SYSTEM DOCUMENT

Project: DPM Final Design Project
Task: Explain the system to be used in the robot design.

Document Version Number: 5.0
Date: November 28^{th} 2019
Author: Brendan Marks

Edit History:
Brendan Marks - 2019/10/19 - Fill in the hardware capabilities section.
Brendan Marks - 2019/10/20 - Fill in the remaining sections of the document.
Stefan Barbu - 2019/10/20 - Fill in the initial system model.
Brendan Marks - 2019/11/15 - Update Structures Section.
Aurelia Haas - 2019/11/28 - Revision of the entire document

1.0 TABLE OF CONTENTS

2.0 SYSTEM MODEL

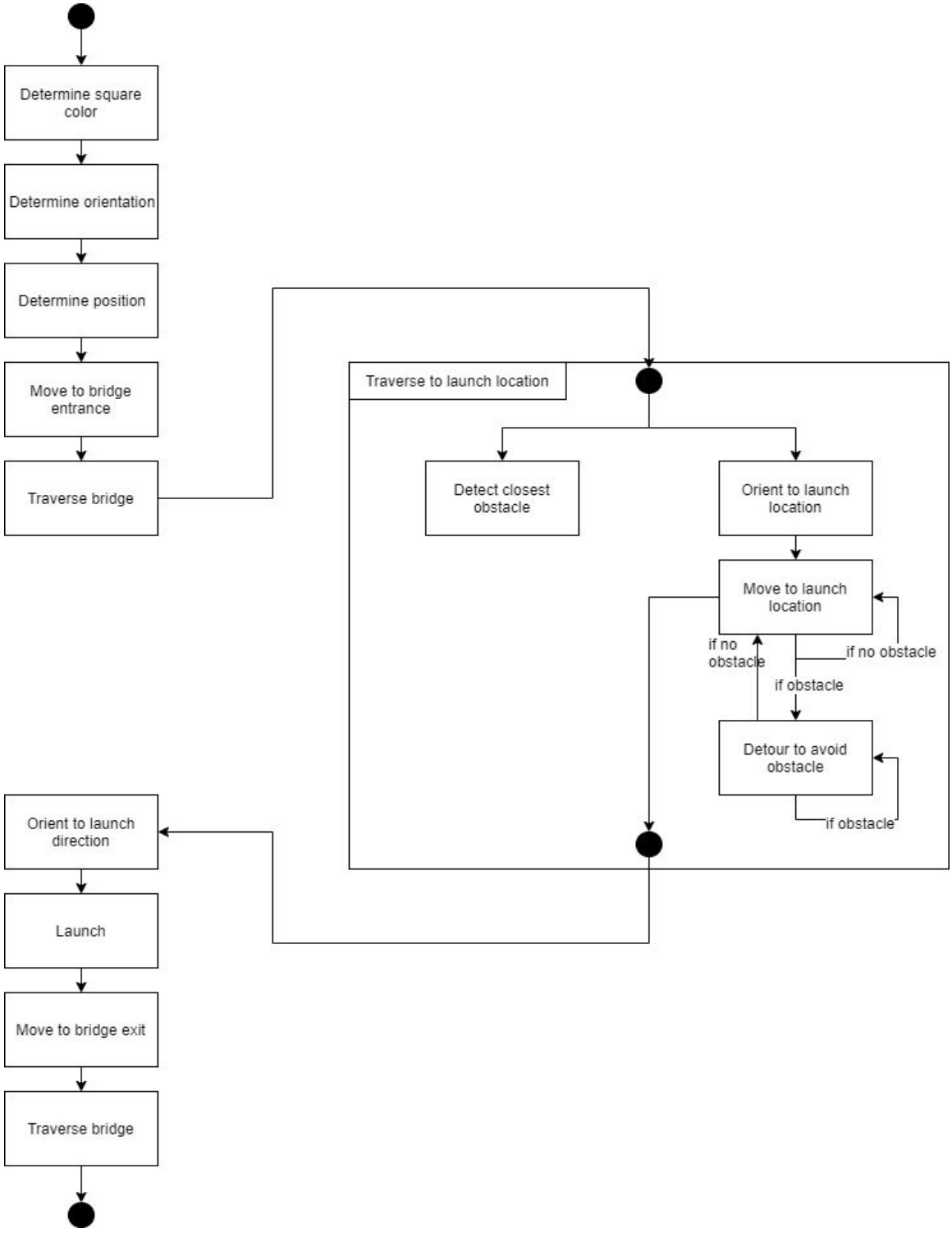

Fig 2.0.1. System Model

3.0 HARDWARE AVAILABLE AND CAPABILITIES

The lego components are the source of several mechanical constraints. Firstly, the lego causes problems related to stability, rooted in the fact that they are somewhat flexible. When designing the robot, not causing too much pressure on any of the pieces needs to be taken into account. Secondly, using lego only allows certain shapes to be created, due to their rectangular nature. Finally, there is the limitation caused by only having a finite number of each piece, meaning that certain designs will not be possible due to a lack of resources.

For electromechanical limitations, one of the most prevalent ones would be the battery, which causes slightly different performance of the robot depending on its voltage. This limits the reliability and the performance of the robot, especially for longer tasks. Another example of this type of limitation can be found in the tachometers of the motors, which are limited in terms of accuracy.

The EV3 processor is a TI AM1808 (ARM926EJ-S core), which has a clock rate of 300 Mhz. This indicates the processing speed of the processor on which the code is run, but it is often required to run multiple threads, which affects its performance. The processor can only run a certain number of threads, before becoming relatively useless. There are also limitations related to the sensors, which can only poll with a limited amount of accuracy and a limited polling rate.

4.0 SOFTWARE AVAILABLE AND CAPABILITIES

Given the current setup of the EV3, the only programming language that we are able to use for the final project is Java. Java is an Object-Oriented programming language, which makes it easier to write the necessary code and it also has automatic garbage collection, which is useful in a system with limited memory. However, compared to a language like C, Java is much more robust and does not utilize the EV3's limited specs as efficiently. Furthermore, all three groups have already written code to execute similar tasks to what will be required in the final group project code, so much of this code may be reusable. In addition, there is the code from both the lecture and tutorial powerpoints, which could prove to be useful in the final project. In both of these previous cases, it must be verified that code does exactly what is wanted for the final project, otherwise changes will need to be made. In terms of tools that could help in the development of the software, there is the Eclipse IDE, which helps us to compile and debug the code. There is also GitHub, which facilitates the sharing of code and the ability for groups to work together by hosting the code in a remote repository.

5.0 COMPATIBILITY

The main limiting factor related to the mechanical design is related to the lego components, but this is already discussed in Section 3.0. As discussed there, the pieces plug together in a rigid and angular manner, which only makes certain designs compatible with lego creation.

The limiting factors for software have also been discussed in Section 4.0, which in brief states that the inherent limitations of Java programming are involved. It also states that the code that have been developed in the lab or provided in class will likely need to be modified since they are not entirely compatible with the goals of the final project. These compatibility issues will be further discussed in Section 6.0 on Reusability.

6.0 REUSABILITY

In terms of the mechanical structure, the labs allowed for a basis that will help in the final project. Firstly, teams had to learn how to properly attach Ultrasonic and Light Sensors for the Wall Following and Odometry labs respectively. It is likely, however, that an entirely new mechanical design will be needed to allow the robot to accomplish all that is needed for the final project.

The software written in the labs will likely be much more useful in terms of reusability than the hardware, since each lab's code relates to important aspects of the final project. The localization lab will be very useful for the robot to locate itself accurately at the start of the task. In addition, the odometry code will be useful to keep track of the robot's general position along its trajectory. Furthermore, the wall following code will be helpful to avoid objects while attempting to complete the task, but we will have to change the algorithm to utilize multiple ultrasonic sensors and follow walls on either side of the robot. The projectile lab will also be useful in the creation of the ball-launching mechanism required. Finally, the navigation lab's code will also be useful for navigating to different required waypoints to complete the final challenge. Overall, all of this code will need to be integrated together, added upon and improved.

7.0 STRUCTURES

Please refer to both the Software and Hardware Documents respectively.

8.0 METHODOLOGIES

Hardware:

- Create an initial robot design, trying to utilize the tools as well as possible.
- Evaluate our design for its capability to meet the requirements adequately and efficiently.
- Try to find possible faults in our initial design for future improvements.
- Design tests to ensure each individual part of the design will be able to accomplish its objective.

Software:

- Evaluate current code resources (ie: lab code from each group), and determine which ones should be chosen to model our final project implementation on. This will be decided by how well the lab code would meet the project requirements and how well it performed in the lab.
- Generate an overall idea of the software system, which tries to find a good balance between having enough complexity to adequately achieve the necessary functionalities, but also takes into account the limited processing power of the EV3.
- Evaluate the software systems ability to meet requirements, and make improvements if necessary.
- Test components of the software to ensure that they work on a unit test level, with integration and system tests coming further down the line.

9.0 TOOLS

For the construction of the mechanical system, the tools available include 3 full Lejos EV3 building kits, whose limitations have already been discussed in Section 3.0. This means that there could be up to 3 EV3 controllers at our disposal for the final project, and their capabilities to run code have been analyzed in Section 4.0. In terms of software tools available, there is the labs and also what has been provided in class and tutorials. A discussion of the usefulness and limitations of these software tools can be found in Section 6.0.

Some external tools that could be useful to expand our knowledge would be the TA assigned to our group, as well as the professors who could answer any questions we may have. There is also documentation about the Lejos EV3 functionalities online, among many other potentially useful resources.

10.0 GLOSSARY OF TERMS

All terms used should be easily understood by anyone with a basis in electrical, computer or software engineering, so no definitions will be needed.