

Artificial Intelligence
Machine Problem 2 – Alpha/Beta Search

Introduction

For this assignment, you will implement the minimax algorithm with alpha-beta pruning in order to find the optimal move for a simple game. This game is played on a 5x5 board with the player starting in the lower left corner, and a monster in the upper left. One of the positions in the board contains gold. The player has to move (up, down, left, right) to reach the gold and come back to the starting position (to exit). The player can also build a wall in one of the four directions, if the square is empty. The monster can also move in the four directions and can also stay still (player must always build or move). The monster wants to eat the player, which happens when they become collocated.

Requirements

You are to modify the give base code program to implement the alpha-beta search for making the computer's move (it's random initially). This will require implementing additional methods for testing for terminal states and finding the utility of states, among others. You can follow the textbook's pseudocode for the algorithm.

How to play

When you start, the game board will appear with P marking the player location, M marking the monster location, and G marking the location of the gold. The # symbols indicate walls. The human player is first to move. The movements are one of 'w', 's', 'a', or 'd', indicating up, down, left, or right directions, respectively. You can addend a 'b' to indicate you want to build a wall in that direction (rather than move there). For example, you can enter 'wb' to build a wall above the current player's position (if possible).

Additional Requirements

1. The name of your source code file should be mp2 . py. All your code should be within a single file.
2. **You cannot import any packages that are not already imported in the base code.**
3. Your code should follow good coding practices, including good use of whitespace and use of both inline and block comments.
4. You need to use meaningful identifier names that conform to standard naming conventions.
5. **At the top of each file, you need to put in a block comment with the following information: your name, date, course name, semester, and assignment name.**

What to Turn In

You will turn in the single mp2.py file using BlackBoard.

HINTS

- It's easiest to use the backtracking method. That is, instead of generating hypothetical states, just apply the moves to the game board, compute the utility, and then backtrack the move. This requires that you add a function to reverse whatever move was made. Don't forget to reverse the state of having the gold.

EXTRA CREDIT (+3)

If you implement all the program requirements, you can get up to three extra credit points if you implement a cutoff search with a cutoff depth of 12. The monster should still play perfectly – win when it's possible. To get any extra credit, your program will still need to work perfectly according to the original specifications. You should also add a prompt at the beginning of the program to choose between original and extra credit versions.

Grading Rubric

Category	Unsatisfactory (0-1 points)	Satisfactory (2-3 point)	Distinguished (4-5 points)
Program Correctness	<ul style="list-style-type: none">• Program does not execute due to errors• Incorrect results for most or all input	<ul style="list-style-type: none">• Program works and completes most tasks appropriately• Program fails to work for special cases	<ul style="list-style-type: none">• Program runs and completes all required tasks• Handles any required special cases• Executes without errors
Programming Style	<ul style="list-style-type: none">• No name, date, or assignment title included• Poor use of white space• Disorganized and messy• No or few comments in the source code• Poor use of variables (improper scope/visibility, ambiguous naming).	<ul style="list-style-type: none">• Includes name, date, and assignment title.• White space makes program fairly easy to read.• Well organized code.• Some comments missing in the source code or too many comments• Good use of variables (few issues with scope/visibility or unambiguous naming).	<ul style="list-style-type: none">• Includes name, date, and assignment title.• Excellent use of white space.• Perfectly organized code.• Source code is commented throughout when needed• Excellent use of variables (no issues with scope/visibility or unambiguous naming).
Following Specifications	<ul style="list-style-type: none">• Incorrect filenames• Incorrect specified identifier names• Source code organization different from requirements• Additional requirements not satisfied	<ul style="list-style-type: none">• Correct filenames and class names• Few issues with other specified identifier names• Source code organization close to requirements• Some additional requirements not satisfied	<ul style="list-style-type: none">• Correct filenames and specified identifier names• Source code organization satisfies all requirements• All additional requirements satisfied

Sample Output

CLASS: Artificial Intelligence, Lewis University

NAME: [put your name here]

1 2 3 4 5

1|M| | | | |

2| |#|G|#|#|

3| |#| | | |

4| | | | | |

5|P|#|#| |#|

Player's Move # 1

Choose your move ['w', 'wb']: w

1 2 3 4 5

1|M| | | | |

2| |#|G|#|#|

3| |#| | | |

4|P| | | | |

5| |#|#| |#|

Depth reached: 21

Number pruned due to a/b: 595018

COMP MOVE TIME = 7.18 seconds

1 2 3 4 5

1| | | | | |

2|M|#|G|#|#|

3| |#| | | |

4|P| | | | |

5| |#|#| |#|

Player's Move # 2

Choose your move ['w', 'wb', 's', 'sb', 'd', 'db']: d

1 2 3 4 5

1| | | | | |

2|M|#|G|#|#|

3| |#| | | |

4| |P| | | |

5| |#|#| |#|

Depth reached: 19

Number pruned due to a/b: 231551

COMP MOVE TIME = 2.7 seconds

1 2 3 4 5

1|M| | | | |

2| |#|G|#|#|

3| |#| | | |

4| |P| | | |

5| |#|#| |#|

Player's Move # 3

Choose your move ['a', 'ab', 'd', 'db']: d

1 2 3 4 5

1|M| | | | |

2| |#|G|#|#|

3| |#| | | |

4| | |P| | |

5| |#|#| |#|

Depth reached: 17

Number pruned due to a/b: 156002

COMP MOVE TIME = 1.8 seconds

1 2 3 4 5

1| | | | | |

2|M|#|G|#|#|

3| |#| | | |

4| | |P| | |

5| |#|#| |#|

Player's Move # 4

Choose your move ['w', 'wb', 'a', 'ab', 'd', 'db']: w

1 2 3 4 5

1| | | | | |

2|M|#|G|#|#|

3| |#|P| | |

4| | | | | |

```
-----  
5| |#|#| |#|  
-----
```

Depth reached: 15
Number pruned due to a/b: 45973
COMP MOVE TIME = 0.57 seconds

```
  1 2 3 4 5  
-----  
1|M| | | | |  
-----  
2| |#|G|#|#|  
-----  
3| |#|P| | |  
-----  
4| | | | | |  
-----  
5| |#|#| |#|  
-----
```

Player's Move # 5
Choose your move ['w', 'wb', 's', 'sb', 'd', 'db']: w
 1 2 3 4 5

```
-----  
1|M| | | | |  
-----  
2| |#|P|#|#|  
-----  
3| |#| | | |  
-----  
4| | | | | |  
-----  
5| |#|#| |#|  
-----
```

Depth reached: 13
Number pruned due to a/b: 10681
COMP MOVE TIME = 0.16 seconds

```
  1 2 3 4 5  
-----  
1| | | | | |  
-----  
2|M|#|P|#|#|  
-----  
3| |#| | | |  
-----  
4| | | | | |  
-----  
5| |#|#| |#|  
-----
```

Player's Move # 6
Choose your move ['w', 'wb', 's', 'sb']: s
 1 2 3 4 5

```
-----  
1| | | | | |  
-----  
2|M|#| |#|#|  
-----  
3| |#|P| | |  
-----
```

4| | | | |

5| |#| |#|

Depth reached: 11

Number pruned due to a/b: 6831

COMP MOVE TIME = 0.11 seconds

1 2 3 4 5

1| | | | |

2| |#| |#|

3|M|#|P| |

4| | | | |

5| |#| |#|

Player's Move # 7

Choose your move ['w', 'wb', 's', 'sb', 'd', 'db']: s

1 2 3 4 5

1| | | | |

2| |#| |#|

3|M|#| | |

4| | |P| |

5| |#| |#|

Depth reached: 9

Number pruned due to a/b: 1674

COMP MOVE TIME = 0.04 seconds

1 2 3 4 5

1| | | | |

2| |#| |#|

3| |#| | |

4|M| |P| |

5| |#| |#|

Player's Move # 8

Choose your move ['w', 'wb', 'a', 'ab', 'd', 'db']: ab

1 2 3 4 5

1| | | | |

2| |#| |#|

3| |#| | |

```
-----  
4|M|#|P| | |
```

```
-----  
5| |#|##| |#|
```

```
-----  
Depth reached: 7
```

```
Number pruned due to a/b: 126
```

```
COMP MOVE TIME = 0.01 seconds
```

```
1 2 3 4 5
```

```
-----  
1| | | | | |
```

```
-----  
2| |#| |#|##|
```

```
-----  
3|M|#| | | |
```

```
-----  
4| |#|P| | |
```

```
-----  
5| |#|##| |#|
```

```
-----  
Player's Move # 9
```

```
Choose your move ['w', 'wb', 'd', 'db']: w
```

```
1 2 3 4 5
```

```
-----  
1| | | | | |
```

```
-----  
2| |#| |#|##|
```

```
-----  
3|M|#|P| | |
```

```
-----  
4| |#| | | |
```

```
-----  
5| |#|##| |#|
```

```
-----  
Depth reached: 5
```

```
Number pruned due to a/b: 40
```

```
COMP MOVE TIME = 0.0 seconds
```

```
1 2 3 4 5
```

```
-----  
1| | | | | |
```

```
-----  
2|M|#| |#|##|
```

```
-----  
3| |#|P| | |
```

```
-----  
4| |#| | | |
```

```
-----  
5| |#|##| |#|
```

```
-----  
Player's Move # 10
```

```
Choose your move ['w', 'wb', 's', 'sb', 'd', 'db']: w
```

```
1 2 3 4 5
```

```
-----  
1| | | | | |
```

```
-----  
2|M|#|P|##|##|
```

3| |#| | | |

4| |#| | | |

5| |#| |#| |#|

Depth reached: 3

Number pruned due to a/b: 5

COMP MOVE TIME = 0.0 seconds

1 2 3 4 5

1|M| | | | |

2| |#|P|#| |#|

3| |#| | | |

4| |#| | | |

5| |#| |#| |#|

Player's Move # 11

Choose your move ['w', 'wb', 's', 'sb']: w

1 2 3 4 5

1|M| |P| | |

2| |#| |#| |#|

3| |#| | | |

4| |#| | | |

5| |#| |#| |#|

Depth reached: 1

Number pruned due to a/b: 0

COMP MOVE TIME = 0.0 seconds

1 2 3 4 5

1| | |P| | |

2|M|#| |#| |#|

3| |#| | | |

4| |#| | | |

5| |#| |#| |#|

GAME OVER

You Lost!