



NEidl / LDiCS

Algorithms in Game Environments

Organizational Matters

Andreas Ruscheinski

Computer Science Division, University of Rostock

Motivation

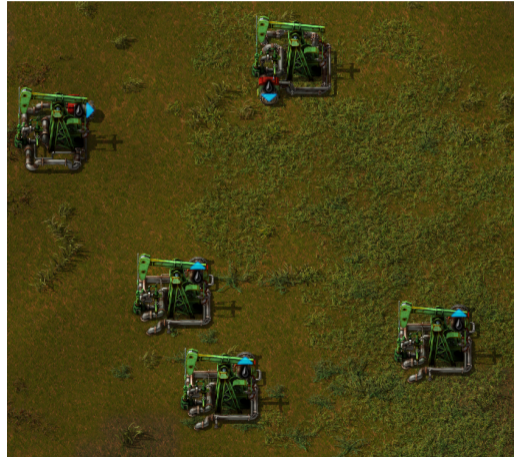
- Games are an interesting testbed for studying, developing and evaluating algorithms
- Reasons:
 - Structured Environment: Clear Rules and Objectives
 - Outcomes of scenarios can be easily measured and compared
 - Games involve complex decision-making and strategic planning
- Challenges:
 - Dynamic 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



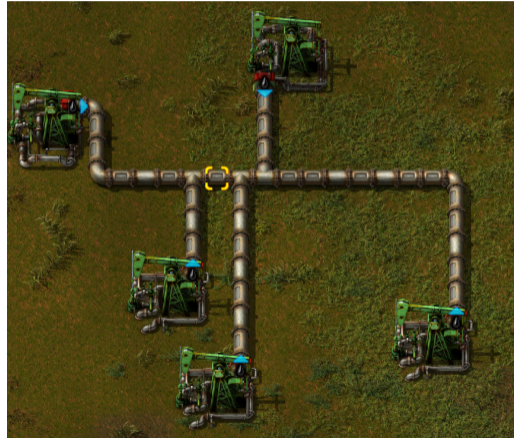
Connect the oil jacks with pipes

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



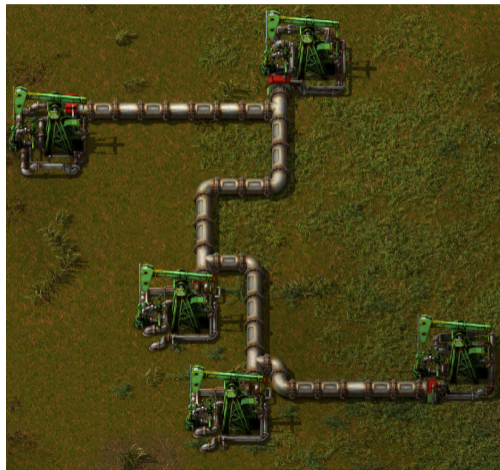
Solution: 37 pipes

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



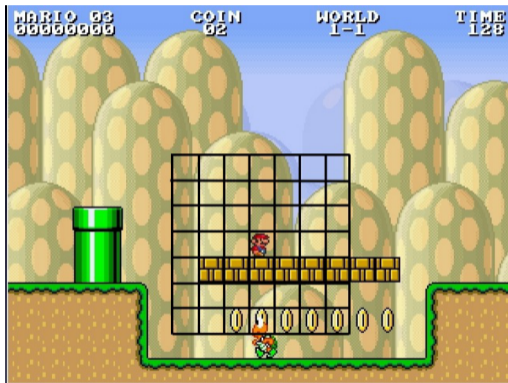
Overview of the environment

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 are centered around the following questions:

- How can game-inspired problems be solved via an algorithm?
- How can a game „optimally“ be played by an agent?

Module Workflow

1. Choice/assignment of topics
 - After the first lecture!
2. Your task:
 - Familiarization with the topic
 - Literature review
 - Development of own approach
 - Presentation + Paper
 - Regular appointments for consultation and advices
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)

Remarks on the Workflow

- Turn in slides and paper as PDF document via email
 - Time stamp of ITMZ mail server applies to all deadlines
- Extension of deadlines **only possible**
 - in case of important reasons
 - before the original deadline has expired
 - with a valid medical certificate in case of sickness
- Own initiative is part of the seminar
 - See it as a dry run for your master's thesis
 - to familiarize with the topic and related work
 - to elaborate on concepts, terms, and ideas
 - Nevertheless, ask for help if (really) necessary
- If you **plagiarize** (parts of) slides, code, and/or paper, you will **fail** the course **without further warning!**

Registration and Contact

- Restriction: max. 15 participants (seats will be assigned at 31.03.2023, 23:59)
- Enrollment in corresponding Stud.IP course
 - Link: https://studip.uni-rostock.de/dispatch.php/course/details?sem_id=32957a90370d6229560bd36740bb5207
- Questions via E-Mail to Andreas Ruscheinski
 - andreas.ruscheinski@uni-rostock.de