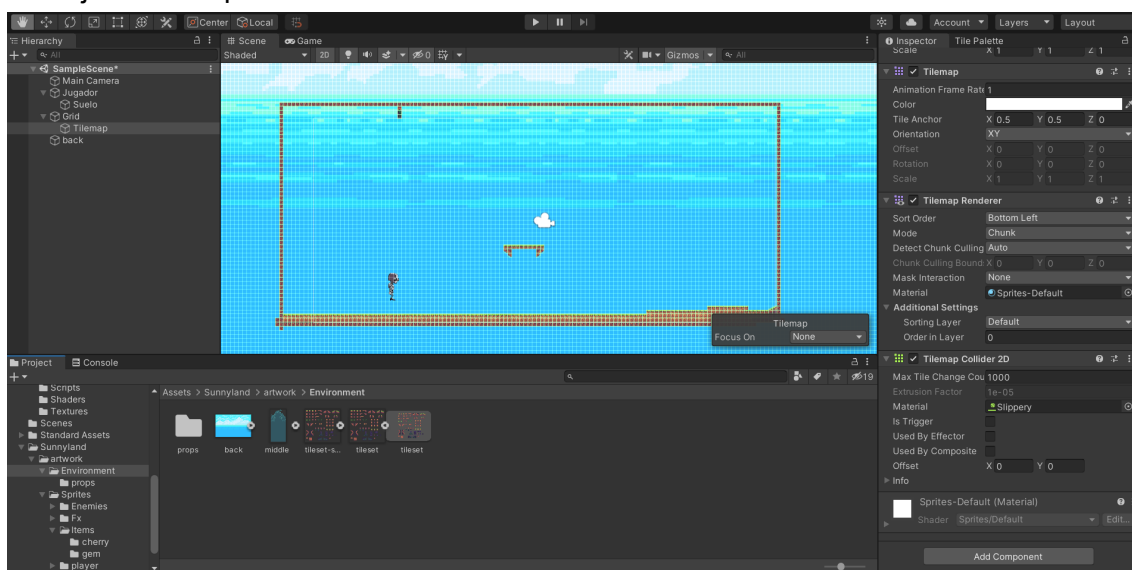


Imagen 1: Creación del mapa

Continuaré con la creación del mapa más adelante, ya que esto es suficiente para poder ir creando las físicas de nuestro personaje.

Creación de físicas y animaciones del personaje

Como quiero que el personaje esté relacionado con el entorno voy a utilizar el que viene en sunnyLand aunque me toca crearlo de nuevo ya que no tiene prefab. Lo primero que hacemos es crear un objeto 2D vacío llamado jugador, al cual vamos a añadir los

componentes “Box collider 2D” y “Rigidbody 2D”, es importante marcar la opción de “Freeze Rotation” dentro de “Rigidbody 2D”. A continuación creamos un script llamado Jugador, que va a ser el que vamos a utilizar para capturar los movimientos. Para empezar programamos el código necesario para que se pueda mover de izquierda a derecha.

Una vez comprobado que funciona bien vamos a crear la opción de saltar pulsando el espacio, para ello vamos a crear un objeto vacío llamado suelo dentro del jugador, a este objeto le añadimos el componente “Box Collider 2D” y lo colocamos cuidadosamente debajo de nuestro personaje, ya que este objeto va a ser el encargado de indicarnos si nuestro jugador está teniendo una colisión con el suelo. Por último generamos un script llamado suelo, se lo asignamos a este objeto y en el script de Jugador programamos el movimiento de salto al pulsar el espacio. Después de haber hecho todo esto nuestro personaje ya es capaz de saltar y moverse a los lados.

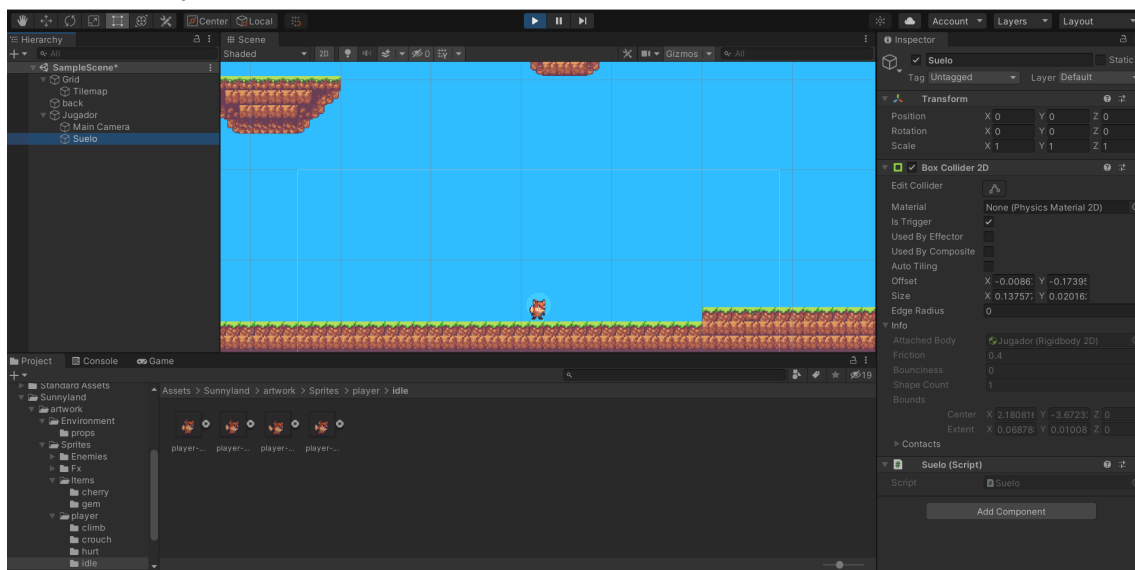


Imagen 2: Personaje principal con movimiento

A continuación vamos a crear los sprites de nuestro personaje para ello abrimos la ventana de animation y creamos una nueva animación, una vez hecho eso arrastramos los sprites a la ventana de animación y empezamos la grabación.

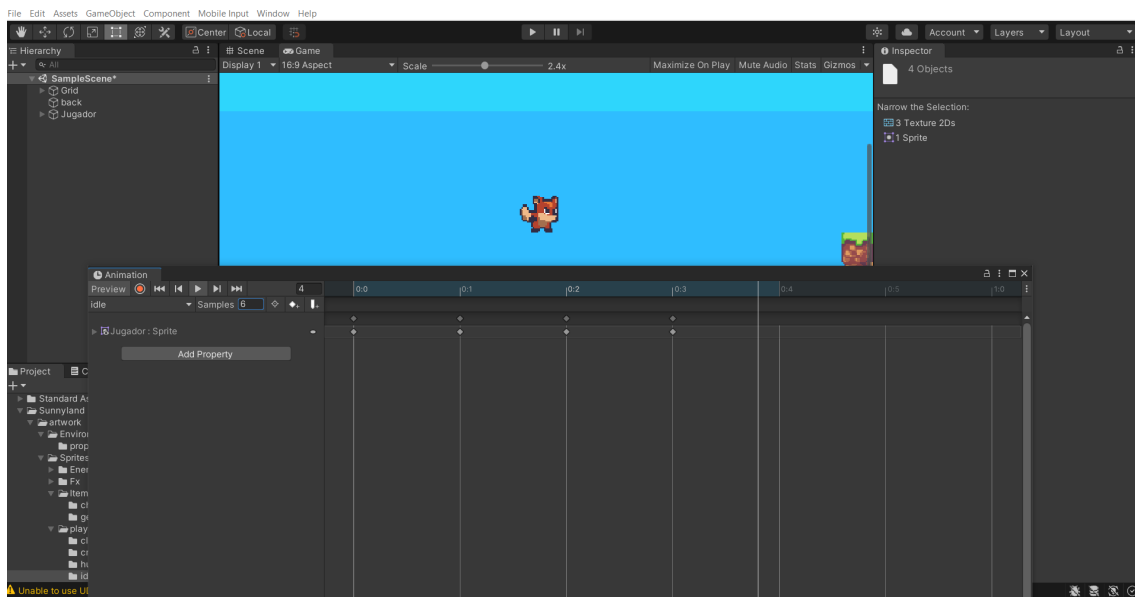


Imagen 3: Creación de sprites

Una vez tenemos creado la animación del personaje cuando está parado vamos a crear la de correr, para ello hacemos lo mismo que hemos hecho antes. Después vamos a configurar la animación para que funcione cuando pulsamos “a” y “d”, para ello dentro de la ventana de animator vemos como se nos ha creado una nueva animación llamada Run, lo que tenemos que hacer es crear dos transiciones desde la animación de parado y salto, también tenemos que crear un nuevo parámetro booleano que nos indicará a través de código si estamos corriendo o no. Por último configuramos las condiciones de las transacciones para que se activen cuando el parámetro “Run” esté a true o false.

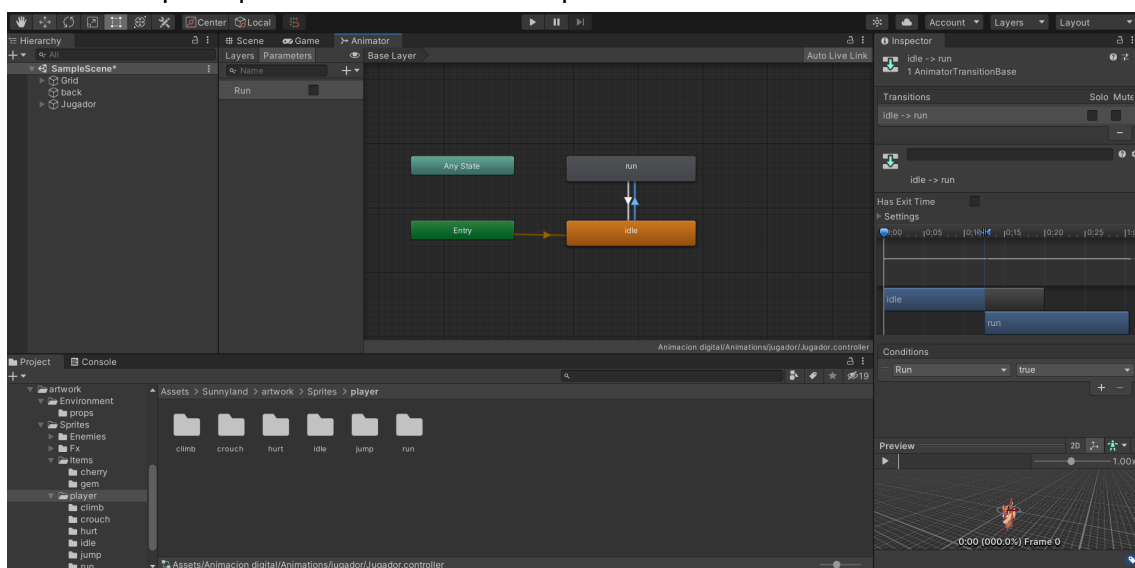


Imagen 4: Creación de la animación de correr

A continuación dentro del código del Jugador vamos a configurar el parámetro “Run” para que se active o desactive, también vamos a cambiar el “spriteRenderer.flipX” para que el personaje se gire sobre sí mismo.

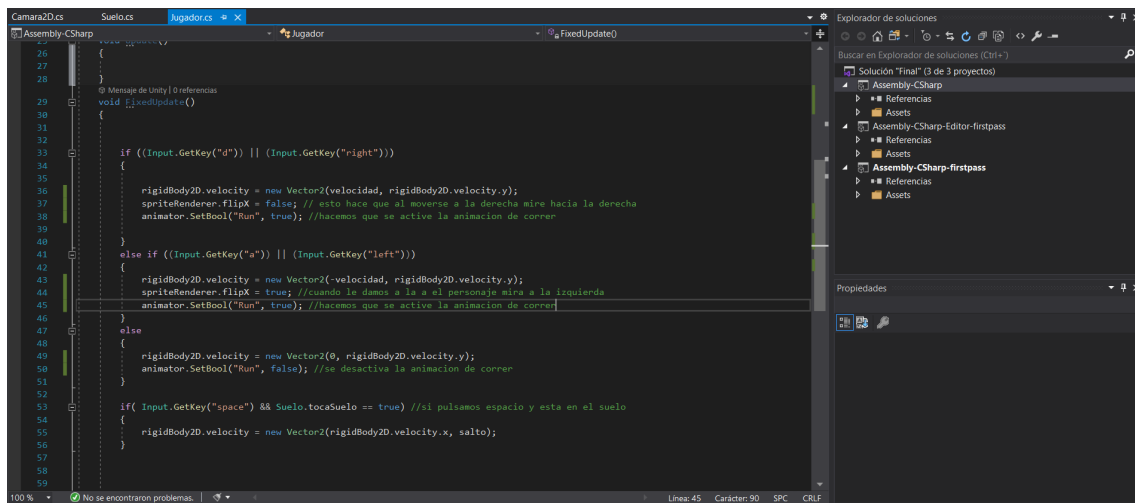


Imagen 5: Código del movimiento del personaje

Por último vamos a hacer lo mismo para el salto de modo que el resultado de las animaciones quede tal como se ve en la imagen 6.

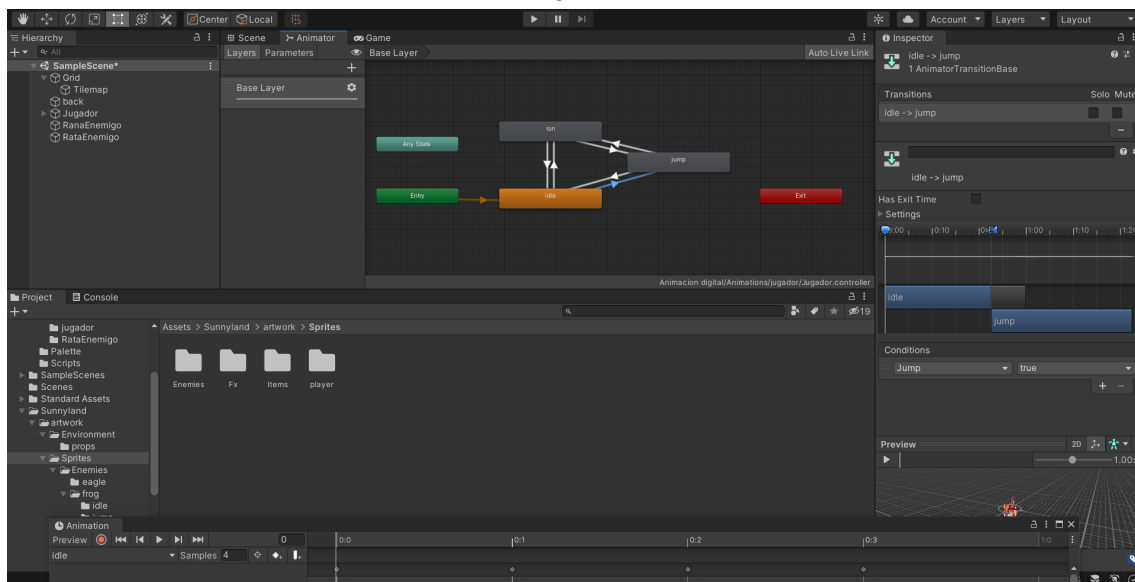


Imagen 6: Animaciones completas del jugador.

Después de haber hecho todo esto decidí crear la opción de que el jugador tenga doble salto para tener mayor jugabilidad, para hacer esto dentro del script "Jugador" que es el encargado del movimiento creo mediante el código para que una vez que el personaje está en el aire pueda saltar de nuevo.

Creación de enemigos

Para darle más jugabilidad vamos a crear un enemigo con el que el jugador pueda interactuar, para ello lo primero que hacemos es crear el objeto "RataEnemigoPadre", este objeto va a ser el padre de los demás objetos de forma que resulte más fácil manipularlo.

Una vez creado le añadimos las animaciones al igual que hicimos con el personaje, en este caso este enemigo solo tiene una, siendo esta la de correr.

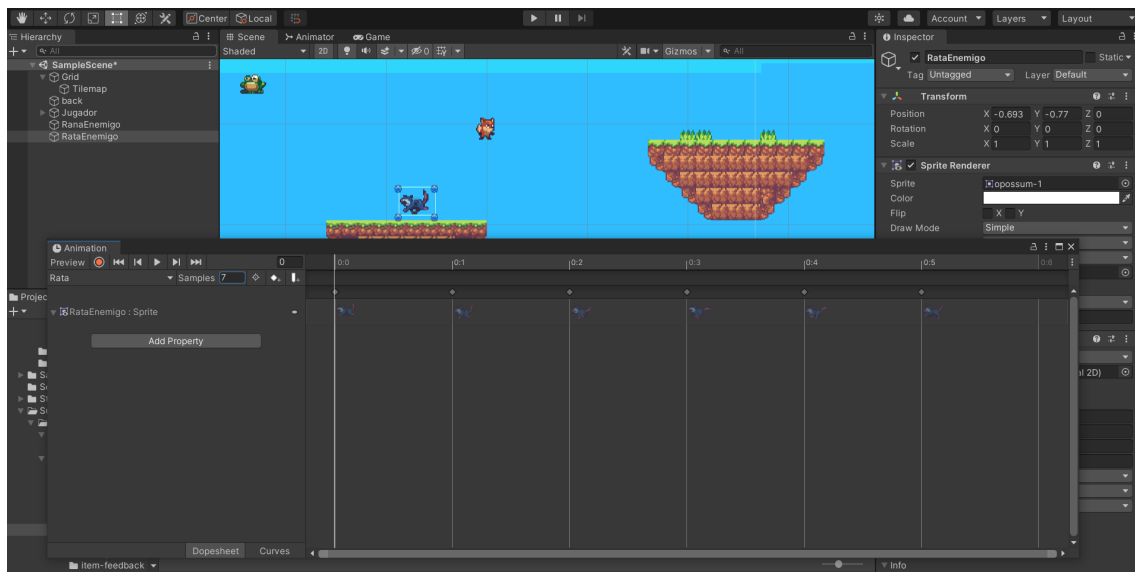


Imagen 7: Enemigo Rata.

A continuación vamos a darle movimiento al enemigo, para ello creamos dos objetos vacíos dentro de “RataEnemigoPadre”, uno de estos objetos vacíos va a estar dentro del propio enemigo, y el otro va a estar en la posición a la que queremos que llegue andando.

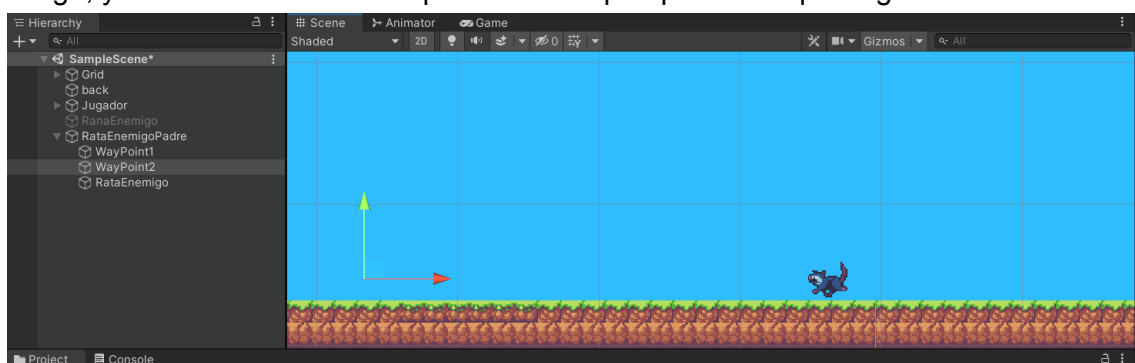


Imagen 8: Objeto vacío 2

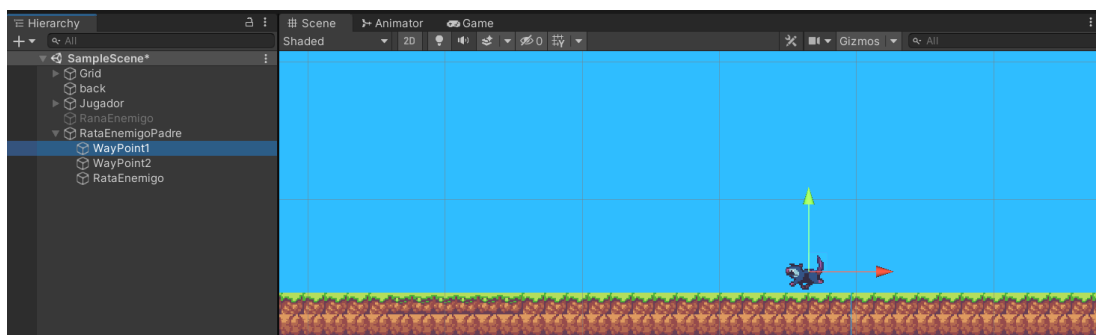


Imagen 9: Objeto vacío 1

Después mediante código hacemos que el enemigo se esté moviendo continuamente de un lado a otro. Para ello he creado un array llamado “movimientos” el cual va a estar

compuesto por los objetos vacíos que hemos creado anteriormente de forma que vamos a estar comprobando si el enemigo ha llegado a esa posición y en caso de que haya llegado pasará a moverse a la siguiente, haciendo que podamos crear diferentes movimientos para el enemigo. También he creado una subrutina que estará comprobando si el enemigo está caminando hacia la izquierda o hacia la derecha, esto para poder girarlo según su dirección.

Por otro lado para que el enemigo sea capaz de matar y de ser matado he creado dos “box colliders” uno de ellos en la parte superior haciendo que si le pisas lo mates y otro por debajo pero que sobresalga un poco, así si el enemigo te ataca por el lado te matará.

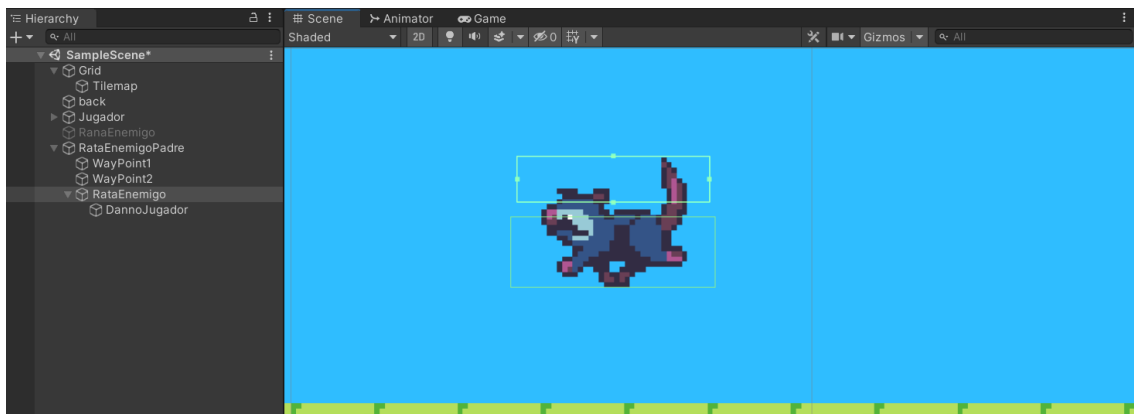


Imagen 10: Visualización de los dos “box colliders”.

Para que cuando matemos el enemigo no desaparezca sin más he le asociamos la animación de muerte cuando el jugador tiene una colisión con el “box collider” de la parte superior, al acabar la animación el enemigo desaparecerá, en cambio si el enemigo te ataca a ti se reproduce la animación de daño en el jugador morirá.

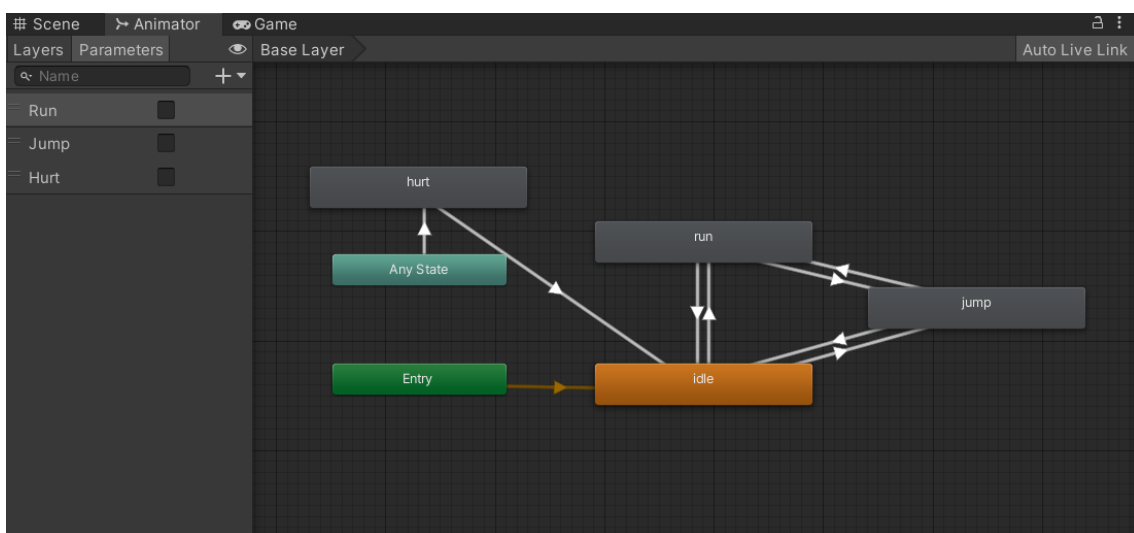


Imagen 11: Animaciones finales del jugador

Por último nos queda hacer que el jugador respawne cuando muera, para hacerlo creamos un script llamado respawn asociado al jugador, dentro de este script iniciamos los valores de x e y con los valores que tiene el jugador al comienzo de la escena y creamos una función

“MurioJugador” que sea llamada cuando se produce la colisión con el box collider del enemigo. He puesto un delay de 0.3 segundos antes de respawnear para que se pueda ver la animación de daño del jugador.

Creación del mapa final

Dado que ya tenemos más o menos las físicas del juego va siendo hora de finalizar el mapa para poder ir juntando todo lo que hemos creado. Como el juego es de plataformas la finalidad va a ser ir saltando de un lado a otro recogiendo frutas y esquivando enemigos.

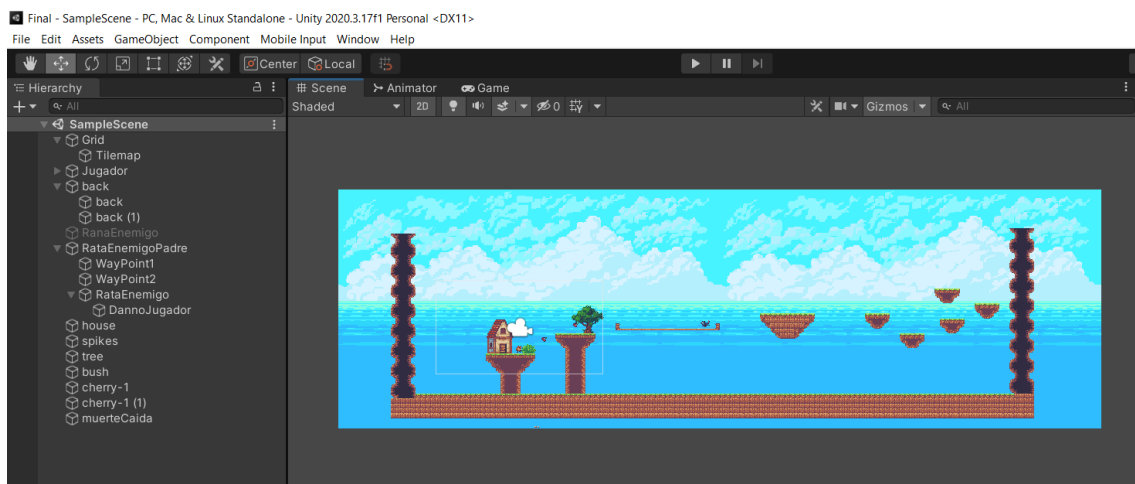


Imagen 12: Mapa creado

Como vemos en la imagen 11 el mapa ya está finalizado, siendo el propósito del juego coger las frutas para ganar, estas frutas las configuraremos más adelante. He añadido un box collider en todo el suelo llamado “muerteCaída” que hará que cuando nos choquemos con él moriremos, he usado el mismo script de muerte por los enemigos. Intenté que el fondo se moviera pero me daba muchos problemas de optimización por lo que he decidido que esté quieto.

Sonidos

Para poder meter sonidos he creado un objeto vacío para cada sonido, uno para el sonido de fondo, otro para cuando el personaje muere, uno cuando salta y otro para cuando el enemigo muere. Para reproducir los sonidos es muy sencillo, para el sonido de fondo simplemente se crea un objeto vacío y se le añade el componente de “AudioSource”, después se le añade el audio que queremos y la opción de “Play on awake”. Para los demás sonidos también hacemos lo mismo pero sin la opción de “Play on awake” ya que queremos que se reproduzca en un determinado momento, para ello en el código creamos el objeto “AudioSource” y cuando se produce la acción en donde queremos que suene el audio llamamos al método Play().

Items

A continuación voy a explicar cómo he metido los items, en este caso he usado el item de cherry de sunnyLand. Lo primero que hacemos es crear el objeto cherry y le añadimos la animación, después creamos otro objeto para la animación de recoger el item y por último hacemos que este ultimo objeto sea hijo de cherry, como se ve en la imagen 13.

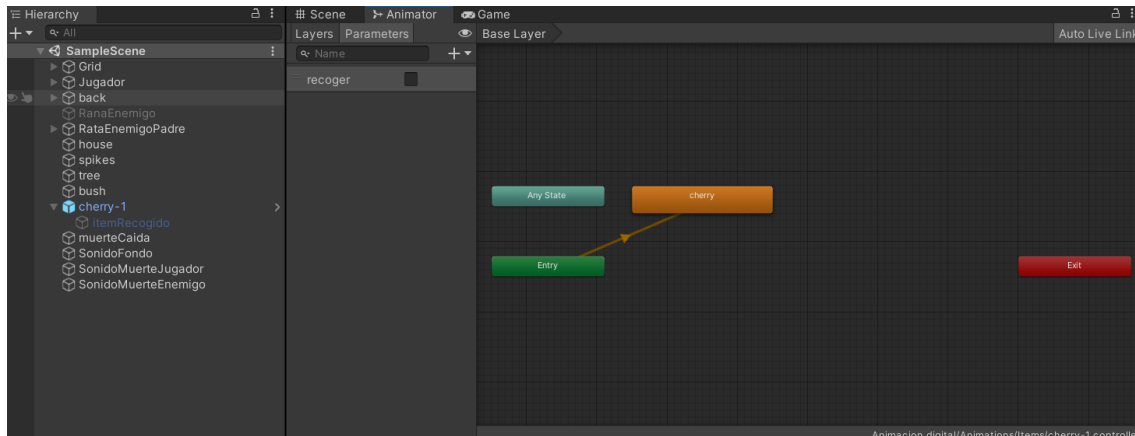


Imagen 13: Item y su hijo

A continuación creamos un script llamado “itemRecogido” en el cual vamos a hacer que cuando haya una colisión entre el jugador y el cherry se desactive el “spriteRenderer” del cherry y el objeto de la animación de recoger se active, por último se destruye el objeto.

Una vez tenemos hecho esto vamos a crear un objeto que será el padre de todos los items, en este caso lo hemos llamado “itemGlobal”, dentro de este objeto metemos todos los cherrys que queramos por todo el mapa y creamos un script llamado “ItemGlobal” que estará preguntando continuamente si le quedan hijos, en caso de que no tenga hijos se pasará a la siguiente escena.

Creación de la escena final

Para crear otra escena primero de todo es importante hacer prefabs a todos los objetos que queramos seguir utilizando, a continuación dentro de la carpeta Scenes creamos otra escena. Después de esto es exactamente igual que lo que hemos hecho anteriormente solo que al tener todo ya creado es mucho más rápido.

Para esta segunda escena he creado otro mapa con un ambiente diferente como podemos ver en la siguiente imagen.

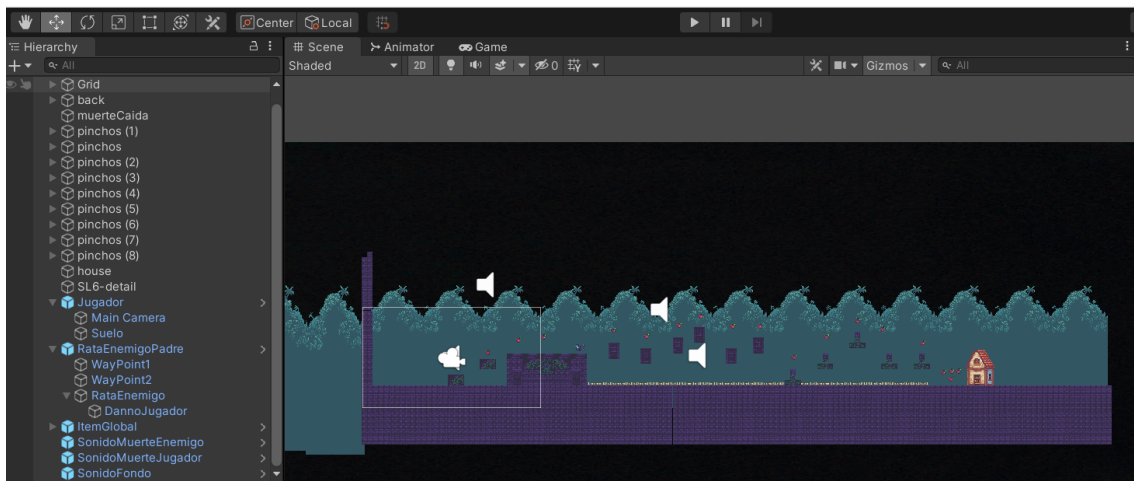


Imagen 14: Segunda escena

Para poder cambiar de escena dentro de “Build settings” añado la escena que acabo de crear y en el script de “ItemGlobal” añado una línea que cambia a la siguiente escena.

Por último nos queda hacer que si no hay más escenas nos salga un texto diciendo que hemos completado el juego, para hacerlo creamos un objeto de tipo texto y lo desactivamos, añadimos el texto que queremos que salte cuando recojamos todas las frutas, en el script “ItemGlobal” añadimos un objeto tipo texto, por último comprobamos que la escena es la última escena y si es así activamos el objeto.

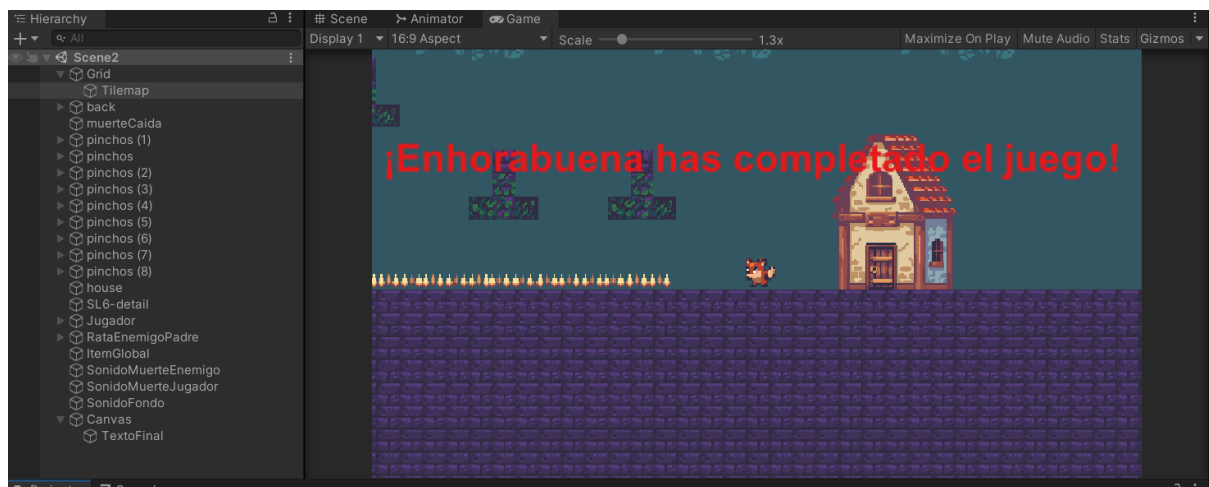


Imagen 15: Mensaje final

Bibliografía

Para hacer muchas de las cosas que he implementado me he basado en tutoriales de youtube como:

<https://www.youtube.com/playlist?list=PLIpYDjUMtti5bWJ1Ugystr-vUg8B5EuP>

https://www.youtube.com/playlist?list=PLNEAWvYbJJ9kZpalg2RfzAc_KZixBgchT

Como hacer un wait: <https://docs.unity3d.com/ScriptReference/WaitForSeconds.html>

Asset de unity: <https://assetstore.unity.com/packages/2d/characters/sunny-land-103349>

Audios: <https://freesound.org/>