

ARMY OUTPOST ON AN ISLAND

GRAPHIC PROCESSING PROJECT

AVRAM ANDREI-ALEXANDRU

GROUP 30434

CONTENTS

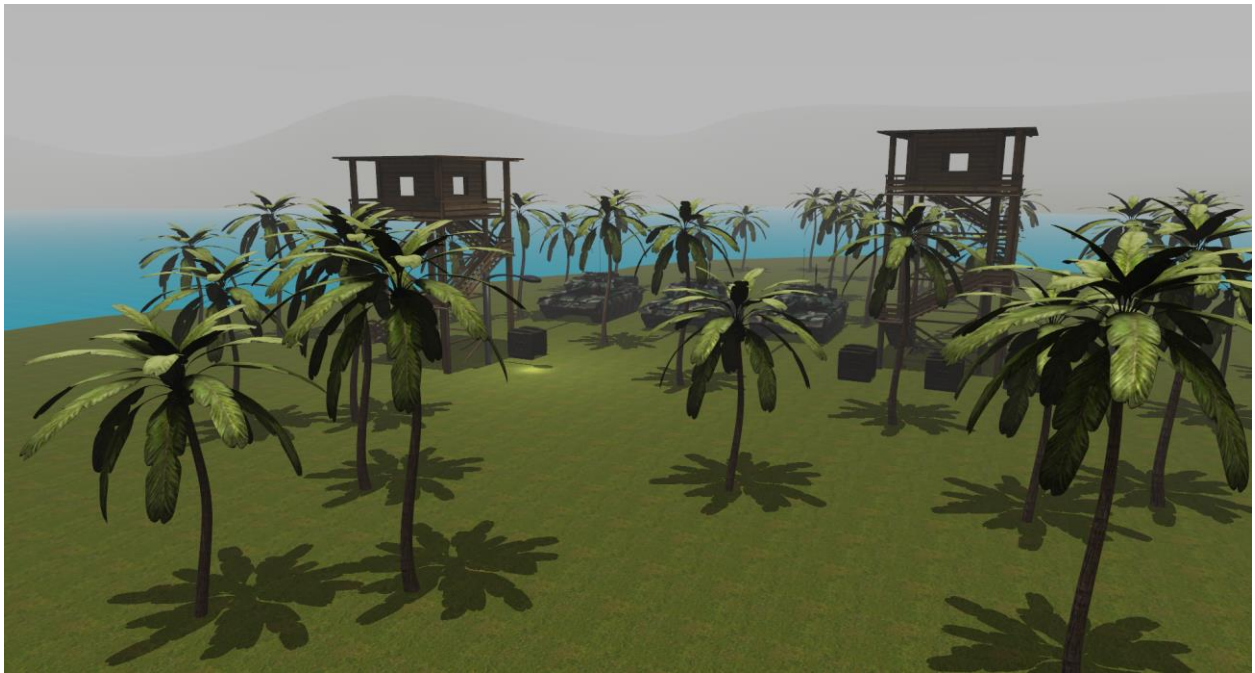
1.	Subject specification	2
2.	Scenario.....	2
2.1	Scene and object description.....	2
2.2	Functionalities.....	3
3.	Implementation details	4
3.1	Functions and special algorithms.....	4
3.2	Graphics model	4
3.3	Data structures	5
3.4	Class hierarchy	5
4.	User manual	6
5.	Conclusions and further development.....	6
6.	References.....	6

1. SUBJECT SPECIFICATION

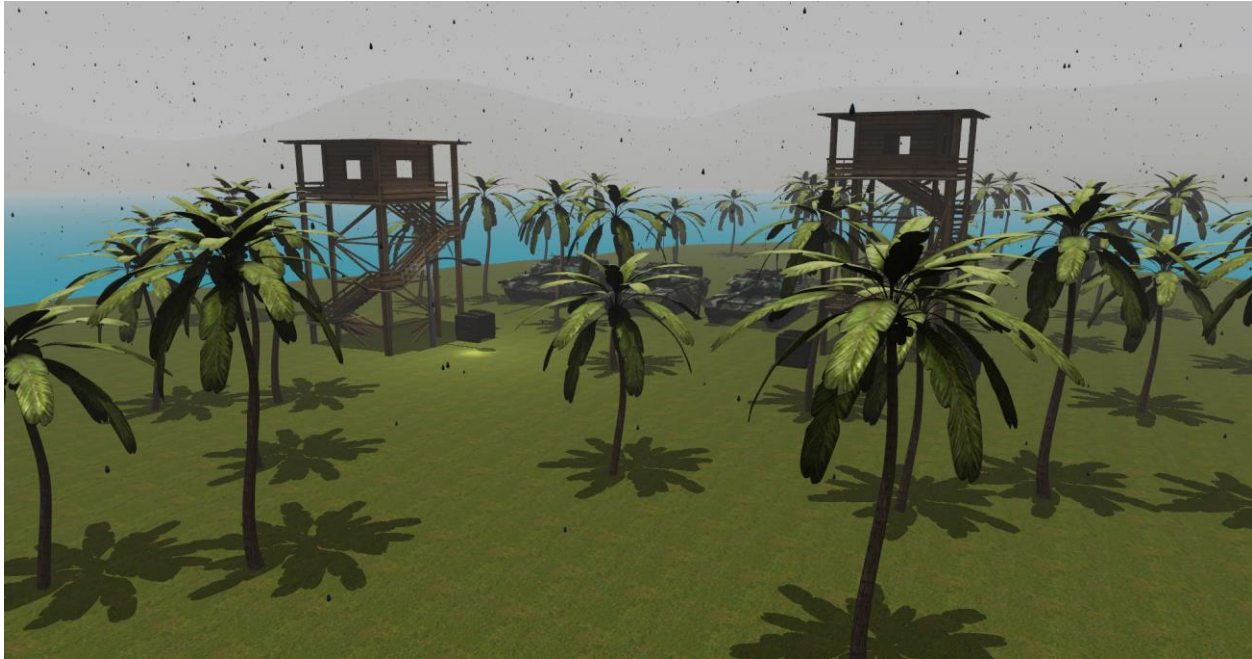
The scope of this project was to build a photorealistic scene containing 3D objects using OpenGL. By using a keyboard and a mouse, the user must be able to manipulate the scene and the camera. The scene also exemplifies multiple light sources, shadow mapping techniques and object animations using scaling, translations and rotations.

2. SCENARIO

2.1 SCENE AND OBJECT DESCRIPTION



Scene without rain



Scene with rain

The scene contains objects related to a military outpost on an island surrounded by hills. The objects that are present in the scene are: watch towers, a soldier, tanks, a lamp post, palm trees and some army crates for storage.

2.2 FUNCTIONALITIES

The project contains the following functionalities:

- Shadow mapping technique to create shadows in the scene
- A directional light
- A point light (present under the lamp post that is near the left watch tower)
- Fog
- Rain
- Visualization of the scene in wireframe, solid and smooth surfaces
- Object animation: Soldier that rotates on spot and rain animation
- Camera animation that shows an overview of the scene
- Detailed and textured 3D objects that correspond to the chosen army theme

3. IMPLEMENTATION DETAILS

3.1 FUNCTIONS AND SPECIAL ALGORITHMS

I will exemplify the algorithm for generating the rain. The algorithm for generating rain is the following:

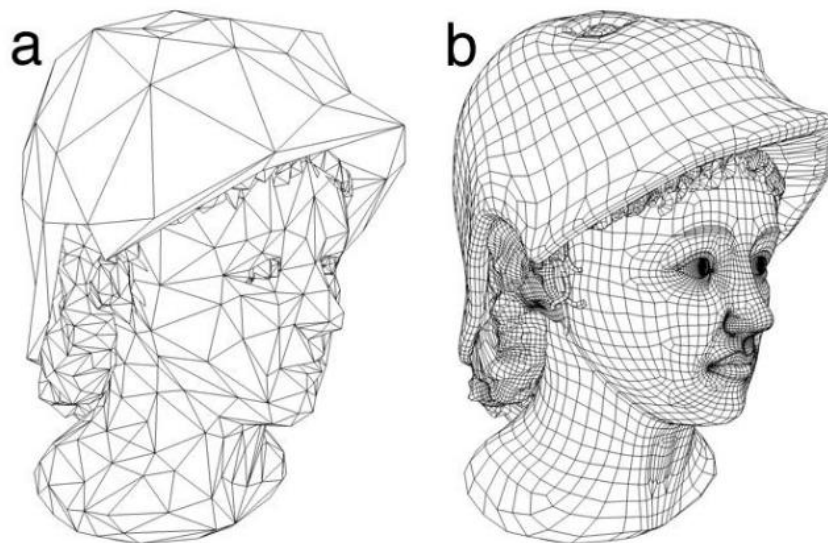
First, when the application is launched the coordinates for the rain drop objects are generated randomly and stored in an array. The values that are randomly generated are the x, y, z coordinates of the objects and the movement speed of the objects. This random generation is done only once to ensure a smooth FPS when running the application. The coordinates and the movement speed are generated in a certain range.

Secondly, when rendering the scene in a for loop going through all the rain droplets (15.000 chosen for the demo) the rain drop is translated to the current location and drawn to the screen. After the drawing takes place the Y coordinate that represents the height of the object is decremented by a value equal to the movement speed of the current object and the object's coordinates are updated to be used in the next call of the render scene. When the rain droplet reaches the ground, its Y coordinates gets reset to certain height to ensure that the object will continue to fall in the part of the scene that is visible to the user.

In summary, the algorithm translates and draws randomly a large number of objects representing the rain droplets.

3.2 GRAPHICS MODEL

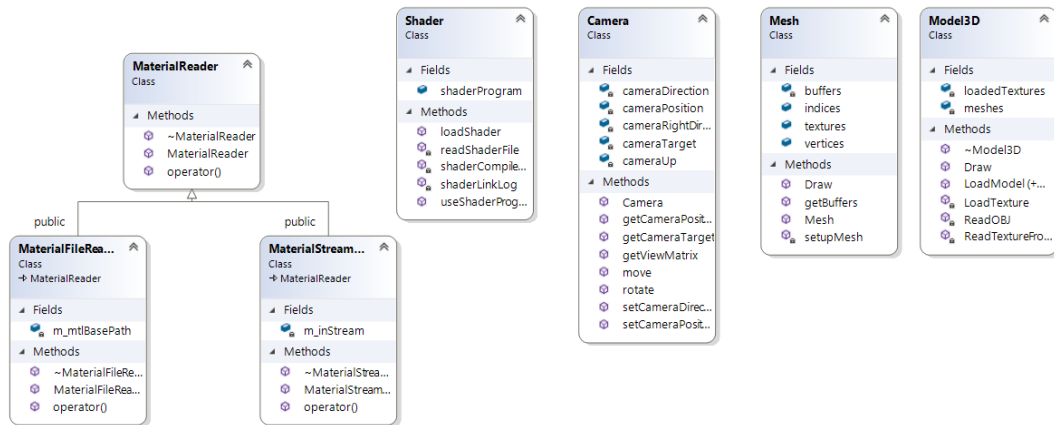
The graphics model present in the project is the polygonal representation of 3D objects.



3.3 DATA STRUCTURES

A 2D float array is used to hold the coordinates and the movement speed for each rain droplet object.

3.4 CLASS HIERARCHY



4. USER MANUAL

KEY	ACTION
1	Start/Stop soldier animation
2	Start camera animation
3	Stop camera animation
5	Start/Stop rain
M	Show/Hide depth map
Q	View wireframe
E	View solid
J	Rotate directional light clock-wise
L	Rotate directional light counter clock-wise
W	Move camera forward
A	Move camera left
S	Move camera backward
D	Move camera right
X	exit

Rotation of the mouse rotates the camera and scroll wheel can be used to zoom in and out.

5. CONCLUSIONS AND FURTHER DEVELOPMENT

The scene can be improved in many ways by adding and animating new objects. Also, multiple algorithms can be implemented like motion blur, bloom, ambient occlusion and many more.

In conclusion, the project was very useful in increasing my knowledge about 3D graphics processing and of the OpenGL API.

6. REFERENCES

<https://learnopengl.com/>

https://moodle.cs.utcluj.ro/pluginfile.php/101841/mod_resource/content/0/gps-04-3D%20ObjRep%20%28P1%29.pdf

<https://free3d.com/>