

# The Digital Ludeme Project: Modelling the Evolution of Traditional Games

Cameron Browne  
Walter Crist

Game AI & Search  
Maastricht University

Mapping Antiquity  
Fribourg University  
17/1/2020



Maastricht University

# Context

Games are ubiquitous

- All humans play games
- All human cultures have their own games
- Games reflect the culture(s) in which they're played

Games offer a window of insight into cultural past

Ancient games rarely recorded

- Passed on by oral tradition
- Huge variety today
- Very little knowledge





# Cultural Contact

Games are cultural artefacts

- Touchpoints between cultures
- Evidence of contact

e.g. Patolli and Pachisi



**Patolli** Mexico (200BC–1200AD)



**Pachisi** India (from 600-1600 AD)

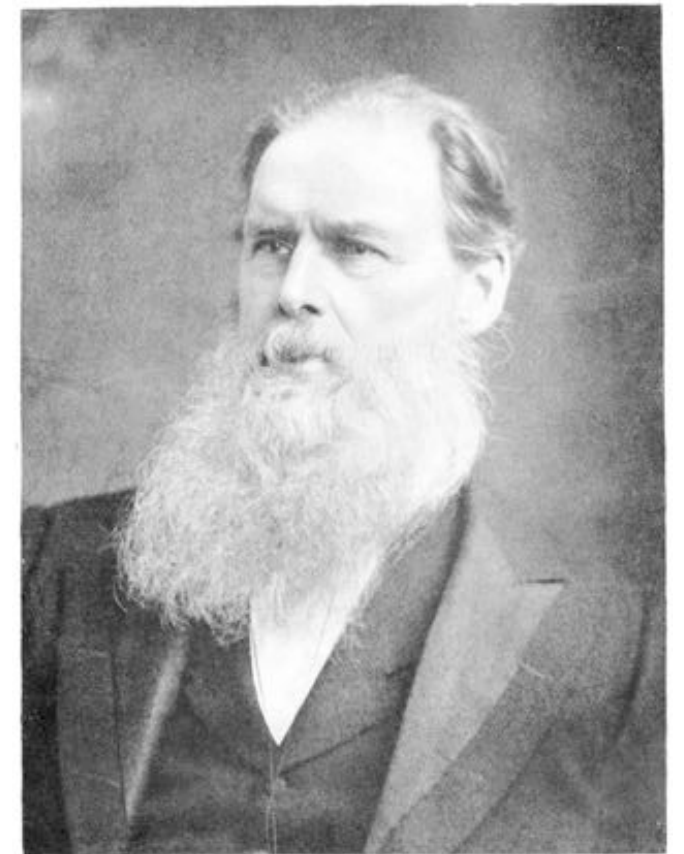
# Cultural Contact

Tyler (1879)

- Evidence of early pre-Columbian contact

Erasmus (1950)

- Coincidence, “Limitation of Possibilities”



E. B. Tyler (1832–1917)



**Patolli** Mexico (200BC–1200AD)



**Pachisi** India (from 600-1600 AD)



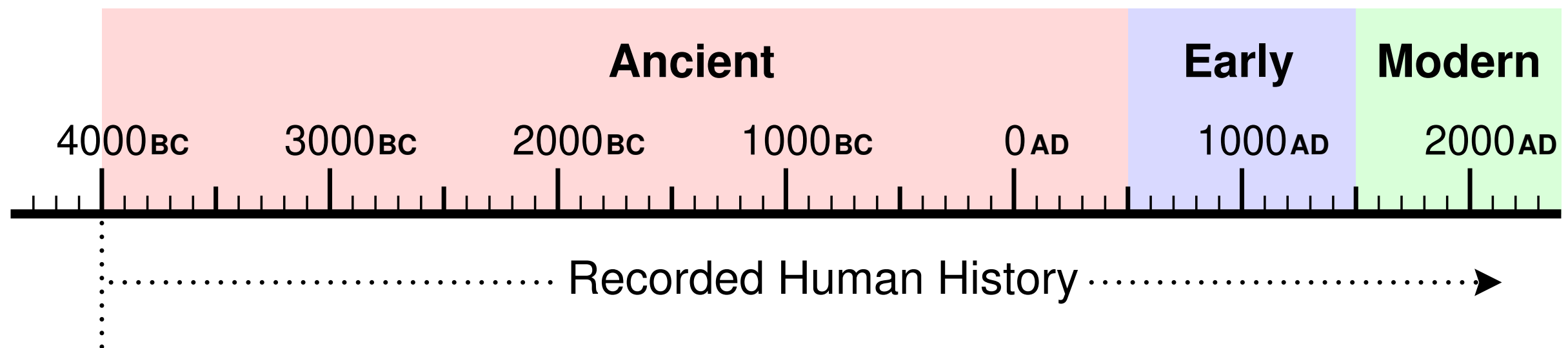
# Games in Human History

Archaeological evidence of games:

- Found worldwide
- Throughout recorded history

Provides clues:

- Reconstruction is a subjective endeavour
- Highly unreliable



# Digital Ludeme Project

Five-year research project

- Funded by the ERC (€2m)
- Maastricht University

Computational study of the world's traditional games

Games as mathematical entities

- Evidence based
- Quantitative approach



European  
Research  
Council





# Objectives

## 1. **Model**

Full range of traditional strategy games  
in a single playable digital database

## 2. **Reconstruct**

Missing knowledge about ancient games more reliably

## 3. **Map**

Spread of games throughout history

**Aim:** To improve our understanding of traditional games  
using available evidence and modern AI techniques



# Team



**Cameron Browne (PI)**

- Game AI (technical lead)



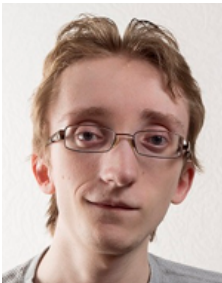
**Eric Piette (Postdoctoral Researcher)**

- Game AI (game engine development)



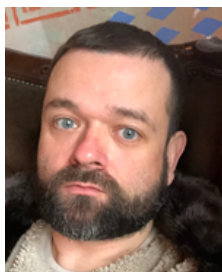
**Matthew Stephenson (Postdoctoral Researcher)**

- Game AI (GUI, networking, data mining)



**Dennis Soemers (PhD Candidate)**

- Game AI (feature learning)



**Walter Crist (Postdoctoral Researcher)**

- Anthropologist/archaeologist (dispersal of games)
- Middle East and Mediterranean specialist



# Scope

## Traditional games of strategy

### *Traditional*

- No proprietary owner
- Some historical longevity
- Connection with local culture

### *Strategy*

- Reward mental skill
- Good decisions beat bad decisions  
e.g. board, tile, card, dice, *etc.*

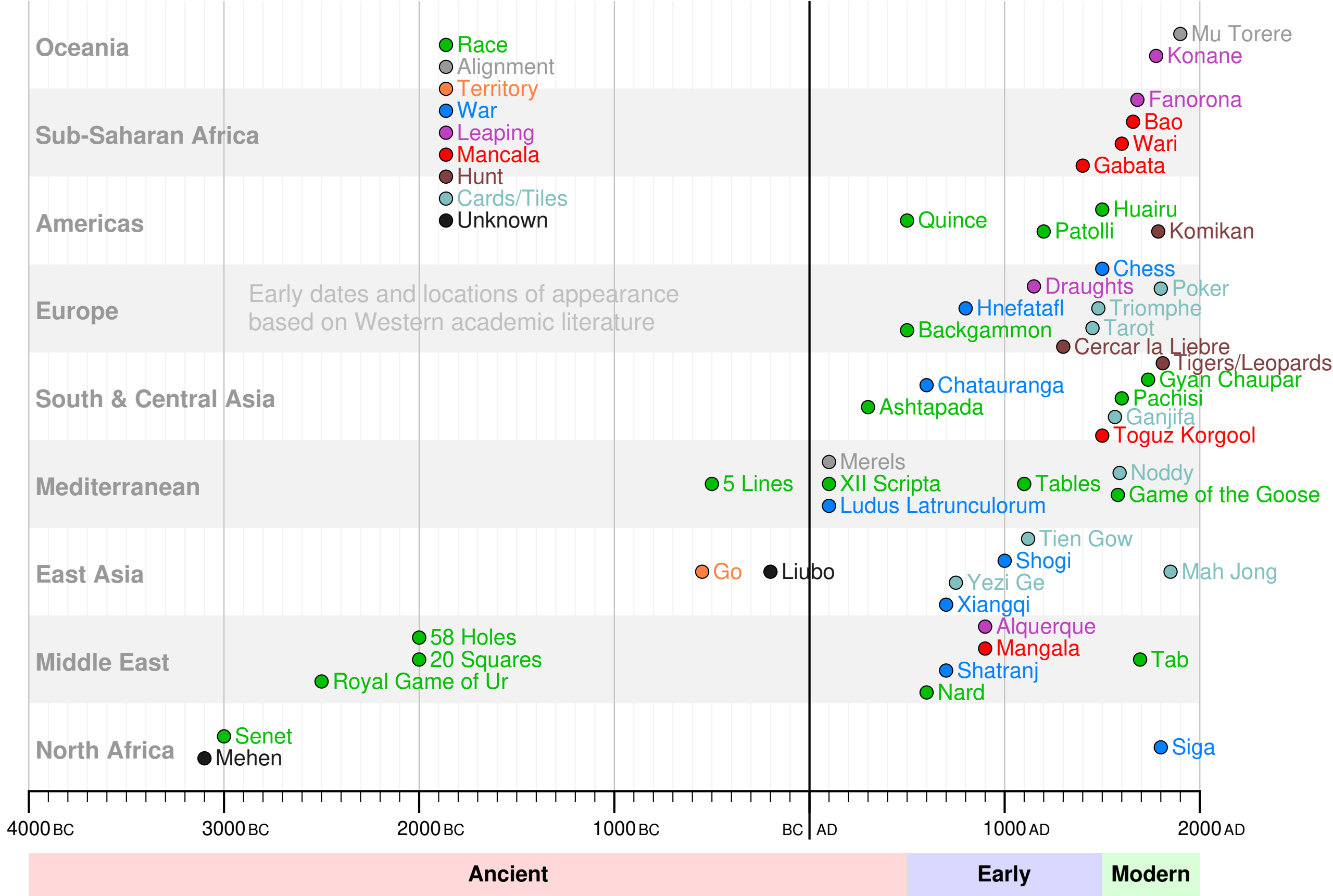
Model the 1,000 most “important” traditional games

- Documented, can be located and dated
- Impact on evolutionary record



**XII Scripta** board from Laodicea, Turkey

# Timeline





# Games as Cultural Heritage

## Tangible cultural heritage

- Physical evidence  
e.g. boards, pieces, components, *etc.*

## Intangible cultural heritage

- Rules
- Social/cultural context
- Restore and preserve

**What is the available evidence?**

# Ancient Egypt

Earliest known games

## Mehen (c.3100BC)

- No rules
- No clues to how it was played



## Senet (c.3000BC)

- Hundreds of sets found
- No rules
- Hieroglyphic art





# Senet

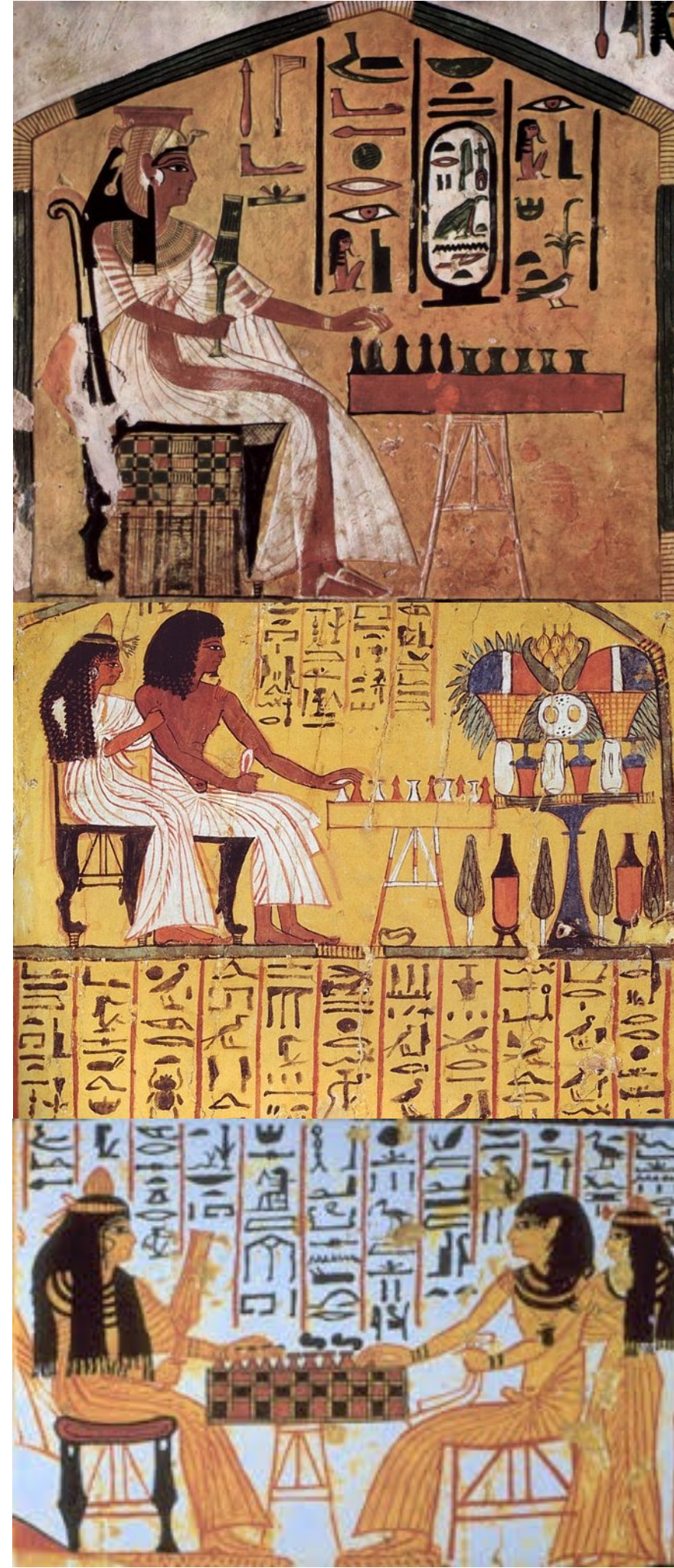
From hieroglyphs

- Two players
- Some starting positions

From evidence

- Two piece types
- Special symbols on board
  - Entry points? Exit points?

About a dozen plausible reconstructions





# First Known Rules

Sumerian cuneiform tablets

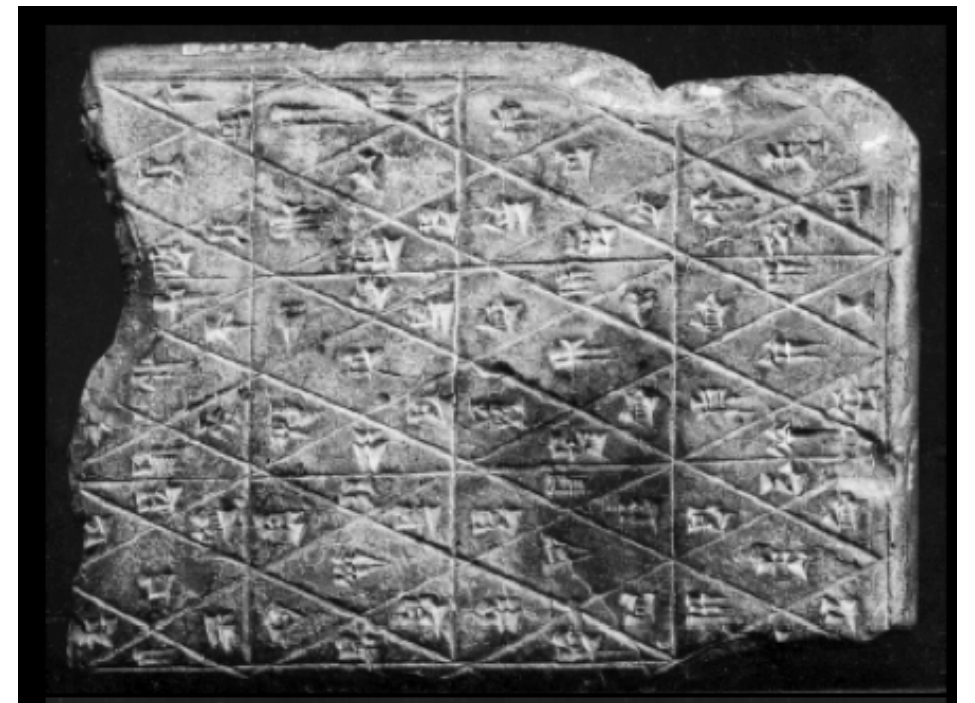
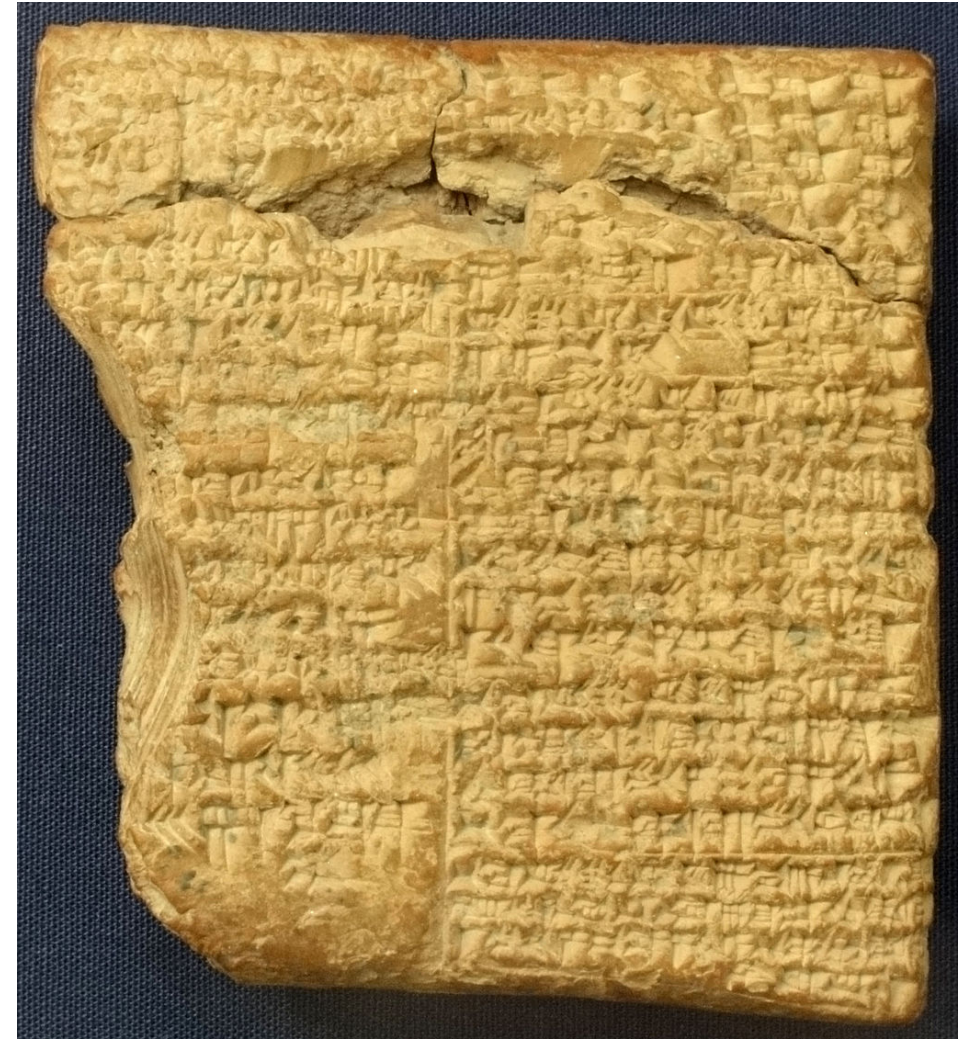
- Mesopotamia, 177BC

British Museum (top)

- One of 130,000

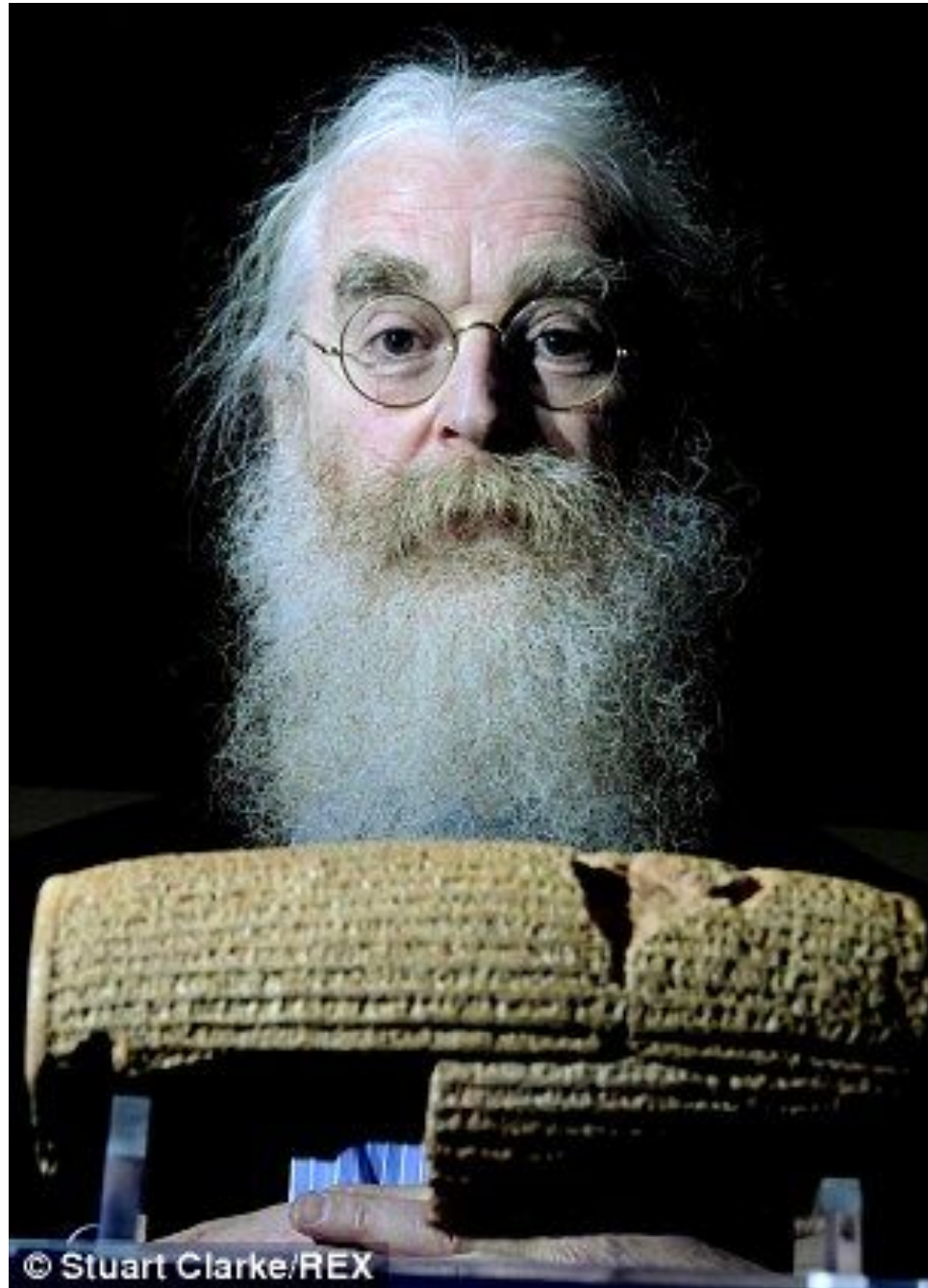
Parisian (bottom)

- Destroyed 1940s
- Photo survived

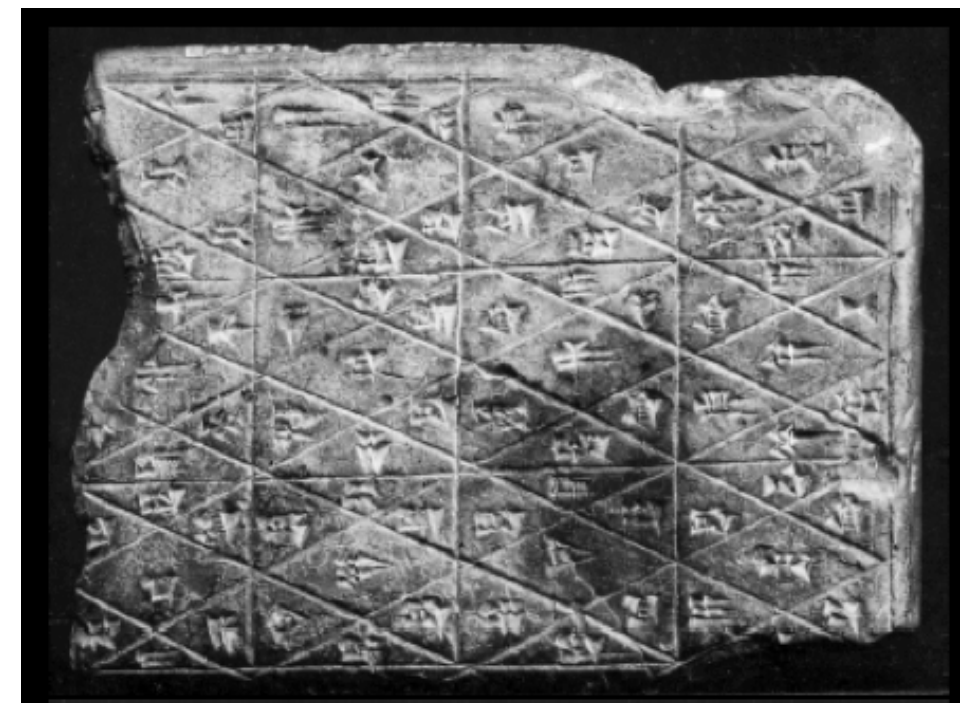
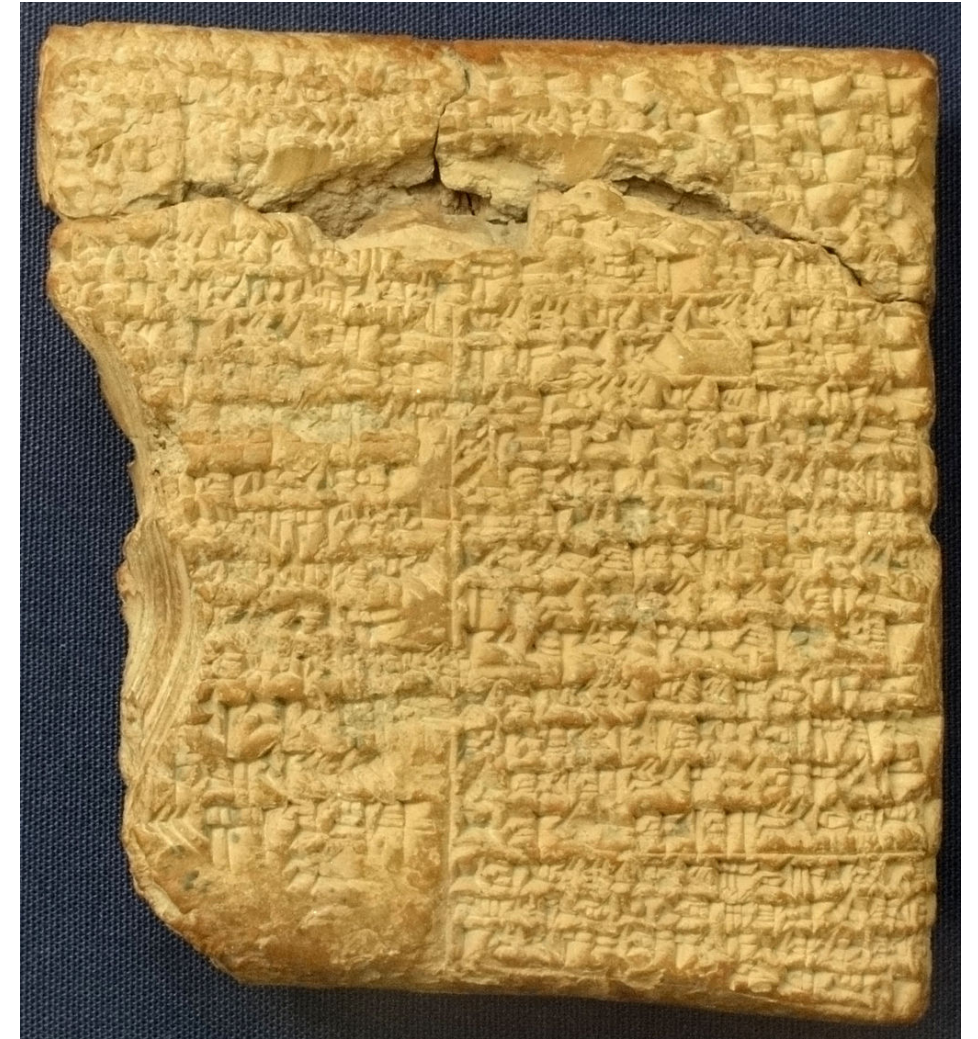




# First Known Rules



**Irving Finkel (1990)**  
Curator, British Museum





# Royal Game of Ur

Played in Mesopotamia

- c.2600BC

Tablets written

- c.177BC

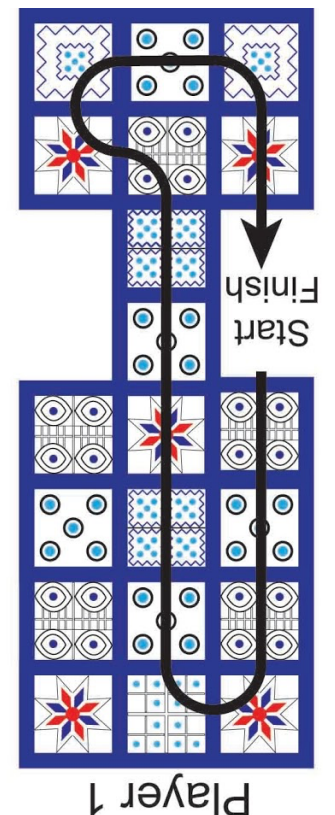
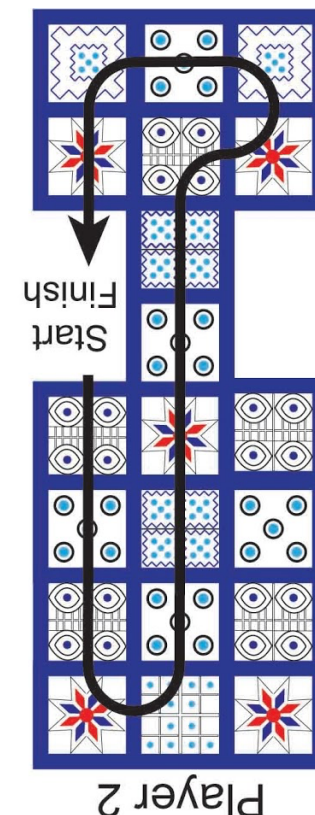
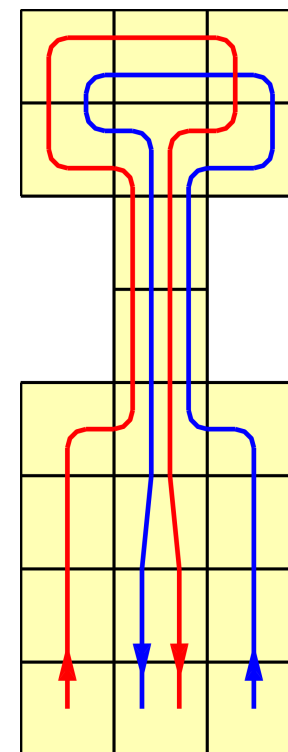
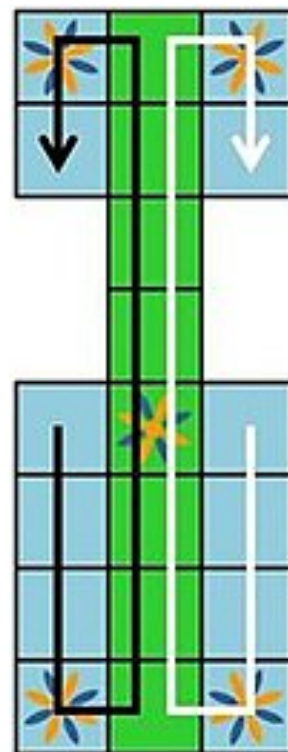


Reconstruction by Irving

- 1990

Still some questions

- Same game? Probably
- Which track?



# Transcription Errors

Mu Torere (New Zealand, 18<sup>th</sup>C)

- Full knowledge
- Living players

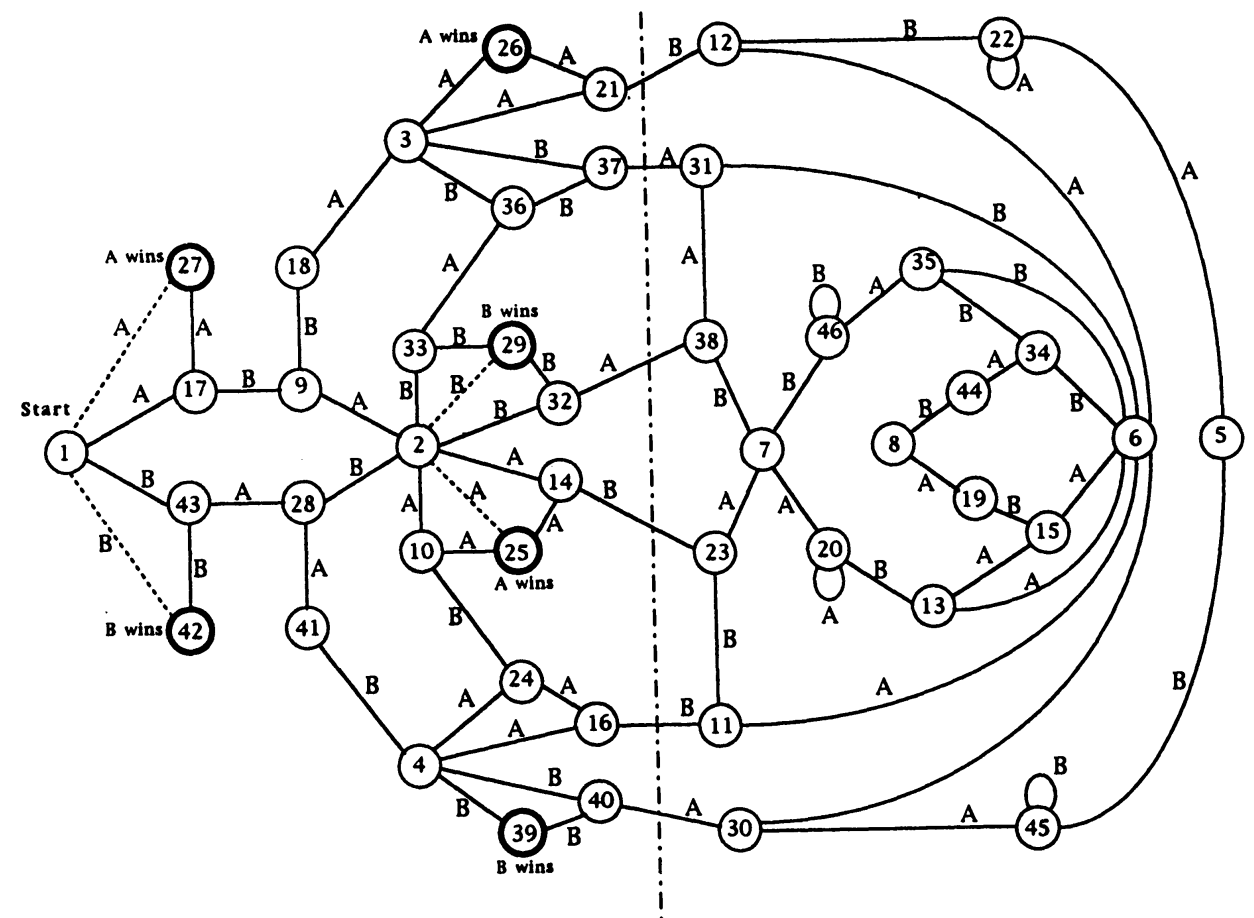
*Move a piece of your colour adjacent to an enemy piece to the adjacent empty point.*



Some accounts simplify this:

*Move a piece of your colour to the adjacent empty point.*

Win on first move!





# Invented Traditions

**Birrguu Matya** (Australian Aboriginal, late 19<sup>th</sup>C)

Marketed as traditional game

Identical to Small Merels

Is a clear outlier

- No other strategy board games in this culture

Meggitt (1958)

- Afghani camel herders
- German missionary





# Invented Traditions

## Surakarta

- Named after traditional Javanese capital
- National game of Java?

But...

- Can't find anyone from Java who knows it!
- Game invented for Ravensburger 1972 edition?



# Approach

How to handle this incomplete, unreliable evidence?

1. Quantify where possible
2. Encode in single consistent format
3. Establish historical/cultural context
4. Find relationships within data

# Ludemes

Game “memes” (Borvo, 1975)

- Units of game-related information
- Building blocks (DNA) of games
- Encapsulate key concepts (in consistent format)

e.g. (tiling square)

(size 3)



# Ludemes

## Game “memes”

- Units of game-related information
- Building blocks (DNA) of games
- Encapsulate key concepts

e.g.

```
(tiling square)
```

```
(size 3)
```

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

# Ludemes

## Game “memes”

- Units of game-related information
- Building blocks (DNA) of games
- Encapsulate key concepts

e.g.

```
(tiling square)
```

```
(size 3)
```

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

```
(game "?"  
  (players White Black)  
  (board  
    (tiling square)  
    (size 3)  
  )  
  (move (add Own Empty))  
  (end (win All (in-a-row 3)))  
)
```

# Ludemes

## Game “memes”

- Units of game-related information
- Building blocks (DNA) of games
- Encapsulate key concepts

e.g.

```
(tiling square)
```

```
(size 3)
```

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

```
(game "Tic-Tac-Toe"  
  (players White Black)  
  (board  
    (tiling square)  
    (size 3)  
  )  
  (move (add Own Empty))  
  (end (win All (in-a-row 3)))  
)
```



# Stanford GDL

## Academic standard

- 15 years

## Programmer's view

- Low level instructions
- Not high level concepts

```
(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 vs GDL

```
(game "Tic-Tac-Toe"
  (players White Black)
  (board
    (tiling square)
    (size 3)
  )
  (move (add Own Empty))
  (end (win All (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))
```

# Ludemes vs GDL

```
(game "Tic-Tac-Toe"  
  (players White Black)  
  (board  
    (tiling square)  
    (size 7)  
  )  
  (move (add Own Empty))  
  (end (win All (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))
```

# Ludemes vs GDL

```
(game "Tic-Tac-Toe"  
  (players White Black)  
  (board  
    (tiling hexagonal)  
    (size 7)  
  )  
  (move (add Own Empty))  
  (end (win All (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))
```



# Ludemes vs GDL

```
(game "Tic-Tac-Toe"  
  (players White Black)  
  (board  
    (tiling hexagonal)  
    (size 7)  
  )  
  (move (add Own Empty))  
  (end (win All (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))
```

# Ludemes vs GDL

```
(game "Tic-Tac-Toe"  
  (players White Black)  
  (board  
    (tiling hexagonal)  
    (size 7)  
  )  
  (move (add Own Empty))  
  (end (win All (no-moves)))  
)
```

## Designer's view

- Encapsulates high level concepts
- Full range of games



```
(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))
```

# How Many Ludemes?

Do we have to implement them all?

- Most of them

About 400 so far

- About 600 expected
- Not actually that many
- High reuse among games

Very achievable!

System is fully extensible

- Just add more as needed

# How To Improve Reconstructions?

Search for alternative rule sets that maximise:

**1. Historical Authenticity**

**2. Game Quality**





# How To Improve Reconstructions?

Search for alternative rule sets that maximise:

## 1. Historical Authenticity

- Rules match: *location, period, cultural context*
- Based on historical data

## 2. Game Quality

- Run self-play trials between AI agents
- Look for obvious flaws
- Look for indications of quality

# Obvious Flaws

Basic indicators of bad games:

## 1. Bias

- All players should have chance of winning

## 2. Drawishness

- Most games should produce a result, not a draw

## 3. Game Length

- Games shouldn't be too short or too long

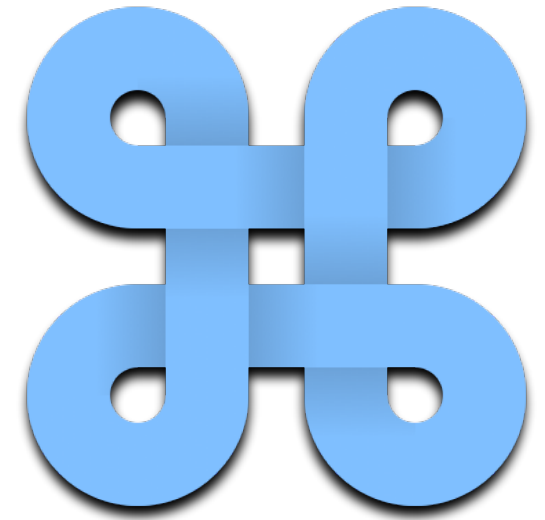
Easy to detect, can eliminate to narrow down choices



# Ludii

# General game system

- Playing, analysing, designing, reconstructing



## Early stages

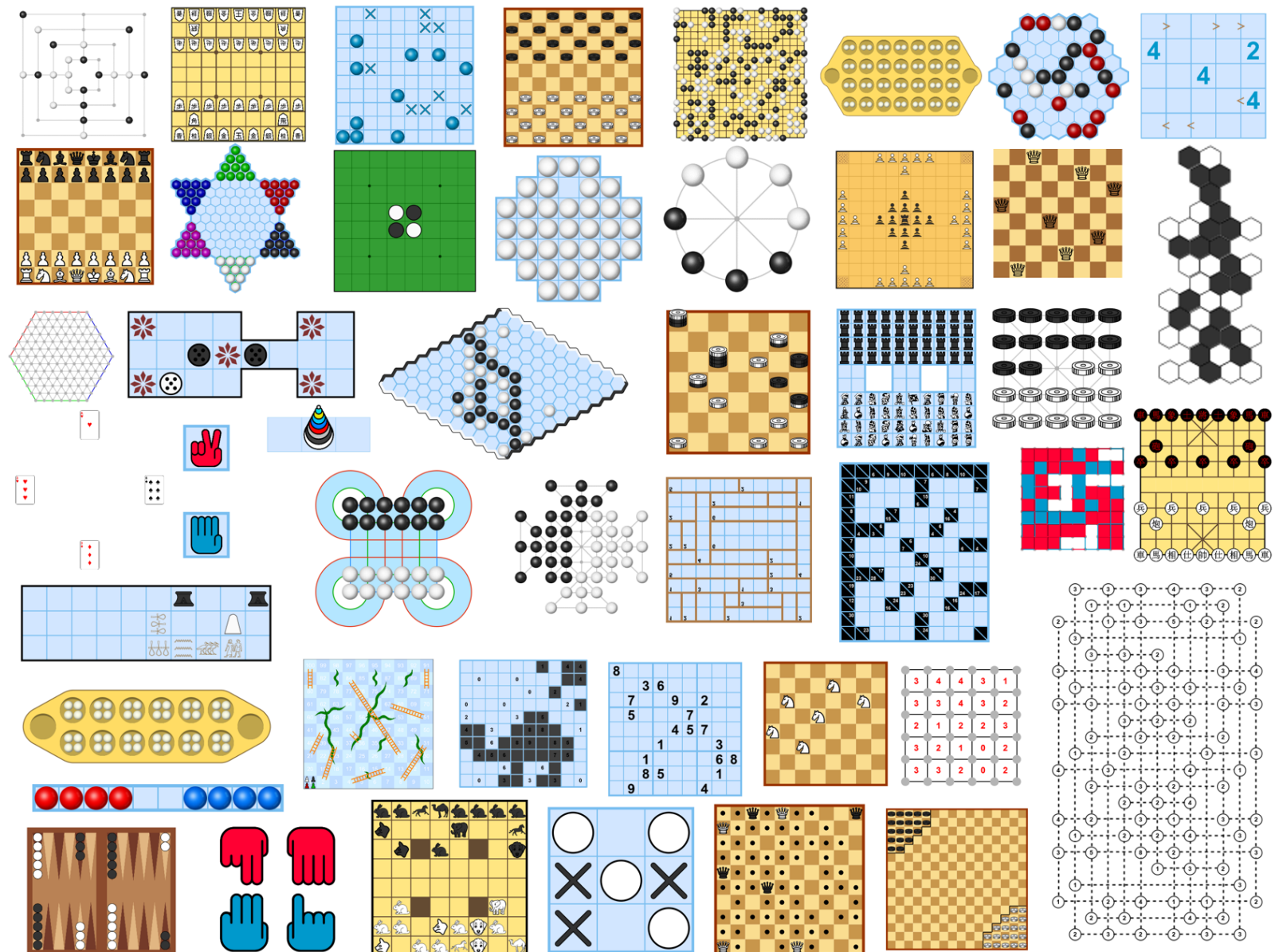
- ~200 games

# Beta version available

- <http://ludii.games>

# Official release

- Soon!
- January/February





# Example: Hnefatafl

Hnefatafl “Viking Chess”

- Scandinavia (c.800AD)
- No rules found

Linnaeus (1732)

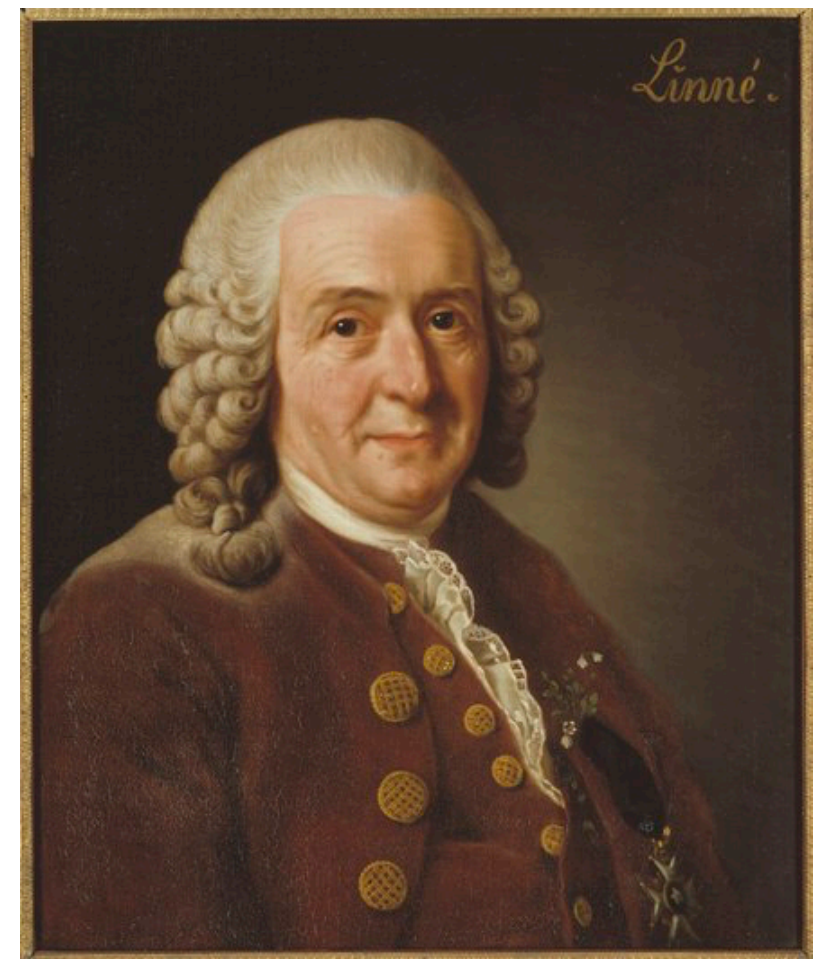
- Saw Tablut, transcribed rules (in Latin)

Smith (1811)

- Translated into English

Murray (1913) *History of Chess*

- Published rules, became de facto



Carl Linnaeus (1707-1778)

# Example: Hnefatafl

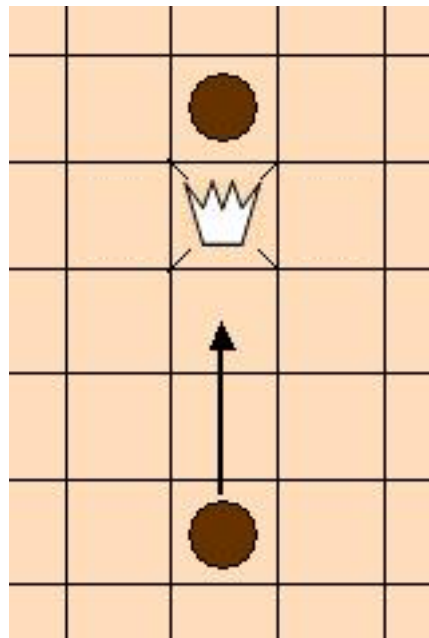
BUT...

Smith made a bad translation of the king capture rule



A. “likewise the king”

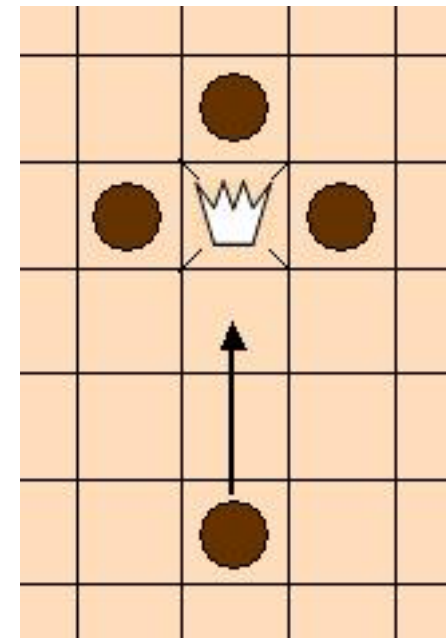
- Flanked
- Easy to capture



B. “except the king”

- Surrounded
- Hard to capture

[DEMO]





# Upper Bound

## Taikyoku Shogi

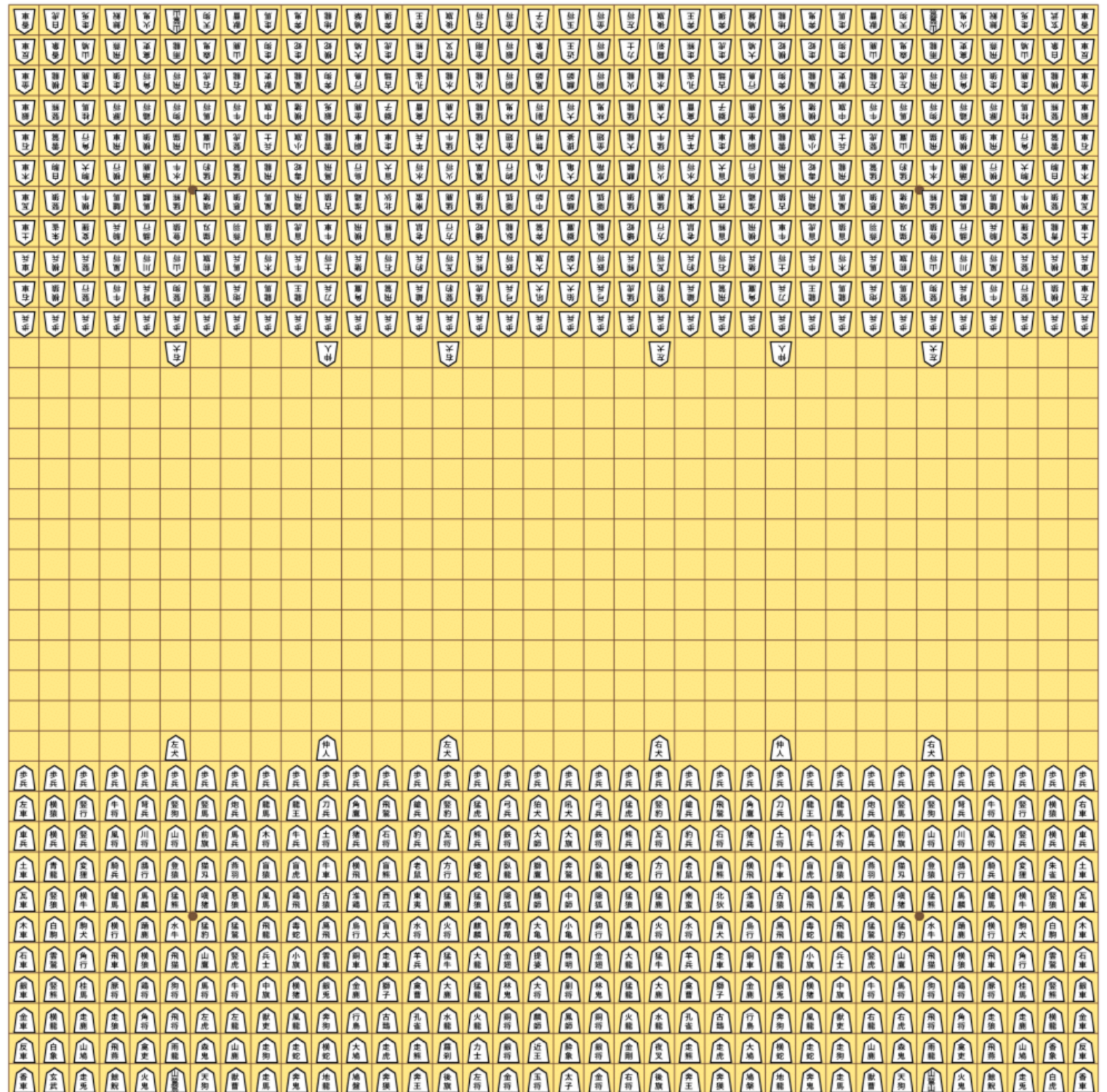
- c.1600
- At least two sets exist
- Has actually been played

## Equipment

- 36x36 board
- 402 pieces each
- > 200 different piece types with distinct moves

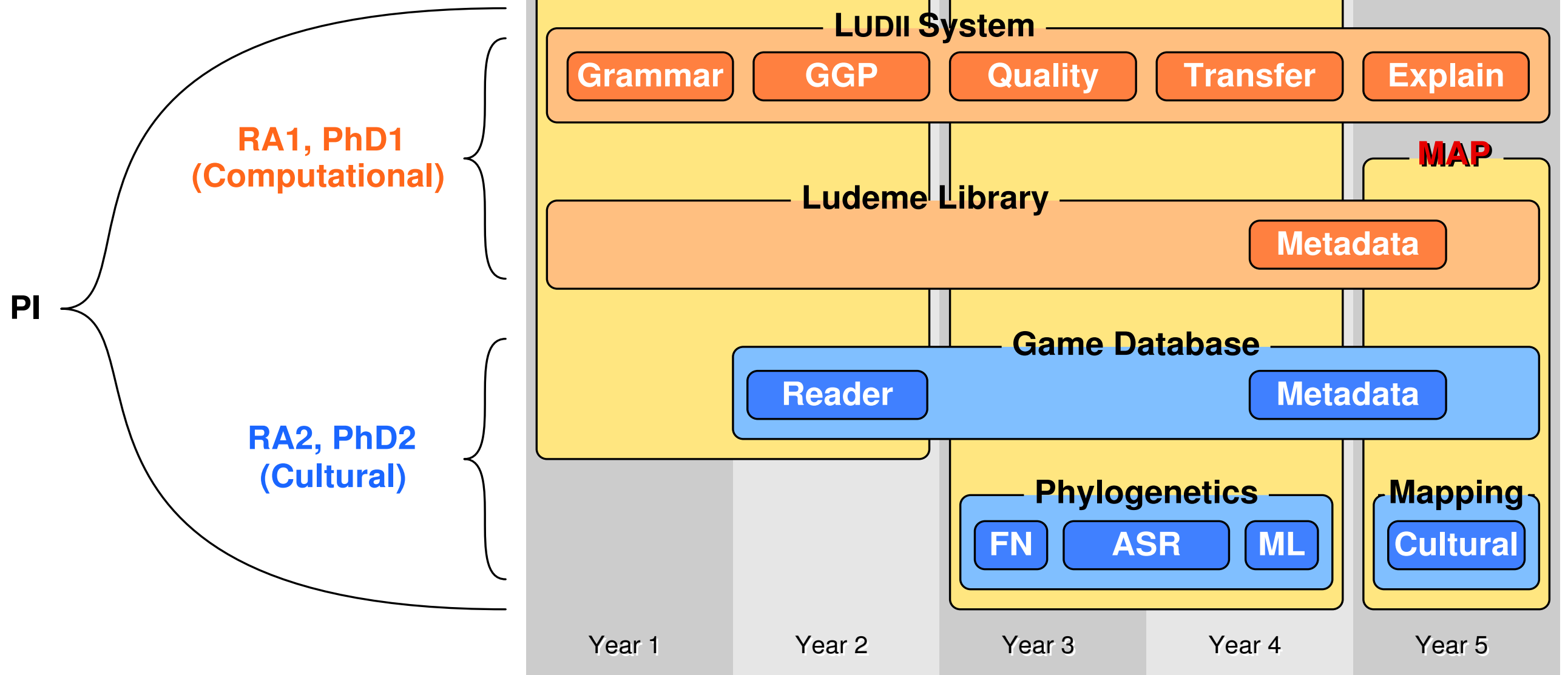
## Complexity

- ~800 legal moves per turn
- ~1,000 turns per game
- Game tree >  $10^{1,000}$   
(Go is  $\sim 10^{360}$ )





# Plan



## 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

# Cultural Transmission Theory

## Games

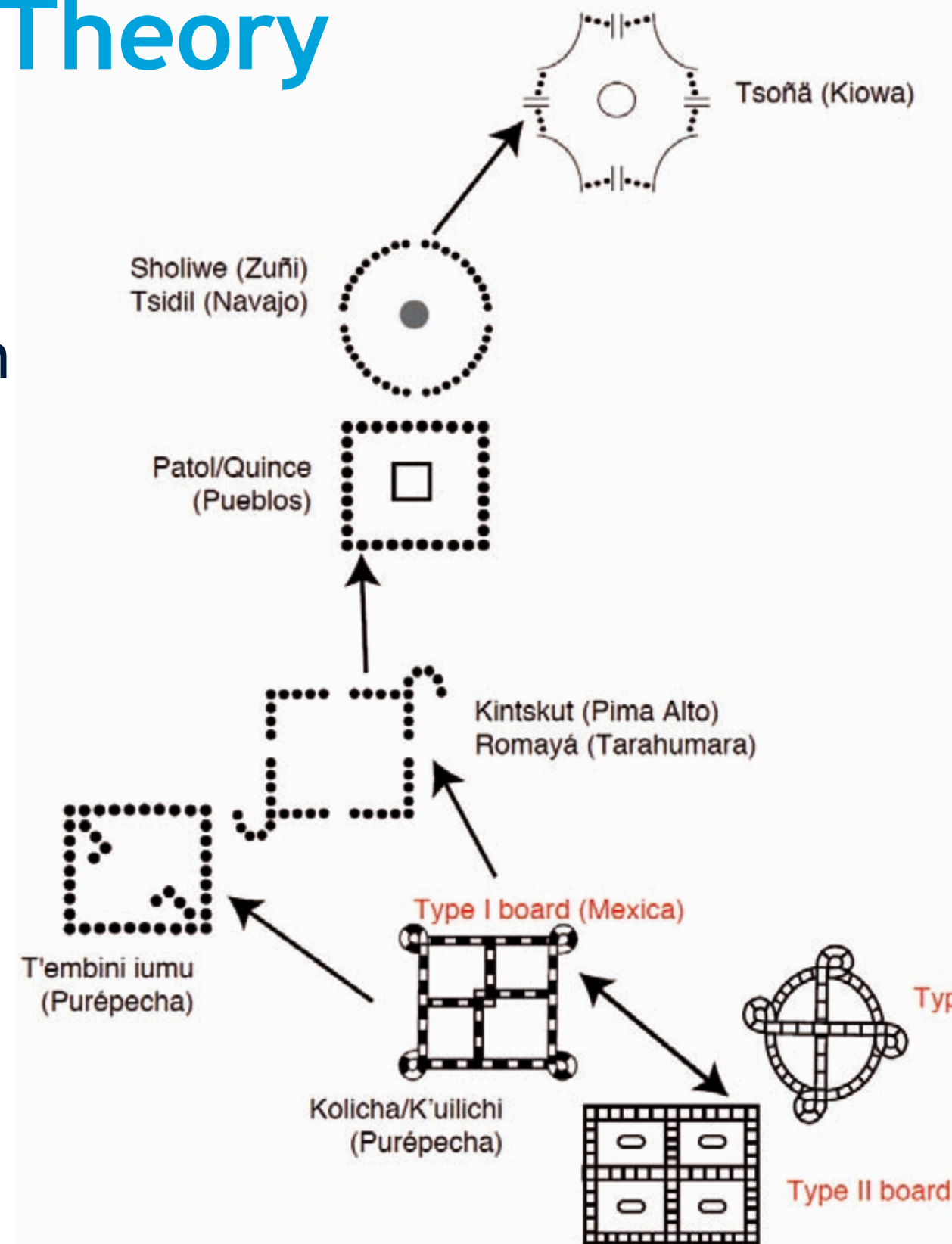
- Vehicles for transmission of ideas
- Transferred from person to person

## Ludemes

- Packages which are transmitted
- DNA of games

## Ludemic Distance

- Number of steps required to change one game into another



# Ludemic Distance

## Genotypic Distance

**Form** - Measured directly from game description:

- Edit distance between rule sets
- Geometric distance between equipment
- Game tree and state space complexity
- Types of actions
- etc.

## Phenotypic Distance

**Function** - Trends that emerge during play:

- Balance
- Drama
- Tension
- Uncertainty
- Strategic depth
- etc.



# Computational Phylogenetics

Ludemic distance:

- In lieu of actual genetic distance

## 1. Family Trees

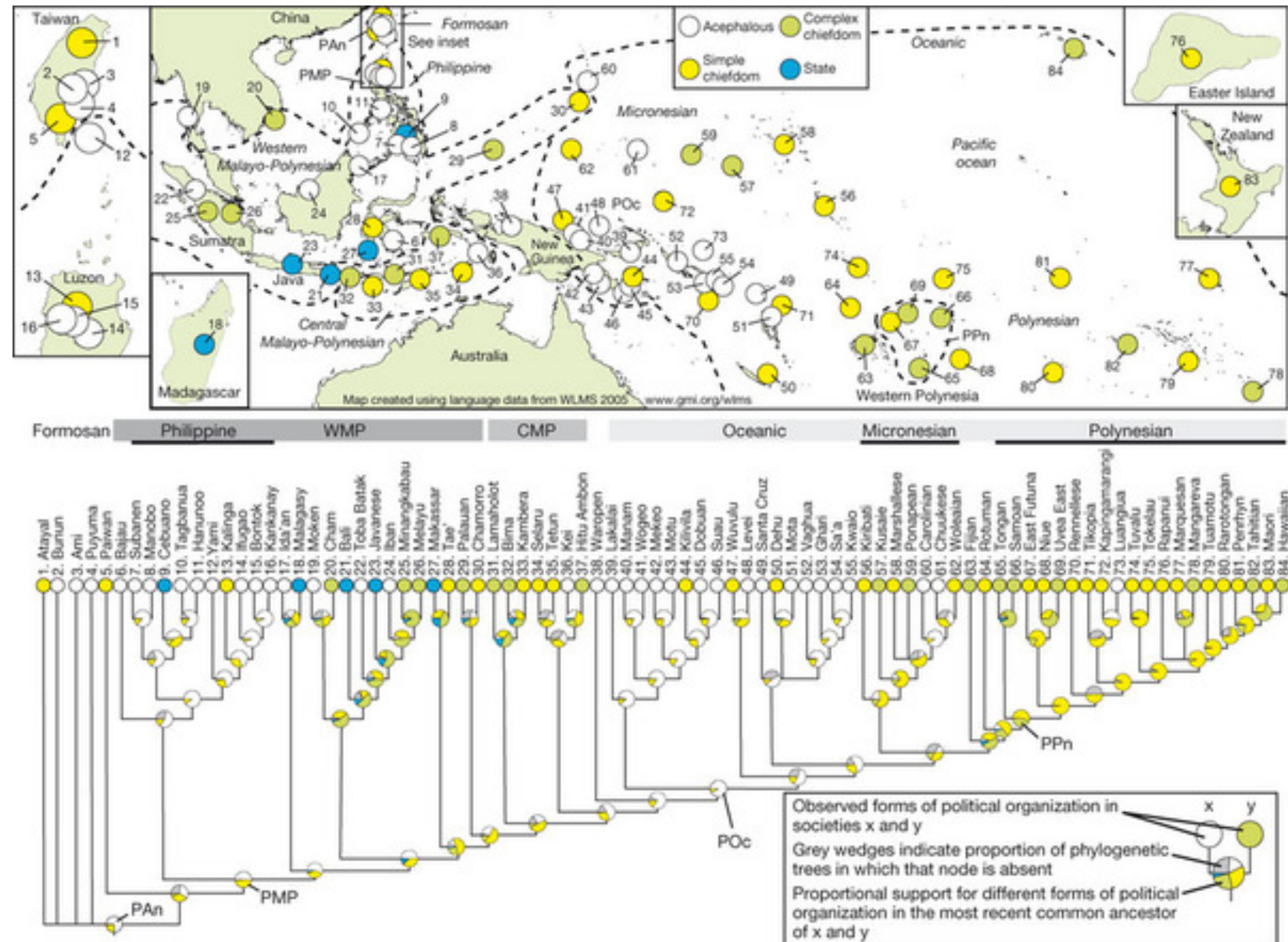
- Key game families

## 2. Ancestral State Reconstruction

- Identify likely traits in ancestors

## 3. Missing Links

- Games that explain gaps in the evolutionary record?



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



# Dispersal of Mathematical Ideas?

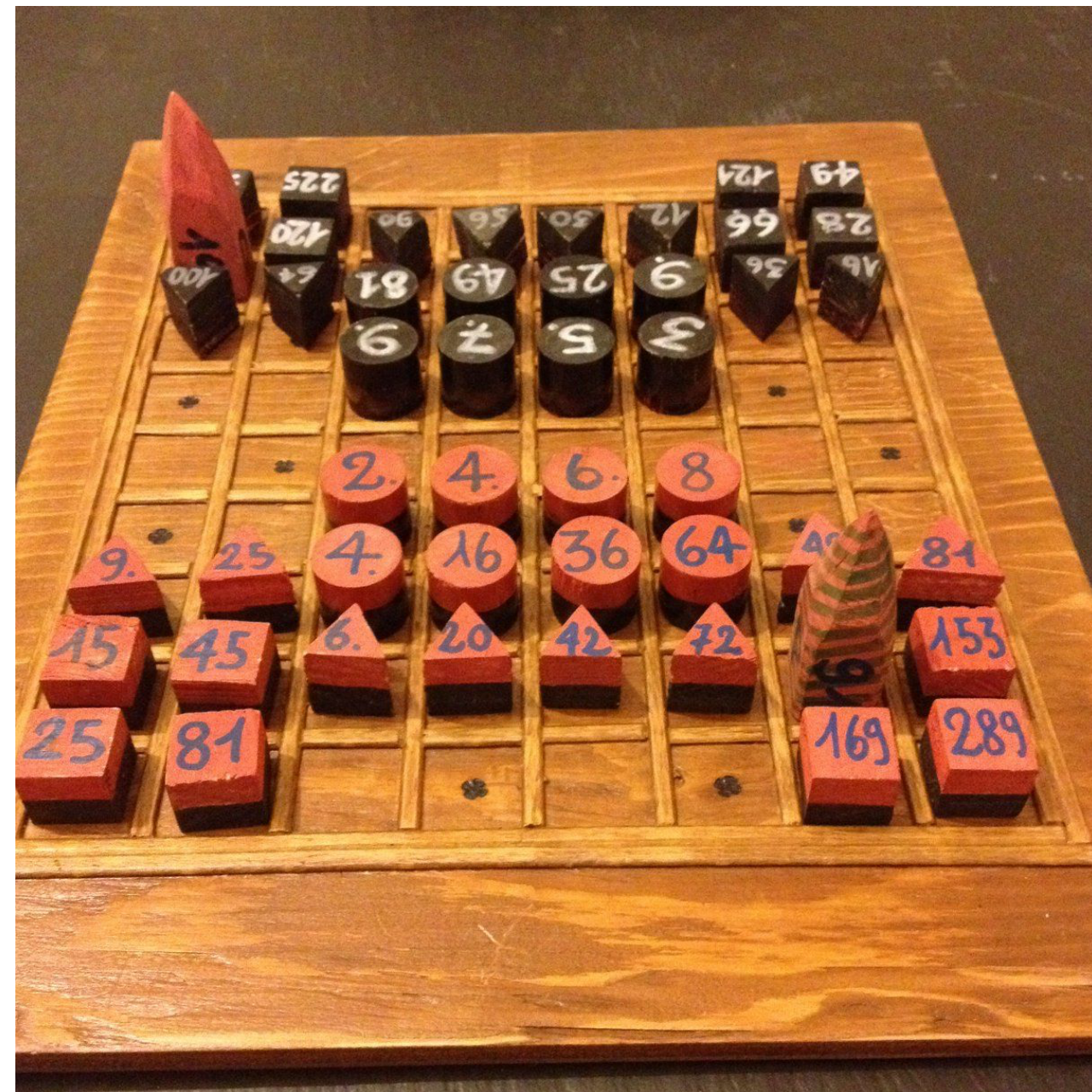
Chart the spread of:

- games
- underlying mathematical concepts throughout history

Tag each ludeme with mathematical concepts

Gives *mathematical profile* for each game

*All* games embody underlying mathematical concepts!



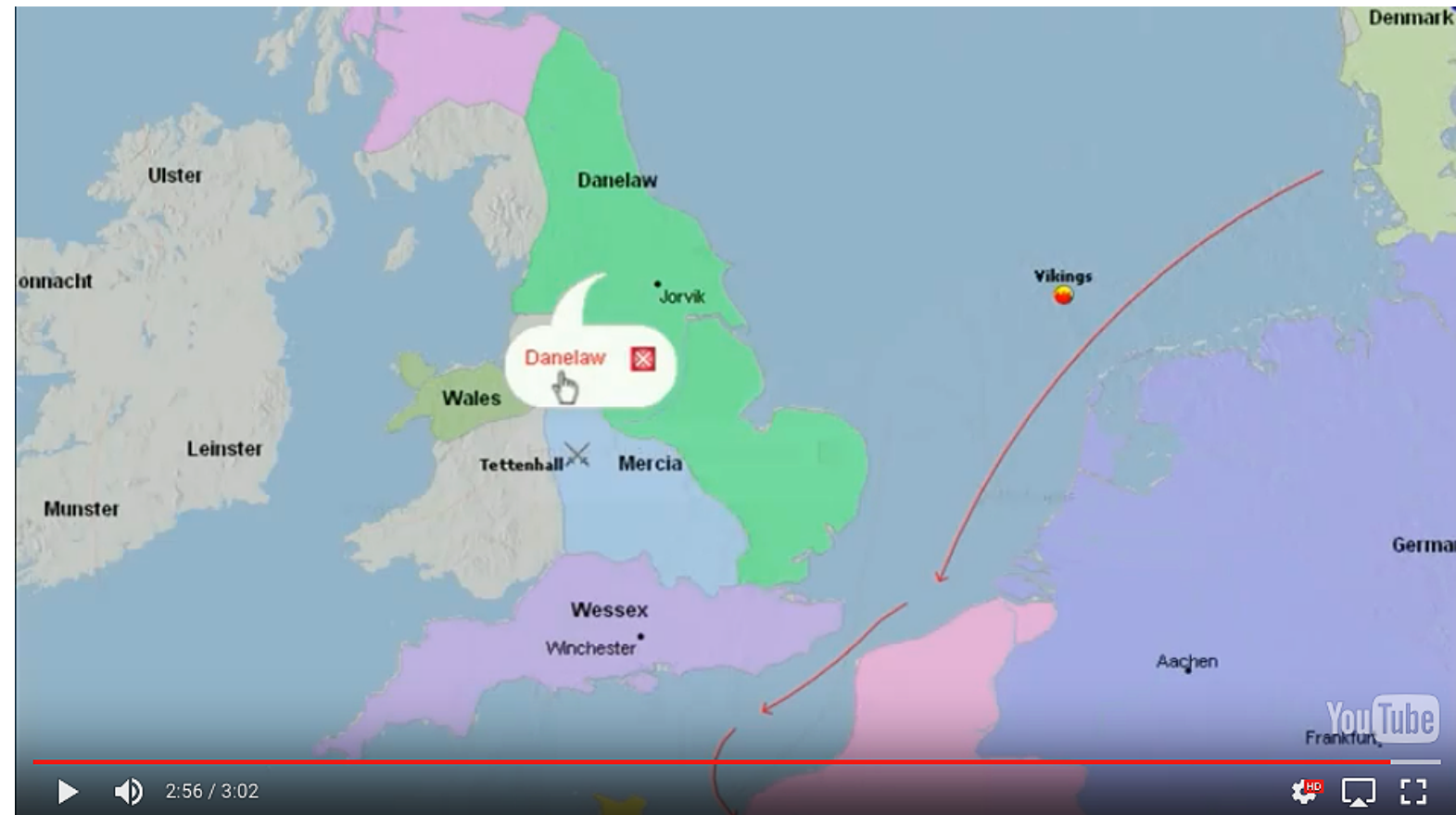
# GeaCron

Geo-temporal database

- Project partners

Yearly maps:

- Last 5,000 years
- 2,000 cultures



Viking route from Norway to Paris (845AD)

**IN:** GPS + date

**OUT:** Civilisation + nation + landmarks + events + routes



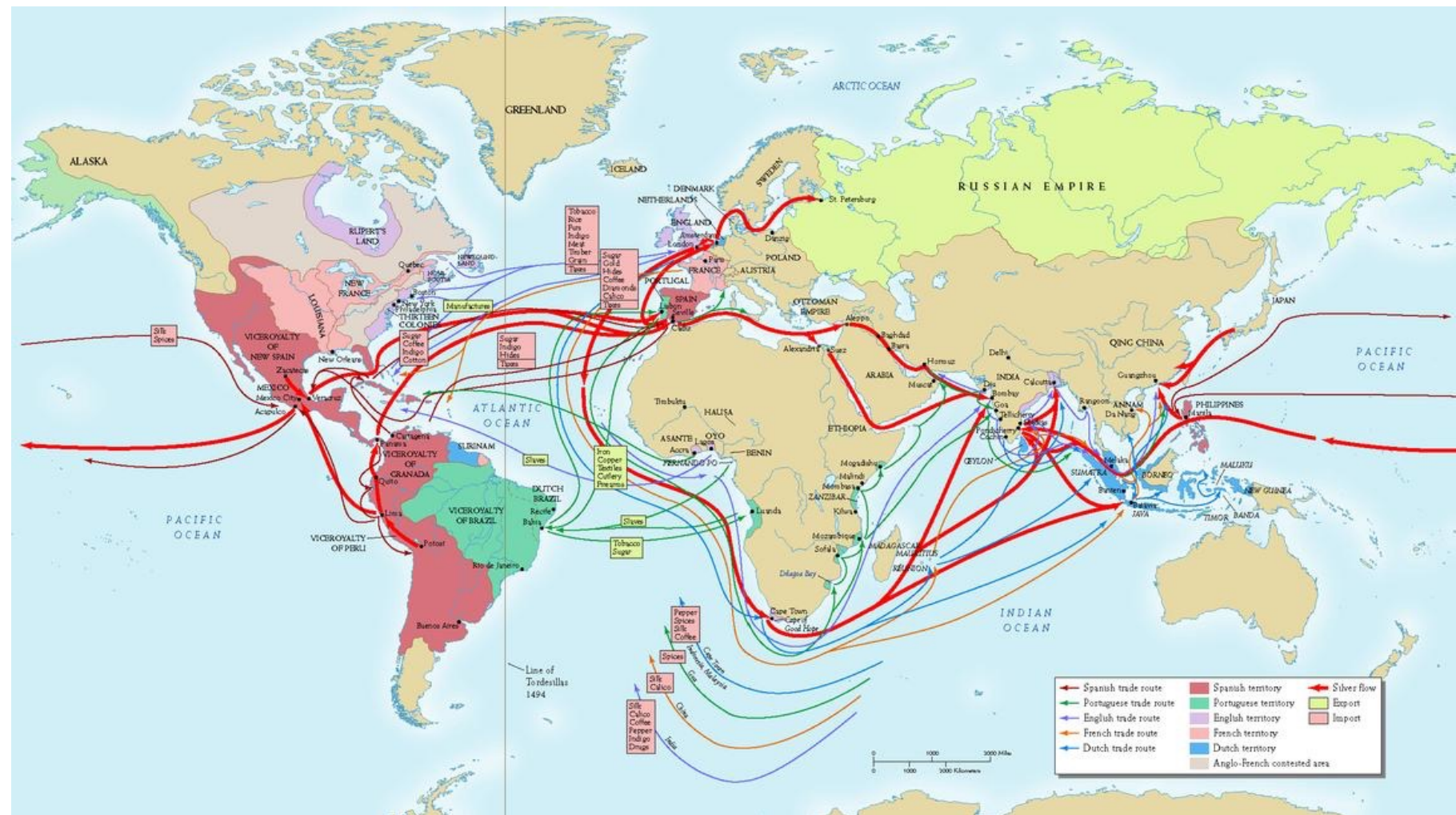
# Spread of Games

Chart spread of games/ludemes throughout human history

Correlate with:

- Trade routes
- Explorer routes
- Military camp'ns
- Crusades
- Diasporas
- *etc.*

GeaCron have provided 275 known routes



Colonial trade routes (1890s)



# Silk Road Trade Routes

Very important in the history of games

- Fertile crescent:
  - Egypt
  - Sumer
- Middle East
- India
- Asia



Map 12.1 The silk roads,



# Data Gathering

Sources:

- Artefacts
- Rules texts
- Artwork depictions
- Ethnography
- Historical accounts
- Field trips



Walter documenting a **58 Holes** board (Azerbaijan)



Two men playing **Bao** (Malawi)



# Data to be Gathered

Data per piece of evidence:

- Name
- Location
- Date
- Game rules
- Social status
- Gender of players
- Age of players
- Spaces in which people played
- Source
- **Confidence**



**Ludus Latrunculorum** board (Sabratha)

# Data to be Gathered

Data per game:

- Name
- Descriptions of variants and reconstructions (for \*.lud files)
- Period
- Region
- Cross-reference known evidence
- Source(s)
- **Confidence** (incorporates evidential confidence)

Historical/cultural profile per game

- Also for component ludemes

The screenshot shows a web-based database entry form. At the top is a navigation bar with tabs: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, and Operation. Below this is a table with columns: Column, Type, Function, Null, and Value. The table contains several rows for different fields, each with a dropdown menu for selection and a text input for values. The fields are: id (int(11)), Typeid (int(11)), Gameid (int(11)), Variationid (int(11)), Sourceid (int(11)), DateStart (date), DateStartBC (tinyint(1)), DateEnd (date), DateEndBC (tinyint(1)), Location (text), KML (varchar(45)), Rules (text), Misc (text), and ConfidenceMultiplier (int(11)). The 'Rules' field has a large text area with a preview of the rules text. The 'Misc' field has a large text area with a preview of the miscellaneous text. The 'ConfidenceMultiplier' field has a dropdown menu and a text input. A 'Go' button is at the bottom right.

Column	Type	Function	Null	Value
id	int(11)			157
Typeid	int(11)			6
Gameid	int(11)			4
Variationid	int(11)			
Sourceid	int(11)			
DateStart	date			0560-01-01
DateStartBC	tinyint(1)			0
DateEnd	date			0636-12-31
DateEndBC	tinyint(1)			0
Location	text			
KML	varchar(45)			
Rules	text			pieces trapped between two opposing pieces cannot move, pieces are placed on any space on the board in the beginning of play, pieces move forwards or sideways
Misc	text			Isidore Origines IVIII.4? "DE CALCULORUM NOTIS. Calculi partim ordinis sunt, partim vagi: idem alios ordinarios, alios vagos appellamus: ut vici qui moventur ordinis non possunt, Isidore dicitur. Vnde et agens homines Isidore vocatur, quibus spes ultra procedendi nulla restat." Translation Rehdiger 1990: "Some of the pieces move regularly, others here and there: therefore the first are called ordinarii, the others vagi, and those that cannot move at all are called"
ConfidenceMultiplier	int(11)			50

# Evidence Maps

## Game Database:

- Interactive plots
- Evidence per game
- Colour coded by date

## Data points:

- Rules
- Where game was played
- When game was played

## Goal:

- At least one d.p. per game
- All 1,000 games

## Open access





# Data Points

Each data point is:

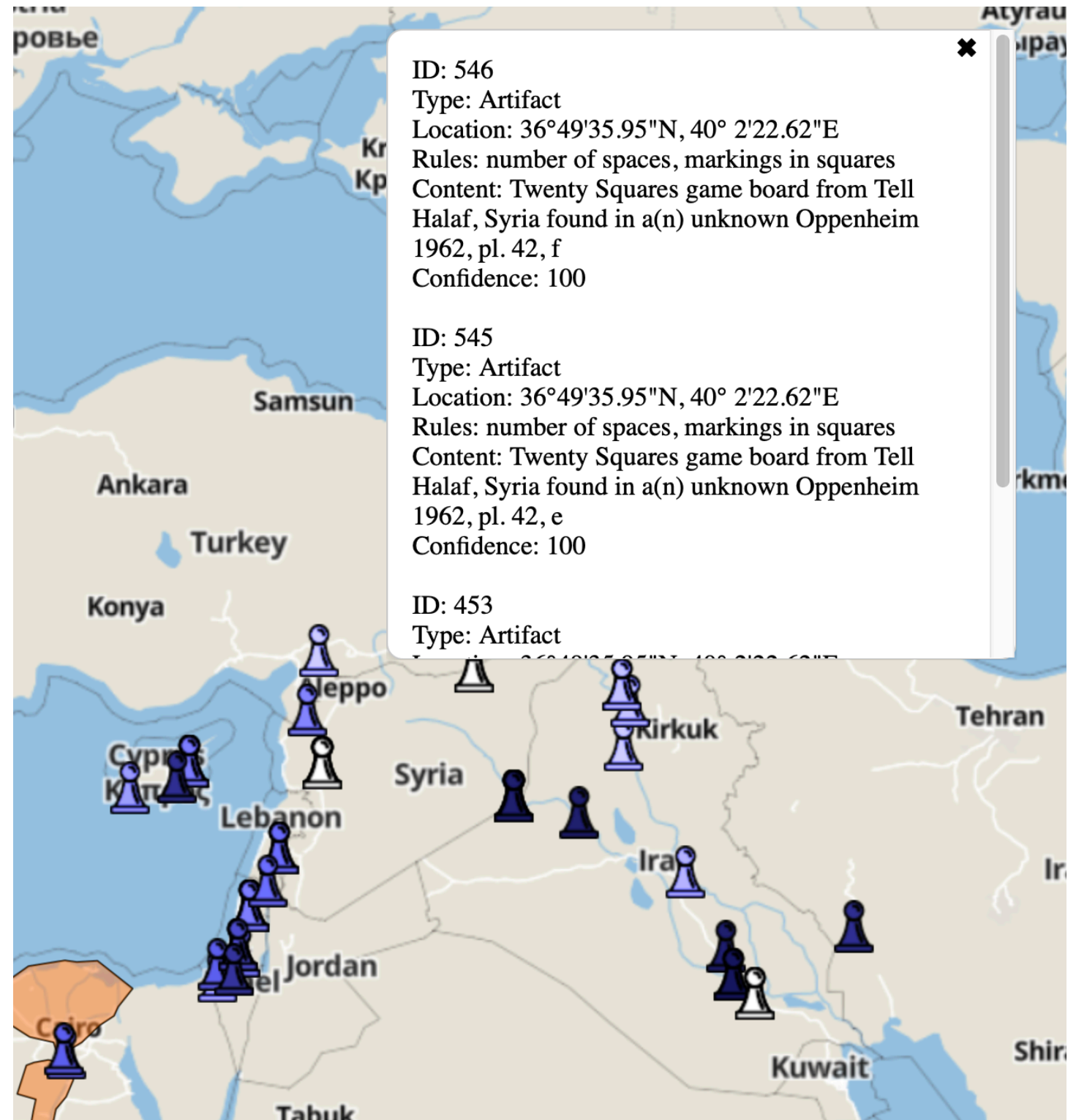
- Artefact
- Artwork
- Historical text
- Literary allusion
- Ethnographic description

Data points can be:

- Point (GPS coordinate)
- Region

Sources listed

Confidence given as %



# Digital Archaeoludology

Traditional game studies:

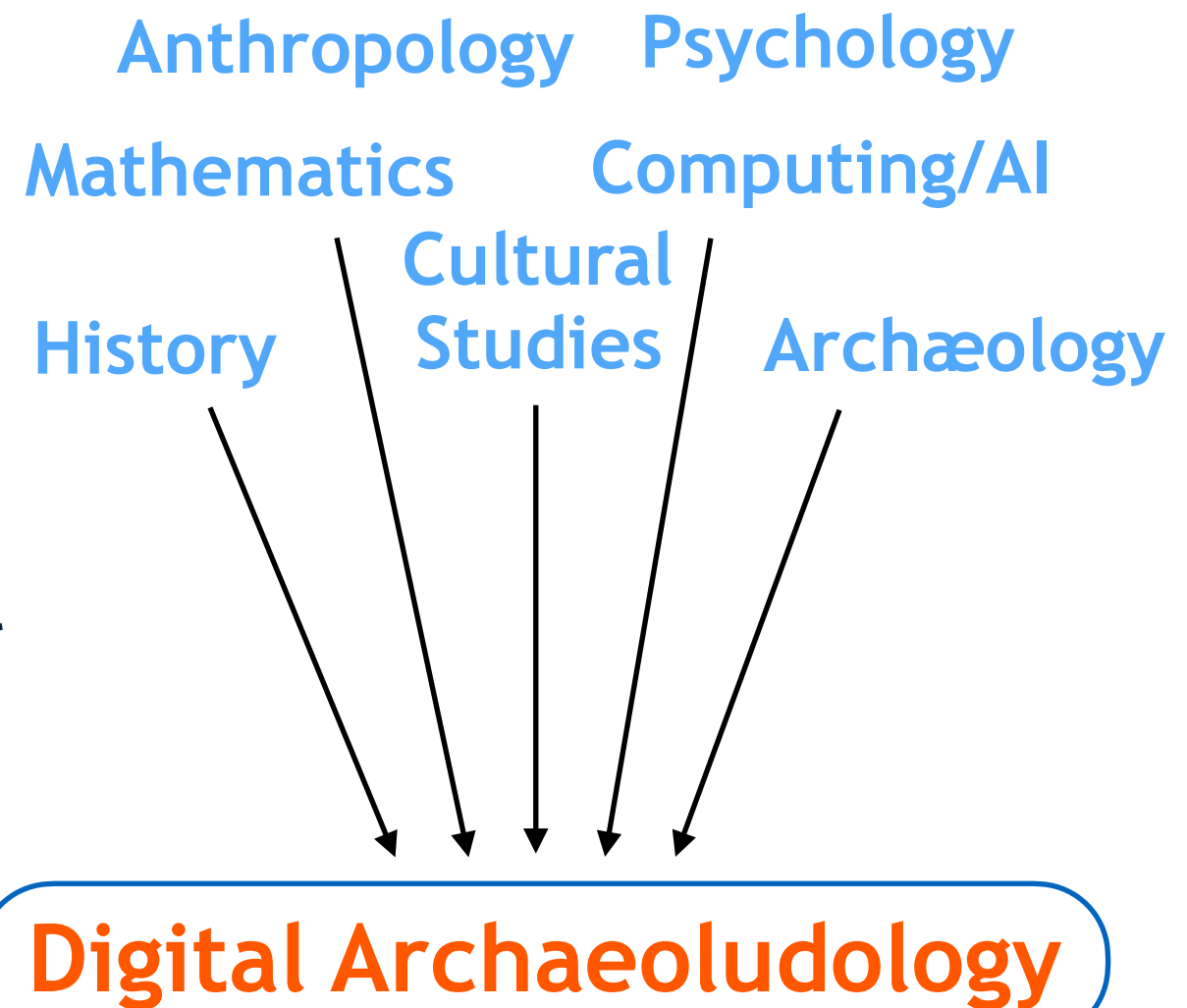
- Wealth of historical analysis
- Little mathematical analysis

Modern game AI studies:

- Huge surge in recent research
- Maths based, no historical context

Seek to bridge this gap

- Single unified research field



**DAL:** *Use of modern computational techniques to harness the available evidence and improve our understanding of ancient games*

# Conclusion

Thank You!

Questions?



<http://ludii.games>



**D**igital  
**L**udeme  
**P**roject

<http://ludeme.eu>



European Research Council

Established by the European Commission

