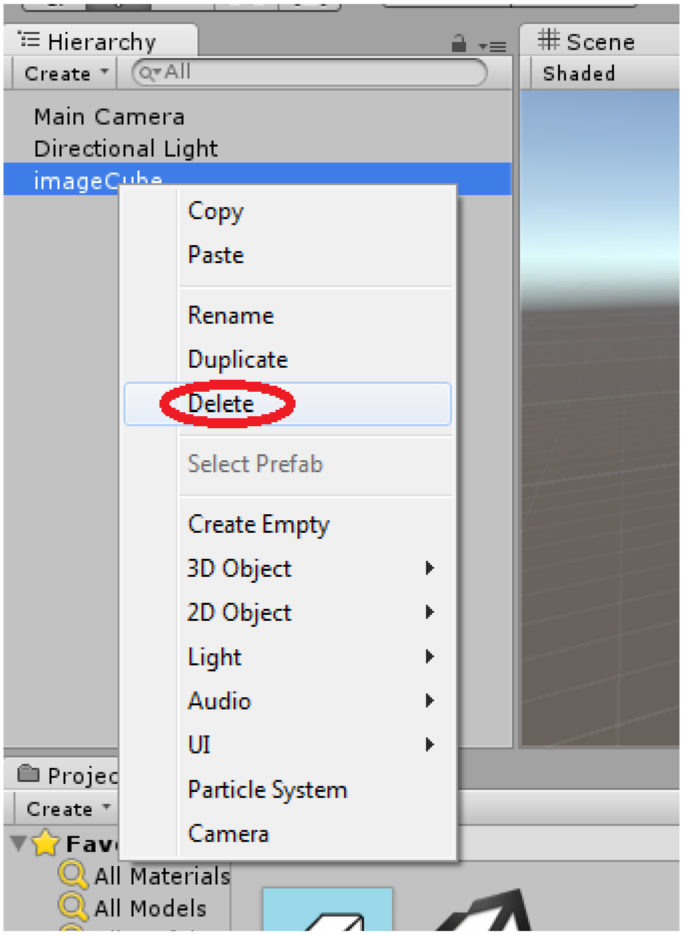
Unity Tutorials 3 – Game Objects

I – GameObjects

Cleaning up the Scene

Let's start by deleting the image we made in the last tutorial. To do that, simply click on it in the hierarchy and press the DEL key or right-click it and click delete. In the right-click menu you can also rename and duplicate the object.



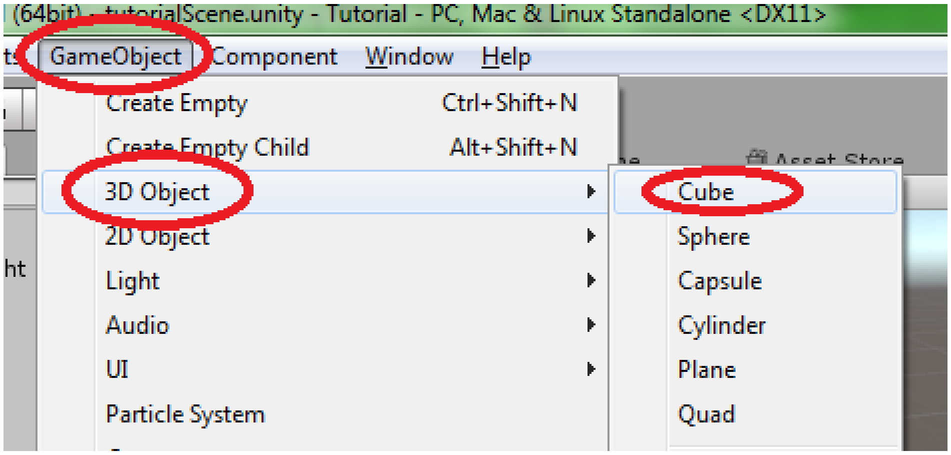
Try deleting the asset too. Same as before, right click the image under Assets and press delete, or highlight it and press the DEL key. Once deleted, you can check your Assets folder to see that it is gone.

Adding 3D Objects

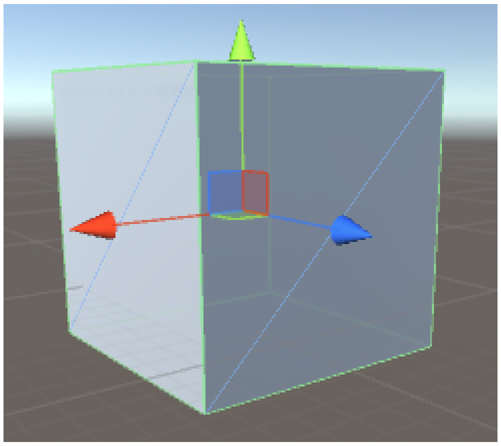
Besides adding completed 3D models, you can also add standard 3D objects, like cubes and spheres, into a scene. You can even group these objects together to create a larger object.

First, go into “GameObject” on the top bar, “3D Object”, and “Cube”.

Unity Tutorials 3 – Game Objects



With the move tool selected, notice that the arrows originate from the centre of the cube.



Pivot Point: The central point from which transformations on an object occur.

Parent: An object that contains other objects.

Child: An object that is contained by another object.

This is known as the object's **pivot point**. With most models, the pivot point should be at the base. To see the effects of the pivot point, select the rotate tool and rotate the cube around. Notice how it doesn't move under the grid? If the pivot where at the base it would.

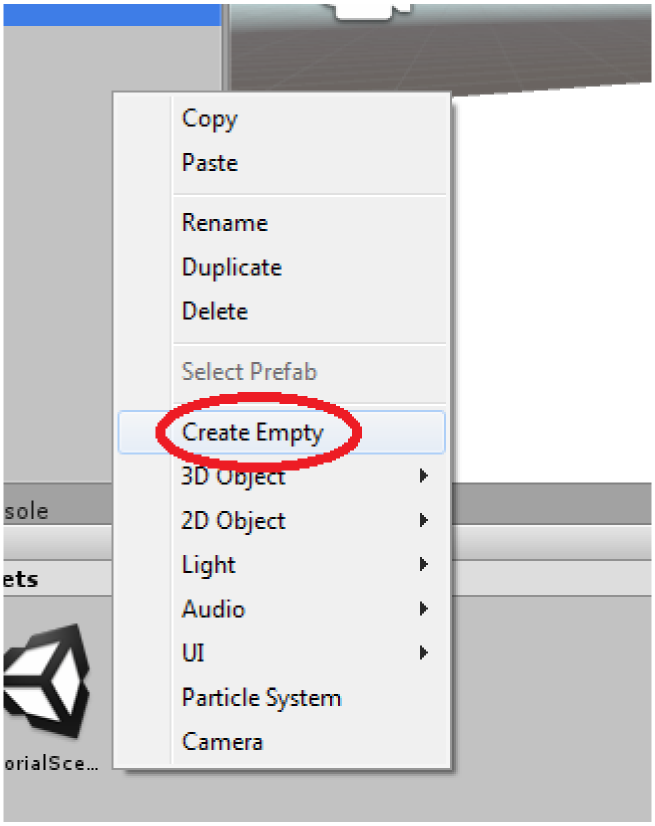
Note: The pivot point of an object cannot be directly changed in Unity. That being said, there is a way to do it which we will get to later.

Now try making one of each 3D object. You should have a cube, a sphere, a capsule, a cylinder, and a plane. Don't worry about the quad. A quad is like a small plane.

Parenting Objects

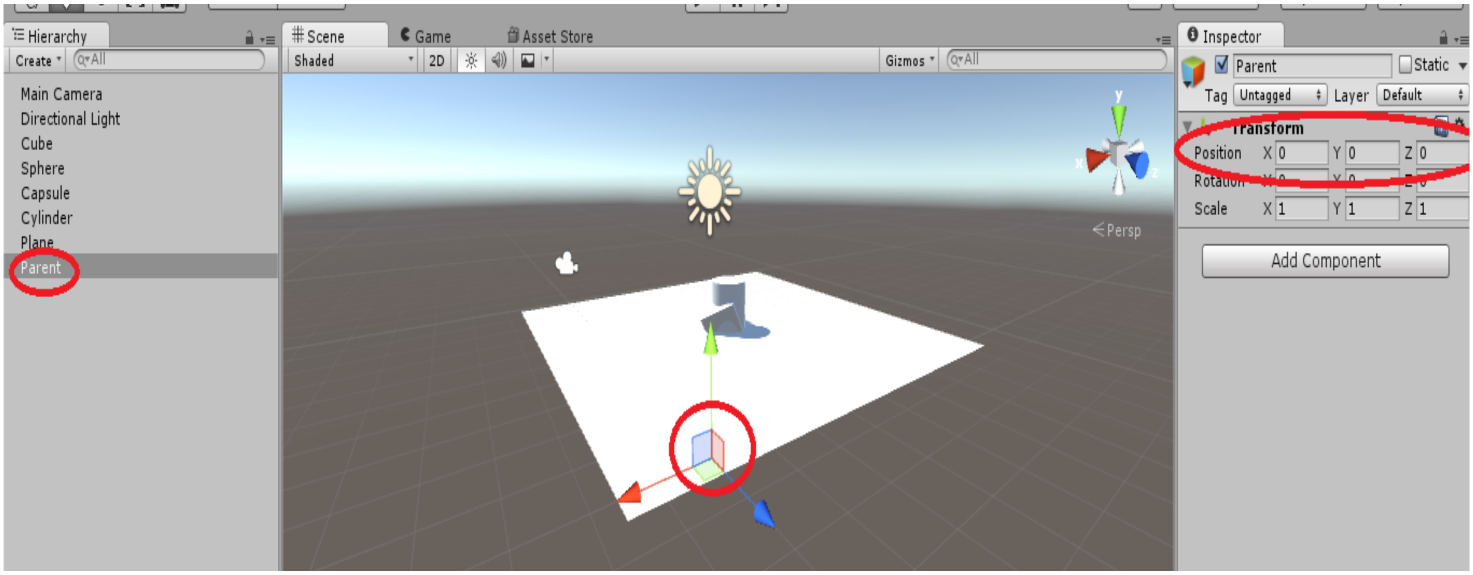
Now we're going to create one large object out of these smaller objects. This process is known as “parenting”. An object that is parented to another object becomes dependant on it. Any transformation that is done to the **parent** object will be done to any **child** object as well.

First you will need to make an empty game object. An empty object is just what it sounds like. It's an invisible object with no attributes. You can find it in the GameObject menu. Click “Create Empty”. You can also right click inside the hierarchy to bring up the GameObject menu. Make sure that you don't have another object selected or the empty object will become parented to it. You can unparent an object by dragging it out of the parent to an empty space in the hierarchy.



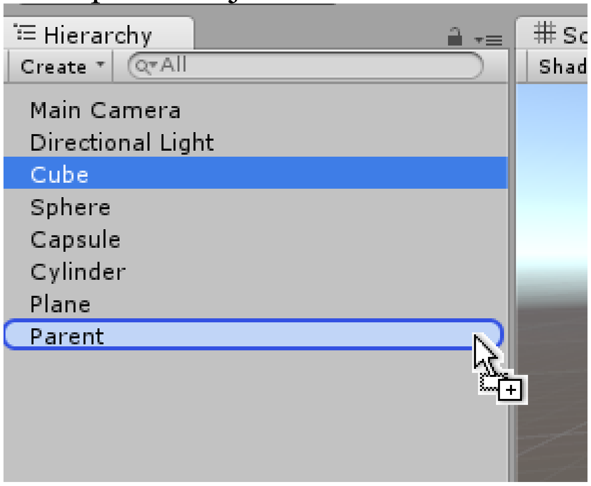
Now before we begin, let's set up our parent object. First, rename it. Next, check its tranform in the inspector. Set the position to (0, 0, 0). This is where we want the centre of our completed object to be.

Unity Tutorials 3 – Game Objects



Now set up your objects however you want them, but first move your plane to (0, 0, 0). Since it will be the central part of our complete object, it would make sense to have its pivot and the parent's pivot at the same location.

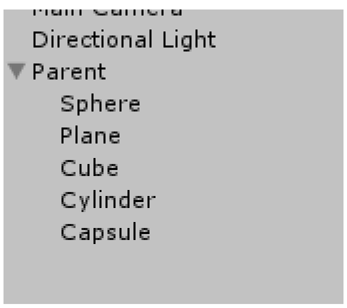
Parenting objects is very easy. Just go to the hierarchy and click and drag each object into the parent object.



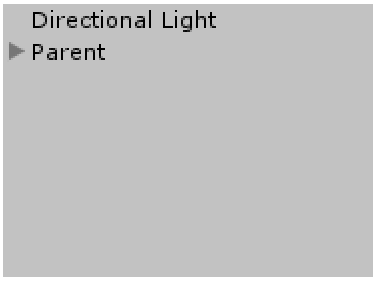
Make sure that you are parenting each object to the actual parent and not to one of the childs in the parent.

You can click the little arrow next to the parent to hide the childs. This is convenient for organizing your hierarchy. For example, say you have multiple of the same enemy in a scene. You can parent all of these enemies to one empty object. The hierarchy can be very difficult to manage once it is full of objects, so keeping it organized is a good idea.

Unity Tutorials 3 – Game Objects

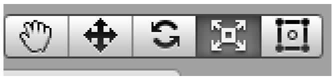


Main Camera
Directional Light
▼ Parent
Sphere
Plane
Cube
Cylinder
Capsule

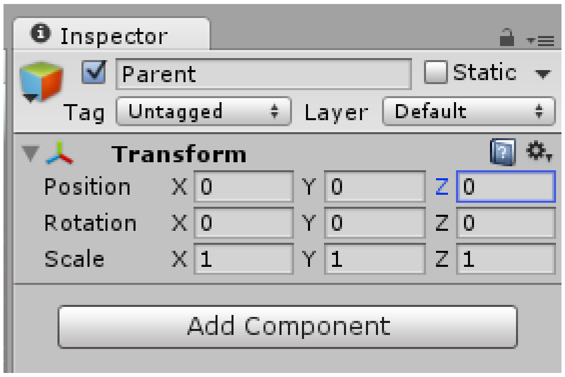


Directional Light
▶ Parent

Let's try using the scale tool on the parent object. Select the parent object then select the scale tool. You can use it just like the move and rotate tools.



You can see that all the parented objects scale with the parent. Now, try resetting the scale. Scale is based on multiplications. Whatever the number is in the transform, the size of the object will be its original size times that number. So, the default scale is 1.



Inspector

☑ Parent □ Static

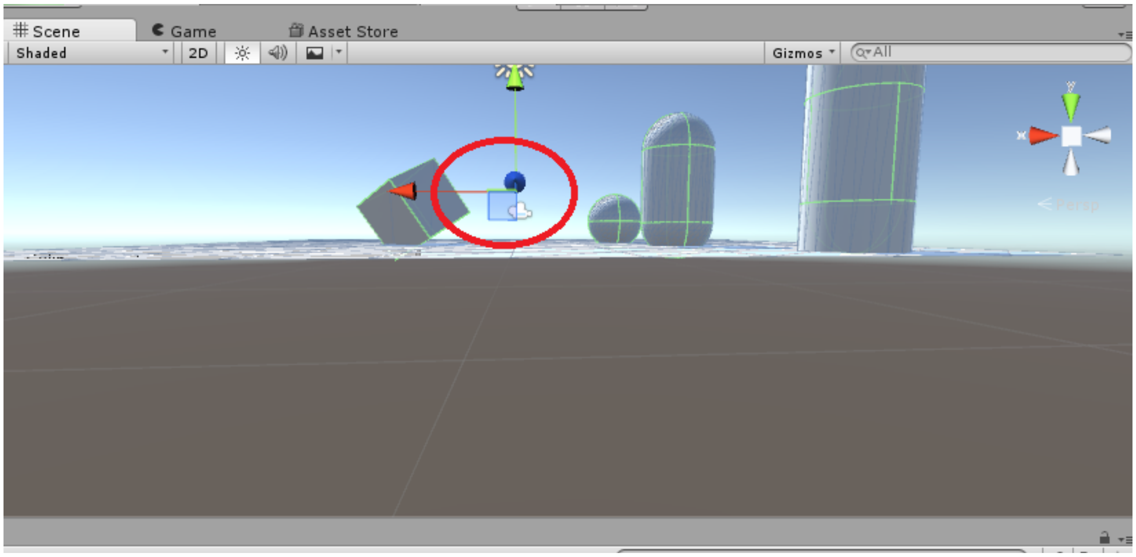
Tag Untagged Layer Default

Transform

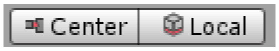
	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

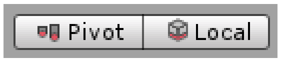
Back to the Pivot

Under the default settings you should be able to see that the tool arrows originate not from the base of our object but from slightly up. If not, keep reading and you'll see why.

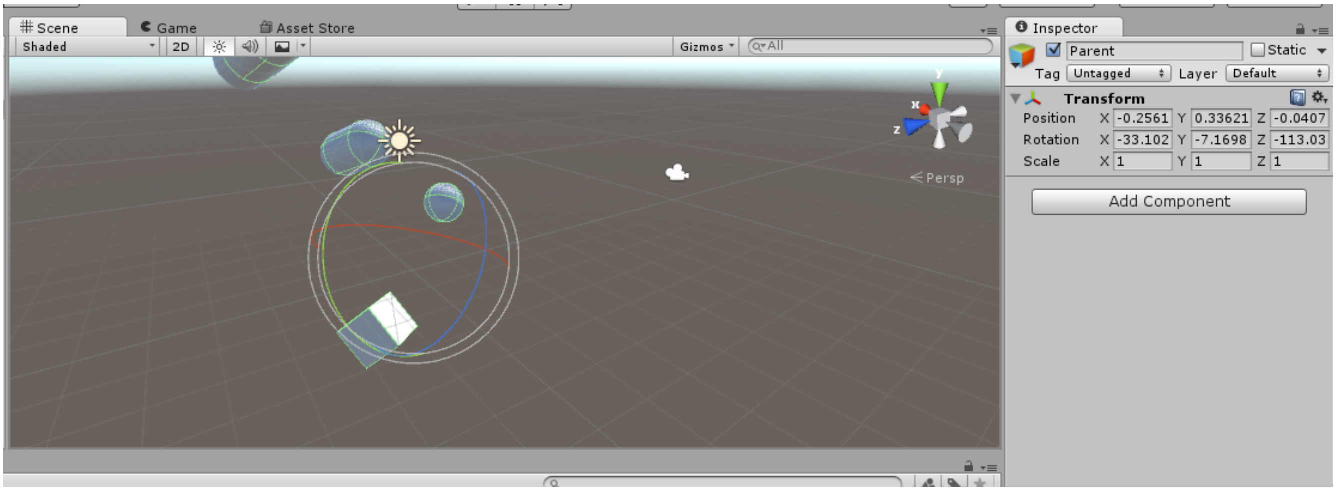


This is because right now we have the tools set to act on the centre of an object and not its pivot. Next to the tools are two buttons. Click the one on the left.





With the parent object selected you should be able to see how the arrow's location changes to where we originally set it. To see the difference, change the button back to “Center” and try rotating the parent object.



☑ Parent □ Static

Tag Untagged Layer Default

Position	X	-0.2561	Y	0.33621	Z	-0.0407
Rotation	X	-33.102	Y	-7.1698	Z	-113.03
Scale	X	1	Y	1	Z	1

You can see that the position of the object does change slightly. This is because the centre of the object doesn't move. Instead, the object moves around it. You might also notice that you can only see the plane from one side. This is the same for every **face** in Unity, but we'll cover that later.

Now change the rotation and position back to (0, 0, 0) and change the “Center” button to “Pivot”. Try rotating the object now.

Notice how not only does the position not change, but the objects now rotate under the grid.

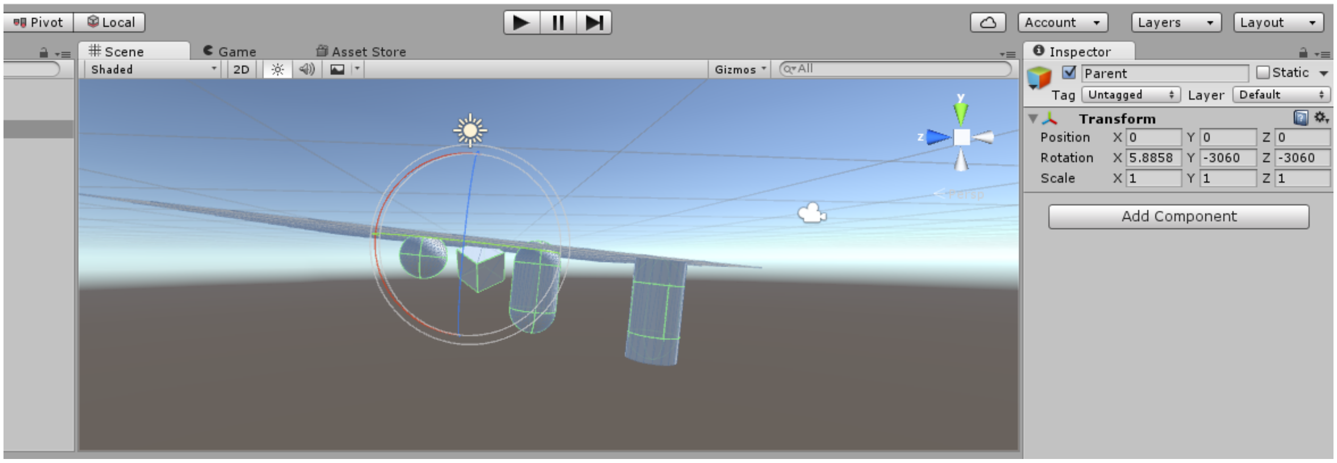
Vector: The smallest part of a polygon. A vector is basically like a point on a graph. Each quad (four sided polygon) has 4 vectors.

Side: The middle part of a polygon. Each side is made up of a line between two vectors, or points.

Face: A face is an area surrounded by 3 or more sides.

Polygon: A 2D shape. Every 3D model is made up of many polygons put together.

Unity Tutorials 3 – Game Objects



☑ Parent □ Static

Tag Untagged Layer Default

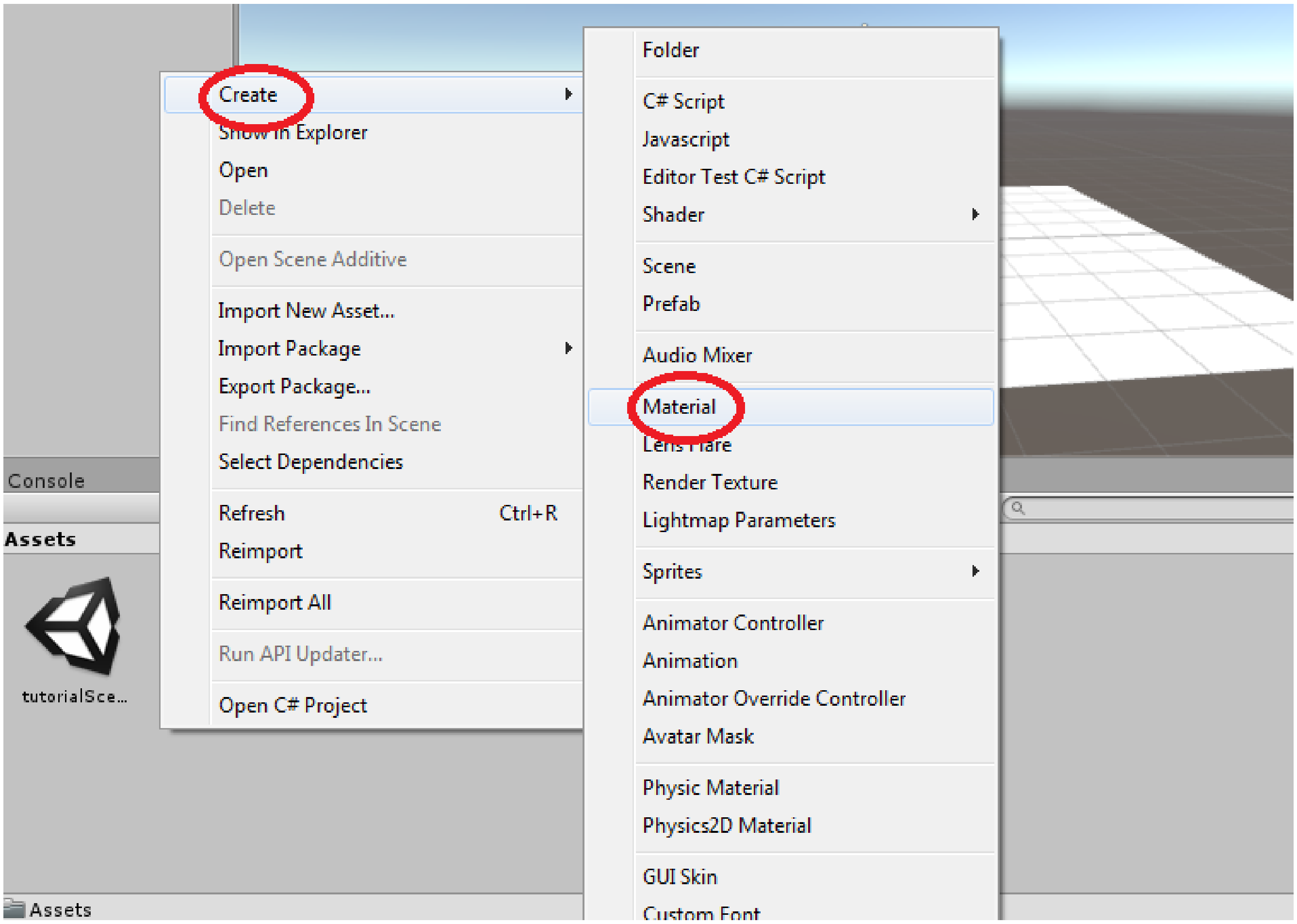
Position	X 0	Y 0	Z 0
Rotation	X 5.8858	Y -3060	Z -3060
Scale	X 1	Y 1	Z 1

Unity Tutorials 3 – Game Objects

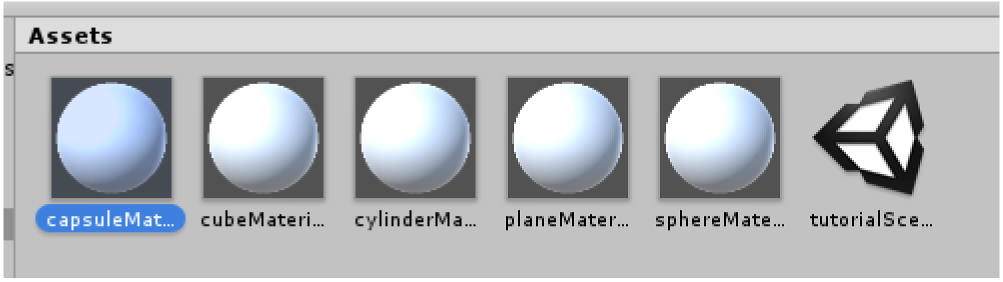
II – Applying Materials

There is an important difference between “textures” and “materials”. A material basically defines what an object should look like in terms of colour. A texture is an image file that can be turned into a material.

Try creating five different materials and apply them to your objects. First, right-click inside the Assets window. Under “Create” you'll see a long list of assets that you can create from scratch. Click on “Material”.



Once you've made your first material you can highlight it and hit CTRL and D. This will duplicate it. Next, rename your materials.



Let's take a look at the inspector for one of the materials. You can see that there are a number of different "maps" as they're called. The albedo is the main texture that decides the base colouration of the material. This is where you would put a **texture map**.

For now, just click on the colour box next to Albedo and choose a different colour for each material. You can also play with the metallic and smoothness settings. Furthermore, if you click on the box next to "Shader" you can change from "Standard" to "Standard (Specular setup)". This will change the **metallic** settings to specular settings. This will not be covered in these tutorials, but you may notice how it can be used to combine two colours.

Normals

Before we apply the materials, remember how you can only see faces from one side? This has to do with which way the normals are facing. In geometry, a normal is a line that is perpendicular to a given object. In graphics, these normals are used to determine which way a face is facing. At first glance, a polygon looks like it has two faces. If you hold up a sheet of paper (a quad) you might say there's two sides to it, or two faces. But in 3D modelling, a quad only has one face, and it is on the side that the normal is facing.

Texture Map: In 3D modelling, a texture map is a flat image designed in such a way that it can be wrapped around the model. Imagine cutting a head mask up the back then flattening it.

Albedo: The base texture or colour for a material.

Metallic: How metallic the material should look, from 0 (non-metallic) to 1 (fully metallic). The more metallic it is the shinier it will appear.

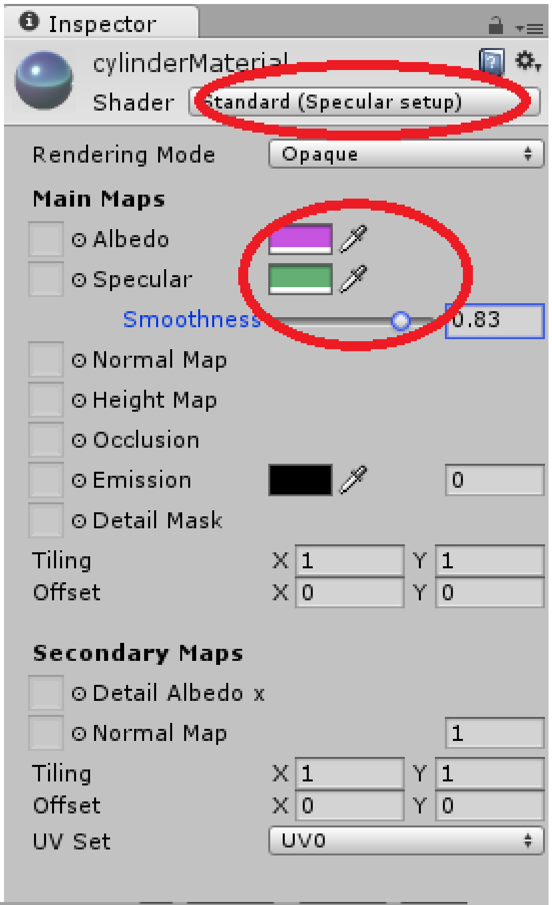
Normal Map: Also known as a bump map. This is a special texture map that dictates where bumps should appear on the material. Using this, you can give the illusion of bumps without actually having to model them. Rugged models such as rocks, guns, and armour use normal maps.

Occlusion: A special texture map that dictates how indirect lighting should be handled on different areas of a model.

Unity Tutorials 3 – Game Objects

Now to apply a material you simply drag it from your assets onto the object you want to apply it to. You can do this either by dragging it onto the object in the scene view or into the hierarchy.

Once it is applied you should be able to see it show up in the inspector.

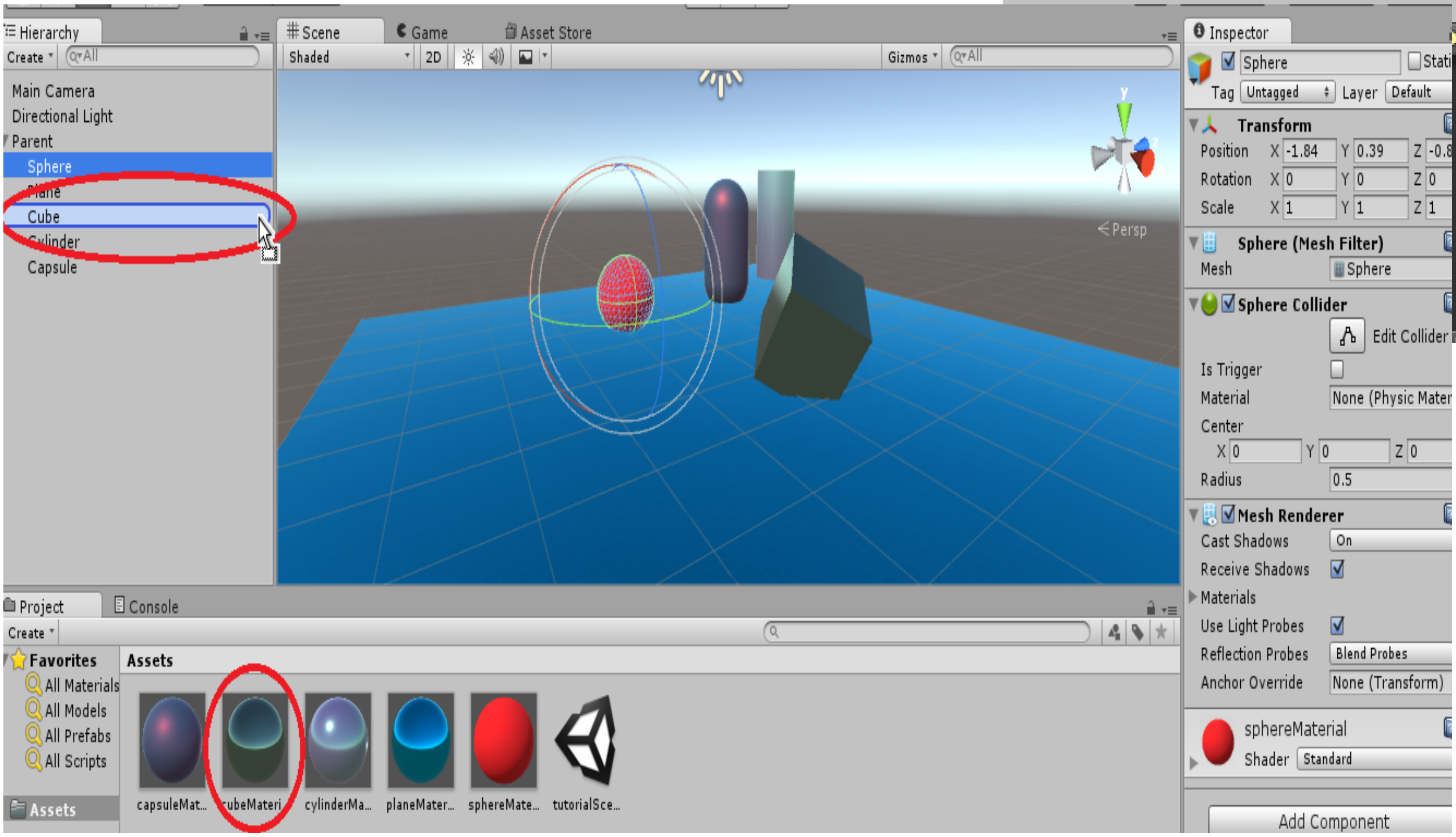


cylinderMaterial
Shader Standard (Specular setup)

Rendering Mode Opaque

Albedo
Specular
Smoothness 0.83
Normal Map
Height Map
Occlusion
Emission 0
Detail Mask
Tiling X 1 Y 1
Offset X 0 Y 0

Detail Albedo x
Normal Map 1
Tiling X 1 Y 1
Offset X 0 Y 0
UV Set UV0



Main Camera
Directional Light
Parent
Sphere
Plane
Cube
Cylinder
Capsule

☑ Sphere □ Stati
Tag Untagged Layer Default

Transform
Position X -1.84 Y 0.39 Z -0.8
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Sphere (Mesh Filter)
Mesh Sphere

☑ Sphere Collider
Edit Collider
Is Trigger □
Material None (Physic Mater
Center
X 0 Y 0 Z 0
Radius 0.5

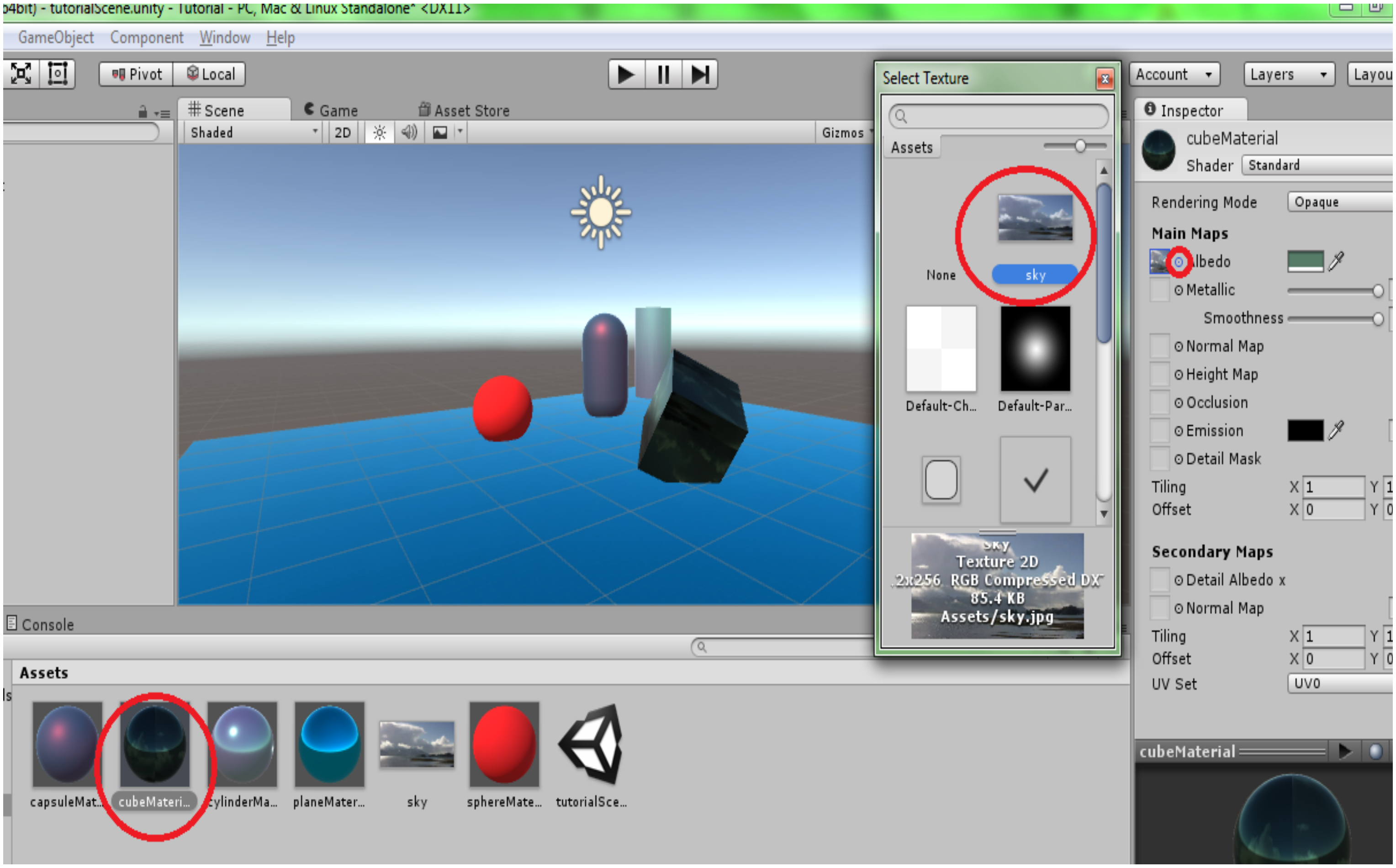
☑ Mesh Renderer
Cast Shadows On
Receive Shadows ☑
Materials
Use Light Probes ☑
Reflection Probes Blend Probes
Anchor Override None (Transform)

sphereMaterial
Shader Standard

Unity Tutorials 3 – Game Objects

Next let's try using a texture instead of a colour. First, find an image on the internet and save it to your Assets folder. Or you can save it somewhere else and import it after.

Select the material you want to apply it to and click on the little circle next to "Albedo". Alternatively, you can drag-and-drop the texture right onto the circle.



sky
Texture 2D
2x256 RGB Compressed DXT
85.4 KB
Assets/sky.jpg

cubeMaterial
Shader Standard
Rendering Mode Opaque
Tiling X 1 Y 1
Offset X 0 Y 0
Tiling X 1 Y 1
Offset X 0 Y 0
UV Set UV0

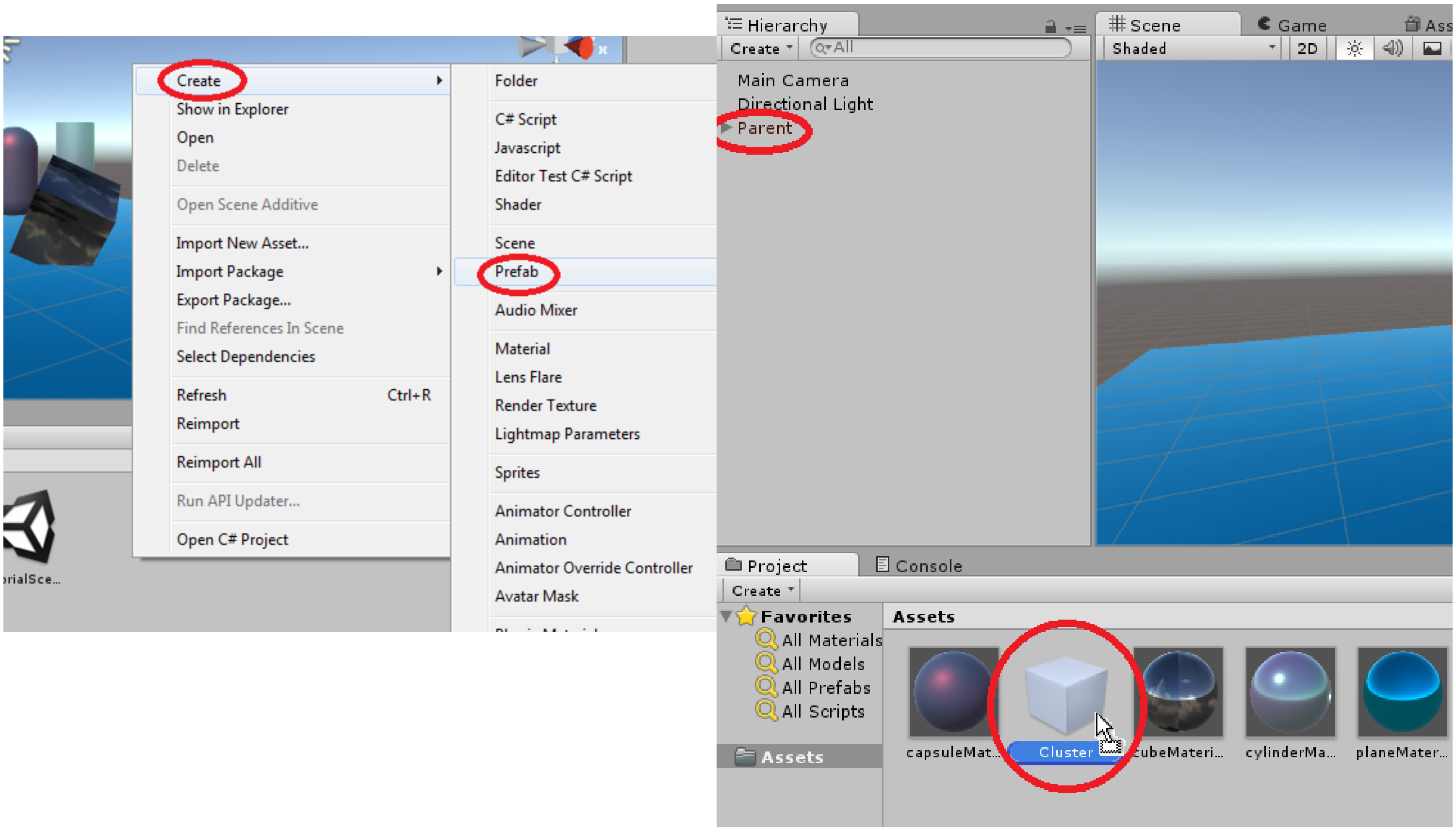
If you clicked the circle, a box will show up listing viable assets. Double-click on the texture that you saved. Remember, since this texture hasn't been mapped to fit the model it will look strange unless it is a pattern. For example, you can safely use a wood pattern as a material for floors or a rock pattern as a material for a boulder. But when it comes to characters, unless you have their bodies split up into numerous objects, a texture map will be required.

Unity Tutorials 3 – Game Objects

III – Prefabs

What is a prefab?
Prefabs are pre-made objects that are stored as an asset. It is wise to always use prefabs when you can.

Making a prefab is very easy. Just like how you created a new material, right-click in the Assets window and under “Create” click “Prefab”.



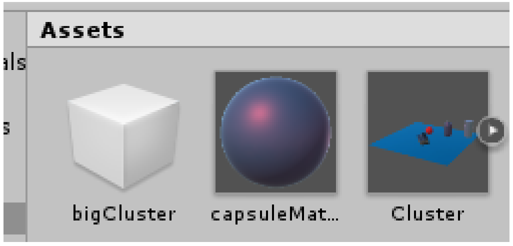
Rename your prefab. Now drag-and-drop your parent object into it.

Notice that the parent object and all of the objects within it are blue in the hierarchy. This means that they were created using a prefab.

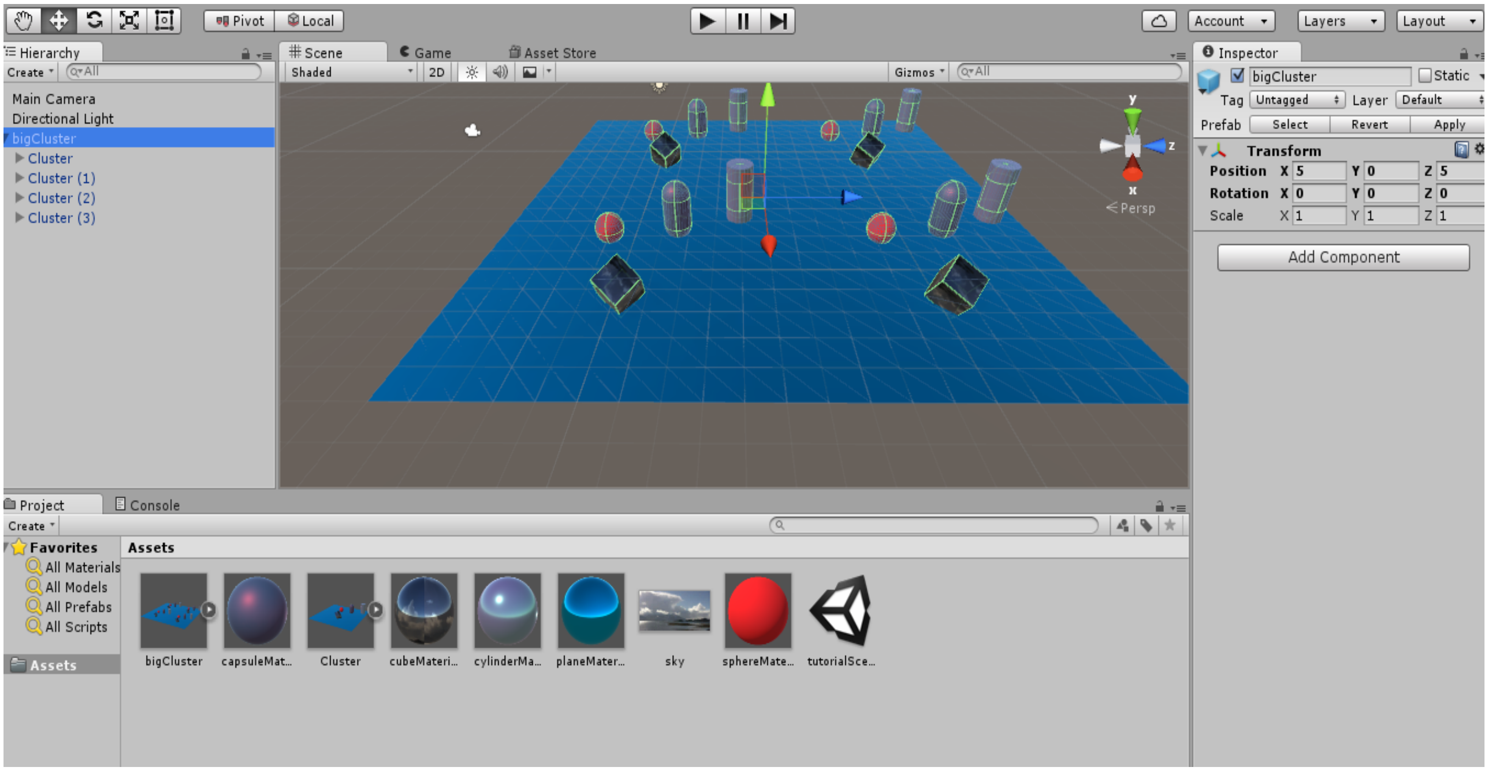
Delete the original parent object. Now drag-and-drop your prefab into the hierarchy or the scene view. This is how you will most often be placing models into your scene. Let's try making an even bigger prefab using this one.

Create a new prefab just as before.

Unity Tutorials 3 – Game Objects



Now drag 3 more prefabs into your scene to make a total of four. After that, move them around so that you make a large square. Now in the hierarchy make an empty object. Before you parent the four prefabs to it you should move it into a proper position. The centre of your large square will be something similar to (5, 0, 5) depending on how you formed it. When that is ready, parent the four clusters to the empty object. Once that is complete you can drag it into the new prefab.



bigCluster ☐ Static

Tag Untagged Layer Default

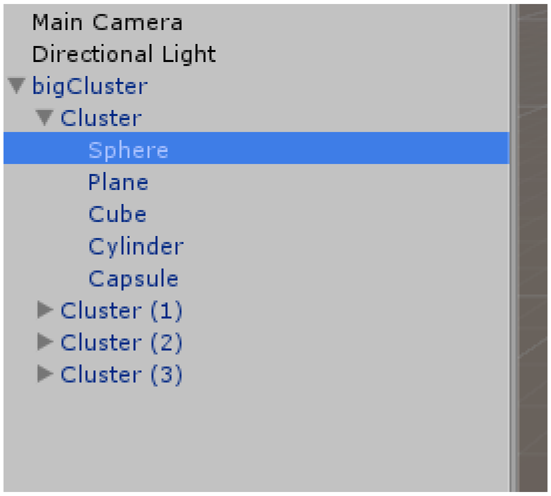
Prefab Select Revert Apply

Transform

	X	Y	Z
Position	5	0	5
Rotation	0	0	0
Scale	1	1	1

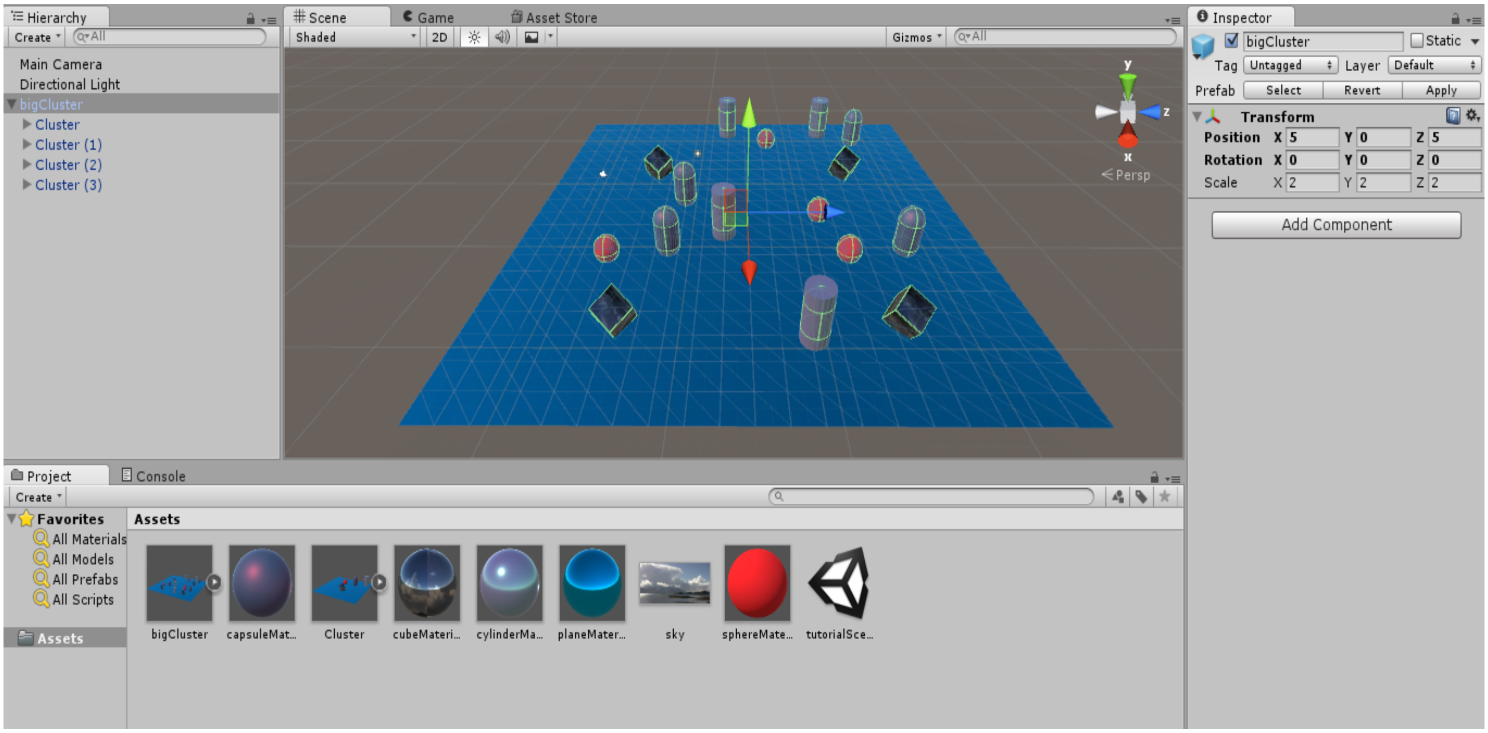
Now let's try rearranging the objects so that the four squares aren't identical. Click the drop-down arrow for one of the four clusters to individually select one of the objects in it.

Unity Tutorials 3 – Game Objects



```
Main Camera
Directional Light
▼ bigCluster
  ▼ Cluster
      Sphere
      Plane
      Cube
      Cylinder
      Capsule
  ▶ Cluster (1)
  ▶ Cluster (2)
  ▶ Cluster (3)
```

Now you can change that object's location separately from all the others. Do this to a few of the objects. Once you are finished, select the top parent (bigCluster) and change the scale to (2, 2, 2). Now drag this object into its prefab in the Asset window again. This will update the prefab.



```
Main Camera
Directional Light
▼ bigCluster
  ▶ Cluster
  ▶ Cluster (1)
  ▶ Cluster (2)
  ▶ Cluster (3)
```

☑ bigCluster					□ Static	
Tag	Untagged		Layer	Default		
Position	X	5	Y	0	Z	5
Rotation	X	0	Y	0	Z	0
Scale	X	2	Y	2	Z	2