Graphical Processing Systems
Semestrial Project

Student: Blaga Cristian-Ioan

Group: 30433

Teacher: Adrian Sabau

Contents

1. Subject specification

The main focus of this project is the development of a photorealistic 3D scene in which different graphics principles and algorithms are deployed. There will be different animations deployed throughout the scene, both autonomous and user controlled. The scene will also be explorable by the user with the use of the keyboard and the mouse.

2. Scenario

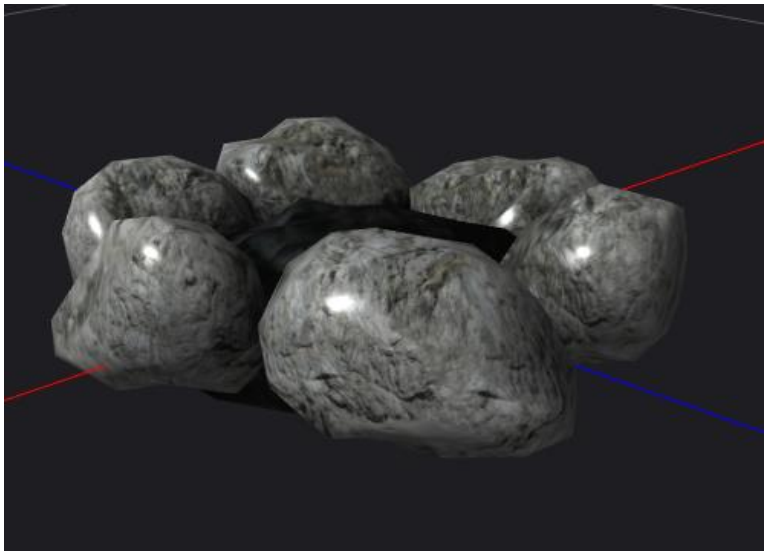
a. Scene and objects description

The scene is built by the use of a lot of 3D objects which will be presented below.

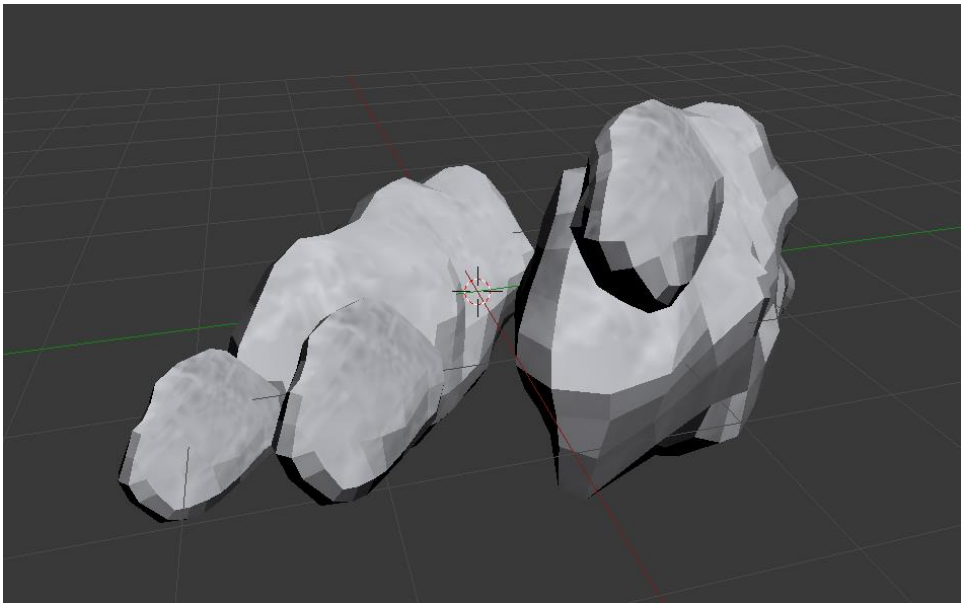
- *Trees*

In the scene, there are two types of trees which are present.

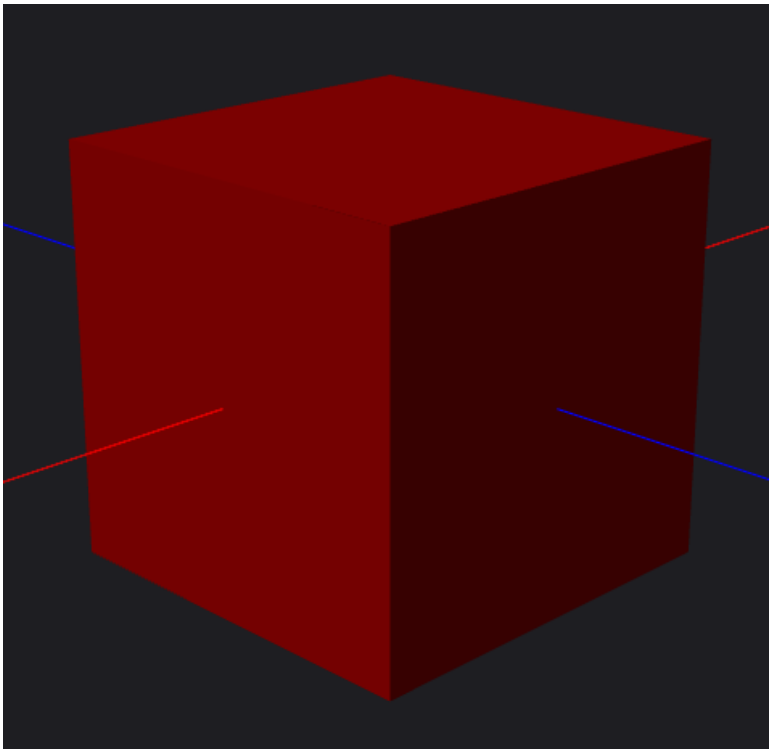
- *Campfire*



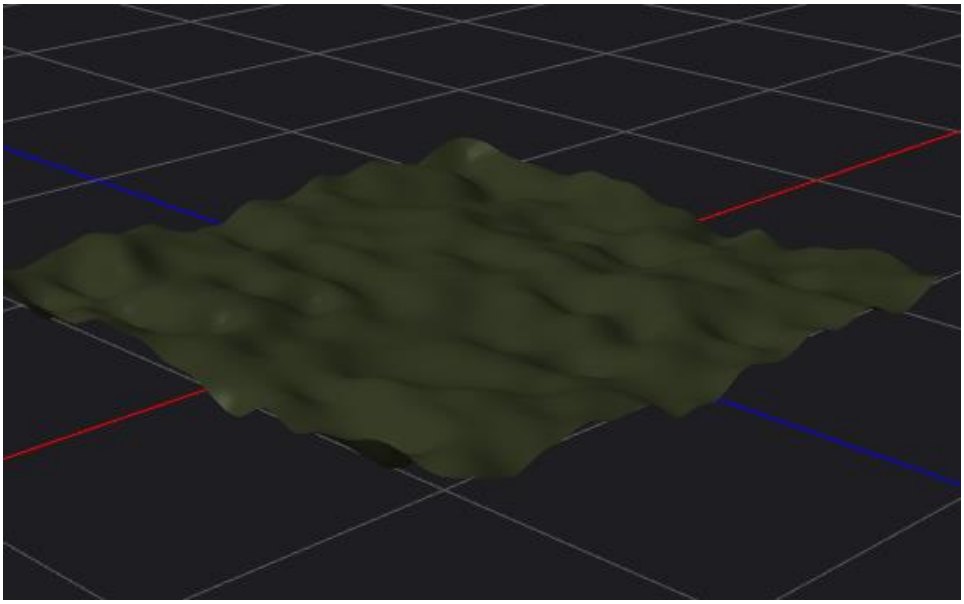
- *Cloud*



- *Cube* (used to show the bounding boxes of the objects in the scene)



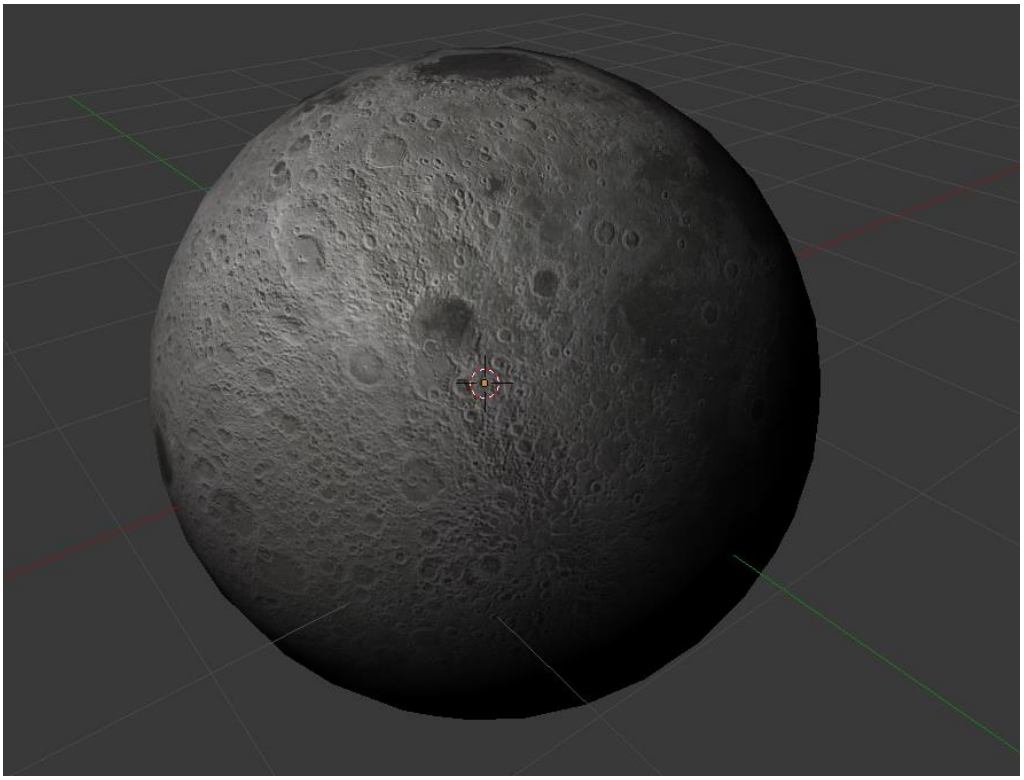
- *Ground*



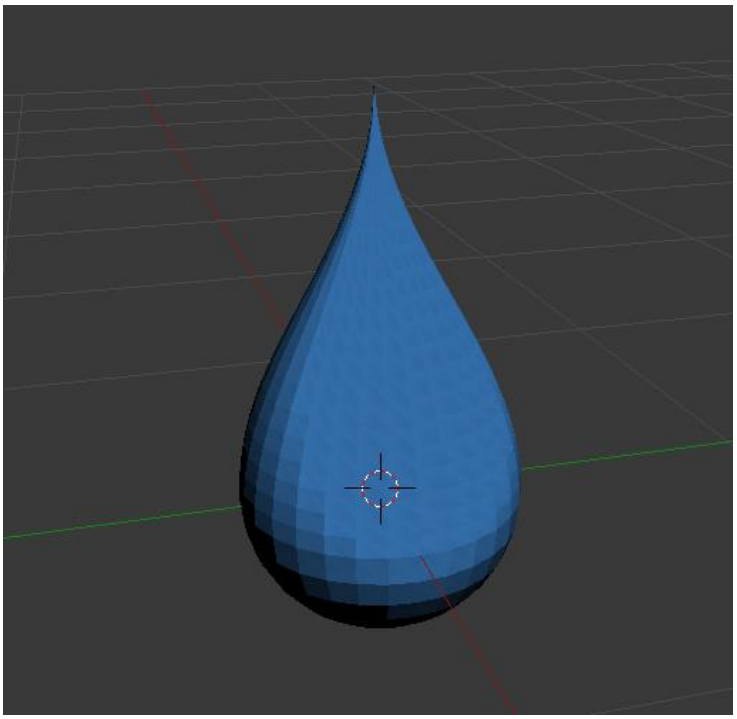
- *Lamp* (which will be positioned on the house)

- *Light Post*

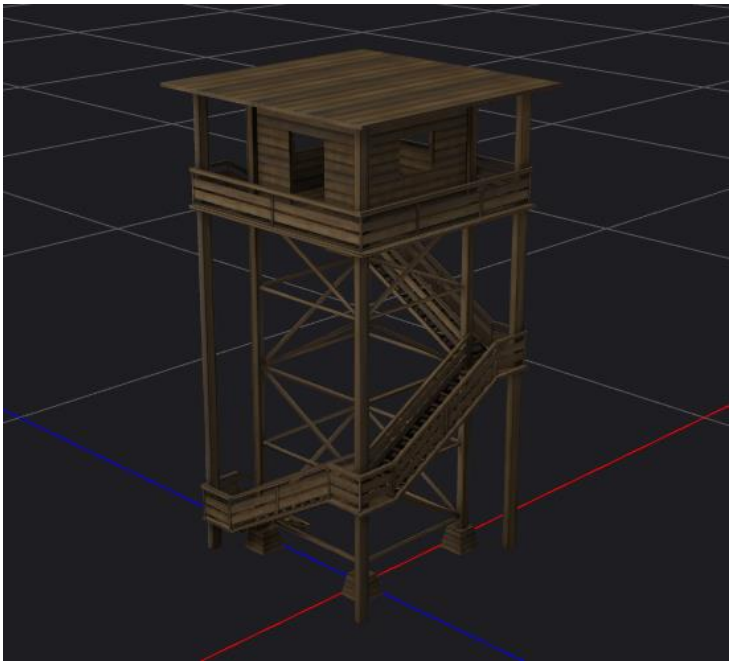
- *Moon*



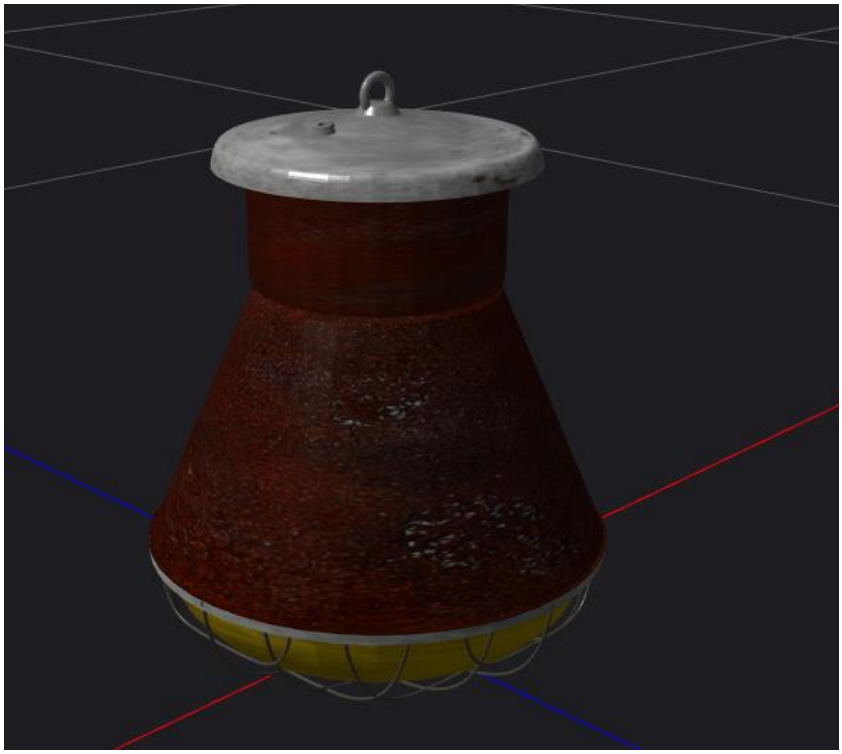
- *Raindrop*



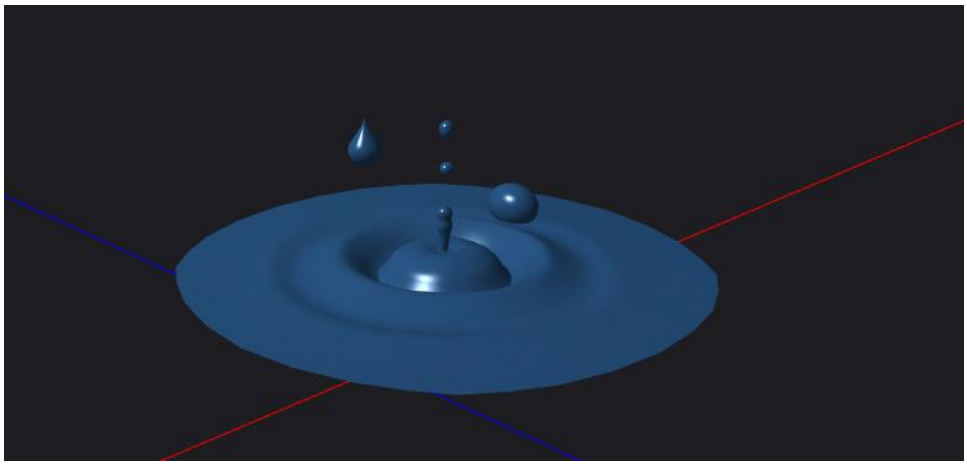
- *Tower*



- *Spotlight Lamp*

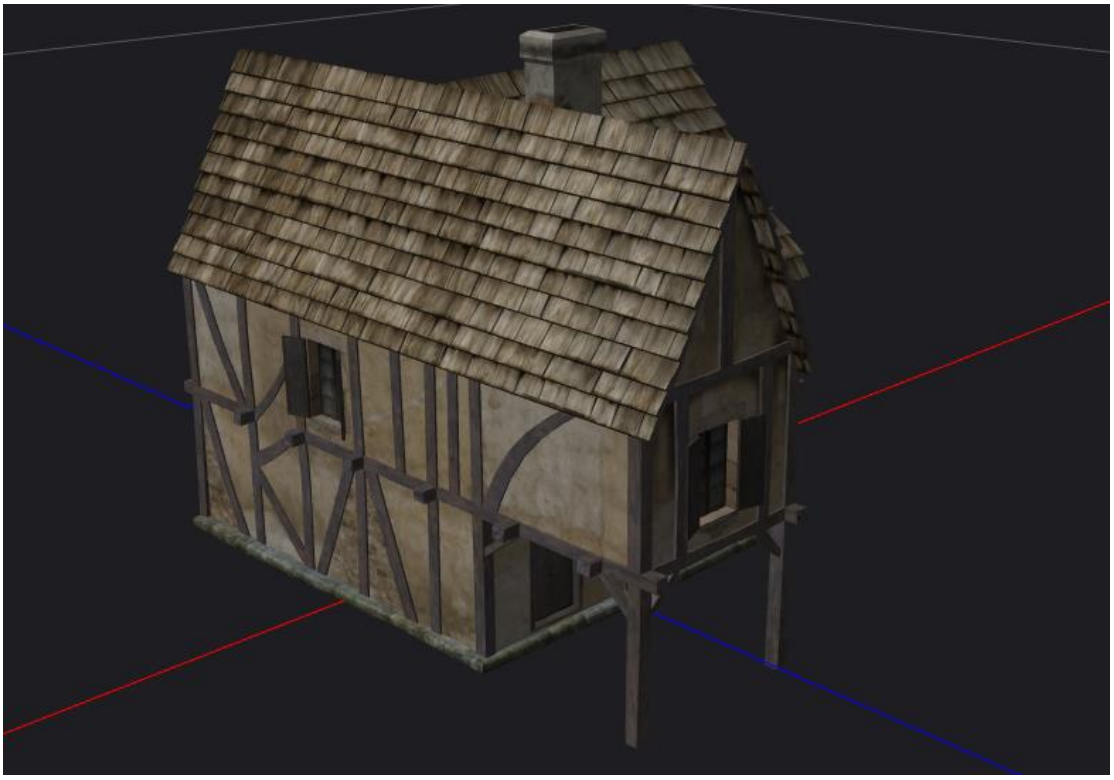


- *Watersplash*



- *Woodlogs*

- *Wooden house*



b. Functionalities

The scene can be explored by the use of the keys WASD on the keyboard together with the mouse. The scene can also be viewed with a self-animated scene animation.

All the objects in the scene (without the water drop, the water splash and the ground) have bounding boxes which can be viewed and which also prevent to enter the objects that have them.

There are multiple light sources:

- Directional light

The light given by the moon. The moon revolves around the scene and lights it. The light direction is the directional vector from the center of the . This type of light is not influenced by the distance between

- Point light

There are two point lights in the scene, one given by the lamp and the other one given by the light spot. The point lights lighten everything around them, no matter the direction, but the longer the distance, the more attenuated the light is.

- Spot light

There is one spot light present in the scene given by the spotlight lamp. This type of light functions on the same principle as the spot light, but gives light into a cone shape. The spotlight lamp can be moved by the user, thus influencing the way the spotlight lightens the scene. The spotlight can also be turned on and off.

The scene also contains the shadows generated by the way the directional light interacts with the entire scene (some objects shadows are not drawn as they would have been redundant; objects like the moon, clouds, water drops, water splash). The shadow computation algorithm is the one taught in the laboratory (the algorithm using a depthMapShader). We have also applied a PCF algorithm on the maps in order to smooth them.

Based on the distance between the viewer and the scene, the objects are also affected by fog. The fog density can be manipulated by the user.

Another functionality present in this project is the rain. By pressing a key, an animation will start containing clouds that gather around the scene. When the clouds have reached their final position, it will start raining. The so called "gravity" (the speed at which the rain drops fall) can be manipulated. When a rain drop hits a bounding box, a water splash object appears. The rain can also be manipulated by a wind.

3. Implementation details

a. Functions and special algorithms

i. Possible solutions

- Light computation

Each of the three different types of light casters have been implemented either by following the laboratory material or internet material. The directional light is computed based on three components: ambient light (the light that exists in the environment), diffuse light (the light that is scattered on an object by the light source; it does not depend on the viewer's position, but it does depend on the light position) and specular light (the light that is reflected by the surface; depends on both light position and viewer's position). The point light is computed exactly the same as the directional light, but we also apply an attenuation based on the distance from the point being lightened to the light source position (the greater the distance, the less the point is lightened). The spot light is computed exactly the same as the point light, but we also apply an additional computation to determine if the fragment's position is inside the cone of the spotlight or not.

- Shadow computation

Shadow is computed only from the perspective of the directional light. The shadow algorithm is the one present in the laboratory material and is build on the following idea. We first render the scene from the light's perspective thus obtaining a depth buffer containing all the points that the light can "see". Afterwards, we render the scene from the viewer's perspective, but, for each point, we check to see if it is being lightened or not. In order to reduce shadow artifacts, we apply a "hack" called shadow bias where we offset the shadow's position. In order to smoothen the edges of the shadow, we apply PCF

(percentage-closer filtering) which basically means that, for each point, the shadow is being computed based on itself and the points around it. The final value of the shadow is then used in computing the directional light (it does not affect the ambient light).

- Fog computation

The fog computation formula has been taken from the laboratory material. Basically, based on the distance between a point and the viewer's position, we compute a fag factor that is influences the final color of the point.

- Scene movement

Both keyboard movement and mouse movement affect the camera state. Keyboard movement affects the camera's position, the other variables remaining the same. Mouse movement, however, affects the camera direction (where we are looking at). Mouse movement is implemented using two angles, pitch (affecting the transversal axis) and yaw (affecting the vertical axis).

- Rain

When the rain animation starts, first the clouds appear. From their initial position to their final position, each cloud is translated by using interpolation of the line based on a number of steps. When the clouds reach their final position, the rain starts. The rain is computed as follows: at each scene rendering, we generate a number of raindrops which have a random value x between the xmin and xmax of the ground, a random value z between zmin and zmax of the ground, and at a height y variable depending on the wind (when there is wind, we still want to have rain in the entire scene, so we have to generate raindrops lower). The raindrops that have been generated previously, change their position based on the variable gravity (a vector with just the y component being not different from 0) and the wind (a vector with x and z variable). When the rain exists the xmin and xmax or zmin and zmax of the ground they disappear. They also disappear when they go below the ground. Also, when they encounter a bounding box, the point disappears and, for a single frame, a water splash appears.

- Bounding boxes

The bounding boxes have been implemented as follows:

- When the object is loaded, we compute the xmin, ymin, zmin, xmax, ymax and zmax values of all the points that make up the object. Thus, with these values, we can compute the two points that are diagonally opposite and make up the initial bounding box of the object.
- Afterwards, if the object is translated or scaled, we apply those transformations on the two points and obtain a new bounding box. However, if the operation is rotation, we need to compute a new bounding box. Thus, from this 2 points, we compute all the 8 points of the bounding box, we apply the rotation on them, and we compute the new xmin, ymin, zmin, xmax, ymax and zmax values which will give us the new bounding box.

ii. The motivation of the chosen approach

All of the approaches presented before have been chosen based both on their difficulty but also on their outcome. I think that these approaches have the best difficulty to result ratio of the algorithms and approaches I have found.

b. Graphics model

All of the models used in this project are 3D models with .obj extensions accompanied by different .mtl files. All of these objects have been from internet resources, modified in Blender, and then exported. In case they did not have .obj extension, they were imported in Blender and, afterwards, exported in the desired extension. Example of internet resources:

- https://free3d.com/3d-models/obj
- https://www.turbosquid.com/
- https://www.cgtrader.com/
- https://sketchfab.com/models

c. Data structures

Besides glm/glsl specific data structures (such as vec3, vec4, mat3, mat4), I have also implemented my own data structures used to develop an easier implementation of the project. The data structures are:

- Object data structures: each object has associated a data structure, containing the translation vector, scale vector, rotation angle, transformationMatrix (if the object is static; in order to improve the efficiency), the bounding box of the object after placing a static object in the scene.
- Light data structures: for each type of light, we have defined data structures (which are sent to the shaders) containing information about each type of light, like ambient, diffuse, specular components, light position, light color etc. Based on the light type, the data structures have different fields.
- A bounding box data structure associated to the Model3D class used to store the initial bounding box of the object.

d. Class hierarchy

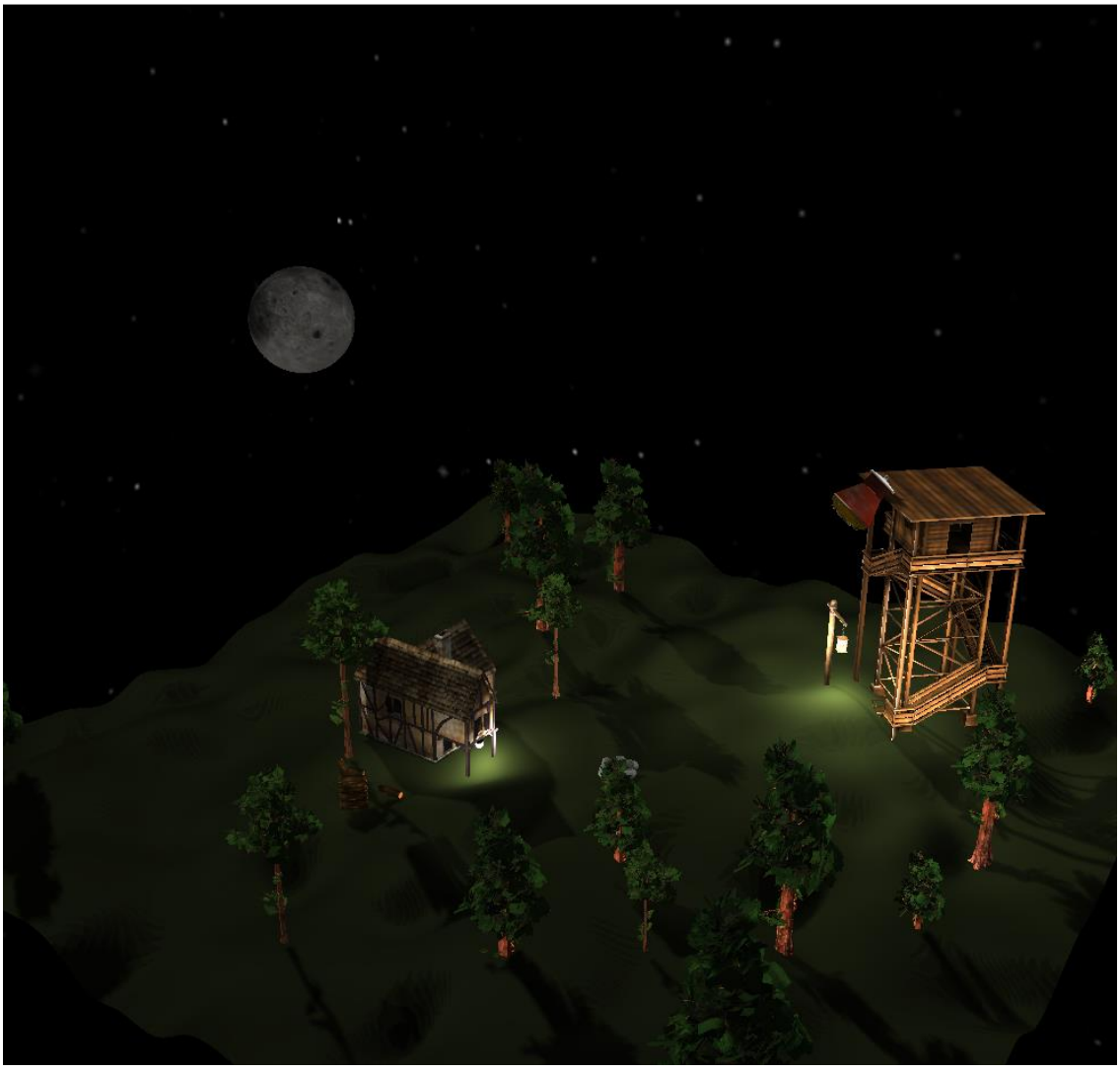
The project is composed by one file containing the main function and some additional C++ helper classes implemented in the laboratory material. The file containing the main function is the core of the project. The other files are used in order to easier represent and manipulate data. Both Mesh and Model3D classes help us in defining 3d models based on .obj and .mtl files. The Shader class eases the use of multiple shader programs. The Camera class facilitates the movement in the scene.

4. Graphical user presentation/ User manual

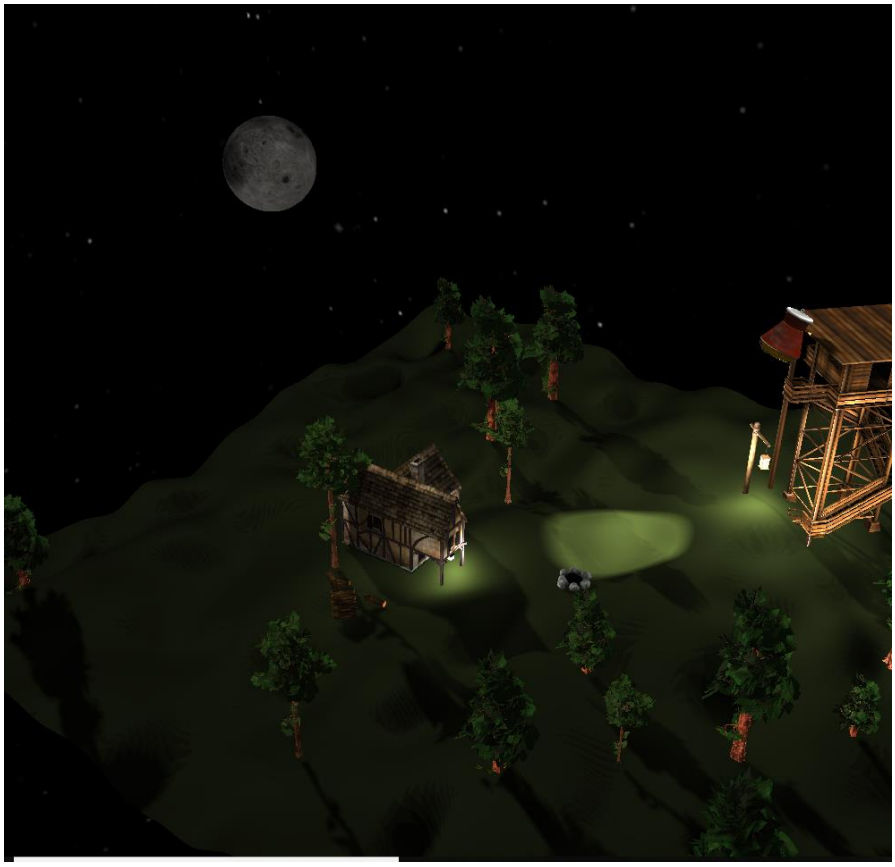
- **1** ~ view wireframe representation of the scene
- **2 ~** view normal representation of the scene
- **3** ~ view point representation of the scene
- **W ~** move camera forward
- **S ~** move camera backward
- **A ~** move camera to the left
- **D ~** move camera to the right
- **U ~** increase rain speed on Y axis
- **J ~** decrease rain speed on Y axis
- **O ~** start cloud animation
- **P ~** stop cloud animation
- **H ~** increase X component of the wind
- **G ~** decrease X component of the wind

- **K ~** increase Z component of the wind
- **L ~** decrease Z component of the wind
- **I ~** decrease the speed of the cloud animation (must be performed before the animation is started)
- **Y ~** increase the speed of the cloud animation (must be performed before the animation is started)
- **X ~** increase fog density
- **Z ~** decrease fog density
- **V ~** start camera animation
- **B ~** stop camera animation
- **M ~** draw bounding boxes
- **N ~** not draw bounding boxes
- **UP Arrow ~** rotate spotlight lamp upwards (implicitly affecting the spotlight)
- **DOWN Arrow ~** rotate spotlight lamp downwards (implicitly affecting the spotlight)
- **Q ~** turn on spotlight
- **E ~** turn off spotlight

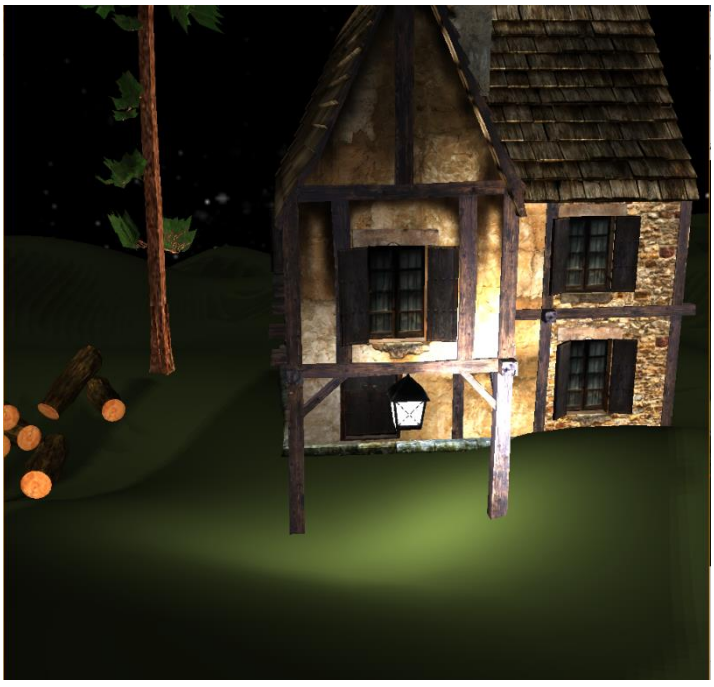
a. General overview of the scene



b. Overview of the scene with spotlight turned on



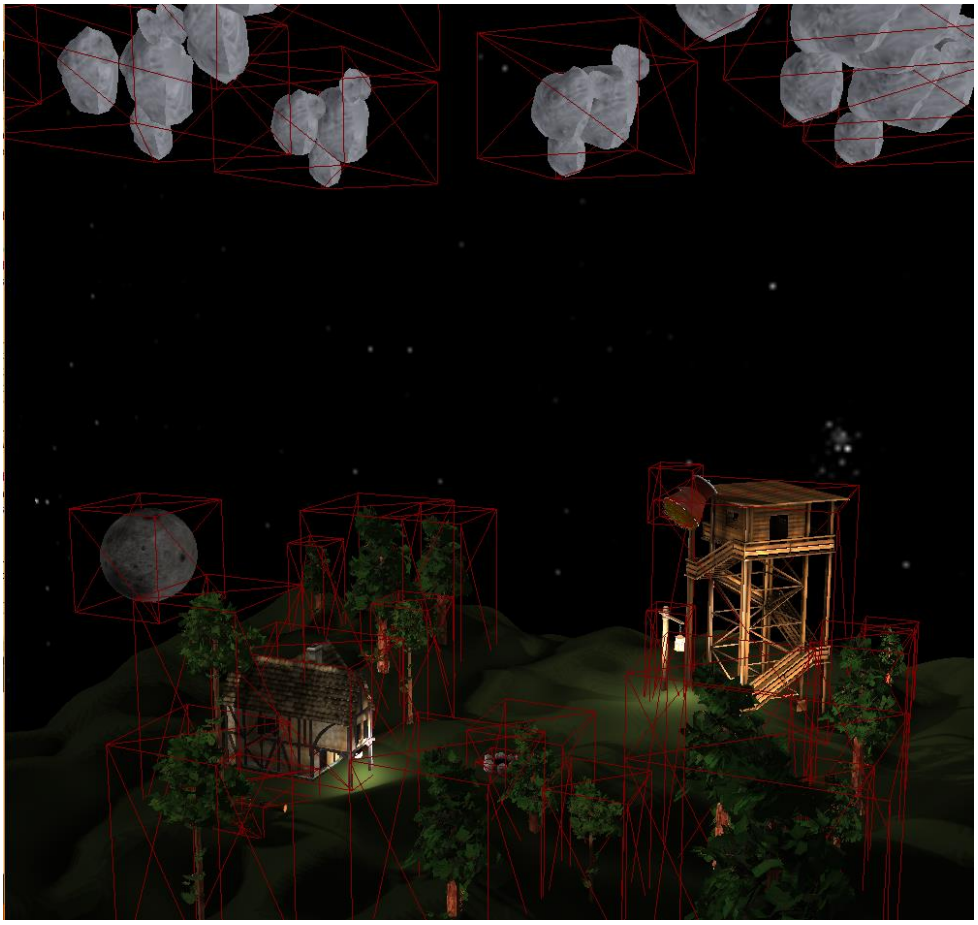
c. Point light inside the lamp near the house



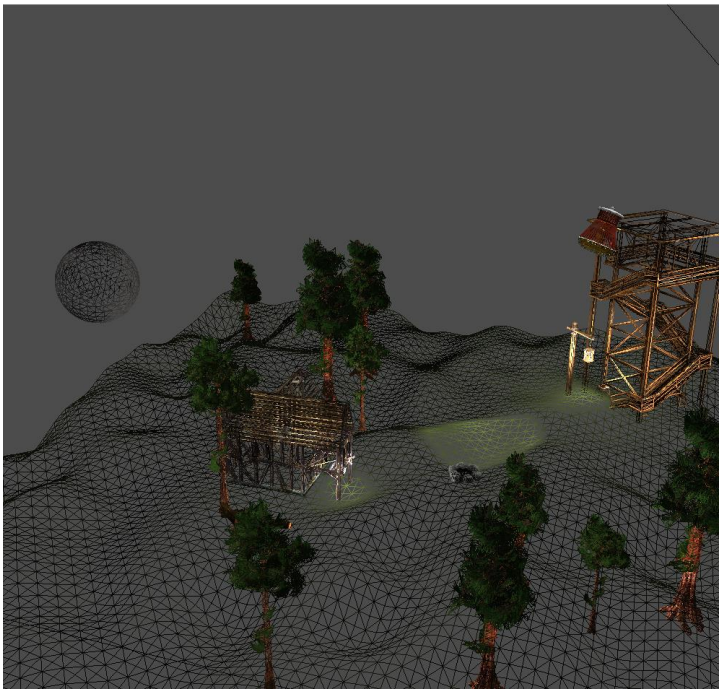
d. Point light near the tower

e. Cloud animation

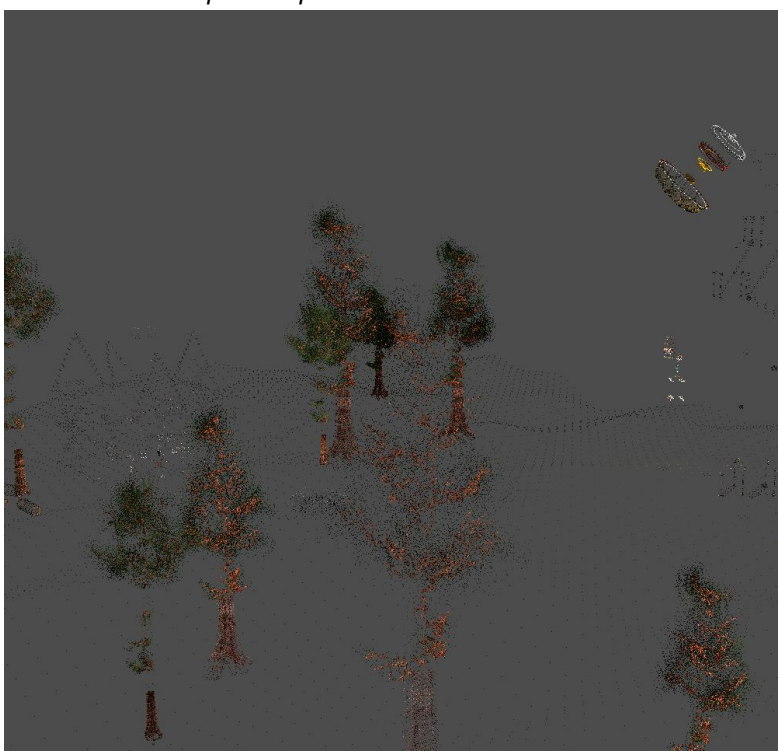
f. Scene overview with drawn bounding boxes



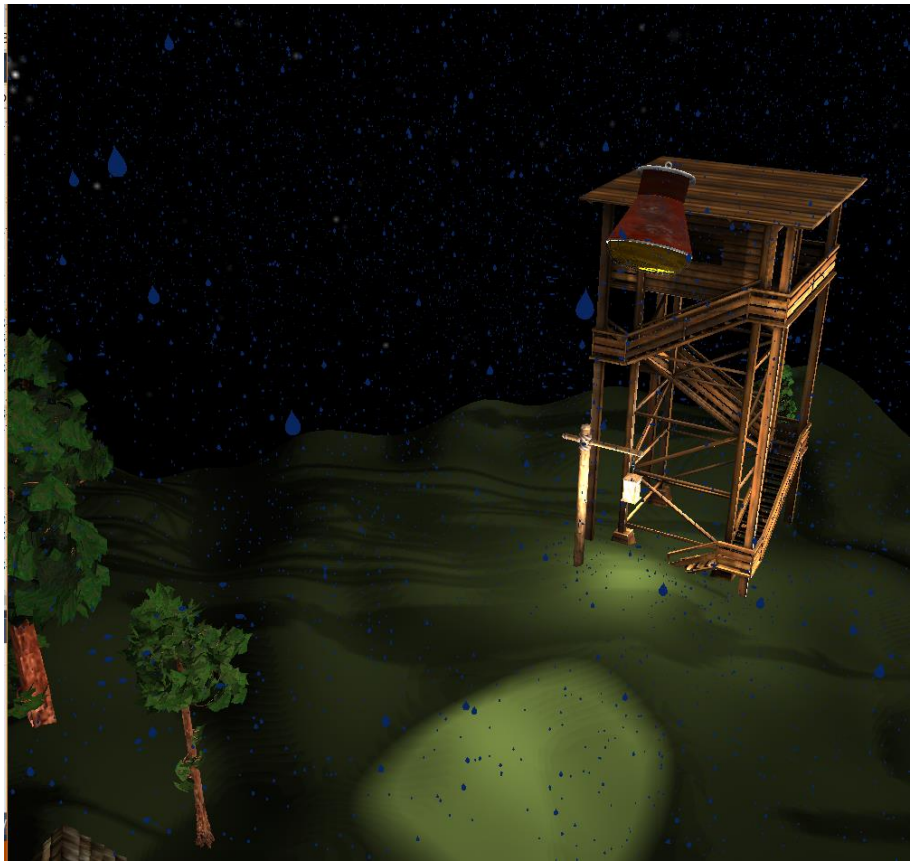
g. Scene overview with wireframe representation



h. Scene overview with point representation



i. Rain with wind effect



j. Fog example

k. Shadow with PCF computation

5. Conclusions and further development

The project is far from perfect, but, for an introductory project in the art of graphics, I would say it is really good. However, some additional improvements that could be made are:

- Increase the photorealism of the scene
- Create a day/night cycle
- Add some human/animal entities in the scene

6. References

My main reference (besides the laboratory materials) was the website:
https://learnopengl.com/Introduction