

# Generador de Mazmorras

—Unity 3D—

Por: Nadia Gonzalez

—Programación para Videojuegos—

Universidad Tecnológica Americana:  
UTECA



## índice

Contexto: .....	3
Consideraciones: .....	4
Implementación: .....	5
Contenido: .....	6
¿Cómo puedo modificarlo en Unity? .....	9
Main Camera: .....	9
El jugador .....	10
Habitación Central: .....	11
Prefabs / Habitaciones .....	13
Room Templates .....	14
Cámara Mini Mapa .....	16



## Contexto:

Este es un generador de mazmorras o habitaciones que se agrupan y acomodan de forma aleatorio, cada vez que se inicia el juego.

Para este generador se utilizan como base el **Grid** de la escena (pues tienen que quedar a la medida para que todos los cuartos se acoplen bien) Por otro lado, también se utiliza la herramienta de **Text Mesh Pro** y **Text** para mostrar una interfaz de datos en pantalla junto con un **Raw Image** para crear un mini mapa.

Este sistema proporciona un sistema básico de movimiento para nuestro personaje junto con uno para mover la cámara con el mouse (primera persona), pues es relevante para poder explorar nuestras mazmorras mejor, este movimiento se genera gracias a las teclas de las flechas o las teclas W, A, S, D junto con el movimiento del mouse.



En pocas palabras este generador de mazmorras/habitaciones hará lo siguiente:

- ✓ Crear diferente cantidad de habitaciones cada vez que se inicie.
- ✓ Cada vez que se inicie el generador también las habitaciones cambiarán de localización.
- ✓ En cada habitación creada se localizará un enemigo.
- ✓ En una de las habitaciones habrá un jefe.

Con su interfaz se podrá ver:

- ✓ Un mini mapa
- ✓ La cantidad de habitaciones creadas
- ✓ La cantidad de enemigos y jefes creados
- ✓ La posición de nuestro jugador

Dicha generación funciona cada vez que abras/inicies la herramienta/aplicación.

## Consideraciones:

- ✓ Este proyecto se trabaja con 10 unidades por 10 unidades de unity para hacer las habitaciones (10 cubos x 10 cubos) si no quedan a la medida o en ese rango las habitaciones, no funcionara el generador.
- ✓ Recuerda que al ser un proyecto 3D su rendimiento dependerá del equipo que estes utilizando.
- ✓ Para el mini mapa se utiliza el **Render Texture**.
- ✓ Este proyecto utiliza la técnica de poner objetos vacíos o llamados en este proyecto como **"SpawnPoint"** en la entrada/salida de cada habitación (A 10 cuadros de la salida/entrada)
- ✓ Este proyecto trabaja con la función **Random** para generar las cosas aleatoriamente.
- ✓ Se utilizan dos cámaras (Main camera y camera)
- ✓ Recuerda agregar los diferentes elementos correspondientes en cada una de las casillas nuevas que proporcionen los scripts en unity (una vez asignados los scripts a los objetos como jugador, etc.)
- ✓ Recuerda que debes tener instalada o instalar la paquetería de **Text Mesh Pro** para que se pueda mostrar la posición de personaje en pantalla (pues en uno de los scripts se creara una variable para colocar dicho texto).
- ✓ Para este proyecto se descargaron paqueterías de la Asset Store (para el uso de modelos 3d y ambientación) En Assets aparecen como **Red Nebula Studios** (dicho paquete se utilizó para la ambientación, y es importante pues los prefabs contienen su ambientación), **Crusader tank** (del cual se utiliza el modelo 3D para el jefe), **Low Polyskeleton** (para el modelo 3D de los enemigos) y **AllSkyFree** (se utilizo para darle una "imagen" el cielo de nuestra escena).

-**Nota:** Puedes eliminar todas esas paqueterías para colocar las de tu elección-



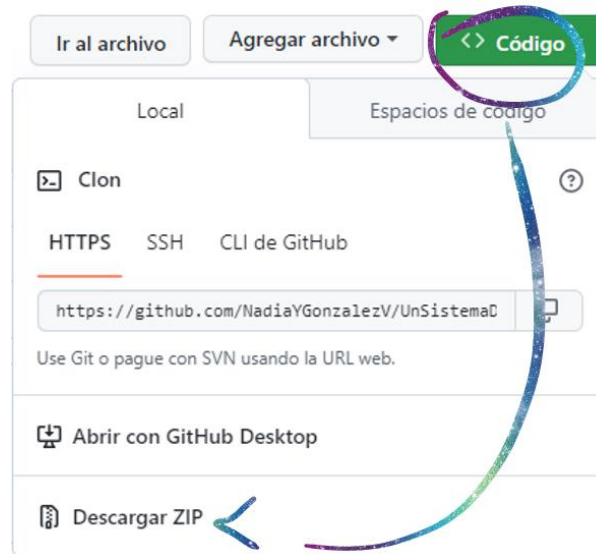


## Implementación:

- ✓ Deberás descargar todo el proyecto para que funcione correctamente en Unity desde GitHub, el repositorio con el sistema está en el siguiente enlace:

<https://github.com/NadiaYGonzalezV/Mazmorras-3D-Generador.git>

- ✓ Una vez que te encuentres en el repositorio, le darás clic al botón de “Código” y posteriormente al de “Descargar ZIP”



- ✓ Una vez descargado el archivo lo ubicas en tu dispositivo (normalmente en la carpeta de descargas)
- ✓ En la carpeta donde se encuentre el ZIP descargado , crearas una nueva carpeta para mover el Zip dentro de ella (esto para tener orden,opcional).
- ✓ Una vez dentro de la nueva carpeta ,extraeras los archivos de la crpeta dando clic derecho y en “Extraer aquí”
- ✓ Listo, tendras acceso a todos y cada uno de los archivos del proyecto, junto al Unity package file.

Si quieres probar la herramienta, se encontrará en la carpeta de “Ejecutables” con el nombre de “Mazmorras 3D Generador” con la propiedad de “Aplicación”.

## Contenido:

Este documento te ayudará a crear una herramienta para generar mazmorras/habitaciones de forma aleatoria al igual que poner enemigos en ellas.

En este sistema se proporcionan varios elementos:

- ✓ 8 scripts que serán los responsables de que nuestra herramienta funcione a la perfección:

- 4 scripts encargados de que las habitaciones se generen bien:

**RoomTemplates:** Este script es el encargado de que las habitaciones aparezcan y se ordenen de forma aleatoria, además de colocar los enemigos en cada habitación y mostrar la interfaz de usuario del “**número de habitaciones**” “**numero de enemigos**”.

**RoomSpawner:** Este script se encarga de decir cuáles son las salidas de cada habitación, pues así busca una que complementa esa salida y puedan construir habitaciones con entradas y salidas.

**Destrucrion:** (Sí, en vez de una “r” debía ir una “t”, pero ese nombre tiene el script en unity) Este script ayuda a que no se coloquen habitaciones en nuestra “habitación principal” y siempre sus 4 salidas conecten a otras habitaciones.

**AddRoomToList:** Este script se encarga de mostrarnos en el inspector la cantidad de salas creadas y el nombre de cada una de ellas.

- 2 scripts para interfaz de usuario (aparte de también “**RoomTemplates**”):

**MostrarPosicion:** Como su nombre lo dice, aquí se utiliza el Text Mesh Pro para actualizar en tiempo real la posición de tu “jugador” y mostrarlo en pantalla.

**MiniMapCamera:** Igualmente, como su nombre lo dice, es el encargado de que la segunda cámara que creemos para el “Mini Mapa” siga al jugador cuando este se mueva y lo muestre en el mini mapa que tenemos como interfaz.

- 2 scripts para el movimiento de nuestro **jugador** y de nuestra **“Main Camera” para** tener una vista en **primera persona**.

**CameraLook:** Este script es el encargado de que nuestra cámara se mueva fluidamente con el movimiento de nuestro mouse sin necesidad de apretar algo en el y siga a nuestro jugador. Además de que no importa donde este mirando el jugador, las teclas se adaptaran para que funcionen a la perfección.

**PlayerController:** Este script (como su nombre lo dice) es el encargado de darle movilidad a nuestro jugador para poder explorar las habitaciones.

A continuación se te presentarán fragmentos de código que podrás encontrar en este proyecto para explicar brevemente su función (en su mayoría los códigos en visual studio se encuentran comentados):

#### MostrarPosicion:

```
public Transform personaje; //colocar a tu jugador
public TextMeshProUGUI textoPosicion;

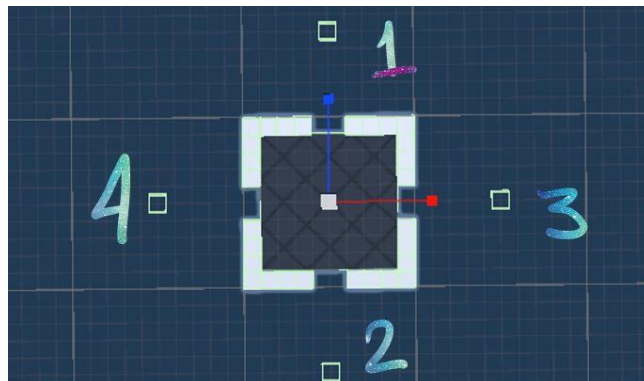
void Update ()
{
    textoPosicion.text = "Tú Posición: " + personaje. position.ToString();
}
```

#### RoomSpawner:

```
public int openSide; //saber el sitio donde se encuentra la puerta o el sitio
abierto

//1 necesitamos una puerta abajo
//2 necesitamos una puerta arriba
//3 necesitamos una puerta izquierda
//4 necesitamos puerta derecha
```

Ten cuidado a la hora de otorgarles un número a tus salidas/entradas en unity porque de ellos depende como se deban acomodar las habitaciones (te dejo el esquema de cómo hacerlo)



### AddRoomToList:

```
//este script es el encargado de mostrar la lista en el inspector  
  
//private RoomTemplates templates;  
  
public RoomTemplates templates;  
  
void Start()  
{  
  
templates=GameObject.FindGameObjectWithTag("Rooms").GetComponent<RoomTemplates>  
( );  
    templates.rooms.Add(this.gameObject);  
  
    templates.roomCountText.text = "Número de Habitaciones: " +  
templates.rooms.Count;  
  
}
```

### -Por último-

- ✓ Paquetes utilizados para la ambientación: **Red Nebula Studios**, **Crusader tank**, **Low Polyskeleton** y **AllSkyFree** (recuerda que si vas a utilizarlos respeta los derechos de autor y otorga el crédito correspondiente)
- ✓ Se proporcionan los Prefabs de las habitaciones utilizadas (el prefab de la habitación con nombre "Room template" fue la base para las demás habitaciones, solo agregando o quitando paredes(cubos).



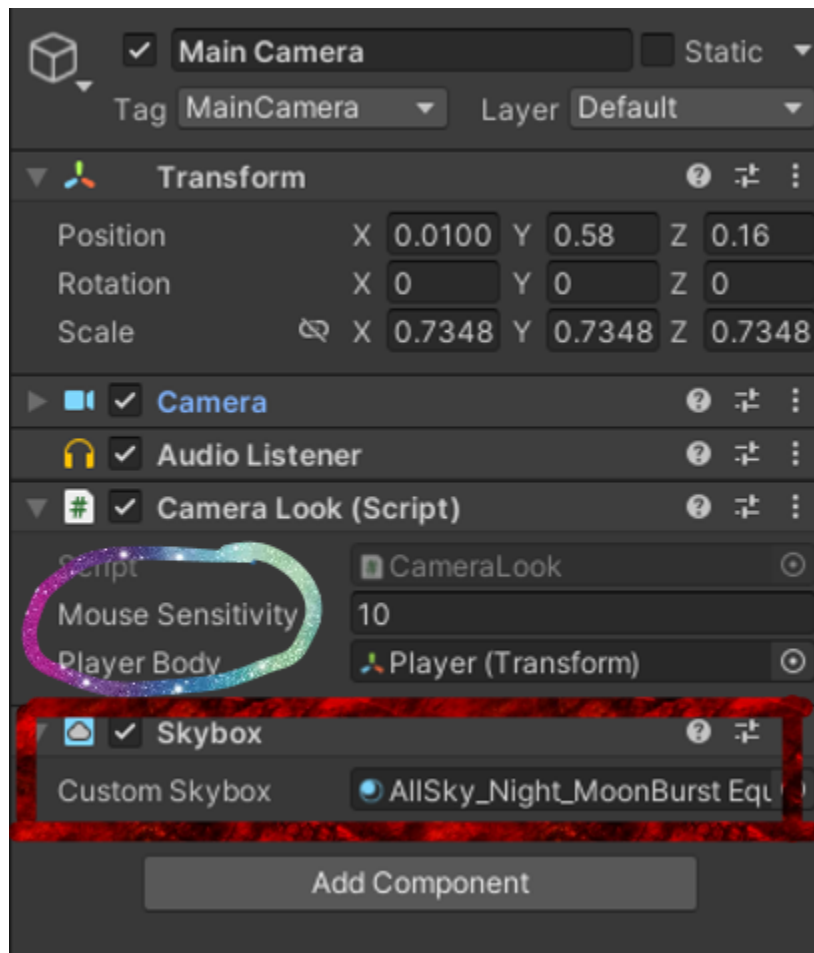


# ¿Cómo puedo modificarlo en Unity?

Este proyecto se puede modificar y personalizar de distintas maneras:

## Main Camera:

- ✓ Mouse Sensitivity: Otorgas la cantidad de sensibilidad que tu mouse tendrá
- ✓ Player Body: Colocas el objeto de tu “Jugador” o “player” para que tu cámara los siga (o en su caso, colocas el objeto que necesites seguir)
- ✓ SkyBox: Este componente es opcional, pues ayuda a colocarle una imagen o textura a nuestro cielo (en el caso de que lo utilices en su casilla “Custom skybox” colocas tu material seleccionado)

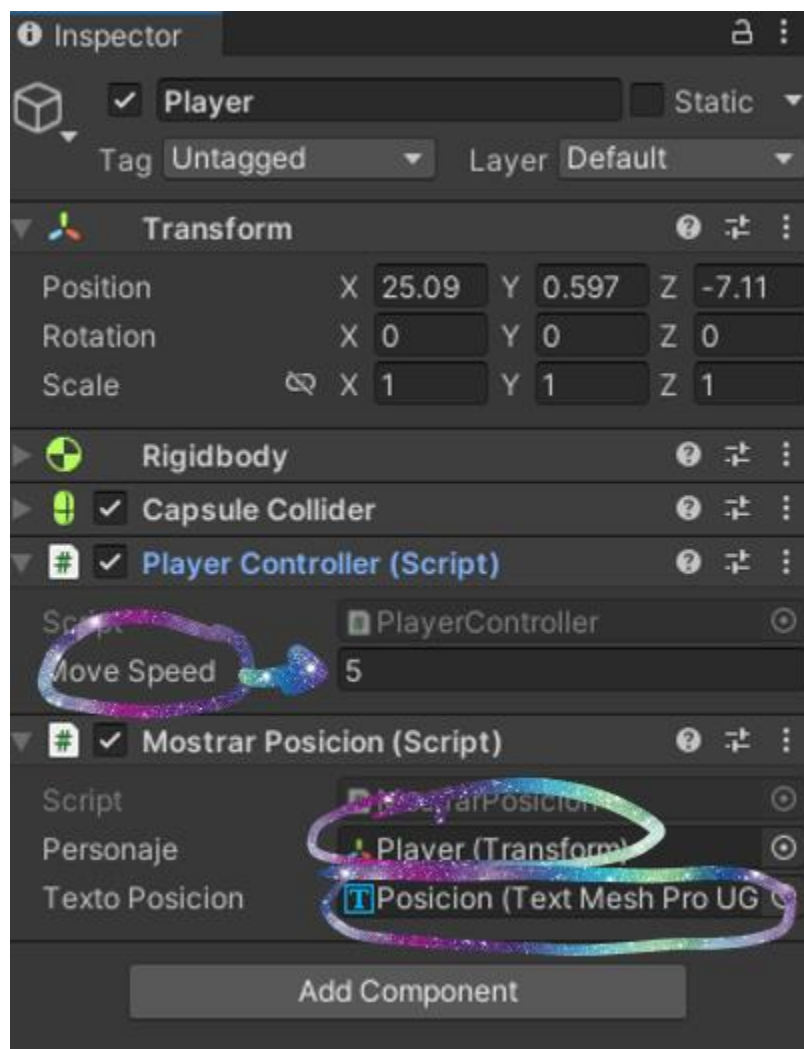


## El jugador

Aquí deberás colocarle los siguientes scripts a tu jugador o en su respectivo caso, al objeto que contenga a tu jugador.

Aquí agregaras los scripts de “**Player Controller**” y “**Mostrar Posición**”.

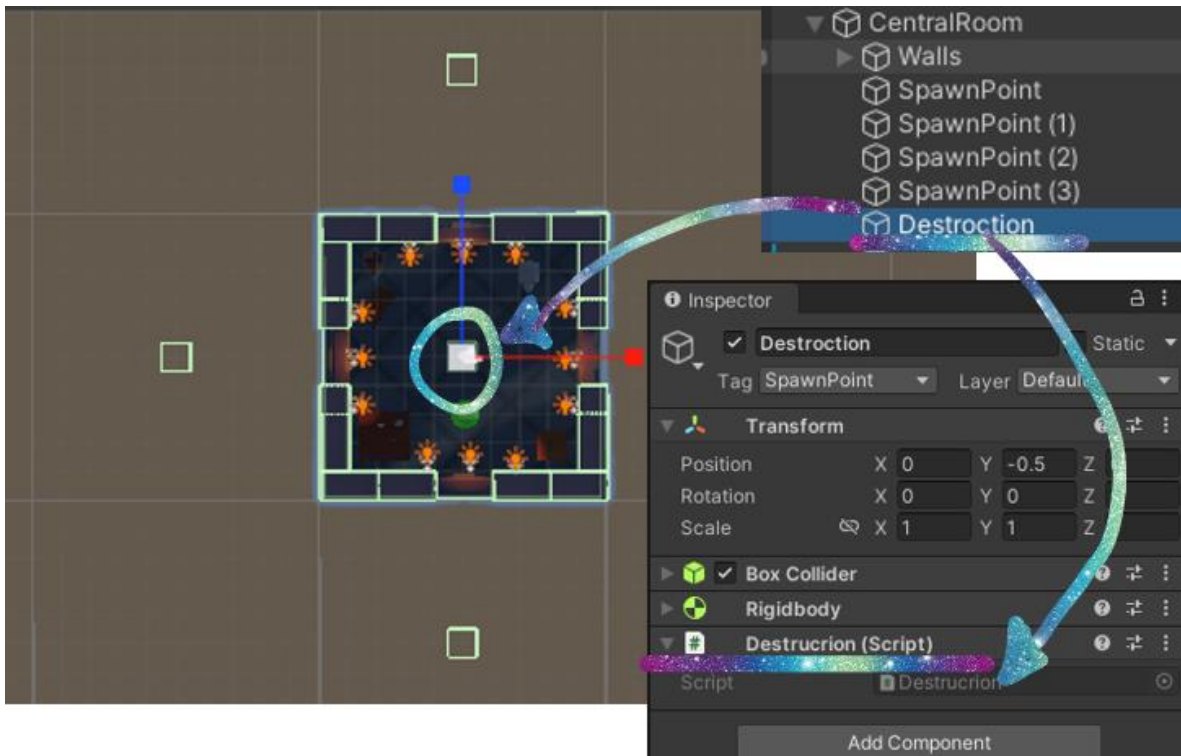
- ✓ Move Speed: la velocidad con la que se moverá tu Jugador.
- ✓ Personaje: Agregaras al objeto que quieres que muestre su posición en pantalla
- ✓ Texto Posición: el Text Mesh Pro que utilizaras para mostrar la posición en pantalla.



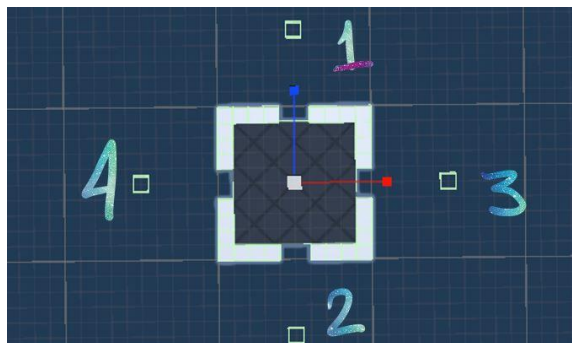
## Habitación Central:

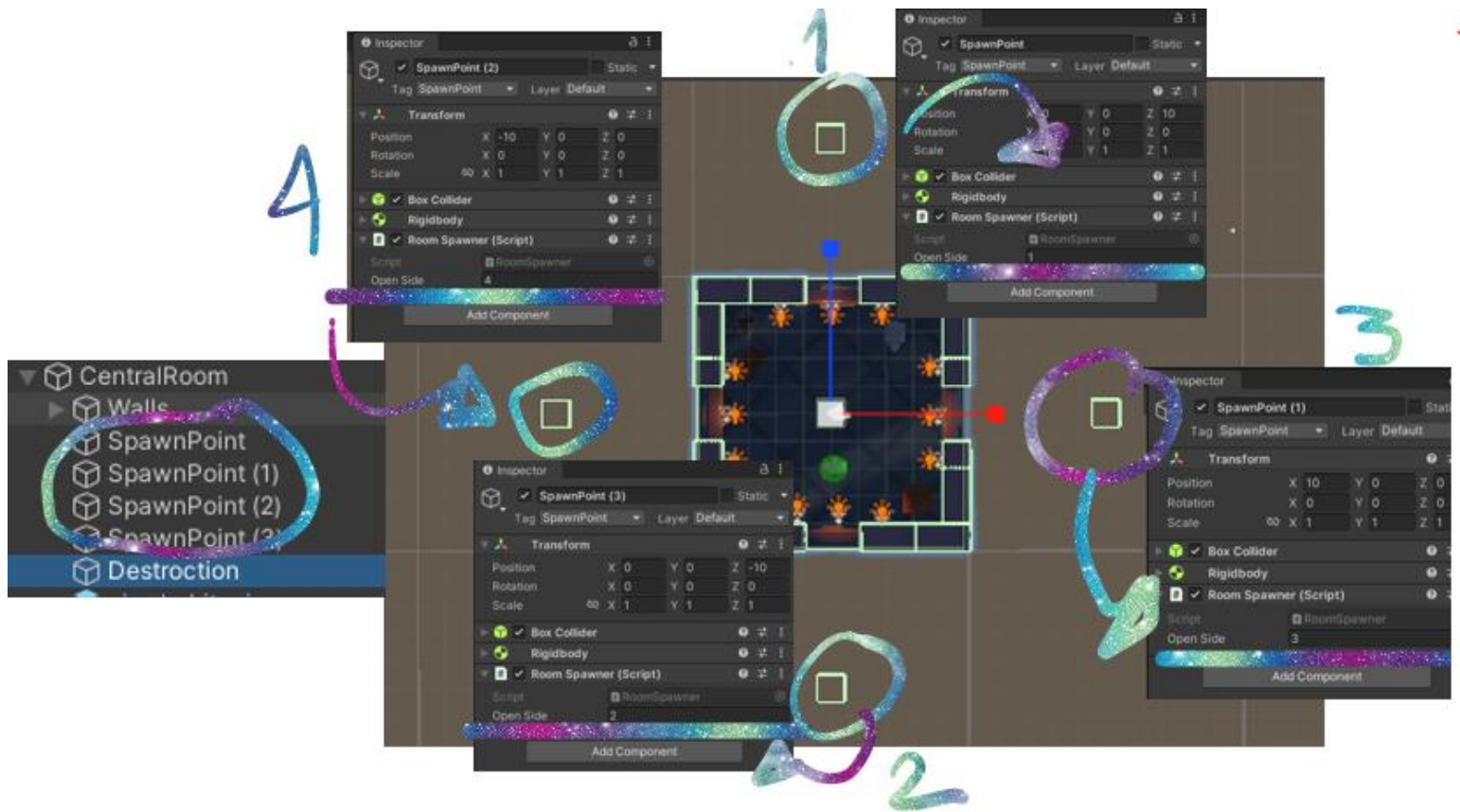
Para esta habitación son varias cosas por ser la habitación central o principal, pues siempre debe ir en medio:

- ✓ A un objeto vacío agregar el script de "Destruction" y el tag SpawnPoint después céntralo (que quede abajo del piso de tu habitación)



- ✓ A cada uno de los SpawnPoint de todas las habitaciones (los cuales son objetos vacíos a 10 cuadros de distancia de cada salida/entrada de nuestras habitaciones, sí, forzosamente debe ser este número) debes agregarle el script de "RoomSpawner" y el Tag SpawnPoint pues si no lo haces, el generador nunca sabrá donde está la entrada/salida y nunca podrá otra habitación continua.





- ✓ Aquí se da un ejemplo grafico de como debe quedar la habitación con 4 salidas de 2 cuadros cada una y asignarle su número correspondiente.
- ✓ A esta habitación será la única que no tenga un prefab

## Prefabs / Habitaciones

Estas habitaciones las puedes personalizar (decoración, etc.), puedes tener distintos tipos de entradas (para este proyecto solo son 2, si quieres usar 3 o mas debes configurarlo)

Recuerda que primero las haces en la escena, luego los vuelves prefabs y después las eliminas de la escena (pues ya son prefabs y no las necesitas en tu escena, pues después gracias a los prefabs se generaran aleatoriamente)

La habitación que está dentro de un círculo duplica la dos veces, esto es por qué:

- ✓ La primera será para diseñar tus demás habitaciones.
- ✓ La segunda será para crear el Closed **Room**, esta habitación no tendrá ningún script ni ningún SpawnPoint, simplemente será el encargado de cerrar habitaciones que lleven a la nada.

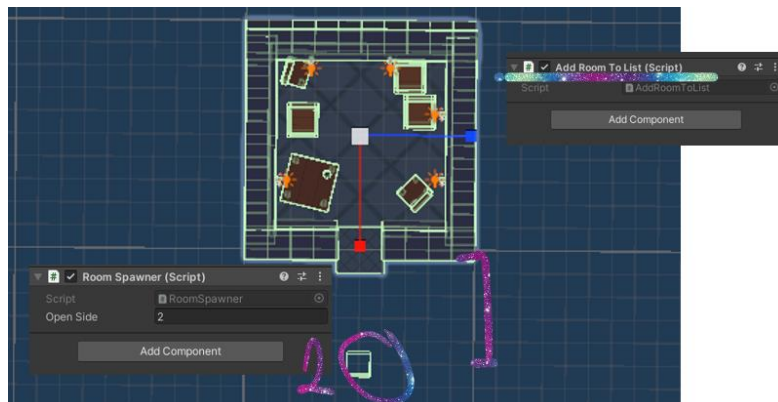
Agrégle los objetos vacíos o **SpawnPoints** junto al script **RoomSpawner** para decir donde son las entradas y salidas.

Ya después simplemente puedes duplicarla y borrar cubos para decir donde estarán los SpawnPoint (recuerda alinearlos a la salida correspondiente y nombrarlos dependiendo donde tengan la entrada)

A continuación te dejo el ejemplo de las salidas/entradas que puedes hacer.



Recuerda agregar a cada habitación (menos a la central) el script **AddRoomToList**





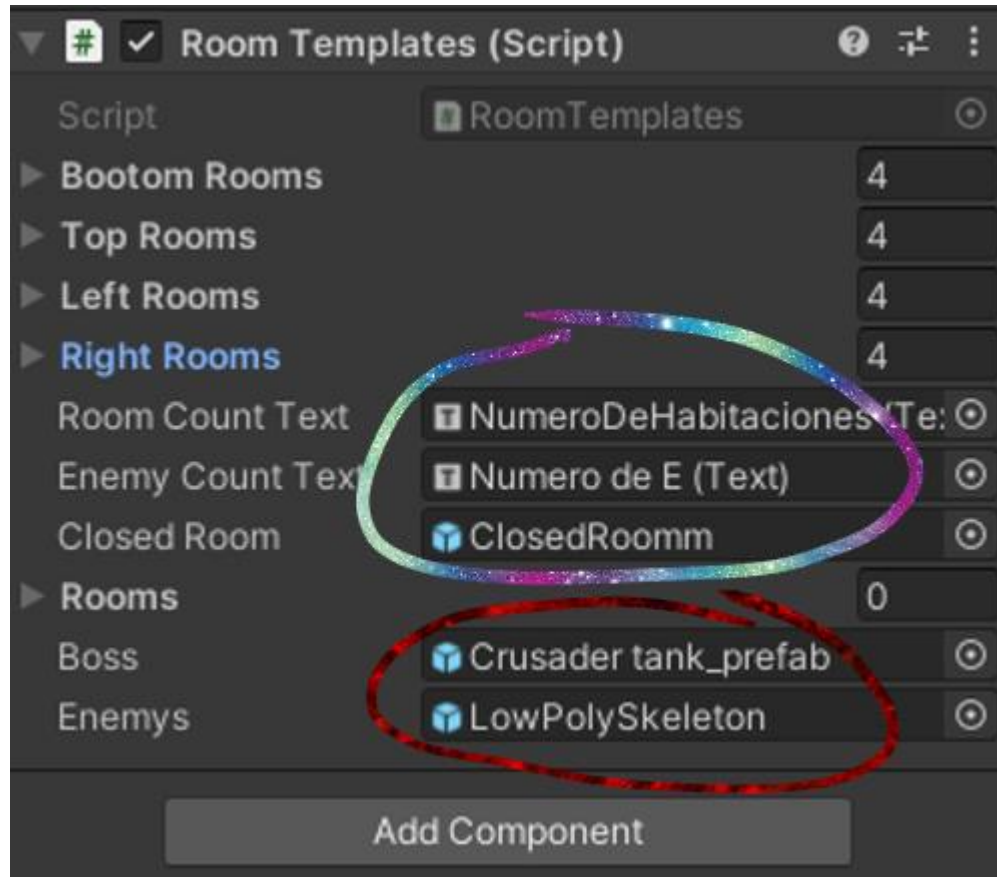
# Room Templates

En este **objeto vacío** también debes personalizar varias cosas, antes que nada agrega el script **RoomTemplates** y el Tag **Rooms**, después realiza lo siguiente:

- ✓ Agrega los prefabs de las habitaciones (menos la central) correspondientes a cada casilla donde tenga su salida/entrada.
- ✓ En este caso son 4 habitaciones por cada apartado, pero puedes agregar las que necesites.



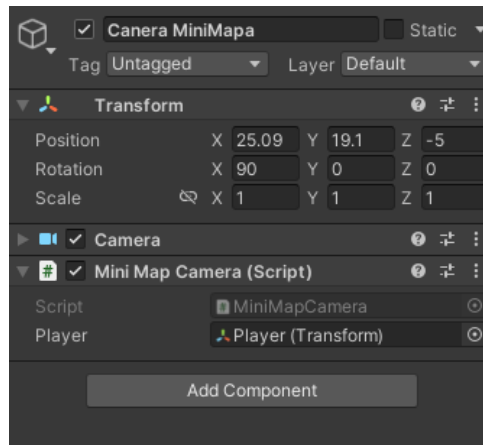
- ✓ Después puedes agregar tus elementos correspondientes para los textos que mostraran en pantalla el número de habitaciones, enemigos, jefes e incluso (encerrados en círculo rojo) los modelos de tus enemigos y jefes.
- ✓ También recuerda agregar tu habitación u objeto que funcionara como **Closed Room**.



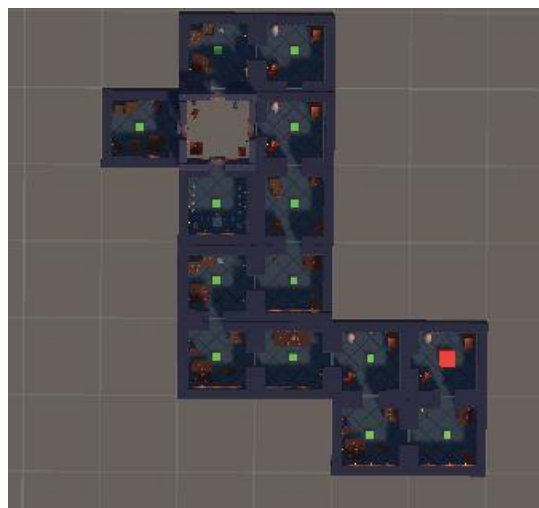
## Cámara Mini Mapa

Puedes personalizar la para que quede en la rotación y posición que tu desees.

Pero recuerda agregarle el script Mini Map Camera y el objeto que desees seguir.



Agrega el Render Texture guardado en tu carpeta de Mini mapa y coloca los RawImage correspondientes.



Espero que esta documentación  
te haya sido de ayuda y que  
puedas disfrutar al máximo mi  
herramienta de generación de  
mazmorras.

Contacto: [yologonzalezv@gmail.com](mailto:yologonzalezv@gmail.com)

