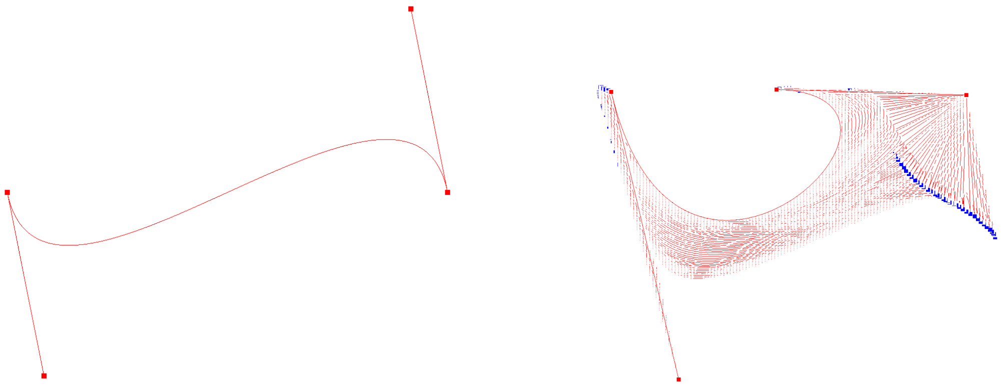
Karina Rigby
CSC305: Assignment 3
March 16

This project has two parts: the Bezier line generator and editor, and the simple animation.

Part 1: Bezier line generator:

To draw the Bezier curve, this program first defines four control points on the window. Draw Bezier generates the line pairs by receiving the four control points as parameters, and then performing **deCasteljau's a**lgorithm to find the next two sets of control points. The program runs recursively, and when it reaches the maximum depth it pushes the two points to be drawn onto the point list. (2%)



Before and after sample transformation

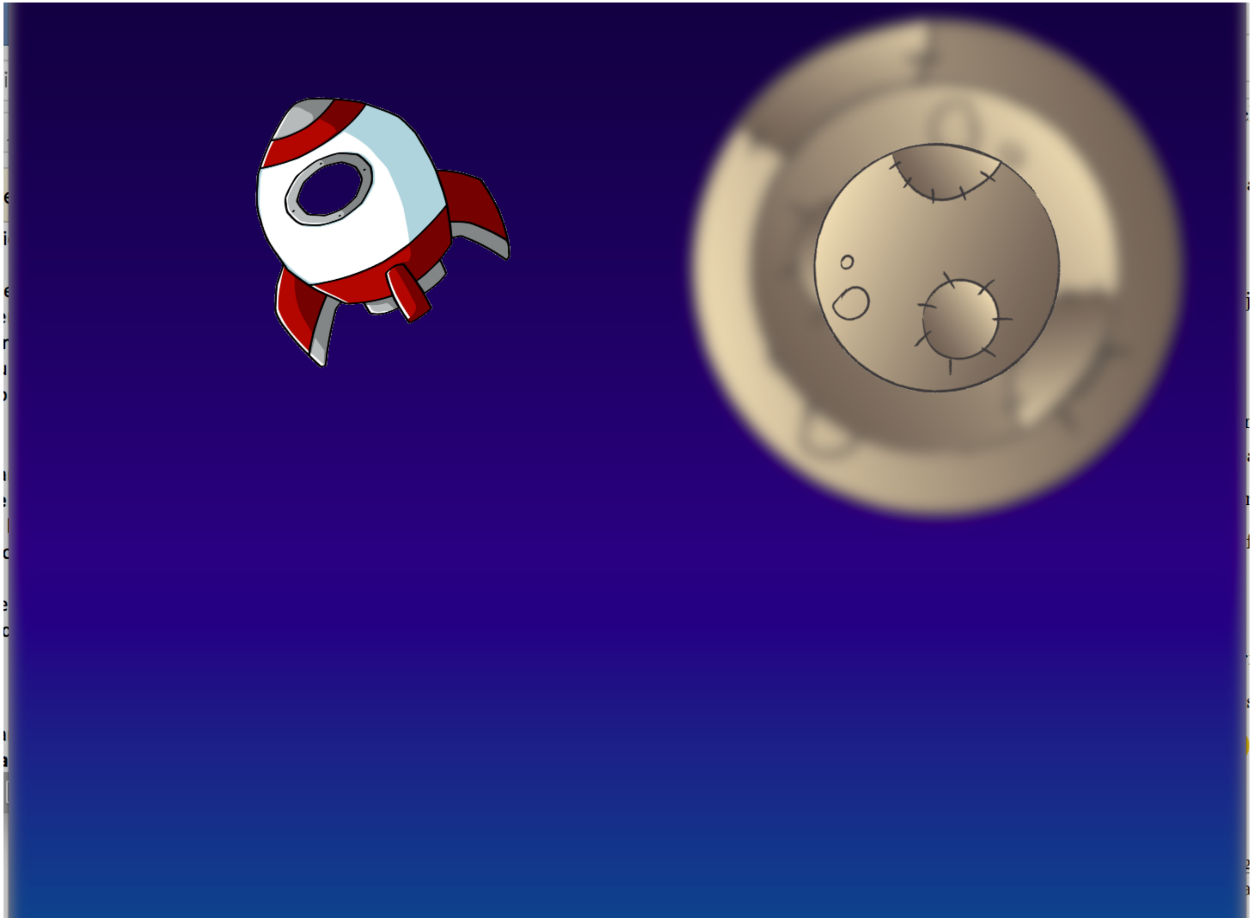
This program features mouse interaction, so although it has initial control points drawn on the screen**, the user can click any control point with their mouse and change it to a different spot. The Bezier line is drawn on the fly** – the user can see the Bezier line moving while they move any of the control points around.

To improve: Currently this program displays the lines from the previous frames. If I put a background texture, this problem would probably go away.

Part 2: The animation:

This program features a rotating moon and a rocket on a Bezier animation path. The moon has **transformation hierarchy**, in that one moon's transformation matrix dictates the other two moons. Two of the rings of the moons are rendered with Gaussian blur in **the fragment shader on the GPU**. (2-3%)

The program uses textures with **alpha-blending** – both the moons and the rocket are on quads and feature transparency. (2%)



The program has a **Bezier animation path** for the rocket, where the rocket translates and scales depending on which point of the path it's on. The animation path was computed in the same way as in part one, except the control points are not editable with the mouse.

I also **saved the animation as a MOV file**, (1%) and I saved my working progress on my git repository (1%).

Total: 80% + 8% = 88%