02 Reflection Cubemap

Cubemaps define what will be reflected on a Reflective Object. In the real world everything in the environment is reflected. When simulating a complex Scene, simulating the reflection from everything would seriously complicate a render with, perhaps, no practical gain in the final outcome. A Cubemap is used to specify what will be reflected and what will show the reflection.

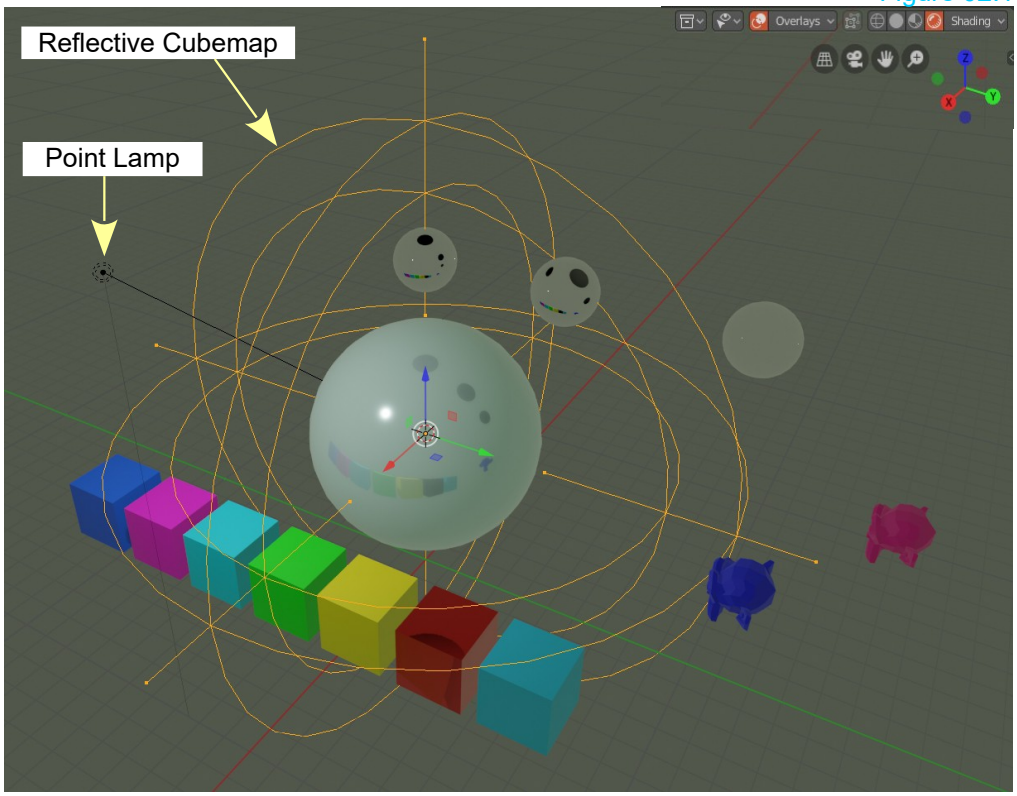
For an Object to reflect light it has to have a smooth metallic surface. Whether the surface is a metal or a shiny smooth plastic depends on the Material settings applied, but for the discussion on reflection consider the **Metallic** and **Roughness** settings in the Material buttons.

When discussing Reflection in general (Supplement 01) it was demonstrated that Reflections could be generated by checking **Screen Space Reflections** in the Properties Editor, Render buttons (Ref: Supplement 01 Reflection). Before investigating Cubemaps have **Screen Space Reflections** deactivated (unchecked). With this feature active, Reflections will be duplicated confusing the demonstration.

Remember for an Object to reflect it is best with a Metallic smooth Material.

Figure 02.1 shows an arrangement in the 3D View Editor in User Perspective View. The Viewport Shading mode is Rendered. The World Background Color has been modified.

Figure 02.1



Before explaining the significance of the Objects in Figure 02.1 consider the Reflective Cubemap itself.

The **Reflective Cubemap** is entered into the Scene from the **Add Menu – Light Probe** category (Figure 02.2). By default a Reflective Cubemap is entered in the Scene as shown in Figure 02.3. This is with the 3D View Editor in User Perspective View. Changing the 3D View Editor to Top, Front or Right Orthographic View shows the Cubemap as two concentric circles (Figure 02.4).

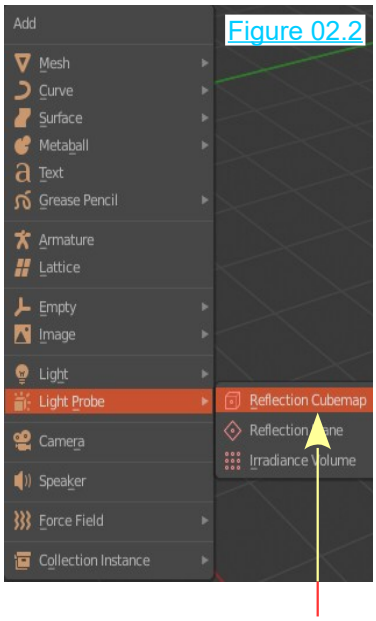


Figure 02.2

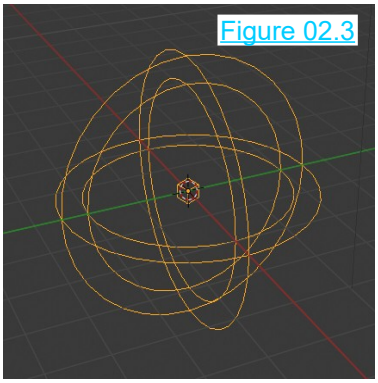


Figure 02.3

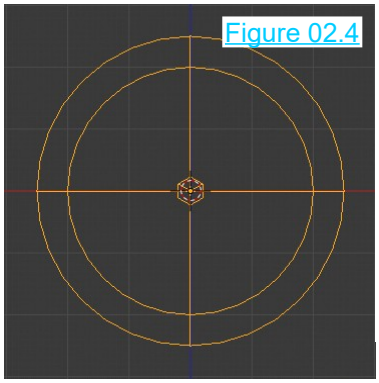


Figure 02.4

User Perspective View Top, Front, Right Orthographic Views Click to Select

Why Cubemap? The concentric circles are defining a volume of space in the Scene. The volume is, in fact, a cubic volume rather than a spherical volume. To demonstrate this, go to the Properties Editor, Object Data buttons (with the Reflective Cubemap selected in the 3D View Editor) and change **Probe Type: Sphere** to **Box** (Figure 02.5). **Size** adjusts the Cubemap size.

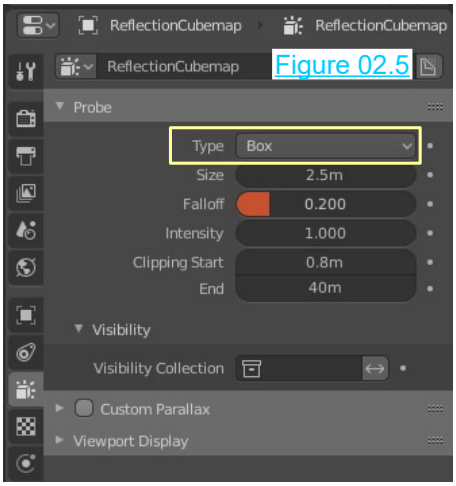


Figure 02.5

The Reflective Cubemap displays as a cube (Figure 02.6) which is what has to be considered when positioning Objects within the spatial volume. For example, in Figure 02.1 the seven colored Cubes would appear to extend outside the volume of the spherical Cubemap but they are within the volume when the Cubemap displays as a cubic volume.

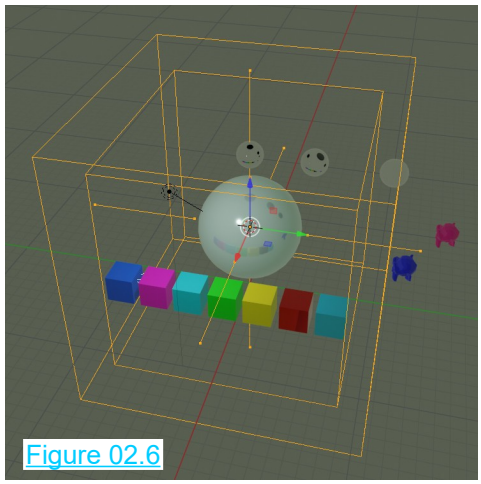


Figure 02.6

OK! Why the consideration to this spherical or cubic volume of space?

Before proceeding, make note that the colored cubes in Figure 02.6 were used simply to demonstrate the difference in positioning Objects in the spherical representation of the Cubemap and the cubic representation.

When considering Reflection there are two types of Objects; those that are being reflected and those that are reflecting. In Figure 02.6 the cubes are being reflected, the spheres are reflecting. You see the reflection of the cubes on the surface of the spheres.

For the purpose of this discussion and to make the distinction between Reflected Objects and Reflecting Objects the expression Primary Object will be used in place of Reflected Object.

Think of it this way. Light from a source bounces off an Object. This is the Primary Object. The light travels to the surface of a second Object, the Reflecting Object, which has a shiny surface, which reflects the light causing you to see a reflection of the Primary Object on the surface of the second Reflecting Object.

The Reflective Cubemap is one type of Light Probe defining a volume of space in a Scene. Only Objects within the bounds of the Cubemap will participate in a Render of Reflections in the Scene. You position and Scale Cubemaps to to encapsulate the Objects you wish to have participate in Rendering Reflection.

Bear in mind, you may have more than one Light Probe of different types in a Scene.

There is one more factor to consider before proceeding.

As well as defining a volume of space (spherical or cubic) a Cubemap has another set of limits called **Clipping Limits**. Clipping Limits are displayed as adjustable horizontal and vertical lines with small dots at each end (Figure 02.7). The ends of the lines also delineate a cubic volumes.

Outer Volume (White Cube)

Clipping Limits (Red Lines)

Inner Volume (Blue Cube)

The dots at the ends of the red lines are at the center of the faces of the inner and outer volumes.

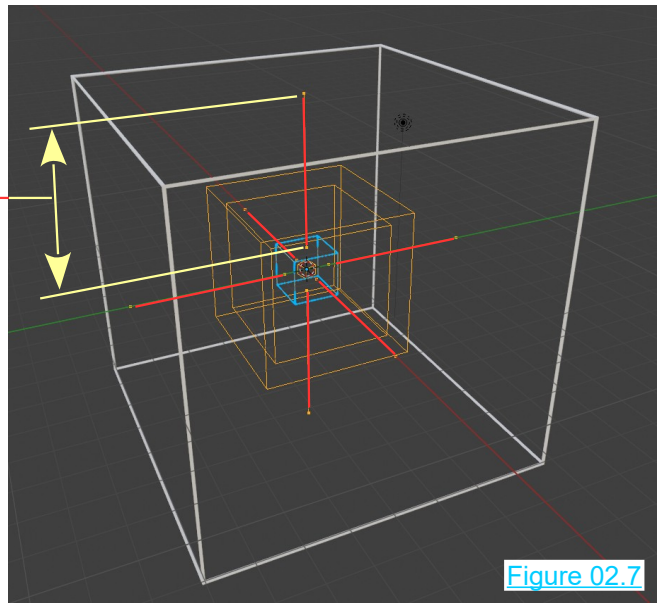


Figure 02.7

To see the **Clipping Limits** in the Viewport go to the Properties Editor, Object Data buttons, with the Cubemap selected. Open the Viewport Display Tab and check Clipping in the lower right hand corner (Figure 02.8).

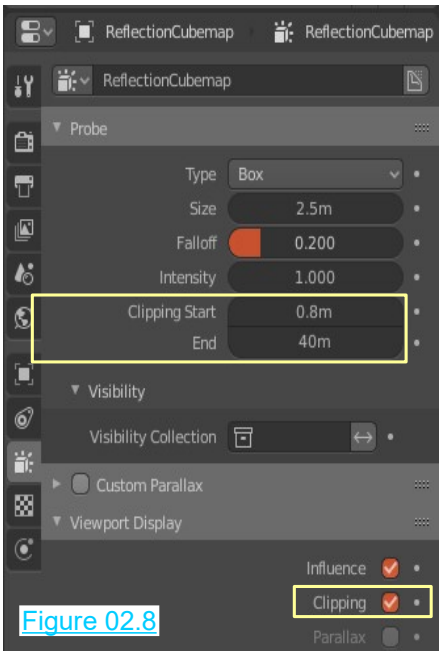


Figure 02.8

To adjust the Clipping Limits alter the values in the Clipping Start and End sliders.

What Goes Where?

Reflecting Objects have to be inside the Cubemap volume.

Primary Objects have to be inside the Clipping Limits.

To place everything in perspective, consider Figure 02.9 which is the arrangement shown in Figure 02.1 having a Box Type Cubemap, instead of the default Sphere Type.

The three UV Spheres in the Scene all have the default gray Material with a Metallic value: 1.000 and Roughness: 0.000 making them **Reflective Objects**.

The seven Cubes have simple Diffuse Material colors as do the two Monkey Objects. They are not Reflective. These are the Primary Objects. There is a single Point Lamp in the Scene.

All the colored Cubes are within the Clipping Limits and show reflected on the surface of the central sphere. These same reflections show on two of the smaller spheres since they are within the volume of the Reflective Cubemap. These two spheres also receive the reflection of the large sphere. The third small sphere is in the **Falloff** area nearly outside the volume of the Reflective Cubemap and, therefore, receives no reflection.

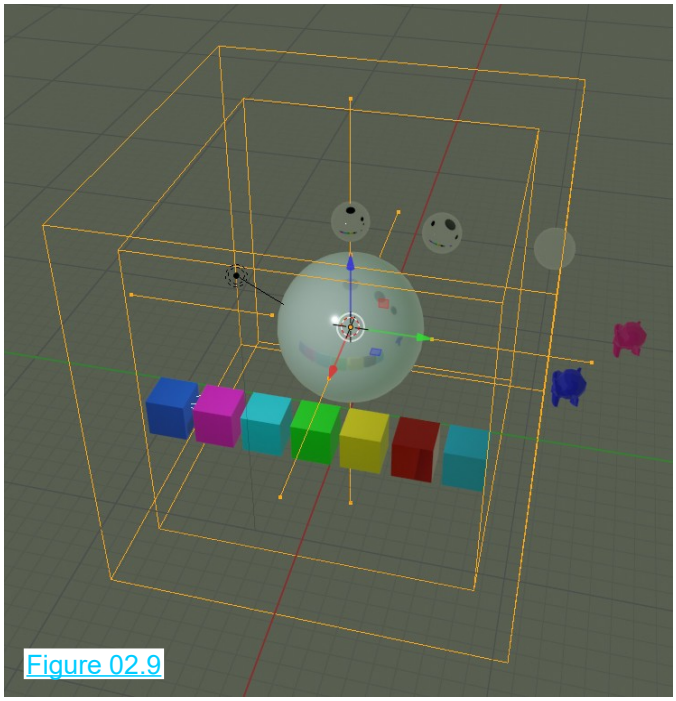


Figure 02.9

The blue Monkey casts a reflection since it is within the Clipping Limits. The red Monkey is outside the Clipping Limit and does not reflect.

Note: This Box Cubemap is not the same as the cubic volume defining Clipping Limits.

You have set up the Scene as shown so why aren't you seeing reflections? Remember you turned off Screen Space Reflections. Well, **leave it turned off.** To get reflections using a Reflective Cubemap you have to **Bake the Cubemap.**

Baking

Properties Editor

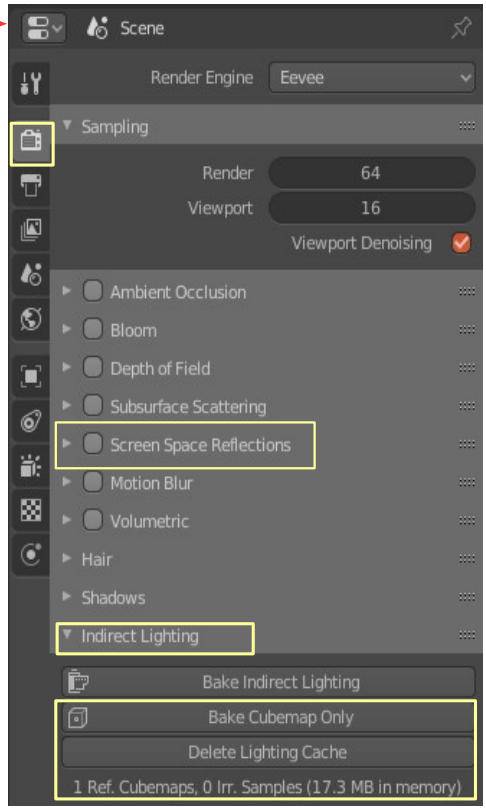


Figure 02.10

Baking is the process of performing the calculations to produce reflections, and saving the data making it available for future use.

With **Screen Space Reflections** active, every change in the Scene would require a recalculation and anything and everything would be involved. In a complicated Scene this would eat up computer power. Baking and storing Cubemap data makes the data available and only calculates reflections for the enclosed volume of the Cubemap.

With the Cubemap set up go to the **Properties Editor, Render buttons, Indirect Lighting Tab** and press **Bake Cubemap Only**. In Figure 02.10 a Cubemap has been previously Baked since you see **Delete Lighting Cache** active and how much memory has been used by the Bake. Click **Delete Lighting Cache** to clear the memory, modify the Cubemap and Rebake after you have made changes.

Figure 02.11 shows another example of using a Reflective Cubemap with a Floor Plane having a Chequer Texture and a UV Sphere within the Cubemap volume.

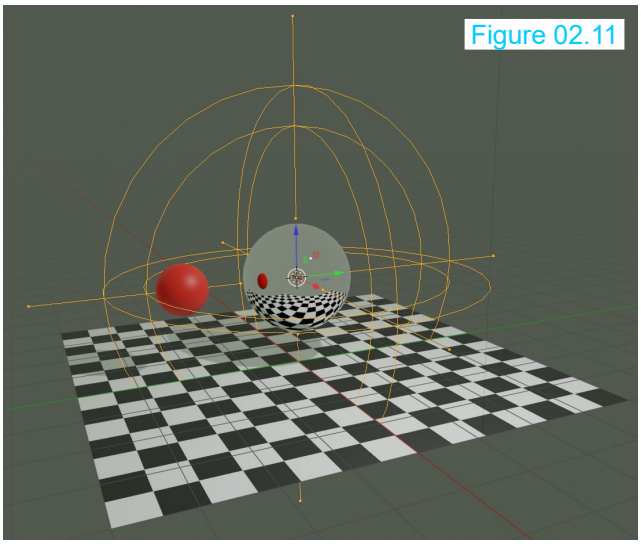


Figure 02.11

The central UV Sphere has a pure white Material color with the Metallic set at 1.000 and Roughness 0.000.

The World Background color is a shade of olive green which is Reflected on the surface of the Sphere.

With the Spheres and the Chequered Floor within the limits of the Cubemap Volume, the red sphere and the floor reflect from the surface of the larger sphere.