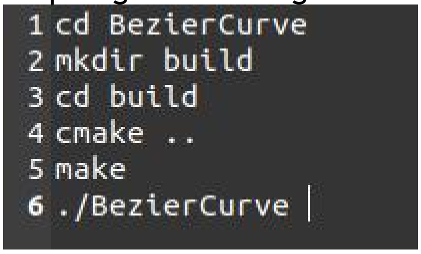
Curves in Computer Graphics

Introduction

i.This assignment is CS4102:P1-Curves in Computer Graphics. My student ID is 190004531, and the date of submission is March 10th.

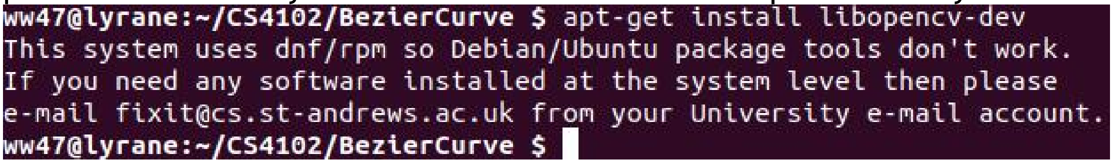
ii.For the implemented functions:
Language and library : C++, Opencv
Basic function: Implemented a window to draw the Bezier curve and the convex hull of the control points.
Realize the use of multiple control points(n > 2) to draw the Bezier curve.
Realized draggable each control point, while displaying of Bezier curves and convex hull in real-time.

iii.Compiling and running instructions with examples:



```
cd BezierCurve
mkdir build
cd build
cmake ..
make
./BezierCurve
```

I also try to compiling in the lab machine. However, I don't have permission to modify and download the most basic ‘opencv’ library.



```
ww47@lyrane:~/CS4102/BezierCurve $ apt-get install libopencv-dev
This system uses dnf/rpm so Debian/Ubuntu package tools don't work.
If you need any software installed at the system level then please
e-mail fixit@cs.st-andrews.ac.uk from your University e-mail account.
ww47@lyrane:~/CS4102/BezierCurve $
```

Design, Implementation and Use

i. Design

This assignment is to learn C++ and the Opencv library. The window visualization does not use QT or javaFrame, but to try some functions of the CV::Mat class. The program is mainly divided into three parts, drawing curves, drawing convex hull and mouse response functions.

ii. Implementation

Drawing curves: Drawing Bezier curve is the easiest part of this assignment, which implemented by a recursive algorithm to get the coordinates of the target point at a certain time *‘t’(0<t<1)*. This is the De Casteljau Recurrence Relation:

$$b_i^0(t) = b_i,$$

$$b_i^n(t) = (1 - t)[b_i^{n-1}(t)] + t[b_{i+1}^{n-1}(t)].$$

Drawing convex hull: A violent method is used to achieve the drawing of convex hull, and the algorithm complexity is $O(n^2)$.

Mouse response functions: This part is the most complicated and time-consuming method in the assignment. I use the left mouse button to let the user place the control point n. When n > 2, the convex hull and curves are displayed in real time.
I use the right mouse button to let user to select one point in the control points. Then press '*Ctrl*' and move mouse to drag the point move in the wondow, while showing the curves and convex hull in real-time. I did not set the conditions for terminating the setting of control points, and the user can continue to add control points after moving the point.

iii. How to use

a). Run the program we can get a window. Click left to put control point n in the window one by one.
b). When n > 2, there are curves and convex hull showing in the window. We can still add more point in the window, and the curves will change into new one by the control points set.
c). We do not need to press any keyboard to stop setting control points.
d). If we want to move one of point in the window, we will do two steps. One is to use right click to select that point, the other is to press Ctrl while move mouse.

Bibliography

docs.opencv.org. (n.d.). *OpenCV: Convex Hull*. [online] Available at: https://docs.opencv.org/3.4/d7/d1d/tutorial_hull.html [Accessed 10 Mar. 2021].

en.cppreference.com. (n.d.). *C++11 - cppreference.com*. [online] Available at: https://en.cppreference.com/w/cpp/11 [Accessed 10 Mar. 2021].

GeeksforGeeks. (2013). *Convex Hull | Set 2 (Graham Scan)*. [online] Available at: https://www.geeksforgeeks.org/convex-hull-set-2-graham-scan/.

Opencv.org. (2019). *OpenCV library*. [online] Available at: https://opencv.org/.