DKE Lunchtime Lecture

Digital Ludeme Project

Modelling the Evolution of Traditional Games

Cameron Browne
Eric Piette
Dennis Soemers

14/11/2018

Maastricht University

Outline

Digital Ludeme Project:
- What it is
- What we are doing
- What we hope to achieve

Eric Piette:
- LUDII general game system

Dennis Soemers:
- Learning strategic features

Context

Games are ubiquitous throughout recorded human history:

- All humans play games
- All human cultures have their own games

Potential source of insight into cultural past:

- Don’t actually know much about ancient games!

Modern understanding based on (unreliable) modern reconstructions

Evidence or Coincidence?

Pachisi

- Traditional game of India
- Invented 6th–16thC

Evidence or Coincidence?

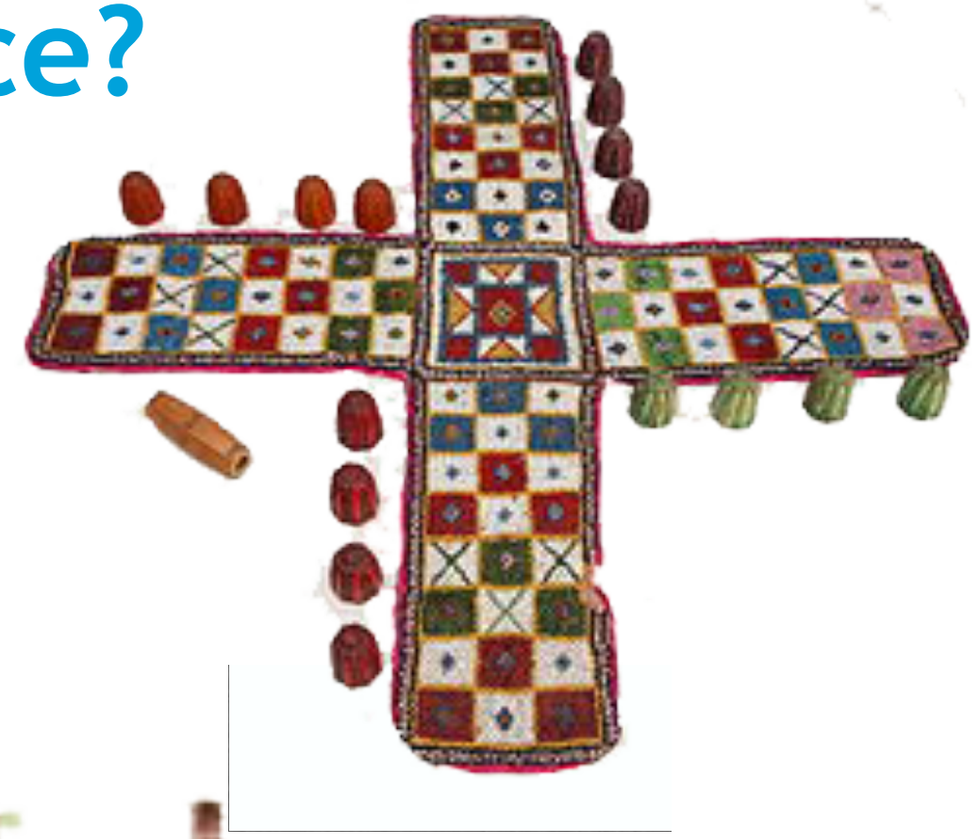
Pachisi
- Traditional game of India
- Invented 6th–16thC

Patolli
- Ancient Mexico
- Played c.200BC

Different rules...
Same board?

Maastricht University

Evidence or Coincidence?



Tyler (1879)
- Evidence of early contact

Erasmus (1950)
- “Limitation of Possibilities”
- Coincidence

Murray (1952)
- Assume coincidence as a last resort

No mathematical models, no software tools

Problem

Ancient games:
- Much archaeological evidence (boards, pieces, etc.)
- Almost no record of rules

Writing rare skill until recently:
- Games less important, last thing to be recorded

Passed on by oral tradition:
- Huge variety today
- Original rule sets lost

Missing Knowledge

Senet

- Ancient Egypt
- c.3100BC

Many sets found:

- e.g. Tutankhamen's
- Pristine condition
- No rules!

Is it even a game?

Missing Knowledge

Hints from hieroglyphic art:
- Two players
- Two colours
- Starting position

Special symbols on board:
- Entry points?
- Exit points?

Dozens of reconstructions:
- No mathematical models, no software tools

Bridging the Gap

Traditional game studies:
- Wealth of historical analysis
- Little mathematical analysis

Modern game AI studies:
- Huge surge in recent research
- Little interest in historical context

Almost no overlap:
- Seek to bridge this gap

Digital Ludeme Project

Digital Ludeme Project

- Five-year research project
- European Research Council:
  - ERC Consolidator Grant (€2m)
- Run at DKE since April 2018

Team:

- 1 x PI (me)
- 3 x RAs (Eric, Matthew, ?)
- 1 x PhD (Dennis)

Computational study of the world's traditional strategy games throughout recorded human history

Maastricht University

erc

European Research Council

Established by the European Commission

Objectives

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

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

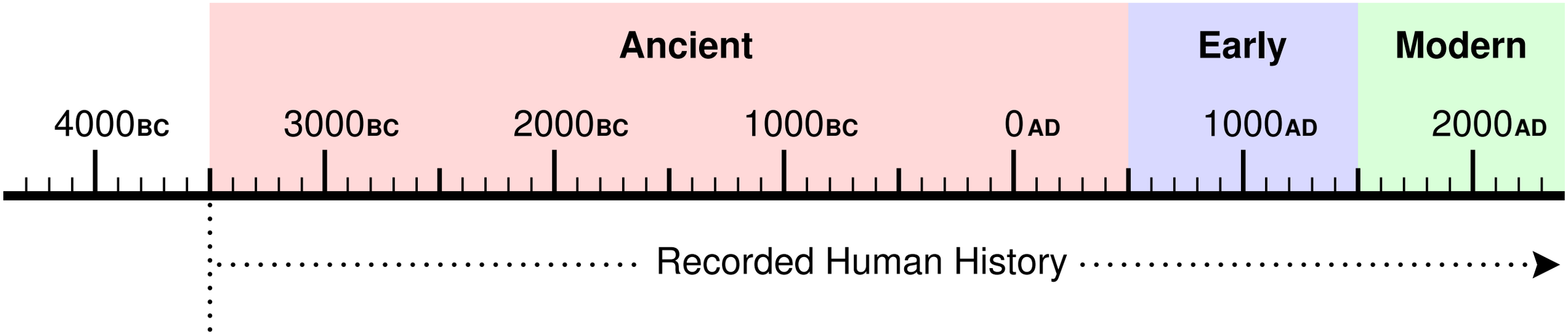
3. ***Map*** Spread of games (and associated mathematical ideas) throughout recorded history

Aim: To improve our understanding of traditional games using modern AI techniques

Scope

Traditional Games of Strategy

- *Traditional*: No known inventor or proprietary owner
- Strategy: Mental skill, *e.g. board, tile, card, etc.*



Range: ~3100BC ... ~1900AD

- 1,000 most influential examples (+ variants, + recon.s…)
- Evaluation of up to 1 million rule sets

Ludemes

- Game “memes”:
  - Units of game-related information
  - Building blocks (DNA) of games
- Encapsulate key concepts

Ludemes

- Game “memes”:
  - Units of game-related information
  - Building blocks (DNA) of games
- Encapsulate key concepts

e.g.

```
(tiling square)
```

```
(size 3 3)
```

Ludemes

- Game “memes”:
  - Units of game-related information
  - Building blocks (DNA) of games
- Encapsulate key concepts

e.g.

```
(tiling square)
```

```
(size 3 3)
```

```
(board
   (tiling square)
   (shape square)
   (size 3 3)
)
```

Ludemes

- Game “memes”:
  - Units of game-related information
  - Building blocks (DNA) of games
- Encapsulate key concepts

e.g.

```
(tiling square)
```

```
(size 3 3)
```

```
(board
   (tiling square)
   (shape square)
   (size 3 3)
)
```

```
(game Tic-Tac-Toe
   (players White Black)
   (board
      (tiling square)
      (shape square)
      (size 3 3)
   )
   (end (All win (in-a-row 3)))
)
```

Stanford GDL

Stanford University:

- Game Description Language (GDL)
- Academic standard

Explicit Instructions:

- Verbose
- Difficult
- Inefficient
- Not that general
- No encapsulation!

Maastricht University

```
(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))
```

Comparison

```
(game Tic-Tac-Toe
  (players White Black)
  (board
    (tiling square)
    (shape square)
    (size 3 3)
  )
  (end (All win (in-a-row 3)))
)
```

```
(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))
```

Comparison

```
(game Tic-Tac-Toe
  (players White Black)
  (board
    (tiling square)
    (shape square)
    (size 5 5)
  )
  (end (All win (in-a-row 3)))
)
```

```
(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))
```

Comparison

```
(game Tic-Tac-Toe
  (players White Black)
  (board
    (tiling hexagonal)
    (shape square)
    (size 7 7)
  )
  (end (All win (in-a-row 3)))
)
```

```
(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))
```

Comparison

```
(game Tic-Tac-Toe
  (players White Black)
  (board
    (tiling hexagonal)
    (shape square)
    (size 7 7)
  )
  (end (All win (no-moves)))
)
```

```
(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))
```

Ludemic Model

Simple:
- Compact (QR codes), human-readable

Powerful:
- General, extensible, evolvable, efficient

Useful:
- Encapsulates key concepts
- **Labels** key concepts

Good for design:
- Allows **full range** of traditional strategy games

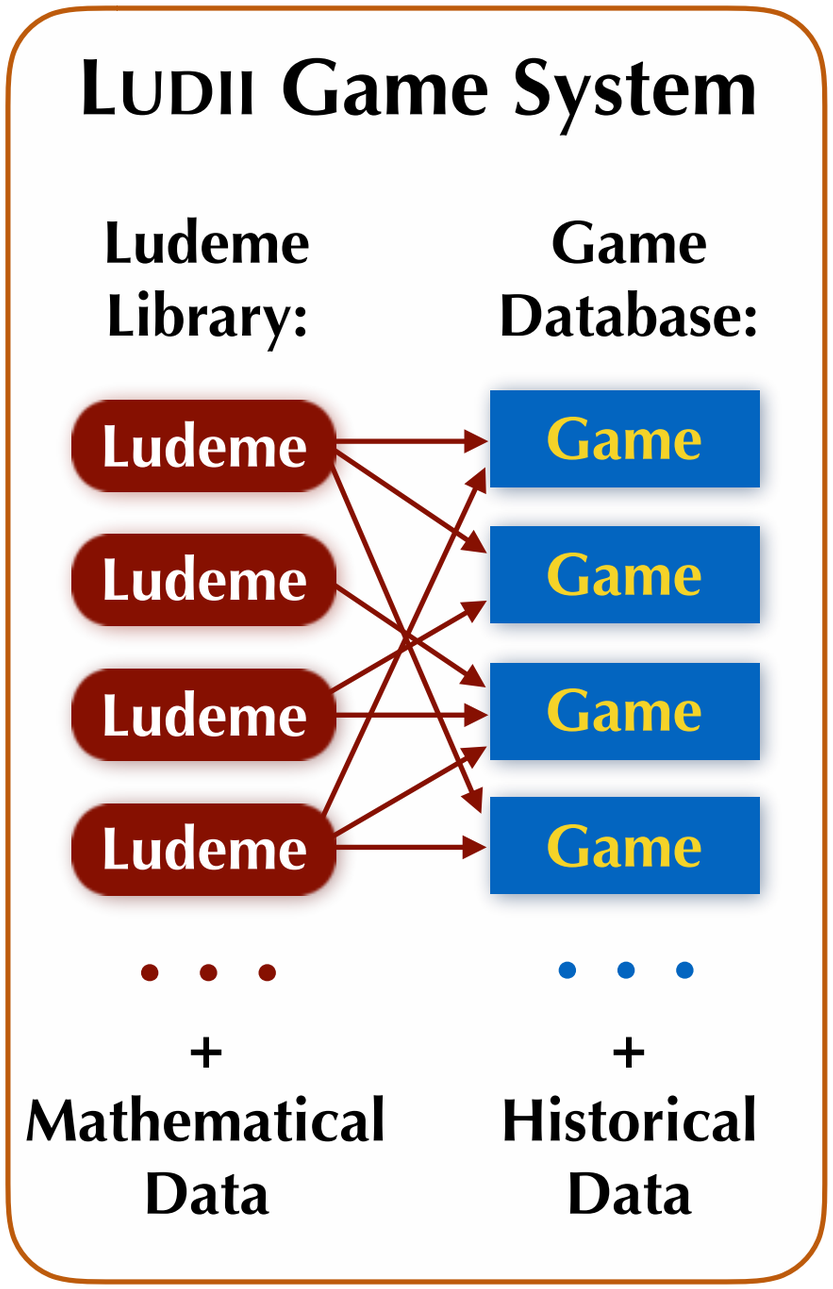
LUDII

LUDII general game system:

1. **Ludeme library**
   - Each ludeme is a Java class
   - Gives each ludeme a **name**
   - Tag with math. concepts

2. **Game Database**
   - Game descriptions
   - Historical data

[Eric to discuss]



Genetic Model of Games

Ludemes = form (*genotype*)
Play = emergent behaviour = function (*phenotype*)

Problem:

- No genetic material = no genetic distance

Solution:

- Use ludemic distance instead:
  - Edit distance between ludeme trees
  - Number of steps required to change one game into another

Computational Phylogenetics

"Family tree" of traditional games:

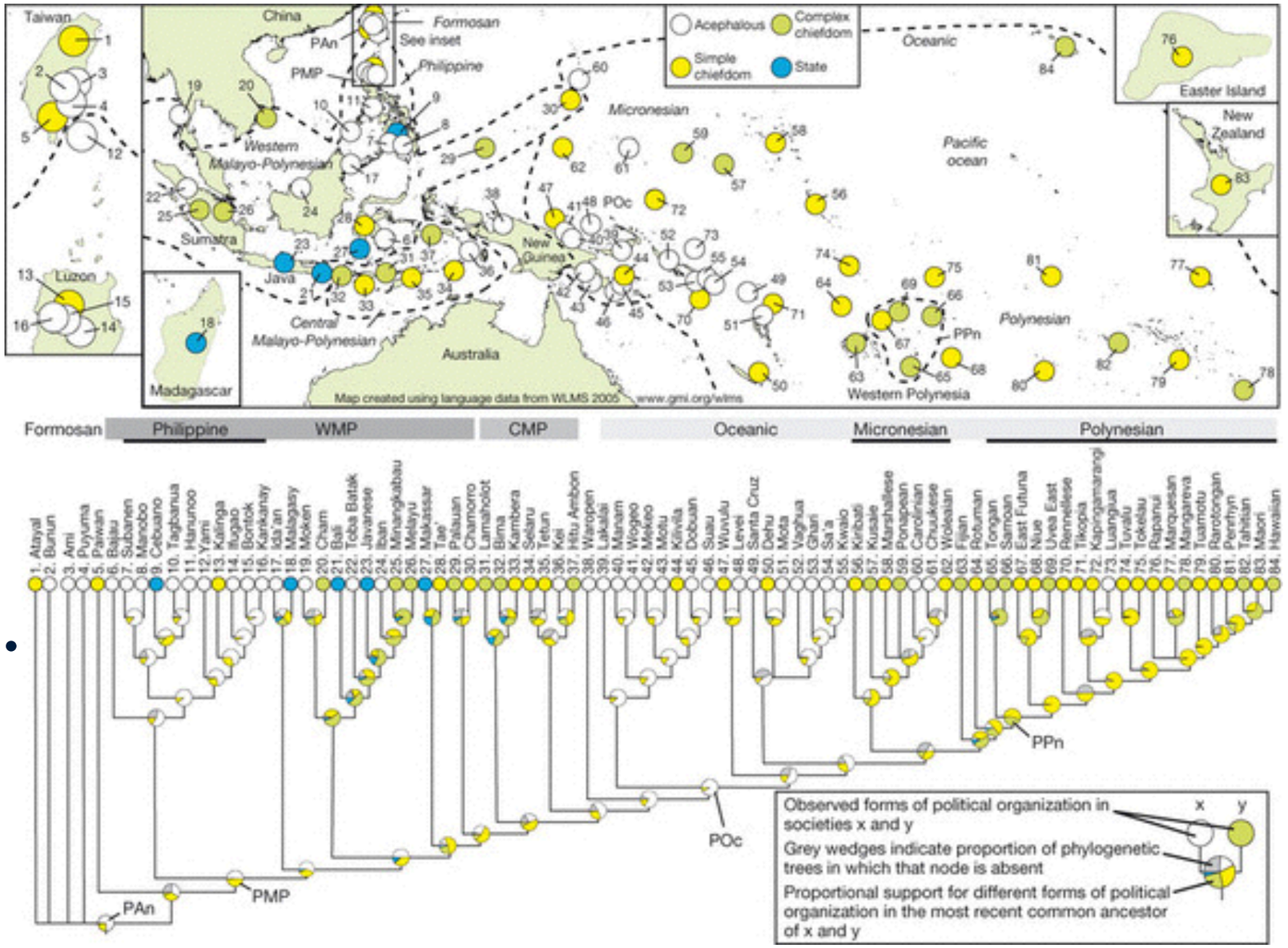
- Principles similar to linguistics

Ancestral State Reconstruction:

- Probabilistic recon. of ancestral traits

Missing links?

- Games that *may* have existed



Phlyogenetic analysis of Austronesian societies
Currie (2010) *Nature*

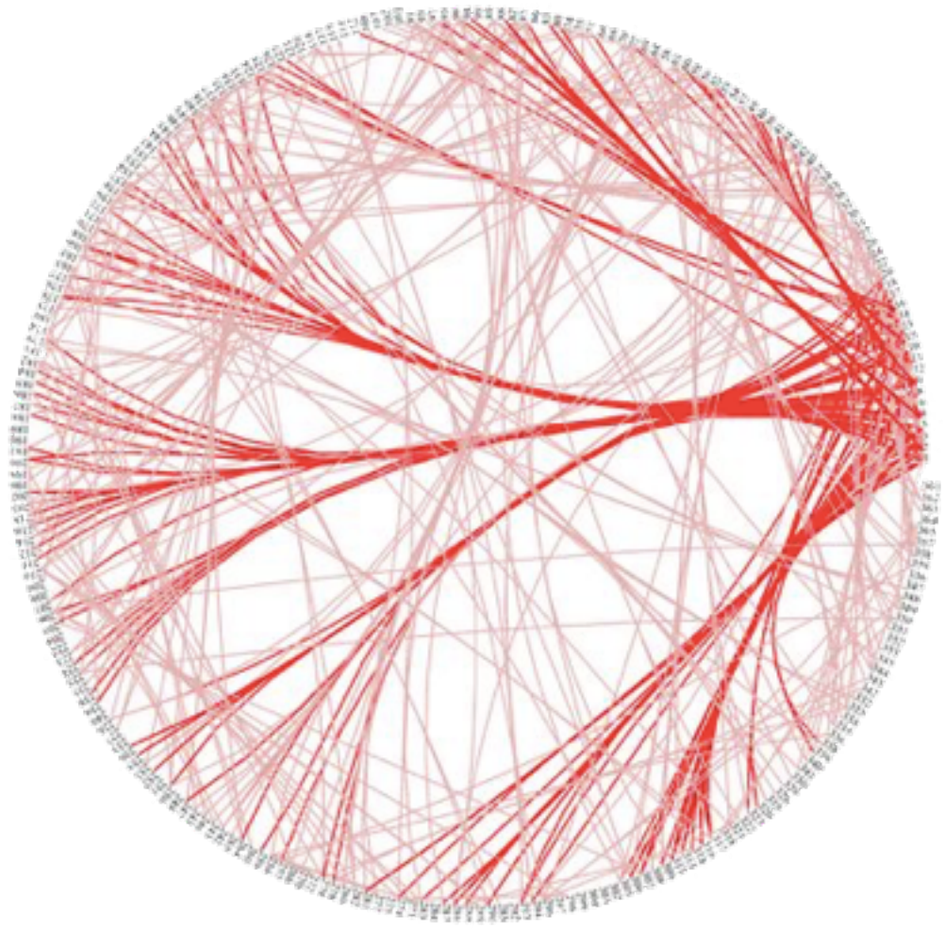
Maastricht University

Horizontal Influence Maps

Problem:

- Classical phylogenetics assumes vertical gene transfer

- Games spread through horizontal gene transfer (*ethnogenesis*):
  - Rules from any source
  - Rules from any time
  - Can't prove origins (*parents*)



HIM of programming languages
Valverde & Sole (2015) *JRSI*

Solution:

- Horizontal Influence Maps (HIM)
- Different view of relationships between data

Historical Authenticity

Given a rule set

How likely is it to have occurred:

- In that time?
- In that place?
- In that culture?

e.g. **Birrguu Matya**

- Australian Aboriginal
- Traditional? No!
- Obvious outlier
- “Invented tradition”

Game Quality

Quality of reconstructed rule sets as games:

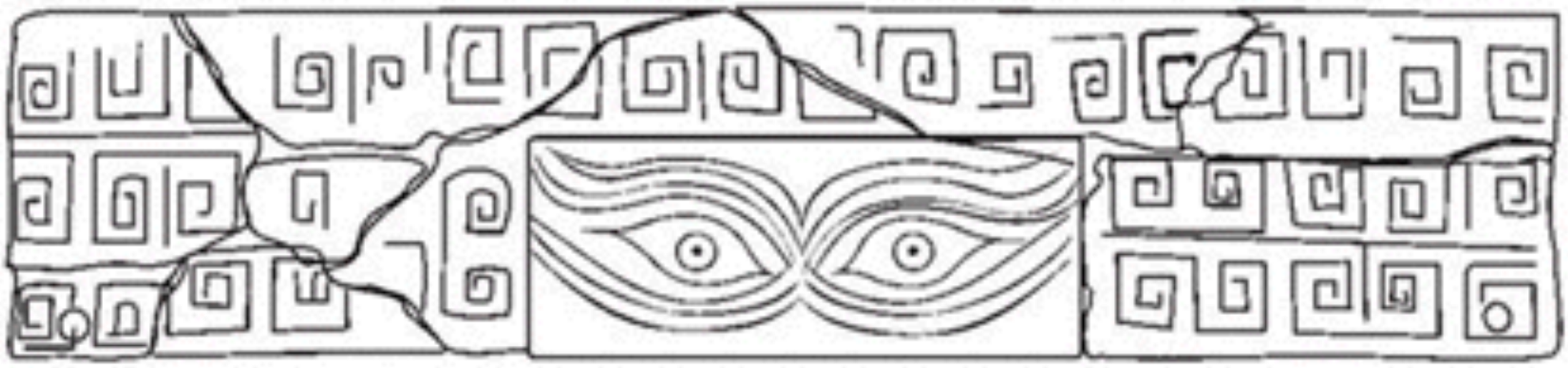
- No universal indicator

Player preferences change:

- Period, culture, individual

Can identify flawed rule sets:

- Biased, drawish, too short or long



Translation Errors

Hnefatafl
- Scandiavia, c.400BC
- No rules found

Linnaeus (1732)
- Saw Tablut played
- Recorded in travel diary (in Latin)

Smith (1811)
- Translated into English

Murray (1913) *History of Chess*
- Published rules, became de facto

Translation Errors

BUT...

Smith's translation had a critical error:

- "…likewise the king…"

not

- "…except the king…"

King almost impossible to capture:

- King's side always wins
- Biased game, corrected ever since
- **Not how it was played**

Transcription Errors

Mu Torere

- Maori, New Zealand, 18thC
- Special opening rule

Some historical accounts omit this rule:

- Game ends after 1 move
- **Not how it is played**

Strategic Potential

Flaws easy to detect: game quality harder to define

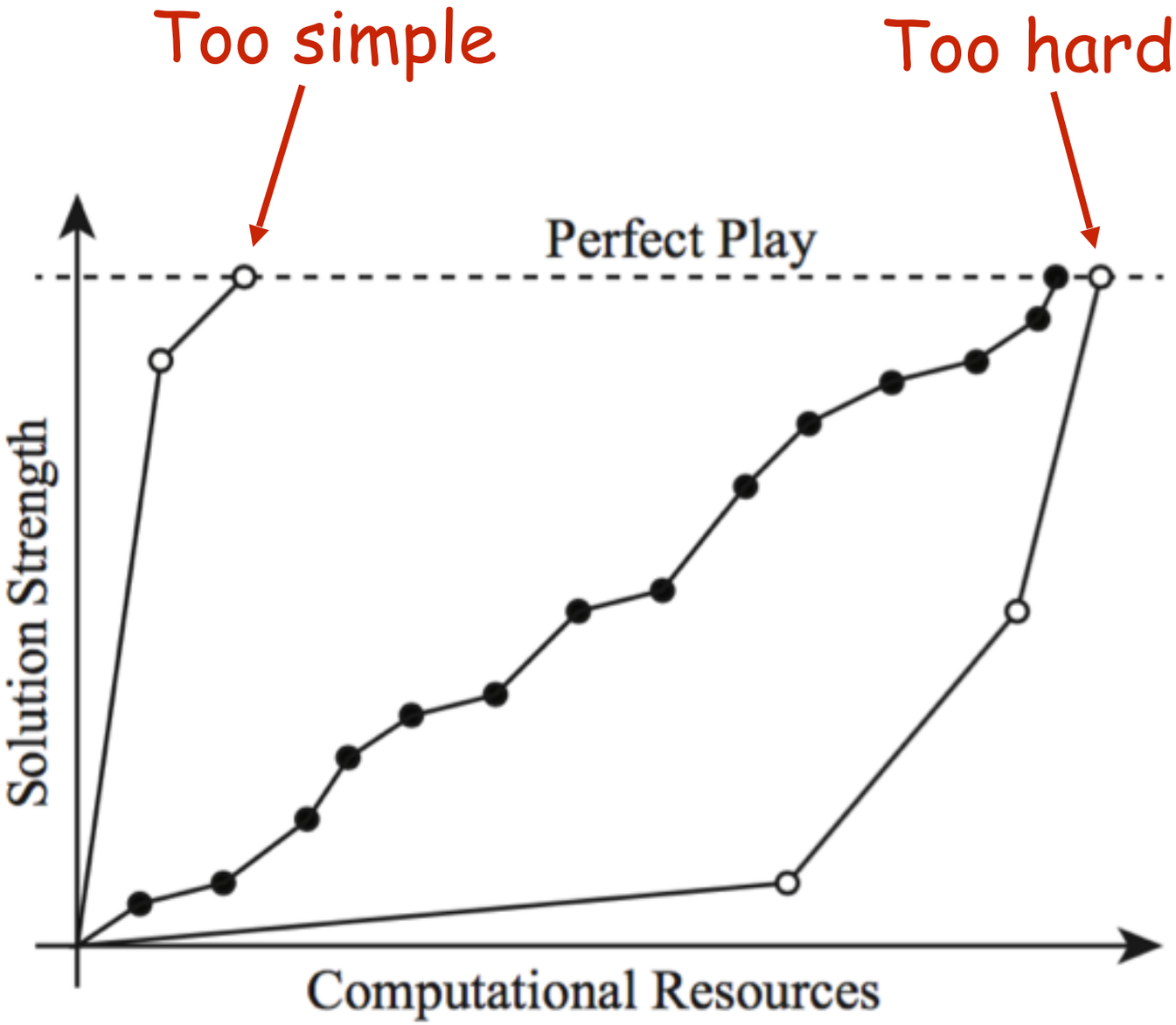
Hypothesis:

- *Strategic potential is a universal indicator of quality*

Strategy ladder:

- Not too easy
- Not too hard
- Linear accumulation of strategies

[Dennis to discuss]



Lantz *et al.* (2017) *AAAI'17*

Putting It All Together

Improve reconstructions by optimising rule sets:

1. Define known constraints (equipment, rules, etc.)

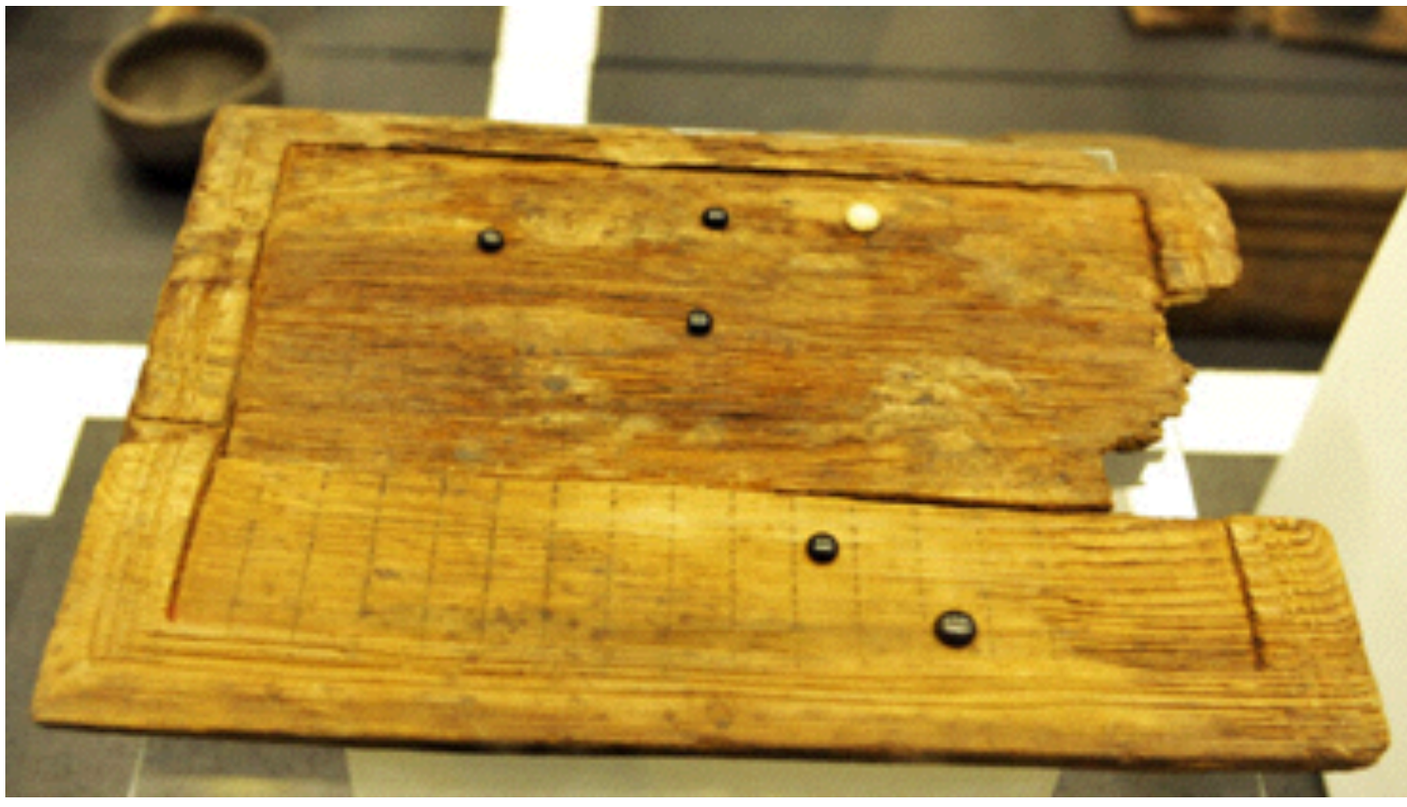
2. Evolve using plausible ludeme combinations

3. Eliminate flawed rule sets

4. Maximise for:
    1. Historical authenticity
    2. Game quality

Forensic Game Reconstruction

Poprad Game (Slovakia)
- Tomb dated to 375AD
- Germanic chieftain



Board:
- 17x15/16 grid (not Go!)

Pieces:
- 2 x Colours
- 1 or 2 x Sizes?

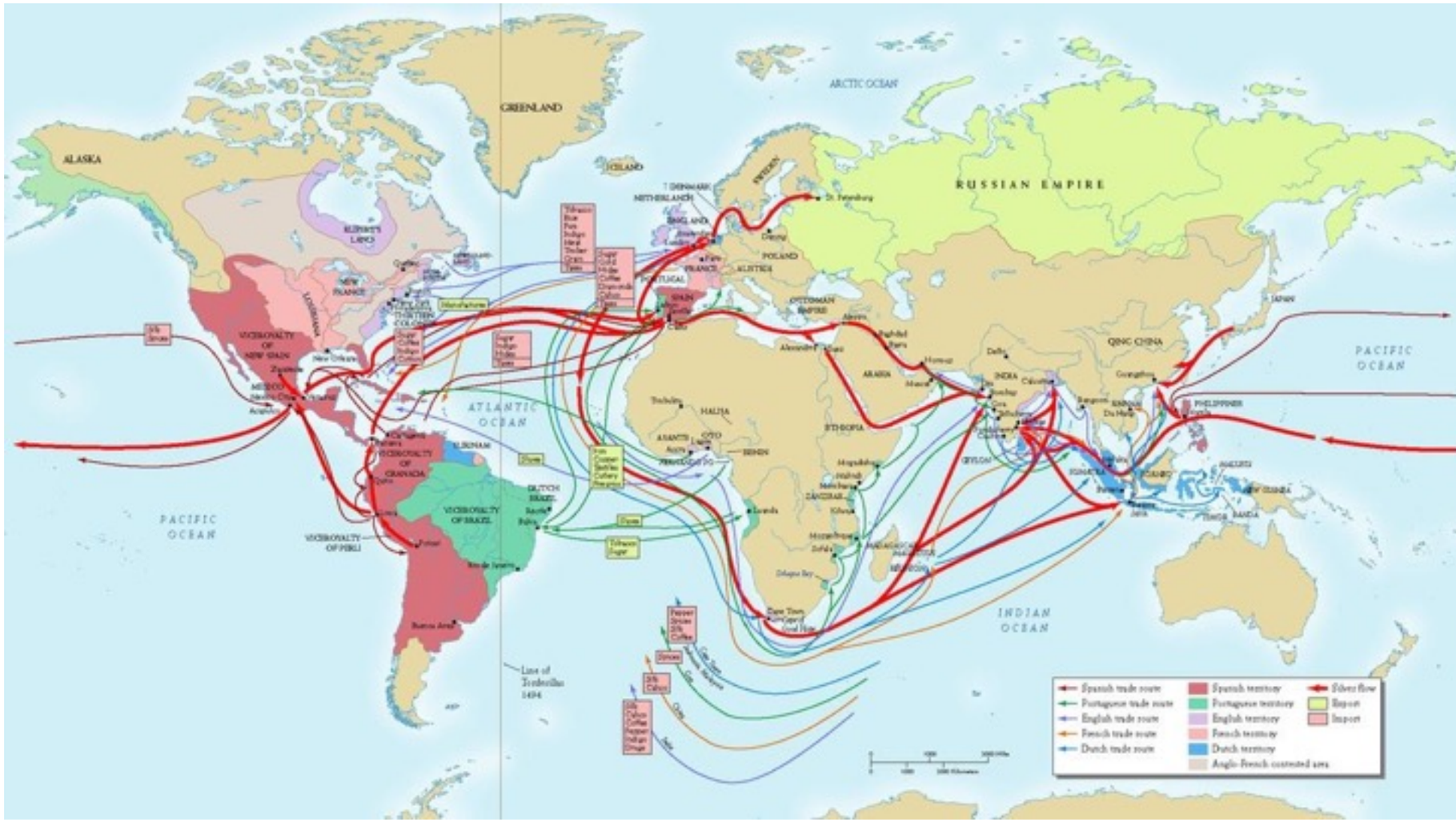
Ulrich Schadler (2018)
- “An impossible task”
- LUDII will turn impossible task into a difficult one

Cultural Mapping

Map spread of games/ludemes throughout human history

Correlate with:

- Trade routes
- Explor. routes
- Diasporas



Maastricht University

Silk Road Trade Routes

Very important in history games!

- Fertile crescent:
  - Egypt
  - Mesopotamia
- India
- Asia

Can we track spread of early games?

Map 12.1 The silk roads,

Maastricht University

GeoCron

Queriable geo-temporal database

Yearly maps:

- 3,000BC—today
- 2,000 cultures

Provide GPS+date:

- Culture/civilisation
- Country/nation/state
- Landmarks
- Historical events (+ routes!)

GeoCron: Viking route from Norway to Paris (845AD)

Historical profile ⇒

Cultural location

Maastricht University

Games and Mathematics

Tagging ludemes with underlying mathematical concepts:

- Geometry
- Logic
- Algebra
- Arithmetic

Map spread of games/ludemes

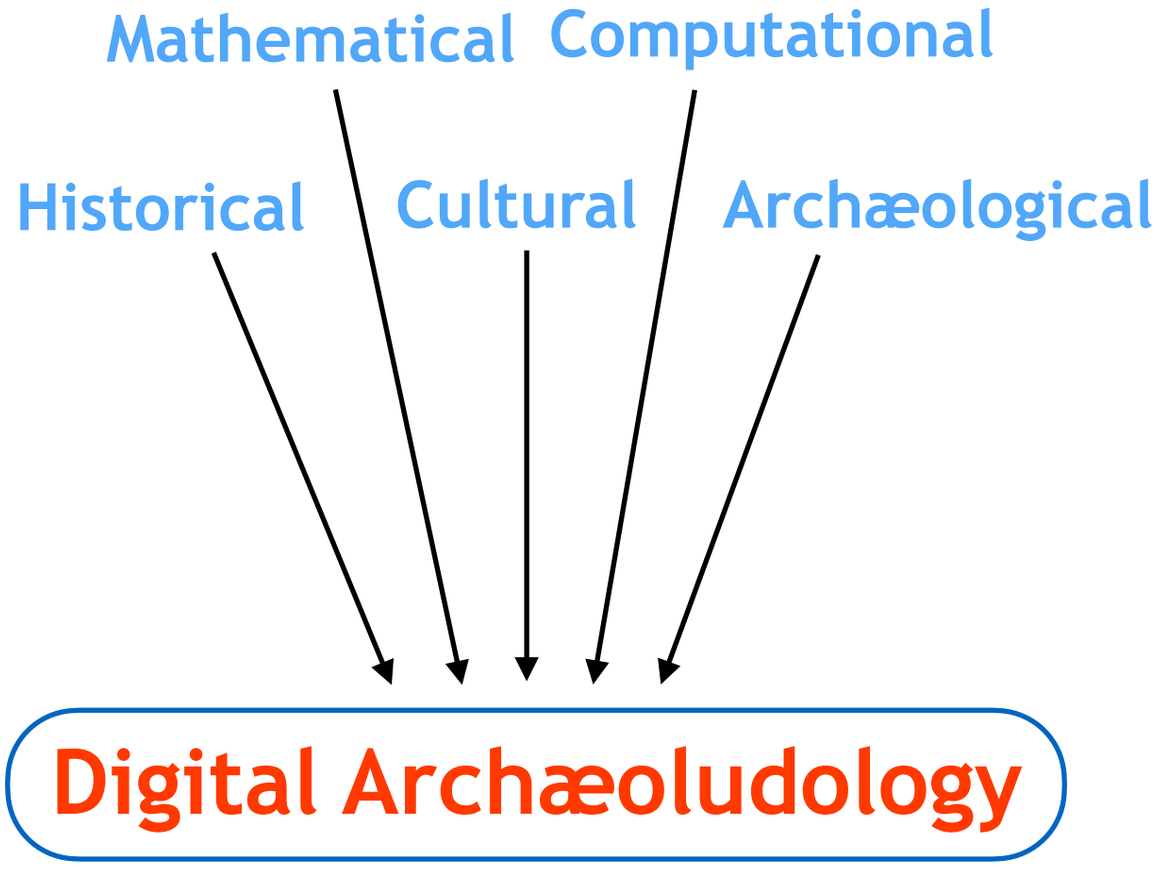
Can we also correlate spread of associated mathematical ideas throughout history?

Rithmomachia (11thC)

Digital Archæoludology

Digital Archæoludogy:

- New field of research
- Several research strands
- Single unified approach



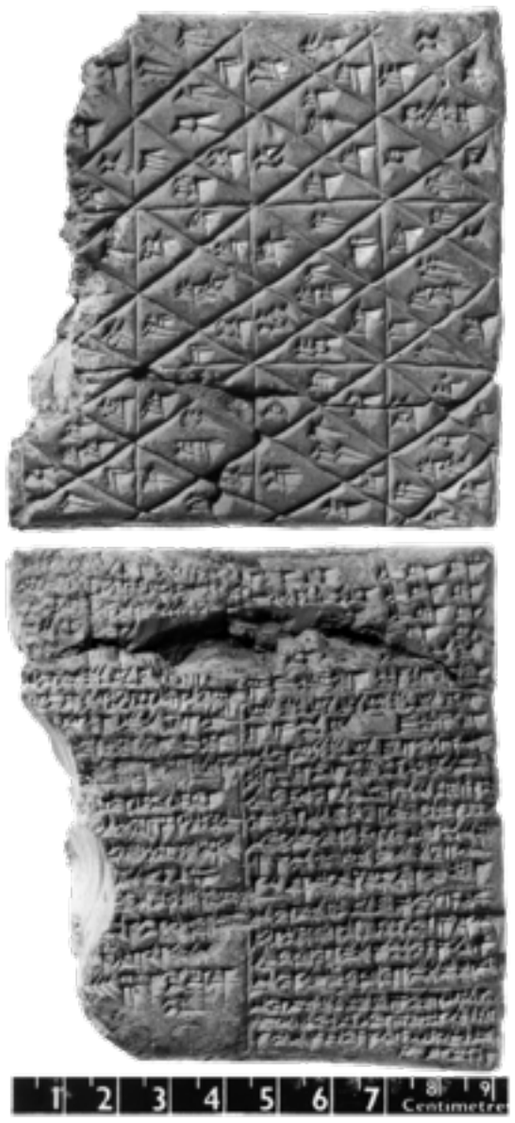
Modern comput. techniques:

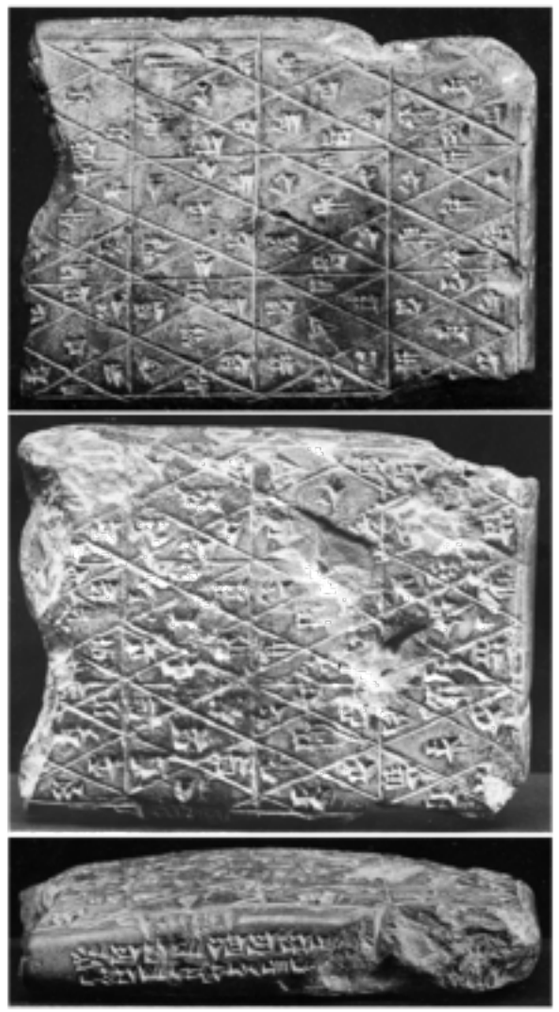
- Analysis and reconstruction
- Incomplete descriptions

Preserving Knowledge

Royal Game of Ur
- Mesopotamia, 2600BC

Tablets dated 177BC:
- **B.M.:** One of 130,000
- **Paris:** Destroyed 1940s





Oldest recorded rules:
- Found by Finkel (1990)
- Lucky find!
- Losing game evidence all the time

Maastricht University

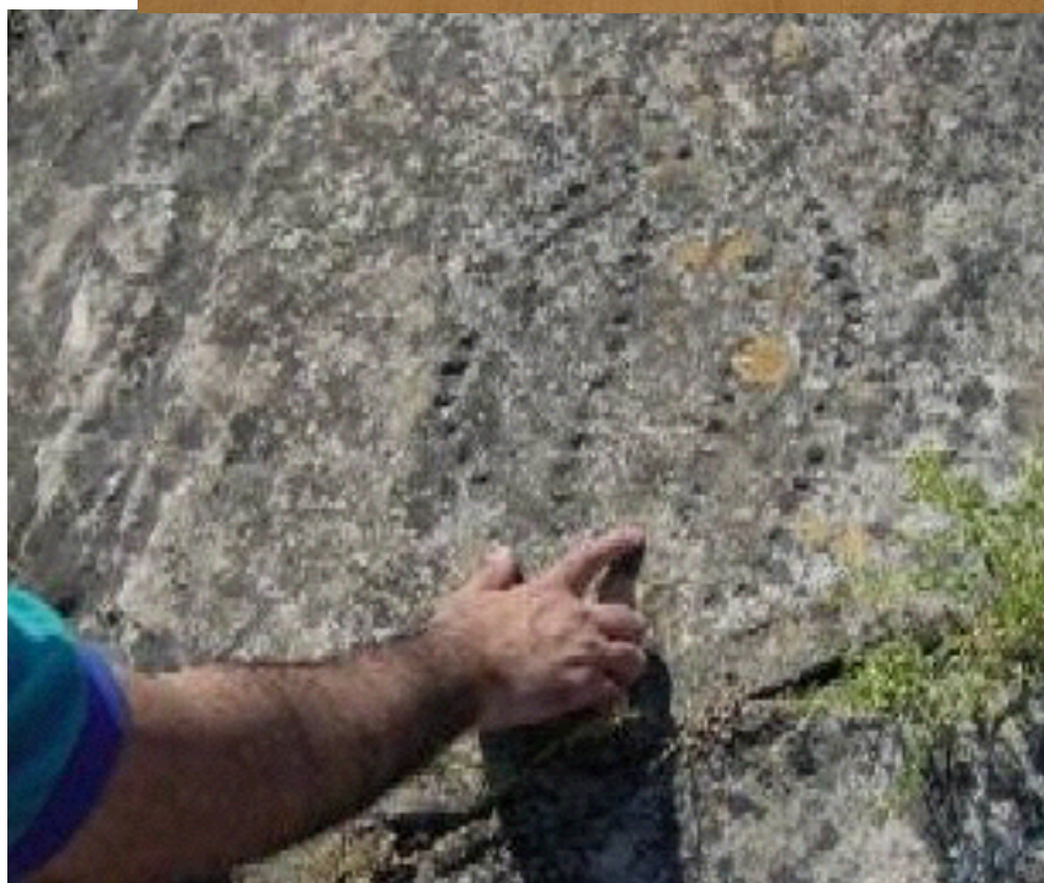
Preserving Knowledge

Hounds and Jackals
- Egypt ~2000BC

Azerbaijan Carving
- Azerbaijan ~2000BC : Game? Art?

Walter Crist (US):
- Evidence of cultural contact
- Site destroyed last year
- Not published

Aim: Help preserve cultural heritage of games

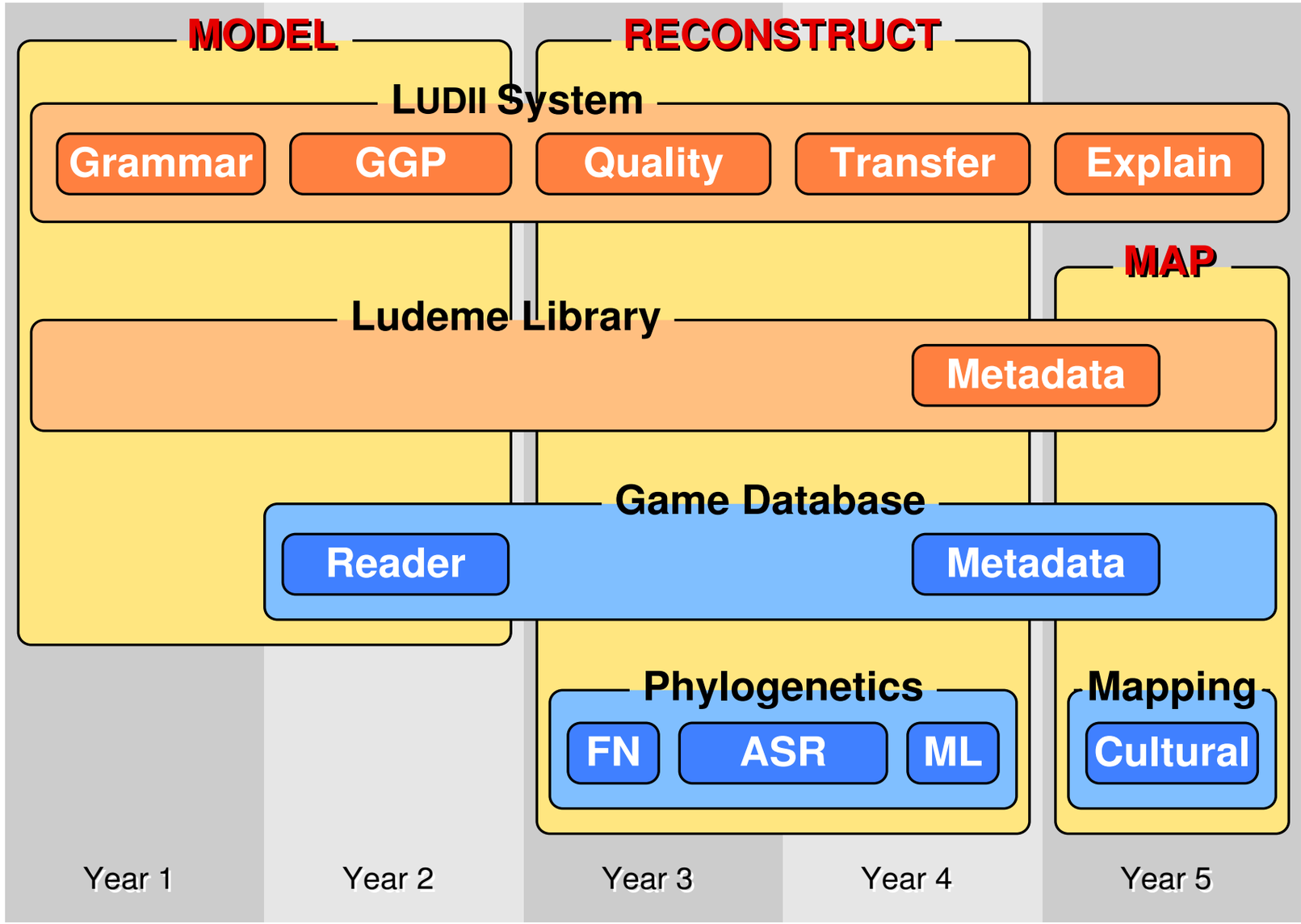


Maastricht University

Workplan

PI

- RA1, PhD1 (Computational)
- RA2, PhD2 (Cultural)



OUTPUTS

LUDII
- GGP system
- Ludemes
- Games + Reconstructions
- Manuals
- Web site
- AI methods

Sympos. 1
- Proceedings

Sympos. 2
- Proceedings

Conference
- Proceedings

Exhibition
- Catalogue
- Interactive Maps
- Public lectures
- Artefacts
- Displays

Other
- 45+ papers
- 3 books
- 2 PhD theses
- Patents?

FN	=	Family Tree/Network
ASR	=	Ancestral State Reconstruction
ML	=	Missing Links

Maastricht University

Team

Principal Investigator
- Cameron

Computational

Postdoc
- Eric

PhD
- Dennis

Cultural

Postdoc
- Hiring now!
- Historian/anthropologist
- Advise cultural aspects

Postdoc
- Matthew

Maastricht University

Eric Piette

- Postdoctoral Researcher
- Digital Ludeme Project

Maastricht University

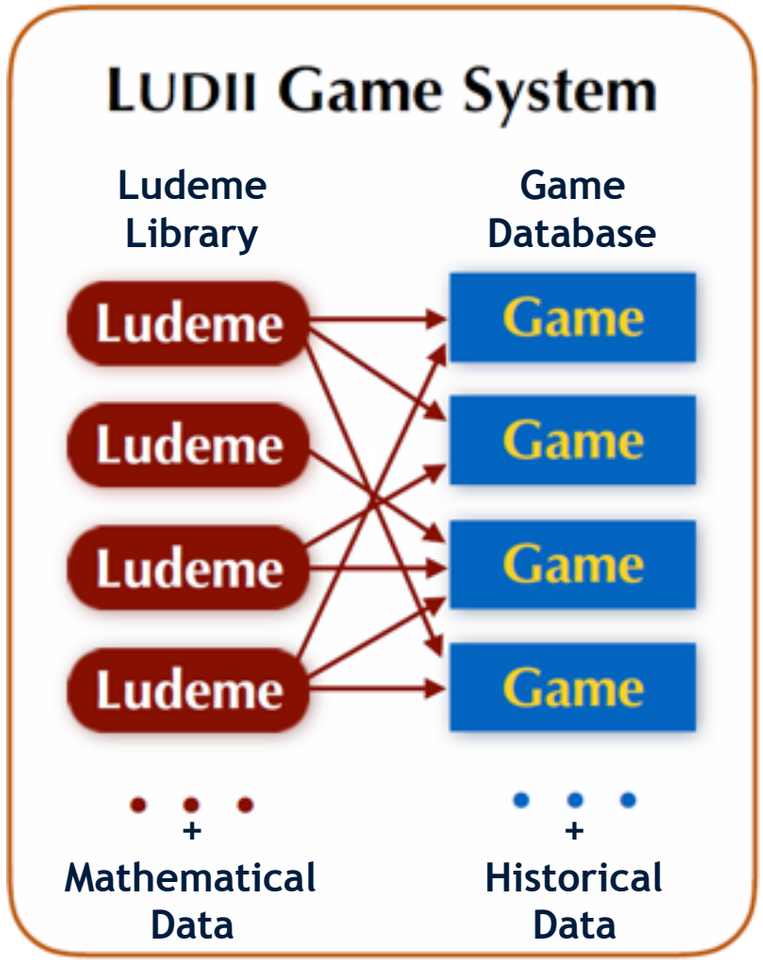
LUDII General Game System

- Play, evaluate, reconstruct
- Full range of traditional games

1. Model

Ludeme Library

- Each ludeme:
  - Java class
  - Meaningful name
  - Tagged with math.keywords



Game Database

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

Java Project

Core Modules:

1. **AI** - Default AI agents for playing games in the DB
2. **Common** - Constants, annotations, etc.
3. **Grammar** - Generates game grammar from ludemes
4. **Library** - Ludemes in structured class hierarchy
5. **Player** - Main play controller (GUI and CLI)

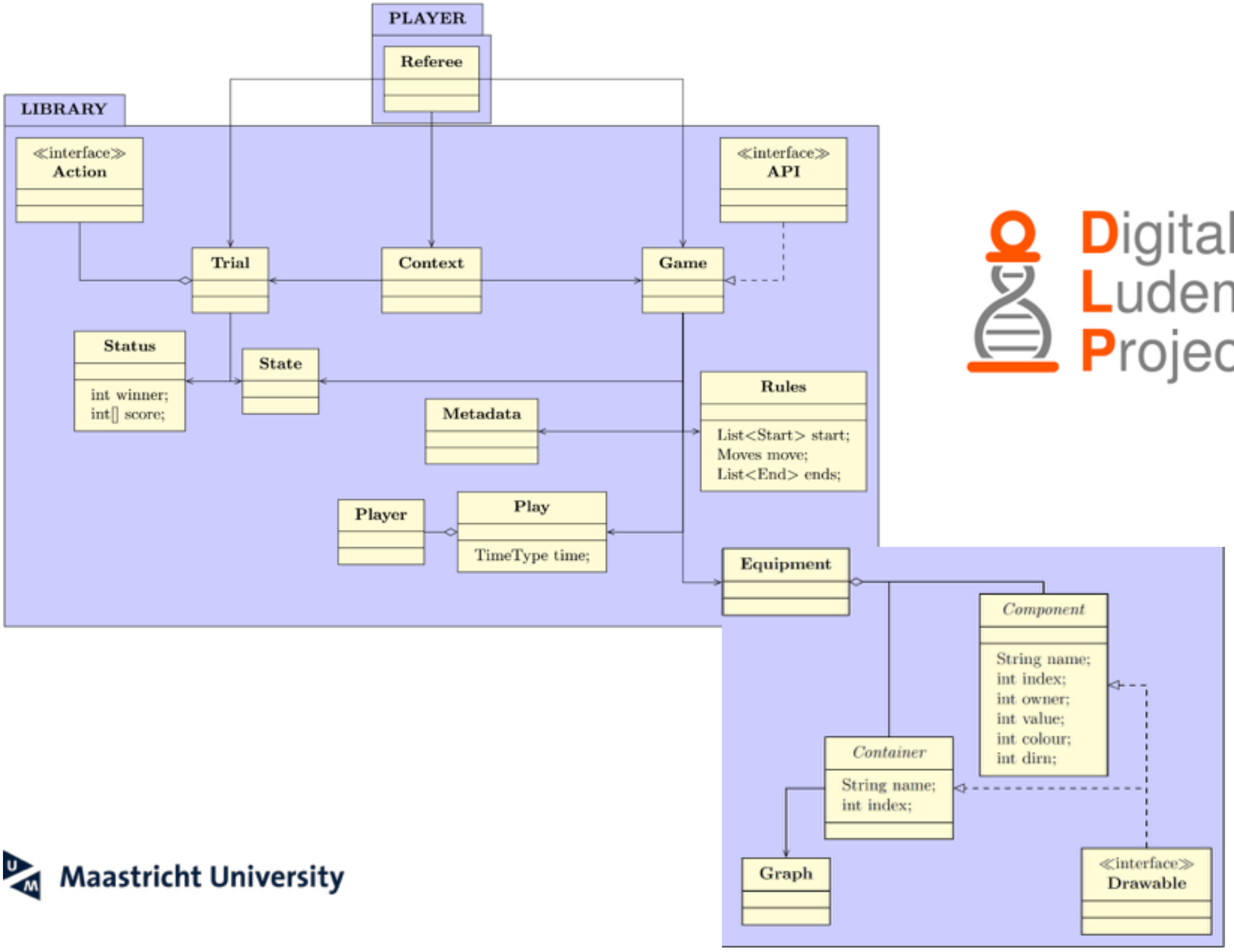
```
(game "Tic-Tac-Toe"
  (players White Black)
  (board (square 3))
  (play
    (add
      (piece Own)
      (board Empty)
    )
  )
  (end (win All (line 3 Own Any)))
)
```

Grammar

Player

Library

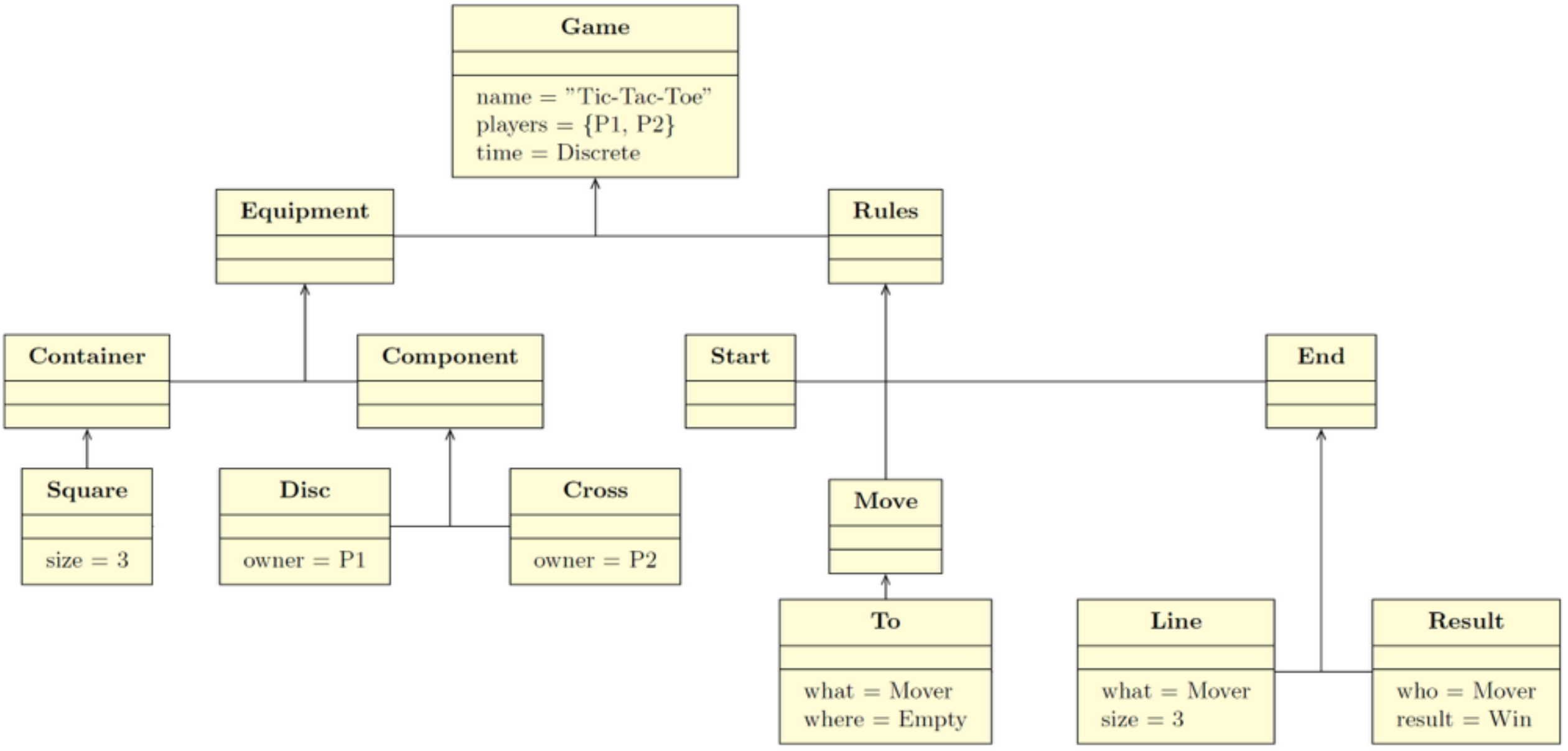
Core of LUDII: Ludeme Library



Digital Ludeme Project

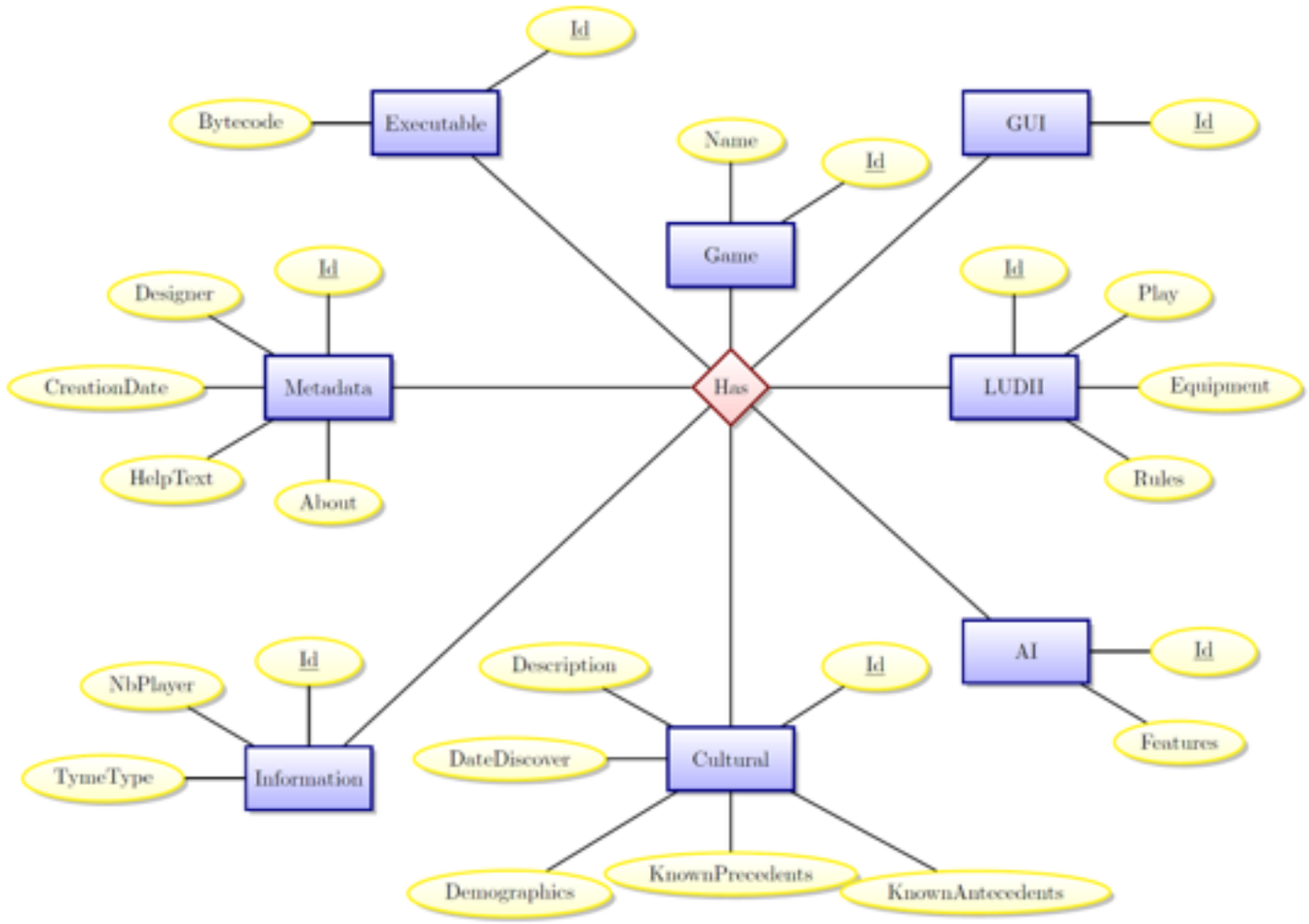
Maastricht University

Example: Tic-Tac-Toe



Game Database

- Goal: 1,000 representatives games
- Their known variants
- Automatically generated reconstructions



For now:

- 30 different games
- 80 variants

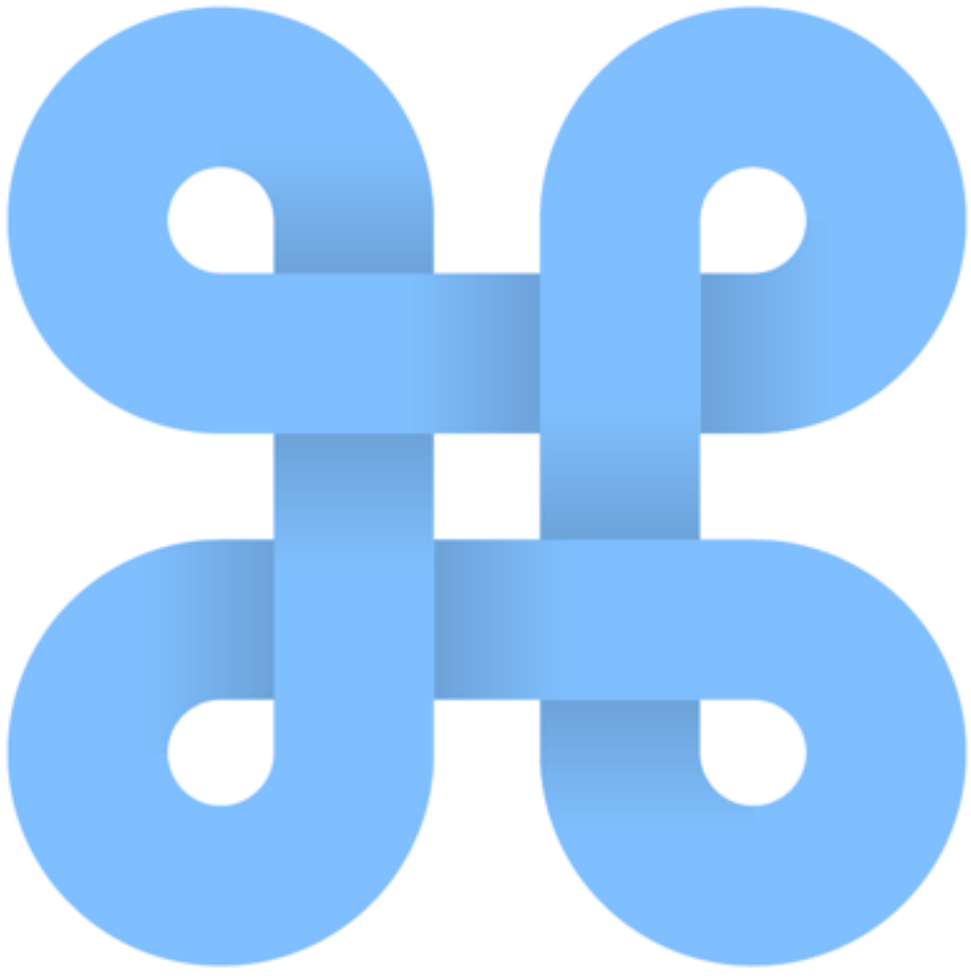
With:

- 141 kind of rules
- 8 types of board
- 28 types of pieces

Ludii Public Portal

Access games in the database:

- Play AI agents
- Play other users
- Evaluate rule sets
- AI tournaments

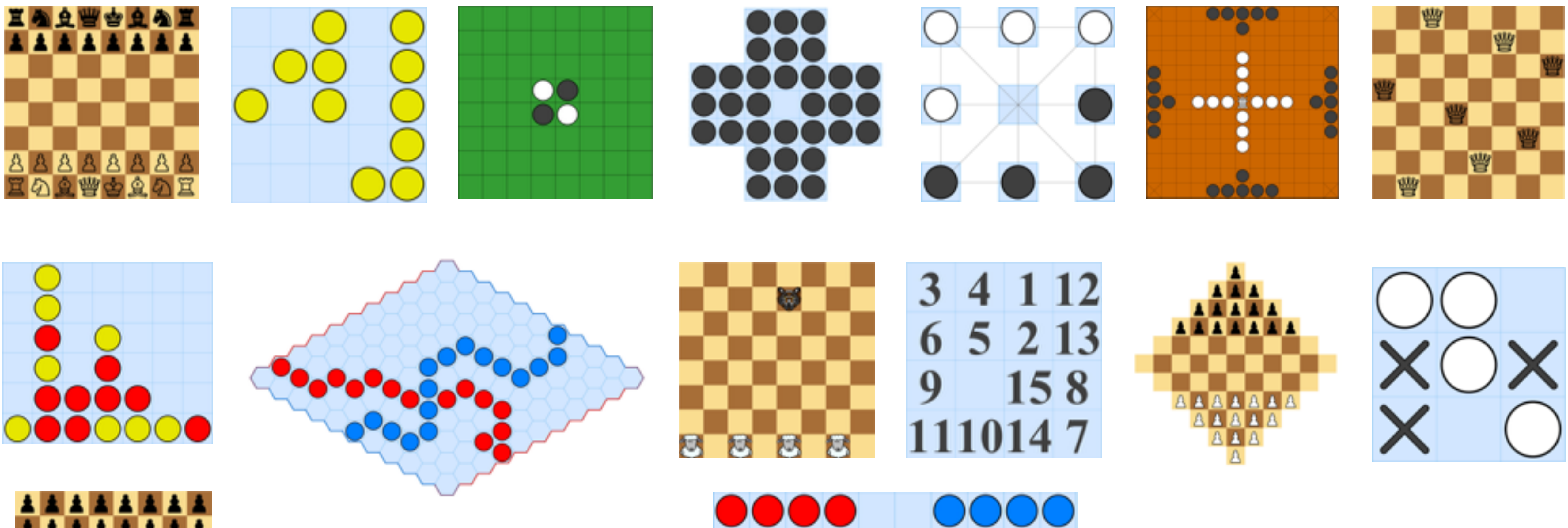


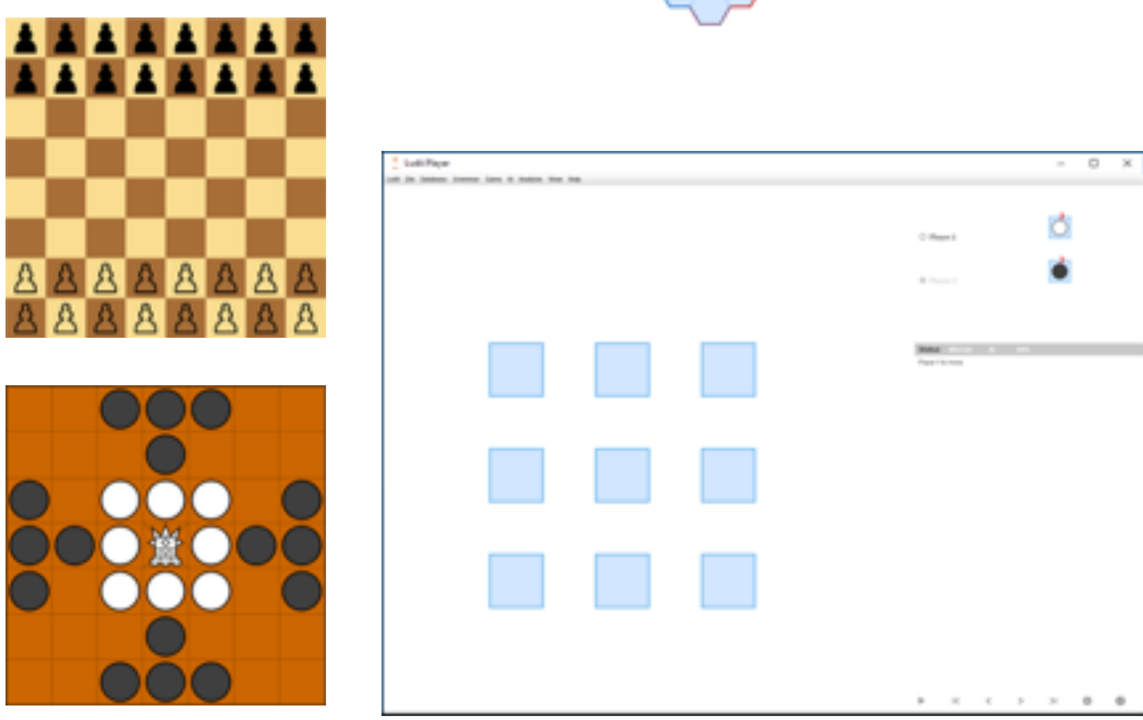
www.ludii.games

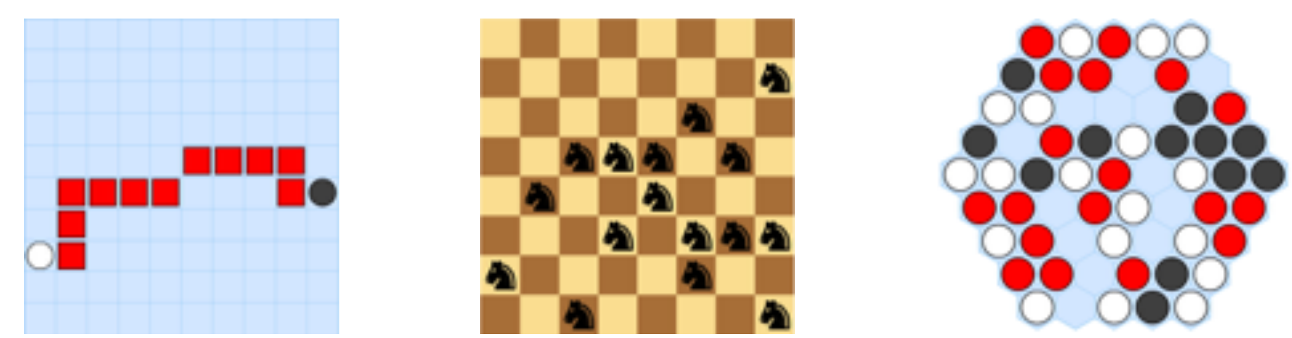
Release:

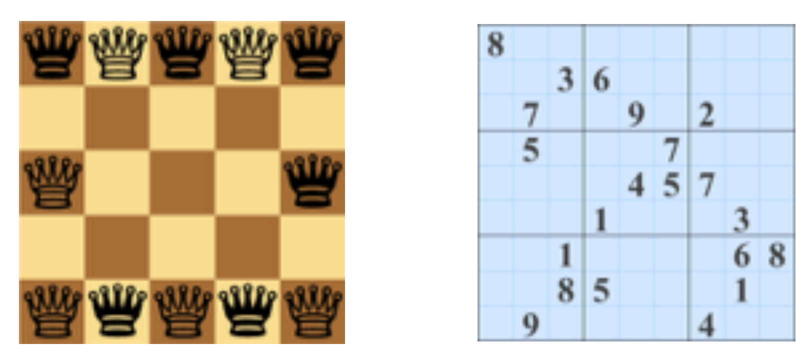
- Mid-2019?

GUI Examples









Maastricht University

Digital Ludeme Project

Preliminary results

Game Name	LUDII	GGP-base	Rate
8 Puzzle	22,092	4,113	5.4
8 Queens	790,200	1,946	406.1
Breakthrough	2,339	1,123	2.1
Chess	308	0.06	5133.3
Connect 4	83,002	13,664	6.1
Hex	8,857	195	45.4
Knight's Tour	84,137	75,000	1.1
Lights Out	20,395	11,799	1.7
Peg Solitaire	8,234	3,172	2.6
Reversi	922	203	4.5
Skirmish	429	124	3.5
Sudoku	100,201	635	157.8
Tic-Tac-Toe	693,568	85,319	8.1
Tron	139,273	121,989	1.1
Wolf and Sheep	6,736	5,532	1.2

Playouts per second
vs
GGP/GDL versions

Dennis Soemers

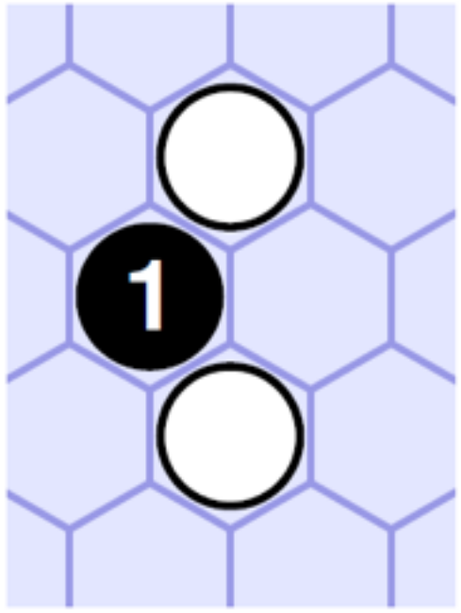
- PhD Candidate
- Digital Ludeme Project

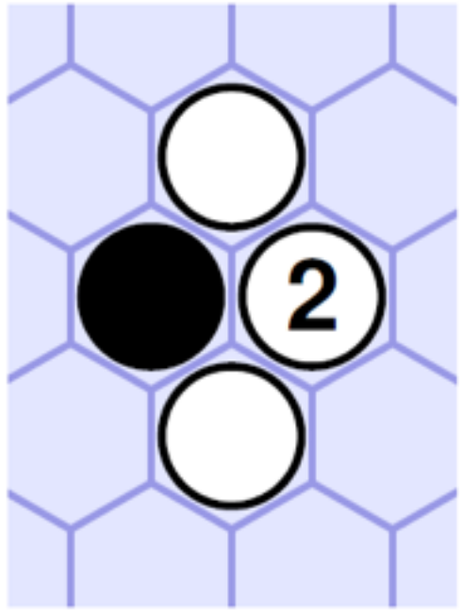
Maastricht University

Features

- Feature ≈ *description of action and parts of the state around it*

- Formally: *state-action* feature
  - Local
  - Binary



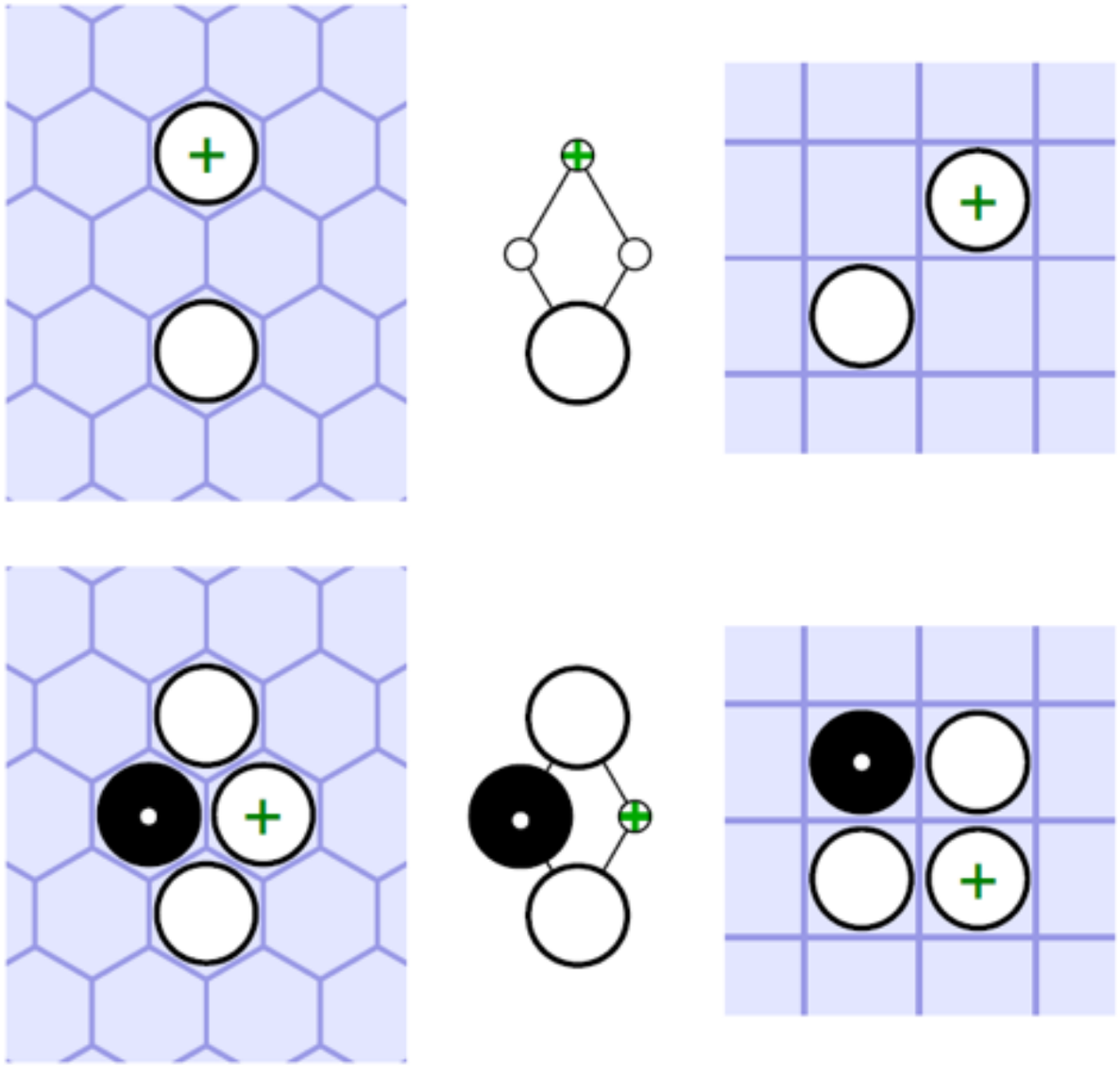


Uses of Features

- Bias play-outs in Monte Carlo Tree Search
  - More plausible play-outs
  - Better evaluations
  - Stronger agent
- May also be used in different ways
  - Bias in Selection phase of MCTS
  - Reinforcement Learning with function approximation
  - …

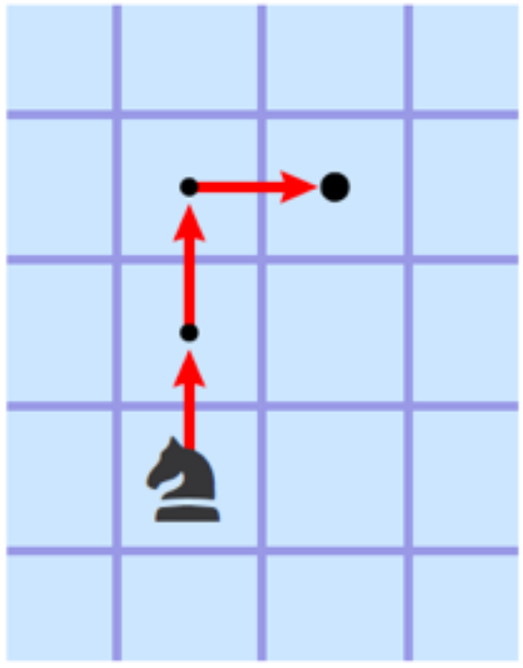
Geometry Independence

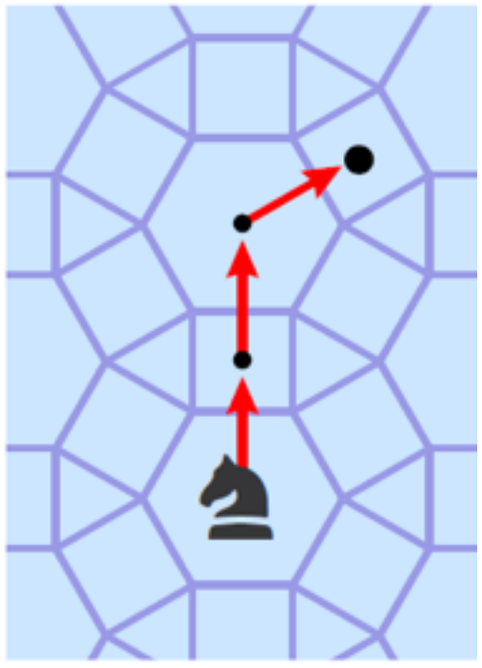
- Features should not be game- or board-specific
- Transfer between games / boards

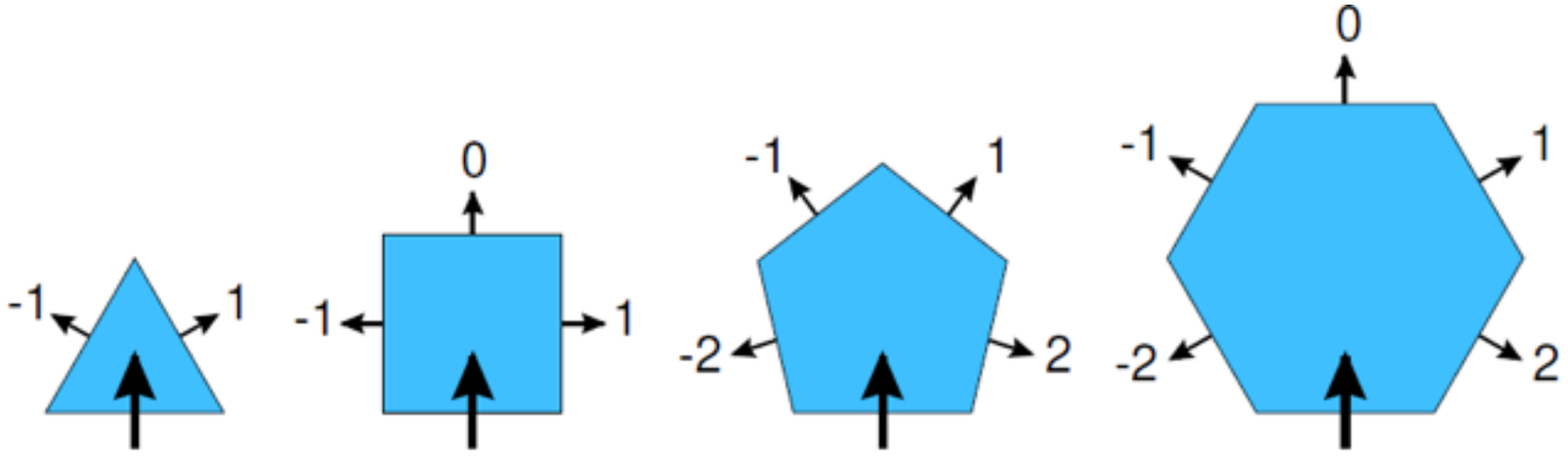


Geometry Independence

- Relative locations given by *Walks*

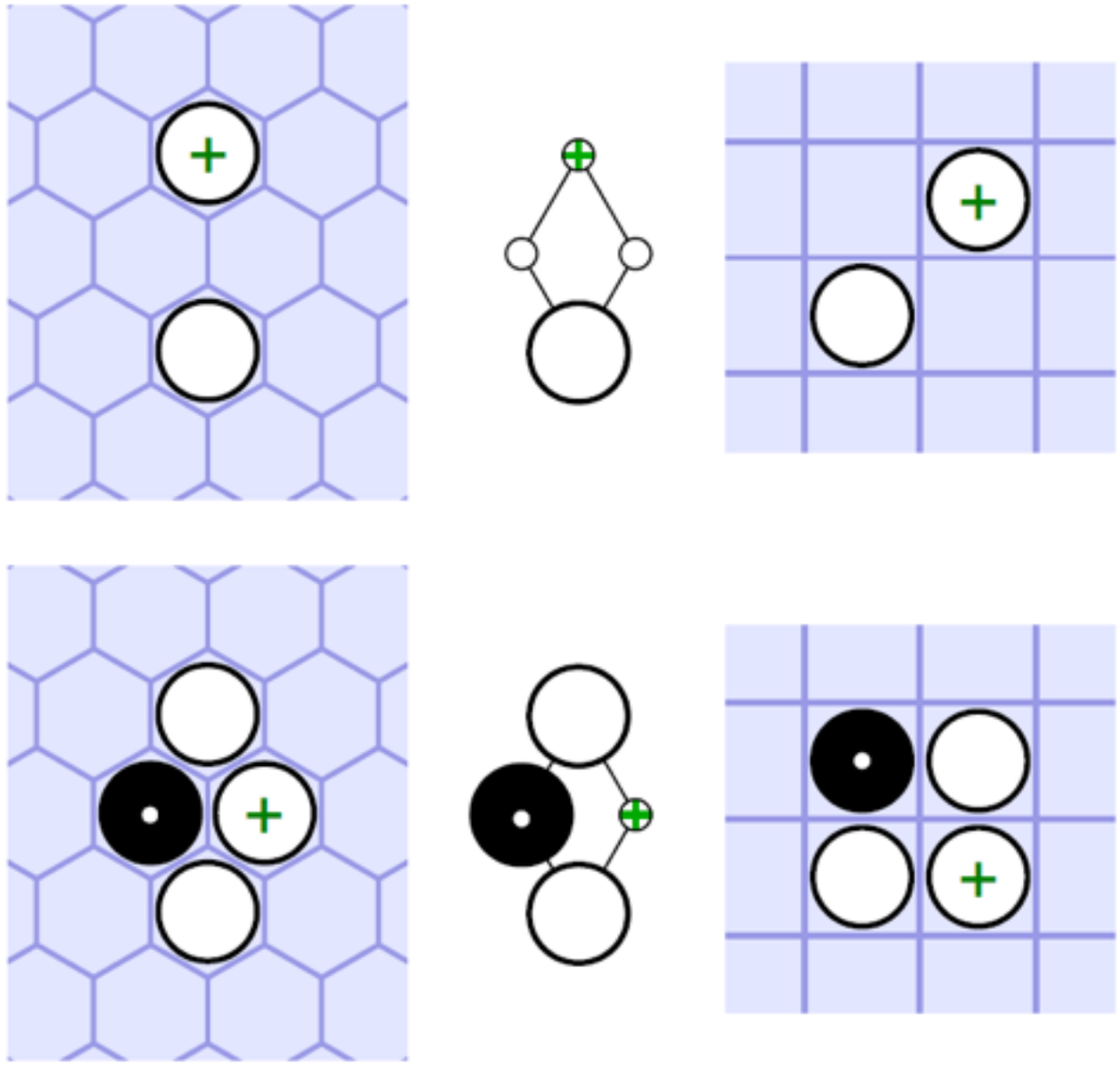






Reactive and Proactive Features

- Reactive features respond to last move
- Proactive features apply to entire board



Generating Features

- Handcrafted features
  - Not feasible for 1,000 different games

- Supervised learning from human games
  - Previously done in Go (1991), Go (2003), Go (2005), Go (2006), Go (2007), …
  - Still not feasible for 1,000 different games

Generating Features

- Exhaustively generate all features
  - LOTS of features
- Learning/mining from self-play games
- Evolutionary algorithms
- …

Generating Features

- Ludemes can help feature generation!
- Transfer useful features between game
- Automatically generate “minimum-required” patterns for legal moves
  - Reduces search space
  - Features *must* have empty position for recommended action in Hex, Tic Tac Toe, etc.

Understanding Games

- Features do not just improve the AI

- Features can provide insight into games and their relations:
  - Similar set of features → related games?
  - Similar rules, different features → can isolate rules responsible for strategies?
  - Features can be linked to ludemes responsible for their generation

Conclusion

Digital
Ludeme
Project

http://www.ludeme.eu

Thank You!
Questions?

Maastricht University

Go
China, 548BC
(Japanese players)