

Simulation Games

Duration of the project: 6 Months

Project Mentors: Mihir Mangesh Ketkar, Sudarshan Sundarajan

Team Members: Hari Chetan, Priyanka Gawande, Sudhanva Nadiger

Introduction (abstract)

In this project we aim to simulate and analyze various games like Ludo and Mafia and understand various strategies that can help beat the game. Along with that, we aim to build our own simulation game, wherein with partial information, a game is simulated and accordingly a player wins the game.

Inspiration:

The following videos were our inspiration for coming up with the idea.

- <https://www.youtube.com/watch?v=ubQXz5RBtU>
- <https://www.youtube.com/watch?v=Vv9wpQIGZDw>

Method

The project started with the team members going through resources and practicing basic codes to get acquainted with python programming, Numpy and Pandas libraries function and concepts like Markov chain, basic game refinement theory.

For Phase Two, the Snake and Ladder game was simulated using Markov's chain model. Using this Snake and Ladder model we calculated the probability distribution of finishing the game in say k steps. A reference paper on Game refinement theory was also discussed among us.

For the next phase, the team was divided into two. While one set of members simulated Ludo, another set of people simulated Mafia using python. For Ludo, first, a single-player game was implemented using the Markov chain model, then more players were added and the dice roll concept was used for simulating the game. Four different strategies i.e. Cautious Greedy, Cautious, Careless Greedy, and Random were simulated and the winning percentage, the average length of the game to win, and the pawns captured or lost by each strategy in an average give were calculated. While for Mafia, using the concept of classes and objects, each type of player was defined separately and given properties on the basis of their skills. Then the game was simulated for a particular composition for some thousand rounds and the GR value was calculated and the most ideal composition on the basis of these results was also found.

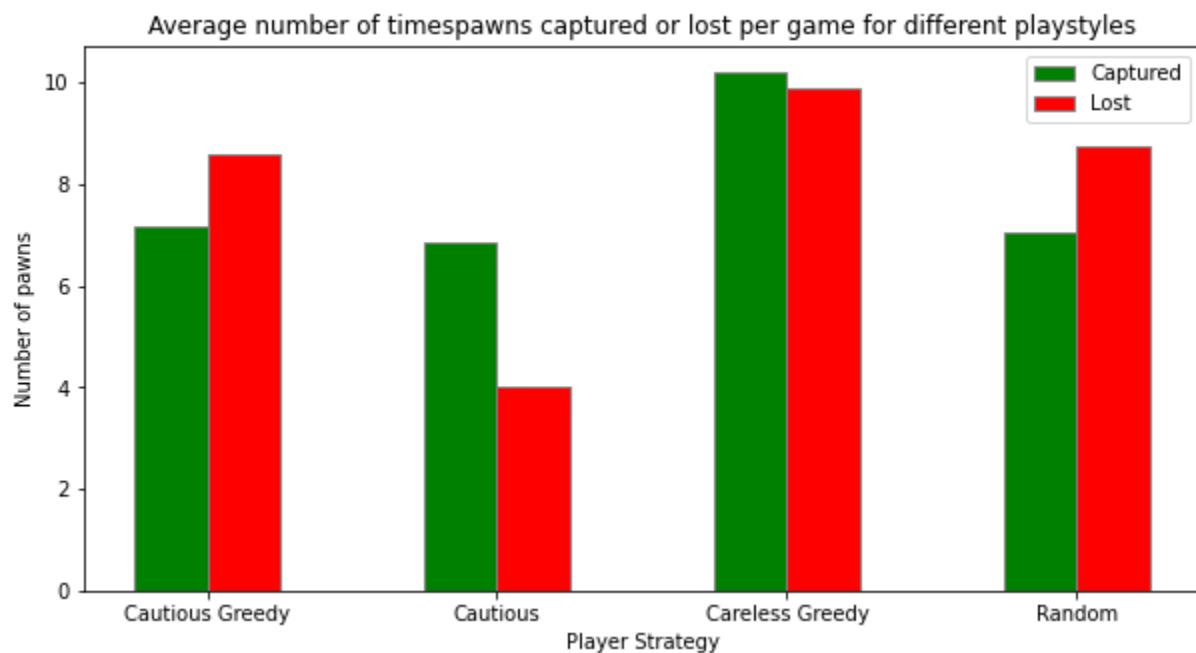
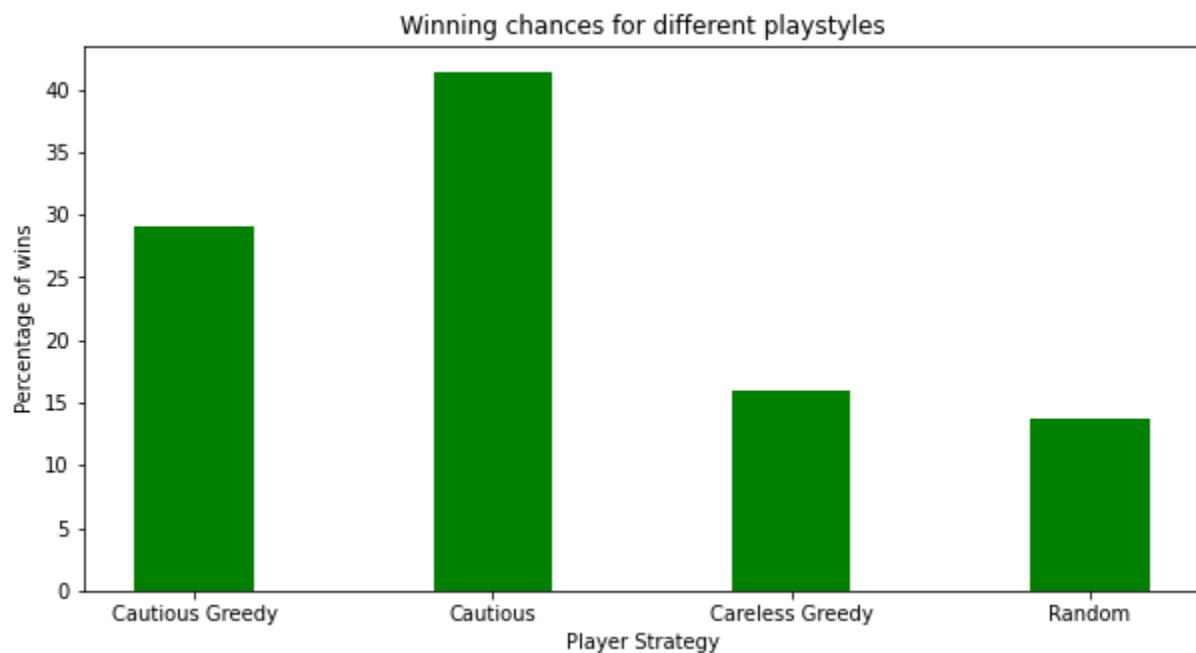
For the final phase of the project, a new game was created from scratch and then coded using Pygame. The new game is as follows:

1. Suppose the number of players is N
2. The game has 10 rounds (may change on the basis of N)
3. For each round, a company will be selling its stocks. The number of stocks will be the floor value of $N/2$.
4. Each player will be given the same amount of money at the beginning of the game. In each round, they will do secret bidding.
5. The game master will sell the stocks to the upper half of the bids but the value of the stock will be the lowest of the winning bid
6. So while the lowest of the winning bidders will stay even, other winning bidders will lose money.
7. After every three turns the person with the lowest value gets to roll a pair of dice and based on the difference of the numbers that showed up on both the dice and the predefined set of rules, an event occurs and the value of money (and not stocks) decreases by five percent.
8. After all the rounds are complete, the person with the highest amount of money and stocks wins the game.

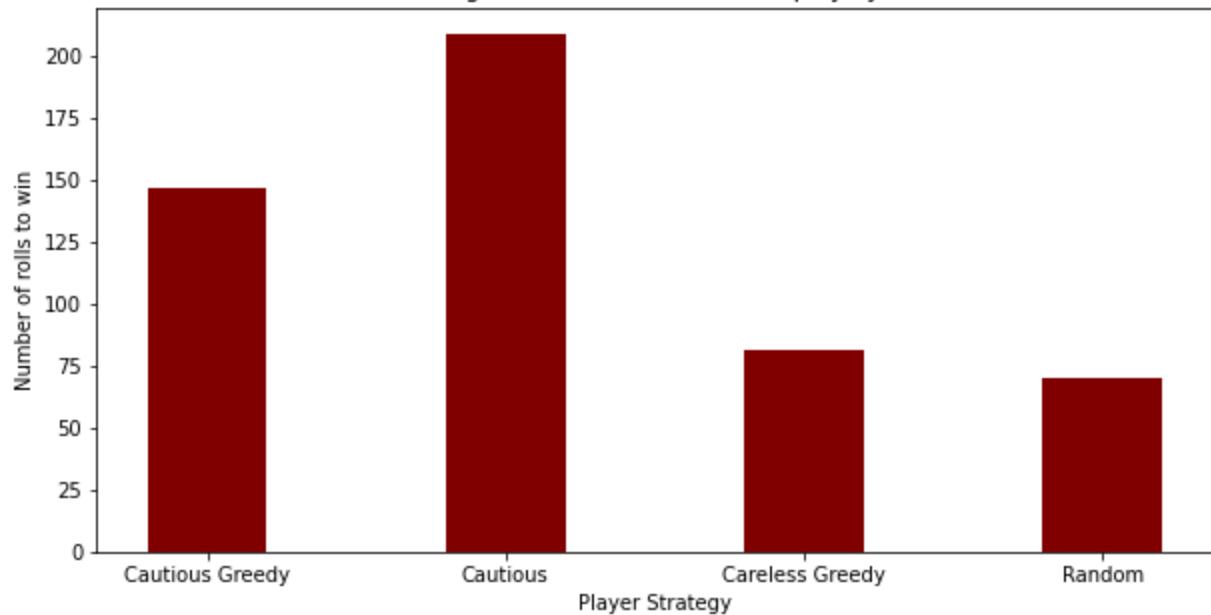
Results:

Game	Results obtained
Snake And Ladder	The average number of moves to complete the game according to our simulation came out to be 65.57
Mafia	By calculating the number of citizen wins, Average death, Rounds, Average Length, and hence the GR value, it was deduced that MFG(15,5,1,1) is the best composition to play Mafia. {MFG(total, mafia, doctor, sheriff)}
Ludo	Out of the four strategies, the winning percentage is highest for Cautious Player (43.5%) followed by Cautious Greedy (~30%), Careless Greedy (~15%) and Random (~12.5%). But the Cautious Player is also the slowest to win the game at ~200 rolls to win while ~70 rolls for Random.
Final Game	The single player version is ready, working on the multiplayer part

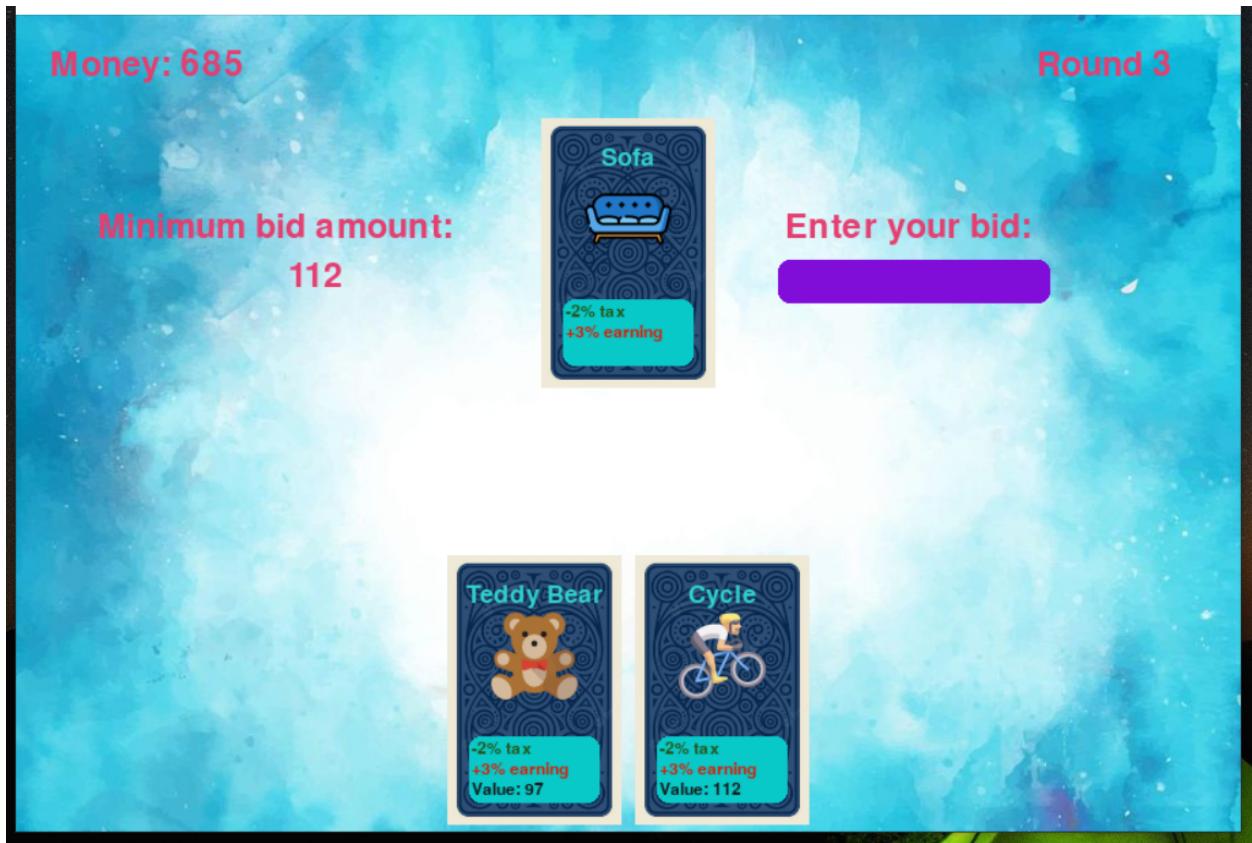
Results of ludo analysis



Average rolls to win for different playstyles



Photos of the Game:



Money: 657

Round 3

Roll Dice

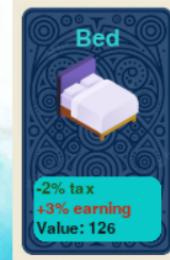


Money: 657

Round 3

6 1

Stock Market Crash



Obstacles faced:

The project started with revising basic python. As we all had experience with python programming before, it was easier. Markov Chain theory looked a bit overwhelming at the start but watching different videos about it made it easy to understand.

The main project started with simulating Snake and Ladders. Implementing the Markov Chain was fun. Printing the random path that we got each time we ran the code was exciting. The only obstacle that we encountered was understanding the Mathematics behind the calculation of Probability distribution which was explained by the mentors whenever required.

For the second phase, the Mafia game had a lot to do with classes and subclasses. There were some small errors regarding the code syntax that we encountered while running the python code but after reviewing the code, they were solved. The Mathematical element in calculating the best composition for having a fun game i.e. having GR value in the most desirable range was a bit complex but the research paper provided and mentor helped a lot. There was some difference in the results obtained and the results the research paper showed which was later on alleged to be because of the difference in the rules that were implemented by us and the rules that the paper used.

For the second Phase, at first, it was hard to understand the strategies, how to implement these strategies and how to build the game, but then after guidance from mentors, the different strategies and how these strategies are related to each other became more clear. And this in turn made simulating these strategies fun.

For the Third Phase, we discussed some ideas for the final game, which was a hard task on its own, and we settled on a final idea. The mentors were very receptive to any ideas we had presented, and we included them as well. The last game required Pygame so some time was spent learning that. It was altogether a very different coding experience. The mentors helped a lot whenever there was confusion about the Mechanics of the Game or how to go on with the making of the Final Game. The prototype for the final game was successfully made, with a little more work left to clean it up and add some functionality.

Technical issues were not much as it was mostly done on Python and its packages, which most of us were equipped with. Overall, the obstacles were mostly centered on understanding the various concepts that went into the project and minor coding errors. The project nevertheless was a very fun experience. At times it felt a bit overwhelming to complete all the assigned tasks, but the mentors were quite understanding. Mentors also helped us a lot whenever we were stuck while writing the codes. To code in Pygame was fun and frustrating at the same time. But doing this project gave us the confidence to take up more such fun projects in the future.

Conclusion

Through this project, we were able to understand the theory behind some common games and hence making us view these games not only from the view of a player but also as a developer. Through this project, we understood how game theory is different from Game refinement theory, how to balance the sophistication of a game and hence giving us a deep insight into the current game and improving the quality of the game. As a player what my strategy should be for maximizing winning chances and as a game developer what should I include to make it more fun and not very sophisticated. For the Mafia game, we were able to find the best composition under the assumption that the game refinement measure is within the sophisticated zone. For Ludo, out of the four strategies depending on what we are looking for in the game as a player or a developer we can choose any one of the four strategies discussed. For a player who is only looking to win, the cautious strategy will be the best (as a game developer, this can be set as an easy mode) but for a fun and quick game, the chaotic and quick nature of the Careless greedy can be used. The cautious greedy can be used as a normal-level opponent.

Future work

Analysis of the three games: In this area, a ton of improvements can be made to ensure quality results and come to a better conclusion for the skill/luck balance. For example, the snakes and ladders simulations were done with just three different boards and we considered the results. The results can be further solidified by checking with multiple configurations of boards. Along with the quality of results, we can look at ways to improve the skill/luck balance of the games by making changes to the actual game.

For the Final Game, the single player version is done. Work is in progress currently for the multiplayer part. Once this is done, it can be analyzed for skill/luck balance, further modified to make it better, and can be used as a recreational activity.

References

Papers:

1. [Possible Interpretations for Game Refinement Measure](#)
2. [Mafia Game Setting Research Using Game Refinement Measurement \(researchgate\)](#)
3. [Complexity analysis and playing strategies for Ludo and its variant race games](#)