The Garden

Richard Borbely
40283185@napier.ac.uk
Edinburgh Napier University - Computer Graphics (SET08116)

Abstract

This paper describes the various features implemented in a 3D scene using OpenGL with the provided libraries. Features include the use of various light types, cameras, texturing, shadowing and other more advanced techniques that required further research such as reflections, refractions, modified Fresnel effect and multiple post-processing effects that can be applied using a graphical interface. **Keywords –** C++, OpenGL, GLSL

1 Introduction

'The Garden' The background story of the scene is about a glass statue who is 'placing' tiny planets into orbit above a marble textured pool. The main attraction of the scene is surrounded by tall hedges, resembling a garden during night time.
Most interesting features are:

- GUI for key bind information and Post-Process effects
- Dynamic shadow
- Reflection and refraction using cubemap
- Fresnel effect

2 Related Work

The provided workbook has been extremely useful in explaining most of the features implemented in the project. To understand the Fresnel effect, I have read this [1] online article which states that a transparent object reflects or refracts light depending on the angle of the viewpoint.

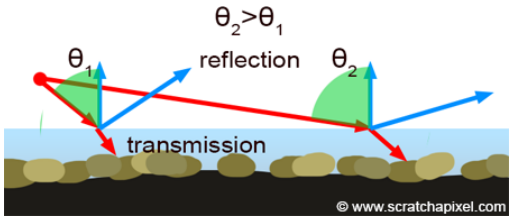


Figure 1: "The ratio of reflected light increases as the angle between the view direction and the surface normal increases"

The effect has been applied to the 'water' in the pool, but to visually benefit more from it I had to take refraction out of the equation. It is shown on Figure 1 that a wider view angle from the surface normal results in more reflection, so the clouds will start to become visible. To make this even more appealing the reflection amount is clamped in the shader, so the surface never becomes completely reflective.

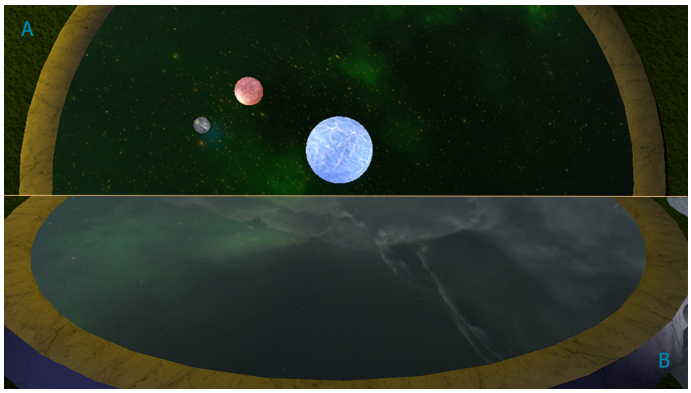


Figure 2: A - Camera looking down perpendicularly, B - Camera looking from the side

3 Implementation

3.1 Skybox The simplest way to change the atmosphere of the scene is by adding a skybox, which is essentially done by creating a large enough cube that engulfs the scene and rendering, texturing the inner sides. This is then called a cubemap, which is later on used for reflections or refractions. The active camera must always be in the center of the cubemap so that the user never feels like they are inside a cube.

3.2 Lights Directional light is used to simulate the moonlight in the scene, this type of light has a constant direction.
Point lights originate from a point and emit light to all directions. They are used for the lamp posts in the scene.
Spot lights are similar but the light rays are restricted in a cone shape. These can be found under the planets in the pool.
All light types use Phong shading to calculate the color of a pixel depending on the material properties. The diffuse component will define the objects base colour and

the specular component defines the area where light is reflected directly into the camera.

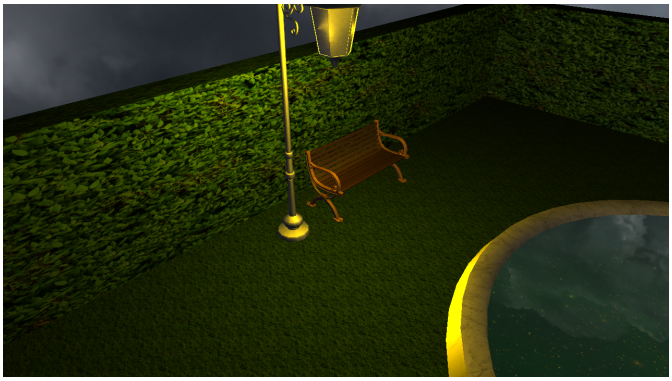


Figure 3: Point lights

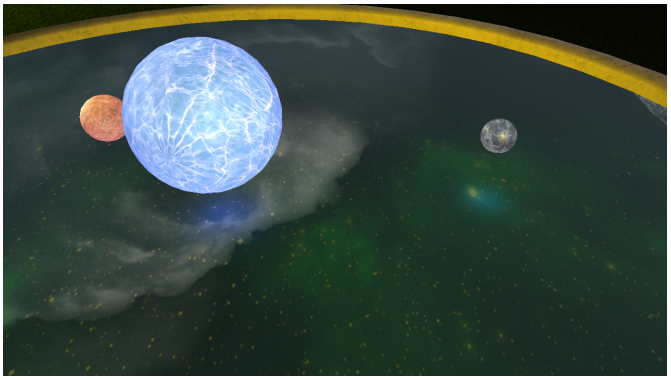


Figure 4: Spot lights

3.3 Shadows Rendering shadows for spotlights is done by first rendering the scene from the perspective of the light and storing the depth information of each rendered object in a 'shadow map'. This shadow map is then compared with a regularly rendered frame to calculate where shadows should be cast by the light source. Moving the light source and repeating the process each frame results in a dynamic shadow which can be very hard on performance.

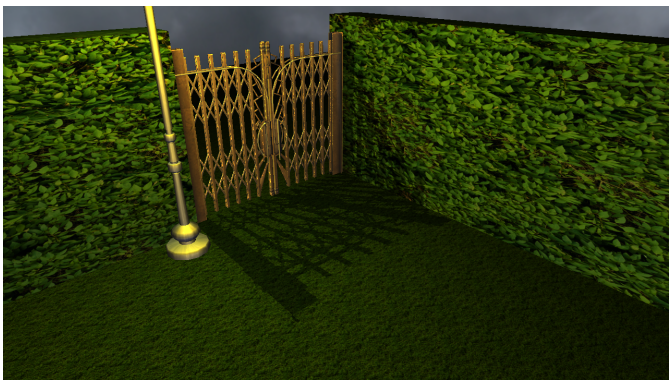


Figure 5: Spot light shadow

3.4 Refraction When light rays pass from one transparent medium (eg. air) to another (eg. water), they change direction. The change in direction is different for each medium and is measured by their refractive index. When calculating the refraction vector in the shader, these indexes need to be included.

$$refractedVector = refract(viewVector, normal, 1.0/1.33);$$

The refracted vector is then used to sample from a cubemap, resulting in a transparent look.

$$refractedColour = texture(cubeMap, refractedVector);$$
$$colour = mix(colour, refractedColour, refractionAmount);$$

Figure 6: Refraction applied to the statue object

3.5 Post Processing To manipulate the post processing effects easily a graphical menu is included. The menu functionality is quite simple; it consists of 5 images for each effect, when the user moves UP or DOWN on the menu list, the selected effect's image used for the menu. LEFT or RIGHT will change the effect's properties, both single press or button hold is registered.

Figure 7: Graphical user interface

The scene is rendered to a frame buffer and then used as a texture in the post processing shader to apply the effects on top of the rendered frame.

Saturation refers to the intensity of colours in an image. An image that is highly saturated has vivid, intense colours, without saturation it becomes greyscale. Greyscale is when pixels only have light intensity information.

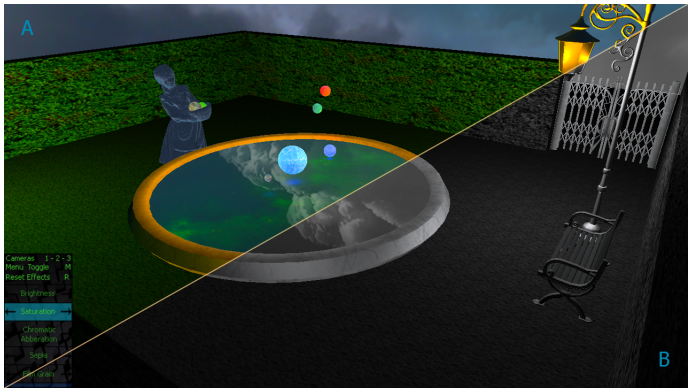


Figure 9: A - 3x more colourful, B - Greyscale

Chromatic Abberation in optics happens when the lens is unable to focus all colours to the same convergence point. The effect is getting some attention in the video game industry to make a scene more photo realistic. It is very easy to compute, this is a simple implementation of it by sampling Red and Blue pixels from a slightly shifted location.

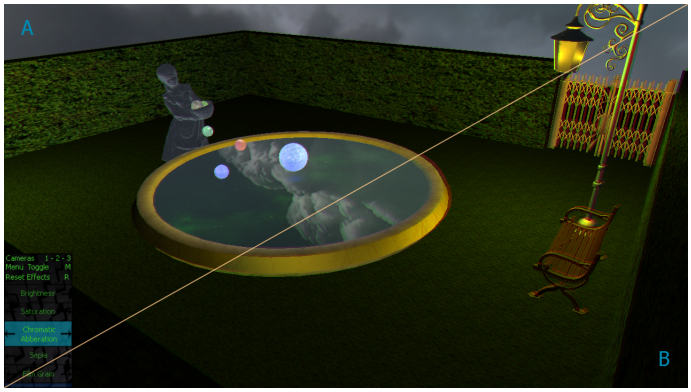


Figure 10: A - slight shift in pixels , B - noticeable shift in pixels

Film Grain is another effect that can be used to simulate an artistic look. It is done by either sampling a noise texture or generating it and applying it to the image. Here it is being generated using the elapsed time as the seed, then going through various calculations that give a consistent and stable noise.

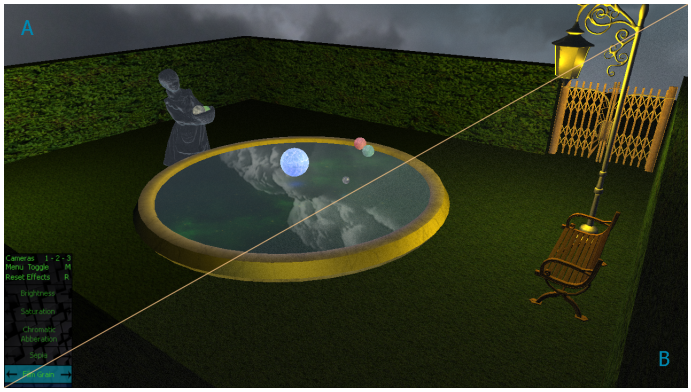
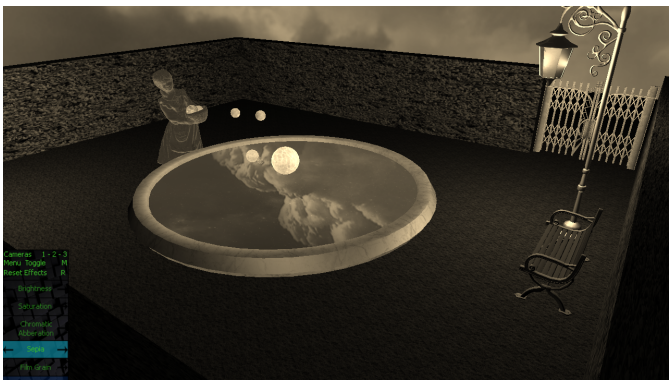


Figure 11: A - small amount of grain, B - noticeable grain



Figure

Figure 8: Sepia effect

Sepia Sepia tone is often used in photography because it resembles the ageing of old photographs.

4 Conclusion

Naively starting out with a scene it is very simple to get lost in it's inner workings. Making something complex is difficult, time consuming and requires a lot of research. However, once the foundations are laid down it becomes a very enjoyable activity. Working on this project has definitely made me a lot more interested in computer graphics and I am eager to test my knowledge in a well refined engine like Unreal Engine.

References

[1] Reflection, refraction (transmission) and fresnel. [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/reflection-refraction-fresnel