

# MP 2: Planning Games

ABHI KAMBOJ

ECE 448

Section R: Prof. Hockenmaier

3/1/2019

## Table of Contents

MP 2: Planning Games .....	1
Table of Contents .....	2
Section I:.....	3
Section II:.....	4
Minimax vs Minimax (Offense vs Defense).....	4
Minimax vs Alpha-Beta .....	4
Alpha-Beta vs Minimax .....	5
Alpha-Beta vs Alpha-Beta .....	5
Conclusion/Analysis for Section II:.....	5
Section III:.....	7
Evaluation Function .....	7
Results.....	7
1 .....	7
2 .....	8
3 .....	8
4 .....	9
Section IV: .....	10
1 .....	10
2 .....	10
3 .....	11
EXTRA CREDIT/CONTRIBUTION:.....	12

## Section I:

**CSP:** We would like a description of your representation of the problem as well as an explanation of the algorithm (as well as heuristics) that you used to find a solution.

In a constraint satisfaction problem, there are variables, domains containing values for each variable and constraints which define how each variable should be assigned values. In my representation of the problem the variables were each of the pentominoes, the domains were the possible locations on the board they could fit, and the constraints were represented as a dictionary (I called b for board) that held locations on the board as values and a list of pentomino objects as keys. I created an initialization function which gave each location in the dictionary copies of all the possible pentominoes values (orientation and index) that could fit in that location. The goal of the program is to narrow down in it list of which pentominoes could fit in the location until all locations have been filled with one pentomino.

My algorithm is intended to be a recursive backtracking algorithm, like a depth first search with heuristics to make it run faster. I first chose an row,col location where least pentominoes can go (least remaining values heuristic). Then I go through the list of possible pentominoes that can go there and out of those that have not been place already, I look for the one with the least amount of remaining locations where it can go (least constraining assignment heuristic). After choosing the appropriate pent for the location, I removed the pent from the list and all other locations' lists and add the pent and location to the solution. I then recurse and call the function again until there are no available locations left.

To improve this algorithm, I would add forward checking. In order to implement forward checking the pent that is being placed would also check for "holes" or spots on the board where no other pent could possibly fit because of its size, and then I would not place that pent if it caused holes.

## Section II:

Ultimate Tic-Tac-Toe: for each of the four games of predefined agents, report the final game board positions, number of expanded nodes, and the final winner.

### Minimax vs Minimax (Offense vs Defense)

```
0 _ X 0 _ _ 0 X _
_ X _ _ _ _ _ _
X _ _ _ _ _ _ _
_ _ _ X _ 0 _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
Total number of expanded nodes: 5246
The winner is maxPlayer!!!
```

Nodes Expanded: 5246

Winner is Offensive Minimax

### Minimax vs Alpha-Beta

```
0 _ X 0 _ _ 0 X _
_ X _ _ _ _ _ _
X _ _ _ _ _ _ _
_ _ _ X _ 0 _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
Total number of expanded nodes: 3565
The winner is maxPlayer!!!
```

Nodes expanded: 3565

Winner is Offensive Minimax

### Alpha-Beta vs Minimax

```
0 _ X 0 _ _ 0 X _
_ X _ _ _ _ _ _
X _ _ _ _ _ _ _
_ _ _ X _ 0 _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _

Total number of expanded nodes: 3263
The winner is maxPlayer!!!
```

Nodes expanded: 3263

Winner is Offensive Alpha Beta

### Alpha-Beta vs Alpha-Beta

```
0 _ X 0 _ _ 0 X _
_ X _ _ _ _ _ _
X _ _ _ _ _ _ _
_ _ _ X _ 0 _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _

Total number of expanded nodes: 1582
The winner is maxPlayer!!!
```

Nodes expanded: 1582

Winner is Offensive Alpha Beta

### Conclusion/Analysis for Section II:

The results here make sense because minimax and alpha-beta pruning are deterministically the same algorithm so all the final board positions should be the same. The only difference is the run time. When

we used one alpha-beta algorithm and one minimax compared to both minimax the number of expanded nodes dropped from about 5000 to about 3000! Then when we used both alpha-beta algorithms the number of expanded nodes dropped all the way to about 1500! This is significantly faster and follows the theory since the alpha-beta pruning reduces the runtime from  $O(b^m)$  to  $O(b^{m/2})$ , where  $b$  is the branching factor and  $m$  is the number of edges in the longest path!

## Section III:

Ultimate Tic-Tac-Toe: for at least 20 games of offensive agent vs your agent, explain your formulation and advantages of evaluation function. Report the percentage of winning time and number of expanded nodes for each game. Report 3-5 representative final game boards that show the advantage of your evaluation function vs predefined offensive evaluation function. If your own defined agent fails to beat the predefined agent, explain why that happened.

### Evaluation Function

After observing the predefined agent's play, I noticed there were many times that one agent simply chose a box where the other opponent could make 2 in a row which would increase the chances of that opponent winning. The predefined evaluation function rewards an agent for blocking the opponent's two in a row and for having one's own two in a row, so I added negative reward for positions allowing the opponent to have two in a row. In addition, I added rewards for occupying the very middle space of the boards, because that is an advantageous position in my personal experience and opinion since it allows the most ways to create a three in a row. Other than these few modifications, my designed evaluation function follows the same rules as the predefined evaluation.

### Results

My agent won the game 95-100% of the time (I ran 20 trials a few times). The number of nodes expanded in each of the 20 rounds for my latest run are as follows respectively:

[2617, 3053, 2379, 3487, 3487, 2008, 1988, 1988, 2008, 2617, 3053, 1931, 4503, 1990, 2154, 4503, 2008, 2379, 2049, 2416]

### 1

My Agent (O) played first starting at board 2. This game shows that my agent didn't really give a chance for the opponent X to get even 2 in a row.

```
Round 17
Randomness: 2 False
The winner is MyAgent!!!
X _ O X _ _ O O X
O O O _ _ _ _ _
_ _ _ _ _ _ _ _
X _ _ X _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
```

2

Round 11: My agent (O) played second, and the game started at board 4. This game shows how my agent was favoring the middle, whereas the other agent kept playing in the corners.

```
Round 11
Randomness: 4 True
The winner is MyAgent!!!
X O _ O X _ _ _ _
_ O _ O _ _ _ _
_ O _ _ _ _ _ _
X _ _ X X _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
```

3

Round 15: My agent (O) went first starting at board 4 and won. This game was interesting since it was longer than the other 19 games and it seems like both players had couple of good blocks. However, I think it still supports my evaluation function because the final win from my agent was through the center of middle board, which my agent gave a higher evaluation to than the other.

```
Round 15
Randomness: 4 False
The winner is MyAgent!!!
O _ X X X O _ _ O
_ X _ _ X _ O _ O
O _ _ _ _ X _ X
_ _ X X O _ _ O _
_ _ _ _ O _ _ _ _
_ _ _ _ O _ X _ _
O _ X _ _ _ _ O _
_ _ X _ _ _ _ _ _
O _ _ _ _ _ _ _ _
```



4

Round 4: My agent (O) went second starting at board 6 and loss. This was the one loss out of about 40 runs, and it seems like a close game. My player had two setups (rows of 2) at boards 5 and 7 however it seems like X's beat my agent to it.

```
Round 4
Randomness: 6 True
The winner is maxPlayer!!!
0 _ X X _ 0 _ _ 0
_ X _ _ _ _ 0 _ 0
0 _ _ X _ _ X X X

_ _ X _ 0 X _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ X
_ _ _ _ _ _ _ _
X X 0 _ _ _ 0 0 _
_ 0 _ _ _ _ _ _
_ _ _ _ _ _ _ _
```

## Section IV:

Ultimate Tic-Tac-Toe: for at least 10 games of human vs your agent, discuss your observations, including the percentage of winning time, the advantages or disadvantages of your defined evaluation functions. Report 3-5 representative final game boards that show the advantages or disadvantages of your evaluation function.

1

Start index: 5

I was X and played first.

The agent was doing a really good job of blocking me for this game. I had MANY different spots on the board where one more move could lead to a 3 in a row, however the agent was able to force me into blocking it which allowed it to win on the other board. The advantage that was shown in this game was the agent's foresight. I had a better position at the current moment on the board at almost every step, however the agent was able to see 3 moves in advance and predict the best outcome which I couldn't do. Moreover, the agent's evaluation function although focusing on blocks also valued playing the center move which is what it ended up winning on!

```
What would you like to move (0-9 on small board)?8
_ X _ O _ _ _ O X
_ X _ _ _ _ X _ _
_ _ _ X _ X _ X X

_ _ O _ _ O _ _ _
_ _ _ _ _ O _ X _
_ _ _ _ _ X X _ _

O O _ _ _ O _ _ O
_ _ _ _ _ O _ _
_ _ _ _ _ O _ _

The winner is AIAgent!!!
```

2

Start index: 3

I was X and played first

This game was relatively quick. I set the game up well and ended up trapping the O player across two different boards. I think this showed me the disadvantage in my evaluation function is that it looks at 3-in-a-row combinations across all the boards at once. Therefore, it can't really deduce strategic/advantageous positions across all the boards (aside from its foresight of being able to see 3 moves into the game). This could be a daunting set back for the agent since the board is constantly switching positions, being able to take that into account when playing could help improve the agent's evaluation of the board.

```

0 _ X _ _ _ X _ X
_ _ _ _ _ X _
_ _ X _ _ _ X _ 0

_ _ _ _ _ 0 _ _ _
_ _ _ _ _ 0 _ _ _
_ _ _ _ _ 0 _ _ _

0 _ 0 _ _ _ _ 0
_ _ _ _ _ X _
_ _ _ _ _ X _ _

The winner is HumanPlayer!!!

```

3

Start Index: 6

I was X and played first

This game was fairly long and the agent put up a good match, however I think disadvantage was that the AI agent was spread to thin. For some of the boards, I set it up so I had multiple ways to make three-in-a-row with X's making it difficult for the agent because they would have to go to that board and block multiple times. The agent played a variety of moves causing it to spread it's O's in many different locations, and maybe if that was taken into account when creating the evaluation function, the AI agent would have performed better on this board

```

What would you like to move (0-9 on small board)?7
0 _ 0 X X 0 0 X X
_ X _ X 0 X _ _ _
X X X _ X _ X _ _

_ 0 _ X 0 0 _ 0 _
_ _ _ _ X _ _ _ _
_ _ _ _ _ _ _ _ _

_ 0 _ 0 _ _ 0 0 _
_ 0 _ _ _ _ _ _ _
_ _ X _ _ _ _ _ _

The winner is HumanPlayer!!!

```

~~EXTRA CREDIT/CONTRIBUTION:~~

I completed the assignment myself.