Lab 7

Setting up a skybox…

Change the Texture.h as follows

Add the following methods…

GLuint loadCubemap(std::vector<std::string> faces);

GLint getTexHandler() { return textureHandler; }

Addthe following method to the texture.cpp

GLuint Texture::loadCubemap(std::vector<std::string> faces)

{

glGenTextures(1, &textureHandler);

glBindTexture(GL\_TEXTURE\_CUBE\_MAP, textureHandler);

int width, height, nrChannels;

for (unsigned int i = 0; i < faces.size(); i++)

{

unsigned char \*data = stbi\_load(faces[i].c\_str(), &width, &height, &nrChannels, 0);

if (data)

{

glTexImage2D(GL\_TEXTURE\_CUBE\_MAP\_POSITIVE\_X + i,

0, GL\_RGB, width, height, 0, GL\_RGB, GL\_UNSIGNED\_BYTE, data

);

stbi\_image\_free(data);

}

else

{

std::cout << "Cubemap texture failed to load at path: " << faces[i] << std::endl;

stbi\_image\_free(data);

}

}

glTexParameteri(GL\_TEXTURE\_CUBE\_MAP, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_CUBE\_MAP, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_CUBE\_MAP, GL\_TEXTURE\_WRAP\_S, GL\_CLAMP\_TO\_EDGE);

glTexParameteri(GL\_TEXTURE\_CUBE\_MAP, GL\_TEXTURE\_WRAP\_T, GL\_CLAMP\_TO\_EDGE);

glTexParameteri(GL\_TEXTURE\_CUBE\_MAP, GL\_TEXTURE\_WRAP\_R, GL\_CLAMP\_TO\_EDGE);

return textureHandler;

}

In the maingame.h add the following…

void Skybox();

GLuint skyboxVAO, skyboxVBO, cubemapTexture;

vector<std::string> faces;

Texture skybox;

Shader shaderSkybox;

In the maingame add the following to the init method…

vector<std::string> faces

{

"..\\res\\skybox\\right.jpg",

"..\\res\\skybox\\left.jpg",

"..\\res\\skybox\\top.jpg",

"..\\res\\skybox\\bottom.jpg",

"..\\res\\skybox\\front.jpg",

"..\\res\\skybox\\back.jpg"

};

cubemapTexture = skybox.loadCubemap(faces); //Load the cubemap using "faces" into cubemapTextures

float skyboxVertices[] = {

// positions

-6.0f, 6.0f, -6.0f,

-6.0f, -6.0f, -6.0f,

6.0f, -6.0f, -6.0f,

6.0f, -6.0f, -6.0f,

6.0f, 6.0f, -6.0f,

-6.0f, 6.0f, -6.0f,

-6.0f, -6.0f, 6.0f,

-6.0f, -6.0f, -6.0f,

-6.0f, 6.0f, -6.0f,

-6.0f, 6.0f, -6.0f,

-6.0f, 6.0f, 6.0f,

-6.0f, -6.0f, 6.0f,

6.0f, -6.0f, -6.0f,

6.0f, -6.0f, 6.0f,

6.0f, 6.0f, 6.0f,

6.0f, 6.0f, 6.0f,

6.0f, 6.0f, -6.0f,

6.0f, -6.0f, -6.0f,

-6.0f, -6.0f, 6.0f,

-6.0f, 6.0f, 6.0f,

6.0f, 6.0f, 6.0f,

6.0f, 6.0f, 6.0f,

6.0f, -6.0f, 6.0f,

-6.0f, -6.0f, 6.0f,

-6.0f, 6.0f, -6.0f,

6.0f, 6.0f, -6.0f,

6.0f, 6.0f, 6.0f,

6.0f, 6.0f, 6.0f,

-6.0f, 6.0f, 6.0f,

-6.0f, 6.0f, -6.0f,

-6.0f, -6.0f, -6.0f,

-6.0f, -6.0f, 6.0f,

6.0f, -6.0f, -6.0f,

6.0f, -6.0f, -6.0f,

-6.0f, -6.0f, 6.0f,

6.0f, -6.0f, 6.0f

};

//use openGL functionality to generate & bind data into buffers

glGenVertexArrays(1, &skyboxVAO);

glGenBuffers(1, &skyboxVBO);

glBindVertexArray(skyboxVAO);

glBindBuffer(GL\_ARRAY\_BUFFER, skyboxVBO);

glBufferData(GL\_ARRAY\_BUFFER, sizeof(skyboxVertices), &skyboxVertices, GL\_STATIC\_DRAW);

glEnableVertexAttribArray(0);

glVertexAttribPointer(0, 3, GL\_FLOAT, GL\_FALSE, 3 \* sizeof(float), (void\*)0);

}

Add the following method…

void MainGame::Skybox()

{

glDepthFunc(GL\_LEQUAL); // change depth function so depth test passes when values are equal to depth buffer's content

shaderSkybox.Bind();

shaderSkybox.setInt("skybox", 0);

//view = glm::mat4(glm::mat3(camera.GetViewMatrix())); // remove translation from the view matrix

shaderSkybox.setMat4("view", myCamera.GetView());

shaderSkybox.setMat4("projection", myCamera.GetProjection());

// skybox cube

glBindVertexArray(skyboxVAO);

glActiveTexture(GL\_TEXTURE0);

glBindTexture(GL\_TEXTURE\_CUBE\_MAP, cubemapTexture);

glDrawArrays(GL\_TRIANGLES, 0, 36);

glBindVertexArray(0);

glDepthFunc(GL\_LESS); // set depth function back to default

}

TODO..

Initialise the skybox shader

Draw the skybox

Run the code, find and fix the error!