COM3503 3D Computer Graphics: Assignment (50%)

Dr Steve Maddock

Deadline: 3pm, Tuesday 7 December

1. Introduction

The assignment will involve using modern OpenGL to render a scene. Scene graphs are required in the modelling process and animation controls are required for hierarchical models.

2. The task

Figure 1 shows a room scene containing three objects, a robot and a window looking out onto a view. The whole scene can be modelled using transformed planes, cubes and spheres. The scene shows five poses of a robot that has entered a museum exhibition room. Pose 1 is on entering the room. Pose 2 is viewing a large mobile phone displayed on a plinth. Pose 3 is viewing a spotlight on the floor as it swings side to side on a stand. Pose 4 is viewing a large egg on a stand. Pose 5 is looking out of the window.

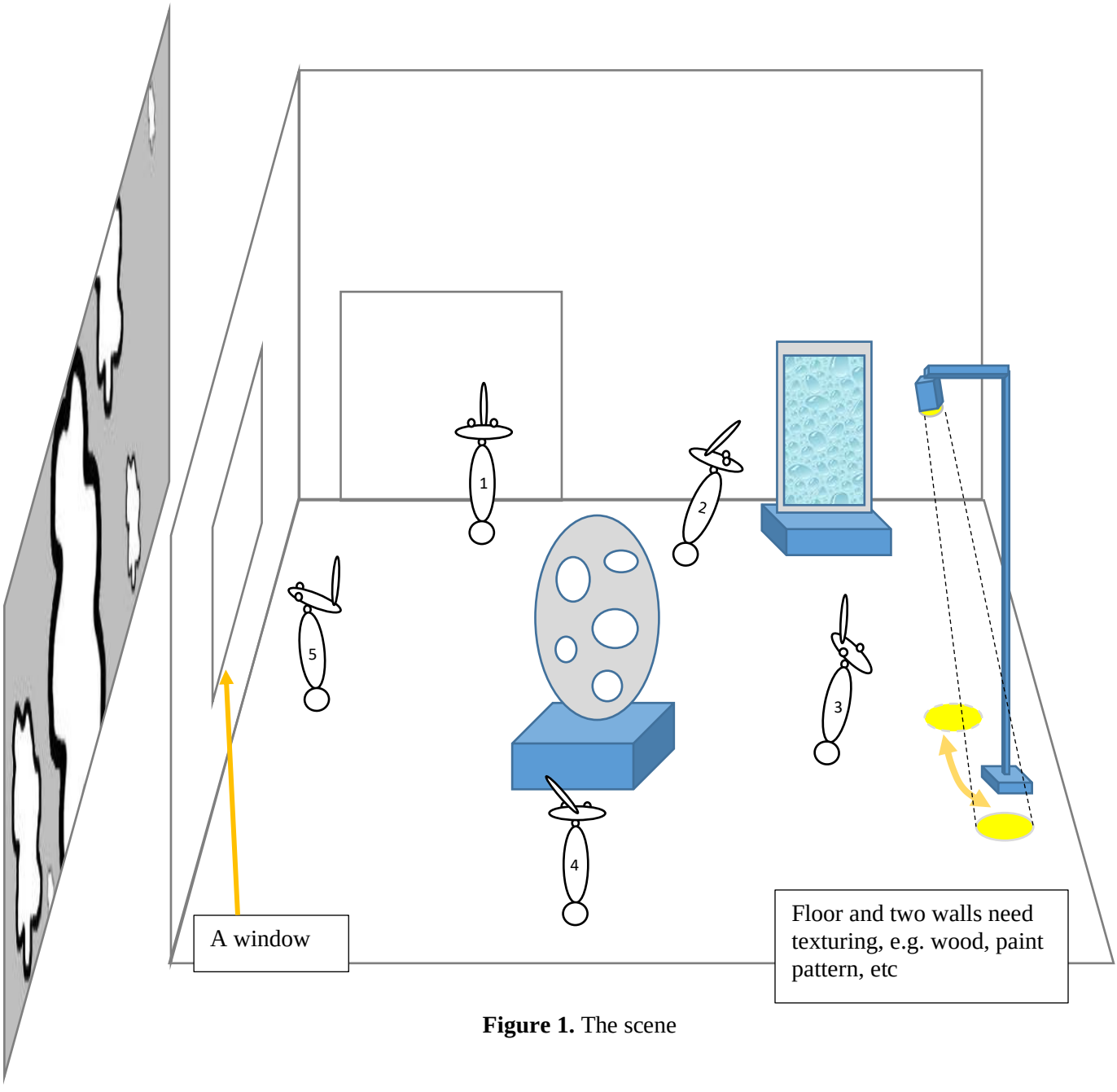

Figure 1. The scene

3. Requirements

You must satisfy all the following requirements.

3.1 The room

- Only two walls and a floor for the room should be modelled.
- The walls and floor should be texture mapped to look like a room in a museum. For example, the floor could be made of wood. The walls should have a paint pattern on them.

3.2 The window

- An outside scene can be seen through the window, for example, this might be a garden scene or a city scene. You could use a picture out of a window in your own accommodation or you could invent a picture.
- Consider how you might do the scene outside:
  - Should it be a texture map pasted onto the wall to look like a fake window and a scene (Figure 2a).
  - Or should there be a hole in the wall and a texture map pasted onto another surface that is a certain distance outside the window? This will mean making the wall from a set of pieces, e.g. eight abutting pieces with the window as a middle area. Figure 2b illustrates this.

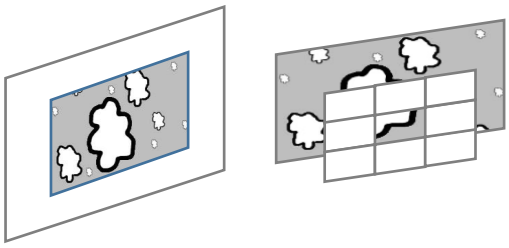

Figure 2. (a) Pasting a texture on the wall; (b) a hole in the wall to see the texture pasted on a plane at a certain distance away.

  - Another option might be a skybox outside the whole room.
- Depending on the approach you choose:
  - How does it look when the camera moves position in the room when looking out of the window? (Is it possible to stand in the room and not see the scene outside through the window?)
  - The scene outside the window should not be the same at different times in the day (e.g. different times of the day and night). How might you achieve this with texturing effects?
- The quality of what you produce for this part of the scene will be part of the marking.

3.3 The robot

- This is a hierarchical model made of different shaped spheres.
  - The hierarchy and associated transformations are more important than the quality of the pieces in the hierarchy. I want you to demonstrate that you understand transformations and a scene graph hierarchy
- The robot should have a head, a neck, a body and a 'foot' (a sphere that can roll). You must design a head that has a minimum of 4 pieces. Figure 3 gives some possible designs. The design of the head is your chance to show a bit of creativity.

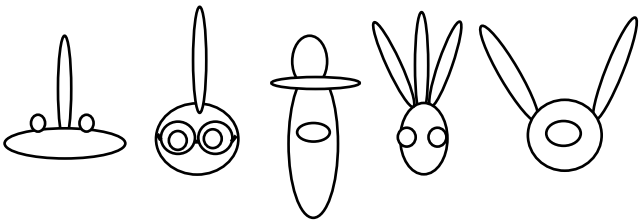

Figure 3. Some possible robot head designs.

- It should be possible for the whole robot to rotate about the foot so the robot can lean forwards or backwards or to one side or the other as part of different poses.
- It should be possible for the head to rotate about the neck so the robot can lean its head forwards or backwards or to one side or the other as part of different poses.
- It should be possible for the features of the head to move around the head to help make the poses distinctive. Depending on your head model, this might be eyes move to look forwards or hat tilts to one side or nose grows or glasses or ears or hairpiece change orientation as part of different poses.
- The robot is required to make 5 different, distinctive poses, one at each location indicated in Figure 1.

3.4 The mobile phone

- This can be modelled using two cubes, one for the phone and one for the stand.
- The mobile phone cube should be texture mapped so that it looks like a mobile phone with a typical screen display.

3.5 The swinging spotlight

- This can be modelled using cubes and a sphere for the spotlight bulb.
- The spotlight mechanism should swing from side to side so that the spotlight is visible on the floor of the room.

3.6 The large egg

- The egg-like object can be made using a cube for the base and a sphere for the egg-like part.
- The egg should be texture mapped with matt and shiny parts. (Hint: diffuse and specular maps.) Figure 1 shows a pattern with shiny spots but a different pattern would also suffice.

3.7 General illumination

- The scene should be illuminated with at least two general world lights which can be positioned anywhere in the world (perhaps with one in the general direction of the sun).
- These general world lights will illuminate all parts of the scene and help visualise the scene during development and testing.
- When you switch off the general light(s), the effects of the swinging spotlight will be much clearer on the floor.
- You do NOT have to do shadows. Do not worry about shadow effects.

3.8 User interface

- A user-controlled camera should be positioned in the scene. Use the camera that was given in one of the exercise sheets – the mouse can be used to change the direction the camera is pointing in and the keys can be used to move about. Do not change the key mappings from the one on the exercise sheet. If you change the key mappings it will make it difficult to mark. It doesn't matter that the camera can see outside the room.
- It should be possible to turn each of the general illumination lights on and off (or dim, i.e. reduce the intensity) from the interface.
- It should be possible to turn the spotlight on and off.
- There should be buttons to show the robot in each of its poses, e.g. pressing a button labelled 'Pose 3' would move the robot to pose 3. This is an immediate movement to the new pose – the robot disappears in one position and reappears in the new position.

3.9 Animation

- This requirement is advanced and you may decide not to do this part, although you would not be able to get full marks.
- The robot should animate between the poses rather than immediately move from one pose to another pose.
- It is perfectly acceptable to animate the Euler angles to achieve movement of the hierarchy. Do not consider using quaternions, as this is beyond the requirements for this assignment.

4. Deliverables

- You should submit a zip file containing a copy of your program code (and any other necessary resources, e.g. image files for the textures and a readme.txt file that describes everything) via Blackboard – this can be done via the link to the assignment handout. You should submit whatever you have done, even if you have not completed all the requirements – for example, you might have produced a model of the room but not done the robot poses. If you submit nothing, you cannot receive any marks.
- **The program MUST compile and run from the command window on a Windows PC or the terminal window on a Mac**. You should assume that the jogl environment (and paths) has already been set up, so you do not have to include this as part of what you hand in. I won't install 'YetAnotherIDE' to make your program work; I want to run the program (and, if necessary, check the compilation) from a command (or terminal) window.
- You must include appropriate comments in your program to identify that you wrote the code, e.g. /* I declare that this code is my own work */ /* Author <insert your name here> <insert your email address here> */
- You can make use of all the code that I have given you on exercise sheets. However, use your comments to state which bits you used or which bits are new.
- The body of the Blackboard submission message should state that the work you have handed in is your own.
- The name of the main class in your program should be **Museum**. That way it is easy for me to run the program. (Last year, I wasted time for some handins trying to work out which was the main class to run.) It would be useful to include a batch/script file to automatically compile and/or run the program.
- *Optional*: You might like to make a short video of your animation. If you do so, **DO NOT** include this in the handin as it will be too big for Blackboard to handle – we tried using Blackboard for this in the past and it crashed the system!! Instead, put the animation on youtube or your personal website and give the URL of the animation in a readme.txt file. Indeed, if you are thinking of a career in the graphics industry, then you should be adding such animation pieces to your personal website (your digital portfolio) to show off what you are capable of.

5. Marking

I will check that the program meets the requirements listed above. To make sure you get some marks, the program **must** compile and do some part of the work requested even if it is not complete. Your program code will be run and exercised thoroughly.

Marks will be available for:

- The quality of the programming (20%)
- Satisfying the requirements (80%)

In assessing the quality of your program code, four aspects will be considered:

- (5 marks) General style: layout; neat, organised code; comments; use of constants and variables; methods not over long;
- (5 marks) Program and data structures for the models: use of separate methods and classes, e.g. separate classes for things like the robot; (The online tutorial does not make use of classes for the more complex examples, partly on purpose. I want you to show that you can structure things appropriately. You must consider the use of classes for the assignment – this demonstrates your programming ability.)
- (5 marks) Neat and tidy coding for dealing with transformations in the scene; How will you organise the construction and use of the scene graphs? It shouldn't all be in one method!!
- (5 marks) Animation/posing control: how tidy and flexible is the coding?

In considering the requirements, four aspects will be considered (including the quality of the work):

- (20 marks) Modelling the scene: the robot must be a hierarchical model. (Consider drawing scene graphs for the scene before starting to program.)
- (20 marks) Texturing: Use of texture mapping in the scene, e.g. basic texture mapping, use of diffuse and specular textures, seams between textures and any extra texturing effects such as the changing window view.
- (15 marks) Lighting and interface controls: lights should behave correctly such that their effect is seen on the scene. Necessary interface controls, as described in the above specification, should also be included.
- (25 marks) Robot pose control (10) and animation (15). Are all the poses produced and distinct? Is the animation plausible and smooth?

6. Unfair means

- The Department's student handbooks (UG and PGT, see below) give detailed information on the topic of unfair means and what happens if unfair means is used.
- Some students in previous years have placed solutions of their assignments on their personal websites. Be careful you are not attracted to these, as using any of their code would be regarded as use of unfair means – this has happened in previous years and students have failed the module as a result of doing this.

7. Late submission

- Standard Department rules will be applied if the work is handed in late – see UG and PGT handbooks below.

Links to handbooks

UG:
https://sites.google.com/sheffield.ac.uk/comughandbook/general-information/assessment

PGT:
https://sites.google.com/sheffield.ac.uk/compgtstudenthandbook/menu/assessment