Heuristic Analysis

This work presents 3 heuristics for achieving better performance than the `ID_Improved` agent. Each one increases the complexity in terms of computation and implementation. Let's analyze each one:

Center moves

Intuition

The more moves at the center this player can make, the better the outcome of the game will be.

Implementation

This function simply subtracts the available opponent's center moves from the available player's center moves. Center moves are defined as the inner `3x3` rectangle of the board.

Center moves with blanks

Intuition

Same as the above but we scale according to the available blank squares.

Implementation

Call `center_moves()` and divide the result by the number of blank squares.

Uber heuristic

Intuition

The ultimate heuristic (as the name suggests :))! Rewards the player more when he has more available moves and center moves as opposed to his opponent. Finally, the result is scaled by the player's remaining moves.

Implementation

The heuristic implements the following formula:

$$Score = \frac{(P_{moves} + P_{center\,moves}) - (O_{moves} + O_{center\,moves}) - B}{B + P_{moves} - O_{moves}}$$

where P is the player, O is the opponent and B is the number of blank squares.

Evaluation

The `tournament.py` script was run `3` times for each individual heuristic. The results are presented in the following table:

Heuristic	ID wins	ID win %	Student wins	Student win %
center moves	347/420	0.826	330/420	0.786
center moves with blanks	346/420	0.823	333/420	0.792
uber	348/420	0.828	363/420	0.864

Conclusion

According to the data presented above, the `uber` heuristic performs better than the `ID_improvement` agent and all other heuristics. I recommend choosing this heuristic when building an agent since it is:

- the only heuristic that outperforms the sample agent and all other heuristics, according to the data
- a composition of the other two heuristics (with additions), thus including their predictive power
- of equal complexity (Big O) to the other heuristics, yet still much more performant

Further tweaking (possibly including weights to some/each parameter) of the formula might provide even more improvements.