Flag Simulation

Summary and Goals
For my final project I decided to explore cloth simulation. I set several goals that I wanted to complete. Some of them included:

- Realistic Physics
- Lighting and Textures
- User Interaction
- Cloth Tearing

I felt the best way to demonstrate the above goals would to be displaying a flag in the wind. Physics would be applied by using gravity and wind to the flag. Users would be able to interact with the flag by clicking on it, tearing a hole into it. To satisfy these goals meant that the flag must be simulated in real-time. To achieve this I used OpenGL.

Why?
Flag simulation appealed to me for a few reasons:

- I enjoyed doing the early labs that involved spring constraints.
- Gordon’s video of a cloth being pulled impressed me.
- I had an idea of how cloth tearing could be achieved.

The Basic Process
The first thing I set out to do was create a grid of vertices that would represent my cloth. Each vertex has the following properties:

- A “pinned” boolean - This determines whether or not a vertex can be moved by constraints or forces such as wind and gravity.
- A “destroyed” boolean - This determines whether or not a vertex is visible and has active constraints.
- A Normal Vector - Used for lighting.
- A Position Vector - Used for positioning.
- An Acceleration Vector - Used for apply gravity and wind.
Once a grid of these vertices was complete I was able to add constraints. I used structural, shear, and bending constraints. Below is an image of my grid with only structural constraints added and the top corner vertices pinned. Gravity is applied but wind is not.

After adding shear and bending constraints I encountered a problem. The vertices would “explode”.

Applying several iterations to the function that satisfies constraints relieved the problem. More iterations made the cloth less fragile and more rigid.

From here it was easy to see unshaded lighting very obviously reduced the quality. To add shaded lighting meant that I would have to recalculate normals for every triangle each frame since position would be constantly changing. Wind also uses normals so both features were added at about the same time and included textures.

Cloth tearing was the feature I looked to most and was not too difficult to implement. For every constraint, if either of the vertices were “destroyed” then I would just skip it. It also meant that for
bending constraints I would also have to check the vertices between the two connected vertices. When users click or press the spacebar I delete all vertices within a radius around the position of their mouse.

What I Learned
This was a great project to learn more about constraints and spring physics. It was a lot of fun to experiment with the different things you can do. I also learned a lot more about OpenGL. I had the most fun implementing the cloth tearing because it was a concept I came up with on my own and did not need to use research material for it.

Problems I Encountered
The main problem I encountered was that when I tore the flag, some pieces would appear to be floating. To figure out what was going on, I made a function to display the springs of the flag. The problem turned out to be that the pieces were being held by a string of constraints… basically it was being held by a thread. To fix it I made it so that constraints would break when they are stretched too much since I felt that changing the visual geometry would be out of the scope for this project.

The Future
Some future research topics include:

- More user interaction such as pinning vertices and letting the user drag to move vertices.
- Different cloth shapes.
- Collision.

Research Links
1. Gamasutra - “Devil in the blue faceted dress”
   - Very good article for understanding cloth.
2. Cloth (Untitled)
   - Describes wind technique used.
3. Mosegaard’s Cloth Simulation
   - Further describes wind technique in addition to in depth constraint solving.