



M.E.N.A.C.E.

Motorized Embedded Network-connected smArt Car Entity

DIT524 - PROJECT: SYSTEMS DEVELOPMENT

Date 2017-02-14

Isak Magnusson
Kosara Golemshinska;
Laiz H. B. de Figueroa;
Melinda Ivók;

Syeda Elham Shahed;
Rema Salman;
Nina Uljanić

PROJECT OVERVIEW

One of the project's main goal is to build a mobile application that will be used to control the car. The application will have a control menu. It will consist of several buttons to direct the car and it will also provide feedback received from the car under certain circumstances.

The main feature also includes avoiding obstacles by the car. The car will constantly register the input from the sensors and calculate the distance from the nearest obstacle. If said distance is under a predefined minimum, the car will change its route autonomously and also based on the instructions from the user under certain situation.

Another feature that will be implemented is blinking lights. The car will use two lights - one on each side - to show a change of direction, i.e., the light on the left side and on the right side will blink when the car is turning to the left and right respectively, and after the turn is completed, the blinking will stop. The lights will also blink when the car faces an obstacle as an alert signal.

A camera will be incorporated to detect a light and follow it. This is an optional feature for this project.

PROJECT ORGANIZATION

The tasks will be split between all team members, and depending on the difficulty of the given tasks, some members might have to work in pairs and some alone. Pairing will depend on the skills of the group members – members with low knowledge will not work alone on a task that might be hard to solve for them on their own, or work with another team member who might not be up to the task.

Besides the role of the developers, each team member will also take on the role of a tester. Since the tasks are split to make the workload easier to handle, we are going to exchange codes between team members and thoroughly test fellow teammate's code.



Since we are using Agile in the development, we needed a scrum master to keep everything in check. Kosara Golemshinska has volunteered to take this role.

PROJECT DELIVERABLES

List of products/functions deliverables:

- * Mandatory deliverables
 - * The car's movement functionalities.
 - * The mobile application (control).
 - * Avoiding obstacles functionality.
 - * Light's blinking functionality, when the car turns either to left or right, and when the car faces any obstacles.
- * Optional deliverables
 - * Following light functionality, where the car should be turned to the automated mode.
- * Documentations' deliverables:
 - * The project Proposal.
 - * The project reports and the presentations for each sprint.
 - * The Project Plan.
 - * The Requirements Specification.
 - * The Design/Architecture Specification.

HARDWARE AND SOFTWARE RESOURCE REQUIREMENTS

List of hardware parts planned to be used in the project:

Camera	Ultrasonic sensors
Wifi module	Wires (M-M)
Raspberry Pi	Wires (M-F)
Gyroscope	USB cable
Android mobile phone	Breadboards
Infrared sensors	2 lights
ZX Gesture sensors	Arduino System
SmartCar platform	4 Wheels
Bluetooth module	



M.E.N.A.C.E.

Motorized Embedded Network-connected smart Car Entity

The list of software that will be used by the group is:

Android SDK	GitHub
Trello	Google Drive
Slack	Virtual Box
Eclipse	Visual Paradigm
Arduino	Eye Automate

WORK BREAKDOWN

To make a well-functioning code, we have to first analyze the requirements of how the car and the functions should be. After that, the designing of the architecture will take place, where we create the software structure with both low level and high level designs. We will also write test cases to make sure our code satisfies the expectations. Only after these comes the actual coding and the debugging. All this activities At the end, our product will be released.

The main functions are: moving the car, controlling, blinking lights, avoiding obstacles, following a light. These can be broken down into smaller features, whose coding and testing will be the activities in our development.

* Moving the car

- it moves in different directions
- it follows the commands from the application
- when in automated mode, it follows the light

* Controlling

- an application is made for controlling which is connected to the car
- the app has a pin code
- only one user can be logged in at a time
- the app has buttons to move the car forwards, backwards, left and right
- the user can change the mode of the car (automated/user-controlled)

* Blinking lights

- the lights blink only when the car turns left or right
- the blinking side is the same the car is turning towards
- the lights also blink when the car is automated
- when the car is stopped by an obstacle, the alert signal blinks while waiting for a user command



* Avoiding obstacles

when it is user-controlled, the car avoids obstacles by stopping and prompting the user for a new direction

the alert signal is on while the car waits

when it is automated, the car evaluates the surroundings and changes direction by itself to achieve the destination.

the car can detect obstacles in movement and in static.

* Following a light

the car is automated while following a light

the blinking lights feature works in this mode as well

the car can detect obstacles

the automated mode can be turned off any time.

PROJECT SCHEDULE

* Sprint 1

The group will write the requirement specification, one of the documentation deliverables. Based on that they will create user stories, and after that the estimation should be done for each user story with the planning poker. And a sprint backlog should be created.

The code standards will be defined at the beginning in order to avoid future issues.

* Sprint 2

The group will design the architecture of the software and create the architecture specification, one of the documentation deliverables.

* Sprint 3

The group will work in the main feature to move the car autonomously.

* Sprint 4

The group will work in other main feature where the car will blink the light autonomously when performing a turning.

* Prototype demo

The car will be doing the movements and the light blinking autonomously.

* Sprint 5

The group will start to develop the control aiming the buttons for giving the car commands for that sprint. And also the avoid obstacle feature will be developed in this sprint.



* Sprint 6

The group will work in the UI for the control and to insert the pin code. Also will create the car's mode for follow user's commands or to be automated.

* Sprint 7

The bonus feature, follow the light, will be attempted to be started and finished by the group in this sprint. Concomitantly the debugging will be performed. And finally the final presentation will be prepared.

* Final presentation

The car should be presented with the mandatory deliverables working, including the mobile application, and, if manageable, with the bonus feature.

RESOURCE PLAN

The group will use the repository provided by the course on GitHub (DIT524-V17/group-9) to store our code and also to share it among our team members. The documentation is normally developed on Google Drive in a group folder, there it can be stored, when it is finished. Also a zip file with all the documentation developed per month will be pushed to the same repository on GitHub and to the group channel in Slack.

COMMUNICATIONS BETWEEN STAKEHOLDERS

The group is meeting on Patricia, second floor; following the schedule below. The breaks on the meetings are to be agreed upon at the start of each meeting depending on the workload and circumstances.

- * Mon: 14.00-15.00
- * Wed: 10.00-12.00
- * Thu: Extra meeting if needed
- * Fri: 10.00-14.00

The tool agreed for performing the group's communication is Slack, then all the information will be posted there with a pin. And by summarizing the decisions and progress made during each meeting so that everybody has access to that information.



The rules agreed among the team members regarding to communication are:

- * If a member is unreachable during the week for more than 24 hours or for more than 48 hours during the weekend without prior notice or a sufficient reason, a log is kept and after the first such occurrence, the team discuss the issue together. If no resolution is reached or the lack of contact without notice/reason repeats, the scrum master will discuss the issue with the teaching assistants. If no resolution is reached again, the scrum master will contact the course teaching staff.
- * If a team member misses 1 meeting without prior notice or a sufficient reason, the team will discuss this with this member in order to guarantee presence. After 2 such missed meetings, the scrum master will contact the teaching assistants. After the third such meeting, the scrum master will contact the course teaching staff.
- * The same 2-3 strike policy described above applies to any inappropriate or offensive behavior between the team members as well as missing a delivery date or submitting a delivery that does not correspond to the agreed upon quality and quantity.

RISK MANAGEMENT

The risk management comprises four processes:

Risk Identification contains the unique Id, date of creation, description.

Risk Analysis contains the likelihood and the impact of the risk analyzed.

Risk Planning contains the preventive measures to lower down the likelihood, the mitigation, and constant monitoring of process to identify new risks.

Risk Monitoring should be integrated on the project activities. It contains the tracking of the risks and the review of it (close, lower the risk probability, so on), the preparation of status reports, search for new risks.

At this point of the project only the first three will be taken into account. And the result of it is expressed on the Tables 1 to 14, in which they are sorted from the highest probability to happen to the lowest probability to happen and from severe to minor consequences.



Table 1. Risk 01 with high probability and severe consequences.

Unique Id: 01	Created on: 11/02/2017
Title	Unfamiliar languages
Description	The students have no knowledge of languages other than Java.
Probability to happen:	High (>50%)
Possible consequence:	Severe
Risk avoidance:	It is not avoidable.
Risk mitigation:	Dedicate a time slot to educate the team on the skills they will need. Turn to the TAs and other students for additional guidance.

Table 2. Risk 02 with high probability and severe consequences.

Unique Id: 02	Created on: 11/02/2017
Title	A developer has lack of key skills
Description	The students don't have necessary knowledge to work on the tasks assigned.
Probability to happen:	High (>50%)
Possible consequence:	Severe
Risk avoidance:	Ask for help from other team members or TAs and try to learn. Try to keep the activities related to the skills each team member has.
Risk mitigation:	Use pair programming to equally distribute the skills availability.

Table 3. Risk 03 with high probability and medium consequences.

Unique Id: 03	Created on: 11/02/2017
Title	Size underestimate
Description	The size of a feature, functionality and task has been underestimated.
Probability to happen:	High (>50%)
Possible consequence:	Medium
Risk avoidance:	Playing risk poker to estimate the user stories can lead us to a more accurate estimation. Afterwards, create subtasks for each user story and estimate as well. The sum of these subtasks estimation should be the final estimation for the related user story.
Risk mitigation:	Reduce the sprint backlog and depending on how long it takes, reduce the requirements number.



Table 4. Risk 04 with medium probability and severe consequences.

Unique Id: 04	Created on: 11/02/2017
Title	Multiple Stakeholders
Description	More than one stakeholder was identified for the project, having that in consideration, different people can have different expectations about the same product feature.
Probability to happen:	Medium (25 - 50%)
Possible consequence:	Severe
Risk avoidance:	It is not avoidable.
Risk mitigation:	Have regular meetings with stakeholders. and evaluate which of the ideas given is more suitable for the system, estimate it, in case it's a new requirement, and include it in the product backlog.

Table 5. Risk 05 with medium probability and severe consequences.

Unique Id: 05	Created on: 11/02/2017
Title	Car hardware problems
Description	The Arduino or any other car component does not function properly.
Probability to happen:	Medium (25 - 50%)
Possible consequence:	Severe
Risk avoidance:	It is not avoidable.
Risk mitigation:	Ask the TAs for assistance.

Table 6. Risk 06 with medium probability and severe consequences.

Unique Id: 06	Created on: 11/02/2017
Title	A developer is not committed to the project
Description	A developer does not take part in the project in some of the phases.
Probability to happen:	Medium (25 - 50%)
Possible consequence:	Severe
Risk avoidance:	It is not avoidable.
Risk mitigation:	Have a meeting where the group can talk with each other; if this does not show results, talk with the TAs and teachers.



Table 7. Risk 07 with medium probability and minor to severe consequences.

Unique Id: 07	Created on: 11/02/2017
Title	A developer misses meetings
Description	Students that for any reason miss more than one meeting consecutively.
Probability to happen:	Medium(25 - 50%)
Possible consequence:	Minor - Severe (Depends on the commitment relation and on the number of meetings missed)
Risk avoidance:	It is not avoidable in some cases, but in other cases if the team members feel that they are unique to the team and have a role to play, those members need to still be actively engaged.
Risk mitigation:	If the student misses more than two meetings, inform the TA and if they miss more than 3 meetings without any good excuse, inform the teachers.

Table 8. Risk 08 with medium probability and medium to severe consequences.

Unique Id: 08	Created on: 11/02/2017
Title	Tasks/Activities delays
Description	The individual tasks/activities are not delivered in time to be integrated with the others tasks/activities.
Probability to happen:	Medium (25 - 50%)
Possible Consequence:	Medium-Severe
Risk avoidance:	Deadline of tasks 2 days prior to the delivery day, if it doesn't have any other task depending on it. Try not to assign inter-depend tasks to different developers. Additionally, the scrum master should keep track of all tasks in development and the issues faced by the team members in order to try to solve it together with the task owner. If the problem carries on, a meeting should be organized to discuss and to try to solve it on a group level and afterwards, possibly on the TA level as well.
Risk mitigation:	Include this task/activity on the next sprint backlog.



Table 9. Risk 09 with medium probability and medium consequences.

Unique Id: 09	Created on: 11/02/2017
Title	Hardware unavailability
Description	If any element breaks, gets damaged, malfunctions, and/or is not available to be used in the project
Probability to happen:	Medium (25 - 50%)
Possible Consequence:	Medium (25 - 50%)
Risk avoidance:	It is not avoidable.
Risk mitigation:	Change the requirements.

Table 10. Risk 10 with low probability and severe consequences.

Unique Id: 10	Created on: 11/02/2017
Title	Requirements change
Description	If any changes of the requirements occur during the development process.
Probability to happen:	Low (<25%)
Possible consequence:	Severe
Risk avoidance:	Requirements ought to be well described, analyzed and discussed with the stakeholders at the beginning and on every sprint planning meeting.
Risk mitigation:	Do the estimation of this new requirement, ask the stakeholders about its priority, include it on the product backlog and see if it is possible to deliver all of it or if the team will need to take out something. Inform the stakeholders about any new addition and verify if it is appropriate to add it.

Table 11. Risk 11 with low probability and medium consequences.

Unique Id: 11	Created on: 11/02/2017
Title	Lack of communication
Description	Students that do not access Slack daily and miss the information available there.
Probability to happen:	Low (<25%)
Possible consequence:	Medium
Risk avoidance:	Engage team members to use the communication tool chosen.
Risk mitigation:	Make stand-up meetings to keep track of the information and the current task progress.



Table 12. Risk 12 with low probability and minor to medium consequences.

Unique Id: 12	Created on: 11/02/2017
Title	A developer leaves the project course
Description	Students that choose to leave the program and/or the course during the development process of the project.
Probability to happen:	Low (<25%)
Possible consequence:	Minor - Medium (Depends on the commitment relation)
Risk avoidance:	It is not avoidable, but in order to decrease the possibility for this to happen, a better commitment of the team members and distribution of the tasks is needed.
Risk mitigation:	Reduce the sprint backlog and depending on which phase it occur in, reduce the requirements number.

Table 13. Risk 13 with low probability and minor to medium consequences.

Unique Id:13	Created on: 11/02/2017
Title	A developer gets sick
Description	Students get sick during the development process for a long period.
Probability to happen:	Low (<25%)
Possible consequence:	Minor - Medium (Depends on the commitment relation)
Risk avoidance:	It is not avoidable.
Risk mitigation:	Reduce the sprint backlog and depending on how long it continues, reduce the requirements number.

Table 14. Risk 14 with low probability and minor consequences.

Unique Id: 14	Created on: 11/02/2017
Title	Computers/Softwares underperformance
Description	The chosen tools and/or the computers do not perform as expected, fail or stop working.
Probability to happen:	Low (<25%)
Possible consequence:	Minor
Risk avoidance:	It is not avoidable.
Risk mitigation:	Try to fix and/or reinstall as soon as possible, work with pair programming practice until the problem is fixed.