Technical Animation
Assignment 3: Cloth Simulation

Release Date: Wednesday, March 2
Due Date: Monday, March 28

Overview:
Cloth simulation has been an important topic in computer graphics since the 1980's. In this assignment, you will implement a particle system and use this system to simulate cloth based on the mass-spring model. You will also explore Maya's ncloth, a more complicated cloth system.

Requirement:
If you are using our starter code, a particle class and a generalized force structure is already in place. You must implement two subclass forces: GravityForce and SpringForce.

Arrange your particles properly and connect them to form a piece of cloth by adding: strench spring and damping forces, shear spring and damping forces, bend spring and damping forces

Deliverables:
Please submit the following:

1. A short video of cloth animated using your method
2. A short video of cloth animated in Maya
3. A webpage or powerpoint presentation describing your goals, techniques you used, and your results.

In addition, please submit your code, brief documentation on how to run it, and a list of features you chose to implement. Your code should be able to run on the Gates cluster machines.

Submit all files to your handin directory at
/afs/cs.cmu.edu/academic/class/15464-s11-users/<andrewid>/assign3/
Note that in order to access this directory, you may first need to obtain AFS authentication tokens:

```
> aklog cs.cmu.edu
```

Starter Code:

There is starter code built on the basic viewer from the previous assignments, located at
/afs/cs.cmu.edu/academic/class/15464-s11/asst3 handout/starter_code.tar.gz
The files we added to the basic viewer are located in the "Character" directory.

Particle.hpp, Particle.cpp
SpringForce.hpp, SpringForce.cpp
Force.hpp, Force.cpp
Cloth.hpp, Cloth.cpp

This code also contains an updated BrowserMode.hpp and BrowserMode.cpp, which loads the particles and springs into the viewer.

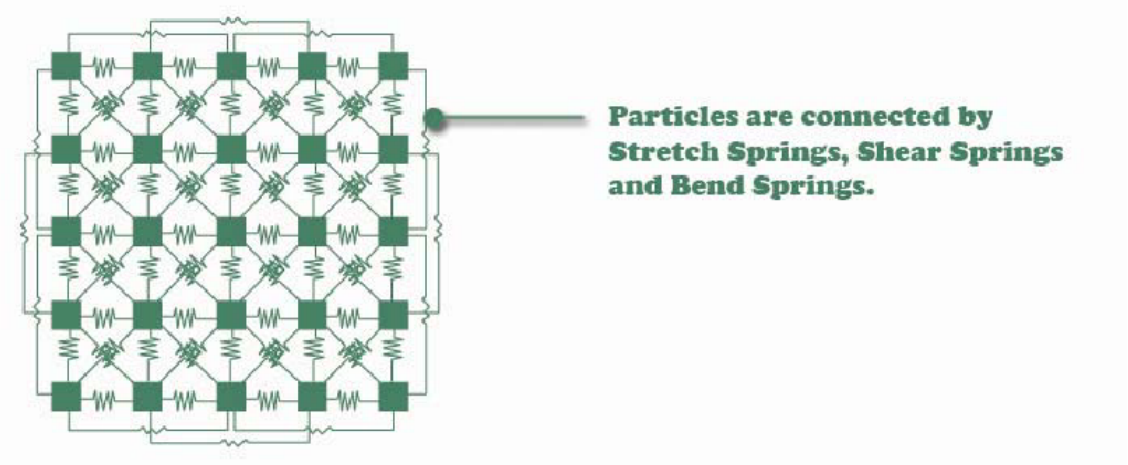
Implementation:

Simulating cloth has two steps: 1. Compute forces and acceleration for each particle 2. Integrate the acceleration to get the new cloth position

We will use the mass-spring model to model forces, and Forward Euler's Method to integrate the motion.

Mass-Spring Model:
The Mass-Spring Model is a very basic method to simulate cloth. It is relatively easy to implement, and with some care can produce good results, so it is a good exercise before exploring more advanced methods.

In the Mass-Spring Model, cloth is simulated as a grid of particles, which are connected to each other by spring-dampers. Each spring-damper connects two particles and generates a force based on the particles' positions and velocities. We will use three types of spring dampers: stretch springs, shear springs, and bend springs. In addition to these spring forces, each particle is also influenced by gravity. The layout of this model is illustrated in the figure below:



Computing Forces:
Gravity: gravity is simply a force porportional to the mass of the particle. We can use the equation:

$$f_g = mg$$

Spring-Dampers: A spring-damper is defined by three constants: its spring constant K_s, a damping factor K_d, and its rest length l_0. We can compute the force using the equation:

$$\begin{aligned} f_1 &= -\left[k_s(|p_1 - p_2| - l_0) + k_d\left(\frac{(v_1 - v_2)\cdot(p_1 - p_2)}{|p_1 - p_2|}\right)\right]\cdot\frac{(p_1 - p_2)}{|p_1 - p_2|} \\ f_2 &= -f_1 \end{aligned}$$

Euler Integration:
Once we've computed the all the forces in the system, we can compute the acceleration of each particle. Using Newton's Second Law:

$$a_n = \frac{1}{m_n} f_n$$

Where a_n is the acceleration of particle n, f_n is the net force acting on particle n, and m_n is the mass of particle n. After we compute the acceleration, we can use it to advance the simulation by a small time-step, Δt:

$$v_{n+1} = v_n + a_n \cdot \Delta t$$
$$p_{n+1} = p_n + v_n \cdot \Delta t$$

Note: Euler Integration is not stable, and may lead to erratic behavior if you use too large a time step. You may want to consider implementing a different method of integration.

Extra Credit:

- There are many other integration schemes you could implement, which may be more accurate and more stable than Euler's Method. Here are a few possibilities:
  - Runge-Kutta
  - Verlet Integration
  - Leapfrog Integration
- Collisions with other objects. You'll have to detect when a collision occurs, then perform collision response, such as by using a penalty function.
- Other forces, such as friction, drag, or wind.
- Other methods other than the mass-spring model.