NEidI / LDiCS
Algorithms in/and Game Environments
Organizational Matters

Andreas Ruscheinski
Computer Science Division, University of Rostock

Motivation

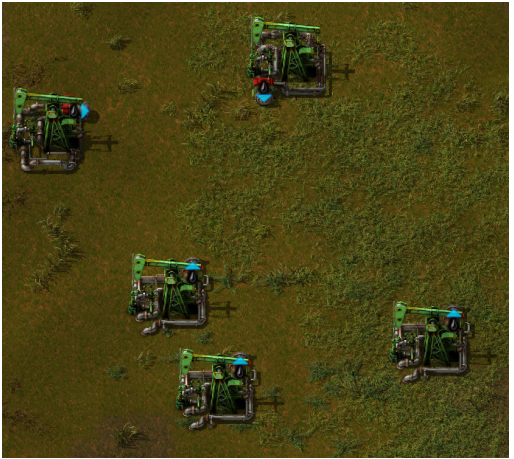
- Games are an interesting testbed for studying, developing and evaluating algorithms
- Reasons:
  - Structured Environment: Clear Rules and Objectives
  - Outcomes of scenarios can be easily measured and compared
  - Games involve complex decision-making and strategic planning
- Challenges:
  - Dynamic and unpredictable nature
  - Limitation of computational resources
  - Not all information is observable

Example 1: Pipe Routing

Problem: Connect all the oil jacks with the least amount of pipes

Challenges:

- Each oil jack can be rotated in four positions
- A lot of different paths for the connections



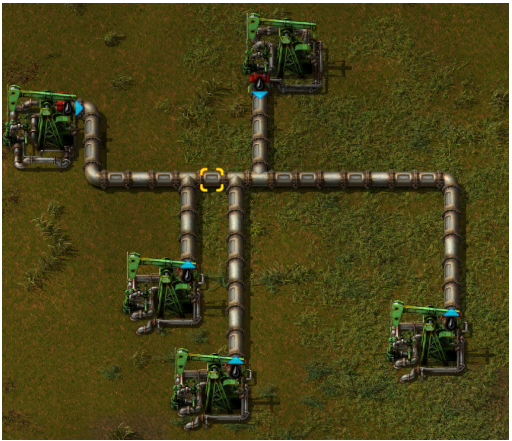
Connect the oil jacks with pipes

Example 1: Pipe Routing

Problem: Connect all the oil jacks with the least amount of pipes

Challenges:

- Each oil jack can be rotated in four positions
- A lot of different paths for the connections



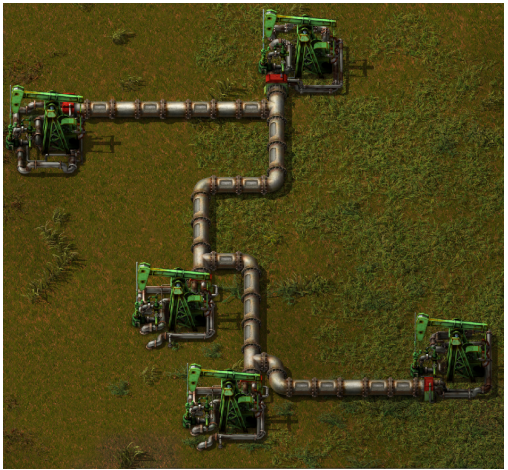
Solution: 37 pipes

Example 1: Pipe Routing

Problem: Connect all the oil jacks with the least amount of pipes

Challenges:

- Each oil jack can be rotated in four positions
- A lot of different paths for the connections



Solution: 30 pipes

Example 2: Infinite Mario Challenge

Problem: Develop an agent that gets as far and fast as possible in a random generated level

Challenges:

- Limited perception of the agent
- Dynamic environment (Moving enemies and platforms)
- Uncountable amount of paths
- 40ms to determine action

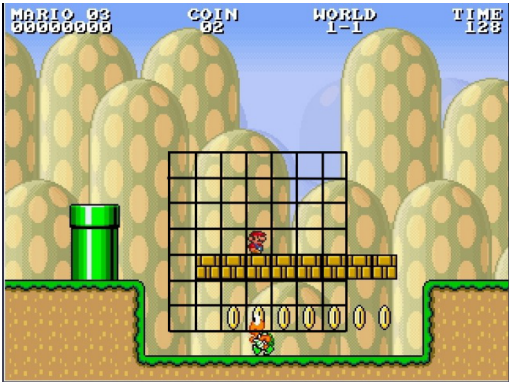
Overview of the environment

Example 2: Infinite Mario Challenge

Problem: Develop an agent that gets as far and fast as possible in a random generated level

Challenges:

- Limited perception of the agent
- Dynamic environment (Moving enemies and platforms)
- Uncountable amount of paths
- 40ms to determine action



Perception of the agent

Example 2: Infinite Mario Challenge

Problem: Develop an agent that gets as far and fast as possible in a random generated level

Challenges:

- Limited perception of the agent
- Dynamic environment (Moving enemies and platforms)
- Uncountable amount of paths
- 40ms to determine action

Inside a path planning algorithm

Topics

All topics revolve around the following questions, but are not limited to them:

- How can game-inspired problems transformed and solved as a algorithmic-problem?
- How can a game „optimally“ be played by an agent?
- Which software-architectures are suitable for developing games?
- How can procedural generation be used in games?
- Where is the sweet-spot between realistic and approximate simulation of a game world?

Module Workflow

1. Choice/assignment of topics
   - After the introductory lecture, you have to come up with your own topic!
   - Group work is encouraged!
2. Your task:
   - Familiarization with the topic + Literature review
   - Development of own approach
   - Presentation + Paper
   - Weekly meetings with all participants to discuss results and challenges
3. Presentation of your work + First version of paper
   - Specific deadlines **TBA** (but before end of lecture period)
4. Final camera-ready paper
   - Specific deadlines **TBA** (but before September)

Registration and Contact

- Restriction: max. 15 participants (seats will be assigned at 28.03.2025, 13:00)
- Enrollment in corresponding Stud.IP course
  - Link: https://studip.uni-rostock.de/dispatch.php/course/details?sem_id=c03ff0e6dd0b4ed4cfa9279964d99e96
- Questions via E-Mail to Andreas Ruscheinski
  - andreas.ruscheinski@uni-rostock.de