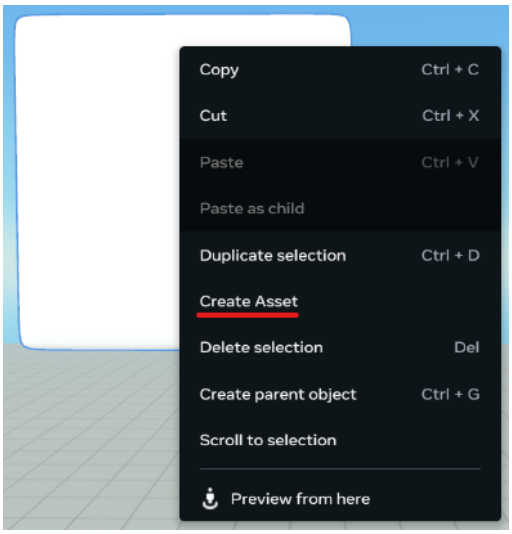
Challenge Steps:

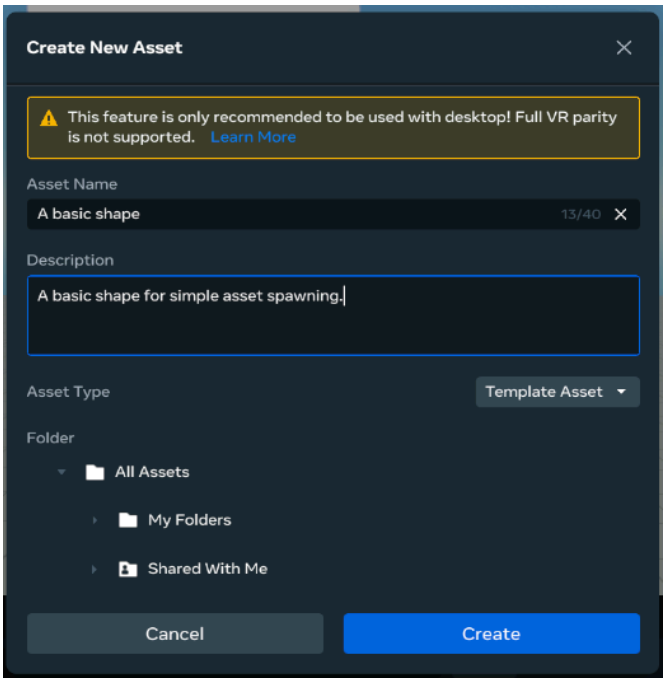
1. Create Your Template Asset

Let's create our first template asset!

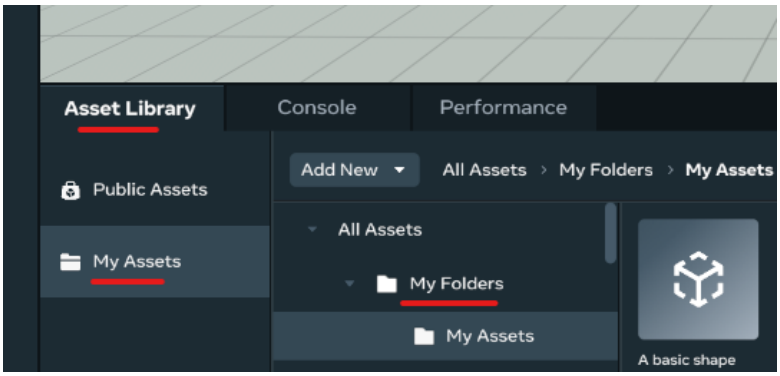
a. Join any world using the desktop editor.
b. You can select an existing object in your world or create a new one. A basic shape is good enough!
c. Right-click the object or parent (if grouped) and select *Create Asset.*



d. In the *Create New Asset* window give your asset a *Name*, *Description*, select a *Folder* to save it in, and click *Create*.



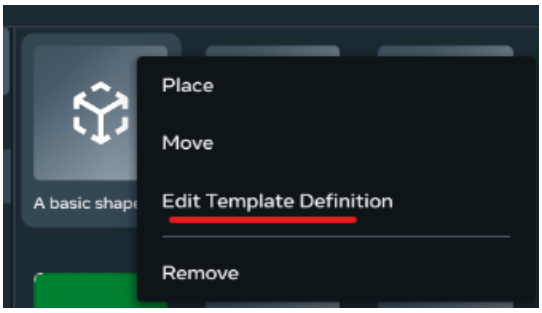
e. That's it! You created your first Template Asset! Your asset folders can be accessed at the bottom of the desktop editor under the *Asset Library* tab > *My Assets* > the folder you selected in the previous step.



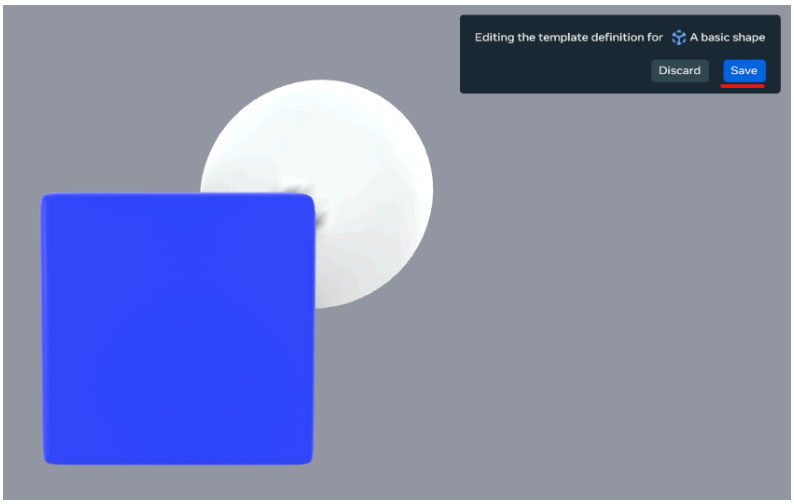
2. Edit your Template Asset

Let's make a change to our template asset. We can add new shapes, change the color, attach a script, or any number of things! You choose.

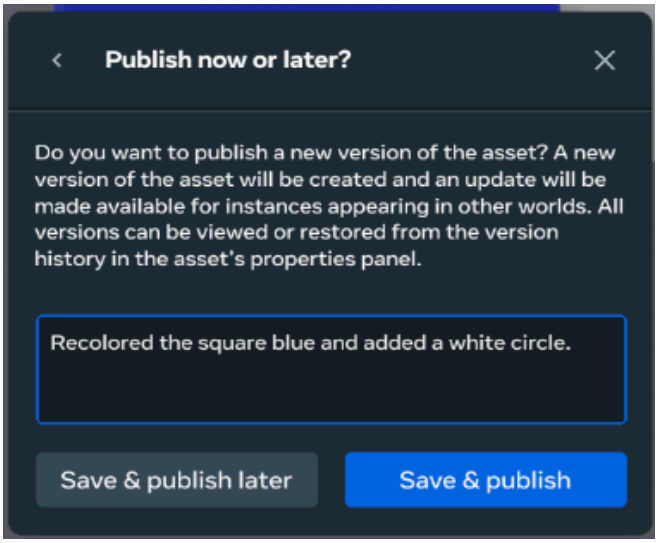
a. Find your newly created asset in your *Asset Library*, right-click on it, and select *Edit Template Definition*.

i. 

b. This will isolate your asset in the viewport and hierarchy. I decided to change my asset to my favorite color and add a circle! You can add new shapes or assets here by dragging them into the viewport. When you are done, click *Save* in the upper right corner!

i. 

c. You will be asked to add a *Description* of your changes and decide whether to *Save & Publish later* or *Save & Publish* now. Add your *Description* and click *Save & Publish*.

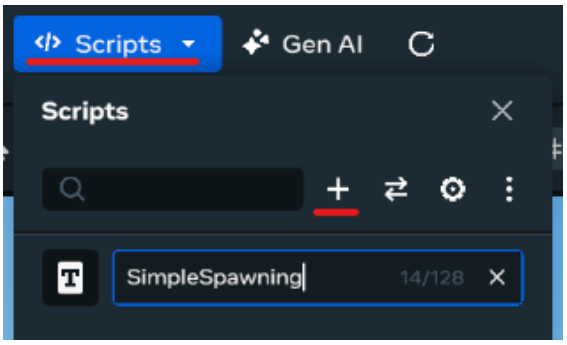
i. 

d. Congratulations. You are almost finished!

3. Spawn your Asset

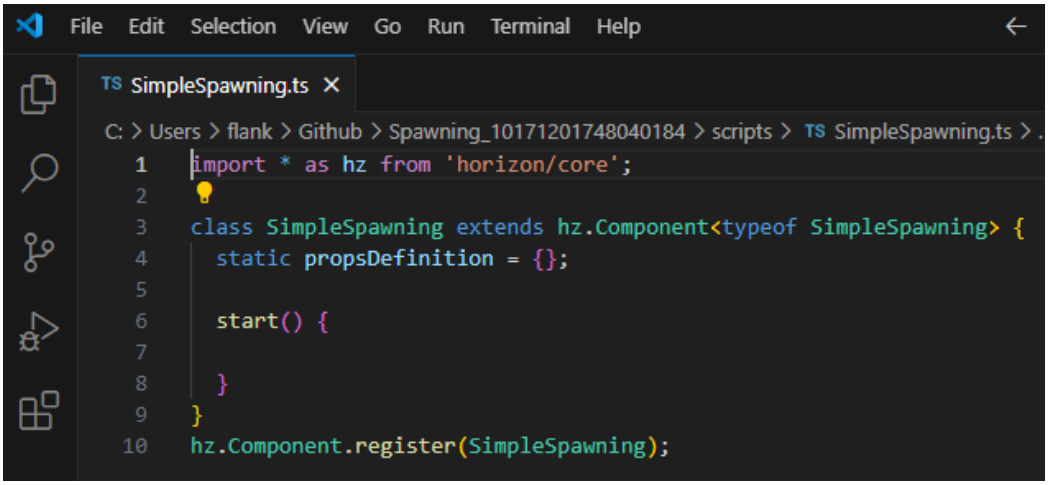
It is time to spawn your asset. So we can see our newly spawned asset, we'll delete the original asset from your viewport/hierarchy by selecting it and pressing the delete key on your keyboard, or right-click and select, *Delete selection*.

a. We need a script to spawn the asset, so let's create one using the *Scripts Panel* and click the plus(+) button. Give it a name and press the enter key on your keyboard.

i. 

b. We can open it in VSCode simply by clicking on it and you should see a window that looks similar to the image below.

i. If you have not installed VSCode or if this does not work for you, view my post on Getting Started: Desktop Editor

ii. 

c. We're going to modify this script to run on a *Trigger Zone* gizmo by first adding the *preStart()* method, connecting to the *OnPlayerEnterTrigger* codeblock event, and redirecting it to our method which we will call *triggerEntered()*. If you are familiar with the *Connect To* codeblock this is very similar.

i. By using the preStart() method to connect to our codeblock event, we ensure our script is ready to hear the event before any actions can be performed in the world. Here is what this looks like:

ii.

```
import * as hz from 'horizon/core';

class SimpleSpawning extends hz.Component<typeof SimpleSpawning> {
  static propsDefinition = {};

  preStart(): void {
    this.connectCodeBlockEvent(
      this.entity,
      hz.CodeBlockEvents.OnPlayerEnterTrigger,
      (enteredBY) => this.triggerEntered(enteredBY)
    )
  }

  start() {

  }

  triggerEntered(enteredBY: hz.Player) {

  }
}
hz.Component.register(SimpleSpawning);
```

d. We need to add a reference to the template asset we created earlier. We're going to reference the asset by its *Asset ID*. To find your asset's Id we need to select the asset from the *Asset Library* and look at the *Property Panel*.

i. 

e. We will take this ID number and create a private variable for it in our script.

i.
```
import * as hz from 'horizon/core';

class SimpleSpawning extends hz.Component<typeof SimpleSpawning> {
  static propsDefinition = {};

  private aBasicShape: hz.Asset = new hz.Asset(BigInt('1059045839366301'));

  preStart(): void {
    this.connectCodeBlockEvent(
      this.entity,
      hz.CodeBlockEvents.OnPlayerEnterTrigger,
      (enteredBY) => this.triggerEntered(enteredBY)
    )
  }
```

f. You’re almost there! Just a few more steps. We nearly have everything we need. Let's add the spawning part. In the triggerEntered() method we created earlier, we need to add the code related to the spawning of our asset. Here is what that looks like. The spawnAsset() method requires 3 parameters. The asset to spawn, the

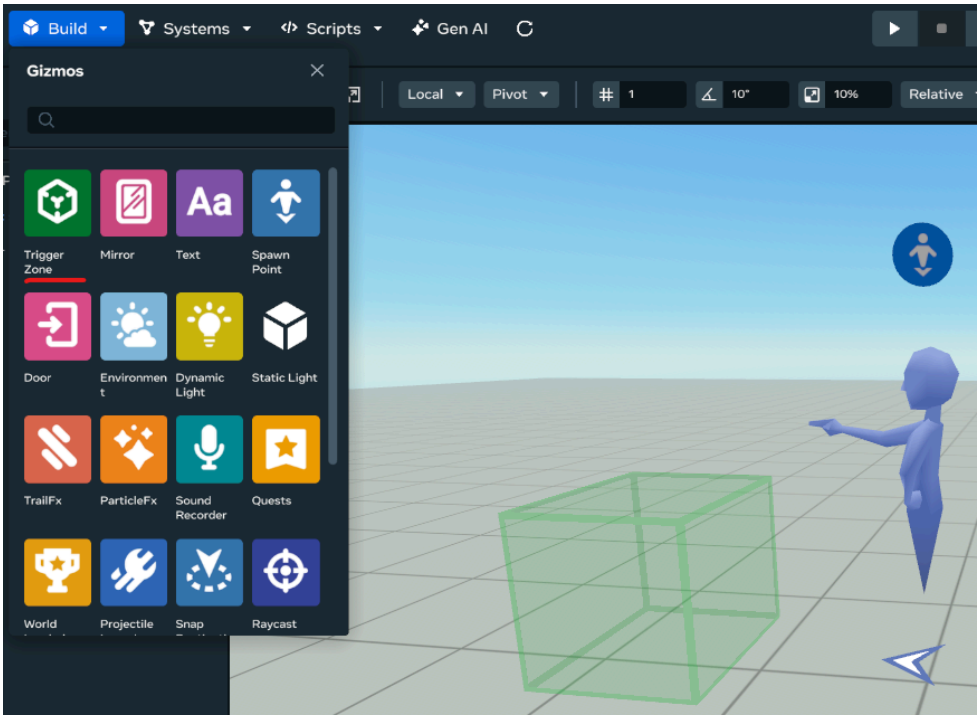
position to spawn it at, the rotation to spawn the asset in, and finally the scale to spawn it.

i.

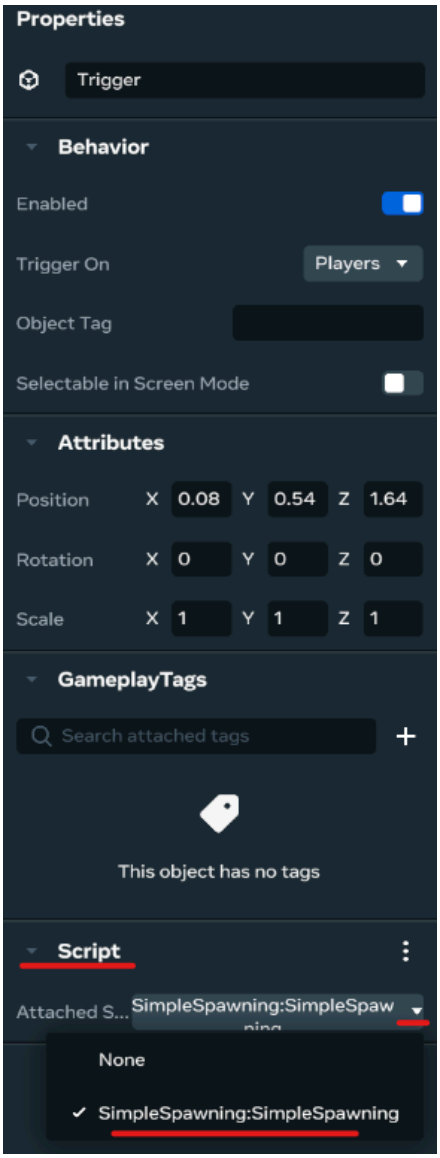
```
20  triggerEntered(enteredBY: hz.Player) {
21
22    this.world.spawnAsset(
23      this.aBasicShape,
24      hz.Vec3.zero,
25      hz.Quaternion.zero,
26      hz.Vec3.one
27    )
28
29  }
```

g. Save your script in VSCode by pressing the *CTRL + S* keys on your keyboard. Go back to the desktop editor and drag a *Trigger Zone* gizmo from the *Build Menu* > *Gizmos* into an area where you can touch it.

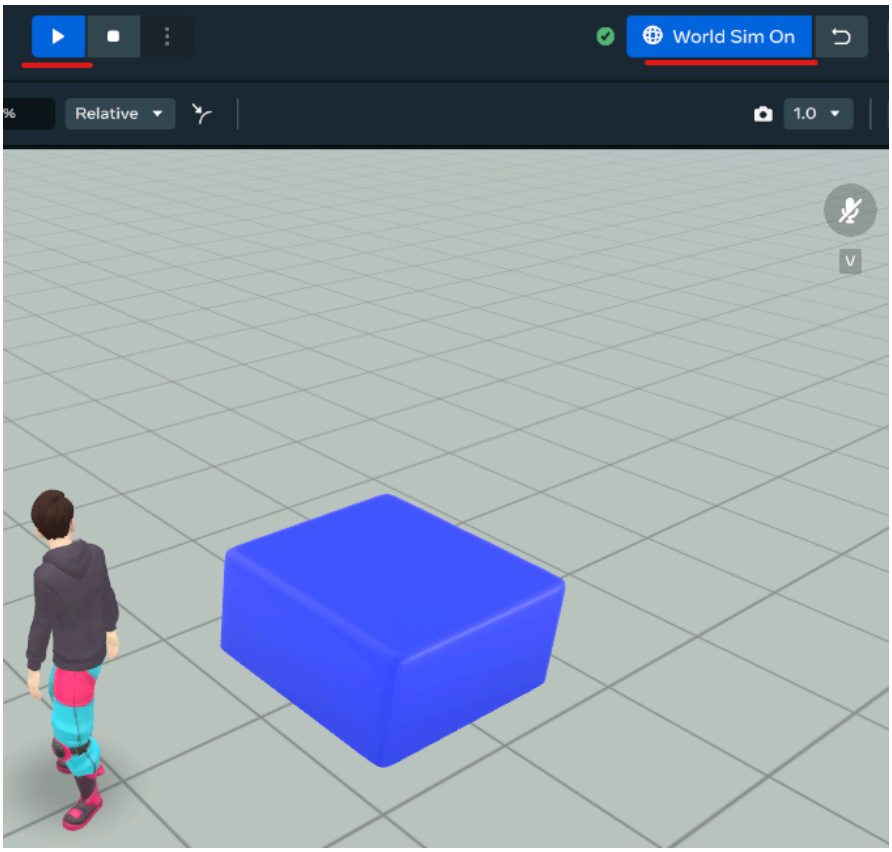
i.



h. Select the *Trigger Zone* gizmo in your viewport and scroll to the bottom of the *Property Panel* on the right. Look for *Script*, hit the dropdown, and find the script you created.

i. 

i. Turn *World Sim On* in the upper right corner of the desktop editor. Spawn yourself into the world with the *Play* button and walk into your *Trigger Zone* gizmo. This should spawn your asset at the center of the world.

i. 

j. Congratulations! You spawned your asset into the world! But your learning doesn't stop here. There is still plenty to know about even simple asset spawning, such as how to spawn an asset in another position, how to delete an asset, and how to utilize the SpawnController. The Meta Learn Center and the Creators Manual are great resources to keep you going! For additional help join me at this week's AMA: Asset Spawning & World Streaming on 3/25/2025 @ 12 PM PST. Register here