

Project Documentation

OpenGL Enhanced City Scene



Student: Urdea Mara Cristina

Group: 30432

Coordinator: Enoiu Diana Cristina

Contents

1. Subject Specification	2
2. Scenario	2
2.1 Scene and Objects Description	2
2.2 Functionalities	2
3. Implementation Details.....	5
3.1 Functions and Special Algorithms	5
3.1.1 Possible Solutions	5
3.1.2 The Motivation of the Chosen Approach	5
3.2 Graphics Model.....	5
3.3 Data Structures	6
3.4 Class Hierarchy	6
4. Graphical User Interface / User Manual	6
5. Conclusions and Further Developments	6
6. References.....	7

1. Subject Specification

This project demonstrates the creation of an interactive 3D city scene in OpenGL, showcasing integration of advanced graphical features and detailed Blender models within a real-time, explorable environment. By incorporating weather effects, lighting, shadowing, and object animations, the project aims to achieve a high level of visual fidelity and engagement.

2. Scenario

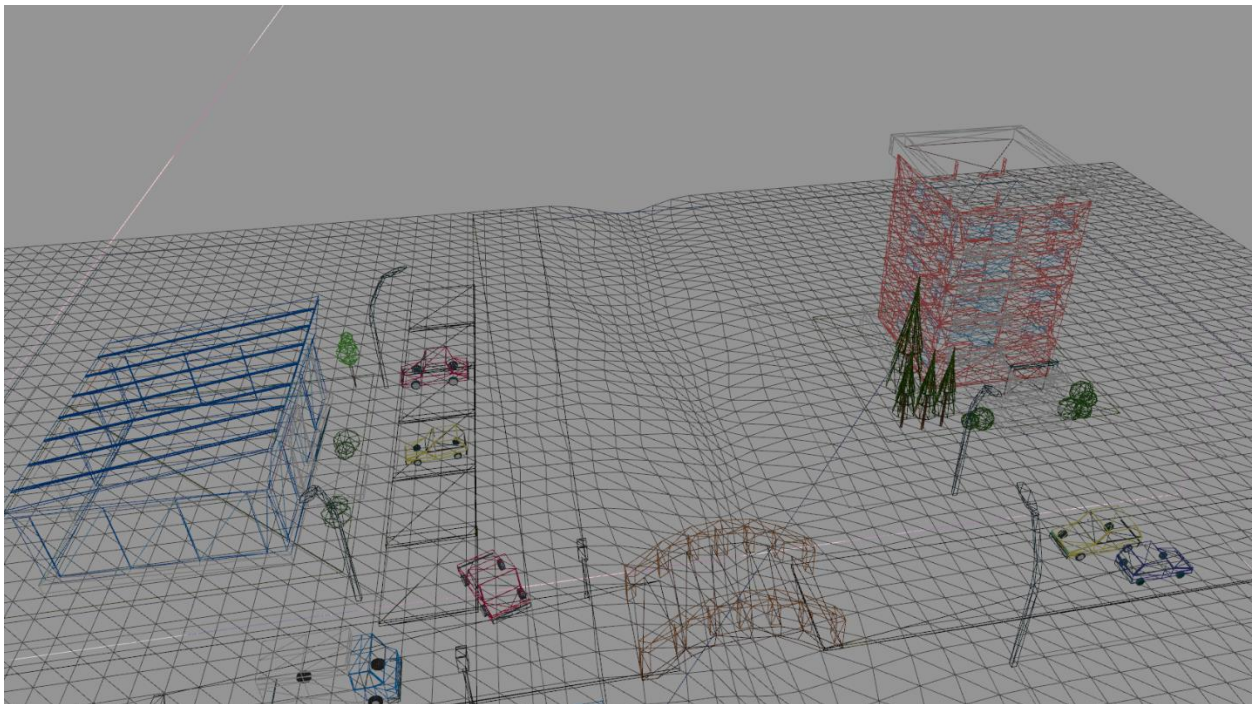
2.1 Scene and Objects Description

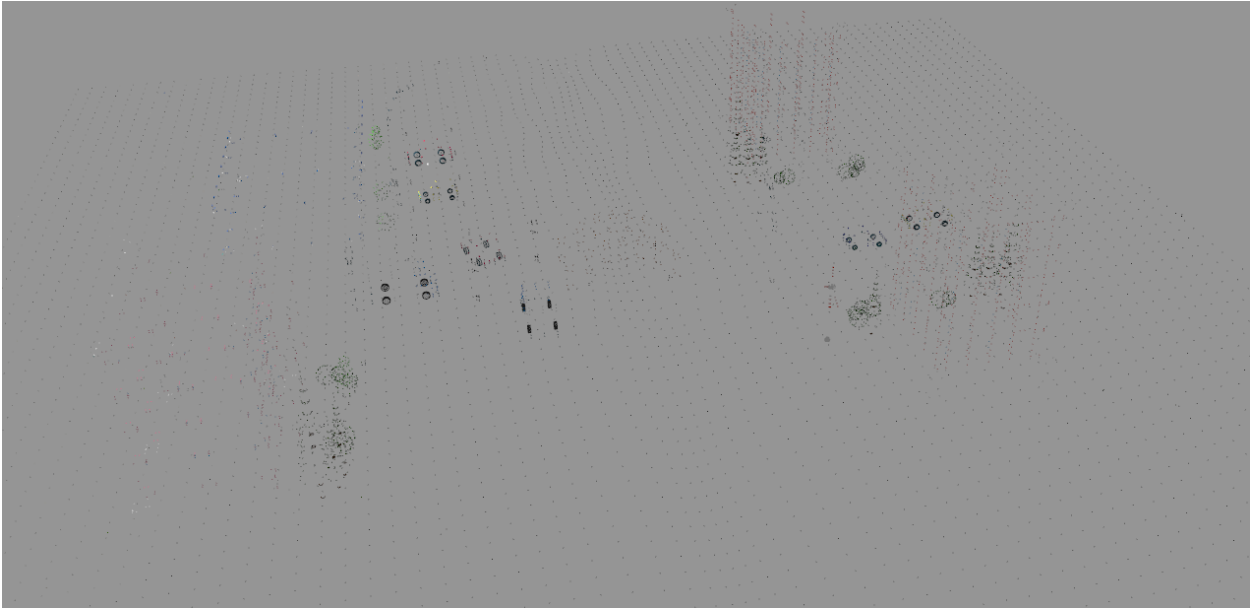
The urban landscape portrays a vibrant metropolis, adorned with diverse architectural marvels, automobiles, and city amenities. Constructed within Blender, every element is meticulously designed to uphold authenticity, ranging from towering skyscrapers to minute details like trees, decorative plants, wind turbines, and lampposts. Noteworthy emphasis is placed on texture application and material attributes to enrich the scene's harmony.

2.2 Functionalities

- ❖ Lighting.
- ❖ Dynamic Weather Effects: fog.
- ❖ Object Animations: Features moving elements like a wind turbine, simulates a circular motion; cars, a smooth translation animation to simulate a real-like motion.
- ❖ Camera animation: User-controlled navigation
- ❖ Different rendering modes: solid, wireframe, polygonal and smooth surfaces.
- ❖ Night mode: Simulates the night over the city.







3. Implementation Details

3.1 Functions and Special Algorithms

3.1.1 Possible Solutions

The OpenGL library encompasses a wide array of functions, including but not limited to:

- ❖ ***glViewport(...)***: Utilized for configuring the Viewport transform.
- ❖ ***glfwCreateWindow(...)***: Employed for window creation.
- ❖ ***glGenTextures(...)*, *glBindTexture(...)*, *glTexImage2D(...)***: Functions for texture generation and manipulation, including depth texture creation for Framebuffer objects.
- ❖ Various others.

Within this project, the pivotal function ***void renderScene()*** orchestrates the transmission of data to shaders, encompassing model and shadow creation, normal matrix computation, as well as handling rotation, translation, and scaling operations.

Additionally, ***void initUniforms()*** plays a crucial role in configuring light properties, encompassing point, directional, and spotlight specifications. ***void initShaders()*** initializes shader programs, while ***void initModels()*** instantiates all 3D models required for the scene.

Other essential functions include:

- ❖ ***void processMovement()***
- ❖ ***void move(gps::MOVE_DIRECTION direction, float speed)***
- ❖ ***void mouseCallback(GLFWwindow* window, double xpos, double ypos)***
- ❖ ***void keyboardCallback(GLFWwindow* window, int key, int scancode, int action, int mode)***

These functions facilitate the execution of user-desired model or camera movements.

The ***mouseCallback()*** function integrates Euler Angles, while mathematical operations and algorithms are facilitated by the GLM (OpenGL Mathematics) library.

3.1.2 The Motivation of the Chosen Approach

In the dedicated laboratory sessions for this course, we acquired knowledge on computing lighting, shadows, and executing translation, scaling, and rotation operations on various objects. With this foundational understanding and a project framework in place, my tasks were clearly outlined, providing a structured path for project completion.

3.2 Graphics Model

The process of sourcing objects and textures from the internet presented challenges, particularly in cases where textures failed to display correctly on objects or where objects themselves did not appear as expected. I spent considerable time searching for suitable objects and textures. For some of them I have performed texturing by myself. Despite these challenges, I found this aspect of the project to be creatively fulfilling and enjoyable.

3.3 Data Structures

The data structures required were pretty simple, already presented and implemented in the laboratory works, so my work was considerably simplified. I also needed some additional functions to calculate the necessities for night mode, fog, lights and so on.

3.4 Class Hierarchy

The project has an Object-Oriented approach, with classes for the different functionalities like camera control, animations, lighting, rendering, skybox, Model3D, Shader etc.

4. Graphical User Interface / User Manual

Users can engage with the scene through keyboard input using the following keys:

- ❖ W – move the camera to the front.
- ❖ A – move the camera to the left.
- ❖ S – move the camera backwards.
- ❖ D – move the camera to the right
- ❖ I – show initial solid and smooth view.
- ❖ U – show wireframe view.
- ❖ V – show point view.
- ❖ Q – move directional light to the left.
- ❖ E – move directional light to the right.
- ❖ F – start the fog.
- ❖ G – stop the fog.
- ❖ N – start the night mode.
- ❖ B – stop the night mode.
- ❖ T – rotate the eolian turbine.

5. Conclusions and Further Developments

In conclusion, I would like to say that I found it pretty difficult to work with Blender, because it was the first time working with it, my laptop encountered a few problems working with this software, like unexpectedly freezing. Finally, I have figured out that working with Blender can be very rewarding and resourceful.

Judging the coding part, I found it very complicated. I spent a lot of time trying to understand the code structures, then working with them. Also, manual debugging was hard too, but necessary.

This project required a significant amount of work and time, but finishing it was really rewarding.

As further developments, they can be:

- ❖ Proper shadowing.
- ❖ Wind, rain, snow.
- ❖ Addition of more elements to the scene.
- ❖ Addition of more animations.

6. References

- ❖ <http://www.opengl-tutorial.org/>
- ❖ <https://learnopengl.com>
- ❖ Courses and Laboratory Works
- ❖ <https://www.cgtrader.com>
- ❖ <https://turbosquid.com>
- ❖ <http://tf3dm.com/>