

START UP

In the console...

- Enter x,y or z to set the colour map to a co-ordinate, or s to map to the colour variable.
- Enter o at the prompt for orthographic projection. Any other key entered will set a perspective projection.
- Enter a background colour. Allowed colours are: green, blue, cyan, magenta, gray, yellow, orange, pink, red, white or darkGray. Black is the default colour and is used for all other input.

CONTROL

Translation

- Hold left click and drag in the left half of the screen to translate left, drag in the right half of the screen to translate right. (X-translation)
- Hold right click and drag in the top half of the screen to translate up, drag in the bottom half of the screen to translate down (Y-translation)
- Use the mouse wheel to move closer to (back) or farther from (forward) the jet model. (Z-translation)

Rotation

- Moving the mouse in the top half of the screen will cause rotation around the X-axis in a positive direction, moving the mouse in the bottom half of the screen will cause rotation around the X-axis in a negative direction (X-rotation)
- The 'a' and 'd' keys can be used to rotate about the Y-axis (Y-rotation)
- The 'w' and 's' keys can be used to rotate about the Z-axis (Z-rotation)

Left mouse click to switch between a WireFrame and Pseudo-Colouring.

NOTES

Please note this is my first time doing any programming with Java.

In particular, I had difficulty in abstracting the UI components in the application and thus implementing a MVC pattern.

A Model in the application represents an object (ie. the jet) and is comprised of Zones. It has variables for position and rotation angles which are manipulated through the control UI and used for drawing.

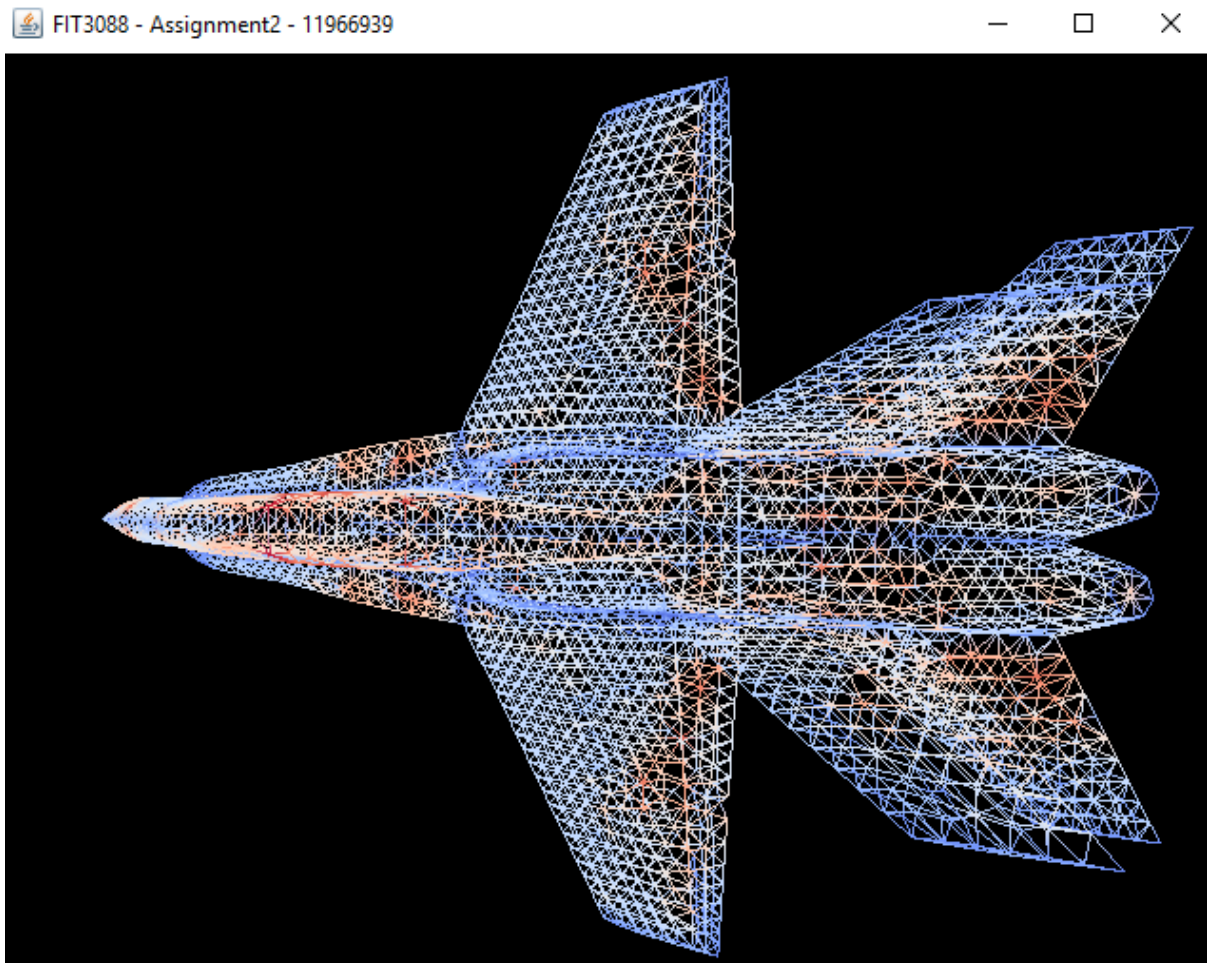
Each Zone has an array of arrays for its vertices and polygon lists and it is these that are accessed by the canvas for drawing.

Accessor methods have not been added mainly due to code simplicity and time. I have stuck to the convention that *private* members are not accessed outside of a class and *public* members are,

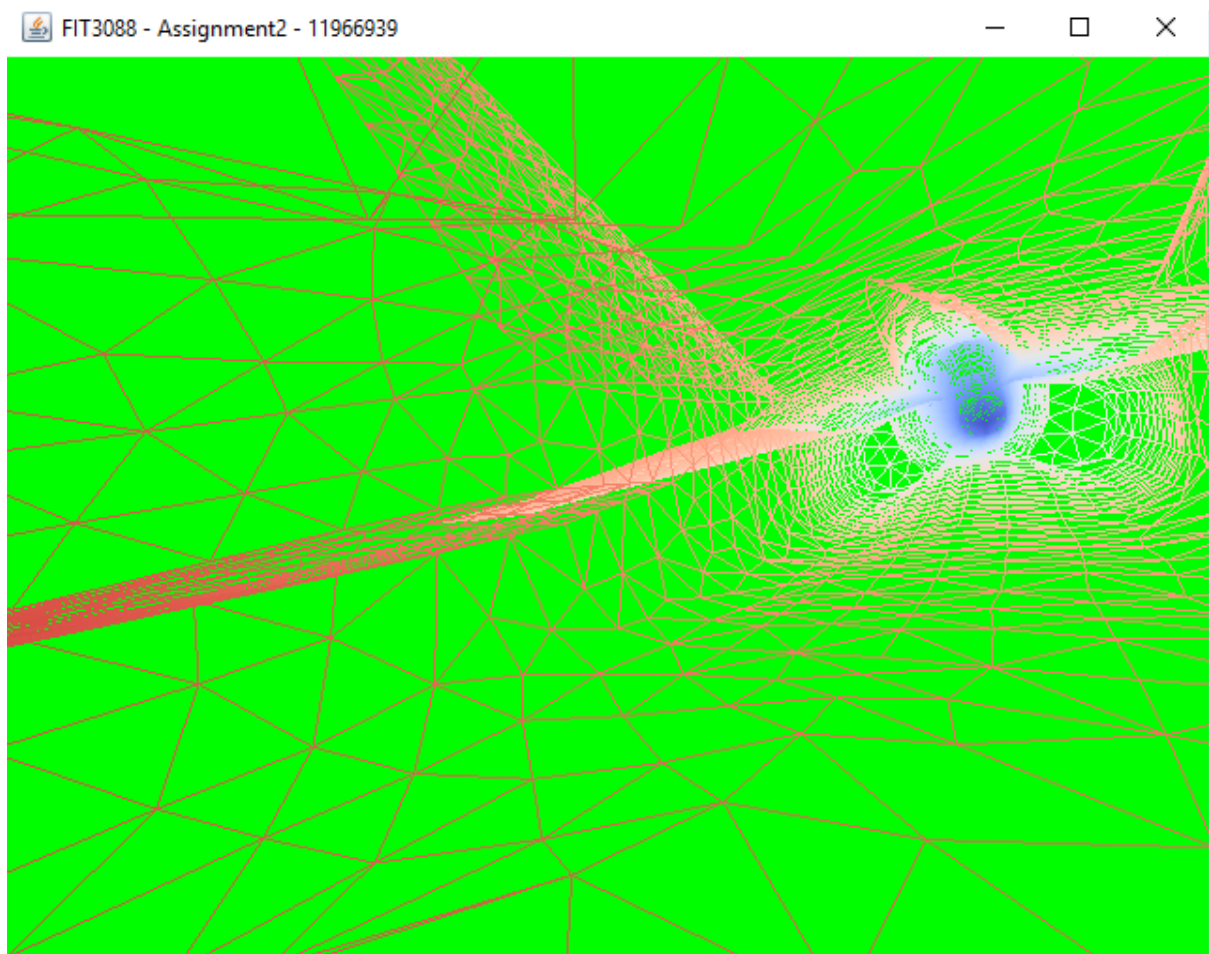
A ColourMap class handles the diverging colour map functionality and a ReadFile class is used by both the Model and ColourMap classes.

SCREEN SHOTS


Screen Shot 1 – Perspective projection, Wire Frame, S-Variable Colour



Screen Shot2 – Perspective Projection, Wire Frame, X – Coordinate Colour



Screen Shot 3 – Orthographic Projection, Fill, S-Variable Colour

 FIT3088 - Assignment2 - 11966939

