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. Be sure to study the course lectures to make sure that you understand how cubic splines work before you start.

Your Task:
- Implement the member functions of the Spline class for.
  - `draw_spline`
    - This function is responsible for drawing a spline.
    - A spline is defined by a number of control points
      - In Spline.h, this is a variable that is set by its caller.
      - In the case of the support code, Spline is created in the MyGLCanvas's constructor.
    - The control points are randomly generated and stored in an array called controlPoints (see Spline.h)
    - In addition to drawing these control points, you also need to draw a “path” between each pair of control points.
      - Between two control points (a starting and an ending control point), you need to render the “dots” (or interpolated points) between them. This is
decided by the constant RESOLUTION (in Spline.h), which is set to be 100

- calculate_Spline
  - This function is responsible for generating a single interpolated point between a starting and ending point.
- In MyGLCanvas.cpp, modify the chunk of code in drawScene(), under:

  ```cpp
  if (followCoaster == 1) {
  ```

  - Modify that section such that the roller coaster becomes a “first person” experience!
    - (HINT!) Look at how the 4 carts are computed and rendered...
    - Alternatively, you can derive the gradient of the spline function
  - (Optional) Implement additional interesting camera path (first person, third person, or another interesting view) that moves along the 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 draw_spline and calculate_spline methods
- MyGLCanvas.cpp and .h – contains the rendering in GL

**Finished Early?**

- Load a ply object and have it move along a path.
- Make your roller coaster move normal to the spine

**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