

This program implements the construction of a roller coaster and a first person perspective ride. The program takes as input a spline (.sp) file and renders a 3d spline using the catmull rom interpolation method. In addition to the spline file, three other texture files are used to texture map scenery onto the program. In order to begin the first person animation, press the “1” key, and the “2” key to pause. Details implementing each level of the coaster are listed below:

Level 1 (Spline): The spline is calculated by iterating through consecutive sets of 4 points given from the spline file. With each set of 4 points, a new set of points are generated in between the second and third point using the catmull rom spline calculation and a fixed u step size.

Note: I coded the recursive subdivision and had it working, but the ride was still not smooth because it lacked a proper way to estimate the curvature between 2 points. The code for the subdivision is commented out at the bottom of the main.cpp file

Level 2 (Ground): One large square area is drawn with repeated squares each with a texture mapped. To ensure realism, the texture is seamless (no perceptible borders).

Level 3 (Sky): gluSphere was used to draw a 200pt radius sphere. A sky texture was mapped onto this sphere. The ground is placed in the middle of the sphere to give the appearance of a dome.

Level 4 (Ride): the ride is implemented by adjusting the camera according to a counter. The counter is updated each time the display function is called. When we need to adjust the camera, we calculate the next point on our roller coaster using the same method used to draw the spline. At each point, the camera looks out from the point on the spline, looks at a point proportional to the tangent line, and uses an up vector. The up vector is calculated using the suggested method of an arbitrary start vector and binormal and normal computation based on previous iterations.

Level 5 (Track): The track is implemented drawing narrow rectangles in between the current and next point on the spline. The positions of these rectangles are on either side of the spline point using the binormal calculation. The cross bar is generated using a rectangle in between the track rails.

Extra credit features:

- 1) Texture mapped steel roller coaster
- 2) Double rail and cross bar
- 3) Texture mapped supports
- 4) Texture mapped sky dome and seamless ground
- 5) Recursive subdivision code written