

# CSC 305: Assignment 2 README

---

The following document outlines the usage, algorithms, data structures and extra features of my assignment 2 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. Add points through the 2D Orthographic views where they can also be dragged around. After 4 points have been added, catmull-rom splines can be seen between the points, excluding the first and last point. The 3D perspective window view can be moved around with the mouse. There are a few options on the left sidebar that have titles explaining what they do.

## Algorithms

The algorithm to draw Catmull-Rom Splines is hard-coded for efficiency. The hard-coded values come from matrix multiplication on MU and P, where MU is the polynomial breakdown of equations for catmull and P is the 4 points needed to interpolate over one segment.

## 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.

## Extra Features

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

- Tension Slider
- Five toggle boxes for: Ground Plane, Axis, Control Points, Control Lines, Catmull Rom
- 2D Orthographic Viewports are scaled around the axis to give the user a better experience, in other words, the user does not have their points being added to the edge of the other 2D Orthographic Viewports