



Artificial Intelligence Project

Intelligent Mancala Game

Submitted By:

Ayman Nasr Zaki - 1600359

Mahmoud Mohamed Anwar - 1601323

Amr Mohamed Mohamed Ahmed Elersy - 1600942

Ahmed Abd El Moamen Mohamed Ali - 1600119

Mostafa Ahmad Mohamed Mohamed - 1601397

Description

The project is a console-based Mancala game that is built from scratch using python and supports AI software agent that uses min-max alpha beta pruning algorithm.

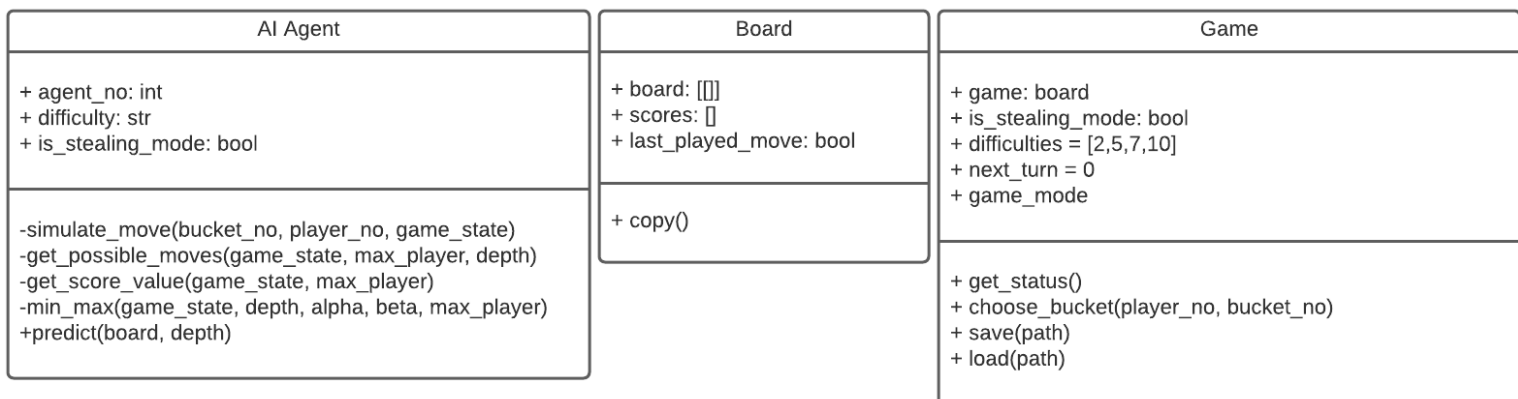
The game supports 2 modes:

- Stealing mode
- Non Stealing mode

Also the game supports:

- Four difficulty levels (easy, medium, hard & very hard)
- Two human players against each other
- AI software agent against human player
- Saving & Loading the game status

Class diagram



Utility functions

Game class:

get_status()	A function that prints to the console the game status after each move
choose_bucket()	A function that chooses a bucket from which to move the mancala rocks,

	all game rules are implemented in this function
save()	A function that saves the game automatically after each move as a pickle file
load()	A function that loads the last game saved

AI agent class:

simulate_move()	A function that simulates a move like choose_bucket but without updating the actual board_state, it only simulates the move for the min-max tree building
get_possible_moves()	A function that takes a board_state and returns all the possible childs from this state, used for tree building
get_score_value()	A function that calculates the leaf nodes scores based on: <ul style="list-style-type: none"> ● Player's Mancala score ● Player's total side rocks
is_terminal_state()	A function that checks if the sent state is a terminal state or not, a terminal state is the state that there is a player side that has no rocks left to play
min_max()	A function that implements the min-max alpha beta algorithm
predict()	A function that takes a board_state and returns the best move based on the difficulty level used

Board class:

copy()	A function that deep copies the board_state to be used for simulating a move
--------	--

User guide

1. Choose 2 human players mode, AI vs Human with the option of whom starts first or load the last saved game state

```
Choose Your Game Mode:
0 - For two player
1 - Vs AI, You start First
2 - Vs AI, AI start with first
3 - Load your last play
Input is: █
```

2. Choose either stealing mode or non-stealing mode

```
Choose Your Game Mode:
0 - For two player
1 - Vs AI, You start First
2 - Vs AI, AI start with first
Input is: 0
Enter 0 for non-stealing mode or 1 for stealing mood : █
```

3. Choose the difficulty level

```
Choose Your Game Mode:
0 - For two player
1 - Vs AI, You start First
2 - Vs AI, AI start with first
Input is: 0
Enter 0 for non-stealing mode or 1 for stealing mood : 0
Enter:
0 - Easy
1 - Medium
2 - Hard
3 - Very Hard
Input is: █
```

4. The game status is printed waiting for the player turn if chosen to play first

```
***** Difficulty is Easy *****
***** Computer score : 0 *****
***** Player 1 score : 0 *****

Board State
[[4 4 4 4 4]
 [4 4 4 4 4]]

Player 1 Choose bucket to Move : █
```

5. The game status is printed also after the agent takes an action

```
***** Difficulty is Easy *****
***** Computer score : 0 *****
***** Player 1 score : 0 *****

Board State
[[4 4 4 4 4]
 [4 4 4 4 4]]

Player 1 Choose bucket to Move : 0
***** Computer score : 0 *****
***** Player 1 score : 0 *****

Board State
[[4 4 4 4 4]
 [0 5 5 5 4]]

***** Computer Turn *****

Action Taken By AI at Depth: 2 = 0.015677213668823242s
***** Computer score : 1 *****
***** Player 1 score : 0 *****

Board State
[[0 4 4 4 4]
 [1 6 6 5 5 4]]

***** Player 1 Turn *****

Player 1 Choose bucket to Move : █
```

Team Work

Design Phase	All team members contributed in the design phase
Implementation Phase	Game rules & board state: Ahmed & Mahmoud Min-Max with alpha beta pruning: Ayman Helper functions and integration with min-max: Ayman, Amr & Mostafa Integrating the game with the agent: Ahmed & Mostafa
Testing Phase	Amr, Mostafa & Mahmoud
Deployment Phase	Deploying the project as executable: Ayman

Github repository link:

<https://github.com/AymanNasser/Intelligent-Mancala-Game>