



AI plays checkers

Jesus Pardo - 22202392

Introduction

- **Create an AI bot that can play Checkers against other bots or real players.**
- **Create and interface for playing against the AI or see how two AI bots can play against each other.**

Methodology

- **Create a checkers game and board using the python module *pygame*.**
- **Use the algorithm Minimax programmed in python 3.9.16 to decide the movements of the AI.**
- **Optimize the algorithm Minimax using Alpha-Beta pruning**

Methodology: Pygame

- **Graphics Rendering**
- **Input Handling**
- **Event Management**
- **Animation and Effects**
- **Sound and Music**
- **User Interface (UI)**
- **Game Logic and State Management**
- **AI Integration**
- **Customization and Styling**
- **Cross-Platform Compatibility**

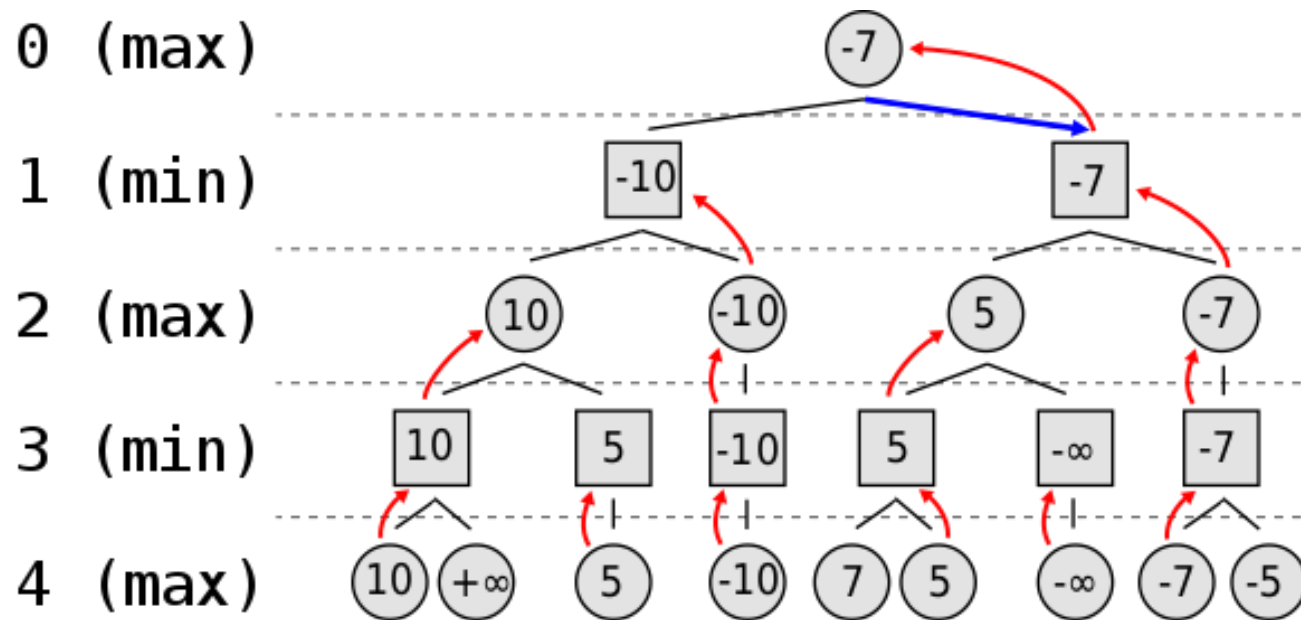
Methodology: Minimax algorithm

- **Objective**
- **Two-Player, Zero-Sum Games**
- **Recursive Approach**
- **Maximizing and Minimizing**
- **Depth-Limited Search**
- **Challenges and Considerations**
- **Evaluation Function**
- **Backpropagation**

Methodology: Minimax algorithm

Evaluation function:

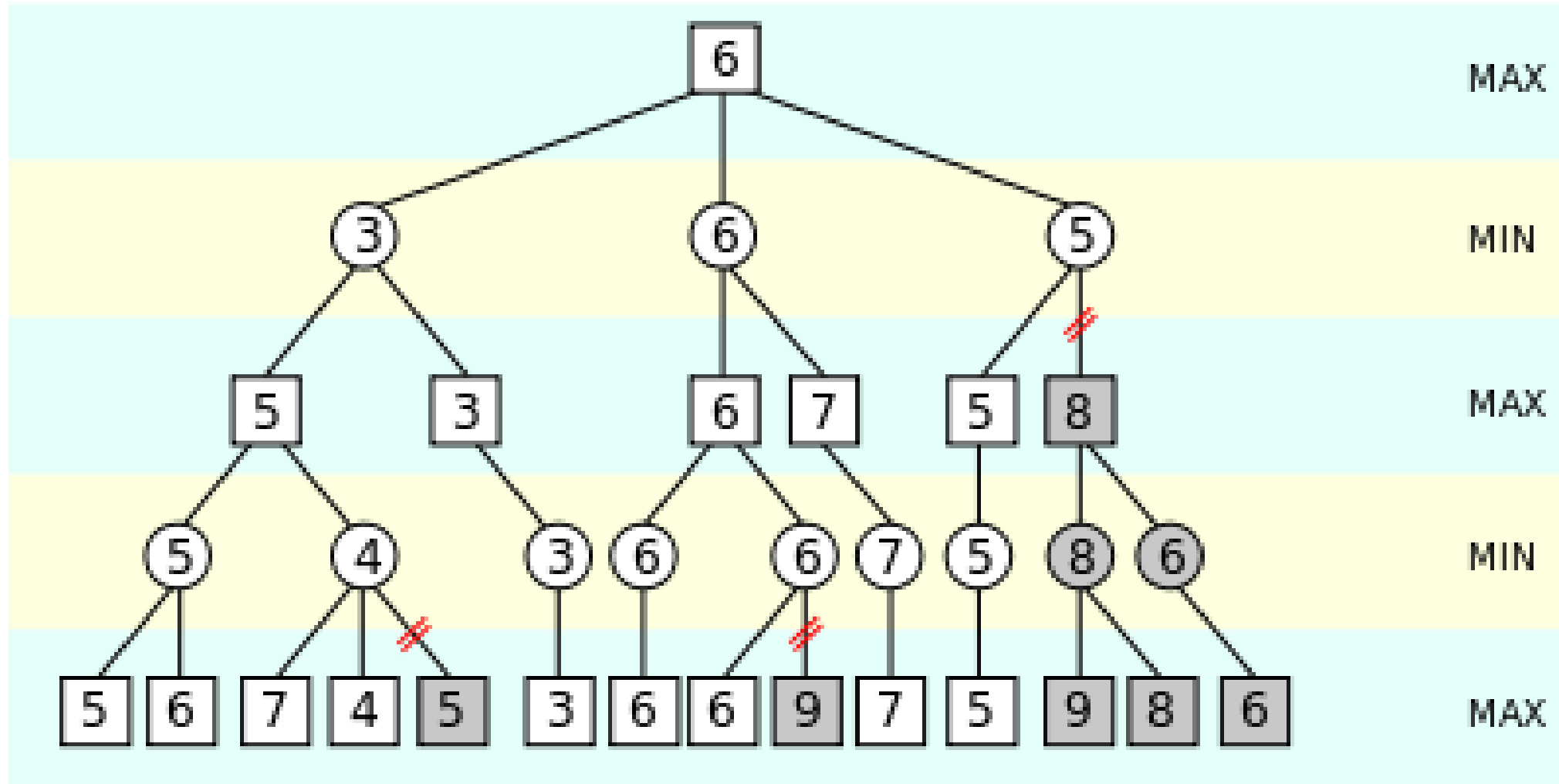
$$\text{Max_player pieces} - \text{Min_player pieces} + (\text{Max_player kings} * 0.5 - \text{Min_player kings} * 0.5)$$



Methodology: Alpha-Beta pruning

- **Optimization Objective**
- **Node Evaluation Order**
- **Alpha and Beta Values**
- **Pruning Conditions**
- **Maximizing Player (Alpha)**
- **Minimizing Player (Beta)**
- **Efficiency Gain**
- **Depth-First Search:**
- **Impact on Complexity**

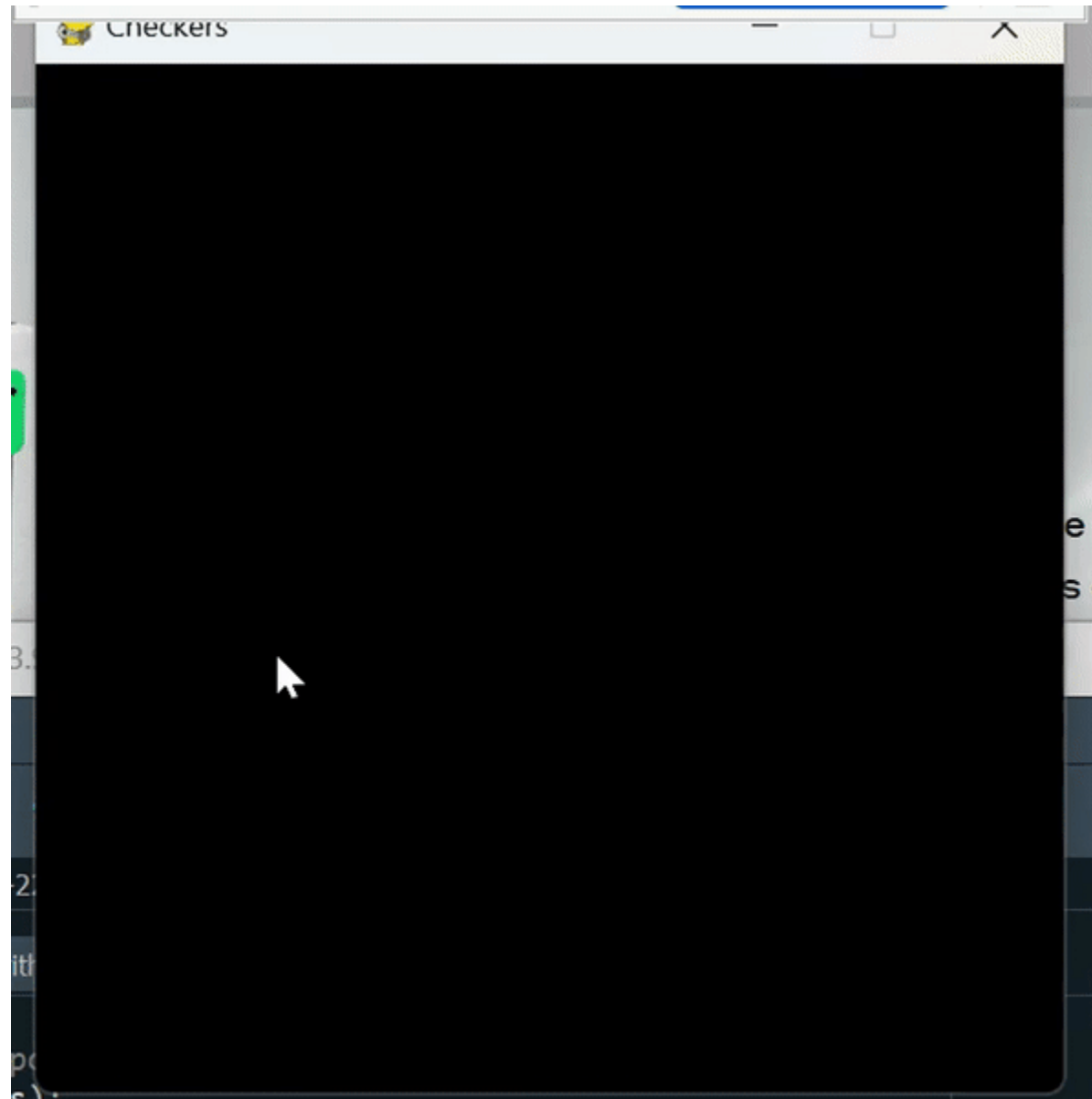
Methodology: Alpha-Beta pruning



Methodology: Repository structure

- **Code Organization and Readability**
- **Modularity and Reusability**
- **Collaboration and Teamwork**
- **Ease of Maintenance**
- **Scalability**
- **Version Control and History**
- **Documentation**
- **Testing and Quality Assurance**
- **Reduced Technical Debt**

The game



Possible Extensions

- **Deep Reinforcement Learning (DRL)**
- **Monte Carlo Tree Search (MCTS)**
- **Ensemble Learning**
- **Dynamic Learning Rates**
- **Curriculum Learning**
- **Hierarchical Reinforcement Learning**
- **Reward Shaping**
- **Exploration Strategies**
- **Adversarial Training**
- **Human-AI Collaboration**
- **Visual Recognition**
- **Multi-Agent Learning**
- **Real-Time Learning**

Thank you for your attention