

GameTable

Key Benefits of Using Games In Computer Science and AI Education

GameTable WG5

UCLouvain Bachelor/Master Programs in Computer Science



- Focus on **problem-solving techniques** through integrated Computer Science projects.
- **Hands-on experience with small-scale and medium-scale projects.**
- Examples of a **few Courses**:
 - Introduction to programming/algorithms/web development
 - Database
 - Computer Networks
 - Logics
 - Algorithms and Data Structure
 - Artificial Intelligence
 - Machine Learning

Games for Computer Science Education

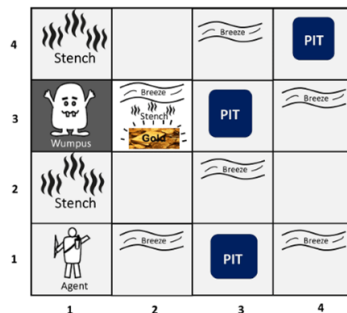
- Games are used throughout our bachelor's and master's degrees at UCLouvain.

Why games in CS?

- Engaging and motivating
- Practical Applications
- Interactive and hands-on learning
- Problem-solving and critical thinking
- Teamwork and Collaboration
- Many computer science concepts can be learned through games**

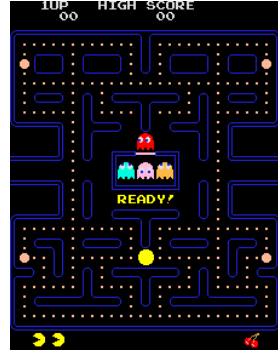
Types of games?

- Puzzles (Mazes, Sudoku, etc.)
- Theoretical games (Wompus World, etc.)
- Board games
- Rarely video games



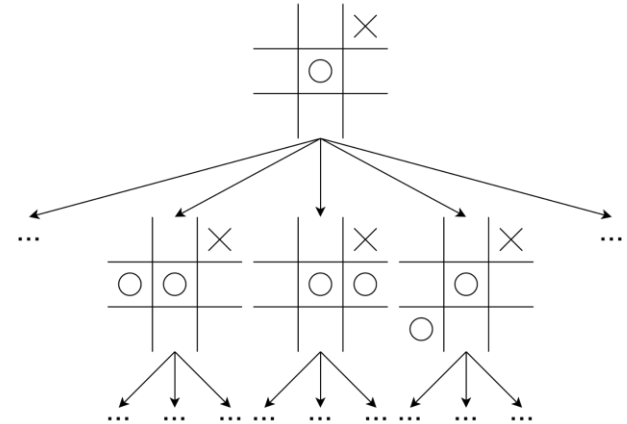
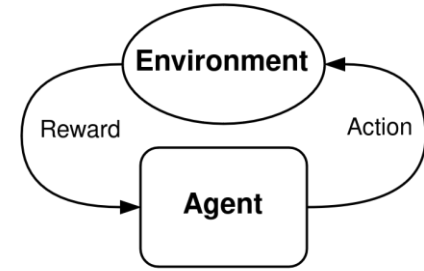
Introduction to Computer Sciences

- Students must create an algorithm to **navigate a player through any maze while avoiding monsters.**
- Introduces students to **logic and basic programming** in a **fun and challenging** way.
- Promote **teamwork** for the remaining years and give students their first experience with the benefits and challenges of working collaboratively instead of alone.



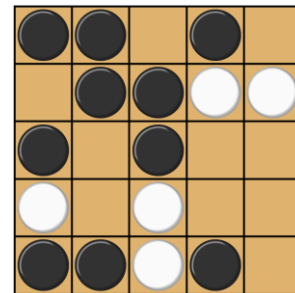
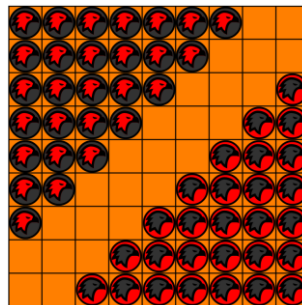
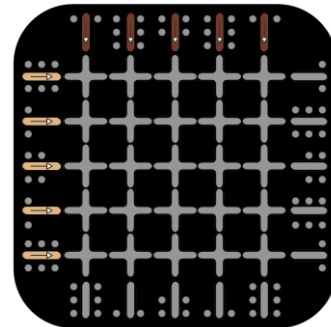
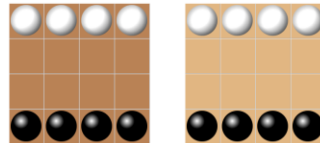
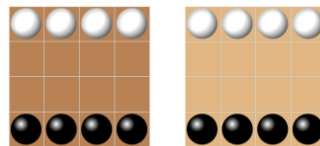
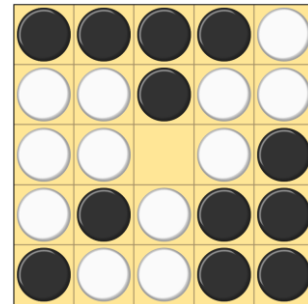
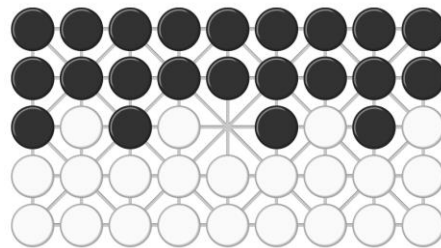
Introduction to Artificial Intelligence

- Lots of AI problems \approx which decision to take in an environment ?
 - Finding the best decision is exploring the tree of all possible decision
- Games are simple environments in which we make decision
 - Everyone has played a game in their life.



AI agent for board games

- **Develop an AI agent for a specific board games**
 - Apply directly what they learn.
 - Play against the agent they create.
- **Side note:** we introduce them to a different board game each year
 - **Historical:** Seega, Yoté, Fanorona
 - **Modern:** Shobu, Squadro, Fenix



Competition between student's agents

- Project last 3–4 weeks, at the end they must submit an AI agent.
- During the final week, a small competition is held where all submitted agents compete
 - A variety of games and scenarios are showcased.
 - Winners are celebrated, and podium finishers receive prizes.
- Encourages engagement and motivation among students.
 - One of the most appreciated and well remembered project in the bachelor.



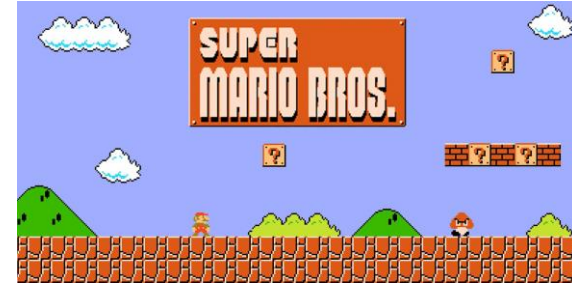
Develop an Android application

- **Design and development of an application of their choosing.**
 - Collaboration
 - Work balancing
 - Meeting deadlines
- Many groups of students choose to develop a game
 - **More engaged and motivated.**
 - **Better division of work.**
 - **Lead to better project outcomes and grades.**



Mastering Programming Paradigms Through Game Development

- Teaching Object-Oriented Programming Through **Video Game Development**
- Demonstrating **key principles** of a programming language through the **gradual development of a familiar game**.
- **Peer evaluation** of projects, followed by **showcasing** and playing the top projects in front of the whole group.



Benefits of games

- Increased student engagement and motivation
 - Games are **fun**
 - Games provide **instant feedback and a sense of accomplishment**
- Improved understanding and retention of CS concepts
 - Games are **practical applications for theoretical concepts**
- Development of soft skills (communication, teamwork, creativity)
 - Games encourage students to **work together, share ideas, and collaborate on projects**

A challenge of using games in CS/AI Education?

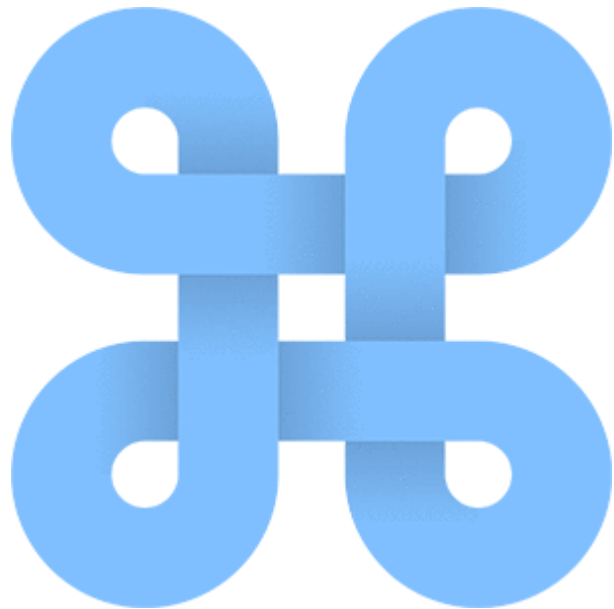
- **Complexity and Time Constraints**

This often requires significant preparation, depending on the course, but the game implementation should always prioritise the core learning objectives, examples:

- For AI classes, the game framework is typically pre-developed by the instructor(s), and students are provided with a simple AI API to focus on implementing AI solutions.
- For development projects, the game is pre-built and presented at the start, with all non-essential code provided to students so they can focus on key tasks.

To summarise: The focus should remain on the learning objectives, not on the game itself.

Wouldn't it be ideal if all games were pre-built or easy to set up?



Ludii



Ludii Library

1300+ games, 2800+ rulesets, ~2,000,000 option combinations

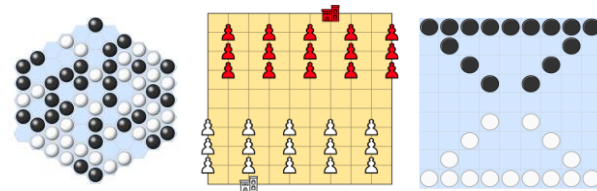


Ludii

- General Game System capable of modelling all games of the DLP project (and more)
- Granular description of the games in ludemes. Clear, simple and short
- Currently, between 2 to 100 times faster than GDL
- Can model:
 - Deterministic / Stochastic
 - Complete / Hidden Info.
 - Puzzles / Multi-Player Games
 - Alternating / Simultaneous
 - Borderless / Any geometry
 - Math, Race, Sow, Space, War Games, ...
 - Stacking games, large pieces, ...
- All playable (humanly) thanks to the GUI **and remotely through our game server**
- Available at: ludii.games
- Source available: github.com/Ludeme/Ludii
- Since 2021, official competition platform of the Computer Olympiads
- Great (AI) teaching platform



Already done at Maastricht University



- **Intelligent Search & Games**

- First year Master AI course (basic search techniques, advanced search techniques, heuristics, Monte Carlo methods, Game Design)
 - Practical assignment (50% of the grade for the course)
 - Implement AI for 2-player game
 - Different game every year
-
- Before Ludii, everything was done from scratch (Game Logic, Interface, AI).
 - Now, the students can quickly design the game and focus only on the AI implementations using all the Ludii logic and interface.

But right now, Ludii **doesn't cover all CS teaching topics**

- **Can not do:**

- Web Development
- Database
- Network



- **But this could be added thanks to:**

- Ludii server
- Ludii database
- Ludii competition framework



Overview of a Game description

```
(game "Amazons"
```

```
(players 2)
```

Number of players

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

Equipment:
Containers, Components, ...

```
(rules
```

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

Starting Rules:
Piece placement, initial score, ...

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

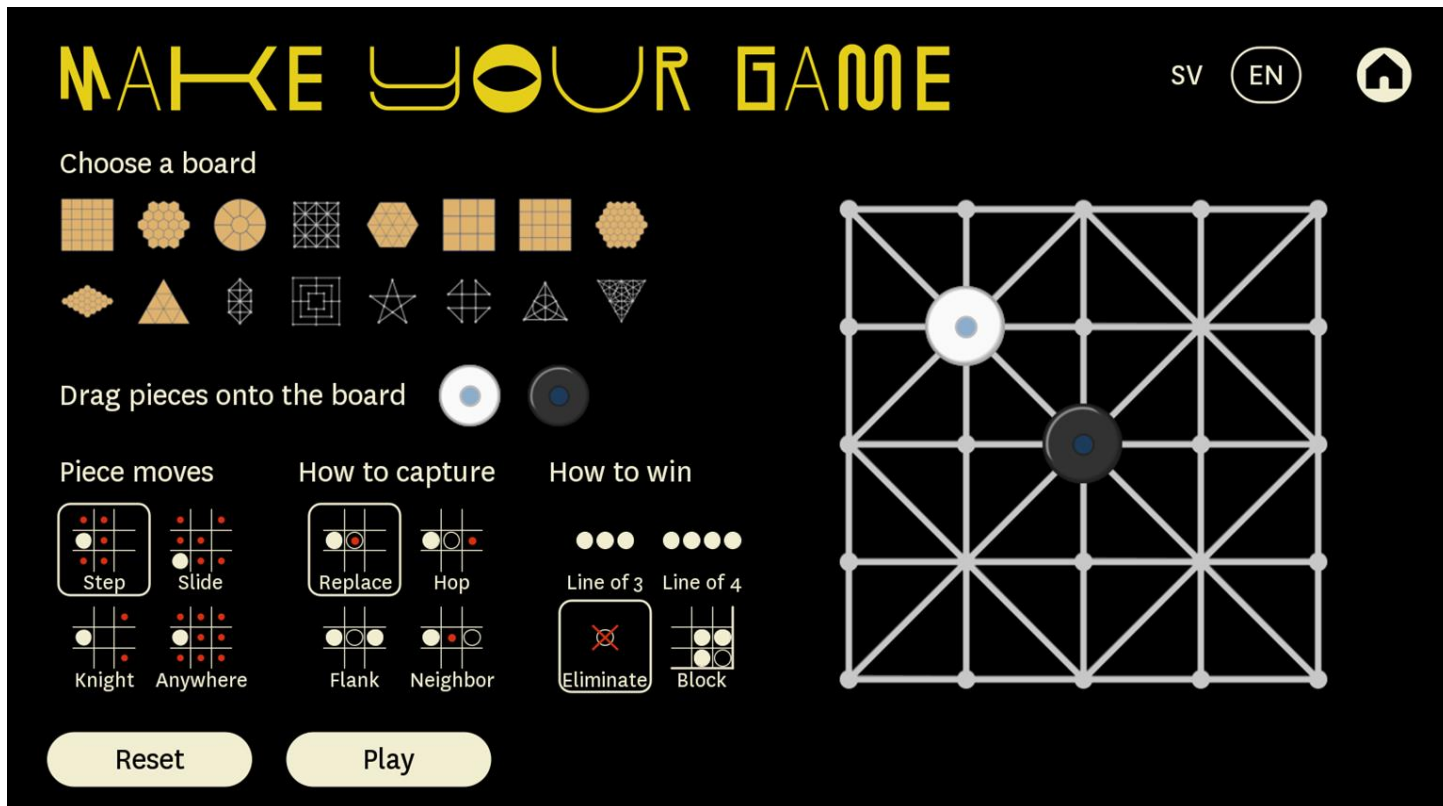
Playing Rules:
Legal moves for each state

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

Ending Rules:
Terminal conditions and outcomes

```
)  
)
```

Making them even simpler to design? And/Or training schools?



Ludii in other disciplines?

- Mathematics?
- Cultural Heritage?
- History?
- Archaeology?
- Others?



Would a software tool **be helpful** in supporting your teaching activities?
If so, how would it benefit you? **How can we help?**
I'd love to discuss this further over the next days.



TIME TO CHAT