Cameron Browne (DKE) Game AI and Search Group

Artificial Intelligence for Ancient Games

PAS Festival
Maastricht

7/9/2019

Maastricht University

Evidence for Ancient Games

Much archaeological evidence

- Boards, pieces, dice, etc.

Rule sets rarely recorded

Passed on by oral tradition

- Huge variety today
- Little known about how ancient games were played

Senet

Ancient Egypt, c.3100BC

Hundreds of sets found
- e.g. Tutankhamen's
  1300BC

No rules!

Maastricht University

Senet

Shown in hieroglyphs

Queen Nefertiti c.1300BC

Maastricht University

Senet

Can deduce
- Two players
- Two piece types
- Various starting positions

Special symbols on board
- Entry/exit points?

Dozens of reconstructions
- Most are plausible

First Known Rules

Sumerian cuneiform tablets

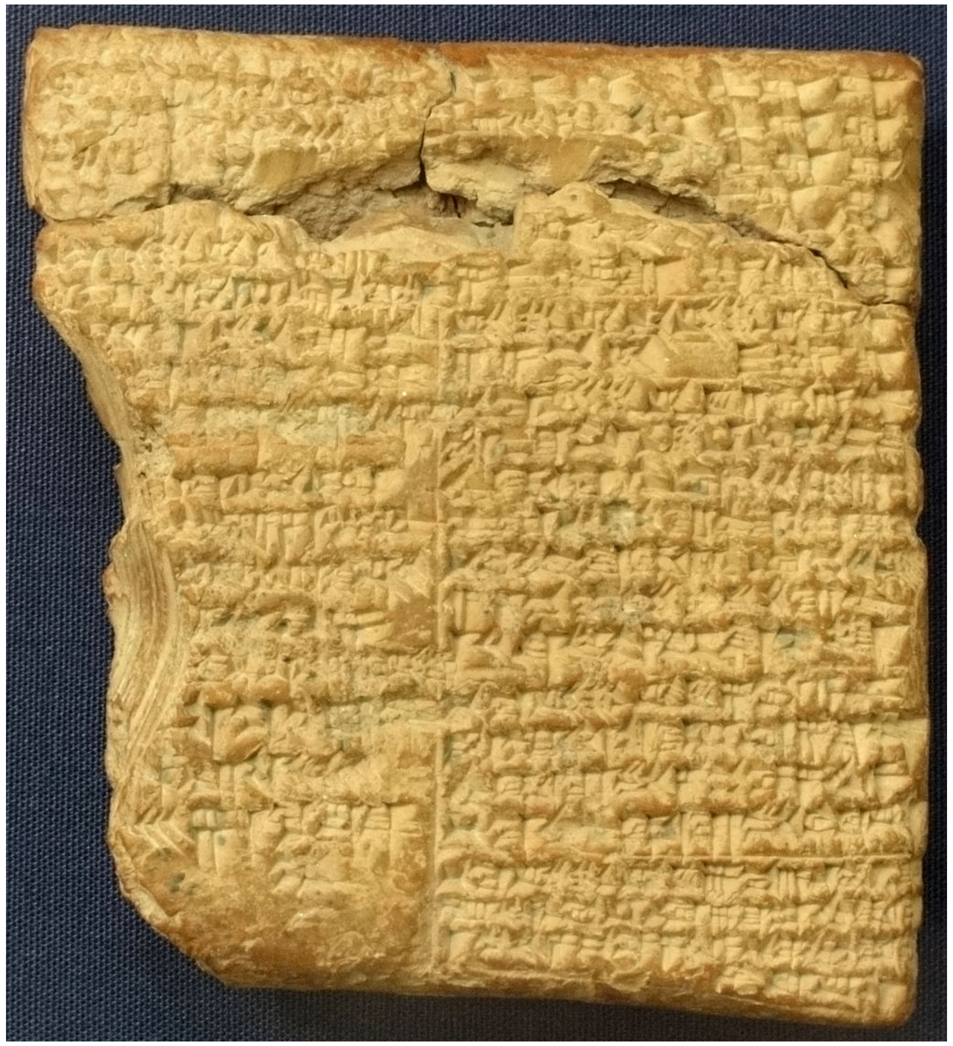
- Mesopotamia, 177BC

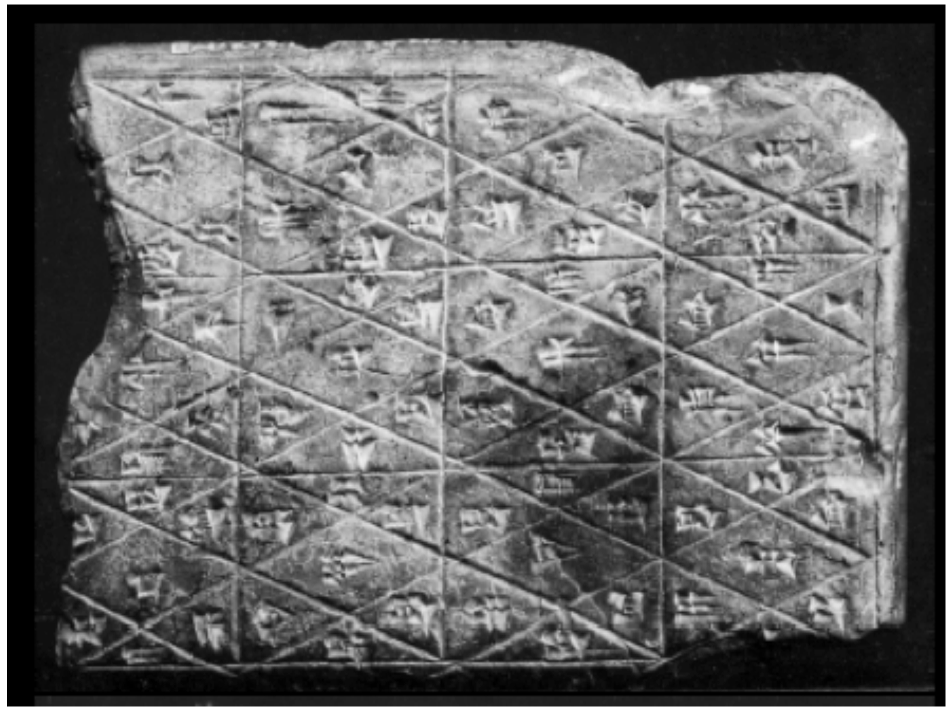
British Museum (top)

- One of 130,000

Parisian (bottom)

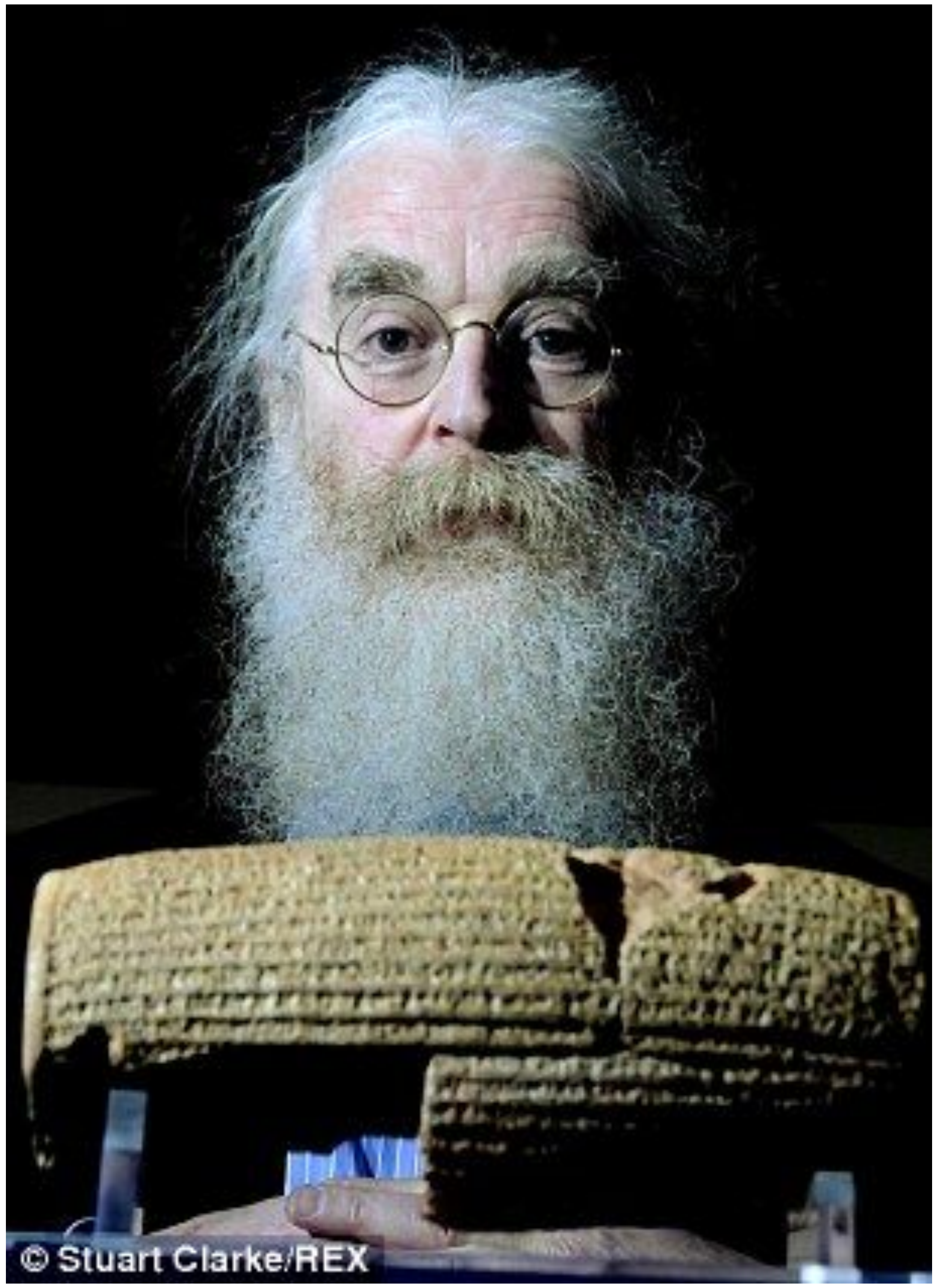
- Destroyed 1940s
- Photo survived



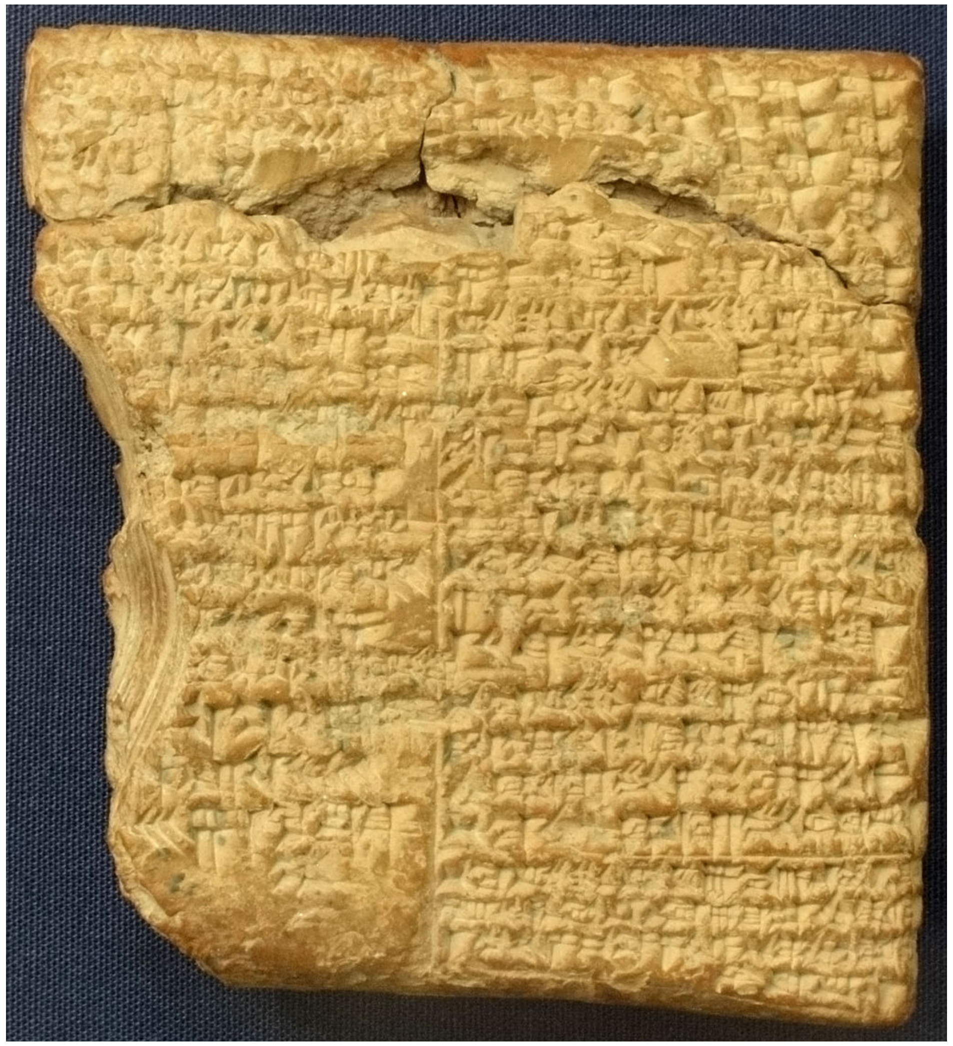


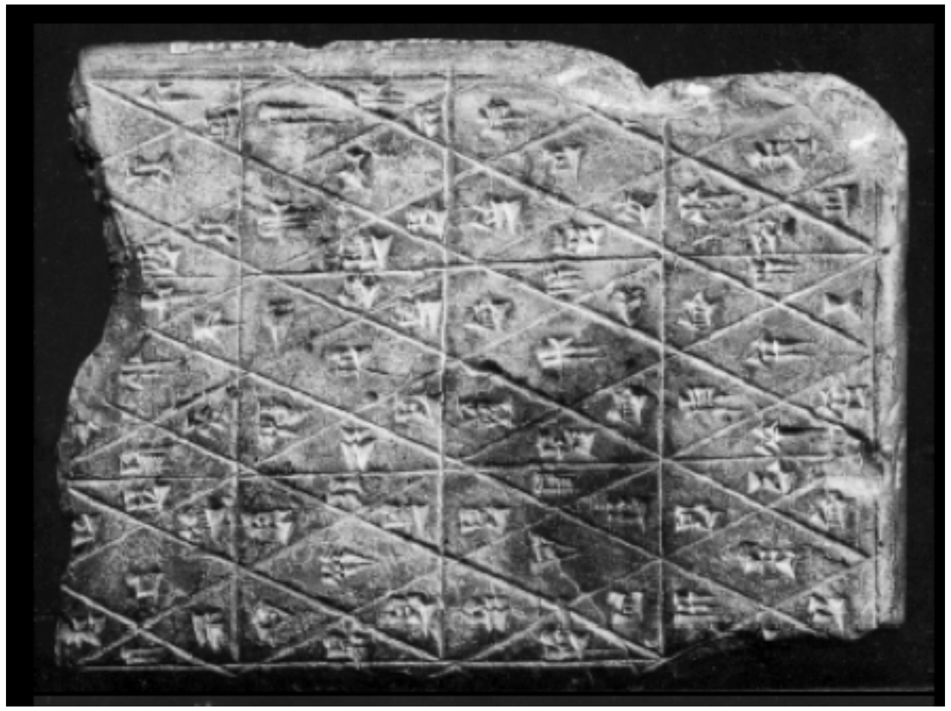
Maastricht University

First Known Rules



Irving Finkel (1990)
Curator, British Museum





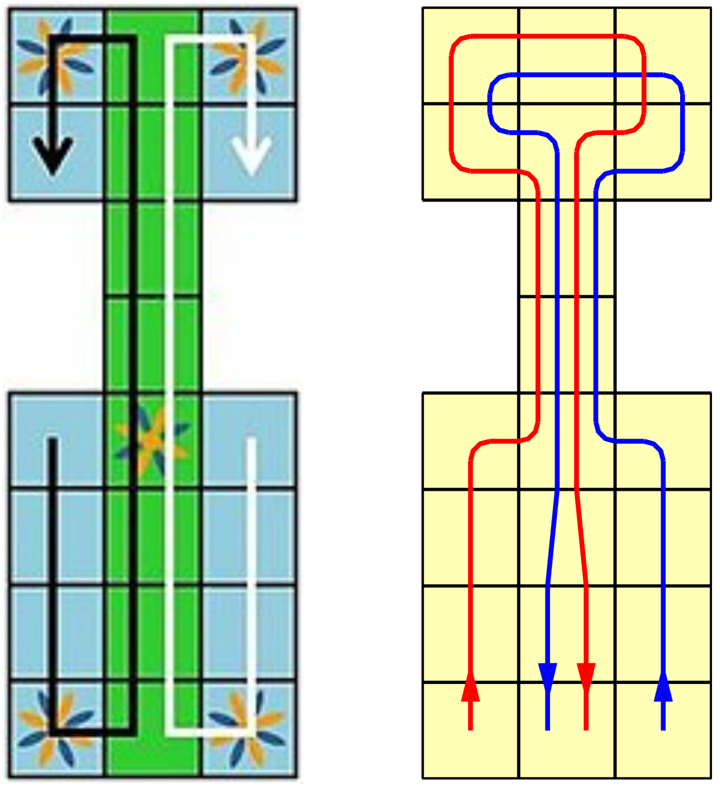
Royal Game of Ur

Played in Mesopotamia
- c.2600BC

Tablets written
- c.177BC

Reconstruction by Irving
- 1990AD

Lots of interpretation
- Same game?
- Which track?



Mu Torere

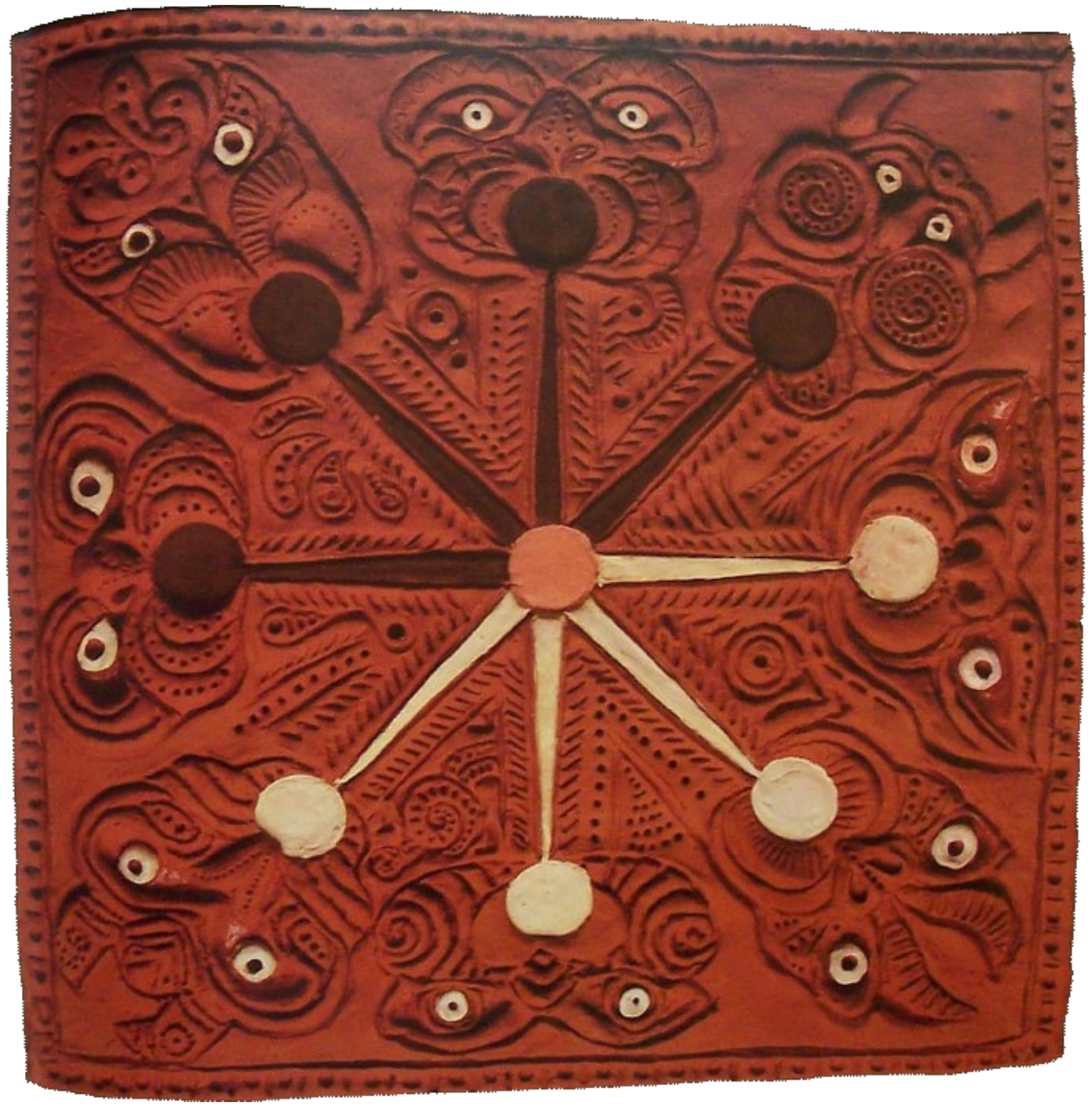
Maori, New Zealand, 18th C
- Living players

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

Some accounts simplify this:

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

Win on first move!



Objectives

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

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

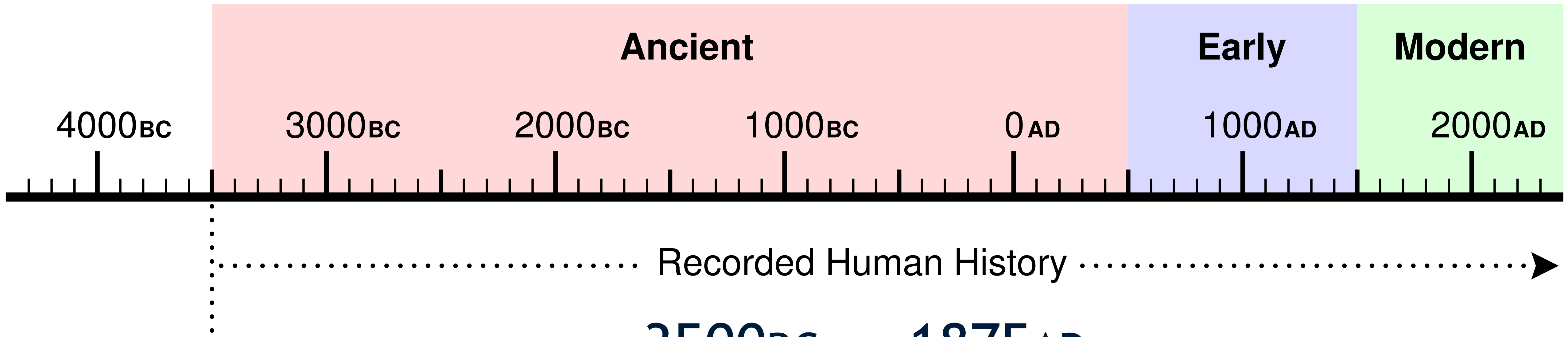
3. ***Map*** Spread of games throughout history

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

Scope

Traditional Games of Strategy

- *Traditional*: No proprietary owner, historical relevance
- *Strategy*: Mental skill, e.g. board, tile, card, etc.



~3500BC – ~1875AD

Identify 1,000 most important traditional games

Q. How to model them in single consistent format?

Maastricht University

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)
```

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))
```

Maastricht University

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))
```

Maastricht University

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 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 immediately

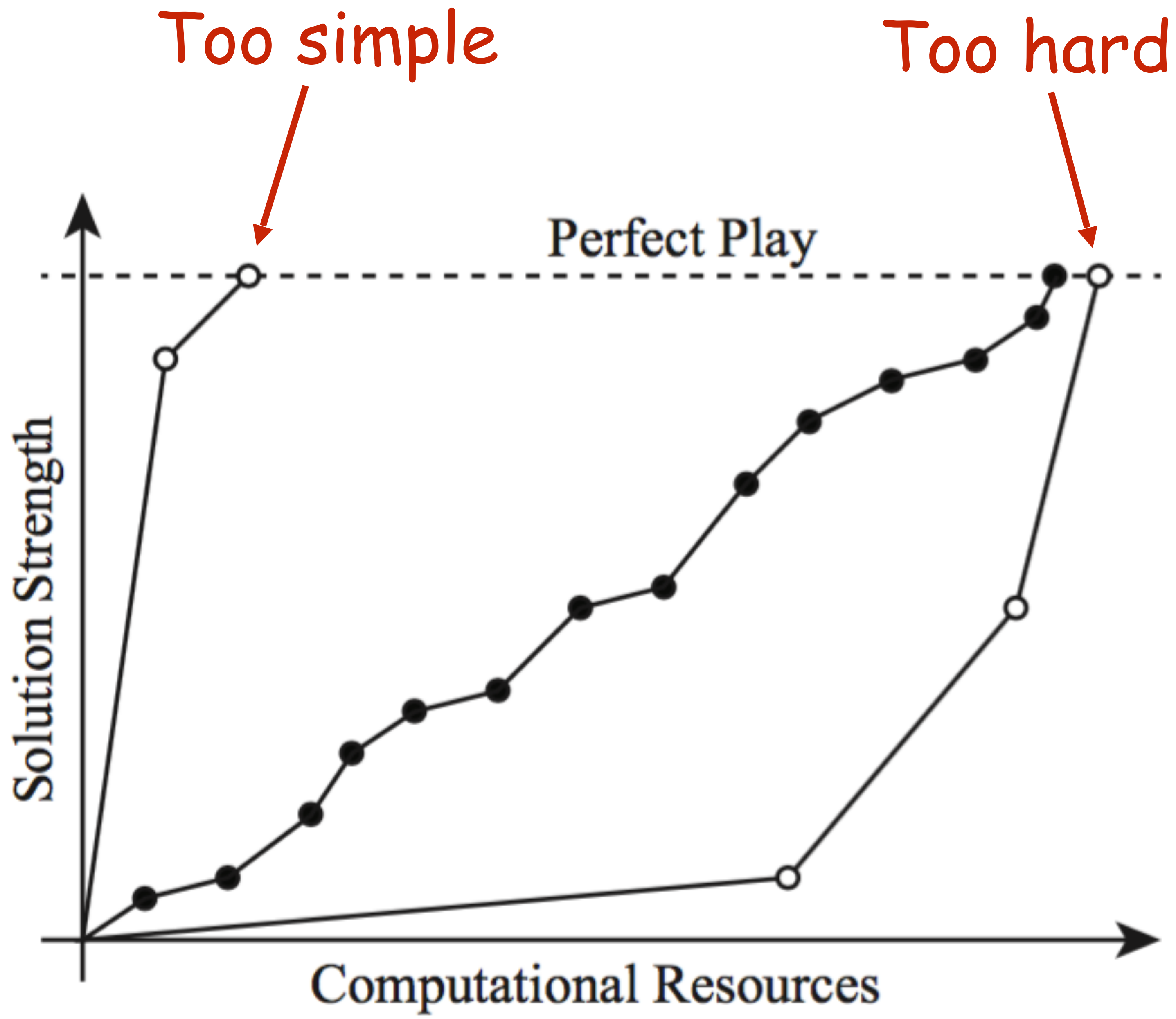
Game Quality

Much harder to define and measure!

Dozens of criteria

Strategic Depth:
- Potential to learn increasingly sophisticated strategies

These are the games that survive



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

Ludii

General game system

- Playing, analysing, designing, reconstructing

Early stages

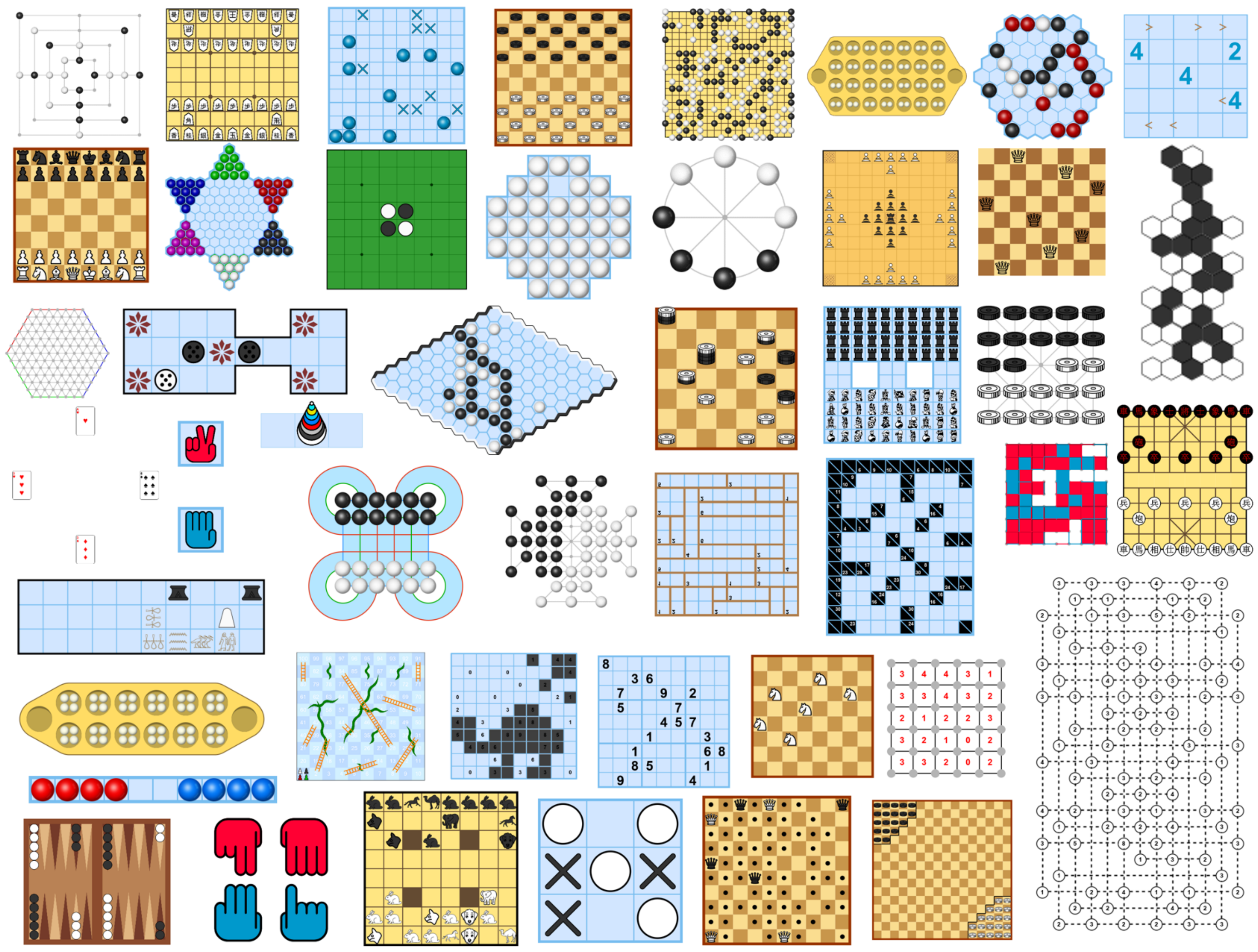
- 100 games

Beta version available

- http://ludii.games

Official release

- 1/1/2020



Maastricht University

Case Study

Hnefatafl "Viking Chess"
- Scandinavia, c.800AD

No written rules found
- Allusions in sagas

Linnaeus (1732)
- Saw Tablut played, wrote down rules (in Latin)

Smith (1811)
- Translated into English

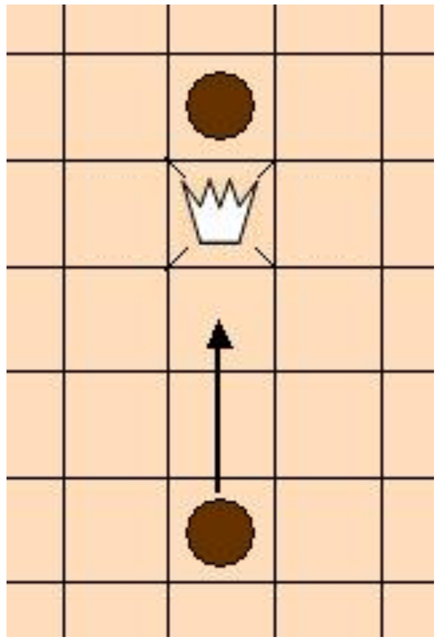
Murray (1913) *History of Chess*
- Published rules, became de facto

Case Study

BUT…

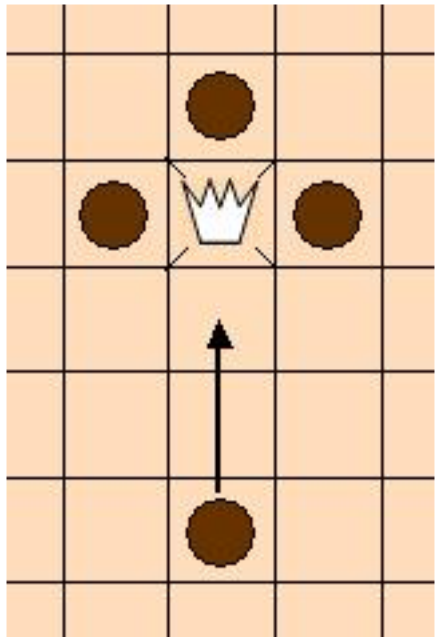
Smith made a bad translation of the king capture rule

A. "likewise the king"
- Flanked
- Easy to capture



B. "except the king"
- Surrounded
- Hard to capture

[DEMO]

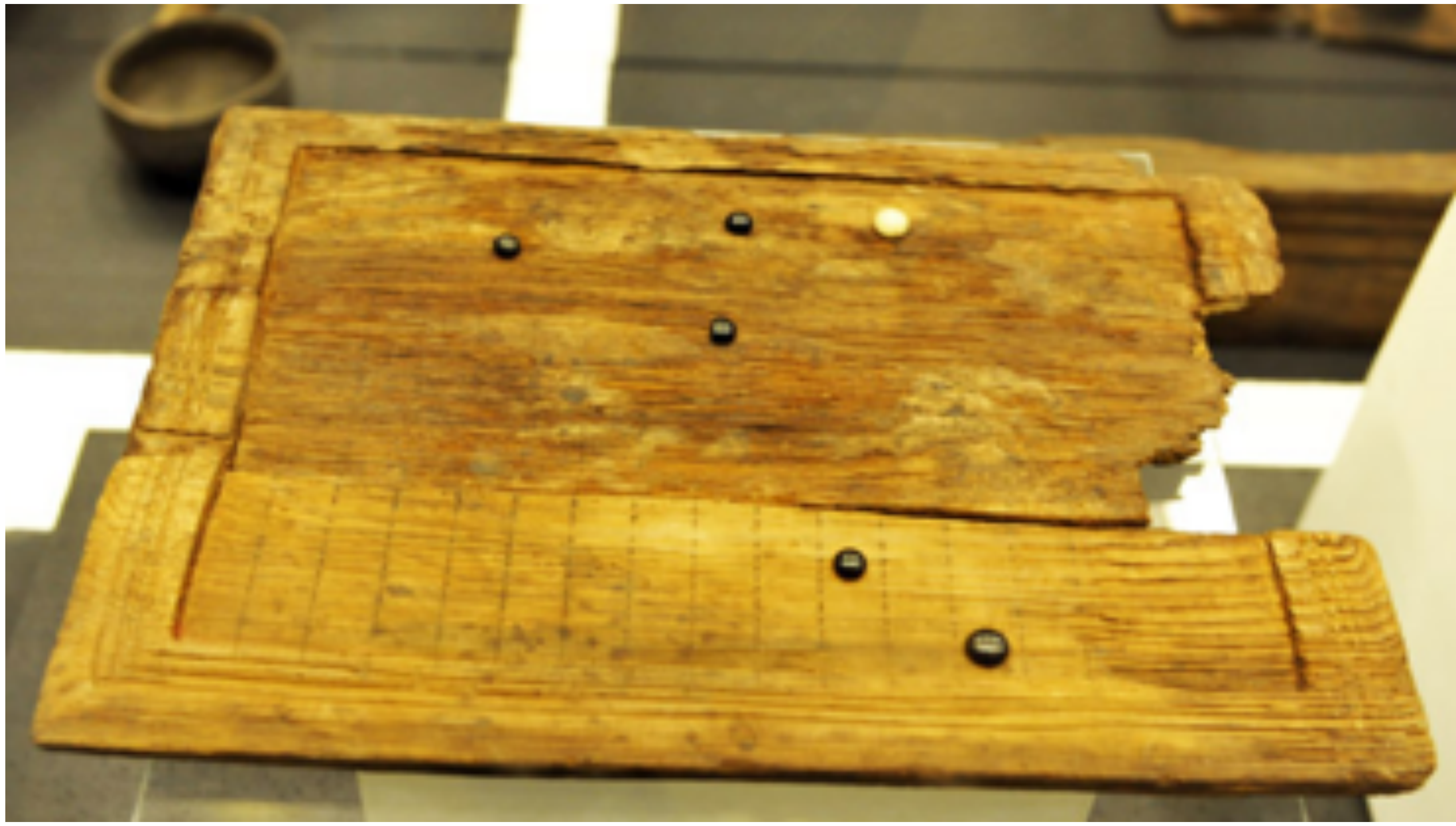


Maastricht University

Forensic Game Reconstruction

Given partial evidence, reconstruct the rules

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



Equipment
- 17x15/16 grid
- 2 x Colours
- 1 or 2 x Sizes?

Ulrich Schadler (2018)
- "An impossible task"
- Ludii could help

Forensic Game Reconstruction

Given:
(players White Black)
(board (rect 17 16)) or (board (rect 17 15))
(pieces (disc White)(disc Black)) or
(pieces (disc White)(disc Black 1)(disc Black 2)) or
(pieces (disc White 1)(disc White 2)(disc Black 1)(disc Black 2))

Search Over:
(start *)
(play *)
(end *)

Prioritise plausible rules
Maximise game quality

Phylogenetics

Ludemes provide measure of “game distance”

Allows phylogenetic analysis

1. **Family Trees**
   - Key game families

2. **Ancestral State Reconstruction**
   - Identify likely traits in ancestors

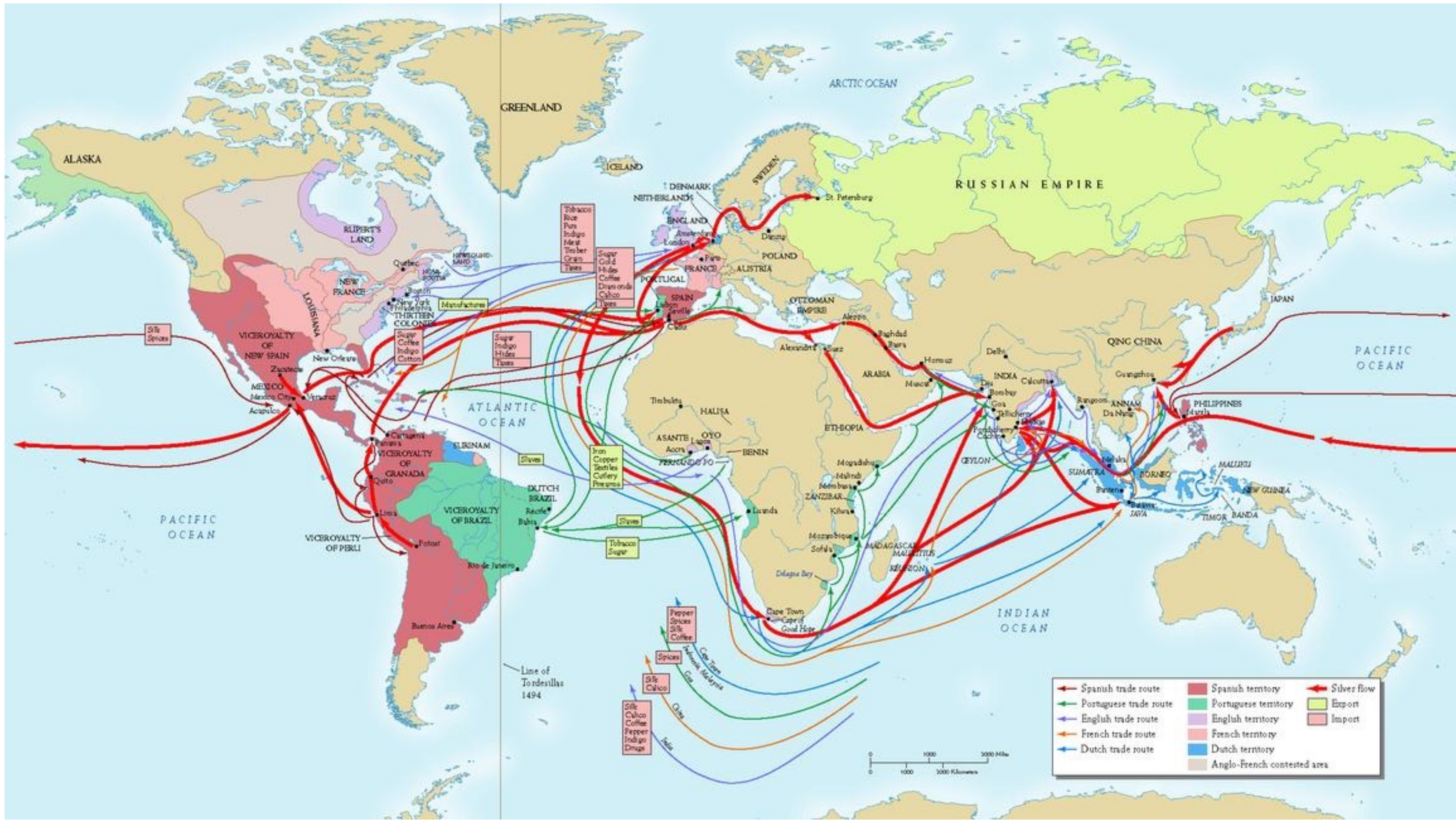
3. **Missing Links**
   - Games that explain gaps in the evolutionary record

Spread of Games

Chart spread of games/ludemes throughout human history

Correlate with:

- Trade routes
- Exploration routes
- Military campaigns
- Crusades
- Diasporas
- *etc.*

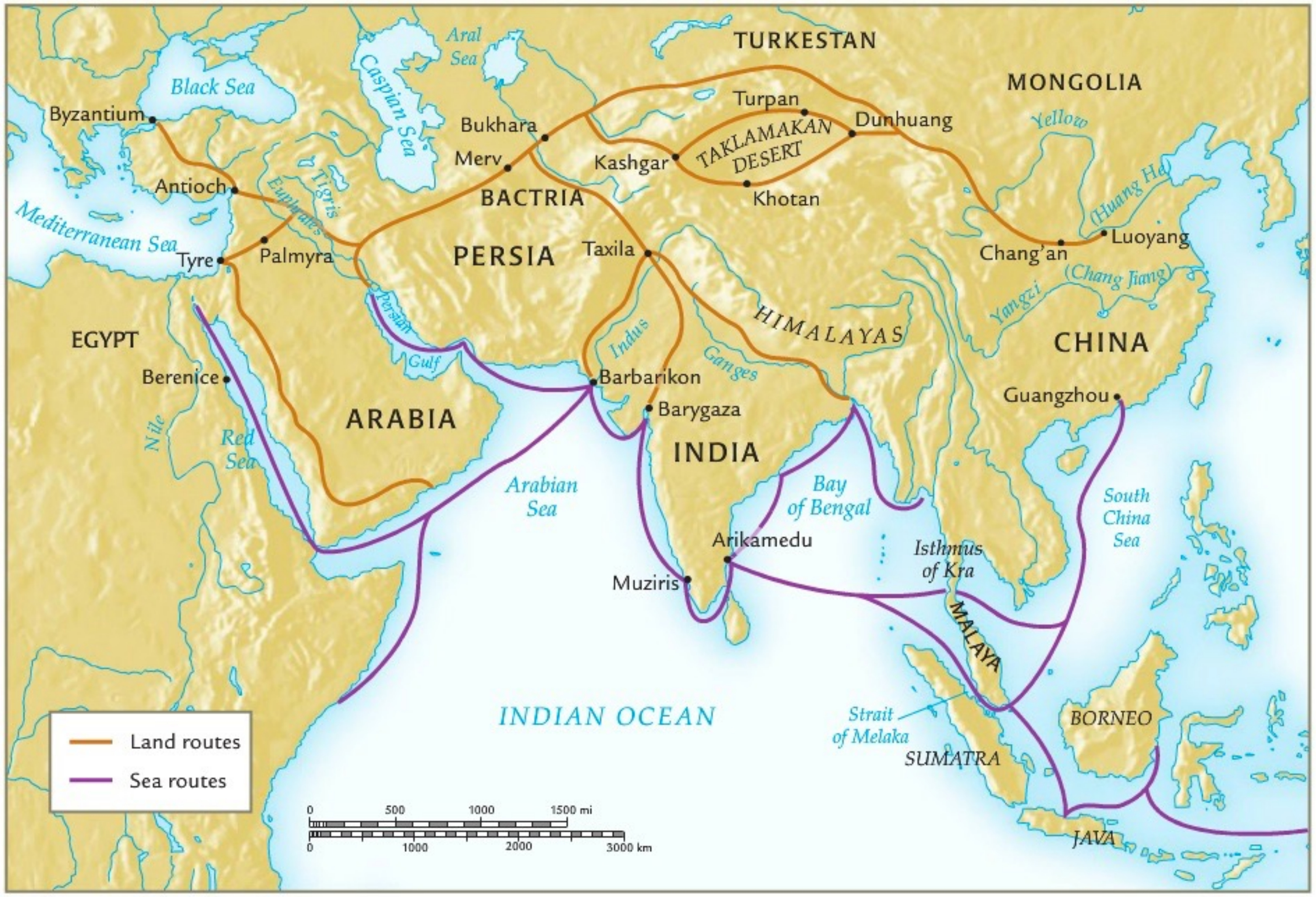


Silk Road Trade Routes

Very important in the history of games

- Fertile crescent:
  - Egypt
  - Sumeria

- Middle East
- India
- Asia



Map 12.1 The silk roads,

Maastricht University

Cultural Contact

Games as evidence of contact

Pachisi

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

Cultural Contact

Games as evidence of contact

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

Patolli
- Ancient Mexico
- Played c.200BC

Different rules…
Same board?

Cultural Contact

Tyler (1879)
- Evidence of early contact
- Centuries before thought possible

Erasmus (1950)
- "Limitation of Possibilities"
- Coincidence

Murray (1952)
- Assume coincidence as a last resort

Can likelihood be quantified?

Preserving Knowledge

Hounds and Jackals
- Egypt ~2000BC

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

Walter Crist (Postdoc, UM)
- Evidence of cultural contact
- Site destroyed last year
- Not published

Aim: Help preserve cultural heritage of games

Maastricht University

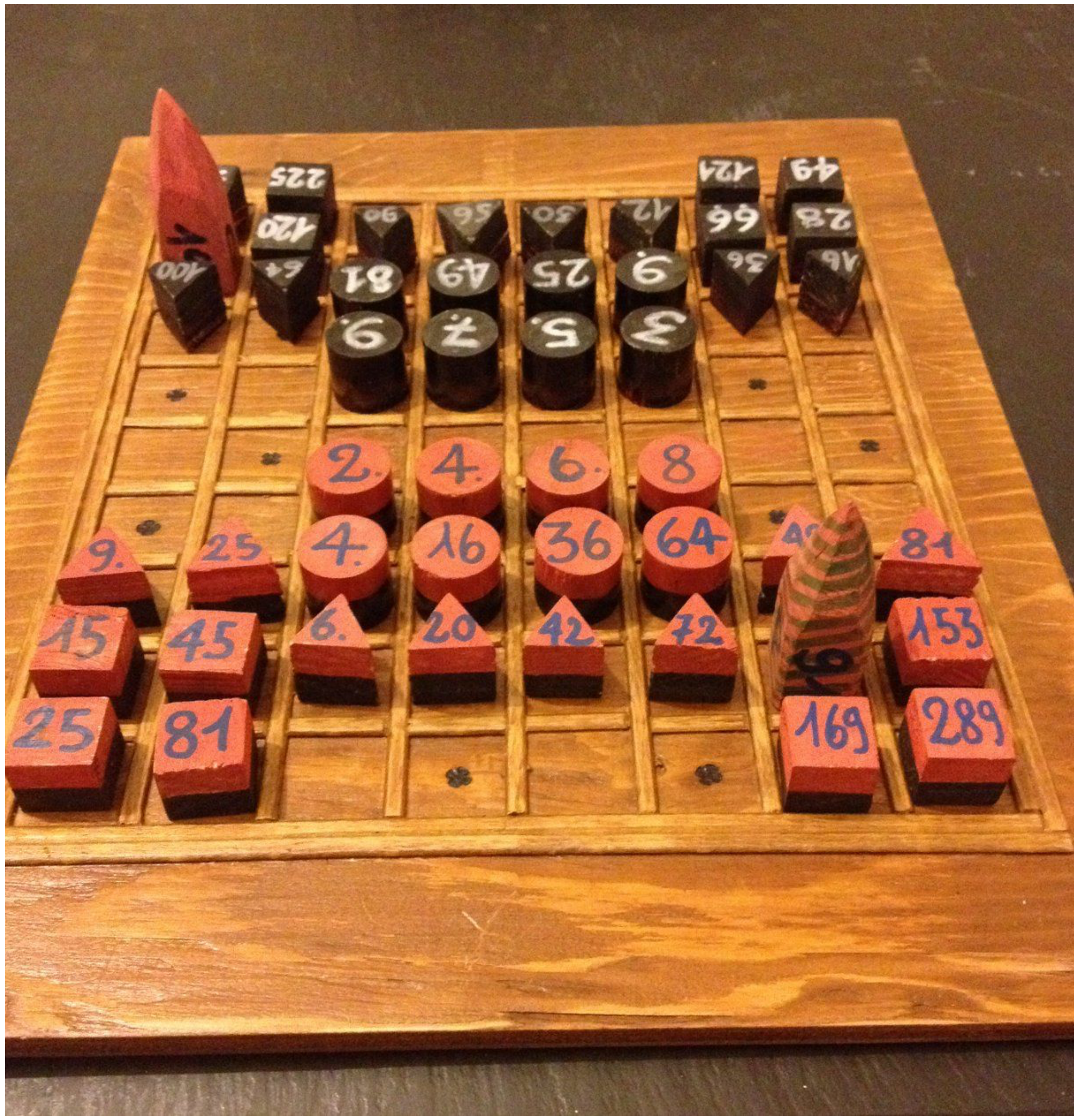
Games and Mathematics

Games are inherently mathematical entities

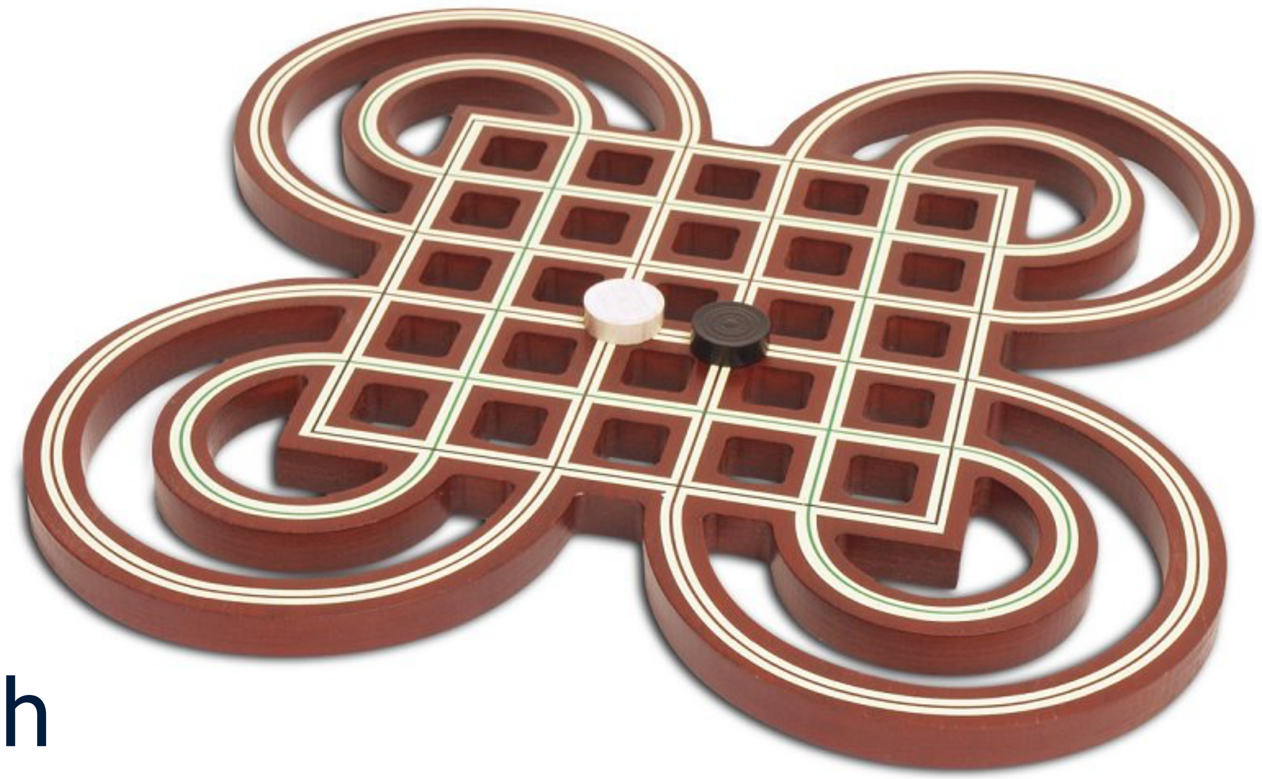
Almost every aspect of a game has underlying math. basis:
- Geometry
- Logic
- Algebra
- Arithmetic

Can we correlate spread of games with spread of of mathematical ideas?

Games as vehicles of math. ideas


Rithmomachia (11thC)

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

Conclusion

Games are an important part of cultural heritage

- Like language, music, art, etc.

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

- Better understanding, better insight

Conclusion

European Research Council
Established by the European Commission

Thank You!

Questions?

Digital Ludeme Project

http://ludeme.eu

http://ludii.games

Maastricht University