**ICG HW3 Report**

**Bling-Phong Shader**

**Fragment shader: vertex shader:**

**一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 軟體, 字型 的圖片

自動產生的描述**

For bling-phone shading, the picture on the left-hand side is fragment shader, and the picture on the right-hand side is vertex shader.

Bling-phone shading is to calculate the halfway vector between viewer and light vector, and use dot(N, H) to replace dot(R, V) to reduce the complexity of calculation.

In vertex shader, I simply convert the model from model view to world view, and pass world position, fragment position and fragment normal vector to fragment shader.

In fragment shader, calculate the light vector and view vector using light position, fragment position and viewer position first, and calculate the halfway vector using light and view vector, then generate model color using model texture. Calculate ambient, diffuse and specular using the following formula:

and add them together to produce the final fragment color.

**Gouraud Shader**

**一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述Vertex shader Fragment shader**

**一張含有 文字, 螢幕擷取畫面, 陳列, 電腦 的圖片

自動產生的描述**

For gouraud shading, the picture on the left-hand side is vertex shader, and the picture on the right-hand side is fragment shader.

Gouraud shading is a shading method which implements phong lighting model at each vertex, so we have to complete the calculations of ambient, diffuse and specular in vertex shader using the formula provided in the picture below, and pass these three parameters and texture coordinate to fragment shader.

一張含有 文字, 字型, 白色, 收據 的圖片

自動產生的描述

In fragment shader, get model color using model texture, and multiply ambient and diffuse passed from vertex shader with model color. At the end, add up the final ambient, diffuse and specular to create the final fragment color.

**Environment Cubemap**

**main**

**一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述**

**Vertex shader fragment shader**

**一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述**一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

The render environment cubemap, pass the necessary uniform values to the shader, bind the cubemap texture and render cubemap in main.

The picture on the left-hand side is vertex shader, and the picture on the right-hand side is fragment sharder.

In vertex shader, calculate pos to transform the vertex from model view to world view, and let gl\_position equals pos.xyww. Thus, we ensure that the skybox appears infinitely far away, which x and y still undergo perspective scaling, and z is fixed to a constant value 1.0, which ensure the skybox depth is always farthest, and w controls the perspective division, ensure the cube’s position respects the camera’s perspective while always enclosing the viewer.

In fragment shader, simply generate the fragment color using the cube texture and texture coordinate passed by the vertex shader.

**Metallic Shader**

**一張含有 文字, 螢幕擷取畫面, 軟體, 作業系統 的圖片

自動產生的描述**

Because the vertex shader of metallic shading is identical to the vertex shader of Bling-Phong shading, so I am only going to explain fragment shader in this part.

The provided picture is the fragment shader, and to complete metallic shading, we have to calculate the reflection of environment and model color by the corresponding texture.

The reflection of environment color is computed using the formula: , and environment color = texture(envTexture, reflection).rgb.

After computing the value of B using the formula . complete the final fragment color by combining the model color and reflection of environment color together using the formula: .

**Glass Shader – Schlick Approximation**

**一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

自動產生的描述**

Because the vertex shader of glass shading is identical to the vertex shader of Bling-Phong shading, so I am only going to explain fragment shader in this part as well.

To complete glass shading with schlick approximation, we have to complete the reflection of environment color and the refraction of environment color.

The method of calculating reflection color is explained in metallic shading, and the method to complete refraction color is to follow the formula: , and refractionColor = texture(envTexture, T).

After complete refraction color and reflect color, calculate using the formula

一張含有 文字, 字型, 白色, 行 的圖片

自動產生的描述

And compute fragColor as \* reflectColor + (1- ) \* refraColor.

**Glass Shader – Empricial Approximation**

**一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述**

Because the vertex shader of glass shading is identical to the vertex shader of Bling-Phong shading, so I am only going to explain fragment shader in this part.

To complete glass shading with empricial approximation, we have to complete the reflection of environment color and the refraction of environment color. and the method of complete reflection color and refraction color is already explained in the above scope.

After the reflection color and refraction color is completed, we have to calculate using the formula: . And compute the fragColor fragColor as \* reflectColor + (1- ) \* refraColor.