

# CMT107 : Visual Computing - Lab Sheet 2

## Transformation and Projection with User Interface

### 1. Use Mouse for Interactive Transformation

Download the file Lab2.7z from the Learning Central and extract it. Create a new project and copy the files extracted from Lab2.7z to the project. Note that .java files should be copied to \src folder, and the shader files should be in the project root folder. Run it and you will see one side of a cube, which is a red square. Move the mouse in horizontal direction (X-direction) with right mouse button down, you will see the cube rotating around a vertical axis (Y-axis). You need to do the following to fulfil rotation around X axis, translation, and scaling.

- In Transformation.java, complete the functions rotateX() and scale() to implement rotation around X axis and scaling operations.
- In VC02.java, add code in the function mouseDragged() to implement the following properties:
  - The object rotates around x and y axis along the mouse motion when the **right** mouse button is down. (The code for calculating the rotation angle around y axis has already been given.)
  - The object moves in the same direction as the mouse motion when the **left** mouse button is down.
  - The object is scaled up or down when the mouse moves down or up and the **middle** mouse button is down.

After finish coding, run it to see whether the cube moves as expected.

### 2. Transformation and Projection

Do the following experiments on VC02.java:

- First run the program to see the rendering result, then comment out the orthographic projection code and uncomment the perspective projection code in the function reshape (). Re-run the program again, and you will see a green window, which means that you are inside the cube and looking at the inside of the back side of the cube.
- Change the eye position in LookAt() inside the function display() -- move the eye in the positive z-axis direction (try different values), and you will finally see a red square, which means you have moved out of the cube, and the whole cube is inside the viewing volume. Explain how and why you can change the z coordinate to put the cube inside the viewing volume.
- Now, replace the line gl.glViewport(x, y, width, height) in function reshape() by gl.glViewport(x, y, width/2, height/2) to see the new rendering result. Explain this result.
- Modify the line again by gl.glViewport(x+ width/2, y+ height/2, width/2, height/2) to see what happens, and explain the reason.
- You may try to comment out T.ortho() lines and uncomment T.frustum() in reshape() to see what happens.