## SIKS-DKE Symposium: New Advances in Game Al

## Artificial Intelligence and the Heritage of Games

Cameron Browne
Game Al \&t Search Group

Maastricht University 13/11/2019

(1) 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-I200AD)
Maastricht University


Pachisi India (from 600-I600 AD)

## Cultural Contact

Tyler (1879)

- Evidence of early pre-Columbian contact

Erasmus (1950)

- Coincidence, "Limitation of Possibilities"



Patolli Mexico (200BC-I 200AD)
M Maastricht University


Pachisi India (600-I600 AD)

## Games in Human History

Archaeological evidence of games:

- Found worldwide
- Throughout recorded history

Provides clues:

- Reconstruction is a subjective endeavour
- Highly unreliable

| 4000вс | Ancient |  |  |  | Early | Modern |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 3000 вс | 2000 bc | 1000вс | Oad | 1000 ad | 2000 ad |
|  |  |  |  |  |  |  |

Maastricht University

## 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 Al techniques

Maastricht University

Cameron Browne (PI)

- Game AI (technical lead)

Eric Piette (Postdoctoral Researcher)

- Game Al (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
- Ancient Egypt and Mediterranean specialist


## Scope

## Traditional games of strategy

## Traditional

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


XII Scripta board from Laodicaea, Turkey

## 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

Maastricht University

## 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
Maastricht University


## Royal Game of Ur

Played in Mesopotamia

- c.2600bc

Tablets written

- c. 177BC


Reconstruction by Irving

- 1990ad

Still some questions

- Same game? Probably
- Which track?



l dəКеІd


## Transcription Errors

Mu Torere (New Zealand, 18 ${ }^{\text {th }}$ 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!

Maastricht University


## Invented Traditions

Birrguu Matya (Australian Aboriginal, late 19th 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

Maastricht University


## 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?

- Quantify where possible
- Encode in single consistent format
- Establish historical/cultural context
- Find relationships within data


## Ludemes

Game "memes"

- 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


Maastricht University

## Ludemes

Game "memes"

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


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

\section*{Ludemes}

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

```

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

```

Maastricht University

\section*{Stanford GDL}

\section*{Academic standard \\ - 15 years}

\section*{Programmer's view \\ - Low level instructions \\ - Not high level concepts}

\author{
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 O) open (not (line o)))
(<= terminal (line x))
(<= terminal (line o))
(<= terminal (not open))

```
```

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

(<= 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)))
)

\section*{Ludemes vs GDL}
```

(role white) (role black)

```
```

(init (cell 1 1 b)) (init (cell 1 2 b)) (init (cell 1 3 b))

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

(<= terminal (not open))

```
(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 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 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 (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 O) 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)))
)

## 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 immediately

## Game Quality

Potential to interest human players

- Much harder to define and measure!
- Difficult to validate


## Strategic Depth

- Potential for increasingly sophisticated strategies
- Universal quality metric?


Strategy Ladder
Lantz et al. (AAAI'17)

## Importance of Strategies

Omega (2010)

1. Play each colour per turn.
2. Multiply your group sizes.
3. Highest score wins.

- Difficult and tedious to play e.g. Who's winning this game?
- No strategy, essentially making random moves

- Not popular


## Importance of Strategies

Omega (2010)

1. Play each colour per turn.
2. Multiply your group sizes.
3. Highest score wins.

- Difficult and tedious to play e.g. Who's winning this game?
- Intuition fails, no strategy, making random moves
- Not popular


$$
\text { White: } 1 \times 2 \times 2 \times 3 \times 4=48 \quad \text { Blue: } 1 \times 2 \times 3 \times 6=36
$$

Red: $\quad 1 \times 2 \times 4 \times 5=40$ Black: $1 \times 4 \times 7=28$

## Emergent Strategies

Key insight (from UCT):

- Partitions of 3 maximise product


## Strategy:

- Make groups of 3 of your colour

Corollary:

- Join enemies into larger groups
- Emergent strategies:
- Connection game
- Anti-connection game


White: $1 \times 2 \times 2 \times 3 \times 4=48$ Blue: $1 \times 2 \times 3 \times 6=36$ Red: $\quad 1 \times 2 \times 4 \times 5=40$ Black: $1 \times 4 \times 7=28$ - UCT agent first to use strategy

- Maths (and AI) saved this game:

Maastricht University

## Strategy Detection

- Learn features to bias MCTS playouts
- Transfer to similar games
- Explain in simple geometric terms
- Feature encode simple strategies
e.g. "Grow small groups" "Don't grow beyond size 3" "Virtually join enemy pieces" "Don’t virtually join friendly pieces




## Hypotheses

Games with deeper* and more easily understood strategies are more likely to:
I. Be played.
II. Be taught.
III. Survive.

* Relative to complexity

Maastricht University

## Ludii

General game system

- Playing, analysing, designing, reconstructing

Early stages

- ~150 games

Beta version available - http://ludii.games

Official release

- January 2020



## Case Study

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



## 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]



## 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


Evolution of Mesoamerican board games (after Depaulis 2018)

## Computational Phylogenetics

Ludemic distance allows phylogenetic analysis

- Horizontal rule transmission

1. Family Trees/Networks

- Key game families

2. Ancestral State Reconstruction

- Identify likely traits in ancestors


Phylogeny of Chess (Kraaijeveld 2000)

## 3. Missing Links

- Games that explain gaps in the evolutionary record?


Horizontal Influence Map

## 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
- Sumeria
- Middle East
- India
- Asia


Map 12.1 The silk roads,

## Digital Archaeoludology

Traditional game studies:

- Wealth of historical analysis
- Little mathematical analysis

Modern game Al studies:

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

Seek to bridge this gap

- Single unified research field


## Anthropology Psychology

Mathematics Computing/AI


Digital Archaeoludology

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
(1) Maastricht University

http://ludeme.eu


## Data Gathering

## Sources:

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


Ludus Latrunculorum board (Sabratha)
Maastricht University


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


## Data to be Gathered

Data per game:

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

Historical/cultural profile per game

- Also for component ludemes

Maastricht University


## What Can We Do With This?

Derive historical/cultural context for each:

- Game
- Ludeme

Evaluate known reconstructions for:

- Plausibility
- Quality

Generate own own reconstructions (maximising for same)
Provide a distribution of reconstructions per game, ranked by relevant metrics

## Forensic Game Reconstruction

Given partial evidence, reconstruct the rules
e.g. Poprad Game (Slovakia)

- Tomb dated to 375AD
- Germanic chieftain

Equipment

- 17x15/16 grid
- $2 \times$ Colours
- 1 or $2 \times$ Sizes?

Ulrich Schadler (2018)


- "An impossible task"
- Ludii could help

D Maastricht University

## Forensic Game Reconstruction

(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))
(start *)
(play *)
(end *)
Use historical context

- Prioritise plausible rules
- Maximise game quality

Maastricht University


## Al Players

Monte Carlo tree search (MCTS)

- Run many (semi)random simulations
- Build search tree from results
- Works well for many games ...but not all

Bias playouts with features

- Geometric piece patterns
- Encourage good/discourage bad
- Generalise to arbitrary grids


Benefits

- Improve player strength
- Encode strategies = depth
- Explain strategies?


"Make-line-of-4" features

