# Computational Game Theory

## 1st Tournament - Prisoner's Dilemma

| | Cooperate | Defect |
|---|---|---|
| **Cooperate** | 3,3 | 0,4 |
| **Defect** | 4,0 | 1,1 |

**Pedro Pais nº 48247**

# Introduction

This is a report on the strategy used to play through the first Tournament of the Computational Game Theory Class.
The first tournament was based around the Iterated Prisoner's Dilemma game, where the incentive is to Defect (see the Normal Form Game in the cover) but you end up gaining more utility as both Cooperate.

# The Strategy

In a single round, the only rational move is to Defect. Selecting the other action will most likely give out a utility payoff of 0. In a system such as this tournament where, besides playing the game several times with one other player, you play the game with the rest of the class and the score is measured in average payoff, the strategy changes.
In this specific game, the players gain much more utility if both Cooperate compared to defecting. While Defecting gives an initial payoff of 4 and (most likely) a payoff of 1 the rest of the rounds, both players Cooperating nets a payoff of 3 every round. This is a much greater increase than Always Defecting.

I assumed that most of my colleagues would not think of looking for a point in the game where it pays off to Defect.

This said, the strategy I chose to implement was the Grim Trigger strategy with the possibility of Defecting when it is worth to do so (if the probability for the next round is very low, for example), assuming that both players always Cooperate. This lets me Cooperate most of the time, earning the most utility I can, while not getting Defected on more than once. However, as soon as my opponent Defects, I stop trusting him and default to Defect in order to bring their score down.

# The Games

The first games were already pre-announced. 20 rounds with full probability of proceeding to the next round. I was ill-prepared for this one as most of my colleagues used the same strategy as me and the ones who defected one or two rounds sooner got the upper hand.

The second games were played with a high probability of proceeding to the next round and an "infinite" (a big number) of rounds. For this game, most of my colleagues played the same strategy but one player used a strategy that relied on randomness and it made some changes in what would be the ranked chart. Most of the differences in ranks could be traced back to it.

The third games were more interesting, they had only two iterations with a high probability of proceeding to the next round. Here, most of my colleagues only took the probability into account, with no regard for the iterations limit. In this case, the only difference between me and the high ranked players was, again, the randomness. Not regarding other players' strategies but regarding the number of iterations. Some matches ended sooner than others which and in these cases, some ended in a Defect - Cooperate matchup, leading to a higher average for those matches.

# Conclusions

After the tournament, I quickly realized that I underestimated my opponents, causing me to lose several points. Also randomness affects these games to a small degree but which caused real changes in the ranking table, perhaps due to a reduced number of rounds. There was only one colleague who was "smarter" than everyone else, Defecting in the third round before the last, expecting everyone to Defect in the last round. He got ranked high in the table because of this.

1st Games strategy used:
https://github.com/pgpais/TJC-48247-1st-Tournament/blob/master/FixedRoundsStrategy.java
2nd Games strategy used:
https://github.com/pgpais/TJC-48247-1st-Tournament/blob/master/ProbabilisticRoundsStrategy.java
3rd Games strategy used:
https://github.com/pgpais/TJC-48247-1st-Tournament/blob/master/FixedRoundsWithProbabilityStrategy.java