McGill University

ECSE 211: Final design project

System Document

Julien Courbebaisse, Romain Nith, Philippe Papineau

supervised by
David A. Lowther

March 6, 2017

1 Task

Project: ECSE 211 Design Project Task: 1v1 Ball Game

2 EDIT HISTORY

2.1 Document Version Number

- 0.0.1: Version presented to Prof. Lowther on the 22/02/2017
- 0.0.2: Version presented to Prof. Lowther on the 06/03/2017

2.2 Edit History

1. 20/02/2017 (Julien, Romain, Philippe): Create basic document
2. 05/03/2017 (Philippe) Reformated the document in LaTeX

Contents

3 SYSTEM MODEL

Fig. 1: Block Diagram of System Model

4 HARDWARE AVAILABLE AND CAPABILITIES

4.1 Lego components and mechanical capabilities

- 3 bricks with 4 Inputs and 4 Outputs each
- 16 motors: fixed max torque and speed
- 3 light sensors
- 3 ultrasonic sensors
- 3 touch sensors
- various structural Lego parts

4.2 Electromechanical limitations

motors: limited by their max torque and speed sensors are limited by their sampling rate and accuracy

4.3 Electronic/processor constraints

processor speed: 300 MHz 16 GB SD storage up to 3 CPUs, (since we have 3 bricks)

5 SOFTWARE AVAILABLE AND CAPABILITIES

Our code will be written in java and code from previous labs will be reused as much as possible. We will use GitHub for version control and separating tasks into modules. Our group will have 3 main contributors which will allow for rapid development of multiple features simultaneously. We will use eGit, a Eclipse plugin to facilitate this process.

For this project, ease of use and generality of operations will be prioritized over speed of code execution and size of code as our storage space and processor speed are both sufficient to do so. No rapid or complex computations will be needed for the purpose of our project. The development of our code will follow objectives from the Gantt Chart. Our program architecture will follow the issued Class Diagram and Flowchart.

6 COMPATIBILITY

(Adherence to requirements within the design environment, e.g. Lego components plug together in certain ways, so everything has to adhere to this. What about software? Are there any compatibility issues there? Will you connect to third party systems, etc.? In this area, you might want to list pieces of code or mechanical structures that have been developed in the lab these speed up development time but might place constraints on how the system will function, i.e. they were constructed with certain assumptions you need to know what those were so that you can make sure that they dont conflict with the current intended usage.)

On the software side, connecting to the Wifi and the DPM server will be our main task in terms of compatibility (beyond the assumed compatibility of our Java environments and JDK). Our previously developed Navigation, Odometer, Localization, and Sensor classes will be compatible, after a few modifications, and integrated in our final design. As for the hardware, a WiFi module will be connected to the robot as a third party system component.

7 REUSABILITY

7.1 Hardware

The critical piece of hardware (lab 5) is the catapult. We may reuse one of the basic design from the three lab 5 robots we have. If we see that none of them are powerful enough, we will try a completely different firing mechanism (slingshot, crossbow). We will make some tests on different chassis used in the labs. We may reuse one of them.

7.2 Software

Any relevant code from the previous labs will be reuse. More precisely, the basic code for the following tasks will be reuse: Odometry Localization Navigation

8 STRUCTURES

8.1 Attack

- Drivetrain: 2 motors with each a wheel attached to it,
- 1 metal ball to support the back of the robot.
- Localization: - ultrasonic sensor to detect obstacles light sensors to locate path within the grid

Firing mechanism ideas:

- Catapult
- Crossbow
- Pushing mechanism

Each of them will use one or more EV3 motors to communicate power.

Drivetrain: 2 motors with each a wheel attached to it, 1 metal ball to support the back of the robot. Localization: - ultrasonic sensor to detect obstacles light sensors to locate path within the grid Firing mechanism ideas: Catapult Crossbow Pushing mechanism Each of them will use one or more EV3 motors to communicate power.

8.2 Defense

Defense mechanism ideas: Wall blocking target Blowing air to deviate the trajectory of incoming projectiles

9 METHODOLOGIES

(Approaches being taken in all parts of the design and, for software, the basic algorithms to be used. These are really lists of the possible candidate solutions for parts of the problem. They come out of the Ideas Generation phase and will allow a critical analysis before the final design is performed.)

8.0 TOOLS

Refer to Section 2.0 Hardware availabilities and capabilities

10 GLOSSARY OF TERMS

Brick: EV3 Lego Main Module Light Sensor: EV3 Lego Light Sensor Motors: EV3 Lego Motors Ultrasonic Sensor: EV3 Lego Ultrasonic Sensor