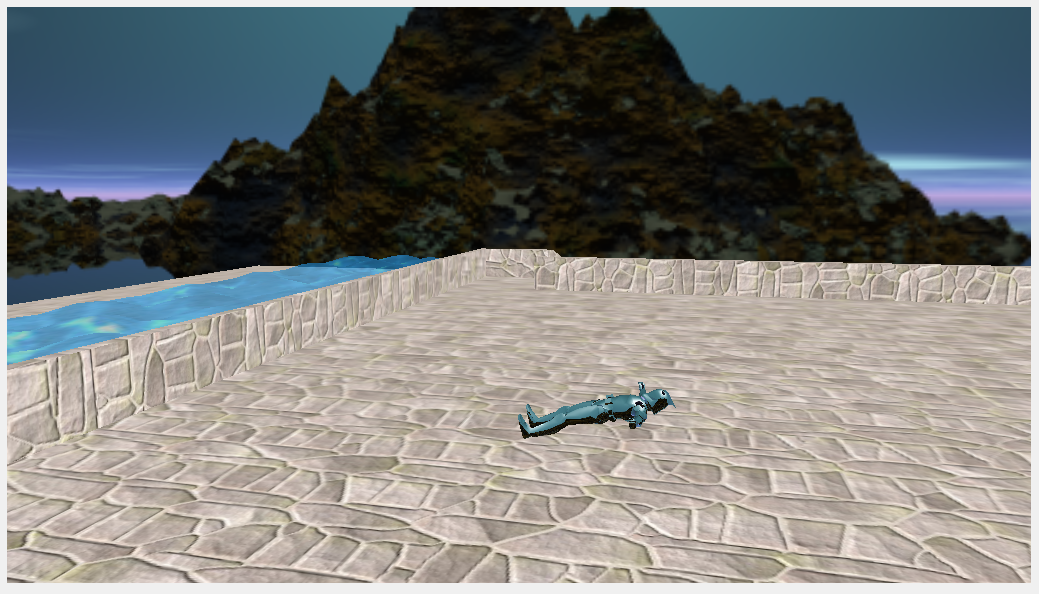
Lab 04 – Cube Maps – Joshua Kauer

The first thing I did was make a sky box. This sky box takes in a cube map made of 6 .bmp images and then put into a cube map. I also only use the position of the cube to actually determine the placement of the textures. Afterwards I made a shader that would take in the skybox and refract or reflect it as a texture on the object. This is shown with the clone trooper. Currently he is refracting the skybox. I think next I may make it so the shader will take in the entire environment and refract it making it look like he has stealth. I also have the clone trooper constantly rotating to help show that the “texture” is constantly being updated to refract what is new and isn’t static.







//Skybox Shader

#version 430 core

layout (location = 0) in vec3 position;

out vec3 TexCoords;

uniform mat4 projection;

uniform mat4 view;

void main()

{

vec4 pos = projection \* view \* vec4(position, 1.0);

gl\_Position = pos.xyww;

TexCoords = vec3(position.x, -position.y, position.z);

}

#version 430 core

in vec3 TexCoords;

out vec4 color;

uniform samplerCube skybox;

void main()

{

color = texture(skybox, TexCoords);

}

//Refract/Reflect Shader

#version 430 core

layout (location = 0) in vec4 position;

layout (location = 1) in vec3 normal;

out vec3 Normal;

out vec3 Position;

uniform mat4 model;

uniform mat4 view;

uniform mat4 projection;

void main()

{

gl\_Position = projection \* view \* model \* position;

Normal = mat3(transpose(inverse(model))) \* normal;

Position = vec3(model \* position);

}

#version 430 core

in vec3 Normal;

in vec3 Position;

out vec4 color;

uniform vec3 cameraPos;

uniform samplerCube skybox;

//Reflect

//void main()

//{

// vec3 I = normalize(Position - cameraPos);

// vec3 R = reflect(I, normalize(Normal));

// color = texture(skybox, R);

//}

//Refract

void main()

{

float ratio = 1.00 / 1.52;

vec3 I = normalize(Position - cameraPos);

vec3 R = refract(I, normalize(Normal), ratio);

color = texture(skybox, R);

}