McGill University

# ECSE 211: Final design project

System Document

*Julien Courbebaisse, Romain Nith, Philippe Papineau, Nayem Alam*

supervised by

David A. Lowther

## March 16, 2017

# Task

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

# EDIT HISTORY

## 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
    - 0.0.3: Version present to Prof. Lowther on the 16/03/2017

## Edit History

1. 20/02/2017 (Julien, Romain, Philippe): Create basic document
2. 05/03/2017 (Philippe) Reformatted the document in LaTeX
3. 13/03/2017 (Nayem): Changed doc’s font to Times New Roman, size 12. Added Fig 1 and Fig 2 on System Model. Added some Firing Mechanism ideas. Added some text onto section 7.1. Added Red/Blue balls in section 4.1.
4. 15/03/2017 (Romain): Updated the mechanical part of the Methodology section and the Attacking Structures.

# Contents

1. [Task](#_1fob9te) 1
2. [EDIT HISTORY](#_3znysh7) 1
   1. [Document Version Number](#_tyjcwt) 1
   2. [Edit History](#_1t3h5sf) 1
3. [SYSTEM MODEL](#_2s8eyo1) 3
4. [HARDWARE AVAILABLE AND](#_3rdcrjn)

[CAPABILITIES](#_3rdcrjn) 3

* 1. [Lego components and mechanical capabilities](#_lnxbz9) 3
  2. [Electromechanical limitations](#_1ksv4uv) 3
  3. [Electronic/processor constraints](#_2jxsxqh) 3

1. [SOFTWARE AVAILABLE AND](#_3j2qqm3)

[CAPABILITIES](#_3j2qqm3) 3

1. [COMPATIBILITY](#_4i7ojhp) 4
2. [REUSABILITY](#_1ci93xb) 4
   1. [Hardware](#_2bn6wsx) 4
   2. [Software](#_3as4poj) 4
3. [STRUCTURES](#_49x2ik5) 4
   1. [Attack](#_147n2zr) 4
   2. [Defense](#_23ckvvd) 5
4. [METHODOLOGIES](#_32hioqz) 5
5. [GLOSSARY OF TERMS](#_41mghml) 5

# SYSTEM MODEL

Fig. 1: Block Diagram of System Model

Fig 2. Design chosen, long robot with catapult, See Design under “Mechanical Architecture

Fig 3. Preliminary design had hardware issues, redesigning with elastic

# HARDWARE AVAILABLE AND CAPABILITIES

## 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
    - red/blue balls used in lab 5

## Electromechanical limitations

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

## Electronic/processor constraints

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

# 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.

# 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.

# REUSABILITY

## 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.

After design one: catapult didn’t work/ wasn’t favored, so we tried elastics but that didn’t work either. So through trial and error, will most probably modify the robot so it has the catapult + elastic.

## 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

# STRUCTURES

## 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:

* + - Catapult: With the Lab 5, the catapult is the most natural first design that came up. It consist of having the ball resting on a rotating arm. The rotating motion is generated by a motor set on a vertical structure attached to the base of the robot. The initial problem coming from the lab is that the distance is drastically different. Instead of three tiles, the launcher has to throw the ball for a minimum of seven. After a number of trials and errors by setting the axle of rotation higher as well as prolonging the rotating arm, the catapult prototype was able to hit the target eight out of ten successful launch. The drawback of this design is that there are very little control on how far the ball can go. Moreover, the trajectory is prone to imprecision due to the flex of the vertical structure. Reloading the ball to the catapult is also an issue: by lowering the projectile arm, the rotational speed decreases and therefore less power is translated at the ball’s exit.
    - Piston: The design consists of a piston pushing the ball out of a container. The latter has a small opening for the projectile to exit only if it gets pushed by the piston. With this medium, reloading the ball is only a matter of directing it to the firing chamber. The angle imprecision found on the catapult is also minimized. However, after several tests, the piston doesn’t provide enough power to propel the projectile far enough (around 3 tiles).
    - Motorized wheel: The idea of this design is to translate the rotational of spin energy of a pair of wheels into the ball. The latter would be pushed from a chamber to two wheels spinning in the opposite direction of each other before getting sent to the air. The benefits are similar to the piston mechanism but also enabling a better control of the distance. Unfortunately, the ball is too heavy for it to exceed at least two tiles. A second pair of wheels set in series to the other pair would have helped with this issue but the two motors didn’t have enough torque for all four wheels. Even after fiddling with the gear ratios, the prototype ended up not working as expected. Adding a second pair of motor could solve the problem but this entails having a larger and heavier chassis as well as a more power demanding system.
    - Crossbow

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.

## Defense

Defense mechanism ideas:

* Wall blocking target
* Blowing air to deviate the trajectory of incoming projectiles

# METHODOLOGIES

The mechanical design was broken down into three development phases: launcher mechanism, defense mechanism and sensors. First, a general base structure consisting of the EV3 brick, two motorized wheel and a supporting ball was used to accommodate the preliminary designs and tests. The most challenging part of the mechanical is the ball launcher. Several prototypes have been made and tested to decide on the most effective design. As for the defense mechanism prototyping, it started around the middle of the project timeline. Finally, the sensors on the robot is meant to support the autonomy capabilities of the robot. Therefore, the mechanical division has to work closely with the software system to fit their needs. The sensors location and type had to be constantly readjusted throughout the project.

(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.)

Refer to [Section 2.0 Hardware availabilities and capabilities](#17dp8vu)

# GLOSSARY OF TERMS

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