COM3503 3D Computer Graphics
Assignment (50%), Summer 2025

Dr Steve Maddock

Deadline: Friday 1st August 2025, 15:00

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. You MUST use the Programmable Pipeline in your solution, i.e. use vertex and fragment shaders.

2. 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.

3. The task

Figure 1 shows a room scene containing a table and a window looking out onto a view. There is a noticeboard on one wall. On the table is an unusual angle-poise lamp, a laptop, and a helicopter-like object.

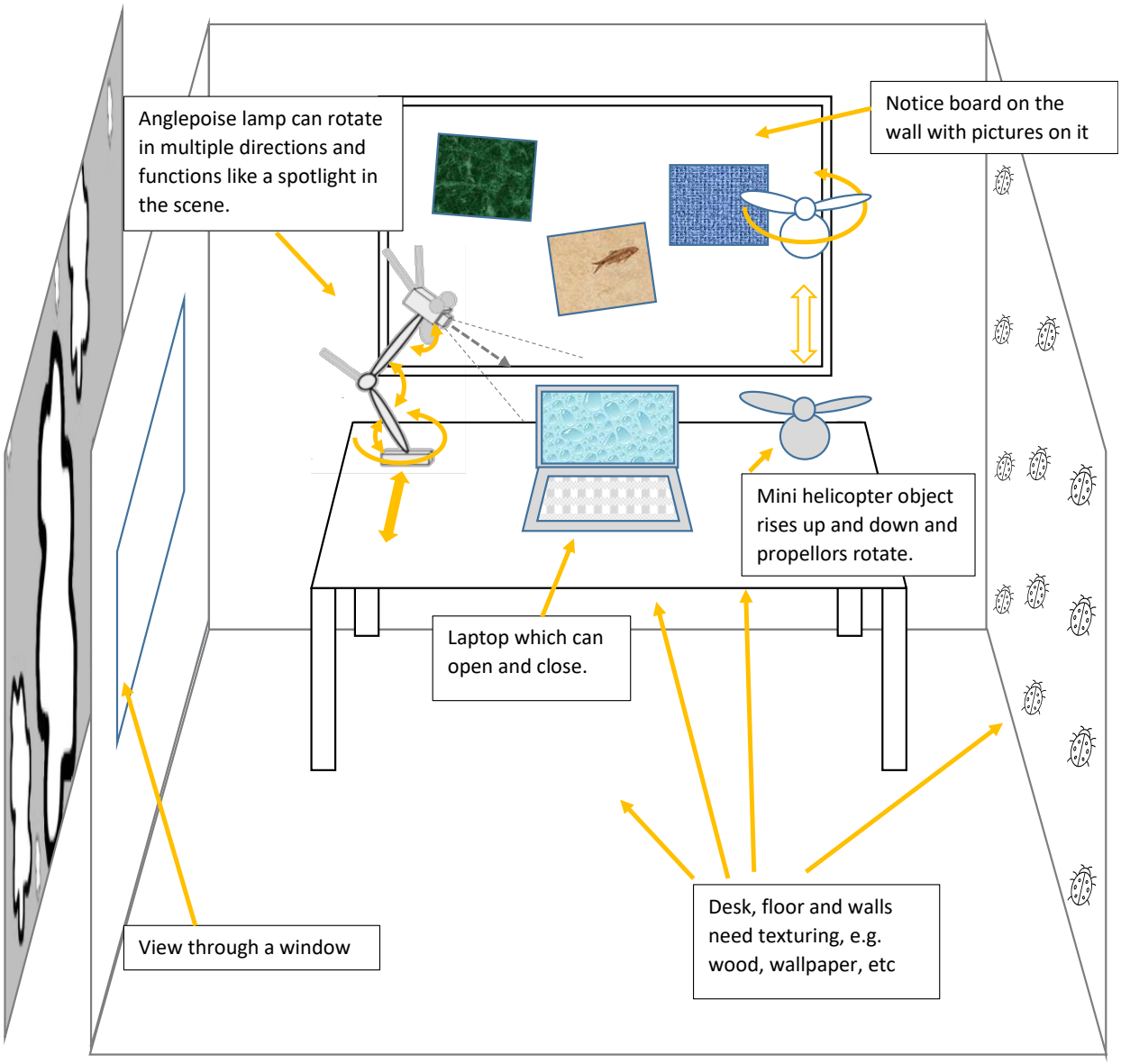

Figure 1. The scene

You must satisfy all the following requirements.

3.1 The room

- Three walls and a floor for the room should be modelled.
- The floor should be textured. You choose a texture.
- The right wall has a repeating texture added to it. Here, a picture of a bug is used (which repeats four times in one direction and three times in the other). You should make your own choice of texture. It must *repeat* over the wall, i.e. the wall is not a single texture image containing multiple pictures.
- The back wall has a notice board with three posters on it. Use texture maps. Show some creativity by including a specular map for one of the posters.
- There is no ceiling

3.2 The window

- The left wall in the room contains 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 a countryside scene, etc.
- 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 a texture map be pasted onto another surface that is a certain distance outside the window (as shown in Figure 1)?
  - Or should a box of textures be added outside the window so a texture can be seen at all angles when looking out of the window?
  - Or should a skybox be used that is outside the whole room?
  - Will the window have a frame?
- 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 always see the scene outside through the window?)
  - The scene outside the window should be different at different times in the day. How might you achieve this with texturing effects?

The complexity and quality of what you produce for this part of the scene will be part of the marking.

3.3 The table

- The table can be modelled as one scaled flat cube on top of four legs made out of cubes. The table should be texture-mapped, for example, to look like wood or some other material.

3.4 The anglepoise lamp

- The hierarchical model of the basic angle-poise lamp (see Figure 2) 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.

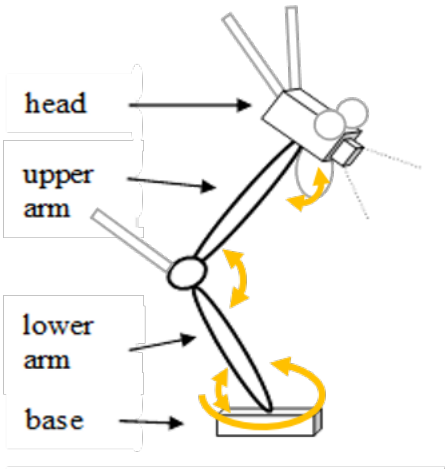

Figure 2. A model of an angle-poise lamp that looks like a strange alien animal

- The base of the lamp can slide forwards and backwards on the table, as indicated by the two-ended arrow in Figure 1. Three other parts of the lamp can articulate as illustrated in Figures 1 and 2. The lower arm can rotate about the connection point with the base in two directions (around the vertical y axis and around the x axis), the upper arm can rotate about the lower arm (like an elbow joint, the x axis) and the head can rotate about the upper arm (in one axis, the x axis). (The x axis could be substituted by the z axis if you prefer.) The lightbulb shines in the same direction that the head is pointing in, as indicated by the dotted lines in Figures 1 and 2.
- A company is using this lamp in an advertising campaign in which they hope to sell more interesting angle-poise lamps. Thus, you need to add 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 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 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.
- 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 the look of the associated cube object) on and off. (You are responsible for working out how to implement a spotlight effect – read the lecture notes and the relevant section in Joey's online tutorial, https://learnopengl.com/.)

3.5 The laptop

- The laptop can be modelled using two scaled cubes with texture maps to show a keyboard and a screen image. It should be possible to open and close the laptop using buttons on the interface (e.g. labelled 'Open' and 'Close').

3.6 The helicopter

- The mini helicopter-like object can be made of a combination of spheres and/or cubes. The propellors must spin as the cube lifts off the desk, rises up and then descends back down to the tabletop. It should do this under menu control, e.g. a button to start and a button to stop.

3.7 General illumination

- The scene should be illuminated with a general world light which can be positioned anywhere in the world.
- This general world light will illuminate all parts of the scene and help visualise the scene during development and testing.
- When you switch off the general light, the effects of the spotlight on the anglepoise lamp will be much clearer.

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 one in the tutorial. 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 the general light on and off (or, more creatively, dim, i.e. reduce the intensity) from the interface.
- It should be possible to turn the lamp's spotlight (lamp bulb) on and off (or, more creatively, dim it, i.e. reduce the intensity).
- Interface controls for opening and closing the laptop and controlling the helicopter should be included.
- 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 your 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. Some of the decorative pieces might move in response to the pose adopted by the lamp – this will demonstrate some creativity.
- There should also be a button (labelled 'Slide') or buttons (labelled 'Slide Forwards' and 'Slide Backwards') to slide the lamp forwards and backwards between two set positions on the table.
- A reset button can be used to reposition the lamp to some neutral pose.

3.9 Poses and animation

- Pose control is described in the previous section.
- The lamp should animate between its poses and positions rather than immediately transition from one pose to another pose. 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.
- Animation between poses is advanced and you may decide not to do this part, although then you would not be able to get full marks.

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 lamp 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 using the standard javac and java commands.
- You must include appropriate comments to identify parts of the code that you wrote, e.g. /* I declare that this code is my own work */ /* Author <insert your name here> <insert your email address here> */. This could be done around major chunks of code and/or at the start of a class to identify the main changes you made.
- 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/chunks/files are new.
- The body of the Blackboard submission message should state that the work you have handed in is your own in addition to the code that was supplied in the tutorial material.

- The name of the main class in your program should be **Main**. That way it is easy for me to compile and run the program. (In previous years, where this has been ignored, I have wasted time for some handins trying to work out which was the main class to use.)
- *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/games 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 you meet the requirements listed above. 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.

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

- (29 marks) Modelling the scene: the anglepoise lamp must be a hierarchical model. Also, consider the laptop, helicopter and table as scene graphs. Consider drawing scene graphs for the full scene before starting to program.
- (28 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.
- (15 marks) Lighting and interface controls: lights should behave correctly such that their effect is seen on the scene. All necessary interface controls, as described in the above specification, should be included.
- (28 marks) Lamp pose and slide control and animation. Laptop open-close animation. Helicopter animation. Is animation smooth? Does it look plausible? Are the random poses for the lamp plausible? The *quality* of your animation will be considered.

6. Unfair means

- The School of Computer Science's student handbooks (UG and PGT) give detailed information on the topic of unfair means and what happens if unfair means is used.
- Generative AI cannot be used for this assignment.

7. Late handin

- Standard School rules will be applied if the work is handed in late: UG and PGT.

Links to handbooks

UG:
https://sites.google.com/sheffield.ac.uk/comughandbook/your-study/assessment

PGT:
https://sites.google.com/sheffield.ac.uk/compgtstudenthandbook/home/your-study/assessment