Physically Based Modeling

CS 15-860 Spring 1994

Assignment Ω: Critters

Due by *Thursday* April 28

In this assignment, you'll simulate a simple 2-D mass-and-spring creature, consisting of a triangle with mass points at the vertices and (active) springs on the edges. You should be able to build this on top of the mass-and-spring program that you wrote earlier.

Here's the scenario: picture your 2-D world as a horziontal surface, e.g. a table top. The triangle critters use their edge springs as muscles, expanding and contracting them by changing their rest lengths. Each vertex has a "sucker" that it can use to grab onto the tabletop. When the sucker is on, the mass point at the corresponding vertex is frozen in place. When the sucker is off, the mass point moves according to the forces acting on it, including a drag force $\mathbf{f}_{drag} = -k_{drag}\dot{\mathbf{x}}$.

Your job is to write a control program that makes the critter locomote by varying the three spring rest lengths, and turning the three suckers on and off, in an appropriately coordinated way. Naturally, there are many ways to do this. Here's a very simple framework you can take as a starting point, based on the idea of periodic actions with phase shifts:

- Everything is synchronized to a *simulation clock* which is advanced by Δt each time step.
- Each spring is controlled by periodically toggling its rest length between two preset values, i.e. "short" and "long." The graph of rest length over time is a square wave, which you specify completely by giving its frequency, phase, and duty cycle, along with the high and low values.
- The same kind of square wave is used to turn the suckers on and off. *Variant: instead of an on-off sucker, toggle the particle's k_{drag} between high and low values. This should give you something that looks more like swimming.*
- The very simplest kind of control program consists of just the six sets of square wave parameters. You probably want all of them to run at the same frequency. Your main tool in getting coordinated behavior is the phase—remember the idea of the 1-D inchworm that grabs and lets go of the ground, while expanding and contracting.

At the very least, see if you can come up with two distinct locomotion "gaits." (There are many possibilities—e.g. I think you can produce a pleasant twirling "waltz" gait, although I've never actually tried this.) Try to go for something with visible dynamics—glides, springiness, etc.—rather than just a static-looking crawl.

Important notes: 1) You're going to need to play with the parameters to get this to work. That will be all but hopeless if you compile them into your code. At

the least, put the parameters in a text file that you can edit easily. Sliders are even better, of course. 2) It's important to get the square wave parameters, the spring stiffnesses, and the solver step size into a reasonable range. Things need to be moving quickly enough for you to see what's going on, and to make your creatures look lively and alive. The springs need to be strong enough to actually propel the creatures. The step size needs to be a lot smaller than the period of the square waves, but not so small that things grind to a halt.

Optional extra stuff. Try making steerable creatures, with settable speed and direction of motion. Try doing simple autonomous control hacks (e.g. making them bounce off the walls, chase/avoid the mouse, etc.) Try doing a vertical flatland, with downward gravity, and with (non-sliding) floor contact in place of suckers. See if you can make your creatures hop.