Quoridor AI Battle

PPCA 2022

What can you learn from this project?

What can you learn from this project?

- **Week 1 ~ 2:**
  - Gain a solid understanding of game tree search algorithms (**Minimax** & **MCTS**)
  - Practice various **pruning and techniques** for optimizing your search
  - Improve your coding skills and experience how powerful AI is in board games

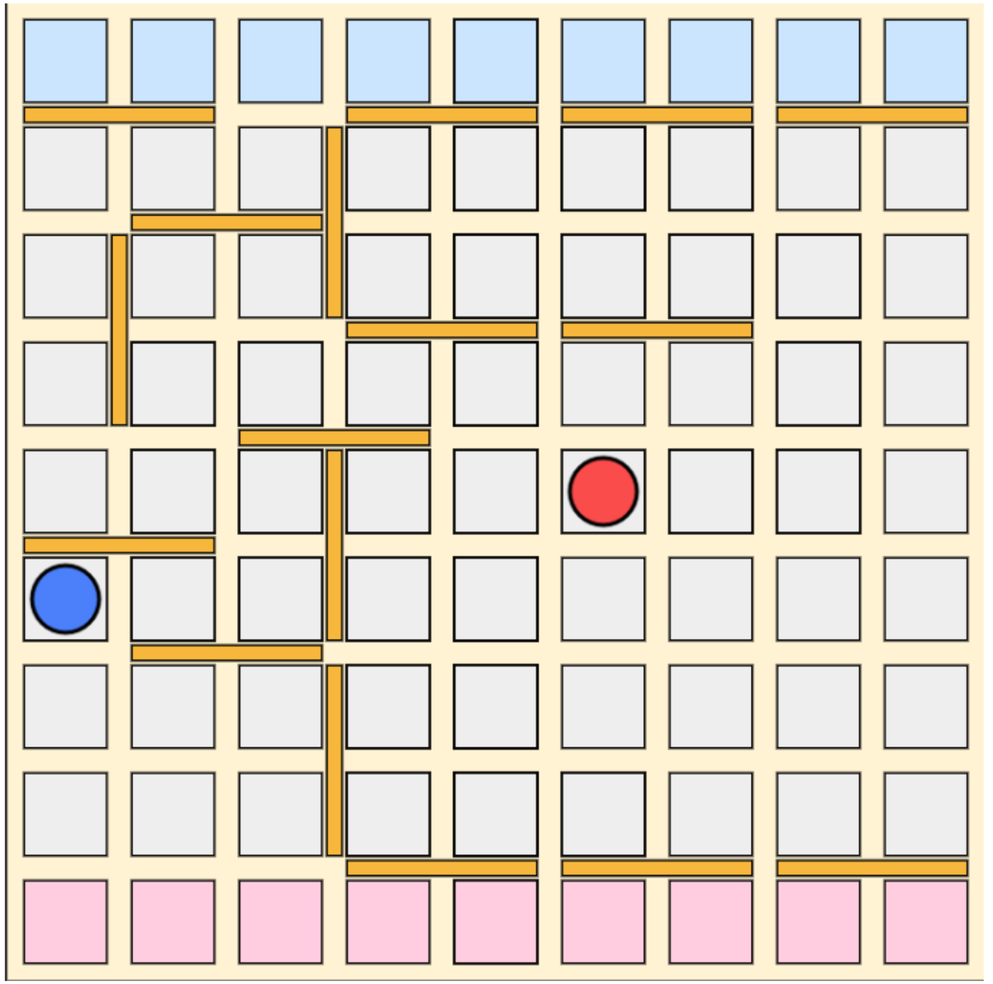
What can you learn from this project?

- **Week 1 ~ 2:**
  - Gain a solid understanding of game tree search algorithms (**Minimax** & **MCTS**)
  - Practice various **pruning and techniques** for optimizing your search
  - Improve your coding skills and experience how powerful AI is in board games
- **Week 3 ~ 4:**
  - Understand the basics of **Neural Network** and classical neural network architectures
  - Learn the basic usage of **Python** and **Pytorch**
  - Write and train your first(?) neural network model

Week 1 ~ Week 2

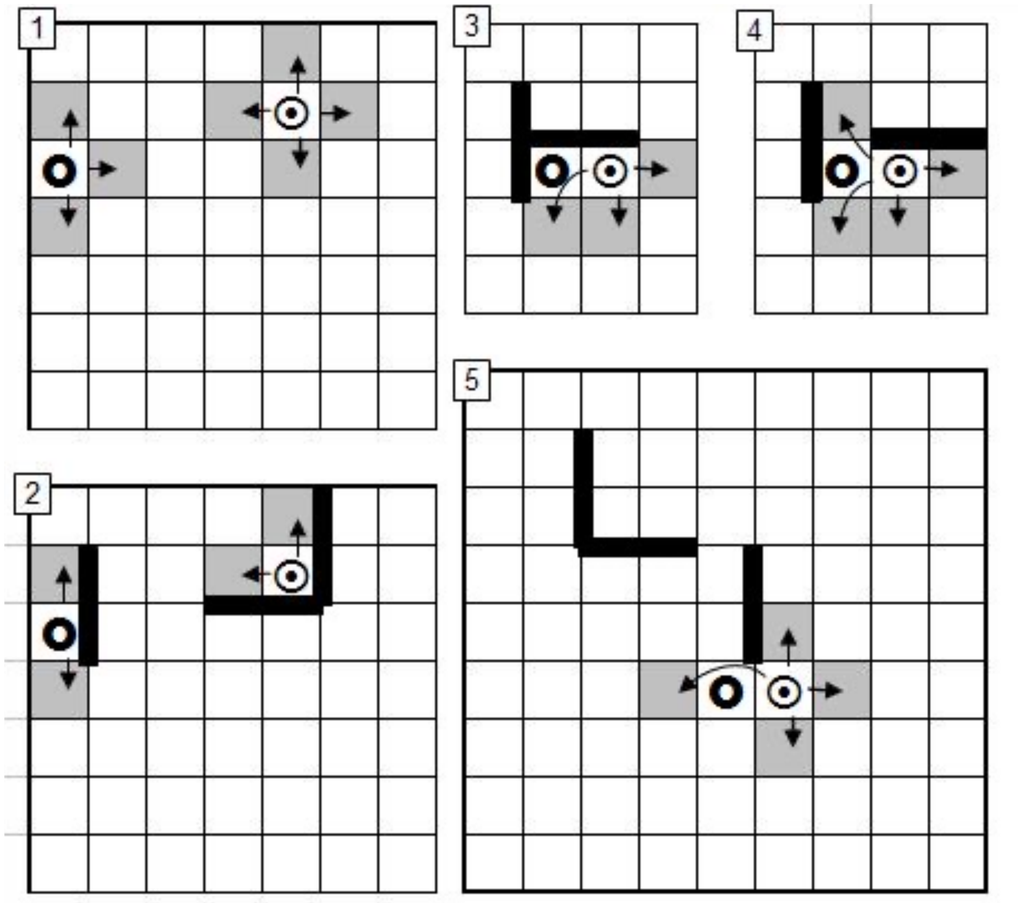
Game Rules of Quoridor

- Play on a 9x9 game board.
- Each player is represented by a pawn (1x1).
- The objective is to be the first player to move their pawn to opposite side of the game board.
- Each player has 10 two-space-wide walls which can be placed in the groove that runs between the spaces.
- Walls block the path of all pawns, which must go around them.

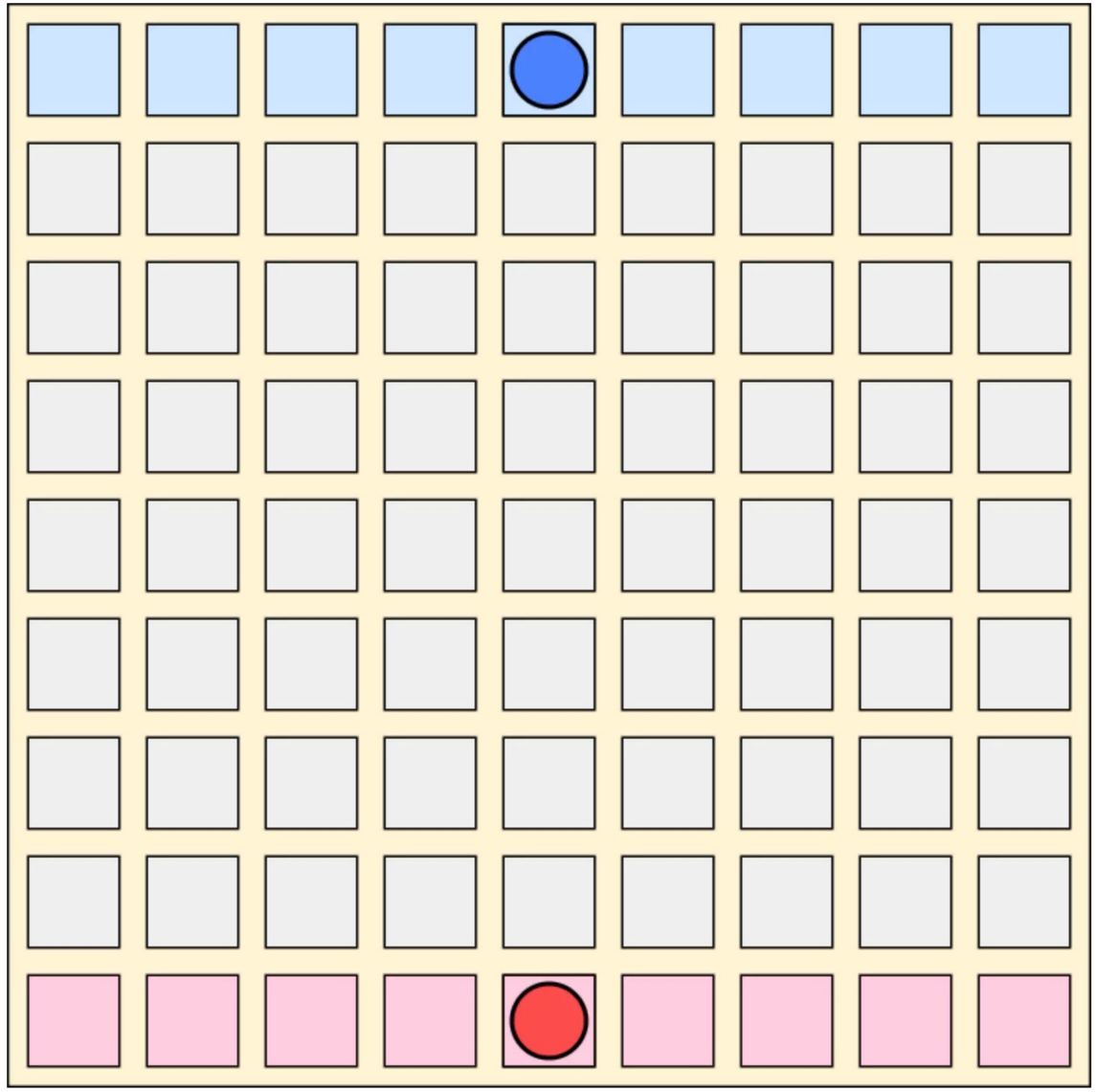


Game Rules of Quoridor

- Each turn a player may either move their pawn, or, if possible, place a wall.
- If adjacent to another pawn, the pawn may jump over that pawn.
- A wall must not be placed which cuts off the only remaining path of any pawn to the side of the board it must reach.



Game Rules of Quoridor



第0步

先手： Sakits

Artificial_Idiot_MCTS

剩余挡板数目:10

后手： Sakits Sakits_Minimax

剩余挡板数目:10

ai0 wins!

■ ‹ ▶ ›

stderr0

```
 act: 0 8 5
rate: 0.491405 1109
act: 0 8 3
rate: 0.48964 585
act: 0 7 4
rate: 0.494986 17940
```

stderr0

```
 0 7 4
vsize:19
0 1 4
0 6 4
vsize:24
0 2 4
```

What you need to do?

What you need to do?

- **Week 1**: Create your AI using game tree search algorithms
  - **Minimax Search** & **Alpha-beta Pruning**
  - **Monte-Carlo Tree Search**

What you need to do?

- **Week 1**: Create your AI using game tree search algorithms
  - **Minimax Search** & **Alpha-beta Pruning**
  - **Monte-Carlo Tree Search**
- **Week 2**: Try to defeat others' AI to get better ranking **with your wisdom**
  - Implement more advanced **pruning techniques**
  - Well-tune your **evaluation function**
  - **Analyze the weaknesses of others' AI** for targeted attacks
  - Apply human intelligence to **design some strong openings**
  - ...

Requirements

- **Language support**: C++ and Python
- **Code length limit**: 50 kb
- **Time limit per turn**: 2s
- **Prohibition**:
  - Open files
  - Hack the server
  - Copy codes from the Internet
    (No AI can be found online that can beat baselines)

Scoring

- **Implement the algorithm (code review)**: 60 %
- **Beat baseline (score by gradient)**: 20 %
  - Average 75% win rate against weak baselines for full score
  - Average 40% win rate against strong baselines for full score
- **Final ranking**: 20 %

- PS: Above scores account for 70% for ACM Class

Week 3 ~ Week 4

Learning Goals

- Learn to program in **Python**
- Understand the basic usage of **Pytorch**
- Explore how **Neural Networks** work
- Have a glimpse of some classic neural network architectures
  - CNN
  - ResNet
  - Transformer
  - ...

What you need to do?

- **Week 3**: **Learning**
  - Follow the given guide and tutorial to learn **Python** and **Pytorch**
  - Learn the basics of **Neural Network**

What you need to do?

- **Week 3**: **Learning**
  - Follow the given guide and tutorial to learn **Python** and **Pytorch**
  - Learn the basics of **Neural Network**
- **Week 4**: **Train a policy neural network** for 9x9 Gomoku
  - The dataset and all neural network unrelated code will be released
  - Only need to design your own NN Architecture and train it
  - Construct game-specific feature planes to improve the performance

Scoring

- **Understanding of NNet (code review)**: 20 %

- **Model performance (beat baseline)**: 10 %

Thanks!

Q & A