TEAM 11

MCGILL ENGINEERING DESIGN

ECSE 211

ECSE 211 DESIGN PROJECT

SOFTWARE DOCUMENT

Version *1.01*

03/02/2018

ECSE 211 TEAM 11

VERSION HISTORY

Title	Software Document			
Description	Keeps track of all software related implementation			
Created By	Patrick Ghazal, Software Team Leader			
Date Created	2st March 2018			
Version Number	**Modified By**	**Modifications Made**	**Date Modified**	**Status**
1.00	Patrick Ghazal	Created the Document.	1st March	
1.01	Luka Jurisic	Peer reviewed the document. Formatted the Document	3rd March	Preliminary Week 2 submission Content complete

TABLE OF CONTENTS

2 ODOMETER

The Odometer class (and its associated classes OdometerData and OdometerExceptions) has been used since Lab 2. Therefore, the group is quite familiar with its functionality and usage, and we do not plan to bring any changes to its implementation. The only possible change here is an adaptation of code to ensure that communication between methods of different classes remains as smooth as possible.

3 LOCALIZATION

For Lab 5, we still had both an ultrasonic- and light-based Localization class. Given that localization will still need to be performed for the final project, it is likely that both classes will be found in our final code, albeit perhaps modified to better fit the needs of the project.
Regarding the ultrasonic localization, we have found it best to use a rising or falling edge routine depending on the initial distance read of the sensor and not to enforce one over the other.
As to the light localization, the procedure could definitely use some improvement, as it is time-costly now. We will make sure to cut the required time for the procedure as much as we can.

4 NAVIGATION

The travelTo and turnTo methods are the centerpiece of the Navigation class and have proven very useful in past labs. We have had to adapt the behavior of the travelTo method for Lab 5 due to necessary obstacle avoidance and block searching, however these changes will be repackaged and re-implemented differently in the code for the project, due to the change in expected behavior.

5 COLOR DETECTION

In Lab 5, colour identification was done using the mean and standard deviation of the RGB intensities, with mitigated success (the detection was successful but only at small sensor-block distances). For this lab, we plan to experiment with the existing functionalities of the sensor, namely the colorID attribute. If it shows better success in recognizing blocks, we shall implement it.

6 DISPLAY

Display is another class inherited from past labs that has little changed since we first started using it. It is practical to assess whether the odometer is accurate or not. To reduce the number of threads running at any one time in the code, it was discussed that we could perhaps implement it within the Odometer class.

7 ROBOT CLASS

The Robot class is a new addition to our code. Its purpose is to contain and manage all the constants and variables related to the robot: motors, sensors, constants such as the track and wheel radius, etc. It will allow for the centralization of this data, which will allow any class to easily find and manipulate it. Previously, some objects were instantiated and/or passed along several times, causing confusion and possibly interfering with the robot's functionality.

8 SEARCH

The search procedure that was implemented in Lab 5 was to circle around the search area, find blocks using a sensor on the side of the robot, visit a block once detected, and act according to the color of the block. If the block was of the target colour, the search would terminate, and the robot would immediately head to the upper right point of the search area. If the block was not of the target colour, the robot would head back to the path it was following and proceed. Here, the target and procedure being slightly different, it is likely that we will adapt our search procedure to the situation. We may use an algorithm such as the wavefront algorithm if it proves useful and relevant to our situation, where we do not know how many blocks there are or even how many.

9 CAPTURE THE FLAG

CaptureTheFlag is the initial entry point into the code for the entire project. Its main method will initialize the variables of the Robot class as well as the necessary threads, then run the demonstration by coordinating the other classes.