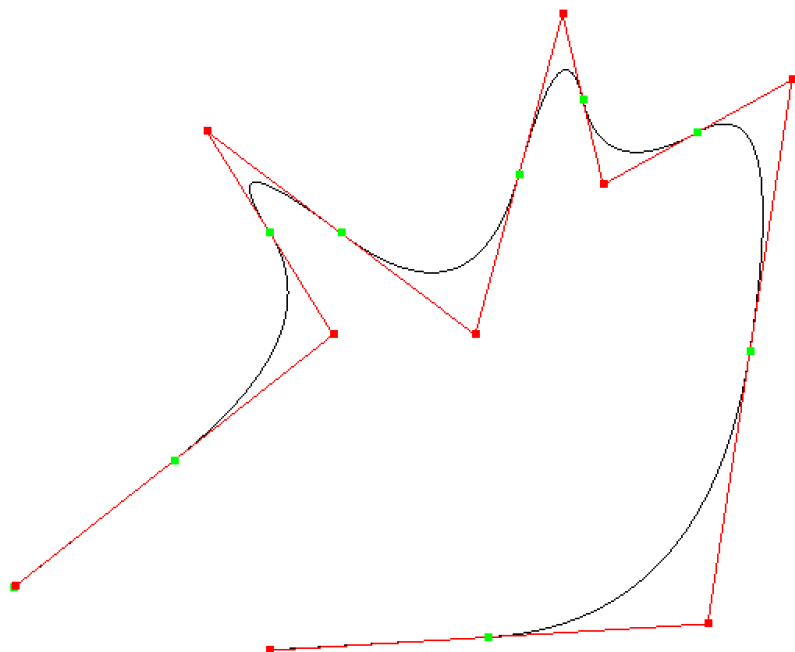
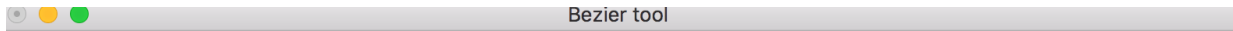
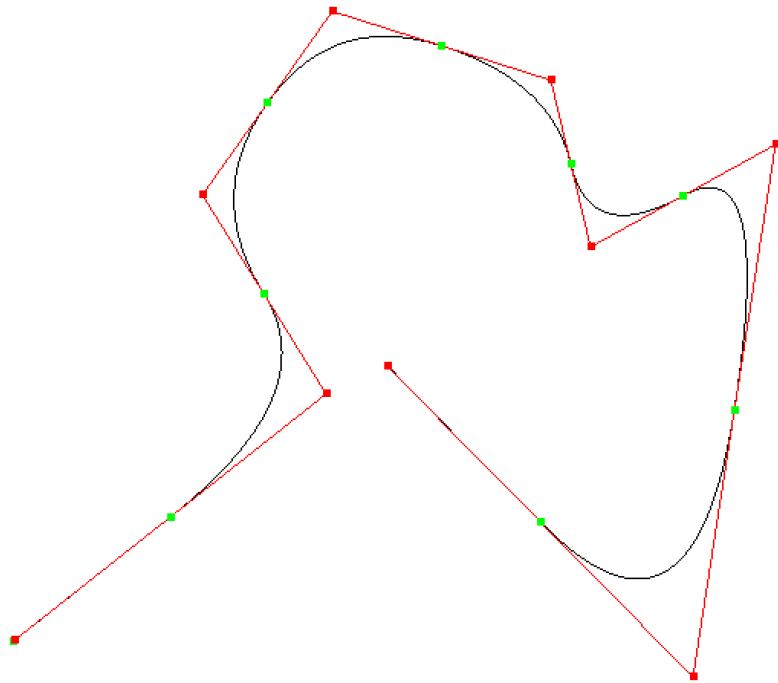
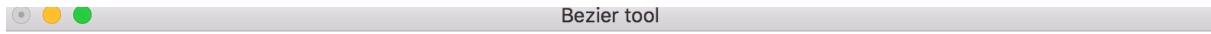
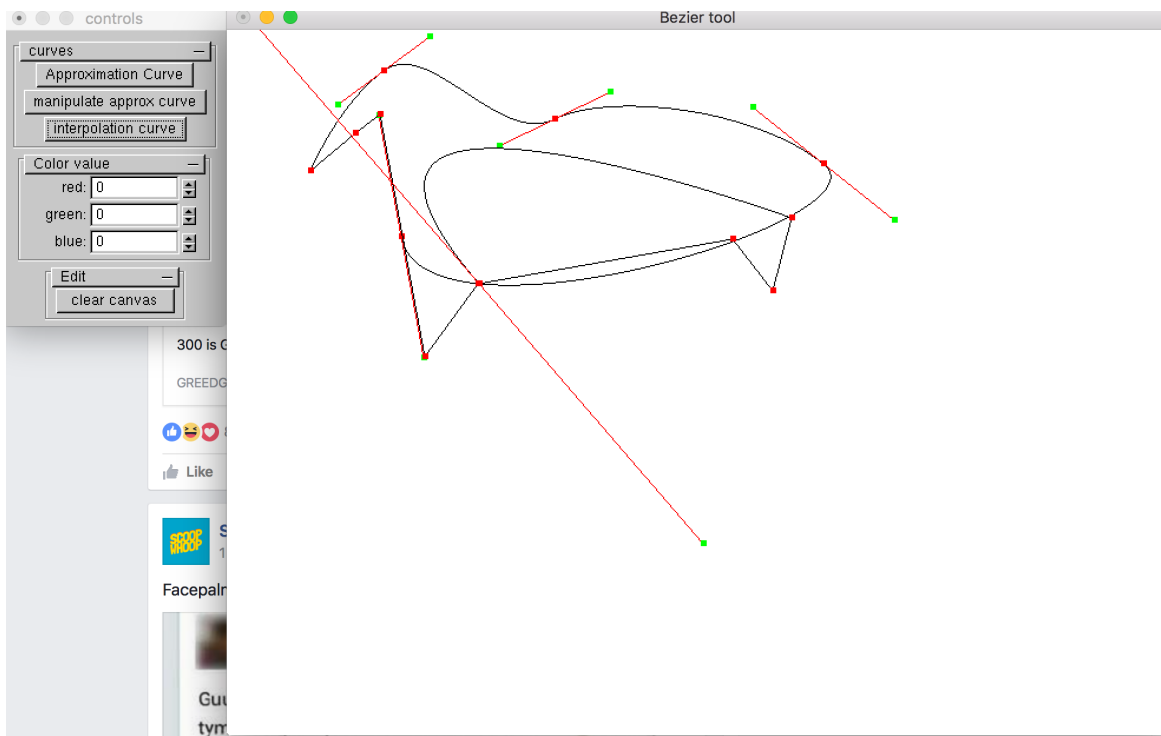


Bezier Tool (Assignment - 02)

Hammad Akhtar (2013060)



**Fig1 is an approximation curve,
Fig2 Curve in Fig1 after manipulation
Fig 3 A pen tool art**



Abstract

Design and create an interactive 2D paint program. It should support lines, circles, polygons. Colouring should be filled and bordered.

Keywords: function, abstract class, enum, GL_LINES, GL_TRIANGLES

1 Bezier Tool

This is the bezier tool designed. It supports following features

1. Curve
 1. add points : used to add new points for piecewise curve
 2. manipulate: on clicking this button one can manipulate the positions of control points using simple drag and drop.
 3. interpolation curve/pen tool: after clicking this one can draw using pen tool.
2. Color Value
 1. Red (0-255)
 2. Green (0-255)
 3. Blue (0-255)
3. Edit options

1.clear

: clearing the canvas completely

2 Implementation

2.1 Line Class

On the first and second mouse press positions are recorded and a line is drawn using those positions as vertex data. Vertex data also includes colour information supplied as argument to the constructor of line class. Line class implements *void draw()* function in which it uses **GL_LINES** enum for drawing the vertex data

2.2 Bezier Class

A series of points are passed to the constructor of Bezier class and a series of interpolated points are generated by varying the interpolation parameter t and passing it to the `Bezier::bezier` function as parameter.

Bezier::bezier is a recursive function. Its base is when vector of points contain only one point. If there are n points such that $n > 1$, then $n-1$ points are generated by linearly interpolating points $i, i+1$ such that i lies in range 1 to $n-1$. These $n-1$ points are passed to the subsequent recursive call.

2.3 Piecewise Approximation Curves

A series of points are passed to the constructor of PiecewiseBezier class and a series of phantom points are generated between points i , $i+1$ such that i lies in range 1 to $n-1$.

2.3.1 Generation of Phantom Points

Now in order to create a phantom point consider two points i and $i+1$. mid point is found between points $i-1$, i and i , $i+1$ lets call them $mid1$, $mid2$ respectively. Now we use $mid1$, i , $mid2$ for drawing a quadratic bezier curve.

2.3.1 Manipulation of points

In order to manipulate control points first one clicks on manipulate button thus entering the manipulation mode. Now one click and drag new control points to their new location. To determine the control point to be manipulated mouse events are used. Mouse click gives the x and y co-ordinates if there is a point in the vicinity of 5.0 units of (x,y) then that point is chosen as the point to be manipulated. There may be multiple co-ordinates in range but selection is made on first come first serve basis. Once point to be manipulated is chosen its position is manipulated as the position of cursor changes.

2.4 Pen Tool (Bonus)

When user left clicks, vector of its current position is stored as pos . Now while left mouse button pressed as the user drags the cursor changed positions are recorded in $pos1$ variable. As $pos1$ changes $pos2$ variable is updated whose value is $pos - (pos1 - pos2)$. Now let's assume after first left button click we get points $pos1$, $pos11$, $pos12$. After second mouse click we get points $pos2$, $pos21$, $pos22$. Now a curve is drawn between points taken in order as $pos1$, $pos11$, $pos22$, $pos2$.

3 Compilation Instructions

- Clone the Assignment02 directory to the desired location on your system.
- Open the terminal and `cd` into the Assignment02 directory
- execute **`cmake .`** command
- execute **`make`** command
- execute **`./Assignment02`** command
- Paint window will pop up with a canvas and a console.

References

SHADER UTILITY CLASS (USED FOR READING SHADERS FROM A FILE AND CREATING SHADER PROGRAMS)

[HTTP://LEARNOPENGL.COM/CODE_VIEWER.PHP?TYPE=HEADER&CODE=SHADER](http://learnopengl.com/code_viewer.php?type=header&code=shader)