McGill

TEAM 4

ECSE 211

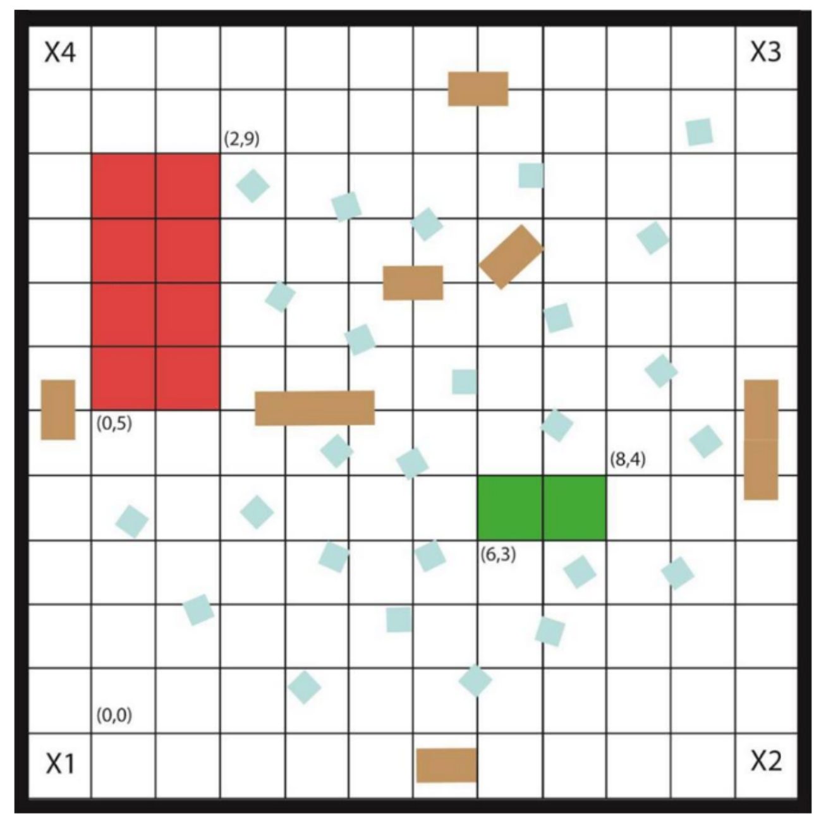
Project Manager - Kareem Halabi - 260616162
Documentation Manager - Jake Zhu - 260621270
Hardware Manager - Benjamin Willms - 260690005
Software Manager - Tristan Toupin - 260688712

Testing Manager - Mamoun Benchekroun - 260565411
General Member (Hardware) - Quentin Norris - 260621167
General Member (Software) - Alexis GJ - 260690465
General Member (Testing) - Richie Piyasirisilp - 260624968

OBJECTIVE

The goal of this project is to design an autonomous robot that is capable of navigating an area, avoiding obstacles, finding blocks, and manipulating them.

SPECIFICATIONS



1*

> The autonomous robot must maneuver an enclosed 12’ x 12’ arena populated with obstacles.

> The robot must be capable of finding and manipulating blue styrofoam blocks.

> Approximately 10-20 styrofoam blocks will be randomly distributed except in the 4 corners.

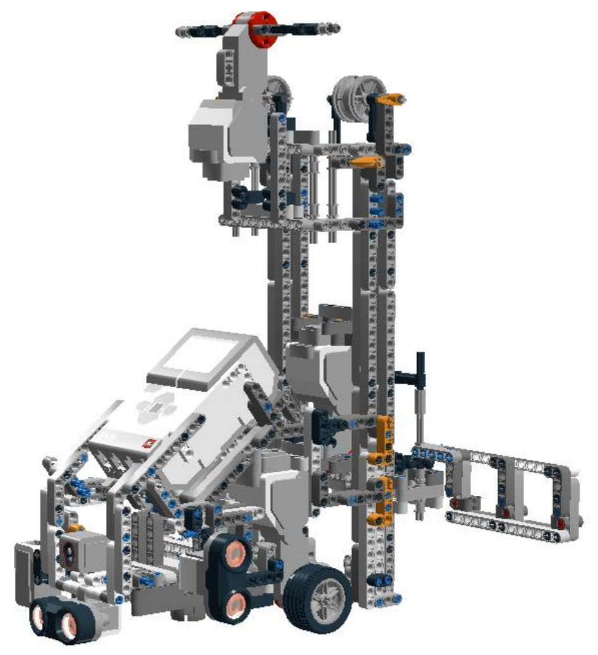
> Red and Green zones will never be smaller than 1x2 squares and never larger than 2x4 squares.

Requirements	Description
Localization	Localize within 30 seconds
Navigation	Nagivate arena while keeping track of location
Avoidance	Avoid obstacles such as the wooden blocks and walls
Object detection	Detect all objects and differentiate them
Object collection	Collect and manipulate the blue blocks
Tower builder	Stack blue blocks within the green zone
Garbage collector	Move blue blocks to designated red zone
Time Limit	5 minutes

THIRD PARTY MATERIALS

Material	Purpose
String	Used to allow the lifting and lowering of the grabbing mechanism
Spoon	Helps the robot cross discontinuities between the boards
Glue	Attaches the spoon to the robot's underside so that the spoon does not slip off when the robot is in motion

HARDWARE



Spoon Based Center Wheel:

> Allows versatility in crossing discontinuities with as little error in odometry as possible.

Single Brick Design:

> Chosen to simplify software, influenced the design of the grabbing mechanism

Compact & Balanced Chassis:

> Ensures that the weight distribution is centered on the wheels and minimized the robot's footprint.

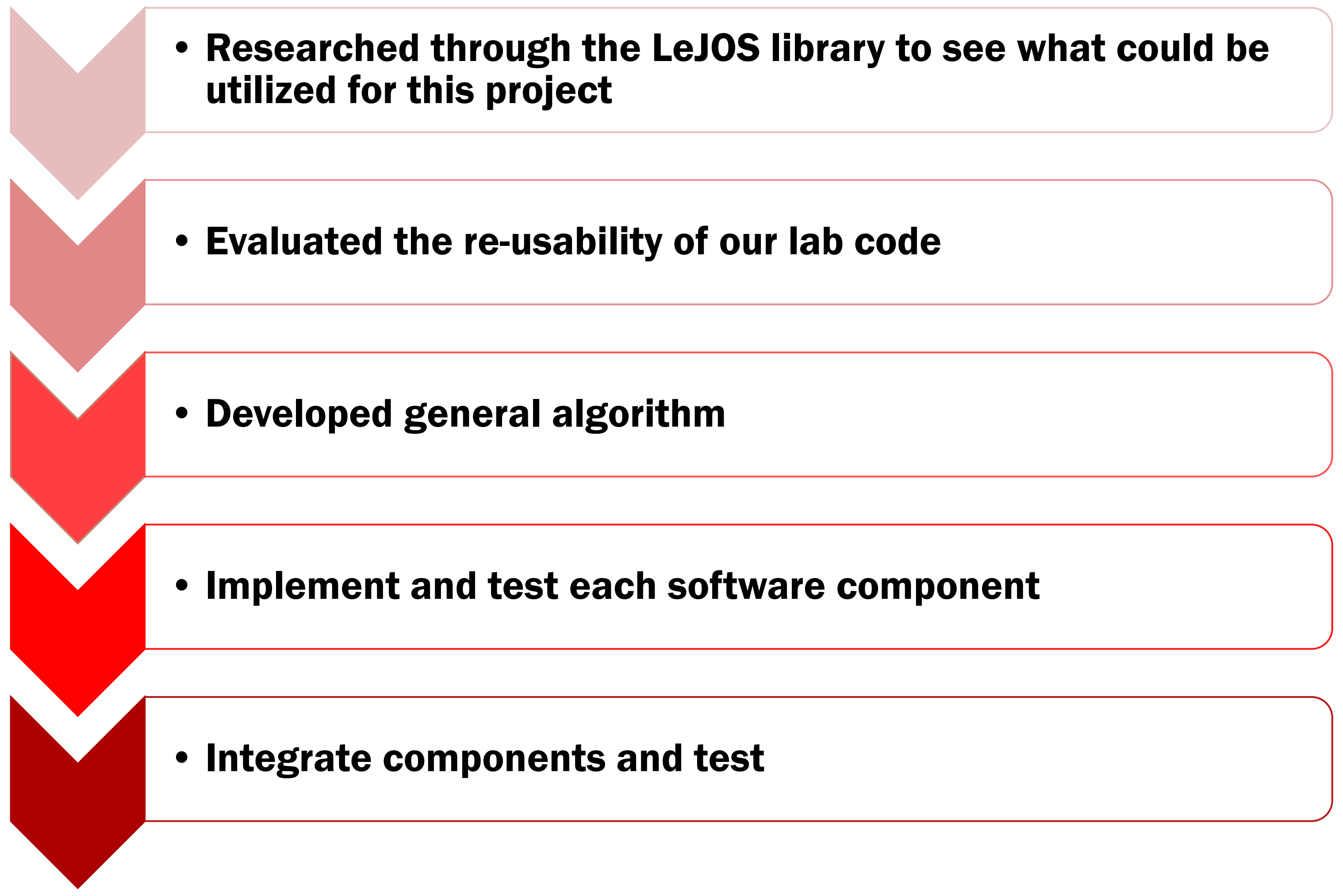
Grabbing Mechanism:

> One-motor grabber with a simple gear system to simplify the grabbing algorithm.
> Retraction of arms to improve balance and mitigate chance of collision and odometer error whilst navigating.
> Back plate utilized as a bumper for localizing and collecting blocks to ensure proper orientation before pickup.

Lifter:

> Constructed to be as light as possible in order to mitigate the robot’s balance issues. The lifter has the height necessary to stack four blocks and the motor is placed at the top to prevent lifter from bending back during heavy lifting.

SOFTWARE DESIGN PROCESS



- Researched through the LeJOS library to see what could be utilized for this project
- Evaluated the re-usability of our lab code
- Developed general algorithm
- Implement and test each software component
- Integrate components and test

TOOLS

draw.io Dropbox eclipse GANTT project
Google Drive git GitHub JavaDoc
LaTeX ObjectAid Overleaf slack

RESULTS

Tests	Test(s) conducted
Localization	> Flat bumper
	> Triangular bumper
	> Rising and Falling edge
Avoidance	> Blocks placed randomly
Object Detection	> Integration test
Object Collection	> Integration test
Tower Builder	> Integration test
Garbage Collector	> Integration test

Tests	Failure points	Possible Improvements
Localization	> Ultrasonic sensor not working properly > Dirty wheels may cause errors when rotating	> More solid wheels to reduce the variance in the wheel radius
Avoidance	> The robot may read the other robot and not detect the blocks in time and collide	> A cone-like structure could be placed around the ultrasonic sensor to avoid false detections
Object Detection	> The robot may read the other robot and not detect the blocks nearest to it	> A cone-like structure could be placed around the ultrasonic sensor to avoid false detections
Object Collection	> The blocks may not be oriented properly even after being flushed against the bumper	> Material with more friction used to lift the blocks
Tower Builder & Garbage Collector	> The blocks being placed may be placed too far off the center and cause the tower to fall	> A scan of the zone to find the exact location of the stack

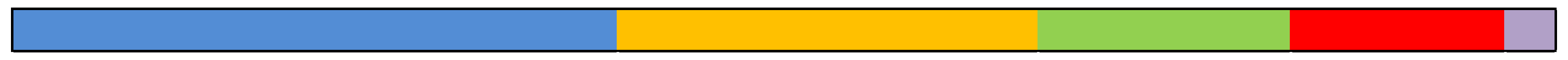
Overall results:

> Bumper localization is fast and simple.
> Optimal search point selection based on available board space and nearby obstacles.
> Custom geometry package handling multiple coordinate systems and synchronization.
> Logging framework to help with testing and debugging.

Limitations and further improvements:

> Obstacle avoidance and red zone avoidance needs to be improved.
> Can’t handle case if obstacles are close to walls.
> Currently avoids nagivating into corners.
> Need smarter searching algorithm.
> Handle more blocks/towers.
> More efficient strategy for garbage collection.

DEDICATED HOURS



	Documentation		Software		Hardware		Testing		Meetings
🟦	170 hours	🟨	117 hours	🟩	70.5 hours	🟥	60 hours	🟪	14 hours

1* The image of the arena is taken from the document “Project Description” available on myCourses.