

## Project 2 Description

As implemented by Allison Gooch

In my applet, I implemented the key frame cubic Bezier curves based on control points, but the user only needs to draw the curve itself to set the points. I made an algorithm that approximates the curve they have drawn to the grid points I have created. Once the curve points are approximated as points on the grid and placed in an array of points, the algorithm finds 4 points from the array - the first point in the array, the last point in the array, and points at  $\text{length}/3$  and  $2*\text{length}/3$ . The points at  $\text{length}/3$  and  $2*\text{length}/3$  are moved further away from the curve to allow the Bezier approximation to lie closer to the drawn curve. I have also stored each point in the Bezier curve to a separate array, which I use to generate the in between curves.

To generate the in between curves in a static image, I used 5 divisions between each curve and performed a linear interpolation between each curve with  $t = 1.0/5$ ,  $t = 2.0/5$ , etc. To generate the curves in the animation and for the warping, I interpolated based on a time increment.

My warping algorithm is admittedly non complex, and simply moves the grid points a certain distance away from their original point based on how far each in between curve is away from its predecessor. The algorithm loops through all the given points on the key frame curve and moves them accordingly along with the linear interpolation curves.

I am not sure specifically what qualifies for extra credit, but I believe my curve drawing function is accurate and more complex than was required for the assignment. I also chose to do this project alone rather than with a partner, which may warrant a bit more credit.