Report

Title: CS365 Lab B – Adversarial Games
Authors: Gordian Bruns
Programming Language: Python 3.7.4

(Note that descriptions of all files such as what is contained in each file is in the README)

I implemented a finish and defense mechanism that works for every utility function, which works as follows: If the current player can finish a game with one move, then the player will automatically take this move to win the game. If the current player can win in two moves, then he will prefer this course of action over others. If the opposite player has a pawn that can finish with its next move, but there is a pawn you can use to capture the opposite pawn, then it is going to do so. Also, Os represent white pawns, while Xs represent black pawns. Note that I change the turn of the players in each layer of the tree. This leads to some more complicated logic when calculating the utilities. So, if it is white's turn and the node is depth 1 or 3, then it actually is black's turn. If it is white's turn and the node is depth 0 or 2, then it actually is white's turn. The same works vice versa. Also, note that the games played on a 8 x 8, 2 board take a long time to finish (> 5 mins).

Evasive Function:

"Evasive" seems to be a fitting description for this utility function since the players just try to not lose pawns and they do not really care whether they win or not.

Examples:

6 x 6, 1:

```
. . . X . .
. O . . . O
. . O O . O
X . . X . .
X . . X X .
. . . . . .
```
Number of captured pawns: 0 (by white), 1 (by black)
Number of moves taken: 24 in total (12 each)

8 x 8, 2

```
. . . . . . . .
. . . O . . O O
O . O O O . . .
O . . O O O X X
X O . X X . . .
X . . X . X X .
. X X X X . . .
O . . . . . . .
```
Number of captured pawns: 3 (by white), 3 (by black)

Number of moves taken: 69 in total (35 by white, 34 by black)

4 x 4, 1

```
. X . .
O . . .
X X X .
. . . .
```

Number of captured pawns: 0 (by white), 3 (by black)
Number of moves taken: 12 in total (6 by white, 6 by black)

Conqueror Function:

With this approach, we notice that a lot more pawns are getting captured. Therefore, the names "conqueror" is fitting as well.

Examples:

6 x 6, 1

```
. . X . . .
. . . . O .
. . . . . .
O . . . . .
. . . . X X
. . X . . .
```

Number of captured pawns: 2 (by white), 4 (by black)
Number of moves taken: 28 in total (14 by white, 14 by black)

8 x 8, 2

```
. . O . O X . .
O . . . O . . O
O . . . . . . .
. . . . . . . .
. . . . . . . .
X . . . . . . .
. . . . . . X X
. . . X . . . .
```

Number of captured pawns: 11 (by white), 10 (by black)
Number of moves taken: 74 in total (37 by white, 37 by black)

4 x 4, 1

```
. . . .
O . . .
. . X .
O . . .
```

Number of captured pawns: 3 (by white), 2 (by black)
Number of moves taken: 17 in total (9 by white, 8 by black)

Additional Functions:

- **balanced**
- **rusher**

(These functions do not seem to be optimal yet since their win rate is not as high as hoped. This could be because of the implemented finish and defense mechanisms, small logical mistakes, or other reasons. I will find appropriate names for my functions and make them better for the final version of this project)

Balanced:

This function considers both the number of remaining white pawns and of remaining black pawns to find a good choice of action.

Rusher:

This function uses an additional distant function that determines the distance of the closest pawn to the finish and considers that in the choice of the best action.

Examples:

balanced (white) vs evasive (black) on 8 x 8, 2

```
. . . . . . . O
. . O . . . . .
O O . O O O . .
. X . . X . . O
. . . X O . . .
. . . . . . . .
. . . . . . . .
. . . . . O . .
Number of captured pawns: 13 (by white), 6 (by black)
Number of moves taken: 83 in total (42 by white, 41 by black)
```

Rusher (white) vs evasive (black) on 8 x 8, 2

```
O O O . . . . .
O O O O O O O .
. . . . . . . .
X X . X . O . O
. . X X . . . .
X X . . . X X X
```

```
. . . . . . X X
X O . . . . X .
```
Number of captured pawns: 2 (by white), 3 (by black)
Number of moves taken: 55 in total (28 by white, 27 by black)

Balanced (white) vs evasive (black) on 6 x 6, 1

```
O O . . . .
. . . . O .
X . . O . .
. . . X . X
. . . X . .
X O . . . .
```
Number of captured pawns: 1 (by white), 1 (by black)
Number of moves taken: 21 in total (11 by white, 10 by black)

Balanced (white) vs conqueror (black) on 6 x 6, 1

```
O . . O . .
. . . O . .
O . . . . .
X . . . . .
. . . X . .
. . . O . .
```
Number of captured pawns: 4 (by white), 1 (by black)
Number of moves taken: 23 in total (12 by white, 11 by black)

Balanced (white) vs conqueror (black) on 4 x 4, 1

```
. . . O
. . . .
. . X .
O . . .
```
Number of captured pawns: 3 (by white), 2 (by black)
Number of moves taken: 11 in total (6 by white, 5 by black)

Rusher (white) vs conqueror (black) on 4 x 4, 1

```
. . . O
. . . .
. . X .
O . . .
```
Number of captured pawns: 3 (by white), 2 (by black)
Number of moves taken: 11 in total (6 by white, 5 by black)

Rusher (white) vs conqueror (black) on 6 x 6, 1

```
O O . O . .
. . . . . .
. . . . . .
. . . . X .
. . . . . .
X . O . . .
Number of captured pawns: 4 (by white), 2 (by black)
Number of moves taken: 21 in total (11 by white, 10 by black)
```

Rusher (white) vs balanced (black) on 8 x 8, 1

```
. . . X . . . .
. . . . . . . .
. O . . . . . .
. . . X . . . .
O X . . . X . X
O . . . . . . .
. . . . . . . .
X . . . . . . .
Number of captured pawns: 2 (by white), 5 (by black)
Number of moves taken: 56 in total (28 by white, 28 by black)
```

Rusher (white) vs balanced (black) on 4 x 4, 1

```
O O . .
X . . .
. X . .
. X O .
Number of captured pawns: 1 (by white), 1 (by black)
Number of moves taken: 9 in total (5 by white, 4 by black)
```

Rusher (white) vs balanced (black) on 6 x 6, 2

```
X . . . . .
. . . . . O
. O . . . .
. . O . . X
. . . . X X
. . X . . .
Number of captured pawns: 7 (by white), 9 (by black)
Number of moves taken: 42 in total (21 by white, 21 by black)
```

Rusher (white) vs balanced (black) on 10 x 4, 1

```
. . . .
. . . .
```

```
X . . .
. . . .
. . . .
. X X .
. . . .
X . . .
. . . .
. . . .
Number of captured pawns: 0 (by white), 4 (by black)
Number of moves taken: 34 in total (17 by white, 17 by black)
```

Conclusion:

It seems as if the function balanced has the best performance. Both balanced and rusher reliably beat evasive and conqueror reliably, but in direct comparison between these two functions, balanced seem to win more often. Since I also implemented a finish and defense mechanism for all utility functions, the balanced utility function seems to work well already with barely any observable flaws (questionable decisions). This stays the same for larger and smaller board sizes, and if we increase or decrease the number of pawns. The rusher utility function is overall also a good function, but it has especially problems when the board does not have many columns and when there are a lot of pawns. Altogether, I am sure that both functions can be improved, but they already do a good job at beating evasive and conqueror, which means that they are at least decent players.