

Lab 5: Splines

Objective

- Understand the use cases, implementation details and differences between Catmull-Rom and Bezier splines

Goals

- Implement Catmull-Rom path traversal with n points
- Draw the full Catmull-Rom path
- Implement cubic Bezier interpolation for one segment

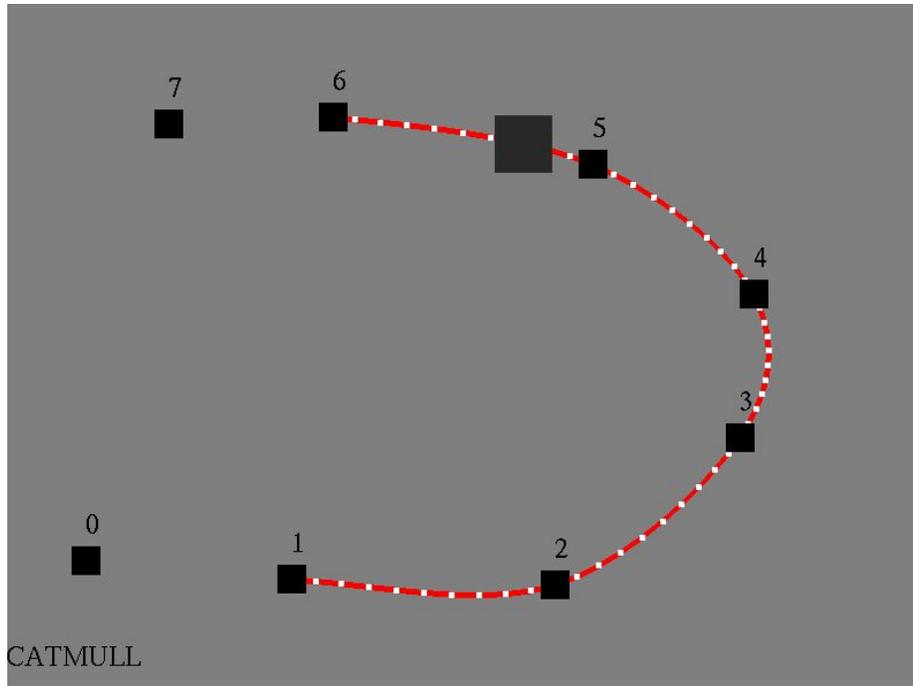
Itinerary

Note:

- You may work individually or in groups of 2 - 3. Only one person needs to submit, make sure everyone's names are in all of the submitted source files.
- Pressing the "L", "C" and "B" keys will toggle between LERP, Catmull-Rom and Bezier interpolation.
- Pressing the spacebar will clear all points

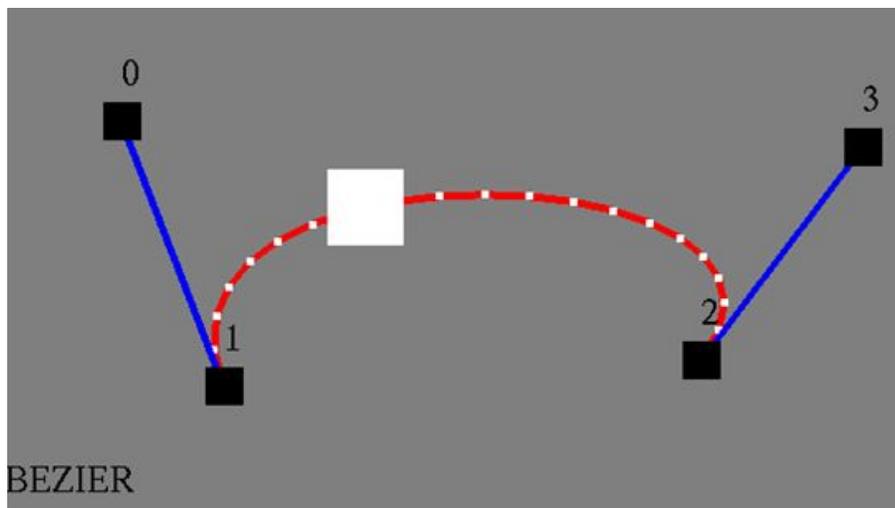
Part 1: Catmull-Rom Path Traversal

1. The provided starter code performs Catmull-Rom interpolation along one segment. The first task is to implement path following by adding functionality for traversal along any number of segments.
2. Open up AnimationMath.h, here you will find a function named "catmullUMP". This function is an implementation of the "UMP" formula described in the lecture slides. Compare this code with the lecture slides to get an understanding of what is happening, you should understand what all of the input parameters are before continuing.
3. Open up main.cpp and go to the "doCatmull" function. This is where the Catmull-Rom interpolation is performed. Currently this function is hardcoded to use the first 4 points placed. Modify it to work for any number of points.
4. Go to the "drawCatmull" function in main.cpp. This is where we will draw the path. Currently it is hardcoded to draw the first segment. Modify the logic in this function to draw the entire path. The result should look like this:



Part 2: Bezier Segment Traversal

1. In main.cpp, go to the “doBezier” function. This function is fully implemented and calls the “bezierUMP” function located in AnimationMath.cpp. You need to implement the bezierUMP function. Refer to the “catmullUMP” function as a reference and the lecture slides for details.
2. Go to the drawBezier function located in main.cpp. This function is fully implemented, take a look at it and get feel for what it is doing. The final result should look like this:



Submission

Submit a zip file the following:

1. Any source files you modify (AnimationMath.h and main.cpp at minimum)
2. Please submit a **zip** file and not individual files.

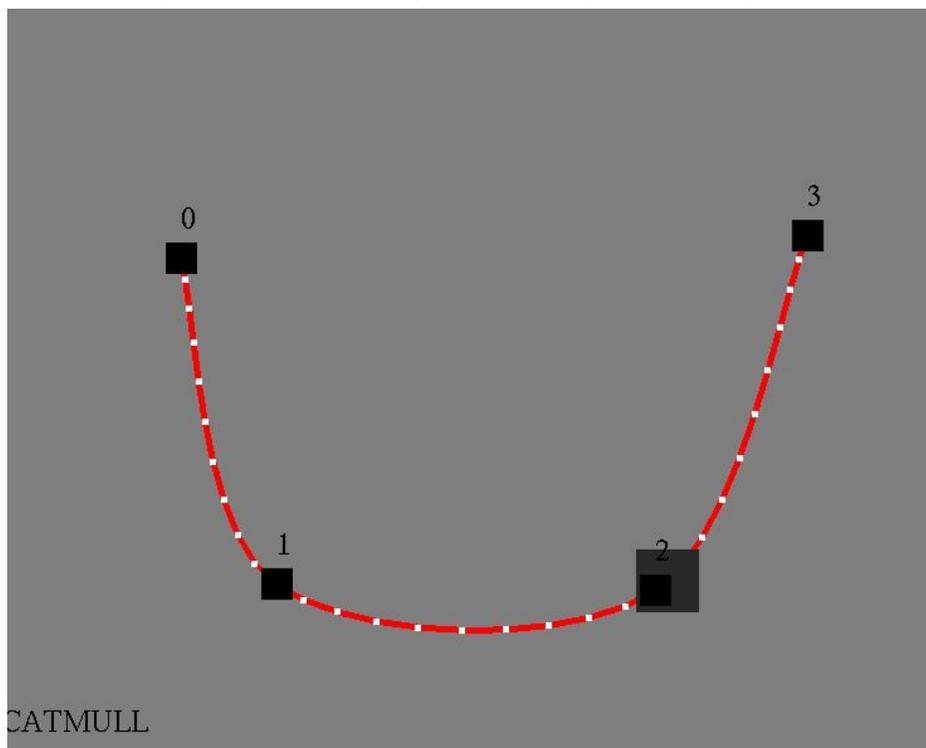
Make sure you put the names and student numbers in a comment at the top of each file!

Failure to follow these submission guidelines will result in a **zero**!

What's Next

Think about the following:

- Try implementing Catmull-Rom interpolation along the end segments, i.e:



- The segment between p_0 and p_1 can be calculated by using p_0 twice
 - $p' = \text{catmull}(p_0, p_0, p_1, p_2)$
- The segment between p_2 and p_3 can be calculated by using p_3 twice
 - $p' = \text{catmull}(p_1, p_2, p_3, p_3)$
- Implement the ability to save / load paths and you've got the beginnings of a path editor going!