First development documents of Mars Rover

Judith van Stegeren and Mirjam van Nahmen

December 1, 2014

Specifications

The main goal of the robot is: find all three lakes and measure the temperature.

Functional requirements:

The robot must be able to ...

- wander the area.
- recognize a lake and determine its color and not go across the lake.
- recognize the white barrier and not cross it.
- notice obstacles in the area and avoid them while wandering.
- lower the temperature sensor and measure the temperature in a lake.
- store the temperature with the color of the lake.
- see the difference between black, white, red, green and blue.

Usability:

The robot must ...

- be programmed by stating its behavior in a DSL and then generating code from that specification.
- not drive too fast, so that if the robot is heading for the edge of the table, the user still has time to interfere.
- provide feedback for the user about measurements it makes.
- provide feedback for the user about the state of the program for debugging purposes.
- provide feedback for the user when the robot is finished (show the table with measurements).
- provide feedback for the user about errors/bugs.

Reliability:

- The robot must stop if an error/bug is detected to avoid to go off of the table.
- The robot components must be tested before it will work with the whole system.
- The sensors must be calibrated before the program starts.

Performance:

- If the light sensors spot the white border, the robot must react immediately to avoid falling off the table.
- If the light sensors spot the color border of a lake, the robot must react immediately to avoid falling in the lake.
- All three lakes must be found within a reasonable amount of time.
- If the robot has already measured a lake, he must skip the measurements if he encounters that same lake again.

Supportability:

- the DSL can also be used as grammar for other languages, it needs only another code generator

Proposal deployment

List of the actuators:

2 motors

1 motor for the temperature sensor

List of the sensors:

1 lamp

1 temperature sensor

2 light sensors

1 color sensor

1 ultra sonar sensor

2 touch sensors

There can be 3 actuators and 4 sensors on 1 brick. Since it is vital that the motor does not fall off the table, we want that the communication between the light sensors and the motors does happen via bluetooth. Also bumping against a rock is an important issue such that it seems to be useful to connect the touch sensors also directly with the master brick.

Since the color sensor, the motor for the temperature sensor and the temperature sensor all have to do with the same functionality, it would seem logical to put these sensors and actuators on the same brick. Also the measurement of the distance with the ultra sonar sensor is not crucial it is logic to put them on the brick for the measurements.

Brick 1 (Master):

crucial processes (driving without bumping or falling off the table)

Motor 1

Motor 2

Light 1

Light 2

Touch 1

Touch 2

Brick 2:

measurements (color of a lake, temperature and distance detection)

Motor Temperature

Temperature

Ultra sonar

Color

Use case:

Robot wanders around the area without falling off the black table, the borders are marked with a white line. We need both motors and both the light sensors.

On detection of a colored lake, the robot lowers the temperature sensor and measures the temperature and store it with the color of the lake. We need the color detector, the motor for the temperature sensor and the temperature sensor itself.

Robot wanders around the area, without bumping into objects. We need both motors and the two touch sensors and the sonar sensor.

If the robot finished the measurement of the last lake it stops wandering and shows the table with the temperatures of all lakes on his display.

If the robot detects a rock with the ultra sonar sensor, it changes the direction and makes a bend to avoid bumping the rock.

If the robot bumps against a rock, the robot should stop both motors immediately. After stopping the robot should go back straight to avoid to fall off the table during driving backwards. Afterwards the robot makes a turn to change the direction and starts again with wandering.

If the robot detects a with line of the border it goes straight backwards and makes a turn to the opposite side of the light sensor which detected the white line.

Risk analysis

Id	Type	Description	Probability	Severity	Weight	Mitigation	Contingency
1	Technical risk	The computer with the installed software breaks down and we lose access to our work.	3	4	High	Push all work to the repository, regularly.	Find a replacement for the broken laptop or use one of the lab computers.
2	Organizational risk	The robot is not available for testing.	3	4	High	Regulary test small pieces of code, so that debugging is not a lot of work.	Postpone testing until later or wait until the robot is free. Use the lab when when we know there won't be other students around (early morning).
3	Technical risk	The DSL does not fit with the implementation.	3	2	Medium	Change the DSL to a lower level of abstraction.	
4	Organizational risk	One of our team members become ill.	3	2	Small	Healthy living	The other one continues working on the project and the ill person works as much as possible from home. Keep communicating via email/chat.
5	Requirements risk	The requirements contains ideas which are not compatible with the idea of the DSL.	2	2	Medium		Change requirements until it is possible to implement.
6	Technical risk	Problems with the bluetooth communication between the two bricks	5	3	Medium	Test bluetooth functionality as soon as possible.	Ask for help from the teacher or other students.
7	Organizational risk	Lab partners have different schedules which makes it hard to find time to work together	5	3	Medium	The lab partners don't plan anything on Wednesday afternoon, so that there is at least one moment in the week when they are both free to work on the project	Reschedule work or work together during the weekends.