Technical University of Cluj-Napoca
Faculty of Automation and Computer Science
Computer Science Department

Old Workhouse Scene

- Graphical Processing Systems Project –

Reckerth Daniel-Peter

Group 30434

Contents

1. Specification

The project's specification is based on constructing a photorealistic representation of a 3D graphic scene using the OpenGL public library and C++ programming language. The theme of this is an abandoned workhouse scene in which different 3D objects are presented including buildings, truck, buses and environment (trees, fence, etc). Different features have been used in order to increase the scene's realism.

OpenGL is a specification developed by Khronos group, although it is considered an API. It describes the desired result or output of each function used in graphics. Alongside, we have included also the GLFW library in order to help us create windows, contexts and surfaces, receiving inputs and events and GLEW, an OpenGL extension which provides efficient run-time mechanism in determining which OpenGL extensions are supported on the target platform.

2. Scenario

2.1 Scene and objects description

The scene contains an old, degraded warehouse which includes different 3D objects:

- an old, deteriorated warehouse
- old-bricked office building
- two rusted vehicles: truck and disassembled car
- an old bus
- environment objects: different kinds of trees, fence, gate
- skydome

The scene is located in an asphalt surfaced yard and the perimeter on a place surrounded by hills and a lake. A figure of the scene is presented below:

Fig. 1: 3D Scene

2.2 Functionalities

Below we list the functionalities of our project:

- scene's visualization and camera movement using keyboard keys and mouse:
    - W, A, S, D for forward, backward, left and right
    - mouse for rotation around camera and scroll for zoom in and out
- changing between light sources: directional, point light using L
- changing between solid and wireframe view of objects using F
- viewing the fog effect using C

Figures exemplifying the functionalities above are shown below.

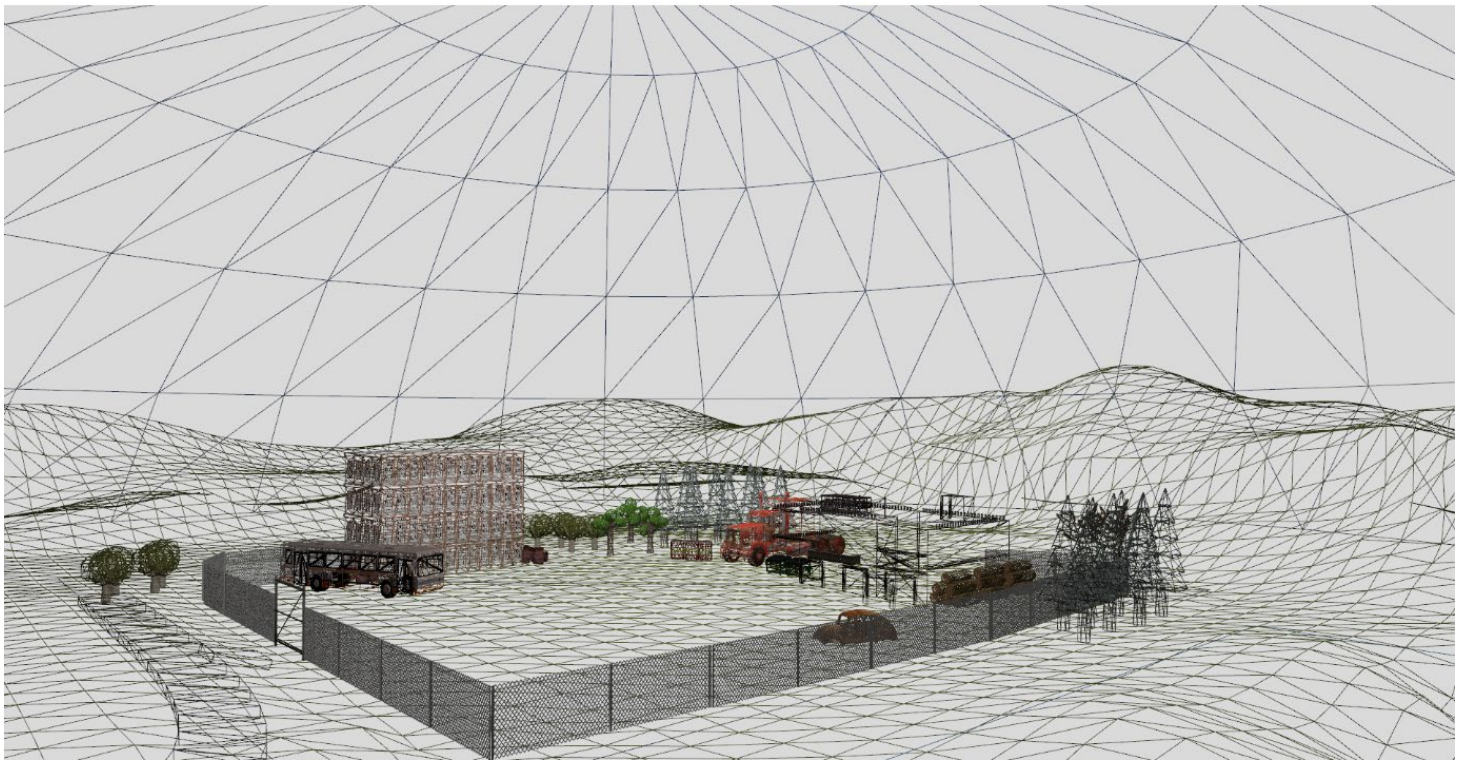

Fig. 2: Wireframe view

Fig. 3: Fog effect

Fig. 4: Directional Light

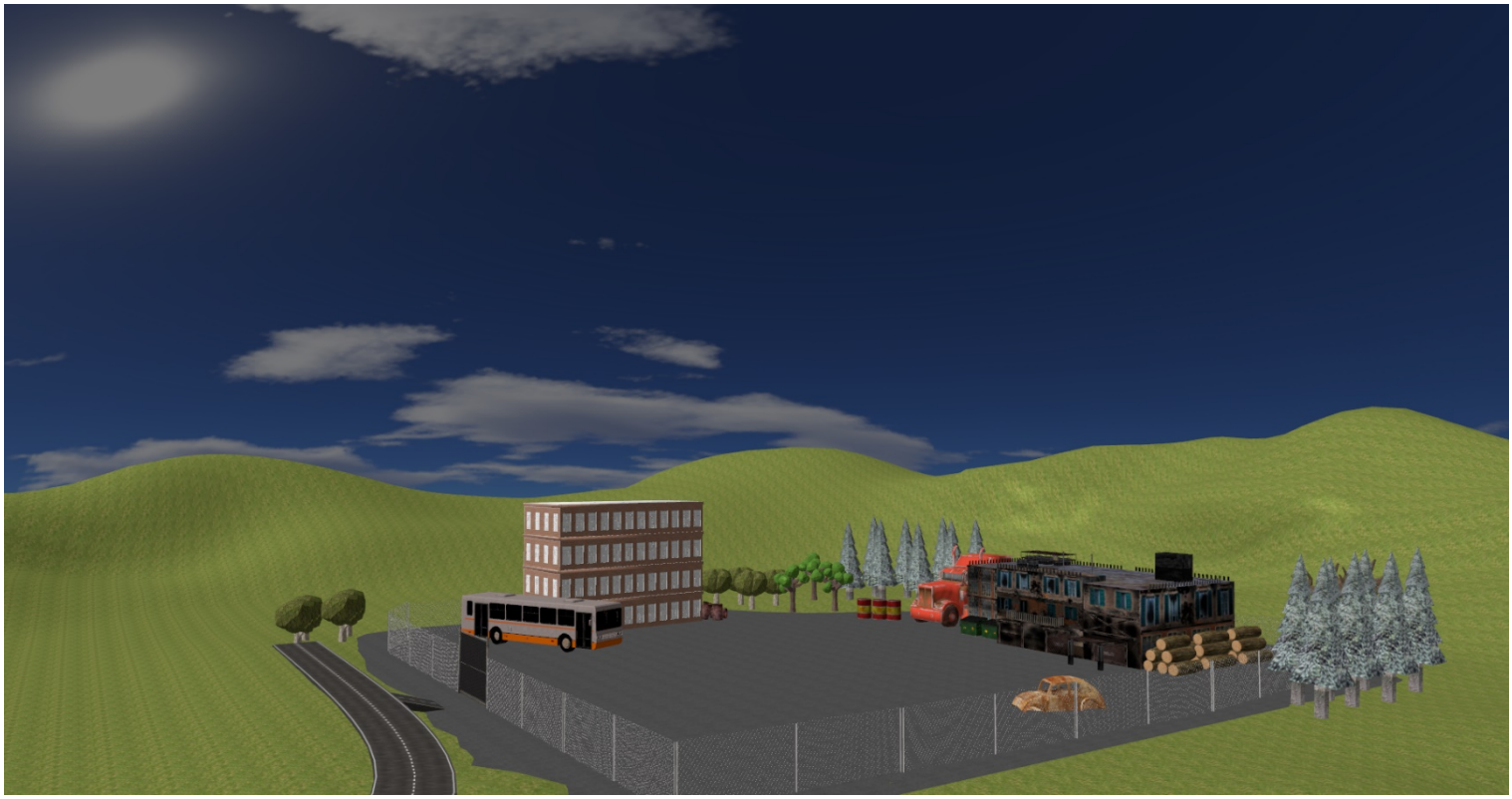

Fig. 5: Point Light

3. Implementation Details

The project's scene was modeled previously using Blender. We have defined the planes (soil, water and warehouse area) by texturing and sculpting. We have imported different objects which were downloaded from the internet but separately textured. A sky-dome was also added to showcase the idea of the horizon.

3.1 Functions and special algorithms

In improving the details and the complexity of the scene we have used functions and special algorithms for camera's movement and effects.

We have studied these functions in the laboratory works and we have implemented them. We have algorithms developed for camera's movement, also the fog effect and the two light sources which are written in the shaders.

3.2 Graphics model

The objects are presented as polygonal models. They are actually polygonal meshes, which are geometric and topological description of bounded or surfaced objects. The main advantage is that it makes the working simple.

3.3Class Hierarchy

The project is comprised of the following classes:

- `Camera.hpp`
- `Mesh.hpp`
- `Model3D.hpp`
- `Shader.hpp`
- `Window.h`

In the camera class we have defined attributes like camera's position, target, front direction, up and right direction. We have functionalities of moving the camera, rotating, and returning the viewing matrix.

The mesh class extrapolates the idea of the polygonal object. We have attributes like vertices, indices, textures and buffers.

In the 3D model class we have meshes and loaded textures. The main functionalities of this class are that we can load the model, drawing it on the screen, reading the object's path, etc.

The shader class helps us manage the written shaders more easily. We have implemented functions the help us load our shaders program, use them, read the shader's provided file, etc.

The Window class help us to work with the window more easily. We have defined the dimensions of our window and provided functions of creating, deleting, and returning the dimensions of our window.

4 User Manual

In order to test the forementioned functionalities of the project, one can run the executable file and test the functionalities. We also mention the effect of the key-binds:

- W, A, S, D – forward, backward, right, left
- Q, E – rotate right, left
- mouse movement: change camera's direction or rotation
- mouse scroll: zoom in/out
- L – directional/point light
- C – activate fog effect
- F – wireframe view
- M – shadow (not working)
- O – open gate (not properly working)

5 Conclusions and further development

The project can be further developed by working more carefully and professionally the scene in Blender and by also adding different functionalities: different lights sources, effects like rain, snow, wind.

Animations could have been added in order to enhance the photo-realism of the project.

6 Bibliography

For developing this project different materials were taken in account.

- lectures, laboratory materials from the UTCN's Graphical Processing Course
- https://learnopengl.com/
- for objects:
    - https://free3d.com/
    - https://www.turbosquid.com/