

CPCS331 – Artificial Intelligence – Spring2020 -Project I

[ Using Alpha Beta, Minimax and MCTS algorithms]

|  |  |  |
| --- | --- | --- |
| Student Name | Student Number | Section |
| Elham Imdad | 1779096 | AA |
| Ghadeer Qalas | 1778773 |
| Lena abdulmanan | 1705454 |

Contents

[Part 1 3](#_Toc35343142)

[1. Minimax and Alpha-Beta pruning technique to solve “Dots and Boxes” Puzzle 3](#_Toc35343143)

[1.1 Introduction 3](#_Toc35343144)

[1.2 Problem formulation as a search problem 3](#_Toc35343145)

[1.2.1 Initial state: 3](#_Toc35343146)

[1.2.2 Actions: 3](#_Toc35343147)

[1.2.3 Transition model: 4](#_Toc35343148)

[1.2.4 Goal test: 4](#_Toc35343149)

[1.2.5 Path cost function: 4](#_Toc35343150)

[1.3 Technical discussion 5](#_Toc35343151)

[1.4 Discussion of results 8](#_Toc35343152)

[1.5 Results 11](#_Toc35343154)

[Part 2 13](#_Toc35343155)

[2. Monte Carlo Tree Search technique to solve “Dots and Boxes” puzzle 13](#_Toc35343156)

2.1 MCTS Algorithm explanation…………………………………………..………………13

2.2 Example…………………………………………………………………………………14

[2.3 MCTS Algorithm Evaluation 18](#_Toc35343157)

Appendix…………………………………………………………………………………………19

3.1 References…………………………………………………………………………………19

3.2 Program listing…………………………………………………………………………….19

## Part 1

## 1. Minimax and Alpha-Beta pruning technique to solve “Dots and Boxes” Puzzle

## 1.1 Introduction

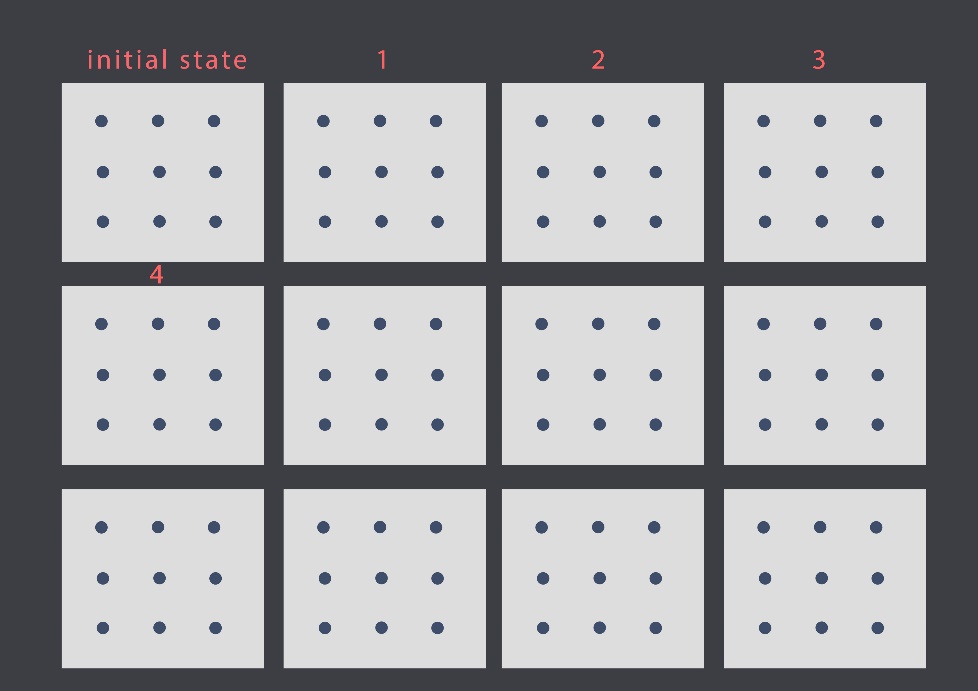
The problem that we’re trying to solve is Dots and Boxes problem. Dots and Boxes is a combinatorial game popular among children and adults around the world. In a game of Dots and Boxes, the players draw a rectangular grid of dots and take turns drawing lines between pairs of horizontally or vertically adjacent dots, forming boxes. A game’s size is defined in terms of the number of boxes. The player who draws the fourth line of a box captures the box. When this happens, the player gains a point and then must draw another line. At the end of the game, the player with the most points wins. If both players captured the same amount of boxes, the match is considered as a tie.

## 1.2 Problem formulation as a search problem

## 1.2.1 Initial state:

The game starts with an empty 3x3 dots matrix or it can be at any size. (note that any matrix larger than 6x6 is not solvable by the computer.)

Example of the initial state of 3x3 matrix:



## 1.2.2 Actions:

Connecting adjacent dots either vertically or horizontally in the grid.

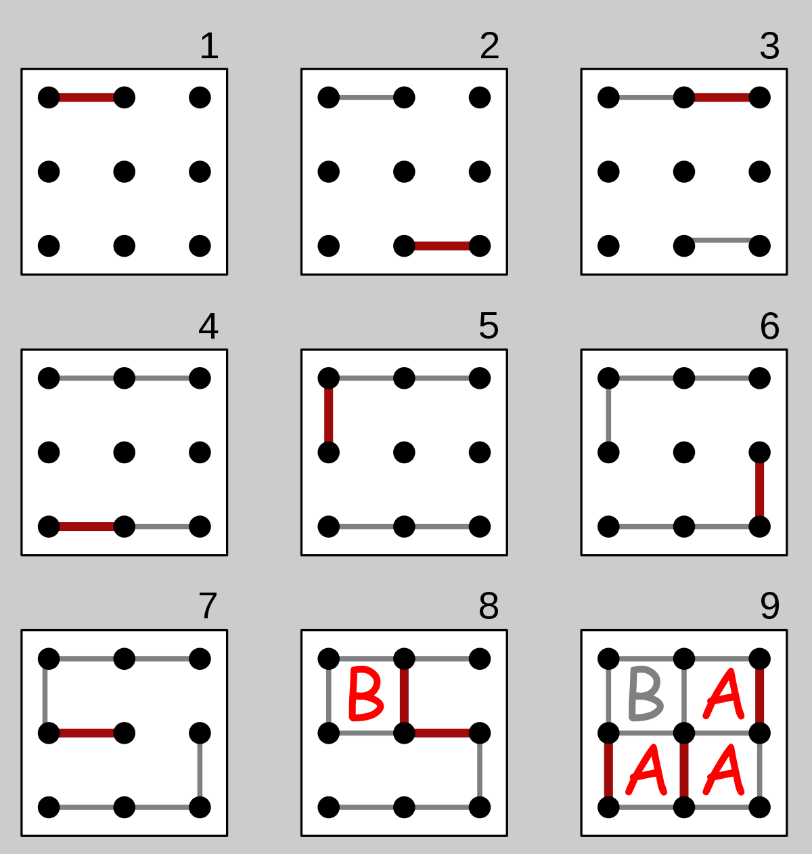
Example:



## 1.2.3 Transition model:

form edges and boxes.

Example:



## 1.2.4 Goal test:

We reach the goal state if all boxes are formed in the grid. The winner is the one who create more boxes, and if the number of boxes is equal then it is a tie.



Goal state

A game of 3 x 3 grid whereas the blue player is the winner.

## 1.2.5 Path cost function:

Path function is not applicable.

## 1.3 Technical discussion

The AI Techniques for solving " Dots and Boxes" Problem used in this report are minimax algorithm, and Alpha beta pruning algorithm:

**1.3.1 MiniMax technique:**

MiniMax algorithm is the most common artificial intelligence algorithm for a computer to participate in two-player combinatorial games. It works by looking ahead from the current game configuration a certain number of moves to see all the possible states the game could reach, building a game tree that is complete to a certain depth. Each state of the game will contain a score, given by an evaluation function. In Dots and Boxes game, this evaluation assigns a good score to a move that will give points to the player. On the other hand, it will give a bad score to a move which allows the opponent to ﬁnish the square in the future. We`re going to implement this method using java.

The Mininmax pseudocode in our problem:

**function** minimax(board, depth, player)

**If** board.depth >= depth or children.isEmpty() **then**

**return** the heuristic value of board

**If** player = "MAX" **then**

value = -∞

**for each** child of board **do**

minMaxVal = minimax(child, depth-1, "MIN")

**If** minMaxVal > value **then**

value = minMaxVal

**return** value;

**else if** player = "MIN" **then**

value = ∞

**for each** child of board **do**

minMaxVal = minimax(child, depth-1, "MAX")

**If** minMaxVal < value **then**

value = minMaxVal

**return** value;

The minimax function returns a heuristic value for leaf nodes (children is empty and nodes at the maximum search depth). Minimax treats the two players (the max player and the min player) separately in its code. If the player is max, then it recursively calls minimax and changing the player to min, decrements depth and returns the maximum value. If the player is min, then it recursively calls minimax and changing the player to max, decrements depth and returns the minimum value.

**1.3.2 Alpha-beta pruning technique:**

we are also going to solve the problem of dots and boxes by using, Alpha beta Pruning algorithm which is an improvement of minimax algorithm, it allows us to take the possible correct values of max and min without having to look at each node in the tree by *pruning* nodes that are not going to affect the final decision. Here is how we are going to implement the algorithm in solving dots and boxes; as in minimax algorithm we have *min* that try to get the lowest possible score and *max* that tries to get the highest possible score, in addition we have ***alpha*** *which is the best choice for max at the current state* and ***beta*** *which is best choice for min at the current state*. alpha initially equals to (-∞) and beta initially equals to (+∞), the values of alpha and beta are past to children and updated at their levels. prune if alpha >= beta. in the code we are going to choose the initial value for alpha as (-1000), and beta as (+1000). The possible moves are going to be represented as a tree that is searched by the algorithm (depth first search) to find the optimal path for max to win, also prune the nodes that are not needed.

**General algorithm for Alpha beta Pruning**

Function alphaBetaPruning(state) returns an action

v 🡨 max-value(state, -∞, +∞)

Return the action in action(state) with value v

Function max-value(state,alpha,beta) returns a utility value

If terminal-test(state) then return utility(state)

v 🡨 -∞

For each a in action(state) do

v 🡨 max(v, min-value,result(s,a), alpha, beta)

If v >= beta then return v

Alpha 🡨 max(alpha, v)

return v

Function min-value(state,alpha,beta) returns a utility value

If terminal-test(state) then return utility(state)

v 🡨 +∞

For each a in action(state) do

v 🡨 min(v, max-value,result(s,a), alpha, beta)

If v <= alpha then return v

beta 🡨 min(beta, v)

return v

**pseudocode for Alpha beta Pruning in Implementing the problem**

Algorithm miniMax(State, Ply\_num)

for i to State.Current.dimY do   
 for j to State.Current.dimX do   
 if State.Current.Mat[i][j] = ' ' and (j, i) not in State.children then  
 State.Make(j, i, True)  
 if Ply\_num < 2 then  
 return (i, j)  
 Minimum\_Score 🡨 1000  
 i 🡨 0  
 j 🡨 0  
 for k, z to State.children.items() do   
 Result 🡨 Algo.Maximum(z, Ply\_num - 1, Minimum\_Score)  
 if Minimum\_Score > Result then  
 Minimum\_Score 🡨 Result  
 i 🡸 k[0]  
 j 🡨 k[1]  
  
 return (i, j)  
  
 Algorithm Maximum(State, Ply\_num, Alpha):

if Ply\_num = 0 then  
 return State.CurrentScore  
  
 for i to State.Current.dimY do  
 for j to State.Current.dimX  
 if State.Current.Mat[i][j] = ' ' and (j, i) not in State.children then  
 State.Make(j, i, False)  
 Maximum\_Score 🡨 -1000  
 i 🡨 0  
 j 🡨 0  
 for k, z to State.children.items() do  
 Result 🡨 Algo.Minimum(z, Ply\_num - 1, Maximum\_Score)  
 if Maximum\_Score < Result then  
 Maximum\_Score 🡨 Result  
 if Result > Alpha then   
 return Result  
 return Maximum\_Score  
  
Algorithm Minimum(State, Ply\_num, Beta

if Ply\_num == 0 then  
 return State.CurrentScore  
  
 for i to State.Current.dimY do  
 for j to State.Current.dimX do  
 if State.Current.Mat[i][j] = ' ' and (j, i) not in State.children then  
 State.Make(j, i, True)  
 Minimum\_Score 🡨 1000  
 i 🡨 0  
 j 🡨 0  
 for k, z to State.children.items() then  
 Result 🡨 Algo.Maximum(z, Ply\_num - 1, Minimum\_Score)  
 if Minimum\_Score > Result then   
 Minimum\_Score 🡨 Result  
 if Result < Beta then  
 return Result  
 return Minimum\_Score

## 1.4 Discussion of results

**1.4.1 minimax**

## A picture containing crossword puzzle, text Description automatically generated

**-3**

**-3**

**-3**

**-3**

**-3**

**-3**

**-3**

**-3**

**-3**

**4**

**4**

**4**

**4**

**4**

**4**

**4**

**4**

**4**

**4**

B

F

F

E

E

D

D

C

C

B

C

B

D

C

B

A

A

A

A

**Level 3**

**Level 2**

**Level 1**

**Level 0**

**Level 7**

**Level 6**

**Level 5**

**Level 4**

Here is a simple tracing of Minimax algorithm Implementation with 3x3 size:

**Level 0:** The game is starting with empty grid of dots.

**Level 1:** Max player start the game by drawing a horizontal line.

**Level 2:** Min player draw a vertical line.

**Level 3:** The two players take turns adding a single horizontal or vertical line between two unjoined adjacent dots. We skipped other steps until the eighth step, in this level, the max player draw a horizontal line.

**Level 4:** In case A, Min player draw a vertical line and gain a box. In case B,C and D, Min player draw a horizontal line.

**Level 5:** In case A, Max player draw a horizontal line and gain a box. In case B, Max player draw a horizontal line. In case C, Max player draw a vertical line and gain a box

**Level 6: :** In case A, E and F, Max player draw another line because he gain a box in the previous step, he gain another box in this step. In case B, Max player draw another line because he gain a box in the previous step. In case C, Min player draw a vertical line and gain two boxes. In case D, Min player draw a horizontal line and gain a box.

**Level 7:** In case A, B, E and F, Max player draw another line because he gain a box in the previous step, he gain another box in this step. In case C, Min player draw another line because he gain two boxes in the previous step, he gain another box in this step. In case D, Min player draw another line because he gain a box in the previous step, he gain another two boxes in this step.

**1.4.2 alpha-beta pruning**

A picture containing crossword puzzle, text

Description automatically generated

**Level 0**

initially

Alpha = -∞

Beta = +∞

**Level 1**

**Level 2**

**Level 3**

**Level 7**

**Level 6**

**Level 5**

**Level 4**

Alpha = -∞

Beta = +∞

**X**

Alpha = -∞

Beta

= ~~+∞ 🡪~~ 4🡪-3

**C**

**B**

**A**

Alpha = -∞

🡪-3

Alpha = ~~-∞~~ 🡪4

Beta = +∞

**J**

**H**

**G**

**F**

**E**

**D**

Alpha

= -∞

Beta

= ~~+∞ 🡪~~

~~4~~ 🡪 -3

Alpha = ~~-∞~~ 🡪 4

Beta = +∞

**O**

**K**

**L**

**M**

**N**

4

4

4

-3

-3

4

Alpha = -∞

Beta

= ~~+∞ 🡪 4~~ 🡪 -3

Here is a simple trace of Alpha-beta pruning algorithm Implementation with size 3x3:

**Level 0:** initially the game start with an empty grid and dots.

**Level 1:** Max player start the game by drawing a horizontal line, and the initial value of Alpha will be set to -∞ and beta to +∞.

**Level 2:** Min player draw a vertical line, and the values of alpha and beta will be passed down.

**Level 3:** The two players take turns adding a single horizontal or vertical line connecting two adjacent dots. We skipped other steps until the eighth step. at this level, max player draws a horizontal line.

**Level 4:** we will consider the third child branch (for simplicity), and the values of alpha and beta will be passed down as alpha = -∞, and beta = +∞.

1. at child A the values of alpha and beta will be passed as alpha = -∞, and beta = +∞.

node A will explore its children D will get value alpha = 4 from K and E will prune L because it has the same value as the current alpha. the value 4 will be passed to alpha so the value of A 🡪 alpha = 4

1. the values of alpha and beta at node X in level 4 will be passed as alpha = 4, and beta = +∞.
2. at child B the values of alpha and beta will be passed as alpha = -∞, and beta = 4.

node B will explore its children F will get value beta = -3 from M and G will prune N because it has the same value as the current beta. the value -3 will be passed to alpha so the value of B 🡪 alpha = -3.

1. C branch will be pruned because -3 <= 4, node X is a minimizer it will take the

value of -3 either way.

## 1.5 Results

Due to the fact that minimax and alpha-beta has the same output we only included the results from alpha beta code.

!! Welcome to the game of Dots and Boxes !!

Be prepared to be crushed by the power of Artificial Intelligence ... !!

Kidding! You totally can beat it!

Press 1 to start the game or press 2 to escape from the inevitable doom!!

1

Please enter the number of rows for the board:

2

Please enter the number of columns for the board:

2

Please enter the number of plies used by the AI:

5

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| \* \* \*

1| 8 5

2| \* \* \*

3| 1 9

4| \* \* \*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 4

Please enter the 'Y' coordinate of your choice (an integer such as 4): 4

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| \* \* \*

1| 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 1 , 0 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| \* - \* \*

1| 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 2

Please enter the 'Y' coordinate of your choice (an integer such as 4): 0

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| \* - | \*

1| 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 3 , 0 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| \* - | - \*

1| 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 0

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - \*

1| 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 0 , 1 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - \*

1| | 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 4

Please enter the 'Y' coordinate of your choice (an integer such as 4): 0

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 4 , 1 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 2

Please enter the 'Y' coordinate of your choice (an integer such as 4): 2

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| \* | \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 3 , 2 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| \* | - \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 2

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 2 , 3 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 | 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 4

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 2 , 1 )

Current Score =====>> Your Score - AI Score = -5

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | | - \*

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 4

Please enter the 'Y' coordinate of your choice (an integer such as 4): 2

Current Score =====>> Your Score - AI Score = -5

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | | - |

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 1 , 2 )

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 3

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 4 , 3 )

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 1

Please enter the 'Y' coordinate of your choice (an integer such as 4): 4

Current Score =====>> Your Score - AI Score = -12

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 3 , 4 )

Current Score =====>> Your Score - AI Score = -21

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* - |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Stop this Madness!!!

!!! Inevitable Doom!!! You were crushed by the AI!!

Process finished with exit code 0

(an integer such as 4): 0

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| \* - | \*

1| 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 3 , 0 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| \* - | - \*

1| 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 0

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - \*

1| 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 0 , 1 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - \*

1| | 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 4

Please enter the 'Y' coordinate of your choice (an integer such as 4): 0

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 4 , 1 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 2

Please enter the 'Y' coordinate of your choice (an integer such as 4): 2

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| \* | \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 3 , 2 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| \* | - \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 2

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 2 , 3 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 | 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 4

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 2 , 1 )

Current Score =====>> Your Score - AI Score = -5

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | | - \*

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 4

Please enter the 'Y' coordinate of your choice (an integer such as 4): 2

Current Score =====>> Your Score - AI Score = -5

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | | - |

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 1 , 2 )

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 3

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 4 , 3 )

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 1

Please enter the 'Y' coordinate of your choice (an integer such as 4): 4

Current Score =====>> Your Score - AI Score = -12

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 3 , 4 )

Current Score =====>> Your Score - AI Score = -21

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* - |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Stop this Madness!!!

!!! Inevitable Doom!!! You were crushed by the AI!!

Process finished with exit code 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 2 , 3 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 | 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 4

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 2 , 1 )

Current Score =====>> Your Score - AI Score = -5

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | | - \*

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 4

Please enter the 'Y' coordinate of your choice (an integer such as 4): 2

Current Score =====>> Your Score - AI Score = -5

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | | - |

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 1 , 2 )

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 3

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 4 , 3 )

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 1

Please enter the 'Y' coordinate of your choice (an integer such as 4): 4

Current Score =====>> Your Score - AI Score = -12

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 3 , 4 )

Current Score =====>> Your Score - AI Score = -21

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* - |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Stop this Madness!!!

!!! Inevitable Doom!!! You were crushed by the AI!!

Process finished with exit code 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 4 , 1 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| \* \* \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 2

Please enter the 'Y' coordinate of your choice (an integer such as 4): 2

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| \* | \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 3 , 2 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| \* | - \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 2

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 2 , 3 )

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 | 9

4| \* \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 4

Current Score =====>> Your Score - AI Score = 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 5 |

2| | | - \*

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 2 , 1 )

Current Score =====>> Your Score - AI Score = -5

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | | - \*

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 4

Please enter the 'Y' coordinate of your choice (an integer such as 4): 2

Current Score =====>> Your Score - AI Score = -5

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | | - |

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 1 , 2 )

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 3

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 4 , 3 )

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 1

Please enter the 'Y' coordinate of your choice (an integer such as 4): 4

Current Score =====>> Your Score - AI Score = -12

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 3 , 4 )

Current Score =====>> Your Score - AI Score = -21

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* - |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Stop this Madness!!!

!!! Inevitable Doom!!! You were crushed by the AI!!

Process finished with exit code 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | | - |

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 1 , 2 )

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 0

Please enter the 'Y' coordinate of your choice (an integer such as 4): 3

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 4 , 3 )

Current Score =====>> Your Score - AI Score = -13

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please enter the 'X' coordinate of your choice (an integer such as 4): 1

Please enter the 'Y' coordinate of your choice (an integer such as 4): 4

Current Score =====>> Your Score - AI Score = -12

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 3 , 4 )

Current Score =====>> Your Score - AI Score = -21

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* - |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Stop this Madness!!!

!!! Inevitable Doom!!! You were crushed by the AI!!

Process finished with exit code 0

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI selected the following coordinates to play:

( 3 , 4 )

Current Score =====>> Your Score - AI Score = -21

0 1 2 3 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0| | - | - |

1| | 8 | 5 |

2| | - | - |

3| | 1 | 9 |

4| | - \* - |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Stop this Madness!!!

!!! Inevitable Doom!!! You were crushed by the AI!!

Process finished with exit code 0

# Part 2

## 2. Monte Carlo Tree Search technique to solve “Dots and Boxes” Puzzle

**2.1 MCTS Algorithm explanation.**

A tree structure is a hierarchy of linked nodes where each node represents a particular state. The structure has nodes, these nodes have none, one or more child nodes. There is a particular way for a solution, The exits path from the "root" node (initial state) to a "goal" node (desired state). Tree search algorithms attempt to find a solution by traversing the tree structure, it's starting at the root node and thoughtfully expanding the child nodes in a specific way.

Monte Carlo algorithm is a tree search algorithm that starts from any state and tries to improve it by producing its successors and choose the one that is more optimal than the current node and the other successors. It becomes useful as it continues to evaluate other alternatives periodically during the learning phase by executing them, instead of the current perceived optimal strategy. The process of Monte Carlo Tree Search can be broken down into four distinct steps, viz., selection, expansion, simulation, and backpropagation. Figure (1) shows each of these steps below:

A close up of a device

Description automatically generated

**2.1.1 Pseudo Code For MCT algorithm**

A screenshot of text

Description automatically generated

**2.2 Example**

We will apply the MCTS algorithm for Dots and boxes problem. The dots and boxes game is the problem of starts each player takes turns adding a single vertical or horizontal line between two enjoined adjacent dots. When a player completes the fourth side of a 1×1 box earns one point and takes another turn. The game ends when no more lines can be placed.

In our example, we will play between two humans or between human and computer.

**2.2.1 Color of players:**

Player #1: holder blue color.

Player #1: holder red color.



**2.2.2 Board of game:**

The real board representation:

A screenshot of a cell phone

Description automatically generated

The board size is 5X5. On the start of a Dot and Boxes, the blue side is always the first player. The UI will count points and time automatically.

* New Game button to start a new game with the computer.
* To stop the game, hit the Stop Game button on the left side.
* If you want to play with other players, you can Un-lead Engine, start a new game without game engine loaded.

**2.2.3 MCTS Algorithm**

A screenshot of a cell phone

Description automatically generated

Here where using MCTS algorithm to implements the game, first of all create root and the children of nodes. Choosing the successor depends on its heuristic function. It must be better than the current state and other successors.

The heuristic function is computed by the total number of boxes and lines. This piece of code calculates the heuristic function:

A screen shot of a computer

Description automatically generated

**After that judge between two players depends on heuristic function:**

A screenshot of a cell phone

Description automatically generated

**2.2.4 Final output:**

A screenshot of a cell phone

Description automatically generated

## 2.3 MCTS Algorithm Evaluation

|  |  |
| --- | --- |
|  | MCTS performance |
| Completeness | Yes |
| Optimality | No, but it finds the optimal solution in only current state. |
| Time complexity | O(mkI/C) where mm is the number of random children to consider per search and kk is the number of parallel searches, and II is the number of iterations and CC is the number of cores available. |
| Space complexity | O(mk) since in each iteration we map mk states over the cluster. |

# Appendix

# 2.4 References

1. Russell and Norving, “ Artificial Intelligence: A modern approach”, 3rd edition.
2. <https://github.com/GhadeerQalas/AI-Project>

2.5 Program Listing

MinMax:

|  |
| --- |
|  |
|  | import java.util.ArrayList; |
|  | import java.util.Collections; |
|  | import java.util.List; |
|  | /\*\* |
|  | \* |
|  | \* @author smoot |
|  | \*/ |
|  | public class Minimax { |
|  | public int minimax(Board board, int ply, String player){ |
|  |  |
|  |  |
|  | board.player = player; |
|  | List<Edge> children = board.getMoves(); |
|  |  |
|  | Collections.shuffle(children); |
|  |  |
|  |  |
|  | if(board.depth >= ply || children.isEmpty()) { |
|  | board.evaluate(); |
|  | return board.eval; |
|  | } |
|  |  |
|  | /\* |
|  | \* If the player is max, then it recursively calls minimax and changing the player to min |
|  | \* returns the maximum value. |
|  | \* Decrements ply |
|  | \*/ |
|  | if(player == "MAX") { |
|  |  |
|  | int value = Integer.MIN\_VALUE; |
|  | int minMaxVal; |
|  | for(Edge e: children) { |
|  |  |
|  | Board child = board.getNewBoard(e); |
|  | minMaxVal = minimax(child, ply-1, "MIN"); |
|  |  |
|  | //max value |
|  | if(minMaxVal > value) { |
|  | value = minMaxVal; |
|  | } |
|  | board.eval = value; |
|  | return value; |
|  | } |
|  | } |
|  | /\* |
|  | \* If the player is min, then it recursively calls minimax and changing the player to max |
|  | \* returns the minimum value. |
|  | \* Decrements ply |
|  | \*/ |
|  | else if(player == "MIN") { |
|  | int value = Integer.MAX\_VALUE; |
|  | int minMaxVal; |
|  | for(Edge e: children) { |
|  |  |
|  | Board child = board.getNewBoard(e); |
|  | minMaxVal = minimax(child, ply-1, "MAX"); |
|  |  |
|  | //min value |
|  | if(minMaxVal < value) { |
|  | value = minMaxVal; |
|  | } |
|  |  |
|  | board.eval = value; |
|  | return value; |
|  | } |
|  | } |
|  | return 0; |
|  | } |
|  |  |
|  | /\* |
|  | \* Makes a move, takes board, ply and player type as the argument |
|  | \* calls the minimax function and saves its value to a value, and if the value equals |
|  | \* the best value of the root, then it returns the best edge. |
|  | \*/ |
|  | public Edge makeMove(Board board, int ply, String player) { |
|  | Board child = board.clone(); |
|  | int value = minimax(child, ply, player); |
|  | Edge edge = new Edge(0,0); |
|  | ArrayList<Edge> children = child.getMoves(); |
|  | for(Edge e : children) { |
|  | Board root = board.clone(); |
|  | root.setEdge(e.getX(), e.getY()); |
|  | if(root.eval == value) { |
|  | edge.setX(root.bestX); |
|  | edge.setY(root.bestY); |
|  | } |
|  | } |
|  | return edge; |
|  | } |
|  | } |

Alpha Beta:

class Algo: # A class for defining algorithms used (minimax and alpha-beta pruning)

def miniMax(State, Ply\_num): # Function for the minimax algorithm

for i in range(State.Current.dimY):

for j in range(State.Current.dimX):

if State.Current.Mat[i][j] == ' ' and (j, i) not in State.children:

State.Make(j, i, True)

if Ply\_num < 2:

return (i, j)

Minimum\_Score = 1000

i = 0

j = 0

for k, z in State.children.items():

Result = Algo.Maximum(z, Ply\_num - 1, Minimum\_Score)

if Minimum\_Score > Result:

Minimum\_Score = Result

i = k[0]

j = k[1]

return (i, j)

def Maximum(State, Ply\_num, Alpha): # Alpha-beta pruning function for taking care of Alpha values

if Ply\_num == 0:

return State.CurrentScore

for i in range(State.Current.dimY):

for j in range(State.Current.dimX):

if State.Current.Mat[i][j] == ' ' and (j, i) not in State.children:

State.Make(j, i, False)

Maximum\_Score = -1000

i = 0

j = 0

for k, z in State.children.items():

Result = Algo.Minimum(z, Ply\_num - 1, Maximum\_Score)

if Maximum\_Score < Result:

Maximum\_Score = Result

if Result > Alpha:

return Result

return Maximum\_Score

def Minimum(State, Ply\_num, Beta): # Alpha-beta pruning function for taking care of Beta values

if Ply\_num == 0:

return State.CurrentScore

for i in range(State.Current.dimY):

for j in range(State.Current.dimX):

if State.Current.Mat[i][j] == ' ' and (j, i) not in State.children:

State.Make(j, i, True)

Minimum\_Score = 1000

i = 0

j = 0

for k, z in State.children.items():

Result = Algo.Maximum(z, Ply\_num - 1, Minimum\_Score)

if Minimum\_Score > Result:

Minimum\_Score = Result

if Result < Beta:

return Result

return Minimum\_Score