

Assignment 3

CMSC 427, Spring 2009

Due: 11:00am Tuesday, April 7 2009

This assignment involves simulation and rendering of fabric with a mass-spring system. For this assignment, you should make the following extensions to your program from the Assignment 2.

1. **(1 point)** Create a piece of blanket similar to how you created the waving flag in assignment 1, as a grid of 50 x 50 squares (you will want to try different values for this later, so set it using `#define`). Apply a texture to it. If you want, you can use the texture of the blanket that's already on the bed, like this:

```
glBindTexture(GL_TEXTURE_2D, 4); // (the texture ID for the blanket is 4.)
```

Scale the blanket to a width and height of two units, centered above the bed at a height of 1.5 units.

2. **(2 points)** Calculate the normal vector at each vertex using the cross product on the two sides of each square. To save calculations, you only need to calculate the length of the normal vector once, since all the squares are the same size. This means you only need to do one multiply (by the inverse of the length) per quad instead of a square root and a divide if you did it the usual way.

3. **(4 points)** Make the blanket into a mass spring system, with 6 springs from each node (the corners of the squares the blanket is made up of). For example, node (3,3) should have a spring connecting to node (3,4), node (4,3), node (3,5), node (5,3), node (4,4), and node (2,4).

To model a spring, you can do the following. At each time step, find out how far apart the two nodes joined by a spring are. If they are too far away, move them directly towards each other by 90% of the difference between how far apart they are and the length of the spring. If they are too close, move them apart in the same way.

Use the Verlet integration scheme for calculating position x_i of a particle at time step i , in which instead of: $x_2 = x_1 + x_{velocity}$, you can calculate it as: $x_2 = x_1 + (x_1 - x_0)$

With the former, the feedback between vibrating springs can grow larger and larger until the whole fabric explodes; the latter is inherently more stable.

4. **(1 point)** At the first time step, apply a small random force to each node to break symmetry. (This will wrinkle up the fabric a bit.) At each successive time step, apply gravity. Find a gravity constant so that the blanket falls slowly and gently.

5. **(1 point)** At each time step, check to make sure each node of the fabric hasn't passed through the bed or the floor. If it has, move the node back to where it was on the previous time step. In the reference executable, the floor is modeled as a plane at $z = -0.59$, the bed as a rectangle from (0.63, 1.0, 0.1) to (-0.63, -1.0, 0.1), the head and footboards as rectangles at (0.63, 0.9, 0.2) to (-0.63, 1.1, 0.2) and (0.63, -0.9, 0.2) to (-0.63, -1.1, 0.2), and the bedposts as spheres at (+/- 0.58, +/- 0.95, 0.55).

6. **(1 point)** If you've done everything right so far, you should now have a blanket that drops onto the bed and drapes over it. To make the demo a little more fun, use the `glutKeyboardFunc` to apply a force to each node of the fabric in any of the six directions when you press one of six keys. (w , a , s , and d to move in x and y and t and g to move in z direction.)