*Technical University of Cluj-Napoca*

*Faculty of Automation and Computer Science*

**GP project**

Indre Bogdan

Group: 30432

1. **Content**

1. Content

2. Subject specification

3. Scenario

3.1 Scene and object description

3.2 Functionalities

4. Implementation details

4.1 Functions and special algorithms

4.2 Graphics model

4.3 Data structures

4.4 Class hierarchy

5. Graphical user interface presentation

6. Conclusions and further developments

7. References

1. **Subject specification**

The subject of the project consists in the photorealistic presentation of 3D objects using OpenGL library. The user directly manipulates by mouse and keyboard inputs the scene of objects.

* (2p) visualization of the scene: scaling, translation, rotation, camera movement

1. using keyboard and mouse
2. using animation

* (1p) specification of light sources (minimum two different lights)
* (0.5p) viewing solid, wireframe objects, polygonal and smooth surfaces
* (1p) texture mapping and materials

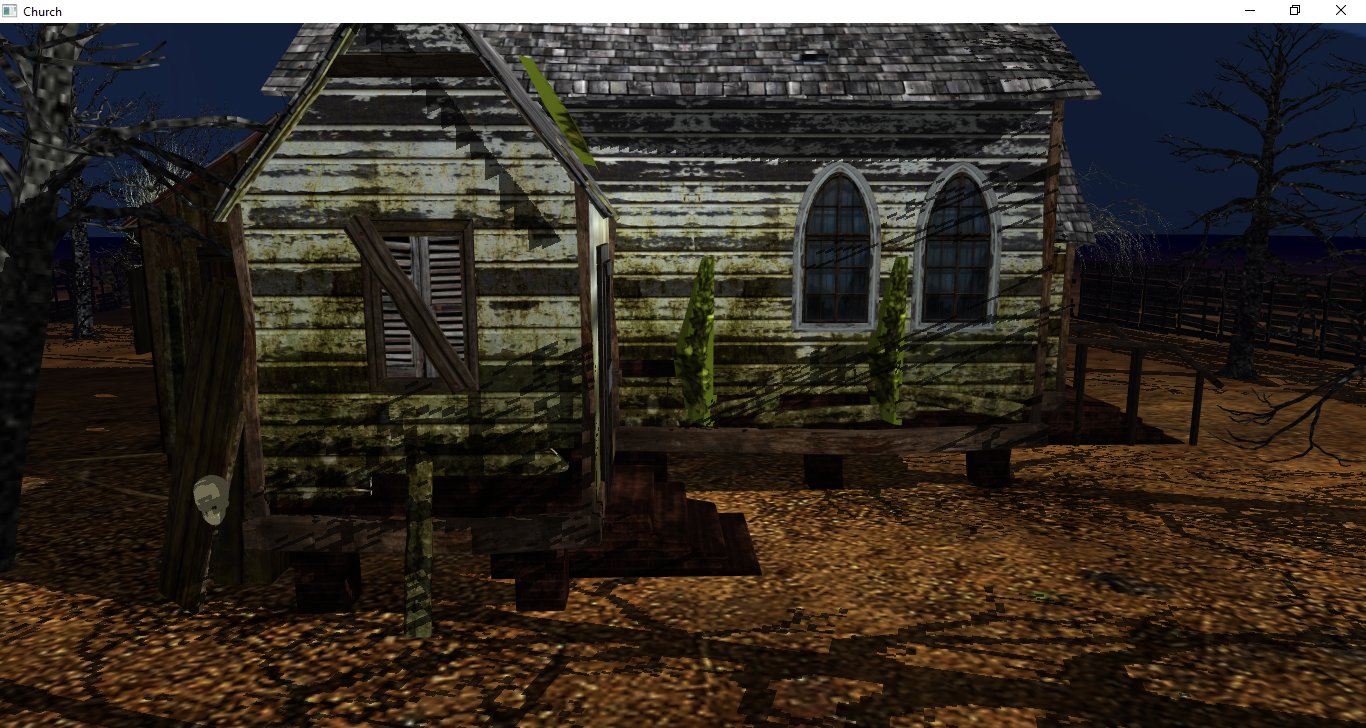
1. textures quality and level of detail
2. textures mapping on objects

* (1p) exemplify shadow computation
* (0.5p) exemplify animation of object components
* (3p) photo-realism, scene complexity, detailed modeling, algorithms development and implementation (objects generation, collision detection, shadow generation, fog, rain, wind), animation quality, different types of light sources (global, local, spotlights)
* (1p) documentation (mandatory)

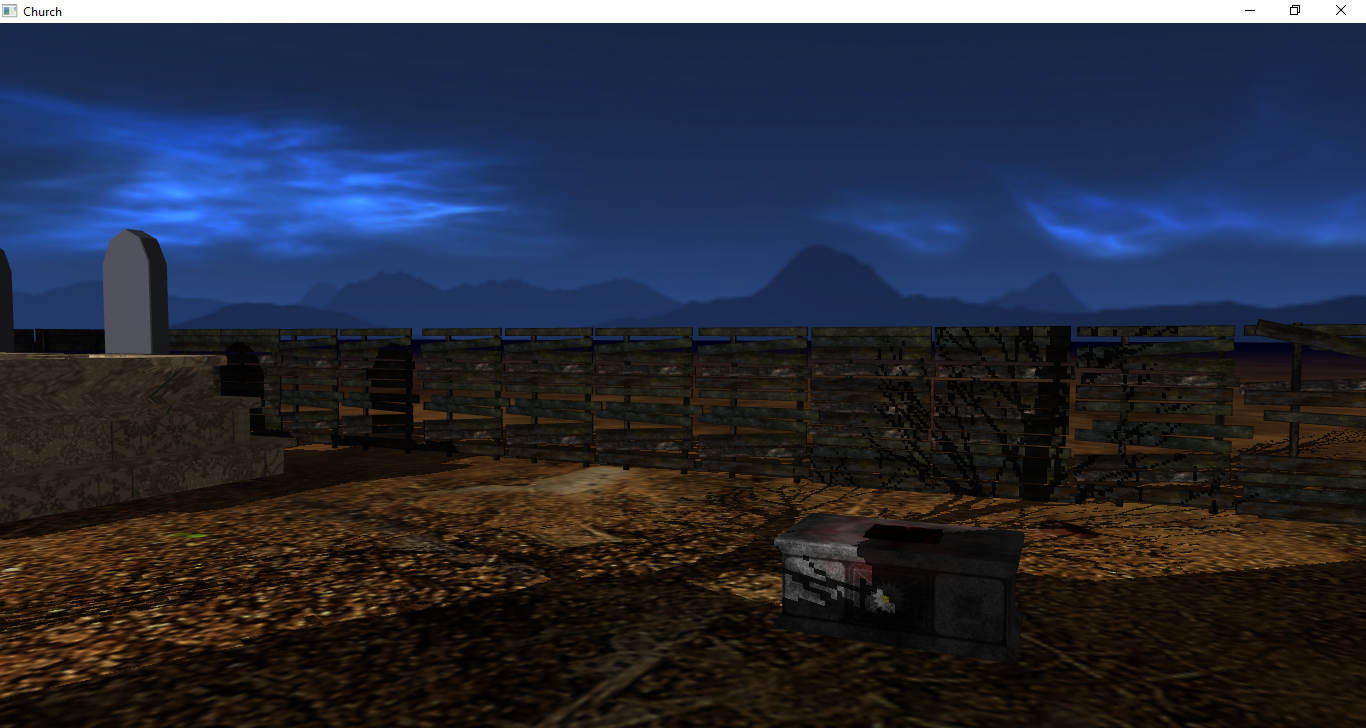
1. **Scenario**

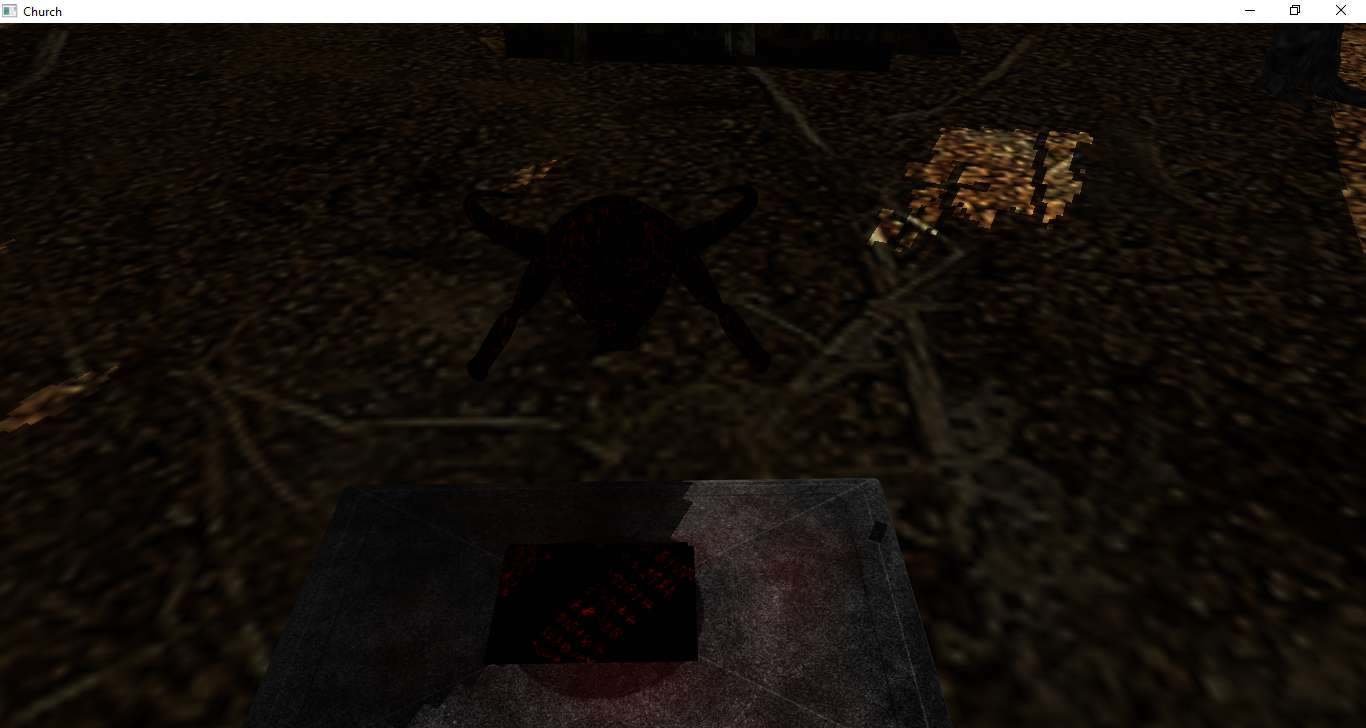
**3.1 Scene and object description**

For this project I decided to model a haunted village. Unfortunately, free good objects and textures are a rarity. That being said, I had to settle for the church of that village (which is very far away from other human dwellings). 

Here we can see the old church with its years of neglect and mismanagement. Around the church there is a wooden fence that marks its yard. Inside the yard we have a small shed, some dead trees and a small graveyard. Did I forgot to mention the flying skull?

It is clearly having a great time. In the back of the church we find more tombstones and a strange altar with an even stranger book on it.





This is Buba. Buba is a friendly demon you can sometimes find around. Right now he is enjoying what he calls a “novel”. He was the last priest to have served in this temple. Too bad boredom got the best of him and he started worshiping…other things. Buba will follow your commands, because of the same thing that got him here: boredom. But watch out, he cannot leave the church yard. He also dislikes light, especially the light of the church.

**3.2 Functionalities**

There are several animations taking place. First, there is the intro. In the intro the user has no control over the camera movement. This will show the yard in its splendor.

What the user has control over is the light in the church tower, the opening of the fence and Buba. After the intro, the user can control the camera. The light can be switched on and off, this makes Buba hide in the dark (he can still be controlled and you can still see his shadow but let him believe it).



The fence can be lifted and placed back. Some graphical features included: directional and point light (ambient, diffuse and specular), fog, shadow computation, camera control, skybox, object collision (Buba and the yard also the camera and the ground), object animation (the skull) and part of an object animation (a part of the fence), intro.

1. **Implementation**

**4.1 Functions and special algorithms**

The scene uses several algorithms to simulate reality. For the lighting the Blinn-Phong model is used and for the shadows the shadow mapping technique presented in Labs 8 and 10 respectively. What I added is the point light with a quadratic function in myCustomShader to alter the ambient, diffuse and specular lights of the surrounding objects.

This is done by sending two new uniforms to the shader (lightPos for the position and lightOn). The computations are done in computePointL().

The camera can be rotated by the mouse using mouseCallback() and in camera.cpp Camera::rotate() and Camera::updateVectors() to find the right direction.

For the intro I simply use a time variable that increments on each frame and the move() and rotate() camera and for the rest of the animations and collisions I used simple translations, scales and rotations to obtain the desired result.

There are only a few objects imported directly in OpenGl because Blender was used to set textures, positions and to create a base object with the static components. The dynamic components were added manually.

I have added a bluish fog to go along with the blue skybox (small changes in shader).

**4.2 Graphical model**

The graphical model used for lighting is Blinn-Phong.

**4.3 Data structures**

Data structures used include basic types and uniforms to pass data to the shaders, in short, nothing special.

**4.4 Class hierarchy**

The classes used are the following: The main (OpenGL\_4\_Application\_VS2015) and several classes that help with the importing of objects (Model3D), camera (Camera), shader use(Shader), Skybox and Mesh for the objects and textures.

1. **Graphical user interface presentation**

The user starts the application and the intro takes place. After the intro he can move the camera with WASD and the mouse. The arrows move Buba, L turns the light on and K “kills: the light. O will lift the fence and C will lower it. 1,2,3 changes the view from fill to line and point.

1. **Conclusion and further developments**

This project has been a great opportunity to understand better the stress that comes after you try and fail to use blender efficiently. I also understand better computer graphics and the way we move a 3D world into a 2D monitor to the best of our capabilities. Many things can be added to this project (walking mode, a hand with a lamp in it in front of the camera, better shadows and lighting, better textures, better objects, better everything), but I do believe that the objective of this project has been reached, that is, to apply some of the things learned in this lab.

1. **References**

1. Laboratories

2. <https://learnopengl.com/Getting-started/Camera>

3. <http://ogldev.atspace.co.uk/www/tutorial20/tutorial20.html>