COM3503: 3D Computer Graphics: Assignment (*40%*)

Dr. Steve Maddock

Deadline: 3pm, Wednesday 5 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 a hierarchical model.

2. The task

Figure 1 shows a model of a large, human-sized, angle-poise lamp, which is decorated to look like an animal. This is made up of 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). The lower arm can rotate about the base, the upper arm can rotate about the lower arm and the head can rotate about the upper arm. The lightbulb shines in the same direction that the head is pointing in, as indicated by the dotted lines in Figure 1. Imagine the lamp is standing on a tabletop in a room looking out of a window. The lamp can also jump around the tabletop. A company is using this lamp in an advertising campaign in which they hope to sell more interesting angle-poise lamps. Your challenge is to create this scenario.

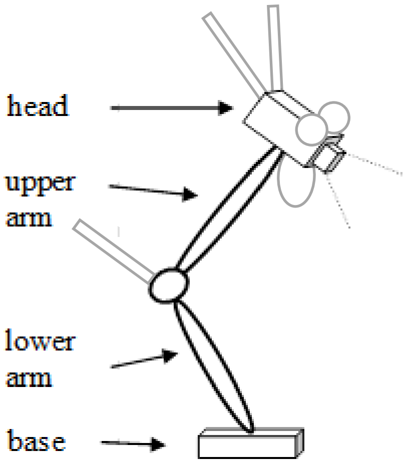

Figure 1. A model of an angle-poise lamp that looks like a strange alien animal

You must satisfy all the following requirements:

- Produce a hierarchical model of an angle-poise lamp, made up of the four basic parts (and the lightbulb) listed above and some decorative pieces that make it look unusual. Do not just copy the decorative pieces used in Figure 1; invent your own pieces. For example, you might add a hat or make the ears more interesting, or add some hair, or horns, or a larger nose, or a different tail or a different base. Use simple objects for the individual parts, i.e. scaled spheres or cubes. For example, in Figure 1, 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 of the lamp look like they are made of wood or brick or whatever, perhaps with shiny reflective markings.
- 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. There must be an option in the interface to turn the spotlight (and 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.)
- The lamp should be placed on a tabletop, with the table under a window in a wall (Figure 2).
- Only one wall and a floor for the room should be modelled. The wall and floor should be texture mapped to look like a room in a house. For example, the floor could be made of wood. The wall may have wallpaper on it.
- The table can be modelled as one scaled flat cube on top of 4 legs made out of cubes. The table should be texture-mapped, for example, to look like wood.
- There should be three objects on the table top (not including the lamp) that should each be textured. Example objects might include: a mobile phone (cube with relevant textures), a picture frame (cube with relevant textures), a paperweight (sphere with relevant textures), a desk tidy (cube with relevant textures and pens sticking out of the top), a cactus plant (spheres with relevant textures), etc. The objects should be made out of simple pieces, with textures used to make them more interesting. There will be some reward for inventiveness.
- At least one of the objects on the tabletop should demonstrate a combination of specular and diffuse texture maps.
- An outside scene can be seen through the window – this might be a garden scene or a city scene. Consider

how you might do the scene outside. Should it be a texture map pasted onto the window, or should it be a texture map pasted onto another surface that is a certain distance outside the window? How does each look when the camera moves position? You need to use the right OpenGL settings to make sure the textures display well. For extra marks, consider switching between different window textures at certain points in time to model a change outside the room, e.g. daytime versus night-time or adding a moving texture to the window view. (Note: You will have to work out how to model the wall with a hole in it for the window, and decide how big the hole for the window should be.)

- The scene should be illuminated with at least two general world lights which can be positioned anywhere in the world. It should be possible to turn the lights on and off (or dim, i.e. reduce the intensity) from the interface. 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 spotlight will be much clearer on the table and the nearby wall.
- A user-controlled camera should be positioned in the room. 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 for me to mark.
- The interface should have a button (labelled 'Random Pose') to make the lamp parts adopt random angles to each other to make a pose. The angles between the parts of the lamp should be within certain ranges so that the result is plausible, i.e. just as you elbow cannot bend 'backwards', the lower and upper arm of the lamp should not be able to bend backwards with respect to each other. Also, the main lamp parts shouldn't intersect each other or the table – don't worry about the decorative parts of the lamp. Thus the lower arm might rotate about the base and the upper arm might rotate about the lower arm and the head might rotate about the upper arm. Also, the tail might rotate about the attachment point between the lower and upper arms. The lamp should animate between the poses rather than immediately assume the new pose. A reset button can be used to reposition the lamp to some neutral pose.
- The interface should have a button (labelled 'Jump') to make the lamp jump from its current position to a new random position nearby to its current position – the lamp must remain on the table top. The lamp must jump in a plausible way, e.g. be smooth and bend/compress more for bigger jumps. How should the decorations behave when the lamp jumps? You are not being asked to physically model a jump, but to provide a simple procedure with parameters that can be used to give a variety of jumps. For example, a curve could be used to control the height of the base of the lamp off the tabletop and the angles between the parts of the lamp could be animated using sine waves to control the size of the angle to make it look like the lamp is compressing and preparing before a jump. Also, you should consider the speed of the animation. 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. Also, do not worry about the decoration parts of the lamp intersecting with the wall, table or with other parts of the lamp.

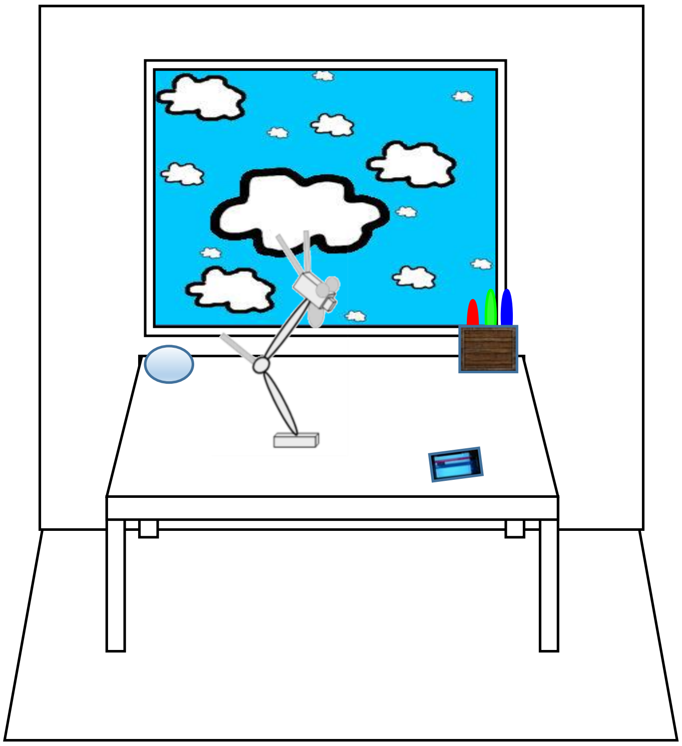

Figure 2: The lamp positioned on the tabletop – the table could be much bigger to give the lamp room to jump

- Each time the jump button is pressed, the lamp will jump from its current position to a new position. The new position will then become the current position. Over time the lamp will move around the table top.
- You do NOT have to do shadows. Do not worry about shadow effects.

3. 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 MOLE – 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 but not done the animation. If you submit nothing, you cannot receive any marks. ***The program MUST compile and run from the command line*** on a standard Windows PC (such as the ones in the Diamond). 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 the command line on my Windows PC.

- 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, state that you have used it as part of your comments, and identify which bits you used.
- The body of the MOLE 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 **Anilamp**. 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 an Anilamp.bat 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 MOLE to handle – we tried using MOLE 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.

4. 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 the room, table and lamp;
- (5 marks) Neat and tidy coding for dealing with transformations in the scene;
- (5 marks) Animation/posing control: how tidy and flexible is the coding?

In considering the requirements, five aspects will be considered:

- (25 marks) Modelling the lamp and other parts of the scene: the lamp must be a hierarchical model. (Consider drawing a scene graph for the lamp model and another for the full scene before starting to program.)
- (20 marks) Texturing: lamp, room (wall and floor), table, objects on table and window. The *quality* of the texturing will be considered, e.g. use of diffuse and specular textures, seams between textures and any extra texturing effects such as the changing window texture.
- (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.
- (20 marks) Jumping lamp animation and pose control. Is the animation smooth? Does it look plausible? The *quality* of the animation will be considered. Are the random poses plausible?

5. Unfair means

- The Department's student handbooks (UG[1] and PGT[2]) give detailed information on the topic of unfair means and what happens if unfair means is used.

6. Late handin

- Standard Department rules will be applied if the work is handed in late: UG[3] and PGT[4].

[1] https://sites.google.com/sheffield.ac.uk/comughandbook-201819/general-information/assessment/unfair-means

[2] https://sites.google.com/sheffield.ac.uk/pgtstudenthandbook2018-19/menu/assessment/unfair-means

[3] https://sites.google.com/sheffield.ac.uk/comughandbook-201819/general-information/assessment/late-submission

[4] https://sites.google.com/sheffield.ac.uk/pgtstudenthandbook2018-19/menu/assessment/late-submission