

Artificial Intelligence - Methods and  
Applications - 5DV181

Othello

Shinorina Shahrin Shaon (mai21ssn)

1. I work in windows and use “bash” command window to run and see the output of “Othello” program. I work on python. To run my code I use both “test\_code” and “Python” scripts file. I use time limit 5. Under “test\_code” directory I run the below command:

```
./othellostart ./othello_naive ../Python/othello.sh 5
```

At python I work for “Black” player. Here othellostart and othello\_naive are from test\_code and othello.sh is python script file.

2. At Othello.py file I use a flag “play”. It is staying “True” until player is finished. If the player is not done othello store the moves in a list and print the moves using “print\_move()”. If the list is empty it prints “pass” and the flag is turn to “False”. I follow this procedure and no moves are missing for players.

I check both computer player and human player. At alpha beta algorithm the search depth is fixed 7.

I check the values of alpha beta with child. So here no node is missing.

3. I use the provided “CountingEvaluator”. Here “white\_square” and black\_square” both were initially 0 and it is increamenting by 1.
4. At my work I don’t mention any time. I mention time on run command. When othello takes too much to complete run I use “**Ctrl+C**” button to terminate.
5. For make\_move() I use row, col, clone(),\_\_is\_own\_square(). I check the all position north, south, east, north\_west, north\_east,south\_east and south\_west. I update players own row column according to the position.

6. For white player the output is:

```

MINGW64: c:/Users/User/PycharmProjects/5dv181ht21/Othello/test_code
white to move
W0XE0000EEEXE0EEEEE0X0XXXEEEEE0E0000EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
(2,5)
  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
--|---|
1 | 0 | X |   | 0 | 0 | 0 |   | 1
--|---|
2 |   | X |   | 0 | 0 |   |   | 2
--|---|
3 | 0 | X | 0 | X | 0 | X |   | 3
--|---|
4 |   | 0 |   | 0 | 0 |   |   | 4
--|---|
5 | 0 |   | 0 | 0 | 0 |   |   | 5
--|---|
6 |   |   |   |   |   |   |   | 6
--|---|
7 |   |   |   |   |   |   |   | 7
--|---|
8 |   |   |   |   |   |   |   | 8
--|---|
  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

It took 1 seconds to make this move...

```

For black player the output is:

```
Black to move
8000000000000000000000000000X0000000X00000X0X0000000000000000XE000
pass
  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
---|---|---|---|---|---|---|---|---
1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
---|---|---|---|---|---|---|---|---
2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
---|---|---|---|---|---|---|---|---
3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
---|---|---|---|---|---|---|---|---
4 | 0 | 0 | 0 | X | 0 | 0 | 0 | 0 | 4 |
---|---|---|---|---|---|---|---|---
5 | 0 | 0 | 0 | X | 0 | 0 | 0 | 0 | 5 |
---|---|---|---|---|---|---|---|---
6 | 0 | X | 0 | X | 0 | 0 | 0 | 0 | 6 |
---|---|---|---|---|---|---|---|---
7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 |
---|---|---|---|---|---|---|---|---
8 | 0 | 0 | 0 | X |   | 0 | 0 | 0 | 8 |
---|---|---|---|---|---|---|---|---
  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

It took 0 seconds to make this move...
```

For 10s the final output is:

```
MINGW64:/c/Users/User/PycharmProjects/5dv181ht21/Othello/test_code

  |  |  |  |  |  |  |  |  |  |  |
4 | 0 | 0 | 0 | X | 0 | 0 | 0 | 0 | 4
  |  |  |  |  |  |  |  |  |  |  |
5 | 0 | 0 | 0 | X | 0 | 0 | 0 | 0 | 5
  |  |  |  |  |  |  |  |  |  |  |
6 | 0 | X | 0 | X | 0 | 0 | 0 | 0 | 6
  |  |  |  |  |  |  |  |  |  |  |
7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7
  |  |  |  |  |  |  |  |  |  |  |
8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8
  |  |  |  |  |  |  |  |  |  |  |
  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

It took 0 seconds to make this move...
*****
Results for ./othello_naive vs. ../Python/othello.sh:
white won with 56 points
Average time for white: 0.4 s (max: 1 s)
Average time for black: 0.5 s (max: 1 s)
```

There is no noticeable difference for different time.

7. I took huge time for Othello.py file. It was difficult for me to understand what will go on with Othello.py file. The make\_move() was easy as I follow the \_\_is\_move() procedure. My another difficult point is to run the file. At running time I get too much error. After all it was really effective work.
8. My code has "TypeError". I can't shortout this typoerror. This is the big limitation of my work.
9. References: <http://dhconnelly.com/paip-python/docs/paip/othello.html>  
<http://www.cse.uaa.alaska.edu/~afkim/csce211/handouts/othello.pdf> and several github repos.