

MANOLITO EL BÁRBARO

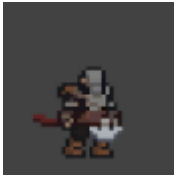
Bienvenido a Manolito el Bárbaro, un emocionante juego de plataformas lleno de desafíos y aventuras. Los jugadores se meterán en el papel de Manolito, un bárbaro de la edad medieval aterrizado por los fantasmas de sus enemigos muertos.

Se trata de sortear o matar a los fantasmas, que son los enemigos y se ponen violentos cuando Manolito se acerca. Para conseguir acabar con estos Manolito dispone de su hacha mágica, esta es mágica porque aunque la lance siempre va a estar ahí. Esta después de un segundo y media desaparece.

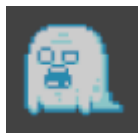
Esta es su hacha



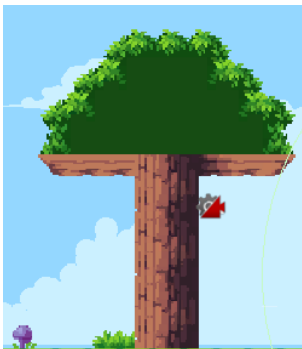
Este es Manolito



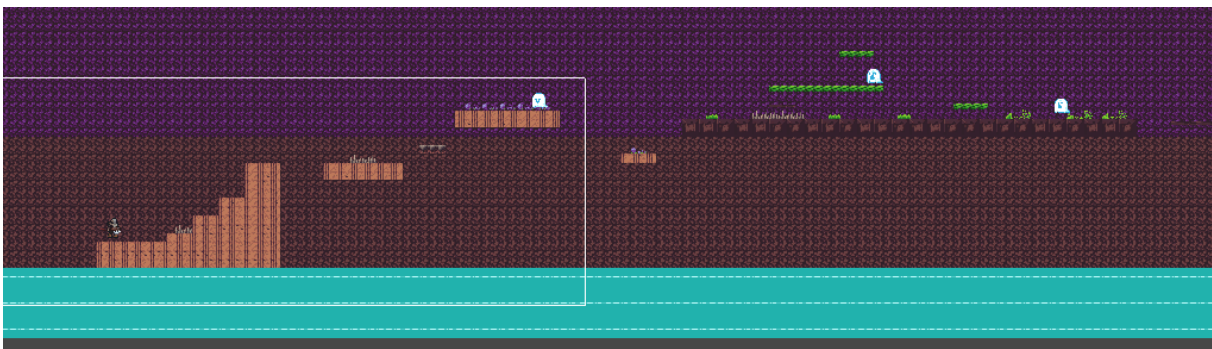
Este es el fantasma



Mi primera escena es tipo bosque, el fondo es de nubes, hay elementos como flores y árboles. Además, hay elementos como vallas y pinchos.



La segunda escena es una especie de cueva, aquí hay más pinchos y plataformas, aquí hay más fantasmas, pinchos y es algo más complicado.



Y la tercera, es una escena para hacer saber al jugador que ha ganado.

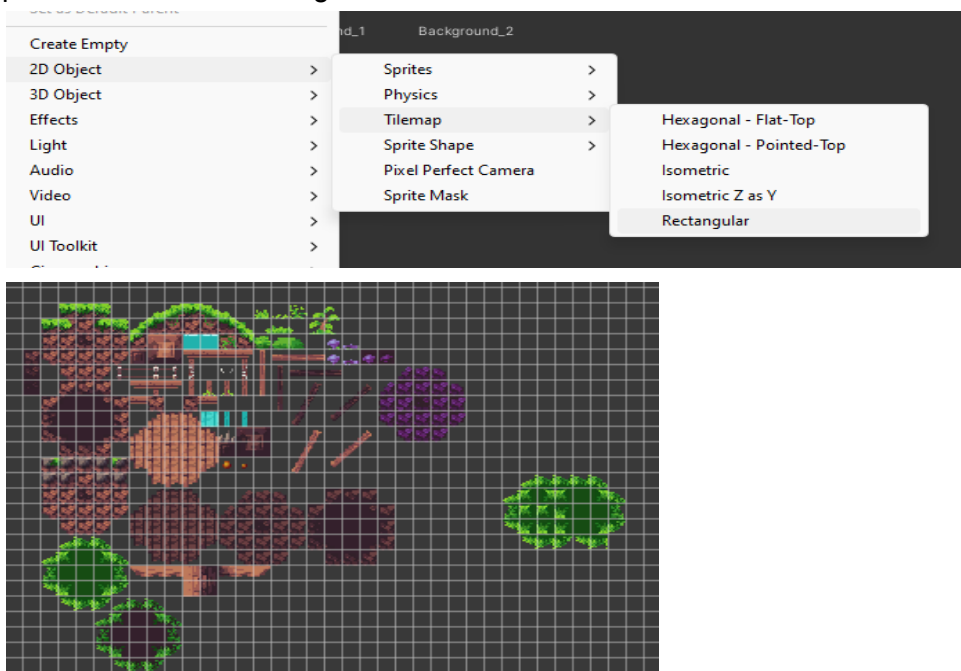


El personaje puede realizar tres movimientos, hacia adelante, o la derecha (según lo quieras ver) con la tecla “d”, para atrás o la izquierda con la letra “a” y puede saltar, con la tecla de espacio, en el aire también puede cambiar de dirección pero no puede disparar si salta. En cada nivel tiene una potencia diferente de salto, la velocidad del personaje sí es la misma.

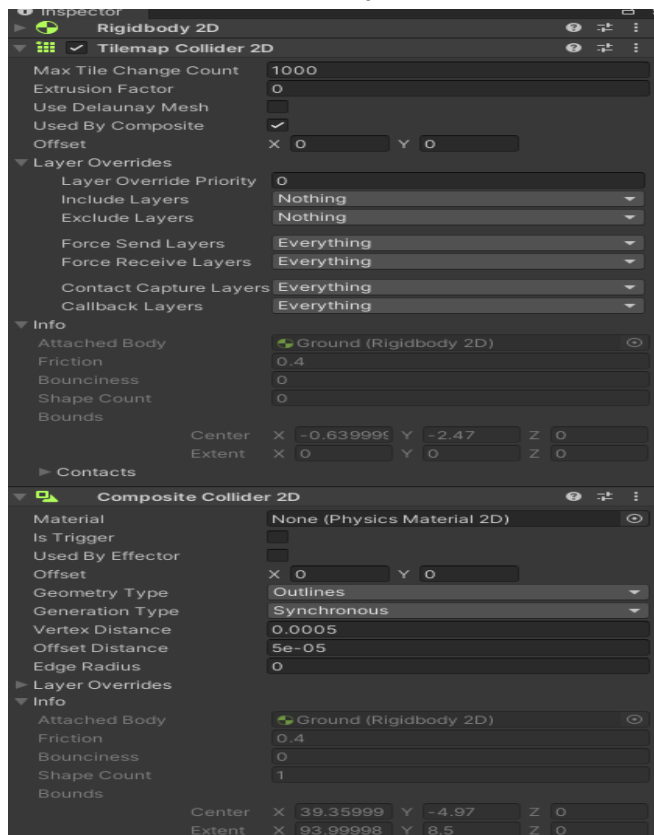
Los peligros son los pinchos, cualquier mínimo roce hace que el personaje muera, igual que tocar al fantasma, este solo se mueve de izquierda a derecha, no puede ni saltar ni cambiar de dirección (en el código está puesta una solución que pensé pero que no funcionó).

Para la creación del personaje me he descargado el asset de la página itch.io, a este no le he hecho slice como he hecho con los sprites del bosque.

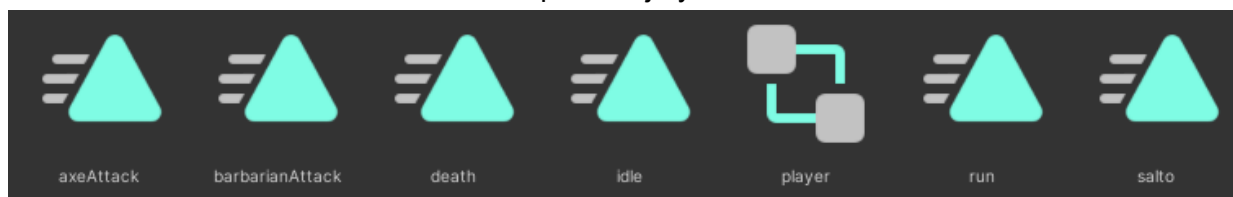
Así se ve mi Tilepalette, el sprite está a 16 píxeles por unidad. Hice slice y quedó perfectamente. Hice un grid de esta forma

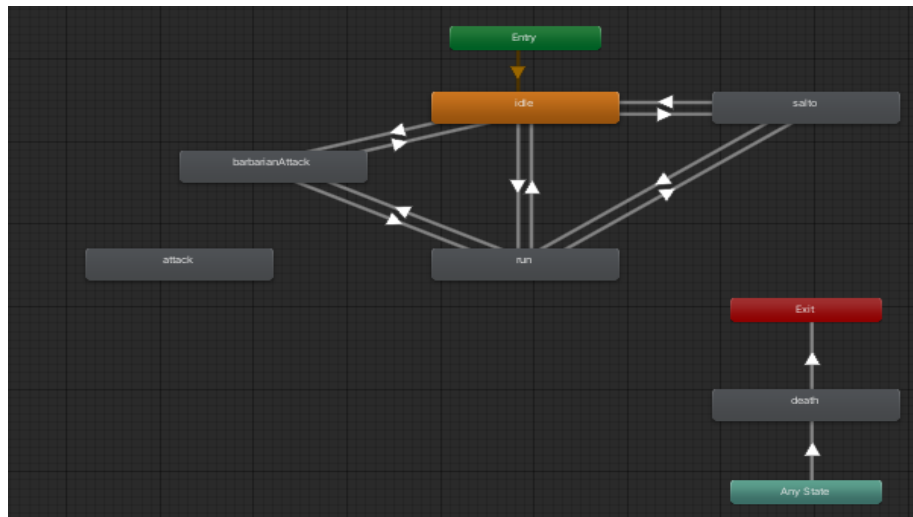


En los videos dice de meter un box Collider para que no se caiga pero hay una forma mejor, que es esta, un Tilemap Collider 2D y Composite Collider 2D. Para la caída al vacío si que he usado un BoxCollider del tamaño del escenario que sea trigger para que al tocar al capsule collider del personaje active la animación de “death”

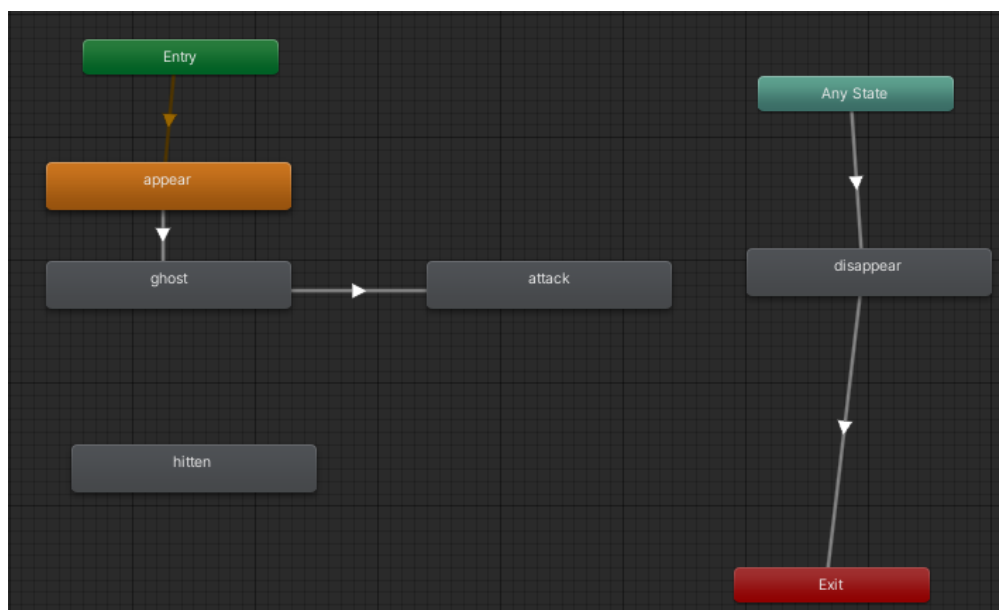
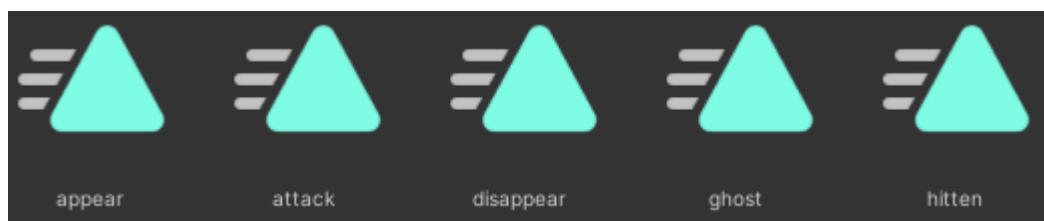


Cada personaje tiene una animación de idle, de atacar y de correr. El fantasma también tiene una animación de aparecer y desaparecer, en este personaje su animación de atacar es como la de idle pero aumentando los samples para que parezca que está como más loco. Estas son las animaciones del personaje y sus estados





Y estas son las del fantasma



Aparte de esto los personajes tienen un box collider que son trigger para que al ser tocados estos mueran. El fantasma también tiene un área de ataque que al entrar en contacto con Manolito el bárbaro hace que su animación de ataque empiece y también que este se empiece a mover. Todos tienen un rigidBody para que se apliquen las leyes de la física en ellos, en el caso de Manolito es dynamic, en el fantasma es kinematic y en el caso del suelo es static. En mi caso no he le he puesto materiales en específico, imagino que esto hubiera ayudado a cosas como la fricción, ya que este se queda pegado a los laterales de las plataformas.

Las hachas también tienen un box collider que cuando tocan al fantasma hacen que este muera y se desaparezca, también tienen su animación al lanzarse y un tiempo para que el objeto se destruya.

Para el código he ido empleando el código de los videos y añadiendo cosas más, como por ejemplo

```
private void OnTriggerEnter2D(Collider2D other)
{
    if (other.gameObject.CompareTag("enemy") || other.gameObject.CompareTag("Respawn") || other.gameObject.CompareTag("colisionInferior") || other.gameObject.CompareTag("pinchos"))
    {
        animPlayer.SetTrigger("death");
    }
}
```

Hago que cuando cualquiera de estos elementos con esas etiquetas toquen al jugador harán que este muera y haga respawn a su sitio fijado en la escena. Para el respawn he hecho una función que haga que cargue la escena, pasándole el nombre de la escena activa con el scene manager.

```
//función para que al morir el personaje, este vuelva a la misma escena
Referencias
public void respawn()
{
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}
```

El resto del código está comentado en el proyecto. En los vídeos se incluía música para los efectos de sonido con el Audio Source pero yo he tenido problemas con esto. Cuando metía dos Audio Source y le metía en cada uno un clip de música diferente me seguía sonando los dos movimientos igual. Para arreglarlo, he buscado y lo he hecho desde el código. Creando una variable AudioSource y metiendolo en la función Jump y después asignándole el clip desde el inspector. Para la música de fondo en la Main Camera le meto un Audio Source y le pongo "Play on Awake" y "Loop", pongo una canción por escena.

```
Yinicial = transform.position.y;
AudioSource audioSource = saltoPlayer.GetComponent();
//AudioSource audioSource = audioSource1;

public void Jump()
{
    if (checkGround.isGrounded)
    {
        rb.velocity = new Vector2(rb.velocity.x, jumpingPower);
        soundEffectJump.Play();
    }
}
```

Mi elemento win es una tercera escena que pone YOU WIN con plataformas. Al igual que en la segunda escena he usado el mismo método para cambiar de escena, un box collider en una puerta.

Para las cámaras le he puesto el cine Machine y que siga el jugador y le mire a él, también le he quitado el damping para que no haga ese efecto, que a mi personalmente no me gusta. y la he centrado a mi gusto.