Software Lab
Computational Engineering Science
Pusher Mechanism

Aaron Albert Floerke, Arseniy Kholod, Xinyang Song, Yanliang Zhu

Supervisor: Dr. rer. nat. Markus Towara*

18th November 2024

*Informatik 12: Software and Tools for Computational Engineering, RWTH Aachen University, info@stce.rwth-aachen.de

Contents

1 Analysis **4**
- 1.1 User Requirements . . . 4
- 1.2 System Requirements . . . 4
- 1.3 Theory . . . 4

2 Design **4**
- 2.1 Principal Components and Third-Party Software . . . 4
- 2.2 Class Models . . . 4

3 Implementation **4**
- 3.1 Development Infrastructure . . . 4
- 3.2 Source Code . . . 4
- 3.3 Software Tests . . . 5

4 Project Management **5**

A User Documentation **5**
- A.1 Building . . . 5
- A.2 Testing . . . 5
- A.3 Running . . . 5

Preface

- administrative information about the project (e.g, topic issued by which institute)
- fit of topic into study program (e.g, sufficient prior knowledge)
- acknowledgement of supervision

1 Analysis

1.1 User Requirements

user requirements explained (includes essential information and references into literature on technical background of the topic, e.g, [1]) based on UML Use Case diagram(s)

1.2 System Requirements

functional and non-functional system requirements explained

1.3 Theory

explain geometry

2 Design

2.1 Principal Components and Third-Party Software

libraries that you built on explained briefly and references to further information

2.2 Class Models

UML Class diagram(s) and description; should link into overall design through reference of application programming interfaces (API) of third-party software

3 Implementation

3.1 Development Infrastructure

programming language, compiler, run time libraries, target platform (hardware, operating system)

3.2 Source Code

overview of source code structure (file names, directories); build instructions; references into source code documentation e.g, doxygen[1]; short (!) code listings

```
1 #include<iostream>
2 int main() {
3   std::cout << "Leave me alone world!" << std::endl;
4   return 42;
5 }
```

if helpful (must come with detailed explanation)

[1] https://github.com/doxygen/doxygen

3.3 Software Tests

e.g, googletest[2]

4 Project Management

who did what, when, and why; organization of collaboration, i.e. [online] meetings, software version control (e.g, git[3]

References

[1] Adam Ries. *Rechenung auff der Linihen und Federn.* Annaberg, 1522.

A User Documentation

A.1 Building

e.g, using cmake[4] and make[5]

A.2 Testing

e.g, `make test`

A.3 Running

documented sample session(s); e.g, `make run`

[2] `https://github.com/google/googletest`
[3] `https://git.rwth-aachen.de`
[4] `https://cmake.org/`
[5] `https://www.gnu.org/software/make/`