Project 3 for CS3451, Spring 2008: Stick animation

Revision 0

Part P3A) Provide an interactive applet in which the operator can specify a series of **strokes** (short line segments) in the plane. Each stroke is simply specified by a mouse stroke (point-click-drag-release action). Provide simple tools (combinations of key presses and mouse actions) for editing the series of strokes. For example, holding 'p' down ('p' for pick), the operator may click close to a stroke to select it. The selected stroke should appear in a different color. Then holding 'm' down, the operator may click and drag to adjust the closest end-point of the selected stroke. Holding 'i' down ('i' for insert) and entering new stroke will insert them after the current stroke. Holding 'd' down and clicking the mouse will delete the nearest stroke. Also provide a tool for saving the strokes on file and for reading them back in. (The saving may not work in a browser but the reading should. Make sure that the applet can read several files, because you will need this option for Part B.) Create a set of strokes that mimics the motion of a juggling stick and save them on file.

Part P3B) Implement 3 interpolation methods for animating the motion of a **stick** that interpolates (or approximates) the consecutive strokes. Provide a simple interface for selecting one of these methods (pressing '1', '2', or '3') and for starting and stopping animation (press 'a'). Make sure that the animation runs at about 30 **frames** per second and takes between 5 and 15 seconds total. Also provide a stroboscopic view where all the frames of the animation are superimposed to show in a single image the whole animation. You may either assume that the animation is a loop (first stroke follows the last one) or not (your choice). Run the applet at http://www.gvu.gatech.edu/~jarek/demos/twistBender/ and press 'a' to see a stroboscopic view and a similar project.

Assume that stroke i starts at point S[i] and ends at point E[i]. The animated stroke starts at point S and ends at point E.

- Method 1 should be a linear interpolation between consecutive strokes: S=s(S[i],t,S[i+1]) and E=s(E[i],t,E[i+1]) as t varies between 0 and 1. Then you increment i.
- Method 2 should use the four-point subdivision to construct a series of starting points rS[i] from the control points S[i] and a series of starting points rE[i] from the control points E[i].
- Method 3 should be your own invention and should have clear advantages over the other two methods (more intuitive results, smoother motion, easier to design with…).

Deliverable: On your web page for this project, you should post the **applet** and **instructions** on how to run it and test all its features, links to the **source code** and data files (and if possible a **zipped file** of the folder with the source code and data file), a **discussion** of the shortcomings of methods 1 and 2 with **images** (or data sets that can be loaded into your program) that illustrate these shortcomings, and a **presentation of your method** (what it does, how you have implemented it, and why it is better than the other two).

Extra credit: If you go beyond these requirements, please clearly mention the extra credit work (software, analysis, research ideas). If you find ideas in papers, books, or online, please reference them clearly and precisely. The total extra credit will be capped to 100% of the nominal points for the project. Extra credit ideas (with **caps** on associated extra points, assuming it is done well):
* Ensure that the strokes and the animated stick have all the same length (10%)
* Instead of straight lines, use Bezier control polygons as strokes and animate a Bezier curve (30%)
* Measure and display the jerk for the end-points of the animated stick (30%)
* Render a texture that moves (scaling, translation, rotation) as if attached to the stick (20%)
* Do the whole thing in 3D (50%)