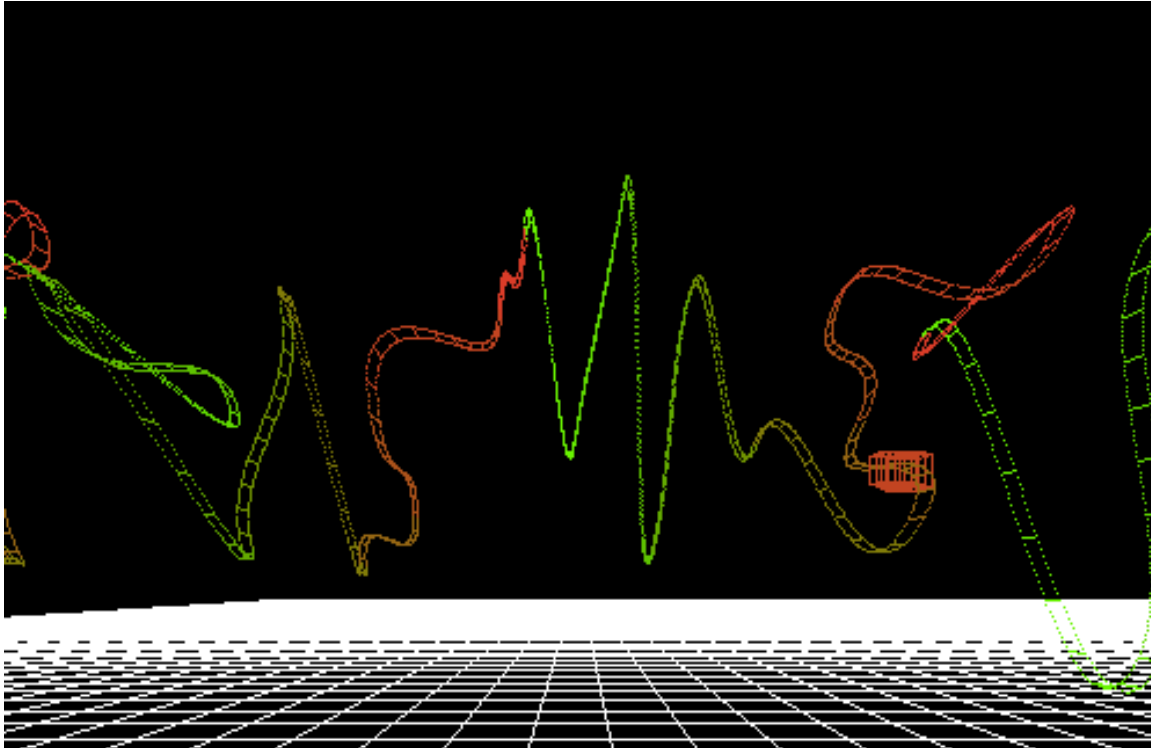


Roller Coaster



Description:

Today you will be implementing a spline in 3D. A spline is a function that allows us to interpolate between a set of points, resulting in a smooth curve. There are examples of the spline provided below. Also, be sure to study the course lectures to make sure that you understand how cubic splines work before you start.

Demo: (Spline Example)

<http://www.cs.tufts.edu/comp/150VIZ/support/spline>

Your Task:

- Implement the member functions of the spline class for.
 - draw_spline
 - This function is responsible for drawing a set of interpolated points between a starting and ending (“goal”) point.
 - calculate_Spline
 - This function is responsible for generating a single interpolated point between a starting and ending point.
 - (Optional) renderAsRollerCoaster
 - draw your spline as a roller coaster (whatever your interpretation of this is)

- In main.cpp, modify the chunk of code in MyGlutRedisplay, under:

```
if (following == 1) {...
```

- (Optional, but strongly encouraged): Modify that section (and maybe a few lines above) such that the roller coaster becomes a “first person” experience!
- (Optional) Implement additional interesting camera path (first person, third person, or another interesting view) that moves along the spline.
 - One suggested strategy to do this is to populate a dynamic array(private member: computed_x_y_z) with a list of points in your spline.
- (Optional sub tasks) Add some interactivity
 - Modify the class so that we can change the velocity
 - Figure out how to add loops! (Challenging!)

Files Given:

main.cpp – Main
spline.cpp and spline.h – Fill in the empty cubic spline and calculate_spline method
controlPoint.h – A point in 3D space along the spline

Finished Early?

- Load a ply object and have it move along a path.
- Make your roller coaster move normal to the spine
- Make your roller coaster accelerate and decelerate based on the slope of the curve

Going Further:

Did you enjoy this in class assignment? Combine the spline with your movie scene! Have your camera move along a spline.

- Make your spline interactive
 - Combine a visual interface with your spline so you can move points and watch the spline transform.
- Try implementing a [Bezier curve](#) which is a different way of interpolating a curve along a point.
- How would you implement loops (so that along the x-axis we can get smooth curves that overlap)?
- Color the points along the spline according to some metric.
 - Some ideas
 - Slope (and once you know slope, you can model your
 - Local max or minimum