**GRAPHIC PROCESSING PROJECT**

Name: Buzan Meda

Group: 30431

Contents

⦁ Subject specification

⦁ Scenario

⦁ Scene and objects description

⦁ Functionalities

⦁ Implementation details

⦁ Functions and special algorithms

⦁ Possible solutions

⦁ The motivation of the chosen approach

⦁ Graphics model

⦁ Data structures

⦁ Class hierarchy

⦁ Graphical user interface presentation/user manual

⦁ Conclusions and further developments

⦁ References

⦁ **Subject specification**

The subject of the project consists in the photorealistic presentation of 3D objects using OpenGL library. OpenGL Mathematics(GLM) is a C++ mathematics library for 3D software based on the OpenGL Shading Language(GLSL) specification.

The scene that I implemented represents a summer island. The user directly manipulates by mouse and keyboard inputs the scene of objects.

⦁ Scenario

⦁ Scene and object description

The scene depicts a small tropical island, with high mountains and a blue sky. Here you can see many trees. Next to them you observe two pirates, ready to step off their boats and onto the island (there is also an additional boat for the user). On the island, there are three giant parrots guarding a treasure chest .



⦁ **Functionalities**

The user can visualize the whole scene: he/she can rotate the camera using the mouse, can zoom in and move using the keyboard and interact with different objects by pressing some specific keys on the keyboard.

He can also play with the fog and change between solid, wireframe objects, polygonal and smooth surfaces.

⦁Implementation details

⦁ Functions and special algorithms

⦁ Possible solutions

OpenGL contains a lot of functions. The following are some of the most frequently used:

⦁ glutInit: to initialize GLUT

⦁ glutCreateWindow: for creating a top-level window

⦁ glBindTexture (…): lets you create or use a name texture

⦁ glTextImage2D (...): for creating depth texture

⦁ …

Now I will present some important functions that I used and the purpose of them.

drawObjects () – is the function where all the objects that are not part of the initial scene are created, also with their shadows. Their normal matrices are computed and the rotation, translation and scaling are done.

initShaders () – is the function where the shaders are initialized

initModels () – is the function where all the 3D models are instantiated

processMovement () – is the function that shows us what action is happening when we press a specific key. For example, we use pressedKey[GLFW\_KEY\_A] when we want to press the A key.

renderScene () – is the function where the scene is rendering, along with the light cube that is the only source of light in this project.

mouseCallBack () – is implemented by using Euler Angles, a mathematics library tailored for OpenGL

⦁ **The motivation of the chosen approach**

We have learnt many of these functions and methods in the laboratories. For example, how to compute light, fog, how to perform different operations (translation, scaling, rotation), how to import and export objects, tutorials for using blender. So, most of the work was based on

information that we knew. This made the work easier and more fun.

⦁ **Graphics model**

All the objects from the scene were downloaded from the internet. Some came with their textures, some came without, so I have to look for a corresponding texture for them. I imported them into Blender to edit them.

The background of the scene was made as a SkyDome, representing the sky, also in blender, using half of a sphere. I edited the mountains also, by deforming the ground.

⦁ **Data structures**

The structures used in this project were simple ones, starting from scalars, vectors, matrices etc. They were used for the light, fog, shadows. They have already been implemented in the project provided by the laboratory, so it was easier to understand and use them.

⦁ **Class hierarchy**

The classes included in the project are: Shader, Model3D, Mesh, Camera and main.

Shader – contains methods for instantiating the shaders

Model3D – contains methods for printing meshes using a specified shader program

Mesh – this class represents a 3D object and includes the vertex (position, normal vector and texture coordinates) and the texture/material

Camera – contains the implementation of the camera movement

Main – contains the main algorithms and methods

The project doesn’t contain a SkyBox class, because we don’t have one. The sky is part of the main scene and it is exported in the scene object.

⦁ Graphical user interface presentation/user manual

The user can interact with the friendly scene in the following ways, by pressing the keys on the keyboard:

9 – start of the preview of the scene

0 – stop of the preview of the scene

1 – show wireframe view

2 – show solid and smooth view(the main view of the scene)

3 – show point view

W – move camera to the front

S – move camera to the back

A – move camera to the left

D – move camera to the right

Up – move the boat to the front

Down – move the boat to the back

Left – move the boat to the left

Right – move the boat to the right

F – start the fog

G – stop the fog

L – move directional light to the right

J – move directional light to the left

The mouse can also be user to rotate the camera.

⦁ **Conclusions and further developments**

In conclusion, I found it quite interesting to work with the blender, because it was the first time when I learnt about it. I had some difficulties withcreating an OpenGl Template to work in in Visual Studio, also working in Blender was a little challenging, and loading the textures and .mtl files, but once I figured it out everything became so easy and enjoyable.

It was an advantage that the code was mainly from the laboratory work, so I had to understand it better and play with the structure and my objects from the scene. I managed to add some animations to some objects, which made the project more fun.

In the future I would like to have a more complex scene and maybe make some objects by myself, in Blender, without importing them from different websites. I also can add rain/snow, night and add other animations to the objects.

⦁ References

⦁ Courses and Laboratories (Graphic Processing, Fall 2022)

⦁ <https://free3d.com/3d-model>

https://www.youtube.com/watch?v=45MIykWJ-C4&t=4935s&ab\_channel=freeCodeCamp.org

⦁ <https://learnopengl.com>

https://www.youtube.com/playlist?list=PLrgcDEgRZ\_kndoWmRkAK4Y7ToJdOf-OSM