CS 6100 Program 2 Game Theory Tournament (40 points) Due February 4, 2010

The Big Picture
We want to have a tournament of all types of agents playing against each other to see which is the best.

Consider one interaction as follows: Two agents (having the same or different strategies) will play each other TRIALSIZE times in one interaction. They need to repeat their play with each other, as some strategies make are ineffective if they just play each other one.

We define the interaction as follows:

OneInteraction
T1 = pickStrategy() // Select the strategy for agent 1
T2 = pickStrategy() // Select the strategy for agent 2
Play the two agents against each other TRIALSIZE times.
T1AveScore = totalPointsEarnedT1/ TRIALSIZE
T2AveScpre = totalPointsEarnedT2/ TRIALSIZE
Record average scores as follows
  summary[T1].interactionCt++
  summary [T1].score+=T1AveScore
  summary [T2].interactionCt++
  summary [T2].score+=T2AveScore

Using the concept of one interaction, run a tournament.

Tournament
Set Up Game Characteristics (what is the mix of agents in your world)
Repeat COUNT times
  OneInteraction (picking agent types from your world)
//Show results
  for (i=1; i <=8; i++)
    cout << "Agent " << i << " scored " << summary[i].score/summary[i].interactionCt

The details
Create an applet (and place it on CSILM) which allows the user to control the input for the following experiment.

Description:
You will conduct a tournament that tests how different strategies perform in repeated-play prisoner’s dilemma game with a variety of other agents. You will allow agents of the following
  1. An agent who always defects
  2. An agent who plays a mixed strategy (randomly picking cooperate half the time and picking defect half the time)
  3. An agent who always cooperates
  4. An agent who employs the tit-for-tat strategy (see below), and
  5. An agent who employs the tit-for-two-tats strategy (see below).
6. An agent who uses the Pavlov strategy (see below).
7. An agent who uses the Win-Stay/Lose-Shift strategy (see below).
8. Your own strategy

The user will select:
- the chance that pickStrategy will select an agent of a given strategy
- TRIALSIZE: how many repeated games are played with the same partner. Assume the number is unknown to the agents themselves
- COUNT: How many times we have an interaction between different pairings of agents.

You should try out how well each strategy works with a trialSize of 5, 100, 200.

Total Score for each agent type in the world will be recorded.

You should use the prisoner's dilemma payoff matrix as shown below with 4 the most preferred outcome and 1 the least preferred.

<table>
<thead>
<tr>
<th>P1/P2</th>
<th>Cooperate 2</th>
<th>Defect 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperate 1</td>
<td>(3,3)</td>
<td>(1,4)</td>
</tr>
<tr>
<td>Defect 1</td>
<td>(4,1)</td>
<td>(2,2)</td>
</tr>
</tbody>
</table>

In the payoff matrix, cooperate means cooperating with the other prisoner by remaining silent, and defect means defecting from the team strategy by confessing to the police. If you play the game 100 trials, your score is the sum of all of the payoffs you received so, for example, if you and your opponent each cooperated, then your score would be 3*100. High scores win.

Present your results in a helpful fashion. Spend some time figuring out which tests are most interesting. Answer the following questions (at least).

a) If all 8 agent types are equally likely in your world, which agent type does the best? Why?
b) Axelrod said that tit-for-tat was the best. Do you get the same results? Why or why not?
c) If the mix contains only your agent type and one other, which agent does best? (Actually this is seven different questions. Who does best if it is you against each of the other types, pairwise.) Do allow two agents of the same type to play each other. It's like you go out into the world and randomly pick a partner (who could have your strategy or some other, depending on the mix in the world).
d) What mix of three or more agent types seems to make your agent type do the best? Is your agent sensitive to the mix?
e) If you only have tit-for-tat and all-defect agents in the mix, who wins? Again, do allow two agents of the same type to play each other.

**Strategies**

In the **tit-for-tat** strategy, the agent begins by cooperating, and then plays whatever strategy the other agent played. For example, let P1 be an arbitrary agent and let P2 be the tit-for-tat agent. Then a sequence of choices might be something like
Notice how the second player started off cooperating, and then just copies the previous choice by P1.

The tit-for-two-tats strategy is similar, except that P2 only defects if the other agent defects twice in a row, but cooperates immediately after the other agent cooperates. So, if I let the sequence (A1,A2) represent the past two choices from the other agent, then Tf2T plays the following strategy:

Thus, the following example represents a proper response for a Tf2T agent.

The Pavlov strategy begins by cooperating. On subsequent turns, whenever the two agents didn't agree on the previous last play then the Pavlov agent defects; otherwise, the Pavlov agent cooperates. So, if I let the pair <A,B> represent the choices that the two players made on the last round, then Pavlov plays the following strategy:

This says that if we cooperated last time then I'll cooperate this time. If we both defected, I'll take a chance that you might want to cooperate this time. If you defected and I cooperated, I'll defect next time. And if you cooperated and I defected, I'll try to get away with defecting again this time. An example sequence of how Pavlov would play against a made-up agent is shown below.

The WinStay/LoseShift (WSLS) strategy begins by cooperating, but then changes its behavior (C goes to D or D goes to C) whenever the agent doesn't win. Winning occurs whenever the agent gets either its most preferred or next most preferred result. For example, let P1 be an arbitrary agent and let P2 be the WSLS agent. Then a sequence of choices might be something like
Whenever WSLS wins (CC or CD --- where P1’s choice is listed first and P2’s choice is listed second), the WSLS agent replays its choice (C or D, respectively); i.e., it stays with the choice. Whenever WSLS loses (DC or DD), the WSLS agent changes its choice (D or C, respectively); i.e., it shifts its choice.