# Introduction

The game we chose to implement is Monopoly which will closely resemble the official rules. Monopoly is a game of strategy, negotiation and luck. The primary objective of Monopoly is to make the other players go bankrupt such that a single player has a monopoly on the board. The board is divided into spaces representing properties, railroads, utilities, and other spaces with the potential to provide benefits or drawbacks for the players. To move from one space to another, players roll the dice and perform the actions dictated by the space they, or another player, lands on. The actions include, but are not limited to, paying rent/tax and drawing cards from chance and community chests, where each drawn card contains its own set of actions. A unique feature of monopoly is its economic system. Monopoly incentivizes the players to invest in properties. For doing so, the players are able to generate income, allowing them to make more investments. The economic system allows players to trade and make deals with each other involving their money or properties. Monopoly is one of the most widely known board games and we are creating one of its many variations in digital form.

- We will be using the python programming language and using pygame as the GUI interface of the program
- Using ai it will make a decision based on the property it lands on after its roll, how much money it has and what properties it has.
- The UI will look like a standard implementation of digital monopoly. The board will take up a majority of the screen space. Action buttons and information about all players owned property and money will be displayed above the board.
- We will carefully plan our system architecture to be able to easily integrate sounds and animations, however this is not a strict requirement in our design.
- A modified version of monopoly where games are limited to a set amount of rolls (in order to reduce the total time it takes to finish a game) is being considered.
- The saved state of the game will be a custom implementation that will be generated and interpreted by the save() and load() functions. The necessary components to save are the player information (position, money, and jail cards owned), status of each property on the board, and which players turn it is.

# User Stories

- As a player, I want to roll the dice and move my game piece on the board to simulate the movement of my character.
- As a player, I want to be able to purchase properties when I land on them, so that I can collect rent from other players who land on them later.
- As a player, I want to be able to build houses and hotels on my properties, so that I can increase the rent I collect from other players.
- As a player, I want to be able to mortgage my properties to receive immediate cash, in case I need to pay debts or make strategic decisions.
- As a player, I want to be able to trade properties and other assets with other players, so that I can negotiate and strategize to my advantage.
- As a player, I want to be able to participate in auctions when a property is not purchased by the player who landed on it, so that I can have a chance to acquire the property.
- As a player, I want to be able to pay rent when I land on another player's property, so that I can continue playing the game.
- As a player, I want to be able to collect rent when other players land on my properties, so that I can accumulate wealth and increase my chances of winning.
- As a player, I want to be able to draw chance and community chest cards, which can provide opportunities or setbacks in the game.
- As a player, I want to be able to participate in jail mechanics, such as paying bail or attempting to roll a double to get out of jail, so that I can strategize and plan my moves effectively.
- As a player, I want to be able to strategically use "Get Out of Jail Free" cards, which can be obtained through chance cards or traded with other players, to avoid paying bail or rolling doubles to exit jail.
- As a player, I want to be able to bankrupt other players by acquiring all their properties and leaving them with no money, so that I can eliminate them from the game and increase my chances of winning.
- As a player, I want to be able to manage my finances, including tracking my cash, properties, and debts, so that I can make informed decisions and stay competitive in the game.
- As a player, I want to be able to track the progress of other players, including their properties, cash, and assets, so that I can plan my strategies accordingly.
- As a player, I want to be able to keep track of the properties that have been mortgaged by other players, so that I can identify potential opportunities to purchase those properties or plan my strategies accordingly.
- As a player, I want to be able to utilize special cards or power-ups that provide unique advantages, such as the ability to skip turns, block opponents from purchasing properties, or decrease rent prices on my properties.

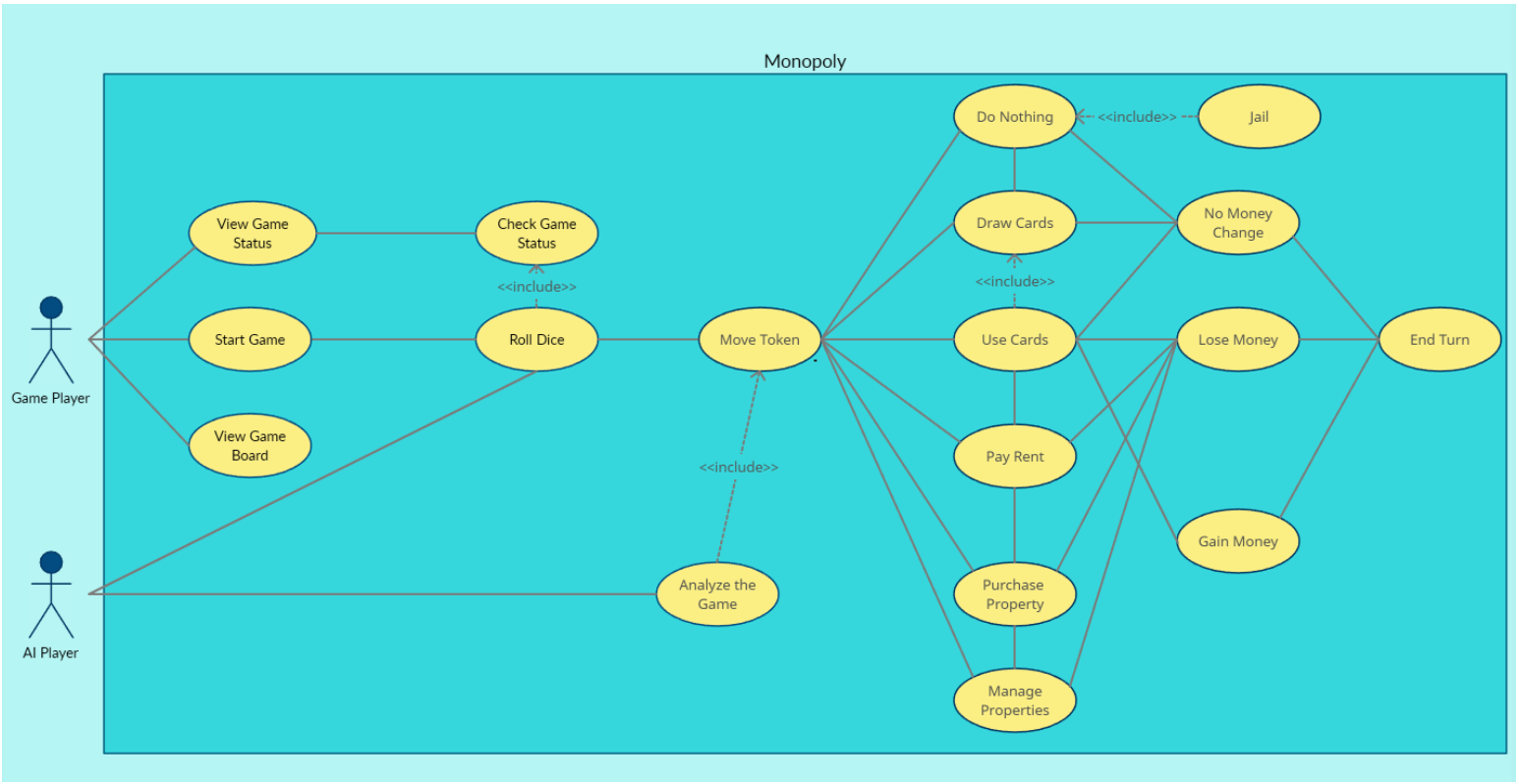
- As a player, I want to be able to view the game board from different angles or zoom in/out to get a better perspective, allowing me to analyze the board and plan my moves more effectively.
- As a player, I want to be able to save the progress of a game and resume it at a later time, enabling me to play multiple sessions without losing our current positions and progress.
- As a player, I want to be able to access a comprehensive rulebook or tutorial within the game, providing clear instructions and explanations for all the game mechanics, to help me understand and learn the game quickly.
- As a player, I want to be able to win the game by acquiring the most properties, accumulating the highest net worth, or being the last player remaining after bankrupting all other players.

## Functional and Non-Functional Requirements

| Functional requirement | Non-functional requirement |
|---|---|
| The game must display all essential elements to the player, such as the board, the pieces, the amount of money, etc. Also, the game must update these elements after each turn. | The game elements, such as the pieces and the amount of money, should be easy for the human player to identify. |
| The player should be able to roll dice and move his/her piece when it is his/her turn. | The time that each AI player needs for each turn should be around 1-2 mins. |
| The player should be able to purchase property or build house/hotel. | The AI player will not agree with unfair trade with other player unless it has no money available. |
| The player should be able to draw a chance or community card. | The number of concurrent players in a game is 2-8. |
| The player should be able to pay the rent when reach other's property. | The game should provide an user interface to the human player when in his/her turn to allow him/her to select the actions. |
| The player should be able to receive the rent when other player reach his/her property. | The game should provide backup save for the player to continue later. |

| | |
|---|---|
| The player should be able to trade property with other player(s). | The game should be available whenever the user wants to play a round or continue from a save file. |
| The player should be able to leave jail through different approaches, either pay or roll a double. | The game must perform without any error in almost all scenarios mentioned in the user stories. |
| The AI player should be able to do all the actions mentioned above, which are the same as the human player. | The game will pop-up the achievement that human player earned when the he/she reach certain condition during the game. |
| The game will announce a player is lost when he/she has neither money nor property. | |
| The game must announce the only remain player who is not bankrupted as the winner. | |

## UML Use-Case Diagram



# UML Class Diagram