MP02 Presentation

HEURISTIC e1

```
1. for every row, column and diagonal R do:
2.   for K in range(s, 0, -1)
3.         if R contains K O's then
4.               V += num*100
5.               break
6.         else if R contains K X's then
7.               V -= num*100
8.               break
9.         end if
10.  end for
11.  return V
```

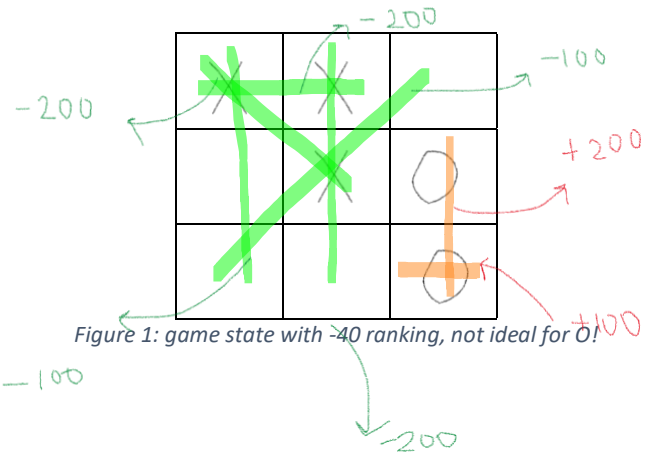

Figure 1: game state with -40 ranking, not ideal for O!

HEURISTIC e2

```
1. for every row, column and diagonal R do:
2.         if R contains s potential consecutive pieces then
3.               if piece is O then
4.                     V += 100
5.               else if piece is X then
6.                     V -= 100
7. return V
```

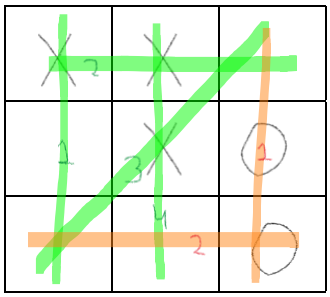

Figure 2: game state with s=3 gives 3 potential wins for X and 2 potential wins for O

ANALYSIS

1. **Effect of depth**
   - **Deeper** level explored → heuristic returns more informative, **better move**
   - **Shallower** level explored → heuristic returns least informative move, **worst move**

```
1.  the size of the board: 4
2.  the number of blocs: 1
3.  enter x coordinate for bloc 1: 2
4.  enter y coordinate for bloc 1: 3
5.  the winning line-up size:3
6.  the maximum depth of the adversarial search for player 1: 3 # player 1 is X
7.  the maximum depth of the adversarial search for player 2: 1
8.  the maximum allowed time (in seconds) for your program to return a move: 1
9.  minimax (FALSE) or alphabeta (TRUE)?FALSE
10. which play mode (H-H, AI-AI, AI-H or H-AI)? AI-AI
11. n=4 b=1 s=3 t=1.0
12. blocs=(2,3)
13. Player 1: AI d=3 a=FALSE e2
14. Player 2: AI d=1 a=FALSE e1
15.
16.  … other moves…
17.
18. ********** MOVE #1 ***************
19. ....
20. .X..
21. ....
22. ..-.
23.
24.  … other moves…
25.
26. ********** MOVE #9 ***************
27. X.O.
28. OXX.
29. XOX.
30. O.-.
31.
32. The winner is X!
33. ----------------------------------------
34. the size of the board: 6
35. the number of blocs: 2
36. enter x coordinate for bloc 1: 1
37. enter y coordinate for bloc 1: 1
38. enter x coordinate for bloc 2: 1
39. enter y coordinate for bloc 2: 0
40. the winning line-up size:4
41. the maximum depth of the adversarial search for player 1: 2 # player 1 is X
42. the maximum depth of the adversarial search for player 2: 4 # player 2 is O
43. the maximum allowed time (in seconds) for your program to return a move: 2
44. minimax (FALSE) or alphabeta (TRUE)?FALSE
45. which play mode (H-H, AI-AI, AI-H or H-AI)? AI-AI
46. n=4 b=1 s=3 t=1.0
47. blocs=(2,3)
48. Player 1: AI d=3 a=FALSE e2
49. Player 2: AI d=1 a=FALSE e1
50.
51.  … other moves…
52.
53. ********** MOVE #1 ***************
54.
```

```
55. .-....
56. .-....
57. .X....
58. ......
59. ......
60. ......
61.
62.  … other moves…
63.
64. ********** MOVE #8 ***************
65. O-....
66. O-....
67. OXX.X.
68. O.X...
69. ......
70. ......
71.
72. The winner is O!
```

2. **Minimax vs alpha-beta**
   In general, **alpha-beta explores d levels faster than minimax.**

   For a game with the same parameters:

```
1.  the size of the board: 4
2.  the number of blocs: 2
3.  enter x coordinate for bloc 1: 1
4.  enter y coordinate for bloc 1: 1
5.  enter x coordinate for bloc 2: 3
6.  enter y coordinate for bloc 2: 0
7.  the winning line-up size:3
8.  the maximum depth of the adversarial search for player 1: 3
9.  the maximum depth of the adversarial search for player 2: 3
10. the maximum allowed time (in seconds) for your program to return a move: 1
```

   - With **alpha-beta**, evaluation time $t \downarrow$ for some **depth** d

```
1.  minimax (FALSE) or alphabeta (TRUE)? TRUE
2.
3.  … some other moves and lines…
4.
5.  i Average evaluation time: 0.012s
6.  ii Total heuristic evaluations: 1125
```

   - With **minimax** , evaluation time $t \uparrow$ for some **depth** d

```
1.  minimax (FALSE) or alphabeta (TRUE)? FALSE
2.
3.  … some other moves and lines …
4.
5.  i Average evaluation time: 0.426s
6.  ii Total heuristic evaluations: 25578
```