COMP612 Computer Graphics Programming

Assignment 2

This is an individual assignment. All work you submit must be entirely your own.

Where your implementation idea or inspiration has been taken or adapted from other published sources, or you receive assistance from a classmate or tutor, those sources should be acknowledged appropriately in the class header and detailed in your developer's logbook.

You must develop using the Eclipse version and jogamp libraries specified on Blackboard. It is your responsibility to submit an Eclipse project that runs on a COMP612 classroom machine.

It is expected that you will work consistently on this assignment, from hand out to due date. Time will be allocated in class each week for you to ask questions, get help, and work on your assignment. Please be aware that this is not an assignment that can be completed at the last minute. Each class you will be exposed to new concepts, and as we work through these concepts you will be able to progress your assignment. It is estimated that this assignment will take approximately 25 hours.

Due Date: Monday 29th April 2019 at 11:59 pm

Submission:

You must submit an electronic copy via Blackboard using the link provided. Your submission must contain:

- A zip file of your Eclipse project and any resources required for the code to run.
- An electronic version of your logbook as a PDF file.
- Please ensure that your java class headers, Eclipse projects and logbook file names contain your first and last name and student ID.

Late assignments, without an approved extension, will be subject to a deduction of 5% (one grade e.g. from C+ to C) of the total mark available for each 24-hour period, or part thereof, up to a maximum of five calendar days. Assignments over five days late will not normally be accepted or marked and students will receive a DNC (Did Not Complete) for that assessment.

If your ability to attempt this assessment or prepare for this assessment has been seriously affected by exceptional circumstances beyond your control you *may* qualify for **special consideration**. Applications for extensions should be made on or before the due date of the assessment and appropriate supporting evidence to be supplied within 5 working days of the application being submitted. Full details of how to apply and to determine whether or not your situation qualifies for special consideration can be found on Blackboard.

Marking:

This assignment is worth 30% of the overall mark for the paper. Marking not only takes into account this assignment specification, the quality of the OO design of your code, creativity, difficulty and quality of your feature extensions and the visual effect of your aquarium.

From the Office of Jacqueline Whalley, Associate Professor
<jwhalley@aut.ac.nz> | <+64 9 9219999 ext. 5203>

Plagiarism:

Please be aware that any piece of your assessment might be tested with the plagiarism prevention software to which AUT subscribes.

Assignment objectives:

This assignment assesses your ability to:

- apply built-in OpenGL matrix transformations (affine transforms)
- comprehend, set-up and use a track-ball camera
- create and animate a simple 3D hierarchical model
- implement a simple key listener for keyboard events
- appropriately scale an environment
- create an OO design that's simple, maintainable, reusable code

Notes:

- Your assignment will be re-compiled before running, so make sure it works on the AUT lab computers, not just on your home computer.
- For this assignment you are supplied with a basic application that sets up the OpenGL lighting for you. This is because we have not covered lighting yet and lighting makes everything look nice in 3D. Because we have lights we must use the COLOR_MATERIAL command. Do not alter this code unless you are confident that you know what you are doing and want to use MATERIALS.
- To render, in 3D, objects with transparency don't forget that you will need to temporarily turn off writing to the depth buffer while drawing the transparent objects.
- If you wish to, you may change the lighting setup but please note that in my demo the aquarium is centred at [0, 0, 0] and the provided lighting was set up for this environment. If you change this for your aquarium then you may need to also alter the position of the lights.
- For this assignment you have also been provided with a trackball camera. You must ensure that it is set up appropriately for your aquarium. The setup will depend on the scale and positioning of your aquarium. In my aquarium every 1 unit maps to 1 metre in the real world.

From the Office of Jacqueline Whalley, Associate Professor
<jwhalley@aut.ac.nz> | <+64 9 9219999 ext. 5203>

Aquarium

Part A

You have been provided with a basic skeleton on Blackboard that has a lighting setup and an origin marker. You may use this as a starting point if you wish or you may start your own project from scratch. If you use the starting code you need to read and understand the code provided. You have also been provided with a TrackballCamera class. You will need to read and understand this code. You should create and draw the camera and experiment with its parameters to get the view you want to achieve before proceeding. Don't forget to record these experiments and your observations in your logbook.

1. Scale

Decide on the scale for your tank in the real world and how this will map to OpenGL. Make a note of this in your logbook and in the header of your GLEventListener class. Once you have decided on your scale all your objects should be designed using this scale to ensure a realistic aquarium scene. Marks are awarded for the scale being noted in your logbook and code, and for consistent use of this scale in the design of your scene (which should be illustrated and discussed in your logbook as you develop).

2. Camera setup

Trackball camera class and methods must be used appropriately and suitable parameters used that allow the entire aquarium and the objects in it to be viewed from different angles.

3. Aquarium

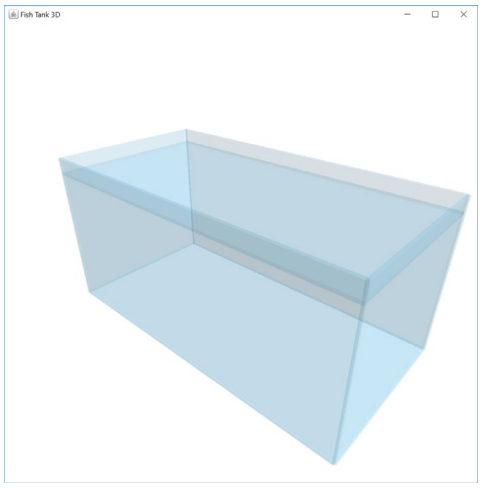
Create a 3D aquarium. Your choice of colour, alpha values and rendering order is particularly important in order to give the impression of glass. Remember that now we are working in 3D you need to create a tank with volume. Your aquarium can be any shape or size you want – the design is up to you.

Don't forget while doing this assignment to record any experiments you undertake, or bugs and solutions you encounter, in your logbook as you are doing the work. You must use screenshots and illustrations to support these discussions.

From the Office of Jacqueline Whalley, Associate Professor
<jwhalley@aut.ac.nz> | <+64 9 9219999 ext. 5203>

4. Water

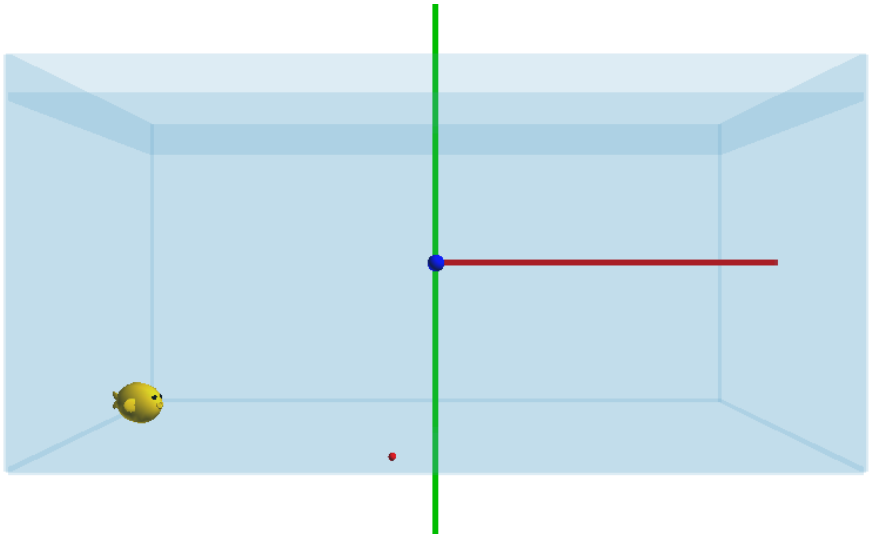
Add water to your tank. Make it look like it has volume and surface tension. As for the aquarium the choice of colour and order of rendering is important to make the water appear inside the tank.



5. Fish

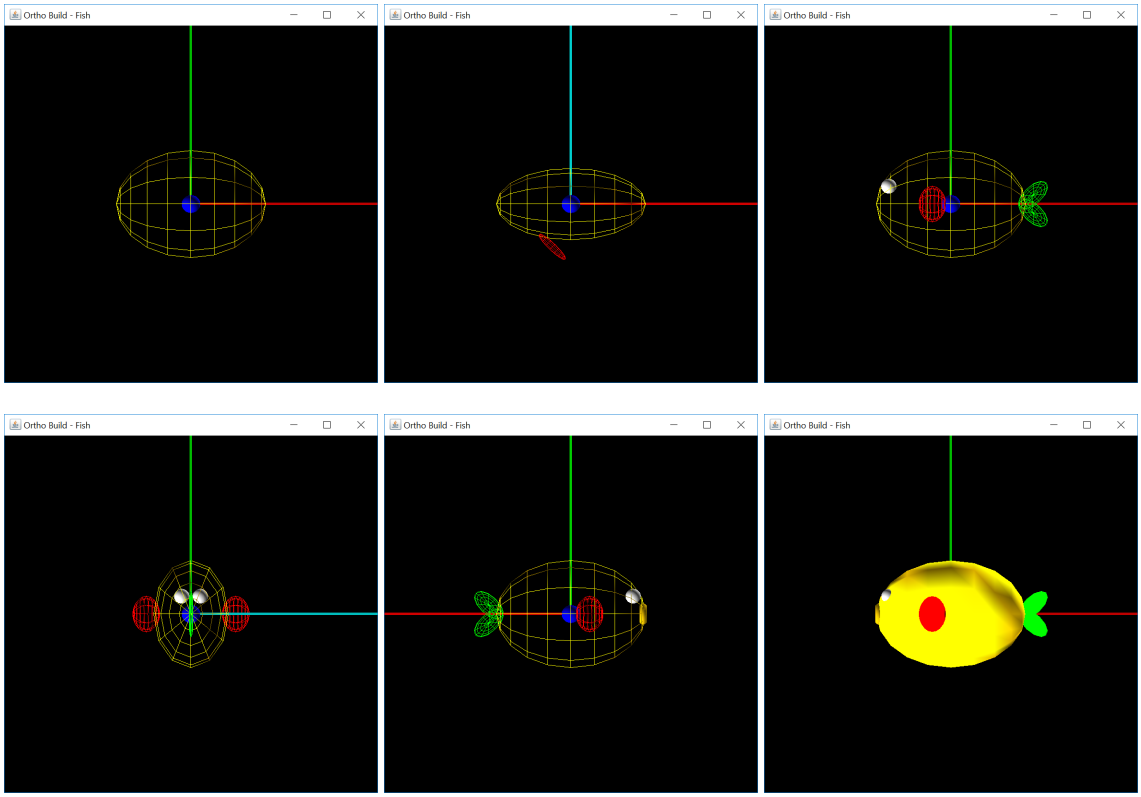
Use the built-in GLUT and GLU geometric shapes as well as hierarchical modelling techniques to build a fish or swimming creature of your own design. The fish should have moving parts and should swim around the tank. You will need to determine how you will implement your fish's swimming.

Recall that hierarchical modelling means that instead of just using an absolute transformation from the world origin for each individual part, you should use a relative transformation between a part and its parent in the hierarchy. For example, the fins should be placed relative to the fish body's coordinate system. This hierarchical structure is absolutely critical in the long run for both modelling and animation, even if it might seem like extra work at first! If you plan this carefully before you start coding you should have no trouble. Only use the OpenGL matrix transformations do not try to manipulate the model-view matrix yourself. In the past students who have ignored this recommendation have not been able to correctly view their model.



My fish was built using spheres and tori, with the code for animation and drawing encapsulated in a Fish class. If you need 3D shapes that are not provided by the GLU and GLUT libraries then please refer to the Parametric Surfaces examples and the course notes on Blackboard.

I strongly recommend building your fish in a separate project using the default orthographic projection and adding key controls to switch between viewing the font, sides, back, top and bottom of the fish model. It is much easier to build a model using the orthographic projection and also to use wireframe rendering and colour to ensure the positioning of the parts of the fish is correct. Once you are happy with your fish then you can add it to your aquarium code. Some of the building stages of my fish are shown below:



6. Animation Control

Allow the user to pause the animation by typing a key. When the animation is not paused you should allow the user to control the speed of the animation via 3 fixed speed settings. Display a listing of all the fully operational key functions on the screen. Your animation should always start on the NORMAL setting. You may wish to implement this early as it can help you visually debug your fish's movement.

```
Key mapping:
-------------------------------------------
SPACE: pause/restart
1: SLOW ANIMATION SPEED
2: NORMAL ANIMATION SPEED
3: FAST ANIMATION SPEED
```

From the Office of Jacqueline Whalley, Associate Professor
<jwhalley@aut.ac.nz> | <+64 9 9219999 ext. 5203>

Logbook

You must hand in your logbook as it forms part of your proof of authorship. Remember, the onus is on you to prove you are the creator of your product; thorough record keeping is essential to this process.

Your logbook should record dates, time spent, a record of bugs and fixes, design details, extension ideas, and details as to how you realised the extensions including illustrations and screen dumps.

Finally, you must provide a small statement in the last entry that critically evaluates what you did well, what you might do differently next time, and identifies the shortcomings of your application.

Part B

If you successfully complete Part A of this assignment you may be awarded up to a grade of B. To achieve a mark in the B+ to A+ grade range you should also complete Part B.

Please note that a well implemented Part A is likely to receive more marks than a poorly executed Part A and Part B.

Extend your aquarium in **two** ways.

Each extension should be significant and will be graded according to technical difficulty, novelty and visual effect. For example more marks will be allocated for animated effects or algorithmically difficult effects. Extensions are expected to be challenging and extend your knowledge. For example a school of fish employing an AI algorithm will gain significantly more marks than non-animated seaweed.

Part B is intended to be challenging and to extend your capabilities by giving you the opportunity to research ideas and to design something interesting. It is expected that you will come up with the solution and undertake appropriate reading independently – however support is available. You should discuss your ideas with the tutor before proceeding – at this stage they will tell you whether or not your idea is achievable and suitable. They will also be able to point you in the direction of appropriate resources and give you some pointers.

For marks to be awarded for Part B you are expected to investigate possible ways to implement your extensions independently and the work must be fully documented in your logbook. In some cases this will require reading ahead in the course materials and trying out the rendering techniques in small test projects prior to implementing them. This investigation should be presented in your logbook in detail and include relevant references to resources you used to help you design and implement your extension.