CS4102 Computer Graphics

Assignment 1: Bezier Curves in Computer Graphics

University of
St Andrews

Student ID: 170009479

1 Introduction

Bezier curve is commonly used in computer graphics to draw curves. As the number of sample points growing, the curve becomes increasingly smoother and closer to the ideal curve. In this assignment, Bezier curve and some related work was implement by Java, which includes basic requirements and one enhancement of specification. The completed tasks are listed as follows:

- It can draw a quadratic Bezier curve by three control points.
- The number of sample points is chosen by users.
- Light source is specified by users.
- The greyscale value of samples changes based on the location of the light.
- Self-occlusion is accounted for, so the light only shows if not blocked.
- **Extension -** It can draw a cubic Bezier curve by four control points.

It should be noted that initially the lighting of the samples was done for the quadratic curve (by choosing the light with the user's 4th click), but in the final software, the light is selected on the 5th click and therefore lights up the cubic Bezier curve.

There are two sections in the main body of the report. Firstly, the functions of different classed are descried. Secondly, the results of the programme are displayed.

2 The functions of different classes

There are 4 classes in this programme, including "Calculation", "Display", "BezierCurve", and "Main".

1) Calculation class:
   a) Calculate the coordinates of a point in a normal vector of the line between two adjacent sample points. Each normal vector also pass through the middle points of the line between two adjacent sample points. To obtain x and y coordinates of a middle point, the formulas, $\mathrm{M(x)} = P_{i-1}(x) + \frac{1}{2}(P_{i-1}(x) + P_i)$ and $\mathrm{M(y)} = P_{i-1}(y) + \frac{1}{2}(P_{i-1}(y) + P_i)$, are used. To obtain x and y coordinates of the point in normal, the formulas, $N(x) = -P_{i-1}(y) + \mathrm{M(x)}$ and $N(y) = P_{i-1}(x) + \mathrm{M(y)}$

b) Do dot product calculation between unit vector of normal and unit vector of middle point to light. The main point of this function is to obtain two unit vectors, thus, the magnitudes of two vectors need to be calculated first.

2) Display class:

This class set the graphic user interface, which includes one frame, one panel one combo box and one label. In combo box, the users can choose how many sample points they want. Furthermore, this class invoke object in "BezierCurve" class to implement the display of Bezier curve and the change of greyscale value.

3) BezierCurve class:

a) This class overrides the function of "paintComponent" and "mouseClicked", implements the function of "lighten" and "lineItersect".

b) In the "paintComponent" function, it draw control points, light point, Bezier curve, sample points of Bezier curve, middle points between adjacent sample points, light line to each sample point, and the colour changed middle points after adding the light.

c) To get Bezier curve. The formula,$B(t) = (1-t)^2P_0 + 2(1-t)tP_1 + t^2P_2, 0 \leq t \leq 1$, is used with the condition of 3 control points. The formula, $B(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2, 0 \leq t \leq 1$, is applied with the condition of 4 control points.

d) The "lineItersect" function is used to determine whether path between sample points and light source is blocked by other segments of the curve. If so, the result is black sample points.

4) Main class is to invoke "Display" class to show the Bezier curve graph.

3 Results display

The results are shown as follows:

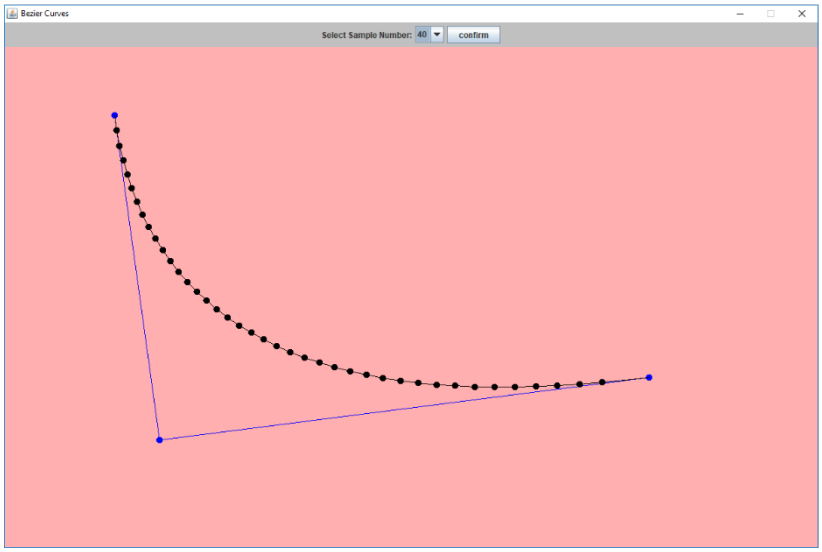

Figure 1 a Bezier curve example with three points and 40 sample points

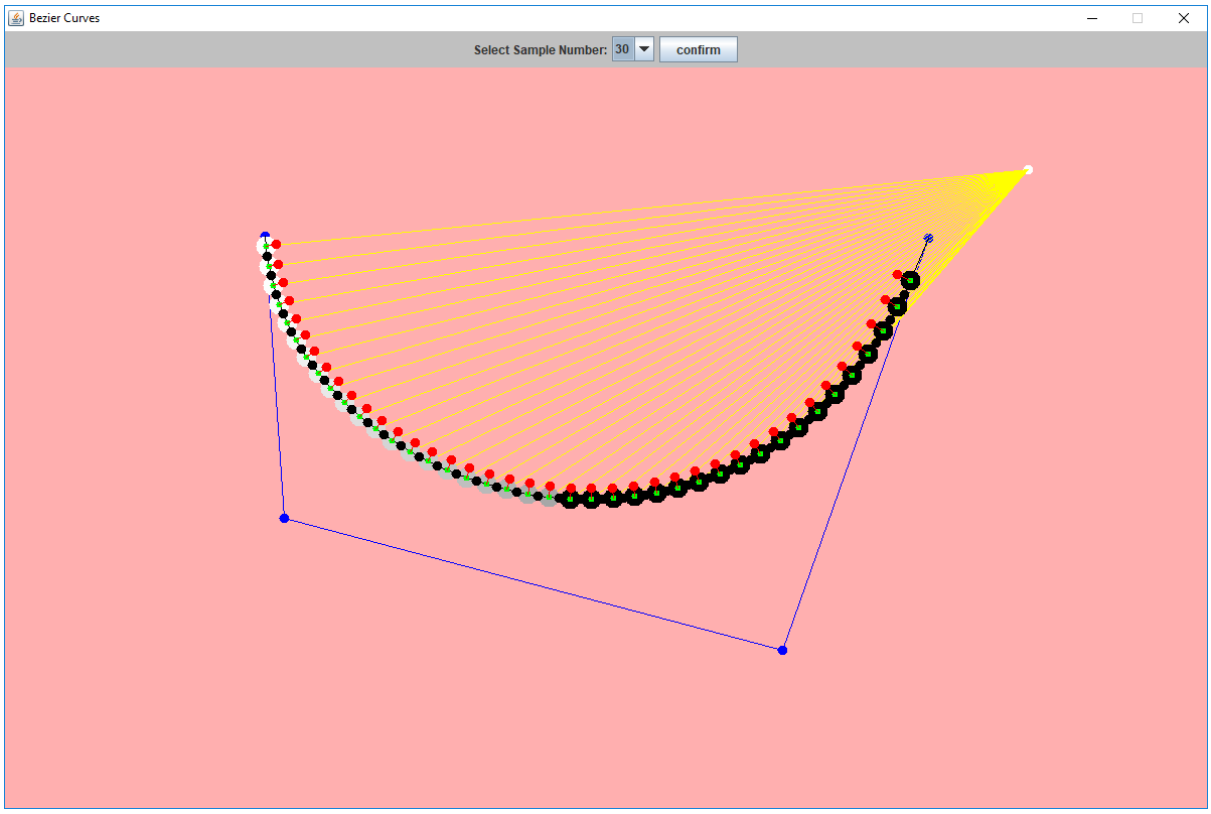

Figure 2 a Bezier curve example with some segments are blocked and 30 sample points

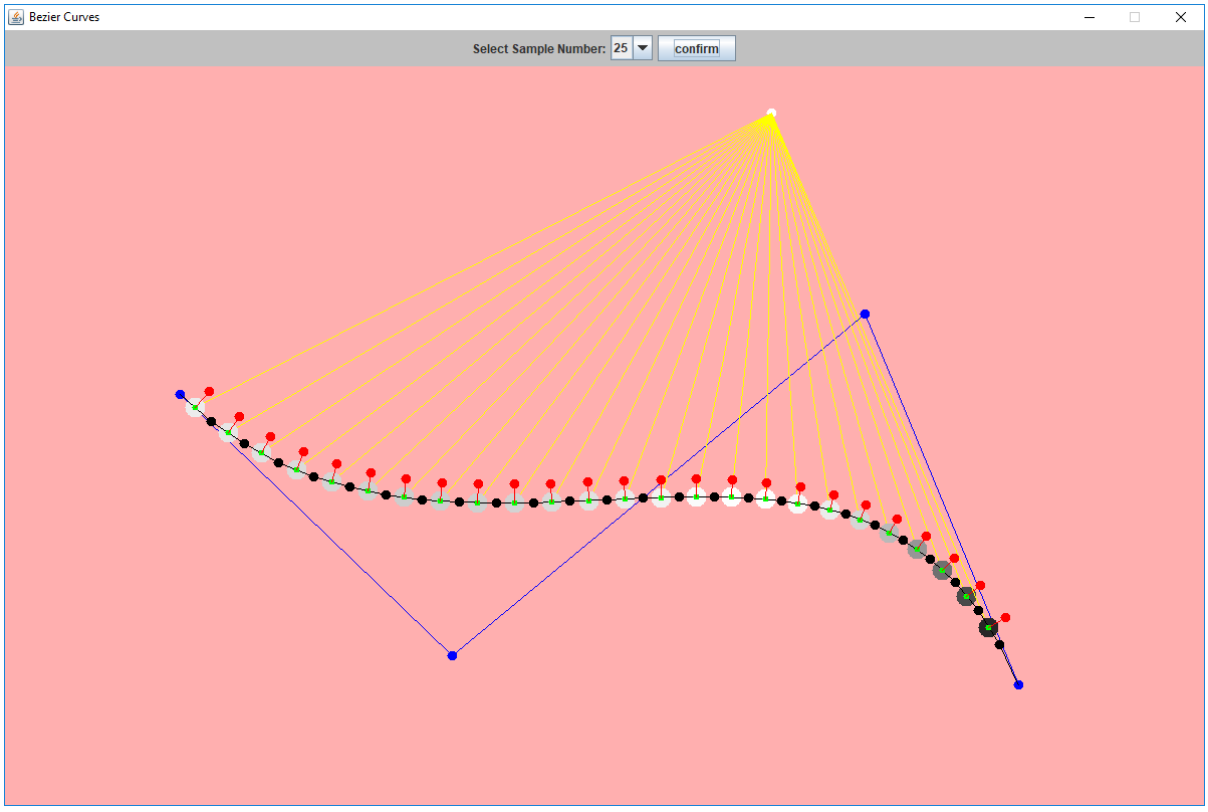

Figure 3 a Bezier curve example without blocking and 25 sample points