# **1 Introduction**

This report will go over the general description of cube mapping, looking at sources of what cube mapping is, and how it is used in currently existing concepts. The report will also explain how cube mapping is achieved in-engine.

# **2 Cube Mapping Review**



**Figure 1.** An illustration depicting how cube mapping is applied to each face of a cube.

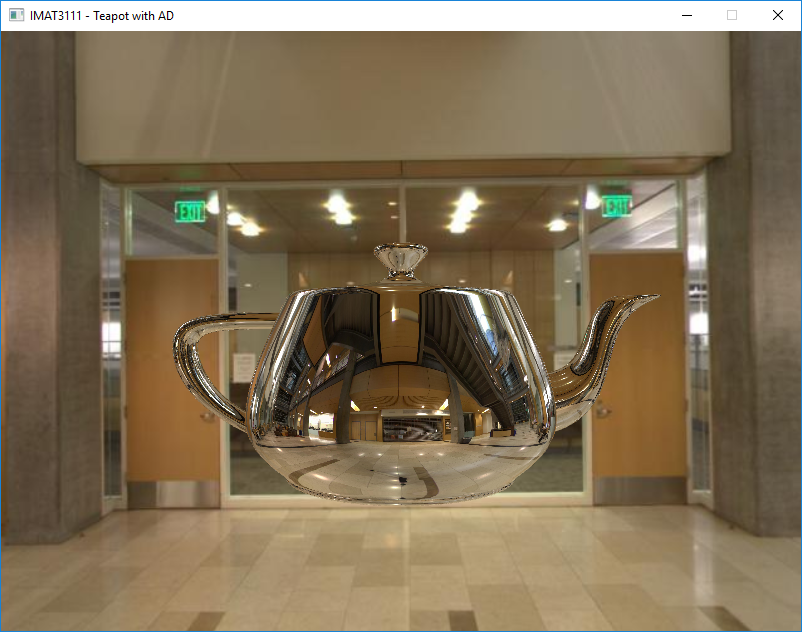
### **2.1 Cube Mapping**

Cube mapping is an environment mapping technique used for reflective objects (Brennan, 2002). The method involves using a cube that encompasses the entire scene within an engine. The cube is then overlayed with images, with one image per face, to form a sort of “skybox” (Niebruegge, 2007). The cube shape is much easier to use compared to other shapes, such as spheres, since the surrounding sides are flat. Providing the cube shape with textures is a lot easier as you would only need 6 flat images taken each at an orthogonal 90-degree view from each side (Niebruegge, 2007). Cube mapping has an advantage over other methods of creating environments as images are projected on flat planes, whereas for example spherical image mapping involves stretching and distorting an image to fit the spherical shape (Tze-Yiu Ho, 2009).

### **2.2 How Cube Mapping is Achieved**

Cube mapping can be created by displaying individual images on each 6 faces of the inside of a cube. The images then combine to form a 360-degree skybox that appears to have no seams between the images. This can then be used to calculate reflections on objects within the scene using shaders that utilise the images displayed on the cube, allowing for some sophisticated reflection effects.

# **3 Code Analysis**



**Figure 2.** The Utah Teapot, seen within the TeapotAd program.

### **3.1 Teapot Ad**

The Utah teapot is an iconic 3D model of a teapot used in various computer-generated research and development (Mendiburu, 2012). The teapot is used within this project to demonstrate reflection and refraction shaders, which are in turn used to demonstrate the effectiveness of cube mapping.

The program has several controls, with the left and right mouse button controlling the camera angle and position, and the scroll wheel controlling the mouse zoom. **The spacebar can be used to swap between reflection and refraction shaders.**

### **3.2 Cube Mapping Code**

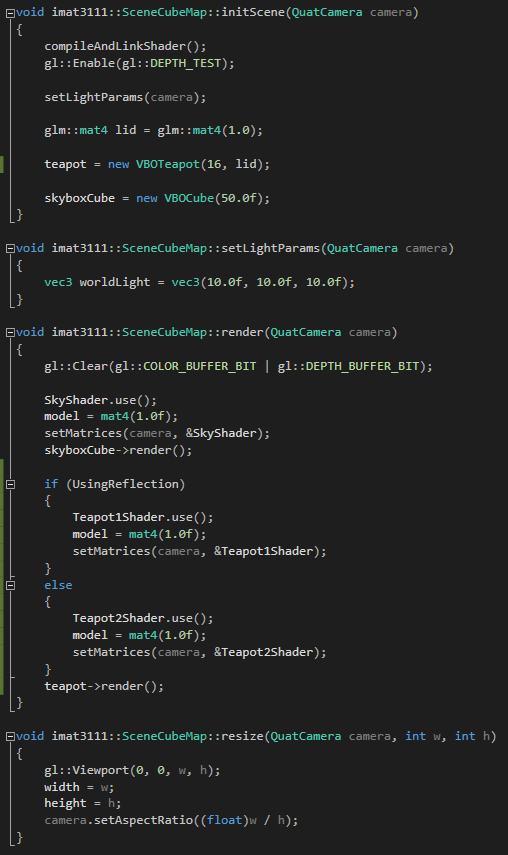


**Figure 3.** Cube Map scene class.

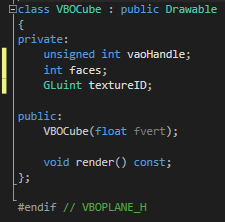


**Figure 4.** Functions used within cube map scene.

The cube map scene is initialised with two fragment shaders loaded in preparation for swapping between reflection and refraction of the teapot, which is swapped via the spacebar button, triggering the activation of the UseReflection Boolean variable.



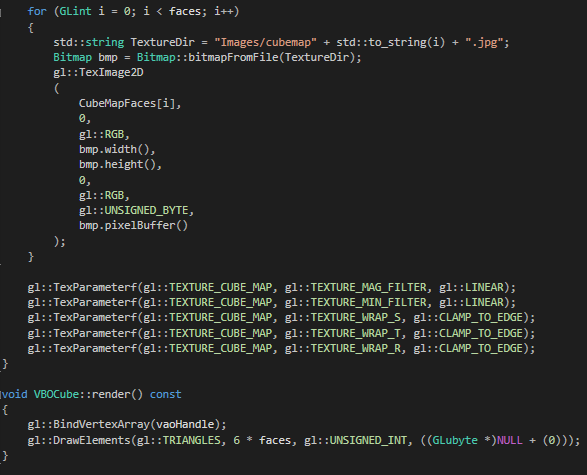
**Figure 5.** Functions used within cube map scene class.



**Figure 6.** Cube object used for cube mapping the skybox within the scene.



**Figure 7.** First half of the cube initialisation function.



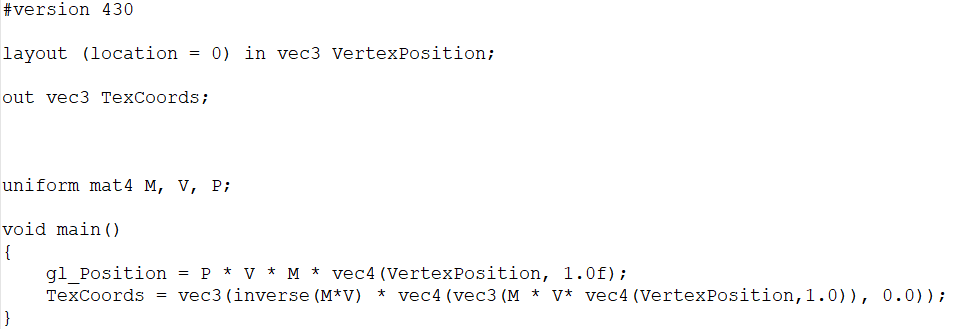
**Figure 8.** Second half of the cube initialisation function, along with the render function.

The cube object initialiser starts by generating blank faces in the 3D environment based on a size input variable. The faces are then textured by individually loading an image per face based on the face number. These images are displayed within the cube in the 3D environment to mimic a skybox.

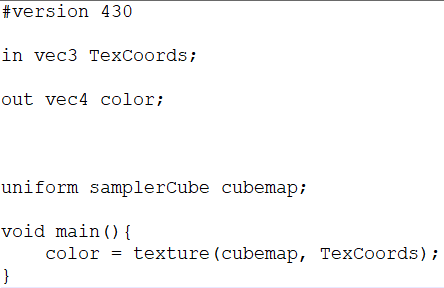


**Figure 9.** Reflection and refraction shaders applied to the teapot side by side.

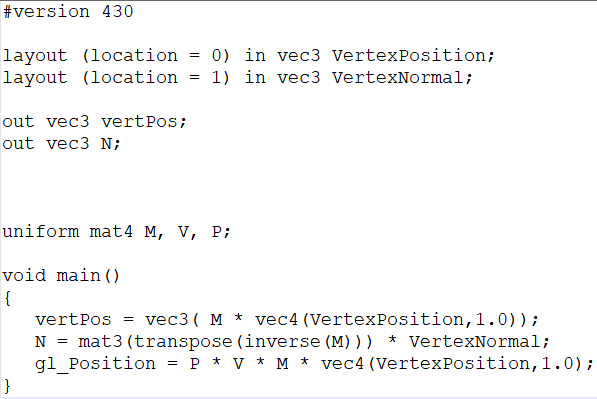
### **3.3 Shaders used**



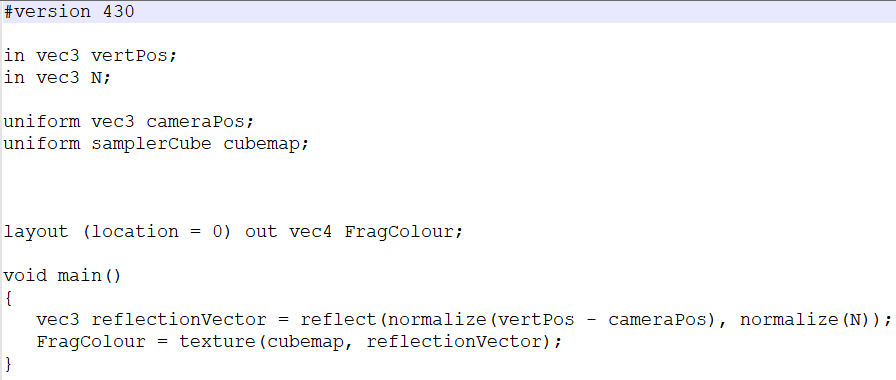
**Figure 10.** Skybox vertex shader.



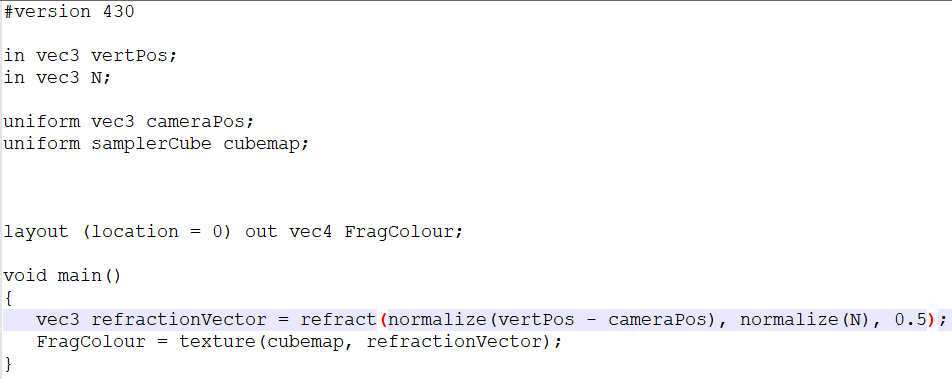
**Figure 11.** Skybox fragment shader.



**Figure 12.** Teapot vertex shader.



**Figure 13.** Reflection fragment shader.



**Figure 14.** Refraction fragment shader.

# **4 Conclusion**

In conclusion, the concept of cube mapping has presented itself as a viable alternative method of creating an environment skybox to other methods such as sphere mapping, as the cube allows for flat images to be used as overlayed textures. The textures can be applied to shaders to generate photo-realistic reflection and refraction effects on objects within a 3D scene.

The Utah Teapot was used to display two shaders, a reflection shader and a refraction shader. These were created and displayed in an OpenGL environment.

# **5 References**

Brennan, C., 2002. Diffuse Cube Mapping. *Direct3D ShaderX: Vertex and Pixel Shader Tips and Tricks,* pp. 287-289.

Mendiburu, B., 2012. Introduction to 3D Cinema. In: *3D movie making: stereoscopic digital cinema from script to screen.* s.l.:CRC Press, pp. 1-2.

Niebruegge, C. M. G. B., 2007. Continuous cube mapping.. *Journal of Graphics Tools,* 12(4), pp. 25-34.

Tze-Yiu Ho, L. W. C.-S. L. P.-M. L. T.-T. W., 2009. Unicube for Dynamic Environment Mapping. *IEEE Transactions on Visualization and Computer Graphics,* 17(1), pp. 51-63.