Simple Checkers AI

Hyunje Kim

Sagar Ranga

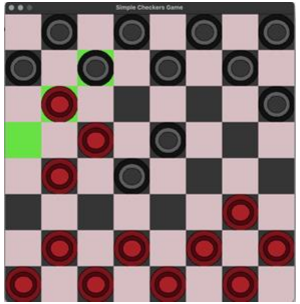
Problem Statement

Goal: To develop an AI powered Checkers game with different difficulty levels to engage a player and offer a good learning/playing experience.

Use Cases

A new player could play against an AI where the difficulty would be a lot easier and the player could practice strategies against an easier opponent. Eventually the player would increase difficulty to challenge themselves

AI Algorithm and Model



Beginner (Random)

- Valid moves
- Random selection

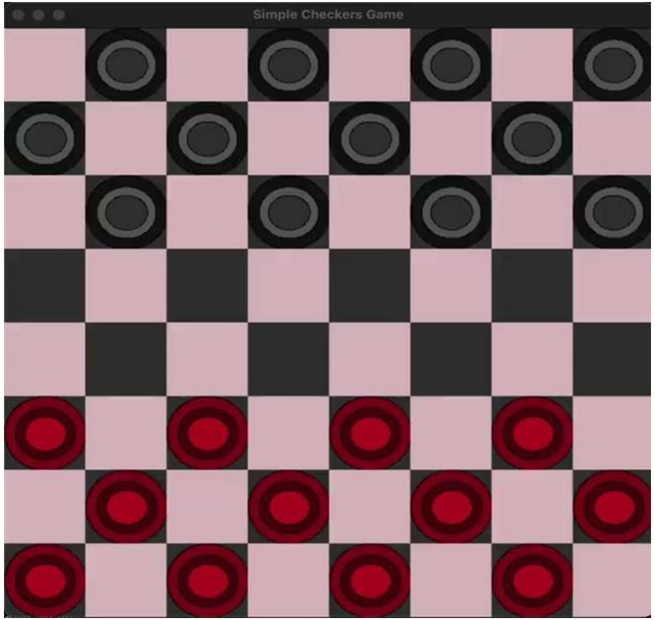
Intermediate (Point aggressive)

- Valid capturing moves
- Random selection

Advanced (Minimax + Alpha-Beta Pruning)

- Evaluate board
- Tree search multiple moves ahead
- Minimize possible loss for worst case scenario
- Cut off branches

Results and Demonstration



Analysis

Beginner (Random)

- Pretty easy to beat but can sometimes raise a challenge

Intermediate (Point aggressive)

- Quite difficult to beat when new to the game.

Advanced (Minimax + Alpha-Beta Pruning)

- Provides a challenge but has rare points where it gets stuck in a corner after making a king.

Lessons Learned

- AI needs to perform better on advanced levels
  - Checkers game have descent search space, so improve on existing evaluation and value-assigning logic.
  - Improve board evaluation function
- AI integration could be better with highlights and forced jumps.
- Implement better alpha-beta pruning
- Implement more algorithms to fine tune AI difficulty furthermore
- Learnt a lot about working with pygame and gui design

Q&A