# Computer Graphics (COMP3271)
# Programming Assignment 1: 2D Transformation

Due date: 11:59pm, Sept. 30, 2022 HKT

**Submit your work (code.cpp) on Moodle before the deadline.**

**Purpose**

This assignment is about 2D transformation and draw fractals based on these transformations.

**2D Transformation Objectives**

- You are required to deal with the mouse interaction and collects these points which are input by the mouse to form triangles.
- Draw these triangles with OpenGL function.
- Generate a fractal uses these transformations based on the transformation matrices created from these triangle input.

**Template**

You will be offered a template so you don't need to do more work to configure the template.
The template has already implemented some initial work, such as initializing two windows, configuring the OpenGL and so on.
The interfaces of the functions which you need to fill in and some data structures you will used are given in 'code.h' and 'code.cpp'.
You may also use your own environment and library. But you might need to schedule a meet and demo your program to TA.

**Your Task**

Your task is to implement the following four functions whose interfaces are given in code.cpp.

>void **MouseInteraction(double x, double y);**

To deal with the mouse interaction and use the points which are input by the mouse to form triangles.

>void **DrawTriangles();**

To draw triangles which have been generated by the mouse interaction in the first window.

>void **AffineMatricesCalculation(Triangles in, Triangles out, Matrix matrix);**

Calculate the transformation matrices from the first triangle to each other triangle.

>void **RecursiveFractal(int k)**

Use the transformation matrices calculated in CalculateMatrices() function to generate a fractal.



**Procedure**: RecursiveFractal (k)

**Begin**
if (k > 0) {
For each triangle $T_i$ and transformation matrix $M_{0\to i}${
(Step 1) glPushMatrix(); // Push current transform matrix
(Step 2) glMultiMatrixd($M_{0\to i}$); // Multiply current matrix with $M_{0\to i}$
(Step 3) RecursiveFractal(k-1);
(Step 4) glPopMatrix(); // Recover transform matrix
}
}
else {
(Step 5) // Draw Triangle Here.
}
**End**

**Requirements**

Two windows should be opened side by side to display triangles input and fractals created by these triangles. Interactive techniques are provided with related functions. You just need to fill in four functions to make the whole thing work. See the course website for detail.

Observe the differences fractals created by input triangles, you can change the value of recursive times and positions in order to get different images from these operations. Try to create some special images from different arrangement of input triangles.