Software design
Team project – Deliverable 1

Team number: 16

Team members:

Name	Student Nr.	E-mail
Eliane Kadouch	2623437	e.r.kadouch@student.vu.nl
Karol Komorniczak	2622762	karol.komorniczak@gmail.com
Maciej Juzoń	2618287	maciek.juzon@gmail.com
Gaweł Jakimiak	2617124	gawel.jakimiak.ib@gmail.com
Jiyoung Oh	2610337	godinggool@gmail.com

Table of Contents

1. Introduction

Author(s): Jiyoung Oh

The goal of our project is to develop a system that searches an unknown environment that was created for a number of moving rovers with different searching features implemented. In order to achieve that we follow the rules of software design. Namely, we present all the requirements of the system and use them to develop a use case diagram, class diagram and sequence diagrams, which will help us in implementing our system in the Simbad Simulator. However, we first need to make certain assumptions for the ROBOSEARCH system. Firstly, the size of the environment is 25 x 25. In the environment, six static obstacles are devised for testing the functions of rovers. Before starting the mission, operator of our system will determine the number of boxes to be found in the environment as well as their coordinates. Additionally, we have decided that in the begging all the boxes will be yellow in order to develop efficient system just for one color and then eventually extend it to find boxes of different colors. Finally, operator will be able to stop the system whenever he or she wants

In principle, every rover in our system can move around anywhere in the environment avoiding obstacles autonomously and searching for the right goal. Our obstacle avoidance strategy is strictly related to our goal finding strategy, as when a rover stumbles upon an object it will begin the process of determining what it is and where to go next. Our robot will have 3 operators, each one of the responsible for different things. Firstly, when the robot will hit anything, the photographer will make use of the robots camera in order to take a series of photos when the robot will be rotating around its own axis. If one of the photos will contain more than 45% of yellow, the photographer will notify central station that one of the boxes was found. If there will not be a photo containing 45% of yellow, photographer will send the best photo (containing the highest ration of yellow to not yellow pixels)to navigator, which based on that will determine the direction in which the rover should drive and finally based on that the driver will rotate the robot and make it drive in given direction in order to reach the desired box. This process will be repeated until all boxes are found.

Moreover, all the operators will be controlled by a central station, which will act as a brain of our system. It will be responsible for assigning tasks to different operator and monitoring the progress of our mission. The search concludes once all the goal objects have been successfully found In the end it will also be responsible for informing the operator about the position of the boxes found, so the user will be able to assess the accuracy of the system thanks to knowing the position of the boxes from the beginning.

2. Requirements Specification

Author(s): Karol Komorniczak, Maciej Juzon, Eliane Kadouch

This chapter contains the specification and UML representations of all the requirements.

2.1 Requirements

Functional requirements

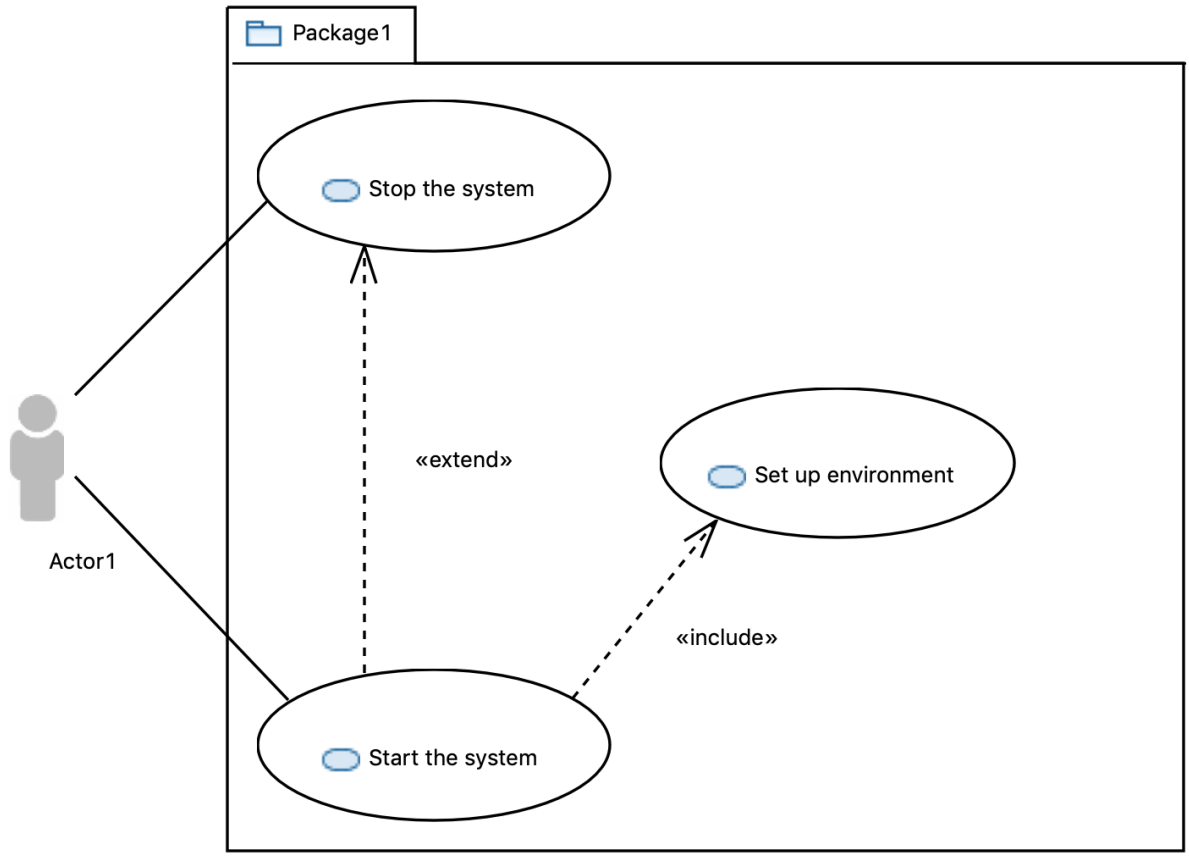
#	Short Name	Description
F1	Obstacle Avoidance	The rovers shall move freely in the environment until its bumpers detects and obstacle
F2	Cooperation/Work Division	The operators shall cooperate and exchange information in order to find the goal
F3	Finding the Goal	The rovers shall stop upon finding the desirable block
F4	Operators	The rovers shall be operated by 3 operators: Photographer, Navigator, Driver
F5	Central Station	Central station shall control operators in such a way that only one can control a robot at a time
F6	Taking photos	The photographer shall use of the robots camera in order to take a series of photos when the robot will be rotating around its own axis.
F7	Classifying	The photographer shall classify the photos based on the ration of yellow to non-yellow pixels
F8	Navigating	The navigator shall calculate direction in which the robot shall drive to find the goal based on the photographer classification
F9	Notifying observers	Central Station shall notify all observers after change of state

Non-functional requirements

#	Short Name	Description & reasoning
NF1	Obstacle avoidance [Performance]	Each rover shall react to the presence of an obstacle within 30 milliseconds.
NF2	Finding the Goal [Performance]	Rovers shall be able to recognize the barrier in front of it whether the border of the environment or an obstacle within 30 milliseconds.
NF3	Reliability	The system shouldn't crash
NF4	Speed	The boxes should be recognized and found in a quick Manner
NF5	Usability	System should be easy to interact with
NF6	Robot accessibility	Robot should be accessible for all observers
NF7	Communication	Robot operator should only communicate via central station

2.2 Use Cases

Use Case Diagram



Name	Start the System
Short description	The operator initializes the system
Precondition	The operator has access to the system
Post condition	The system has been initialized
Error situations	The system is unavailable
System state in the event of an error	The rovers cannot be initialized
Actors	Operator
Trigger	The operator wants to enable the system
Standard process	(1) Operator initializes the system (2) Operator can set up the environment

Name	Set up environment
Short description	The user is decides how many boxes will have to be found and decides where they will be positioned in the environment
Precondition	The system has been initialized
Post condition	The environment is set up and mission can be started
Error situations	Operator inputs 0 as number of boxes to be found
System state in the event of an error	System ask the user to input number of boxes again
Actors	Operator
Trigger	The operator wants to set up the environment
Standard process	(1) Operator is asked to give number of boxes to be found (2) Operator is asked to give x coordinate (3) Operator is asked to give z coordinate (4) The boxes are positioned and robots start moving

Name	Stop the System
Short description	User can stop the system whenever he or she wants
Precondition	The system has been initialized
Post condition	The system is stopped
Error situations	The system is not initialized
System state in the event of an error	The system cannot be stopped
Actors	Operator
Trigger	The operator wants to stop the system
Standard process	(1) Operator stops the system

3. Implementation remarks

Author(s): Gaweł Jakimiak

For the first assignment, we have only made minor changes to the environment and robots code. We have prepared completely different implementation for deliverable 2, so we will not edit this part, as we think that in the second part we show sufficient knowledge of simbad framework ☺

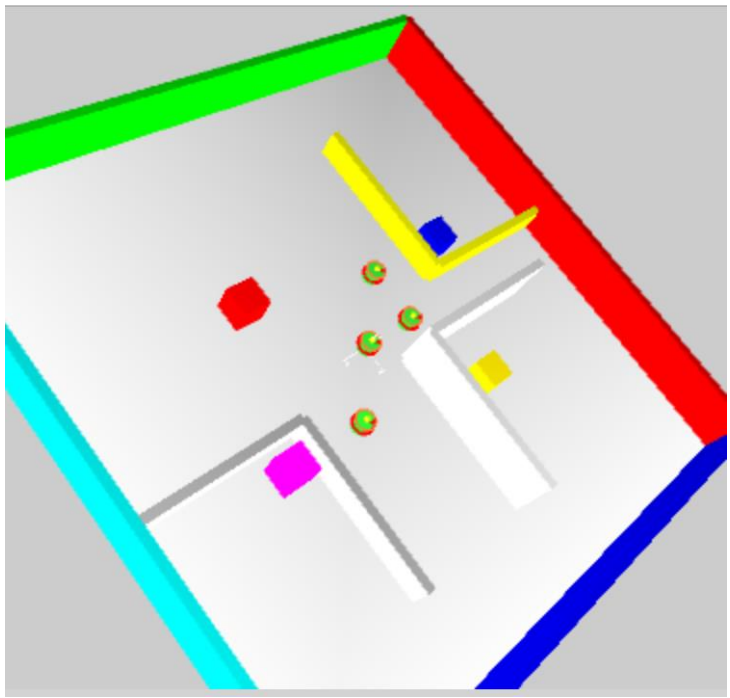
Robots:

- Added new robot
- Changed behavior of the first three robots so now they act differently:

- Robot - Original code, no alterations
- SRobot - We have set higher translational velocity
- Brobot - Robot starts from the initial position after every collision
- Qrobot - Turns 180 degrees after every collision

Environment:

- Added some walls
- Changed the color of the walls
- Enlarged the size of the world
- Added two boxes



4. References

Osterweil, L. (2006). *Proceedings / 28th International Conference on Software Engineering*. New York, NY.: Association for Computing Machinery, pp.534-543.

Seidl, M., Scholz, M., Huemer, C. and Kappel, G. (2015). *UML @ classroom*. Cham: Springer.

Simbad.sourceforge.net. (2019). *Simbad 3d Robot Simulator.*. [online] Available at: http://simbad.sourceforge.net/index.php [Accessed 22 Feb. 2019].