paperwork

IDs:

026507202

203371935

Heuristics:

For our heuristic functions, we decided to balance between a straight win/loss and weighted sum several mini-heuristics:

1. The difference between weighted pieces. Each “regular” piece worth 1, but edge piece worth more, and corner piece worth a lot more. The more pieces you have compared to the opponent, the better.
2. The difference between possible moves. The more moves a player can make from the given state, the more “control” he/it has on the game, and therefore the better the state he/it is currently in.
3. The higher percentage of currently placed pieces. Similar to (1), but this value is normalized by the total amount of pieces placed thus far, having more influence on early states of the game
4. The higher percentage of currently possible moves. Similar to (2), but this value is normalized by the total amount of moves both player can currently make, having more influence on mid states of the game
5. The difference between secured pieces. The more unchangeable pieces a player has, the better

Having these mini-heuristics set, we ran iterations of min-max-alpha-beta using heuristics with different weights, to determine which weights set is superior.

Min-max vs min-max-alpha-beta:

We ran several iterations of min-max vs min-max-alpha-beta, and immediately noticed that min-max-alpha-beta is roughly twice as fast, at all tested depths.

Min-max and min-max-alpha-beta variations:

In both algorithms, we slightly changed the edge case of zero next generations to keep going as long as the opponent has at least one next generation, as the player might have been “skipped”.

Graphics:

For graphics we used the standard built-in tkinter of python. We only used basic primitives to allow a simple but functioning GUI.