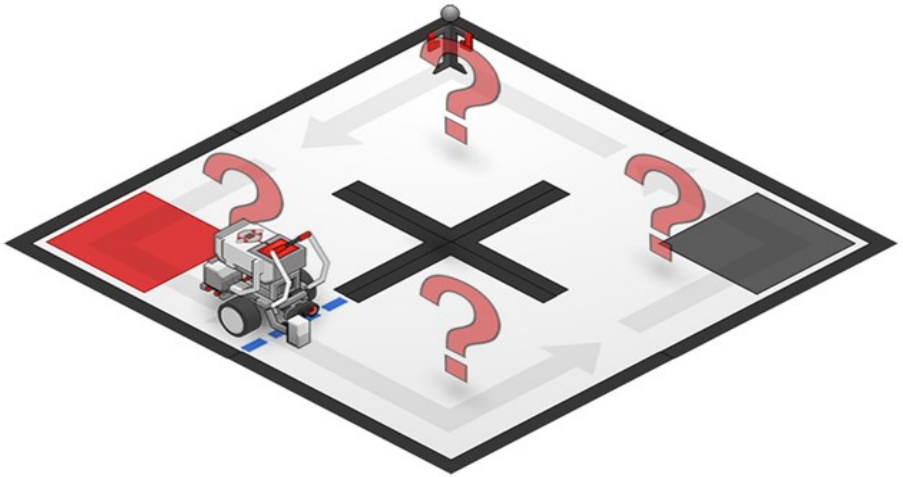
PHASE 2

1. WRITE 1 PROGRAM THAT WILL TRAVEL ALL 4 ROOMS
2. THE ROBOT MAY START AT ANY ROOM'S ENTRANCE
3. THE ROBOT'S TRIP CAN BE EITHER CLOCKWISE OR COUNTER-CLOCKWISE
4. THE ROBOT MUST RETURN TO WHERE IT STARTED
5. THE LOCATION OF THE ROOMS WILL BE RANDOMIZED EACH RUN



☐ **Show Teacher:**

Program 1 – Fire Room _____

Program 2 – Rescue Room _____

Program 3 – Walled Room _____

Program 4 – Clear Room _____

Final 4 Program _____

☐ **Document programs 1 through 4 and the final four programs on separate slides. Include program, detailed comments on program function and robot behavior and pictures supporting the challenge.**

Advanced Programming Lego Mindstorm EV3

Name (first & last) _____

Team Name _____

Table Number _____

Period number _____

EV3- Task Assignment
"Moving Straight"

As a team, work to develop the project outlined below. Each of these steps must be *demonstrated* successfully to one of the other teams. That team will then sign off indicating you were successful.

Date	**Robot Building**	Sign-off
	Robot constructed according to instructions	

Date	**50 CM Challenge**	Sign -off
	Mini Challenge Successful	

Date	**Cargo Retrieval**	Sign -off
	Mini Challenge Successful	

***Close Shave Challenge (NXT Video Trainer 2.0- Click Moving Straight Number 14)*:**

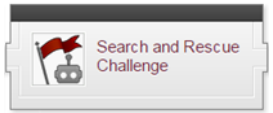
Use the provided layout and have another team sign off on your demonstration, before arranging with the teacher for your ***Close Shave Challenge*** presentation. On the back of this sheet, explain in detail the mathematics you used to complete ***Close Shave***, and be prepared to explain it as you demonstrate the challenge.

Date	***Close Shave Challenge***	Sign-off
	Math for Close Shave shown	
	Successful Close Shave	
	Successful completion of the challenge (on back)	(Teacher)

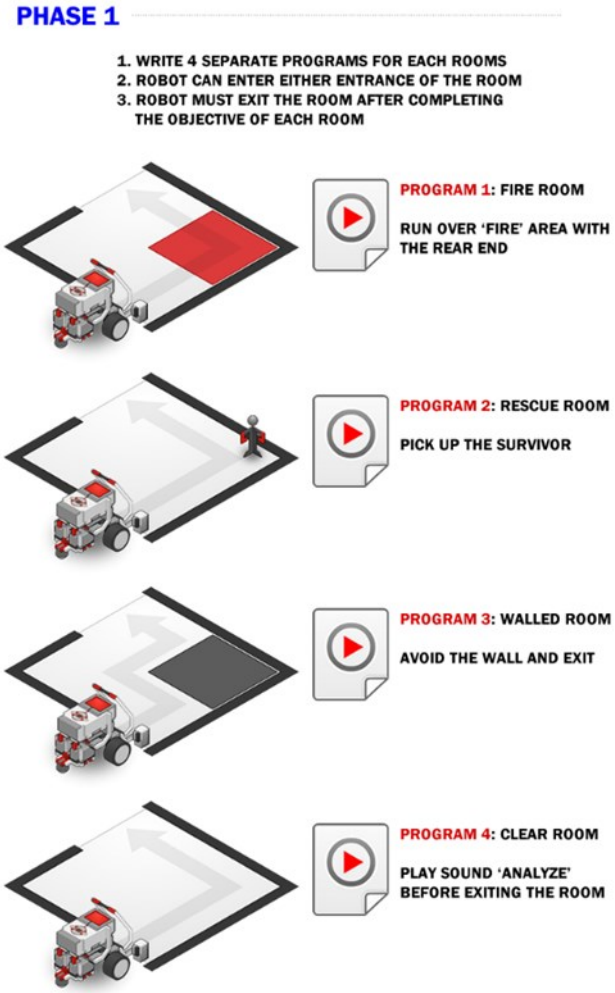
2. Which of the following is not an important element in a good Engineering Process?

◊ Researching the problem
◊ Planning the development
◊ Prototyping the solution
◊ Testing the prototype
◊ Commercializing the product
◊ Ignoring resource limitations

☐ **Do: Final Challenge – Search and Rescue**



In this Challenge, you will use everything you've learned to create a rescue robot that will enter a 4-room building. The robot must perform 4 unique actions for 4 unique rooms, which will be randomized in order to simulate a hazardous area where you can never know what will be encountered. The robot must complete all 4 rooms, and return to the starting point.



☐ **Watch: Resources 3: Project Planning**

1. What main topic does this video address?
◊ How to build a car out of LEGO bricks
◊ How to write the most efficient program code
◊ How a robot "thinks" about its surroundings
◊ How to coordinate a team of people working together on the same problem

2. What does a Design Specification do?
◊ Explain the importance of the problem being solved
◊ Align team members' ideas of what is being built
◊ Distribute the work evenly among team members
◊ Keep track of the days that people have shown up to work on the robot

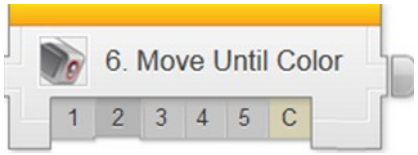
3. Which of the following can help your team finish its task on time?
◊ Assigning clear responsibility for each part to a specific team member
◊ Working separately until the day the project is due
◊ Prioritizing parts that need to be finished before other parts can be worked on
◊ First and third answer, but not the second

☐ **Watch: Resources 4: Engineering Process**

1. What main topic does this video address?
◊ How to coordinate a team of people working together on the same problem
◊ What engineering is, and how it works
◊ Imagining the solution to a problem
◊ Types of terrain that are most suitable for robot use

NOTES:

Introduction to Programming
Lego Mindstorm EV3
Behaviors: Sensors – Move Until Color 1-5 & C
(found online in Robotics Academy EV3)



Check off þcompleting sections and challenges in the following order:

☐ **Watch Color 1 : Introduction to Autonomous Vehicle**

1. What are some of the challenges a self-driving car must overcome?
◊ Finding a path to the destination
◊ Following the road
◊ Obeying traffic laws and signals
◊ Avoiding other vehicles
◊ All of the above

2. What will the robot need to detect with the Color Sensor?
◊ The speed of the robot
◊ Distance from the vehicle in front of the robot
◊ Width of the road
◊ Color of a traffic light

☐ **Do Gyro 2 : Robot Configuration**

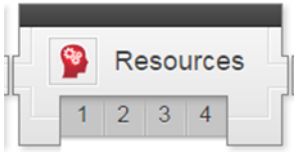
Configure Color Sensor to Robot per instructions.

☐ **Watch Color 3 : Wait for Green**

1. What does this program do?
◊ Wait for the Color Sensor to see a Red object, then move forward
◊ Wait for the Color Sensor to see a Green object, then move forward
◊ Wait until an object is moved out of the way, then move forward
◊ Move forward until it sees a Green object

Introduction to Programming
Lego Mindstorm EV3

Final Challenge: Resources *(found online in Robotics Academy EV3)*



☐ **Watch: Resources 1: Flowcharts**

1. What is a Flowchart?
◊ A graphical representation of a robot's plan of action, including decisions
◊ A series of pipes and wires that illustrate electricity flow
◊ The overall map of a program's progress
◊ The document that tracks the number of weeks left in the project cycle

2. Why are Flowcharts important?
◊ They help programmers visualize the decision-making process on the robot
◊ EV3 programs can be written in Flowcharts and loaded directly into the robot
◊ Robots internally reprocess all instructions into flowcharts in order to think
◊ The flowchart calculates the trajectory of the rejected plants.

☐ **Watch: Resources 2: Iterative Design**

1. What method does this video recommend for building a solution to a problem?
◊ Construct the entire solution all at once
◊ Solve all the parts separately, then combine them in one step
◊ Solve a part, add it to the solution, test the combined whole, then repeat
◊ Build almost everything at once, then try to get the last part to fit

NOTES:

2. When multiple colors are checked in the "Set of Colors" area, what will the Wait Block do?

- ◊ Wait for ANY of the colors to be seen
- ◊ Wait for ALL of the colors to be seen at the same time
- ◊ Wait for ALL of the colors to be seen at least once each
- ◊ Wait for ALL of the colors to be seen in the order indicated by thenumbers

The EV3 Color Sensor can detect 7 different colors, plus the absence of color.
Each of these 8 colors is labeled with a different number.

☐ Do Try it! 1: No Color

What does the No Color "color" mean in the Set of Colors menu? Change your program so that it waits for "No Color" (Option 0). Make sure you unselect all the other colors.

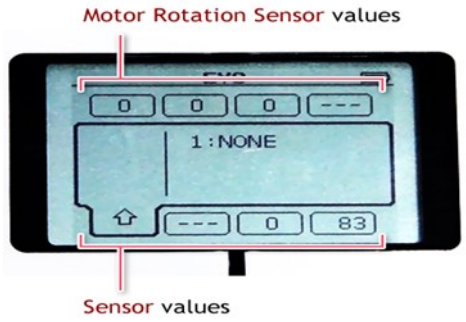
Place various objects in front of the Color Sensor, and run the program. What triggers the Wait For No Color Block? Try it!

☐ Do Try it! 2: Port View: Color Sensor Values

You can see the Number value of the currently detected color directly on the EV3's view screen, in the Port View Mode.

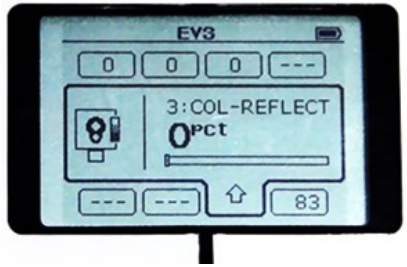
1. Use the Left and Right Buttons on the EV3 to navigate to the EV3 Apps menu (), and press the Enter Button to select 'Port View Mode'.

The 8 blocks at the top and bottom of the screen represent the 8 ports on the EV3.



- Motor Rotation Sensor values are displayed across the top.
- Sensor Values are displayed across the bottom, depending on what is plugged in.

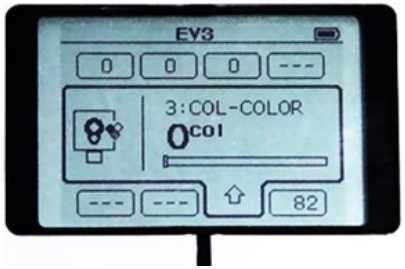
2. Use the Left and right Buttons so that the Color Sensor's Port View block is selected.



"**COL-RELECT**" means the sensor is reading amount of colored light being reflected.
This is not what we want, since we want to see exactly what color the sensor is seeing, not the amount of color.

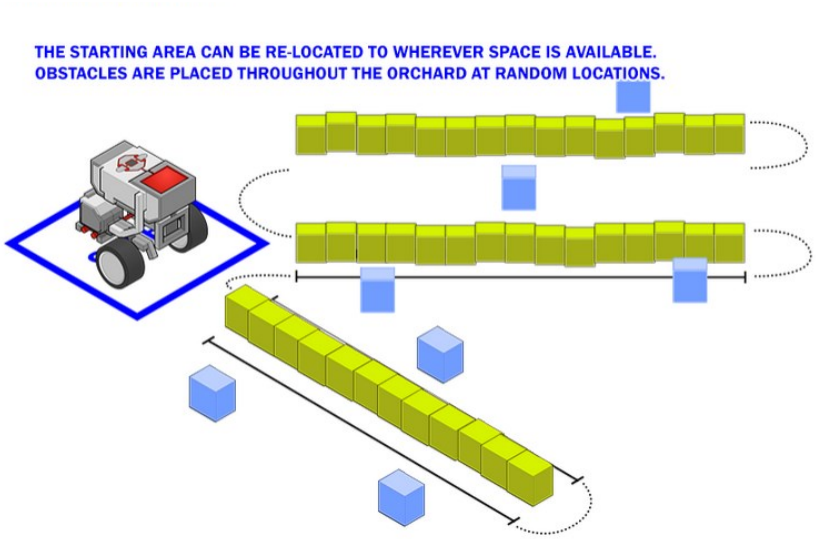
3. Press the Enter Button on your EV3 while the Color Sensor's Port View block is selected.
Move the Up and Down Buttons to select the type of reading **'COL-COLOR**', and press the Enter Button to set new reading type.

4. Now the Color Sensor's Port View will display the color value. (**'COL-COLOR'**)



NOTES:

☐ **Watch & Do: Switch-Loops 6: Obstacle Orchard Challenge**



☐ **Show Teacher _____**

☐ **Document with program snapshot , detailed comments and pictures:**

- ☐ **Obstacle Detection: Move Until Black**
- ☐ **Obstacle Orchard Challenge**

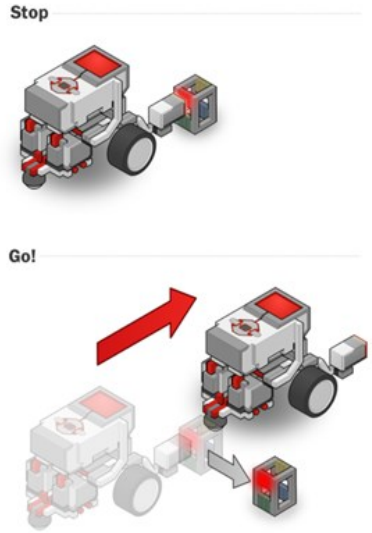
1. Point the Color Sensor directly at the Red side of the Color Crate. What value do you see?

2. What value should you see if the Color Sensor is pointed at the Blue side of the Color Crate?

3. Point the Color Sensor away from the Color Crate and any strong lights. What reading does it give?

☐ **<u>Do Mini Challenge 1: Railroad Crossing</u>**

Instead of red and green lights, some traffic signals simply use signs that raise and lower in the path of traffic.
Program your robot so that instead of waiting for a green light, it waits for the red stop sign to be taken away.



☐ **<u>Watch Did you know? How the Color Sensor Works</u>**

☐ Watch Color 4: Forward Until Red

1. What does this program do when run?

Move forward when the robot sees a red object
Move forward until the robot sees a red object
Move backward when the robot sees a red object
Move faster when the robot sees a red object

☐ Do Mini Challenge 1: Froward To Stop Line

In addition to signs and lights, self-driving cars also need to obey pavement markings, like this stop line telling where to stop.

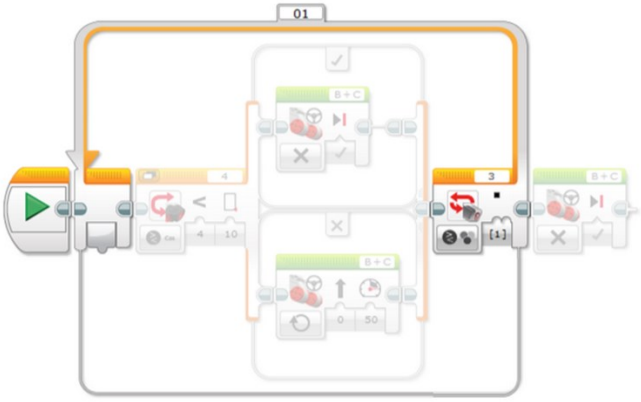
Modify your Color Sensor attachment, so that it faces downward, and program your robot to drive forward until it reaches the line.

☐ Show Teacher: Forward Until Yellow (Yellow line on dark surface)
☐ Show Teacher: Forward Until Black (Black line on light surface)

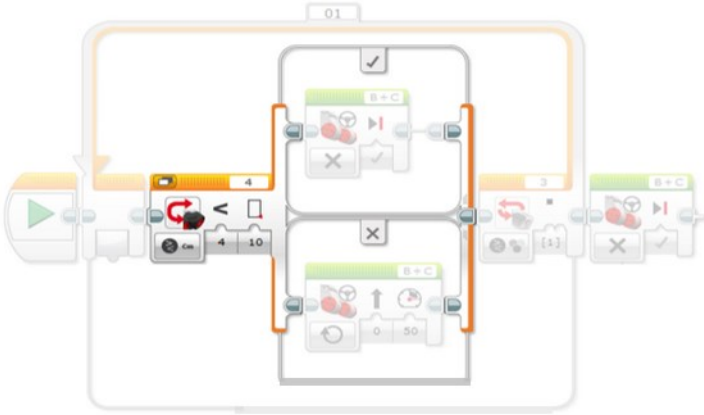
☐ Color 5 : Color Sensor Review

The program shown below is sample code for making your robot wait until it sees the color green. Provide a detailed explanation of how the program blocks work and how the robot responds.

☐ Program Review: Switch-Loops 5



Provide a detailed explanation of the highlighted program block and how it will work.



Provide a detailed explanation of the highlighted program block and how it will work.

How do repeated decisions allow the robot to watch both sensors at once?

◊ The program resets and restarts at the beginning every time the readings of any sensor changes
◊ The robot calculates the path with both sensors at first and runs algorithms for the best way to run the course
◊ The robot activates a specific sensor at certain times throughout the course
◊ Every operation is fast, and does not block other commands from running

In the final version of the program, the robot ends up processing the Switch inside the Loop...

◊ Only once, ever, because it's a Switch
◊ Four times because the Loop goes for four rotations
◊ Thousands of times, because the robot processes the loop very quickly and encounters the Switch many times

☐ **Do: Mini Challenge 1: Obstacle Detecting Move Until Black Line**

Modify the Obstacle Detection program you wrote so that it will move safely (stopping when an obstacle is in front of it, moving when there is none) until the Color Sensor detects a black line on the table, rather than until the robot has traveled a certain number of rotations.

☐ **Show Teacher _____**

The program shown below is sample code for the Railroad Crossing mini-challenge. Provide a detailed explanation of how the program blocks work and how the robot responds.

Provide a detailed explanation of how the program blocks work and how the robot responds.

☐ Watch Color 6: Traffic Signal Challenge

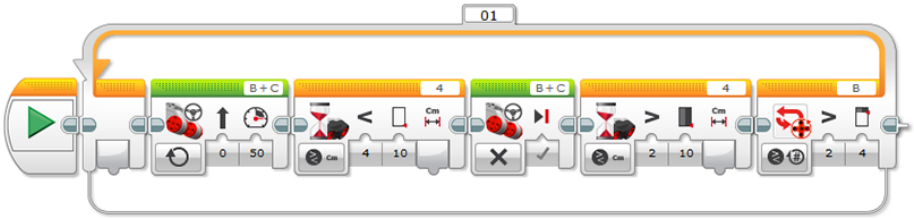
In this challenge, you will program your EV3 robot to through three different intersections, each of which has a traffic signal. The traffic signal, which can be either the colored block or the red/green card, is held by hand at a set height. Unlike a camera, the detection range of the Color Sensor is short, so you will need to modify its placement on the robot so that it can see the traffic signal and react appropriately.

The robot does not need to stop on its own after passing through all three intersections (it can be stopped by hand).

☐ Document the following programs on a new slide titled Color Sensor (including a diagram and text explaining your programs).

- ☐ Wait Until No Color (Railroad Crossing)
- ☐ Forward To Yellow Line
- ☐ Forward To Black Line
- ☐ Traffic Signal Challenge

Why doesn't this program work for Obstacle Detection?



- ◊ The Wait Blocks prevents the flow from reaching the end of the loop and checking Rotation Sensors
- ◊ The program ends immediately when it detects an object, regardless of the Loop
- ◊ The Loop makes the robot go backwards
- ◊ The first Move Steering block only goes for one rotation, then ends the program

Instead of using Waiting approach and long movements, the solution you will learn next will involve:

- ◊ Rapid checking of sensors
- ◊ Sensor recombination fork
- ◊ A new multiple-sensor Wait Block
- ◊ A new type of Loop Block

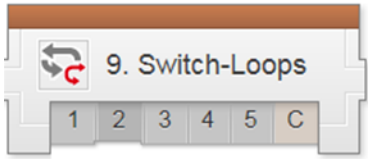
☐ Watch & Do: Switch-Loops 4: Obstacle Detection

Instead of thinking about the four-rotation Obstacle Detection as one big movement, the video suggests thinking about it as...

- ◊ Using a brand new type of Wait Block that allows multiple sensor checking
- ◊ Downloading the program into the robot and using built-in options to run parts of the program at certain times
- ◊ A long series of tiny movements that add up for four rotations
- ◊ All of the above are valid suggestions

Introduction to Programming
Lego Mindstorm EV3
Decisions: Switch-Loops 1-5 & C

(found online in Robotics Academy EV3)



☐ **Watch: Switch-Loops 1: Introduction to Autonomous Tracker**

What is the difference between this version of the Orchard Challenge and the original version in the Turning Chapter?

◊ There are more trees
◊ The robot must complete the challenge within a certain time limit
◊ There will be randomly placed obstacles in the robot's path
◊ There is no difference

☐ **Do: Switch-Loops 2: Robot Config. (Ultrasonic Sensor)**
☐ **Watch & Do: Switch-Loops3: Obstacle Detection Failures**

Why doesn't this program work for Obstacle Detection?

◊ The program detects an object and stops before moving all four rotations
◊ The program doesn't run because it's waiting to detect something in order to run
◊ The Move Steering Block holds up the program, preventing it from checking other sensors in the mean time
◊ All of the above are valid reasons on why the program doesn't work

Engineering Challenge #1
Light-Activated Robo-Dragster

Design, build & program a Dragster-Bot that:

5 pts- Begins racing at the removal of the starting "flag"

5 pts- Stops as a result of reaching the finish line (race is over when dragster comes to a stop)

5 pts- Stays within its racing lane

5 pts- Covers the track in the shortest time possible

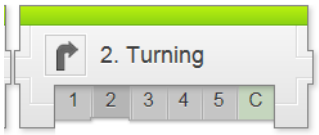
5 pts- Uses only the parts from one kit

5 pts- The Light Sensor must be at the back of the robot

Points /30

+5 Fastest Dragster in Class

Introduction to Programming
Lego Mindstorm EV3
Behaviors: Turning 1-5 & C *(found online in Robotics Academy EV3)*

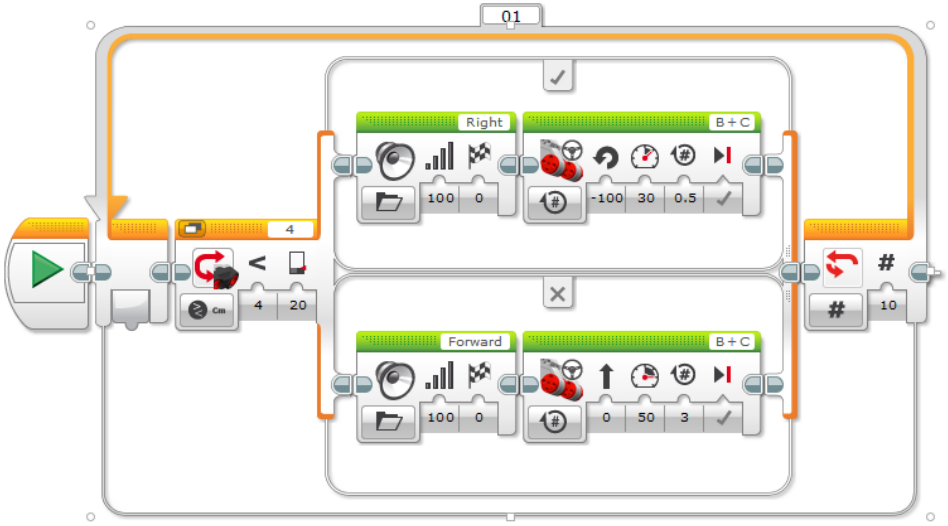


☐ **Watch Turning 1: Introduction with Crop Tractor**

1. Why is it important to be able to drive through an orchard?
◊ To perform specialized tasks to different types of crops.
◊ GPS is required while navigating around the orchard.
◊ To perform tasks like inspection and spraying which cannot be done as effectively through other means.
◊ All of the above.

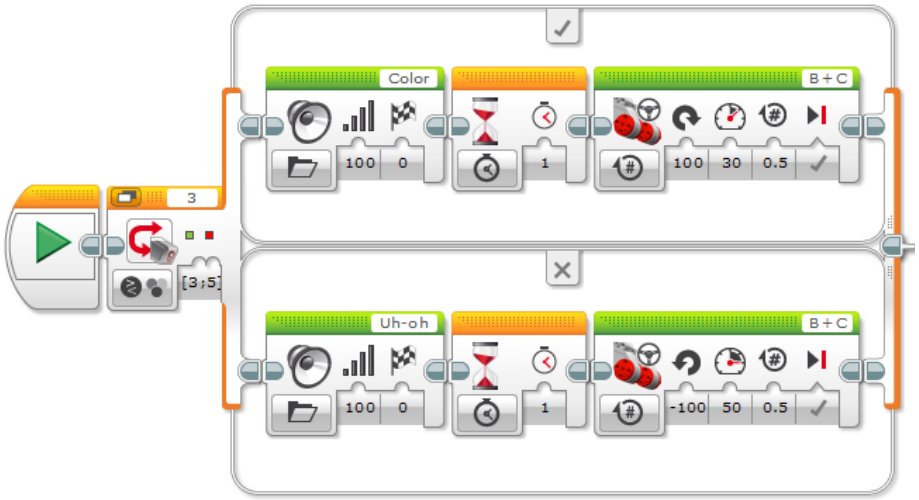
2. What is the advantage of the Autonomous Tractor over a human driver?
◊ Reduces the need for humans to perform the repetitive task of driving through the orchard over and over.
◊ Reduces exposing human to hazard areas while performing inspections.
◊ Autonomous Tractor can travel through an area where a human driver may get lost.
◊ There is no big advantage.

3. In addition to basic turning, what additional, new knowledge will help you complete this challenge?
◊ How a robot moves straight.
◊ How a robot move back and forth.
◊ How a robot turns, and different types of turns.
◊ All of the above.



Explain, in detail, what the above program shown will do.

☐ **Watch: Switches 6: Strawberry Plant Challenge**
☐ **Build: Switches 6: Robot Plant Sorter**
☐ **Do: Switches 6: Strawberry Plant Challenge**
(Blue & White = Good plants. Red, Yellow & Black = Bad plants)
☐ **Show Teacher _____**
☐ **Document with program, comments and pictures:**
- ☐ **Move If Clear**
- ☐ **Color Sensor Compare**
- ☐ **Maze Runner**
- ☐ **Strawberry Plant Challenge**



Explain, in detail, what the switch block shown will do.

☐ **Turning 2: Robot Config (build EV3 driving base)**

☐ **Watch Turning 3: Turning in Place** ***(Name your new project, "TURNS")***

1. What does the robot do when the *Turn Right* program is run?

◊ Move straight forward
◊ Spin to the robot's right without moving forward at all
◊ Spin to the robot's left without moving forward at all
◊ Spin for 360 degrees

2. TRUE or FALSE: With "Rotations" on the Move Steering Block to 1, the whole robot rotates 1 time.

TRUE: the robot will turn around 1 time
FALSE: the wheels will turn 1 time, not the body

☐ **Do Mini Challenge 1: 90 Degree Turn**
☐ **Observed by teacher _____**
☐ **New slide in "Program Documentation" titled TURNS (90° Turn)**

Try It! #1: Direction of Turn

Moving the Steering slider all the way to the right makes the robot turn to the right, in place. What happens if you move it all the way to the left?

The white pointers on the EV3's tires help you to see how much the wheels are rotating. Run your *Turn Right* program again, and watch the pointer on the robot's right wheel.

3. How much did the robot's wheel turn during this movement?

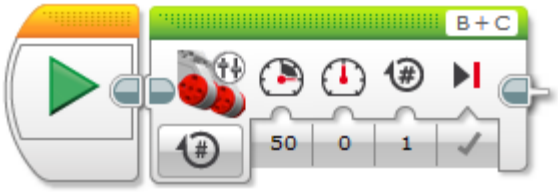
◊ 1 rotation
◊ 1 degree
◊ Enough to make the robot spin completely around one time
◊ One lap around the table

4. What does the "1 rotation" refer to in the Move Steering Block's controls?

◊ 1 full rotation of the robot's body during a turn
◊ 1 rotation of the robot's wheels
◊ 1 time that the robot is picked up and turned around
◊ 1 rotation of the Earth and its axis

☐ Watch Turning 4: Other Turns

1. In the movement you programmed, the left motor was told to move forward at 50% power, and the right motor was told to...



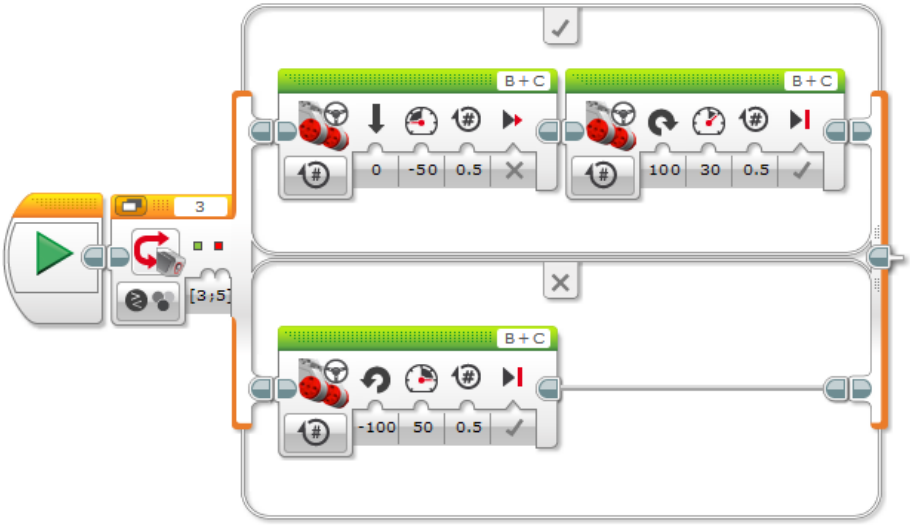
◊ Move forward at 50% power
◊ Move backwards at 50%
◊ Stay in place
◊ Spin freely

2. What kind of turn did the robot produce with one motor running and one motor stopped?

◊ Goes straight
◊ Turns in place
◊ Turns "wide"
◊ Backs up

☐ *Do Mini Challenge 1: Dizzy Drill*

Program your robot to run out to an obstacle, go around it, then come back. Use a small object as an obstacle, and program your robot to travel around it. The robot may start and finish in any position behind the start line.

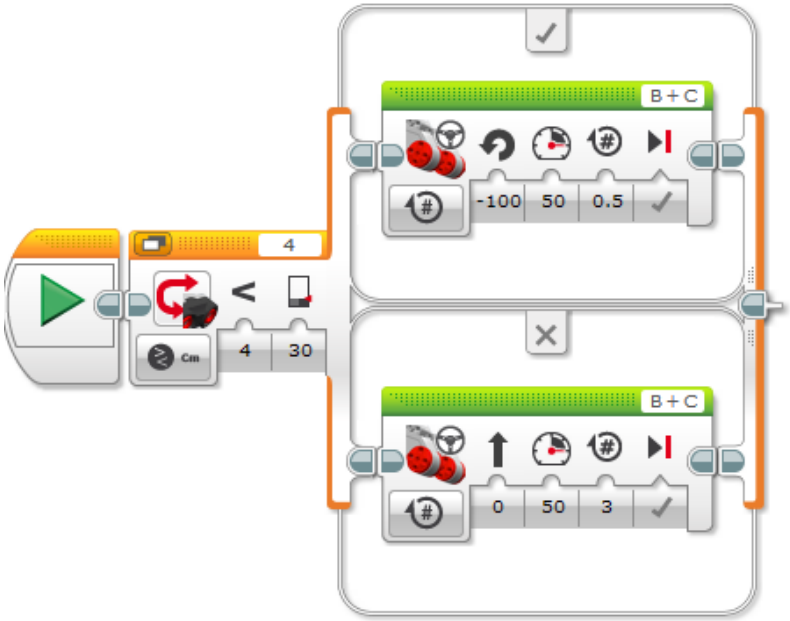


Explain, in detail, what the switch block shown will do.

- ☐ **Do Mini Challenge 1: Smarter decisions challenge**
- ☐ **Show Teacher _____**

- ☐ **Answer Switches 5: Switches Review [follow directions!!]**

Start statements with "If" and "Then", followed by statements starting with "If Not" and "Then".

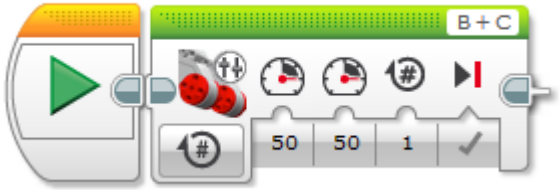


Explain, in detail, what the switch block shown will do.

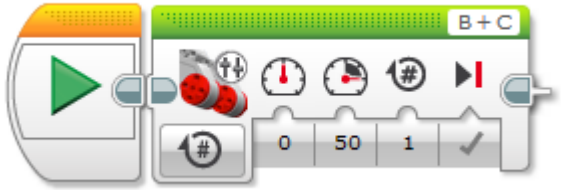
- ☐ Dizzy Drill Observed by teacher _____
- ☐ "Program Documentation" (Dizzy Drill)

Try It! #1: Different Motions

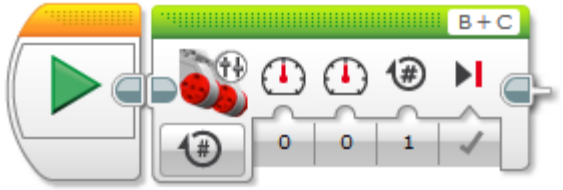
You can create many different types of motion by combining different motor speeds. Try each of the following to see what you get!



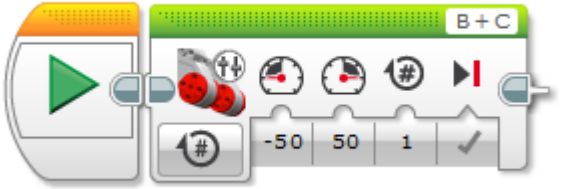
What happens?



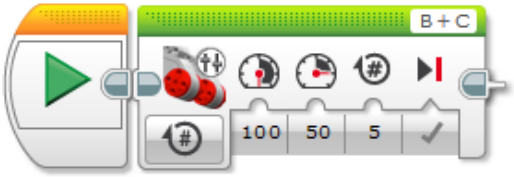
What happens?



What happens?



What happens?

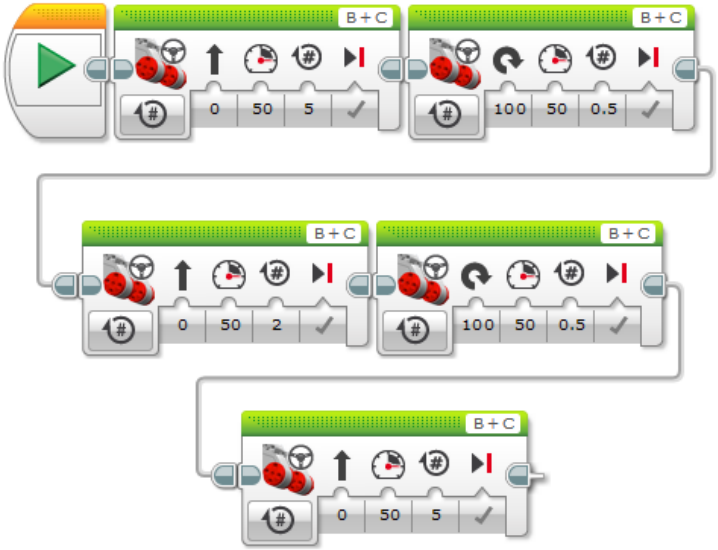


What happens?

Turning 5: Turning Review

What is the programming blocks name?

How does the tribot behave with this program?



How does the tribot behave with this program?

☐ **Do: Mini Challenge 1: Color Sensor Compare Switch**

☐ **Show Teacher: Mini Challenge 1: Color Sensor Compare Switch**

(*turn right if blue or white and turn left for any other color*)

☐ **Do: Mini Challenge 2: Color Name Reader**

(Create a program that will say the name of the detected color ((white, blue, red, yellow, & black out loud, by playing the sound for that color word)

☐ **Show Teacher: Mini Challenge 2: Color Name Reader**

☐ **Watch & Do: Switches 4: Looped Decision**

What happens when you place a Switch inside a Loop?

◊ The robot runs in a circle until told to stop

◊ The robot makes a decision once, then repeats the result many times

◊ The robot makes a decision many times, taking whatever action is appropriate each time

◊ You cannot place a Switch inside a Loop

When a Switch is placed inside a Loop…

◊ Both the Switch and Loop operate normally, but the arrangement is useful

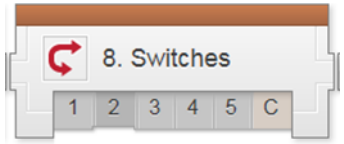
◊ The software interprets the Switch+Loop structure as a special construct

☐ **Do Try It! 1: Maze Runner**

Adjust your Forward movement to go one tile per movement, rather than 3 rotations.

What happens?

Introduction to Programming
Lego Mindstorm EV3
<u>Decisions: Switches 1-5 & C</u> *(found online in Robotics Academy EV3)*



☐ **Watch: Switches 1: Introduction to Strawberry Plant Sorter**

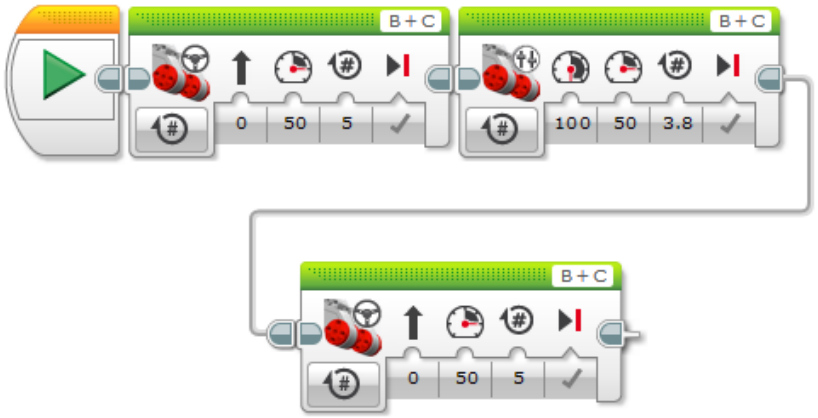
A Switch will allow your robot to…
◊ Write very long programs
◊ Write multiple programs
◊ Repeat behaviors within a program
◊ Make a one-time decision

☐ **Do: Switches 2: Robot Config (Ultrasonic Sensor)**

☐ **Watch & Do: Switches 3: Move If Clear**

The robot will move forward...
◊ if there is no object in front of the Ultrasonic Sensor when the program starts
◊ if there is an object in front of the Ultrasonic Sensor when the program starts
◊ if an object passes in front of the Ultrasonic Sensor at any time
◊ until an object passes in front of the Ultrasonic Sensor

The robot makes its decision about whether to move forward or turn...
◊ Once, when the Switch is reached in the program
◊ Once, when the Switch sees an object
◊ Continually while the program is running
◊ The robot always moves, no matter what



How does the tribot behave with this program?

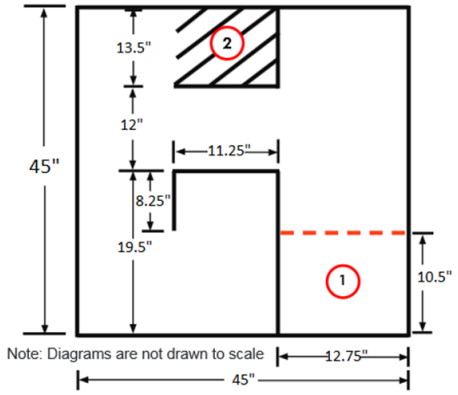
Task Assignment

Engineering Challenge #2

Walled Maze Challenge

READ On NXT Video Trainer 2.0 -> Behaviors -> Turning -> 12, follow the diagram below to set up and complete the *Maze Challenge* to receive your final points.

- Program your robot to travel from box 1 to box 2 and BACK to box 1 in the least amount of time. You may NOT use any sensors, however, other any other parts from your kit may be used.



Total: _____ / 25

Introduction to Programming
Lego Mindstorm EV3
Behaviors: Sensors – Move Until Touch 1-5 & C
(found online in Robotics Academy EV3)

☐ **Watch Touch 1: Introduction**

1. Why are sensors important to robots?

◊ They allow multiple commands to run in order
◊ They give the robot information about its surroundings
◊ They allow robots to repeat similar tasks
◊ All of the above

2. What is the advantage of Sensor Control over Sequential Commands?

◊ The robot can remember hazard areas
◊ The robot can perform actions a lot faster
◊ The robot can react to its environment
◊ There is no big advantage

☐ **Touch 2: Configure Robot with Touch Sensor**

☐ **Watch Touch 3: Wait for Touch**

1. What does the robot do when the WaitTouch program runs?

◊ Runs continuously until the Touch Sensor is pressed in
◊ Waits for 1 second, then moves 1 rotation
◊ Waits for the Touch Sensor to be pressed in, then moves 1 rotation
◊ Runs for 1 rotation

2. The program waits BEFORE it moves because...

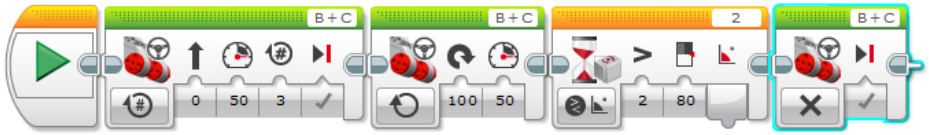
◊ The Wait Block comes first in the program
◊ The Wait Block always takes priority over Move Blocks

☐ **Watch Did you know? How the Gyro Sensor Works**

☐ **Gyro 5 : Gyro Sensor Review**

Provide a detailed explanation of how the following program will run.

Provide a detailed explanation of how the following program will run & how the robot will react.



☐ **Gyro 6 : Mower Challenge**

☐ **Watch Challenge Review**

☐ **Modify Robot for Scoop Attachment**

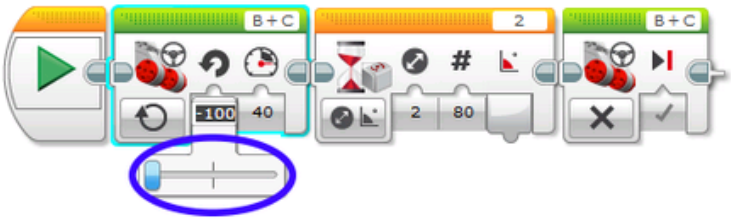
☐ **Document the following programs on a new slide titled Gyro Turns (including a diagram and text explaining your programs).**

- ☐ Gyro Sensor Mini Challenge Around Square Box
- ☐ Gyro Sensor (with Scoop) Mower Challenge

3. What is the difference between a workaround and a solution?

◊ A workaround is preferred over a solution
◊ A solution removes the source of the problem, while a workaround only reduces its effects
◊ A workaround removes the source of the program, while a solution only reduces its effects
◊ None, as both are equally the same

☐ Do Try It! Try it! 1 Left Turns



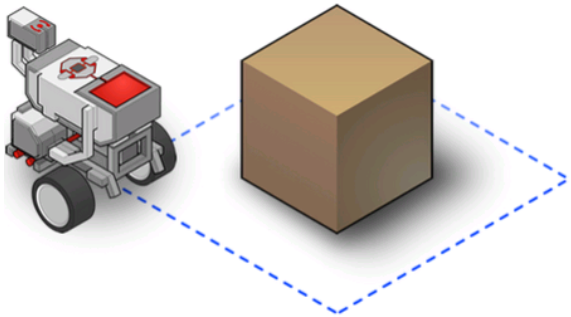
Does the same Wait for Gyro block work for left turns?

Try changing your program to turn 90 degrees to the left instead.

What happens?

☐ Do Mini Challenge 1: Square Box

Program and make the robot complete a full lap around a square box, using the Gyro Sensor to control all of its turns.



☐ Do Try it! 1: Already Pressed

What happens if you're already holding down the Touch Sensor's button when you start running the program?

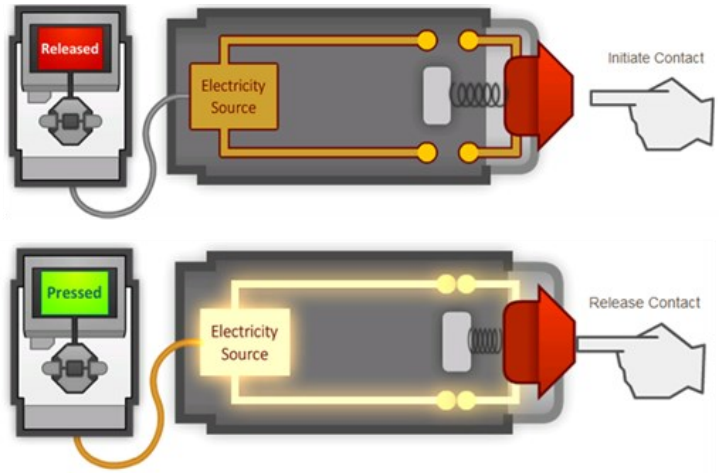
☐ Do Try it! 2: EV3 Buttons

The 5 buttons on the front of the EV3 (not counting the Cancel button) can be used as Touch Sensors!
Try changing the Mode of the Wait Block to:
"Brick Buttons > Compare > Brick Buttons" and running your program.
Once it's running, press the middle button on the front of the EV3!
What happens?

☐ Did you know? How the Touch Sensor Works

When the Touch Sensor is pressed, it _____ an _____ circuit, allowing current to flow.
If the Touch Sensor is released, the _____ is _____ and no _____ flows.

The flow (or lack) of current is detected by the EV3, allowing it to determine the Touch Sensor is pressed.



☐ Watch Touch 4: Forward until Touch

1. What does a Move command do when its Mode is set to “On”?

◊ Turn the motors on
◊ Turn the motors on for a certain number of rotations
◊ Turn the motors on until the Touch Sensor is triggered
◊ Combines with the next block to make a special command

2. What does a Move command do when its Mode is set to “Off”?

◊ Turn the motors off
◊ Waits for the Touch Sensor to be pressed
◊ Wait for the Touch Sensor to be pressed, then turn the motors off
◊ End the program

☐ Do Try it! 1: Forward Until Release

The Wait - Touch block can wait for the sensor to be "Released" as well as "Pressed".

What happens if you set the Wait - Touch block to "Released" and run it with an empty box holding down the sensor?

☐ Do Mini Challenge 1: Vacuum

☐ Teacher Sign-Off

Program the robot to touch all four walls of a room, using its Touch Sensor to know when it has reached each one.

☐ Touch 5: Touch Sensor Review

1. Provide a detailed explanation of how the following program will run.

☐ Do Gyro 2 : Robot Configuration

Configure Gyro Sensor to Robot per instructions.

☐ Watch Gyro 3 : Turn for Angle (Part 1)

1. Because of the way it is attached to the robot, the Gyro Sensor measures:

◊ The amount the robot's body turns
◊ The amount the robot's wheels turn
◊ The amount the robot is "tipping" forward or backward
◊ When an oscilloscope is nearby

2. When the robot actually ran, what happened?

◊ The robot turned its body exactly 90 degrees
◊ The robot turned its body slightly more than 90 degrees
◊ The robot moved forward 90 degrees
◊ The robot spun in place forever

☐ Watch Gyro 4 : Turn for Angle (Part 2)

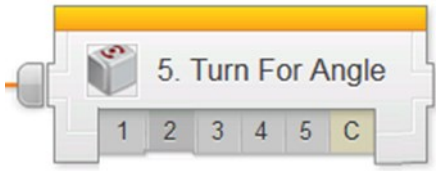
1. Which of the following factors contributes to the "overturning" problem?

◊ Sensor accuracy limitations
◊ Delay in sensing and signal transmission
◊ Physical momentum
◊ All of the above

2. Which of the following workarounds can help to reduce the "overturning" problem?

◊ Replace the Gyro sensor
◊ Telling the robot to wait until a value that comes "before" the one you actually want
◊ Use a different numbered port
◊ Press the cancel button as soon as it completed its turn

Introduction to Programming
Lego Mindstorm EV3
Behaviors: Sensors – Turn For Angle 1-5 & C
(found online in Robotics Academy EV3)



Check off þcompleting sections and challenges in the following order:

☐ **Watch Gyro 1 : Introduction to Golf Course Mower**

1. What does a gyro sensor help the robot to do?

◊ Move a precise distance
◊ Turn more precisely
◊ Use GPS
◊ None of the above

2. Why does the autonomous mower use a gyro sensor if it already has GPS?

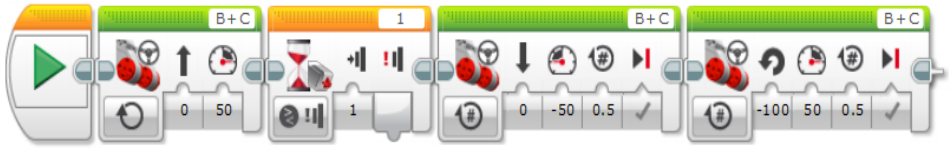
◊ Gyro sensor is more accurate than GPS
◊ GPS is sometimes blocked or unavailable
◊ GPS can be slow sometimes
◊ Robot moves faster using gyro sensor

3. Why should your robot use a gyro sensor even if it already has rotation sensors?

◊ Wheels sometimes slip and lose accuracy
◊ Wheel's rotation sensor cannot be used to detect body rotation
◊ Gyro sensor works independently from GPS
◊ All of the above

2. Provide a detailed explanation of how the following program will run.

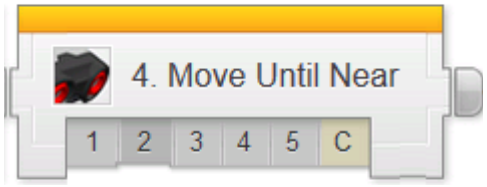
3. Provide a detailed explanation of how the following program will run.



☐ **Do Arm Position Challenge** ☐ **Teacher Sign-Off**
☐ **Teacher Review of completed work.**

☐ **Update Program Documentation (new slide titled Touch Sensor)**
- ☐ **from Try it! 2: EV3 Buttons program (with comments)**
- ☐ **from Try it! 1: Forward Until Release program (w/comments)**
- ☐ **from Mini Challenge 1: Vacuum program (w/comments & drawing)**
- ☐ **Arm Position Challenge program (with comments & drawing)**

Introduction to Programming
Lego Mindstorm EV3
Behaviors: Sensors – Move Until Near 1-5 & C *(found online in Robotics Academy EV3)*



☐ **Watch:** Ultrasonic 1 : Introduction to Hexarotor

1. Why might the Rangefinders (Laser or Ultrasonic) be preferred over Touch Sensors for detecting walls and obstacles?

◊ Hitting wall could damage the wall
◊ Hitting a wall could damage the robot
◊ It is quicker to detect obstacles at a distance
◊ All of the above

☐ Do: Ultrasonic 2: Robot Config

☐ Watch: Ultrasonic 3: Wait for Near

1. What does the Wait Block wait for before playing the "Hello" sound?

◊ The Ultrasonic Sensor to detect an object less than 50 cm away
◊ The Ultrasonic Sensor to detect an object more than 50 cm away
◊ The robot to travel less than 50 cm
◊ The Ultrasonic Sensor's reading to change by up to 50 cm

2. How do you select the sound file the robot plays?

◊ By clicking in the upper-right "File Name" blank on the block
◊ By speaking into the speakers of the EV3
◊ By adding a File Block

Task Assignment

"Arm Control"

As a team, work to develop the programs outlined below. Each of these completed programs must be *demonstrated* to one of the other teams. That team will then sign off indicating your program was successful.

Date	Program	Sign-off
	Gripper Close Program	
	Direction Challenge	

Fruit Picker

When you have completed this challenge, have another team sign it off before arranging with the teacher for your demonstrations. Be prepared to discuss your Flow Chart, NXT-G program details and how you overcame any difficulties that arose.

Date	Challenge	Sign-off
	Fruit Picker	
	Successful Fruit Picker Completion!!!	**(Teacher)**

Task Assignment

"Line Following"

As a team, work to develop the programs outlined below. Each of these completed programs must be *demonstrated* to one of the other teams. That team will then sign off indicating your program was successful.

Date	Program	Sign-off
	Follow Straight Line for 15s (step #8)	
	Follow Straight Line Optimized for 10s (step #9)	
	Optimized to follow "S" Curve (step #10) _____(best time)	

Click on the last step, #11, and complete the **RoboSlalom Challenge**. When you have completed this challenge, have another team check it off before arranging with the teacher for your demonstration. Be prepared to discuss your Flow Chart, NXT-G program details and how you overcame any difficulties that arose.

Date	Challenge	Sign-off
	RoboSlalom Challenge	
	Successful RoboSlalom Challenge Completion!!! **Best time:_____**	**(Teacher)**

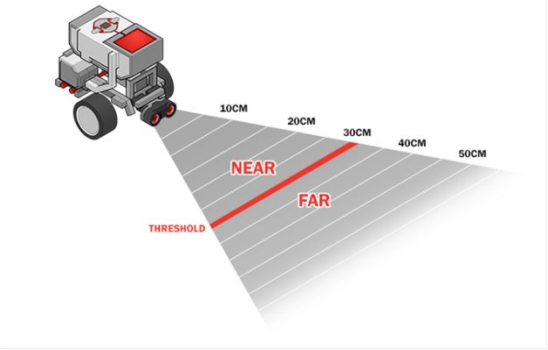
Do: Mini Challenge 1: Threshold Value

The Wait - Ultrasonic Sensor Block uses a "_____" to define what it is waiting for.

Rather than look for a specific value (like 1cm or 200cm), it sets a "_____" value that divides all the possible Ultrasonic Sensor values into two categories:

If the distance value is **above** the Threshold, it is considered "_____"

If the distance value is **below** the Threshold, it is considered "_____"

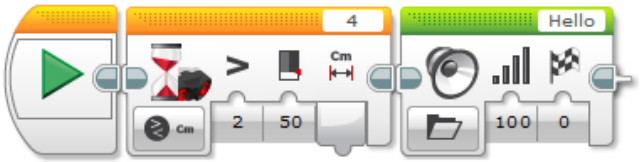


This way, the Wait Block does not have to worry about the difference between an object at 29cm and an object at 30cm; it only has to worry about whether the value is above or below the Threshold.

- ☐ Show Teacher: Mini Challenge 1: Threshold Value 10cm
- ☐ Show Teacher: Mini Challenge 1: Threshold Value 100cm

- ☐ Do: Missing Object Alarm (Wait for Far)

  Change the Wait Block's "Compare Type" setting to Greater Than (2). What does the Wait Block do if you set it to wait for a value Greater Than the Threshold instead of Less Than?



Place an object in front of the robot, and download and run the program. Now, try moving the object and see what happens.

☐ Watch: Did You Know?: How the Ultrasonic Sensor Works

☐ Show Teacher: Try It! 2: Sound Sentences

How would you get the EV3 to say one sound after another?

☐ Do: Try It! 3: Sensor Change Mode

What happens after changing the Mode of the Wait Block to Ultrasonic Sensor-Change – Distance Centimeters with direction setting to "Any" (2) and Amount setting to "5"?

☐ Watch: Ultrasonic 4: Froward Until Near

If the robot is facing a wall, it will move until...

◊ The Ultrasonic Sensor has traveled 50 cm

◊ The Ultrasonic Sensor is 50 cm from the wall

◊ The robot moves 50 cm total

◊ The robot detects an obstacle beyond 50 cm away

☐ Show Teacher: Mini Challenge 1: Backward Until Far

☐ Ultrasonic 5 : Ultrasonic Sensor Review

Suppose the robot is tracking the left edge of the line, and sees Black. Which way should it move?

◊ Forward and to the left, because the left edge is that way

◊ Forward and to the right, because the left edge is that way

◊ Straight forward because the line will curve eventually

◊ Turn left in place, because the left edge is that way

Which basic program pattern is being used in this Line Tracking behavior?

◊ Sequential Movements

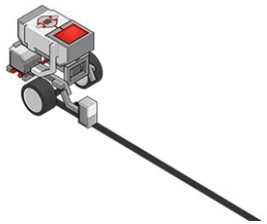
◊ Parallelism

◊ Repeated Decisions

◊ Variables

☐ **Do: Mini Challenge: Track Line for Rotations**

Program the robot to stop line-tracking after moving for four-rotations.



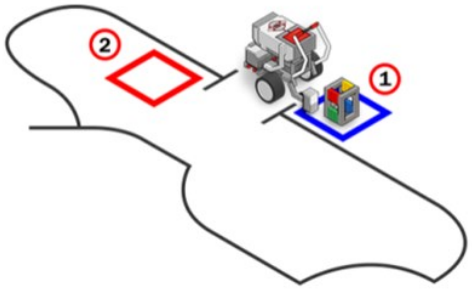
☐ **Show Teacher _____**

☐ **Do: Challenge**

Show Teacher _____

In this challenge, you will program your EV3 robot to grab a cargo crate from the pickup spot, follow the line track and drop the crate off in the drop-off zone.

☐ **Document Mini Challenge & Challenge (include pictures, and detailed descriptions of program function and robot behavior)**



Introduction to Programming
Lego Mindstorm EV3
Decisions: Line Follower 1-2 & C *(found online in Robotics Academy EV3)*

☐ **Watch: Line Follower 1: Introduction to Automated Material Transport System (AMTS)**

What environmental feature will the robot use to help it navigate in this challenge?

◊ A line on the ground
◊ IR beacons on walls
◊ Volume levels of environments
◊ It is facing Northward

What programming concept will be used to create this behavior?

◊ Discrete behaviors
◊ Multitasking
◊ Repeated decisions
◊ Bad decisions

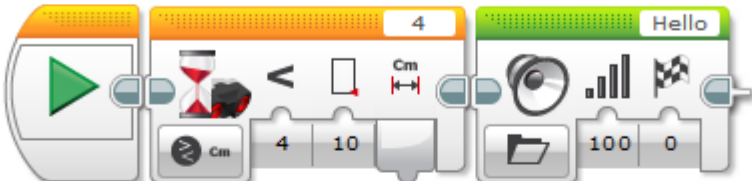
☐ **Do: Robot Configuration (Color Sensor – Facing Down)**

☐ **Watch Line Follower 2: Challenge (Line Tracking)**

If the robot is seeing Black, what can it tell about its position?

◊ It is over the line
◊ It is outside the line to the left
◊ It is outside the line to the right
◊ It is facing Northward

Write a detailed explanation for the following program blocks:



☐ Do: Update Program Documentation (with programs, comments and drawings in new slide(s))

- ☐ Threshold Value 10cm
- ☐ Threshold Value 100cm
- ☐ Sound Sentences
- ☐ Backward Until Far

FINAL CHALLENGE

Follow the directions to complete the ***MAZE CHALLENGE.*** Your program must use the ultrasonic sensor for navigation. When you have completed this challenge, have another team check it off before arranging with the teacher for your demonstration. Be prepared to discuss your pseudocode, your program details (including comments) and how you overcame any difficulties that arose.

Date	Challenge	Sign-off
	Successful Walled Maze Challenge	
	SUCCESSFUL WALLED MAZE CHALLENGE DEMONSTRATION	**(Teacher)**

NOTES:

Criteria	Points Possible	Points Earned
Research: Use the Internet to research the appearance and behaviors of your animal. **Record at least 3 characteristics (appearance or behaviors) and describe how you will use your NXT or EV3 kit and/or program to simulate each characteristic.**	6 (1 pt per characteristic, 1 pt per description)	
Design: Use **only the parts from your NXT or EV3 kit**. **Your design must be your own** (do not use building tutorials found online). **Make a sketch of your robot before you build it, then make a sketch of the final product and list what you changed and what remained the same**. You must include **at least TWO sensors.**	6 (1 pt each for before/ after sketch, 1 pt each for changes/ same, 1 pt per sensor)	
Build: Robot is **completely built by the day of the Zoo Tour.**	3	
Program: Create a program that causes your robot to behave like the animal. You must **use the two sensors** that you included in your design. Your program should **be ready by the day of the Zoo Tour.**	4 (1 pt per sensor used, 2 pts ready on time)	
***Peer Evaluation:** Does your robot look like your animal?	3	
***Peer Evaluation:** Does your robot behave like your animal?	3	
TOTAL POINTS	25	

Design Challenge: Robo-Zoo

For this project, you will research, design, build, and program a robot that looks and behaves like an animal of your choice. Two groups cannot choose the same animal.

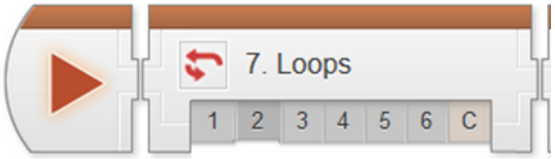
Use the criteria in the table below to help you complete your project by the deadline.

*On the day of the Zoo Tour, your classmates will evaluate your design and program for the "Peer Evaluation" Criteria. For each question, Yes = 3, Sort Of = 2, No = 1. Class scores will be averaged to determine your grade for these portions.

Draw your DESIGN BELOW:

Introduction to Programming
Lego Mindstorm EV3
Decisions: Loops – Move Until Near 1-5 & C

(found online in Robotics Academy EV3)



☐ **Watch: Loops 1: Introduction to Container Handling Robot**

1. Repetitive tasks can be accomplished efficiently by...
◊ Writing very long programs
◊ Writing multiple programs
◊ Repeating behaviors within a program
◊ Manually controlling the robot

☐ **Do: Loops 2: Robot Configuration**

☐ **Watch: Loops 3: Looped Movement**

1. What does the Loop do?
◊ Send the program "flow" back to an earlier point in the program, causing it to repeat some instructions
◊ Choose between two different possible sets of commands to run
◊ Repeat a branching decision quickly, to enable continuous control of the robot
◊ Run all the programs on the robot in a continuous cycle

2. How do you add a Loop to a program?
◊ Click the Loop Block, then drag a box around the commands you want to put inside the Loop
◊ Select the blocks you want to put inside the Loop, then right-click and choose "Place in Loop"
◊ Right-click in an empty area and choose "Make Loop" from the menu that appears
◊ Drag a Loop into the program, then drag commands into it

☐ **Do: Mini Challenge 1: Square Lap 1**

Program a Looped behavior that makes the robot travel around a square box forever.

How many complete loops did you succeed in completing before the robot went off course?

☐ **Watch: Loops 4: Loop with Count Control**

1. A Loop set to Count Mode will send the Program Flow back...

◊ Every time, forever
◊ Only a limited number of times
◊ If the Touch Sensor is not pressed when the Flow reaches the end of the Loop
◊ If there is nothing after the Loop

2. What does it mean for a Loop to be "Conditional" in the EV3 Programming Software?

◊ It only sends the Flow back under certain conditions
◊ The entire Loop can be skipped under certain conditions
◊ The Loop runs faster after it is trained, or "conditioned"
◊ The code runs every time, no matter what

3. What is the "condition" in this Loop based on?

◊ The distance the robot has traveled
◊ The value of the Touch Sensor
◊ The number of times the Loop has sent the Flow back
◊ The number of seconds the Loop has been running

☐ **Do: Try it! 1: Other Counts**

Try changing the number in the Loop's Count setting to 2, then running the program.

What happens?

Engineering Challenge #3 – Sumo-Bots

- Use the Engineering Process to develop your Sumo-Bot.
- The Commercialization step will be the Sumo-Bot competition.
- For each of the other steps of the Engineering Process, document your progress by completing the associated Deliverables and your Engineering Process Log.
- Have another team sign off on your deliverables before submitting them for grading (including all the elements for your Design Review).
- Present your Design Review for your classmates and teacher

Date	Deliverables	Sign-off
	Requirements	
	Research Report	
	Design Concept	
	Project Plan	
	Design Review Elements	
	SUCCESSFUL SUMO-BOT DESIGN REVIEW!!	**(Teacher)**

Task Assignment

Engineering Challenge #3 – Sumo-Bots

Sumo-Bot Match Procedures

This will be a double-elimination competition.

Each match will begin with a one minute “call to start” warning. Additional 30s, 15s, and 5s warning will also be given. Teams unprepared to start on time will forfeit that match.

This event will take place on a ~4’x4’ whiteboard with a black border. The arena will be a white square 36” on a side.

Sumo-bot wrestling will begin with opponents in opposite corners and will continue until either one opponent is completely out of the arena, or one Sumo-bot is disabled.

If, after two minutes, there is no clear winner the competition will be halted. After a second one minute “call to start”, an additional one minute re-match will begin. If there is no clear winner after the second match, both Sumo-bots move to the loser’s bracket.

The winning robot must either push its opponent completely out of the arena, or disable its opponent (render opponent unable to maneuver, while victor Sumo-bot still can). If a robot is not pushed off the mat, but is flipped, the flipped Sumo-bot is considered disabled and loses the match.

Sumo-Bot Specifications

Components: All robots will be constructed only from a single 9797 kit (items as listed on the 9797 placards) and the following additional allowed items:

Large Wheel (2)

Angled Snap Beam (8)

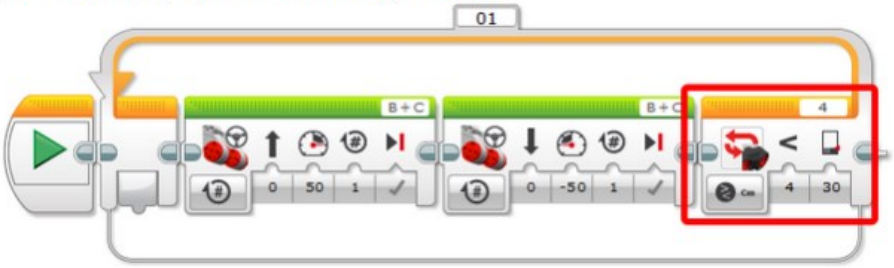
Claw (2)

☐ **Do: Mini Challenge 1: Square Lap 2**

Program your robot to make just one lap around the box.

☐ **Watch: Loops 5: Loop with Sensor Control**

1. When the Loop is set to Ultrasonic Sensor as shown below, what "condition" will cause it to pass Program Flow through (instead of sending it back)?



◊ The robot sees something within 30 centimeters immediately after moving forward
◊ The robot sees something within 30 centimeters immediately after backing up
◊ The robot sees something farther than 30 centimeters away
◊ The program repeats itself 30 times

2. If you want to make a Loop end based on a sensor reading, you should…
◊ Set the Loop's mode to "Sensor"
◊ Set the Loop's Mode to the sensor you want
◊ Select the sensor you want while the program is running
◊ Set the Loop's Mode to Count Sensors

3. When does a Sensor Loop check the sensor?
◊ Continuously while inside the Loop
◊ At the beginning and end of the Loop
◊ At the end of the Loop only
◊ At the end of every block within the Loop

☐ **Do: Try it! 1: Other Sensors**

Set the Loop's mode to Touch Sensor instead of Ultrasonic Sensor. Make sure there is a Touch Sensor on the robot. Run the program and try to get the robot to stop!

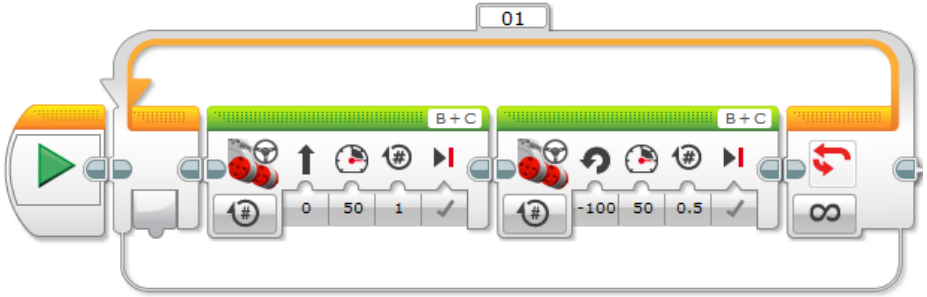
What happens?

☐ **Do: Mini Challenge 1: Square Lap 3**

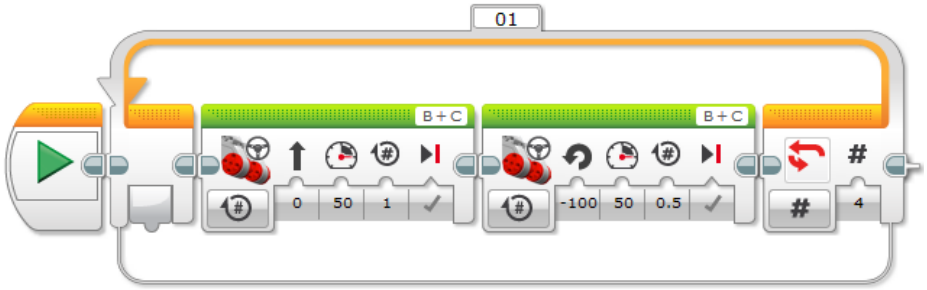
Make your robot run laps around the box until it encounters an obstacle in its way. The obstacle will be placed so that it is visible to the robot immediately after it turns.

☐ **Loops 6: Loop Review**

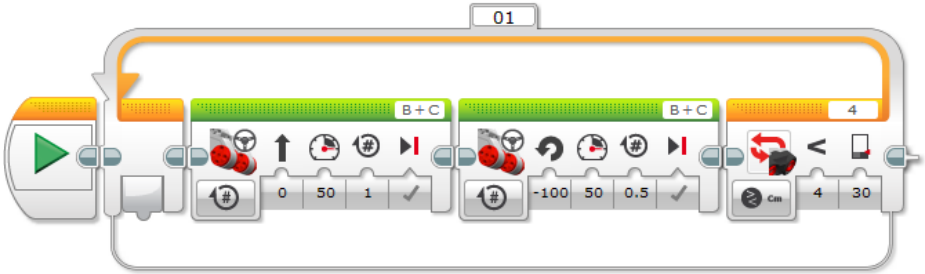
Provide a detailed explanation of how the <u>program runs</u> AND how the <u>robot responds</u>.



Provide details on what the difference is between the following program loop and the previous program loop?



Provide details on what the difference is between the following program loop and the previous program loop?



☐ **Update and Document the following programs. BE sure to include a <u>diagram of the robots movement</u> AND detailed instructions about <u>how each program block functions</u>.**

☐ **Square Lap 1** ☐ **Square Lap 2** ☐ **Square Lap 3**

☐ **Container Challenge**