

컴퓨터애니메이션 (CA) #HW2

Implement 2D curve editing interfaces B-splines



Self-scoring table

| | 1 | 2 | 3 | 4 | Total |
|-------|-----|-----|-----|-----|-------|
| Score | 1/1 | 1/1 | 1/1 | 1/1 | 4/4 |

2018707068 김경환

KwangWoon University

Code Analysis:

이번 과제에서는 Uniform Cubic B-spline curves을 위한 2D curve editing interface를 구현한다.

먼저 B-splines를 통해 curve segment를 그리는 방법에 대해 설명하겠다. Uniform Cubic B-spline curves는 controlPoints 중에 인접한 Control Point 4개를 순차적으로 지정하고, 해당 Control Point 4개와 0에서 1까지 균일하게 나뉜 t값을 사용하여 curve 위의 point 계산한다. 더 자세한 B-Spline curve 위의 point를 계산하는 방법은 t값을 Bell-shaped basis function에 대입하고 Control Point 4개와 곱하여 값을 얻어낸다. 이렇게 얻어낸 Curve Segment를 그릴 때는 B-Spline 위의 Point값을 GL_LINE_STRIP으로 그려낸다. 또한 Draw Control Polygon을 수행할 때는 B-Spline 위의 Point를 계산할 때 사용한 Control Point 4개를 GL_LINE_STIPPLE과 GL_LINE_STRIP을 사용하여 그린다.

위 과정을 통해서 n개의 Control Point에 대한 n-3개의 curve segment를 Uniform Cubic B-splin을 통해 그릴 수 있게 되었다.

이제 2D curve editing interface의 구현에 대한 설명을 하겠다.

첫 번째로 Add는 마우스 좌클릭을 눌렀을 때 control point의 개수를 표현하는 변수 N을 1 증가시키고, control point의 좌표를 저장하는 C++ STL의 2차원 vector에 1차원 vector를 추가한다. 그리고 마우스 커서의 좌표를 얻어와서 N번째 vector에 커서의 x, y, 0을 순서대로 추가한다. 여기서 Curve는 2D에 그려지므로 z좌표는 0을 대입한다. 그리고 마우스 커서의 좌표는 screen 좌표에서 얻어지므로 이를 world 좌표로 바꿔주는 과정을 수행한다. 이러한 원하는 마우스 커서의 좌표를 얻기 위해 좌표를 변환하는 과정은 4가지 기능 전부에서 이루어진다.

두 번째로 Remove는 마우스 좌클릭을 눌렀을 때 control point를 전부 탐색하며 마우스 커서의 위치와 가장 가까운 control point를 찾아내고 이를 remove한다. 이때 커서의 위치와 control point의 위치의 차이가 임의로 부여한 epsilon 보다 작을 때만 이를 수행한다.

세 번째로 Drag는 먼저 D키를 눌러 Drag Mode로 변경했을 때 해당 Mode가 여러 번의 Drag 이후에도 계속 유지되도록 하기 위해 마우스 왼쪽 버튼을 놓았을 때 pickMode를 COMPLETE로 바꿔준다. 그리고 Drag를 위한 좌클릭을 눌렀을 때 만약 pickMode가 COMPLETE라면 DRAGGING으로 바뀌주고 DRAGGING에 알맞은 일을 수행해주게 된다. 이제 DRAGGING에 알맞은 일은 먼저 Remove와 유사하게 모든 control point를 탐색하며 마우스 커서와 가장 가까운 control point를 찾고 해당 control point를 move의 이동에 따라 바뀌는 마우스 커서에 맞춰 좌표를 변경해준다.

참고로 두 번째와 세 번째에서 마우스 좌표와 control point 사이의 최단 거리를 구하는 방법으로는 두 point를 시작과 끝으로 갖는 벡터의 norm을 구함으로 계산할 수 있었다.

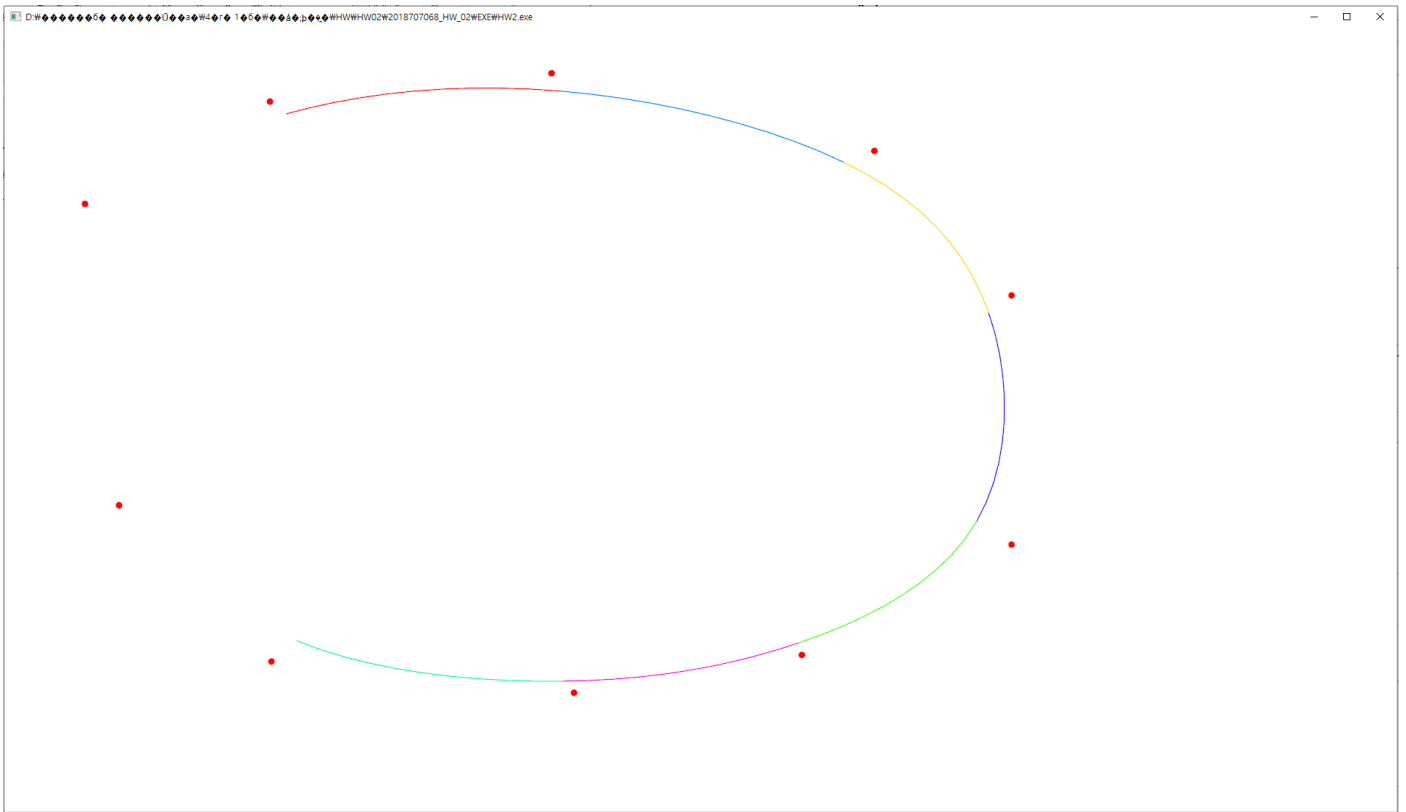
네 번째로 Insert는 먼저 마우스 point와 가장 가까운 Control Polyon을 알아내기 위해 모든 control point를 맨 처음부터 순차적으로 2개씩 뽑아 Edge를 만들어낸다. 해당 Edge와 마우스 point 사이의 최단거리를 구해 epsilon 보다 작을 경우 수선의 발을 계산하여 control point에 추가한다. 이때 curve segment의 순서를 고려하여 control point의 순서에 맞춰 control point를 적절한 위치에 삽입한다.

참고로 점과 직선의 최단거리를 구하는 방법으로는 Edge을 이루는 두 점을 p1, p2라고 하고 점을 p라고 할 때 $\mathbf{V1}(p-p_1)$ 와 $\mathbf{V2}(p_2-p_1)$ 의 cross product를 구한 뒤 $\mathbf{V2}$ 의 길이를 나눠줌으로써 구할 수 있다. 한편 최단거리는 $\mathbf{V1}$ 과 $\mathbf{V2}$ 의 dot product 값이 p1과 p2 사이의 거리보다 작고 $\mathbf{V1}$ 과 $\mathbf{V2}$ 가 이루는 각이 예각일 때만 최단거리가 유의미하다. 그리고 이때 수선의 발 위치를 구하는 방법은 점 p1에서 $\mathbf{V2}$ 방향으로 $\mathbf{V1}$ 과 $\mathbf{V2}$ 의 dot product만큼 더해주므로써 구할 수 있다.

1. Add 10 control points (a key):

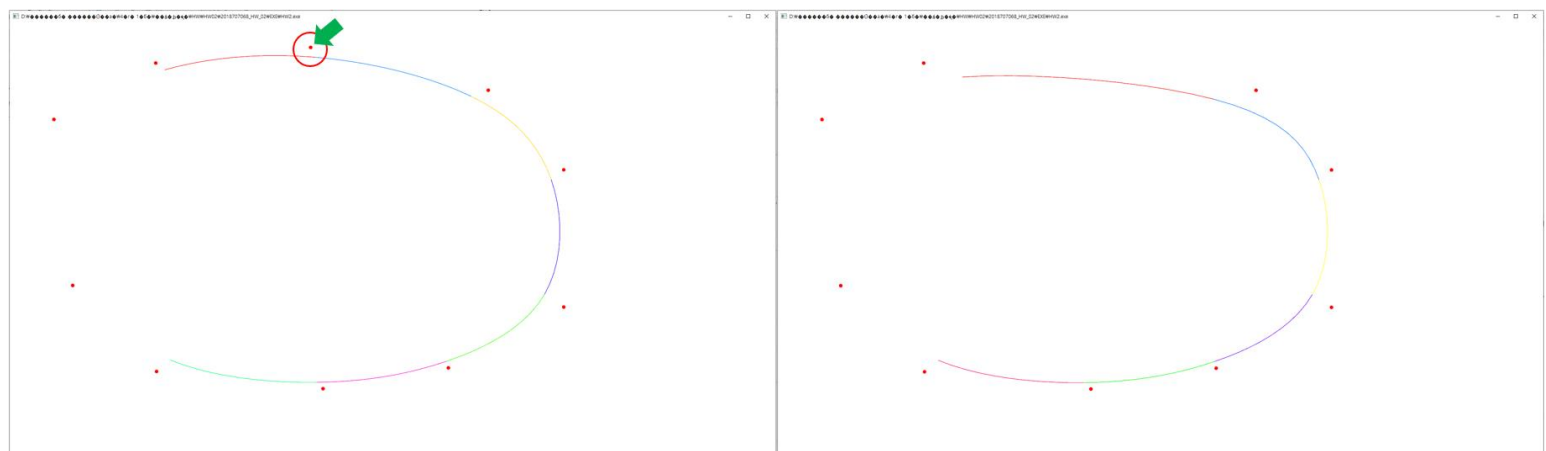
Mouse Cursor의 Click으로 4개의 Control Point를 추가해 총 4개의 Control Point로 그려진 모습이다.

Mouse Cusor의 Click으로 3개의 Control Point를 더 추가해 총 7개의 Control Point로 그려진 모습이다.



Mouse Cursor의 Click으로 3개의 Control Point를 더 추가해 총 10개의 Control Point로 그려진 모습이다.

2. Select/remove 3 control points (r key):



빨간색 원과 초록색 커서가 가르키는 Point를 클릭해서 Remove한 모습이다.

가장 왼쪽부터 1번이라고 했을 때 3번째 Control Point를 삭제했다.

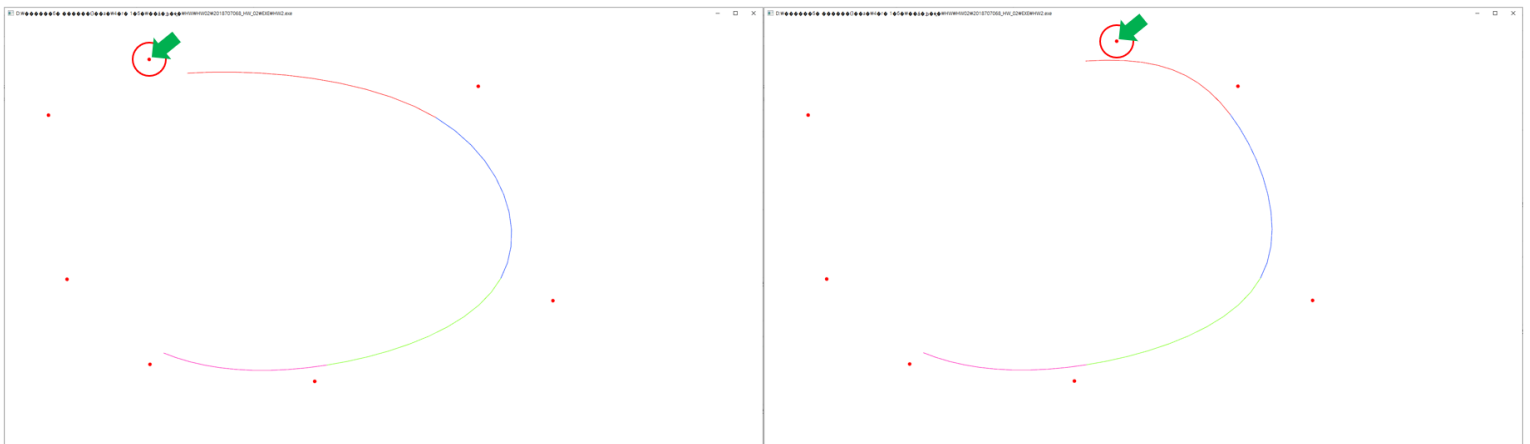


빨간색 원과 초록색 커서가 가르키는 Point를 클릭해서 Remove한 모습이다.
가장 왼쪽부터 1번이라고 했을 때 6번째 Data Point를 삭제했다.



빨간색 원과 초록색 커서가 가르키는 Point를 클릭해서 Remove한 모습이다.
가장 왼쪽부터 1번이라고 했을 때 4번째 Data Point를 삭제했다.

3. Select/drag 2 control points (d key):



가장 왼쪽부터 1번이라고 했을 때 왼쪽 사진에서 빨간색 원과 초록색 커서가 가르키는 Point에 위치한
1번째 Data Point를 드래그해서 오른쪽 사진에서 빨간색 원과 초록색 커서가 가르키는 Point에 위치하
도록 위치시켰다.



가장 왼쪽부터 1번이라고 했을 때 왼쪽 사진에서 빨간색 원과 초록색 커서가 가르키는 Point에 위치한
5번째 Data Point를 드래그해서 오른쪽 사진에서 빨간색 원과 초록색 커서가 가르키는 Point에 위치하

도록 위치시켰다.

4. Select edges of the control polygon and insert 2 control points (i key):

7개의 Data Point에 의해 만들어진 6개의 Control Polygon 중 1번째 Control Polygon 위에 커서가 위치하도록 하고 클릭했을 때 해당 cursor에서 edge에 내린 수선의 발을 control point로 삽입하였다. 해당 수선의 발은 control polygon 위에 내려진 것이기 때문에 control polygon 위에 control point가 추가된 것을 확인할 수 있다.

8개의 Data Point에 의해 만들어진 7개의 Control Polygon 중 4번째 Control Polygon 위에 커서가 위치하도록 하고 클릭했을 때 해당 cursor에서 edge에 내린 수선의 발을 control point로 삽입하였다. 해당 수선의 발은 control polygon 위에 내려진 것이기 때문에 control polygon 위에 control point가 추가된 것을 확인할 수 있다.