MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE

UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Graphic Pocessing Project

Name:Stroie Anca Gabriela

Gr:30432

Cuprins

Introduction

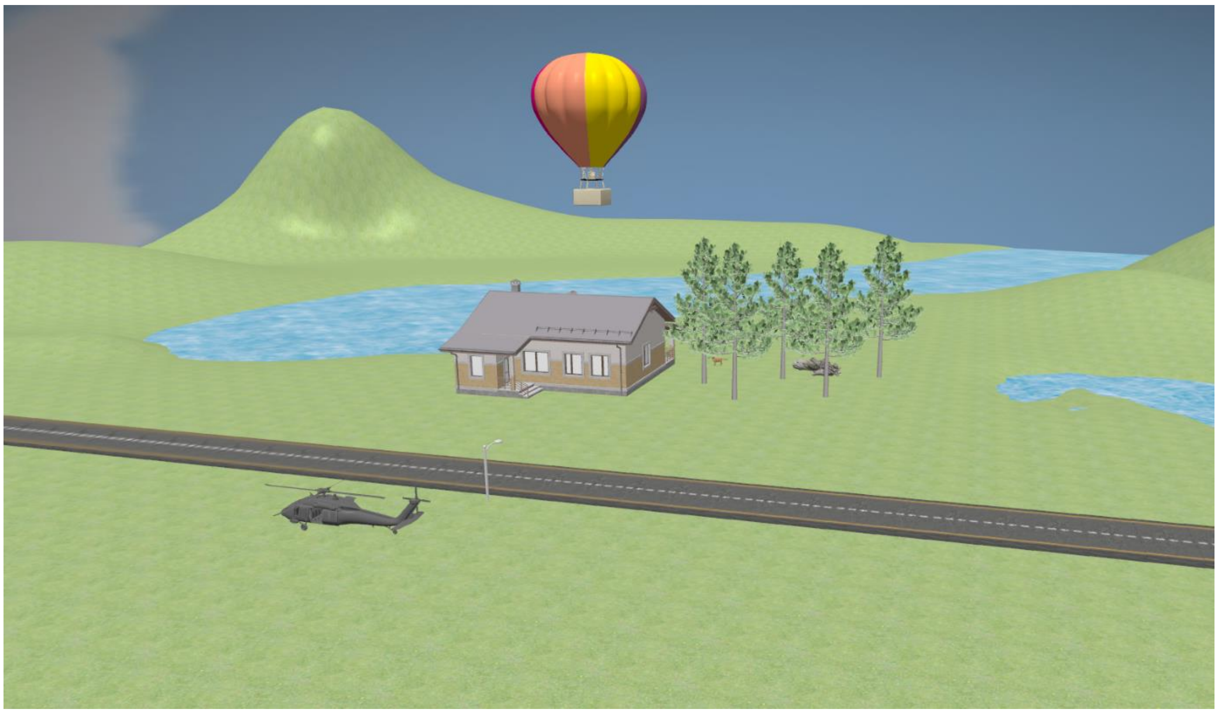
This project is about creating an interactive 3D scene using OpenGL technology. It allows users to explore a detailed 3D environment, much like in a video game, using their keyboard and mouse. The scene includes various objects such as houses, animals, vehicles, and uses special effects to enhance the experience. It's not just an exploration of a 3D world, it also demonstrates the use of 3D libraries like OpenGL, GLFW, and GLM.

Technologies Used:

- OpenGL for 3D graphics rendering.
- C++ for programming logic.
- Blinn-Phong lighting model for realistic visual effects.
- Shader programming for detailed rendering control.

Scenario

The scene features a mountainous terrain created in Blender, with realistic grass textures. A central house is the main focus, surrounded by other structures, a water body, roads, animals, and a helicopter. Interactive elements like a hot air balloon and the helicopter add life to the scene. Users can change light settings and fog effects.



Functionalities

The scene uses both directional and point lighting. Users can move around using keyboard and mouse controls. The camera's speed is adjustable. Users can control the rendering of the scene, activate or deactivate lights and fog. Animations of various elements, like the hot air balloon and the helicopter, enhance the scene's realism. Users can interact with these elements through keyboard commands.

Implementation Details

Functions and Algorithms

Key functions include `renderScene()` for drawing objects, `initModels()` for loading models, and `initUniforms()` for managing shader variables. The camera is controlled by `mouseCallback()` and `processMovement()`.

Graphic Model

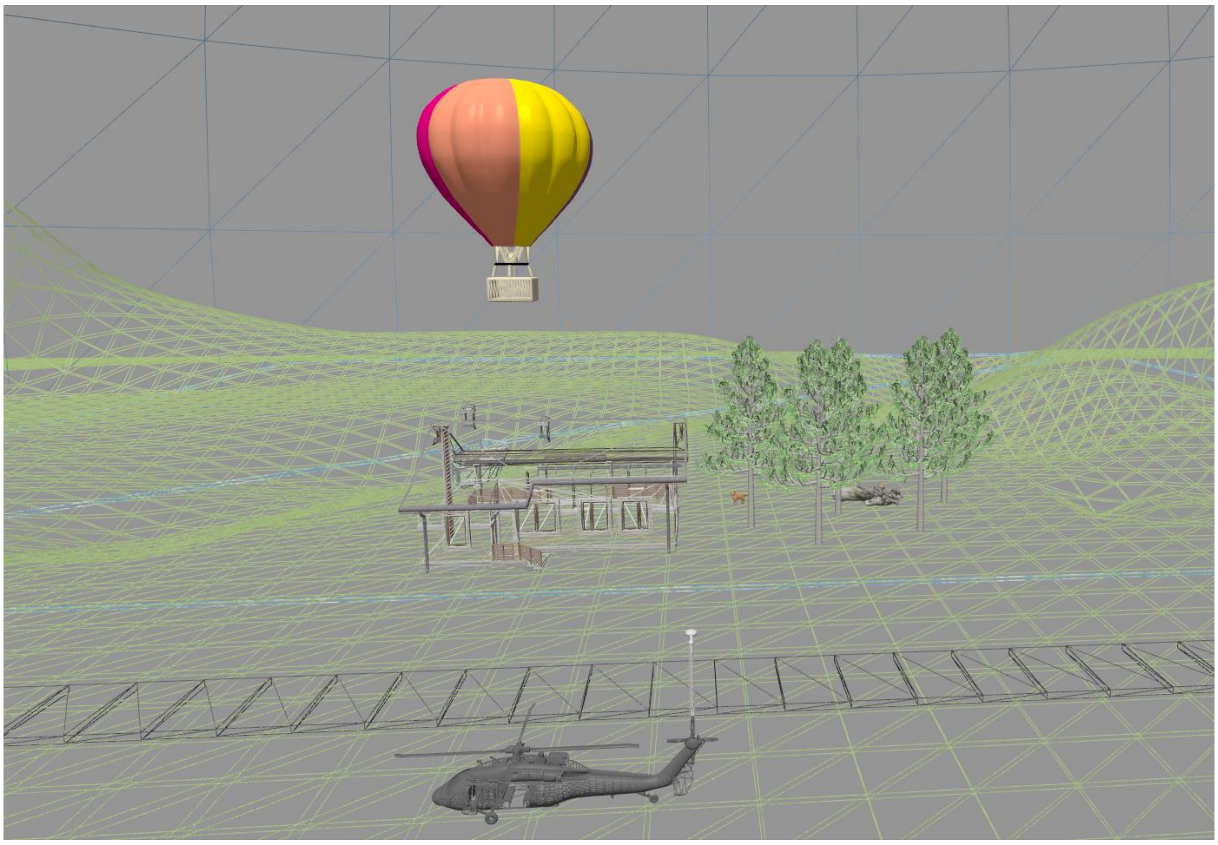
The terrain is custom-made in Blender. Each object was carefully texture-mapped for realism. Roads and other structures were also manually created in Blender. Several individual objects were imported for in-program manipulation.

Data Structures

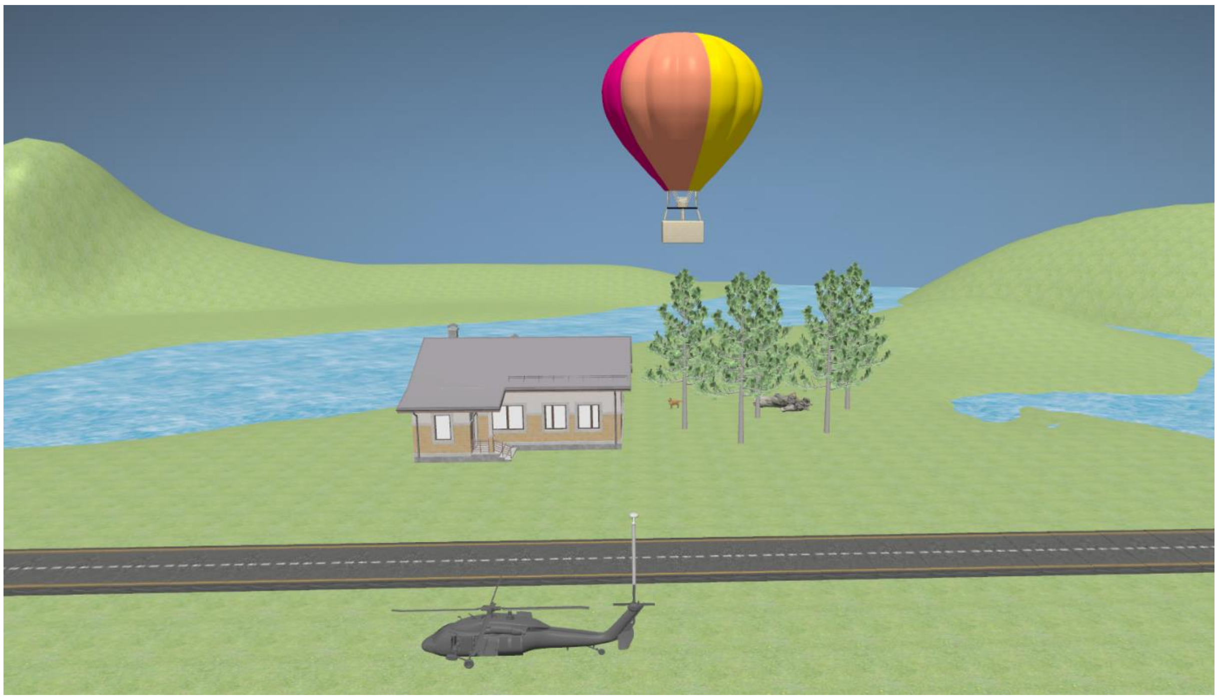
Vectors are used for camera positioning, GLints for shader variables, and standard C++ types for other functionalities. The Camera class has been modified for better functionality.

Graphical Interface Presentation

Visualisation Modes



Wireframe visualisation



Smooth visualisation

Point view

Visualisation of point light

With fog

User manual

The user can interact with the scene using the following keys on the keyboard:

- W move camera to the front
- S move camera to the back
- A move camera to the right
- D move camera to left
- Q start fog
- Z stop fog
- O strat pointlight
- P stop pointlight
- 1 move the rotor of the helicopter
- E wireframe view
- F smooth view
- V point view

Conclusions and further developments

Some further developments:

- Multiple light sources
- Wind

- Rain
- Addition of more elements in the scene

References

- https://learnopengl.com/
- https://free3d.com/
- https://www.turbosquid.com/
- https://www.youtube.com/playlist?list=PLrgcDEgRZ_kndoWmRkAK4Y7ToJdOf-OSM