Simulating Monopoly to determine different strategies to increase the chances of winning

BY

2020B4AA1846G	Bhavya Mehta
2020B4A71588G	Devansh Agarwal
2018B4A80046G	Harsh Talwar
2020B4A71093G	Hrushikesh Kadam
2020B4A71955G	Kushal Maheshwari
2020B4AA0758G	Tejas Khadke

Course Instructor: Dr. Anupama Sharma

Birla Institute of Technology and Sciences-Pilani, K.K Birla Goa Campus

May, 2022.

Abstract

Monopoly is a fast-dealing property trading board game and is one of the most popular and well-known board games across the world, with multiple variations throughout history. Every game is different and is difficult to generalise and through our stochastic model of Monopoly, we will be trying to determine various strategies and techniques to increase the chances of winning.

We will be achieving our goal by simulating a board of monopoly with all its intrinsic properties, for example, Go To Jail space, Go space, Jail space, etc. Following this, we will be simulating the probability rolls which include the 2 dice, and also other components which are based on random draws such as the community chest and chance spaces.

Landing probability is defined as the probability of a player ending up on a particular tile. We will approximate the landing probability of each space by running a Monte Carlo Simulation which will approximate the distributions by repeated random sampling. The performance of each property depends on the revenue, the cost, and the landing probability. Based on these parameters and strategies that we developed we create a heuristic to score properties and hence, devise strategies that increase the chances of winning an arbitrary game.

Table of Contents

List of Illustrations

Introduction

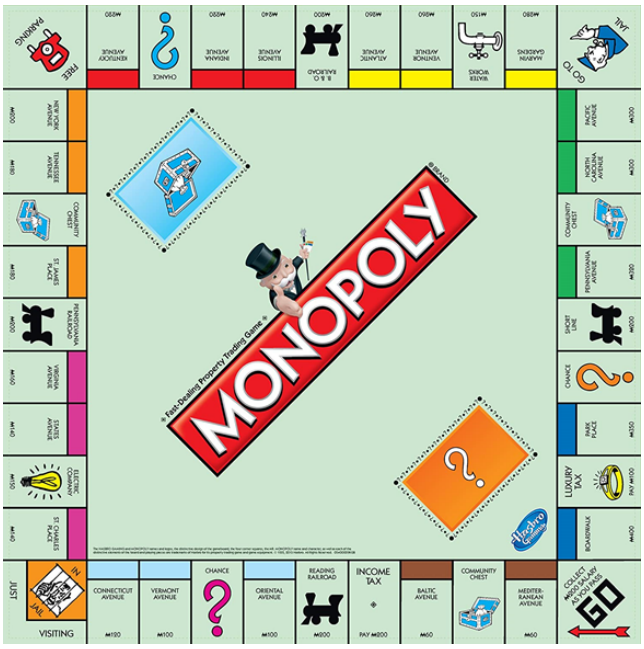

Fig 0: Monopoly Board By Hasbro

Rules of Monopoly

Monopoly is a board game in which one's goal is to buy and develop the property while driving competitors bankrupt. The goal is to become the game's wealthiest player by purchasing, leasing, trading, and receiving payments on properties.

It is a 2-6 player game in which each player rolls two dice during their turn and moves their pawn across the board. The board tiles include properties, train stations, the utilities, the taxes, the chance, the community chest, and the jail.

The game starts with all the players placed on GO. Anyone player starts the game by rolling the dice and moves ahead. A player gets an extra chance to play if they get a double, but if they get three doubles in a row, they will be sent to jail. They can either use a jail-free card, roll a double, or pay the fine to get out of jail.

If the player lands on a property, station, or utility, then they can buy it. If they choose not to buy it then it gets auctioned where the other players bid for that property. If the property is owned by some other player, then the player has to pay the rent.

If a player lands on chance or community chest, then they draw a card and perform the task mentioned on the card. If a player lands on super tax or income tax, then they pay the required amount. After completing every turn and passing GO, each player gets a 200 salary. To build more houses on a set, make sure to distribute them evenly amongst the set.

Markov Chain

A finite-state discrete-time system is a Markov process (also known as a finite Markov chain). It is a system with a finite number of attitudes or states that progresses sequentially from one state to the next, and for which the probability of passing from state i to state j is represented as p_{ij}, which depends only on i and j, not on the system's history in achieving state i. We are only concerned with the case when p_{ij} is independent of the moment at which the transition from i to j occurs. The chain is considered to have stationary transition probability in this situation.

The chance of being on any of the 40 spots in a given round is the basic quantity of interest for strategic considerations in Monopoly. The turn appears to be a natural unit of time, with the state being the position at the start of the turn. This process, however, is not Markovian. For instance, when a pawn is in the prison state, the person in charge may opt to keep the pawn in jail if no doubles are thrown. This option is only available if the preceding two states have been completed i.e. if a pawn has been in jail for two turns, it must be released. As a result, the memoryless property does not apply.

Another feature of the game which makes the process non-Markovian is the chance and community chest position-transfer cards. There are three positions each called chance and community chest, and upon reaching such a position the top card of 16 chance cards or 16 community chest cards is drawn. Of these cards, 10 chance cards and 2 community chest cards require a movement to another position. Except for one "get out of jail free" card in each stack, which may be retained until needed, the chosen card is placed at the bottom of the stack after completing its instructions. As a result, the order in which these cards appear, and therefore their impact, is determined not by the pawn's location, but by the stack's starting order and the history of its use.

In conclusion, as the game model in its true sense is not memoryless because of certain factors a few of which were mentioned above, we make assumptions to avoid dealing with these non-memoryless interdependent random variables, in order to study the model using Discrete-Time Markov Chain (DTMC) theory.

The following formula represents the memoryless property of the Markov Chain process:

$$P(X_n = x_{i_n} \mid X_{n-1} = x_{i_{n-1}}, \ldots, X_0 = x_{i_0}) = P(X_n = x_{i_n} \mid X_{n-1} = x_{i_{n-1}})$$

To find landing probabilities using Markov chains we first design the transition matrix. We do this by finding the transitional probability for every pair of tiles on the board. As there are 40 tiles on the board, a 40 x 40 transition matrix is formed whose any element a_{ij} represents the probability of landing on the j^{th} tile from the i^{th} tile.
For instance, transition probabilities for the 12th square (pink 1) is calculated below.

Initial Square	Final Square	Value to be rolled	Probability
12	13	1	0
12	14	2	1/36
12	15	3	1/18
12	16	4	1/12
12	17	5	1/9
12	18	6	5/36
12	19	7	1/6
12	20	8	5/36
12	21	9	1/9
12	22	10	1/12
12	23	11	1/18
12	24	12	1/36
12	All other squares	>12	0

Table 1: Transition probabilities for 12th square(pink 1)

The above table is similar for all squares except chance, community chest and go-to-jail. For the 'go-to-jail' square all transition probabilities are zero except 'jail'(while is 1). For chance and community chest transition probabilities also depend on which card is chosen. Let us consider chance. As there are 16 cards, the probability of choosing a particular card is 1/16. There are 9 cards out of the 16 that ask the player to go to another square. Hence, those squares will have a transition probability of 1/16. There is a 7/16 probability that the player will stay on chance and roll the dice. Therefore, the probabilities of rolled values will be multiplied by 7/17. The transition probability table for 8^{th} square (chance 1) is shown on the next page..

Initial Square	Final Square	Value to be rolled	Probability
8	1	none(chance card)	1/16
8	5	none(chance card)	1/16
8	6	none(chance card)	1/16
8	8	0	0
8	9	1	0
8	10	2	1/36*(7/16)
8	11	3	1/18*(7/16)
8	12	4	1/12*(7/16+1/16)
8	13	5	1/9*(7/16+1/16)
8	14	6	5/36*(7/16+1/16)
8	15	7	1/6*(7/16)
8	16	8	5/36*(7/16)
8	17	9	1/9*(7/16+1/16)
8	18	10	1/12*(7/16)
8	19	11	1/18*(7/16)
8	20	12	1/36*(7/16)
8	25	none(chance card)	1/16
8	40	none(chance card)	1/16
8	others	>12	0

Table 2: Transition probabilities for 8th square(chance 1)

A 40 x 40 matrix is constructed after finding transition probability values for all the squares. The matrix is shown below:

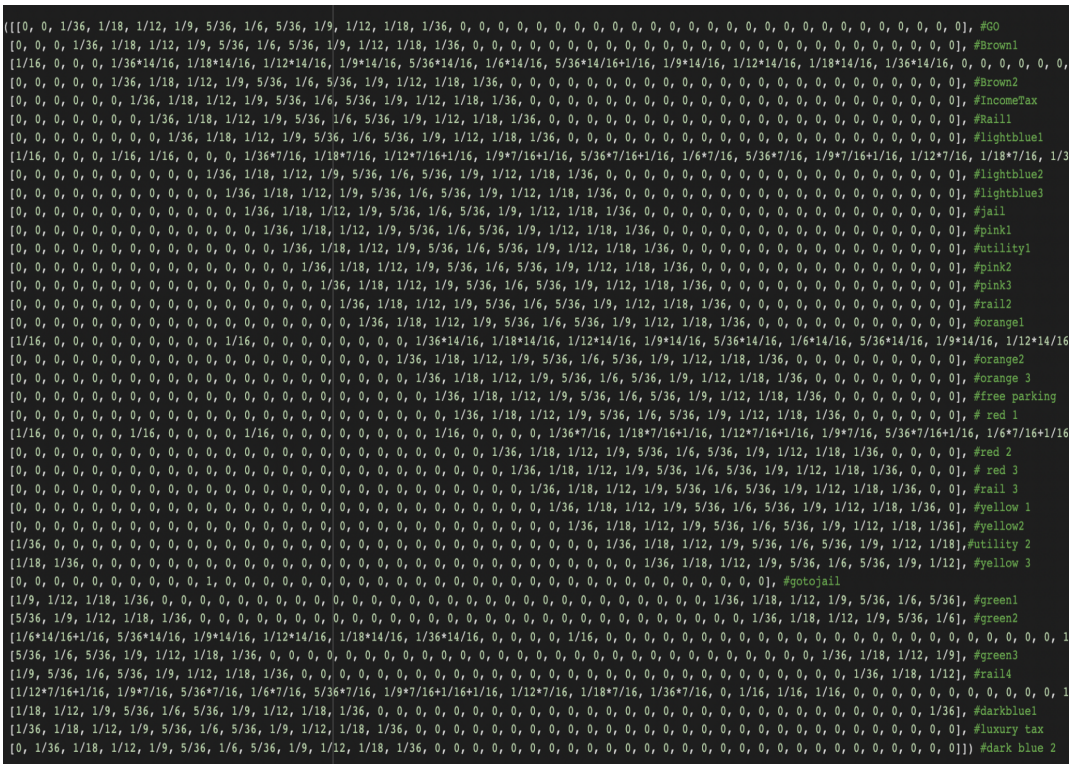

Fig 1: Transition matrix

The above transition matrix is represented as a heatmap, where the higher the colour concentration, the higher the probability.

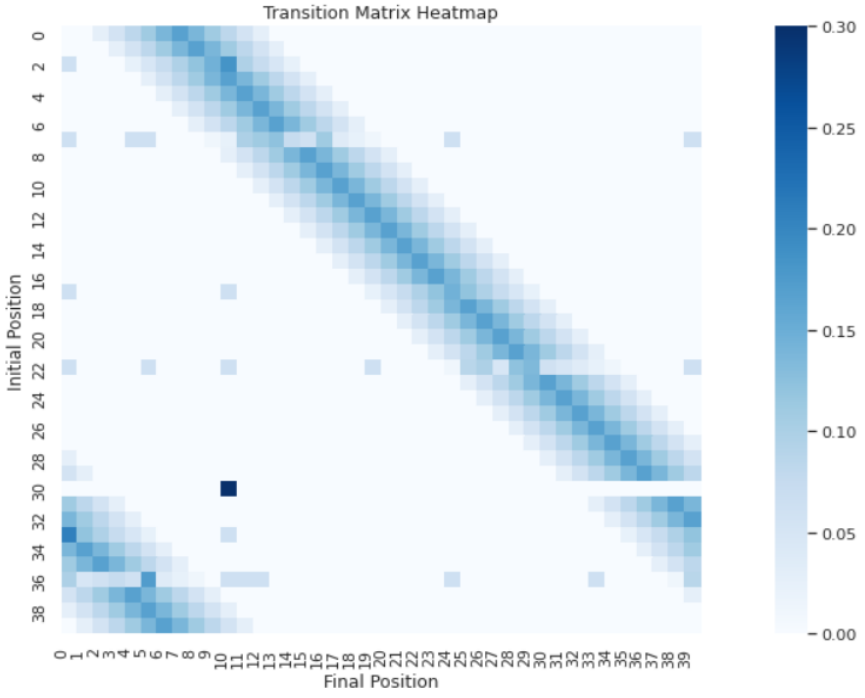

Fig 2 : Heatmap of transition matrix

Monte Carlo Simulation

A Monte Carlo simulation or Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. The underlying concept is to use randomness to solve problems that should ideally be solved using the deterministic approach. It is often very useful when there exists an intervention of different random variables which makes the theoretical approach almost impossible to deal with.

In monopoly, the term landing probability is very significant. It is defined as the probability of a player ending up on a tile at the end of a tile at the end of their turn i.e. right before the next roll if they roll a double. We can very conveniently use the method of Monte Carlo simulation to calculate this probability while not making the broad assumptions that were required to study the model using the Markov Chain theory.

In order to apply this method, we encode the game by simulating the board, the players rolling the dice, the properties, etc. The game is simulated for over a million times. We record the frequency of players ending up on each tile. Then we approximate the landing probability of a tile as the frequency of ending up on that tile divided by the total number of times the dice were rolled. The more games we simulate, the more confident our approximation is.

List of strategies

In our model, we have considered determining and analysing the results of 3 strategies that are:-

1.) Calculating the landing set probability to find which colour set is the most landed on.
2.) Determining the properties that have the highest probability of being owned by the winner.
3.) Determining the set with the highest probability of bankrupting a player.

Strategy 1: Landing probability of a set and tiles

Aim: To find the properties and colour sets with the highest probability of landing on the particular property or colour set.

This strategy has been run by two different methods:-

- Monte Carlo Simulation
- Markov Chain

First we will discuss the Monte Carlo Simulation method and its assumptions.
Assumptions:

- Trading, auctioning, and mortgaging are not considered.
- In every simulation, a game runs for a random number of turns which is determined by a Gaussian distribution with a mean of 30, and a standard deviation of 3
    a. Mean is taken as 30 on the basis of a research paper.
    b. The standard deviation is arbitrarily assigned 3 to introduce randomness.
- Total number of players is 4.
- An imprisoned player will try and leave jail as soon as possible, either by using the “Get Out of Jail Free” card or paying the fine (which does not hold true for the latter end of a game).
- Money is not considered while simulating these runs.
- Standard Monopoly Rules are followed except for the ones mentioned above.

Since we are finding the probability of a player landing on a certain tile, money does not play as a factor, hence we assume money to be infinite. To avoid certain complications, we have not considered trading, auctioning, and mortgaging.

Results of Monte Carlo Simulation

TILE	%PROBABILITY	TILE	%PROBABILITY
Blue 1	2.045756	Orange 1	2.866006
Blue 2	2.485266	Orange 2	2.97784
Brown 1	1.97571	Orange 3	3.109597
Brown 2	2.160338	Pink 1	2.851135
Sky 1	2.483743	Pink 2	2.433463
Sky 2	2.570502	Pink 3	2.547612
Sky 3	2.501194	Red 1	2.825751
Green 1	2.568283	Red 2	2.706874
Green 2	2.499551	Red 3	3.157763
Green 3	2.366512	Yellow 1	2.662313
Community Chest 1	1.81237	Yellow 2	2.626789
Community Chest 2	2.654299	Yellow 3	2.517072
Community Chest 3	2.263809	Station 1	2.958115
Chance 1	1.144576	Station 2	2.889
Chance 2	1.212095	Station 3	2.849056
Chance 3	0.950768	Station 4	2.292689
Free Parking	2.883401	Utility 1	2.711486
Go	2.934605	Utility 2	2.743939
Go To Jail	0	Income Tax	2.412347
Jail	6.309528	Super Tax	2.038847

Table 3: Result of Monte Carlo simulation

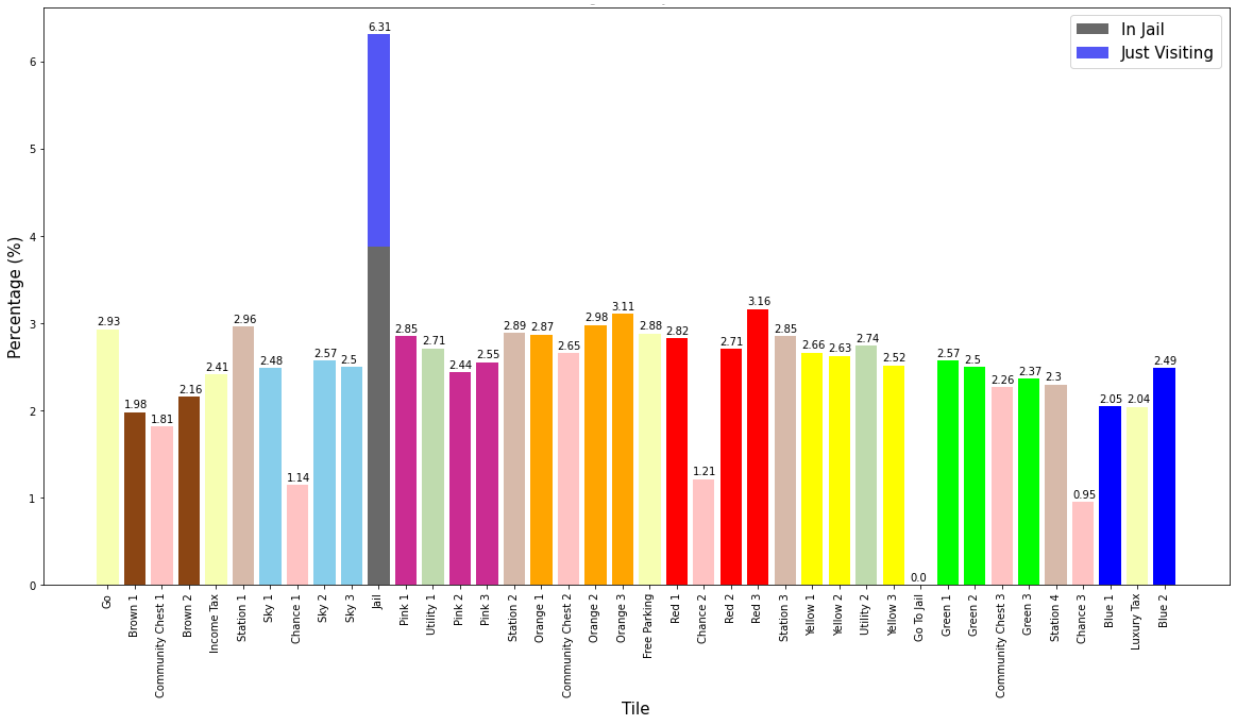

Fig 3: Tile landing probability, Monte Carlo

Here we calculated the individual landing probability of each tile by running 1 million simulations. A few of the conclusions that can be drawn from the data are as follows:-

1) Orange has the highest average landing probability among all the properties. The explanation to this revolves mainly around two ideas.
    - Firstly, the jail has the highest individual landing probability.
    - Secondly, the value that appears on the die is biased near the median.
2) When a player in jail rolls the dice, there is a 39% chance that the player lands on orange, 14% chance on pink, and 6% chance on red.

$$P(x) = \begin{cases} \frac{1}{36} & \text{if } x \in \{2, 12\} \\ \frac{2}{36} = \frac{1}{18} & \text{if } x \in \{3, 11\} \\ \frac{3}{36} = \frac{1}{12} & \text{if } x \in \{4, 10\} \\ \frac{4}{36} = \frac{1}{9} & \text{if } x \in \{5, 9\} \\ \frac{5}{36} & \text{if } x \in \{6, 8\} \\ \frac{6}{36} = \frac{1}{6} & \text{if } x = 7 \\ 0 & \text{otherwise.} \end{cases}$$

Fig 4: PMF of two dice rolled

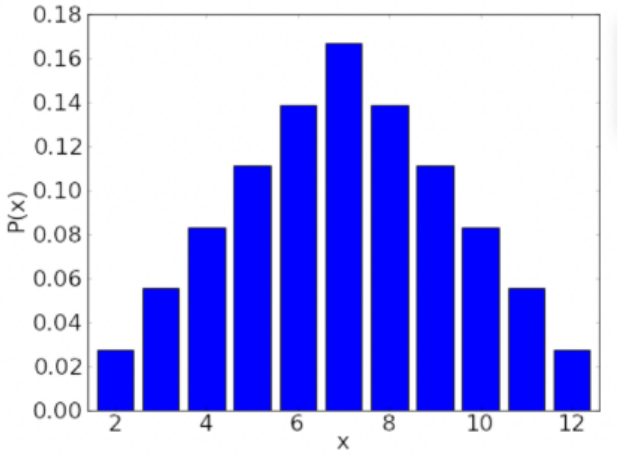

Fig 5: Bar graph representation of PMF

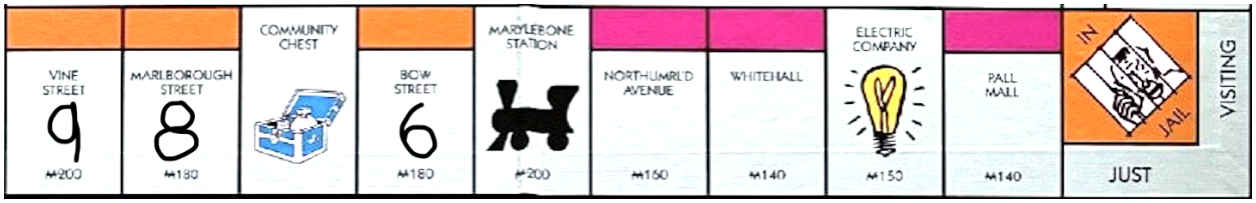

Fig 6: Orange set is the 6th, 7th and 9th square after 'Jail'

The above Fig 4 and Fig 5 clearly shows that the probability of the dice rolling 6-8 is maximum which is calculated theoretically by conditional probability theory and by running 1 million simulations which returned Fig 5. In Fig 6, positions 6, 8, and 9 are marked right after jail. Since all these 3 positions are orange tiles only, this increases their probabilities.

3) Furthermore, the brown set has the lowest landing probability mainly because it is {1,3} steps from 'Go' which is also the starting point.

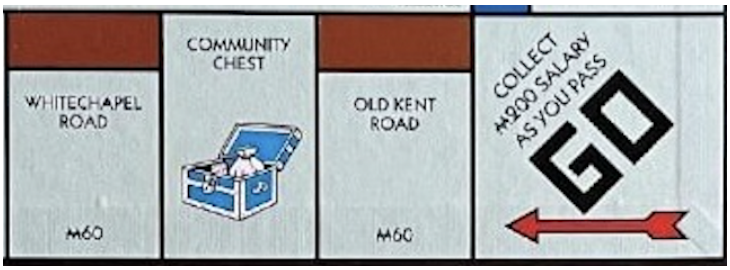

Fig 7: Brown set is 1st and 3rd after 'GO'

Landing Probability

Brown 1	1.97571
Brown 2	2.160338

4) All sky, green and yellow coloured tiles have average landing probabilities as no special features can be observed for any of them.

Average landing % probability = 100/(No. of tiles)
= 100/40
= 2.5

Sky 1	2.483743
Sky 2	2.570502
Sky 3	2.501194
Green 1	2.568283
Green 2	2.499551
Green 3	2.366512

Yellow 1	2.662313
Yellow 2	2.626789
Yellow 3	2.517072

5) There are 3 chance tiles and all have a low probability because chance cards mostly ask the player to advance to some other tile. These chance cards are mentioned below,

I. Advance to Go.
P(Go) = 0.935_____on the higher side

II. Advance to Pink 1.
P(Pink 1) greater than P(Pink 2) and P(Pink 3)

Pink 1	2.851135
Pink 2	2.433463
Pink 3	2.547612

III. Advance to Blue 2.
P(Blue 2) is greater than P(Blue 1)

Blue 1	2.045756
Blue 2	2.485266

IV. Advance to Red 3.
P(Red 3) is greater than P(Red1) and P(Red 2)

Red 1	2.825751
Red 2	2.706874
Red 3	3.157763

V. Advance to Nearest Station. There are 4 stations but only 3 chance tiles. Station 4 is the one which is not nearest to any chance tile.

Station 1	2.958115
Station 2	2.889
Station 3	2.849056
Station 4	2.292689

Now, focusing on the Markov Chain method, the assumptions differ from Monte Carlo Simulation due to its memoryless property.
The assumptions made for Markov Chain are as follows:-

1) Trading, auctioning and mortgaging are not considered.
2) Rolling doubles does not give the player an extra chance to roll the dice.
3) Each time a card is drawn from the deck of 'Chance' or 'Community Chest', it is selected randomly.
4) Going to jail and visiting jail is considered the same.
5) Money isn't considered in this model.
6) Standard Monopoly Rules are followed except the ones mentioned above.

Due to Markov Chain's memoryless property, we could not consider rolling doubles as an extra chance. We have shuffled the deck of chance and community chest cards, everytime a card was drawn out. In this method, we have considered the probability of going to jail and visiting jail as the same. Thus, in the chart below, the probability of 'Go to Jail' is not zero like we received in Monte Carlo Simulation (Fig 3.)

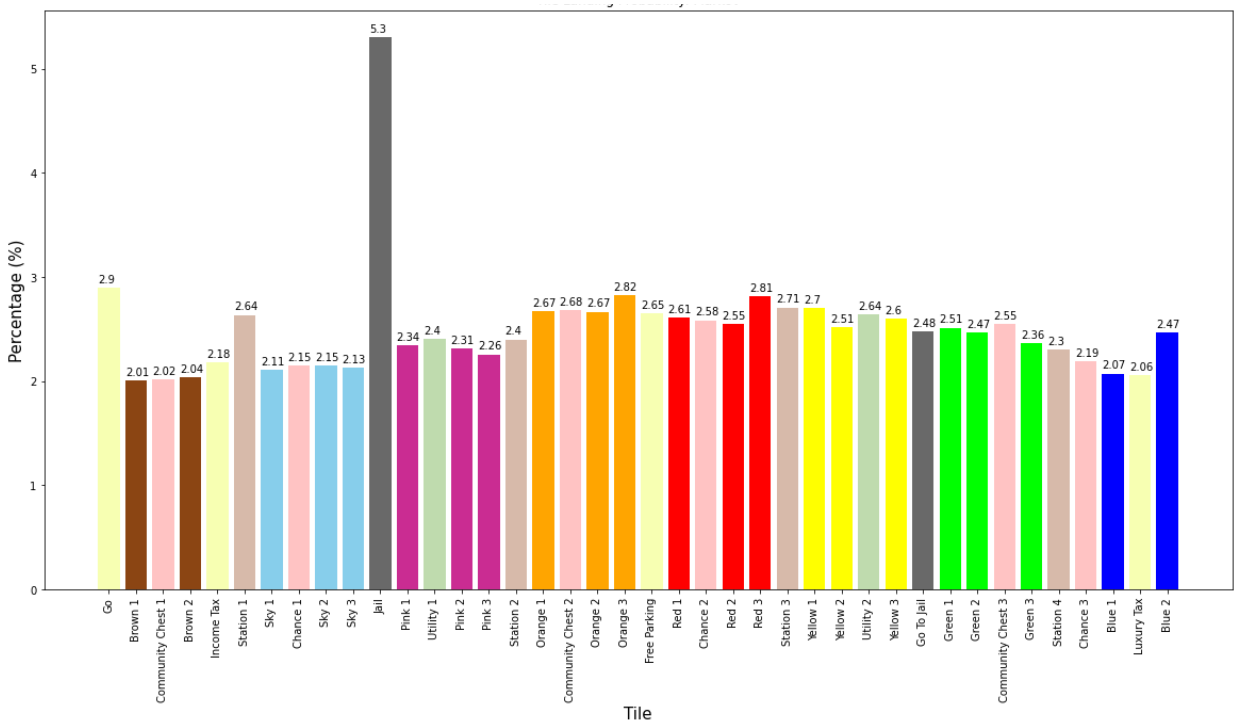

Fig 8: The landing probability, Markov chain

Markov Chain v/s Monte Carlo Simulation

From the results obtained, we can see that the probabilities obtained from Monte Carlo simulation and Markov Chain differ by a very small margin. This deviation in the results is majorly due to the difference in assumptions in the two methods.

	name	%
4	Station	10.991554
10	Orange	8.954937
12	Red	8.687348
8	Pink	7.837747
13	Yellow	7.806548
5	Sky	7.551845
15	Green	7.431034

Fig 9: Monte Carlo

	name	%
4	Station	10.046337
10	Orange	8.160558
12	Red	7.978127
13	Yellow	7.814742
15	Green	7.338399
2	Community Chest	7.251967
6	Chance	6.921154

Fig 10: Markov Chain

This implies that the stations have the highest landing probability whereas the Orange set leads in colour sets with the Red set just behind.

Strategy 2: Common Winning Conditions

Aim: To determine which properties, utilities, and stations were owned by the winner at the end of the game.

Assumptions:

The following assumptions are made to avoid high-level complexity in the model and to make it solvable.

- Sets have been randomly distributed among all the players.
- Total number of players playing the game is 4.
- Each property has a hotel.
- Players can not mortgage their properties.
- Players aren't allowed to sell the hotels.
- Properties of a bankrupt player are discarded and cannot be bought by the other players.
- Players are positioned randomly on the board and start with a random amount of money.
- A game ends either when 3 players go bankrupt or it exceeds 40 turns.
- The number of simulations run is 1 million.

In this Strategy, we have run 1 million simulations using the Monte Carlo random sampling method. We cannot use the Markov chain method in this strategy as it will have many complex calculations and matrices. Furthermore, we have included money as a factor for determining the winner which makes calculating the probability memoryless. As a result, we can't use a Markov chain because the events must be memoryless.

Oscillation of Results

- If none of the three players are bankrupt after 40 turns, the winner is determined by who has the most money.
- To ensure that the result converges and does not oscillate, the maximum deviation between the values of two consecutive runs is plotted (n^{th} and $n\text{-}1^{th}$)
- From the graph (fig 1), we can conclude that the result starts converging very quickly and is in fact almost 0 at around 500 runs which is low compared to the number of iterations that the game runs for which is 1 million times.
- Thus, we can conclude that the results that we obtain are extremely stable and will not deviate by the end of a million simulations.

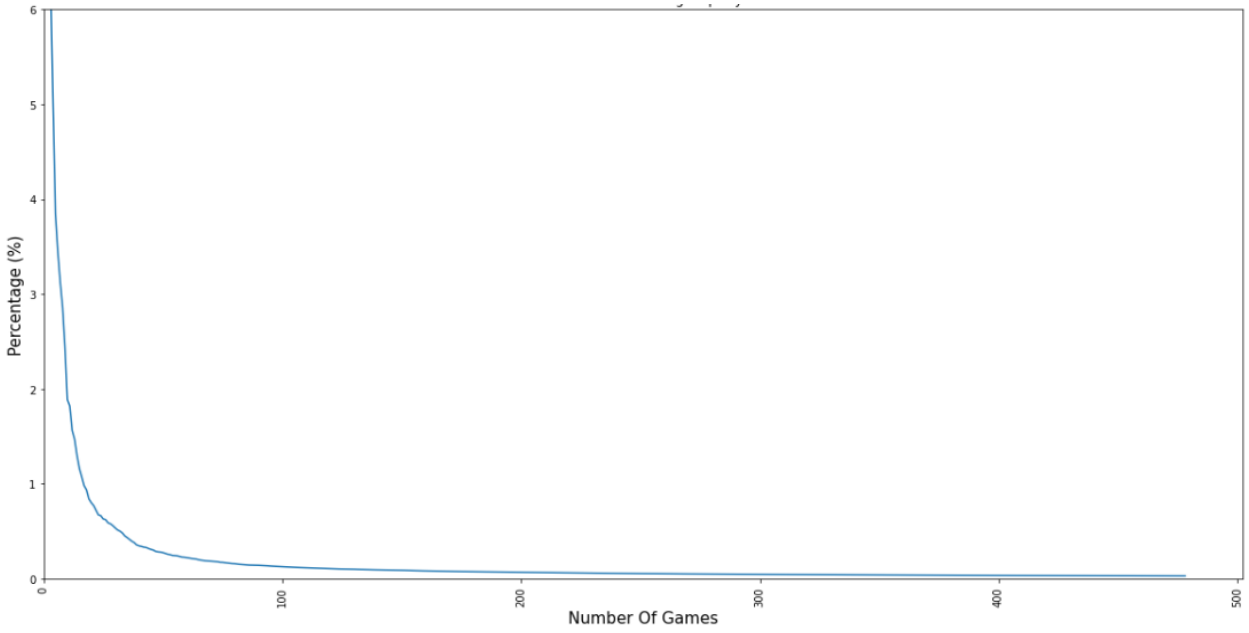

Fig 11: Deviation for winning property

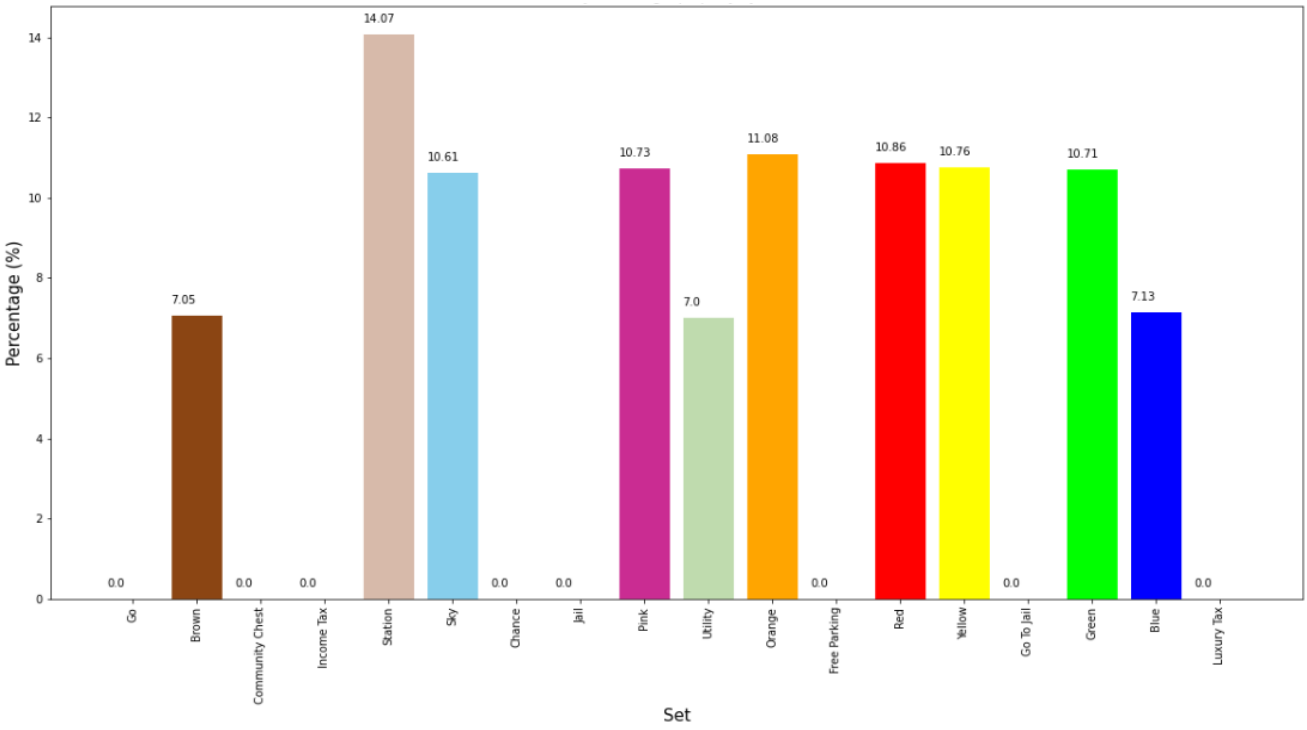

Fig 12: Probability of owning a property by the winner

Results

The winner owned a station 14.07% of the time, which is the highest, and the winner owned a utility only 7.0% of the time, which is the lowest, as shown in Fig 2. Among the colour sets, brown and dark blue sets have the lowest probability i.e 7.05% and 7.13% respectively and the orange colour set has the highest probability 11.08%.

Strategy 3: Determining the set with the highest probability of bankrupting a player

Aim: To find the set/property where each player gets bankrupt based on landing probability and rent of the property.

Assumptions:

The following assumptions are made to avoid high-level complexity in the model and to make it solvable.

- Sets have been randomly distributed among all the players.
- Total number of players playing the game is 4.
- Each property has a hotel.
- Players can not mortgage their properties.
- Players aren't allowed to sell the hotels.
- Properties of a bankrupt player are discarded and cannot be bought by the other players.
- Players are positioned randomly on the board and start with a random amount of money.
- A game ends either when 3 players go bankrupt or it exceeds 40 turns.
- The number of simulations run is 1 million.

This strategy gives information about the colour set that has the highest probability of making a player bankrupt. Here the simulation has been run 1 million times using the Monte Carlo random sampling.

Oscillation of Results

- If none of the three players are bankrupt after 40 turns, the winner is determined by who has the most money.
- Just like in the previous strategy, to ensure that the result converges and does not oscillate, the maximum deviation is between the values of two consecutive runs is plotted (n^{th} and $n\text{-}1^{th}$)
- Unlike the previous graph, this one converges to 0 at around 2000 runs instead of 500 which is still quite low compared to the number of iterations that the game runs for which is 1 million times.
- A player's ability to go bankrupt on a specific tile is more dependent on the number of runs as opposed to the properties held by the winner.
- Thus, we can conclude that the results that we obtain are extremely stable and will not deviate by the end of a million simulations.

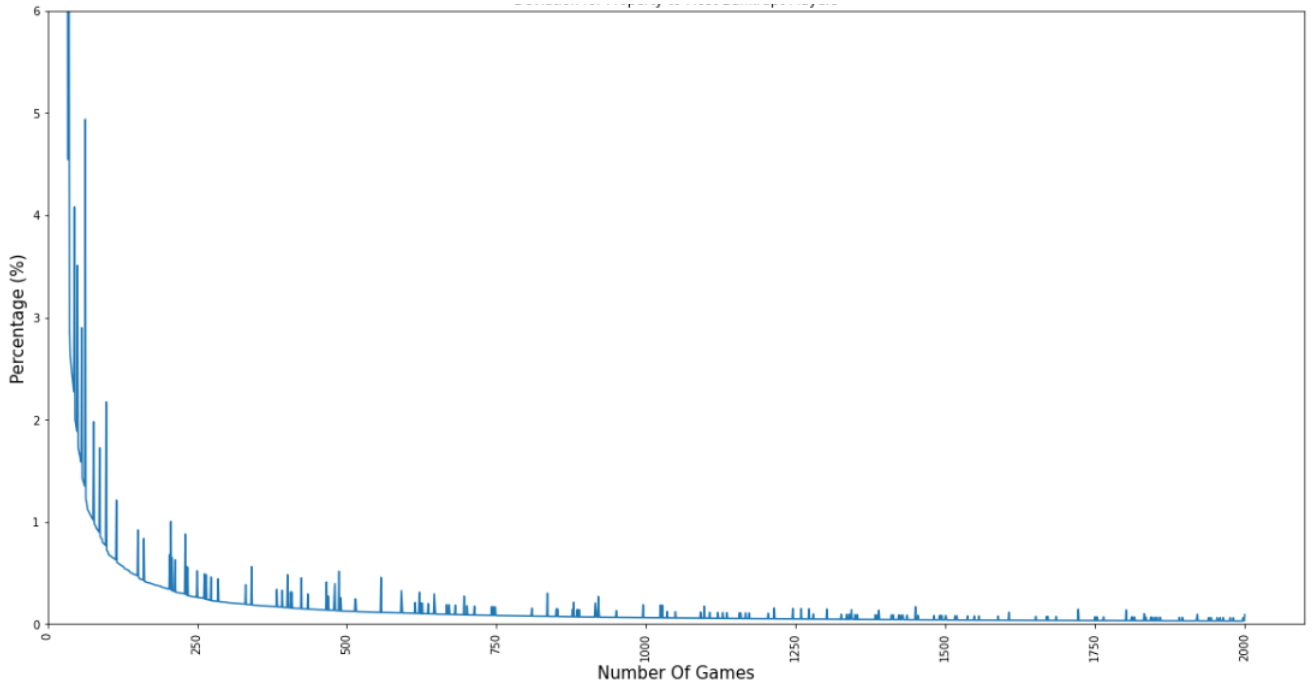

Fig 13: Deviation of Property to most Bankrupt players

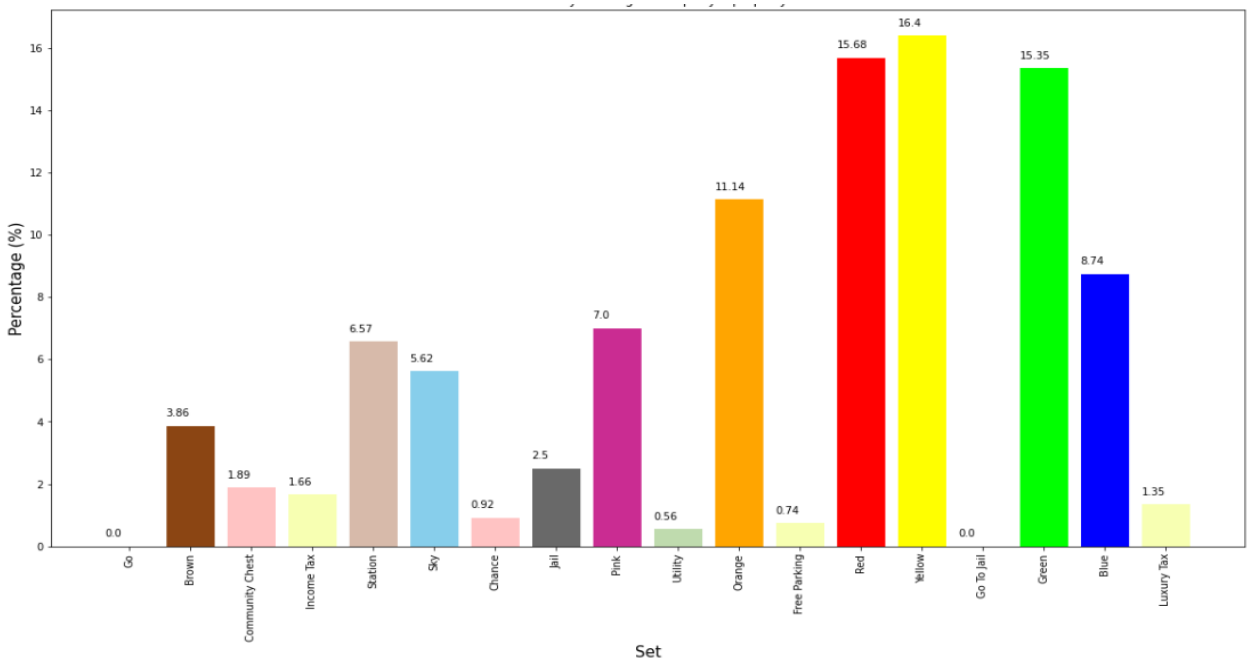

Fig 14: Probability of being bankrupt by a property

Results

The Yellow colour set has the highest probability of making a player bankrupt i.e 16.4% followed by the Red colour set and Green colour set i.e. 15.68% and 15.35% respectively.

The highest chance of a player landing on the monopoly board is with the jail tile, and the most probable and likely dice outcomes are 6,7,8 where the orange set is. It's possible that a player will go bankrupt there, but it's more likely that he or she will make it through. However, the player is very likely to land on the yellow or red tiles on the next roll, increasing the chances of going bankrupt.

The Brown colour set has the least probability of making a person bankrupt.

This is because the number of properties is 2, and the return from the hotel on a brown property is less, the brown colour has the least chance of bankrupting players. Furthermore, the likelihood of a brown colour landing is low.

Combination of Results

The main objective of this model is to combine multiple strategies together to give an idea of how well a property performs based on different parameters and if there was a property that in fact could be labelled the "Best Set in Monopoly to Buy" according to the strategies that were chosen based on how well it performed in these 3 strategies. However, this is not the case. The price to buy and develop a property must also be taken into consideration as the property could perform extremely well but then the cost to buy and develop the property is also extremely high which would make it less viable considering the limited amount of funds with each player.

To develop a heuristic, min-max normalisation was used to score each property in the three strategies and the total investment required to develop that set.

Min-max normalisation performs a linear transformation on the original data. Suppose that min_A and max_A are the minimum and maximum values of an attribute, A. min-max normalisation maps a value, v_i, of A to v'_i in the range [new_minA, new_maxA] by computing

$$v'_i = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

```
for i in range(len(score)-1):
  score1.append(score[i]/np.max(score[i])*10)
score1.append(10-(score[3]*10/(np.max(score[3])-np.min(score[3]))))
```

For our new normalised score, a range of 0 to 10 was arbitrarily chosen for simplicity. According to this, the best-performing property in a particular strategy is given a score of 10, and the worst-performing one is given 0. This is reversed for the investment cost parameter as higher is worse while lower is better, and so the lowest investment cost is given a score of 10, and the highest investment cost is given a score of 0. So, the property is scored out of 40 according to its performance in all 4 parameters (higher is better) (Fig. 15).

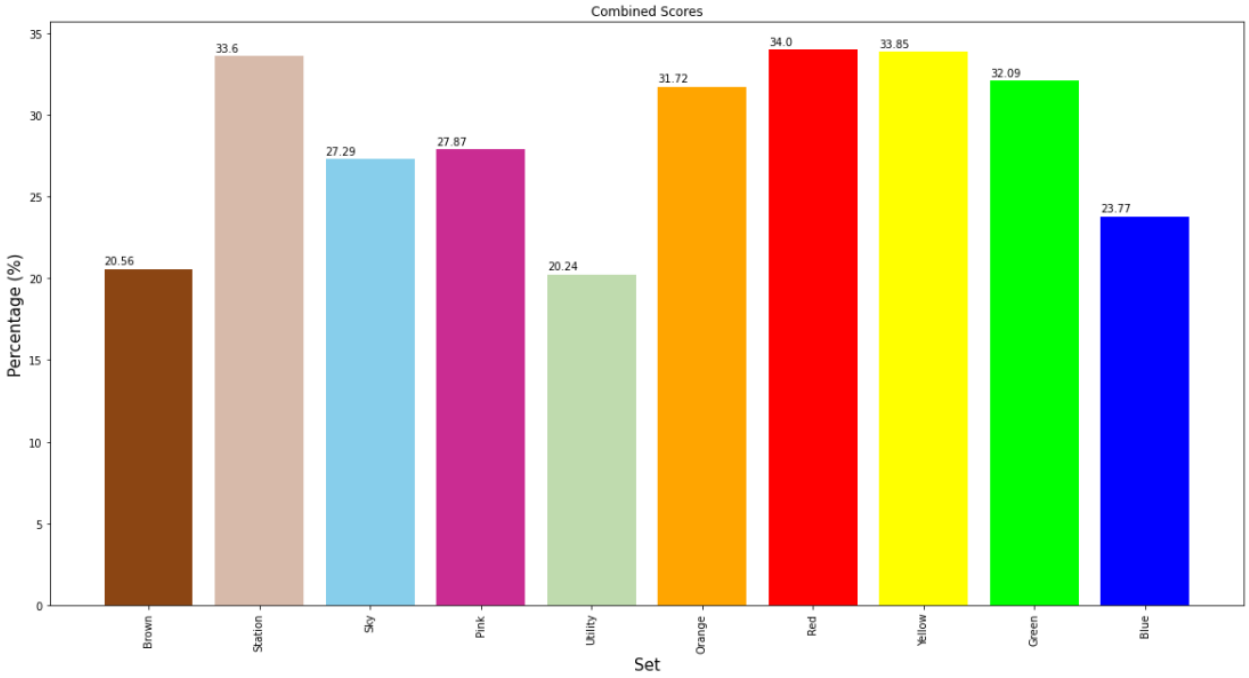

Fig 15: Scores out of 40 for all the sets (higher is better)

From Fig 15, it is trivial to observe that the red and yellow sets are the best performing sets, followed closely by railway stations. This is a surprising result as the red set was never the best in a single strategy, but having a consistent podium position seems to have helped it edge over the other sets, which performed much better in some strategies while worse in others. Another surprising result is the railway station, which is often overlooked because of its relatively low rents. This set ends up with a podium finish, which can be attributed to having the highest landing probability of all sets and one of the lowest investment costs among all the sets and would not have been easily overlooked if the investment costs had not been taken into account.

There are also some indisputably bad properties. The light blue set gets a meagre amount of rent (until the property is developed), the dark blue set suffers from having one fewer property, and the brown set suffers from both of these problems, making these the worst three property sets to buy in the entire game

All these results are based on a few assumptions that are bound to skew the data somehow or the other. Every game played is different, and the number of possible game states makes it difficult to generalise. It is also challenging to take into account the unpredictable human nature if trading and mortgaging are allowed and would require Evolutionary Learning, Deep Learning, or Neural Networks to make bots smart enough to make decisions on their own accord, which is beyond the scope of this study.

Conclusion

Monopoly is ultimately a very intricate and complicated balancing board game — one in which players try to increase their earnings while keeping enough money to pay the rent on other players' properties. However, concentrating on the landing probability of each set, the winning probability of each set, and the probability of a set bankrupting a player provides a distinct viewpoint on the game.

While actual figures will differ with each roll of the dice, strategising may be done using the mathematical notion of expectation and some fair assumptions about how the typical game would play out.

Since we have simulated these results based on Monte Carlo simulation, these values are bound to vary between runs, albeit only slightly, always tending and converging towards similar results after a significant number of simulations have been run.

Anyone who has been at it for over three hours of a never-ending game of Monopoly is aware that it is a game of attrition and willpower more than it is a game of probability and mathematics.

References

- https://www.hasbro.com/common/instruct/00009.pdf
- Bishop, Richard & Ash, Robert. (1972). Monopoly as a Markov Process. Mathematics, Magazine. 45. 26-29. 10.2307/2688377.
- MONOPOLY AS A MARKOV PROCESS - University of Illinois .... https://faculty.math.illinois.edu/~bishop/monopoly.pdf
- The Indisputable Existence of Santa Claus: The Mathematics of Christmas *by*
- *Hannah Fry and Thomas Oléron Evans. Copyright © 2016, 2017 by Dr. Hannah Fry and Dr. Thomas Oléron Evans. Published in 2017* by The Overlook Press, Peter Mayer Publishers, Inc.
- Jiawei Han, Micheline Kamber, Jian Pei, Data Preprocessing, ISBN 9780123814791,
- https://doi.org/10.1016/B978-0-12-381479-1.00003-4.
- https://blog.ed.ted.com/2017/12/01/heres-how-to-win-at-monopoly-according-to-math-expert /
- Code: MonopolyModel.ipynb - Colaboratory