

# CS 770/870

## Assignment 7: Primitive Animation

November 24, 2008

**Due: Monday, 12/8 by 23:59:59**

**Last allowed submission: 12/12 23:59:59**

The purpose of this assignment is to give you some experience defining and coordinating the animation of multiple shapes/objects. For this assignment, you will generate a short animated scene that includes at least one class involving pre-defined *articulated* animation, and shapes that animate along piecewise-linear paths provided in a file. You will also be required to provide a gui that allows for rotation and zooming of the camera or scene, as well as start/stop buttons (plus any other animation controls you would like to add toward the final 10 points).

### *File/command syntax*

In addition to the syntax that specified shapes in the previous assignments, you must also support the following syntax.

- 1) The piecewise-linear path will be specified using the following syntax. Note that this is a multi-line command that begins with “**new**” and ends with “**end**”. Each line after “**start**” provides the next waypoint along the piecewise-linear path, giving the (relative) time that an object should reach the point, and the (x, y, z) coordinate of the waypoint. In most cases, *time1* will likely be 0.

```
new path path_name start  
  time1 x1 y1 z1  
  time2 x2 y2 z2  
  ...  
  timen xn yn zn  
end
```

- 2) The way a file will specify how an object will follow a path is by telling the object to execute an “**animationPath**” command. This command tells the object which path to animate along (*path\_name*) and may also provide a *time\_offset* to be added to every time on the path. The *time\_offset* argument is useful in allowing several objects to follow the same path at different times. One way to qualify toward the last 10 points is to also accept (and demonstrate) a *time\_scale* argument (perhaps having one shape slightly faster than the other, trying to chase it). Note that your program should start with a simulation time at or near 0.

```
tell shape_name animationPath path_name [time_offset]
```

- 3) The way a file will specify how a camera will follow a path is by telling the camera to execute an “**animationPath**” command. This should cause the camera’s eyepoint to follow the path. It is up to you which direction the camera looks, but you can qualify toward the last 10 points if you make the camera always “look” in the direction the camera is moving.

```
tell camera animationPath path_name [time_offset]
```

***Point allocation***

- 35 Ability to animate an articulated object in-place. For full credit, the object must consist of at least 2 sets of 2 joints on which motion is occurring. Examples for humanoid objects would be running, swimming, hitting a baseball, etc.
- 20 Ability to animate an object along a piecewise-linear path specified in a file.
- 10 Ability to animate an articulated object along a piecewise-linear path specified in a file. The articulated object must be doing its articulated animation smoothly across the whole path.
- 10 Ability to animate a camera along a piecewise-linear path specified in a file.
- 15 Ability to animate multiple objects without them interfering with each other. You must provide some way to stop individual animations mid-way through so that we can see the effects when all objects are animating versus when only some are animating.

Features that might qualify for the last 10 points include but are not limited to things like additional animation controls (speeding up/slowing down, looping, etc.), smooth curve paths, collision detection with some response (stopping or bouncing), and/or simulated physics animations.