CSC 305: Assignment 1 README

The following document outlines the usage, algorithms, data structures and extra features of my assignment 1 submission. Note that this has been tested in **QT 4.8.6 with QT Creator 3.2.**

# Usage

To use my application, start the program with Command + R. The buttons are labeled with their use. The buttons can be used in any order, and will do as they are labeled.

# Algorithms

There aren’t that many algorithms, given that most of this assignment was about mapping matrices to polygons in OpenGL through QT. I did implement a few basic LinAlg library functions such as Inverse to invert a matrix using the CoFactor method and VectorTransform to multiple any arbitrary QVector3D by any arbitrary QMatrix3x3.

# Data Structures

QVector: a dynamic array. I used this data type because of the convenience given with built-in class member function calls such as .last() which returns a reference to the element at the last index. Also nice because it is dynamic, which means I don’t have to worry about size.

QVector3D: an object that stores x, y, and z, being used to represent a point in the Cartesian plane, with z as the perspective value. I used QVector3D because it allowed for built-in class member function calls.

QMatrix3x3: a 3x3 matrix type, stores a 2D array equivalent to [3][3]. I used this type because it allowed for automatic matrix multiplication using an overloaded version of multiply.

And finally a bunch of basic types including int, double, bool, double[], etc.

# Extra Features

Features that I included that were not listed as requirements include:

* The vertices of each polygon have their own colour set.
* After applying transformation matrices, the user is still able to add points where they clicked