CSCI 1010 FALL 2015 FINAL PROJECT: **ROBOT MAZE NAVIGATION**

The Logical Operators

Cal Lohrmann, Samsara Counts, Mike Fan, & Abigail Smallman

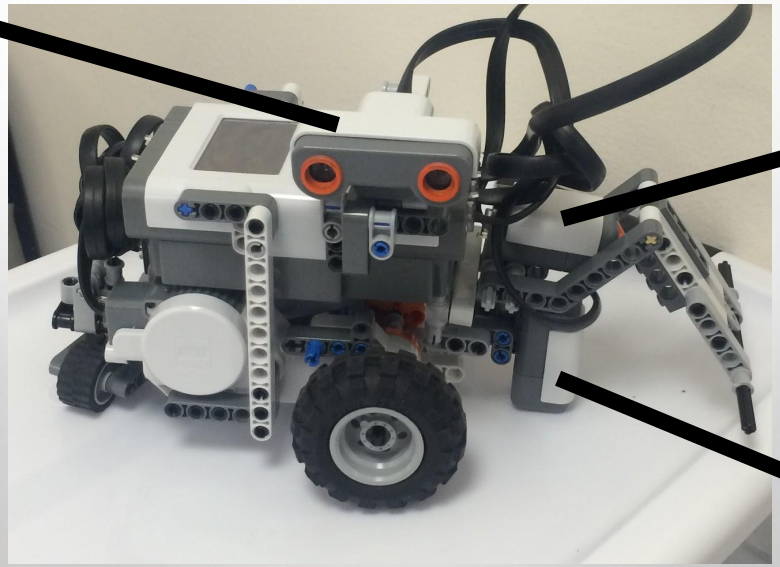
THE GEORGE WASHINGTON UNIVERSITY
SCHOOL OF ENGINEERING AND APPLIED SCIENCE

SOLUTION STRATEGY: SENSORS

- Our design uses three sensors:
  - **Ultrasonic Sensor** - to detect which side the wall is on when our robot senses blue, so it knows to turn the other way (so we can detect a wall without wasting time moving then hitting it)
  - **Color Sensor** - to follow a line and detect the blue tape at intersections
  - **Touch Sensor** - when the robot hits a dead end or wall, the touch sensor senses it and our robot makes a U-Turn

SOLUTION STRATEGY: DESIGN

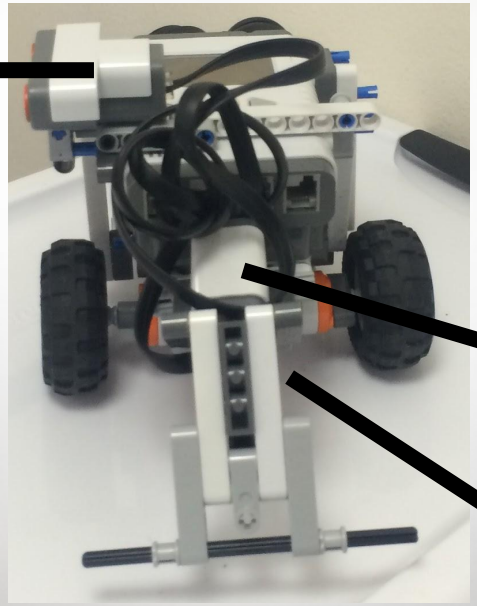
SIDE VIEW



THE GEORGE WASHINGTON UNIVERSITY
SCHOOL OF ENGINEERING AND APPLIED SCIENCE

SOLUTION STRATEGY: DESIGN

FRONT



THE GEORGE WASHINGTON UNIVERSITY
SCHOOL OF ENGINEERING AND APPLIED SCIENCE

SOLUTION STRATEGY: IN DEPTH

- When the CS detects not blue, the right and left wheels alternate in a forward motion.
- When the CS detects blue, the stack is entered which records whether the robot turns left or right.
  - The robot always favors turning right, so the US first checks if the robot can turn right by sensing if there is an obstacle/wall there.
  - If the US detects a wall on the right side of the robot, it will turn left and vice versa. The robot will then find the line and follow it until the next blue tape is detected or it runs into something (detected by the TS).

SOLUTION IMPLEMENTATION

The Stack

- The stack records which way the robot turned at the blue tape.
- If the robot runs into something, it will follow the line back to the tape, turn in the appropriate direction (as indicated below) and the stack gets rid of the incorrect turn, instead adds what it should have done to the top of the stack.
  - Right before -> Turns right again -> add "straight" to top of stack
  - Left before -> Turns left again -> adds "straight" to top of stack
  - Straight before -> Turns right -> adds "left" to top of stack

SOLUTION IMPLEMENTATION: PROBLEM-SOLVING

Problem:

- Our robot couldn't make sharp turns

Solution:

- Make one wheel turn forward, while the other turns backward

Problem:

- HSV Converter did not work

Solution:

- adapt code to fit original RGB values

STACK ON THE WAY BACK

- After the robot hits the aluminum foil, it turns around and follows the black line until it hits blue. Then, it pops the top of the stack, assigning that value to a variable.
- Reads what the variable is, robot does the opposite (unless it's straight):
  - Top of stack is right -> goes left
  - Top of stack is left -> goes right
  - Top of stack is straight -> goes straight

LESSONS LEARNED

- **We learned:**
  - Github and the command line are incredible time-savers
  - Very detailed plans for each lab = more productivity
  - Being considerate of everyone's workload/time in general leads to a more successful group
- **As a team, we overcame:**
  - adjusting to everyone's different styles of working
  - the limitation of only being able to run 32-bit Eclipse on one computer

LESSONS LEARNED, CONT.

- **Important takeaways from working with robots:**
  - **Unit testing** is an effective strategy to save debugging time, especially when working with unpredictable hardware
  - Testing code as soon as possible and tweaking as needed is an effective way to approach coding machines that work in real time
  - When implementing ranges of values in code, real values (from various sensors) directly from sensors are best

THE END

Thank you!

THE GEORGE WASHINGTON UNIVERSITY
SCHOOL OF ENGINEERING AND APPLIED SCIENCE