

Instructivo Binary Space Partitioning

Integrantes

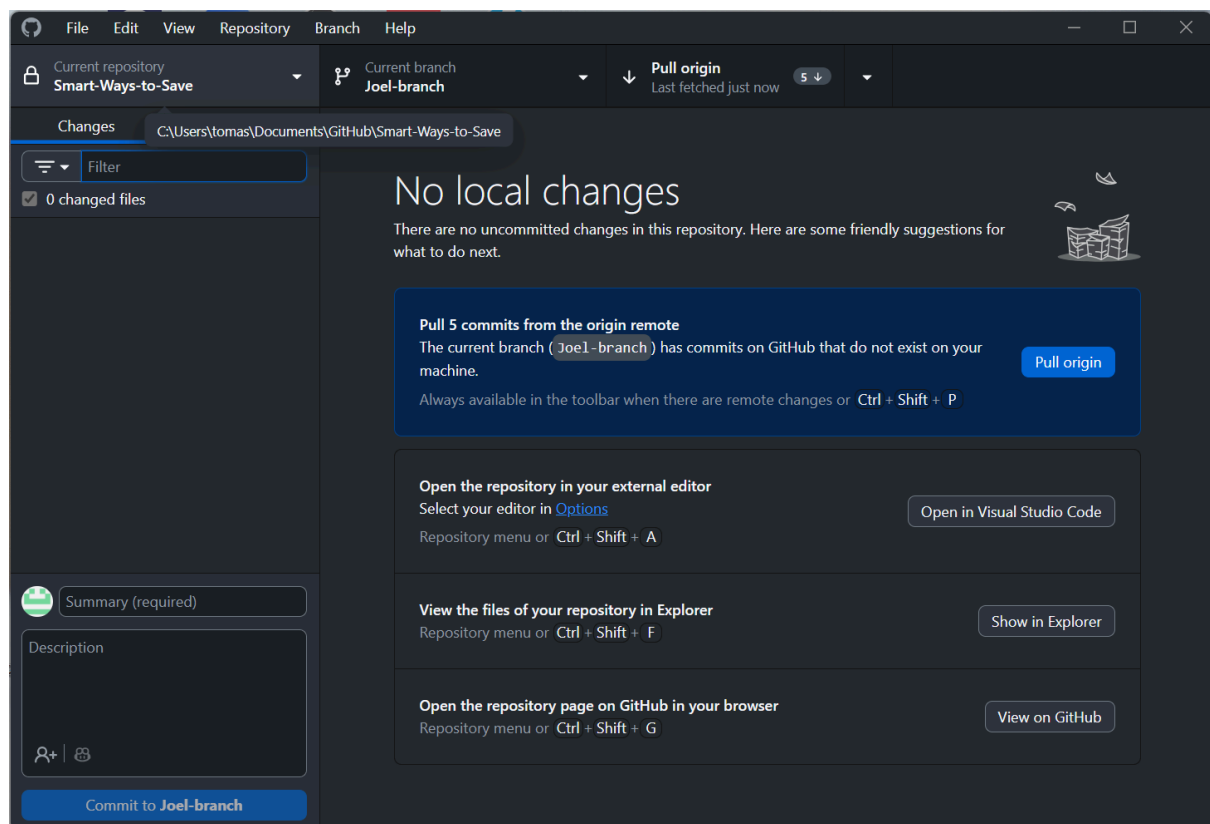
Joel Díaz
Tomás Concha

Overview

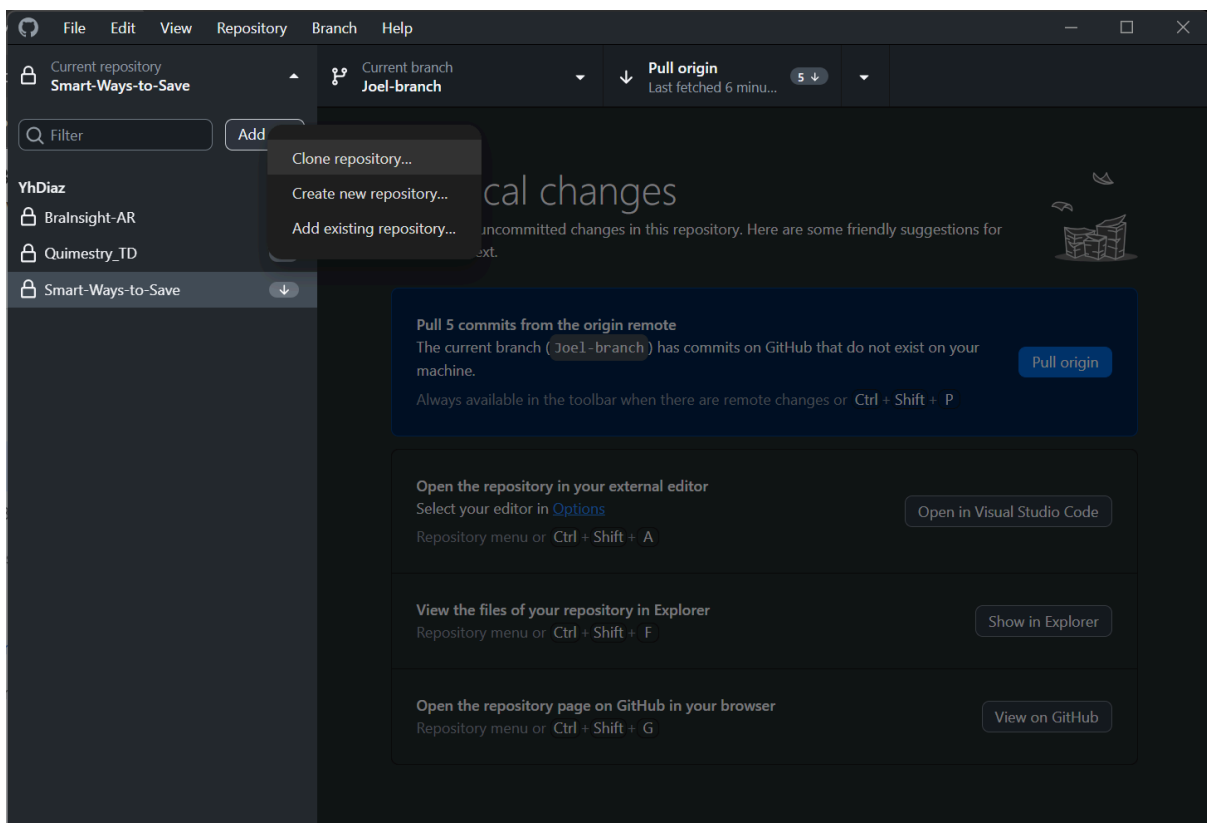
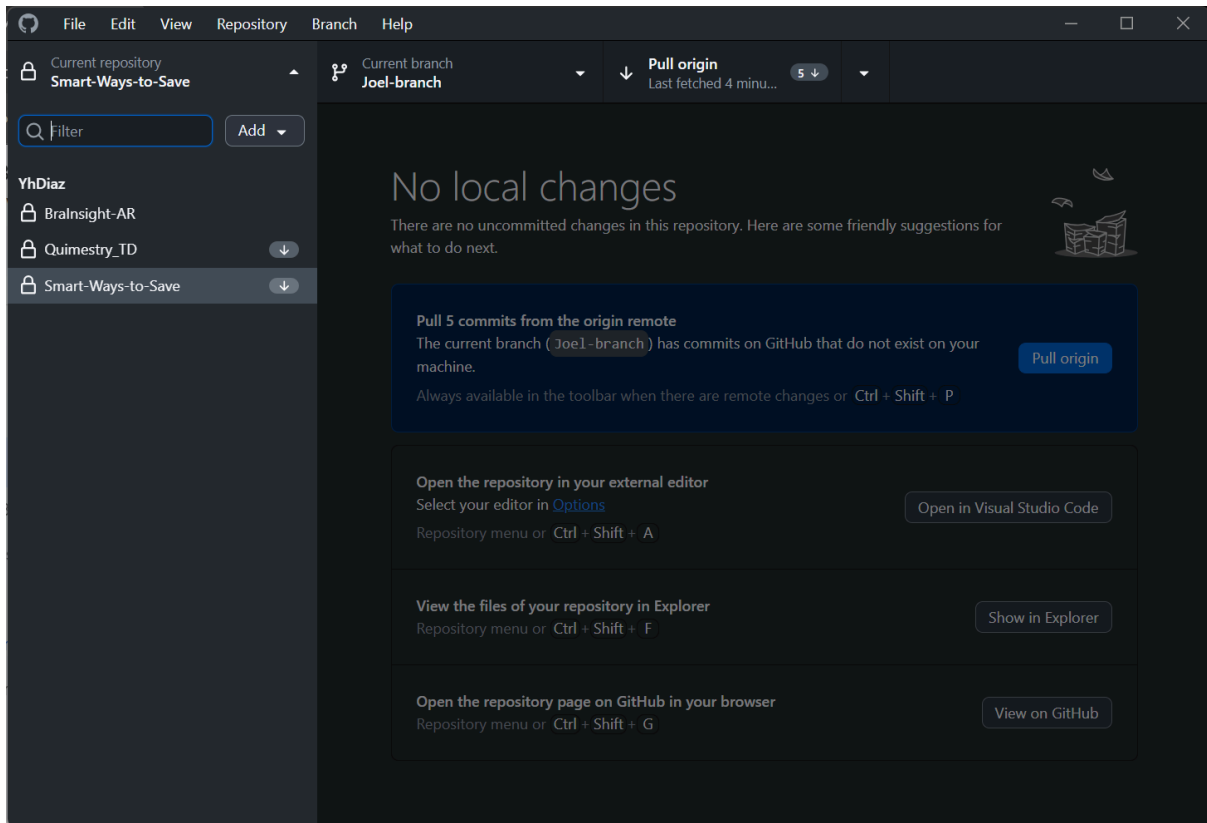
Este instructivo viene de la mano con una prueba piloto de un código de generación procedural que utiliza el algoritmo *Binary Space Partitioning* para crear mazmorras *roguelite*. A continuación se detalla el proceso para acceder al proyecto, y la descripción de cada una de las variables que pueden ser modificadas para alterar los resultados de generación.

Instrucciones de Instalación

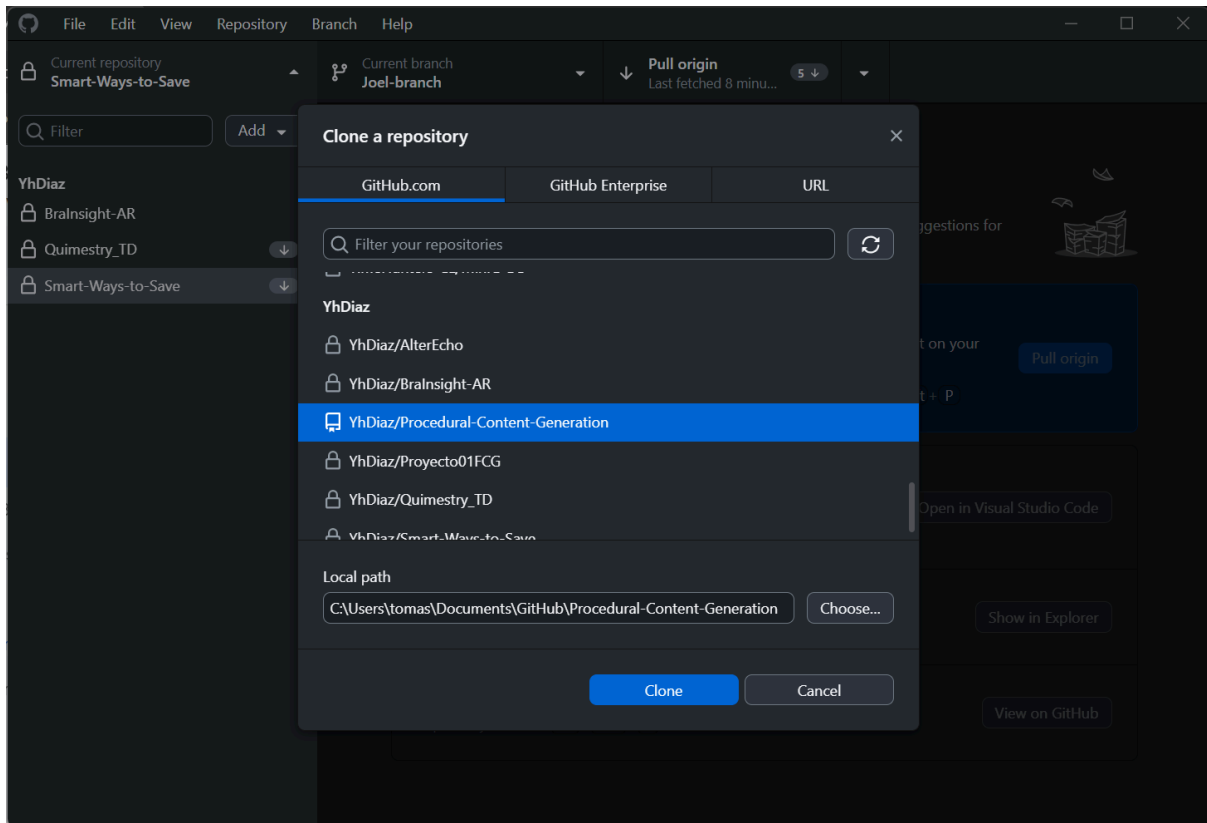
Teniendo una cuenta de GitHub y habiendo instalado GitHub Desktop, acceder a dicho programa y seleccionar el botón de *Current Repository* como se indica en la imagen de abajo.



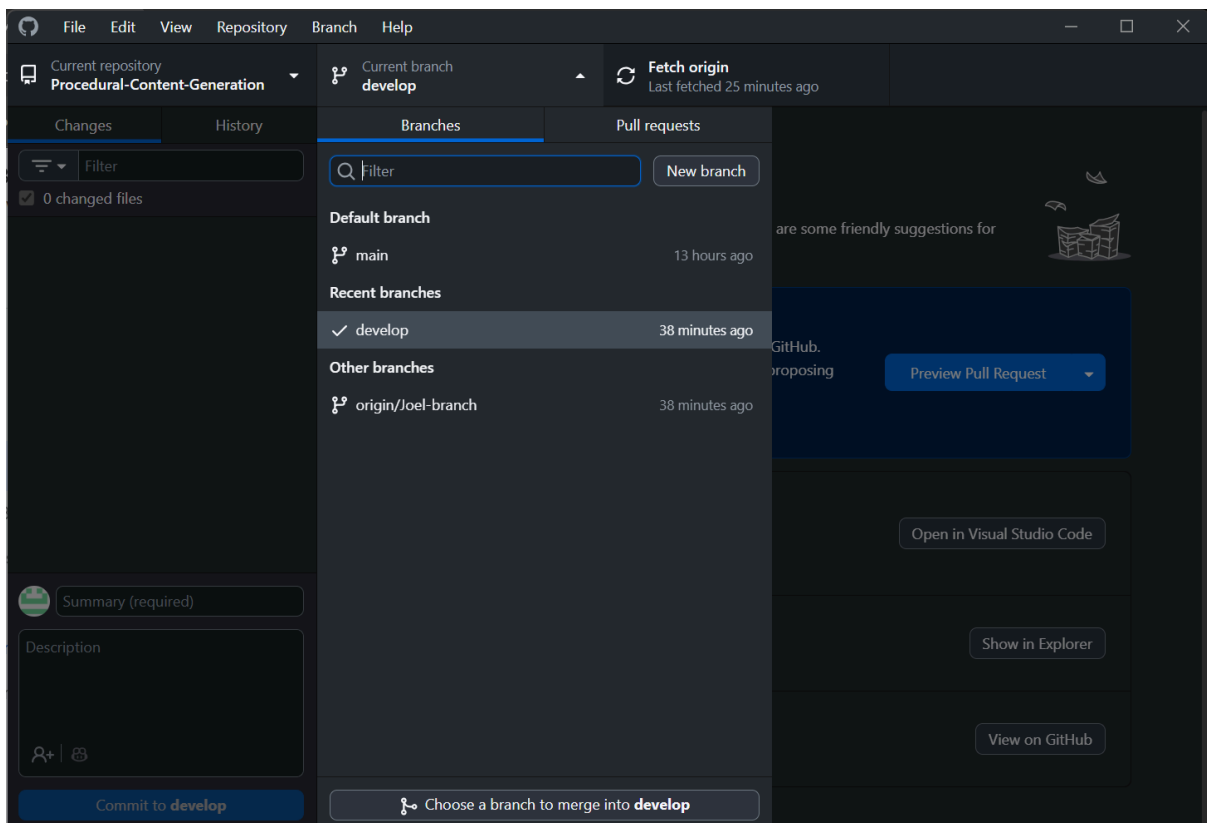
A continuación, presionar el botón *Add*, el cuál mostrará un drop down menu, y se debe seleccionar la opción *Clone Repository*.



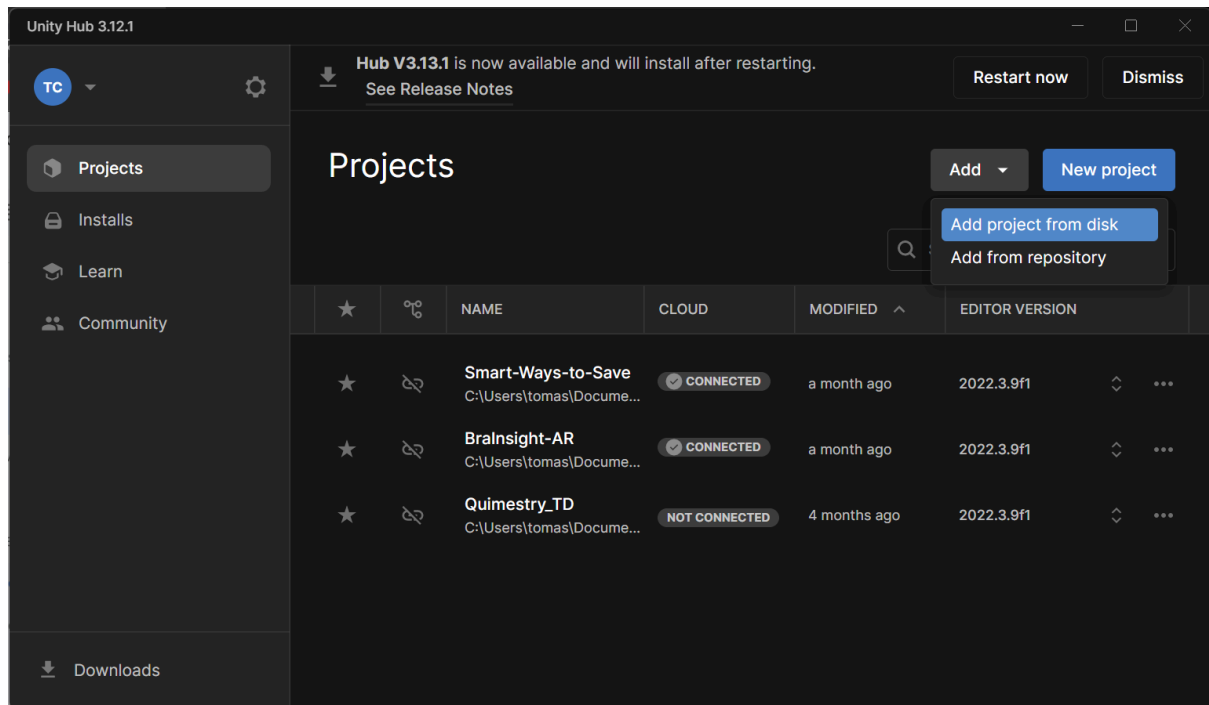
En la ventana emergente, buscar el repositorio de nombre Procedural-Content-Generation, y presionar el botón *Clone*.



Para poder visualizar los últimos cambios, navegue a la rama *develop*.



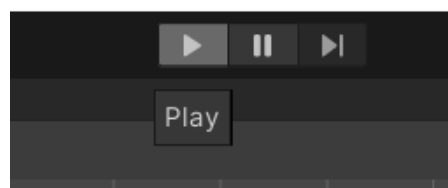
Con eso estará listo para abrir Unity y el proyecto (instalar la versión 2022.3.9f1 para evitar problemas), añadiendo el proyecto desde el disco, y buscando la ruta donde se haya clonado el repositorio en su dispositivo.



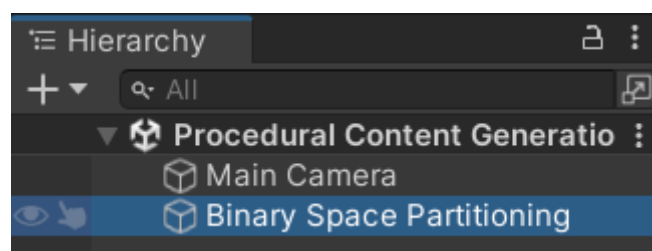
Instrucciones de uso

Una vez abierto el proyecto, desde el Inspector de Unity se pueden modificar los parámetros del algoritmo para obtener distintos resultados. Para esto debe seguir los siguientes pasos de preparación:

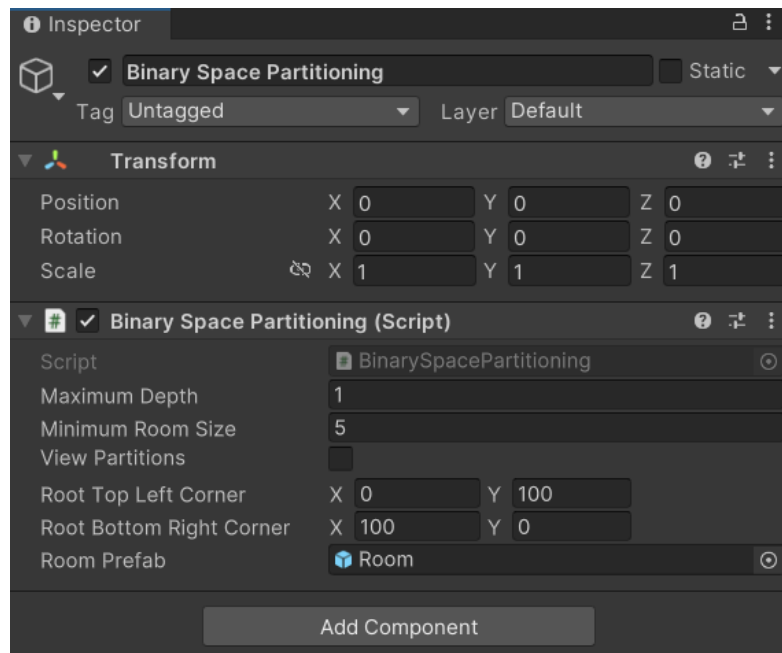
1. Ejecutar el programa presionando el botón Play del editor o con el comando Ctrl + P.



2. Seleccionar el objeto Binary Space Partitioning en la pestaña Hierarchy.



3. Modificar los valores del componente Binary Space Partitioning (Script) en el Inspector.



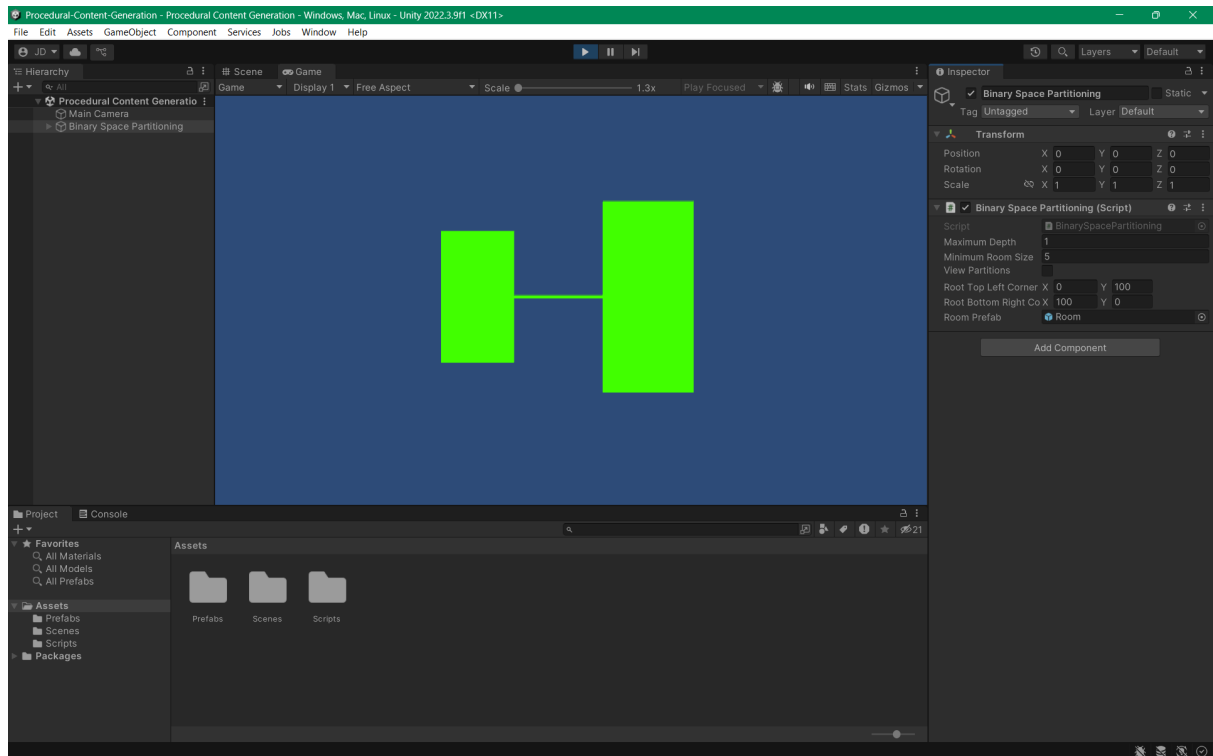
Nota: El campo Room Prefab no se debe modificar.

Dentro del script Binary Space Partitioning, existen las siguientes variables serializadas que pueden ser modificables para alterar las generaciones. A continuación se describe qué modifica cada uno de estos valores, junto con rangos recomendados de los mismos:

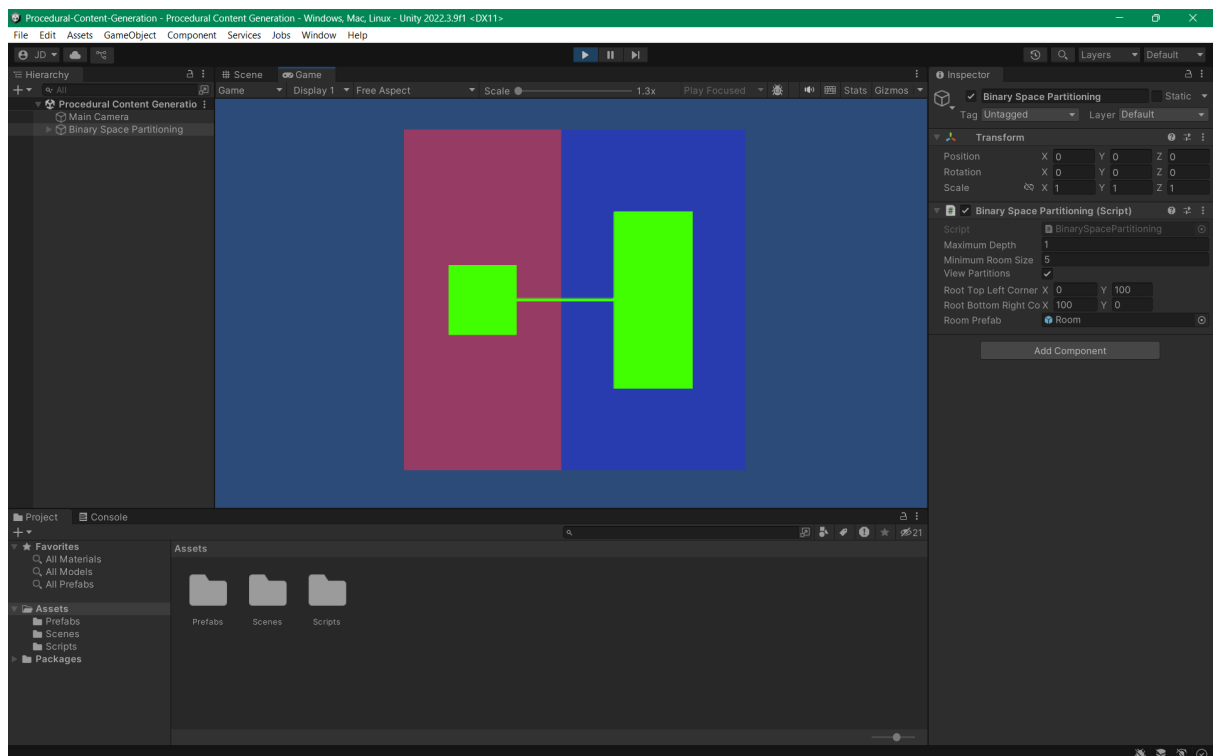
- **Minimum Depth:** Indica la profundidad que tendrá el árbol binario, lo cual se traduce en la cantidad de nodos hoja y mazmorras que se crearán. Mientras más alto sea este valor, más mazmorras se generarán. El número final de mazmorras será una potencia de 2 (2 elevado a Minimum Depth). Se recomienda probar con valores entre 1 y 10.
- **Minimum Room Size:** Indica el tamaño mínimo de una partición. Se recomienda probar con valores entre 5 y 20, dependiendo de la profundidad del árbol; Entre mayor sea la Minimum Depth, se recomiendan valores más pequeños (considérese que el valor es relativo al tamaño que utiliza un cuadrante dentro del espacio que tiene disponible, definido por Root Top Left y Bottom Right Corner).
- **View Partitions:** Esta variable activa (true) o desactiva (false) la visualización de los cuadrantes en los que se divide el mapa.
- **Root Top Left Corner y Root Bottom Right Corner:** Indican el tamaño de la grilla y, por lo tanto, de las particiones y mazmorras. Se recomienda probar con tamaños entre 50x50 y 100x100.

Dentro de los posibles resultados que puede obtener según los parámetros que modifique se encuentran los siguientes:

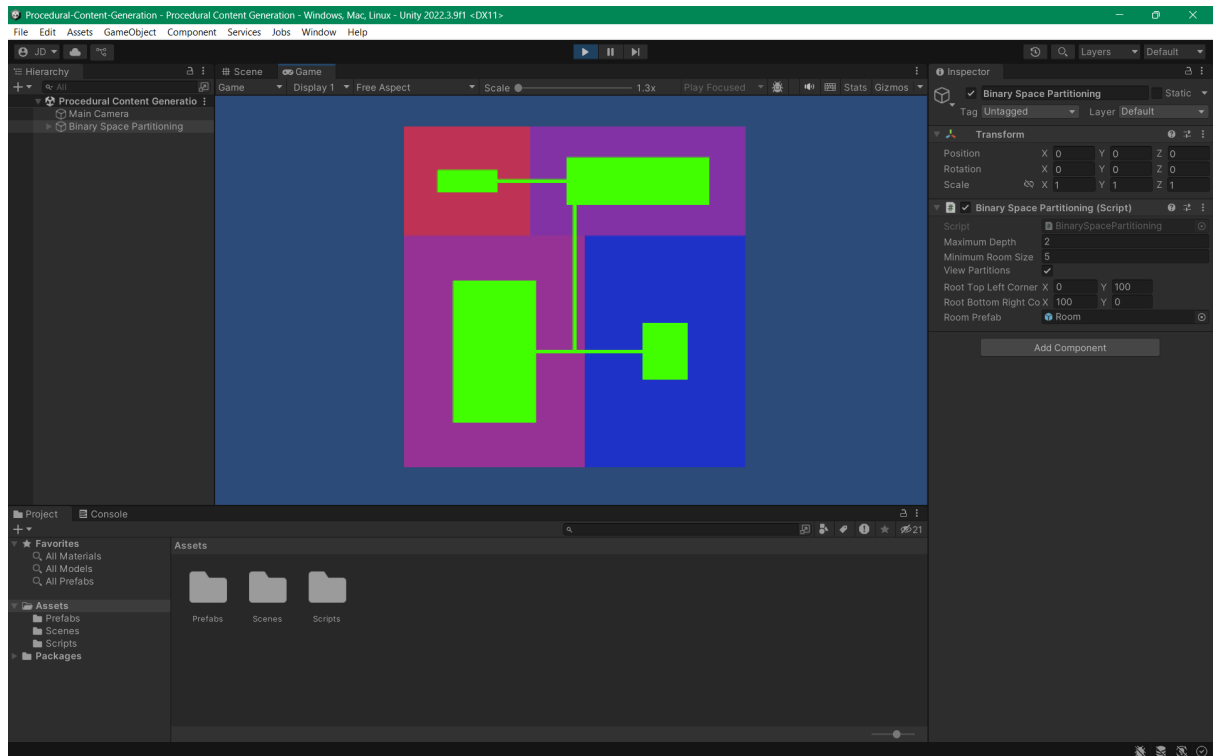
1. **Minimum Depth = 1**



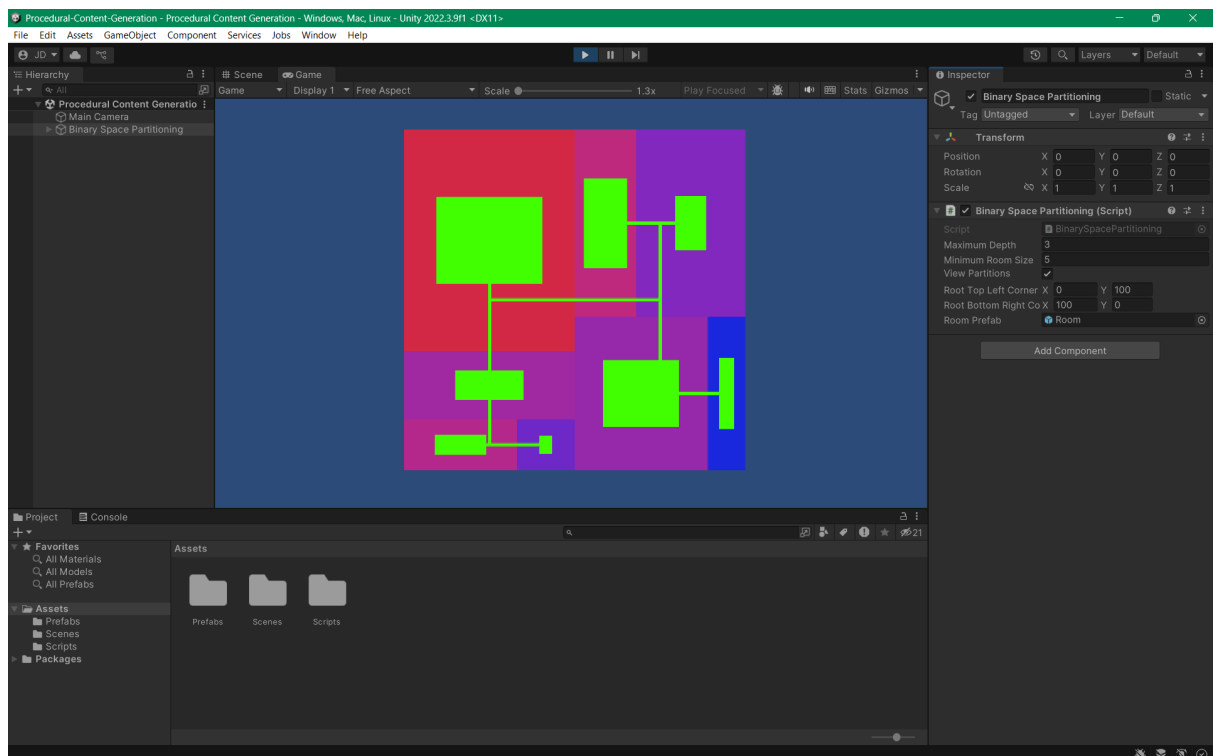
2. View Partitions = true



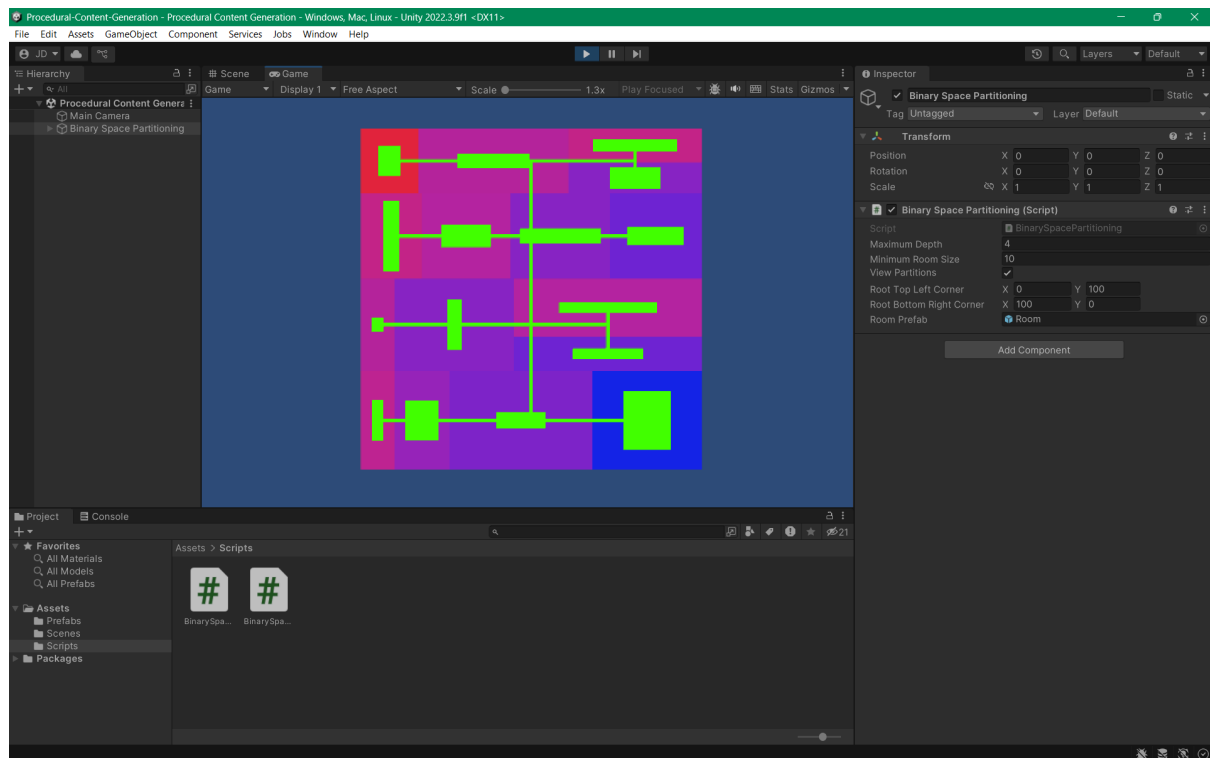
3. Minimum Depth = 2



4. Minimum Depth = 3



5. Minimum Depth = 4, Minimum Room Size = 10



6. Grid 75x75 (Root Top Left Corner = (0,75), Root Bottom Right Corner = (75, 0))

