README

CSC 242 Project 1

Name: Huilin Piao hpiao2@u.rochester.edu

31529129

Collaborator: Mengmeng Yu myu26@u.rochester.edu

31527057

Part 1

In this part we creates a 3*3*3 ConnectFour with three classes: State, Board, and ConnectFour State:

We use a state class to restore input into an array list.

Board:

In Board class, we first use a 3*3 arraylist to restore all the positions in the chess board and update the chess board whenever user and agent plays the chess. Who goes first will be assigned to x pieces and the other one is assigned to o pieces.

Next, we create a row_index method to make sure that when player choose a column to play, the chess always falls to the bottom. We also make sure that the arrayList getSpace() which restores all the available positions that agent can play on only contains positions that is at the bottom of the chess board. We use isEnd() and isWin() method to record the result of each game. Once the game tends to have a winner or finally tie, method returns true.

The most important method in Board class is minimax function. Minimax() is used to calculate the optimal movement for the agent. For each turn, agent will calculate the score for each available positions by looking at the next step opponents can play with depth = depth + 1. Agent then chooses the best position that maximizing its score and minimize players' profit. If

player will win afterward, the current score returns -1. Oppositely, the score returns 1. Otherwise, score returns 0. Once minimax function finds the optimal position, we set Agent play on it.

ConnectFour:

In this class, our main method requests player to input letters and transfer it into a certain position in the chess board. Then we store it into ArrayList State and update the action in chessboard. We use two variables, step and first to record the process of the game. Step increases for every turn. If player chooses red, player plays first and plays whenever step%2 = 1. If player chooses yellow, it will work oppositely.

ConnectFour also recalls the calAction() and boardupdate() method from board class to show game process to player.

Part 2

In this part we create a 6*7*4 ConnectFour with three classes, State, BoardP2 and ConnectFourP2.

State:

Same as Part 1

Board:

Mostly same as Part 1 but implementing an alpha beta pruning algorithm to reduce the time for searching. It enables the project to return best score without searching for the whole board. We also set a depth limit of 5. If agent or player cannot win in this depth limit, the agent will choose more optimal position. If we don't set a depth limit, the project still follows structure of alpha-beta pruning algorithm. However, it might take a long time for project to search for all positions and generate the optimal one.

ConnectFour:

Mostly same as Part 1 but we expand the chess board to a large 6*7 size.

How to run:

javac ConnectFour.java ConnectFourP2.java State.java Board.java
BoardP2.java
java ConnectFour
Java ConnectFourP2

Part 1 Example: a b c 1 2 Do you want to player RED (1) or YELLOW (2)? Next to play: RED/X Your move [column]? a b c 1 | | 2 | | 3 X | | Next to play: YELLOW/0 1score-1 2score-1 2score0 I'm thinking... Best move:2 a b c 1 | | 2 | | 3 X | | 0 Next to play: RED/X Your move [column]? а a b c

```
1 | | | ---+---
2 X | |
---+---
3 X | | 0
Next to play: YELLOW/0
0score1
I'm thinking... Best move:0
a b c
1 0 | |
---+---
2 X | |
3 X | 0
Next to play: RED/X
Your move [column]?
a b c
1 0 | |
2 X | X
---+---
3 X | | 0
Next to play: YELLOW/0
0score1
I'm thinking... Best move:2
a b c
1 0 | | 0
---+---
2 X | X
3 X | | 0
Next to play: RED/X
Your move [column]?
a b c
1 0 | 0
---+---
3 X | X | 0
Next to play: YELLOW/0
1score1
I'm thinking... Best move:1
a b c
1 0 | | 0
2 X | 0 | X
---+---
3 X | X | 0
Agent win!
```

```
Example for part 2:
MENGMENGtekiMacBook-Pro:Desktop mengmengyu$ java ConnectFourP2
           С
               d
                   e f
                           g
1
2
3
4
5
6
Do you want to player RED (1) or YELLOW (2)?
Next to play: RED/X
Your move [column]?
d
           С
               d
                   e
1
2
3
4
5
             | X |
6
Next to play: YELLOW/0
Computer choose4score0
Computer choose5score0
Computer choose5score0
Computer choose5score0
Computer choose5score0
Computer choose5score0
Computer choose5score0
I'm thinking... Best move:6
       b
           c d
                   e
2
```

3

| | | L | L | L | L | - | L |
|----------------------------------|--|--------------------|---|--|--|-----------------------|--------------|
| 4 | | | | | | | |
| 5 | | | | | | | |
| | | to p nove | | | | | 0 |
| 1 | а | b I | c I | d I | e | f | g I |
| 2 | | | | : | | | |
| 3 | | | : | : | | | |
| 4 | | | | | | | |
| 5 | | + | | | - | | |
| Co Co Co Co Co Co | omputomputomputomputomputomputomputomput | ter o | choos choos choos choos choos | se4s of se4s of se4s of se5s o | core(core(core(core(core(core(|))))) | : 4 g |
| 1 2 | | ⊦ | ⊦ | ⊦ | ⊦ | ⊦ | |
| 3 | | ' ⊦ | ¦ ⊦ | ' ⊦ | | | |
| 4 | | + | | ' } | + | | - |
| 5 | | + | | + | | | ⊦ |
| | | to p | lay: | X RED, | 0 /X | X | 0 |
| 1 | a | b | c | d | e | f | g I |
| 2 | | + | + | + | + | | ⊦ |
| 3 | | + | + | + | + | | ⊦ |
| 4 | | + - | | + | | ⊦ | ⊦ |
| | | + | - | - | - | - | |

| 5 | ļ | | | | | | |
|----------------------|-------------------|--------------------------------------|---|----------------------------------|----------------|---------------------------|---------------|
| Com Com | put put put | o pi er d er d er d inki | lay: choos choos choos ing. | X YELL Se4s c Se4s c | OW/OcoreOcore1 |))) L nove: | |
| 1 | a | b | | d | e | f | g |
| 2 | , | | | | | | |
| 3 | - | | | | | | |
| 4 | - | | | | | | |
| 5 | - | | | | 0 | | |
| 6 Nex You e | | | lay: | X RED/ Lumn] | ′X | X | 0 |
| 1 | a | b | С | d | e | f | g |
| 2 | + | + | | ⊦ | + | | |
| 3 | + | | | ⊦ | - | ⊦ | |
| 4 | + | | | ⊦ | X | | |
| 5 | + | | | | 0 | | |
| Com Com | put put | er o | lay: choos choos | YELL Se2sc Se4sc | ore0 |) L | 0 :2 |
| 1 | a | b | С | d I I | e | f | g |
| 2 | + | | | | | | |
| 3 | + | + | | ⊦+ | | | |
| 4 | + | | | ⊦ | Χ | | |
| 5 | + | | 0 | | 0 | - | |
| | | o pi | X Lay: | X X RED/ Lumn] | 0 ′X | | |

| 1 | а | b | | d | | f | g |
|----------------------------|--|--|--|--|---|----------------------------|---------------|
| 2 | | | | | | | |
| 3 | | | | + | | + | |
| 4 | | | X | | X | + | |
| 5 | | | 0 | | 0 | | |
| Co | mput | to pi ter d ninki | X lay: choos ing. | YELL Se2so J Be | 0 _0W/(core: |) l | • |
| 2 | | ⊦ | | ⊦ | | + | + |
| 3 | | ⊦ | 0 | | | + | + |
| 4 | | | ⊦ X | | X | ++ | |
| 5 | | | 0 | + | 0 | + | |
| | | l to p | X lay: | X RED, | 0 ′X | + X | 0 |
| 1 | a | b | c | d | e | f | g |
| 2 | | | | | | | |
| 3 | | | 0 | | | | |
| 4 | | · —¬ | X | . —¬ | X | . — | - |
| 5 | | , ——¬ | 0 | X | 0 | , ——¬ | - |
| Co Co Co Co Co | omputomputomputomputomputomputomputomput | ter of te | lay: choos choos choos choos choos choos choos | X YELL se1sc se2sc se4sc se4sc se5sc d | OW/(core(core(core(core(core(core(core(core |)))))) | :1 g |

| 2 | 1 | | | | | |
|--|--|--|--|--|---|------------|
| 3 | ! ! | 0 | + : | +· : | +· : | + |
| 4 | | X | • | X | +· | + |
| 5 | | 0 | X | + 0 + | +· | + |
| Your | 0 to p move | | X RED <i>,</i> | 0 /X | X | 0 |
| g a 1 | b | c | d I | e | f | g |
| 2 | .+ ! | | ! | +· : | +· ! | + |
| 3 | | 0 | | - | +· | + |
| 4 | | X | | X | + | + |
| 5 | | 0 | • | 0 | | X |
| 6 Next Compu Compu | to p | choos | YELI se1s | _0W/(core(| D | 0 |
| Compu Compu Compu | iter (iter (iter (| choos choos choos | se3so se3so se4so | core(core(core: | 0 0 1 | |
| Compu Compu Compu | iter (| choos choos choos ing | se3so se3so se4so | core core core: est i | 0 0 1 | :1 g |
| Compu Compu Compu I'm t | iter o iter o iter o :hink: | choos choos choos ing | se3so se3so se4so Bo | core core core: est i | 0 0 1 nove | |
| Compu Compu I'm t a 1 | iter o iter o iter o :hink: | choos choos choos ing c | se3sose4sose4so d | core(core) core core est (| 0 1 move f + | |
| Compu Compu I'm t a 1 2 | ter (iter (ite | choos choos ing c | se3se se4se se4se d + + | core(core) core core est i e + + X | 0 1 move f + | |
| Compute Computer and a second and a second a sec | ter (| choos choos ing c | se3sose4so se4so d X | core(core) core(core) est (core) | 0 1 move f + + | |
| Comput Comput I'm to a 1 2 3 4 5 Next Your | ter (iter (ite | choos choos ing c | se3se se4se se4se d X X RED, | core(core(core)) core(core) est (core) est (core) | 0 1 move f + + + | g |
| Computation Comput | ter (iter (ite | choosehoosehoosehoosehoosehoosehoosehoos | se3sese3sese4sese4sese4sese4sese4sese4s | core(core(core)) core(core) est (core) est (| 0 1 1 move f + + + X f | g |
| Comput Comput I'm t a 1 2 3 4 5 Next Your f a | ter (iter (ite | choosehoosehoosehoosehoosehoosehoosehoos | se3se se4se se4se d X X RED, Lumn] | core(core(core)) core(core) core(core) est (core) est (| 0 1 1 move f + + + X f | g |
| Comput Comput Comput I'm to a 1 | ter (iter (ite | choosehoosehoosehoosehoosehoosehoosehoos | se3sese3sese3sese3sesses | core(core(core)) core(core) core(core) est (core) est (| 0 1 1 move f + + + X f | g |

| 5 | | | 0 | | 0 | X | X |
|--------|--------------|-----------------------------------|-----------|--------------------|----------------|-----------------|----------|
| 6 | - | | | | 0 | | 0 |
| | xt t | o p | lay: | YELI | 0W/(core1 | | • |
| | m th | ninki | ing | | est n | nove: f | |
| 1 | a | | | | | | g |
| 2 | | | 0 | | | | |
| 3 | | | 0 | | | | |
| 4 | | | Х | | X | | |
| 5 | | | 0 | | 0 | X | X |
| Yo | xt t | 0 co p | X Lay: | | 0 /X | X | 0 |
| g 1 | a | b | c | d I | e | f | g |
| 2 | | + | 0 | | ⊦ | | |
| 3 | 1 | | 0 | | + | | |
| 4 | | ' | Х | | Х | | X |
| 5 | | | 0 | • | 0 | X | |
| Со | mput | 0 co p ¹ cer o | X Lay: | X YELI se0s | .0W/(core: | X | |
| 1' | m tr a | nink: b | _ | . . Ве d | | nove: f I | : 2 g |
| 2 | | | 0 | | ⊦ | | |
| 3 | | | 0 | | + | | |
| 4 | | - | | | ¦ X | } | X |
| 5 | | 0 | | ├ X | | ├ X | |
| | xt t | 0 co p | X Lay: | | /X | X | 0 |
| a 1 | a | b | c 0 | | e | f | g |

| | + | ⊦ | | - | | | |
|-----------------------------------|--|------------------------|---|----------------------|---|--|--|
| 2 - | | | 0 ⊦ | | | | ⊦ |
| 3 _ | | | 0 | | | | |
| 4 _ | | | X | | X | | X |
| 5 | | 0 | 0 | X | 0 | X | X |
| Con | ct t iput | to pl | X Lay: | X YELL se2sc | 0 _0W/(core1 | X | |
| 1 | a | b | c 0 | d | e | f | g I |
| 2 | | ⊦——- | 0 | | | + | |
| 3 | | | 0 | ⊦ | 0 | | |
| 4 | | | ⊦ X ⊦ | | | | X X |
| 5_ | | 0 | 0 | ' X X ⊦ | 0 | X | X |
| 6 Nex You | t t | 0 to p ¹ | X Lay: | X RED/ | 0 ′X | | 0 |
| | 11 11 | iove | [CO | lumn] | · • | | |
| b 1 | a - | b | c 0 | d | e | f | g |
| b | | b | c 0 | d | e | f | g |
| b 1 _ | | b | c 0 0 | d | e 0 | | g |
| b 1 - 2 | | b | c 0 0 | d | e | | g |
| b 1 - 2 - 3 | | b | c 0 0 0 X | d | e | | |
| b 1 | a + + + X xt t | b | c 0 0 0 X X X X X | d | e 0 X X 0 0 0 0w/(core@ | X X | X X |
| b 1 | a + + + X xt t | b | C 0 0 0 X | d | e | X X | X X |
| b 1 2 3 4 5 6 Nex Com Com I'm | a | b | C 0 0 X X | d | e | X X) | X X 0 |
| b 1 | a | b | C 0 0 X X X C C C C C C C | d | e | X X) | X X 0 |
| b 1 2 3 4 5 6 Nex Com Com I'm 1 2 | a | b | C 0 0 0 X | d | e | X X) | X X 0 |

| | | | L | - | L | L | |
|----------|-----------|-------------------------|----------------------------|--------------------|------------------------|--------------------------|-------------|
| Ne Yo | xt 1 | 0 to p ¹ | X Lay: | | 0 ′X | X | • |
| a 1 | a | • | c 0 | | | f | g |
| 2 | | | 0 | | | + + | |
| 3 | | • | 0 | | 0 | | |
| 4 | | X | X | | X | • | X |
| 5 | X | 0 | 0 | | 0 | X | |
| Со | xt t | 0 to pl | X Lay: Choos | X YELL se1s | 0 _0W/(core: |) X | |
| 1 | a | | 0 | d | | f - | g |
| 2 | | 0 | 0 | | | | |
| 3 | | • | 0 | | 0 | | |
| 4 | | X | X | | X | • | X |
| 5 | Χ | 0 | 0 | | 0 | X | |
| | X xt 1 | 0 to p ¹ | X Lay: | | 0 ′X | X | 0 |
| 1 | а | b | c 0 | d | e | f | g |
| 2 | | 0 | 0 | | X | | |
| 3 | | 0 | 0 | | 0 | | |
| 4 | | X | X | | X | | X |
| 5 | Χ | 0 | | X | 0 | X | X |
| Со | mpu1 | to pi ter d ninki | X Lay: choos ing. | X YELL se0so | OW/(core: est n | X D 1 move: | |
| 1 | a | b 0 ⊦ | c 0 + | d ⊦ | e | f + | g ⊦ |
| 2 | | 0 | 0 | İ | X | l | l |

| | + | - | | - | <u> </u> | - | + |
|----------|-------------------|------------------------|--------------------|-----------------------------------|---------------------|-------------|--------------------------|
| 3 | | | | | | | |
| 4 | | X | X | | Х | | X |
| 5 | X | 0 | 0 | ⊦ X | 0 | X | X |
| Ne | X xt 1 | 0 to p ¹ | X Lay: | X X RED/ Lumn] | 0 ′X | | |
| 1 | a | 0 | 0 | d ⊦ | | | g |
| 2 | | 0 | 0 | | Χ | | |
| 3 | | 0 | 0 | | 0 | | |
| 4 | Χ | X | X | r==== -=== | Χ | | X |
| 5 | Χ | 0 | 0 | X | 0 | X | X |
| Ne Co | X xt 1 mpu1 | 0 to pl | X Lay: Choos | YELL | 0 _0W/(core: | X) L | |
| I' | | | _ | B∈ | | | |
| 1 | | 0 | 0 | | 0 | | g |
| 2 | | 0 | 0 | ⊦ | Х | | |
| 3 | | 0 | 0 | ⊦ | 0 | | |
| 4 | Χ | X | X | | Χ | | X |
| 5 | Χ | 0 | 0 | X | 0 | X | X |
| Ne | X xt t | 0 to p | X Lay: | X RED/ Lumn] | 0 ′X | + X | |
| 1 | | 0 | 0 | d | 0 | | g - |
| 2 | | 0 | 0 | | Х | | |
| 3 | | 0 | 0 | | 0 | | X |
| 4 | X | Х | Х | ⊦ | Χ | | + X + |
| 5 | | 0 | 0 | X | 0 | X | X |
| 6 Ne | Χ | 0 | X | X YELL | 0 | X | |

| | | ter d ninki | | | | | : 6 |
|---------|-------------------|--------------------------------|--------------------|---------------------|---------------------|---------------|---------|
| 1 | a | b 0 | С 0 | d | e 0 | f | g |
| 2 | | 0 | 0 | | Х | | 0 |
| 3 | | ⊦ 0 ⊦ | 0 | | 0 | | X |
| 4 | Χ | X X | Х | | Х | | X |
| 5 | Χ | 0 | 0 | Х | 0 | X | Χ |
| Ne | X xt 1 | 0 to p ¹ nove | X Lay: | X RED, | 0 ′X | | |
| 1 | | 0 | 0 | d | 0 | | |
| 2 | | 0 | 0 | | Х | | 0 |
| 3 | | | 0 | | 0 | | X |
| 4 | Χ | X X | Х | | X | | X |
| 5 | Χ | 0 | 0 | Х | 0 | X | Х |
| Со | X xt 1 mpu1 | 0 to pl | X Lay: choos | X YELI se2s (| 0 _0W/(core: | X) L | 0 |
| I' | m th a | ninki b | _ | B∈ d | | | :0 a |
| 1 | | | 0 | | 0 | | ΙΧ̈́ |
| 2 | | 0 | 0 | | Х | | 0 |
| 3 | 0 | | 0 | | 0 | | X |
| 4 | Χ | | Χ | | X | | X |
| 5 | X | 0 | 0 | Х | 0 | X | X |
| 6 Ag | | 0 win | Х | | | | 0 |