CIS 716.5 Spring 2010, Project II

Description

This project builds on the last one. Instead of having the interaction between agents being a choice about what t-shirt to wear, we will have the agents interact by playing a simple two-player game. We will combine this two-player game with the same kind of learning that we saw in the previous project. Here agents will learn what strategy to use when playing the game.

Implement this game

Write a Netlogo simulation in which agents play a simple two-player game like the Prisoner's Dilemma.

Each agent plays the game using some strategy (by "strategy" I mean something like Tit-for-tat) and initially each agent randomly selects which strategy to use.

After that, the game proceeds in rounds. On each round every agent is paired up with exactly one other agent, and the agents play the game. Depending on how both agents play — and on the game that they play — each agent gets a score. Agents total up their scores over time.

After some number of rounds, agents make a decision about whether to change strategy.

Things your simulation must do:

1. The simulation must include at least two games, one of which is the Prisoner's Dilemma (for the other, you can use any of the games that we talked about in class, or any two-player game that you care to invent).
2. The simulation must include at least three strategies, and one of these must be Tit-fot-tat.
3. The simulation must include some decision making about which strategy to use and this should be based on the scores that different agents obtain through playing their strategy. You can base this on things you implemented in the t-shirt example, or you can invent a completely new mechanism.
4. The simulation must update *on ticks* (see the Netlogo manual to get a full explanation of this), and it must display a count of the total number of ticks that have elapsed in a simulation.
5. The simulation has to show the number of agents with each strategy at each tick in the simulation window.
6. The simulation must display the number of agents playing each strategy
7. The simulation must stop if all the agents end up playing the same strategy.
8. The simulation must allow the user to pick the proportion of agents that start playing each strategy.

Aside from this, I don't mind how you solve the problem (and the more, different, solutions I get, the better :-)

If you want a hint, look at the picture on the webpage — that is a screenshot of my solution when the agents are playing the Prisoner's Dilemma. This time around the agents didn't move (I did everything with patches), but it is also fine to use agents that move.

Document

All you are going to hand in is the Netlogo program so make sure you:

1. Write lots of comments in your code. If I don't understand what your code does, you won't get full credit.
2. Write a description of your program in the Information tab. Here you should write an overview of the problem, the strategies for playing the game, and the ways in which agents make their decision about which strategy to use.

Experiment

Once your simulation is complete, experiment with different initial proportions of strategies, and different games, seeing whether the agents all converge on the same strategy and if they do how many ticks it takes. Since there is randomness in the simulation, you'll need to do several runs (say 10) of each combination in order to get a reliable idea of how they perform.

Describe the results of your experiments, including the statistics you collect, in the Information pages of your project.

Hand it in

Save your model as `<my-name>-pd.nlogo`, where you replace `<my-name>` with your own name (so my program would be called `parsons-pd.nlogo`) and email it to me at `parsons@sci.brooklyn.cuny.edu`.

The subject line of your email should say: 716.5, Project 2.

If you don't get an acknowledgement within 24 hours, send me a follow-up email.

The due date for the project is May 16th.