CS 3451 Fall 2014 - Project 5
9/25/2014
Audrey Nelson

Coon's Patch
- Project 5 -

Audrey Nelson

Project Definition:
Create two quartic Bezier curves and two cubic Neville curves controlled by the user. Then, using Coon's Patch, interpolate between the left and right curves using the top and bottom curves as guides. If the user presses 'a', the continuous animation from the left curve to the right curve will be shown. If the user presses 's', the animation stops and the projected curves are controlled by the mouse.

Breakdown of Code:

Quartic Bezier Curve:
To compute a quartic Bezier curve you must compute each point along the curve at location s. To do this, you must loop through values of s from 0 to 1 and calculate the point along the curve for that value. The point along the Bezier curve at location s is computed in my code through the bezier method. This method takes in the 5 points that control the curve (A1, A2, A3, A4, A5) and the desired location s. To compute the point at s you must first find the location at s along each line segment that connects the points using a Lerp. These locations will become the set of points for the next iteration of Lerps. Continue taking Lerps at s between the points of the previous steps until there are only two points left. From here, the point on the Bezier curve at s is the point returned from the Lerp at s of those 2 points.

```
pt bezier(pt A1, pt A2, pt A3, pt A4, pt A5, float s){
     pt B1 = L(A1, A2, s);
     pt B2 = L(A2, A3, s);
     pt B3 = L(A3, A4, s);
     pt B4 = L(A4, A5, s);
```

```
    pt C1 = L(B1, B2, s);
    pt C2 = L(B2, B3, s);
    pt C3 = L(B3, B4, s);

    pt D1 = L(C1, C2, s);
    pt D2 = L(C2, C3, s);

    return L(D1, D2, s);
 }
```

Cubic Neville Curve:

To compute a cubic Neville curve you must compute each point along the curve at time t. To do this, you must loop through values of t from 0 to 1 and calculate the point along the curve for that value. The point along the Neville curve at time t is computed in my code through the neville method. The neville method takes in the four points that control the curve (A, B, C, D) and the desired time t. To find the point on the Neville curve at time t, you must calculate the Lerp of ABCD. To find the Lerp of ABCD, you must take the Lerp of ABC and BCD, which are calculated using the Lerps of AB, BC, and CD. The Lerps used for a Neville curve also include a, b, c, and d, which are the times at which the curve should pass through the corresponding point.

```
pt neville(pt A1, pt A2, pt A3, pt A4, float t){
    float a = 0, b = 0.333, c = 0.667, d = 1;
    pt L_AB = L(a, A1, b, A2, t);
    pt L_BC = L(b, A2, c, A3, t);
    pt L_CD = L(c, A3, d, A4, t);

    pt L_ABC = L(a, L_AB, c, L_BC, t);
    pt L_BCD = L(b, L_BC, d, L_CD, t);

    pt L_ABCD = L(a, L_ABC, d, L_BCD, t);
    return L_ABCD;
}
```

Bilinear:

The Bilinear point is defined by the vertical location, s, and the horizontal location, t, as well at the four corners that bound the area. To find the Bilinear point at (t, s) I wrote the function bilinear, which takes in the four bounding corners (A, B, C, D), as well as t and s. To calculate the bilinear point you must first take the Lerp of the two left corners with respect to s, as well as the Lerp of the two right corners with respect to s. Then the bilinear point is obtained by taking the Lerp of the two previous Lerps with respect to t.

```
pt bilinear(pt A, pt B, pt C, pt D, float s, float t){
   return L(L(A, B, s), L(C, D, s), t);
```

```
}
```

Coon's Patch:
The Coon's Patch location of a point is defined by the vertical location, s, and the horizontal location, t. To get the value of the point at (s, t) you must first find that point along the Bezier curve, B, find that point along the Neville curve, N, and find that point with respect to the bilinear, Bi. Then the point at (s, t) on Coon's Patch is calculated by adding the points gained from the Bezier and Neville calculations and subtracting the Bilinear point (B+N-Bi).

```
pt coon(float s, float t){
    pt B = L(bezier(C3, L3, L2, L1, C1, s), bezier(C4, R3, R2, R1, C2, s),
t);
    pt N = L(neville(C3, B1, B2, C4, t), neville(C1, T1, T2, C2, t), s);
    pt Bi = bilinear(C3, C1, C4, C2, s, t);
    float x = B.x + N.x - Bi.x;
    float y = B.y + N.y - Bi.y;
    return new pt(x, y);
}
```

Summary:
By using the Coons patch calculation at a specific time t for all values of s the result is the vertical curve between the two Bezier curves at time t along the Neville curves. If you then scroll through each t value, using all s values every time, the animation is shown of the transformation of one Bezier curve into the other along the Nevile curves.