02.12.2016

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

Präsentation Blatt 6

AUFGABE 1 – TIC TAC TOE

Aufgabe 1a – zufälliger Agent

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

```
public Move getMove(GameBoard board) {

        List<Move> possibleMoves = board.getPossibleMoves();
        int size = possibleMoves.size();

        if (size == 0) {
                return null;
        }
        int randomMove = (int) Math.round((Math.random() * (size - 1)));
        return possibleMoves.get(randomMove);
}
```

Aufgabe 1b - Reaktiv

02.12.2016

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

Regeln:

- Wenn ich gewinnen kann, tue ich das.
- Wenn der Gegner gewinnen kann, verhindere ich das.

1) Erzeuge “Zwickmühlen”

2) Verhindere “Zwickmühlen”

SONST: Zufällig

➔ Kann noch geschlagen werden

Aufgabe 1b - Reaktiv

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

```
public List<Move> getWinningMoves(GameBoard board, Player player) {
        List<Move> winning = new ArrayList<Move>();
        List<Move> possibleMoves = new ArrayList<>(board.getPossibleMoves());
        for (Move move : possibleMoves) {
                board.move(player, move);
                if (board.isWonBy(player, move)) {
                        winning.add(move);}
                board.undoMove(player, move);    }
        return winning;
}

public List<Move> getForkMoves(GameBoard board, Player player) {
        List<Move> forking = new ArrayList<Move>();
        List<Move> possibleMoves = new ArrayList<>(board.getPossibleMoves());
        for (Move move : possibleMoves) {
                board.move(player, move);
                List<Move> winningMoves = getWinningMoves(board, player);
                if (winningMoves.size() > 1) {
                        forking.add(move);       }
                board.undoMove(player, move);    }
        return forking;
}
```

Aufgabe 1b - Reaktiv

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

```
public Move getMove(GameBoard board) {
        List<Move> winningMoves = getWinningMoves(board, current);
        if (!winningMoves.isEmpty()) {
                return winningMoves.get(0);}

        List<Move> winningOpponentMoves = getWinningMoves(board, opponent);
        if (!winningOpponentMoves.isEmpty()) {
                return winningOpponentMoves.get(0);}

        if (level >= 1) {
                List<Move> forkMoves = getForkMoves(board, current);
                if (!forkMoves.isEmpty()) {
                        return forkMoves.get(0);}

                if (level >= 2) {
                        List<Move> forkOpponentMoves = getForkMoves(board, opponent);
                        if (!forkOpponentMoves.isEmpty()) {
                                return forkOpponentMoves.get(0);}
                }
        }

        return super.getMove(board); //random move
}
```

Aufgabe 1c – Minimax

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

```
minimax(level, player)  // player may be „current" or "opponent"
if (gameover || level == 0)
   return score
children = all legal moves for this player
if (player is current, i.e., maximizing player)
   // find max
   bestScore = -inf
   for each child
      score = minimax(level - 1, opponent)
      if (score > bestScore) bestScore = score
   return bestScore
else //(player is opponent, i.e., minimizing player)
   // find min
   bestScore = +inf
   for each child
      score = minimax(level - 1, current)
      if (score < bestScore) bestScore = score
   return bestScore


// Initial Call
minimax(9, current)
```

Aufgabe 1c – Minimax

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

Score Berechnung (Einfach):

- +1 für Sieg , -1 für Niederlage, Sonst 0

Score Berechnung (Heuristisch):

- +100 für 3 in einer Reihe = Sieg
- +10 für 2 in einer Reihe (ein leeres Feld)
- +1 für 1 in einer Reihe (zwei leere Felder)
- Sonst 0 (leere oder nicht gewinnbare Reihe)
- Negative Scores für Gegner

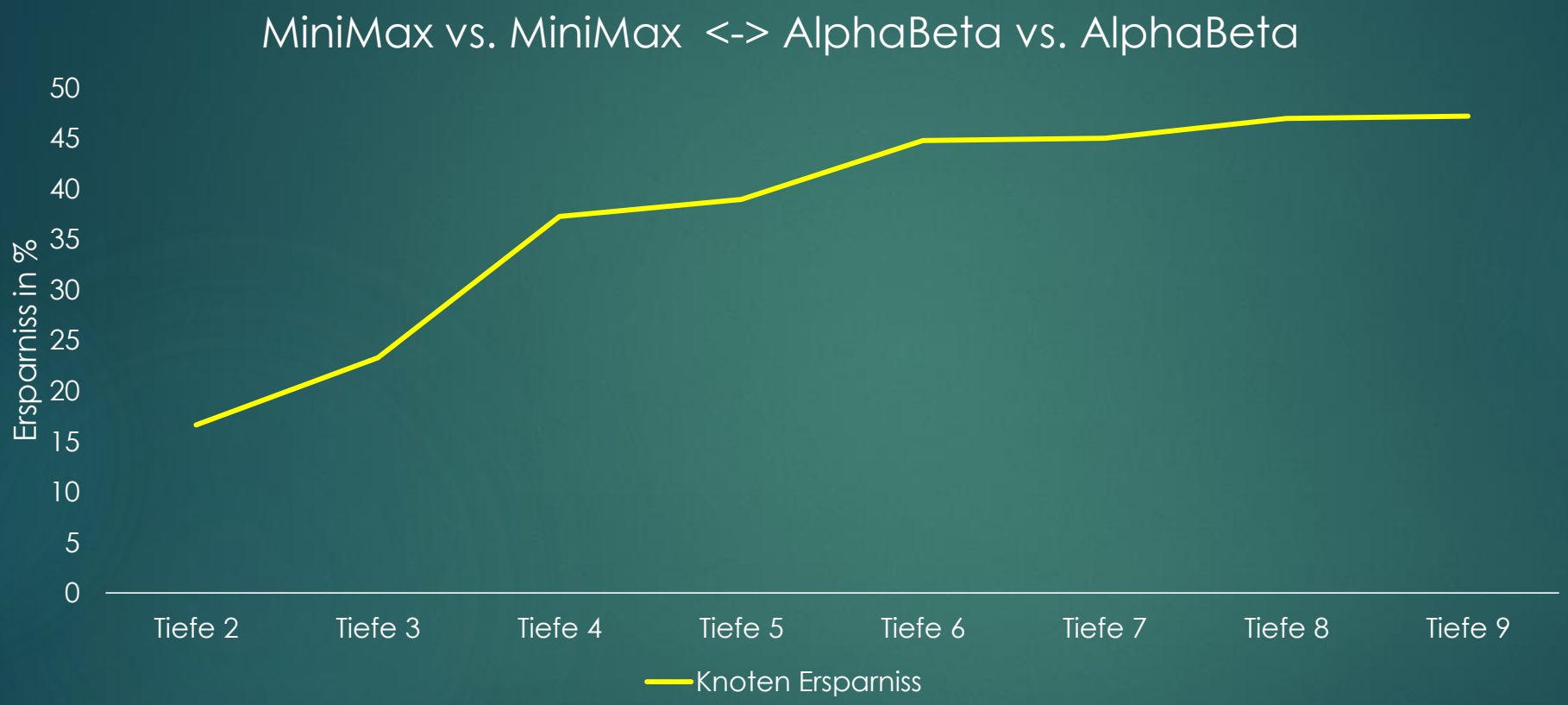
Aufgabe 1d – Alpha-Beta

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

```
alphaBeta(level, player, alpha, beta)  // player may be „current" or "opponent"
if (gameover || level == 0)
   return score
children = all legal moves for this player
if (player is current, i.e., maximizing player)
   // find max and store in alpha
   for each child
      score = minimax(level - 1, opponent, alpha, beta)
      if (score > alpha) alpha = score
      if (alpha >= beta) break;  // beta cut-off
   return alpha
else (player is opponent, i.e., minimizing player)
   // find min and store in beta
   for each child
      score = minimax(level - 1, current , alpha, beta)
      if (score < beta) beta = score
      if (alpha >= beta) break;  // alpha cut-off
   return beta


// Initial Call
minimax(9, current, -inf, +inf)
```

Aufgabe 1d – Alpha-Beta



Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

- Laufzeit wird mit steigender Suchtiefe deutlich verbessert

Aufgabe 1e – Heuristiken

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

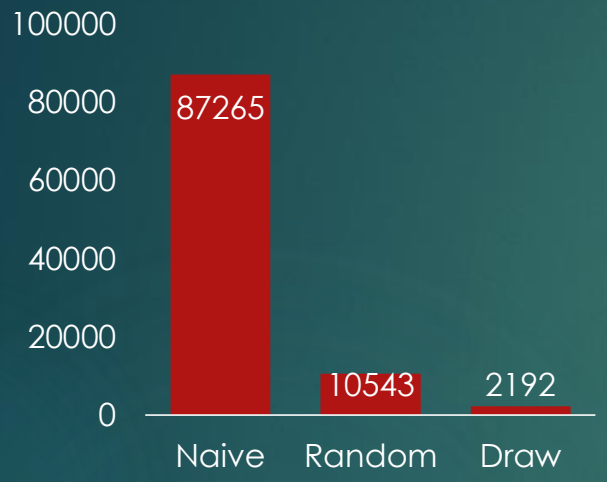
- Tic-Tac-Toe: siehe Folie 7
- 4 Gewinnt: (Negative Punkte für Gegner)
  - 1000 Punkte für Sieg
  - 100 Punkte für 3 in einer Reihe (noch gewinnbar)
  - 10 Punkte für 2 in einer Reihe (noch gewinnbar)
  - 1 Punkt für 1 in einer Reihe (noch gewinnbar)
  - Doppelte Punktzahl wenn Reihe in beide Richtungen noch gewinnbar
  - 0 Punkte für leere oder nicht gewinnbare Reihen

Strategie Vergleich

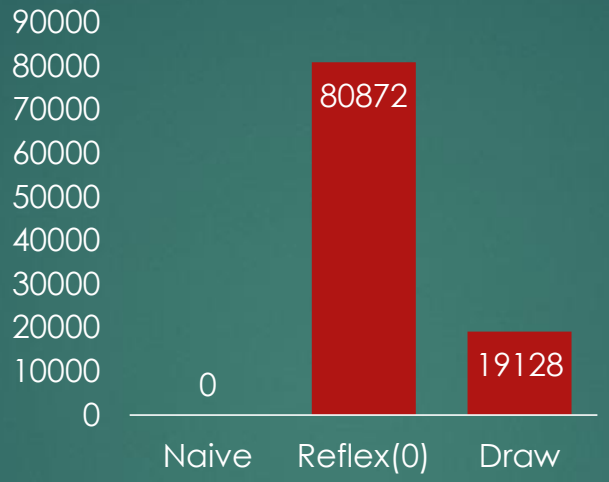
02.12.2016

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

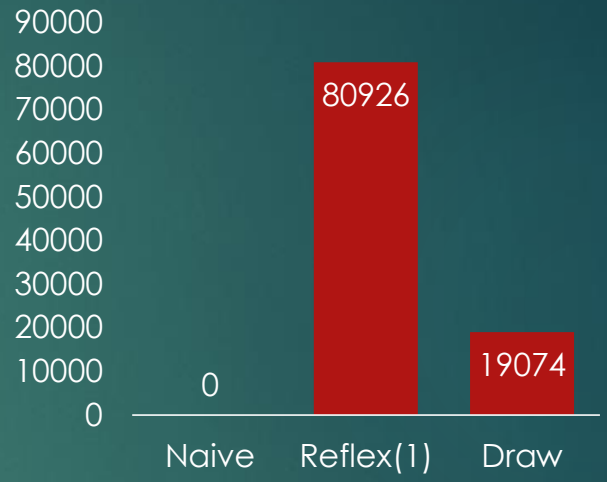
Naive vs. Random



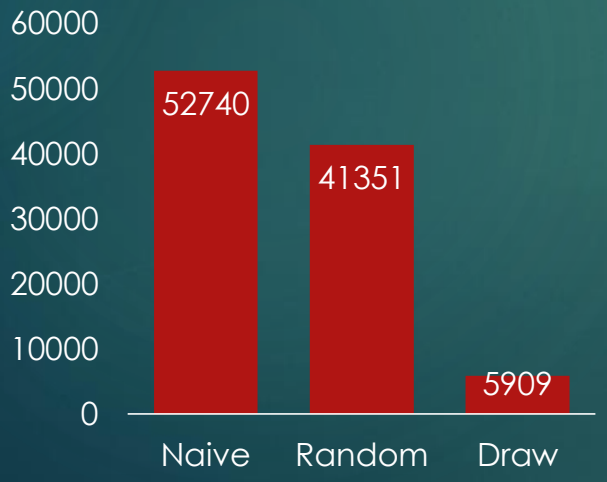
Naive vs. Reflex(0)



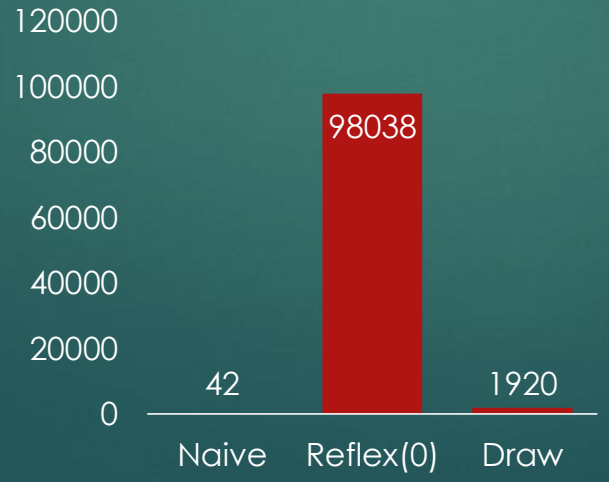
Naive vs. Reflex(1)



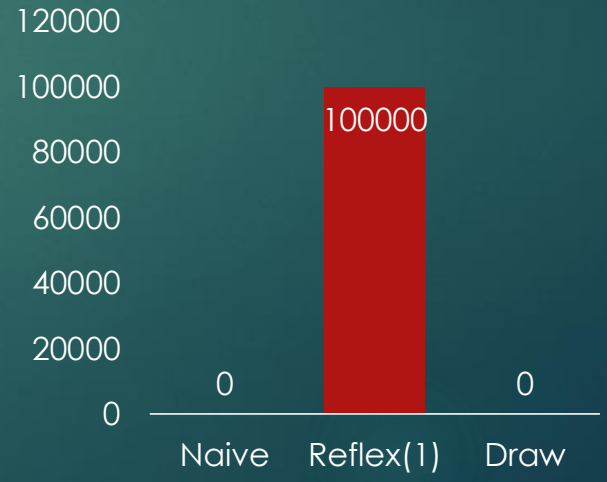
Random vs. Naive



Reflex(0) vs. Naive



Reflex(1) vs. Naive

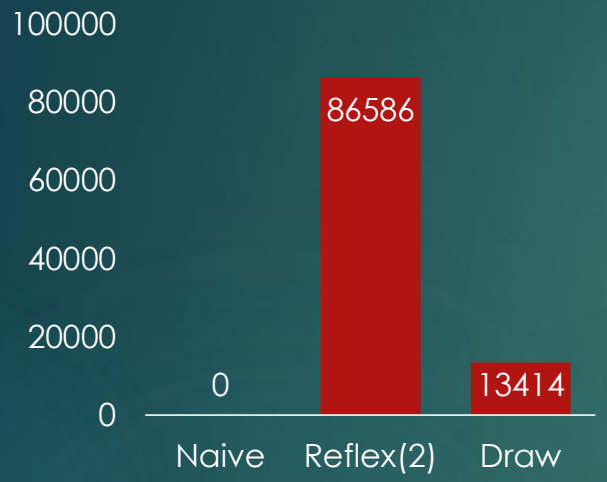


Strategie Vergleich

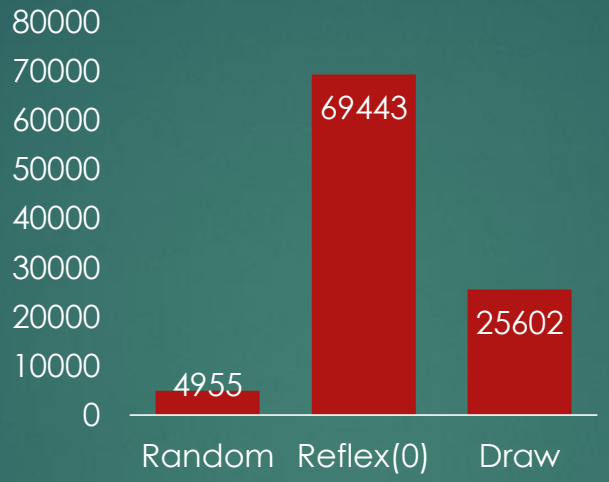
02.12.2016

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

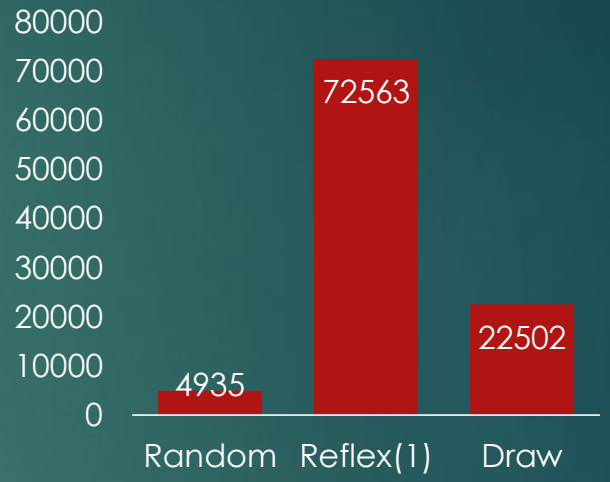
Naive vs. Reflex(2)



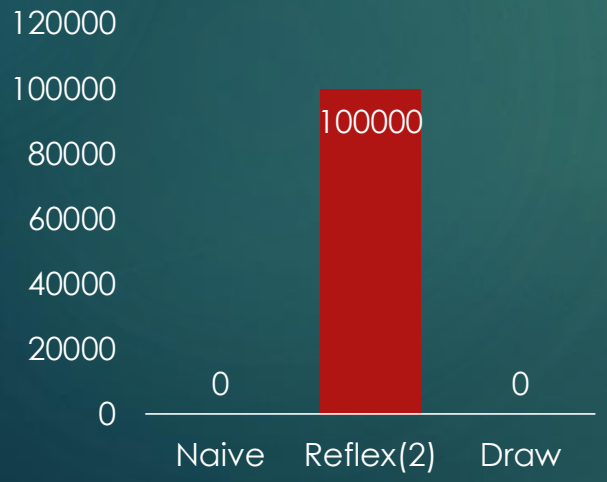
Random vs. Reflex(0)



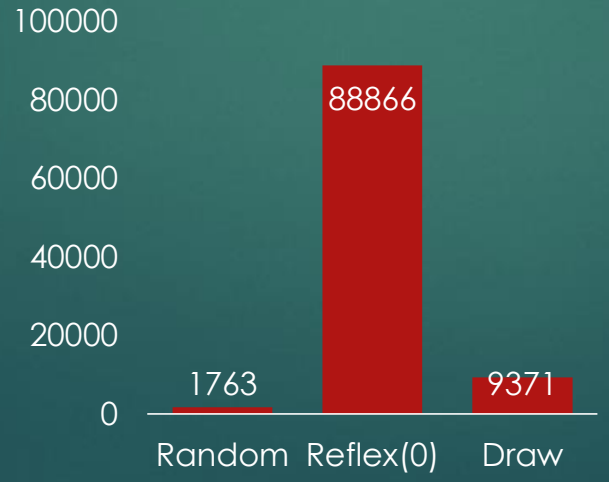
Random vs. Reflex(1)



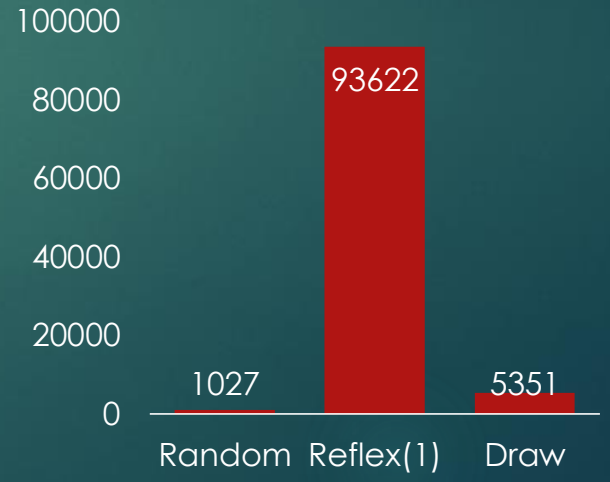
Reflex(2) vs. Naive



Reflex(0) vs. Random



Reflex(1) vs. Random

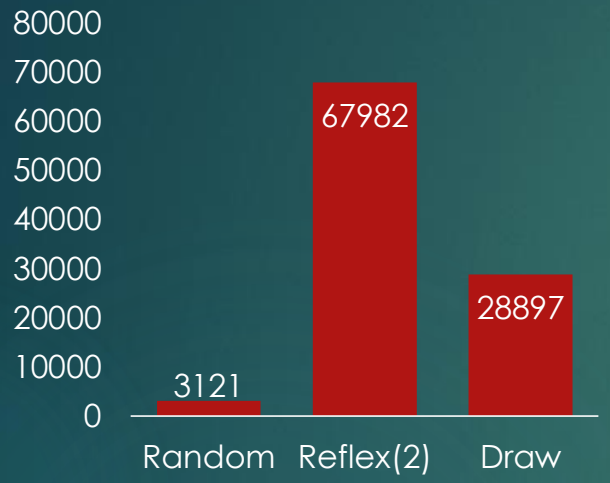


Strategie Vergleich

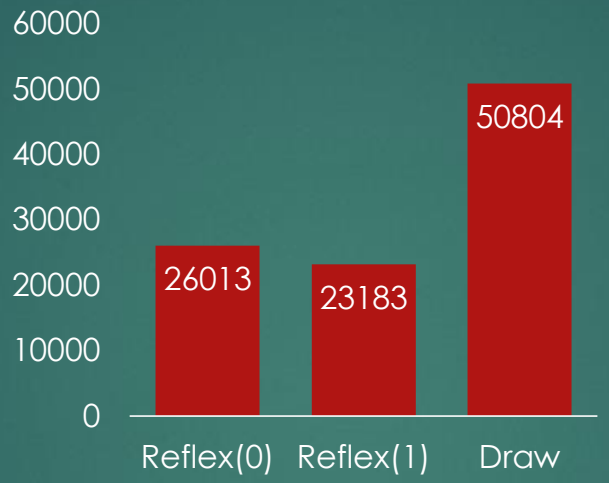
02.12.2016

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

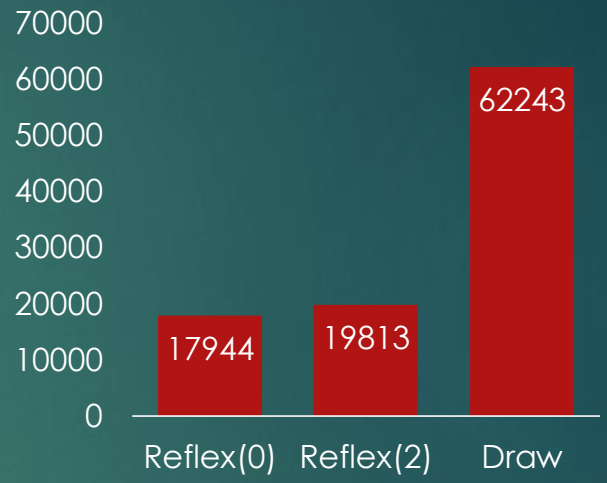
Random vs. Reflex(2)



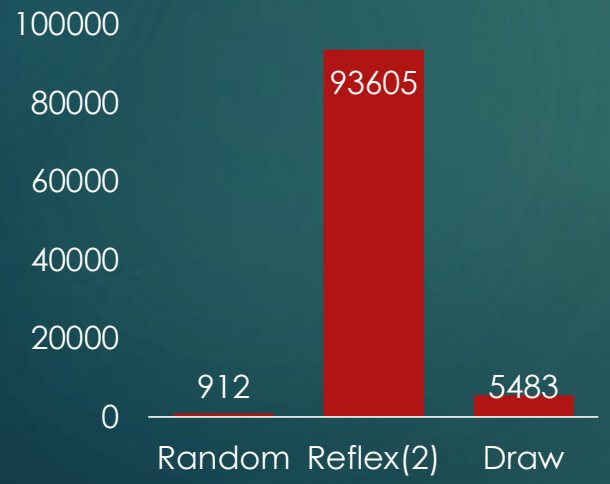
Reflex(0) vs. Reflex(1)



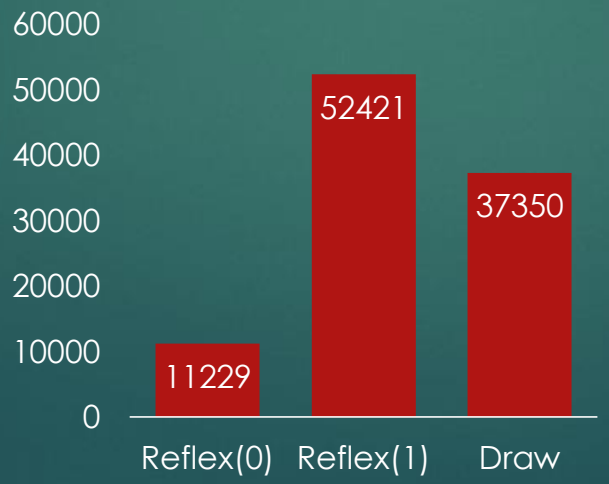
Reflex(0) vs. Reflex(2)



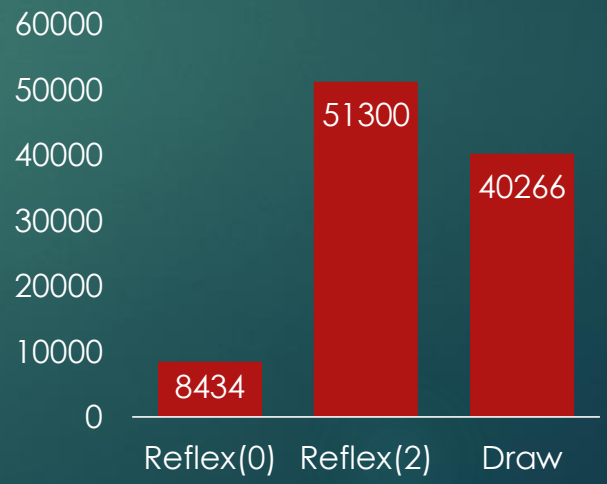
Reflex(2) vs. Random



Reflex(1) vs. Reflex(0)



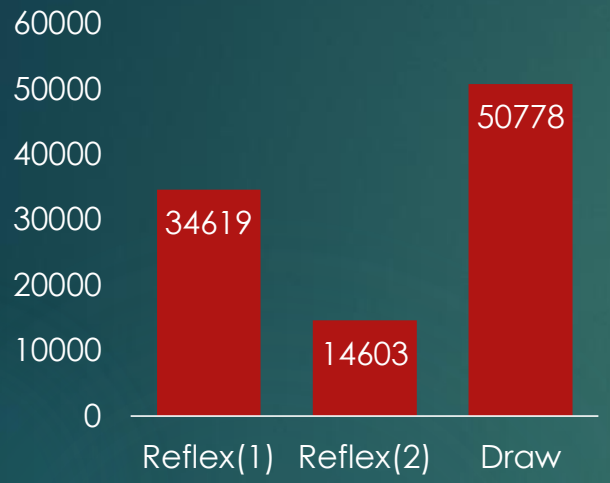
Reflex(2) vs. Reflex(0)



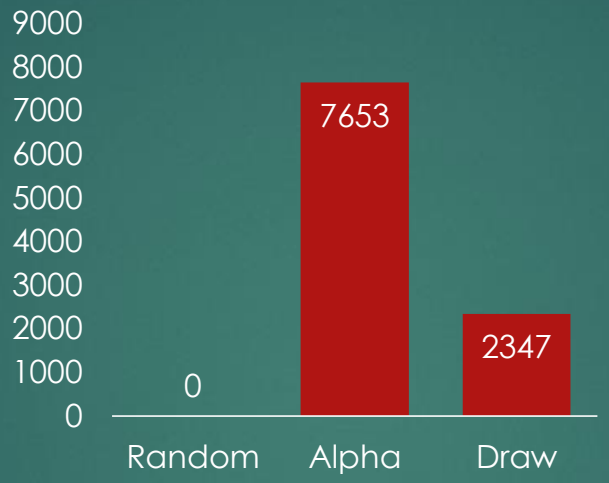
Strategie Vergleich

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

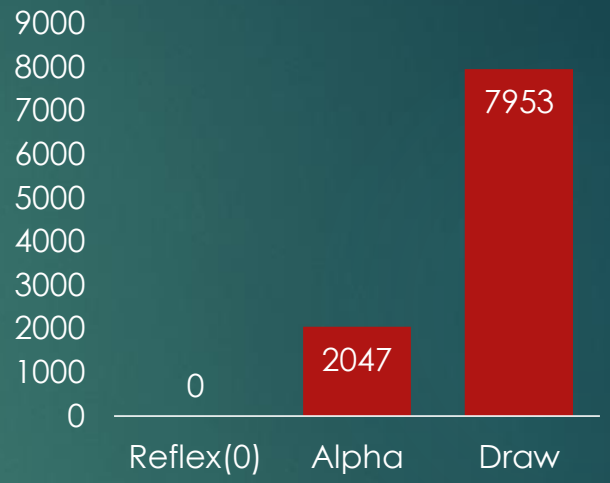
Reflex(1) vs. Reflex(2)



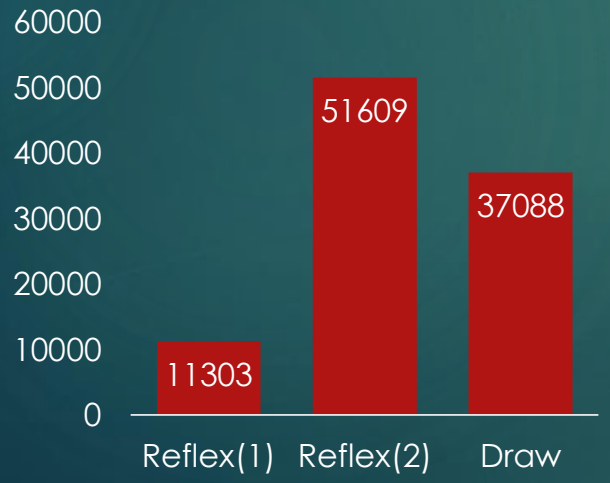
Random vs. Alpha



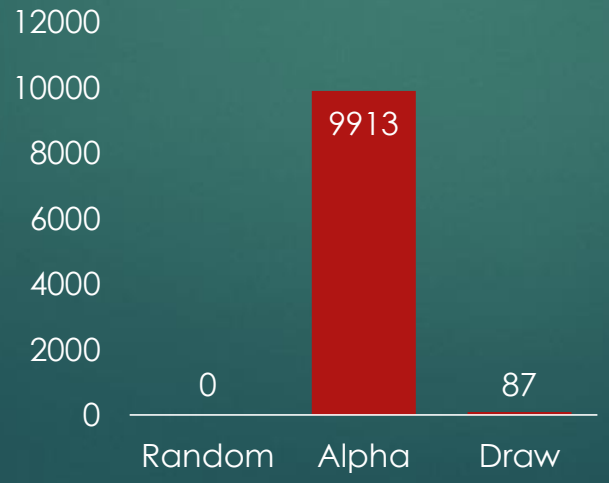
Reflex(0) vs. Alpha



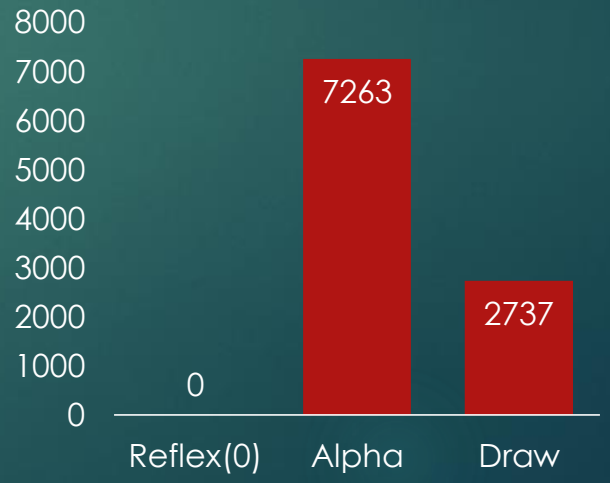
Reflex(2) vs. Reflex(1)



Alpha vs. Random



Alpha vs. Reflex(0)

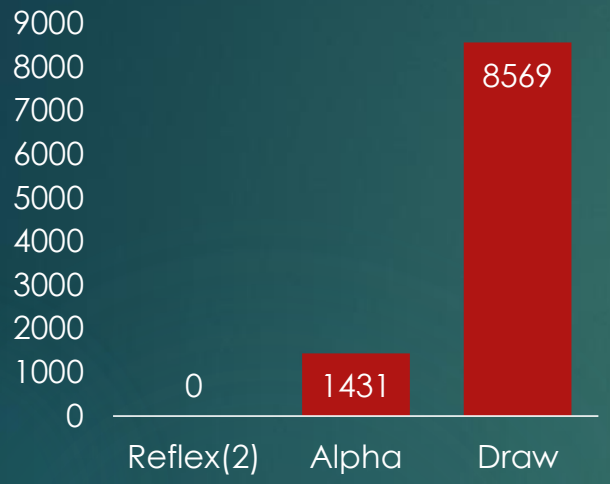


Strategie Vergleich

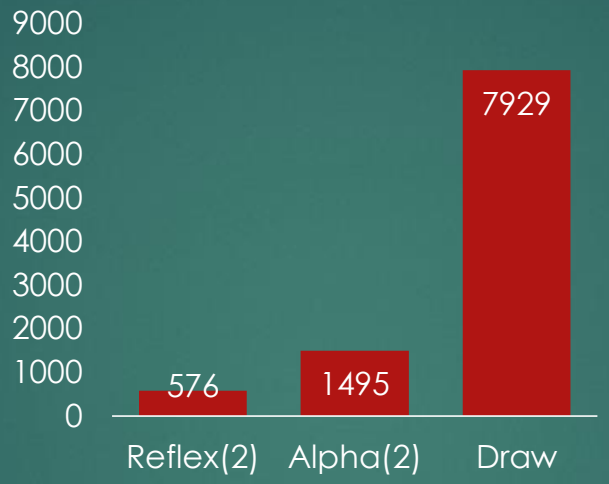
02.12.2016

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

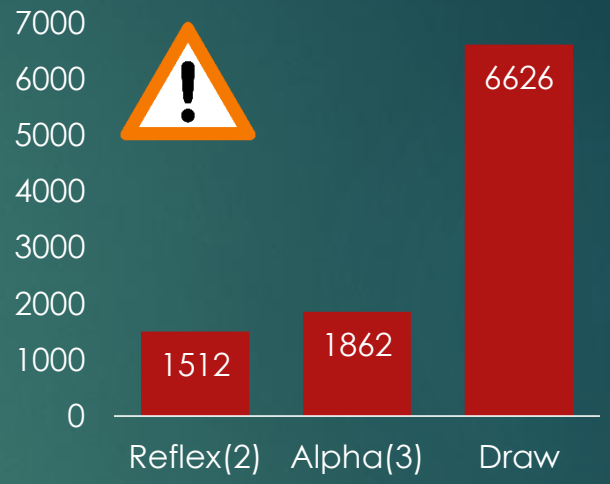
Reflex(2) vs. Alpha



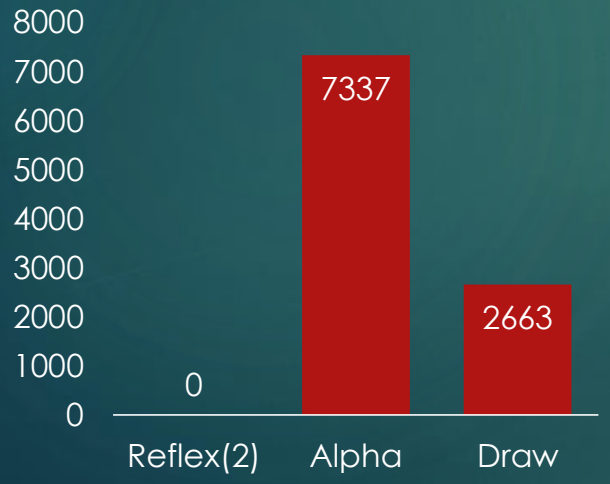
Reflex(2) vs. Alpha(2)



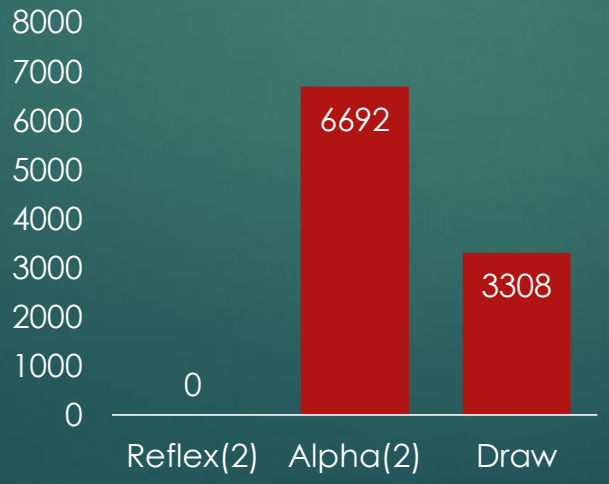
Reflex(2) vs. Alpha(3)



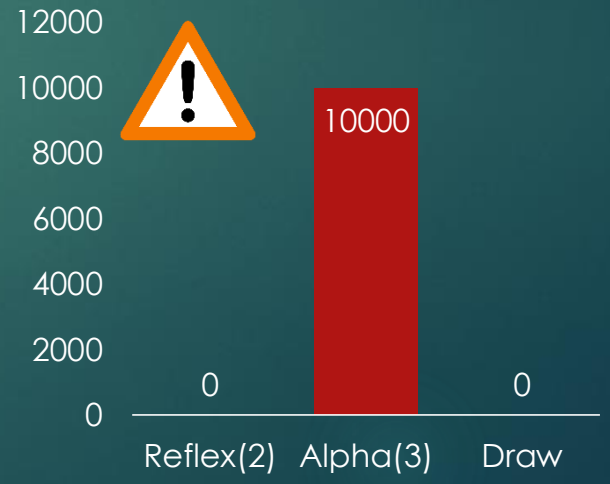
Alpha vs. Reflex(2)



Alpha(2) vs. Reflex(2)



Alpha(3) vs. Reflex(2)

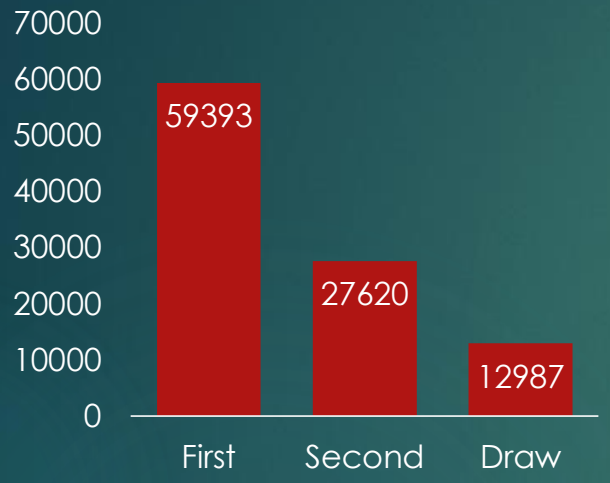


Strategie Vergleich

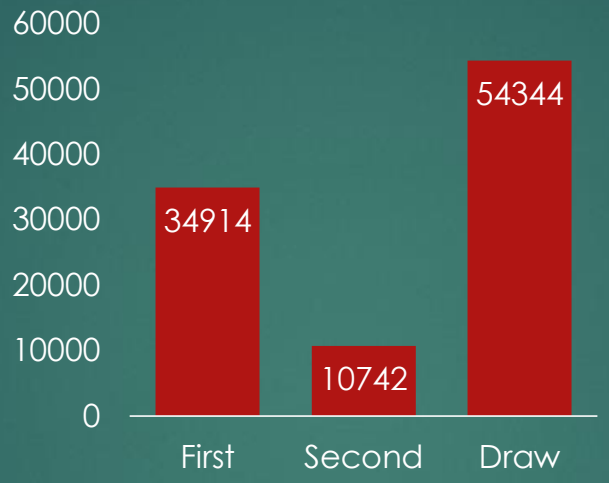
02.12.2016

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

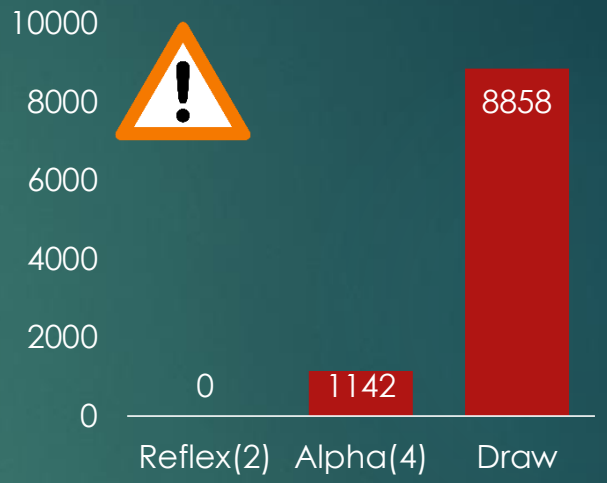
Random vs. Random



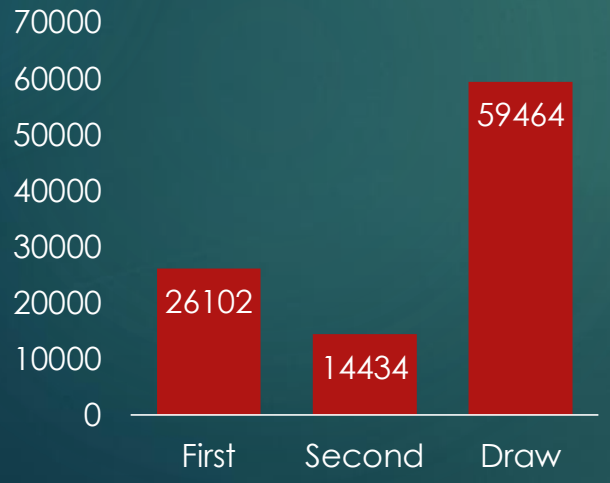
Reflex(2) vs. Reflex(2)



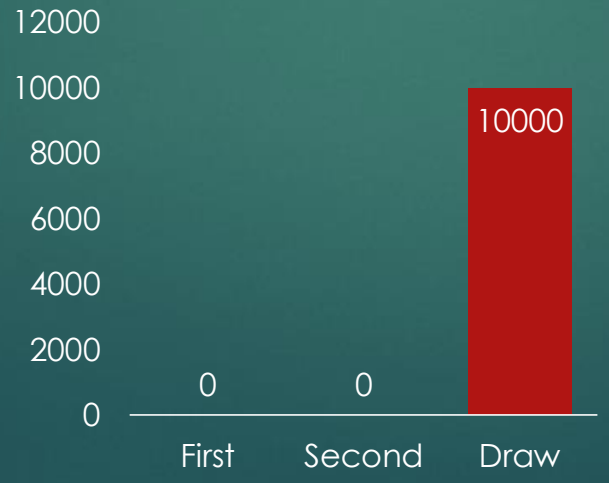
Reflex(2) vs. Alpha(4)



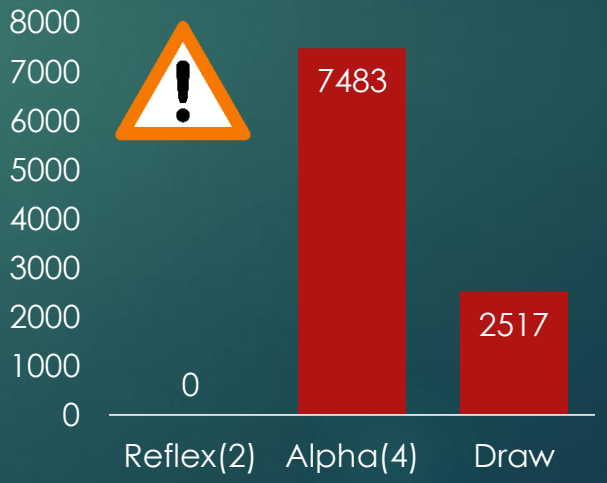
Reflex(0) vs. Reflex(0)



Alpha vs. Alpha



Alpha(4) vs. Reflex(2)



Minimax bzw. Alpha-Beta Algorithmus:

https://www.ntu.edu.sg/home/ehchua/programming/java/JavaGame_TicTacToe_AI.html

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

Gruppe 3
(Martin Schörner,
Matthias Gröbner,
David Winter)

Ende

Vielen Dank für die Aufmerksamkeit!