Homework 6

Software Design **Allen B. Downey**
Spring 2008

Due: Tuesday 4 March
The reading for this assignment is Chapters 13–15 of *How to think...*.

6.1 Turtle Tag

1. Download `Tagger.py` from the class web page:

   ```
   wget http://wb/sd/code/Tagger.py
   ```

   It contains code very similar to `Wanderer.py` from Homework 2. You might find that it makes more sense now :)

   The object of this homework is to make the turtles play tag. When the program starts, it will choose one of the turtles to be "it". This turtle will chase the others, and when it catches one, it tags the other turtle. The other turtle becomes "it," and the game proceeds.

   Some additional rules are:

   (a) Turtles have to stay in bounds. You can use an algorithm like what you used on Homework 2.

   (b) After a turtle is tagged, it has to sulk for a while before it can start chasing the other turtles.

   (c) It should be possible, eventually, for the slow turtle to tag the faster turtles.

   When you run `Tagger.py` you will see that there are buttons in the control panel labelled Run, Stop, Step and Clear. Press Run to start the game and Stop to call a time-out. When you press Step, the World object invokes the `run` method once on each turtle. This feature might be helpful for debugging.

   Here are some suggestions for methods you might want to have in the Tagger class:

   - Compute the distance to another turtle.
   - Turn toward another turtle. HINT: see the documentation of `math.atan2`.
   - Find the closest other turtle. Note: the `world` object has an attribute named `animals` that is a list of all the Turtles currently in the world.
   - Tag another turtle.
   - Change color (maybe the turtle who is "it" should be red and a sulking turtle should be blue).

   WARNING: Turtles already have an attribute named `tag`, so you should not create a method named `tag`.

6.2 Turtle Segregation

This is a JFFE.

1. Using `Tagger.py` as a template, create a file named `Bigot.py` that defines a new kind of Turtle called `Bigot`. Create about 100 red Bigots and about 100 blue Bigots at random locations in TurtleWorld. Let them run around for a while. They should mingle without regard to color.

2. Now change the `step` function for Bigots so that each one finds the nearest 5 neighbors and checks their color. If two or fewer of the neighbors is the same color, the Bigot gets uncomfortable and speeds up.

3. Play around with the parameters and see what effect this behavior has on the distribution of Bigots in the world. Parameters you can vary include the comfort threshold (how many same-color Turtles does a Bigot need to feel comfortable) and the speed (how much does it speed up). You also might vary the number of colors Bigots come in.

This experiment is based on Schelling's model of segregation. If you Google those terms, you will find lots of information and some demonstrations. We study this model and other discrete models of physical and social systems in Computational Modeling, which will be offered in Fall 2008. The only prereq for CompMod is Software Design.