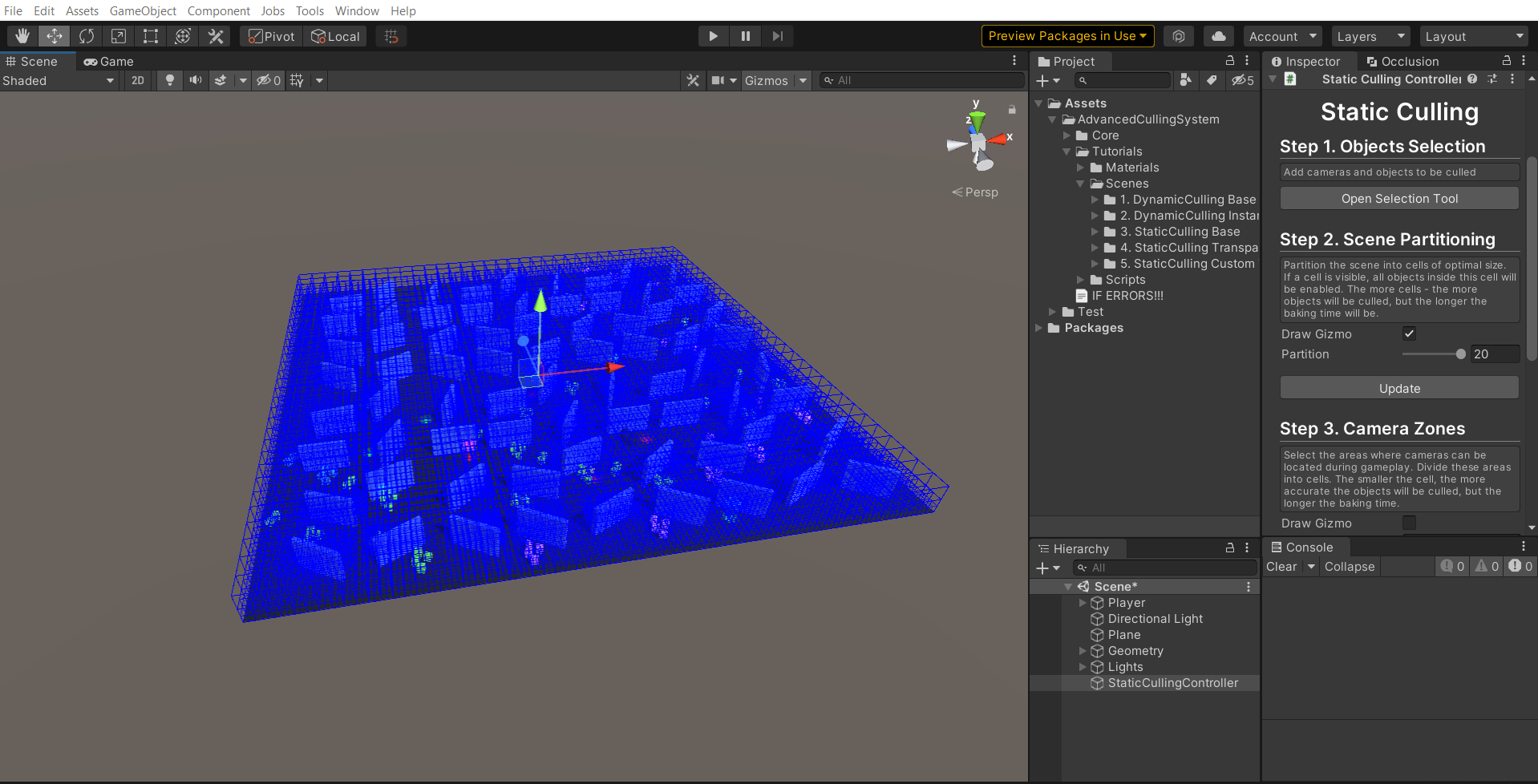
Static Culling base

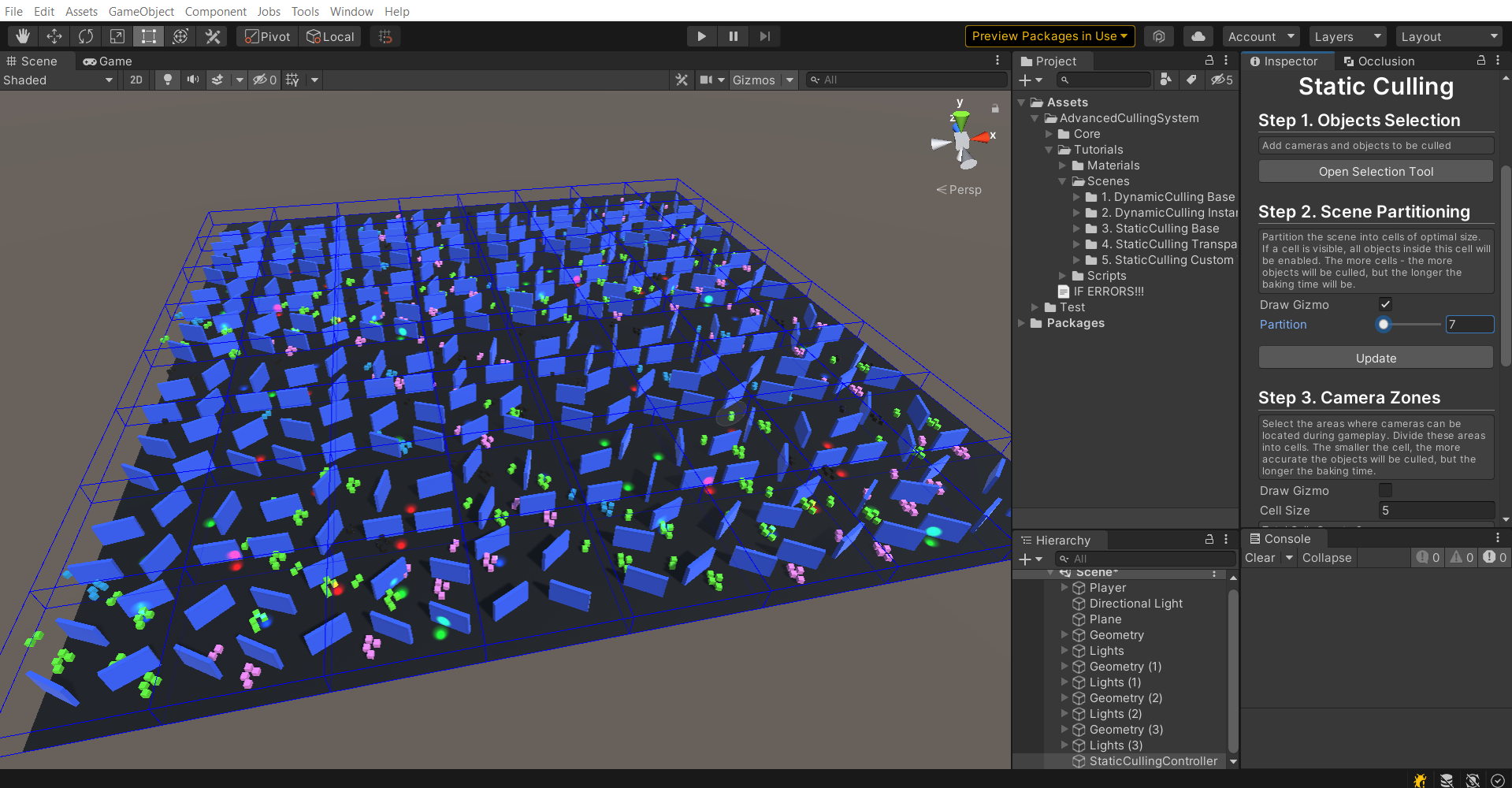
**StaticCulling** is a module that allows **preprocessing the visibility** of objects, and then simply turning them on and off during runtime. This approach offers **deep customization options** and generates less load during runtime compared to Dynamic Culling. It's suitable for **small and medium scenes**, but not recommended for very large ones due to **higher memory requirements** compared to Dynamic Culling.

The StaticCulling solution involves **dividing the scene's space** into three-dimensional cells. Then, it calculates which objects are **visible from each cell**, and the data about these visible objects are **recorded and stored**.

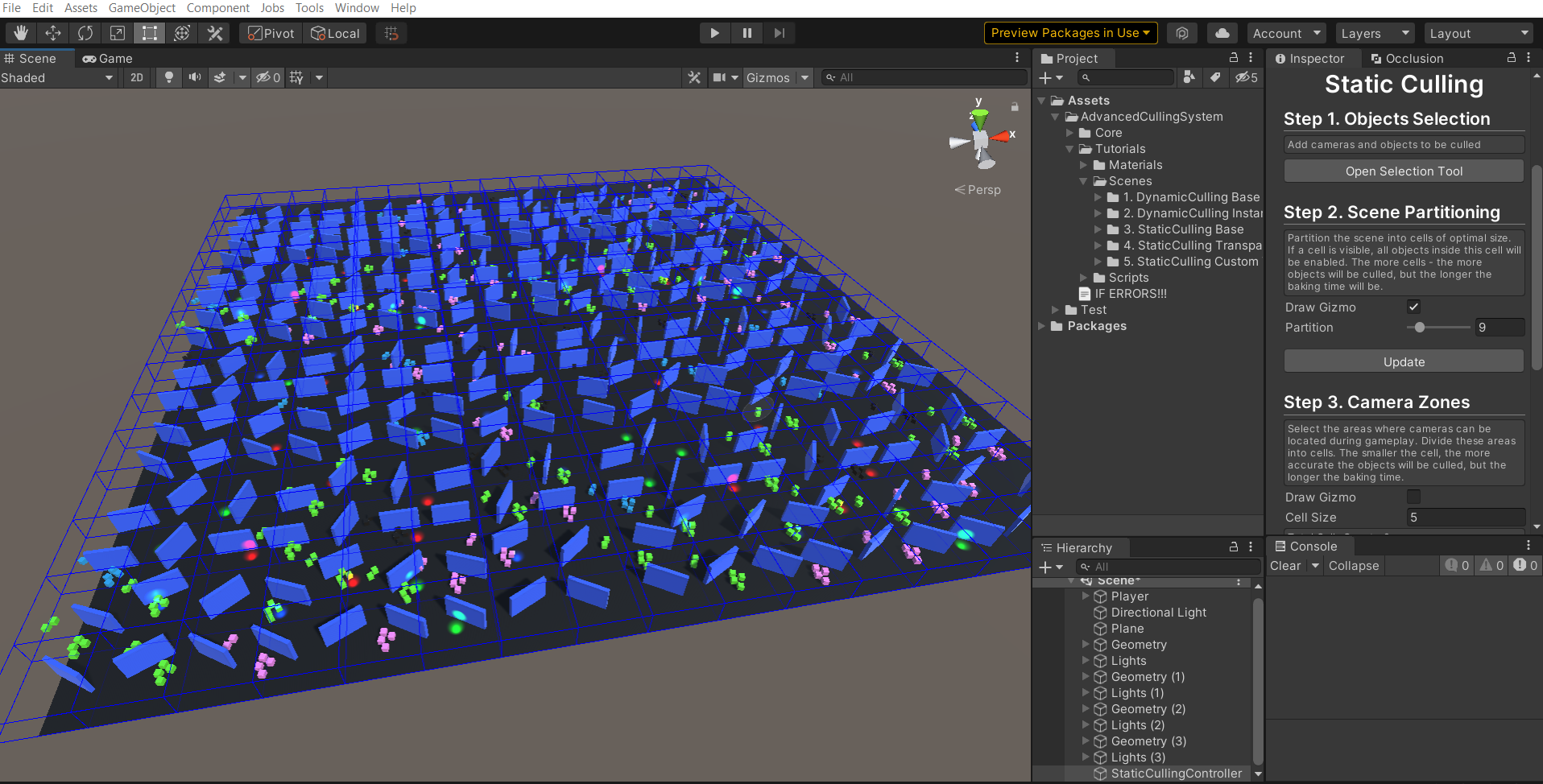
1. First, **create a StaticCullingController**. To do this, go to 'Tools -> NGSTools -> Advanced Culling System -> Static'.
2. **Object Selection**. Open the window **by clicking "Open Selection Tool"** and add the objects you want to cull, following the prompts in this window.
3. **Scene Partitioning**. Partition the objects in the scene into cells. In order to speed up the processing of the scene, the **visibility of a cell is determined first**, and then the visibility of objects in that cell. You need to strike a **balance between the number of cells** and the number of objects in that cell. Try to have up to **ten to twenty objects in a cell**, but do not select a Partition value that is **too large**. An example is shown in the picture below



Too big “Partition” value



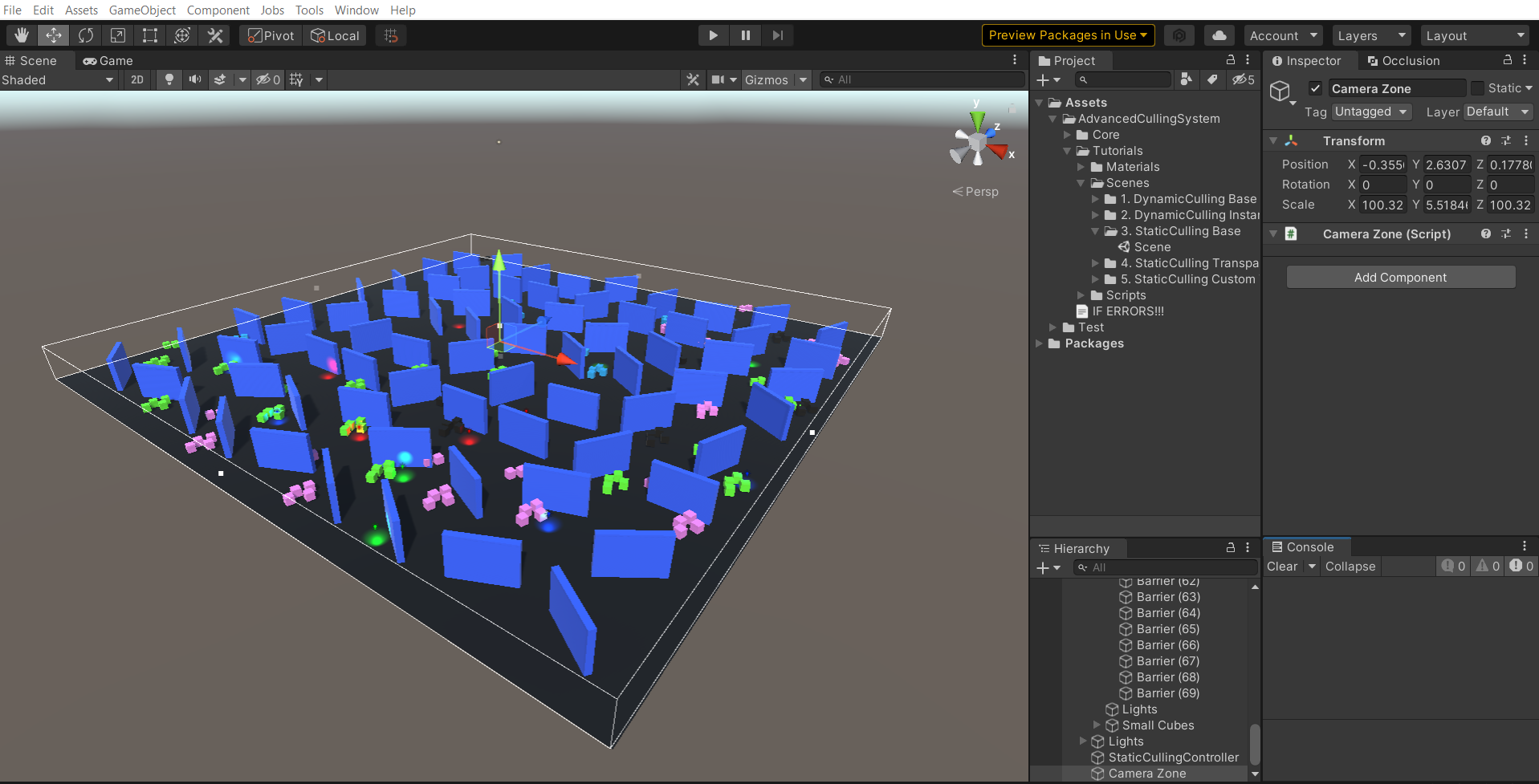
Not enough “Partition” value



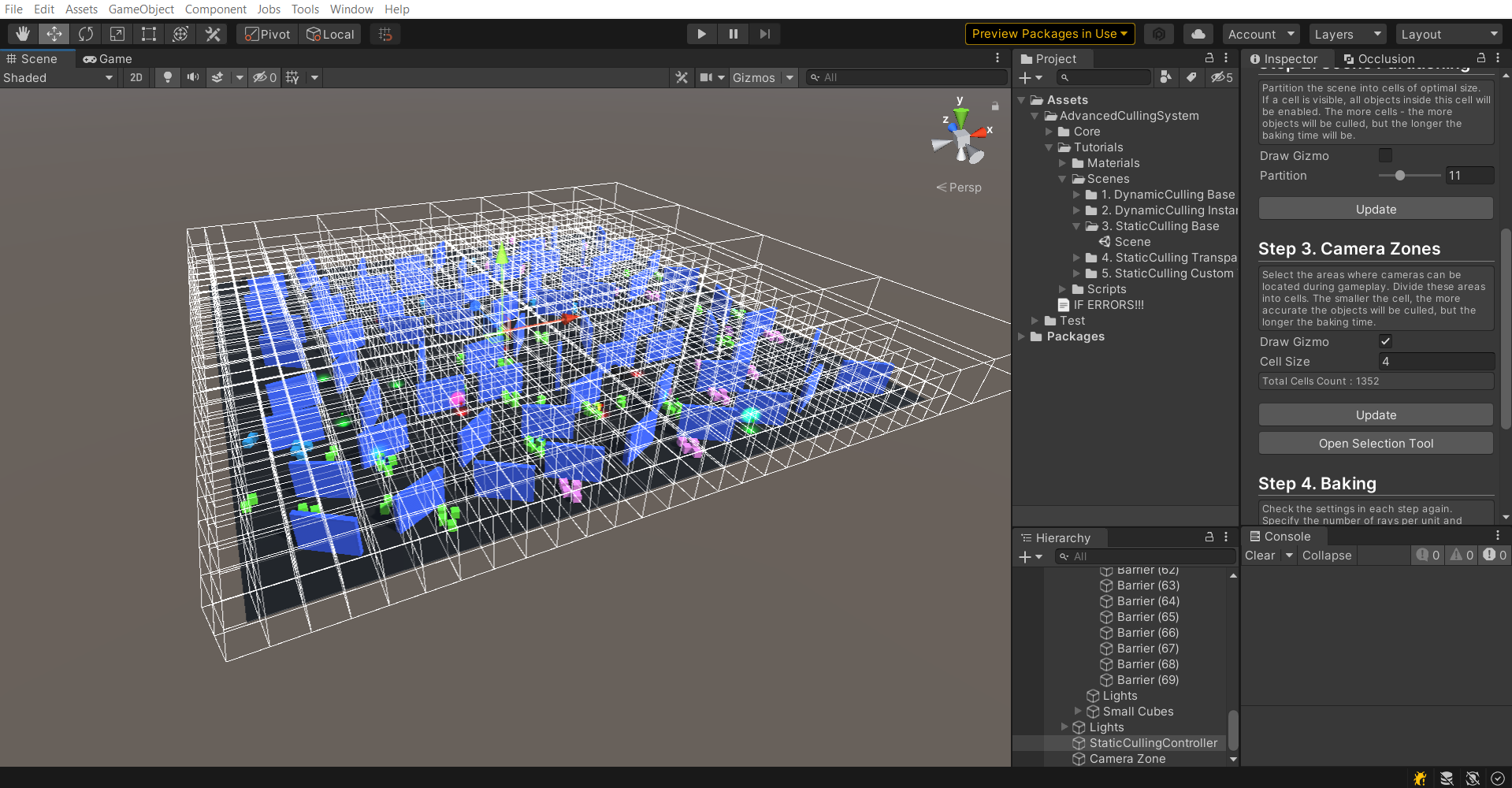
Optimal value

1. **Camera Zones**. Specify the areas where the **camera can be located during gameplay**. The more precisely you specify these zones, the **less memory will be used** to store information about visible objects.

**Select the optimal Cell Size**. All Camera Zones will be **divided into cells** of the size you specify. Bake will determine which objects **are visible from each cell**, then this data will be **saved and used in runtime**. The smaller Cell Size you choose - the more objects will be culled, but the **more memory will be used**.



Optimal CameraZone



CameraZone divided into cells

1. **Baking**. Specify the RaysPerUnit value. The larger the value, the **more accurate the visibility** of objects will be determined, but the **longer preprocessing** will take.
2. **StaticCullingCamera**. Cameras that will participate in object culling must have a **"StaticCullingCamera" component**. If you specified these cameras in Step1, the component **will be attached to them automatically.**

Turn on DrawCells to see which cells are **now affecting the camera**.

Increase the **tolerance** value - if you want to take information from more cells **near the camera**. The more tolerance - the **less artifacts**, but **the more visible objects.**

1. **StaticCullingSource**. The objects you select in Step 1 will **automatically have the StaticCullingSource component** attached to them. Make changes to this component to **control the baking and culling process.**