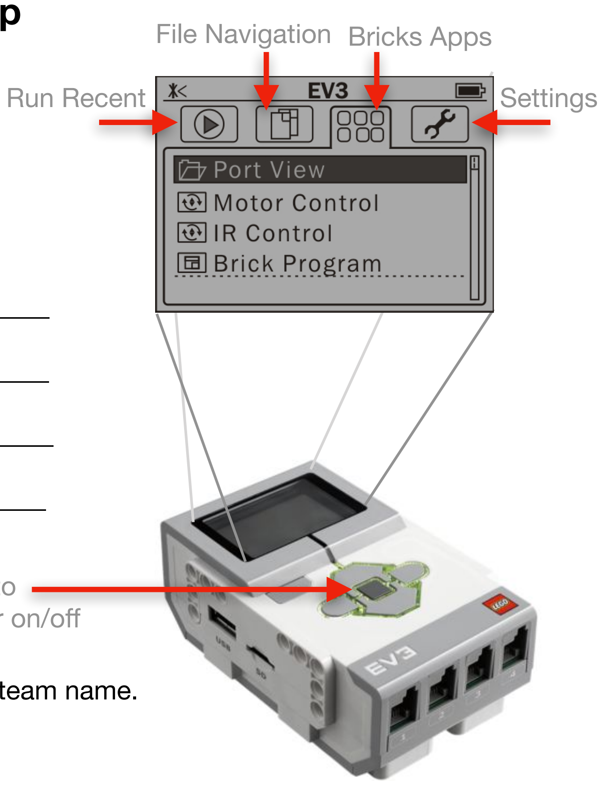
Middle Engineering Academy: EV3 Programming & Robotics

Created in collaboration with:

STUDENT NAME:	TEAM NAME:

Table of Contents

Challenge 1: EV3 Brick Orientation & Setup



In your kit, find your EV3 intelligent brick.
Turn it on by pushing the middle gray button.

There are four tabs at the top. Take some time to explore each of these tabs. What do you think each of the tabs is used for?

Run Recent: ______________________________

File Navigation: ______________________________

Brick Apps: ______________________________

Settings: ______________________________

CHANGE THE NAME OF YOUR BRICK:
- Navigate to the *Settings* tab.
- Navigate to *Brick Name* (bottom).
- Use the onscreen keyboard to rename the brick with your team name.

CLEAN OFF YOUR BRICK:
Let’s clean up your brick by deleting old programs.
- Navigate to the *File Navigation* tab.
- Click the center button to select an old project.
- Click the center button again. The EV3 Brick will ask you if you want to delete.
- Click right and then click the center button to delete the project.
- Click right and then click the check mark to confirm.
- Repeat these steps until all old projects have been deleted from your brick.

Observation: Take note of where the battery pack is on your robot, and the port used for charging it. Locate the charger in your kit. What do you think are some best practices for charging your robot?

____

____

____

____

____

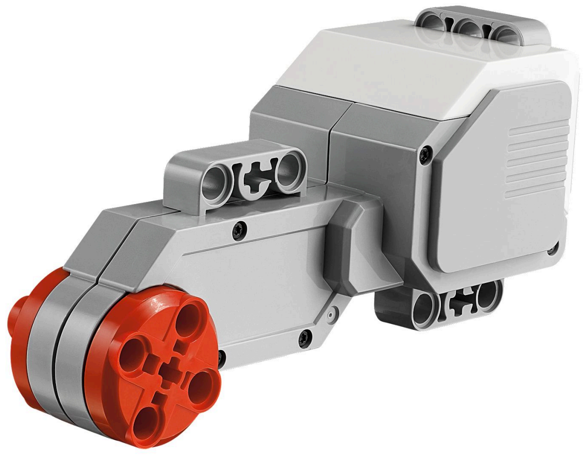
____

____

____

Challenge 2: Introduction to Motors

READ ABOUT MOTORS

Large Servo Motor	Medium Servo Motor
	
The *Large Motor* is a powerful "smart" motor. It has a built-in *Rotation Sensor* with 1-degree resolution for precise control. The *Large Motor* is optimized to be the driving base on your robots.	The *Medium Motor* also includes a built-in *Rotation Sensor* (with 1-degree resolution), but is smaller and lighter than the *Large Motor*. That means it is able to respond more quickly than the *Large Motor.*
The *Large Motor* runs at 160-170 rpm, with a running torque of 20 Ncm (slower but stronger).	The *Medium Motor* runs at 240-250 rpm, with a running torque of 8 Ncm (faster but less powerful).

ATTACH & TEST MOTORS

- In your kit, find a medium motor and connect it to *Port A*.
- Find a large motor and connect it to *Port B*.
- Use the EV3 Brick buttons to navigate to the *Brick Apps* tab.
- Select *Motor Control* and use the app to make both your motors turn.

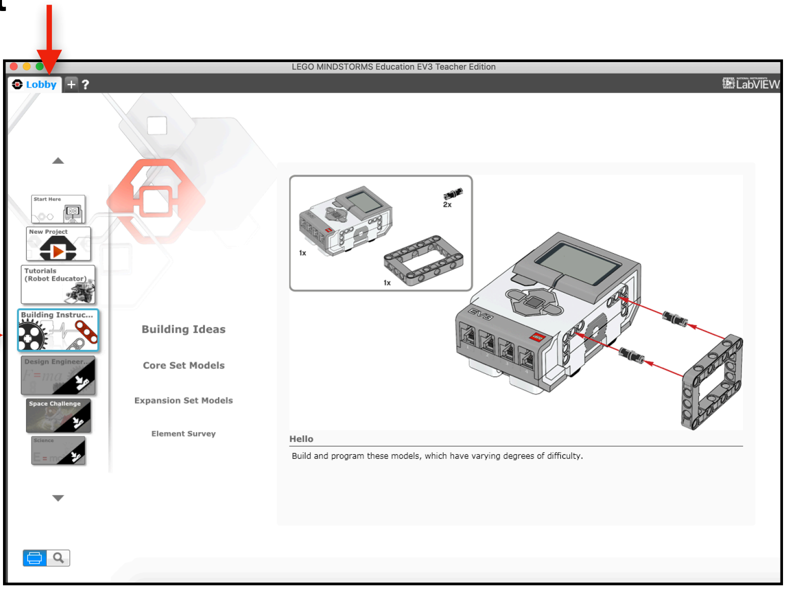
Conclusion: What do you think the main differences are between these motors? What would you use the large motor for? And the medium motor?

Challenge 3: Build a Robot

It’s time to build a robot!

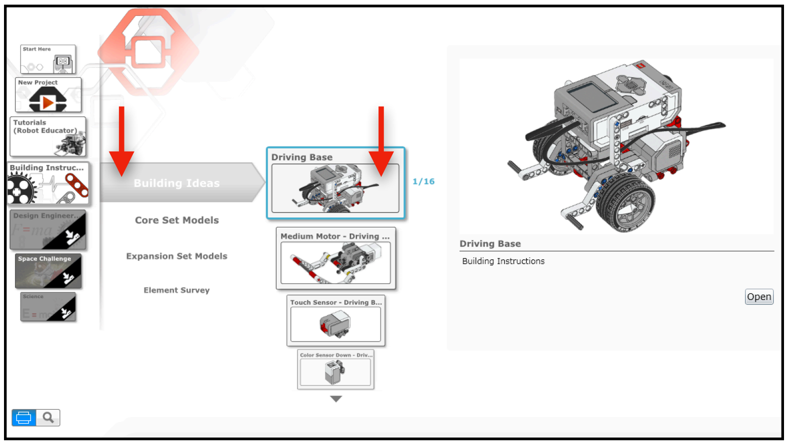
- Open up LEGO Mindstorms application on your computer.
- Go to the LEGO Mindstorms Lobby.
- Click on “Build Instructions.”
- Select “Building Ideas.”
- Click on “Driving Base.”

Follow the step-by-step instructions to build the robot



Hello

Build and program these models, which have varying degrees of difficulty.



Driving Base

Building Instructions

Reflection: Which what challenges did you face while building your robot? How did you overcome those challenges?

____

____

____

____

____

____

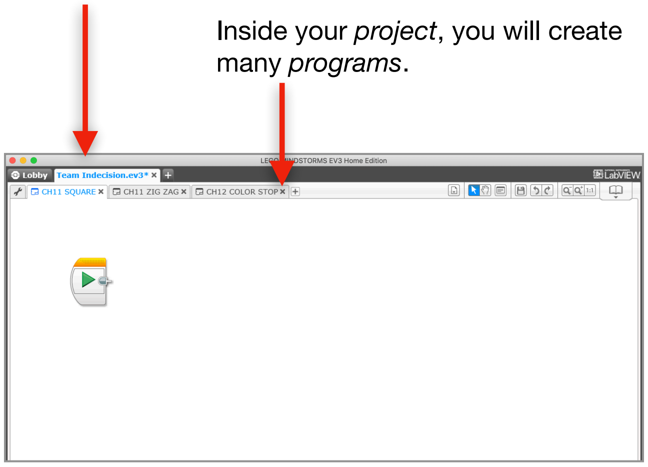
Challenge 4: Orientation to LEGO MINDSTORMS

CREATE PROJECT

- Log in to your assigned computer.
- Open the LEGO MINDSTORMS program.
- Click on File > New Project > Program.
- Click on File > Save Project As...
- Name your project as your team name.
- Save your Project file on the desktop for easy access later.
- Click Save.

NOTE: a *project* is almost like a folder—it can store many different *programs* inside it. Your team should only have ONE *project*, named with your team name.

Inside your *project*, you will create many *programs*.



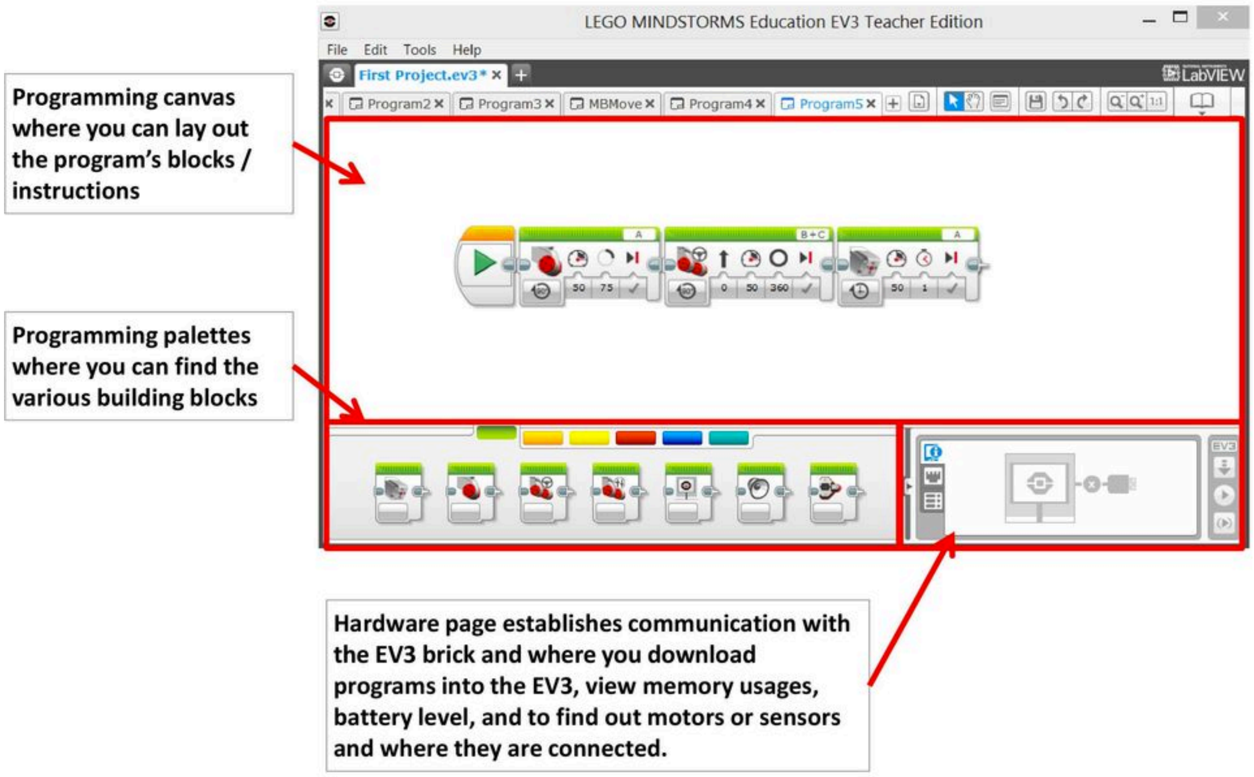
HOW TO CREATE PROGRAMS

- To create new programs, click the + button.
- To rename programs, double click the text.

EXPLORE LEGO MINDSTORMS WORKSPACE

There are three different areas in the workspace: Programming Canvas, Programming Palettes, and Hardware Page.

NOTE: SAVE YOUR PROGRAMS OFTEN!



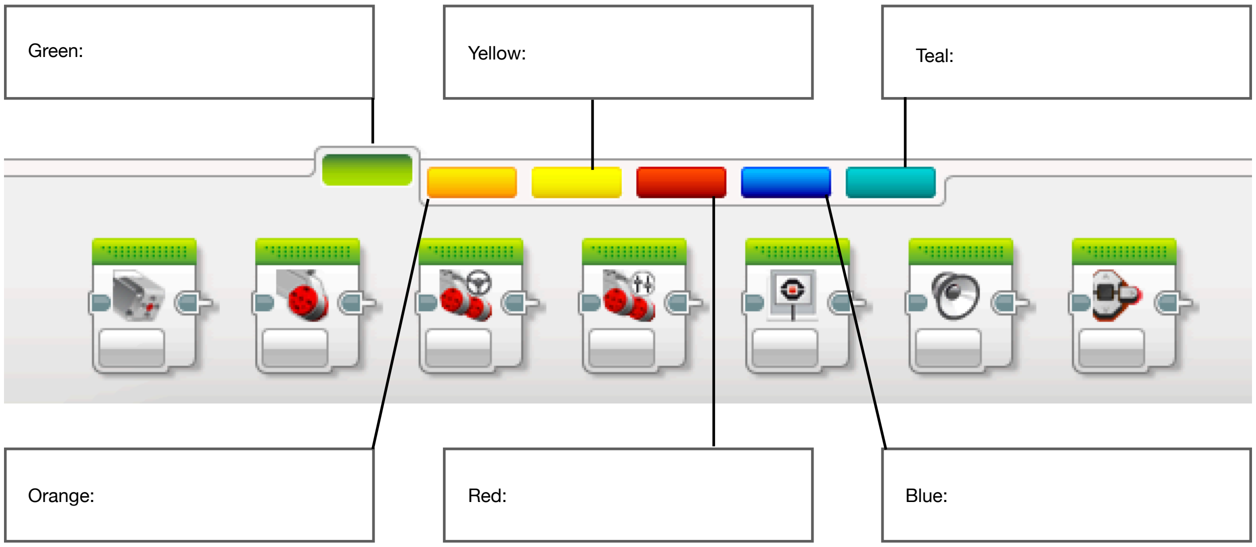
Programming canvas where you can lay out the program's blocks / instructions

Programming palettes where you can find the various building blocks

Hardware page establishes communication with the EV3 brick and where you download programs into the EV3, view memory usages, battery level, and to find out motors or sensors and where they are connected.

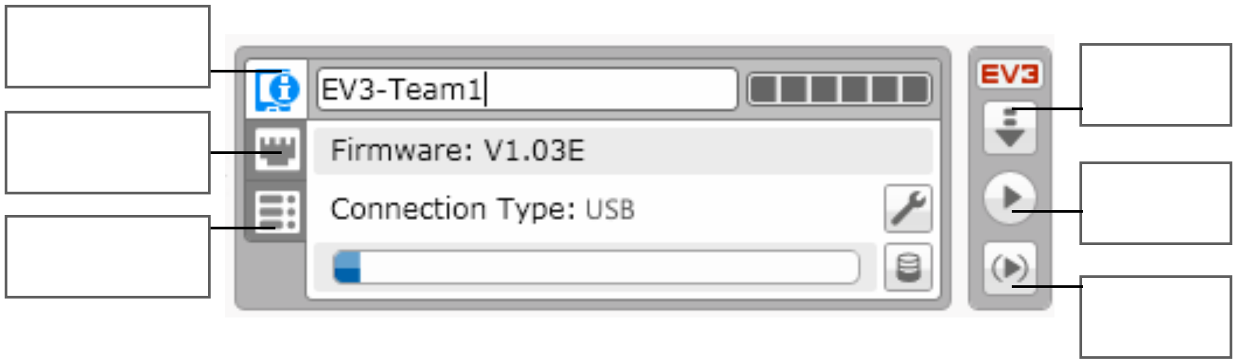
EXPLORE PROGRAMMING PALETTE

- Along the bottom of LEGO MINDSTORMS, you will find 6 different tabs. These are called your *Programming Palettes.*
- Hover your mouse over each *Programming Palette* to learn the name of it.
- Label each palette below:



HARDWARE PAGE

- Connect your robot to your computer using the USB download cable.
- Now that your computer is connected to your EV3 brick, does the *Hardware Page* (bottom right corner) look different?
- Hover your mouse over each *Hardware Page* menu item and label it on the diagram below.



Find *Port View* on the *Hardware Page* and answer the questions below:

Turn your left wheel by hand. What happens to the *Port B* value? ______________________________

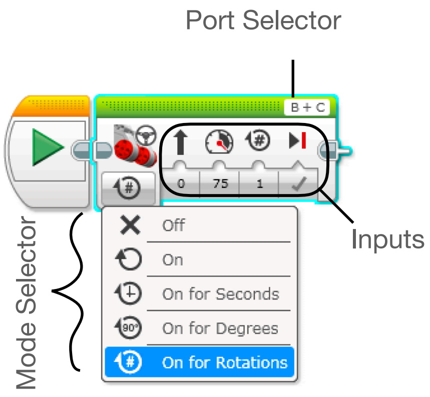
____

Disconnect the cable from *Port B*. What happens to the *Port B* value? __________________________

____

Reconnect your cable. Does the motor reappear? ____

Challenge 5: Make Your Robot Move



STEERING BLOCK ORIENTATION

Drag a green *Move Steering* block to the canvas. Click on different settings of the block. Take note of the 5 modes that can be used to program a *Move Steering* block.

MOVE DEGREES

- Build a program that uses a *Move Steering* block to make the robot move forward for 1000 degrees at a power level of 40.
- Save your program as CH5 DEGREES.
- Download your program and keep your robot connected to the computer.
- Choose a smooth surface (IE: Desk) and a rough surface (IE: carpeted floor) to conduct your test
- Run the program 2 times on each surface.
- Use *Port View* in the *Hardware Page* to answer the following questions:

Trial	**Surface**	**How many degrees did the left motor *actually* turn?**	**How many degrees did the right motor *actually* turn?**
1	Smooth		
2	Smooth		
3	Rough		
4	Rough		

Conclusions: In the space below, use complete sentences to draw your conclusions based on the tests you just conducted. How precise and consistent are the motors? How do different surfaces impact the motors?

POWER LEVELS

- Build each of the following programs and save them:
    - Robot moves 3 seconds at power level 40 (CH5-1 PWR 40)
    - Robot moves 3 seconds at power level 80 (CH5-2 PWR 80)
    - Robot moves 3 seconds at power level -40 (CH5-3 PWR -40)
    - Robot moves 3 seconds at power level -80 (CH5-4 PWR -80)
- Download and run each program.
- Use *Port View* in the *Hardware Page* to answer the following questions:

Trial	**Program**	**Direction Motors Turn**	**How many degrees did the right motor turn?**
1	CH5-1 PWR 40		
2	CH5-2 PWR 80		
3	CH5-3 PWR -40		
4	CH5-4 PWR -80		

Conclusions: In the space below, use complete sentences to draw your conclusions based on the tests you just conducted. What is the relationship between power level, time, and distance? Can you guess why you cannot program the *Move Blocks* to travel a certain distance? What would be the downfall in relying on degrees, rotations, or time to program a set distance with your robot?

Competition 1: Sprints

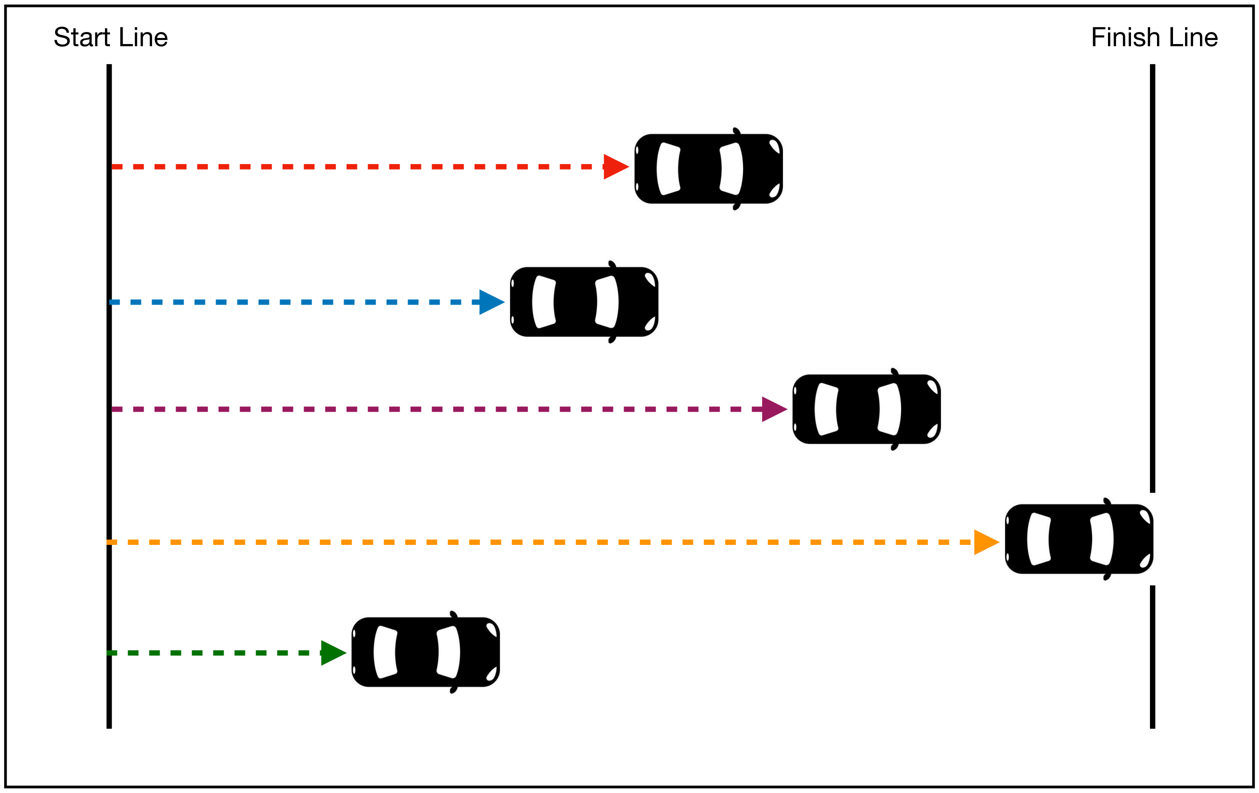
OBJECTIVE
Your mission is to program your robot to make a mad dash from the start line to the finish line. The robot with the fastest time wins!

Save your program as COMP1 SPRINT.

RULES
- The official timer(s) must record all official sprint times.
- Teams may practice their sprint times in advance, but may only record their official time once.
- All robots must begin fully behind the start line, with no component extending over the line.
- A robot is considered "finished" when any robot component breaks the barrier of the finish line.
- Heats of robots racing will be based on the number of available official timers.
- Rematches due to circumstances beyond robot or human player control are determined at the discretion of the official timers.

HINTS AND TIPS
Some variables you may want to consider in order to impact your robot's sprint time:
- Programming motor power
- Wheel size used
- Robot weight
- Drag, or other factors impacting your robot's ability to drive straight
- Gearing (direct drive vs. geared drive)
- Battery power (fully charge your robot!)

Use the graph paper on the following page to take notes, document changes, or keep track of your team's progress.

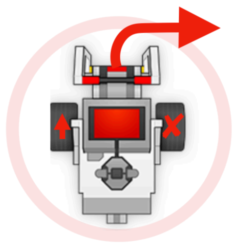


NOTES & OBSERVATIONS

Challenge 6: Make Your Robot Turn

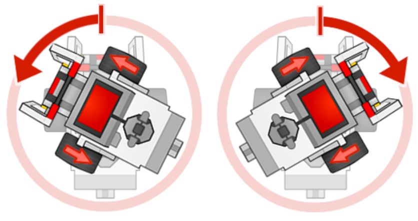
HOW DO ROBOTS TURN?

There are three different ways to make your robot turn.



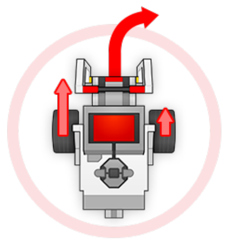
PIVOT TURN

For a pivot turn, one wheel stays completely still while the other wheel moves forward or backward. The robot "pivots" on one wheel.



POINT TURN

For a point turn, both wheels move in opposite directions at the same speed. This results in a very tight turn on the point.



SWING TURN

For a swing turn, both wheels move forward, but one is moving faster than the other. This makes the robot curve.

PROGRAM TURNING

Use a *Move Tank* block to build a program that accomplishes the following tasks. Name your program CH6 TURNING.

- Pivot turn 90 degrees to the right.
- Pause 3 seconds.
- Point turn 90 degrees to the left.
- Pause 3 seconds.
- Swing turn a full circle, approximately 2 feet in diameter.
- Stop.

Note: try experimenting around with *on for degrees*, *on for rotations*, and *on for seconds.*

Conclusion: In the space below, use complete sentences to draw your conclusions. In which situations would you use each type of turn for your robot? Do you prefer one over the other? Which one seems more precise—rotations, degrees, or seconds?

____

____

____

____

____

____

____

OTHER PROGRAMS (OPTIONAL)
Are you ready for an extra challenge? See if you can make your robot perform these additional turning tasks:

Program Name	**Program Desciption**	**Staff Sign**
CH6-1 CIRCLE	Have a team member sit on the floor. Your robot should begin in one place and drive an entire circle around the team member, returning to the same place as it began.	
CH6-2 SQUARE	Have a team member sit on the floor. Your robot should drive a square around the team member. You may choose a point or pivot turn at the corners.	
CH6-3 SQ LOOP	Try the same program as above, but for an added challenge, use a *Loop Block*!	

Reflection: In the space below, use complete sentences to reflect on the other programs you tried. Which programs did you try? How did they work? What difficulties did you encounter? How did you overcome those difficulties?

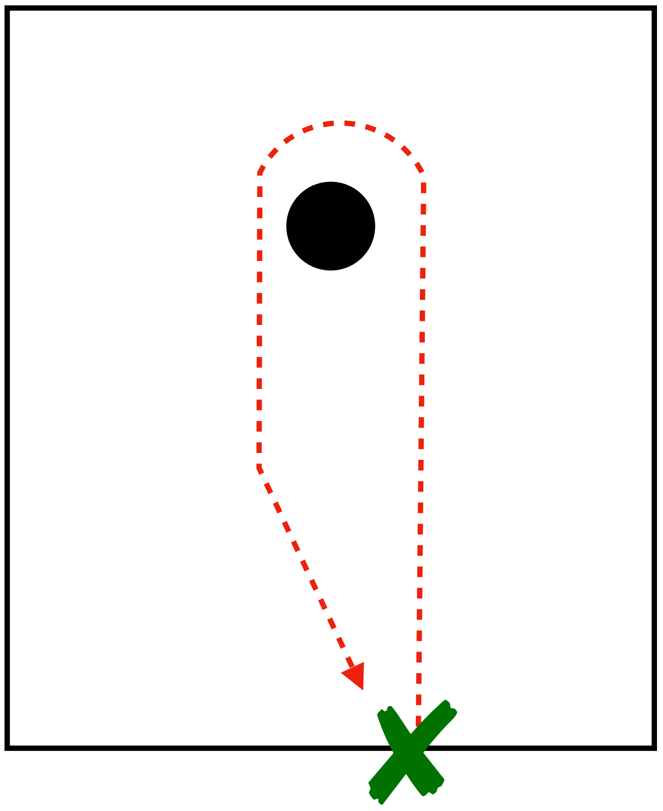
Competition 2: Go Fetch!

OBJECTIVE
You mission is to send your robot out to retrieve a ball from the field and return with it to the starting position as quickly as possible.

Save this program as COMP2 FETCH.

RULES
- Your robot must begin on the green X and return to the green X with the ball in its possession.
- You will need to add some kind of attachment or arm to contain the ball.
- Your robot may take any route necessary to retrieve the ball, as long as your robot returns to the same starting position.
- The timer begins as soon as the human player hits “run” on the program for the first attempt to retrieve the ball, and ends when the ball is successfully brought to the green X. If ball retrieval requires multiple attempts, the timer continues to run between attempts.
- You may rescue and re-start your robot as needed, but do not touch the ball.
- You may request that the referee reset the ball as needed.
- The team that retrieves the ball the quickest wins!

Use the graph paper on the following page to take notes, document changes, or keep track of your team’s progress.



Challenge 7: Touch Sensors

Touch Sensor (Port 1)

ATTACH A TOUCH SENSOR

- In your kit, find a *Touch Sensor*.
- Attach the touch sensor to the front of your robot.
  Make sure that it is in front of everything else on your robot.
- Connect it to *Port 1*.

WHAT DOES A TOUCH SENSOR DO?

- Use the EV3 Brick buttons to navigate to the *Brick Apps* tab.
- Select *Port View* and view *Port 1* to answer the following questions.

What state is the *Touch Sensor* when it is NOT touched? ____

What happens when you press the *Touch Sensor*?____

Conclusion: In the space below, use complete sentences to draw your conclusions. What is the purpose of a *Touch Sensor*? What do you think some practical uses might be for a *Touch Sensor*?

____

____

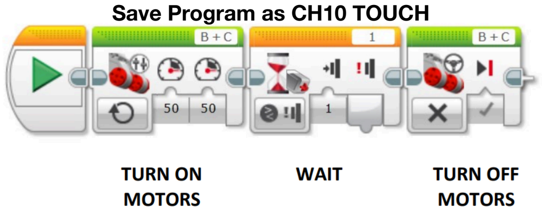
____

____

PROGRAMMING WITH A TOUCH SENSOR

- Create a new program.
- Enter the following code into a new program.
- Save program as CH7-1 TOUCH.
- Download the program to you *EV3 Brick*.
- Place the robot on the floor about 2 feet from a wall.
- Run the program.

Save Program as CH10 TOUCH



Observation: In the space below, use complete sentences to make observations. What happened when the robot hit the wall? Did it stop? If it didn't stop, why was that? How did you fix the programming or robot design so that the robot stops when it touches a wall?

____

____

____

____

OTHER PROGRAMS (OPTIONAL)
Are you ready for an extra challenge? See if you can make your robot perform these tasks using a touch sensor. Make sure you create a new program every time.

Program Name	Program Description
CH7-2 TOUCH SOUND	Robot plays a sound when the *Touch Sensor* is pressed.
CH7-3 TOUCH IMAGE	Robot displays an image on the screen when *Touch Sensor* is pressed.
CH7-4 TOUCH TURN	Robot drives straight until hits something, then it stops, reverses, turns 90 degrees.
CH7-5 TOUCH TURN LOOP	Robot performs the same tasks as above, but keeps repeating these actions forever until the program is canceled.
CH7-6 TOUCH GO	Robot sits completely still until the *Touch Sensor* is pressed. Then it begins speeding forward while making some kind of wild sound.

Reflection: In the space below, use complete sentences to reflect on other programs you tried. Which programs did you try? How did they work? What difficulties did you encounter? How did you overcome those difficulties?

Challenge 8: Ultrasonic Sensors

Ultrasonic Sensor (Port 2)

ATTACH AN ULTRASONIC SENSOR

- In your kit, find an *Ultrasonic Sensor* (the one that looks like Wall-E).
- Attach it to your robot facing forward. Make sure no parts of the robot obstruct the sensor.
- Connect it to *Port 2*.

WHAT DOES AN ULTRASONIC SENSOR DO?

- Use the EV3 Brick buttons to navigate to the *Brick Apps* tab.
- Select *Port View* and view *Port 2* to answer the following questions.

What happens when you point the *Ultrasonic Sensor* at an object and move it closer or further?

____

____

What is the maximum distance that can be measured by the sensor? State your units! ________________

What is the minimum distance that can be measured by the sensor? State your units! ________________

What distance does it measure when you point the sensor at an object 300cm away?________________

What distance does it measure when you point the sensor at an object within 1cm?_________________

PROGRAMMING WITH AN ULTRASONIC SENSOR

It's time to see if you can program an *Ultrasonic Sensor*! Build a program that performs these tasks:

1. Drives forward until it detects an object within 10cm.
2. Stops.
3. Does a point turn 90 degrees to the right.
4. Continues this action infinitely (this will require a *Loop Block).*

Name your program, CH8-1 AVOID WALLS.

Conclusion: In the space below, use complete sentences to draw your conclusions. What is the purpose of an *Ultrasonic Sensor*? How accurate is it? What do you think some practical uses might be for an *Ultrasonic Sensor*?

____

____

____

____

____

____

____

OTHER PROGRAMS (OPTIONAL)
Are you ready for an extra challenge? See if you can make your robot perform these tasks using an *Ultrasonic Sensor*. Make sure you create a new program every time.

Program Name	Program Description
CH8-2 BACK AND FORTH	Add a *Touch Sensor* pointing backwards—the opposite direction of your *Ultrasonic Sensor*. Robot drives forward until it detects an obstacle within 10cm, then reverses until it touches an obstacle. It continues this motion forever. Test this program by placing the robot between two obstacles—it should just keep bouncing back and forth between them.
CH8-3 SILLY	Program your robot to do something silly when it detects an object within 10cm. Perhaps it will display an image or make a sound or start spinning circles. Your decision!

Reflection: In the space below, use complete sentences to reflect. Which programs did you try? How did they work? What difficulties did you encounter? How did you overcome those difficulties?

Challenge 9: Gyro Sensor

Gyro Sensor (Port 3)

ATTACH A GYRO SENSOR

- In your kit, find a *Gyro Sensor.*
- Attach it to your robot so that it is parallel with the floor, pointing forward.
- Connect it to *Port 3*.

WHAT DOES A GYRO SENSOR DO?

- Use the EV3 Brick buttons to navigate to the *Brick Apps* tab.
- Select *Port View* and view *Port 3* to answer the following questions.

What happens when you turn the robot clockwise? ____

What happens when you turn the robot counter-clockwise? ____________________________________

Conclusion: In the space below, use complete sentences to draw your conclusions. What is the purpose of a *Gyro Sensor*? How do the numeric readings work? What do you think some practical uses might be for a *Gyro Sensor*?

____

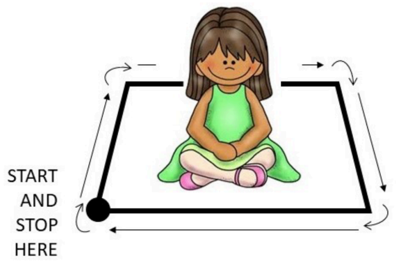
____

____

____

PROGRAMMING WITH A GYRO SENSOR

- Have one person on your team sit on the floor.
- Program your robot to travel around the person, in a square pattern, using the *Gyro Sensor* to measure and make precise turns at the corners.
- For an extra challenge, see if you can use a *Loop Block* in your program!
- If you're stuck, use this video for help: https://www.youtube.com/watch?v=HBh2vDtWcI4
- Save your program as CH9-1 GYRO SQUARE.



Observation: In the space below, use complete sentences to make observations. How did your program work? Did it work perfectly the first time, or require some troubleshooting?

____

____

____

____

OTHER PROGRAMS (OPTIONAL)
Are you ready for an extra challenge? Ready to combine some sensors? See if you can make your robot perform these tasks using a *Gyro Sensor*. Make sure you create a new program every time.

Program Name	**Program Description**
CH9-2 AVOID WALLS	Robot drives forward until it detects an object within 10cm, then stops, turns 90 degrees to the right, and continues to drive forward. This program should continue forever, until program is stopped.
CH9-3 ZIG ZAG	Robot drives a zig zag line.

Reflection: In the space below, use complete sentences to reflect. Which programs did you try? How did they work? What difficulties did you encounter? How did you overcome those difficulties?

Challenge 10: Color Sensors

Color Sensor (Port 4)

ATTACH A COLOR SENSOR

- In your kit, find a *Color Sensor*.
- Attach the *Color Sensor* to your robot, facing down, within 1 cm of the ground.
- Connect it to *Port 4*.

WHAT DOES A COLOR SENSOR DO?

- Use the EV3 Brick buttons to navigate to the *Brick Apps* tab.
- Select *Port View* and view *Port 4* to answer the following questions.

Set your *Color Sensor* to COL-REFLECT mode. What reading is recorded when the sensor is placed close to each of the following surfaces?

Surface	Reading	Surface	Reading
A white surface		A black or dark surface	
A reflective surface		A carpeted surface or clothing	

Set your *Color Sensor* to COL-AMBIENT mode. What reading is recorded under the following conditions?

Surface	Reading	Surface	Reading
Pointed at bright lights		Sitting on the ground	

Set your *Color Sensor* to COL-COLOR mode. What reading is given by the sensor if you place it close to objects of the following colors?

Surface	Reading	Surface	Reading
White		Black	
Red		Blue	
Green		Yellow	

PROGRAMMING WITH A COLOR SENSOR

Write a program to make the robot go forward until it detects a color of your choice.

Save your program as CH10-1 COLOR STOP.

Conclusion: In the space below, use complete sentences to draw your conclusions. Did you use the COL-COLOR or the COL-REFLECT mode? Why? What are some ways you might use the *Color Sensor* on your robot?

____

____

____

____

OTHER PROGRAMS (OPTIONAL)
Are you ready for an extra challenge? See if you can make your robot perform these tasks using a *Color Sensor*. Make sure you create a new program every time.

Program Name	**Program Description**
CH10-2 STOPLIGHT	Robot waits until it sees a green object to go. When it sees a red object, it stops.
CH10-3 SPEAK COLOR	Robot says the name of a color when it detects it (red, green, blue), and is silent when there is no color detected. Hint: This should use a *Switch Block* inside a *Loop Block*.
CH10-4 COLOR DRIVE	Robot say "red" when it drives over a red object, "green" when it drives over a green object, and "blue" when it drives over a blue object. The rest of the time, it should be silent. Make it stop after 30 seconds. Hint: This should use a *Switch Block* inside a *Loop Block*.

Reflection: In the space below, use complete sentences to reflect. Which programs did you try? How did they work? What difficulties did you encounter? How did you overcome those difficulties?

____

____

____

____

____

____

____

____

____

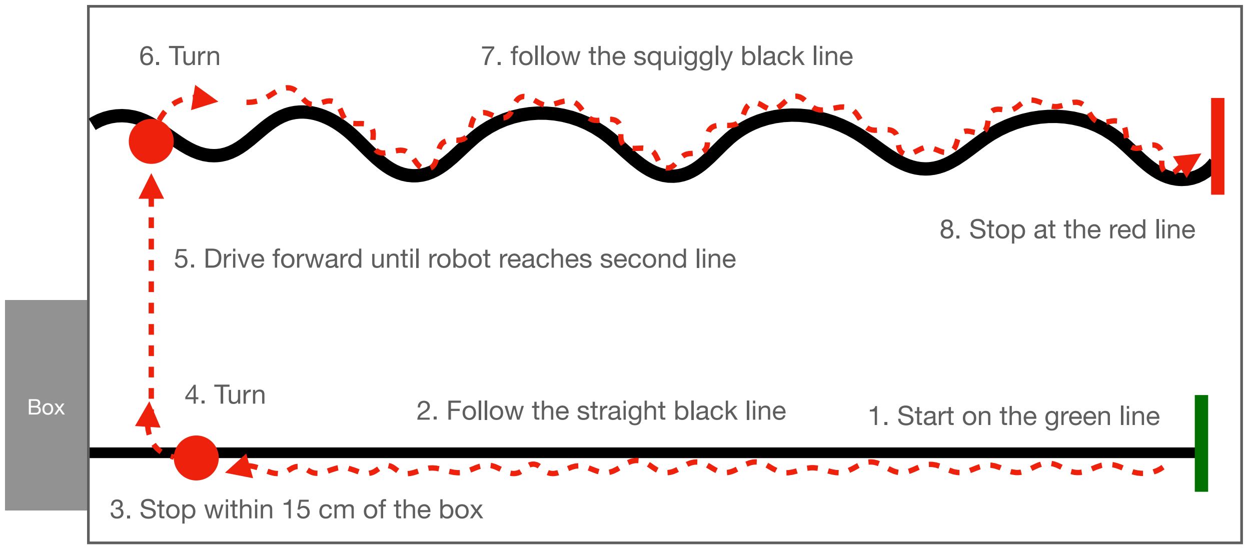
Competition 3: Line Follow

YOUR OBJECTIVE
Your objective is to write a program that gets your robot through the line follow route with the fastest time possible.

Save this program as COMP3 LINE FOLLOW.

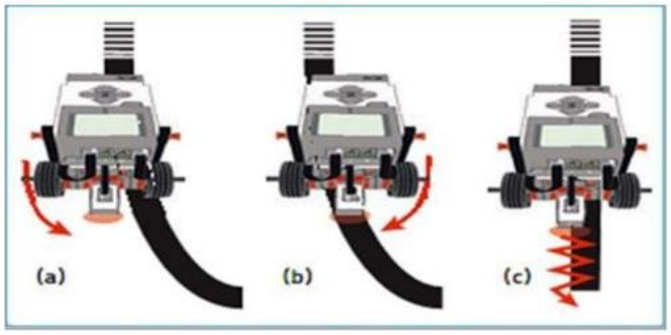
RULES

- Robots must begin on the green line, follow all 8 steps in the diagram, and stop on the red line.
- The timer begins the second the participants hits “go” on the program, and ends the second the robot stops on the red line.
- Participants may try the line follow route multiple times.


6. Turn
7. follow the squiggly black line
8. Stop at the red line
5. Drive forward until robot reaches second line
Box
4. Turn
2. Follow the straight black line
1. Start on the green line
3. Stop within 15 cm of the box

HOW DOES LINE FOLLOWING WORK?
In order to make a line following program work, your robot needs to zig zag back and forth.

(a) When color sensor detects white, the robot must curve towards the black line.
(b) When color sensor detects black, the robot should swivel back towards the white.
(c) As the robot repeats the motion, it moves forward along the black line.

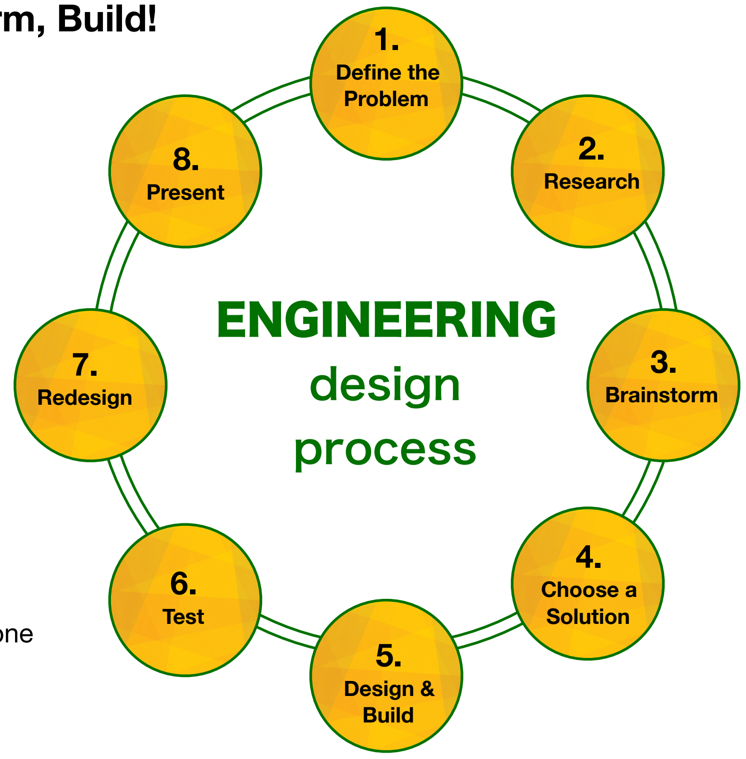


NEED HELP PROGRAMMING?

- Simple line following program: https://www.youtube.com/watch?v=E-_Tm6OVdNI
- More advanced line following program: https://www.youtube.com/watch?v=ye3MhVA9Rhs

NOTES & OBSERVATIONS

Challenge 11: Research, Brainstorm, Build!



THE ENGINEERING DESIGN PROCESS

Engineers use a design process when creating solutions to problems. We are going to use the same process throughout this camp, as we brainstorm, design, and test our robots and programming.

DEFINE THE PROBLEM

Your robot design must include:

- 1 EV3 Brick
- 2 large motors as drive motors
    - Left attached to *Port B*
    - Right attached to *Port C*

Your robot must be capable of:

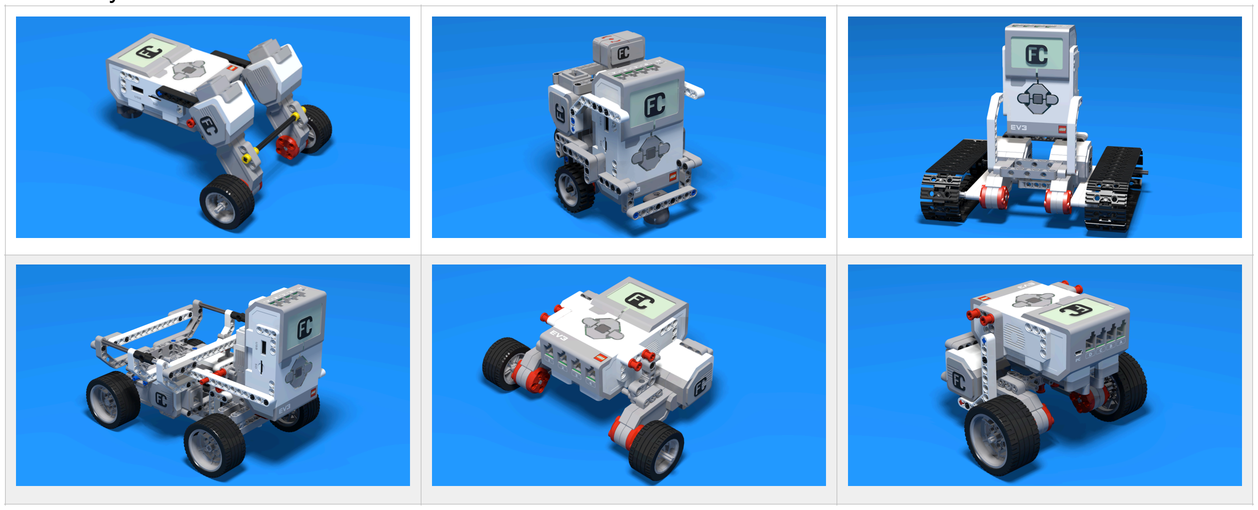
- Moving forwards and backwards
- Moving left and right wheels independently of one another
- Being charged without taking it apart

Other Details:

- Leave all other letter and number ports accessible so you can add sensors/motors as needed
- Do not add any sensors yet (don't worry, you will get to add them in future challenges)

RESEARCH

During this step, you have the opportunity to be inspired by other robot designs! Take a moment to check out some of these robot design ideas. Notice what works well, and what you'd like to do differently.



BRAINSTORM
Use the space below to sketch out ideas for your own robot design.

CHOOSE A SOLUTION
Once you are done brainstorming ideas, it is time to choose a design to build. Consider these questions:

- Does this design meet all the criteria outlined in step 1?
- Does this design look robust?
- Do we have enough time and resources to build this design?

Remember: You are engineers! Engineers work in teams and often have to make compromises.

NOW BUILD YOUR ROBOT!

Reflection: Which robot design did you choose to build? Why did you choose that design? how did you and your teammates come to an agreement?

Competition 4: Weight Pull

YOUR OBJECTIVE

Your mission is to redesign and program your robot as needed to make it successfully pull the greatest amount of weight possible. The robot that is able to pull the most weight wins!

Robots should be programmed to drive forward in a straight line infinitely. Save your program as COMP4 WEIGHT PULL.

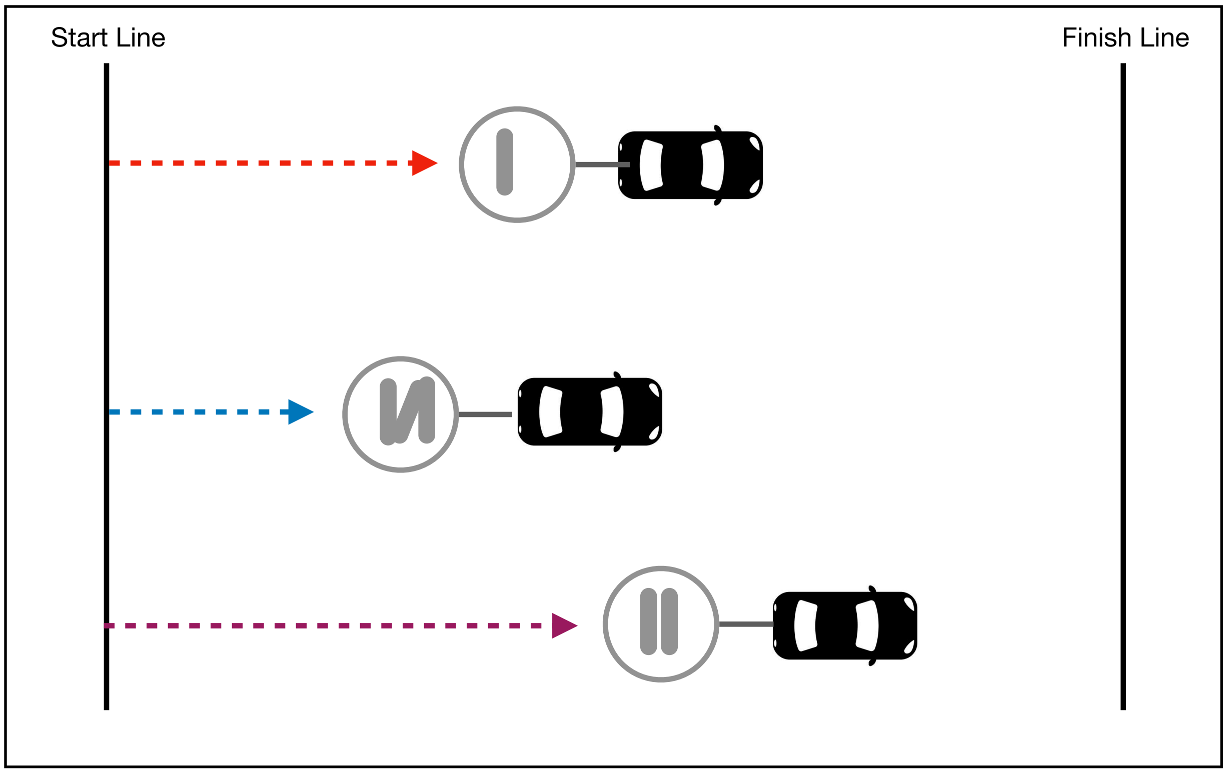
RULES

- Robots will be pulling paper bowl “sleds” with rolls of pennies on board as weight.
- Your robot must have a designated place for attaching a hook that is attached to the weight-pulling sled. It is best if this is directly anchored onto your EV3 Brick.
- Your robot must successfully pull the designated weight across the field.
- Your robot will begin pulling 1 roll of pennies. Once it crosses the field successfully, it will repeat the distance, but with 2 rolls of pennies. Robots will continue crossing the field with an additional roll added each time.
- When a robot is unable to reach the finish line, the distance will be measured and used in case of a tie.

HINTS/TIPS

Consider adjusting these variables to prepare your robot to be able to pull the most amount of weight:

- Programming (motor power)
- Robot weight
- Wheels used
- Gears (direct drive vs. geared)



NOTES & OBSERVATIONS

Competition 5: Sumo Bots

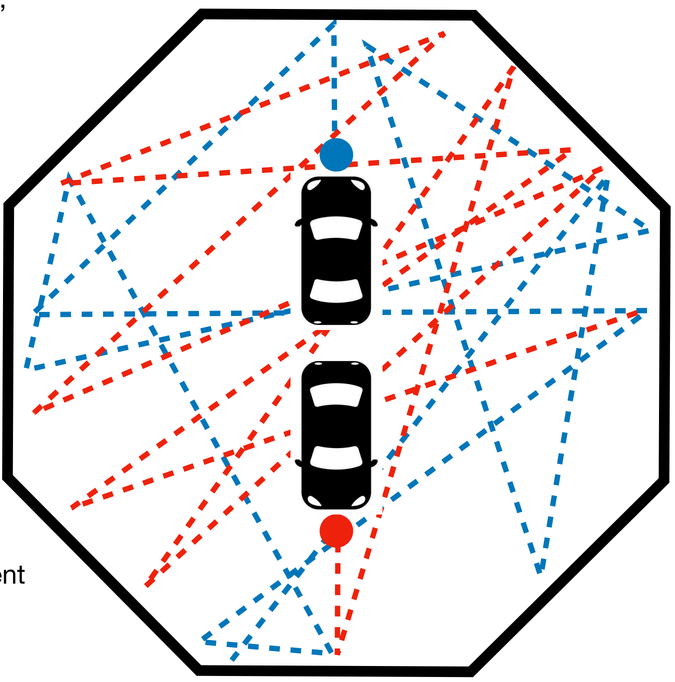
YOUR OBJECTIVE

Your objective is to program a robot that will go up against an opponent robot inside a Sumo Ring. The Sumo Ring is a white field with a black tape line defining its edges. Robots should be programmed to move forward until it detects a black line, then turn around and move forward again, and continue this movement infinitely. When robots collide inside the ring, one robot will eventually push its opponent out of the ring. The robot that remains inside the ring is declared the winner of that round.

Save your program as COMP 5 SUMO.

RULES

- Your program must include a 3 second delay before moving, to allow time for participants to clear the ring.
- Your robot is considered "out of the ring" when the entire EV3 brick or a majority of the robot's wheels are past the black perimeter.
- No humans are allowed to touch the SumoBots during matches, unless directed to do so by the referee.
- The referee may declare a stalemate and restart the match. if multiple SumoBot collisions have occurred without resulting in a winner, or SumoBots' paths do not appear as if they will lead to a collision. If needed, the referee will provide instructions on altering SumoBot starting positions.
- If a robot flips its opponent onto its side or back, immobilizing it, that robot wins.
- Rematches are at the discretion of the referee.
- A single elimination bracket will be used to determine tournament champions.



SUMOBOT ADDITIONS

Here are some design and programming variables you might consider adjusting in order to give your SumoBot an advantage over your opponents:

- Adding armor, hammers, and other battle-bot type additions
- Experiment with gears, different wheels, and robot weight
- Adding failsafe programming to put robot into overdrive when at risk (IE: incorporating more sensors to detect crossing the line from a different direction, etc.)

NEED HELP PROGRAMMING?

- How to Program SumoBots: https://www.youtube.com/watch?v=ZC2WVCtvtM8

NOTES & OBSERVATIONS