

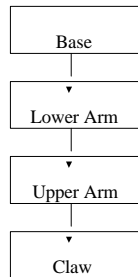
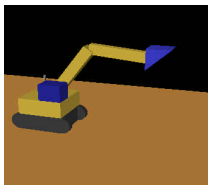
Animator Project

CSE 557

In this project...

1. Create your own hierarchical model
2. Create keyframe animations
3. Create *smooth* keyframe animations
4. Create particle systems

Hierarchical models

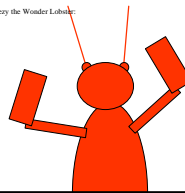


Hierarchical models

First requirement: make your own hierarchical model

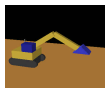
- model using simple primitives (box, cylinder, sphere, triangle)
- build up the hierarchy using OpenGL transformation commands

e.g., Wheezy the Wonder Looby



```
...body stuff...
glPushMatrix();
glTranslated(0.5, 0, 0);
glRotated(lArmAngle, 0, 1, 0);
...left claw...
glPopMatrix();
glPushMatrix();
glPushMatrix();
glTranslated(-0.5, 0, 0);
glRotated(rArmAngle, 0, 1, 0);
...right claw...
glPopMatrix();
```

Keyframe animation



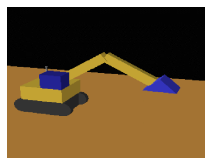
t = 0.0 sec



t = 2.0 sec

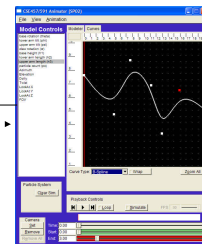
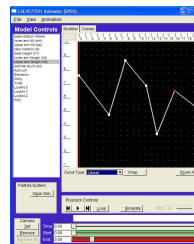


t = 3.0 sec



BEST ANIMATED SHORT
INVOLVING A BACKHOE

Animation curves



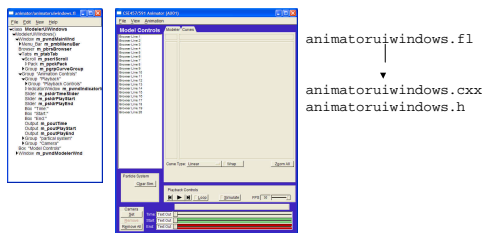
Animation curves

- You'll need to add three kinds of curves: Bézier, Catmull-Rom, B-spline
- For each, make a subclass of CurveEvaluator
 - given a set of control points, calculate many sampled points on the curve
 - allow each curve to be “wrapped” so that the curve is continuous if the animation is looped

Particle systems

- A particle system is a collection of point-objects with forces acting on them.
- We keep track of each particle's position and velocity. At each timestep, update the position and velocity based on the forces acting on the particle.
- You need to be able to *bake* your particle system; i.e., save the calculated positions of each particle at each timestep so that it can be easily replayed.

FLUID



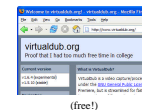
- laying out widgets in FLUID: ☺
- writing code in FLUID: ☹

Making movies

- The animator saves your movie as a set of sequentially-named .bmp files.
- To make a movie:



OR



...or your favorite other program...