WVPL 2021 Makers Club Amazing Maze Season

Olavi Kamppari

Jan 6, 2022

Season Overview

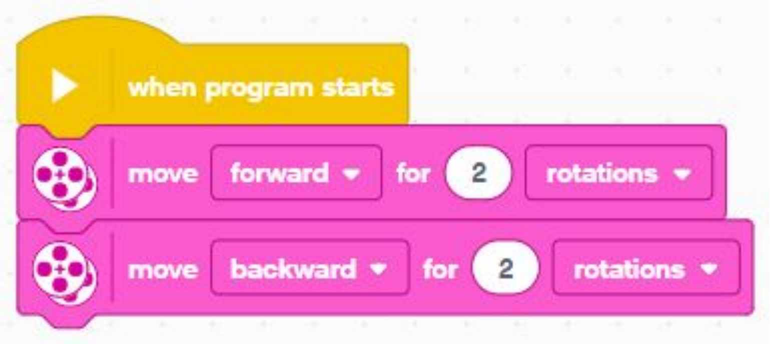
- Club members will learn to
  - program LEGO Mindstorms robots
  - solve challenges
  - compete in the games between the 2 teams.
- The final game is to find a **route through a maze** in shortest time.
- The programming is done with the graphic **Scratch language** (EV3 Classroom).
  - https://education.lego.com/en-us/downloads/mindstorms-ev3/software#downloads
- The programs will
  - read different sensor
  - move the robot
  - manipulate the actuators.
- The club members will
  - build or modify the robots
  - design control strategies to compete in the games.

Season Schedule

1. Jan 6, How to start with EV3 Classroom
2. Jan 13, Basic moves and turns, calibration
3. Jan 20, Actuator, touch sensor, mission design
4. Jan 27, Distance sensor, motor encoder, parallel parking
5. Feb 3, Color sensor, proportional control, gain and bias
6. Feb 10, Color sensor, advanced line follower
7. Feb 17, Improving line follower strategies
8. Feb 24, Wall follower, around the block competition
9. Mar 3, Wall collision handling
10. Mar 10, Maze resolver design
11. Mar 17, Maze resolver tuning
12. Mar 24, Demonstration to parents and guests

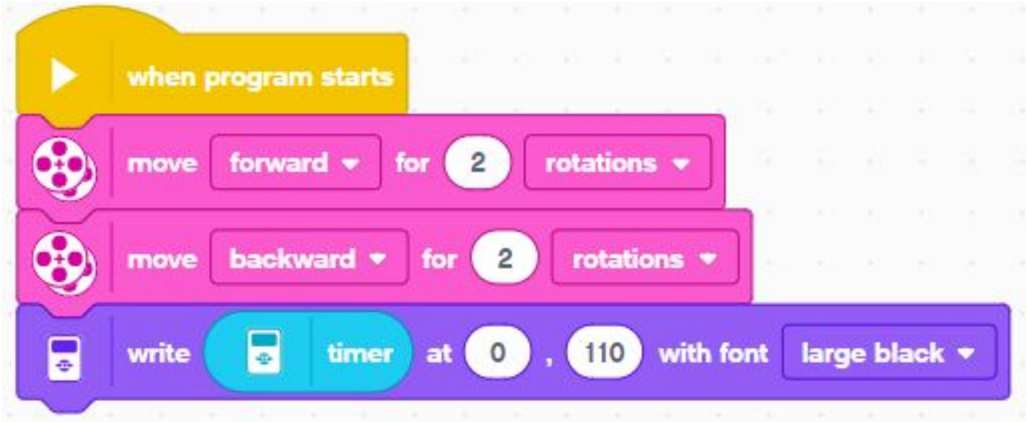
Lesson 1, How to start with EV3 Classroom

1. Review the Training Robot
   a. Driving motors on ports B and C, actuator motor on port D
   b. Touch sensor (port 1), color sensor (port 3), distance sensor (port 4)
2. Add Scratch Blocks
   a. Movement: Move Forward for 2 rotations
   b. Movement: Move Backward for 2 rotations
3. Connect Brick
   a. Turn brick on
   b. In upper left, select the red brick icon
   c. Select the target brick
4. Run the Program
   a. In lower right corner, select the blue start button
   b. After execution, select the red stop button



Lesson 1, Measure execution time

1. Add Block
   a. Display: Write
2. Add Value
   a. Sensors: Timer
3. Change Parameters
   a. X: from 1 to 0
   b. Y: from 1 to 110
   c. Font to large black
4. Run the Program
   a. Observe the value 3.011 on the bottom of the display.
   b. This means that it took 3.011 seconds to go first forward and then back.



Lesson 1, MyBlock

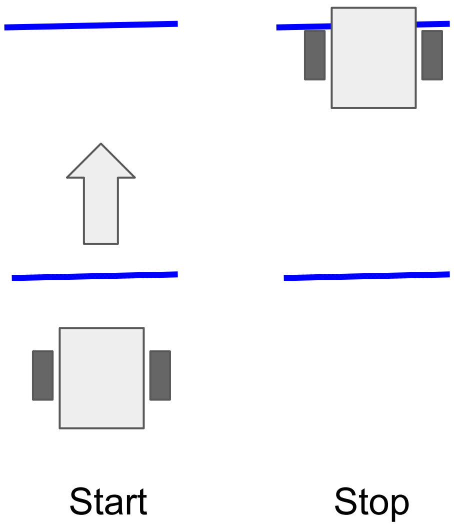
- The native moves are defined in rotations, degrees, and time
  - One wheel rotation is 360 degree
- For planning the robot movements, it is better to use centimeters as units
- When the wheel diameter is 2.76 cm, then the circumference is 17.3 cm
  - In other words one wheel rotation moves the robot by 17.3 cm
  - For 1 cm movement, the wheel has to turn 360/17.3 degree (= 20.81)
- We will create a MyBlock that takes two parameters
  - Distance in cm
  - Speed in percentage (100 is the maximum)
    - Negative speed means backwards movement
- The magic number (20.81) is hidden in this block definition

Lesson 1, MyBlock creation and usage steps

- Select MyBlocks: Make a Block
- Change the block name to Go
- Add two numeric parameters, cm and speed
- Under define Go
  - Add Move Block
  - Add Multiply Operator *
- Modify the main program
  - Add two Go blocks
- Run the program and observe
  - How the time dropped to 2.37
    - The default speed is 50 %
    - The Go blocks did run at 100 %
- The time savings are modest
  - Due to acceleration and turnaround

Lesson 2, Calibration of Go Block

1. Use a tape that is different color than the background
    a. Such as blue or black on beige or light grey
2. Make two parallel tape lines at around 60 - 80 cm distance (L)
    a. Measure the distance in mm (such as L = 713 mm)
3. Start the robot below the first line
4. After the robot stops
    a. Record the degrees for motors B and C (such as 1484)
    b. Repeat several times to get a median value (M)
5. Calculate the value for degree per cm
    a. Deg/cm = M / (L/10)
    b. For example 1484/(713/10) = 1484/71.3 = **20.81**

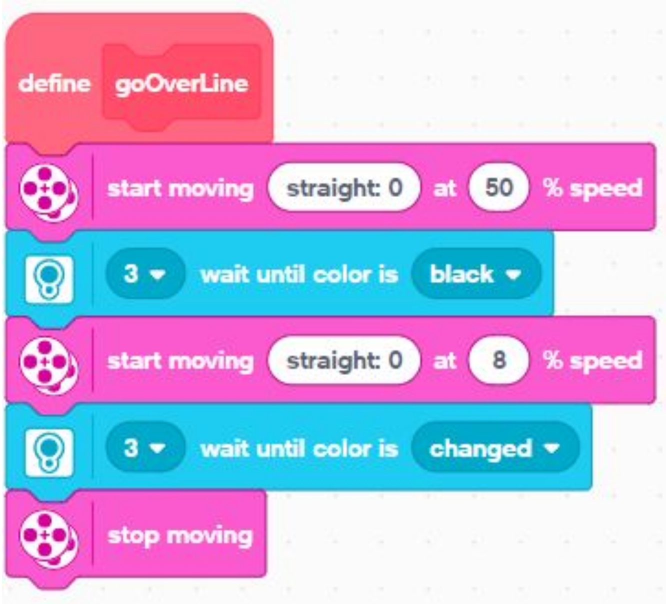


Lesson 2, Distance Calibration

1. Go over 1st line
2. Reset the counters
3. Go over 2nd line

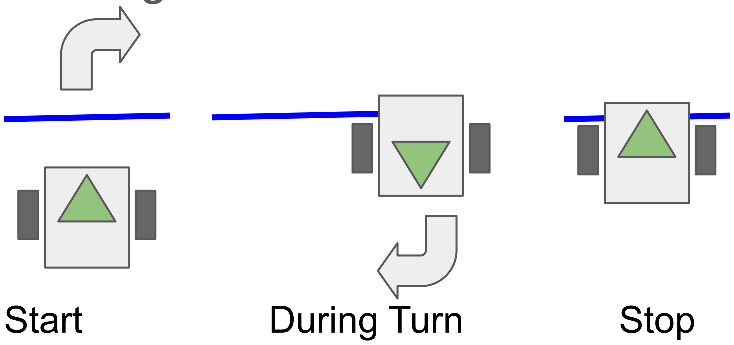
Go Over a Line

1. Move fast
2. Until the line starts
3. Move slowly
4. On the line
5. Stop after the line ends



Lesson 2, Calibration of Turn and Spin Blocks

1. Use a tape that is different color than the background
   a. Such as blue or black on beige or light grey
2. Start the robot below the line
3. After the robot stops after 360 degree turn
   a. Record the degrees for motor B (such as 1875)
   b. Repeat several times to get a median value (M)
4. Calculate the value for travel degree per turn degree
   a. travelDeg/turnDeg = M / 360
   b. For example 1875/360 = **5.208**
5. In theory, 90 degree right turn
   a. Requires that motor B turns
      i. 90 * 5.208 = 469 degrees



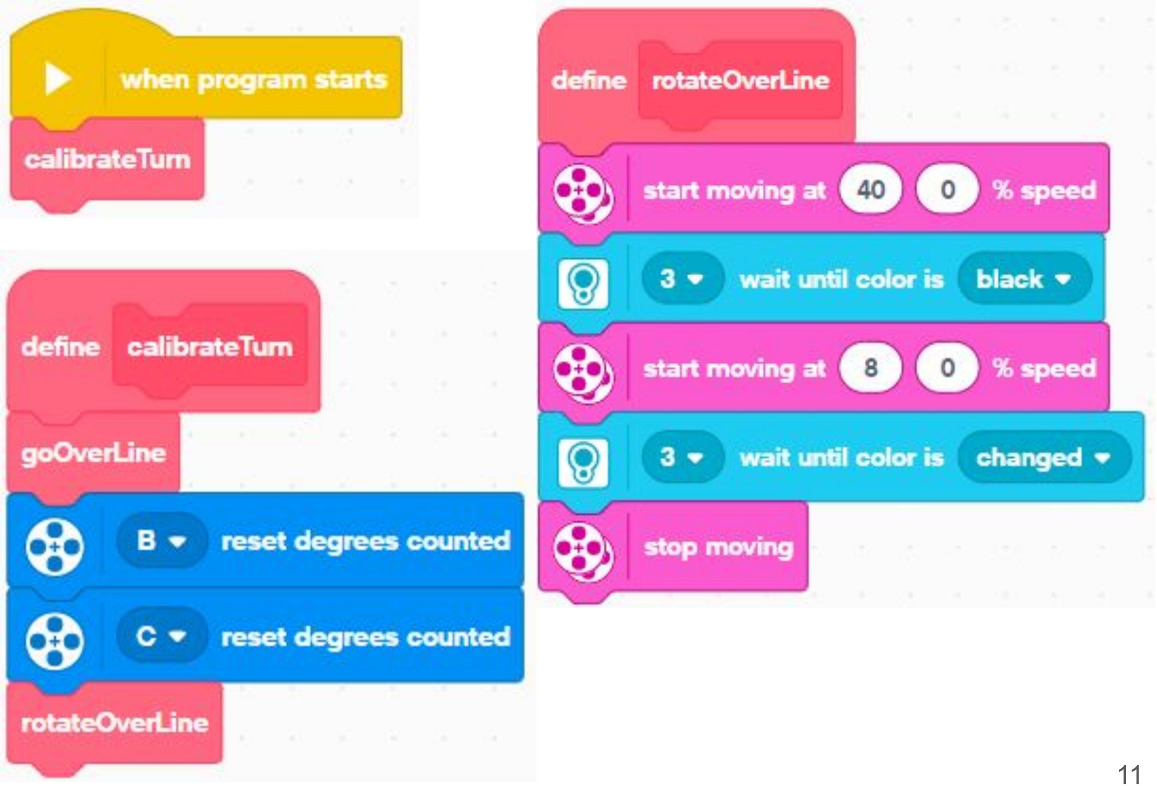
Lesson 2, Turn Calibration

1. Go over the line
2. Reset the counters
3. Rotate over the line

Rotate Over the Line

1. Move 40% with motor B
2. Until the line starts
3. Move 8% with motor B
4. On the line
5. Stop after the line ends

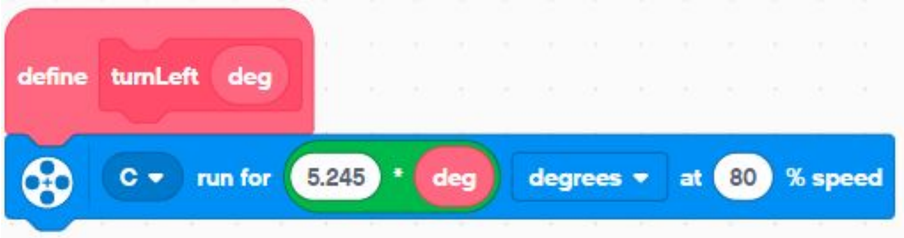
Note, the motor C is running at zero speed during the turn. Still, at the end the C counter is a small negative number caused by the slack in the motors.



Lesson 2, Fine Tuning the Turns and Spins

- The calculated travel degree per turn degree (5.208) is a good starting point for iterative adjustments for the single motor turns
  - turnRight
  - turnLeft
  - backRight
  - backLeft
- The two motor spins need half of those degrees, in this case 2.604.
  - spinRight
  - spinLeft
- After testing the turns and spins with 90, 180, 360, and 720 degrees it can be observed that the numbers have to be adjusted for correct results

Lesson 2, MyBlocks for 1 Motor Turns



Lesson 2, MyBlocks for 2 Motor Spins