User's Manual for Programming Assignment No.1
For course CS6670, fall 2007
Submitted
By
Leena Kora.
(Uid: 0527667)

Abstract

The main aim of the assignment was to produce a graphics program, which helps the user to create and edit 2D Bezier curves. The user should also be given options like opening any curve dataset file and editing them. The second part of the program was to allow the user to create and edit 2D Rational Bezier curves.

Implementation

About the logic (Algorithms used)

1) I have used the iterative method of evaluating the Bezier curve, which is given the text book.
It is very simple. It includes copying all the control polygons in a temporary list and using it in the calculation of curve points using two for loops and storing the evaluated point in a final list. Evaluation function is called for each value of 't'. I set 't' to have 200 values from 0 to 1 and rendered the curve points using opengl command "GL_LINE_STRIP".

2) In case of rational 2D Bezier curves, I used the same algorithm with some necessary changes made. I multiplied 'x' and 'y' coordinates with 'w' coordinate of all the control points before using them in the evaluation function. Before storing the evaluated points into the final array, I divided those points with their 'w' coordinate. As I explained earlier, I have used slider for setting the 'w' values. When the slider is used and a control polygon is clicked, the 'w' coordinate of that control point is set with the slider value and curve is evaluated and displayed on the screen.

3) To implement file reading and storing, I have used fstream of c++ language and have written the code to read the files with the specified format.

About the GUI (Graphical User Interface)

1) I have used the template, which was provided to us to manage and organize the GUI well. I have created two classes called 'Bezier' and 'RationalBezier', for creating the curves, which are inherited by 'PolyLine' class.

2) As per the condition, I have used C++ for the programming. As I never worked using "FLTK", I spent a lot of time coding for the GUI and also was very curious to code using it.

3) I have added toggle buttons for building the GUI, so the user has to make sure that, once it's done with the button, it should be toggled off.

4) I added two toggle buttons to represent the two types of curves (2D Bezier curve and rational 2D Bezier curve). As the main purpose of the assignment is to render "Bezier Curves", I have coded such that at least one of them has to be on.

5) I have added one more toggle button to toggle the display of control polygon of the currently displayed curve. The display of the curve can also be toggled on and off.

6) I have also added toggle buttons for adding and deleting points of the control polygon. Users have to toggle them 'on' to add or to delete the points. I have made these buttons to behave as radio buttons because either a point can be deleted or a new point can be added. The user has to toggle them 'on' and has to click on their desired position on the screen to add or remove points.

7) I have made the display window to show nothing so that user can start adding points and can create their own curves, but with toggling at least of the curves display button "on".

8) User can move the control polygon points by just dragging, pushing and leaving. Just make sure that no add or delete buttons should be 'on'.

9) Users can also open any curve dataset file(with the mentioned format) by clicking the "File" menu and selecting "Open" option.

10) Users can also save the curves by selecting the "save" option from the "File" menu. When the 'save' operation is triggered, a dialog box is opened and the user has to enter the name of the file.

11) Users can clear the curve (also the display screen) by selecting “Clear Curve” option from the “file” menu and they can start creating a new curve by adding points or by reading the points from a file.

12) The “Exit” option from the “file” menu can close the application.

13) The users have to toggle off all the ‘add’ and ‘delete’ buttons before using any options from the menus.

14) In case of rational 2D Bezier curve, user can change the rational value of a point by toggling ‘on’ the button labeled as “click to change the rational value”. The user can set the rational value with the help of slider placed at the left of the GUI. I have placed a text display object which displays the value set in the slider. When the user clicks on any point of the control polygon, the curve is updated, leaving the control polygon unchanged. When the rational value change toggle button is set on, no other add or delete buttons should be on. Just to make the GUI friendly, I have set all the ‘add’ and ‘delete’ buttons to toggle ‘off’, when the rational value change button is set ‘on’. But the user has to make sure that that button should be toggle off after done or should not toggle any other buttons, when this button is being used.

Testing for this Project

I tested this assignment for better quality and performance.

1) I just deleted all the points and the degree of the curve turned to -1. After that I deleted again, and it segment faulted. This means that the user can do this by mistake, so I added some lines of code to avoid segment faulting. I just check if degree is less than or equal to -1 and if yes then the control does not go through the delete code.

2) I just toggled “add the point in the middle” button, when the number of control points was zero. The program just segment faulted, so added some lines of code to deal with that. When the degree is -1 and that middle button is toggled on, then it has to add the point at the end and from next attempt it should use the algorithm to add it in the middle.

3) The user can by mistake set the slider to indicate ‘0’, which make the program to behave weird since ‘w’ can not be zero in case of rational curves. To avoid it I have set the slider to show the values from 0.1 to 10.0.

4) When there is only one point in the control polygon points list, I set the control not to go through the evaluation function to calculate the final points. I just displayed the control polygon point and one curve point to optimize the code.