

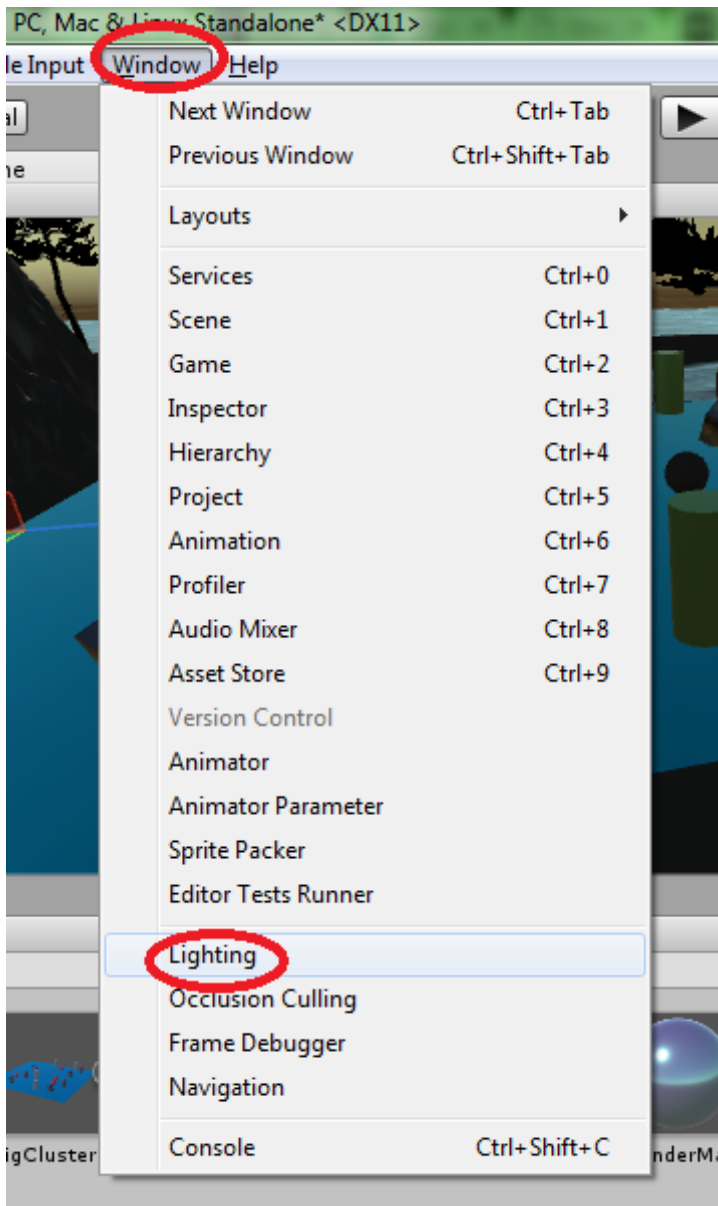
# Unity Tutorials 5 – Lighting and Cameras

## I – Lights

### Getting Rid of All Light

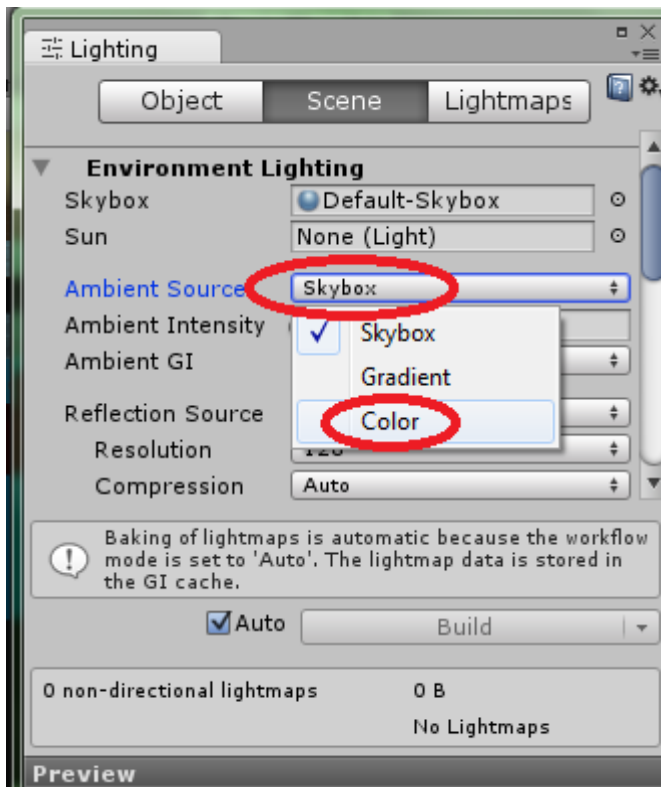
In order to see the effects of lighting properly we will have to get rid of the default light. Whenever you make a new scene you are given a default Main Camera and a default Directional Light. Try deleting the Directional Light.

You can see that your scene is still dimly lit. This is due to the **Ambient Light**. Go to “Window” and “Lighting”.



## Unity Tutorials 5 – Lighting and Cameras

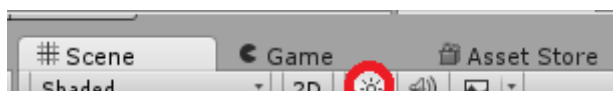
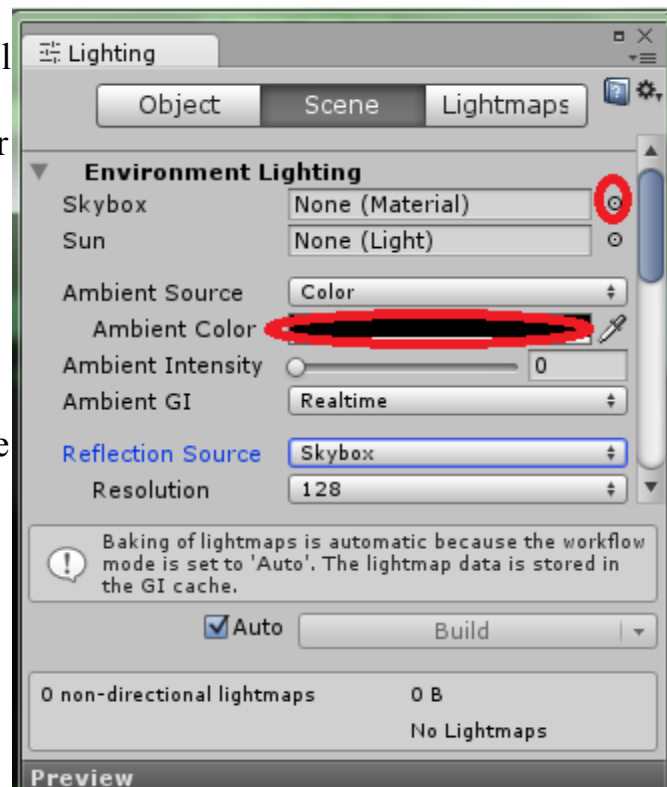
Right now our Ambient Source is the **skybox**. Try changing it to “Color”.



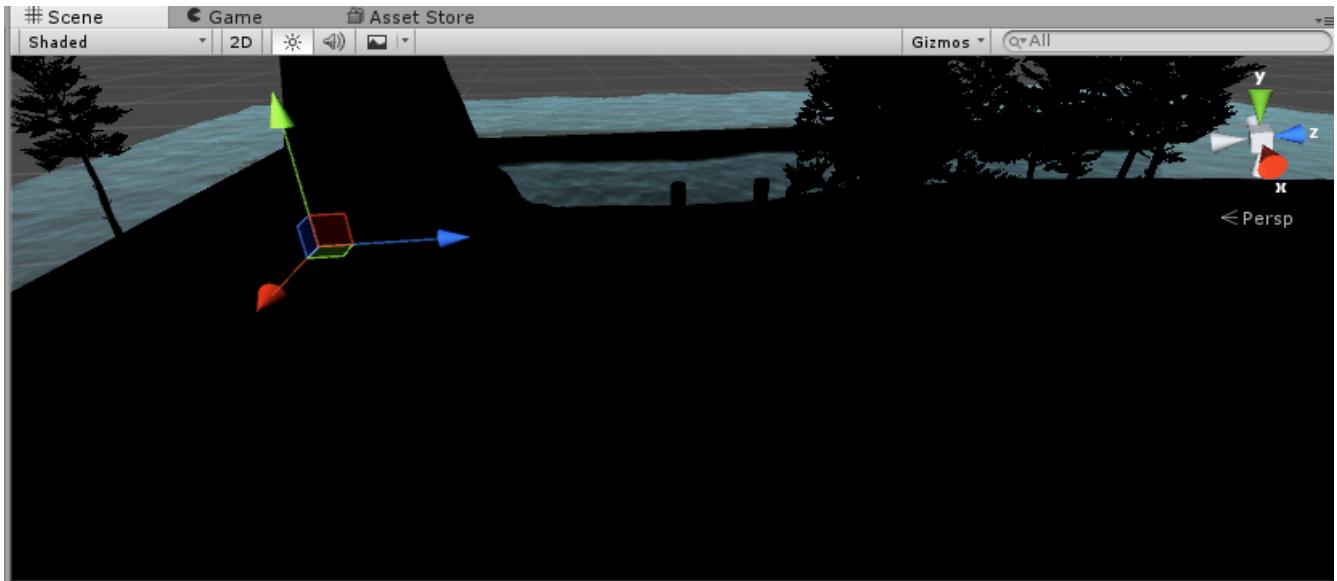
**Ambient Light:** Omnipresent light that does not have a source.

**Skybox:** An image that is wrapped around a scene to simulate a sky. Traditionally, scenes are enclosed in a cube shape with a texture applied to the inside.

Now to get rid of all light sources, click on the small circle next to “Skybox” and scroll up to the top of the list that comes up. Double-click on “None”. Now change your Ambient Color to pure black using the colour picker. You could change the Reflection Source to “Custom” but since we no longer have a Skybox that will not be necessary. Finally, to see your scene with the same lighting that the player will see, click on the sun button at the top of the scene view to turn on “scene lighting”. Without scene lighting on there will be an ambient light in the scene view as if its camera had a light attached to it.

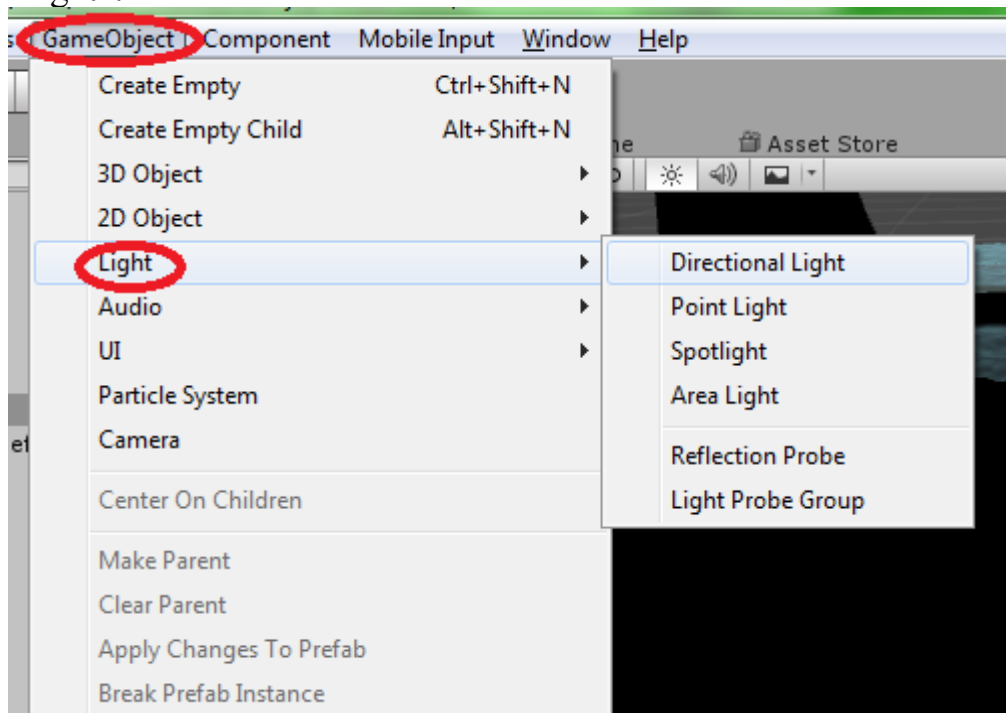


## Unity Tutorials 5 – Lighting and Cameras



### Directional Lights

Now let's try making some lights. Lights can be found under “GameObject” and “Light”.

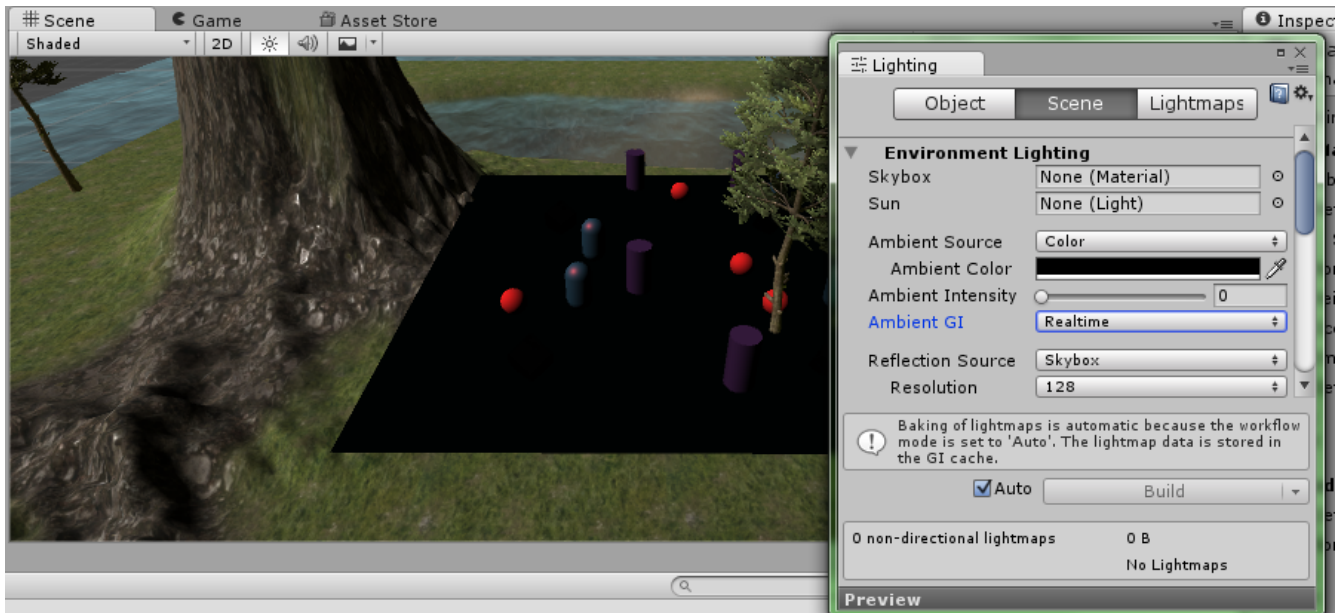


Create a “Directional Light”.

## Unity Tutorials 5 – Lighting and Cameras

Directional lights are similar to ambient lights in that they are generally used to cover an entire scene. They represent large and distant light sources, like the sun.

Even with such a large light, you may notice that any totally metallic or smooth materials do not reflect light at all. This is because there is no Skybox, which is the default reflection source.

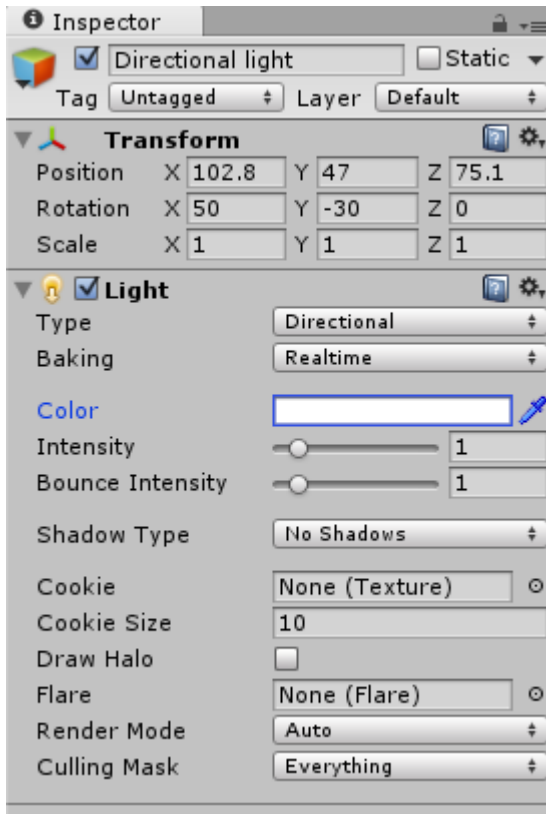


Before we set up a new reflection source, take a look at the inspector for the directional light.

You may notice that changing the position of the light will not change anything. Try changing the rotation instead. Remember that it is the direction of directional lights that make a difference. Even though it has a position in the scene, the source of the light is “far away” enough that moving it won't make a difference.

You can use “Type” to change the type of the light without having to make a new one. You can use “Color” to change the colour of the light. “Intensity” dictates how bright the light is. “Bounce Intensity” is the intensity of indirect light. Indirect light is basically “second-hand” light that has bounced from one object to another. For example, imagine you have a ceiling light with a bowl-shaped fixture. The ceiling above will have a bright circle of light on it. The rest of the room is lit up by the light that has bounced off the ceiling.

## Unity Tutorials 5 – Lighting and Cameras

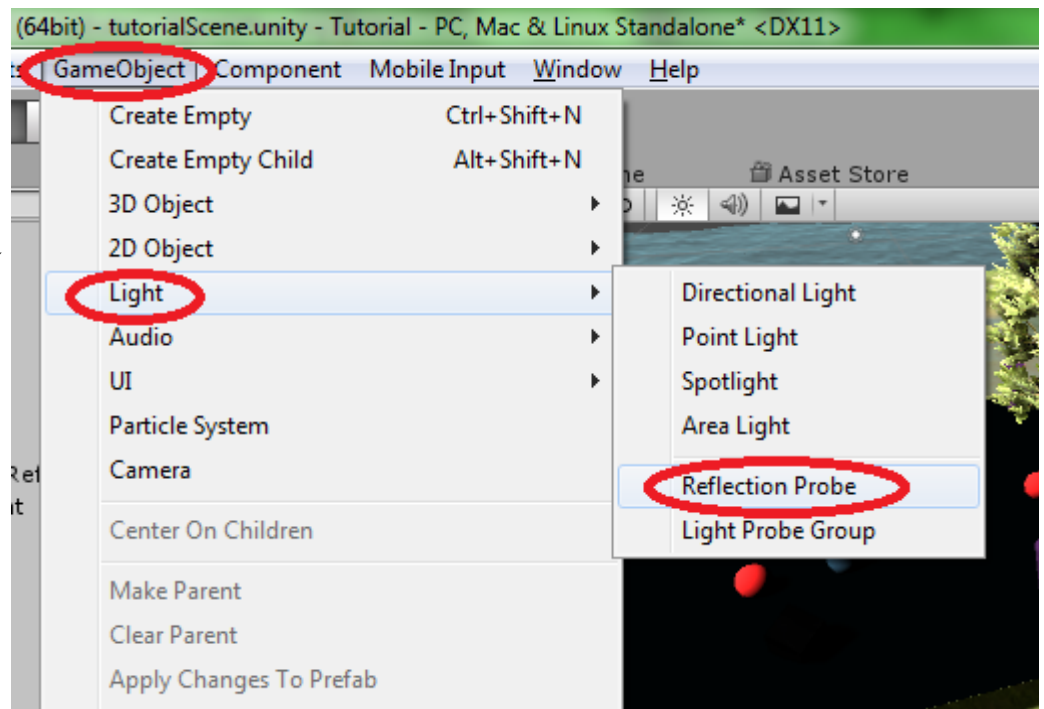


Take a look at the “Shadow Type”. Right now it is set to “No Shadows”, which is not realistic. Change it to “Hard Shadows” or “Soft Shadows”, depending on your preference. Soft shadows are blurred.

### Reflection Probes

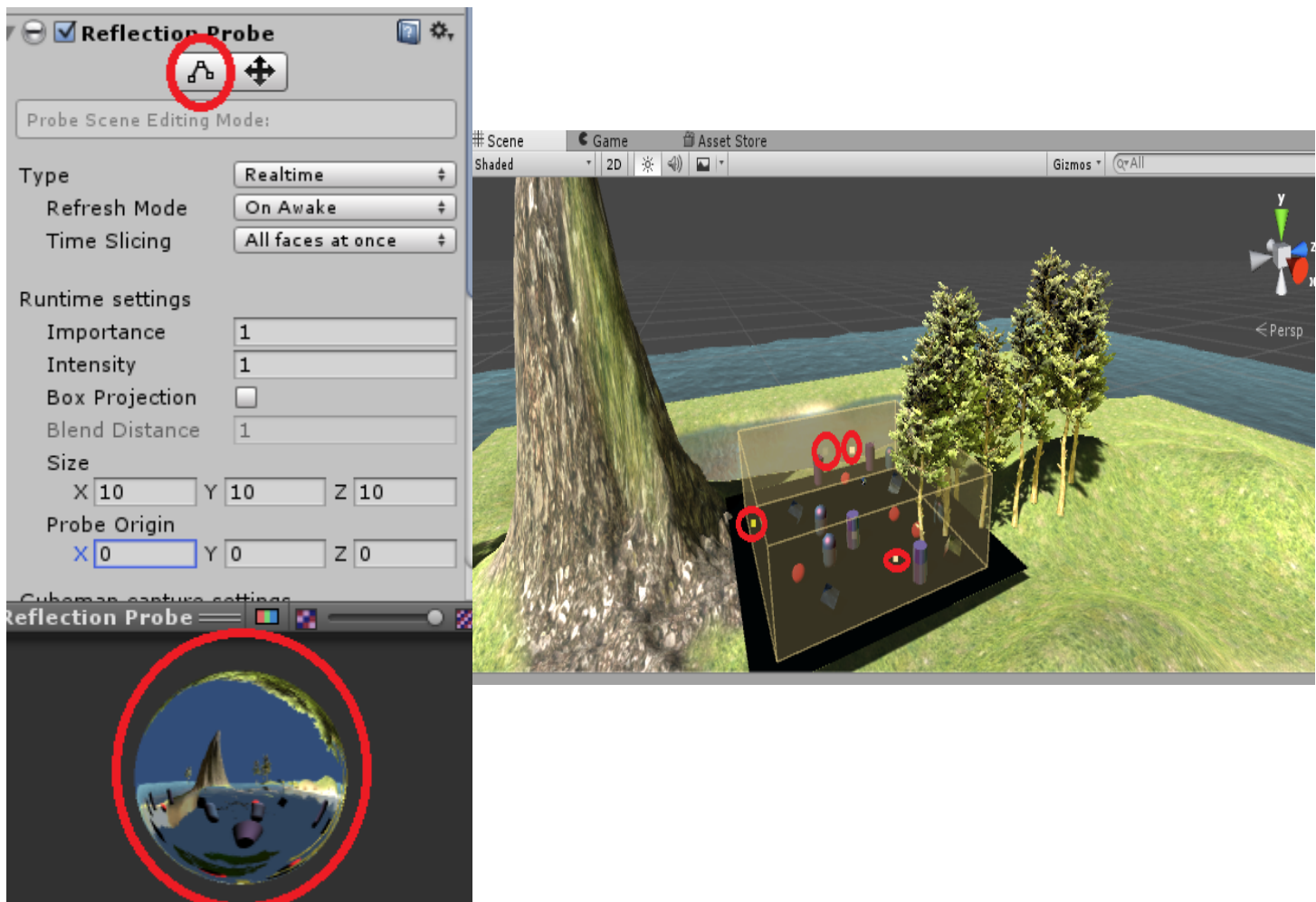
Under “GameObject” and “Light” again, click “Reflection Probe”.

The reflection probe is like a 360 degree camera that captures everything around it.



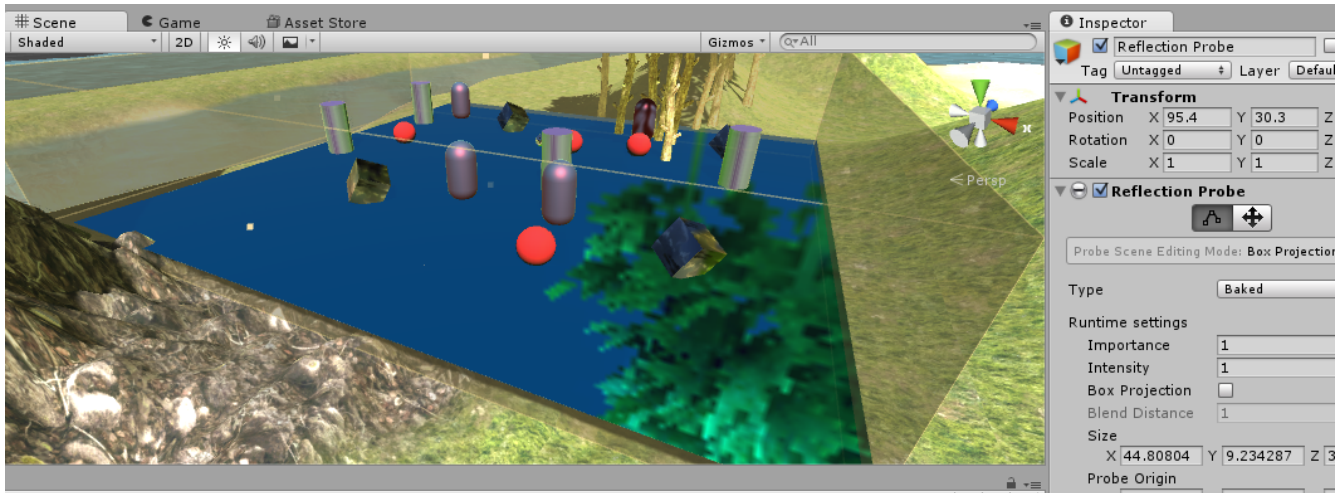
## Unity Tutorials 5 – Lighting and Cameras

Without a reflection source, you must use reflection probes. We only have one area with reflective objects so we only need one reflection probe. Use the move tool to position the probe over top of it. You can click on the icon shown below to change the boundaries of the probe. After clicking on it, notice the little dots that appear within the probe box. You can click and drag these to resize it just like with the scale tool. There is a preview in the inspector that shows approximately what the reflections will look like. Objects not within the boundaries will not benefit from the reflection probe.



You can change the “Type” from “Realtime” to “Baked” because our cluster object will not be moving. Therefore, the reflections will not have to update. “Baking” will be covered later.

## Unity Tutorials 5 – Lighting and Cameras



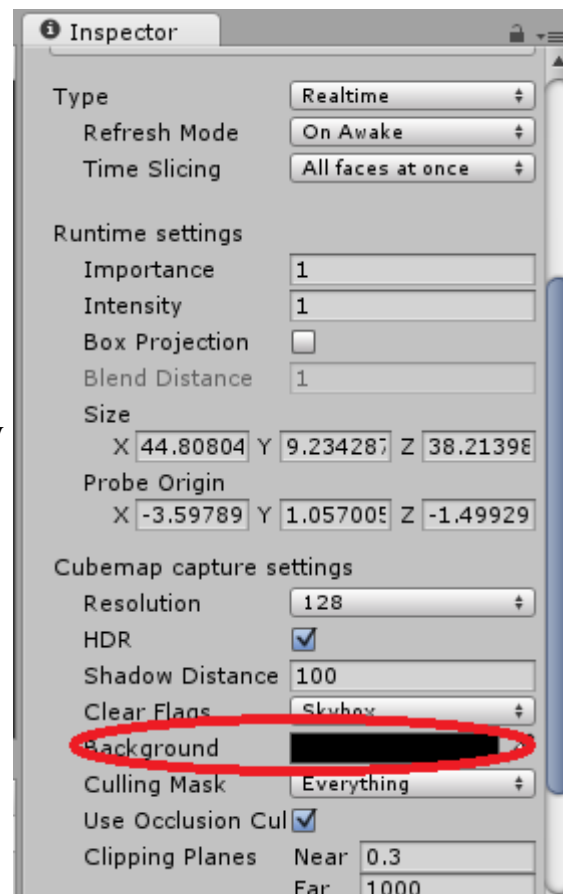
### Point Lights

Change the “Type” of your Directional Light to “Point” and change the rotation to (0, 0, 0).

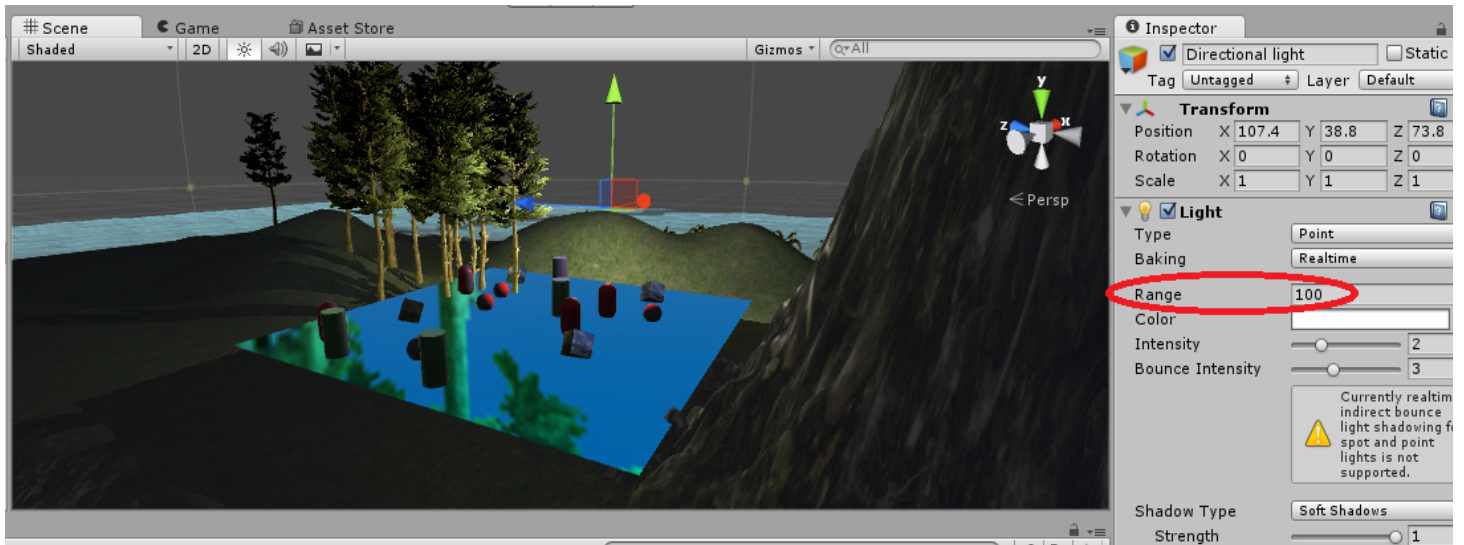
*Notice that even in complete darkness, the reflection probe is creating light. Change its “Background” colour to pure black to fix this. Be aware that the default colour that was there was to simulate a sky. A black background will look strange in an open area.*

A point light sends light in all directions from a central emitter. At first you won't be able to see any light. This is because the default “Range” is very small. Change the range to 100 and place it over your cluster object.

Point lights are great for fires and lamps.

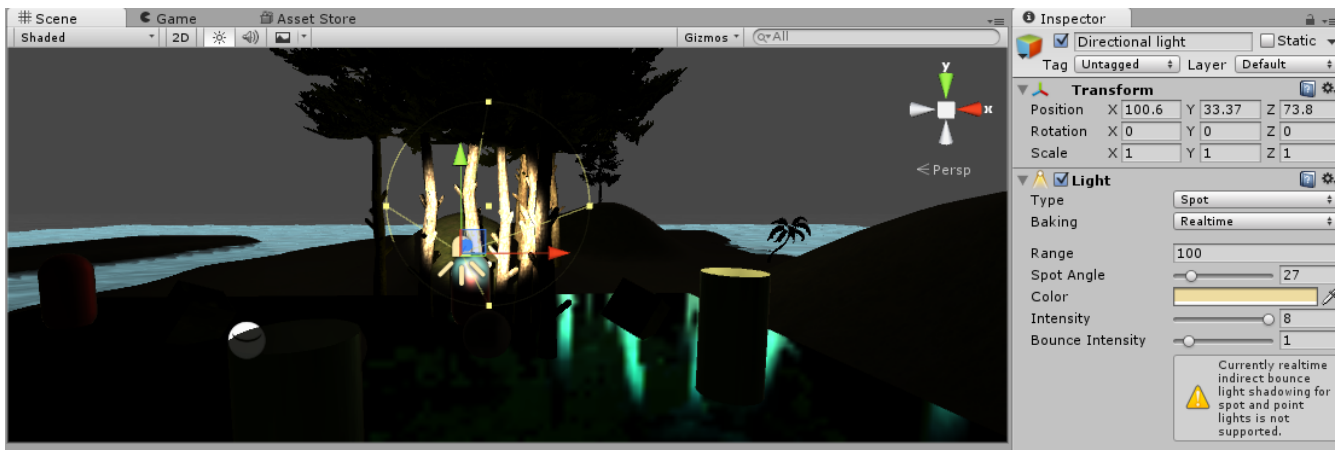


## Unity Tutorials 5 – Lighting and Cameras



### Spot Lights

Now change the light's type to “Spot”. Spot lights illuminate objects within their cone-shaped field.

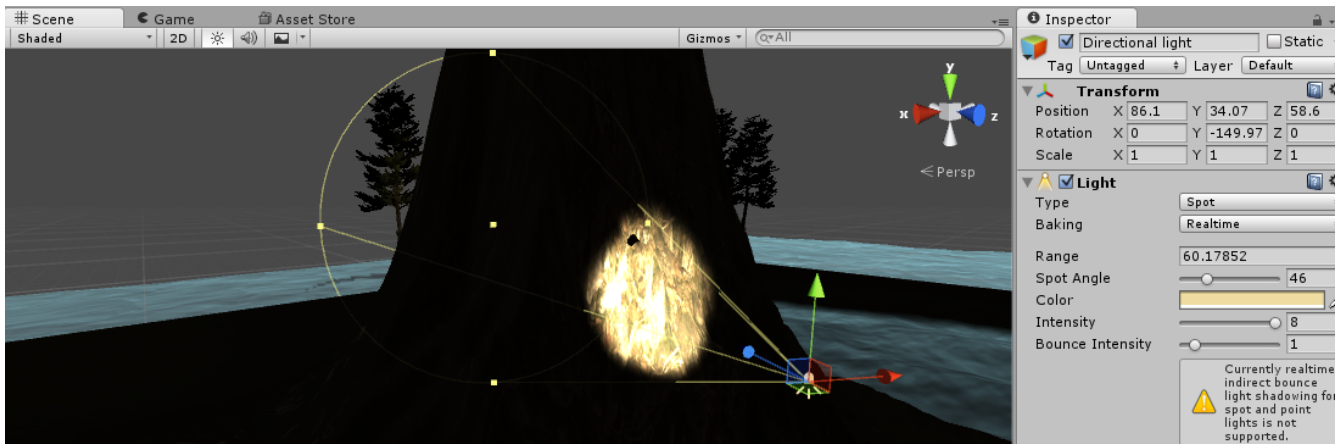


There are 3 key components to a spot light: the range, the angle, and the transform. The position and rotation of the light as a GameObject will determine the source of the light. Try moving it around. You will notice that it will always point at the same thing. To change where the light is pointing, change its rotation. The “Spot Angle” determines how big the cone is. The “Range” determines how far from the source the light reaches.

Spot lights are perfect for flashlights and flood lights.

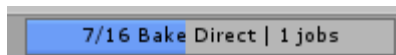


## Unity Tutorials 5 – Lighting and Cameras



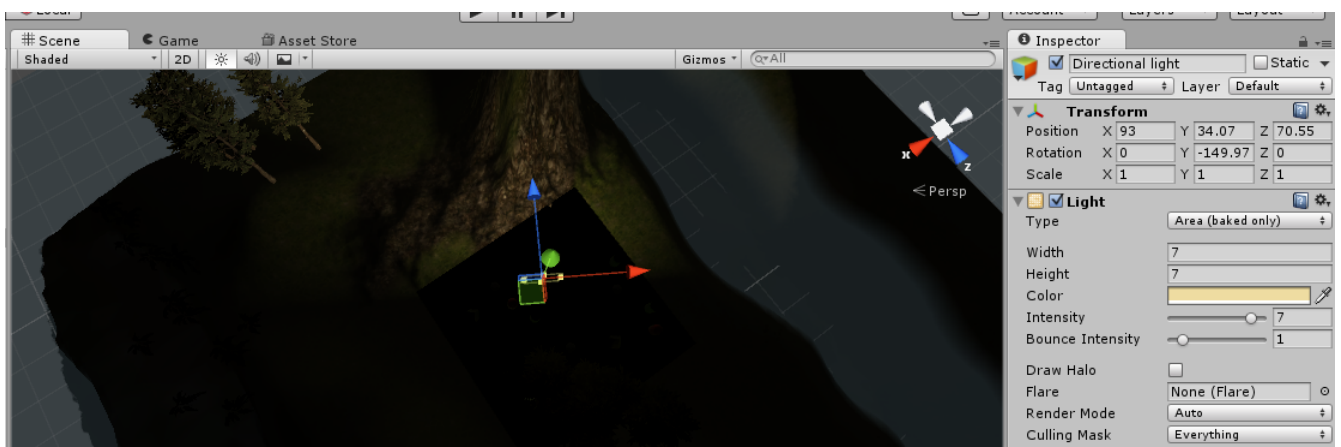
### Area Lights

Note: This needs to bake every time it is updated, which may take some time. If you don't see anything, wait a few seconds. Progress can be seen in the bottom-right corner of Unity.



Try changing the “Width” and “Height” to 10 first.

Area lights light up anything in front of their rectangle. They are great for “glows” like televisions, monitors, and vending machines.



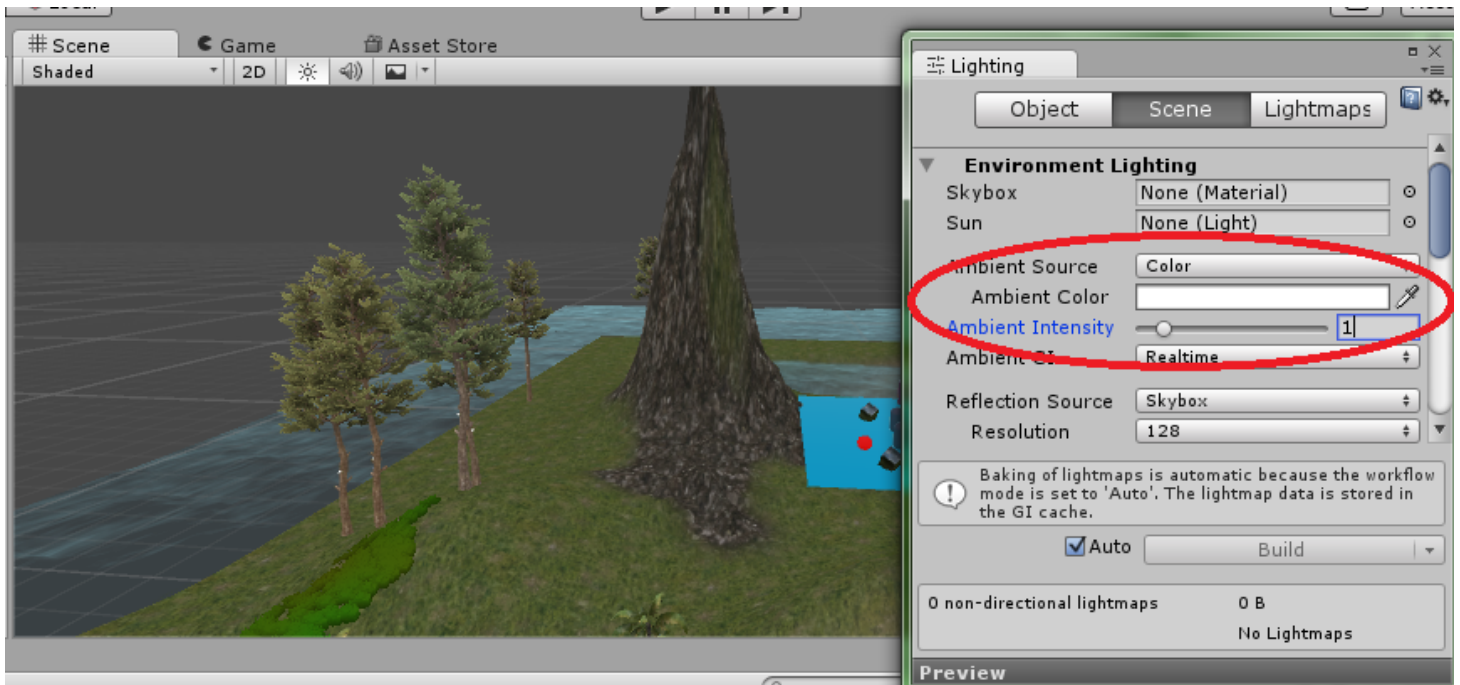
### Baking

*Baking is essentially the act of making the appearance of something permanent. Baked lights and shadows will never move no matter what happens during the game. For example, say you have a box that can be pushed by the player. If there is a light that has*

## Unity Tutorials 5 – Lighting and Cameras

*been baked onto the box, the box will still be lit up even after it has been moved away from the source. So why use baked lighting? Well, realtime lighting is constantly updated as the game is played. This can be hard on the system and slow the game down. Any combination of lights and objects that will never move should be baked.*

To bake an ambient light, begin by creating ambient lighting. You can delete the Directional Light if you wish. Go into the “Lighting” window found under “Window” and change the “Ambient Color”. Make sure the intensity is above 0 too or there will be no light.

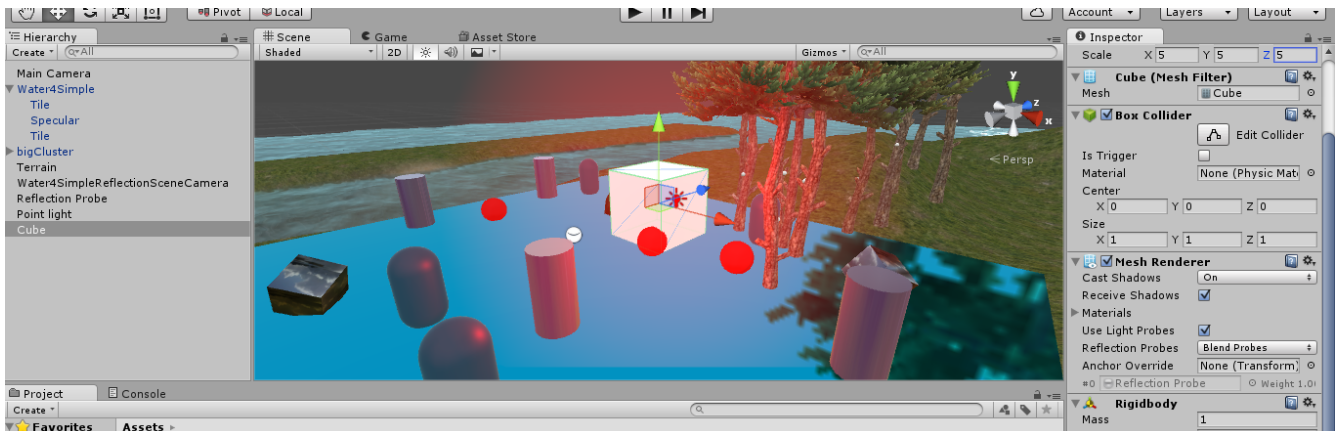
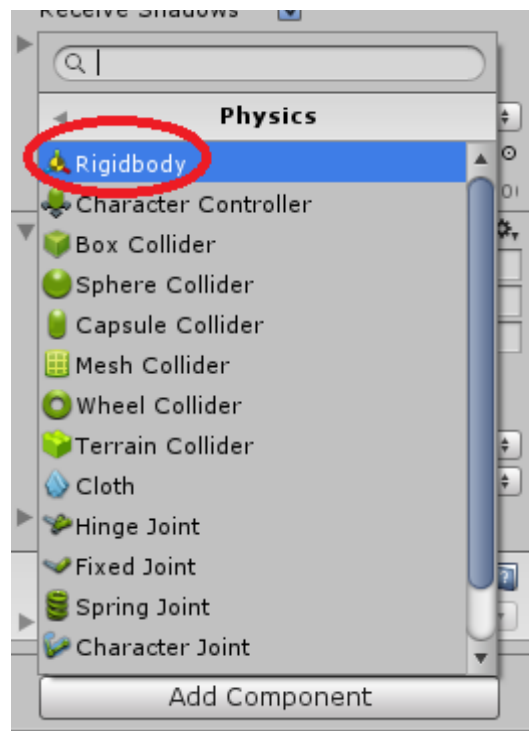
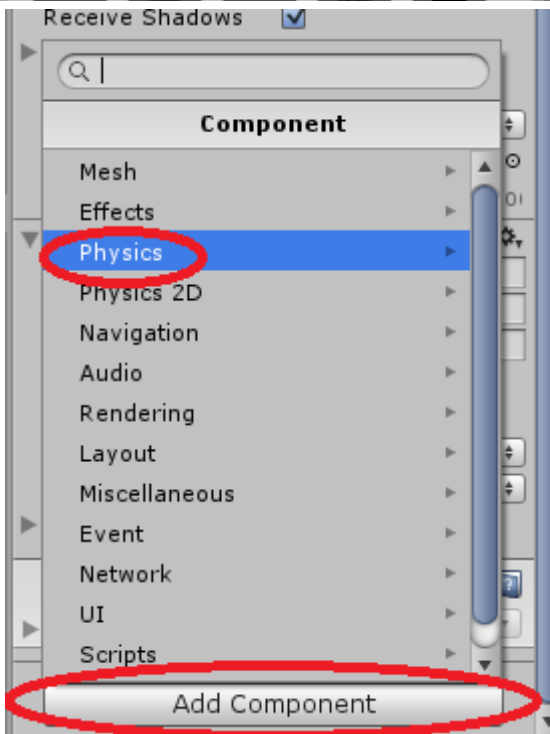
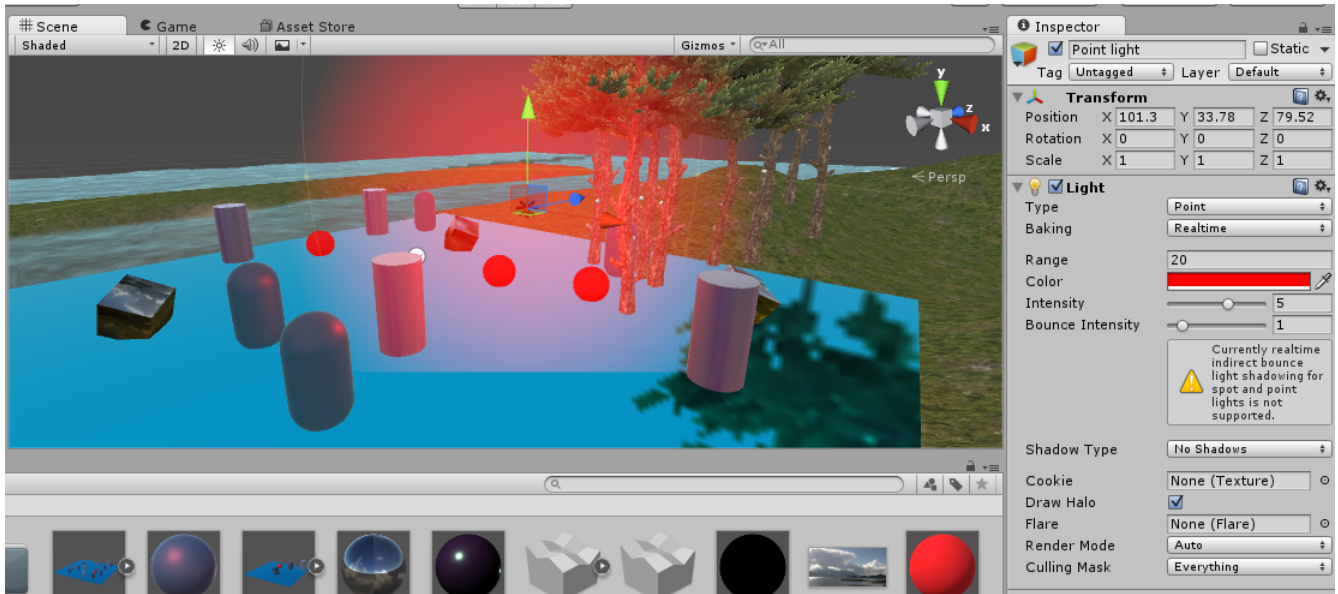


Ambient light is great to use in place of a directional light for cloudy outdoor scenes. Objects will now always be at least this bright.

Now try creating a point light and set it up like the one shown below.

We are going to create a cube that the player can push around. Create a cube and position it over your cluster. In the inspector for the cube, click on “Add Component”, “Physics”, and “RigidBody”.

# Unity Tutorials 5 – Lighting and Cameras

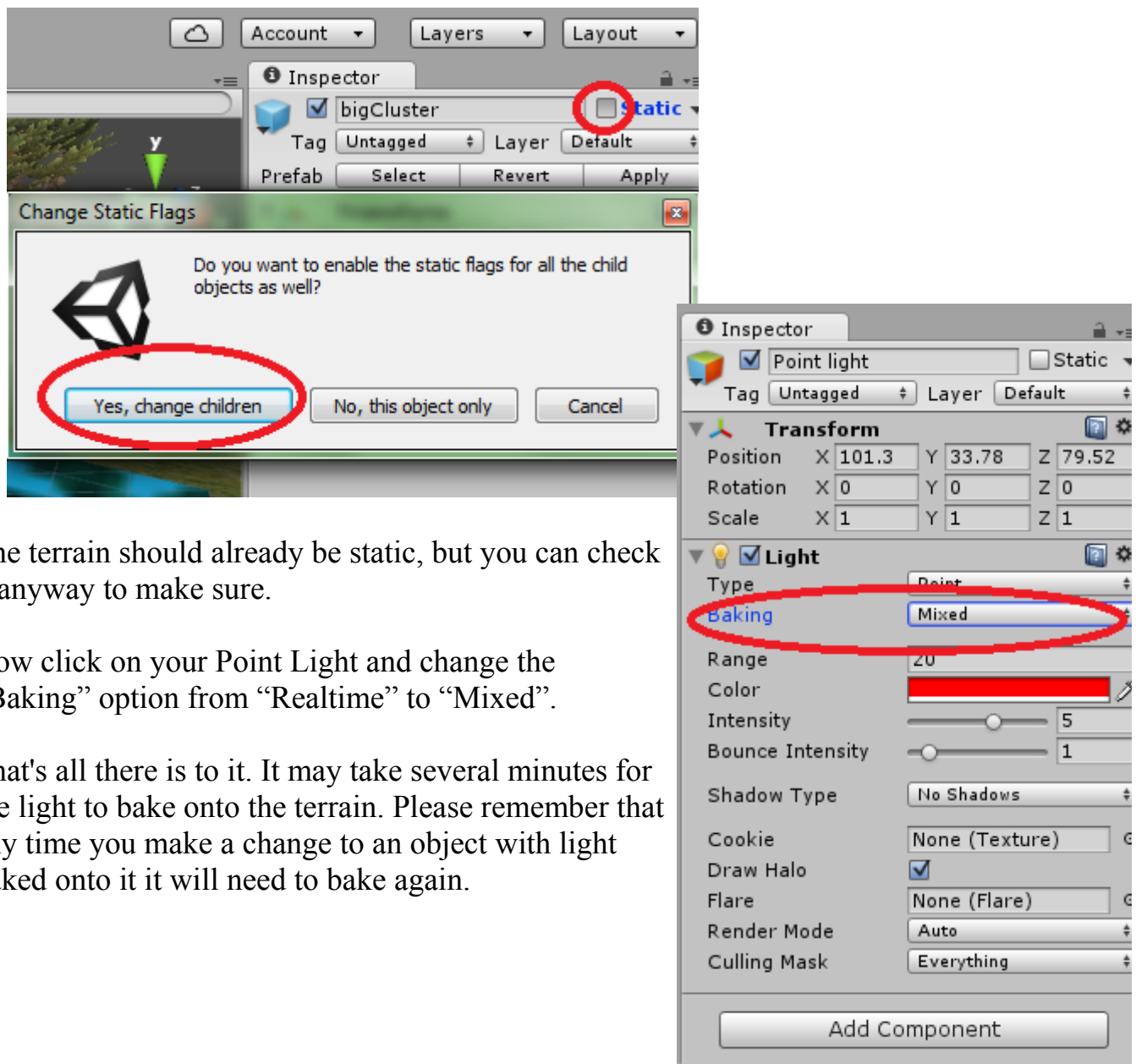


## Unity Tutorials 5 – Lighting and Cameras

So what do you do if you want to bake this light onto the cluster object but not that moveable cube? Unity allows for the option of “mixed” lighting. Remember that you can mark an object as “Static”. If a light has mixed baking then it will only bake onto objects that are static. If we mark the cluster as static but not the box and set this red point light to mixed, then it will be baked onto the cluster and save on processing power, but not the box so the box will no longer bit lit up red when it is pushed away from the light source.

We will not see this in action in this tutorial, but we can at least set it up for now.

Click on the cluster object, then click on the box next to “Static” in the inspector. Click “Yes, change children” to change all of the objects contained within it to static.



The terrain should already be static, but you can check it anyway to make sure.

Now click on your Point Light and change the “Baking” option from “Realtime” to “Mixed”.

That's all there is to it. It may take several minutes for the light to bake onto the terrain. Please remember that any time you make a change to an object with light baked onto it it will need to bake again.

# Unity Tutorials 5 – Lighting and Cameras

## II – Cameras

### ***What is a camera?***

*Cameras are used to give the player eyes. They behave much like real cameras and can be panned, tilted, and zoomed.*

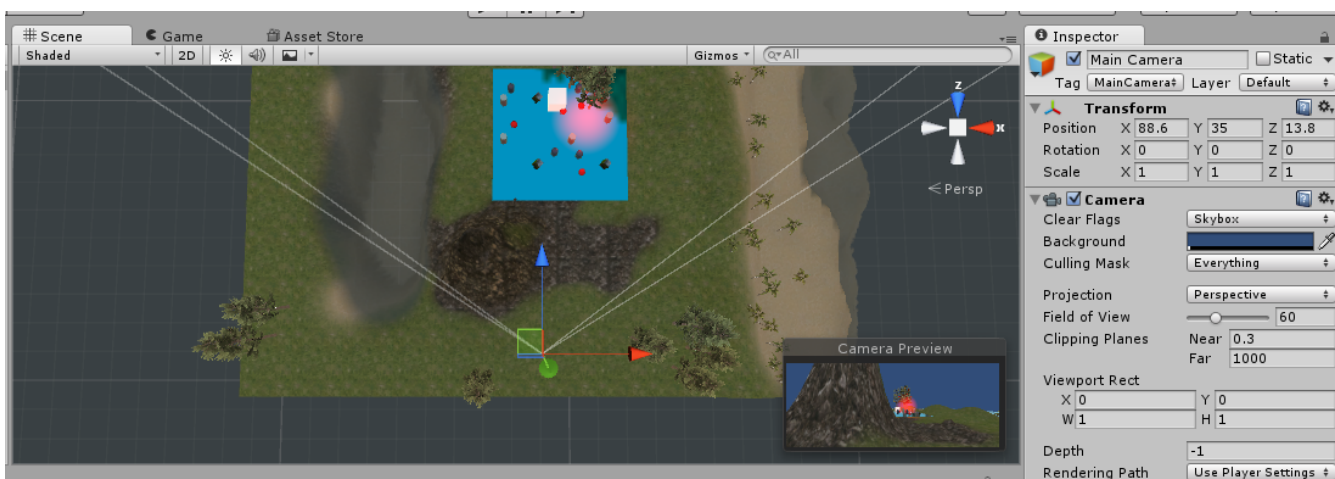
### **Positioning Cameras**

Positioning a camera is easy. With only one camera in your scene this will automatically be the view that is shown on the screen. You can have multiple cameras in one scene to show multiple views at once, or to switch between different views quickly. In the inspector you can find a “Depth” property. When using multiple cameras the camera with the highest depth will be displayed at front of the screen. If all cameras are full-screen then only the view with the highest depth will be seen.

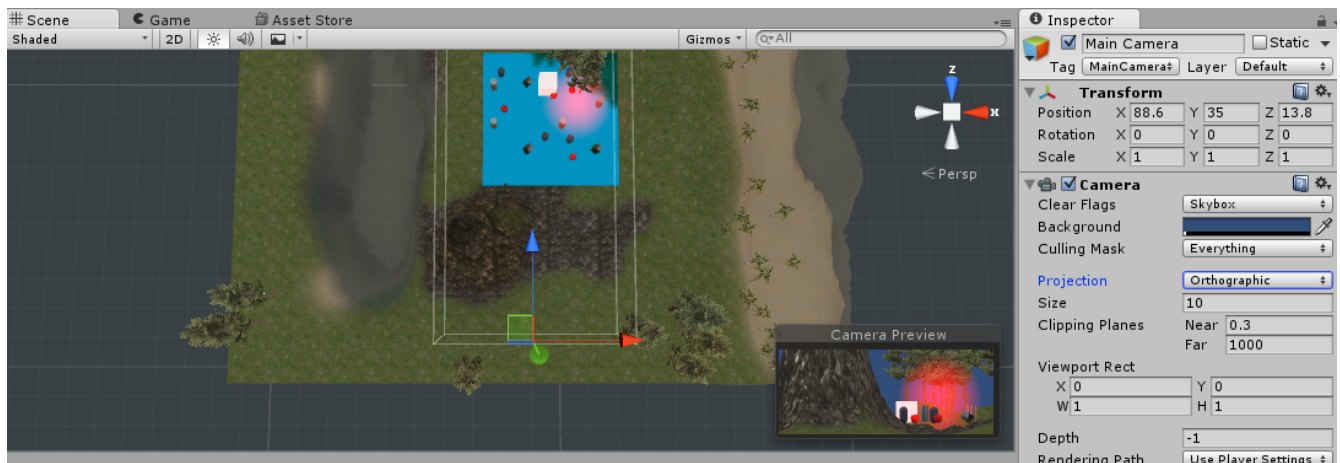
Try moving the Main Camera so that it captures most of the terrain. You can see what the player will see using the camera preview that shows up in the Scene View or by switching to Game View.

### **Projection**

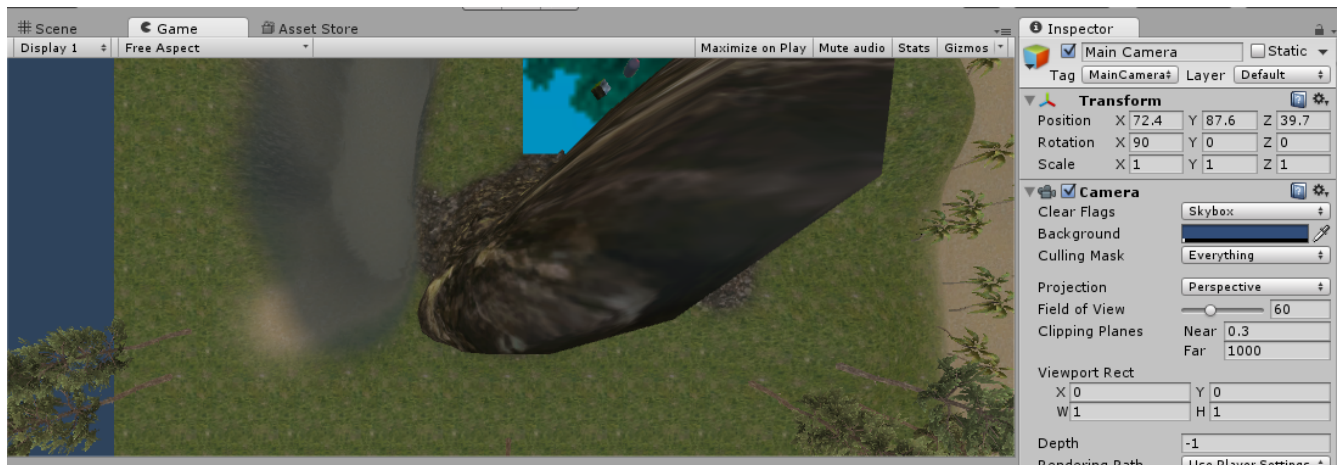
There are two projection modes: perspective and orthographic. Perspective mode uses a “field of view”, which is 60 by default. The larger the field of view, the more the camera will capture at once. In this mode the field grows larger the farther it is from the camera. Orthographic mode is different in that instead of using an increasing field, the field of view is determined by a box with a set width.



# Unity Tutorials 5 – Lighting and Cameras



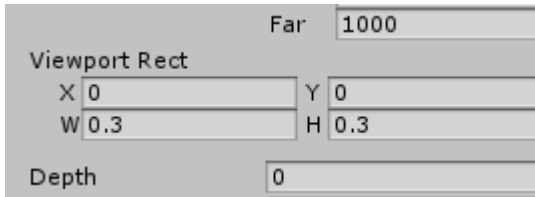
Rotate your camera 90 degrees along the X axis and place it over top of your spire or similar object. This is called a “bird’s eye” or “top-down” view. Now switch between orthographic and perspective. You can see that “orthographic” gives no sense of depth.



# Unity Tutorials 5 – Lighting and Cameras

## Multiple Cameras

Go to “GameObjects” and click “Camera” to create a new camera. Notice that the “Depth” is 0. This camera is now in front of the Main Camera on the screen. Now position it somewhere different from the Main Camera and change the Width and Height under “Viewport Rect” to 0.3.

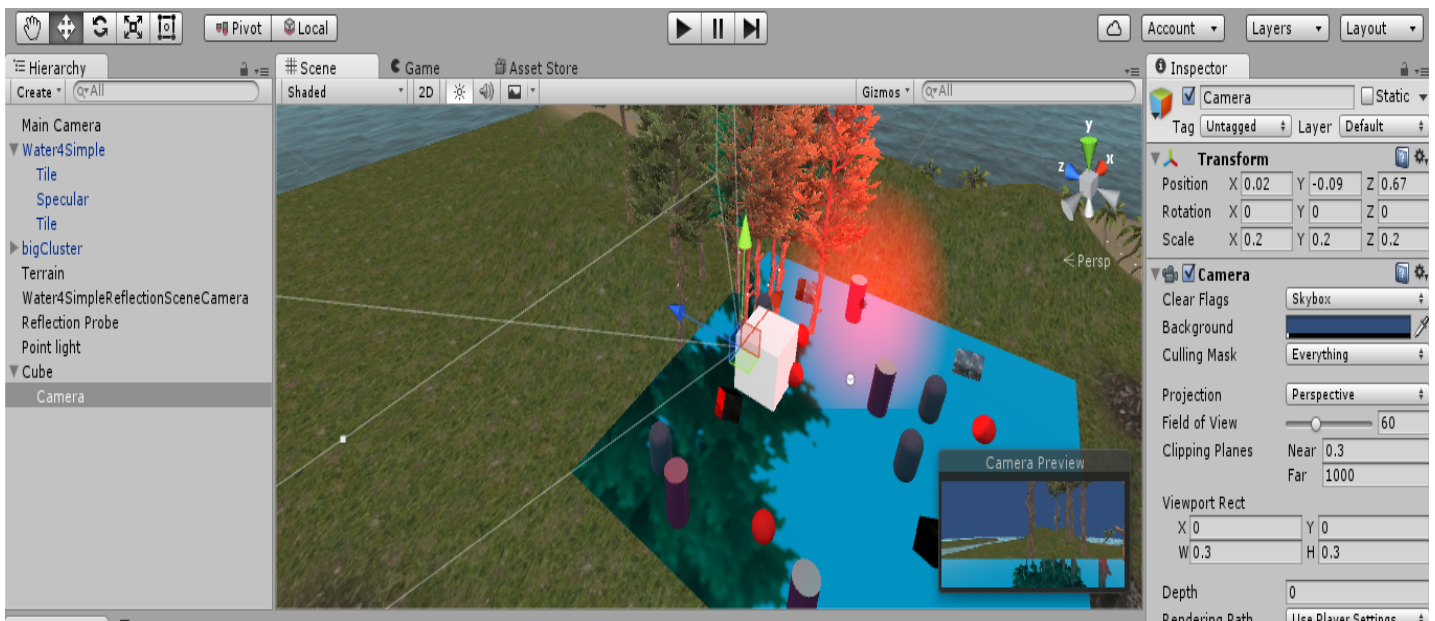


Now check your Game View. You now have a second camera in the corner of the screen. Changing the X and Y of the Viewport Rect will move this little camera somewhere else on the screen.

## Cameras that Follow the Player

Remember that parented objects move with the parent. So what happens when you parent a camera to the player character? The camera will move with it! Try parenting the new camera to the movable cube you made earlier.

Now position the camera so that it is in front of the box. Check all angles to make sure it is positioned correctly. Later on when we move that box around that camera will move with it and you will be able to see what that box sees.



## Unity Tutorials 5 – Lighting and Cameras

### **Rail Cameras**

You may want a camera to move independently from the player character. For example, 2D side-scrolling platforming levels where the player has to keep up with the camera.

This can be done by creating a script that moves the camera so many units every frame.



# Unity Tutorials 5 – Lighting and Cameras

## III – Particles

Go to “GameObject” and click “Particle System”. A particle system emits a flow of particles, simple images, from a spawn point. There are many uses for particle systems, such as flames, rushing water, bubbles, snow, rain, etc.

Particle systems are very complex and you are encouraged to play around with these settings on your own. Below are a number of simple simulations and their settings for reference.

### Fire

The screenshot displays the Unity 3.5.7 interface with a 3D scene featuring a fire particle system. The scene includes trees, a body of water, and a fire effect. The Particle System Inspector is open, showing the following settings:

Property	Value
Duration	5.00
Looping	<input checked="" type="checkbox"/>
Prewarm	<input checked="" type="checkbox"/>
Start Delay	0
Start Lifetime	5
Start Speed	1
Start Size	4
3D Start Rotation	<input type="checkbox"/>
Start Rotation	0
Randomize Rotation Dir	1
Start Color	[Color Picker]
Gravity Modifier	0
Simulation Space	Local
Scaling Mode	Local
Play On Awake*	<input checked="" type="checkbox"/>
Max Particles	1000
Emission	<input checked="" type="checkbox"/>
Shape	<input checked="" type="checkbox"/>
Shape	Cone
Angle	3
Radius	1
Length	5
Emit from:	Base
Random Direction	<input type="checkbox"/>

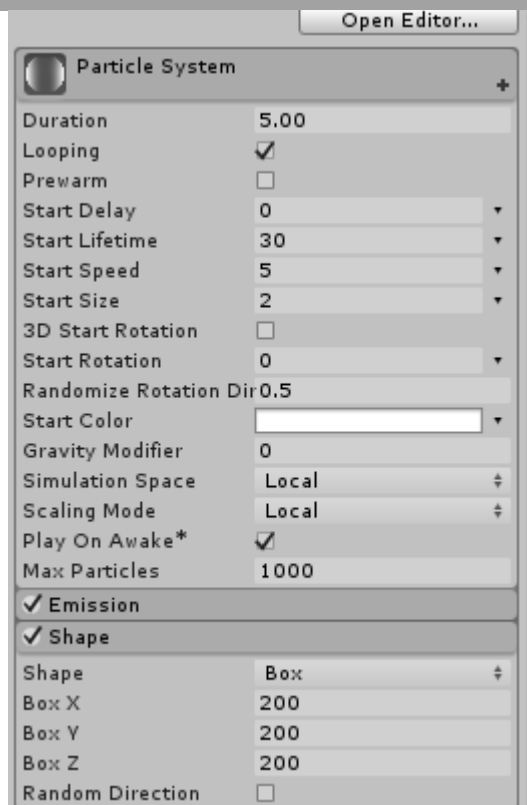
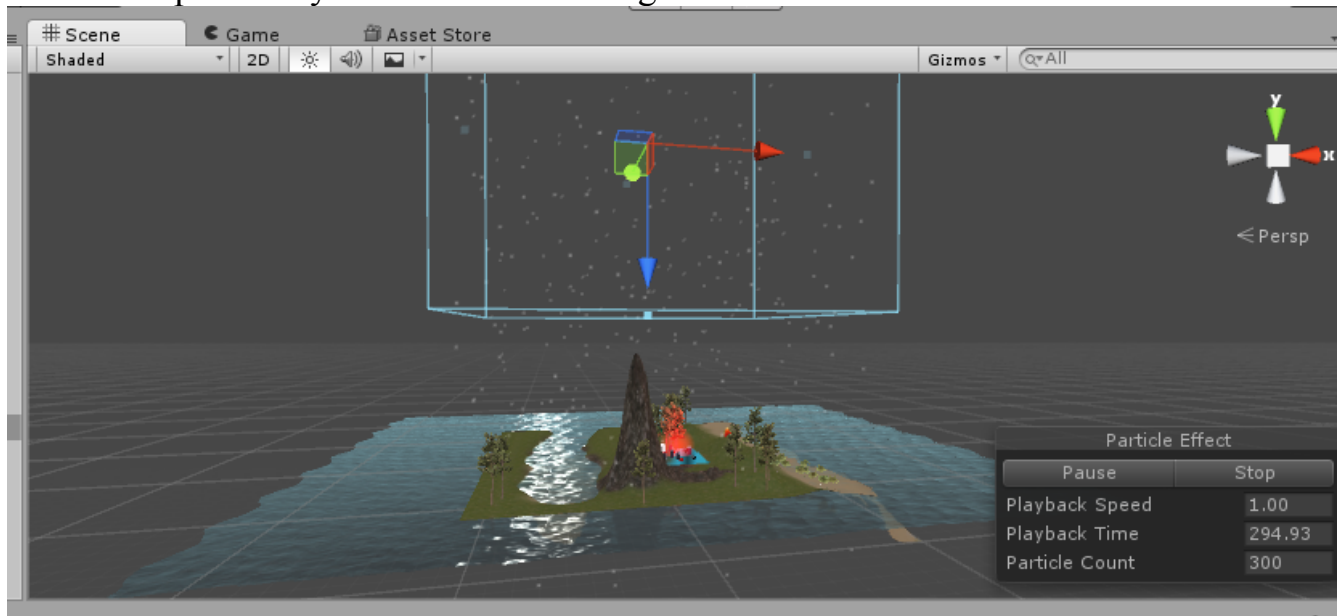
Additional settings in the Particle System Inspector include:

- Velocity over Lifetime:
- Limit Velocity over Lifetime:
- Inherit Velocity:
- Force over Lifetime:
- Color over Lifetime:
- Color: [Color Gradient]
- Color by Speed:
- Size over Lifetime:
- Size: [Size Gradient]
- Size by Speed:
- Rotation over Lifetime:
- Rotation by Speed:
- External Forces:
- Collision:
- Sub Emitters:
- Texture Sheet Animation:
- Renderer:
- Resimulate:  Wireframe:
- Default-Particle: [Particle Icon]
- Shader: Particles/Alpha Blended P1

# Unity Tutorials 5 – Lighting and Cameras

## Snow

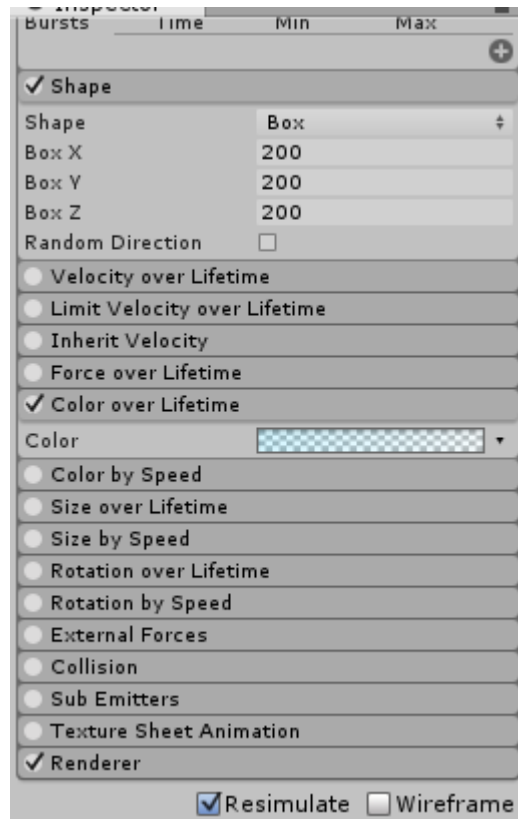
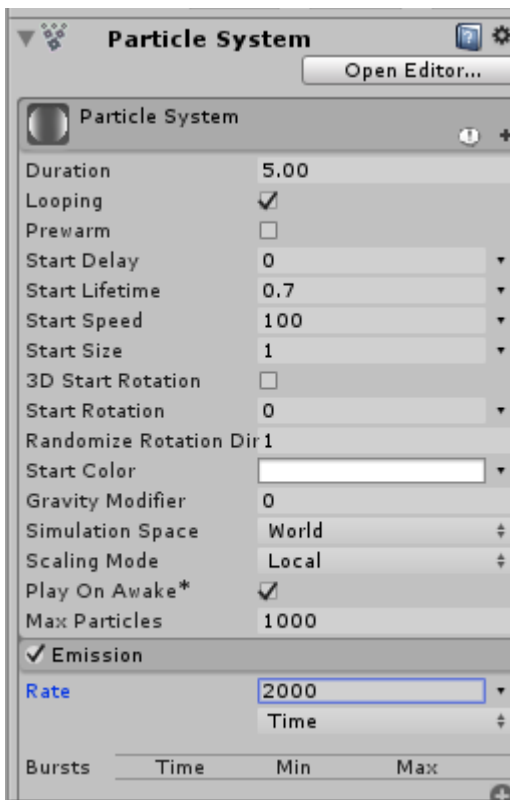
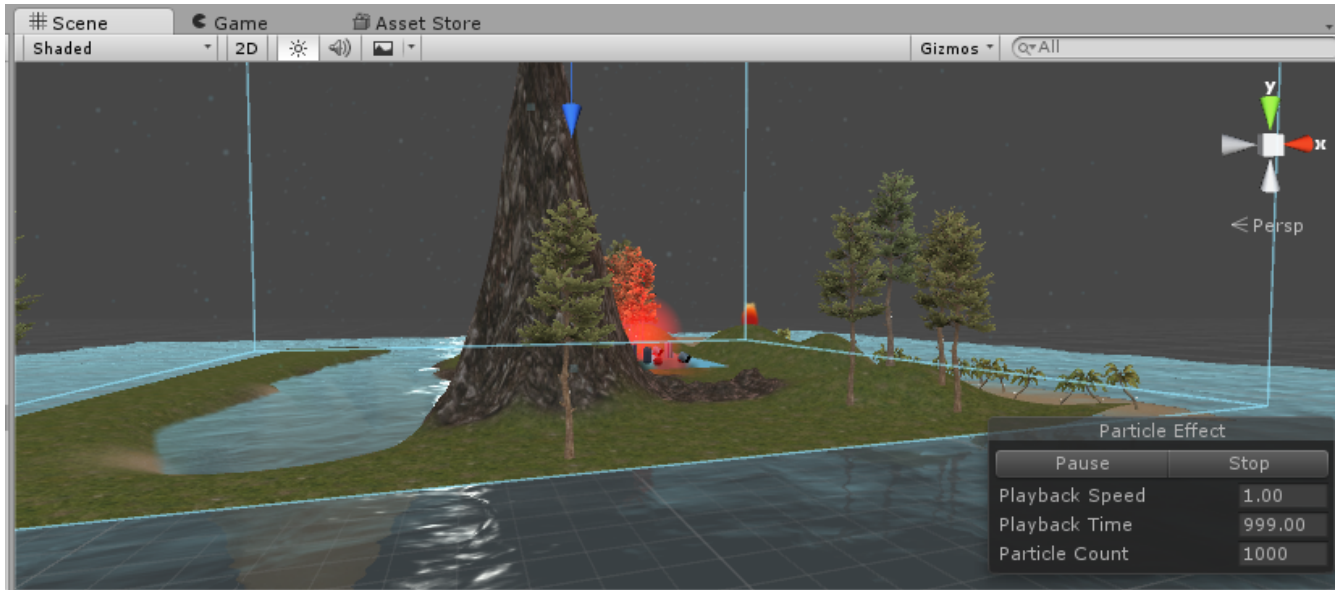
Note: This particle system is rotated 90 degrees on the X axis.



# Unity Tutorials 5 – Lighting and Cameras

## Rain

Note: This particle system is rotated 90 degrees on the X axis.



# Unity Tutorials 5 – Lighting and Cameras

