

System Specification

Robo Ducks

Project Name	Roboducks
Project Leader	Florentin Gewessler
Document state	In process
Version	V. 0.1

Revisions

Date	Author	Change
test date	test name	test use

Contents

1	1 Initial Situation and Goal	3
1.1	1.1 Initial Situation	3
1.2	1.1.1 Application Domain	4
1.2.1	1.1.2 Glossary	5
1.2.2	1.1.3 Model of the Application Domain	6
1.2.3	1.1.4 Overview of the Business Processes	6
1.2.4	1.1.5 Description of the Business Processes	6
1.3	1.2 Goal Definition	6
2	2 Functional Requirements	7
2.1	2.1 Use Case Diagrams	7
2.2	2.2 Use Case Details	7
2.2.1	2.2.1 Characteristic Information	7
2.2.2	2.2.2 GUI to call the use case	7
2.2.3	2.2.3 GUIs for the standard use	7
2.2.4	2.2.4 Scenarios for non-standard uses (bad cases or work around cases)	7
2.2.5	2.2.5 GUIs for the non-standard uses	7
2.2.6	2.2.6 Workflow	7
2.2.7	2.2.7 Open Points	7
3	3 Non-functional Requirements	8
4	4 Quantity Structure	9
4.1	4.1 System Architecture and Interfaces	10
4.2	Acceptance Criteria	11
4.3	List of Abbreviations	12
4.4	References	13

1 Initial Situation and Goal

1.1 Initial Situation

Our goal is to participate in the German Open Standard Platform League. The German Open Standard Platform League is a soccer league where all teams participate using the same robot, the NAO robot from SoftBank Robotics. These robots play fully autonomously and each one takes decisions separately from the others, but they still have to play as a team by using communications. The teams play on a green field with white lines and goal posts, with no other landmarks, and the ball consists in a realistic white and black soccer one. These game characteristics generate a very challenging scenario, which allows improving the league every year. Our competition are mostly teams from universities like the team B-Human which comes from the university Bremen. In this field it is very important to have sponsors because the most schools and universities can not effort too many robots. We have talked to some of the other teams and they said they would need about 50.000 Euros a year. As we mentioned earlier, our main goal is to participate in the German Open Standard Platform League but we can break this down to many sub goals. The first sub goal would be that we finish the framework, which we call “Duckburg”, till the 24.12.2018. The framework is the base of our software which will contain the basic functions of a system.

1.2 Application Domain

Our software will be used on the NAO-Robots. With our software it will be possible that the robots play soccer in teams of five. Our software will also be optimized for the limited resources of the NAO Robots. The robots only have a ATOM Z530 CPU with 1.6 Gigahertz, 1 GB of RAM and up to 10 GB of storage. Furthermore, our software has to be very modular and expandable. This is granted through a specific system architecture which we will elaborate later in the system architecture and interfaces section.

All in all, we can summarize our project in three major steps. The first one would be, that the Duckburg, our basic framework, is fully implemented because that is a major requirement to start further development.

The second step would be the development of the different Engines and Agents which will do specific tasks like walking or kicking a ball. The challenges are very varied depending on which task you like to develop. For example if we want to write an Engine and the associated agents we will have major problems with to balance of the robot but if we want to develop a Engine for recognising a ball we will have to face other problems.

The last step would be the the communication between one robot to another one. Here we have to deal with the limitations of the wlan module of the robots which allows only a small number of kilobytes between the robots.

1.2.1 Glossary

1.2.2 Model of the Application Domain

1.2.3 Overview of the Business Processes

do not exist

1.2.4 Description of the Business Processes

1.3 Goal Definition

2 Functional Requirements

2.1 Use Case Diagrams

2.2 Use Case Details

2.2.1 Characteristic Information

2.2.2 GUI to call the use case

2.2.3 GUIs for the standard use

2.2.4 Scenarios for non-standard uses (bad cases or work around cases)

2.2.5 GUIs for the non-standard uses

2.2.6 Workflow

2.2.7 Open Points

3 Non-functional Requirements

4 Quantity Structure

4.1 System Architecture and Interfaces

4.2 Acceptance Criteria

4.3 List of Abbreviations

4.4 References