

# ROBO POLO TEAM

---

**Team members:** Jacob Koontz, Linh Le, Mohammed Mohammed Salih, Ivan Mondragon, Angela Zhang

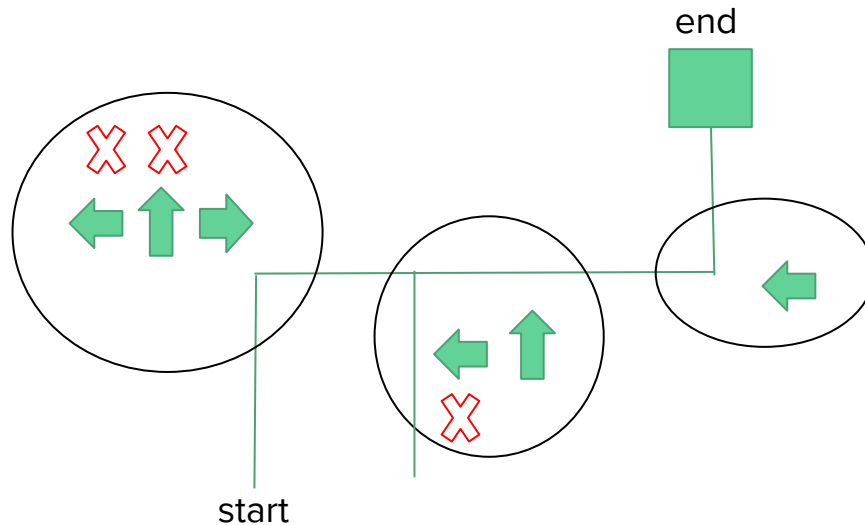
# Useful Resources

- Sample code from Arduino
  - a. Follow line
  - b. Maze solver

<https://www.pololu.com/docs/0J21>

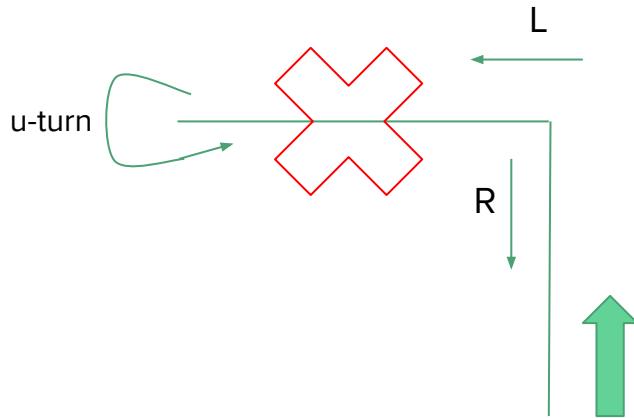
# Algorithm

- Theory in shortening paths guaranteed to find end in non-loop mazes
  - Left Hand on Wall Algorithm
    - Left turn > Straight ahead > Right turn



# Optimizing Algorithm

- Using different observations to create backtrack patterns
  - Simplify turns when backtracking
    - LBR = B
      - Left, u-turn, right = dead end
    - LBS = R
    - RBL = B
    - SBL = R
    - SBS = B
    - LBL = S

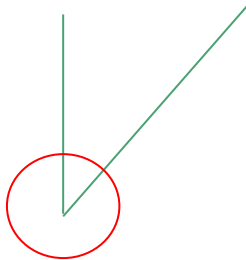
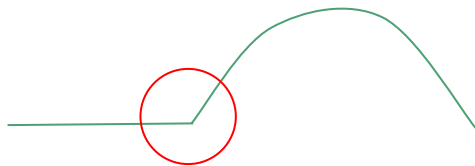


# Optimizing

- Shortened delays
  - Faster reaction to environment
    - Example: turn faster
- Increased speed
  - Faster line segment travel between intersections
    - Around 25% faster
  - Last segment is fast
    - No more turns there

# Obstacles

- Simplifying algorithm
  - Shortening path correctly
- Curves
  - Overturns/underturns
  - Sensor adjustment
- Sharp turns
  - Increase delay time



# Obstacles

- Finishing maze
  - Sensor adjustment (must cover 5 sensors)
- Placement of line segments
  - Some were too close to each other which affected sensor reliability
- Board size
  - Maze was too big, kept falling off board
- Optimizing affects everything
  - Could not solve maze at first
  - Could not detect line properly
    - Overshoot/undershoot
  - Changed speed and delay
    - Tried many methods, wasn't sure what worked

**Thank you! Questions?**