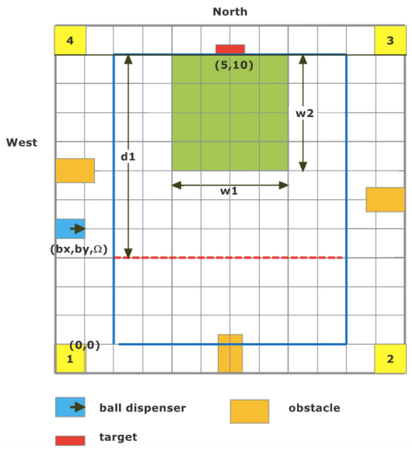
McGill | Faculty of Engineering

LeBot James

Team 11

Project Objective:

"Construct an autonomous robot to play a one-on-one game that is a cross between soccer and basketball. The robot must be able to play either forward or defense and be capable of navigating the field without hitting obstacles..."



Main Tasks:

- Localize within 30 seconds
- Retrieve a ball
- Shoot ball through target
- Block opponent shots (Defense)

Subtasks:

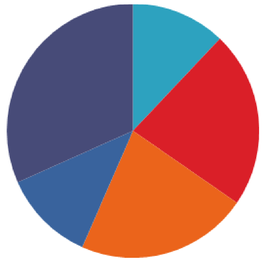
- Odometry + Correction
- Navigation
- Obstacle Avoidance

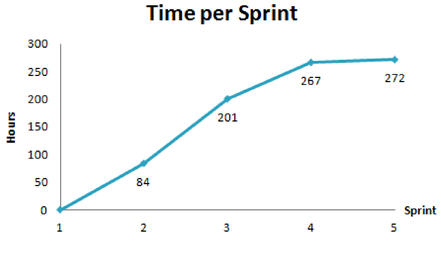
Project Management:

Our Team:

Team Member:	Role:	Total Hours Worked:	% Diffrence From Team Average:
Alex Lam	*Documentation Manager*	54	-0.74%
Durham Abric	*Design Team Manager*	55	1.10%
Ethan Lague	*Hardware Manager*	53	-2.57%
Ian Smith	*Testing Manager*	54	-0.74%
John Wu	*Software Manager*	56	2.94%
	Team Total:	***272 Hours***	
	Team Average:	**54.4 Hours**	

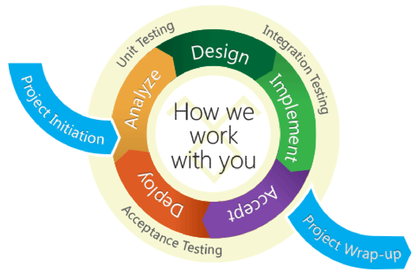
Budget:





Agile Design Process:

Sprint Model:



Agile Manifesto:

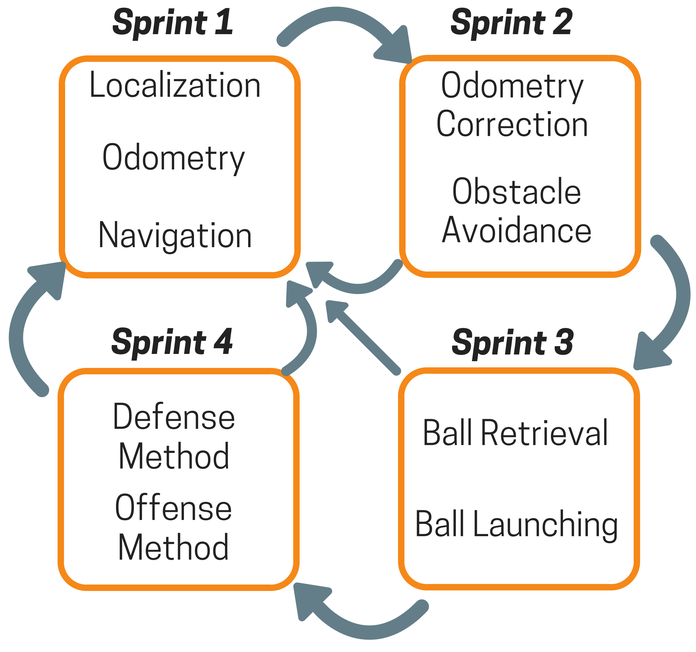
Individuals + Interactions over *Processes + Tools*

Working Product over *Comprehensive Documentation*

Collaboration over *Negotiation*

Responding to Change over *Following a Plan*

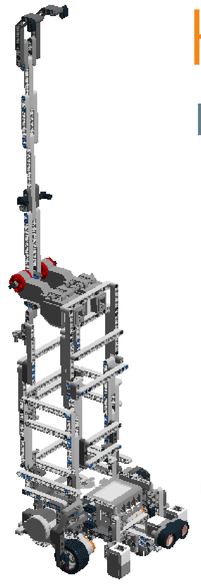
Our Design Process:



Design Tools + Components:

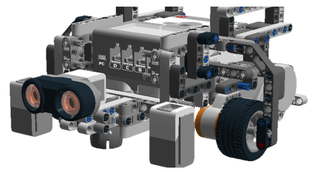
IntelliJ IDEA · Travis CI · GitHub · slack · JUnit 5 · busybot · Overleaf · mindstorms EV3 · Java

Hardware Design:

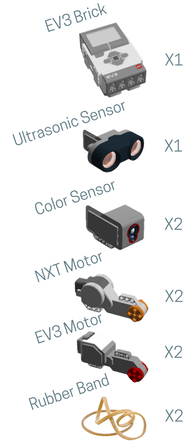


Design Principles:

- Minimal sensors to limit threading & battery usage
- Uncomplicated ball retrieval & launching mechanism
- Focus on navigation and obstacle mapping
- Sturdy/reliable wheelbase
- Large silhouette for defense



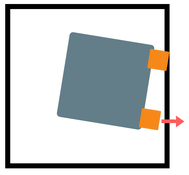
Components:



Software + Algorithms:

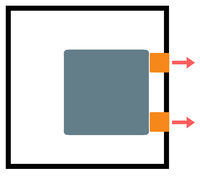
Odometry Correction:

Step 1:



- When first light sensor detects gridline, stop corresponsing motor
- Other wheel continues moving

Step 2:

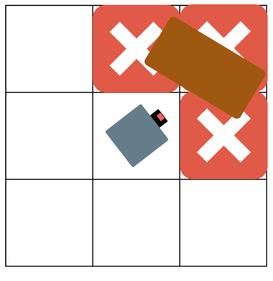


- When second light sensor detects gridline, start other motor again
- Correct heading (0,90,180,270)
- Correct X/Y coordinate

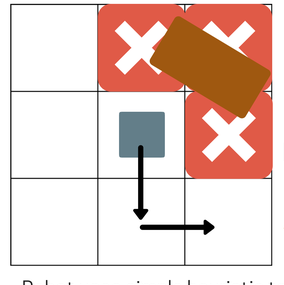
Navigation + Avoidance:

Mapping:

- Ultrasonic sensor detects obstacles in certain squares
- Robot registers those square as *not allowed*
- Only searches squares it hasn't previously



Navigation:



- Robot uses simple heuristic to rank possible moves
- Each move has a priority
- If move square contains obstacle, next priority move is executed
- Greedy, recursive algorithm