



# COMP30020: Introduction to Graphics

## Individual Assignment 1

### Bézier Curves

September 24, 2021

---

#### Table of Contents

1	Drawing a Bézier Curve . . . . .	1
2	Submission . . . . .	2

---

## 1 Drawing a Bézier Curve

Previously you implemented a drawing app, learning how to rasterise lines and triangles. In this assignment you will be learning how to draw curves using piecewise polynomial interpolations. Given a set of control points, one method of approximating a curve is the Bézier curve. You will achieve this by implementing DeCasteljau's algorithm. For more details see class handouts (in particular sections on Hermite & Bézier Curves). A screenshot of your app is shown in Figure 1.

Examine the code in `void GLWidget::keyPressEvent(QKeyEvent *event)` before starting to implement any of the required functions. Try to determine what each key does and how you can best test each function.

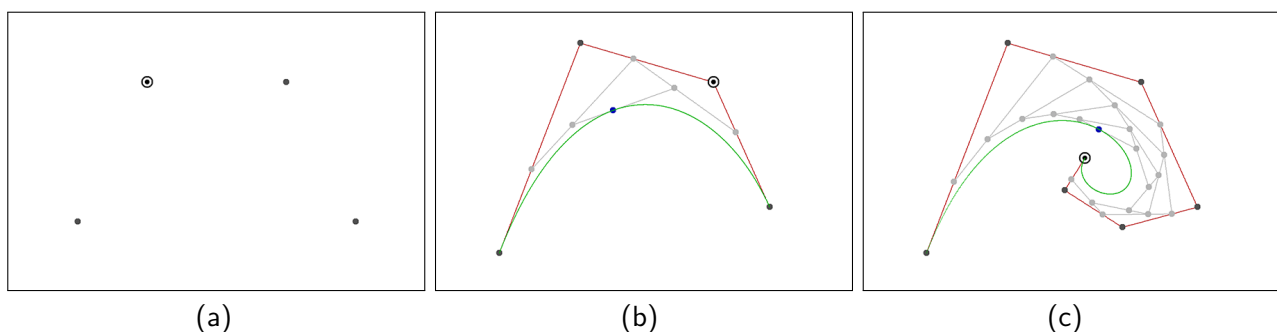


Figure 1: Screenshot of Bézier App using DeCasteljau's Algorithm

### Tasks

In the `glwidget.cpp` file you will find three functions you must implement. To make things easier, you should implement these functions in the following order:

1. `void GLWidget::drawControlPolygon()` to draw the control polygon .
2. `void GLWidget::drawDeCasteljau(float t)` to draw the DeCasteljau algorithm steps.
3. `void GLWidget::drawBezierCurve()` to draw the Bézier curve using DeCasteljau's algorithm.

Test that each function works before moving onto the next one. All of the above functions are called from `void GLWidget::paintGL()`. Study the `void GLWidget::keyPressEvent(...)` function to see how to turn each feature on.

Finally, write some suitable code statements in `void GLWidget::mousePressEvent(...)` which should permit the user to insert an additional vertex to the curve at the mouse pointer location, using a right mouse click.

## 2 Submission

- Submit your project folder (.cpp, .h and .pro) in an archive named:  
`<Firstname>_<Surname>_<StudentNumber>.zip`
- DO NOT SUBMIT NON-COMPILING CODE
- WORK ON SOMETHING SMALL, GET IT COMPILING AND WORKING. ONLY THEN MOVE ON TO THE NEXT PART