Acadia Robotics

BottleSumo Time Trial Workshop

EV3 Classroom App

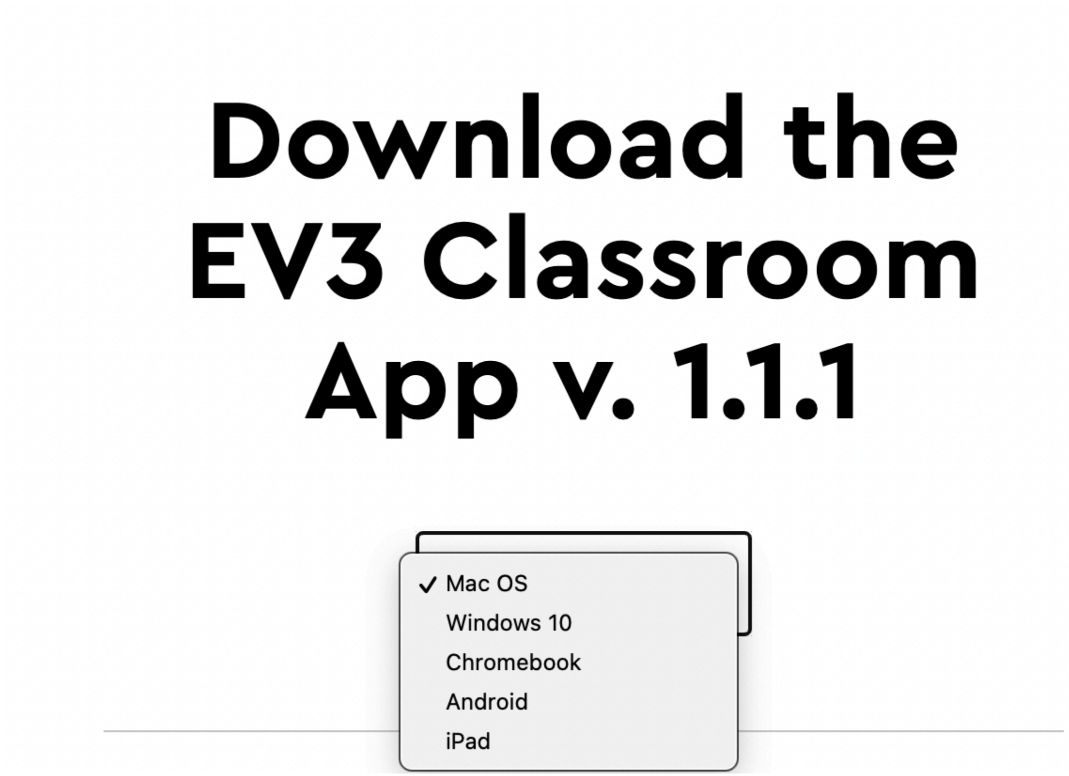
Scratch based programming.

Acadia Robotics May 1-16, 2021

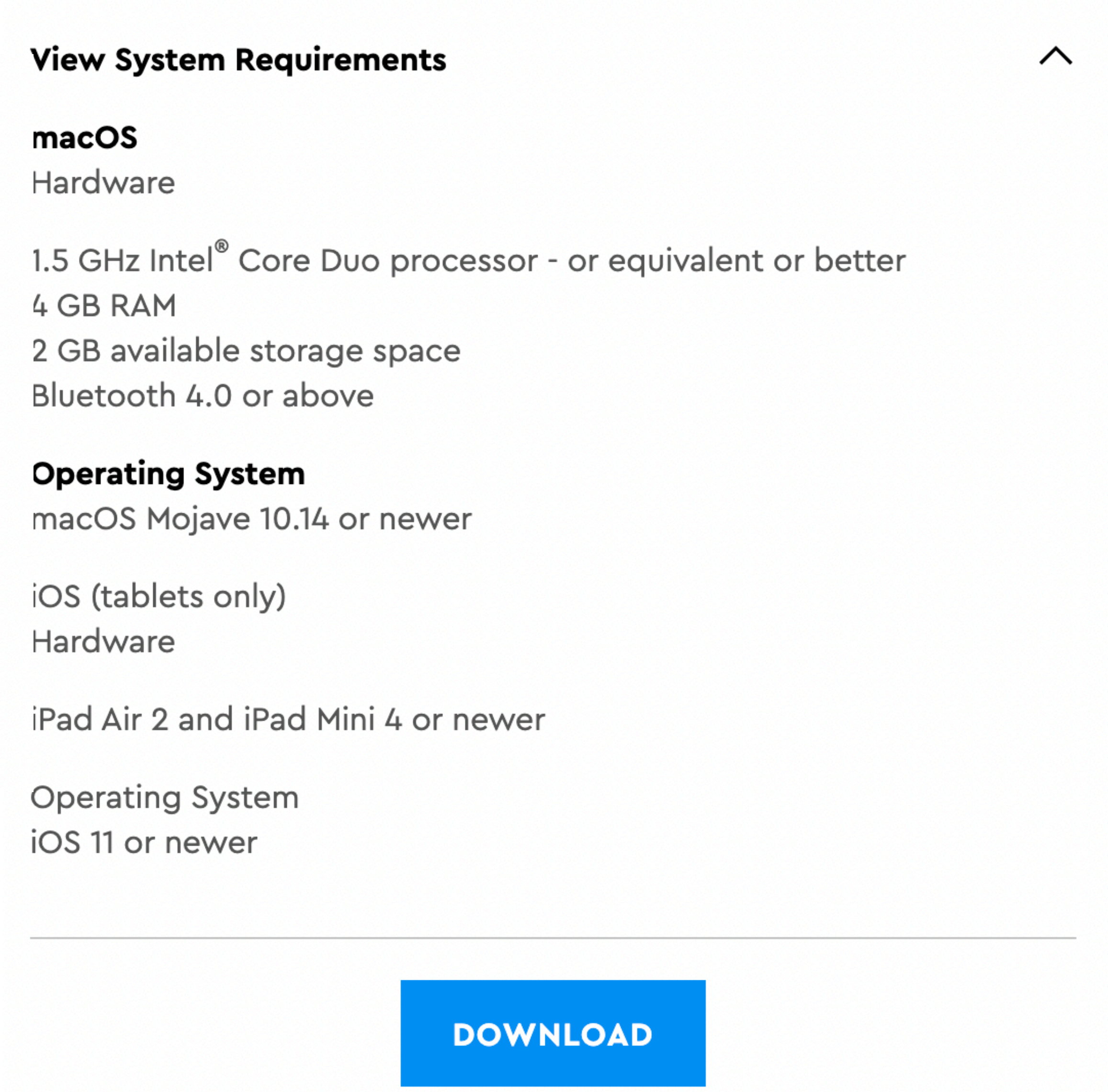
Getting Started

Download the EV3 classroom APP

https://education.lego.com/en-us/downloads/mindstorms-ev3/software



Download the EV3 Classroom App v. 1.1.1



View System Requirements

macOS
Hardware

1.5 GHz Intel® Core Duo processor - or equivalent or better
4 GB RAM
2 GB available storage space
Bluetooth 4.0 or above

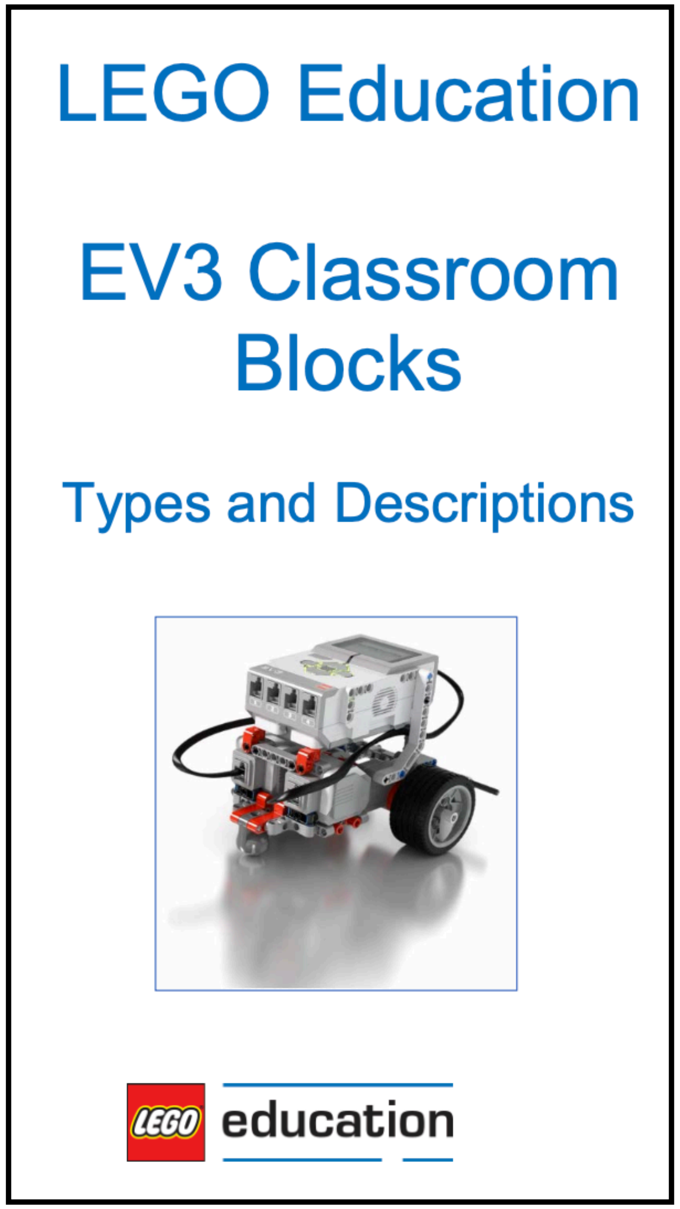
Operating System
macOS Mojave 10.14 or newer

iOS (tablets only)
Hardware

iPad Air 2 and iPad Mini 4 or newer

Operating System
iOS 11 or newer

LEGO Education
EV3 Classroom Blocks Defined



LEGO Education

EV3 Classroom Blocks

Types and Descriptions

education

EV3 Classroom programming Lessons are on our website:
https://robots.acadiau.ca/EV3_timetrial_workshop_classroom_app_scratch.html

LEGO Education created an EV3 Classroom Blocks Types and Descriptions Booklet to use as visual reference.

The booklet can be found on our website near the bottom of our lessons list.

https://robots.acadiau.ca/EV3_timetrial_workshop_classroom_app_scratch.html

Direct link to the PDF file:

https://robots.acadiau.ca/files/sites/robots/Uploaded%20Pictures%20and%20Materials/2021-2022%20Files/May%202021%20BottleSumo%20time%20trial%20event/EV3%20Classroom%20Coding%20Blocks%20Visual%20Overview.pdf

Workshop to build programming skills

BottleSumo Time Trial event

1. Consider robot design/configuration
2. Programming
3. Robot TASKS
    a. Find the edge of the table
        - Stay on the table
        - Find the edge with two sensors

    b. Finding the bottle

    c. Push the bottle off table

    d. Robot must stay on the table.

Robot Design

Rules on the robot size can be found in the BottleSumo Time Trial rules handout.
https://robots.acadiau.ca/Science-odyssey-may-2021-BottleSumo-event.html

- One colour/light sensor - two is recommended for a better performing robot
- Your light/colour sensor(s) should be IN FRONT of your wheels. *See diagram*.
- Your team may choose any robot design they want as long as the team follows the robot rules provided for the event.
- For this tutorial the motors and sensor are assumed to be in the pictured configuration.

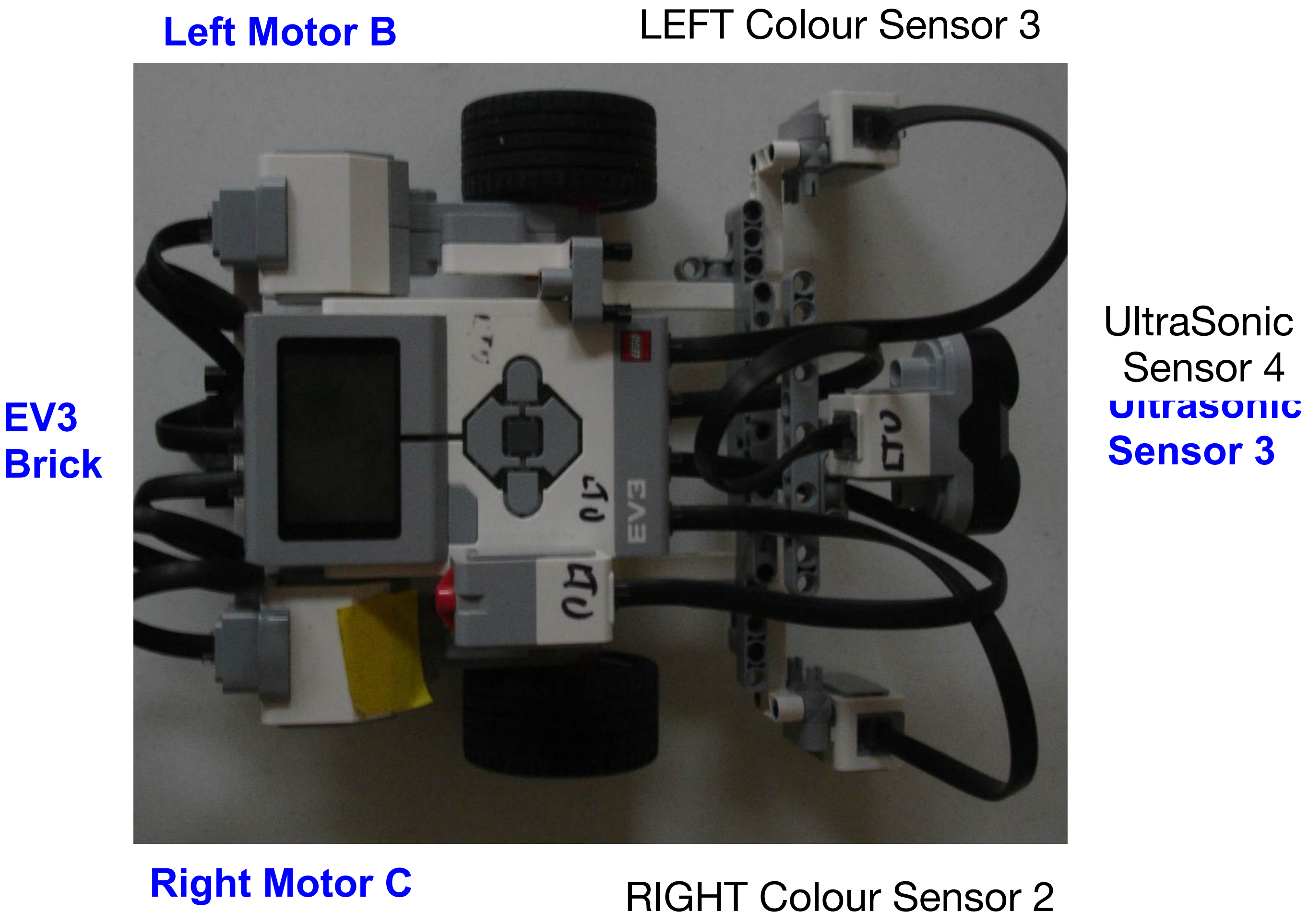


Photo courtesy Robofest and edited for our workshop.

If your robot is using different ports for your motors or sensors, please adjust your programming to reflect the ports you are using.

Motor and Sensor Ports

Reminder to adjust programming based on your own robot configuration.

Our robot is set up with sensors in these ports.

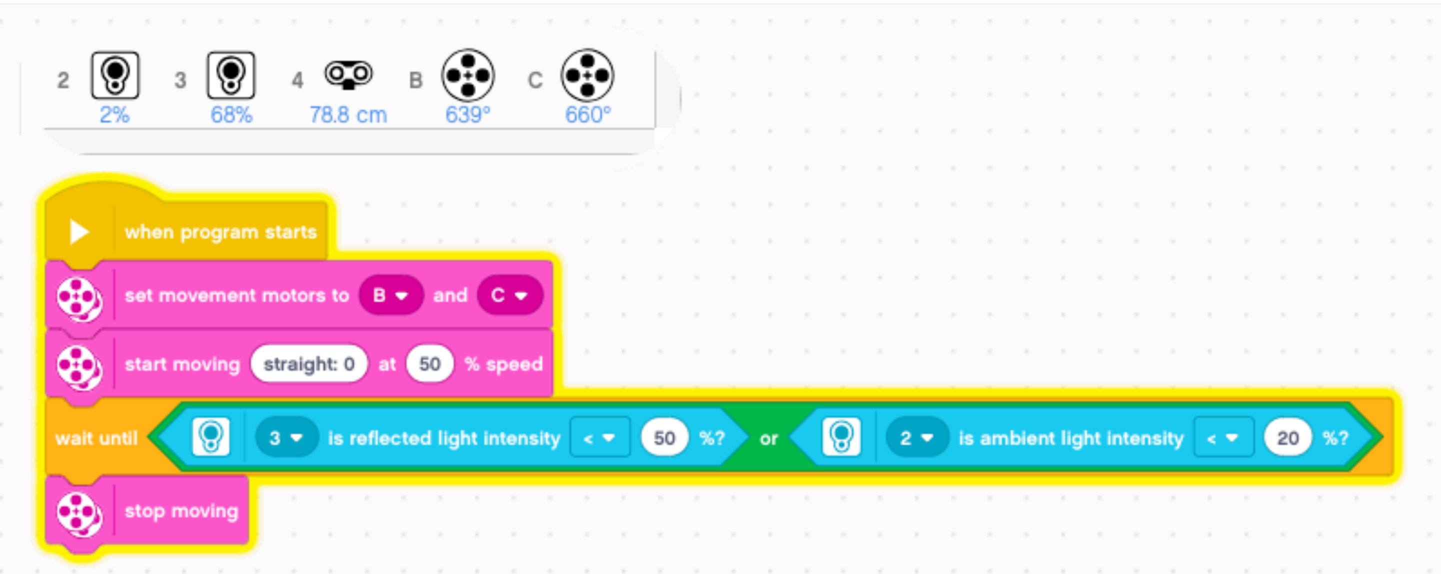
- Left Motor connects to port B
- Right Motor connects to port C

- Left Light/colour sensor connects to port 3
- Right Light/colour sensor connects to port 2
- Ultrasonic sensor connects to port 4

How to use your sensor(s) readings

While your robot is connected to your device, you have a live look at what your robot's sensors 'see'.

REMEMBER your robot configuration:
Port 2 - Right colour sensor
Port 3 Left colour sensor
Port 4 - Ultrasonic Sensor
Port B - Left motor
Port C - Right Motor



2 2% 3 68% 4 78.8 cm B 639° C 660°

```
when program starts
set movement motors to B and C
start moving straight: 0 at 50 % speed
wait until 3 is reflected light intensity < 50 %? or 2 is ambient light intensity < 20 %?
stop moving
```

This program is the "robot finds the edge" example.

In this example the robot has its **RIGHT** colour sensor **off** the table and the **LEFT** colour sensor **ON** the table.

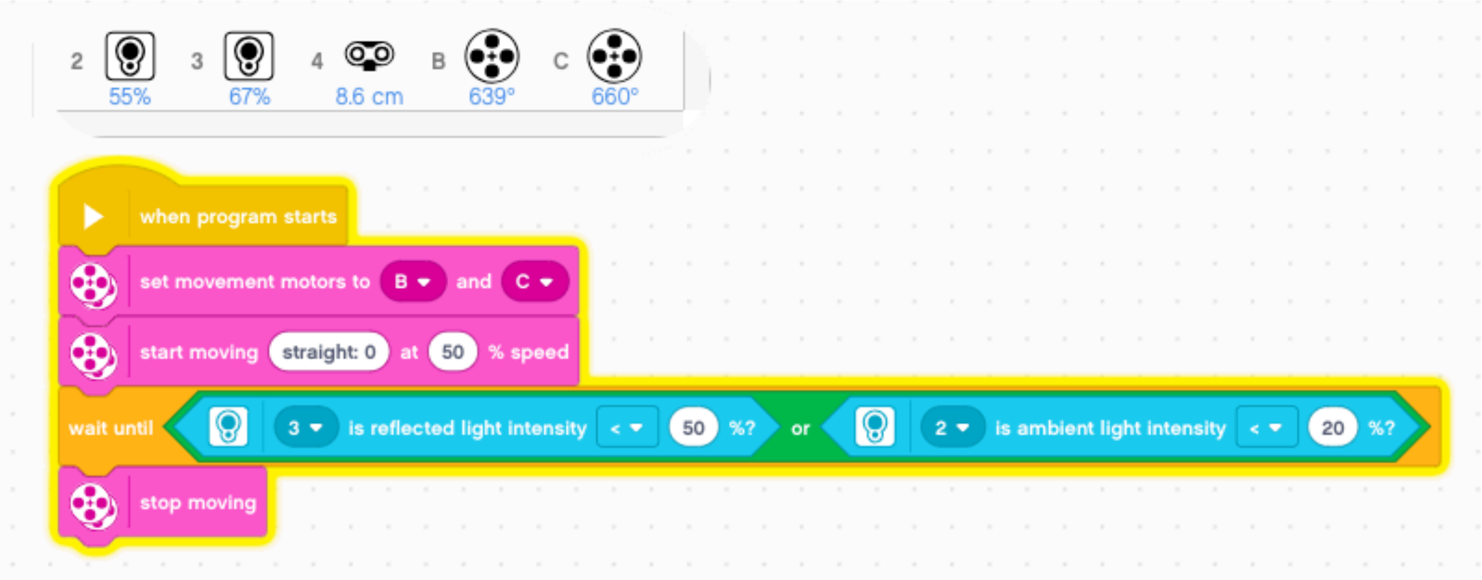
PORT 2 colour sensor reads 2% reflected light (this sensor is OFF the table)
PORT 3 colour sensor reads 68% reflected light (this sensor is ON the table)
PORT 4 ultrasonic sensor 'sees' something 78.8 cm away

IMPORTANT:

Depending on the conditions where you are using your robot, your robot sensor values may be different, you will need to adjust your programming to your robot's readings.

How to use your sensor(s) readings - Part 2

While your robot is connected to your device, you have a live look at what your robot's sensors 'see'.



```
2 55%   3 67%   4 8.6 cm   B 639°   C 660°

when program starts
set movement motors to B and C
start moving straight: 0 at 50 % speed
wait until 3 is reflected light intensity < 50 %? or 2 is ambient light intensity < 20 %?
stop moving
```

REMEMBER your robot configuration:
Port 2 - Right colour sensor
Port 3 Left colour sensor
Port 4 - Ultrasonic Sensor
Port B - Left motor
Port C - Right Motor

This program is the "robot finds the edge" example.

In this example the robot has its BOTH colour sensor are **ON** the table.

PORT 2 colour sensor reads 55% reflected light (this sensor is ON the table)
PORT 3 colour sensor reads 67% reflected light (this sensor is ON the table)
PORT 4 ultrasonic sensor 'sees' something 8.6 cm away

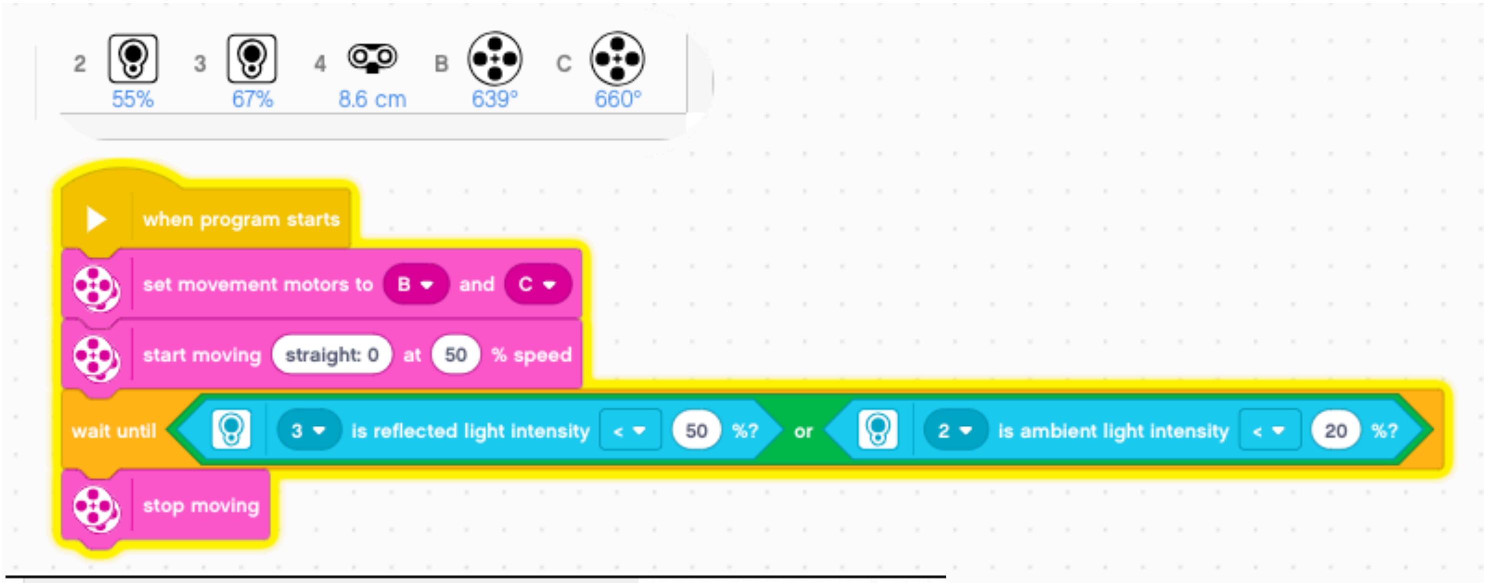
IMPORTANT:

Your robot COLOUR sensors may not show the same reflected light %, you will need to find the THRESHOLD of the two values when programming your robot. You will want your robot to react to it's environment.

How to use your sensor(s) readings - Part 3

Thresholds

While your robot is connected to your device, you have a live look at what your robot's sensors 'see'.

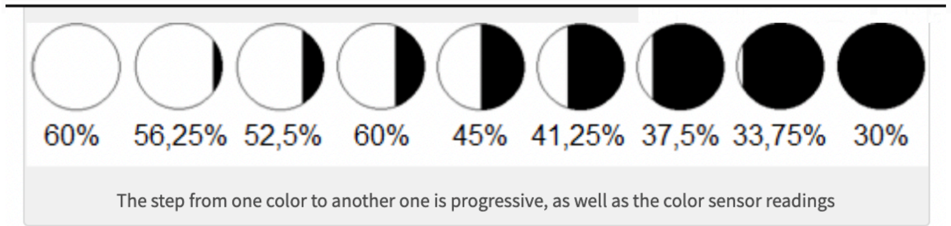


2 55%
3 67%
4 8.6 cm
B 639°
C 660°

```
when program starts
set movement motors to B and C
start moving straight: 0 at 50 % speed
wait until 3 is reflected light intensity < 50 %? or 2 is ambient light intensity < 20 %?
stop moving
```

REMEMBER your robot configuration:
Port 2 - Right colour sensor
Port 3 Left colour sensor
Port 4 - Ultrasonic Sensor
Port B - Left motor
Port C - Right Motor

This program is the "robot finds the edge" example.

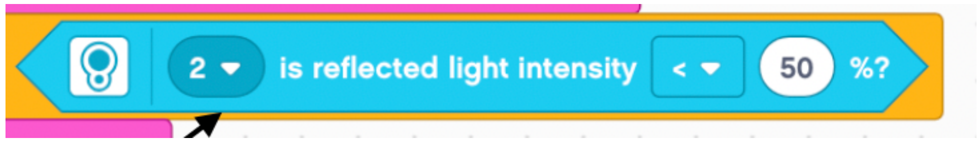


The step from one color to another one is progressive, as well as the color sensor readings

Your robot's colour sensor will read the reflected light as it moved from light (on table) to dark (off table).

Think about it:

In the programming example above; if I want my robot to stop when either colour sensor is **OFF** the table I need to set my threshold to LESS than 50%.



```
2 is reflected light intensity < 50 %?
```

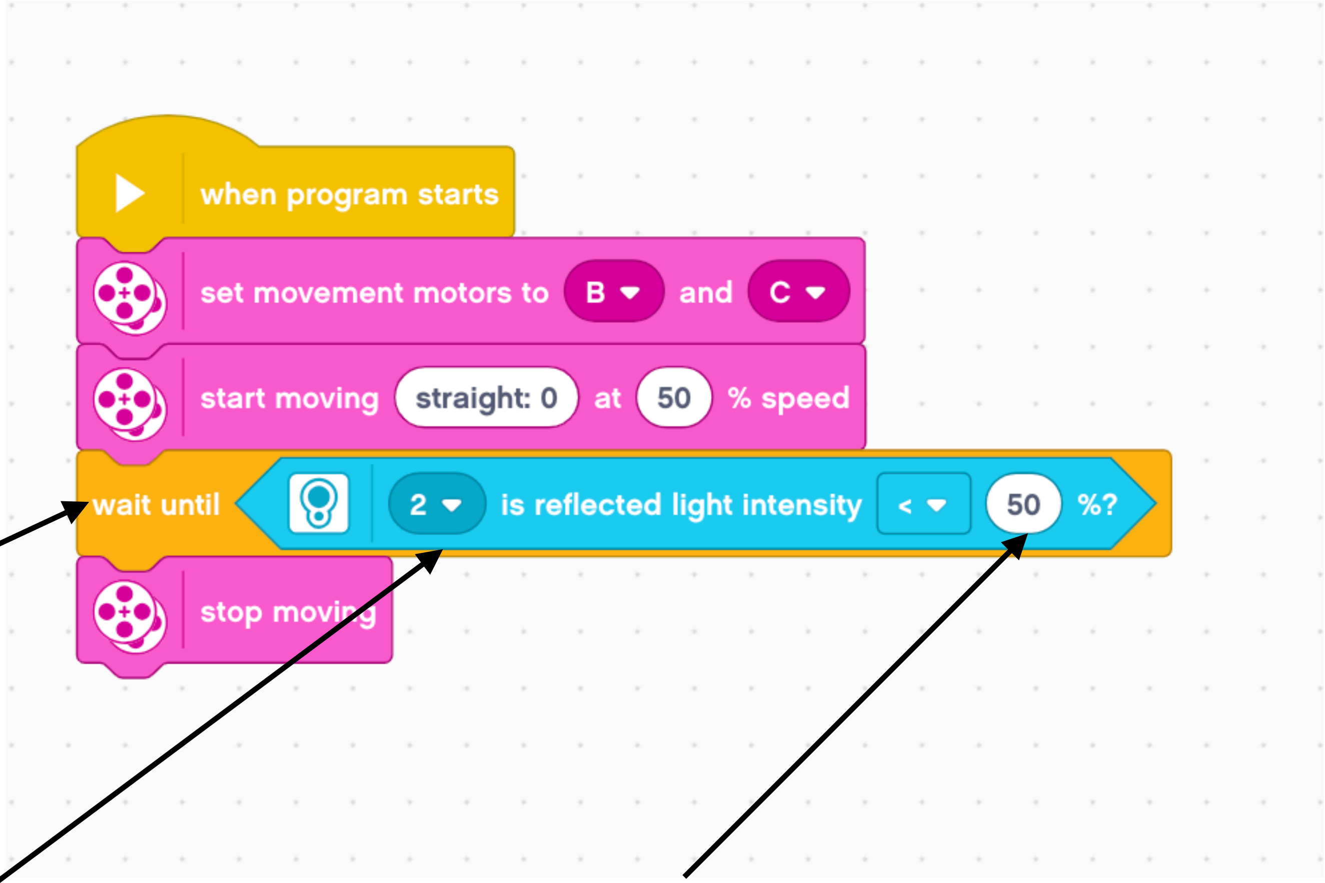
BottleSumo Time Trial
Robot Objective

Your autonomous robot needs to do to the following to complete the challenge.

1. Robot does not fall off the table.
2. Robot needs to find the bottle(s).
3. Robot needs to push the bottle(s) off the table.
4. Robot must remain on the table.
5. Time how long it takes for your robot to clear the table of bottles.

Find the table edge with One Colour Sensor

- Robot needs to stay on the table for the duration of the time trial
- Program the robot to find the edge of the table and STOP
- Use a WAIT block
- In this example when the right colour sensor (port 2) finds the edge of the table, the robot will stop.



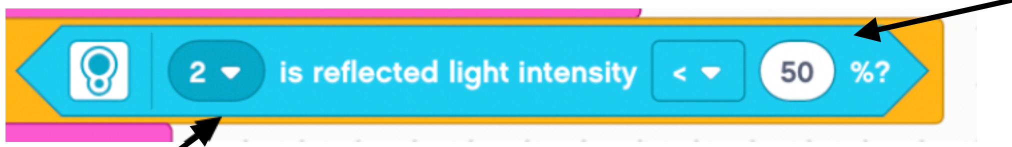
```
when program starts
set movement motors to B and C
start moving straight: 0 at 50 % speed
wait until 2 is reflected light intensity < 50 %?
stop moving
```

Your team needs to find out what value to put here.

Think about it…

Issues your robot will have when trying to find the edge of the table.

- If the robot stops too soon or too late, try adjusting your light sensor threshold.



2 is reflected light intensity < 50 %?

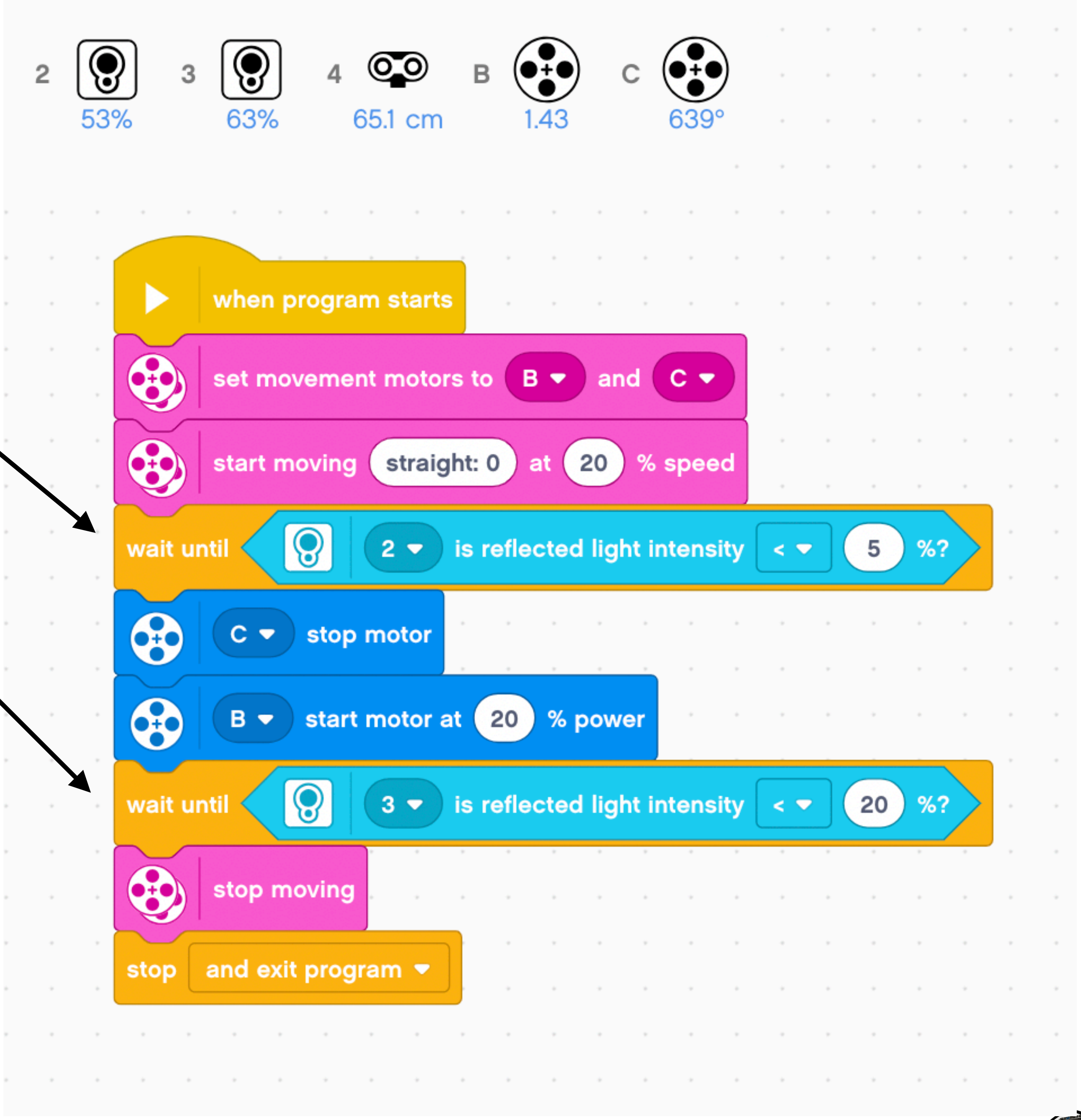
- The Robot is only are looking for the edge of the table with one light sensor in **port 2.**

  Question: What will happen to your robot if the other colour sensor in **port 3** goes off the table first?

Square robot to edge of table -Part 1

Two colour sensor method

- You robot will stop when the first colour sensor finds the edge of the table.
- The motor on the same side of the robot will stop and the other motor will keep moving until the second colour sensor finds the the edge of the table and stops.
- The **<u>wait until</u> block** is used twice in this example.



```
when program starts
set movement motors to B and C
start moving straight: 0 at 20 % speed
wait until 2 is reflected light intensity < 5 %?
C stop motor
B start motor at 20 % power
wait until 3 is reflected light intensity < 20 %?
stop moving
stop and exit program
```

QUESTION:

What will happen if your robot finds the edge with the colour sensor in port 1?

Square robot to edge of table - Part 2

New skill: Using two colour sensors

QUESTION:
What will happen if your robot finds the edge with the colour sensor in port 1?

- **Answer:** your robot will stop and not be squared to to edge of table.
- **Solution**: program your robot to align if EITHER sensor detects edge of table first.
- Introduce “**or**” statements -Two conditions
- Introduce “**if - else**” statements into your programming.
    - **IF** right colour sensor sees table edge - do this action
        - **Else** - do another action

Square robot to edge of table - Part 3

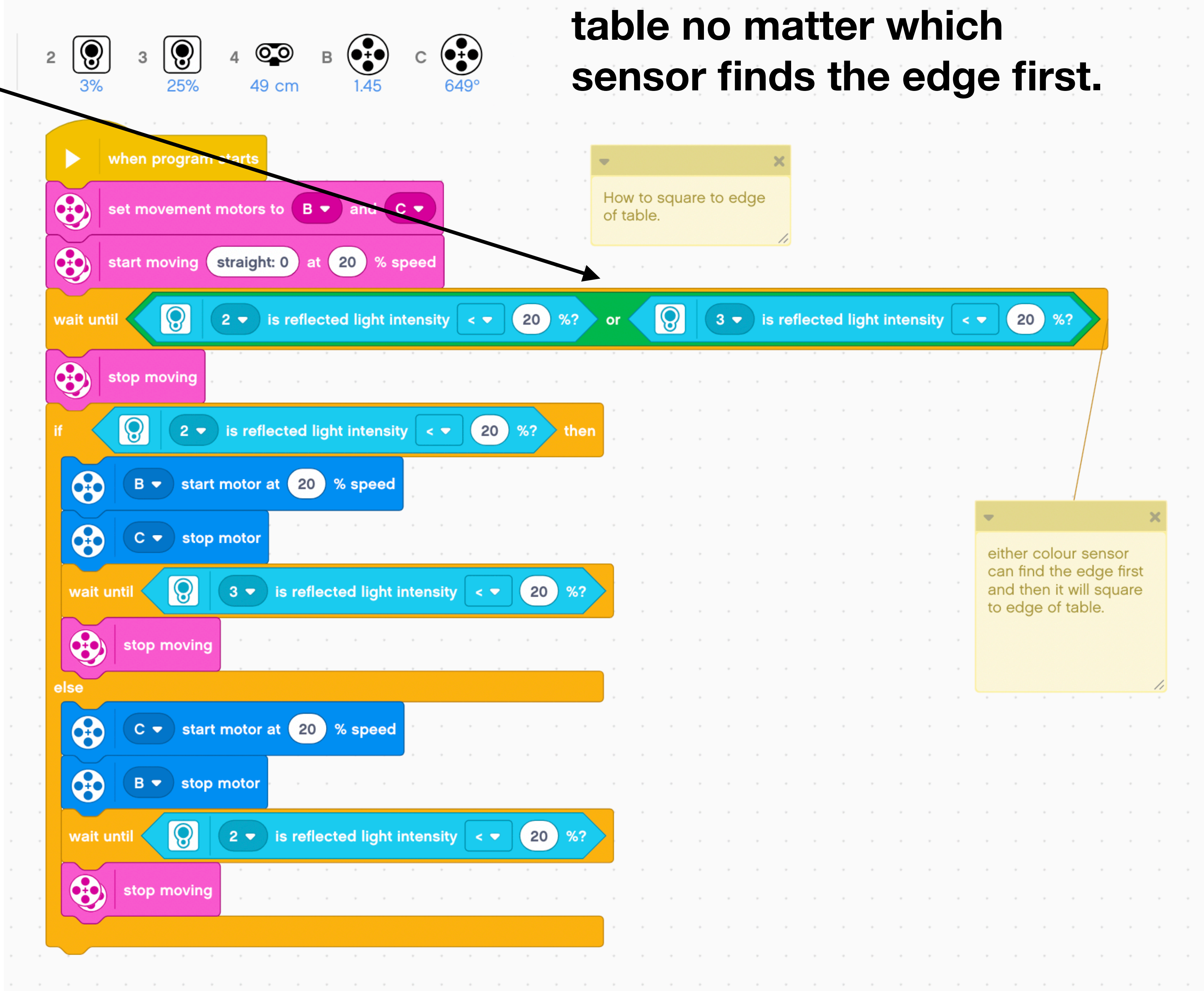
OR statement
(Green OPERATIONS block)

Robot will stop moving when either colour sensors finds the edge of the table.

IF ELSE condition
(Orange Variables block)

- You want your robot to align to the table edge but you won't know which colour sensor finds the edge first.
- Your robot needs to consider the options it has based on which colour sensor finds the edge first.
- What happens if your left colour sensor finds the edge first?
  - Your robot needs to stop the left motor and move the right motor UNTIL the right colour sensor also finds the edge.
- What happens if your right colour sensor finds the edge first?
  - Your robot needs to stop the right motor and move the left motor UNTIL the left colour sensor also finds the edge.

Robot aligns to edge of table no matter which sensor finds the edge first.



```
when program starts
set movement motors to B and C
start moving straight: 0 at 20 % speed
wait until 2 is reflected light intensity < 20 %? or 3 is reflected light intensity < 20 %?
stop moving
if 2 is reflected light intensity < 20 %? then
    B start motor at 20 % speed
    C stop motor
    wait until 3 is reflected light intensity < 20 %?
    stop moving
else
    C start motor at 20 % speed
    B stop motor
    wait until 2 is reflected light intensity < 20 %?
    stop moving
```

How to square to edge of table.

either colour sensor can find the edge first and then it will square to edge of table.

Can you see how using these new statements "OR" & "IF ELSE" allows your robot to be more precise?

Find and Avoid

Edge of the table

Question:

What do you want your robot to do if it finds the edge of the table but it hasn’t found the bottle?

Answer:

The robot needs to keep searching for the bottle.

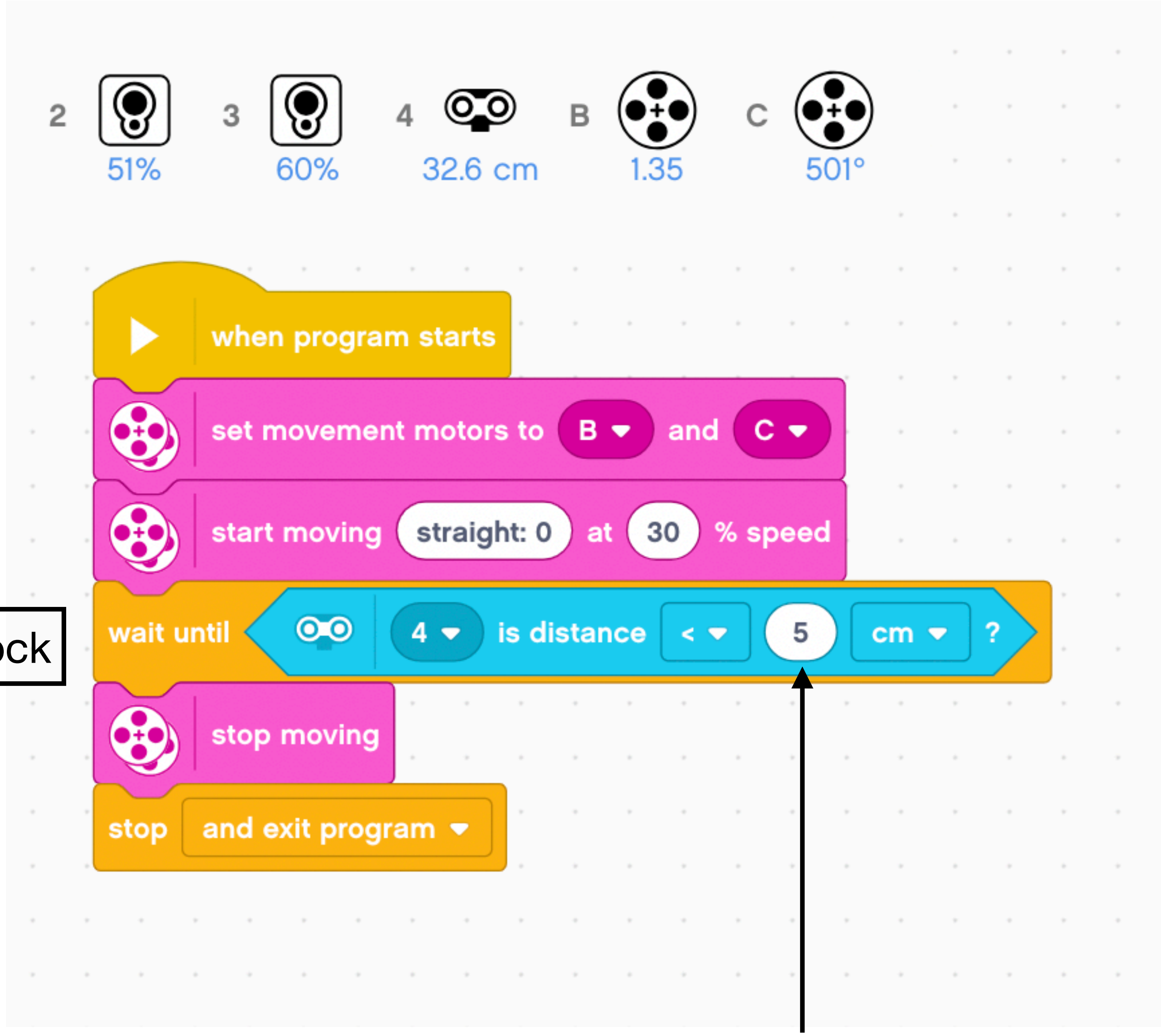
- Back up the robot and turn to keep looking for the bottle.

TEST IT: Use the previous lesson to program the back up and turn AFTER your robot stops at the edge.

Ultrasonic Sensor - Find the Bottle

Stop the robot when it is close to bottle

Robot will stop when the ULTRASONIC sensor (port 4) is less than 5cm away from the Bottle



2 51% 3 60% 4 32.6 cm B 1.35 C 501°

```
when program starts
set movement motors to B and C
start moving straight: 0 at 30 % speed
wait until 4 is distance < 5 cm ?
stop moving
stop and exit program
```

Ultrasonic “Wait Until” Block

Set the distance you want your robot to stop.

TEST IT: put the bottle in the path of the robot, at least 10 cm away, or more.

Think about it:
This example only works if the bottle is in FRONT of the ultrasonic sensor.

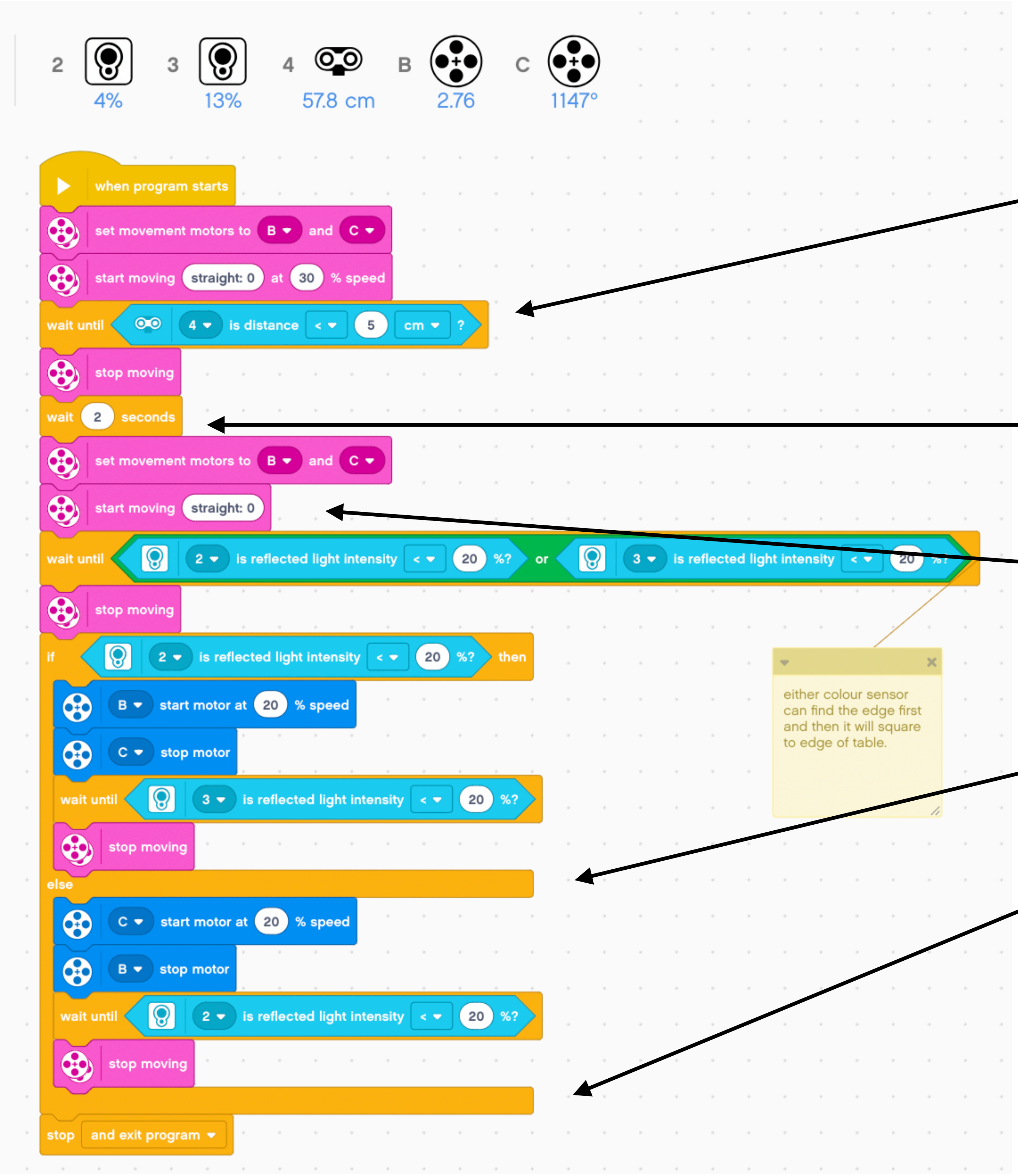
Push bottle off table

Think about it…

You want to push the bottle off the table but your robot has to stay on the table.

- You know how to find the bottle (ultrasonic)
- You know how to stay on the table (colour sensor(s)
- How will you combine these two tasks to push the bottle off the table AND make sure your robot stays on the table?

Push only the Bottle Off Table



```
2 4%   3 13%   4 57.8 cm   B 2.76   C 1147°

when program starts
set movement motors to B and C
start moving straight: 0 at 30 % speed
wait until 4 is distance < 5 cm ?
stop moving
wait 2 seconds
set movement motors to B and C
start moving straight: 0
wait until 2 is reflected light intensity < 20 %? or 3 is reflected light intensity < 20 %?
stop moving
if 2 is reflected light intensity < 20 %? then
    B start motor at 20 % speed
    C stop motor
    wait until 3 is reflected light intensity < 20 %?
    stop moving
else
    C start motor at 20 % speed
    B stop motor
    wait until 2 is reflected light intensity < 20 %?
    stop moving
stop and exit program
```

either colour sensor can find the edge first and then it will square to edge of table.

Follow the code.

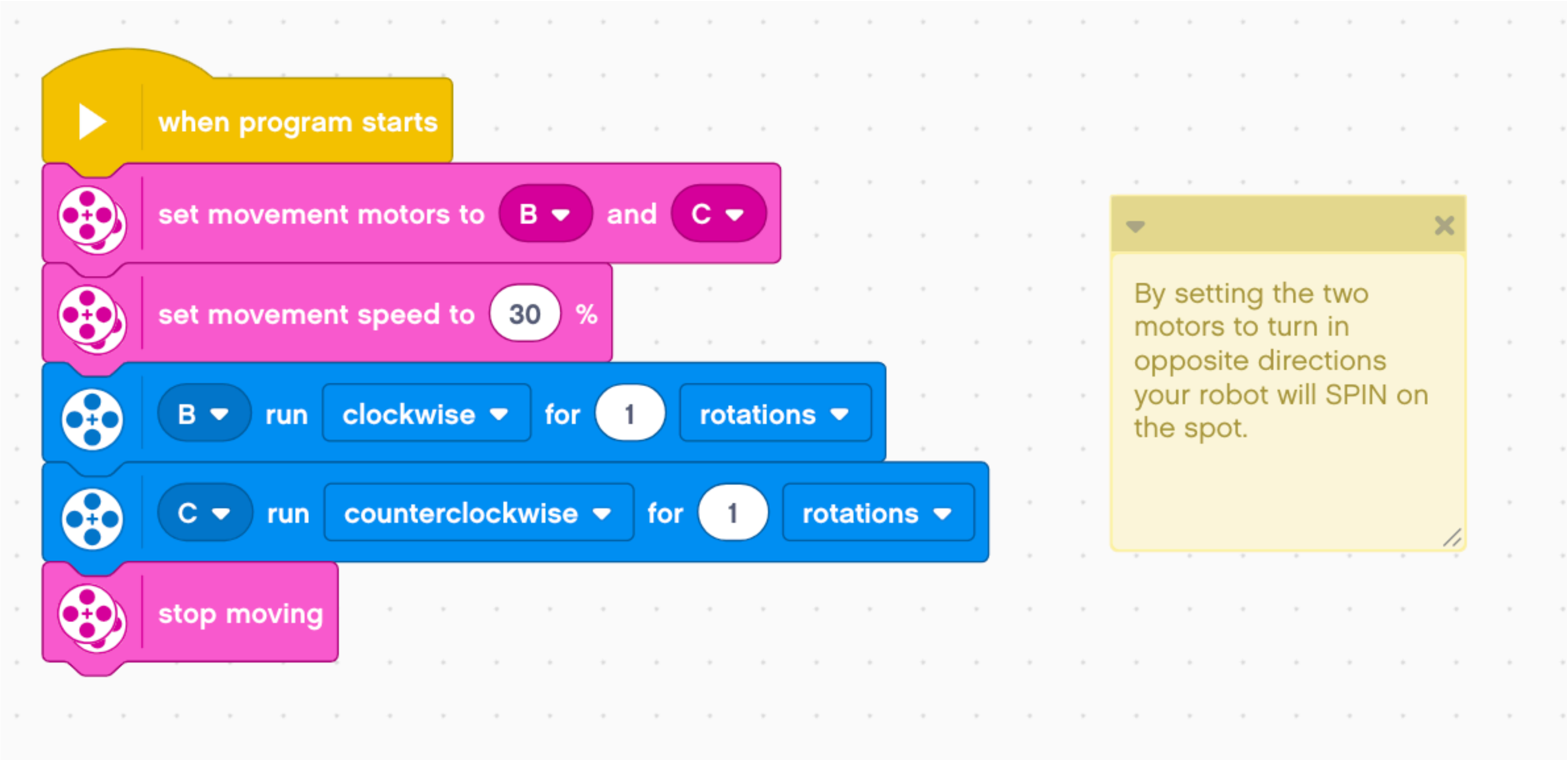
1. Robot drives until it finds the bottle with ultrasonic sensor.
2. Robot pauses for 2 seconds - this is to show you that the ultrasonic sensor ‘sees’ the bottle.
3. Robot starts moving to push robot off table.
4. Robot uses colour sensor program you learned to stay on the table.
5. Robot squares to the edge of table and stops.

TURNS
Find the bottle

Your robot will need to search for the bottles.

SPIN the robot

- Your robot needs to search for the bottle so it can be moved off the table.



```
when program starts
set movement motors to B and C
set movement speed to 30 %
B run clockwise for 1 rotations
C run counterclockwise for 1 rotations
stop moving
```

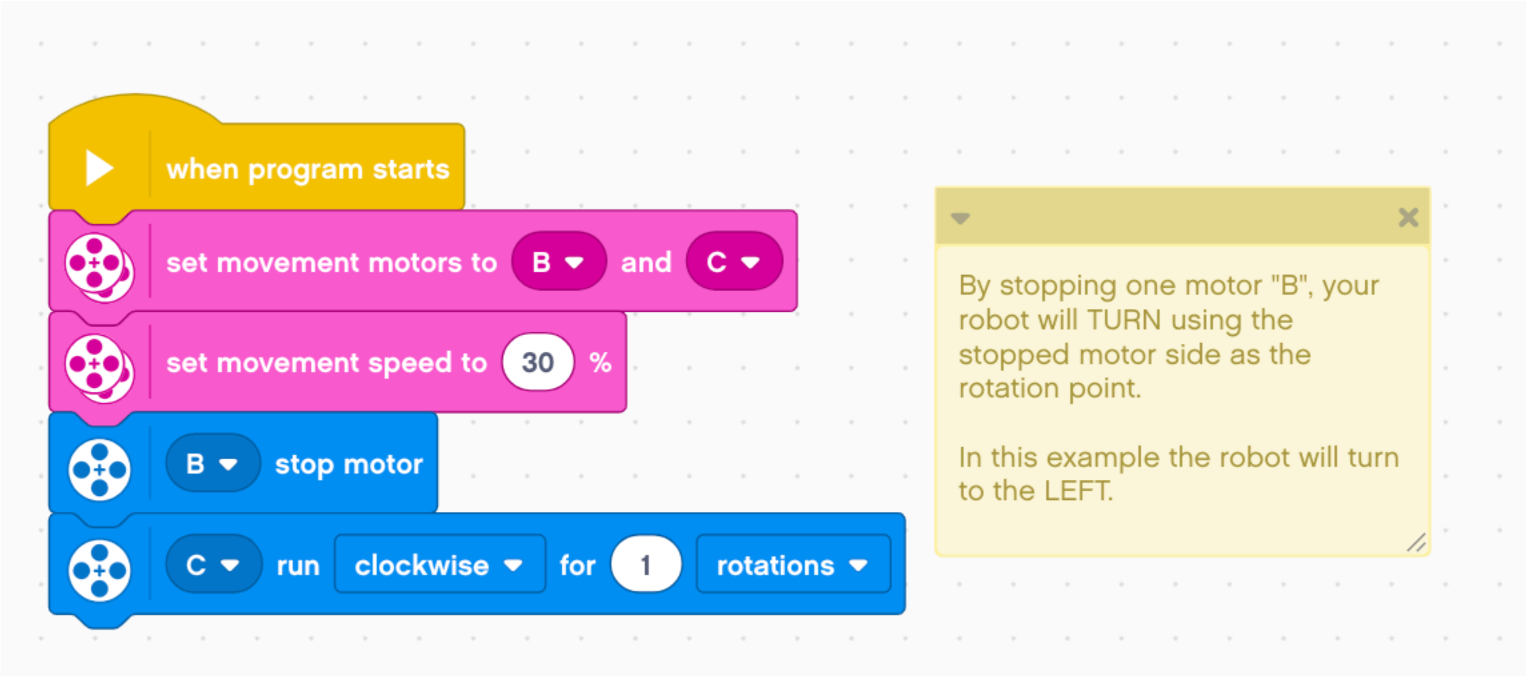
By setting the two motors to turn in opposite directions your robot will SPIN on the spot.

TURNS
Find the bottle

Your robot will need to search for the bottles.

TURN the robot

- Your robot will stop one motor and turn using the motor that is moving.



```
when program starts
set movement motors to B and C
set movement speed to 30 %
B stop motor
C run clockwise for 1 rotations
```

By stopping one motor "B", your robot will TURN using the stopped motor side as the rotation point.

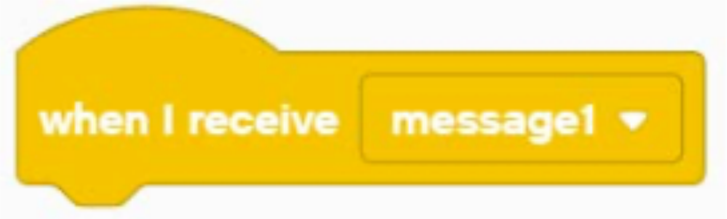
In this example the robot will turn to the LEFT.

TEST IT: Experiment with the other motor

Broadcast

EVENT BLOCKS Defined

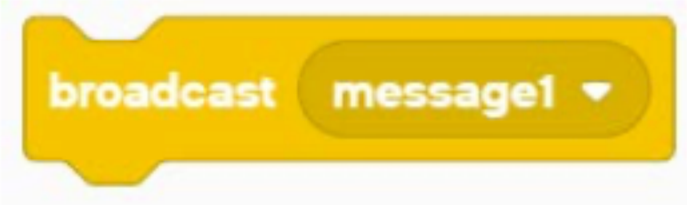
When I Receive Message BLOCK



```
when I receive message1 ▾
```

Runs the blocks attached to it when a specific message is broadcasted by either the Broadcast Messsage or the Broadcast Message and Wait Block.

Broadcast Message BLOCK



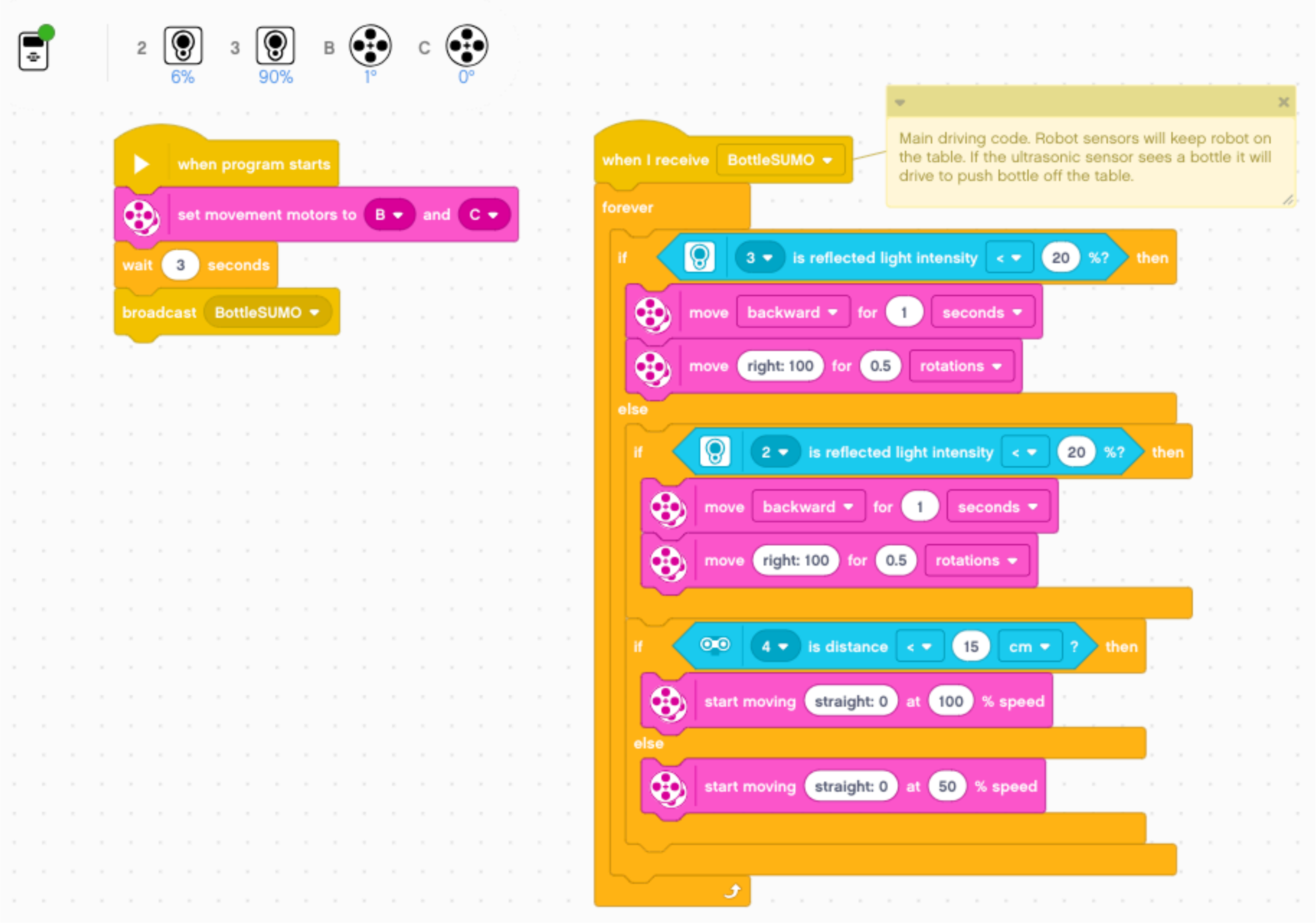
```
broadcast message1 ▾
```

Broadcasts the specified message. All of the WHEN I RECEIVE MESSAGE BLOCKS that have been set to the specified message will activate.

This Broadcast Block sends the specified message and immediately proceeds to the next block.

Simple BottleSumo Code
Using Action Blocks

See if you can follow the coding blocks to understand how this simple BottleSumo program works.



```
when program starts
set movement motors to B and C
wait 3 seconds
broadcast BottleSUMO
```

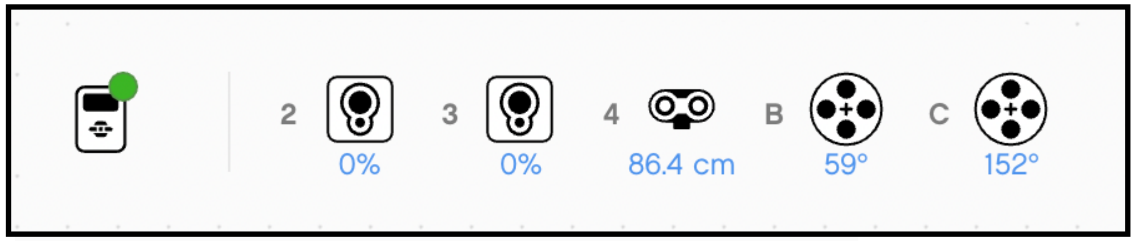
```
when I receive BottleSUMO
forever
    if 3 is reflected light intensity < 20 %? then
        move backward for 1 seconds
        move right: 100 for 0.5 rotations
    else
        if 2 is reflected light intensity < 20 %? then
            move backward for 1 seconds
            move right: 100 for 0.5 rotations
        if 4 is distance < 15 cm ? then
            start moving straight: 0 at 100 % speed
        else
            start moving straight: 0 at 50 % speed
```

Main driving code. Robot sensors will keep robot on the table. If the ultrasonic sensor sees a bottle it will drive to push bottle off the table.

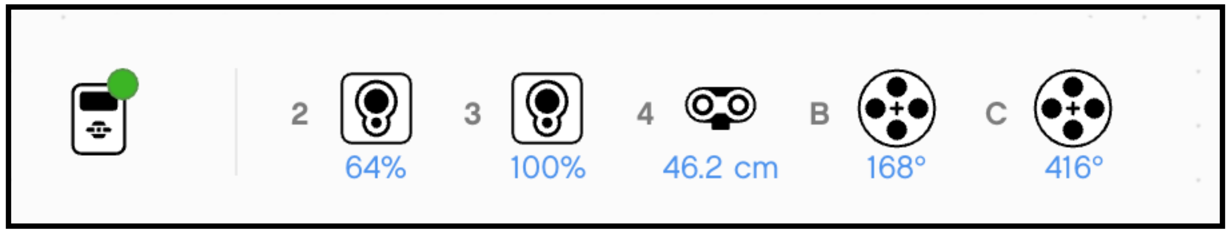
Simple BottleSUMO

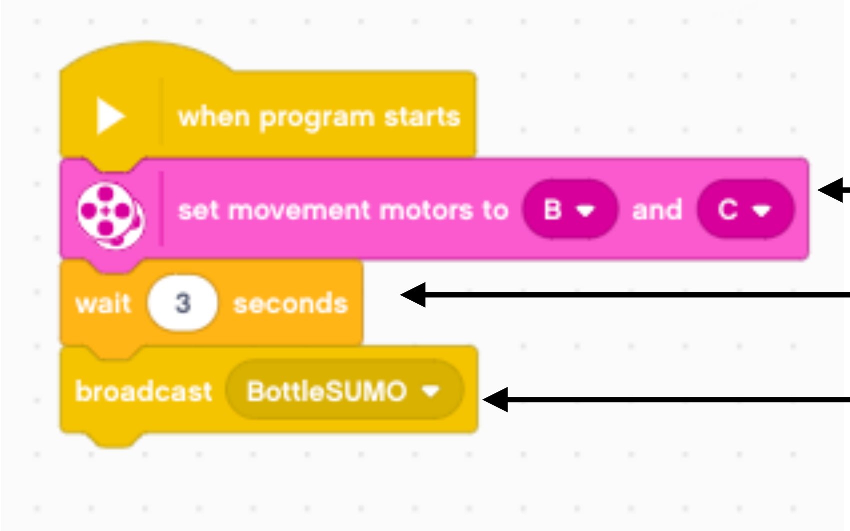
Using Event Blocks

Sensor reading with both colour sensors OFF the table.


2 0%   3 0%   4 86.4 cm   B 59°   C 152°

Sensor reading with both colour sensors ON the table.


2 64%   3 100%   4 46.2 cm   B 168°   C 416°


```
when program starts
set movement motors to B and C
wait 3 seconds
broadcast BottleSUMO
```

This program sets the motors to B & C

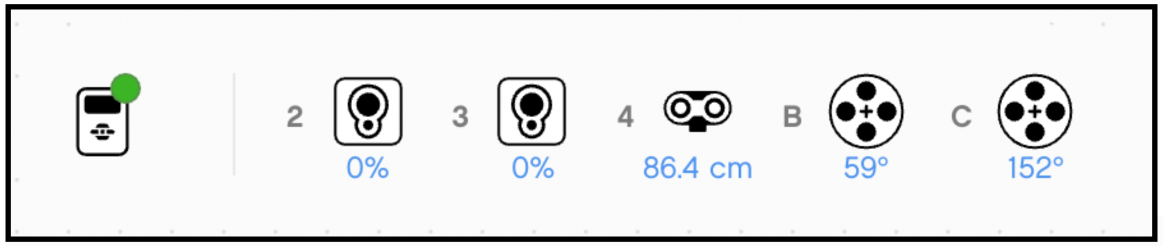
Waits 3 seconds.

Then moved to the Broadcast Message block, which we defined as BottleSUMO.

The next slide will show how we DEFINED the broadcast message block to run the BottleSUMO program.

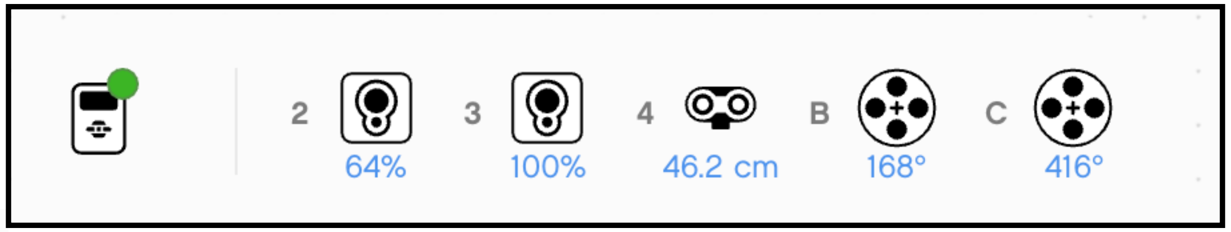
Simple BottleSUMO - Using Event Blocks

Sensor reading with both colour sensors OFF the table.

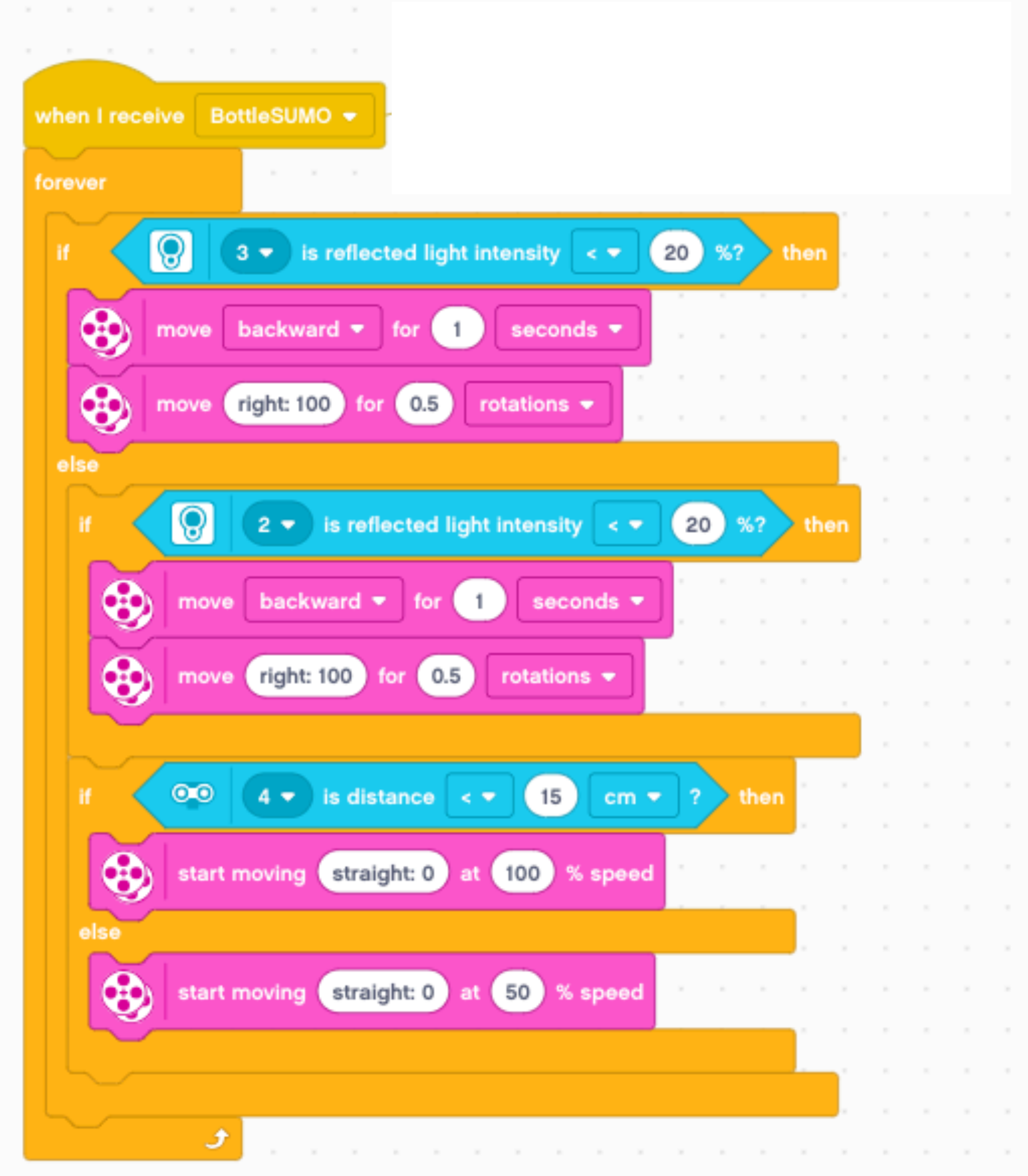


2 0% | 3 0% | 4 86.4 cm | B 59° | C 152°

Sensor reading with both colour sensors ON the table.



2 64% | 3 100% | 4 46.2 cm | B 168° | C 416°



```
when I receive BottleSUMO
forever
  if 3 is reflected light intensity < 20 %? then
    move backward for 1 seconds
    move right: 100 for 0.5 rotations
  else
    if 2 is reflected light intensity < 20 %? then
      move backward for 1 seconds
      move right: 100 for 0.5 rotations
    if 4 is distance < 15 cm ? then
      start moving straight: 0 at 100 % speed
    else
      start moving straight: 0 at 50 % speed
```

Think about it.

Your BottleSUMO robot needs to:

a. Stay on the table

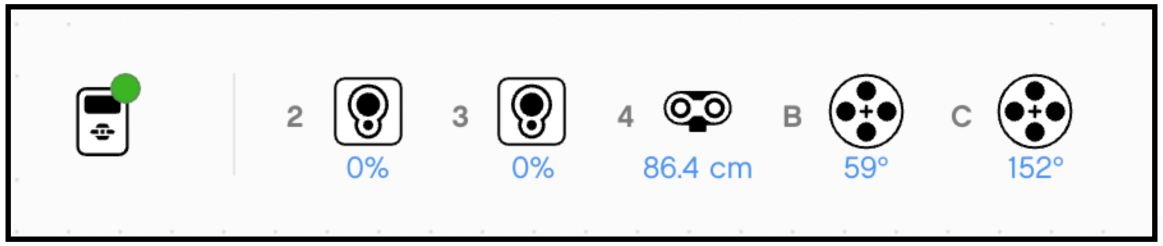
b. Finding the bottle

c. Push the bottle off table

d. Robot must stay on the table, after the bottle(s) are removed.

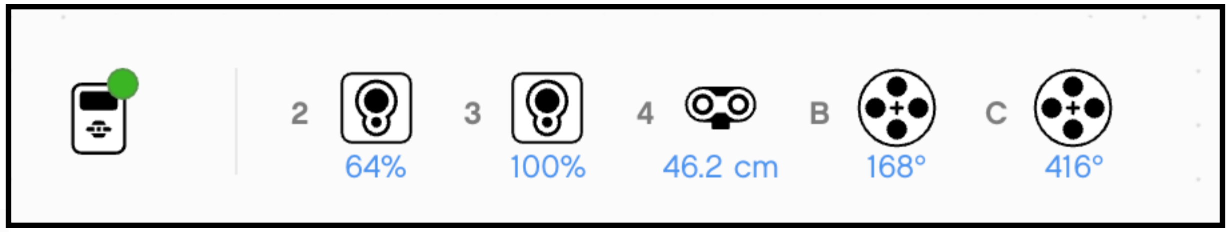
Simple BottleSUMO - Using Event Blocks

Sensor reading with both colour sensors OFF the table.

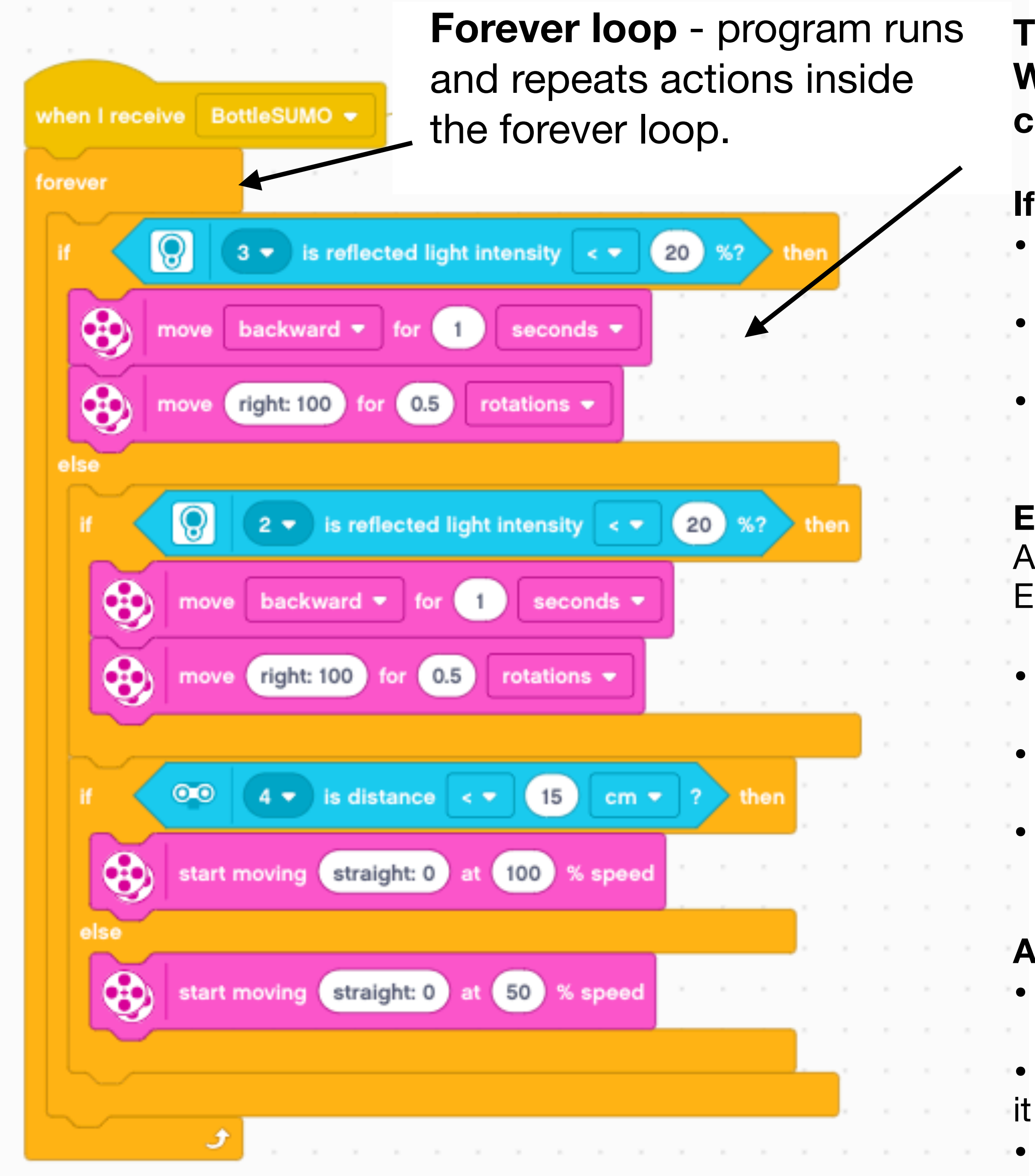


2	3	4	B	C
0%	0%	86.4 cm	59°	152°

Sensor reading with both colour sensors ON the table.



2	3	4	B	C
64%	100%	46.2 cm	168°	416°



```
when I receive BottleSUMO
forever
    if 3 is reflected light intensity < 20 %? then
        move backward for 1 seconds
        move right: 100 for 0.5 rotations
    else
        if 2 is reflected light intensity < 20 %? then
            move backward for 1 seconds
            move right: 100 for 0.5 rotations
        if 4 is distance < 15 cm ? then
            start moving straight: 0 at 100 % speed
        else
            start moving straight: 0 at 50 % speed
```

Forever loop - program runs and repeats actions inside the forever loop.

The robot looks for the bottle with the Ultrasonic sensor WHILE checking to see if it's on or off the table using the colour sensors.

If:

- FIRST program check colour sensor 2 (left side) to see if it's off the table.
- If it is we want the robot to move away from the edge of the table.
- Robot backs up for 1 second and then moves right for 0.5 rotations.

Else:

Another If statement is added under the Else portion of the first If/ Else statement.

- program check colour sensor 3 (right side) to see if it's off the table.
- If it is we want the robot to move away from the edge of the table.
- Robot backs up for 1 second and then moves right for 0.5 rotations.

Add an If/ELSE below the new If statement.

- program uses the ultrasonic sensor (port 4) to see if the bottle is within 15 cm of the robot's sensor.
- **IF -** we want the robot to move towards the bottle to push it off the table.
- **ELSE** - If their is NO bottle

Simple BottleSUMO

Finding a better solution

Think of all the programming lessons you learned in this workshop.

The simple BottleSUMO program you built ‘works’.

The current program is randomly moving around the table. Each action is based on how the sensors react.

Can you think of anyway to improve how your robot **searchers** for the bottle(s)?
Can you think of a way to ensure the bottle is pushed off the table?

We won’t give any more solutions to the BottleSUMO workshop.

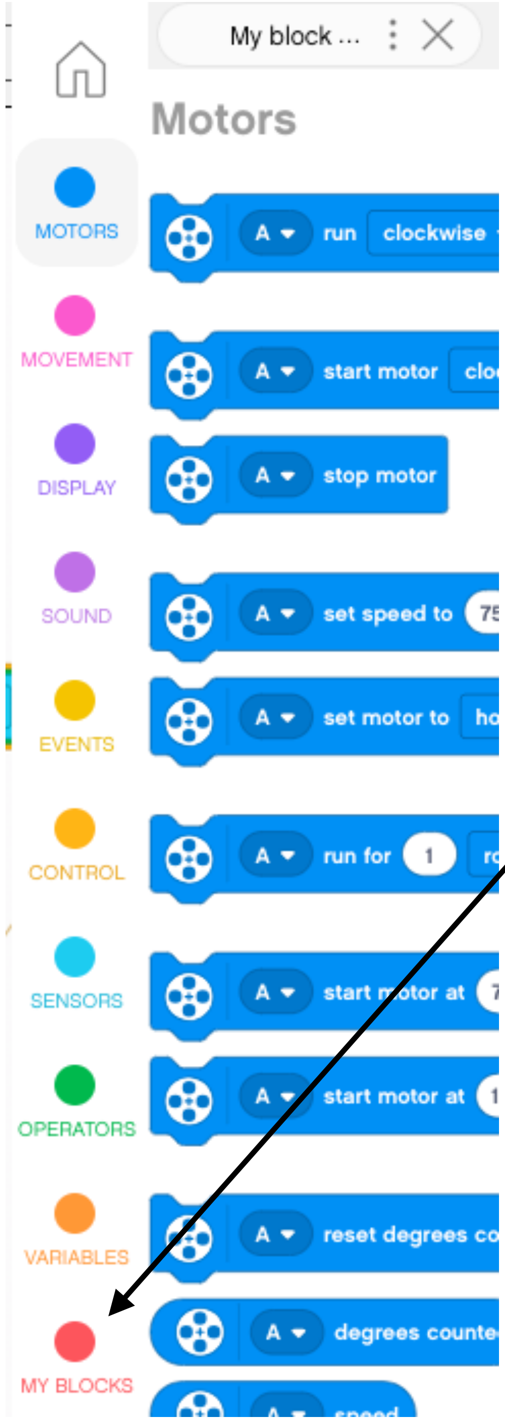
There is more than one way to program your robot to find the bottles.

Good Luck

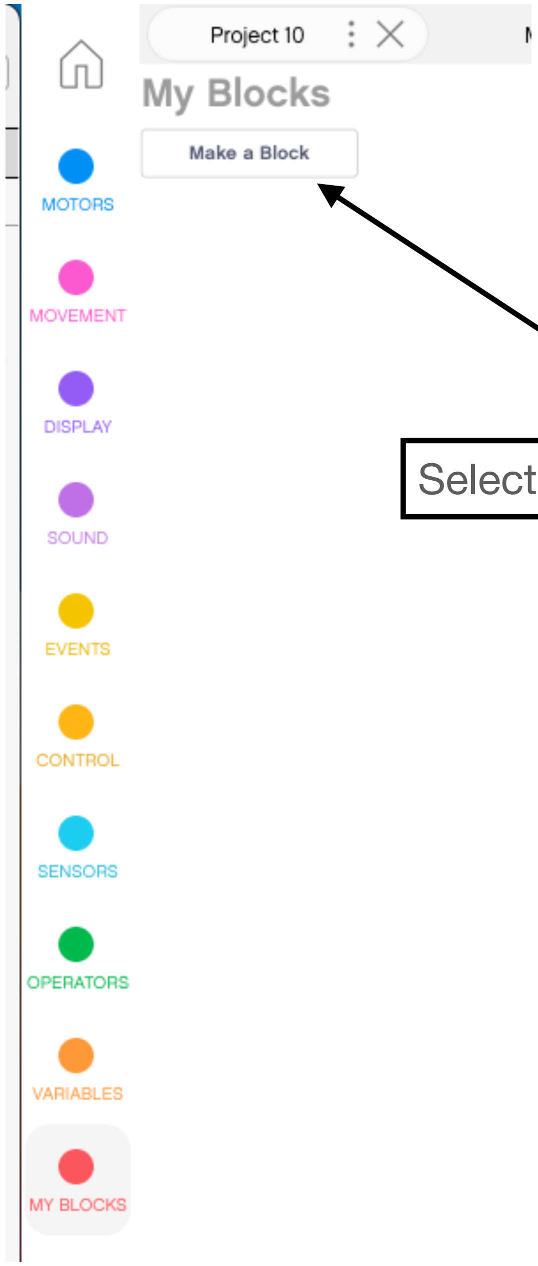
Optional
Programming Lessons

MY BLOCKS

How to create shortcuts for code you need to use often.



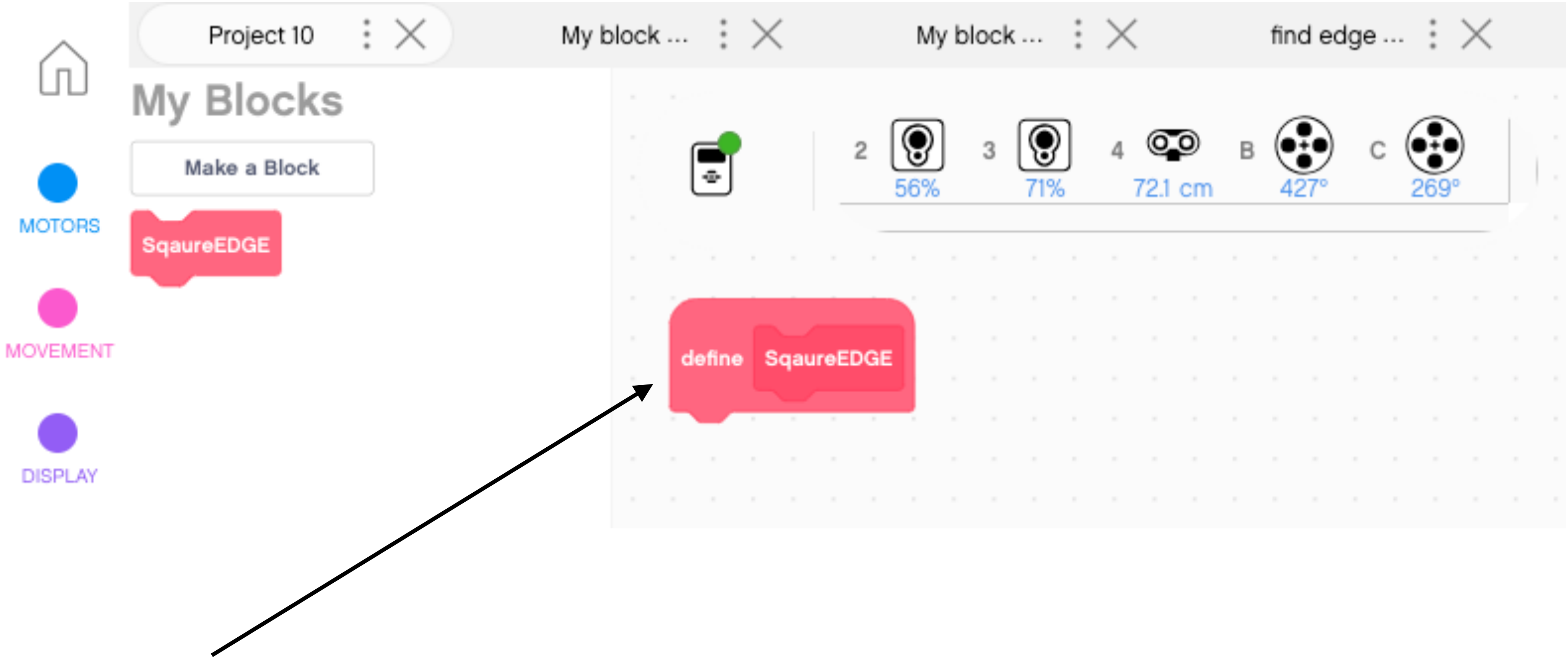
Select **MY BLOCKS**
From menu



Select **Make a BLOCK**

MY BLOCKS

How to create shortcuts for code you need to use often.



DEFINE BLOCK:

This blocks allows you to create your own block.
The MY BLOCK is a group of blocks that's attached to the DEFINE BLOCK

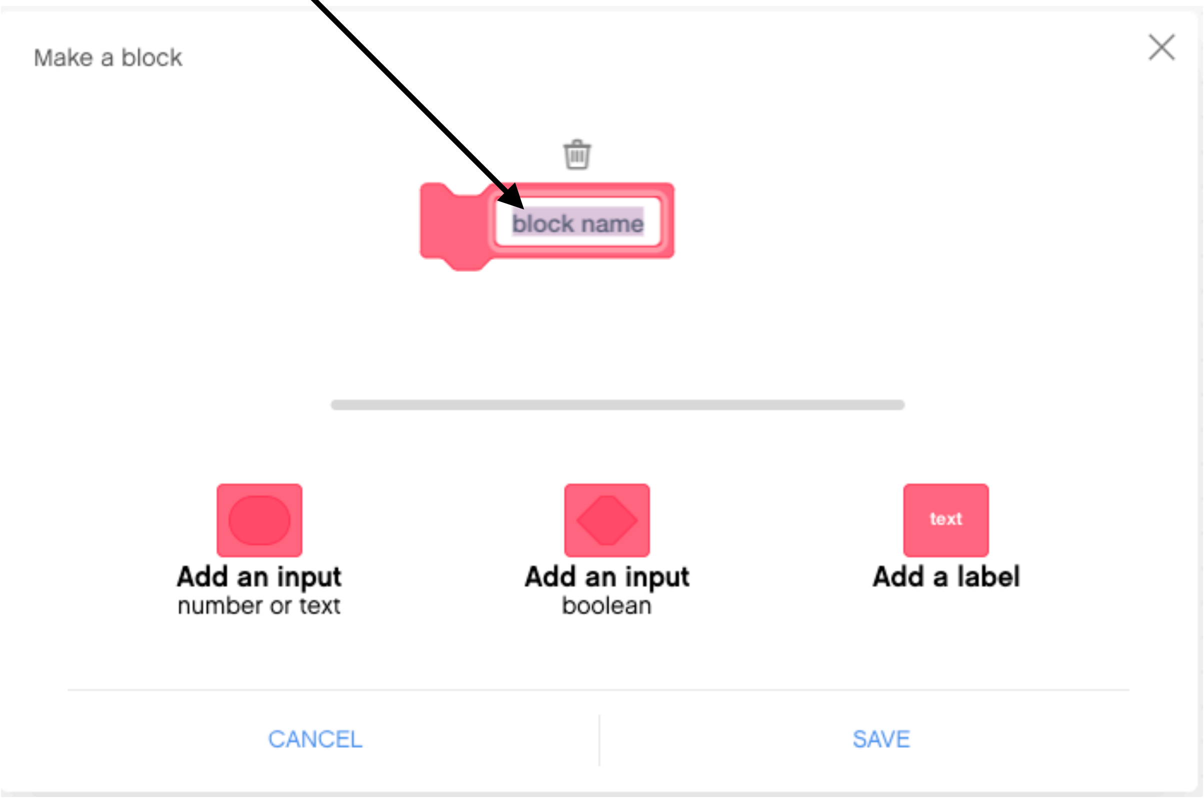
After creating your MY block you now need to DEFINE it.

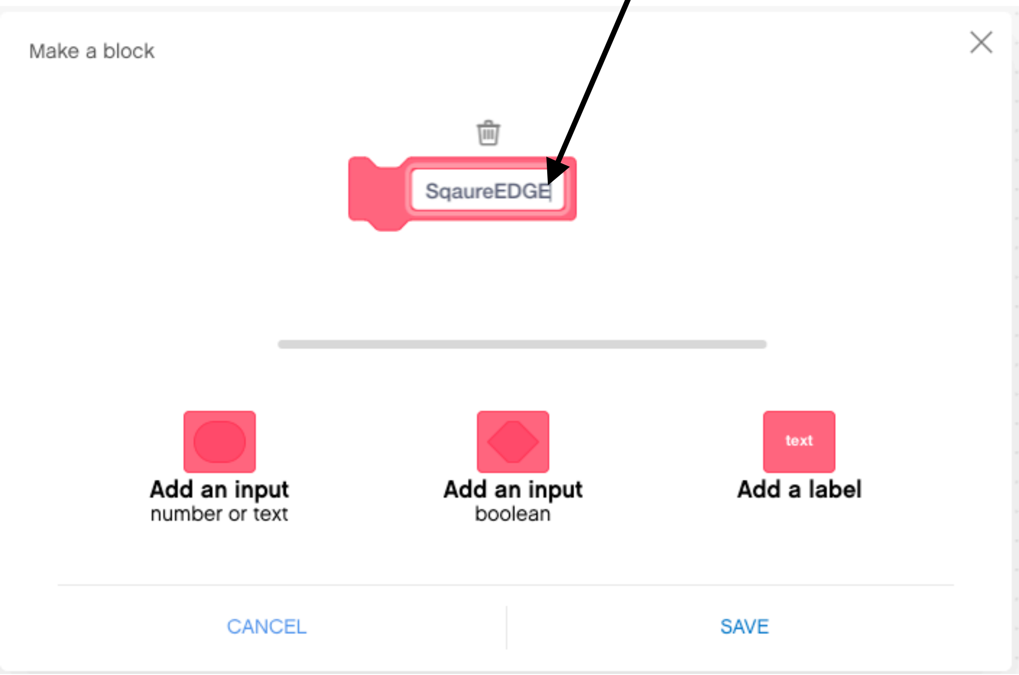
MY BLOCKS

How to create shortcuts for code you need to use often.

NAME your My Block so it describes the action of the My block.

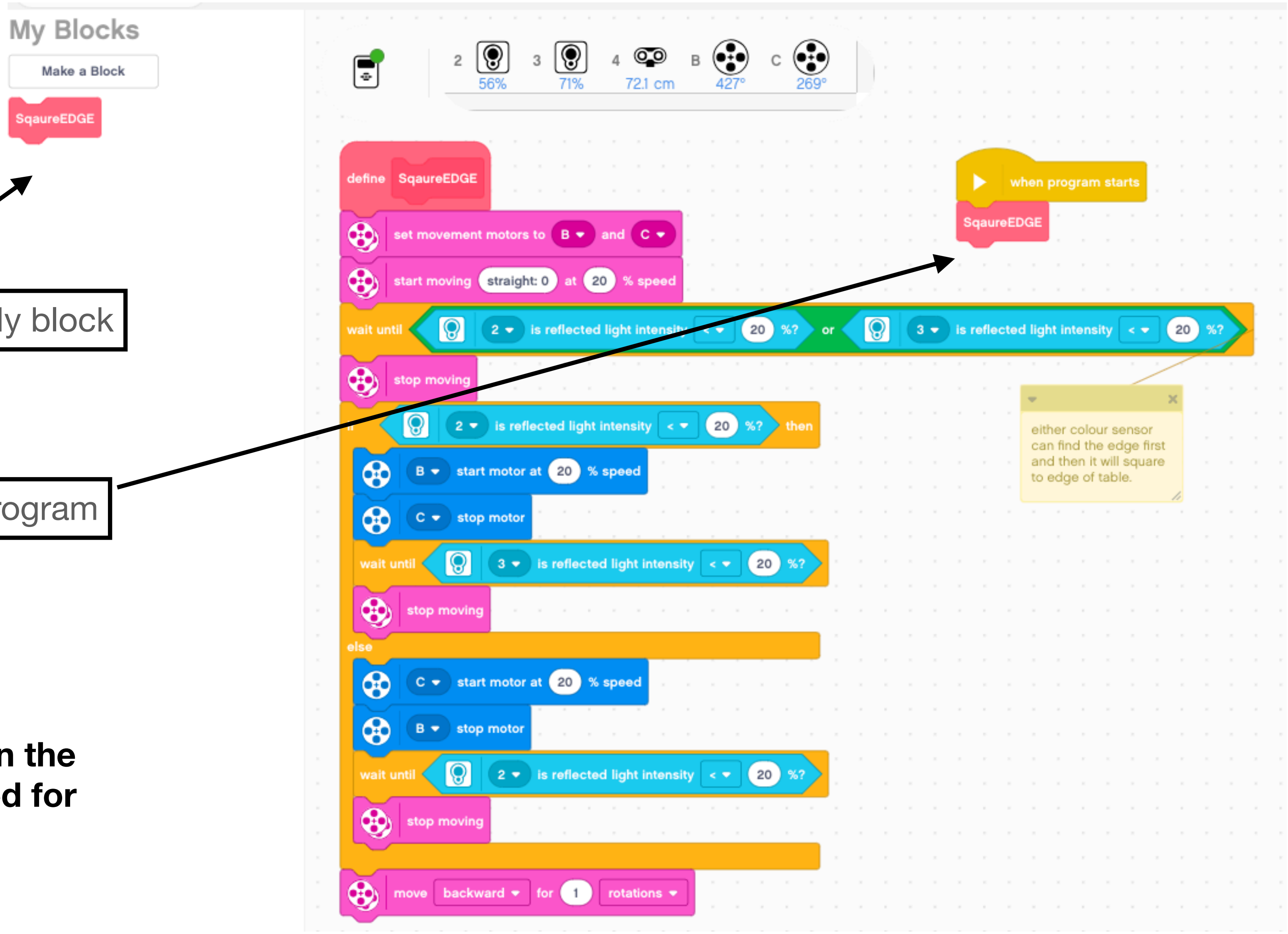
For this example we want our My Block to square to the edge of the table.





MY BLOCKS

How to create shortcuts for code you need to use often.



```
define SqaureEDGE
set movement motors to B ▾ and C ▾
start moving straight: 0 at 20 % speed
wait until 2 ▾ is reflected light intensity < ▾ 20 %? or 3 ▾ is reflected light intensity < ▾ 20 %?
stop moving
if 2 ▾ is reflected light intensity < ▾ 20 %? then
    B ▾ start motor at 20 % speed
    C ▾ stop motor
    wait until 3 ▾ is reflected light intensity < ▾ 20 %?
    stop moving
else
    C ▾ start motor at 20 % speed
    B ▾ stop motor
    wait until 2 ▾ is reflected light intensity < ▾ 20 %?
    stop moving
move backward ▾ for 1 rotations ▾
```

```
when program starts
SqaureEDGE
```

either colour sensor can find the edge first and then it will square to edge of table.

Select the defined My block

Insert it into your program

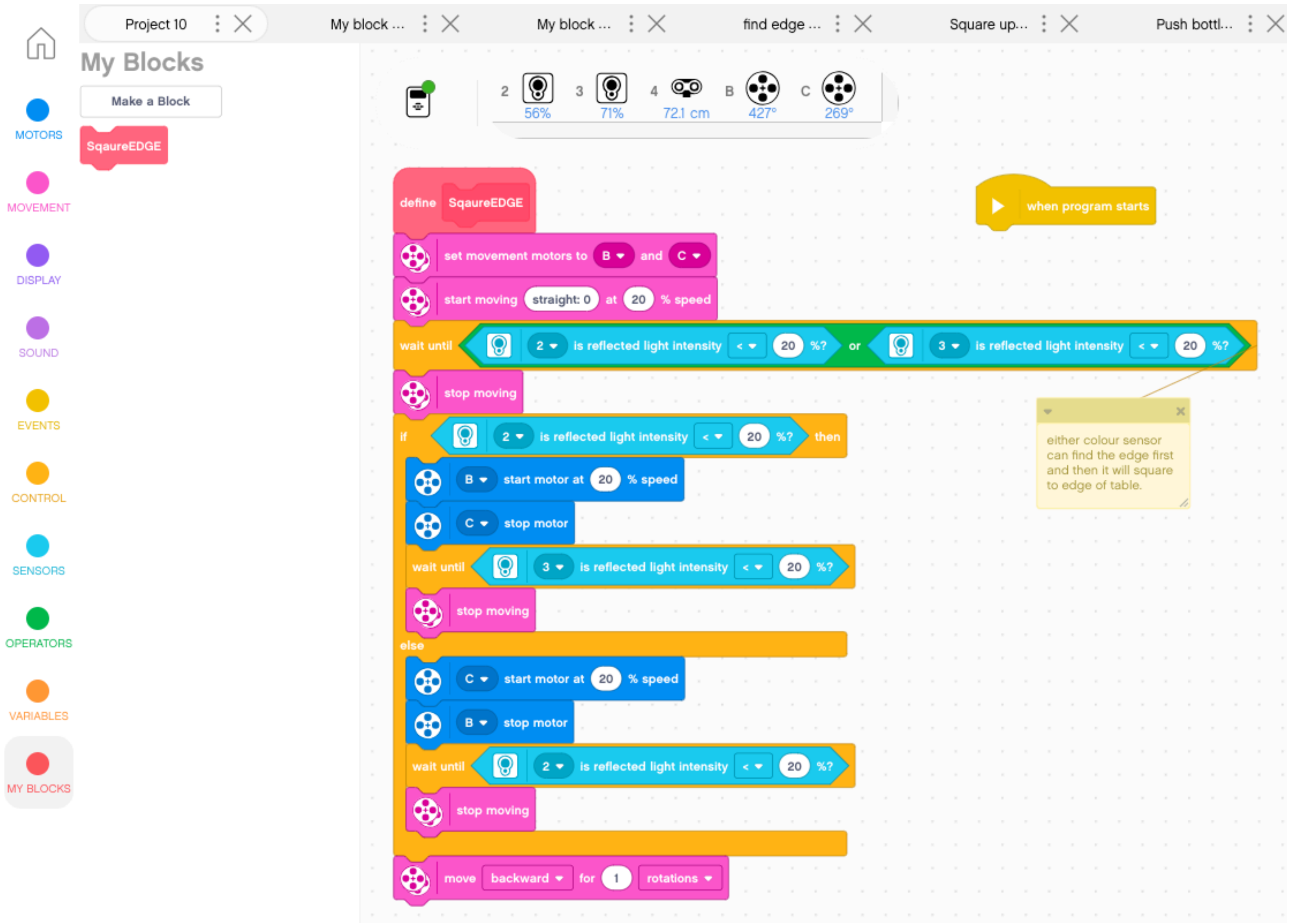
Your program will run the MY Block you defined for Square Edge

Your turn:

Create a **My Block** for a robot behaviour you will use often.

MY BLOCKS

How to create shortcuts for code you need to use often.



```
define SqaureEDGE
set movement motors to B ▾ and C ▾
start moving straight: 0 at 20 % speed
wait until 2 ▾ is reflected light intensity < ▾ 20 %? or 3 ▾ is reflected light intensity < ▾ 20 %?
stop moving
if 2 ▾ is reflected light intensity < ▾ 20 %? then
    B ▾ start motor at 20 % speed
    C ▾ stop motor
    wait until 3 ▾ is reflected light intensity < ▾ 20 %?
    stop moving
else
    C ▾ start motor at 20 % speed
    B ▾ stop motor
    wait until 2 ▾ is reflected light intensity < ▾ 20 %?
    stop moving
move backward ▾ for 1 rotations ▾

when program starts
```

either colour sensor can find the edge first and then it will square to edge of table.

In this example we COPIED the **square to edge of table** code and PASTED it under the define block.