Organization of the lecture

Algorithms and Data Structures 2 – Motion Planning and its applications

University of Applied Sciences Stuttgart

Dr. Daniel Schneider

Moodle

- Link

About me

2007-2011: Bachelor of Mathematics

2011-2013: Master of Mathematics

2013-2017: PhD in Computer Science.

"Rigid and Deformable Motion and Disassembly Planning, with a Focus on the Digital Mockup Process in the Automotive Industry"

2017-2020: IT Project lead in the Digital Factory of Daimler AG

2020-today: IT Solution Architect for Artificial Intelligence at Mercedes-Benz AG

Example Projects:

1) Proprietary Software for the change management of a car production line
2) Implementation of an OOTB Software Solution for the digital validation of newly planned cars.

   → *Examples for the here presented algorithms*
3) Artificial Intelligence for quality insurance.

Short summary of the lecture

Data Structures and Algorithms 2

Questions we will answer in this lecture:

- How can we teach a computer to move objects in a given environment.
- Object like: Robots, Cars, Vehicle Parts, boats, UAVs (unmanned, uninhabited, unpiloted aerial vehicle)….
- What data structures are needed to guide an object.
- What algorithms exist for solving that problem

Literature

Basic book (Part I + II)

- Planning Algorithms by **LaValle**, http://planning.cs.uiuc.edu/

Many papers:

Motion Planning: The Essentials by **LaValle,** http://msl.cs.illinois.edu/~lavalle/papers/Lav11b.pdf

….

Remark

- The literature is used to look up details.
- The ideas and exercises are presented in the lecture and practically done in the study work

About the lecture

The presentations alone is NOT sufficient for understanding and following the lecture.

The following is needed:

- Active participation in the lecture.
- Understanding the data structures and algorithms.
- Asking questions?
- Work on the practical work study.

About the lecture

- The lecture will be held with a presentation.
- The lecture is organized in parts of lectures. One part does not necessary means 1.5h. There are lectures that will be short and lectures that will take more time.
- Each part of the lecture tries to capture a single topic.
- I try to recap in each lecture the main ideas of the previous ones. Your time to ask questions.

What skills are needed for the lecture?

- Basic mathematic knowledge. Especially in linear algebra.
- Good programming skills. Meaning:
  - You must be able to write code on your own.
  - You must be able to debug code on your own.
  - You must be able to program basic user interfaces (GUI)
  - **Note:** This lecture is not a lecture about coding. Coding is a prerequisite
- You are allowed to use **any** higher programming language as long as you can fulfil the above mentioned. Ideas: Java, C++, Python. Even: Webapps stuff (JS aka. React, Angular, with math backend)

  → use the one you are most familiar with.

What skills are needed for the lecture?

- Basic knowledge of algorithms and data structures like:
  - How can you sort.
  - How are graphs defined.
  - How can you find a shortest path in a graph (aka Dijkstra algorithm)
  - How does hashing work.
  - How can you travers data structures efficiently (linking vs. indexing)
  - Basic knowledge about complexity. How much operations are needed for a task?

How is the lecture organized?

Theory (50%) About 40% math, 60% IT

- Learn the data structures and algorithms.
- Learn the math behind those algorithms.

Practical work study (50%) About 10% math, 90% IT

- Implement the learned ideas with examples.
- Read papers and implement them in the your program.
- Get benchmark examples and benchmark the performance of your algorithms.
- I will help out with explanations of the algorithms. Help out with programming only for Java, C++ and Python→ As said: you need to be able to work on your own.

About the practical work study

Part of passing the exam is passing the practical work study

- At the end there will we simple benchmarks for your algorithms. If you algorithm can solve them → passed
- Presentation of your program → passed
- Presentation of at least one paper idea with your program → passed
- Answer questions about your code → passed

About the exam

Written Exam

- Duration of the exam: 120min
- Pen and Paper exam. No lecture material, no books, no media.
  - Again: You need to have an understanding of the algorithms. Practical work helps a lot. What you have programmed → You can easily explain.
- Simple calculator allowed but probably not needed.
- Exercises that we do during the lecture (and similar to them) can be part of the exam.
- Exercises as homework can also be part of the exam.
- You need to know your study work and what you have implemented.
- The exercises in the exam will we sorted with ascending difficulty.
- Content of the practical work study is also part of the exam