Project 3 Report

Summary of Algorithm

My `CustomPlayer` uses the minimax algorithm with iterative deepening (ID), alpha-beta pruning ($\alpha\beta$). I used the basic evaluation function `liberties_own - liberties_opp` with a custom Square Table (SQ) that assigns score to each board position. The actual square table is written below:

In [ ]:
```
SENTINEL = float("-inf")
SQ_SCORE = [
    -3, -1, -1, -1, -1, -1, -1, -1, -1, -1, -3, SENTINEL, SENTINEL,
    -1,  0,  0,  0,  0,  0,  0,  0,  0,  0, -1, SENTINEL, SENTINEL,
    -1,  0,  0, .5, .5, .5, .5, .5,  0,  0, -1, SENTINEL, SENTINEL,
    -1,  0,  0, .5,  1,  1,  1, .5,  0,  0, -1, SENTINEL, SENTINEL,
    -1,  0,  0, .5,  1,  2,  1, .5,  0,  0, -1, SENTINEL, SENTINEL,
    -1,  0,  0, .5,  1,  1,  1, .5,  0,  0, -1, SENTINEL, SENTINEL,
    -1,  0,  0, .5, .5, .5, .5, .5,  0,  0, -1, SENTINEL, SENTINEL,
    -1,  0,  0,  0,  0,  0,  0,  0,  0,  0, -1, SENTINEL, SENTINEL,
    -3, -1, -1, -1, -1, -1, -1, -1, -1, -1, -3, SENTINEL, SENTINEL,
]
```

Performance against Sample Players

	Random	**Greedy**	**Minimax (3)**
Minimax (3)	94.0%	70.8%	49.5%
Minimax (3) + **SQ**	95.8%	77.8%	52.5%
Minimax + ID	95.8%	84.5%	74.0%
Minimax + ID + **SQ**	94.2%	85.2%	77.8%
Minimax + ID + $\alpha\beta$	97.2%	87.2%	73.5%
Minimax + ID + $\alpha\beta$ + **SQ**	96.8%	88.8%	80.0%

Each number in this chart is a result of 100 rounds (400 games) on 4 processes with fair matches flag enabled.

Questions

What features of the game does your heuristic incorporate, and why do you think those features matter in evaluating states during search?

The square table heuristic incorporates the idea that the center of the board is more likely a safer position in the game of Isolation. This heuristic is influential since position near the center increases the likelihood of high number of liberties for the next few moves.

Analyze the search depth your agent achieves using your custom heuristic. Does search speed matter more or less than accuracy to the performance of your heuristic?

Addng the square table heuristic had no visible impact to the search depth, since it only consists of two list lookups. This heuristic is still a relatively simple heuristic, capitalizing on high-depth searches rather than accurate evaluation.

Other

Attempts

There are other ideas I have in improving the model that led to subpar results. Here is a chart with a full list of attempts.

	Random	**Greedy**	**Minimax (3)**
Minimax (3)	94.0%	70.8%	49.5%
Minimax (3) + 8L	93.5%	48.8%	55.2%
Minimax (3) + 2L	91.5%	62.2%	51.5%
Minimax + ID	95.8%	84.5%	74.0%
Minimax + ID + $\alpha\beta$	97.2%	87.2%	73.5%
Minimax + ID + $\alpha\beta$ + CH	95.2%	89.8%	76.2%
Minimax + ID + $\alpha\beta$ + SQ	96.8%	88.8%	80.0%
Minimax + ID + $\alpha\beta$ + SQ10	97.0%	89.8%	76.8%
Minimax + ID + $\alpha\beta$ + SQ20	96.0%	89.8%	79.2%
Minimax + ID + $\alpha\beta$ + SQ40	96.8%	91.0%	79.5%

Each number in this chart is a result of 100 rounds (400 games) on 4 processes with fair matches flag enabled.

8 Liberties (8L)

For the first move, I choose a position with 8 liberties rather than a random position.

2 Liberties (2L)

For the first move, I choose a position with 2 liberties rather than a random position.

Custom Heuristic (CH)

This is a variant of the basic `liberties_own - liberties_opp`. For the first 20 plies (10 moves), it uses an aggressive heuristic `liberties_own - 2 * liberties_opp`, and for the rest of the game, it uses a defensive heuristic `2 * liberties_own - liberties_opp`.

Square Table (SQ)

Like the PCSQ table in chess, I assigned points to each square of the 9x11 board. The square table rewards being in center and penalizes edges and corners.

Square Table Cutoff 20 (SQ10)

Same as above, but I disabled the square table after 10 ply to make the piece focus on survival rather than positioning.

Square Table Cutoff 20 (SQ20)

Same as above, but I disabled the square table after 20 ply.

Square Table Cutoff 40 (SQ40)

Same as above, but I disabled the square table after 40 ply.

Untested Ideas

- Evaluation function dependent on color (ex. White - Defensive, Black - Aggressive)
- Opening Book
- Principal Variation Search (PVS) / Negascout
- Monte Carlo Tree Search (MCTS)