GT IA JEUX – Seminar – 28/10/2021

Ludii: General Game System

future of GGP and its applications

Éric Piette

Digital
Ludeme
Project

erc
European Research Council
Established by the European Commission

Maastricht University

Digital Ludeme Project (DLP)

Five-year research project:

- Funded by the ERC (€2m)
- DKE

1. **Model**
   Full range of traditional strategy games in a single playable digital database

2. **Reconstruct**
   Missing knowledge about ancient games

3. **Map**
   Spread of games throughout history

erc European Research Council

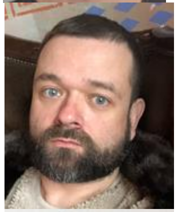
DLP Team

Cameron Browne, Principal Investigator

Éric Piette, Postdoctoral Researcher

Matthew Stephenson, Postdoctoral Researcher

Walter Crist, Postdoctoral Researcher (Cultural Stream)



Dennis J.N.J. Soemers , PhD Candidate

+ many Internships or Student Projects (Master or Bachelor)

General Game Playing (GGP)

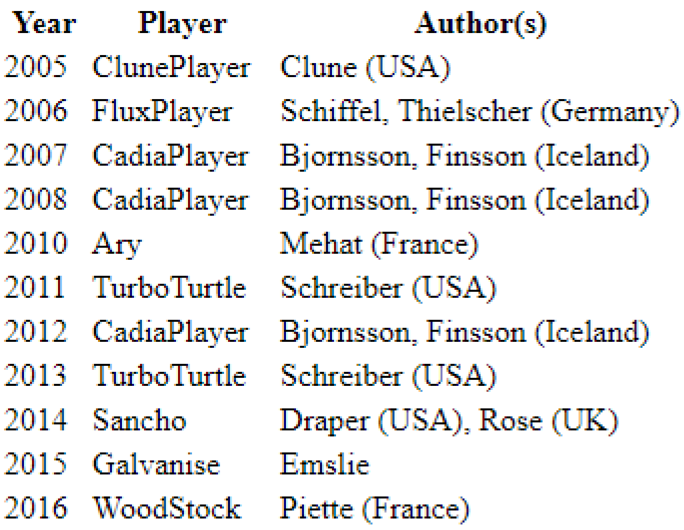
Make a General Game Player able to play to many games (1968, Pitrat)

Different languages/systems:

- METAGAMER (1992): Only for Chess variants
- Zillions of Games (1998): Proprietary System
- **GDL (2005):** Deterministic Games
- TOSS (2011): GDL variant with counters
- VGDL (2013): Realtime games (Atari-like Games)
- GDL-II (2014): Stochastic Hidden information Games
- GDL-III (2017): Epistemic Games
- RBG (2018/19): Deterministic Games

General Game Description (GDL)

- Academic standard (2005-20??)
- **N players, finite, deterministic, full information games**
- Exists for 16 years, **52 (really different) games**
- A lot of General Game players exist: **MCTS**, (S)CSP, ASP, logic programming, …
- International General Game Playing Competition (IGGPC) (2005-2016)
- Lead to a really important number of publications: http://www.general-game-playing.de/literature.html
- Complicated and Long game descriptions
- A low number of moves/playouts per second
- No automatic User Interface (no GUI)



Year	Player	Author(s)
2005	ClunePlayer	Clune (USA)
2006	FluxPlayer	Schiffel, Thielscher (Germany)
2007	CadiaPlayer	Bjornsson, Finsson (Iceland)
2008	CadiaPlayer	Bjornsson, Finsson (Iceland)
2010	Ary	Mehat (France)
2011	TurboTurtle	Schreiber (USA)
2012	CadiaPlayer	Bjornsson, Finsson (Iceland)
2013	TurboTurtle	Schreiber (USA)
2014	Sancho	Draper (USA), Rose (UK)
2015	Galvanise	Emslie
2016	WoodStock	Piette (France)

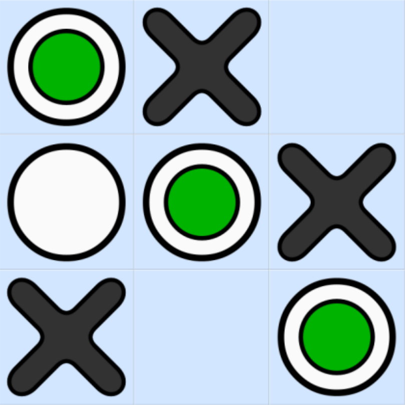
A GDL Game...

Which game is that?

```
(role white) (role black)
(init (cell 1 1 b)) (init (cell 1 2 b)) (init (cell 1 3 b))
(init (cell 2 1 b)) (init (cell 2 2 b)) (init (cell 2 3 b))
(init (cell 3 1 b)) (init (cell 3 2 b)) (init (cell 3 3 b))
(init (control white))
(<= (legal ?w (mark ?x ?y)) (true (cell ?x ?y b))
    (true (control ?w)))
(<= (legal white noop) (true (control black)))
(<= (legal black noop) (true (control white)))
(<= (next (cell ?m ?n x)) (does white (mark ?m ?n))
    (true (cell ?m ?n b)))
(<= (next (cell ?m ?n o)) (does black (mark ?m ?n))
    (true (cell ?m ?n b)))
(<= (next (cell ?m ?n ?w)) (true (cell ?m ?n ?w))
    (distinct ?w b))
(<= (next (cell ?m ?n b)) (does ?w (mark ?j ?k))
    (true (cell ?m ?n b)) (or (distinct ?m ?j)
    (distinct ?n ?k)))
(<= (next (control white)) (true (control black)))
(<= (next (control black)) (true (control white)))
(<= (row ?m ?x) (true (cell ?m 1 ?x))
    (true (cell ?m 2 ?x)) (true (cell ?m 3 ?x)))
(<= (column ?n ?x) (true (cell 1 ?n ?x))
    (true (cell 2 ?n ?x)) (true (cell 3 ?n ?x)))
(<= (diagonal ?x) (true (cell 1 1 ?x))
    (true (cell 2 2 ?x)) (true (cell 3 3 ?x)))
(<= (diagonal ?x) (true (cell 1 3 ?x))
    (true (cell 2 2 ?x)) (true (cell 3 1 ?x)))
(<= (line ?x) (row ?m ?x))
(<= (line ?x) (column ?m ?x))
(<= (line ?x) (diagonal ?x))
(<= open (true (cell ?m ?n b))) (<= (goal white 100) (line x))
(<= (goal white 50) (not open) (not (line x)) (not (line o)))
(<= (goal white 0) open (not (line x)))
(<= (goal black 100) (line o))
(<= (goal black 50) (not open) (not (line x)) (not (line o)))
(<= (goal black 0) open (not (line o)))
(<= terminal (line x))
(<= terminal (line o))
(<= terminal (not open))
```

And with Ludii ?

```
(game "Tic-Tac-Toe"
 (players 2)
 (equipment {
  (board (square 3))
  (piece "Disc" P1)
  (piece "Cross" P2)
 })
 (rules
  (play
   (move Add (to (sites Empty)))
  )
  (end
   (if (is Line 3)(result Mover Win))
  )
 )
)
```



```
(role white) (role black)
(init (cell 1 1 b)) (init (cell 1 2 b)) (init (cell 1 3 b))
(init (cell 2 1 b)) (init (cell 2 2 b)) (init (cell 2 3 b))
(init (cell 3 1 b)) (init (cell 3 2 b)) (init (cell 3 3 b))
(init (control white))
(<= (legal ?w (mark ?x ?y)) (true (cell ?x ?y b))
    (true (control ?w)))
(<= (legal white noop) (true (control black)))
(<= (legal black noop) (true (control white)))
(<= (next (cell ?m ?n x)) (does white (mark ?m ?n))
    (true (cell ?m ?n b)))
(<= (next (cell ?m ?n o)) (does black (mark ?m ?n))
    (true (cell ?m ?n b)))
(<= (next (cell ?m ?n ?w)) (true (cell ?m ?n ?w))
    (distinct ?w b))
(<= (next (cell ?m ?n b)) (does ?w (mark ?j ?k))
    (true (cell ?m ?n b)) (or (distinct ?m ?j)
    (distinct ?n ?k)))
(<= (next (control white)) (true (control black)))
(<= (next (control black)) (true (control white)))
(<= (row ?m ?x) (true (cell ?m 1 ?x))
    (true (cell ?m 2 ?x)) (true (cell ?m 3 ?x)))
(<= (column ?n ?x) (true (cell 1 ?n ?x))
    (true (cell 2 ?n ?x)) (true (cell 3 ?n ?x)))
(<= (diagonal ?x) (true (cell 1 1 ?x))
    (true (cell 2 2 ?x)) (true (cell 3 3 ?x)))
(<= (diagonal ?x) (true (cell 1 3 ?x))
    (true (cell 2 2 ?x)) (true (cell 3 1 ?x)))
(<= (line ?x) (row ?m ?x))
(<= (line ?x) (column ?m ?x))
(<= (line ?x) (diagonal ?x))
(<= open (true (cell ?m ?n b))) (<= (goal white 100) (line x))
(<= (goal white 50) (not open) (not (line x)) (not (line o)))
(<= (goal white 0) open (not (line x)))
(<= (goal black 100) (line o))
(<= (goal black 50) (not open) (not (line x)) (not (line o)))
(<= (goal black 0) open (not (line o)))
(<= terminal (line x))
(<= terminal (line o))
(<= terminal (not open))
```

Ludemes

A *ludeme* is a “game meme”

- Unit of game-information
- Building blocks or “DNA” of games

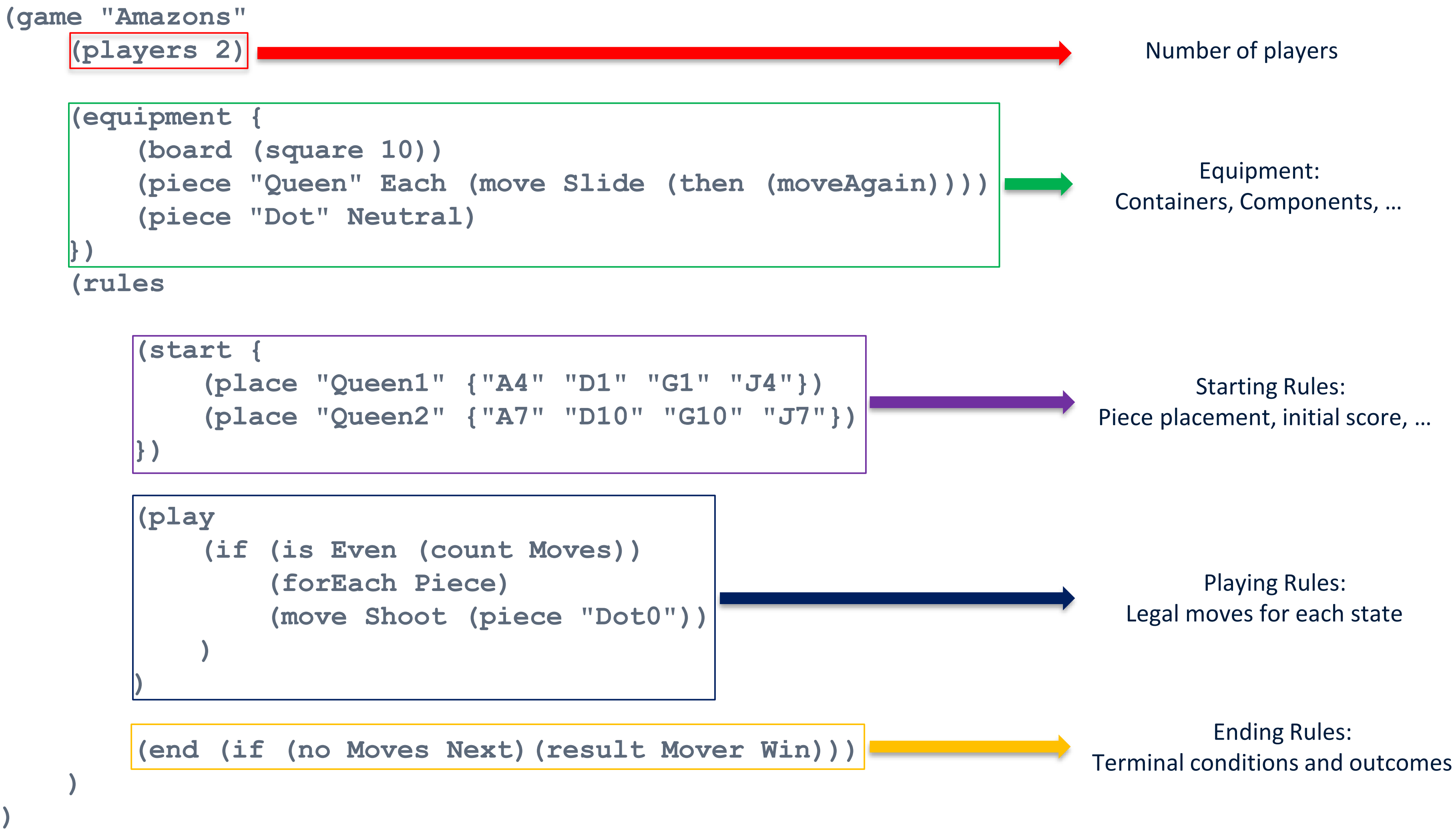
Break games down into components:

- Equipment + geometry
- Rules

```
(piece "Disc" P1)
```

```
(is Line 3)
```

Overview of a Game description



```
(game "Amazons"
    (players 2)

    (equipment {
        (board (square 10))
        (piece "Queen" Each (move Slide (then (moveAgain))))
        (piece "Dot" Neutral)
    })
    (rules

        (start {
            (place "Queen1" {"A4" "D1" "G1" "J4"})
            (place "Queen2" {"A7" "D10" "G10" "J7"})
        })

        (play
            (if (is Even (count Moves))
                (forEach Piece)
                (move Shoot (piece "Dot0"))
            )
        )

        (end (if (no Moves Next)(result Mover Win)))
    )
)
```

Number of players

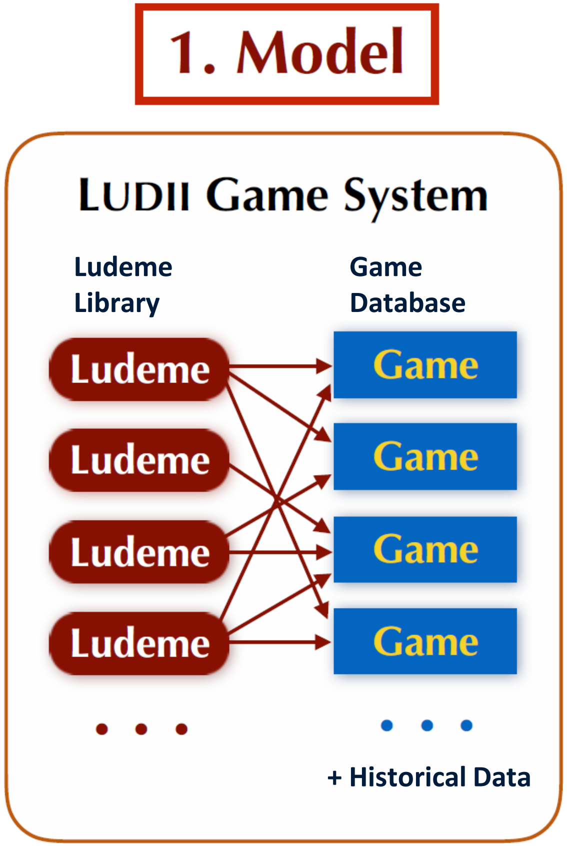
Equipment:
Containers, Components, …

Starting Rules:
Piece placement, initial score, …

Playing Rules:
Legal moves for each state

Ending Rules:
Terminal conditions and outcomes

Compilation of a game



Ludeme Library

- Each ludeme:
  - Java class
  - Meaningful name

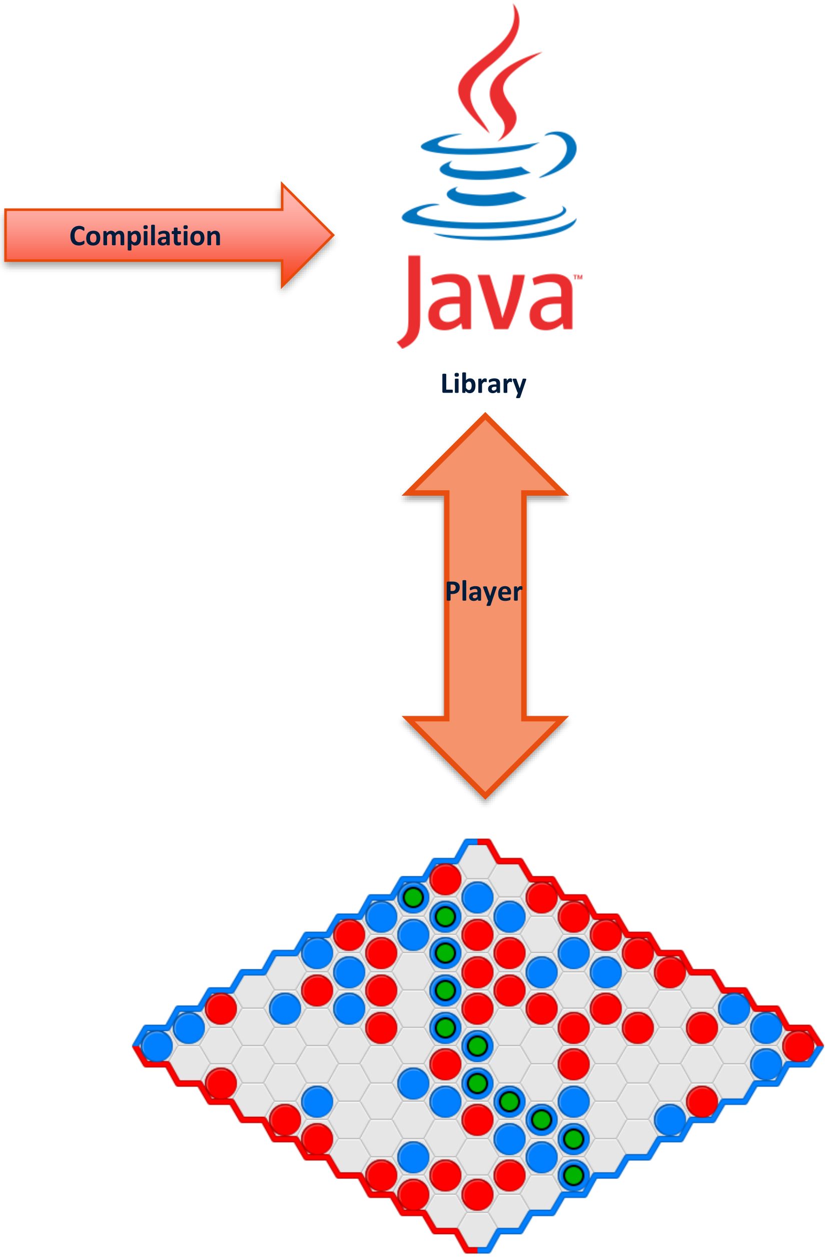
Game Database

- Each game:
  - S-expression
  - Ludeme tree
  - Compiles to bytecode
- Tagged historical data

- *Browne Cameron, A Class Grammar for General Games, CG 2016*

Ludii scheme

```
(game "Hex"
   (players 2)
   (equipment {
      (board (hex Diamond 11))
      (piece "Marker" Each)
      (regions P1 { (sites Side NE) (sites Side SW) } )
      (regions P2 { (sites Side NW) (sites Side SE) } )
   })
   (rules
      (play (move Add (to (sites Empty))) )
      (end (if (is Connected Mover) (result Mover Win)))
   )
)
```



Compilation

Library

Player

Ludii

- **General Game System** capable of modelling all the games of the DLP project (and more)

- **Granular** description of the games in **ludemes.** Clear, simple and short

- **Currently, between 2 to 10 times faster than GDL**

- **Can model**:
  - Deterministic / Stochastic
  - Complete / Hidden Info.
  - Puzzles / Multi-Player Games
  - Alternating / Simultaneous
  - Borderless / Any geometry
  - Math, Race, Sow, Space, War Games, …
  - Stacking games, large pieces, …
- **All playable (humanly) thanks to the GUI and remotely through our game server**

- **Available at: ludii.games**

- **Open source: github.com/Ludeme/Ludii**

- **Paper to Cite:** ***Piette, É. et al., Ludii – The Ludemic General Game System, ECAI 2020***

Ludii Library

1000+ games, 1200+ rulesets, ~2,000,000 option combinations

AI techniques in Ludii

- Random
- Flat Monte-Carlo
- Alpha-Beta (Max^N, Paranoid, BRS+)
- MCTS techniques (Selection, Expansion, Playout, Backpropagation)

Selection

UCB1
UCB1-GRAVE
UCB1 Tuned
MC-GRAVE (Generalized RAVE)
Exit (Expert Iteration)
Progressive Bias
Progressive History

Playout

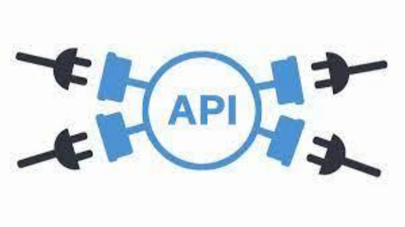
Random
MAST (Move-average)
NST (N-gram)

Backpropagation

Monte Carlo
AlphaGo
Qualitative Bonus
Heuristic

- Score-Bounded MCTS
- Open-loop MCTS

Make your own AI!



- For **implementing Java/Python-based AIs** that can play Ludii games:
  - github.com/Ludeme/LudiiExampleAI
  - github.com/Ludeme/LudiiPythonAI

- Join the **competition**
  - github.com/Ludeme/LudiiAICompetition

Results
&
Works in Progress

Strategies

Real-World Board Game Knowledge

- Agents can use **Heuristics** (~15 availables)
  Examples:
  - Score
  - Distance proximities (centre, corners, specific regions, …)
  - Lines/connections of pieces
  - Material

  → **General/Global Strategies**

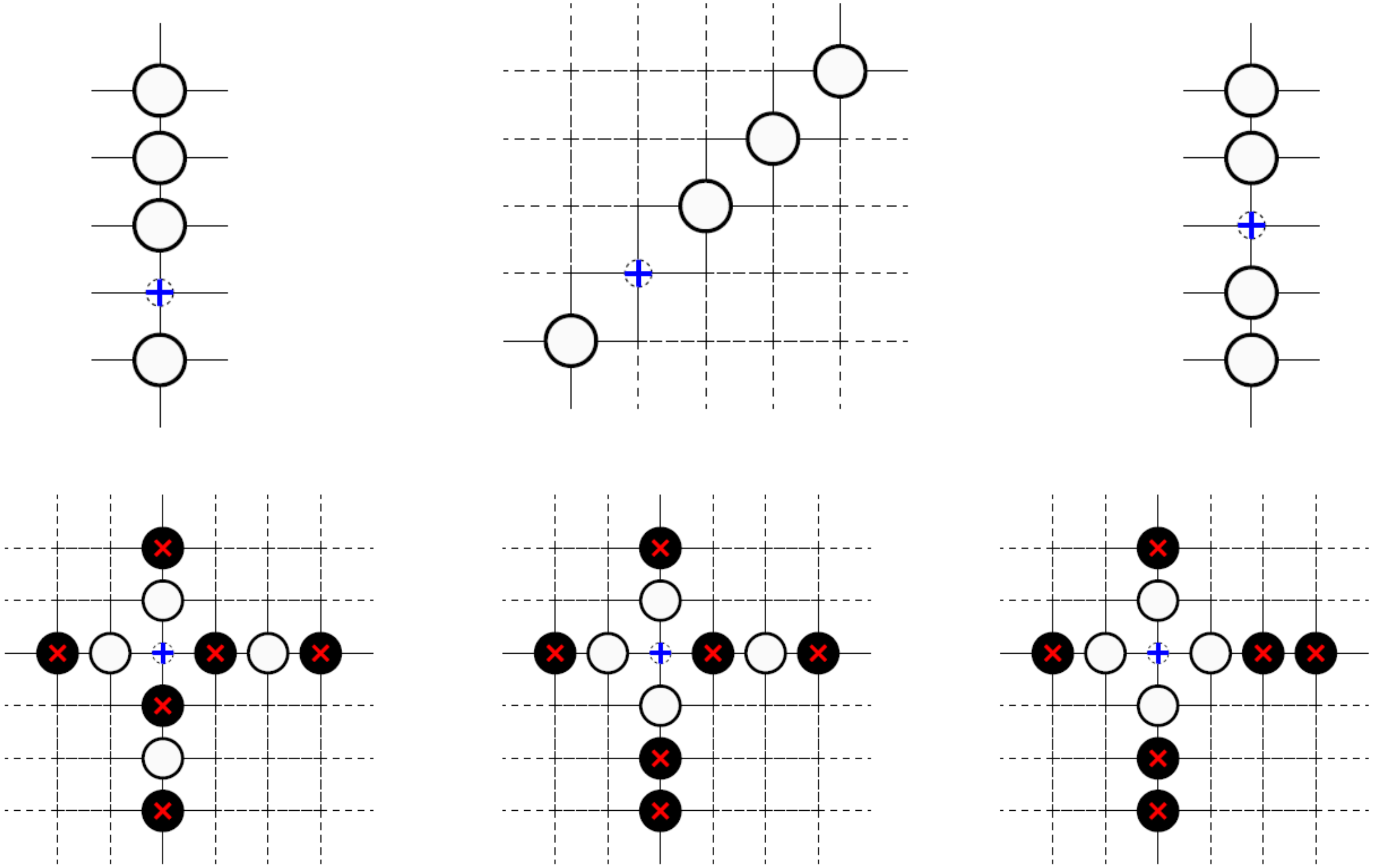
- However, spatial semantics (patterns) matter too
  In others words → **Local Strategies** (also called tactical)

→ **Spatial State-Action Features** (Dennis Soemers PhD 2018-2022)

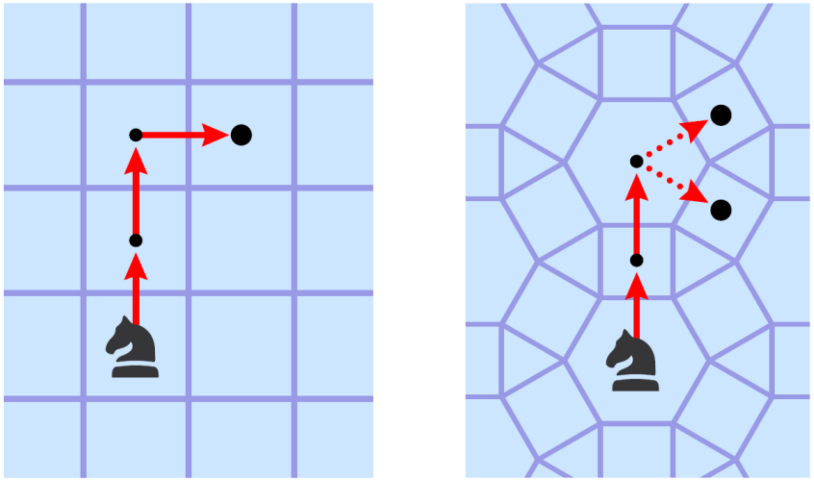
Some papers:

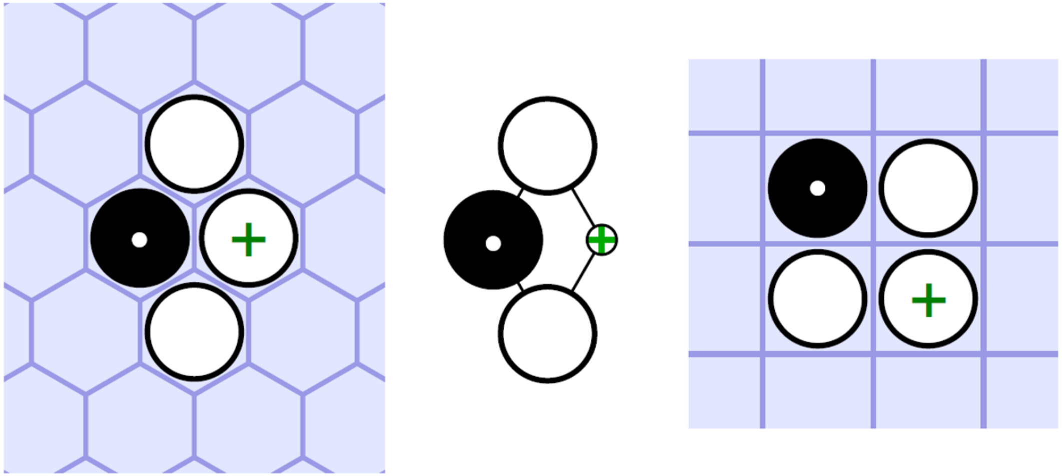
- Soemers et al., **Spatial State-Action Features for General Games**, JAIR (to appear 2022)
- Soemers et al., **Manipulating the Distributions of Experience used for Self-Play Learning in Expert Iteration**, COG 2020
- Soemers et al., ***Learning Policies from Self-Play with Policy Gradients and MCTS value Estimates**, COG 2019*
- *Soemers, Piette and Browne, **Biasing MCTS with Features for General Games**, CEC 2019*
- *Browne, Soemers and Piette, **Strategic features for General Games**, KEG@AAAI 2018*

Spatial State-Action Features



Spatial State-Action Features





Feature Learning

- Start with atomic features
  - Features with only one “test”

- Combine pairs of existing features into new features such that:

  - Correlation with objective function of policy training is maximised
    - → candidate feature is useful for improving policy

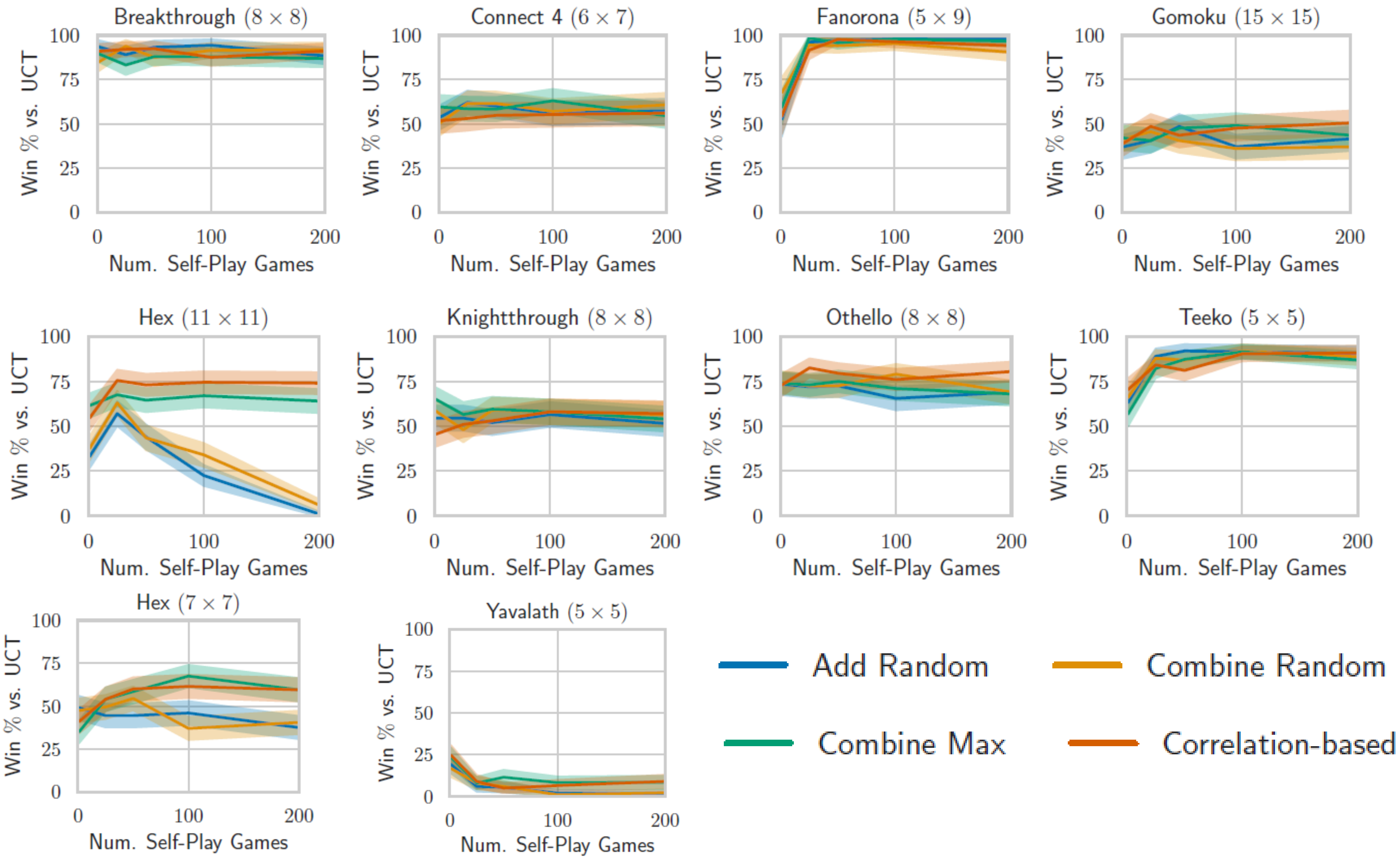
  - Correlation with activity of either constituent is minimised
    - → candidate feature provides new information

Experiments

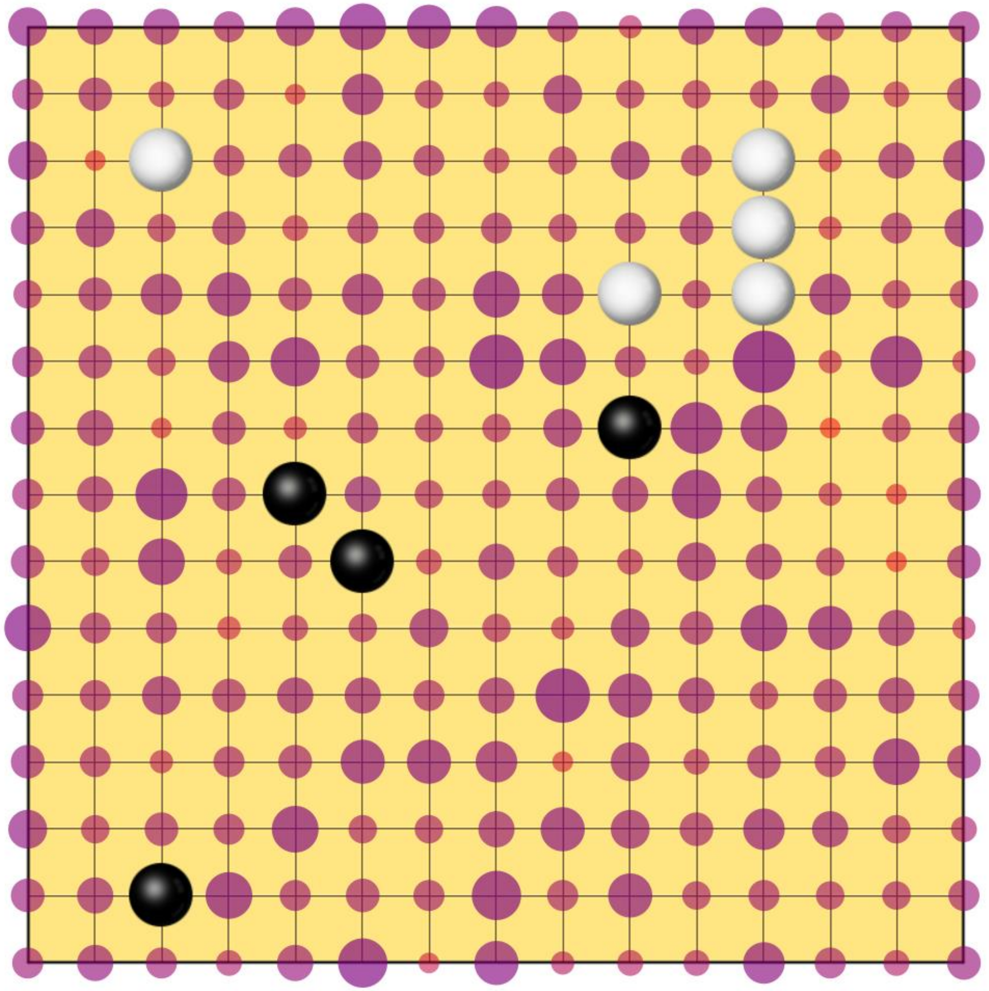
- 200 games of self-play training
  - No, not 200 million!

- One policy update after every move
- One new feature after every game
- 5 sec. pondering time per move
- Evaluation vs. plain UCT

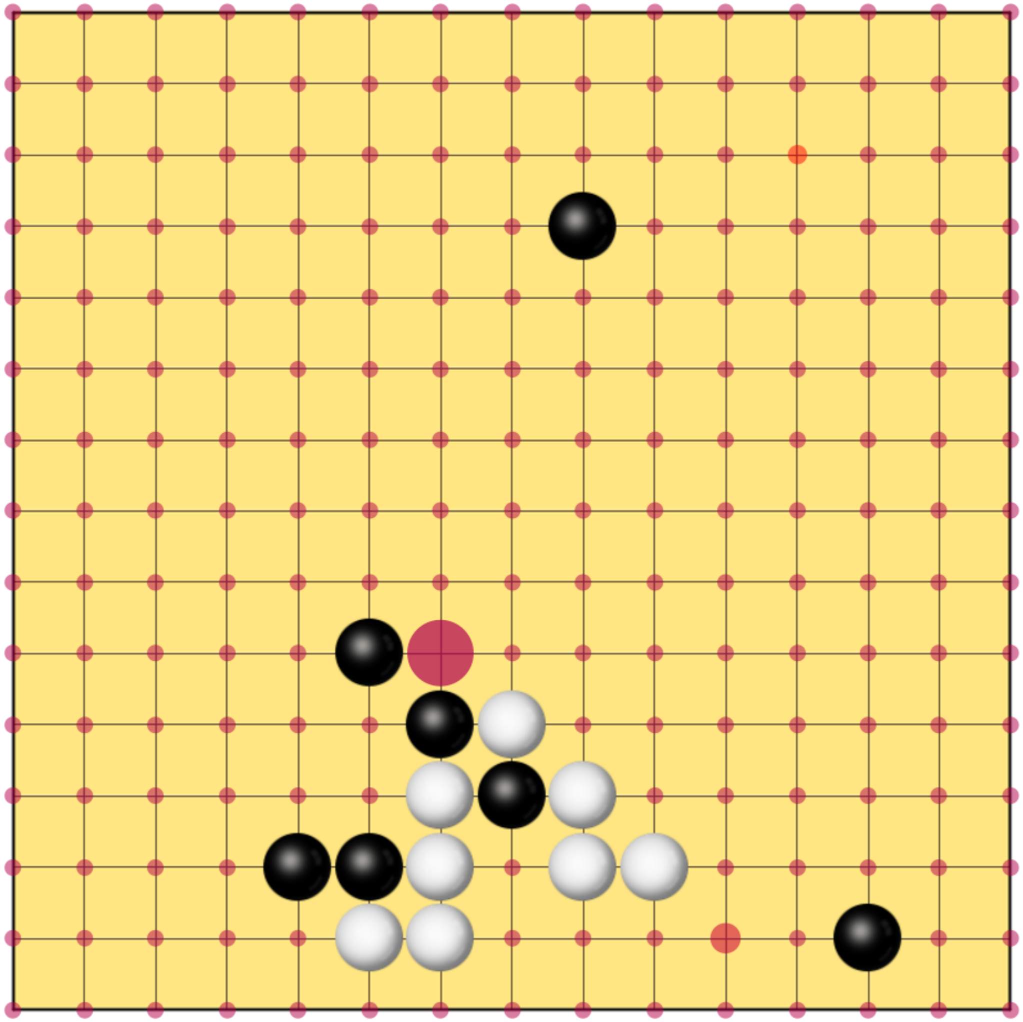
Results


Breakthrough (8 × 8)
Connect 4 (6 × 7)
Fanorona (5 × 9)
Gomoku (15 × 15)
Hex (11 × 11)
Knightthrough (8 × 8)
Othello (8 × 8)
Teeko (5 × 5)
Hex (7 × 7)
Yavalath (5 × 5)
Win % vs. UCT
Num. Self-Play Games
0
25
50
75
100
0
100
200
Add Random
Combine Random
Combine Max
Correlation-based

AI (UCT) without features in Gomoku



AI (Biased MCTS) with features in Gomoku



Transfer learning

Transfer of Fully-Convolutional Networks

Transfer Learning:

- Train in *source domain*
- Evaluate in *target domain*
  - Against opponent trained in target domain

Two ways:

- **Zero-shot**: no training at all in target domain
- **Fine-tuning**: also train in target domain after source domain

Transfer of Fully-Convolutional Networks

- **Inputs (states)** and **outputs (actions)** have:
  - Two spatial dimensions
  - One non-spatial dimension (channels in CNNs)

- **How to transfer?**
  - Fully-convolutional architectures handle spatial dimensions
  - Identify equivalence relations between channels based on Ludii's consistent state/action representations

Transfer Results

- Small boards → Big boards
  - Very strong results, even zero-shot

- Big boards → Small boards, Square <-> Hexagonal Boards
  - Can help with fine-tuning

- Transfer between many different line-completion games
  - Some good zero-shot transfer to large games
  - Often helpful with fine-tuning

- Transfer between several Shogi variants
  - Primarily helpful from small games to large games
  - **97% wins** with **zero-shot** transfer Minishogi → Shogi!

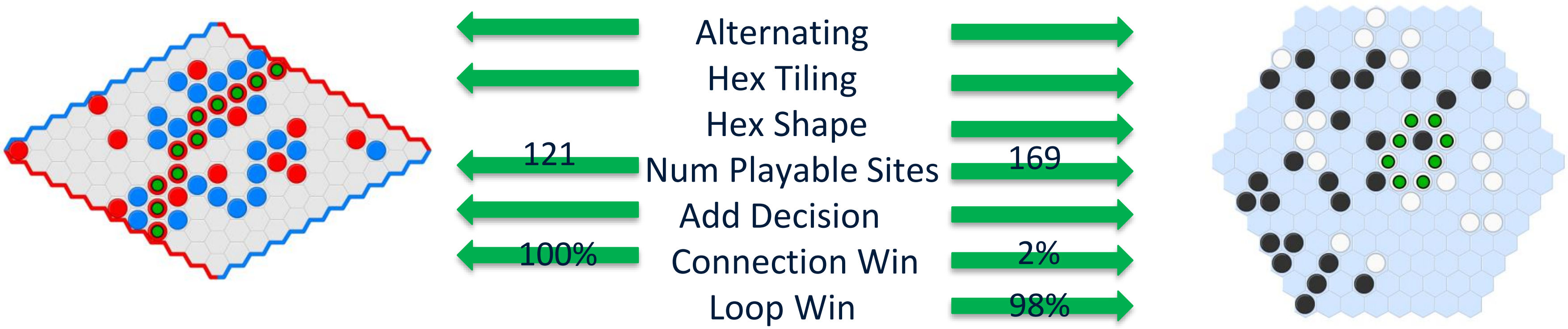
- **Soemers, et al., *Transfer of Fully Convolutional Policy-Value Networks Between Games and Game Variants, (Maybe AAAI 2022, in review)***

Game Concepts

Board Game Concepts

- A **game concept** = Feature expressed in-game terms commonly used by game players and designers which can be associated with the game or an element of play

- Examples:



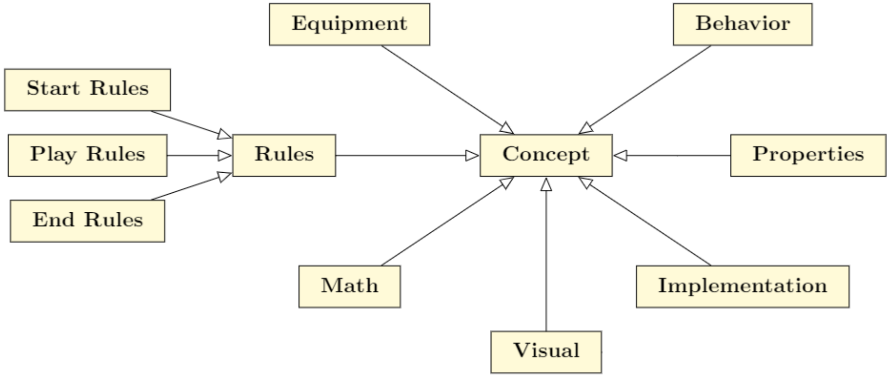
Hex		Havannah
←	Alternating	→
←	Hex Tiling	→
	Hex Shape	→
← 121	Num Playable Sites	169 →
←	Add Decision	→
← 100%	Connection Win	2% →
	Loop Win	98% →

Hex

Havannah

- In General Game Playing (board games), the concepts **are under-exploited**

Game Concepts in Ludii



- **Properties**: Related to the format of the game
- **Equipment**: Related to the board and pieces
- **Rules**: Related to each rule (start, play, end)
- **Math**: Related to fields of Mathematics
- **Behavior**: Related to well-known game metrics
- **Visual**: Related to the game's graphical style
- **Implementation**: Related to implementation details

To organise the concepts, a taxonomy is proposed:
ludii.games/searchConcepts.php

- *Piette, É. et al., General Board Game Concepts, COG 2021*

Compilation concepts

- **Compilation concepts** = All the **binary concepts** and **static numerical concepts**.

- **Binary concept** = **existence of specific ludemes or combinations of ludemes**
  Examples:
  - **Stochastic** concept can be activated by multiple concepts in isolation such as (roll) or (value Random …)
  - **Hop Capture** is triggered if a (move Hop …) and a capturing effect within it, e.g. (remove …), is used.

- **Static numerical concept** = the value set during the compilation.
  Examples:
  - **Num Component Types** = The size of the list of component types.
  - **Num Playable Sites** = The number of graph elements used within the game.

- The computation of these concepts is integrated to the compilation of the game and takes a **few ms**.

Associated to several levels of abstraction

- A **state** i is noted s_i with initial state = s_0 and terminal state = s_{ter}
- An **atomic action** is noted a
- A **move** i is noted M_i is defined by n actions such as M_i: <a_{i0}, …., a_{ij}, …., a_{in}>
- A **trial** is noted τ: <s_0, $M_{0,}$ …, s_i, M_i, …, $M_{(ter-1)}$, s_{ter}>

- The compilation concepts **can be associated with any level of abstraction**:

- **Action concepts**, examples:
  - ActionMove generating from (move Slide …) → Slide Decision.

- **Move concepts**
  - The concepts of a move M are the concepts of all the atomic actions composing it.

- **State concepts**
  - The concepts of a state s are the concepts of each element composing it (legal moves, pieces, board, …)

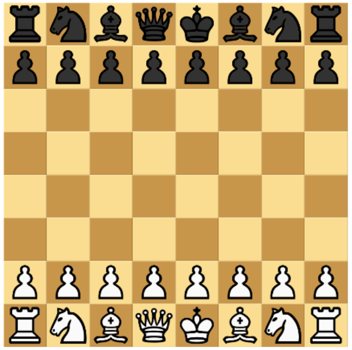
- **Trial concepts**
  - The concepts of a trial are the concepts of each state and each move composing it.

Playout concepts

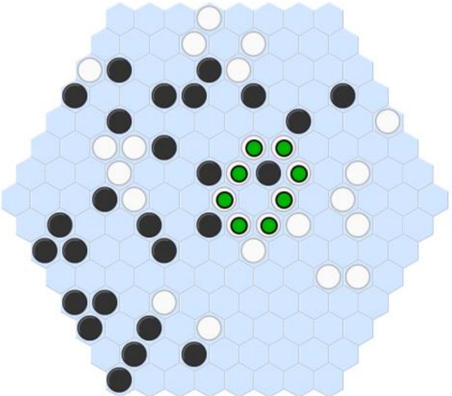
- **Playout concepts:** At a game level, concepts requiring playouts.

- **Frequency concepts**: Rate per playout to be activated.
- **Behavior concepts**: Game Metrics (Branching Factor, Game Length)

Examples with UCT (1s per move) playouts:



Capture Frequency	11%
Checkmate Frequency	90%
Draw Frequency	10%
Slide Frequency	51%
Leap Frequency	15%
Step Frequency	34%
Branching Factor	28.31
Game Length	158.8



Add Frequency	100%
Connection Win	2%
Loop Win	98%
Branching Factor	145.6
Game Length	47.8

- **Important distinction** depending on the intended application. Tasks that have to be responsive cannot use the concepts requiring playouts, as these take longer to compute.

Data Mining

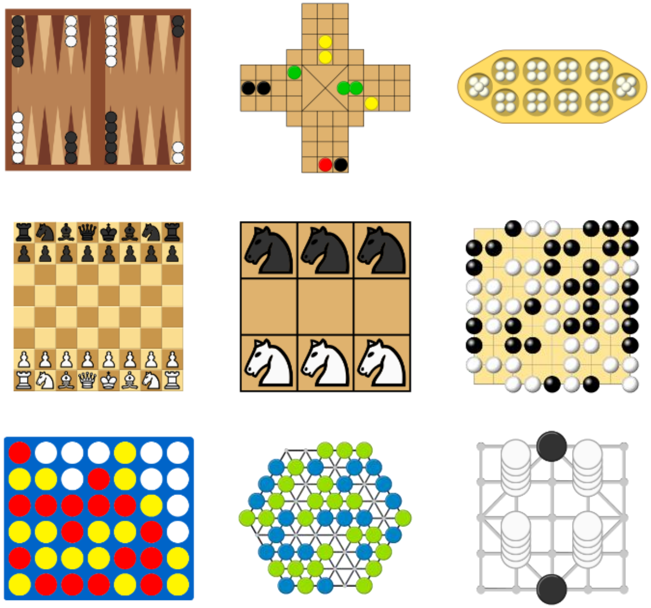
Data

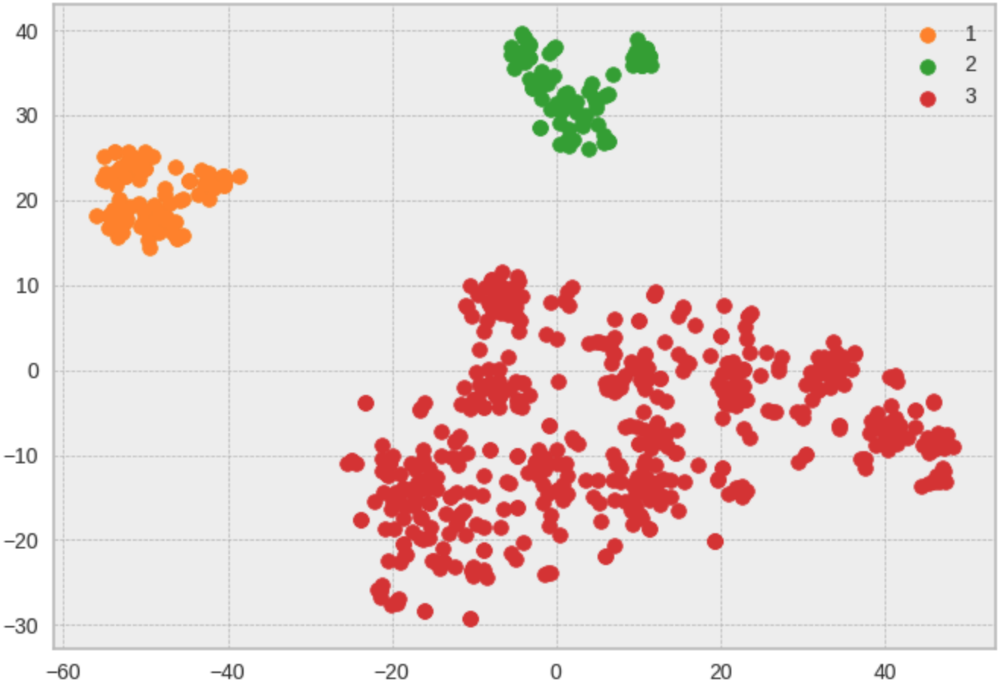
Lots of different datasets available:

- **Historical information** (period, location, culture)
- **Game concepts** (equipment, rules, visuals, math)
- **AI performance** (agents, heuristics, features)
- **Game Metrics** (bias, length, complexity)
- **User data** (most played game, online ratings)

Ludii Game Clusters

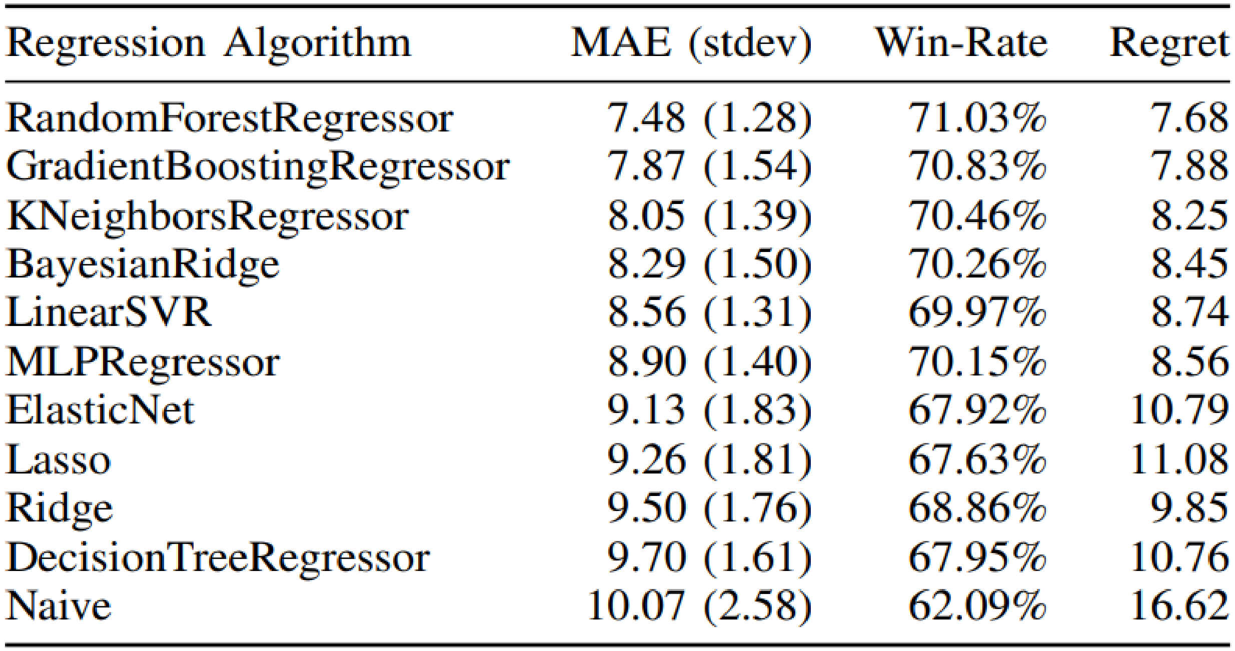
Group similar games together based on their concepts.





AI Performance Prediction

Use these clusters to predict how AI agents or heuristics will perform



Regression Algorithm	MAE (stdev)	Win-Rate	Regret
RandomForestRegressor	7.48 (1.28)	71.03%	7.68
GradientBoostingRegressor	7.87 (1.54)	70.83%	7.88
KNeighborsRegressor	8.05 (1.39)	70.46%	8.25
BayesianRidge	8.29 (1.50)	70.26%	8.45
LinearSVR	8.56 (1.31)	69.97%	8.74
MLPRegressor	8.90 (1.40)	70.15%	8.56
ElasticNet	9.13 (1.83)	67.92%	10.79
Lasso	9.26 (1.81)	67.63%	11.08
Ridge	9.50 (1.76)	68.86%	9.85
DecisionTreeRegressor	9.70 (1.61)	67.95%	10.76
Naive	10.07 (2.58)	62.09%	16.62

- *Stephenson, M. et al., General Game Heuristic Predictions Based on Ludeme Descriptions, COG 2021*

Game Distance

Measure the distance between two games

- How close are the game's cultures?
- Do they have similar rules and mechanisms?
- Are the same playing strategies effective?
- Did one of these games influence the other?

Use this distance measure to help with game reconstruction

Digital Archaeoludology

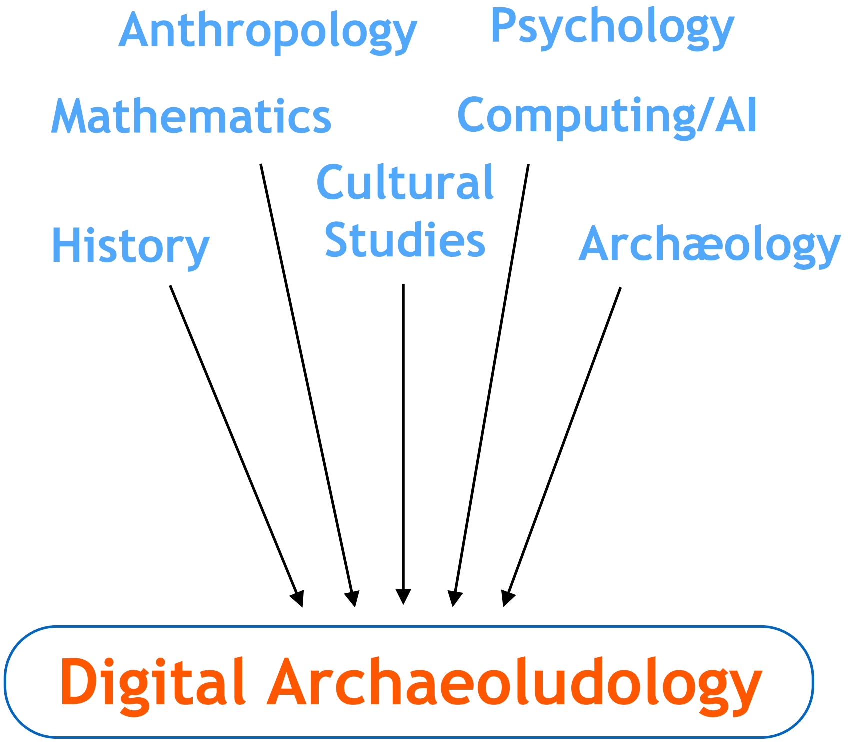
Digital Archaeoludology

New research field:

- to improve our understanding of ancient games
- by applying modern computational analysis
- to the available (partial) evidence

Bridge between:

- Traditional game studies
- Current game AI research



Maastricht University

Game Reconstruction

- Games which are closer are more likely to share concepts.
  - Equipment
  - Rules
  - Gameplay

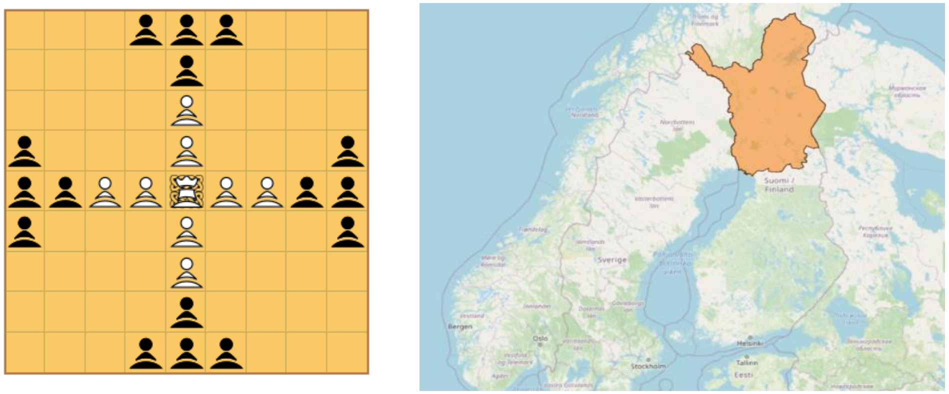
- When reconstructing games, we can take inspiration from other nearby games.

Example

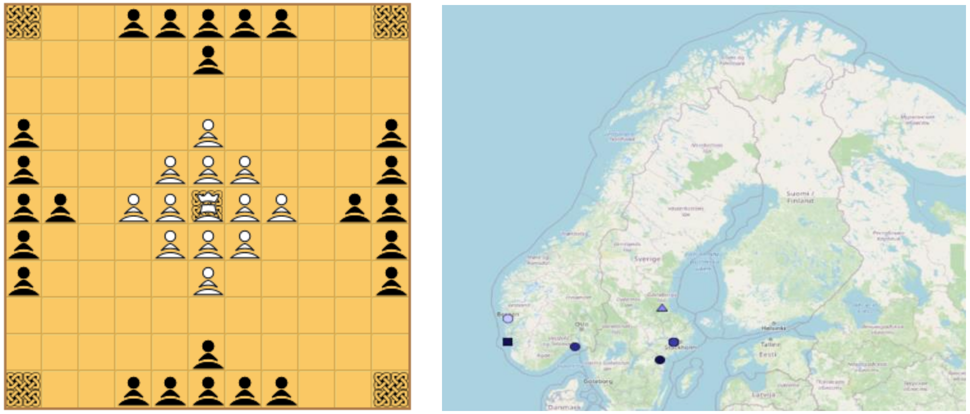
- Use rules of Tablut as a basis for reconstructing other Tafl games.

- Similar time and place.
- Similar board design.

- Probably similar rules.



Tablut (1732CE)

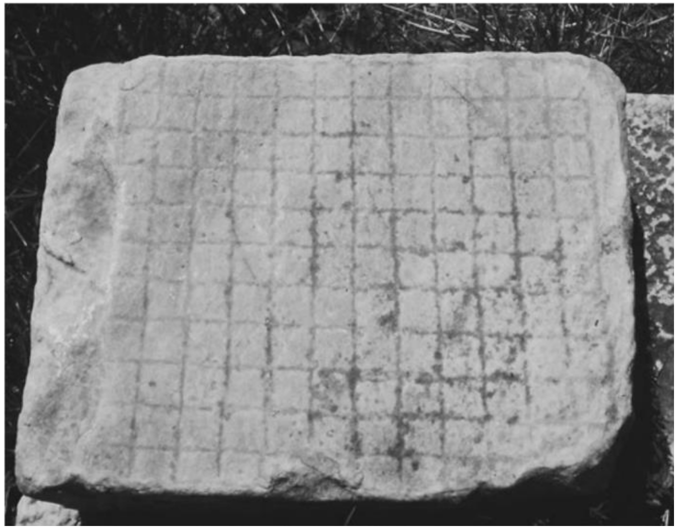


Hnefatafl (790 - 1413CE)

Case Study: Ludus Latrunculorum

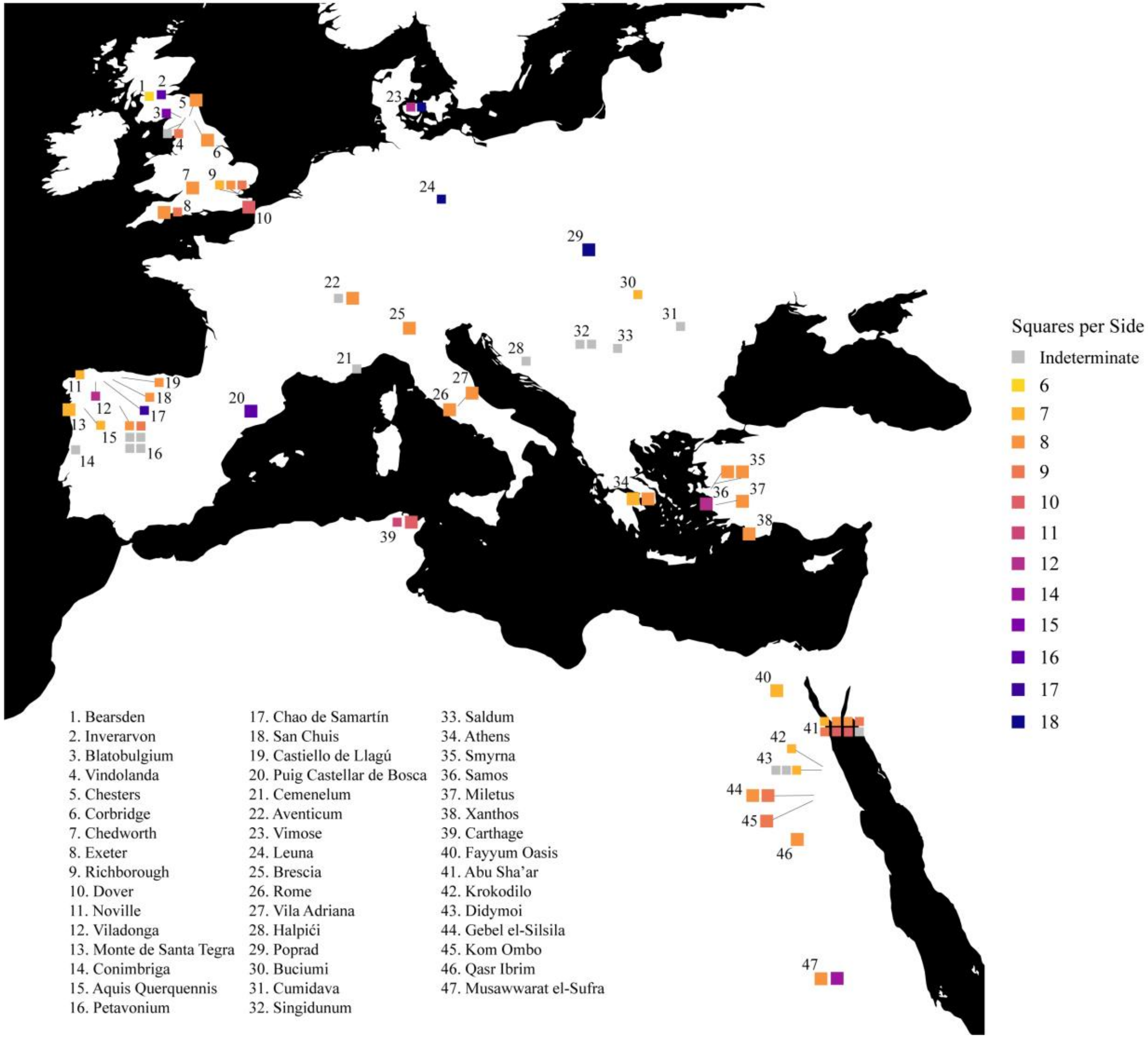
- Four rules known: Square or rectangular board
- Orthogonal Moves
- Custodial Capture
- Placement Phase

6x7, Castro de Santa Tegra, Spain



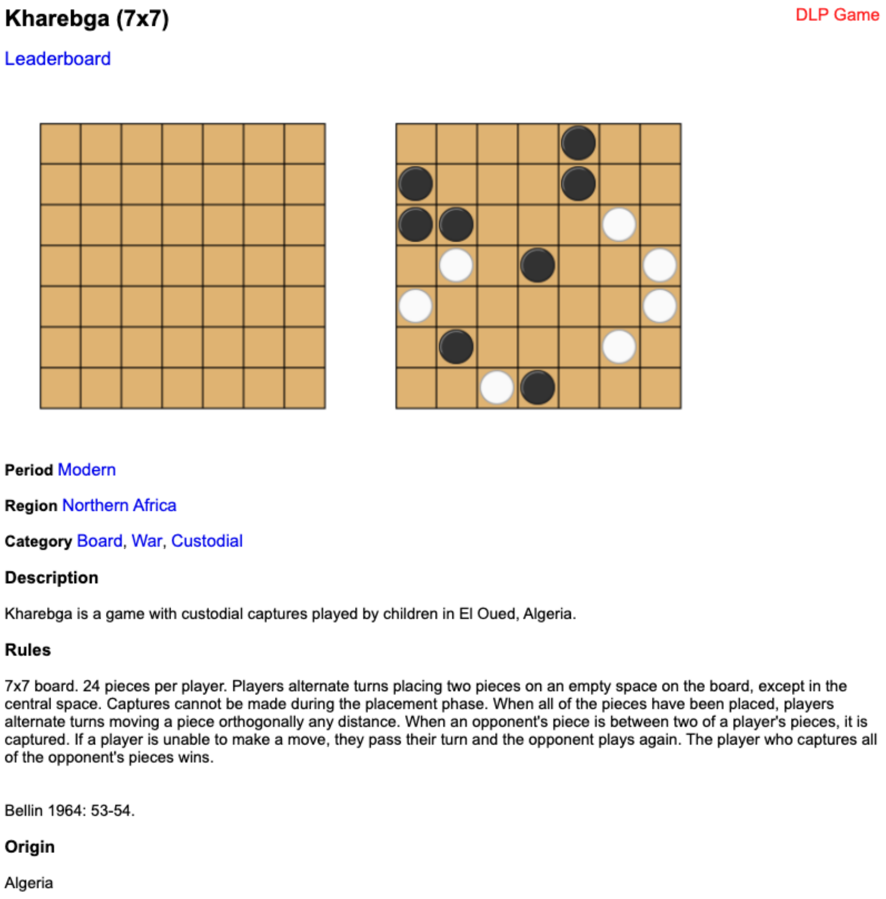
11x12, Temple of Hera, Samos, Greece

Quadrangular boards in and around the Roman Empire



1. Bearsden
2. Inverarvon
3. Blatobulgium
4. Vindolanda
5. Chesters
6. Corbridge
7. Chedworth
8. Exeter
9. Richborough
10. Dover
11. Noville
12. Viladonga
13. Monte de Santa Tegra
14. Conimbriga
15. Aquis Querquennis
16. Petavonium
17. Chao de Samartín
18. San Chuis
19. Castiello de Llagú
20. Puig Castellar de Bosca
21. Cemenelum
22. Aventicum
23. Vimose
24. Leuna
25. Brescia
26. Rome
27. Vila Adriana
28. Halpići
29. Poprad
30. Buciumi
31. Cumidava
32. Singidunum
33. Saldum
34. Athens
35. Smyrna
36. Samos
37. Miletus
38. Xanthos
39. Carthage
40. Fayyum Oasis
41. Abu Sha'ar
42. Krokodilo
43. Didymoi
44. Gebel el-Silsila
45. Kom Ombo
46. Qasr Ibrim
47. Musawwarat el-Sufra

Candidate Rulesets



Kharebga (7x7)

DLP Game

Leaderboard

Period Modern

Region Northern Africa

Category Board, War, Custodial

Description

Kharebga is a game with custodial captures played by children in El Oued, Algeria.

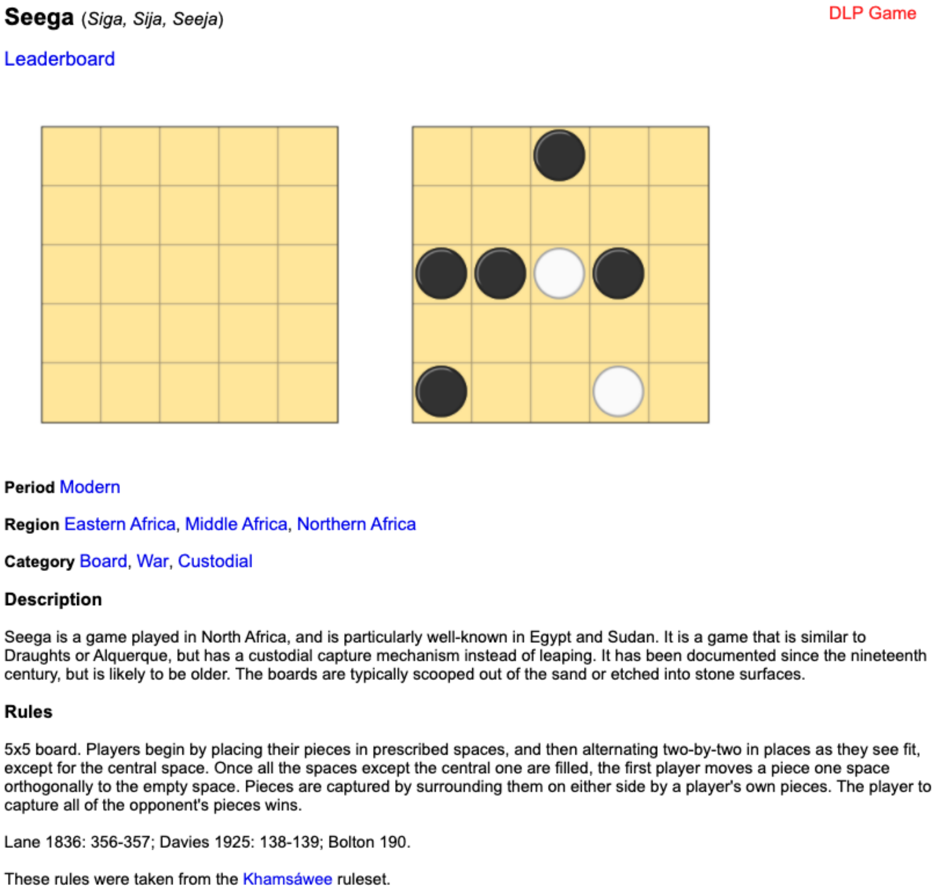
Rules

7x7 board. 24 pieces per player. Players alternate turns placing two pieces on an empty space on the board, except in the central space. Captures cannot be made during the placement phase. When all of the pieces have been placed, players alternate turns moving a piece orthogonally any distance. When an opponent's piece is between two of a player's pieces, it is captured. If a player is unable to make a move, they pass their turn and the opponent plays again. The player who captures all of the opponent's pieces wins.

Bellin 1964: 53-54.

Origin

Algeria



Seega (*Siga, Sija, Seeja*)

DLP Game

Leaderboard

Period Modern

Region Eastern Africa, Middle Africa, Northern Africa

Category Board, War, Custodial

Description

Seega is a game played in North Africa, and is particularly well-known in Egypt and Sudan. It is a game that is similar to Draughts or Alquerque, but has a custodial capture mechanism instead of leaping. It has been documented since the nineteenth century, but is likely to be older. The boards are typically scooped out of the sand or etched into stone surfaces.

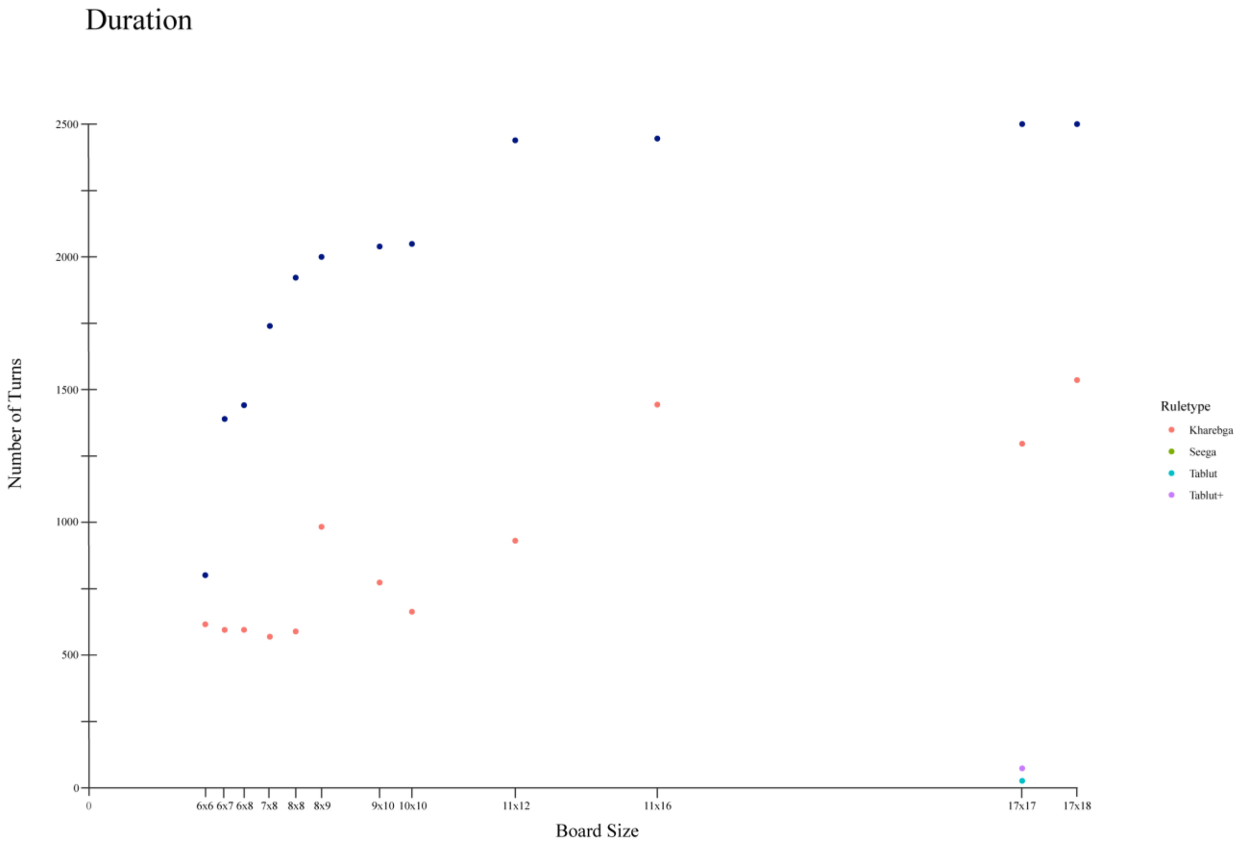
Rules

5x5 board. Players begin by placing their pieces in prescribed spaces, and then alternating two-by-two in places as they see fit, except for the central space. Once all the spaces except the central one are filled, the first player moves a piece one space orthogonally to the empty space. Pieces are captured by surrounding them on either side by a player's own pieces. The player to capture all of the opponent's pieces wins.

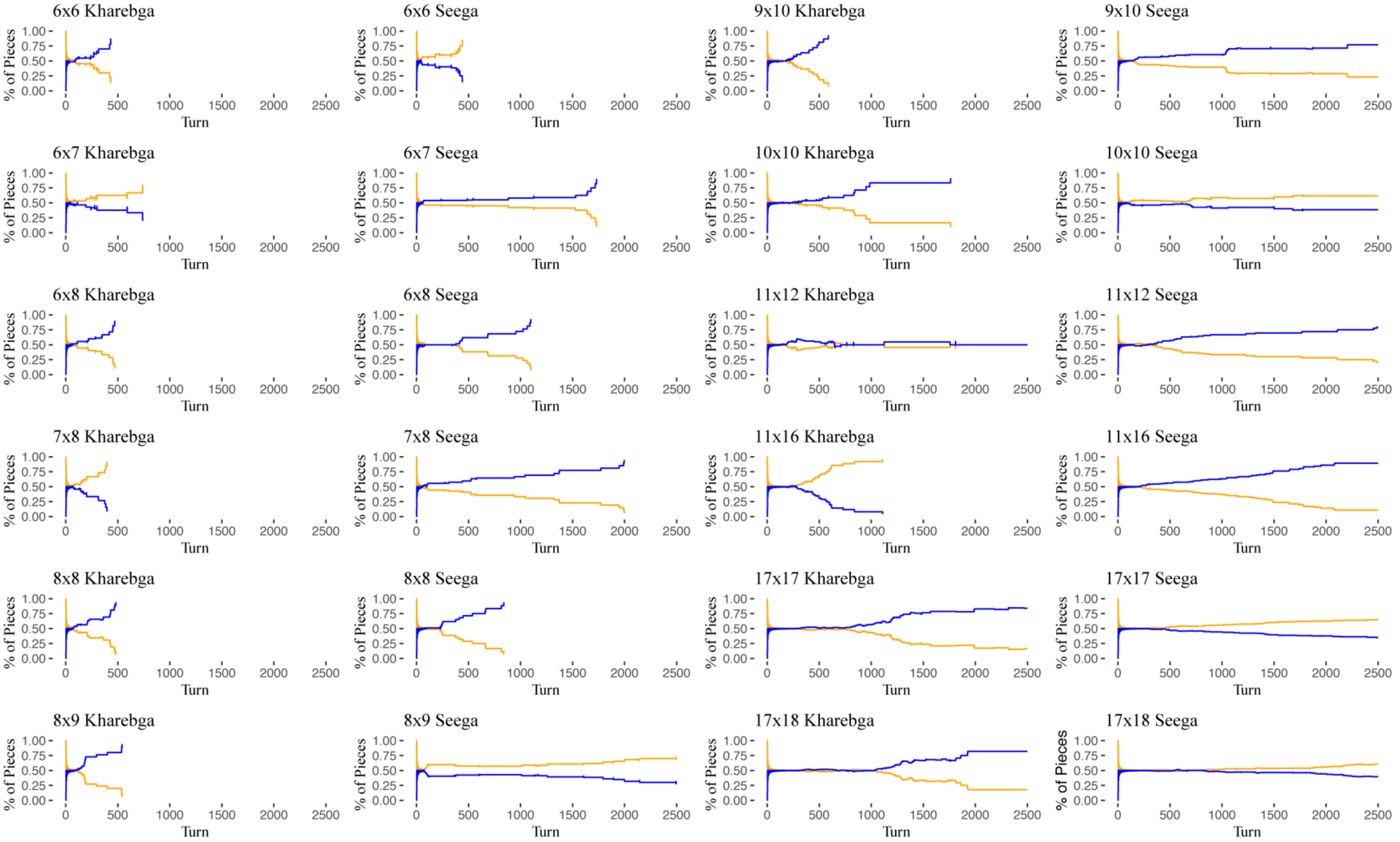
Lane 1836: 356-357; Davies 1925: 138-139; Bolton 190.

These rules were taken from the Khamsáwee ruleset.

Length of Games



Percentage of Pieces on the Board per Player



6x6 Kharebga

6x6 Seega

9x10 Kharebga

9x10 Seega

6x7 Kharebga

6x7 Seega

10x10 Kharebga

10x10 Seega

6x8 Kharebga

6x8 Seega

11x12 Kharebga

11x12 Seega

7x8 Kharebga

7x8 Seega

11x16 Kharebga

11x16 Seega

8x8 Kharebga

8x8 Seega

17x17 Kharebga

17x17 Seega

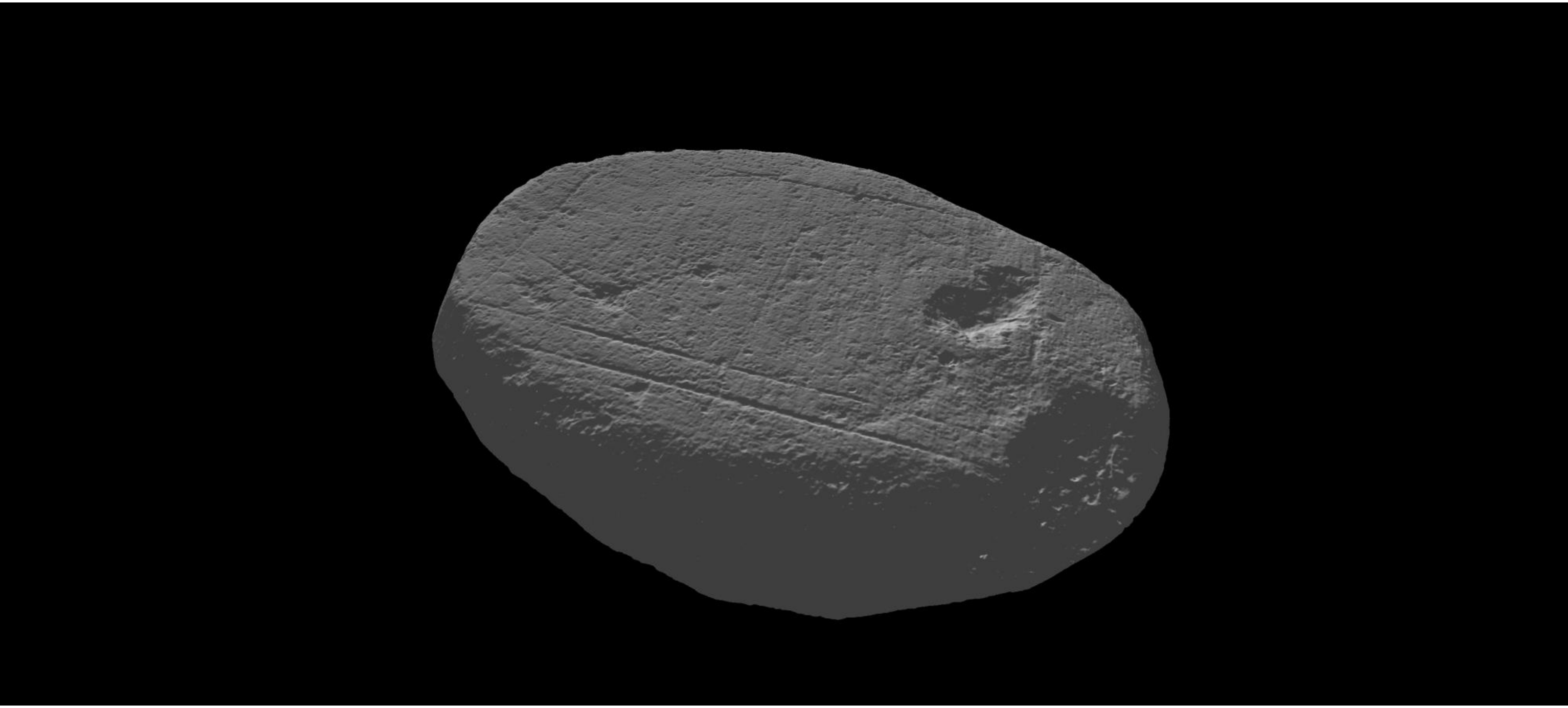
8x9 Kharebga

8x9 Seega

17x18 Kharebga

17x18 Seega

“Thermenmuseum Game”



Automatic Generation of Board Game Manuals

Maastricht University

Automatic Generation of Board Game Manuals

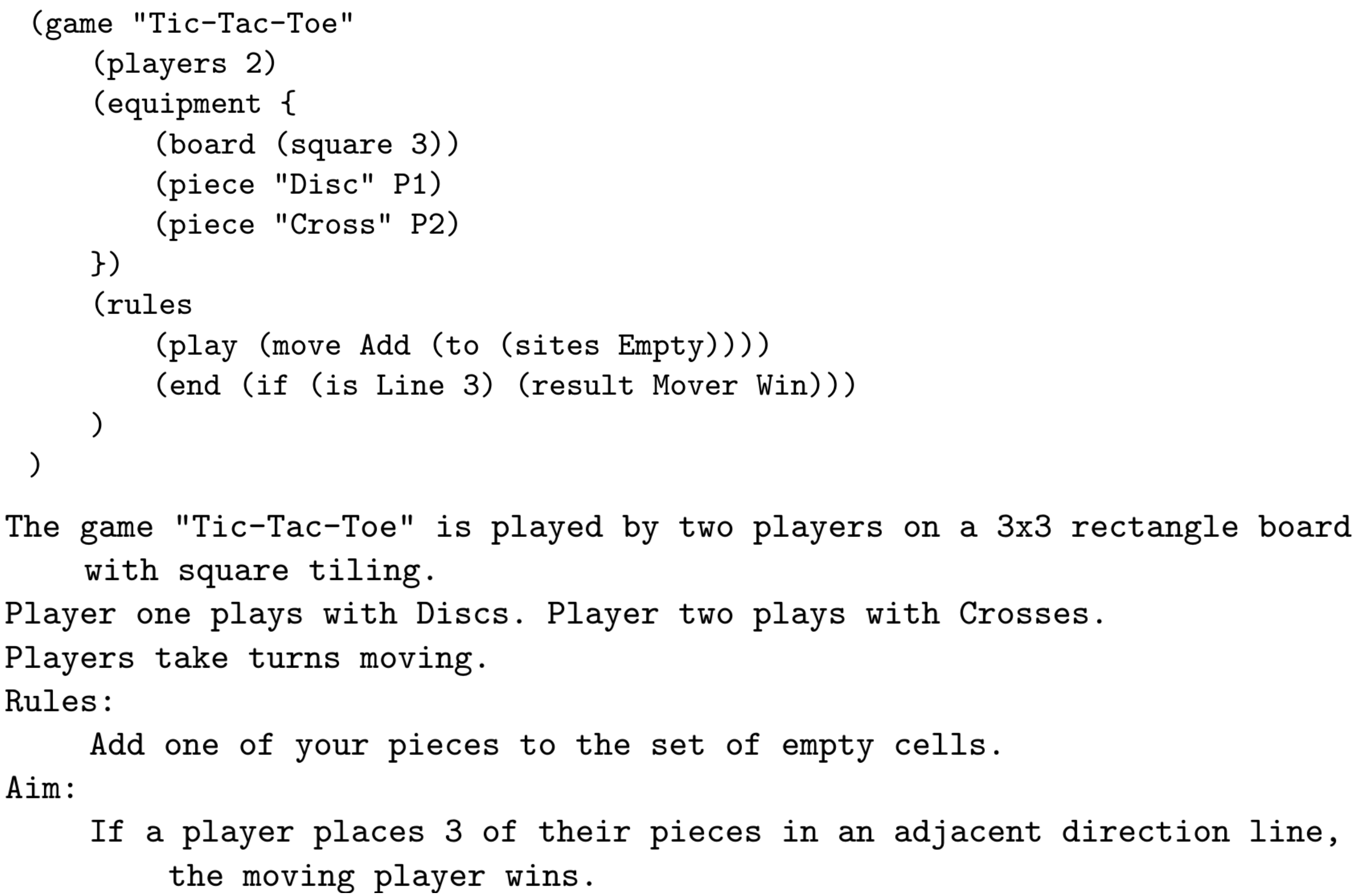
Generate manuals for Ludii games.

- English rule translations
- Initial board setup
- Move visualisations
- Winning / losing moves
- Strategy Explanation

English rule translations

Convert Ludii game descriptions into plain English.

Example: Tic-Tac-Toe



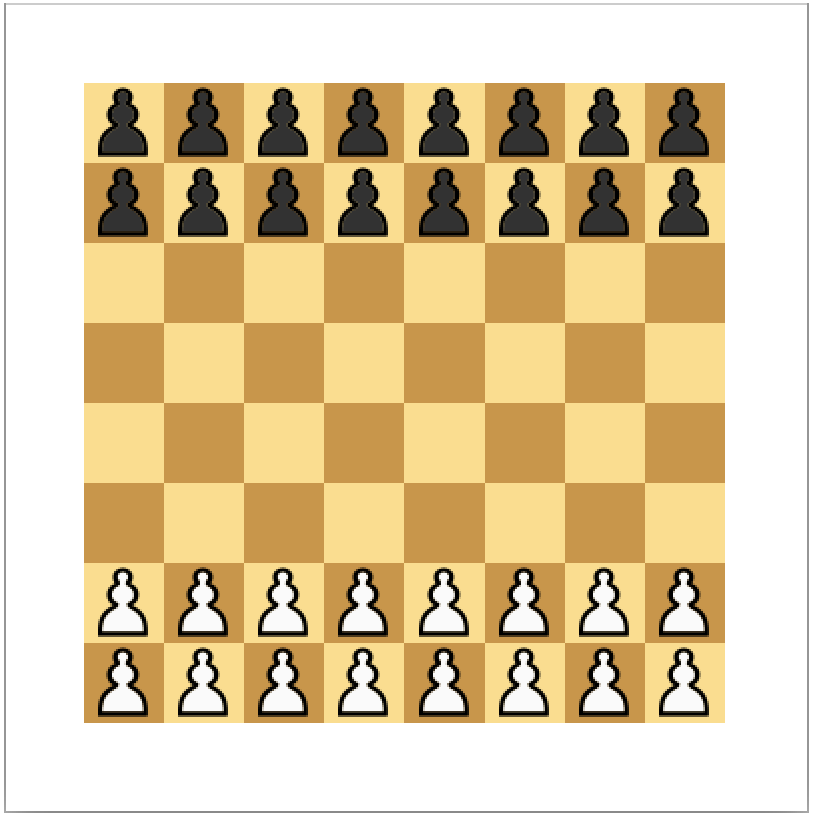
```
(game "Tic-Tac-Toe"
    (players 2)
    (equipment {
        (board (square 3))
        (piece "Disc" P1)
        (piece "Cross" P2)
    })
    (rules
        (play (move Add (to (sites Empty))))
        (end (if (is Line 3) (result Mover Win)))
    )
)
```

```
The game "Tic-Tac-Toe" is played by two players on a 3x3 rectangle board
    with square tiling.
Player one plays with Discs. Player two plays with Crosses.
Players take turns moving.
Rules:
    Add one of your pieces to the set of empty cells.
Aim:
    If a player places 3 of their pieces in an adjacent direction line,
        the moving player wins.
```

English rule translations

Initial board setup

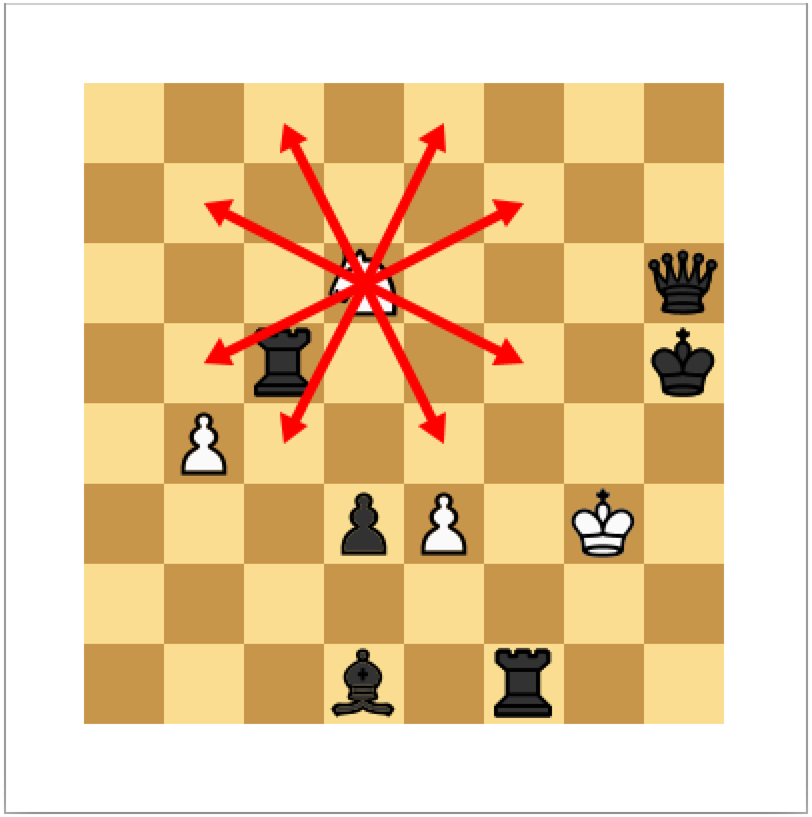
Example: Breakthrough

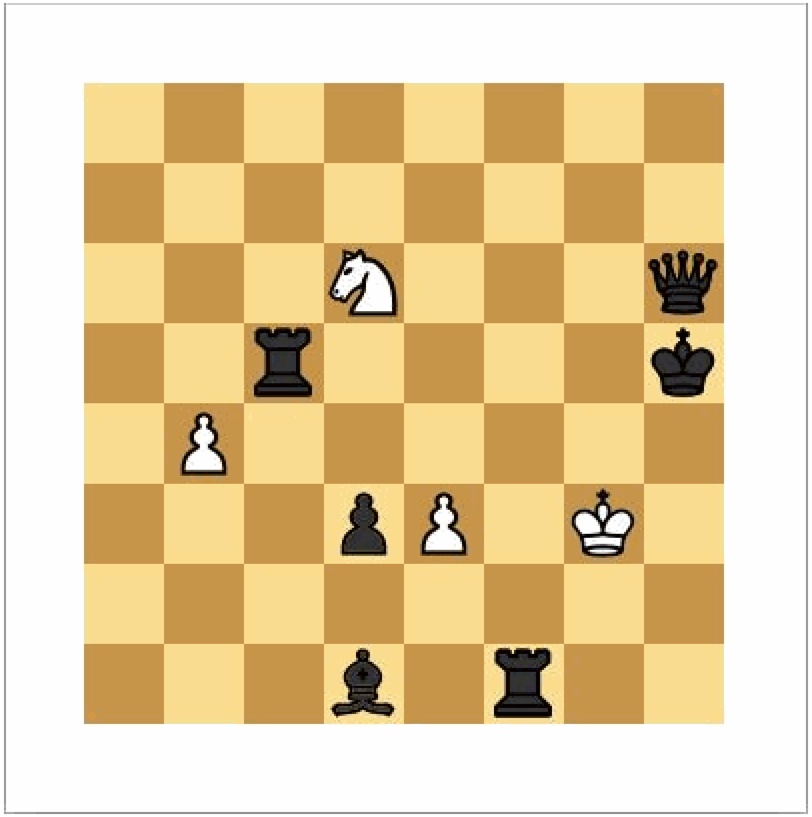


English rule translations

Move Visualisations

Example: Chess (knight)

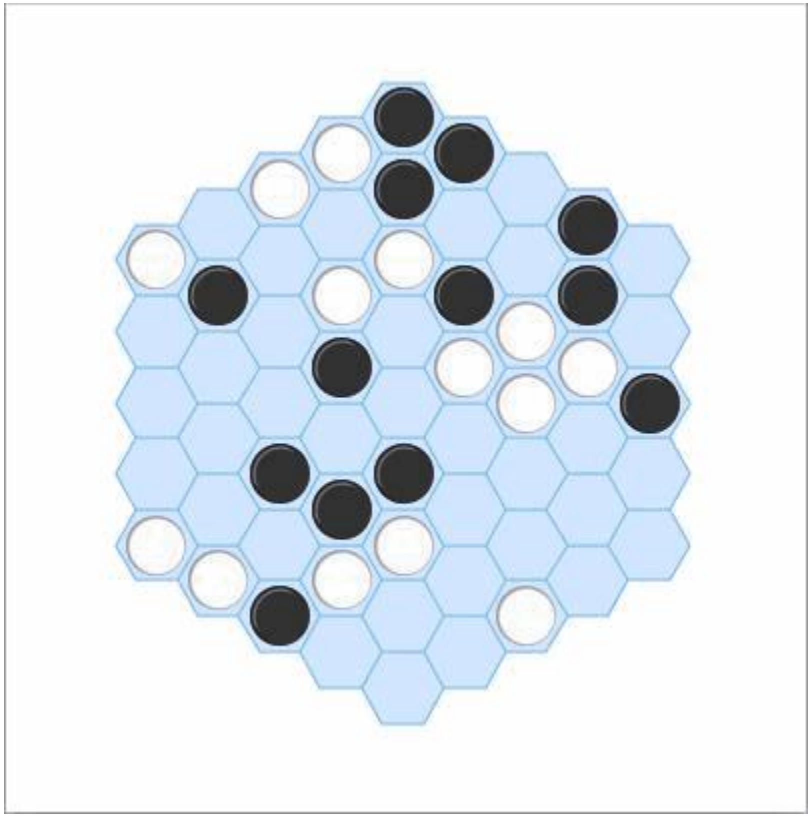


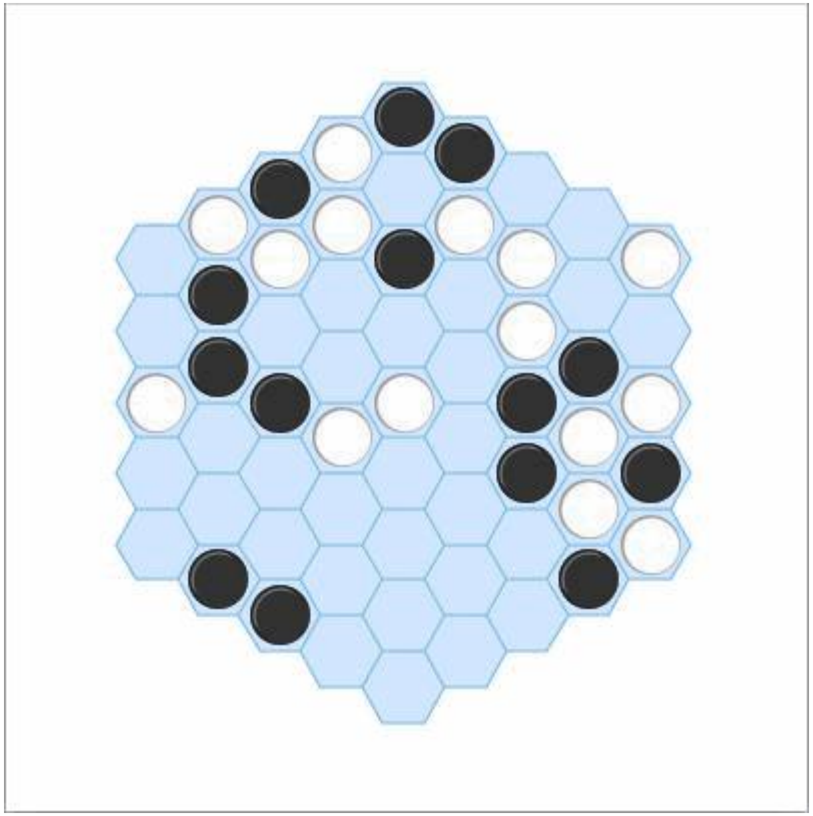


English rule translations

Winning / losing moves

Example: Yavalath

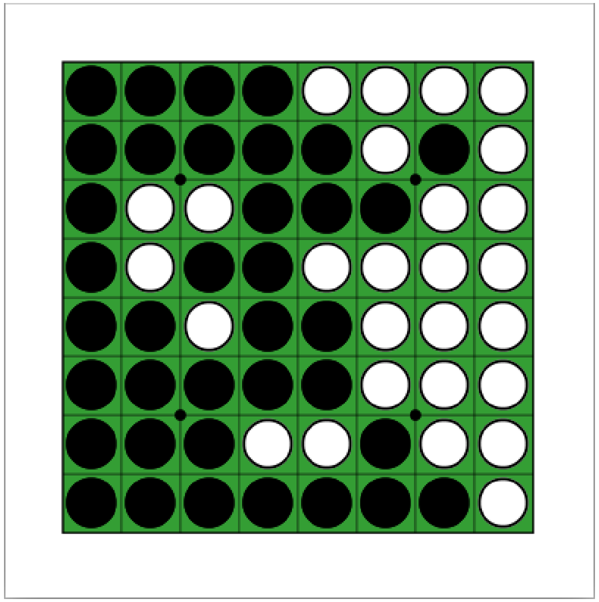




English rule translations

Game Strategies

Example: Reversi



mobility simple
Number of legal moves.
You should try to maximise the number of moves you can make (very low importance)

influence
Number of legal moves with distinct destination positions.
You should try to maximise the number of spaces you can move to (very high importance)

score
Score variable of game state corresponding to player.
You should try to maximise your score (very low importance)

Automatic Generation of Board Game Manuals

A few examples:

ludii.games/manuals/menu.html

- *Stephenson, et al., Automatic Generation of Board Game Manuals, ACG 2021*

Thank You

Questions?

Digital Ludeme Project

http://ludeme.eu

http://ludii.games

European Research Council

Established by the European Commission

Maastricht University

Twitter: @LudiiGames