

# FIANCO

## INTELLIGENT SEARCH & GAMES

STUDENT: MARIA AURORA BOTA STUDENT NUMBER: 6369347

# CONTENTS

- LUDEME
- HEURISTICS
- ITERATIVE-DEEPENING & MOVE ORDERING
- TRANSPOSITION TABLE
- VARIABLE DEPTH SEARCH: MULTI-CUT
- EXPERIMENTAL TUNING: PLAYING AGAINST MC-GRAVE
- REFERENCES

# LUDEME

- Based on Draughts Game
- Conditioned to move only Forward-Left and Forward-Right
- “DoubleCounter” changed to capture only once
- Changed winning move condition

```
(game "Maria"
  (players { (player N) (player S) })
  (equipment
    {
      (board (square 9))
      (piece "Counter" P1 N)
      (piece "Counter" P2 S)
      (piece "DoubleCounter" Each)
      (regions P1 (sites Bottom))
      (regions P2 (sites Top))
    }
  )
  (rules
    (start
      {
        (place
          "Counter1"
          (difference
            (expand (union (sites Bottom) (sites {"B2" "H2" "C3" "G3" "D4" "F4"})) steps:(- 1 1))
            (sites Phase 2)
          )
        )
        (place
          "Counter2"
          (difference
            (expand (union (sites Top) (sites {"B8" "H8" "C7" "G7" "D6" "F6"})) steps:(- 1 1))
            (sites Phase 2)
          )
        )
      }
    )
  )
)
```

# HEURISTICS

## 1. Getnumberofpieces

- Score based on the difference between the number of pieces each player own

## 2. Getleadingpiceas

- Getting the leading pieces in the game based on the row and direction
- Gives more weight to the pieces that are deeper into opponent's territory rewarding a bit aggressively

## 3. Getedgepieces

- Considering the edges as board controlling area, strategically important
- Trying to maintain a balanced number of pieces on the edge so after a threshold the score is diminished

## 4. Getnumberofdefenders

- Rewards players for having pieces on the first row (Player 1) and last row (Player 2) that act as defenders
- It also penalizes the opponent if they fail to occupy key rows
- It is blocking the opponent from winning

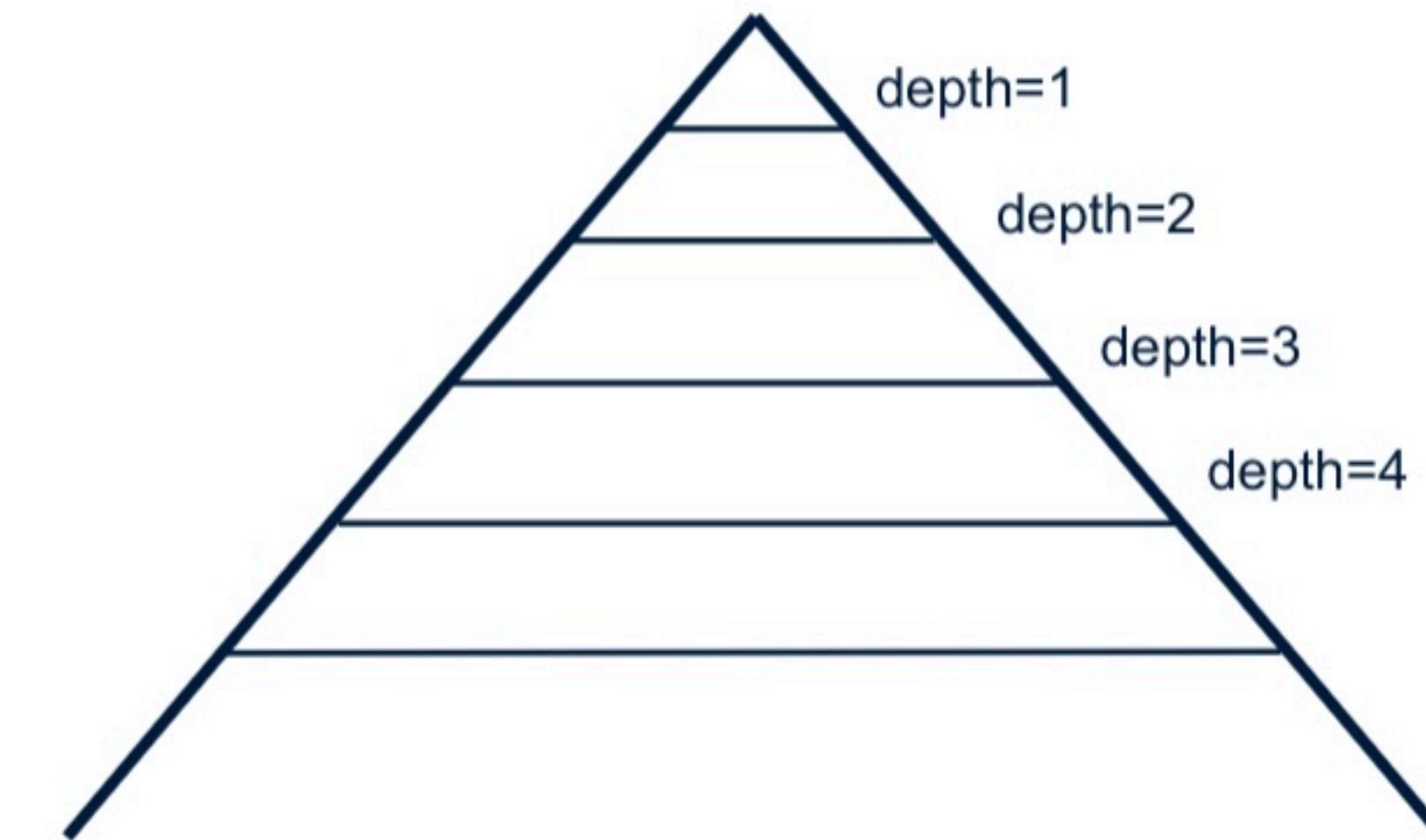
## 5. Getwinner

- The winner gets a very large score

# ITERATIVE DEEPENING & MOVE ORDERING

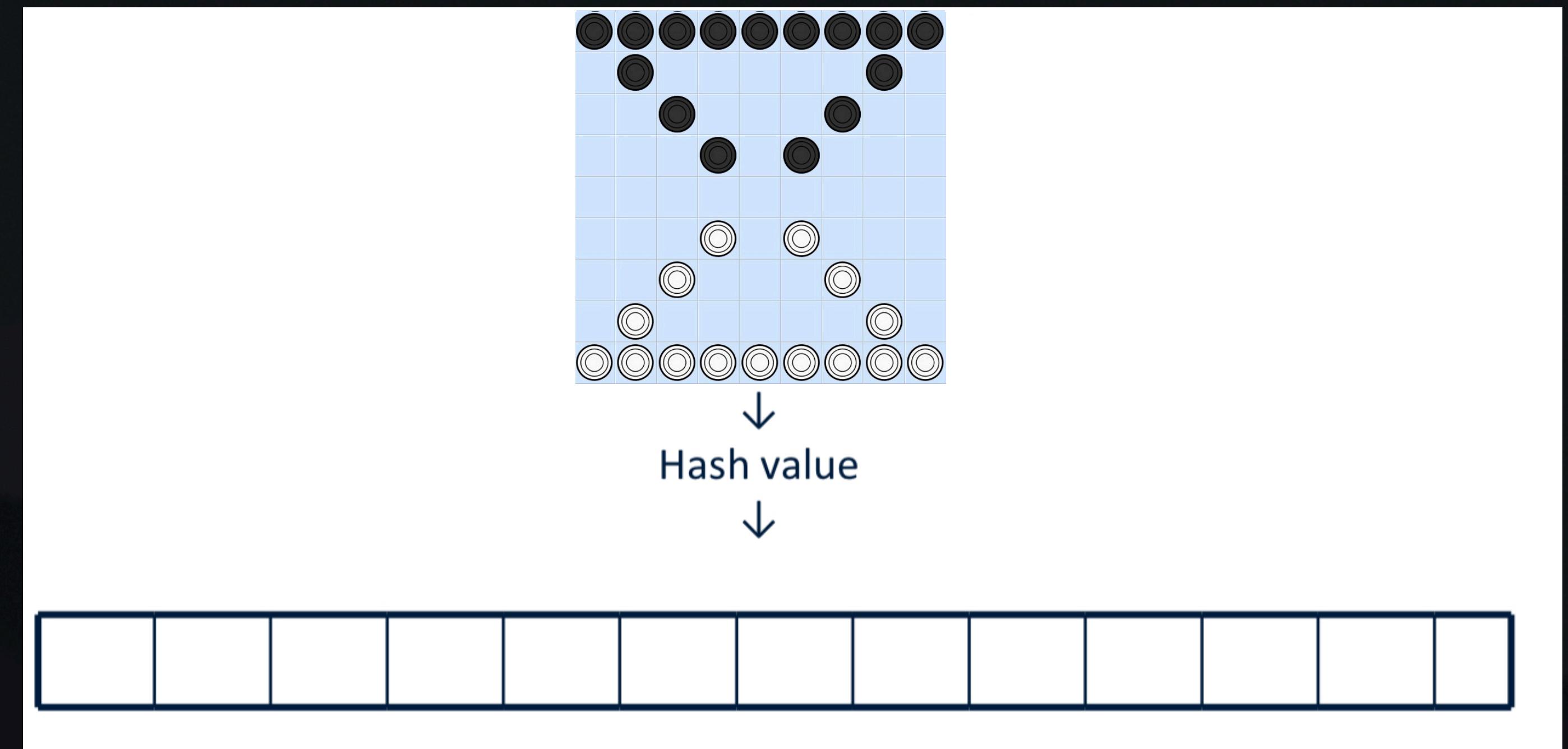
- NegaMax implementation for both alpha-beta search and iterative-deepening
- Inspired [Link](#) used similar functions
- The moves are sorted based on their scores in order to rank them from best to worst

Iterative-deepening depth-first search  
*Fixed depth*



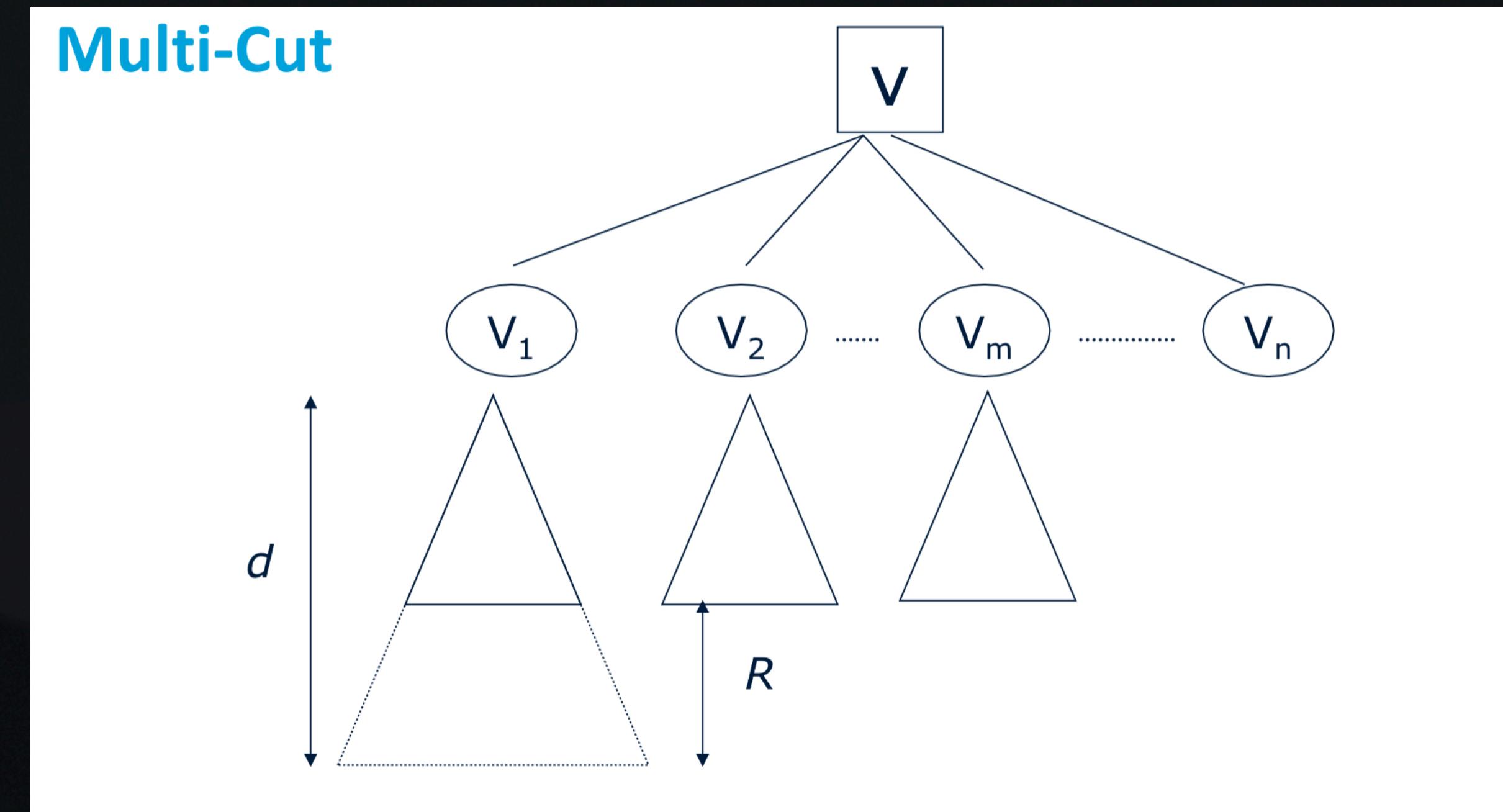
# TRANSPOSITION TABLE

- Significant improvement in playing (was always winning against MC-GRAVE)
- Inspired from the Class slides pseudocode



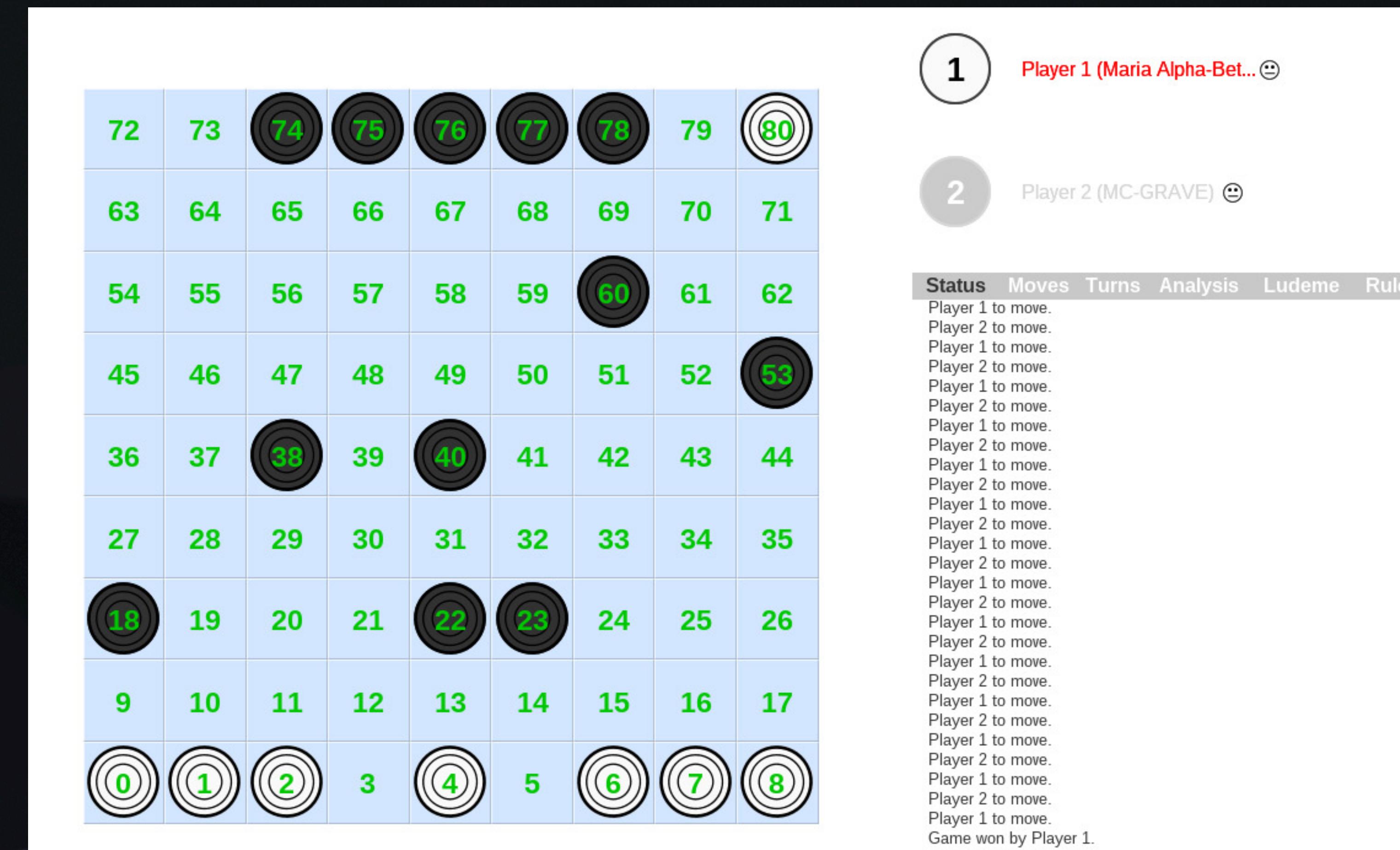
# VARIABLE DEPTH SEARCH: MULTI-CUT

- Parameters settings:  
 $C=2, M=10, R=2$
- Inspired from the class slides  
pseudocode and directions



# EXPERIMENTAL TUNING: PLAYING AGAINST MC-GRAVE

- The game wins in the most games against MC - GRAVE
  - It beats it as both black and white



# REFERENCES

- Lurdeme/Ludii git repo <https://github.com/Lurdeme/Ludii/blob/master/AI/src/search/minimax/AlphaBetaSearch.java>
- Ludii game <https://ludii.games/details.php?keyword=English> Draughts
- E. Piette, D. J. N. J. Soemers, M. Stephenson, C. F. Sironi, M. H. M. Winands, and C. Browne, Ludii - the ludemic general game system, Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020) (G. De Giacomo, A. Catala, B. Dilkina, M. Milano, S. Barro, A. Bugarín, and J. Lang, eds.), Frontiers in Artificial Intelligence and Applications, vol. 325, IOS Press, 2020, pp. 411-418.
- Mark Winands, Intelligent search and games slides 2024