

Report

-CSCI3260 Course project-

Name: CHOI, Jun Seok / SID: 1155100532

1. Overall Rendering Scene

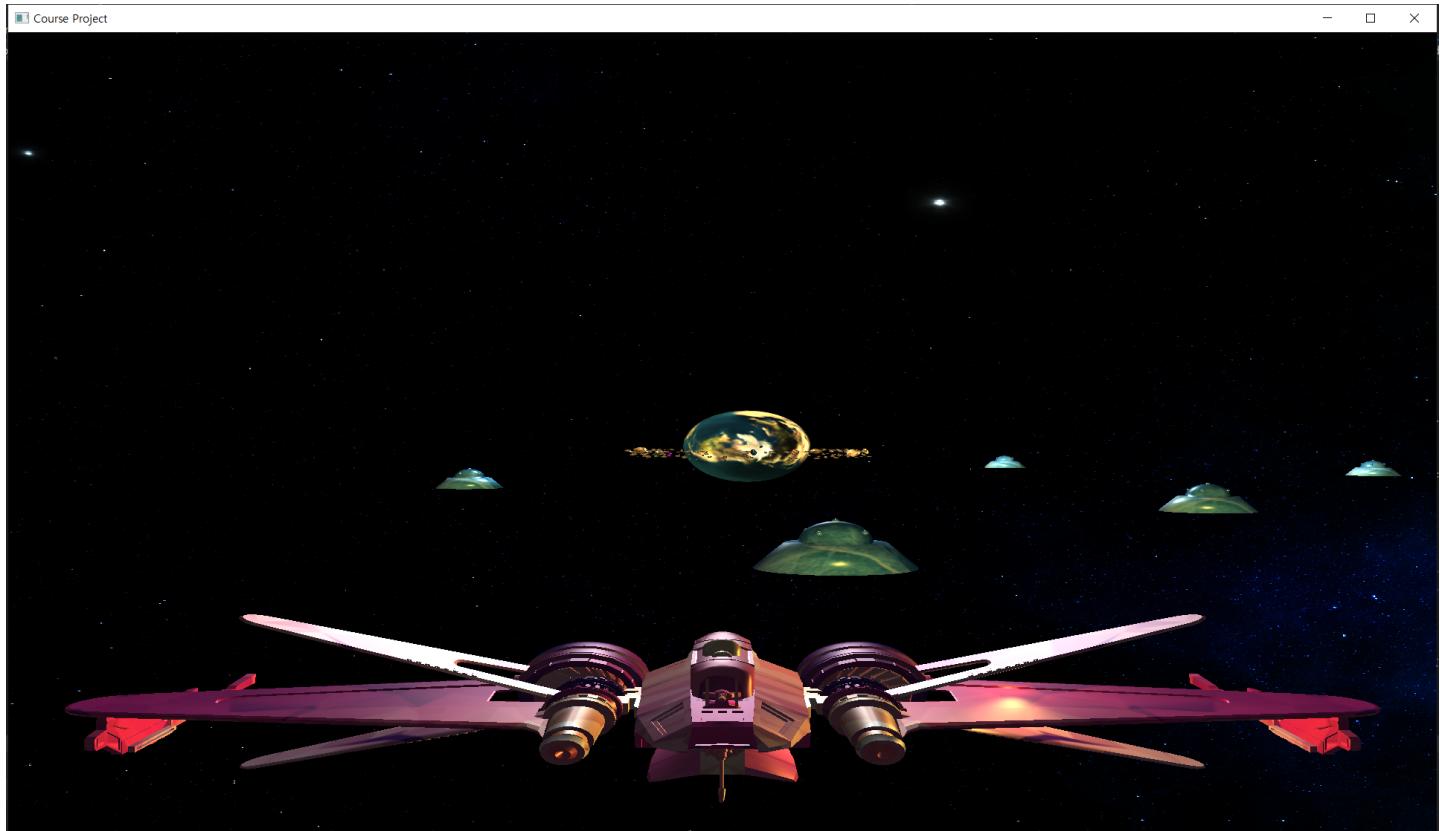


Fig1. Overall scene

2. Point light Rendering scene

Point light features:

1. **Ambient:** `glm::vec3(0.1f)`
2. **Diffuse:** `glm::vec3(1.0f)`
3. **Specular:** `glm::vec3(1.0f)`
4. **Position:** `glm::vec3(0.0f, 10.0f, -20.0f)`
5. **Coefficients:** constant=1.0f, linear=0.007, quadratic=0.0002. Which covers 600 distances.

- **Spacecraft:**

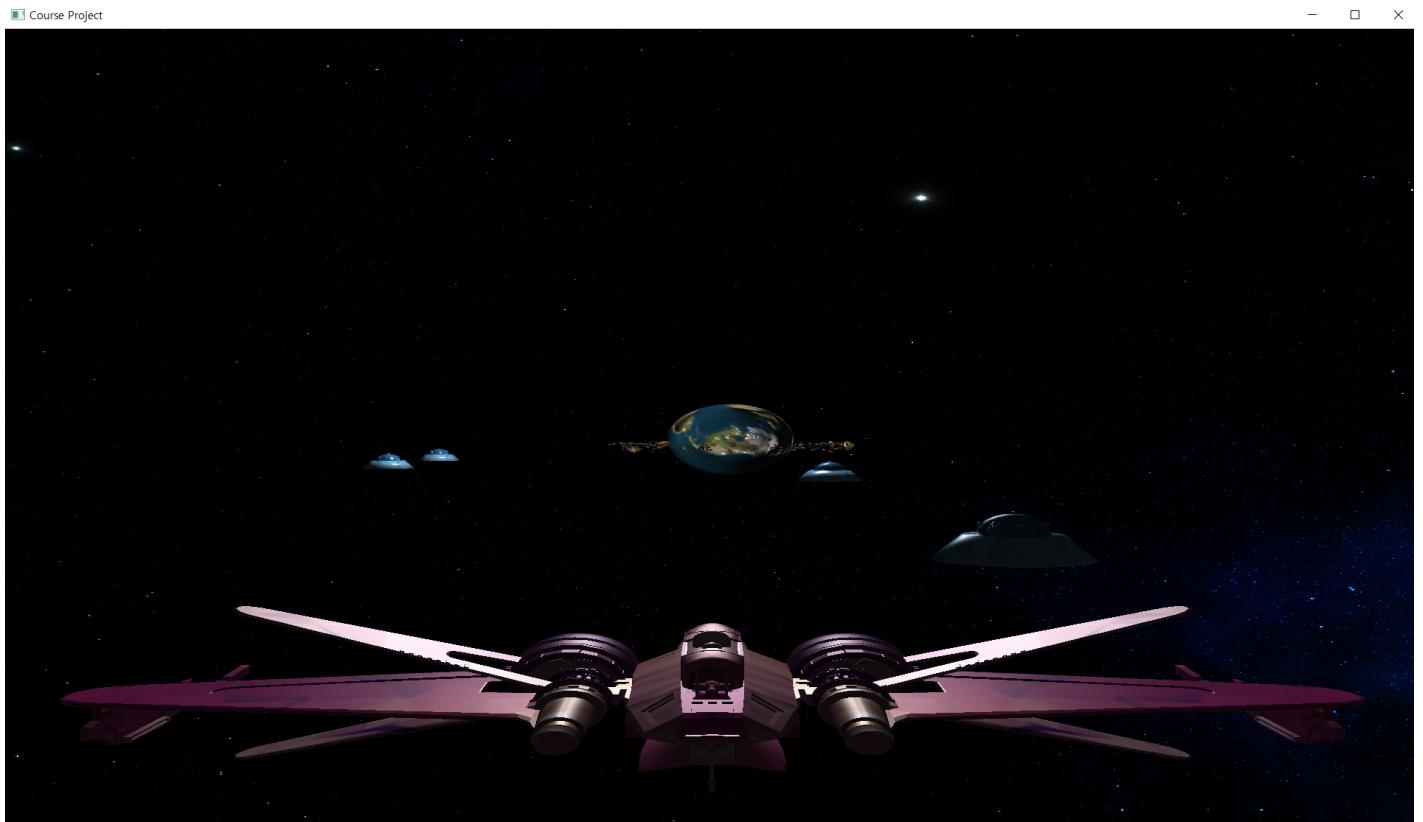


Fig2. Light source is located at the front-side of the spacecraft

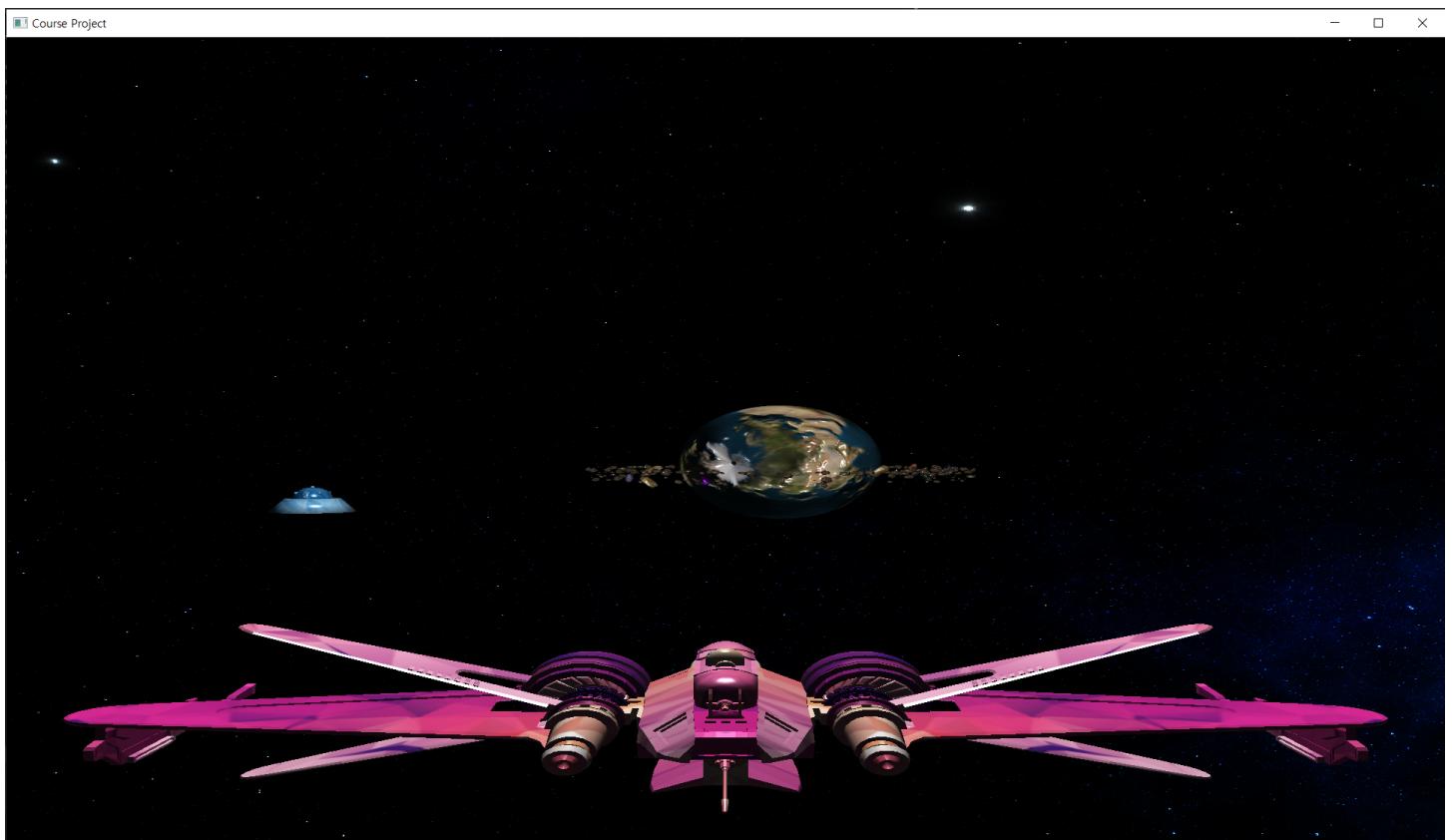


Fig3. Light source is located at the back-side of the spacecraft

- Local vehicle:

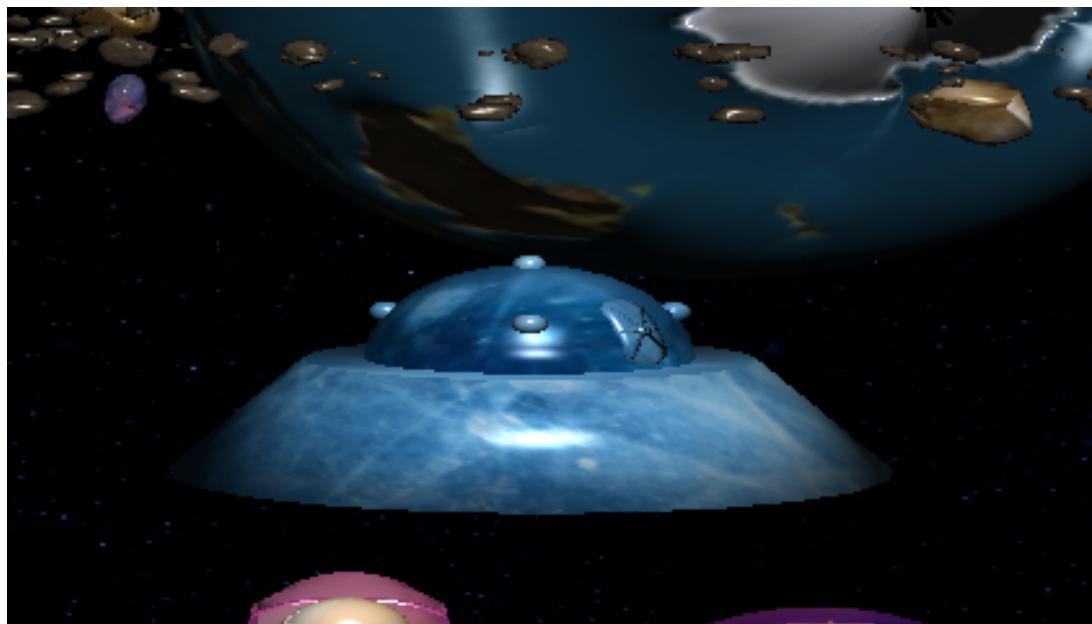


Fig4. Local vehicles interacting with point light

- Planet and rocks:

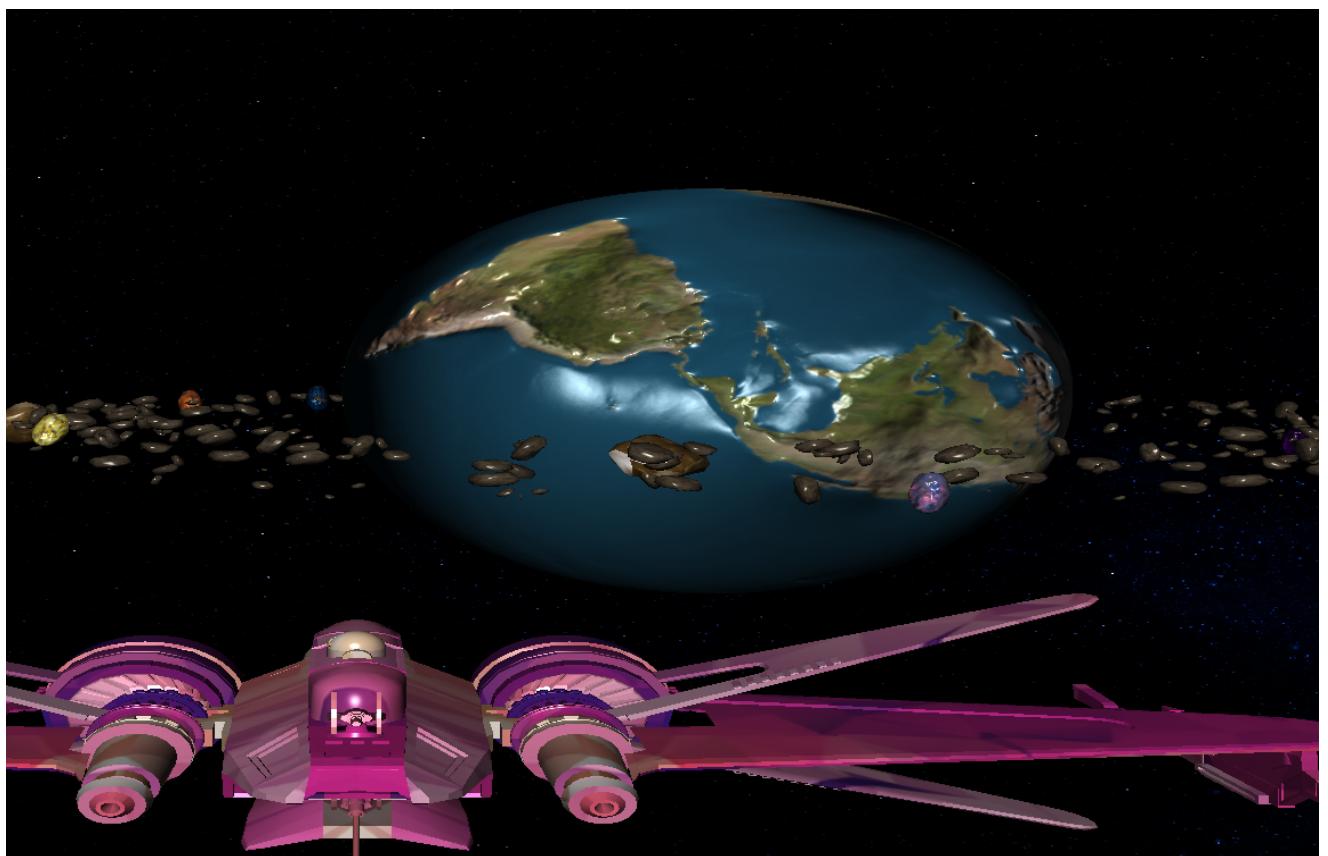


Fig5. Planet and rocks interacting with point light

3. Spacecraft interaction scene

- Alerting scene:

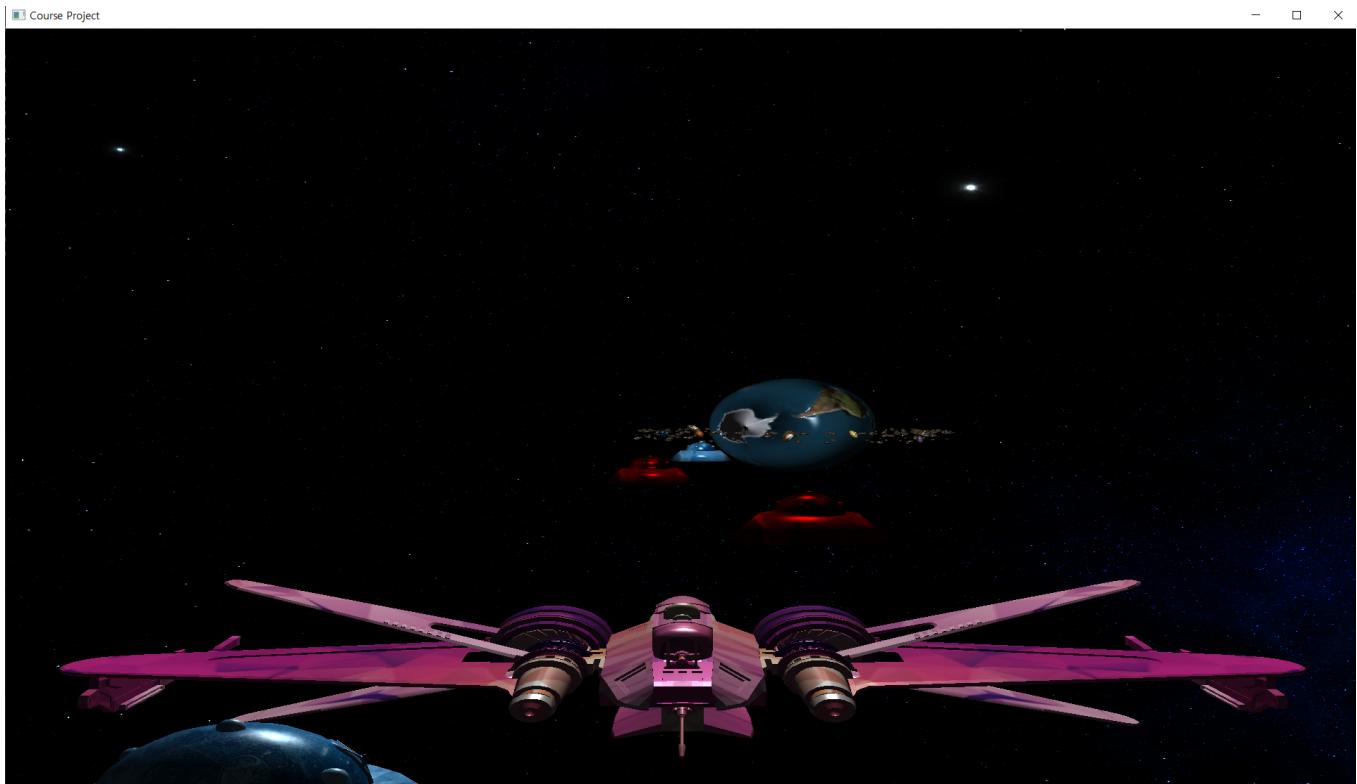


Fig6. Two local vehicles alerting after collide with spacecraft

- Before collecting golds:



Fig7. Before collecting a gold

- After collecting a gold:

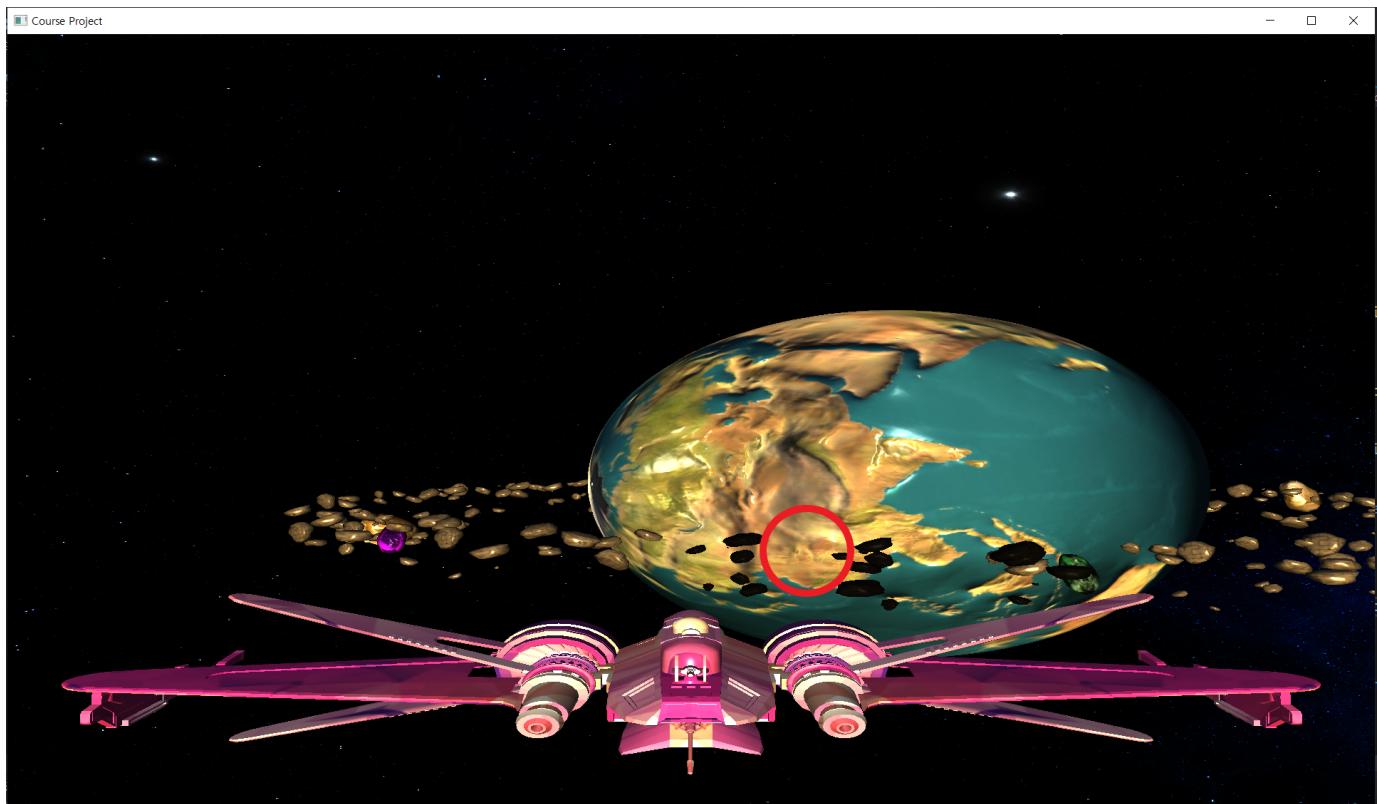


Fig8. After collecting a gold

- After collecting all golds:

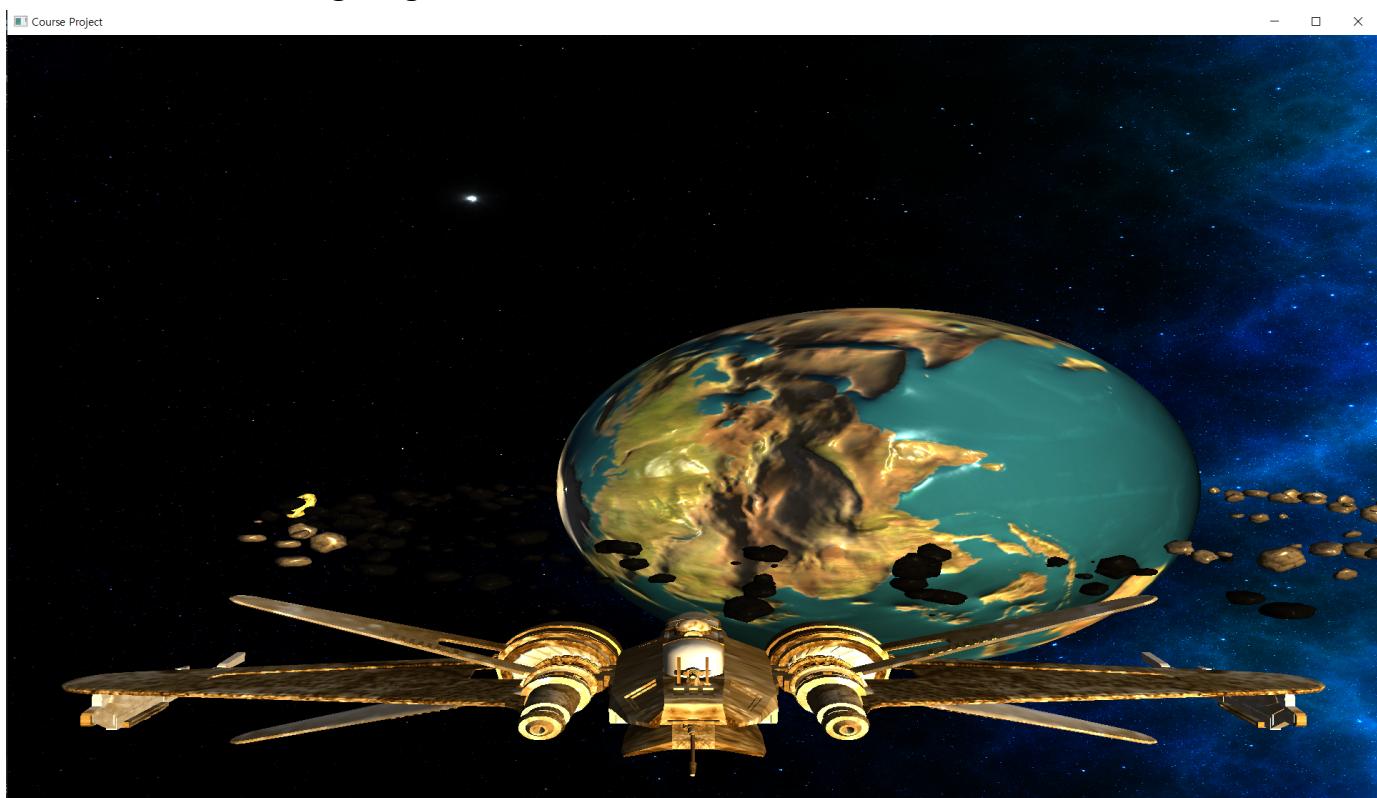


Fig9. After collecting all golds, spacecraft wears gold texture

4. Additional features

Directional light:

- **Directional light features:**

1. **Ambient:** `glm::vec3(0.1f)`
2. **Diffuse:** `glm::vec3(0.876f, 0.654f, 0.210f)`
3. **Specular:** `glm::vec3(0.876f, 0.654f, 0.210f)`
4. **Position:** `glm::vec3(1.0f, 1.5f, 5.0f)`

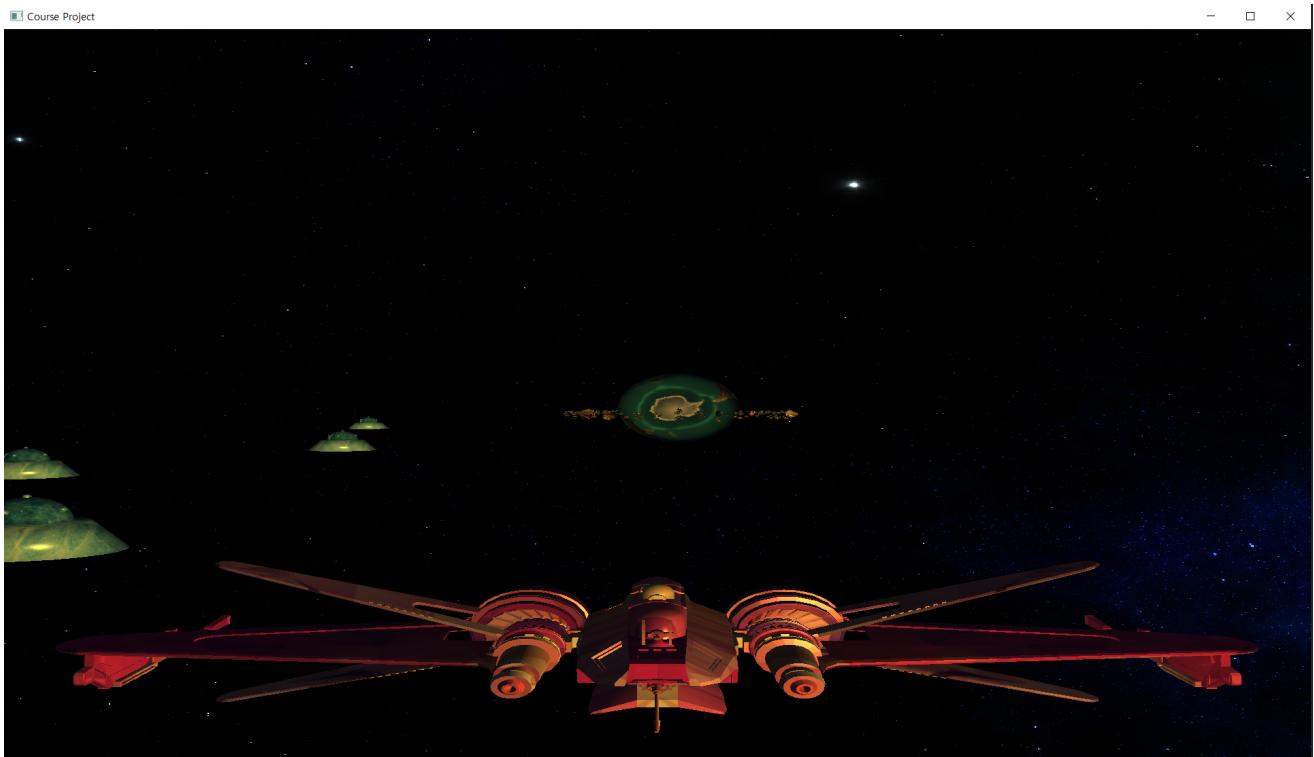


Fig10. Scene with directional light only

Normal mapping:

- Planet
- Stones

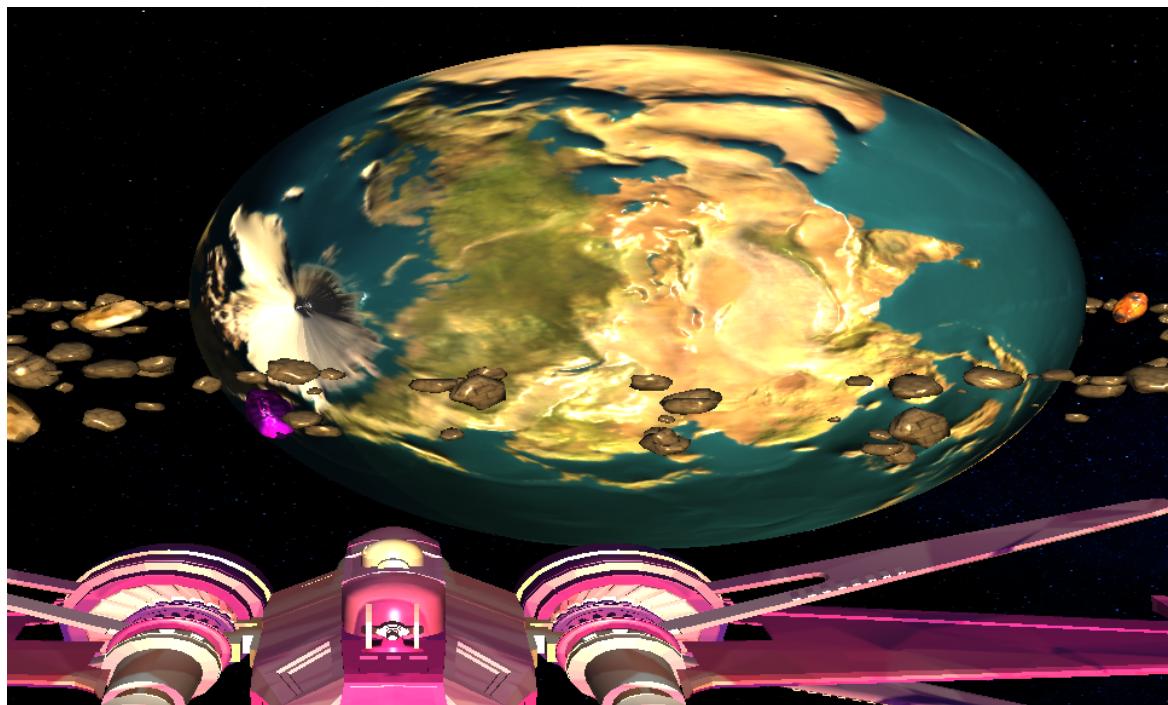


Fig11. Planet with normal mapping

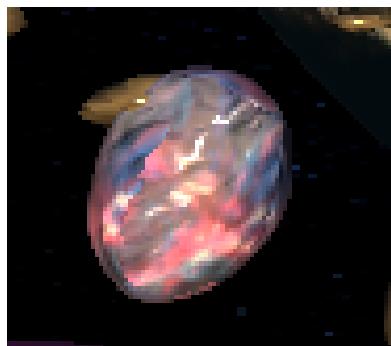


Fig12. Stone with normal mapping

More kinds of treasures(6 infinity stones):

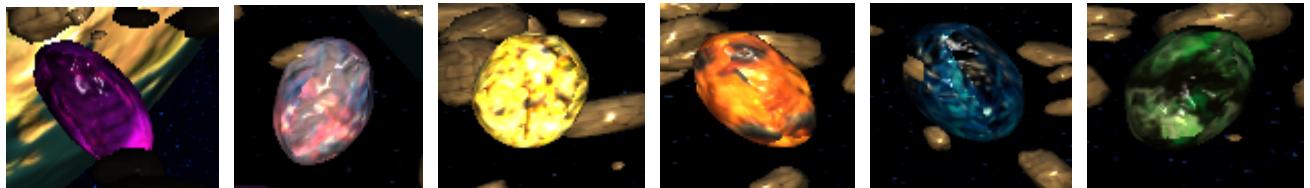


Fig13,14,15,16,17,18. From the left, power, reality, mind, soul, space, time

After collecting all 6 infinity stones => Planet destroyed!



Fig19. Destroyed planet

5. Details

- I used two shaders. One for normal mapped objects(planet and stone) and another one for other objects.
- Six stones are the same object but different textures.
- Rotation speed of golds and stones is half of the rotation speed of rocks for distinction.
- The position, inclination and scale of rocks, golds and stones are randomized within a limited boundary.
- There are two light sources(point and directional) only. If you press 1, directional light is toggled and press 2, point light is toggled.
- For the collision threshold, I calculated the average of the size of three coordinates(x, y, z) for each object and set the value as radius of each object. Then, set the half of the sum of radius of each object as the threshold.
- The position of the spacecraft is set by the inverse of the view matrix with some other transform matrix. It means the spacecraft is dependent on the camera position and the camera is independent.
- For mouse movement, I experimented that the maximum offset of the mouse is less than 200.0f per one frame and I set the 200.0f as threshold for the detection of the mouse movement. Hence, there is no sudden jump of the camera direction even if the mouse cursor is out and in at the different side of the window. However, there is still some problem that if the cursor is in through left(or right) and out through y-axis without moving more than 200.0f along the x-axis and in through left(or right) the x-axis again, the camera view direction suddenly jumps a bit.
- There are five local vehicles moving along the x-axis. Each local vehicle has a different speed and patrol boundary.
- For alerting texture, I simply used a red color vector(1.0, 0.0, 0.0).
- I used the assimp library and stbi_image.h for loading models and textures.