

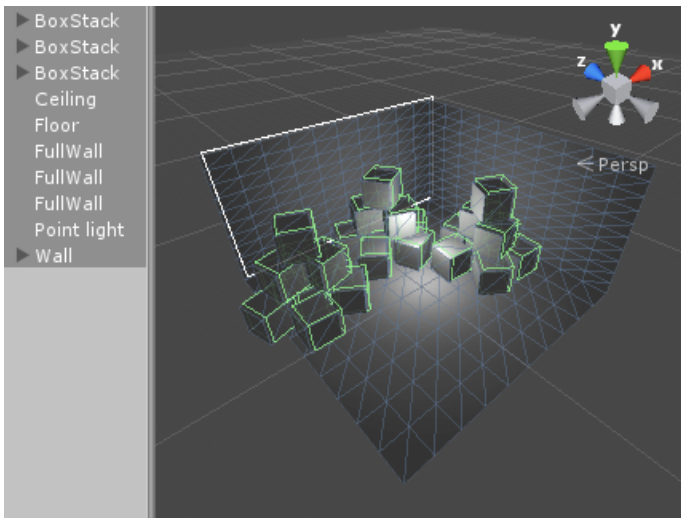
CORE



QUICKSTART

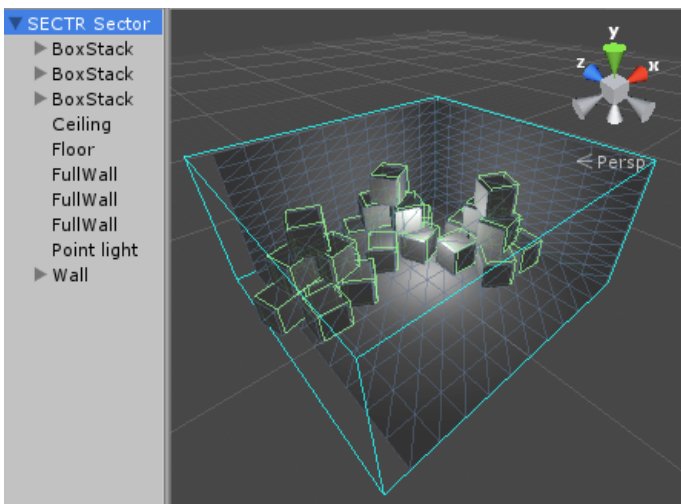
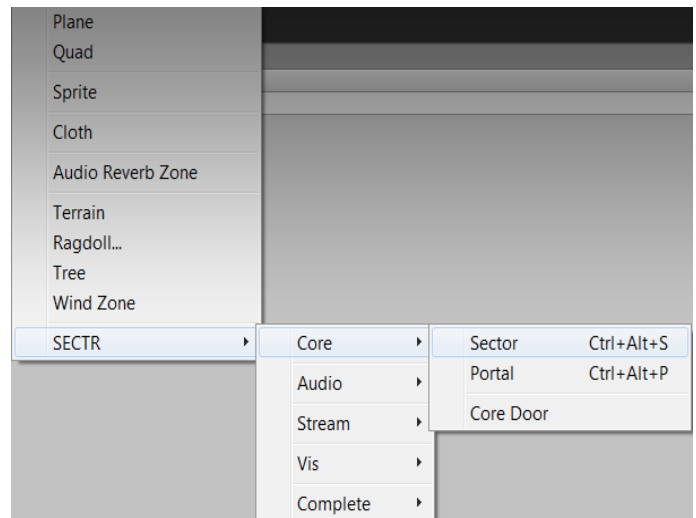
Creating Sectors

The first step in preparing a level for SECTR is creating a few Sectors.



First, select the objects that you want to define the space of your Sector, like the walls, floor, ceiling, or anything else that you want.

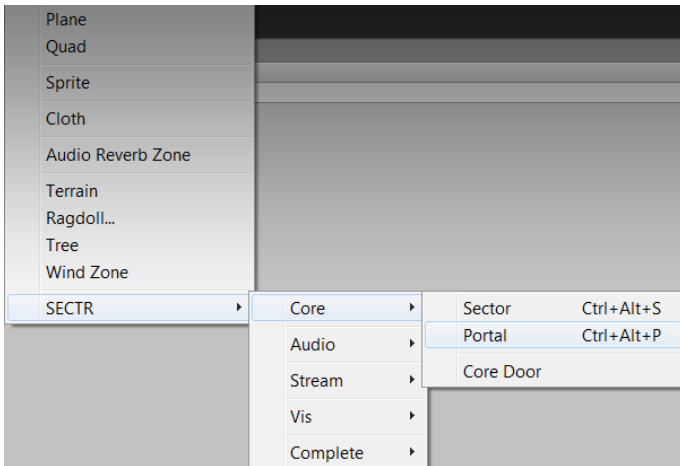
Next, go to Game Object->Create Other->SECTR->Core and choose Sector or press Shift + Alt + S.



Bam! New Sector. If the selected objects weren't already grouped under a single transform, they will be auto-grouped for you.

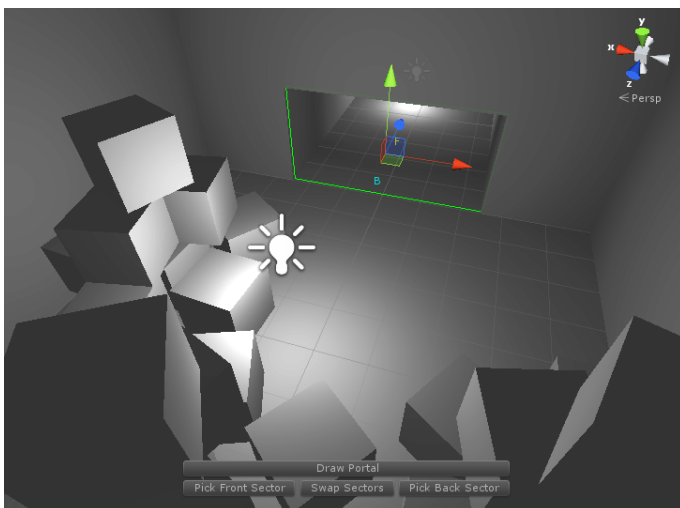
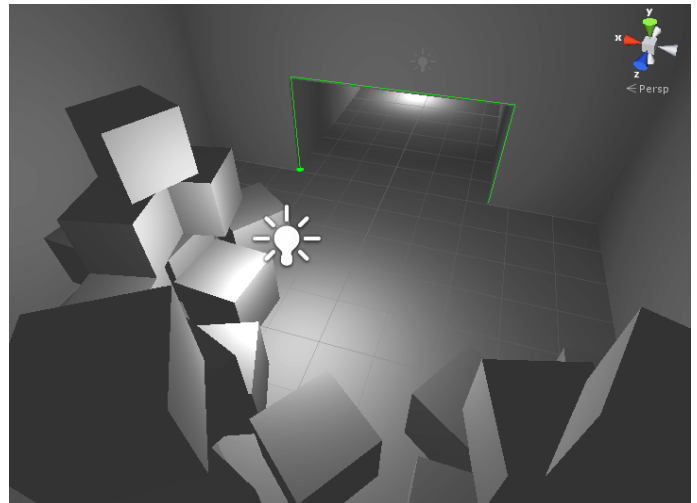
Adding Portals

Once you've created a couple of Sectors, it's time to connect them with a Portal.



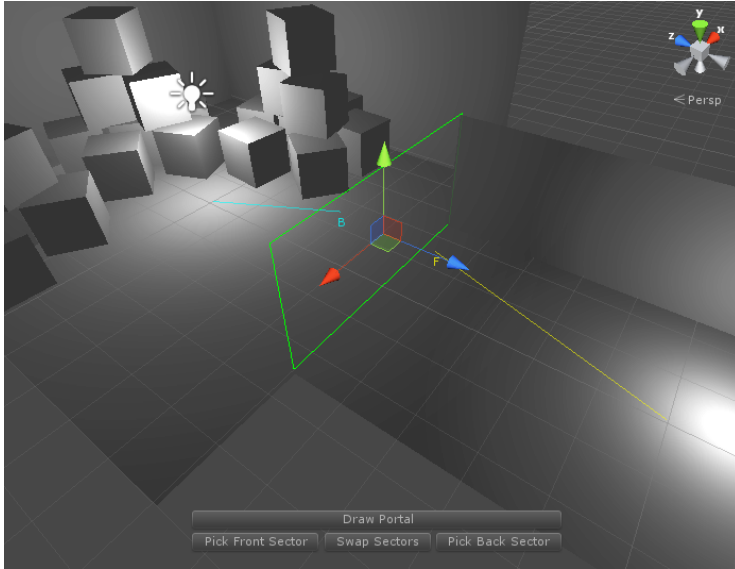
First, go to Game Object->Create Other->SECTR->Core and choose Sector or press Shift + Alt + S. This will activate the portal drawing tool.

Simply click to place the verts of your Portal. Portals can have as many sides as you want, but they have to be planar and convex. The portal drawing tool will turn red if the current position would make an invalid portal.



When you're happy with the geometry of your portal, either press Enter or click on your first vert to complete the Portal.

Now that the Portal is created, the last step is to connect the Portal to the two Sectors on either side.



Simply click the Pick Front Sector button, and then click on any object in that Sector. Do the same thing for the Back Sector, and you're all done.

You'll know that you got it right if the B and F lines draw in the right direction. If you accidentally got them backwards, no problem, just click the Swap Sectors button.

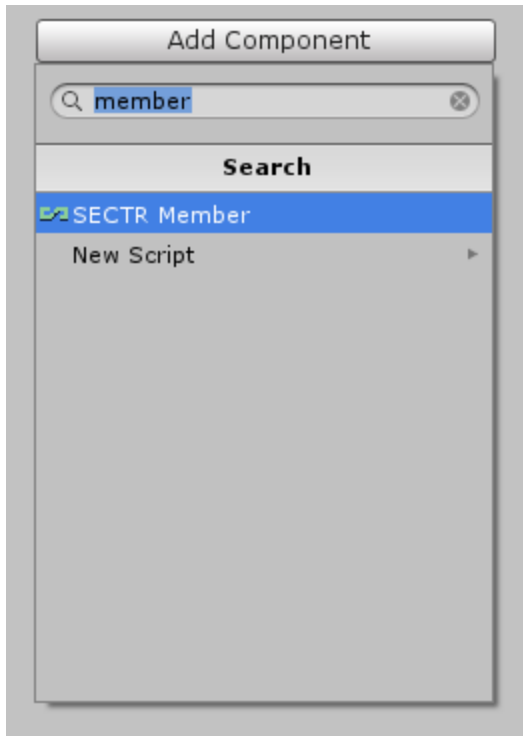
Pro Tip

You don't have to draw out meshes for every portal. You can assign any mesh resource using the Inspector. Just make sure that the mesh is planar and convex.

Adding Dynamic Objects

Sectors and Portals define and connect your space, but most of the interesting objects in the game (like the player, enemies, cameras, etc) are dynamic and need to be able to move from one Sector to the next. As these objects move around, it's important that they be able to keep track of what Sectors they are in.

The solution to this is very simple...



For any object that moves around and wants to know what Sector(s) it's in, simply add a Sector Member component to it. That's it!

Pro Tip

Member components will take care of itself and all of it's children. You don't need to add a Member to each child. This feature is an important part of more advanced techniques, which you'll learn about in SECTR VIS.

Cool! Now What?

Now that you've added Sectors and Portals, you have a logical representation of the spaces in your game and how they are connected. You can use this data to make anything you want, from smarter AI to advanced audio. SECTR CORE includes full source; you can build whatever you want on that foundation. If you like the quality of the tools and code in SECTR CORE, we encourage you to check out our paid modules, which do a lot of very cool things, very quickly, very easily.

STREAM



Makes it easy to save memory, increase performance, and decrease load times by splitting your scene into multiple chunks and streaming them in realtime.

AUDIO



Brings the latest, cutting edge audio production tools and technologies to Unity, including an unparalleled suite of editor extensions and runtime components that let you create rich, complex soundscapes with ease and play them back with a minimum of CPU overhead.

VIS



A high performance, low memory, fully dynamic occlusion culling solution for Unity, with support for the complete set of Unity rendering primitives including lights, shadows, particles, meshes, and terrain.

SECTR CORE

QUICKSTART

Questions or Problems?

support@makecodenow.com