

# AlphaBeta-based agent for playing Boku

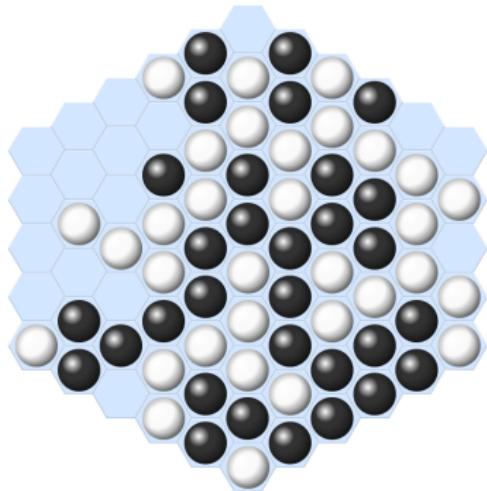
José Manuel Ros Rodrigo

Department of Advanced Computing Sciences  
Maastricht University

28/10/2023

## Main features

- Ludii
  - Incremental evaluation function
  - Iterative deepening
  - Forced move detection
  - Transposition table: bucket scheme
  - Killer moves
  - Multi-cut



# Table of contents

- ① Ludeme
- ② Heuristics
- ③ Iterative deepening
- ④ Memoization: TT and killer moves
- ⑤ Variable depth search: multi-cut
- ⑥ Tuning: playing against MC-GRAVE
- ⑦ References

# Implementation of Boku using Ludii

- Based on Tic Tac Toe's implementation.
- Custodial Ludeme to implement capture moves.
- Use of pending states to remove opponent's marbles.

```

(define "SandwichedPiece"
  (custodial
    (from (last To))
    Orthogonal
    (between
      (exact 2)
      (if:(is Enemy (who at:(between))))
        (apply (set Pending (between))))
      (to if:(is Friend (who at:(to))))
        (then (moveAgain)))
    )
  )
)

(game "Boku"
  (players 2)
  (equipment
    {
      (board (rotate 30 (hex {6 6 5 6 6 5})))
      (piece "Ball" Each)
    }
  )
  (rules
    (play
      (if (no Moves Mover) (pass)
        (if (not "SameTurn")
          (move Add (to (difference (sites Empty)
            (last To))) (then "SandwichedPiece"))
          (move Remove (sites Pending))
        )
      )
    )
    (end {
      (if (is Line 5 what:(id "Ball" Mover)) (result Mover Win))
      (if (and (no Moves Mover) (no Moves Next))
        (result Mover Draw))
    })
  )
)

```

# Table of contents

① Ludeme

② Heuristics

③ Iterative deepening

④ Memoization: TT and killer moves

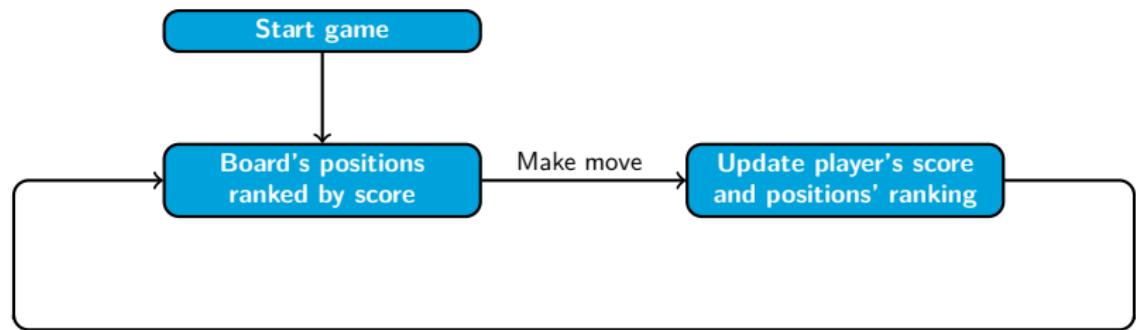
⑤ Variable depth search: multi-cut

⑥ Tuning: playing against MC-GRAVE

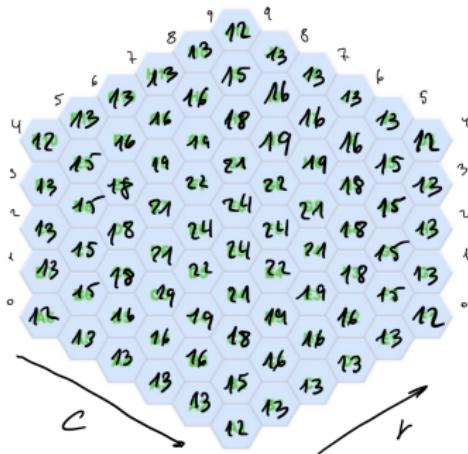
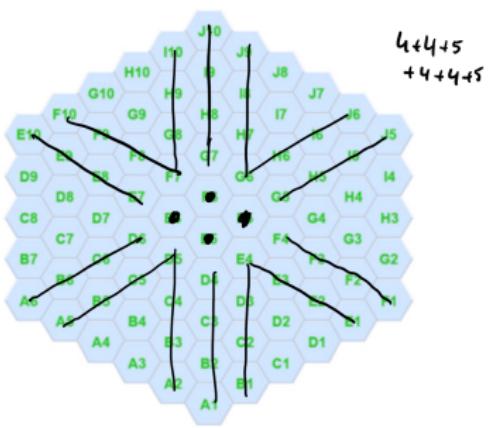
⑦ References

# Intuitive idea

## Game loop

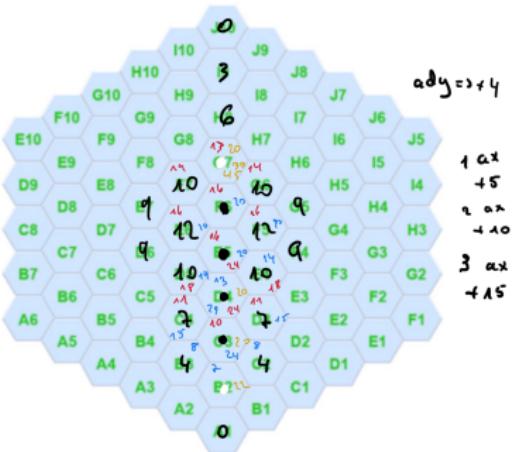


# Base board's scores: degree of freedom



# Updating the board

- Adjacent cells.
- Line within the same row.
- Line within the same column.
- Line within the same diagonal.
- Expensive but gives a really decent move ordering.



# Evaluation

- If last move = Add, then your score increases by your estimated value of the cell.
- If last move = Remove, then opponent's score decreases by their estimated value of the cell.
- The estimated value of the node is equal to your score minus opponent's score.

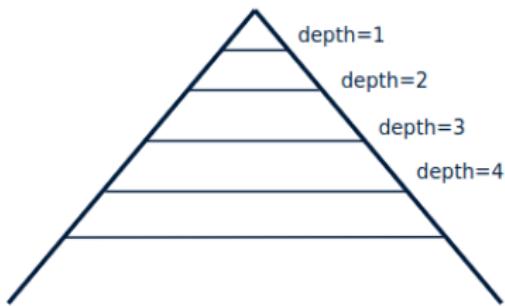
# Table of contents

- ① Ludeme
- ② Heuristics
- ③ Iterative deepening
- ④ Memoization: TT and killer moves
- ⑤ Variable depth search: multi-cut
- ⑥ Tuning: playing against MC-GRAVE
- ⑦ References

# Relevant features

- Forced move detection.
- Plays for 15 seconds.
- Negamax implementation.
- Reaches depth 7-8.

Iterative-deepening depth-first search



# Table of contents

- ① Ludeme
- ② Heuristics
- ③ Iterative deepening
- ④ Memoization: TT and killer moves
- ⑤ Variable depth search: multi-cut
- ⑥ Tuning: playing against MC-GRAVE
- ⑦ References

# Remarks

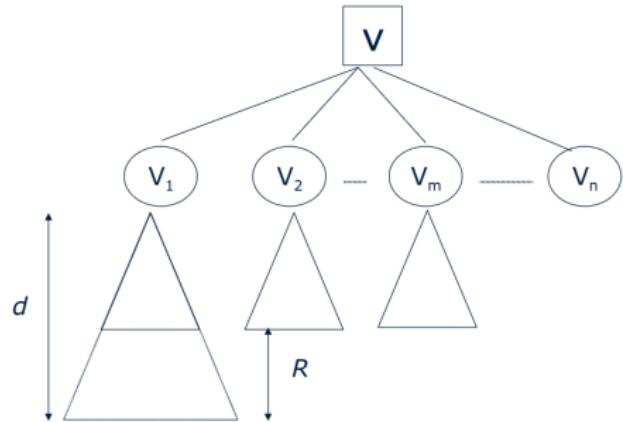
- Primary hash key of 12 bits to avoid collisions.
- Bucket replacement scheme (128-deep).
- Transposition table also used in move ordering.
- 2 killer moves per level of depth.

# Table of contents

- ① Ludeme
- ② Heuristics
- ③ Iterative deepening
- ④ Memoization: TT and killer moves
- ⑤ Variable depth search: multi-cut
- ⑥ Tuning: playing against MC-GRAVE
- ⑦ References

# Multi-cut

- Default parameter setting.
- $C = 3, M = 10, R = 2$ .
- Huge gains: one extra ply.

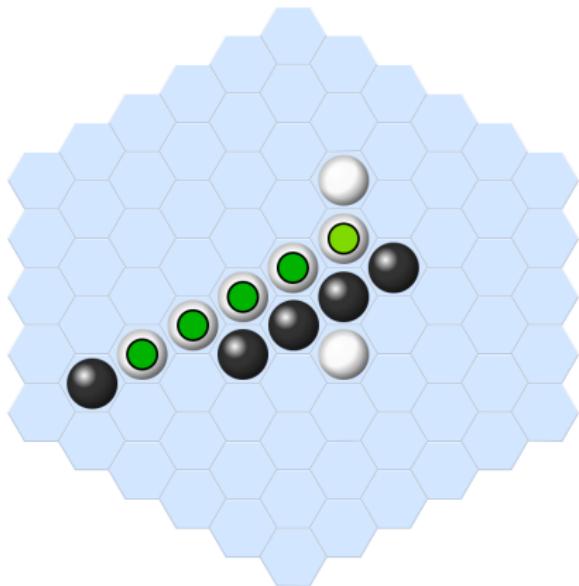


# Table of contents

- ① Ludeme
- ② Heuristics
- ③ Iterative deepening
- ④ Memoization: TT and killer moves
- ⑤ Variable depth search: multi-cut
- ⑥ Tuning: playing against MC-GRAVE
- ⑦ References

# Agent evaluation and tuning: defeating MC-GRAVE

- Evaluation by observing its behaviour against a skilled opponent.
- MC-GRAVE playing for 10 sec.
- Possible to beat as white.



# Table of contents

- ① Ludeme
- ② Heuristics
- ③ Iterative deepening
- ④ Memoization: TT and killer moves
- ⑤ Variable depth search: multi-cut
- ⑥ Tuning: playing against MC-GRAVE
- ⑦ References

# References I

-  Mark Lefler, *Chess programming wiki*, 2023.
-  É. 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 course's slides*, 2023.