System Document

Project: ECSE 211 Final project
Task:
Document Version Number: 1.3 **Date:** 14/10/2017
Author: Justin Tremblay
Edit History:
[14/10/2017] Justin Tremblay: First draft. [15/10/2017] Justin Tremblay: Filled in most of the parts relating to software. [16/10/2017] Alex Hale: added TODO notes for sections we can't fill in yet; edited some writing; changed version system to WEEK.EDIT (e.g. it is currently week 1, edit 3 => 1.3)

Table of Content

1.0 System Model

TODO: block diagram of basic functions that are needed - should gradually be updated as implications of client requirements become clearer

1.1 Hardware Available and Capabilites

The hardware for this project will be the LEGO Mindstorm kit, including one EV3 brick, an assortment of motors and sensors, and a wide variety of connecting pieces for construction. Since the team is made up of three laboratory groups, up to three kits will be available, but it is unlikely that more than two EV3 bricks will be used.

2) TODO: electromechanical limitations

The EV3 brick contains a low-power ARM processor, so our computing speed is quite limited. Our

system memory is stored on a micro SD card, which is fast enough that it won't bottleneck our speed before the processor.

1.2 Software Available and Capabilites

This project will be written in Java, making use of the LejOS library for the LEGO Mindstorms EV3 brick.

Java is a high-level object oriented programming language with features like multi-threading and garbage collection. Its advantages are that it is easy to use and multi-platform. Its disadvantages are that, since it runs on a virtual machine, it adds a lot a overhead and makes it hard to optimize code on platforms such as the EV3, which has very limited resources. Java's memory management model is also a big disadvantage: since it uses garbage collection, it doesn't allow much control over the memory usage of the program and adds a lot of overhead compared to using pointers and manual memory management in languages such as C and C++. The last big disadvantage is how the language is compiled. When Java is compiled, it is converted to class files that are then compiled using JIT (just in time) compilation at run time, which limits the optimization of code, often making it suboptimal.

2.0 Compatibility

2.1 Hardware Compatibility

The physical robot design is limited to the items available in the LEGO EV3 Mindstorm kit. This kit has hundreds of pieces, but parts can only be connected together in a limited number of ways, limiting the design possibilities. It may be possible to produce custom parts using 3D printing - this option will be investigated later if necessary.

2.2 Software Compatibility

Every piece of software that was developed during the semester has been written in Java, meaning that every piece code we have can easily be adapted to work with an other.

Every tool we are using is multiplatform so it accomodates people using Windows, Linux and MacOS. Tools like Git also take care of the different text formatting of the different operating systems (line endings) and converts it all to the same format, making it very easy to work with multiple platforms.

3.0 Reusability

3.1 Hardware Reusability

Not much of the hardware design will be carried forward from the labs, because the robots in the labs made many assumptions. For example, the labs required the robot to have only one or two sensors and two or three motors equipped, whereas the project robot will have at least four of each (or more). The project robot will need to have an arm to grab the zipline, and in general will need to be larger and more stable to accomodate more sensors and motors.

2) TODO mechanical structures that will be copied for use in multiple parts of the robot

3.2 Software Reusability

As mentioned above, since every piece of code from the laboraties is written in Java, everything that has been done during the semester is reusable for the final project.

Software tools such as the odometer, the navigation and the localization will be essential to this project and will be taken form the previous laboratories and improved using the implementions of the multiple groups in the team.

4.0 Structures

4.1 Hardware Structure

1) TODO reasons for hardware choices 2) TODO reasons for electrical choices

4.2 Software Structure

1) TODO design of software structure

5.0 Methodologies

5.1 Hardware Methodology

1) TODO list of possible candidate solutions for parts of hte problem - come out of idea generaton and allow for critical analysis before final design is performed

5.2 Software Methodology

Software is going to be developed using the iterative development process. We are going to be constantly testing the software to find issues and adjust the design accordingly.

6.0 Tools

- Git / Github
- Eclipse
- Google Drive
- QOwnNotes / Markdown
- Slack
- lejOS
- Lego Mindstorms

7.0 Glossary of Terms