Module: Intro to Graphics & Interactive 3D Modelling
Module Code: IMAT2902
Title: Constructing and Displaying an Airport Model.
Module Tutors: Peter Cooke
Submission Deadlines:

Deadline for on-line submission of a report, viewer and render program: 23.59 Friday 22nd April 2016

Marks

This is an individual assignment and is worth 60% of the total coursework marks and 30% of the marks for the module as a whole for the IMAT2902 Intro to Graphics & Interactive 3D Modelling module and will be marked out of 100. See the Marking Scheme at end of this document**.**

Extensions

An extension will only be given at your lecturer's discretion. You will also be expected to provide supporting evidence (medical certificate, doctor's letter etc.) to prove you need extra time to complete the assignment. The request for an extension should be made in writing before the deadline.

If you have been identified as requiring additional time due to specific learning requirements it is in your interests to discuss this with your tutor as soon as possible once the assignment has been issued.

Deadline Penalties

Work submitted up to 14 days late past the deadline – capped at 40%
Work submitted after 14 days late past the deadline – Zero marks

Plagiarism

All cases of plagiarism will be dealt with under University Regulations.

Constructing and displaying an Airport model.

For this assignment you need to create a C++, SFML and Open GL program to display the scene at a small airport, complete with at least two different aircraft, along with some airport buildings including a control tower and the airport's ground map, showing runway(s) and some other details.

The scene should be created in the form of separate 3d models, loaded as Wavefront OBJ files, along with their associated texture files.

The map of the airport should also take the form of an OBJ model and can be based on the ground map of an existing airport (use Google maps or similar to help you here). A smaller airport is a **much** better option, for example Stoughton Airport. The airport buildings should be built (or sourced) using Maya or a similar 3d modelling tool such as Blender or Sketch-Up.

You do **not** need to create the aircraft models yourself; your lecturer will supply a small number of pre-build models which are free for educational use. In addition you are free to find and import suitable models from online sources to extend the realism of your scene.

Your program should display the full 3d view of the airport with both lighting and texturing, and allow the viewer to navigate around the world, using a suitable and documented combination of keyboard and or mouse controls, with a camera using gluLookAt or similar functionality.

Ideally your program should be built from a number of clearly defined and co-operating Classes and be clearly documented and extensible. For example your code might have a Scene Class with an STL vector of model, or model instance, references. You will be marked on the software design of the cooperating classes you use.

The Airport components could be hardwired in the program code, or preferably, loaded from a text, xml or similar file identifying each models position and orientation.

Marks will be awarded for the detail and appearance of the whole scene and for the quality of those models you have created yourself along with the suitability of any you have sourced. For the buildings and the map model you should choose a suitable balance between adding 3D model geometry detail

and using textures or photo-textures.

You should also submit a 500-word structure report which will contain a critical evaluation of the work you have done. This should focus on the work you have done with OpenGL. In the report you should identify the airport you are modelling, if any and explain how you gathered the necessary information to build your model.

Learning Outcomes

By completing this assignment, you will have learned how to:

1. Import a model created elsewhere in a standard format and render it in your own program
2. Create a 3d model using 3d modelling software.
3. Design a set of classes to load and render multiple models from external sources
4. Apply lighting to a polygon model
5. Apply textures to polygon surfaces
6. Manipulate the viewing transform to support camera movement within the model

Mark Scheme (100)

40 Quality of Open GL Code (Commenting – consistent method/attribute naming etc.) Structure of the program (suitable Classes created and used)

20 Quality of the scene displayed – use of lighting and realistic texture. Suitable scaling for the models. Reasonable accuracy of the airport map if based on an actual airport.

10 Quality of user created models along with the suitability of any externally sourced models.

10 Flexibility of the scene engine (use of external file for model loading and positioning).

20 Quality of report and associated documentation.