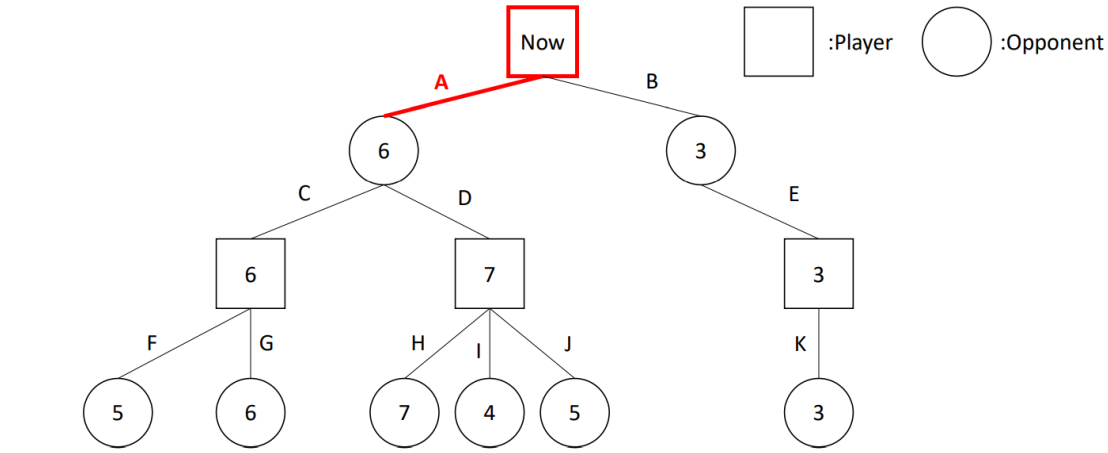
```
double Heuristic(ARR _board, int curPlayer) {
    int count[3] = {};
    double V = 0, D = 0, C = 0, CS = 0, MC = 0;
    ARR w;
    w[0] = {20, -3, 11, 8, 8, 11, -3, 20};
    w[1] = {-3, -7, -3, -1, -1, -3, -7, -3};
    w[2] = {11, -3, 2, 2, 2, 2, -3, 11};
    w[3] = {8, 1, 2, -3, -3, 2, 1, 8};
    w[4] = {8, 1, 2, -3, -3, 2, 1, 8};
    w[5] = {11, -3, 2, 2, 2, 2, -3, 11};
    w[6] = {-3, -7, -3, -1, -1, -3, -7, -3};
    w[7] = {20, -3, 11, 8, 8, 11, -3, 20};

    // Position Values
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            if (_board[i][j] == curPlayer)
                V += w[i][j];
            else if (_board[i][j] == getNextPlayer(curPlayer))
                V -= w[i][j];
            count[_board[i][j]]++;
        }
    }
    // DISC COUNT
    if (count[curPlayer] > count[getNextPlayer(curPlayer)])
        D =  (100.0 * count[curPlayer]) / (count[curPlayer] + count[getNextPlayer(curPlayer)]);
    else if (count[curPlayer] < count[getNextPlayer(curPlayer)])
        D = -(100.0 * count[curPlayer]) / (count[curPlayer] + count[getNextPlayer(curPlayer)]);
    else D = 0;

    // Valid Moves Count
    count[curPlayer] =                get_valid_spots(_board, curPlayer).size();
    count[getNextPlayer(curPlayer)] = get_valid_spots(_board, getNextPlayer(curPlayer)).size();
    if (count[curPlayer] > count[getNextPlayer(curPlayer)])
        MC =  (100.0 * count[curPlayer]) / (count[curPlayer] + count[getNextPlayer(curPlayer)]);
    else if (count[curPlayer] < count[getNextPlayer(curPlayer)])
        MC = -(100.0 * count[curPlayer]) / (count[curPlayer] + count[getNextPlayer(curPlayer)]);
    else MC = 0;

    // Corner Instability
    count[1] = count[2] = 0;
    for (Point c : corners) {
        if (_board[c.x][c.y] == EMPTY) {
            for (int j = 1; j <= 7; j += 2) {
                Point p = c + directions[j];
                if (is_spot_on_board(p)) count[_board[p.x][p.y]]++;
            }
        }
    }
    CS = -20.25 * (count[curPlayer] - count[getNextPlayer(curPlayer)]);

    // Corners Captured
    count[1] = count[2] = 0;
    for (Point c : corners)
        count[_board[c.x][c.y]]++;

    C = 25 * (count[curPlayer] - count[getNextPlayer(curPlayer)]);

    double score = (10 * V) + (11 * D) + (80 * MC) + (375.78 * CS) + (805.131 * C);
    return score;
}
```

https://courses.cs.washington.edu › Final_Paper ▾ PDF

final copy 2 - University of Washington

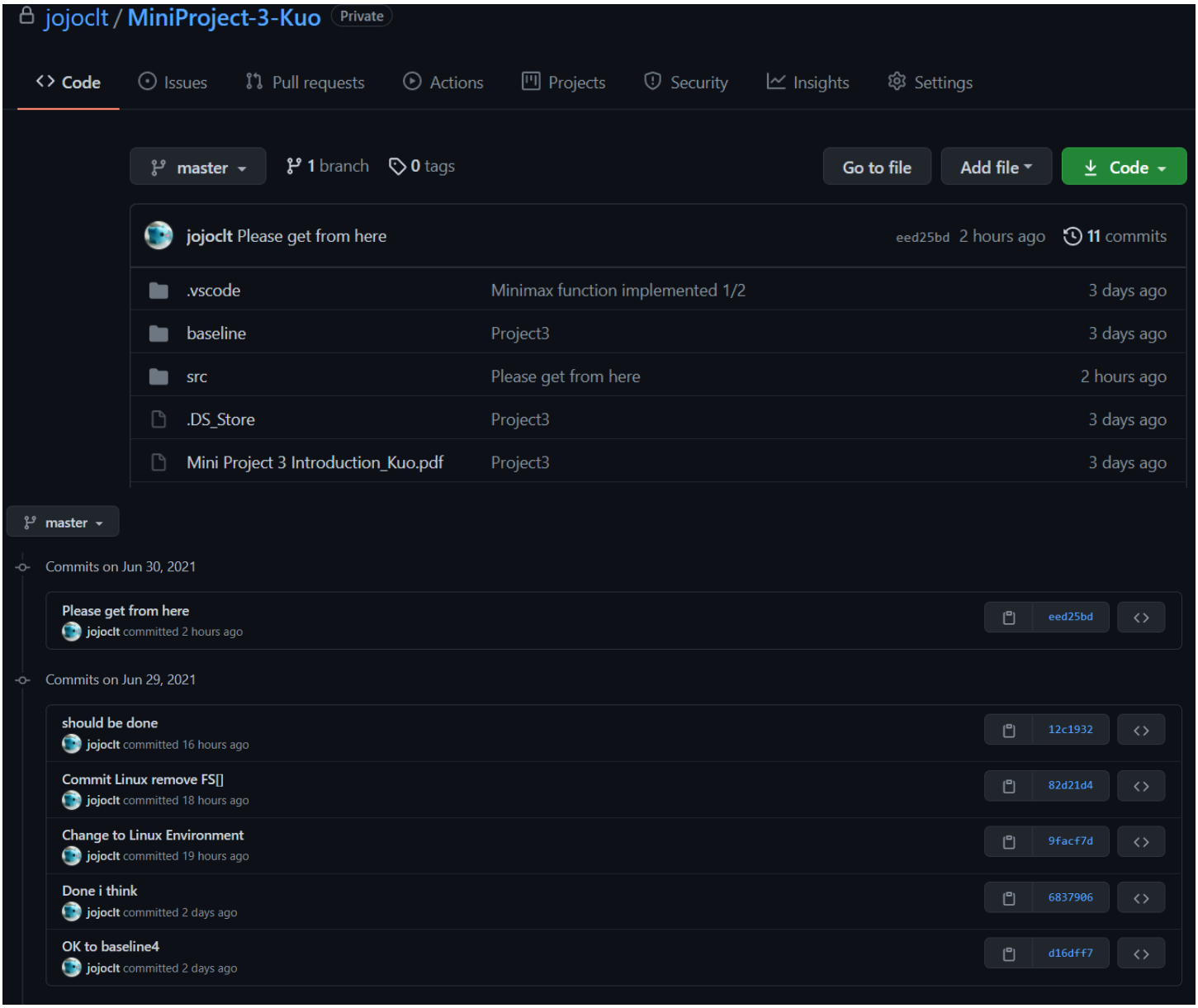
by V Sannidhanam · Cited by 3 — **not** very clear as to how the various **Othello heuristics** ... **complexity** of accurate calculations, most **heuristics** tend ... cannot be flanked at any point of **time** in the.

Player picks move A to be the next move



```cpp
double MiniMax(ARR _board, int depth, int curPlayer) {
    if (depth == 0) return Heuristic(_board, curPlayer);

    vector <Point> nextMove = get_valid_spots(_board, curPlayer);

    if (nextMove.size() == 0) return MiniMax(_board, depth - 1, getNextPlayer(curPlayer));
    //Maximizing
    if (curPlayer == player) {
        double val = -inf;
        for (Point p : nextMove) {
            ARR _state = _board;
            put_disc(_state, p, curPlayer);
            val = std::max(val, MiniMax(_state, depth - 1, getNextPlayer(curPlayer)));
        }
        return val;
    }
    // Minimizing
    else {
        double val = inf;
        for (Point p : nextMove) {
            ARR _state = _board;
            put_disc(_state, p , curPlayer);
            val = std::min(val, MiniMax(_state, depth - 1, getNextPlayer(curPlayer)));
        }
        return val;
    }
}
Point MiniMaxDecision(int depth, std::ofstream& fout) {
    double bestVal = -inf;

    for (auto p : next_valid_spots) {
        ARR _state = board;
        put_disc(_state, p , player);
        double tmp = MiniMax(_state, depth, getNextPlayer(player));

        if (tmp > bestVal) {
            bestMove = p;
            bestVal = tmp;
        }
        fout << bestMove.x << " " << bestMove.y << "\n";
    }
    return bestMove;
}
```

```cpp
double AlphaBeta(ARR _board, int depth, int curPlayer, double a, double b) {
    if (depth == 0) return Heuristic(_board, curPlayer);

    vector <Point> nextMove = get_valid_spots(_board, curPlayer);
    if (nextMove.size() == 0) return AlphaBeta(_board, depth - 1, getNextPlayer(curPlayer), a, b);
    //Maximizing
    if (curPlayer == player) {
        double val = -inf;
        for (Point p : nextMove) {
            ARR _state = _board;
            put_disc(_state, p, curPlayer);
            val = std::max(val, AlphaBeta(_state, depth - 1, getNextPlayer(curPlayer), a, b));

            a = std::max(a, val);
            if (a >= b) break;
        }
        return val;
    }
    // Minimizing
    else {
        double val = inf;
        for (Point p : nextMove) {
            ARR _state = _board;
            put_disc(_state, p , curPlayer);
            val = std::min(val, AlphaBeta(_state, depth - 1, getNextPlayer(curPlayer), a, b));

            b = std::min(b, val);
            if (b <= a) break;
        }
        return val;
    }
}
Point AlphaBetaDecision(int depth, std::ofstream& fout) {
     double bestVal = -inf;

            // std::ofstream log("TRY.txt");
    for (auto p : next_valid_spots) {
        ARR _state = board;
        put_disc(_state, p , player);
        double tmp = AlphaBeta(_state, depth, getNextPlayer(player), -inf, inf);

        if (tmp > bestVal) {
            bestMove = p;
            bestVal = tmp;
        }

            // log << bestMove.x << " " << bestMove.y << "\n";

        fout << bestMove.x << " " << bestMove.y << "\n";
    }
    return bestMove;
}
```


jojoclt / MiniProject-3-Kuo
Private
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
master
1 branch
0 tags
Go to file
Add file
Code
jojoclt Please get from here
eed25bd 2 hours ago
11 commits
.vscode
Minimax function implemented 1/2
3 days ago
baseline
Project3
3 days ago
src
Please get from here
2 hours ago
.DS_Store
Project3
3 days ago
Mini Project 3 Introduction_Kuo.pdf
Project3
3 days ago
master
Commits on Jun 30, 2021
Please get from here
jojoclt committed 2 hours ago
eed25bd
Commits on Jun 29, 2021
should be done
jojoclt committed 16 hours ago
12c1932
Commit Linux remove FS[]
jojoclt committed 18 hours ago
82d21d4
Change to Linux Environment
jojoclt committed 19 hours ago
9facf7d
Done i think
jojoclt committed 2 days ago
6837906
OK to baseline4
jojoclt committed 2 days ago
d16dff7