

COM6503: 3D Computer Graphics: Assignment

Dr Steve Maddock

Deadline: 3pm, Wednesday 7 December

1. Introduction

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

Learning outcomes: After completing this assignment, you will be able to:

- Use data structures and mathematics in representing and manipulating 3D objects
- Produce interactive software that makes use of a graphics API

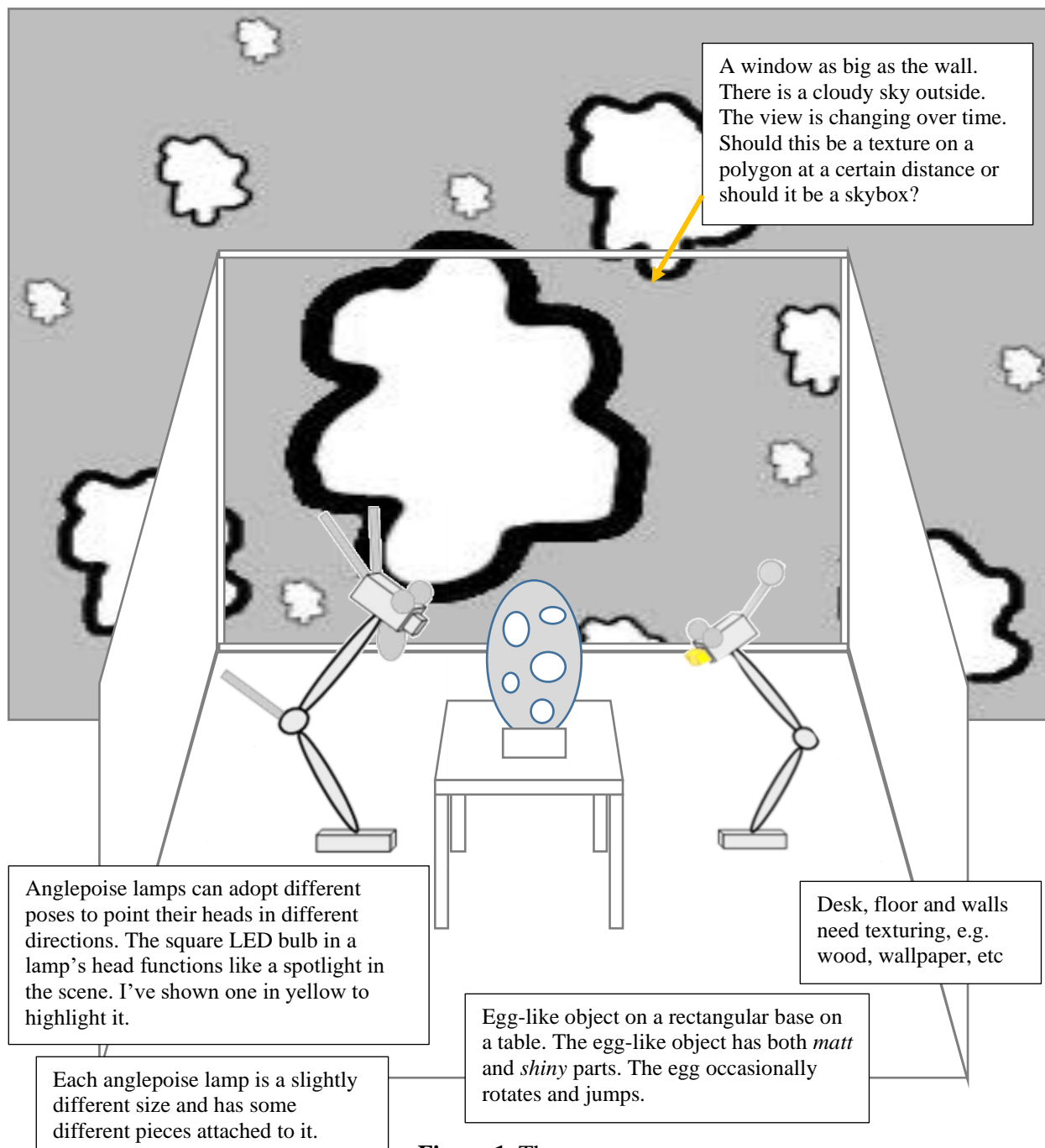


Figure 1. The scene

2. The scene to create

A company is creating an advert for a range of new angle-poise lamps that look like alien animals. Your task is to create a scene for this.

Figure 1 shows a room scene containing a table, an egg-like object, two anglepoise lamps and a window looking out onto a view. The whole scene can be modelled using transformed planes, cubes and spheres.

The lamps are inspecting the egg. The base of each lamp is fixed, but each lamp can each articulate its joints so its head points in different directions. The egg-like object occasionally jumps and twists (rotates), as though there is something inside trying to get out.

3. Requirements

You must satisfy all the following requirements.

3.1 The room

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

3.2 The window

- The back wall in the room is a large window.
- An outside scene can be seen through the window, for example, this might be a garden scene or a city scene or some advertising text (e.g. “new alien animal anglepoise lamps”) that scrolls past the window.
- Consider how you might do this:
 - Should the scene be a texture map pasted onto the wall to look like a fake window and a scene?
 - Or should there be no wall and instead a texture map pasted onto another surface that is a certain distance outside the window (as illustrated in Figure 1)? Should there be a small frame for the window?
 - Or should a skybox be used that is 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 change whilst the program is running, e.g. the clouds might move.
- The quality of what you produce for this part of the scene will be part of the marking.

3.4 An anglepoise lamp

- The hierarchical model of a basic angle-poise lamp should be made up of five parts: a base, a lower arm,

an upper arm, and a head, which contains a protruding lightbulb (a nose) in the shape of a cube (the details of the lighting technology used are not important). There are also some additional pieces – see below.

- The lamp base is in a fixed position in the room. (If you wish, you can animate the base to move a little bit, e.g. slide around, but this is not required for the assignment.)
- Three parts of the lamp can articulate as illustrated in Figure 2. The lower arm can rotate about the connection point with the base in two directions (in a circular way, and the lower arm can also articulate with respect to the base like an elbow joint), the upper arm can rotate about the lower arm (again, like an elbow joint) and the head can rotate about the

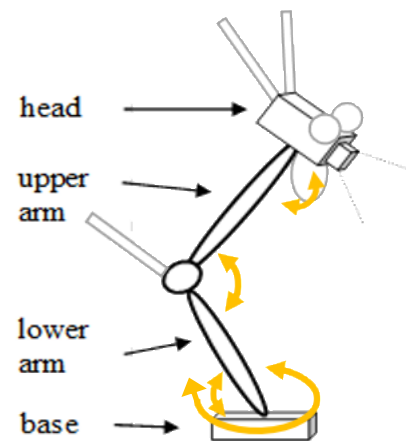


Figure 2. A model of an angle-poise lamp that looks like a strange alien animal.

- upper arm (again like an elbow joint).
- The lightbulb shines in the same direction that the head is pointing in.
- There should be some decorative pieces that make the lamp look unusual or like an animal (see Figure 2). Do not just copy the decorative pieces used in Figure 2; invent your own pieces. For example, you might make the ears more interesting, or add pointy hair, or horns, or a larger nose, or a different tail.
- Use simple objects for the individual parts, i.e. scaled spheres or cubes. For example, in Figure 2, the head is made from a combination of a cube for the lamp head and a cube for the bulb, with some other cubes and spheres for the decorations. Cubes and spheres are the only pieces needed to model the lamp. 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 pieces of the lamp must be texture-mapped. For example, you might decide to make the pieces spotty

to look like a particular animal. As another idea, you might texture map a picture of feet onto the base.

- The head of the lamp must include an object (e.g. a nose bulb), modelled as a cube or sphere, which mimics the effects of a spotlight. The object gives the position of the spotlight and the orientation of the head gives the direction the spotlight is pointing in. The spotlight will thus illuminate the scene in the direction the lamp head is pointing in. (The dotted lines in Figure 2 are just to illustrate the direction the spotlight is pointing in.) There must be an option in the interface to turn the spotlight (and look of associated object) on and off. (You are responsible for working out how to implement a spotlight effect – read the relevant section in Joey’s online tutorial.)

3.6 The large egg-like object

- 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.
- 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 spotlights on the anglepoise lamps will be much clearer.
- 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 the tutorial material – 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 ones in the tutorial. If you change the key mappings, it will make it difficult to mark. It doesn’t matter that the camera can move and see outside the room.
- It should be possible to turn each of the general lights on and off (or dim, i.e. reduce the intensity) from the interface.
- It should be possible to turn each spotlight (lamp bulb) on and off.
- There should be buttons to move each anglepoise lamp into three different poses, so six buttons to do this (three for each lamp). One pose for each lamp is its default, looking at the egg. The other two poses

for a lamp could be anything, e.g. looking at the other lamp or looking under the table or looking towards the window. However, these poses should be obviously visually different for each lamp.

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.
- Each lamp should smoothly animate between its poses rather than immediately transition 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 and a single lamp, but not done two lamps or the different lamp poses or animation. 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 in the tutorial material. However, use your comments to state which bits are new.
- The body of the Blackboard submission message should state that the work you have handed in is your own (except for the bits I supplied).
- The name of the main class in your program should be **Hatch**. That way it is easy for me to run the program. (In previous years, I have 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 will be run and exercised thoroughly.

Marks will be available for:

- The quality of the programming (10%)
- Satisfying the requirements (90%)

In assessing the quality of your program code, the following aspects will be considered: general neatness and organisation; use of data structures, classes and functions; organised scene graphs; animation flexibility.

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

- (24 marks) Modelling the scene: each anglepoise lamp must be a hierarchical model. (Consider drawing scene graphs before starting to program.)
- (24 marks) Texturing: Use of texture mapping in the scene, e.g. basic texture mapping, use of diffuse and specular textures, extra texturing effects such as the changing window view.
- (18 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.
- (24 marks) Lamp pose control and animation. 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 world-readable websites – where possible, they have

been asked to take these down. 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>

PGT:

<https://sites.google.com/sheffield.ac.uk/compgtstudenthandbook>