

COM3503 3D Computer Graphics: Assignment (50%)

Dr Steve Maddock

Deadline: 3pm, Wed 10 December 2025

1. Introduction

The assignment will involve using modern OpenGL to render a scene – this means a programmable pipeline must be used. Scene graphs are required in the modelling process and animation controls are required for hierarchical models.

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

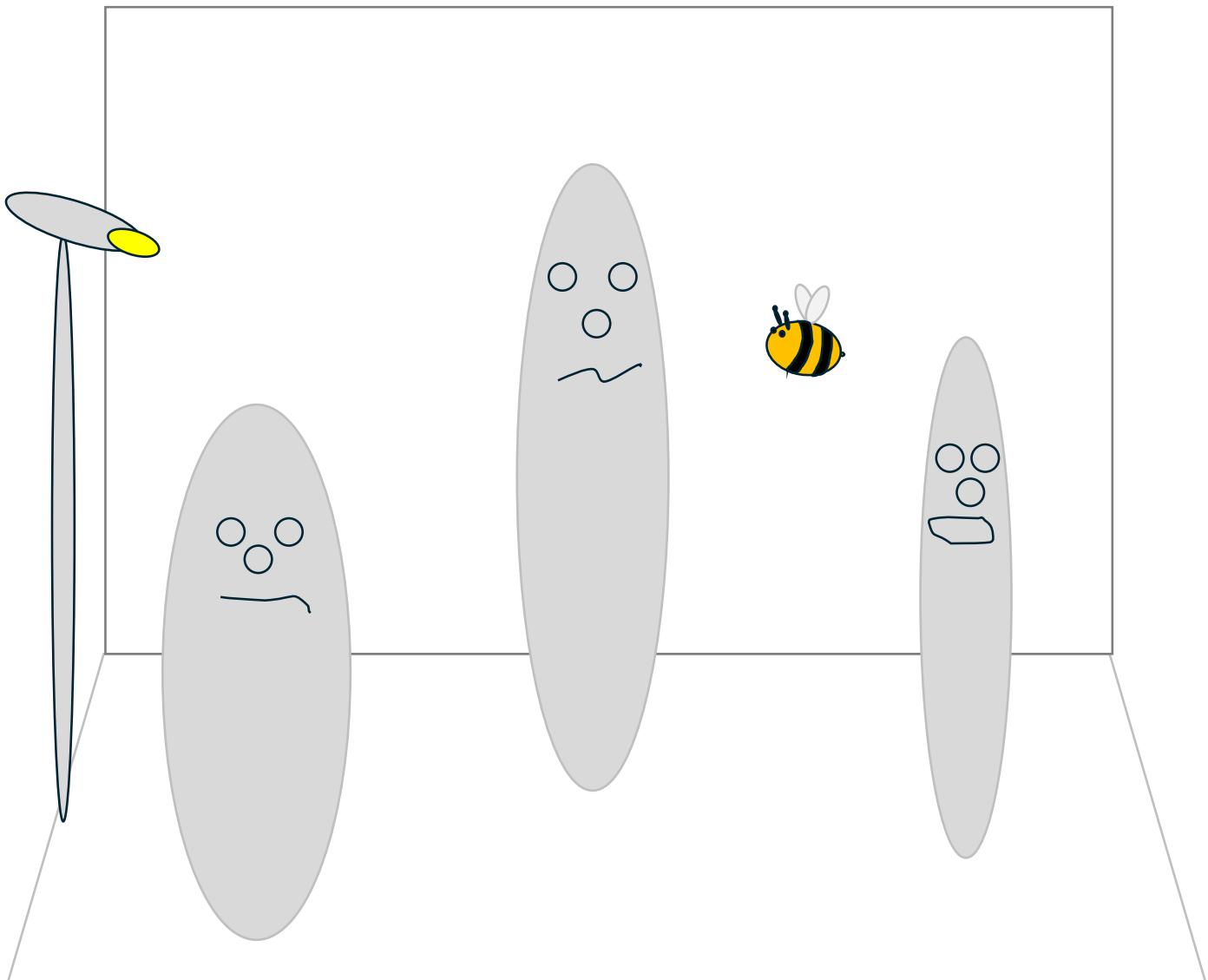


Figure 1. A scene.

3. Requirements

Figure 1 shows a scene which contains three menhir-like statues (each made of a different material and each with a face), a security spotlight that can point its beam of light in any direction, and a bee that flies around the statues. The scene is set in a ‘backdrop’ – see below.

You must use either the Phong or Blinn-Phong approach for the fragment shader. You must use only standard diffuse, specular and emission maps for the texture mapping. Do not consider normal maps or other advanced texturing approaches. Whilst this may limit the appearance of some objects, it will help in controlling the amount of time spent on the practical work.

You must satisfy all the following requirements.

3.1 The backdrop

- The backdrop could be a sculpture park or a square in a city centre or a building quadrangle (like Regent Court), or you could choose another backdrop.
- The backdrop needs to be textured to look like the backdrop you have chosen. For example, if you have chosen a park, should there be grass on the ground and trees in the background? If you have chosen a city square, should there be buildings (i.e. pictures of buildings) around the square and what is the ground made of (possible stones or concrete)?
- Think about the detail of the backdrop. Would layers help improve the look? Would a skybox give a full environment feel? How does the backdrop look when the camera moves?
- The backdrop should change in some way, e.g. different times of the day or different weather or the clouds moving across the sky. You choose.
- The quality of what you produce for this part of the scene will be part of the marking. Some alternatives indicated above are more advanced than others and will attract higher marks. You must choose what to try to implement.

3.2 The statues

- Each statue is a single scaled sphere. In Figure 1, the base of the sphere is on the ground. Instead, you could allow it to penetrate the ground if that looks better – this will not lose marks.

- Each statue must be made of a different material. One must be a stone appearance (e.g. limestone, sandstone, marble), one must be a metal appearance (e.g. brushed steel) and one must be a kind of wood. You must source the texture maps to use for the texture mapping.
- Each statue should also use texture mapping for a unique face. This could be a separate texture map blended with the base material texture map during rendering, or could be mixed into the base material texture map when you make that.
- Each statue’s face should change its expression when the bee flies nearby. This will require a proximity test for each statue. Each statue should have its own reaction to the bee, e.g. one statue might be scared and another might be surprised and another might be angry.
- The faces should be cartoon in appearance.
- Figure 1 shows the statues in a simple triangular arrangement. You could choose a line if you prefer or the statues could face each other or two could stare at one. It is your choice.

3.3 The bee

- This is a hierarchical model made of different shaped spheres. The hierarchy and associated transformations are the important aspect. I want you to demonstrate that you understand transformations and a scene graph hierarchy
- The bee should have a body, two wings, a tail, two eyes and two antenna (where an antenna is made of two spheres). Extra pieces could also be added. The design is your chance to show a bit of creativity.
- The wings should flap when the bee is flying.
- The bee can fly to any part of the scene and is especially interested in the statues.
- Aspects of the bee might change when a statue makes a facial reaction to the bee (e.g. the bee’s eyes might grow larger).
- Exaggerated cartoon effects for the bee’s movements and poses (e.g. cartoons often have exaggerated eye effects) would show extra creativity and attract higher marks.

3.4 The security spotlight

- This can be modelled using spheres. Consider a scene graph.
- The spotlight mechanism should be static until initiated by a button on the interface. This is the switch to switch it on when it gets dark.

- Once switched on, the spotlight will continually sweep the area containing the statues, i.e. move in a circular pattern that covers each statue in turn.

3.5 General illumination

- The scene should be illuminated with at least one general world light which can be positioned anywhere in the world. Two world lights would show extra complexity.
- 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) (using an interface option – see next section), to simulate nighttime, the effects of the spotlight will be much clearer on the rest of the scene.
- You do NOT have to do shadows. Do not worry about shadow effects. (The general world light(s) will illuminate all polygons with a normal pointing towards it (them) and the spotlight will illuminate all objects in the direction it is pointing in which are inside its spotlight area as there are no shadow effects to show light not reaching particular points.)

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 should be possible to turn the general light(s) on and off (or, more creatively, dim, i.e. reduce the intensity) from the interface.
- It should be possible to turn the spotlight movement on and off and also to turn the light on and off (or, more creatively, dim it, i.e. reduce the intensity) from the interface.
- There should be a button to switch between continuous and pose modes in the interface.
- In pose mode, there should be buttons to move the bee to be in specific key positions in front of each statue so the proximity test for the statue can be shown. This would require three buttons, one for each statue. The bee should immediately jump to each new position. This is

an immediate movement to the new position – the bee disappears in one position and reappears in the new position.

- In continuous mode, the bee will just fly between five key positions in the scene. Three of these positions are in front of the statues (to trigger the proximity test) and the other two can be somewhere else in the scene. You choose.

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 bee can be in pose mode, where buttons in the interface are used to jump from position to position immediately or it can be in continuous movement mode, where the bee should animate between the positions in its path in a smooth way. Creativity in the continuous movement will attract higher marks. Consider exaggerated movements as the bee flies, as well as speed of movement.
- 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.
- As part of the handin, you should include a screenshot of the program when it first starts running. This should be named: yourname_screenshot.jpg
- 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 scene but not done the 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 (on my machine), 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 **Buzz**. That way it is easy for me to compile and run the program. (Last year, I 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

The marking for this assignment is aligned with Practical Skills in the Generic Marking Criteria used in the University.

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.

In considering the requirements, four aspects will be considered (with weightings shown):

- (29%) Modelling the scene: There are a range of things to model to complete the scene, e.g. the backdrop, the statues, the bee, the security spotlight. Scene graphs should be used appropriately, e.g. the bee must be a hierarchical model. (Consider drawing scene graphs before starting to program.)
- (28%) Texturing: Use of texture mapping in the scene, e.g. use of diffuse and specular textures, good use of textures (e.g. consideration of seams and distortion), and any extra advanced texturing effects for, say, the changing backdrop of the changing statue faces.
- (15%) 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 be included.
- (28%) Bee pose control and animation. Is the animation plausible and smooth? Are cartoon effects included? Do proximity tests work? Does the spotlight sweep smoothly?

6. Unfair means

- The School's student handbooks (UG and PGT) 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/GitHub – 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 because of doing this.

7. Late submission

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