

# Assignment 3 Report

Janic Moser, 1155210428

November 1, 2023

## 1 Frames

- Figure 1 shows the overall Scene of the Project.
- Figure 2 shows the lightning of the sun, planet, asteroids, crafts and spacecraft.
- Figure 3 shows the rotation along the z-axis based on the cursor position

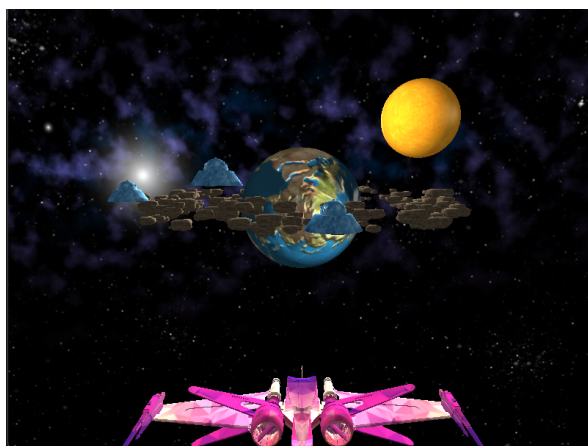


Figure 1: Overall scene of the program

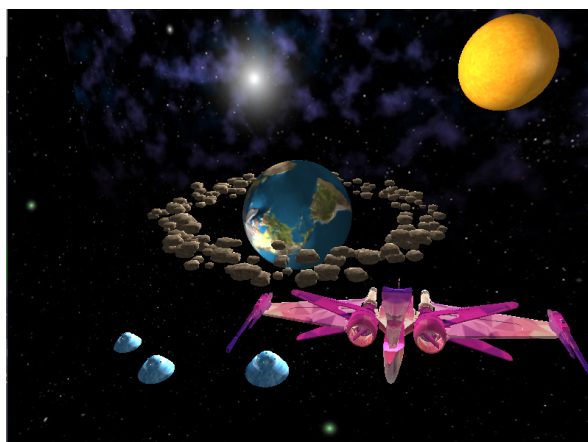


Figure 2: Shows Lightning



Figure 3: Rotation along z-axis

## 2 Requirements

- Render one planet, one spacecraft and at least one craft: Visible in figure 1
- Self-rotation for the planet: Visible in demo
- Render a skybox: Visible in figure 1
- Basic light rendering: Visible in figure 2
- Render an asteroid ring cloud: Visible in figure 1
- The rotation of the rocks: Visible in demo
- Correct viewpoint: Visible in demo, and figure 1/2/3
- Normal mapping for the planet: Visible in figure 1
- Use mouse to control the translations of the spacecraft: Visible in demo and figure 2
- Use keyboard to control the translations of the spacecraft: Visible in demo
- BONUS: Another light Source: Sun, visible in figure 1
- BONUS: Additional meaningful object: Sun, visible in figure 1
- BONUS: Interesting interactions: Mouse position leads to rotation of spacecraft in z-axis. Visible in figure 3

## 3 Implementation details

### 3.1 Previous assignments

A lot of methods/techniques were used from previous assignments. Such as basic lightning, object loading, shader, interactions etc.

### 3.2 Implementation based on tutorials

The implementation of the normal mapping, skybox and camera interaction are heavily based on tutorials 7-9, with slight adjustments.

### 3.3 Self rotation

Self rotation of the planet was achieved by incrementing the rotation angle everytime we call `paintGL()` by a small amount. The same technique was used for the rotation of the asteroid cloud.

### **3.4 Asteroid cloud**

The asteroid cloud consists of 200 rock objects. We reused the model and just changed the transformation matrix for each object. The parameters get initialized in main.cpp, lines 244-251.

### **3.5 Rotation based on cursor position**

The rotation angle along the z-axis of the spacecraft is based on the cursor position. For concrete implementation, please consult main.cpp, line 541.