Initial Observations and Payoff Matrix

The report is divided in three pages: the first reads in a sort of chronological story of my observations and modifications to build my player. The second describes my agent. Last page covers agent evaluation.

The problem given to us is a spin on the two-player prisoner's dilemma. With the payoff values given to us in the three player prisoner's dilemma, I am able to create the following matrix:

	(Player 2, 3)			
Player 1	(C, C)	(C, D)	(D, C)	(D,D)
Coop (C)	(6, 6, 6)	(3, 3, 8)	(3, 8, 3)	(0, 5, 5)
Defect (D)	(8, 3, 3)	(5, 0, 5)	(5, 5, 0)	(2, 2, 2)

*Note: (Player 1, Player 2, Player 3): (Payoff x, Payoff y, Payoff z)

We play the game as Player 1. We notice that we can maximise our payoff by Defecting, following the nasty strategy. This is because whenever we Defect we get a better payoff than when Cooperate. Therefore, our dominant strategy is to Defect in all scenarios. The dominant strategy equilibrium lies in (Defect, Defect, Defect) with all the players defecting.

I further check for the Nash Equilibrium which is all players defecting, since, in this scenario no player would be better off by changing their moves. In every other scenario, one player can benefit by changing their strategy from Coop to Defect. The Nash Equilibrium is always the Dominant Strategy Equilibrium, however, the opposite is not true.

We can further find the cases for Pareto Optimal and those that maximise Social Welfare. The Pareto Optimal is when no player can become better off without it diminishing the payoff of another player. The Pareto Optimal in this scenario is all the strategies except the dominant strategy, where all participants defect or when two players defect. Since, when all players defect, or two players defect every agent can cooperate and still get a better payoff. In other words no one will be worse off by changing their decision. The Pareto Optimality therefore is for two or three agents to cooperate.

The social welfare strategy is for all three players to co-operate. In this scenario the payoff for all the agents sum up to 18. We can reason this out as follows: all get equal sentences, and collectively as a unit they are better off than if they had chosen the nasty route.

The Catch

One might wonder, why we need to carry on after having found our dominant strategy. The issue is that there are occasions where one agent is all three of the players or two of the three players. If we follow the dominant strategy in this scenario, the social welfare is only 6. The lowest that can be obtained for social welfare. This, subsequently, hurts our total score in the competition greatly. In other words, when we are competing with ourselves, we are focusing on Social Welfare rather than dominant strategy. Therefore, one way to summarize the challenge is as follows: follow the Coop strategy when you are competing against one or more of yourselves. Lastly, when competing with two agents of other players follow the dominant strategy. The issue with this is, we are not aware of the scenarios where we are competing with our own player. Subsequently, in the following page I to design a strategy which tackles this issue and tries to follow the best strategy.

Agent Description

The observations above lead to the following agent design, where we try to predict if one or or more of agents we are playing against is our player. To be able to do such a prediction, we use the help of opponent history and our own history.

The Agent Architecture is as follows:

1) Iterate through all of the opponent history
    a) If both opponents have played a different move than my agent, we just defect. All three agents have to be different and we can use our dominant strategy: defect.
    b) If one of the agents has played a different move, we count the number of times they have played a different move and how many of those have been defect.
2) There are a couple of possibilities left at this point
    a) The three agents are different but just getting the same history
        i) In which case we want to use our dominant strategy: defect. (Payoff: Either 8, 5 or 2, opponent gets 3, (0 or 5) or 2).
        ii) To prevent the above scenario where we keep cooperating since the histories are the same with rival agents I implemented the following:
            (1) Play Defect in round 10 instead of cooperating even if the histories are the same
    b) There are two agents which are same and one is different
        i) Since we have two agents, we want to try and maximize our social welfare since the ranking is based on points. Coincidentally, we follow the Paleto Optimality:
            (1) Cooperate when the opp agent (not ours) defects (Payoff = 6, opponent gets: 8)
                (a) If the agent defects and we also do (Payoff = 4, opponent gets: 2)
            (2) Cooperate otherwise (Payoff = 12, opponent gets: 6)
                (a) If we defect when the agent cooperates: Payoff = 10, opponent gets: 0
        ii) As we have seen we don't need to guess if one other agent is the same and the other's different. We can just leave it to the end since all of the above strategies are to cooperate.
    c) The three agents are ours
        i) In this scenario we want to play the strategy which maximises social welfare which is to cooperate (Payoff: 6+6+6 = 18)
3) As we can see, after we iterate through all of the opponent's history, we maximise our payoff by cooperating regardless of whether we are playing against 2 or three of our agents. So in the end we just return 0, and cooperate.

Next page covers agent evaluation.

Agent Evaluation

To evaluate the agent, I modify the run tournament to run a thousand times instead of just once. I then take a total score of all the tournaments and additionally I create a ranking points. Ranking points are based on your rank in every tournament, so in tournament i, if you're first place you get 0 points, second you get 1 point and the nth ranking gets n-1 points. After the 1000 tournaments, my agent is able to rank first in all of them which can be seen by the 0 ranking points. The agent also gets the highest overall score: 176009.8 points and in the screenshot we can also see the result of the one tournament where the agent scores 177.45964 points which is way more than the second play with 154.5732 points. I do note, however, that there are edge cases where if I was to increase the tournaments to 10000 I loose one or two tournaments. However, it has always been second place even in those where I don't come first. This is probably because in those cases agents like freaky player get extremely lucky and are able to score very high through pure randomness and not skill. One such scenario is while we try to aim to maximise our own points, with our strategy we end up giving the opponent some points as well. So cumulatively we supply some points to a player who ends up performing extremely well (through sheer luck) against the rest and winning.

```
NicePlayer: 140.01851 points.

Tournament Results
agarwal_samarth_Player: 177.45964 points.
TolerantPlayer: 154.5732 points.
RandomPlayer: 151.05472 points.
T4TPlayer: 148.29945 points.
NastyPlayer: 146.51527 points.
NicePlayer: 146.40216 points.
FreakyPlayer: 141.12003 points.

Cumulative Tournament Results
NicePlayer: 138630.6 points. 5108 ranking points
NastyPlayer: 147529.25 points. 3099 ranking points
RandomPlayer: 141301.47 points. 4522 ranking points
TolerantPlayer: 151535.69 points. 2047 ranking points
FreakyPlayer: 143978.69 points. 3984 ranking points
T4TPlayer: 150716.06 points. 2240 ranking points
agarwal_samarth_Player: 176009.78 points. 0 ranking points
(base) Parkhis-MacBook-Pro:assignment2 parkhiagarwal$
```