15-863 Computer Assignment #2: HAIL STORM!

Instructor: Doug L. James

Due: Thursday, March 6, 2003

In this assignment, we will **create a hail storm and watch it squish a banana bicycle seat in slow motion.** The hail storm is modeled as a system of rigid particles that fall at a fixed velocity $\vec{v}^{hail}$. The deformable seat is modeled with a quasistatic Green's function (GF) model as in [3]. You will be given significant starter code (in Java) that includes the deforming seat's vertex-based GF matrix ($G = 1, \nu = 0.5$), as well as triangle models for the seat and its matching metal hardware (see Figure 1).

Figure 1: *Banana bicycle seat (before the storm)*

You must time-step the motion of each small hail stone, detect collisions with the seat, and integrate the gravitational and seat contact forces. Model the particle-seat interaction as a simple point contact constraint between the particle and the *surface contact point (SCP)* on the seat's triangle mesh (the SCP is identified by it's triangle and barycentric coordinate).

1. *Constraint handling:* This is a very strong hail storm, so there will be times when more than one hail stone hits the bicycle seat at a time. Assume that the hail stone's contact force is distributed barycentrically to the vertices of the contacted triangle. Use the Green's function matrix to enforce the multiple contact constraints to keep the particles in contact with the seat. (Note that this is slightly different from class where constraints were applied directly at the vertices. Initially, you might contrive things so that only vertices are hit.)

2. *Collision detection:* You need to detect collisions between the hail stones and the deforming seat. Implement a (simple) collision detection scheme for the deforming triangle mesh. Good candidates are spatial subdivisions, or a hierarchy such as a sphere tree [2] or an AABB-Tree [4]. For simplicity, and consistent with "slow motion," you can assume that the hail stones will only travel a tiny distance between frames

3. *Friction:* Most of the hail stones will hit the flat top of the seat and bounce off, whereas those hitting nearer the sides will tend to slide. Can you find a simple way to model particle sliding?

4. *Damping:* Assume that a damping force acts on the hail stone when it is in contact. This will lead to a restitution effect so that the stones won't bounce back up into the sky. Let the force be proportional to the velocity of the contact point. Can you suggest a better force model?.

5. *How many stones can you bounce at one time?* The maximum number of stones that you can handle is determined not only by the Green's function matrix multiply, but also by the collision detection and constraint handling.

6. *Hail accumulation:* If you're ambitious, once you've done the previous, you are only a hair away from building a basic snow simulator [1]. BONUS: Can you make it so that some of the tiny hail stones accumulate like snow on the seat?

References

[1] Paul Fearing. Computer Modelling of Fallen Snow. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 37–46, July 2000.

[2] P. M. Hubbard. Collision Detection for Interactive Graphics Applications. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):218–230, September 1995.

[3] Doug L. James and Dinesh K. Pai. A Unified Treatment of Elastostatic Contact Simulation for Real Time Haptics. *Haptics-e, The Electronic Journal of Haptics Research (www.haptics-e.org)*, 2(1), September 2001.

[4] Gino van den Bergen. Efficient Collision Detection of Complex Deformable Models using AABB Trees. *Journal of Graphics Tools*, 2(4):1–14, 1997.