MINISTRY OF NATIONAL EDUCATION

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA
FACULTY OF AUTOMATION AND COMPUTER SCIENCE
COMPUTER SCIENCE DEPARTMENT

Graphics Processing Documentation

STUDENT NAME: Trasculescu Tudor-Ioan
GROUP: 30434
Coordinating teacher: Nandra Cosmin

Contents

1. Subject

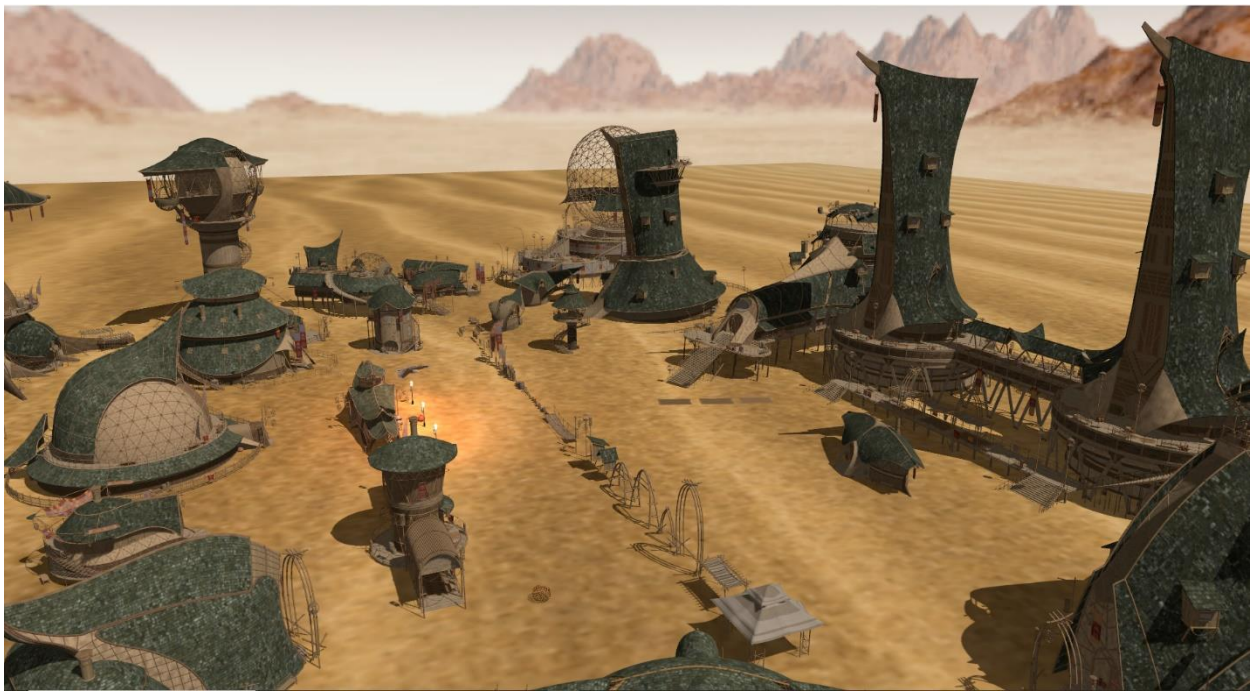
The subject of the project consists in the photorealistic presentation of 3D models using the OpenGL library. The user will be able to directly manipulate the scene of objects through mouse and keyboard inputs. The project will include the following functionalities:

- Visualization if the scene: scaling, translation, rotation and camera movement using keyboard and mouse.
- Animation of object elements.
- Viewing of solid, wireframe objects, polygonal and smooth surfaces.
- Texture mapping and materials including textures quality and level of detail and textures mapping on objects.
- Exemplification of shadow computation.
- Exemplification of animation of object components.
- Some effects such as fog, wind.
- Collision detection regarding some houses using bounding boxes.

2. Scenario

2.1 Scene and object description

The scene will consists of several 3D objects, which will be photorealistically rendered using OpenGL library. The scene is very complex. It has 3GB of data inside the models folder. It shows an abandoned village from the desert where only an eagle was left to watch above his owner’s house. A view of the scene can be seen on the next page.



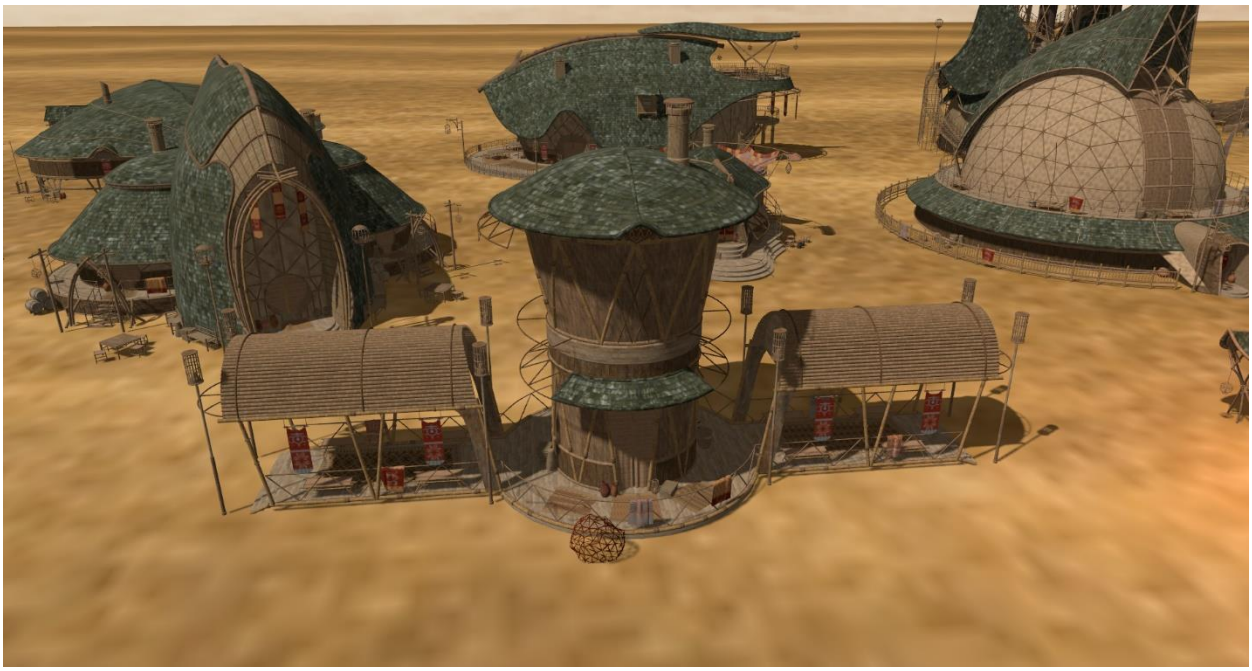
2.2 Functionalities

The user will be able to interact with the scene through keyboard and mouse inputs. Based on the input from the user the wind will start and stop and some tumbleweeds are going to be generated in order to simulate this better. The project consists of multiple light sources (directional, point light and spot light). The spot light will be seen from the user's perspective and only on night mode. The animation of the object components can be observed on the eagle. While it is rotating around the house, it's wings are also moving around it's own axis. At daytime the shadows can be observed. Some other effects can be generated from the user's input, such as fog or the viewing of the scene: wireframe, point and smooth surfaces.

3. Implementation details

3.1. Functions and special algorithms

In order to detect collisions there are implemented some movements on a tumbleweed, such as translation (moving around the scene), but also rotation (around it's own axes) depending on the user's input: FORWARD, BACKWARD, LEFT, RIGHT. This algorithm consists in bounding some objects and checking whether the coordinates of the tumbleweed intersect the boundary of the objects. If it does not intersect, the tumbleweed can move freely, but otherwise it will stop moving, but it will never stop rotating. This is done to enhance the realism of the scene.



As shown in the figure above, the tumbleweed will not go through the house. On this picture, one can observe the level of detail of the shadows. The shadows are drawn from the directional light's point of view using the depth map of each fragment of the scene. The algorithm implemented here samples the shadows 49 times and the resulting color of the fragment is given by the average of the iterations. This gives a more realistic shadowing of the scene.

The preview of the scene will take the user through the most important parts and will show the best features of the objects and texture of the objects. The preview is just moving the camera following a Bezier curve between 3 points chosen by me.

3.2. Graphics model

http://web.archive.org/web/20230110171944/https://cgpersia.com/ this is the site from where all the objects and textures were downloaded. After the objects were downloaded, all the modelling of the scene was made in Blender. This took a while due to the size of the scene and complexity of the objects.

3.3 Data structures

The only data structures required were some simple ones, to hold together the attributes of the different types of light. In order to compute the animation of the eagle some vectors and matrices were used. Some special data structures are used for skybox 3D models and also for shadows in 2D texture. All the data structures

used for the project will include arrays, linked lists and matrices, which will be used to store and manipulate the information related to the scene and objects, such as their positions, rotations and textures.

3.4. Class hierarchy

The project is organized using object-oriented programming, with a class hierarchy that include classes for the scene, objects, lights, textures and animations. These classes have various attributes and methods, such as those for rendering, manipulation and animation of the objects and scene:

- Camera: contains the implementation for camera movement(up, down, front, back, left, right)
- SkyBox: contains skybox implementation (load, draw, load textures, initialize)
- Mesh: represents a 3D object. Includes the following data structures:
  Vertex (position, normal vector and texture coordinates
  Texture (id, type and path)
- Shader: contains methods for creating and activating shader programs

4. Graphical user interface presentation / user manual

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

- W – front movement of camera
- S – backward movement
- D – move to right
- A – move to left
- Z – normal visualization of the scene
- X – edge visualization
- C – point visualization
- N – activating night mode

- M – activating day mode
- F – start fog
- G – stop the fog
- H – enhance the fog
- O – generate the tumbleweeds and simulate wind
- P – stop the wind and make the tumbleweeds disappear
- U – preview of the scene using Bezier curves
- I – stop the preview

Some picture of the scene, showing the contents of the manual can be observed below:

Night mode:

Enhanced fog:

5. Conclusions and further developments

The project demonstrates the use of the OpenGL library for the photorealistic presentation of 3D objects, as well as the implementation of advanced features such as animation and shadow computation. In future developments, additional features such as rain, shadow computation for the local lights, not only for the directional light, more complex animation of the objects, such as a man that can be able to simulate a realistic walk or even run. Some enhancement of the project would be to be able to have collision detection on all objects of the scene.

6. References

- https://learnopengl.com
- http://web.archive.org/web/20230110171944/https://cgpersia.com/
- Graphic processing laboratories