Computer Graphics : Assignment 1

Problem Statement:
Piecewise Bezier curves provide good local control and C1 continuity at joins. In the given code, a sequence of left mouse clicks on the application canvas are used to draw a piecewise linear curve connecting the points. Left mouse click ends adding of points and thereafter left mouse clicks can be used to select and move the control points. Study the code to understand it.
You need to change the given program to draw interpolating piecewise Cubic Bezier curves. You should
1. Describe your strategy to create the interpolating piecewise Bezier curve and to enforce C1 continuity.
2. Implement drawing of interpolating piecewise cubic Bezier curve from points taken progressively (i.e, even while the user is adding control points).

Part 1: Strategy to create the interpolating piecewise Bezier curve and to enforce C1 continuity

A cubic Bezier curve is defined by four control points: two anchor points (start and end) and two control points that influence the curve's shape(we call it intermediate control points).For an interpolating Bezier curve, you want it to pass through specific points. These points will serve as your anchor points(p0,p3), and the intermediate control(p1,p2) points need to be determined to ensure that C1 continuity is maintained.

To make interpolating piecewise Bezier curve we can use simple formula:

P(t) = (1 - t)^3 * P0 + 3 * (1 - t)^2 * t * P1 + 3 * (1 - t) * t^2 * P2 + t^3 * P3

Where p0, p3 are anchor points and p1and p2 are intermediate control points.

One of the mathematical strategy to ensure C1 continuity is points p2 and p3 in first bezier curve must be inline/collinear with p0 and p1 of second bezier curve while joining/connecting them(equation of line is C1 continuous).

As we determine p2 of first curve randomly and p3 by user; we can make p1 of second curve to be collinear with them(p0 of second curve is same p3 of first curve). Assuming Ratios of distance p2(first curve)p3(first curve):p2(first curve)p1(second curve) to be 1:K We can use:

$$[x_2 = \frac{k \cdot x_0 - x_1}{k - 1}]$$

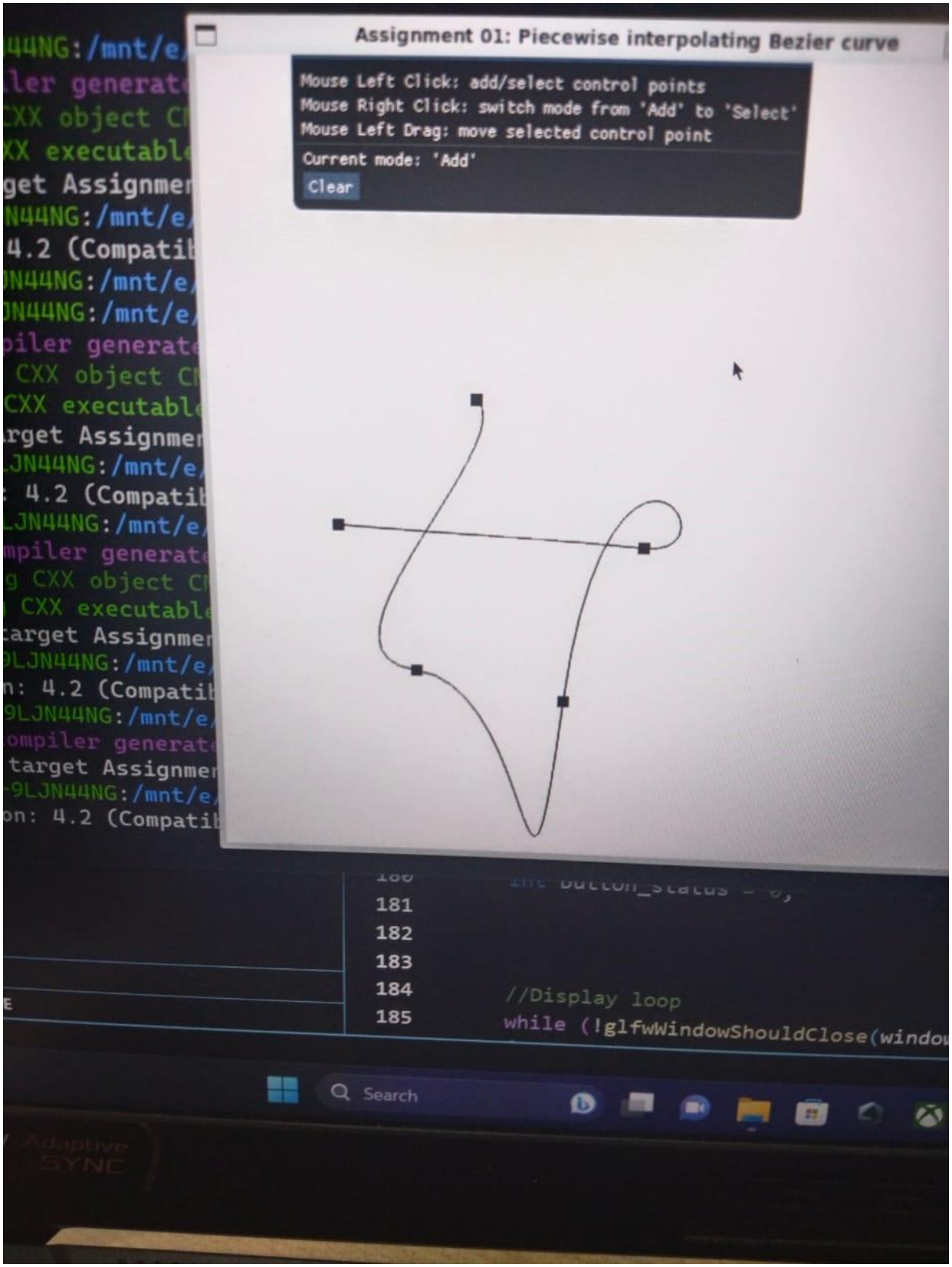
To determine p1 of second curve.

Part 2: Implement drawing of interpolating piecewise cubic Bezier curve from points taken progressively.

I implemented the calculatePiecewiseCubicBezier function to compute the cubic Bezier curve based on the given control points.I calculate the intermediate control points x[1] and y[1] based on if it's the first control point it's calculated randomly else we make sure C1 continuity as per method mentioned in part 1, x[2] and y[2] are calculated as well. Inside a loop, I compute points along the cubic Bezier curve for a specific segment using the cubic Bezier formula(mentioned in part 1) and push these points into the cubicBezier vector.

In the Main Loop:
I added code to update the VAO/VBO for the cubic Bezier curve whenever the control points are updated.
This ensures that the cubic Bezier curve is properly updated when new control points are added or edited.

Finally, I added code to draw the cubic Bezier curve inside the rendering loop.


Assignment 01: Piecewise interpolating Bezier curve
Mouse Left Click: add/select control points
Mouse Right Click: switch mode from 'Add' to 'Select'
Mouse Left Drag: move selected control point
Current mode: 'Add'
Clear
181
182
183
184 //Display loop
185 while (!glfwWindowShouldClose(windo
Search