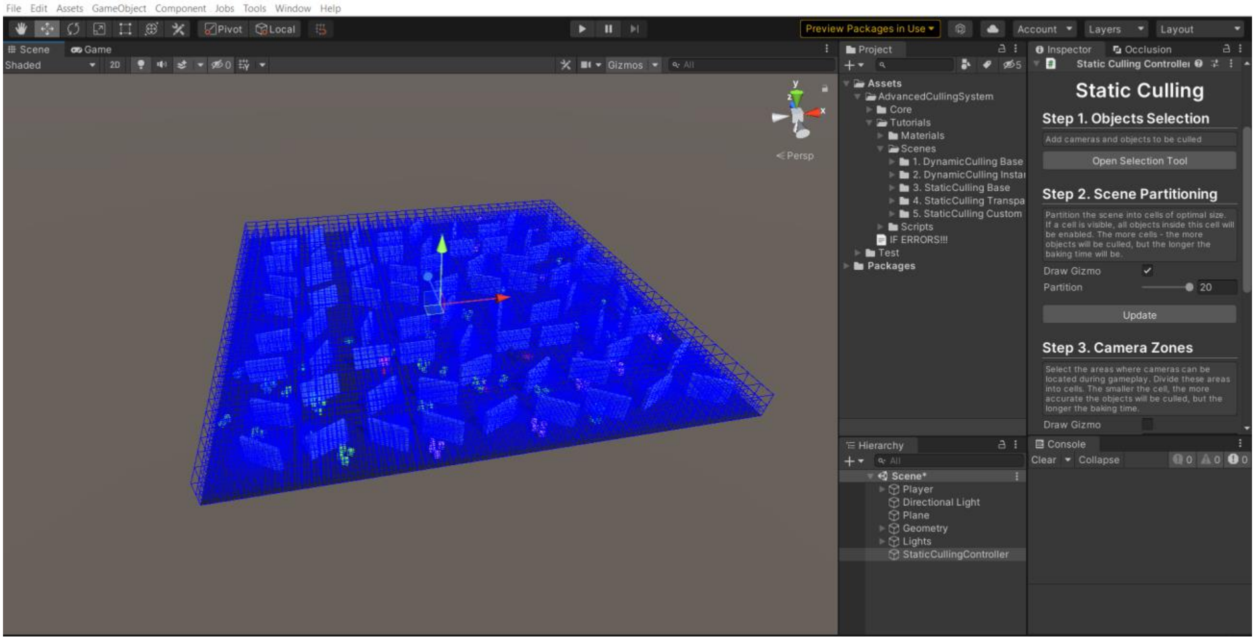
Static Culling base

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



Static Culling

Step 1. Objects Selection

Add cameras and objects to be culled

Open Selection Tool

Step 2. Scene Partitioning

Partition the scene into cells of optimal size. If a cell is visible, all objects inside this cell will be enabled. The more cells - the more objects will be culled, but the longer the baking time will be.

Draw Gizmo ☑

Partition 20

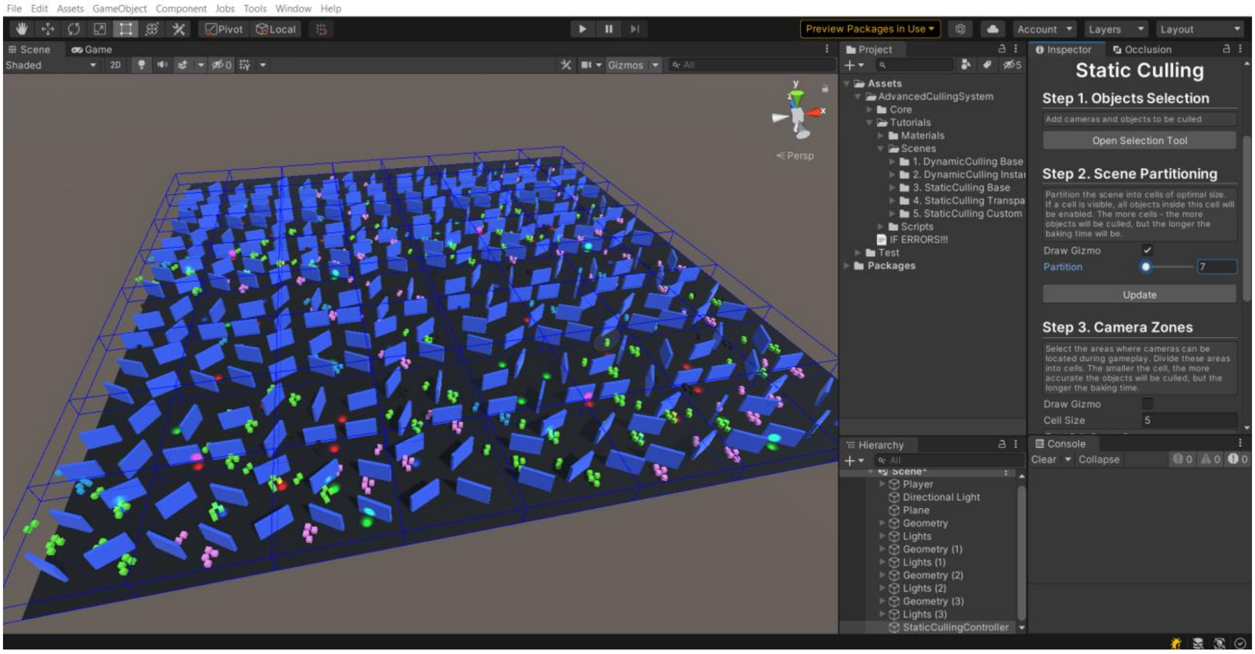
Update

Step 3. Camera Zones

Select the areas where cameras can be located during gameplay. Divide these areas into cells. The smaller the cell, the more accurate the objects will be culled, but the longer the baking time.

Draw Gizmo ☐

Too big “Partition” value



Static Culling

Step 1. Objects Selection

Add cameras and objects to be culled

Open Selection Tool

Step 2. Scene Partitioning

Partition the scene into cells of optimal size. If a cell is visible, all objects inside this cell will be enabled. The more cells - the more objects will be culled, but the longer the baking time will be.

Draw Gizmo ☑

Partition 7

Update

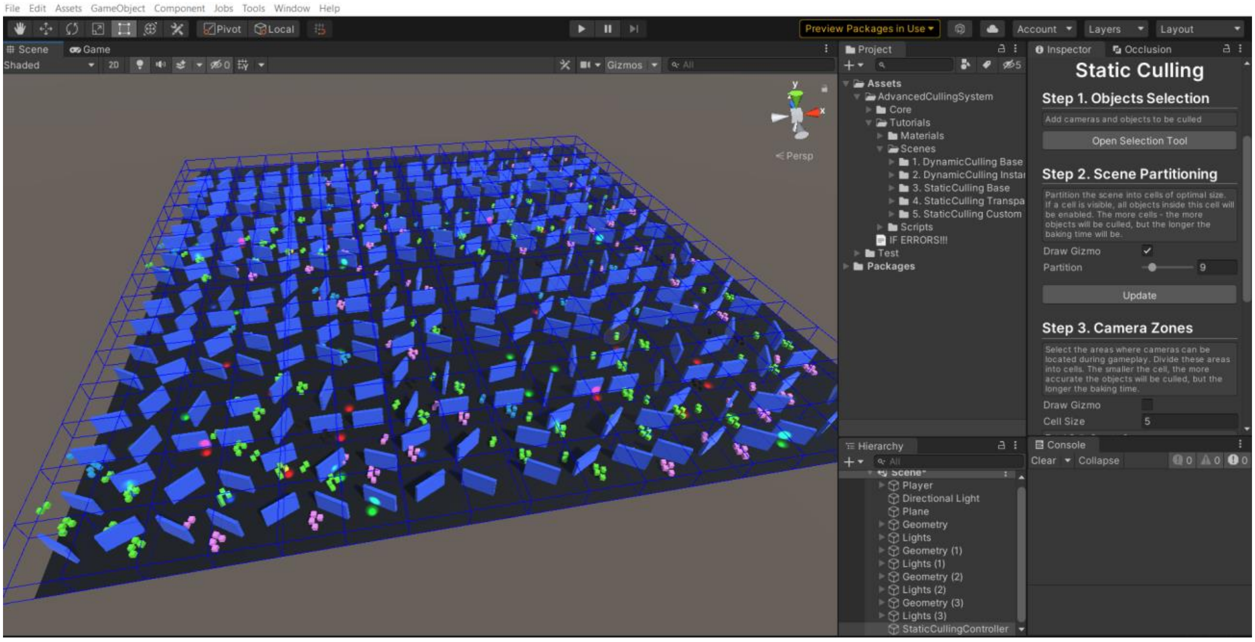
Step 3. Camera Zones

Select the areas where cameras can be located during gameplay. Divide these areas into cells. The smaller the cell, the more accurate the objects will be culled, but the longer the baking time.

Draw Gizmo □

Cell Size 5

Not enough “Partition” value



Static Culling

Step 1. Objects Selection

Add cameras and objects to be culled

Open Selection Tool

Step 2. Scene Partitioning

Partition the scene into cells of optimal size. If a cell is visible, all objects inside this cell will be enabled. The more cells - the more objects will be culled, but the longer the baking time will be.

Draw Gizmo ☑

Partition 9

Update

Step 3. Camera Zones

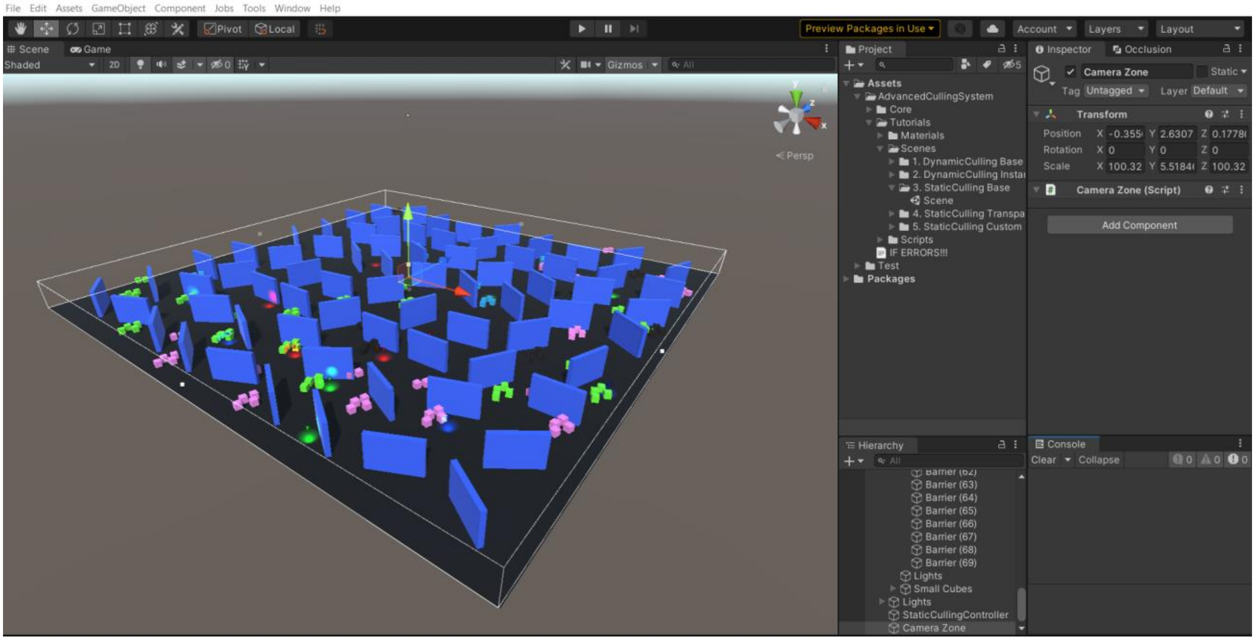
Select the areas where cameras can be located during gameplay. Divide these areas into cells. The smaller the cell, the more accurate the objects will be culled, but the longer the baking time.

Draw Gizmo □

Cell Size 5

Optimal value

4. **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**.



Camera Zone ☑ Static

Tag Untagged Layer Default

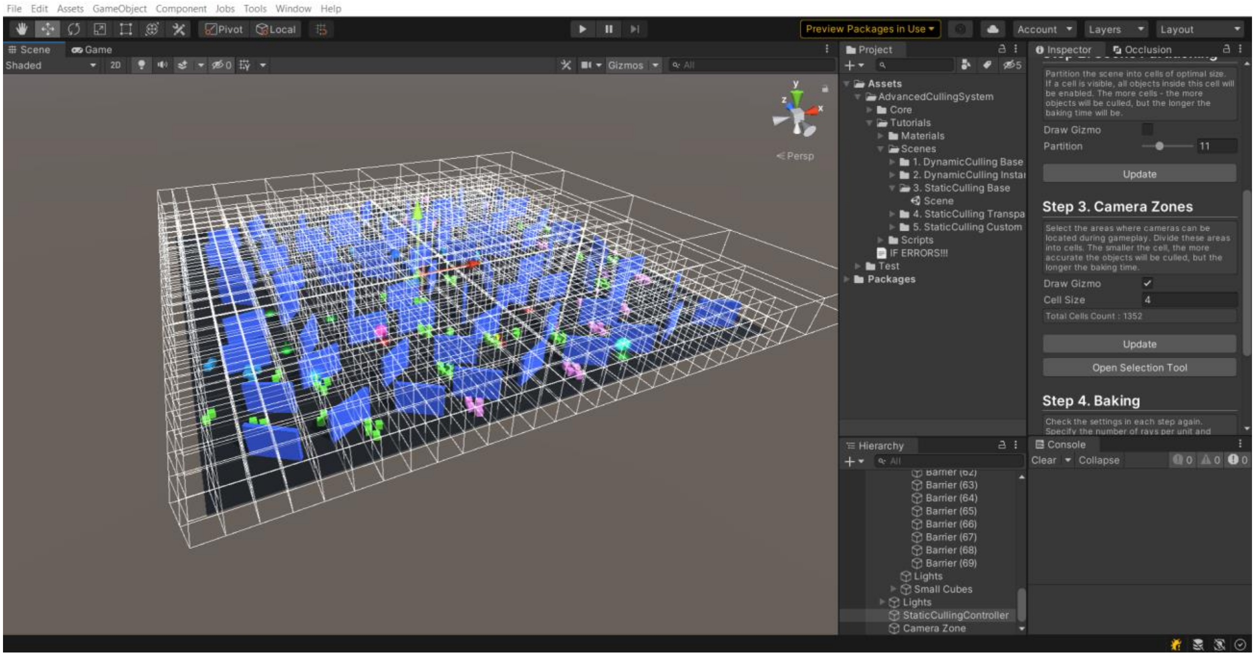
Transform

	X	Y	Z
Position	-0.355	2.6307	0.1778
Rotation	0	0	0
Scale	100.32	5.5184	100.32

Camera Zone (Script)

Add Component

Optimal CameraZone



Partition the scene into cells of optimal size. If a cell is visible, all objects inside this cell will be enabled. The more cells - the more objects will be culled, but the longer the baking time will be.

Draw Gizmo □

Partition 11

Update

Step 3. Camera Zones

Select the areas where cameras can be located during gameplay. Divide these areas into cells. The smaller the cell, the more accurate the objects will be culled, but the longer the baking time.

Draw Gizmo ☑

Cell Size 4

Total Cells Count : 1352

Update

Open Selection Tool

Step 4. Baking

Check the settings in each step again. Specify the number of rays per unit and

CameraZone divided into cells

5. **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.
6. **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.**

7. **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.**