Review of heuristics used in Project 2
By Brennon York
On 2017/02/19

For this project I created three separate heuristics based on ideas I've had throughout my time working on and studying for this project.

The first attempt, custom_score, is a metric based on the idea that, with the limited mobility of the Knight-style chess player, the closer one is to the edge of the board the fewer options will be available to the individual both currently and in the following moves. Measuring this did not show any notable improvement from the baseline ID_Improved heuristic.

The second heuristic, custom_score_2 and custom_score_2', are computed based on euclidean distance. This came as a thought from the reference within the learning as well as the idea around blocking space from your opponent. In this light I thought we could create a score based on the inverse euclidean distance (custom_score_2) to optimize for the player to be as close to the enemy as possible. This, as you'll see, did not work out although I also though, what if we computed the inverse as well (custom_score_2') which optimized for the player being as far away from the opponent as possible. This was the first heuristic that demonstrated a marginal, yet constant, positive increase from the baseline (as in this beat the ID_Improved baseline on at least three of the five attempts). With this heuristic alone we show comparable scores to the ID_Improved heuristic.

The last heuristic I came up with involved the idea of using multiple heuristics with a weighting scheme. For this I chose the above custom_score_2', the ID_Improved algorithm, and the total number of blank spaces with weights of 60%, 30%, and 10% respectively. I thought this would be successful as it would leverage multiple different ideas that have been previously successful on their own (except for the number of blank space) and put a basic weighting scheme to sum the values. This, however, was not effective as I believe rather than creating net greater sum of multiple great heuristics it instead, as the weights would suggest, it took the lesser of each of them.

First we compute a relative score based on (Student - ID_Improved) to create a zero-based metric using ID_Improved as a baseline. We then chart the data as singular elements across the best three of five run trials. We begin by presenting the raw data as follows:

Heuristics	Trial 1	Trial 2	Trial 3	Avg. Over All
custom_score	70.71	65.71	67.14	67.85
ID_Improved	75.00	73.57	73.57	74.05
baseline_score	-4.29	-7.86	-6.43	-6.19
custom_score_2	67.14	72.14	67.14	68.81
ID_Improved	77.14	72.14	74.29	74.52
baseline_score	-10.00	0.00	-7.15	-5.72

Heuristics	Trial 1	Trial 2	Trial 3	Avg. Over All
custom_score_2'	66.43	71.43	75.00	70.95
ID_Improved	65.71	68.57	74.29	69.52
baseline_score	0.72	2.86	0.71	1.43
custom_score_3	66.43	75.71	78.57	73.57
ID_Improved	70.00	72.14	77.86	73.33
baseline_score	-3.57	3.57	0.71	0.24

Given the data and the way we computed each baseline_score I'd recommend using the custom_score_2' moving forward. While custom_score_3 showed strong promise its results fluctuated too much for me to assume strong, constant correlation. Using the euclidean distance to ensure we're as far away as possible from the other player works the best because it ensures we have the highest number of possible positions. Additionally it has a demonstrable and constant improvement against the default ID_Improved heuristic. Last, this heuristic works so well because the 'Knight'-style movement of the players requires a minimum of two spaces for a valid move. Staying as far away as possible allows for the larger number of possibilities to be made.

Moving forward I would like to continue exploring the ideas within custom_score_3 whereby we combine multiple different smaller heuristics with a weight given the varying scores. Determining the weights would, in itself, be a machine learning problem that I'd like to solve. Additionally I believe there might be a better heuristic leveraging euclidean distance but optimizing for getting the player within two squares from the opponent. I feel this would work well because it would actively put the player in the position to reduce and block the opponent from moving into an available space.