CSE 202 - Project Ludo

Andrew Ghafari - A59020215
Chirag Dasannacharya - A59016593
Jay Jhaveri - A59017531
Niraj Yagnik - A59018067

I - Introduction

We decided to build our project to analyze the game Ludo. Ludo is a board game that originated in India and is now a common game in almost all Indian households. The game is played with two or more players with its main objective being to transfer all your pawns (there are 4 per player) from the base to the center of the board (called home) before anyone else. In our project, we will analyze algorithmically the nature of the game and its tradeoffs.

The game is played on a squared grid, having 72 boxes (52 of which are common to all players), 8 of the common ones being safe spots. The game starts with each player having 4 pawns in a corner of the board, and they must move them, according to the result of a dice roll from their respective base to the center of the board. The winner is the player who finishes this task the earliest.

We decided to make a project about this game due to its nature, since it has a limited set of rules but also requires players to make decisions and trade-offs. Some of these choices are which pawn to move and how to strategically position pawns to threaten opponents. The game also involves elements of chance, as the dice roll determines how far a player can move their pawn on each turn.

II - Rules of Ludo

Before we move further into the model we intend on implementing, we first have a look at the basic rules of Ludo which we plan on analyzing.
The basic rules of Ludo go as follows:

1. The game consists of four players, and each player starts with four tokens placed on the base, the large starting area of the player's color (marked in the image below with white stars). Figurative illustration of the ludo board is depicted in *Fig 1*.
2. Players roll a die to move their tokens clockwise around the board, on the gray squares and starting from the respective colored square by their base.
3. A pawn’s journey can only start after getting a dice roll of 6, that takes it from its base to the associated safe spot right outside it.
4. The player decides what pawn they want to move based on their strategy and situation.

5. The chosen pawn moves as many spaces as the value on the die. In case of a six, the player gets an additional turn as well.
6. Each pawn must complete an entire lap from the starting square to the home triangle to finish the journey.
7. The player's primary goal is to move all their tokens from the starting square to the finishing or home triangle (marked with a trophy, which is of the same color as the player's).
8. If a player lands on a square occupied by another player's token, the opponent's token has to go back to its starting square (this may be termed a 'strike').
9. Strikes are incentivized by setting an opponent back and giving the striker an extra turn on the die.
10. The board consists of a total of eight safe zones on the board - the four starting squares, marked with the color of the associated base (these squares are diagonally opposite the arrows), and the four squares marked with a star.
11. Pawns inside the safe zones cannot be struck and are safe from opponents. Any safe zone can accommodate pawns of different colors, and there is no limit on the number of pawns that can land in a single safe zone.
12. When a player's piece reaches the home column of its color, the pawn heads down the column towards the center of its home triangle. A pawn can only be moved to the home triangle with an exact roll. When a player's die roll lands its piece on the home triangle, that piece completes its journey.
13. The first player to move all their tokens to the finishing or home triangle wins the game.

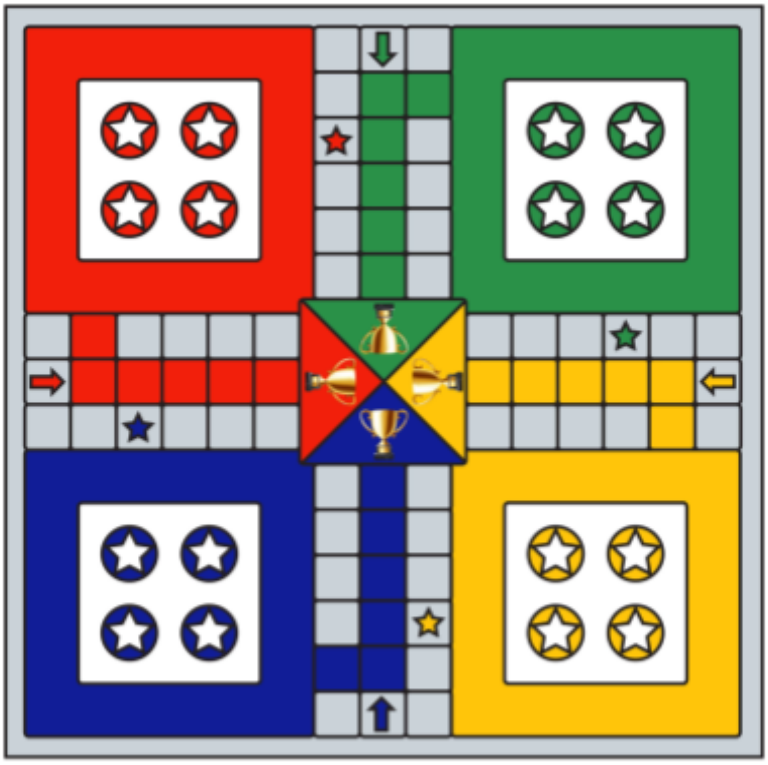

Fig 1: Ludo Board

III - Model

For our purposes, we consider a reduced version of Ludo with 4 players each with only two pawns. This introduces all the qualitative nuances of the game, while simplifying the quantitative requirements. Concepts from the resulting algorithm for two pawns should be extensible to a game with 4 pawns as well.

Ludo is a turn-based board game, where the state of the game can be encapsulated entirely by the current board (pawn positions, current player). Knowledge of previous moves does not make any difference to future decisions, *except as a study of opponents' playing styles*. As a result, given such knowledge in advance, an optimal strategy in Ludo is necessarily a greedy one - the best overall strategy must necessarily incorporate the best current decisions, given the current game state.

Our algorithm, as a result, must be expected to take as input the current pawn positions, the players' order, some assumptions about players' playing styles and the current value of the dice roll (assumed just played). It must generate, as an output, a single choice of which pawn to move by the value of the dice roll.

It is to be expected that the algorithm works differently for different opponents' styles (some may prioritize the progress of one pawn at a time, some may choose to strike freely, despite the risk of stranding their pawns in an unsafe position, some may choose to play safely by moving only from one safe spot to another when no other players have pawns in the vicinity). To begin with, we input this data to the algorithm as prior knowledge, later versions may choose to learn during the game.

Further, as an additional simplification, we initially choose not to model the rule that a player plays again on a six nor that a pawn breaks out of the base on a six only. This can be updated as work on the algorithm progresses and further observations are made.

IV - Increasing Complexity

Assuming we successfully implement an algorithm with the base cases mentioned above. A few ways we can increase the complexity would be:

- Addition of 2 pawns per player (making it 4 in total)
- Needing a dice roll of 6 to get out of the base.
- Implementing additional chances on a dice roll of 6
- Modeling different gameplay personalities. For example, one player may be random, one greedy, one erratic, one goes for a kill no matter what, etc.
    - Further, instead of having personalities as prior knowledge, we may integrate a personality detection sub-algorithm inside our main algorithm.
- Increasing the number of players from 4 to infinity. (The board, safe spots, # of tiles, and max number of dice will increase linearly as the number of players increases.)
- Introducing the concept of team-based ludo: wherein the end goal would change to the winning of the team instead of a solo win condition. This could further have two alternatives:
    - Team killing off
    - Team killing on