IMD3002: 3D Computer Graphics

Term Project: Part B - Game Engine Prototype Report

Submitted to: Dr. Chris Joslin

Submitted By: Ayan Ray 100619458

Submitted on: March 3rd , 2008

1.0 Introduction

This report was written to provide an update on the progress of the game that will be produced for the successful completion of the term project of IMD 3003: 3D Computer Graphics. As mentioned prior to the writing of this report, I will be cooperating and developing with Victor Rucareanu, 100650800, to complete this project. The topic of this game engine will be creating a mission-based airplane game written in Open GL. This report is broken down into the objectives, specifications completed, specifications to be completed, and other challenges.

2.0 Objectives

The main objective of this term project is to create a scalable, usable, and effective game engine that will include one playable game. For this term project, Victor Rucareanu and I have chosen to create a flight simulator action game with primitive AI and mission based game objectives, which are all scalable within the game engine, as mentioned in Part A of the term project.

2.1 Game Engine Objectives

The objectives of the game engine are (1) to create a scalable, reusable game engine that can be used to play similar flight simulator games if not any type of game, (2) use OpenGL to work with graphics cards to render polygons in cell-shaded style rendering, (3) use common game engine creation techniques in industry (based on future research on specific requirements such as shading) so that it allows us, if we wish, to transition into working for a game company.

2.2 Game Play Objectives

The main objectives for the actual playable game that will mold the game engine is to create a mission-based airplane game that places the user in different scenes with different objectives and they must complete those objectives before moving to the next level. The user will be required to take-off and land the plane at the start and end of each mission. Examples for the missions are to escort a plane to a certain destination or shoot down all enemy planes.

3.0 Specifications Completed

The following specifications have been completed or have been partially completed:

i. ***Scalable, modular infrastructure*** *dependent on classes and objects to allow virtually infinite missions.*

This has been developed with the formation of xEngine. xEngine is our 3D Game Engine that has subdivided the 3D engine into the following components: Cameras, Collision Detection, Primitives, Scenes, Types (Vectors etc.), and Utilities. This specification is 90% complete because it was required to lay the framework for the rest of the specifications. However, there are possibly further organization structures that could be developed to make this engine more scalable.

ii. ***Macro-collision detection****, which subdivides the world into sections, and checks these sections continuously for what objects are inside and what objects are hitting each other in an effort to speed up performance.*

This has been developed separately from xEngine and implemented in mode 4 by pressing key 4. In addition, pressing "e" will create new targets that if hit enough times with the aim, will eventually fall from the sky. We have implemented it into the engine by the means of an Octree that breaks the world into separate areas and checks collision detection on the objects within that area. This specification is 90% complete, as the airplane only now requires to be added to the Octree display list for collision detection objects. To complete this specification, we will be adding the collision detection methods demonstrated to an Octree collision list and further organizing the methods into the engine, making it not dependent on the actual game.

iii. ***Airplane controls*** *that changes speed, pitch, yaw, and roll of the airplane. The overall airplane will have direction, speed, and damage to determine the actual world speed of the airplane.*

This has been created as part of the xEngine. It is 100% complete with the creation of the Object3D class, which allows for all movement and rotations of any object in 3D space.

4.0 Specifications to be Completed

To meet the full objectives of the game engine, the following specifications have yet to be completed but are still planned to be completed:

iv. ***Advanced Micro-Collision detection*** *on the airplanes to decide how much damage projectiles will do to airplanes.*

This has not been developed yet. However, we are planning to create hidden 3D spheres around the plane to test for micro collision of objects. If the object hits a certain sphere, it will reduce the damage relative to the sphere it has hit. Since we have used spheres already for the macro-collision, this step should not be as difficult.

v. ***Mission-based objectives engine*** *that decides when a mission is considered "completed" and when it is appropriate to restart or continue to the next level*

This has not been created yet as basic game play must be completed. We have decided that it is also dependent on the AI engine for the planes. The mission engine will be provided with two levels. After completing one level, the next level will automatically load. The levels that will be designed are (1) shoot down all the planes, and (2) protect a plane from other planes until it reaches the boundary of the area.

vi. ***Basic AI*** *for enemy and friendly planes*

For the purposes of this report, simple AI has been setup and only one rule has been created. This task is virtually endless so that is why it is noted as "Basic". We believe we need to create a few more simple rules that all planes need to abide by to complete the Basic AI section. These rules are staying in the boundary, moving towards a target position, and firing at your airplane.

vii. *Use of a* ***Rendering engine****, which renders the scene in the "cell-shaded" style and will be changeable.*

The rendering engine for cell-shaded style has not been created but the rendering ability has been componentized within the Scene.renderCamera() function. This allows us to test different rendering of objects and will implement the cell-shaded style render as we have examples of this technique.

5.0 Other Challenges

This section talks about other challenges that have been encountered so far. Currently, working as a team has proved to be more challenging than working individually. We have to manage our time so we can work together, manage our work so that the different sections can merge seamlessly, and teach each other what we have learnt from the different specifications. This creates project management challenges and is beneficial work experience that complements the skills we have learnt in OpenGL.

6.0 Conclusion

In conclusion, we believe we have completed more than 30% of the required work at this milestone. Considering that we have more specifications, it should be an appropriate amount of work for two people. We have still a few key specifications to complete, but with the framework being laid down, it will be easier on integrating the different specifications.