Kastell

Computer Graphics I (CSA2207) Assignment

Sandro Spina

October 29th 2012

This is the assignment description of unit CSA2207, Computer Graphics I, for the year 2012/2013. Your work on this assignment may contribute up to 40% of the total mark for this unit and should be carried out individually. Under no circumstances should code be shared. Assignments should be submitted by hand on optical medium; student might be asked to describe and demonstrate the implementation submitted with the documentation. Please remember that plagiarism will not be tolerated; the final submission must be entirely your work.

Deliverables

You will submit your project source code, executables and a PDF file containing the project documentation on optical medium. Ideally binaries should be made to run straight from CD/DVD-ROM without the need for any installation unless specified in the accompanying document, in which case a detailed installation guide must also be provided. **DO NOT** submit with the assignment a print out of your source code. You should only print the report in which you describe what you've done.

Background / Description

For this assignment you are asked to create, render and navigate a 3D virtual world consisting of an environment in which elements of a tower defence game are implemented. You are allowed to create (or use any that are freely available given this is not an arts course) 3D geometry to populate your world. All your geometry should be contained within a spatial data structure, or scene graph, which is used to optimise the rendering of your world. You are allowed (or rather expected) to be creative and come up with different ideas in the design of this game.

The application may be developed in C#, Java, C or C++. You are encouraged to use mature game engines, such as Ogre3D and the jMonkeyEngine in this assignment.

Task Breakdown and Marking

The project is broken down into a number of tasks, each of which outlines a different aspect of the final deliverable. No marking is assigned to the various components simply because each one of them can be dealt with at varying levels of difficulty.

Task 1: Playing Field Creation

The playing field represents the terrain over which your game characters will move. Its complexity can range from a simple checker board to a much more complex terrain, for example one which uses procedural generation techniques. In any case you will need to provide the necessary mechanisms which will enable your characters to move (or perhaps stay still) realistically across the game environment. A game engine, physics engine may be used at the stage. Alternatively you may maintain a data structure which describes the valid locations where you characters can move to.

You should now have a rough idea of the gameplay mechanics you intend implementing.

Task2: The Avatar Camera

A basic camera class needs to be implemented here so that the user is able to move around the playing field. It might be a good idea at this point to establish any constraints on the camera position and orientation. For example, the camera is not allowed to get close or under the terrain.

Task3: Setting up castles, defences.

What is your castle? It could be something which is hidden. It could be something heavily fortified. How are you going to defend it? Placement of turrets, pits, traps, soldiers, etc. For this task you need to implement your NPC game mechanics. Do you have soldiers launching grenades or firing machine guns?

Task4: Collision Detection and Gameplay Execution

How are enemies killed? How is your castle conquered? How does your game evolve over time? When will your turrets fire at the enemies? When will you enemies fire at your turret? How will your enemies approach the castle? These questions should all be answered in this task.

Task5: Lighting your World

Illumination is an important aspect of your world. You now need to make use of shader programs which implement different types of illumination. These light sources should then be integrated into the virtual world (through your scene graph) in order to illuminate its different parts. Clearly the positioning (and movement) of all these light sources in the world is very important. Remember to use an ambient component in your implementation of the lighting equation so that every part of the environment is visible.

Task6: Assignment Write-up

This task is **extremely important** as you are expected to clearly describe how you architected and built your virtual world components, outlining any important design decisions taken. You are encouraged to use diagrams to explain the architecture where necessary; moreover, feel free to use pseudo-code to illustrate the algorithms implemented. If the deliverable has short-comings or bugs or even lacks some of the expected functionality, you must put this down in the write-up.

Final Notes

Remember that the first step in achieving a solid, stable and efficient deliverable is that of sitting down and designing the system before anything else. Identify how everything is supposed to work at a high-level, and then start by singling out the various components of your virtual world. Once you identify the components, start categorising the entities and their relationships, how they interact and work within their sub-systems, and how the sub-systems communicate together within the system. Outline all this design, and make sure you are able to describe to yourself precisely what each component is responsible of and how it would be carrying out its duties. Make sure you do this before doing any programming as this will save you time discarding broken designs or trying to patch badly designed components to make them talk, and will let you concentrate on the actual problem at hand. Good luck, feel free to experiment and try to enjoy this assignment. The effort put into this assignment will be duly noted during marking.