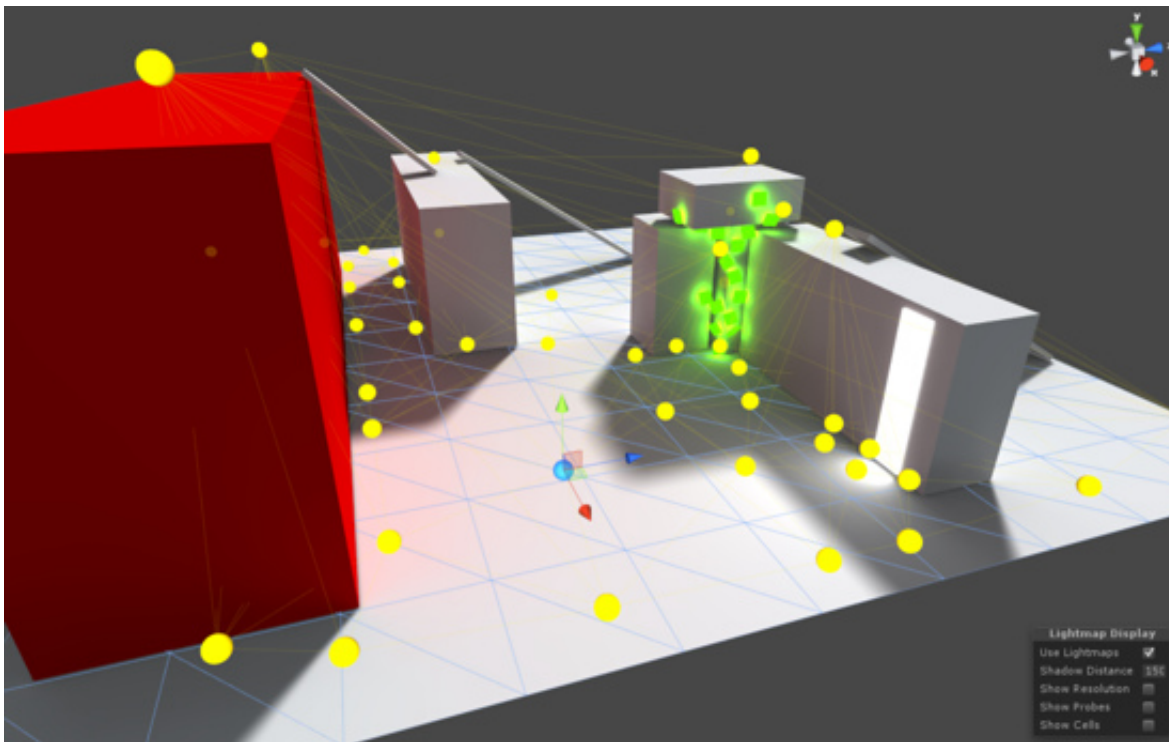


Light Probes

Although lightmapping adds greatly to the realism of a scene, it has the disadvantage that non-static objects in the scene are less realistically rendered and can look disconnected as a result. It isn't possible to calculate lightmapping for moving objects in real time but it is possible to get a similar effect using **light probes**. The idea is that the lighting is sampled at strategic points in the scene, denoted by the positions of the probes. The lighting at any position can then be approximated by interpolating between the samples taken by the nearest probes. The interpolation is fast enough to be used during gameplay and helps avoid the disconnection between the lighting of moving objects and static lightmapped objects in the scene.

The Light Probe Group component (menu: **Component -> Rendering -> Light Probe Group**) can be added to any available object in the scene. The inspector can be used to add new probes to the group. The probes appear in the scene as yellow spheres which can be positioned in the same manner as GameObjects. Selected probes can also be duplicated with the usual keyboard shortcut (ctrl+d / cmd+d).



Choosing Light Probe positions

Remember to place probes where you want to sample light or sample darkness. The probes need to form a volume within

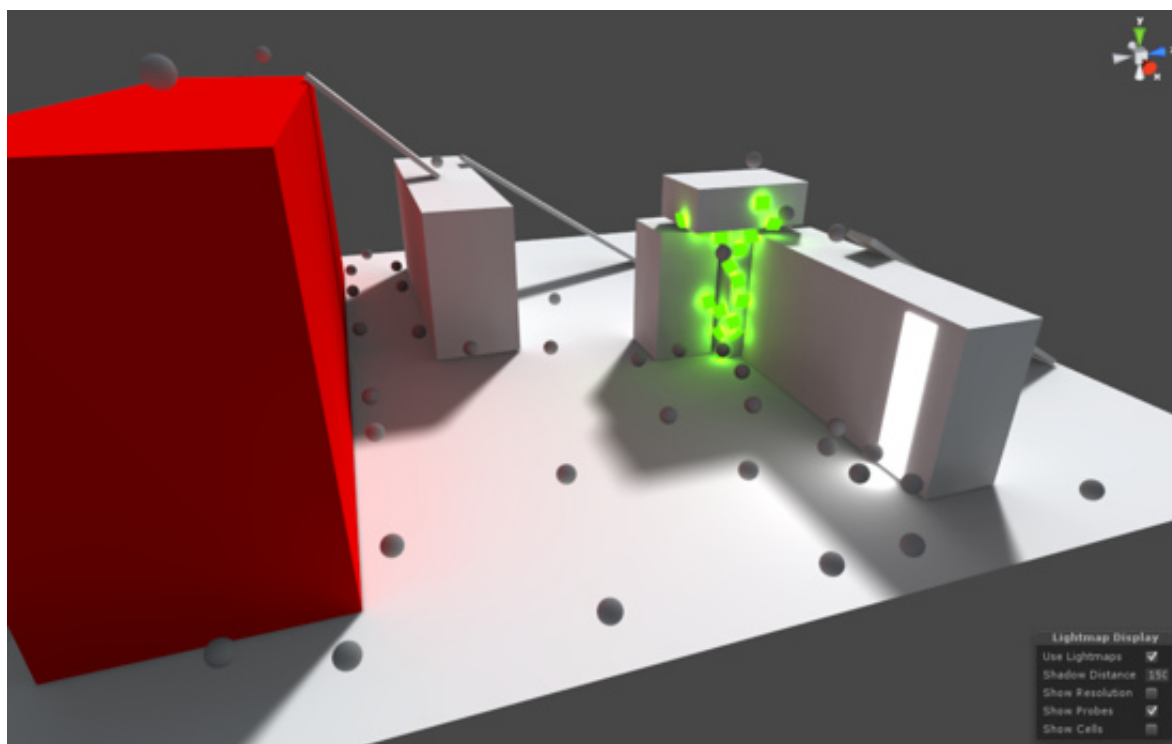
the scene for the space subdivision to work properly.

The simplest approach to positioning is to arrange them in a regular 3D grid pattern. While this setup is simple and effective, it is likely to consume a lot of memory (each light probe is essentially a spherical, panoramic HDR image of the view from the sample point). It is worth noting that probes are only needed for regions that players, NPCs or other dynamic objects can actually move to. Also, since lighting conditions

are interpolated for positions between probes, it is not necessary to use lots of them across areas where the light doesn't change very much. For example, a large area of uniform shadow would not need a large number of probes and neither would a brightly lit area far away from reflective objects. Probes are generally needed where the lighting conditions change abruptly, for instance at the edge of a shadow area or in places where pieces of scenery have different colors.

In some cases, the infrastructure of the game can be useful in choosing light probe positions. For example, a racing game typically uses waypoints around the track for AI and other purposes. These are likely to be good candidates for probe positions and it would likely be straightforward to set these positions from an editor script. Similarly, navigation meshes typically define the areas that can be reached by players and these also lend themselves to automated positioning of probes.

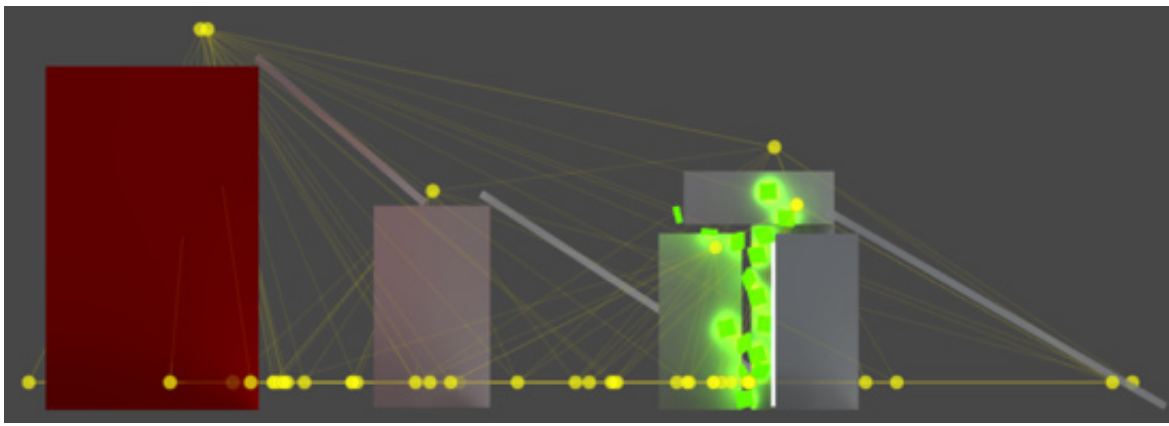
Here light probes have been baked over surfaces where our characters can walk on, but only where there are interesting lighting changes to capture:



Flat 2D levels

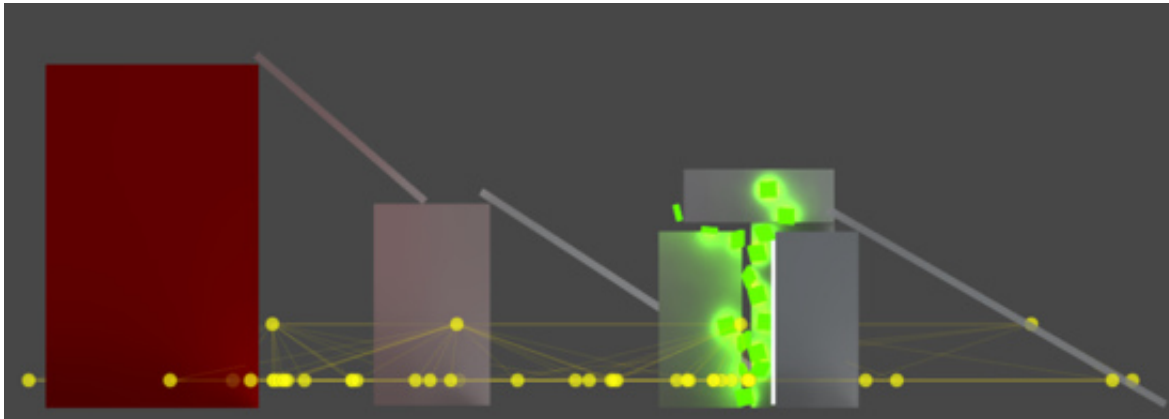
As it is now, the light probe system can't bake a completely flat probe cloud. So even if all your characters move only on a plane, you still have

to take care to position at least some probes in a higher layer, so that a volume is formed and interpolation can work properly.



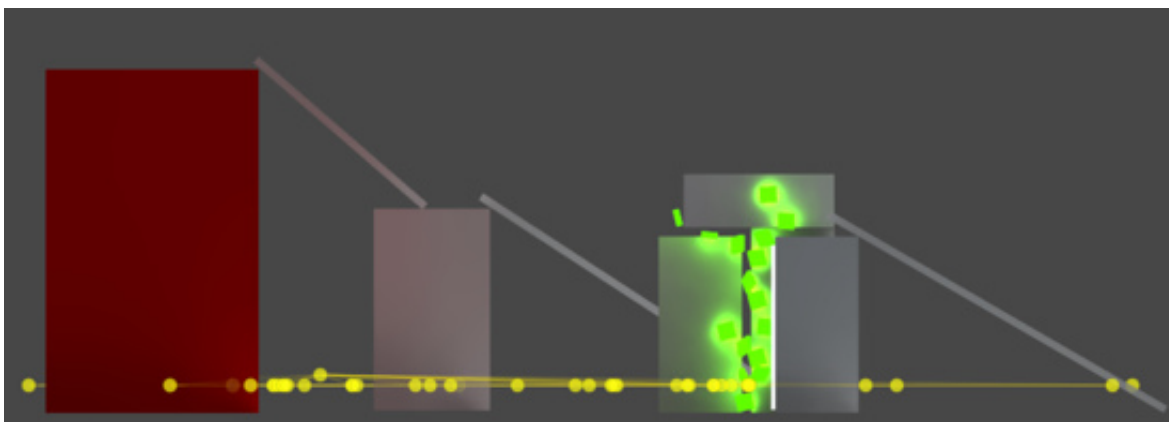
boxes, so it's good to sample lighting up there as well.

Good: This is the original probe placement. The characters can move up the ramps and up onto the



a little bit higher, so that a volume is formed and thin cell are avoided.

Good: Here we assume the characters can only move on the plane. Still, there's a couple of probes placed



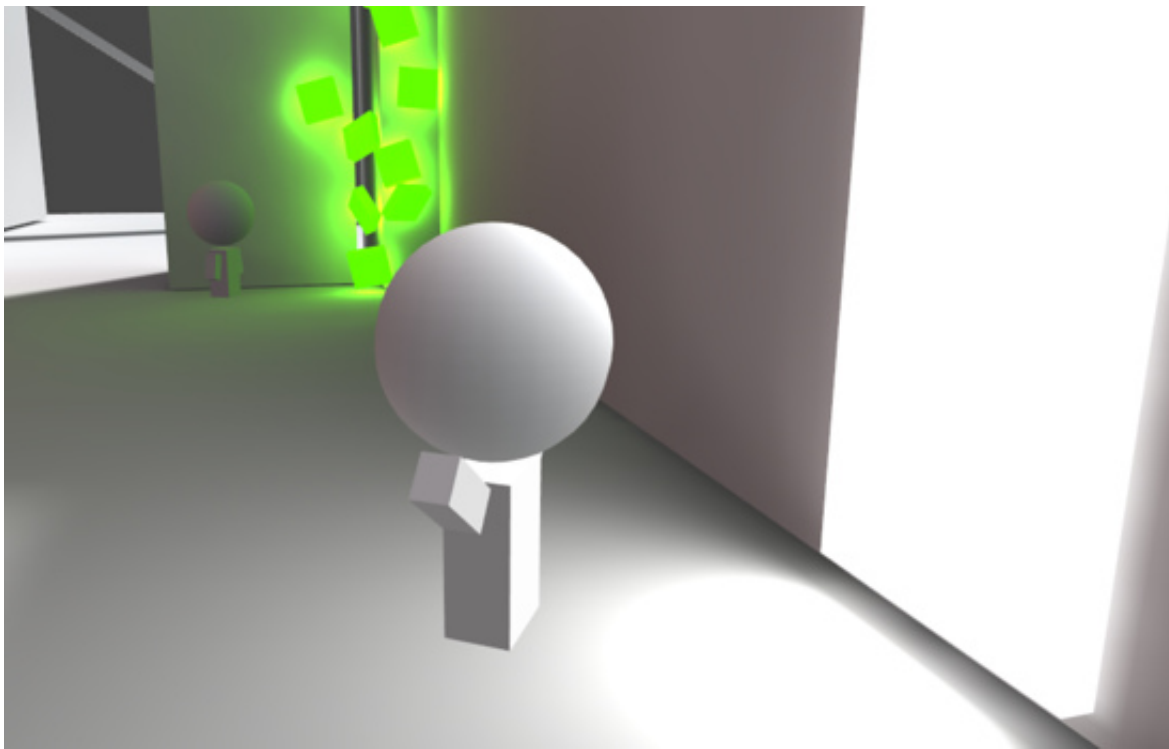
interpolation results.

Bad: The probes are placed too flat, which creates really long and thin cells and produces unintuitive in-

Using Light Probes

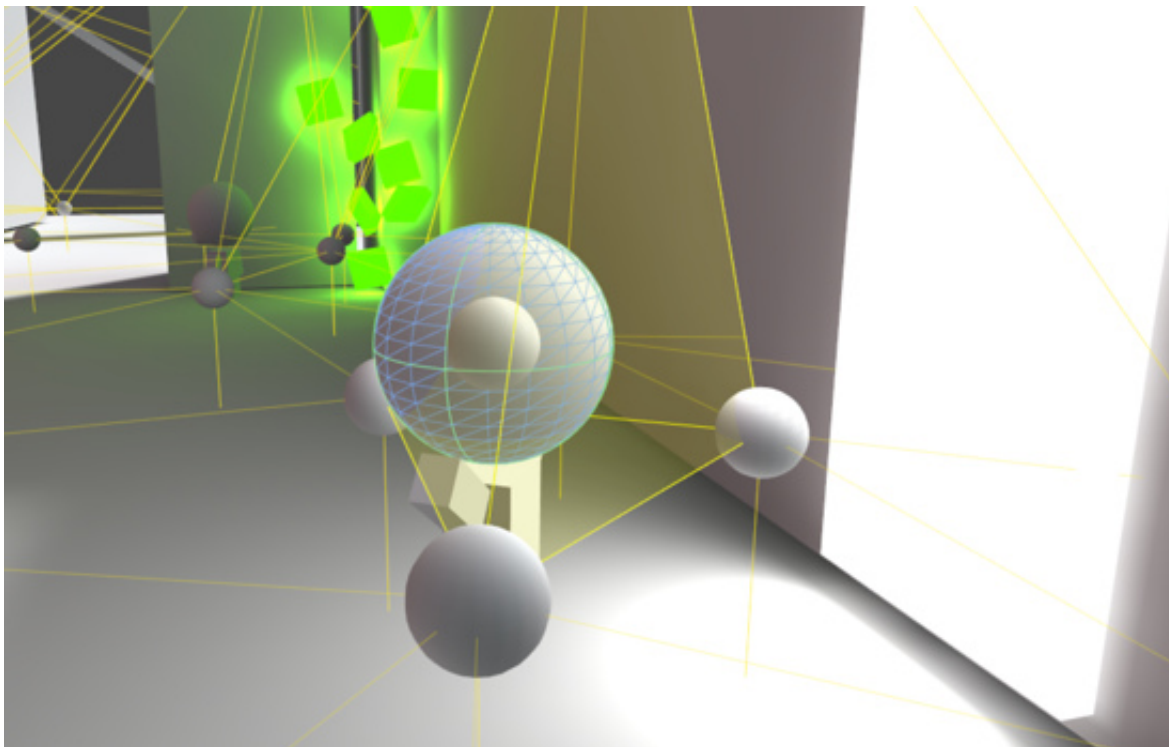
To allow a mesh to receive lighting from the probe system, you should enable the Use Light Probes option on its Mesh Renderer:

The probe interpolation requires a point in space to represent the position of the mesh that is receiving light. By default, the centre of the mesh's bounding box is used but it is possible to override this by dragging a Transform to the Mesh Renderer's Light Probe Anchor property (this Transform's position will be used as the interpolation point instead). This may be useful when an object contains two separate adjoin-



ing meshes; if both meshes are lit individually according to their bounding box positions then the lighting will be discontinuous at the place where they join. This can be prevented by using the same Transform (for example the parent or a child object) as the interpolation point for both Mesh Renderers.

When an object using light probes is the active selected object in the Light Probes Scene View mode, its interpolated probe will be rendered on top of it for preview. The interpolated probe is the one used for rendering the object and is connected with 4 thin blue lines (3 when outside of the probe volume) to the probes it is being interpolated between:



Dual Lightmaps vs. Single Lightmaps mode

In Single Lightmaps mode all static lighting (including lights set to 'Auto' lightmapping mode) is

baked into the light probes.

In Dual Lightmaps mode light probes will store lighting in the same configuration as 'Near' lightmaps, i.e. full illumination from sky lights, emissive materials, area lights and 'Baked Only' lights, but only indirect illumination from 'Auto' lights. Thanks to that the object can be lit in real-time with the 'Auto' lights and take advantage of dynamic elements such as real-time shadows, but at the same time receive indirect lighting added to the scene by these lights.

Page last updated: 2012-10-16