

Memoria Pacman

Integrantes del grupo:

Juan David Lectamo Caicedo
Johann Andrei Ocampo Torres

Universidad de Huelva
Abril/18/2022

Inicio del proyecto:

- **Preconfiguración:**

Antes de iniciar el proyecto se realizó una preconfiguración en la cual buscábamos cada una de las entidades que comprenden el pacman y una vez encontradas se elaboró una matriz de colisiones la cual establecería las bases donde programaremos el juego.

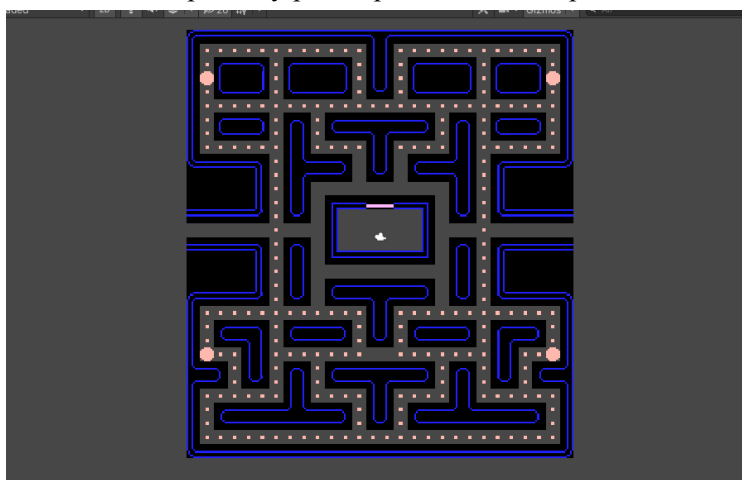
- **Sprites:**

Una vez configurada la matriz de colisiones, el primer objetivo que se buscaba al iniciar el proyecto fueron los sprites para el diseño del juego, estos se consiguieron sin problemas de derechos de autor. Las entidades principales las cuales requerían sprites eran: Las paredes del laberinto, el Pacman, fantasmas(Blinky, Inky,Pinky, y Clyde) los pellets(puntos de ingiere el pacman), power pellet(punto que al obtenerlo brinda al pacman la capacidad de eliminar los fantasmas) y sprites complementarios que permiten realizar las animaciones de cada uno de las entidades(por ejemplos que los fantasmas sean azules y blancos después de ingerir una power pellet o que la posición de los ojos de los fantasmas cambien según el tipo de dirección que tomen.

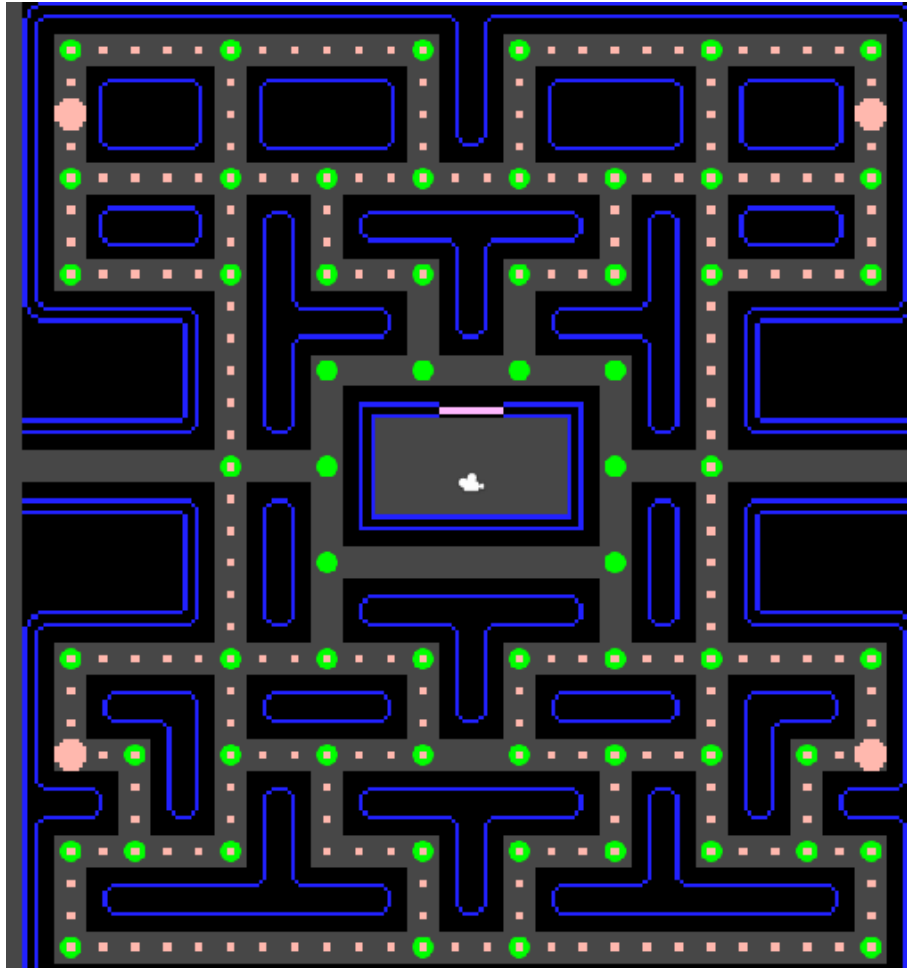


- **Escena de laberinto:**

Ya teniendo todos los sprites, el primer paso es organizar la escena del laberinto y la ubicación de cada una de las pellets y power pellets. Una vez posicionado todo este es el resultado.



Cuando el mapa fue finalizado se crearon objetos llamados nodos y que fueron posicionados de manera estratégica a lo largo del mapa. La finalidad de estos nodos es establecer puntos de guía para el movimiento de los fantasmas por el mapa. cada nodo es posicionado para definir las posibles direcciones que puede tomar un fantasma a la hora de seguir a pacman; esta es la base que permite que la inteligencia artificial de los enemigos funcione.



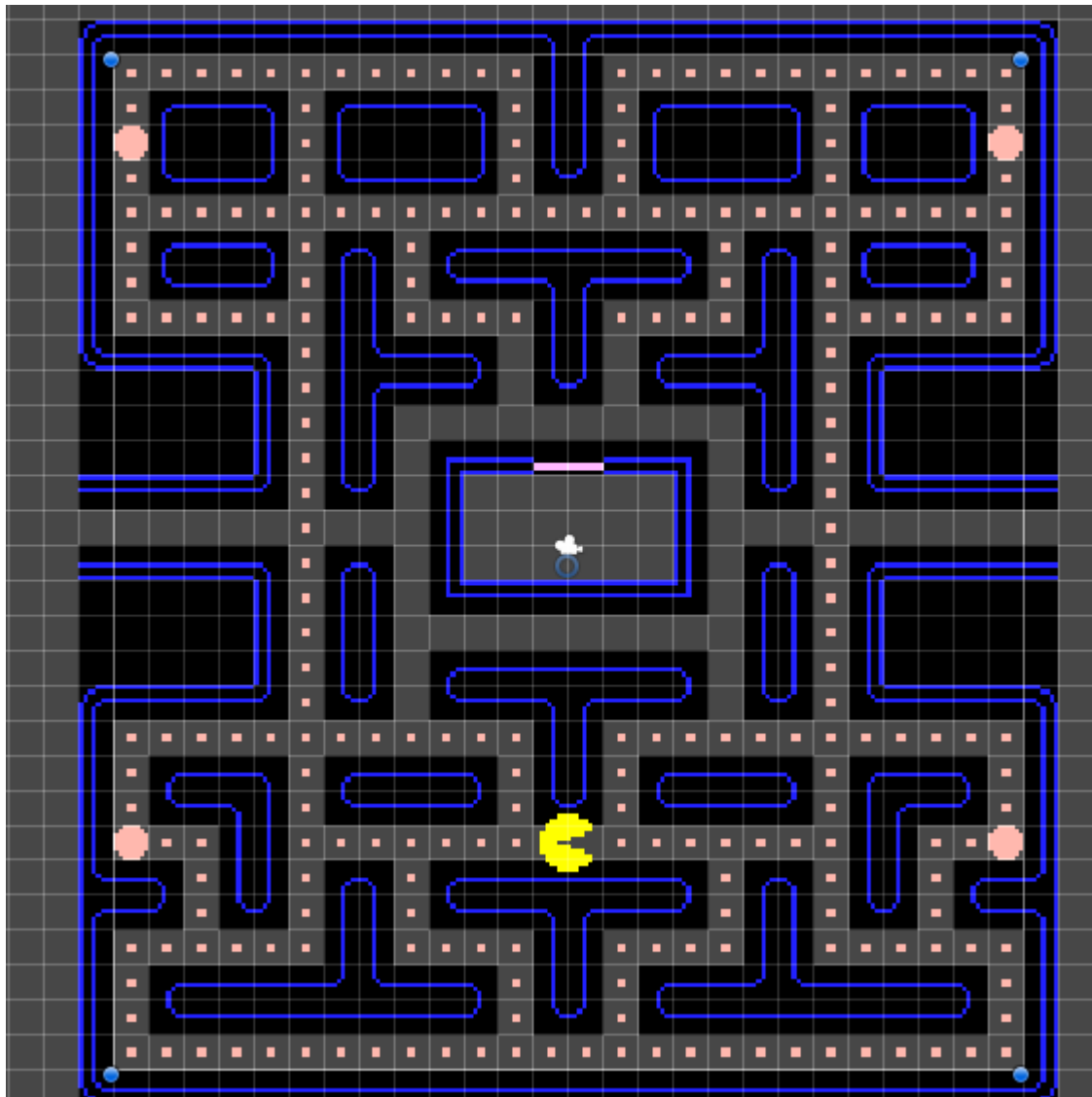
- **GameManager:**

El juego debe tener un punto central que sea el encargado de manejar la parte general del juego, tal y como es la disminución de las vidas en el momento que un fantasma se tope con pacman; conteo del puntaje a la hora de ingerir las pellets, power pellets o fantasmas en modo vulnerable; reiniciar los niveles una vez al jugador se le acaben las vidas o una vez ingiera todas las pellets; etc. Por esto fue creado inicialmente el script llamado GameManager y desde este cada uno de estos aspectos y más es ejecutado.

- **Animaciones, Pac Man y Ghost:**

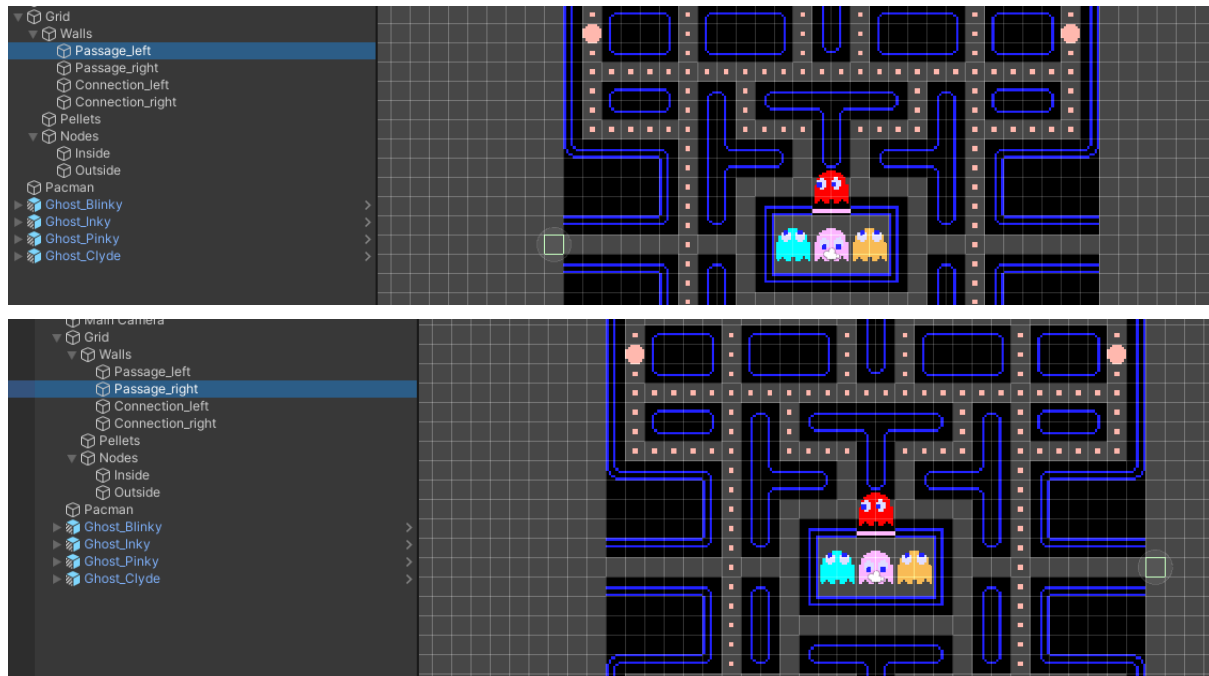
Para concentrarnos en aspectos funcionales de los elementos del juego como el pacman y los fantasmas lo primero que se realizó fue un script llamado AnimatedSprite y es el que permite se ejecute una secuencia de sprites de modo que se realice animaciones como el cerrar y abrir de la boca del pacman, el movimiento de la parte inferior de los fantasmas, etc. Una vez establecida la funcionalidad de la animación de los elementos se crea el script del pacman en el cual definimos cada una de las acciones y movimientos que puede tomar el jugador, al ser este un juego 2D las únicas opciones son arriba, abajo, izquierda y derecha.

También se crea el Script llamado Ghost que va a definir aspectos generales de cada uno de los fantasmas.



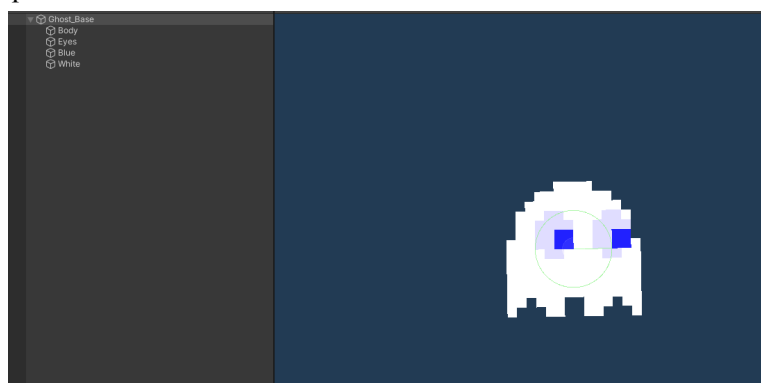
- **Pasillos:**

Una funcionalidad que debe ser implementada en el mapa es el pasillo que conecta cada extremo del mapa, y para esto se crea un script llamado Passage para hacer que cuando un objeto del juego pase por ahí, sea “teletransportado” al extremo contrario. Esto se logra mediante la implementación de objetos vacíos en los extremos los cuales sirven de referencia en el eje para trasladar un objeto de un lado al otro.

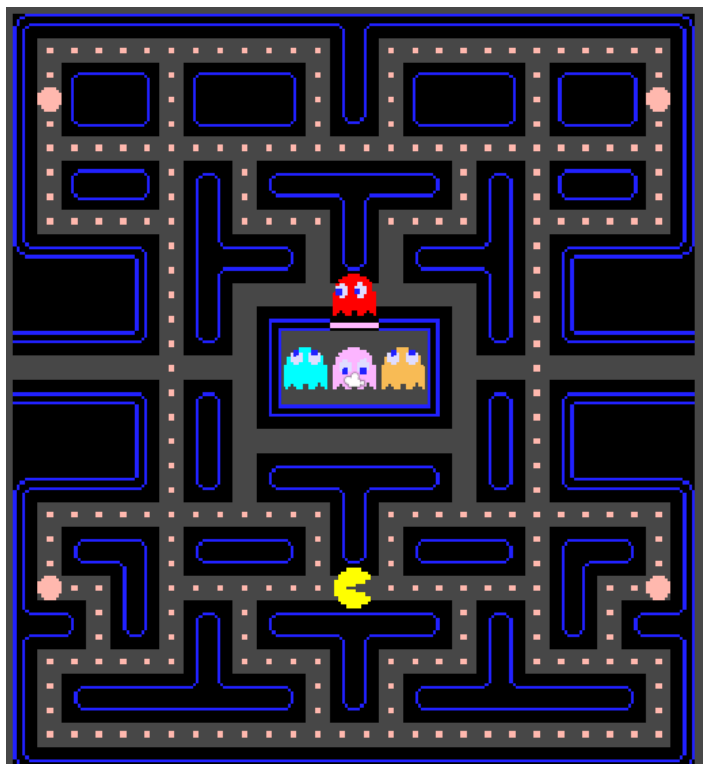
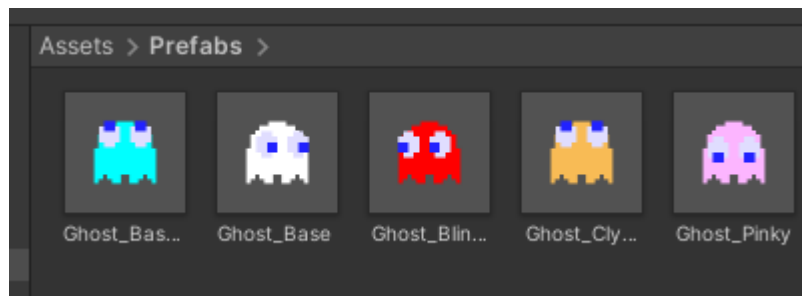


- **Configuración de fantasmas:**

El siguiente paso en la lista es la configuración de los fantasmas, en total hay 4 fantasmas cada uno llamado: Blinky(fantasma rojo), Inky(fantasma azul), Pinky(fantasma rosa) y Clyde(fantasma naranja). ya que cada uno de estos solo se diferencia por el color es más conveniente si se crea un prefab asset general del cual cada uno de los fantasmas derivará, de esta manera cada uno de los aspectos que comparten entre sí será igual exceptuando los específicos como el color; de esta manera es más dinámica la configuración del concepto general de fantasma. Esta base para el fantasma fue llamada Ghost_Base la cual estaba compuesta de subitems necesarios para configuraciones posteriores como el cambio de color por la powerpellet o el movimiento de los ojos segun la direccion que tome un fantasma.



Con el Ghost_Base creamos cada una de las variantes de los fantasmas y las posicionamos en el mapa según el orden de salida.



- **Inteligencia artificial de enemigos:**

Ahora que se poseen los elementos fantasmas viene la parte más compleja del proyecto la cual es la creación de la inteligencia artificial de los enemigos. Para esto lo primero que se hizo fue crear un script llamado GhostBehaviour el cual sería la clase abstracta donde derivan los demás comportamientos de los fantasmas y los cuales son GhostScatter(para que el fantasma deambule), GhostChase(para que el fantasma persiga directamente al pacman para eliminarlo), GhostHome(cuando el fantasma está en la base ya sea porque aún no es su turno de salir o porque fue eliminado mediante una power pellet) y por último el GhostFrightened(hace que al ingerir una power pellet los fantasmas sean vulnerables).

Cada uno de estos comportamientos dependen de los nodos que posicionamos por el mapa, gracias a esto se crea una lista de grafo que permite que el fantasma tome posiciones y además tienen tiempos de duración y escenarios en los cuales se alternan entre sí.

- **Comportamiento Ghost Scatter.**

El comportamiento de GhostScatter está diseñado para que el fantasma deambule aleatoriamente por el mapa y así el jugador tiene margen de error; para esto se programa este comportamiento de tal manera que según los nodos más cercanos a los que este el fantasma se escoja uno de forma aleatoria. El comportamiento de GhostChase consiste en revisar en la lista de nodos cual es el nodo más cercano a la posición del pacman y mediante el cálculo de la distancia euclidiana determinar los nodos que establecen el camino más corto hacia el pacman, de esta manera se simula que el pacman está siendo perseguido.

- **Comportamiento Ghost Chase.**

Los comportamientos de GhostScatter y GhostChase se alternan entre sí cada cierto tiempo de tal manera que los fantasmas están 7 segundos deambulando después 20 segundos persiguiendo y vuelve a iniciar el ciclo en deambulando.

- **Comportamiento Ghost Home.**

El comportamiento GhostHome hace que se realice la animación del fantasma saliendo de la base y una vez afuera adopte los comportamientos GhostScatter y GhostChase. Es establecido inicialmente en Inky, Pinky y Clyde ya que estos son los que están inicialmente en la base y también es establecido en el momento que un fantasma es eliminado y regresa a la base para después volver a salir.

- **Comportamiento Ghost Frightened.**

El comportamiento GhostFrightened es el que establece cuando un fantasma es vulnerable tras consumir una power pellet, este comportamiento cambia la animación de los fantasmas por color azul y blanco. Cuando este comportamiento está activo los fantasmas empiezan a alejarse del pacman y esto se logra implementando la misma dinámica que en el comportamiento de GhostChase solo que en lugar de buscar la mínima distancia se busca la máxima distancia entre el pacman y el fantasma según los nodos más cercanos; cabe recalcar que durante este comportamiento la velocidad de los fantasmas se ve disminuida para que el jugador pueda alcanzarlos fácilmente.

- **Animación de movimiento de ojos en Ghosts:**

Por último se realizó un script para que la posición de los ojos en los fantasmas fuera acorde a la dirección en que se desplazaban, de esta manera si el fantasma va hacia la derecha sus ojos también irán a la derecha y así con las demás direcciones 2D.

- **Puntuación:**

El sistema de puntuación es sencillo, cada que se consume un pellet, power pellet y ghost se incrementa el score de acuerdo al ítem.

- **High Scores.**

El sistema de guardado de puntos se hace mediante la serialización de la clase HighScores que implementa Serializable, en esta clase se guarda los 10 scores más altos y los nombres del jugador. La ventana que le pide el nombre al jugador es la siguiente:



- **Audio del juego:**

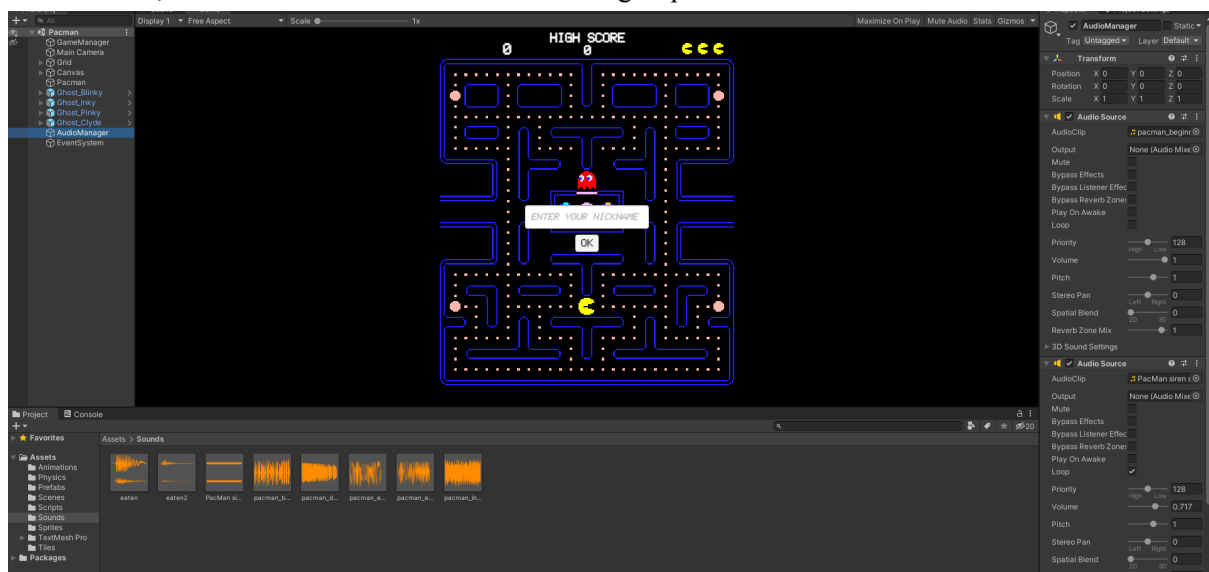
El audio del juego se insertó en cada scene, una por cada scene excepto en la escena del pacman. En la escena del pacman, al inicio suena un sonido de inicio y posteriormente suena otro, esto se consiguió mediante un método asíncrono que espera unos segundos a que termine el primer sonido antes de empezar, de manera tal que no se detenga el hilo principal de ejecución, dicho método se encuentra en la clase General Audio.

```

1  using UnityEngine;
2  using UnityEngine.Audio;
3  using System.Threading.Tasks;
4
5  public class GeneralAudio : MonoBehaviour {
6      public AudioSource beginning;
7      public AudioSource other;
8
9      public void Start() {
10         beginning.Play();
11         Other();
12     }
13
14     protected async void Other() {
15         await Task.Delay(4000);
16         other.Play();
17     }
18
19     public void Stop() {
20         other.Pause();
21     }
22 }
23

```

Por otro lado, en cada escena se creó un AudioManager que controla el sonido en cada escena.



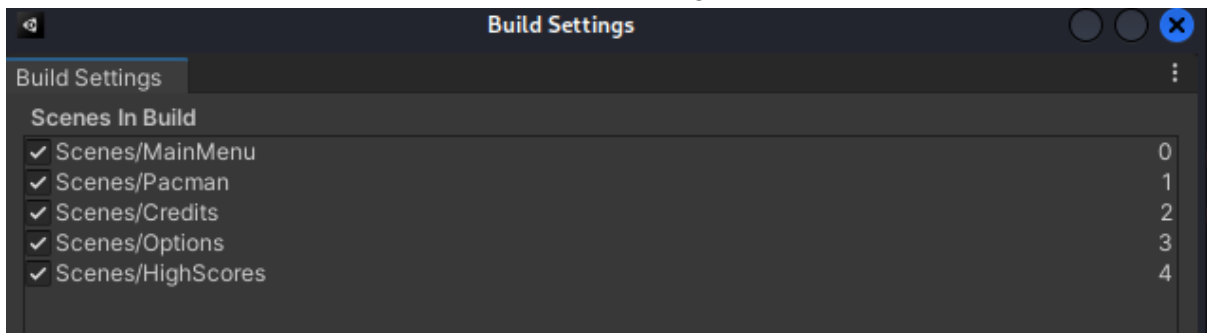
- **Vidas del jugador:**

El contador de vidas se representó de manera gráfica de una manera intuitiva para el jugador, en el que cada vez que el jugador muere se elimina un pacman de la interfaz. No hay manera de conseguir más vidas, siempre se cuenta máximo con 3 vidas por partida.



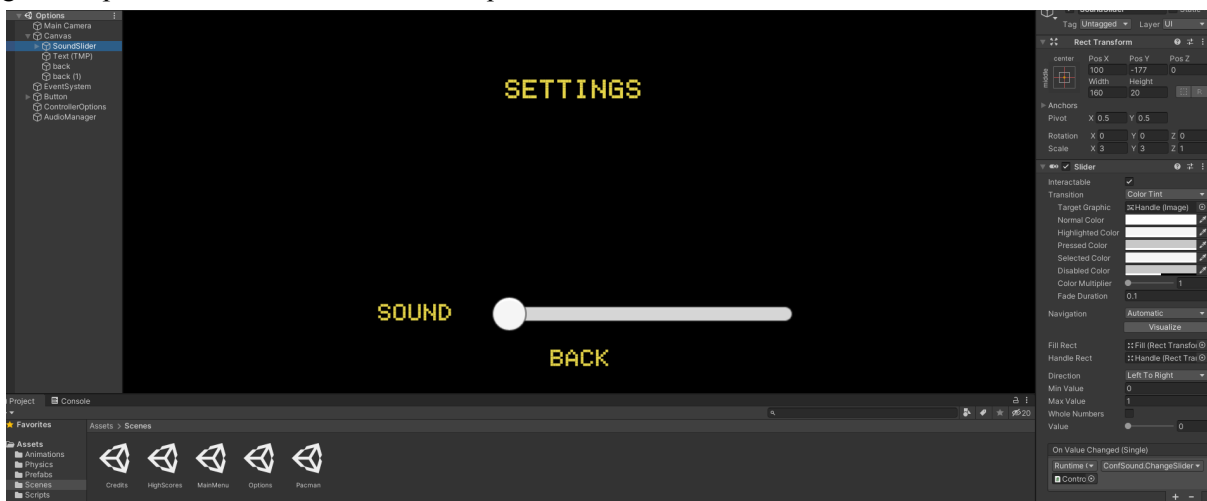
- **Menú principal:**

El menú principal y las otras escenas tienen asociado el script scene switcher que recibe un entero como parámetro y dependiendo de la entrada del jugador se cambia de escena, a continuación se muestra el entero asociado a cada escena en el Build Settings.



- **Configuración:**

La configuración del juego es un menú sencillo en el que el jugador ajusta el volumen del juego en general, para esto se usó un soundSlider que controla el volumen.



- **Recursos consultados:**

- https://www.youtube.com/watch?v=TKt_VIMn_aA
- <https://www.youtube.com/watch?v=Q4LMPXDT0rs>
- <https://www.classicgaming.cc/classics/pac-man/sounds>
- https://www.youtube.com/watch?v=BfUWguM5_jQ