CSED451 Computer Graphics (Spring 2021)

Assignment #3-1 & #3-2

3D Drawing

Assignment #3-1 Due date: April 23th, 2021, Friday, 11:59pm

Assignment #3-2 Due date: May 7th, 20201, Friday, 11:59pm

The goal of assignment 3 and 4 is to reimplement the shooting game in 3D. In assignment 3, you focus on 3D wire-frame rendering and 3D camera control. In this assignment, **you don't have to implement lighting, shading, and texture mapping; the three categories will not be counted as additional scores.**

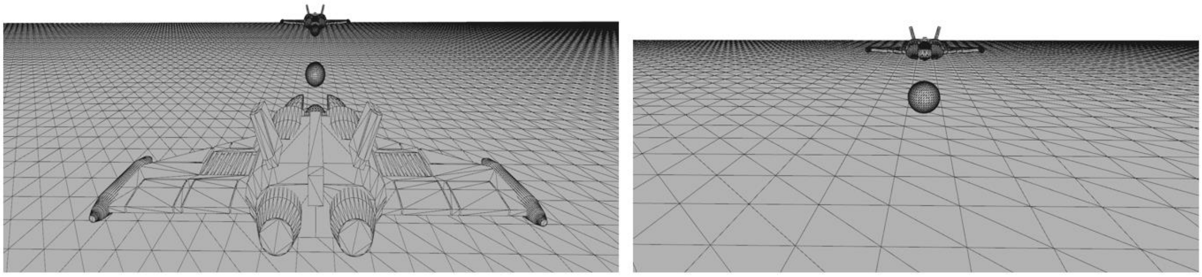

Figure 1. The prototype of the 3D game. (Left: 3rd-person view. Right: 1st-person view)

Requirements

(Important)

- Implement the same game in assignment 3-1 and 3-2, satisfying the requirements described in this document.
- In assignment 3-1, use the fixed pipeline of OpenGL, i.e., use traditional OpenGL APIs.
- In assignment 3-2, you cannot use the fixed pipeline of OpenGL. For example, glBegin() or glEnd() cannot be used. Design your own rendering pipeline by implementing shaders using GLSL.
- In assignment 3-2, you can use GLM (a header only library). (https://glm.g-truc.net/0.9.9/index.html)
- You have to submit a report for each of assignment 3-1 and 3-2.

- **Characters**
  - Find or make 3D spaceship models to represent player and enemies.
    - (Keep in mind) in assignment 4, you'll implement texture mapping. Therefore, you'll want to use UV-mapped meshes in this assignment. Check if the meshes are UV-mapped before using them in your project.
    - You might just want to use a bunch of cuboids or capsules to design a character. You can create such simple geometric primitives manually with code, or design them in other tools like Blender

(https://www.blender.org/) and import/export them by parts.

        - In this assignment, you don't need to consider hierarchical structure of the character and the animation loop.
- **Background**
    - The ground is expressed in a grid. (Refer to Figure 1)
    - The celestial bodies (star, planet, satellite) in the planetary system are expressed as spheres.
        - The hierarchical structure and animation loop of the planetary system are applied in the same way as in assignment 2.
- **System**
    - Bullets and items are represented as simple 3D objects.
    - A certain area of the ground is set as the boundary of the game. (Refer to fig. 2)

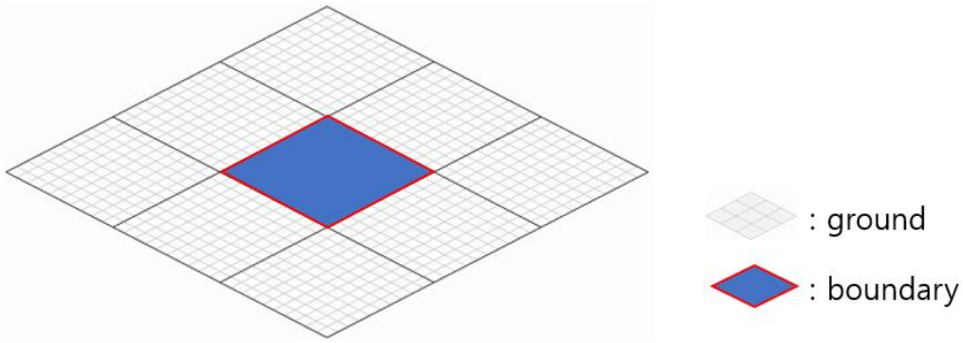

Figure 2. Example of boundary setting

- **Viewing**
    - Implement two viewing modes. (Refer to fig. 1)
        - 3rd-person view: the camera is somewhere behind the player (Default)
        - 1st-person view: the camera is in front of the player
    - You can change viewing mode by pressing 'v'.
- **Rendering**
    - Implement two rendering modes.
        - Mode 1: wire-frame without hidden line removal (Default)
        - Mode 2: wire-frame with hidden line removal
    - You can change rendering mode by pressing 'r'.

Misc.

- You can either implement yourself or use library when loading mesh models.
- Other unspecified requirements are the same as assignment 2.
- If you want to implement other views, provide the views in the requirement and add additional modes for custom views.
- You are free to implement any details that are not specified and record them in the report.
- If you implement additional functions unspecified in this document, you can get extra points, up to 10% of the full score.
    - This should be stated in the report.