What Is a Program?

What Is a Program? Pre-Quiz

1. **What is a program?**

2. **What is an algorithm? Give an example.**

What Is a Program? Pre-Quiz Answers

1. **What is a program?**

   A program is a sequence of instructions written to direct a computer to perform a task.

2. **What is an algorithm? Give an example.**

   An algorithm is a clear and specific procedure for solving a problem in a finite number of steps.

   ***Example*: The "addition algorithm" is a procedure for how to add any two numbers.**

What is a program?

In today’s world, computers are everywhere!

They are capable of running all sorts of programs to perform a huge variety of tasks. Just think about all the applications (apps) that are available for smart phones and tablets. They can do everything from playing chess to setting alarms to enhancing pictures.

Each app is a program designed to perform a specific task. As computers become even more widespread, the ability for people—like you—to be able to program becomes more and more important. Programming is how you tell a computer how to perform a task that you want it to do.

Objective

In today’s lesson, we:

- **Investigate programs and algorithms — *What are they?***
- **Become familiar with simple commands that are used in programs**
- **Learn how to use LEGO MINDSTORMS EV3 software to write programs that the LEGO robot can execute**

What is a program?

- A program is a sequence of instructions written to direct a computer to perform a task.
- All computers can run programs.
- *Example*: A calculator is a computer that is programmed to perform arithmetic operations.
- Similarly, the LEGO MINDSTORMS EV3 intelligent brick is a computer that we can program to perform different tasks, such as moving through a maze.

What is an algorithm?

An algorithm is a clear and specific procedure for solving a problem in a finite number of steps.

Example:

Addition algorithm: A systematic process that always produces the correct answer when numbers are to be added.

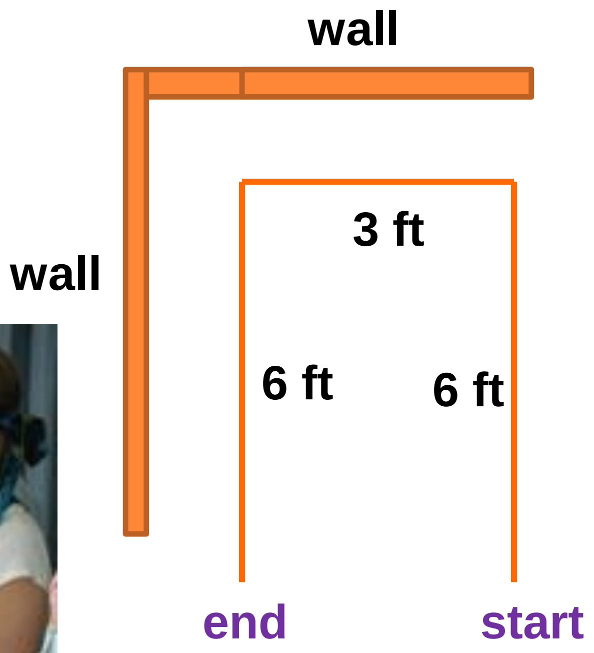
$$\begin{array}{r} {\scriptstyle 1\ 1\ \ } \\ 1\,2\,3 \\ \underline{+\,7\,8\,9} \\ 9\,1\,2 \end{array}$$

What is Programming?

- Programming is designing an algorithm to solve a problem.
- You need to have commands that are clear and precise… the robot will follow them exactly, so each step is important.

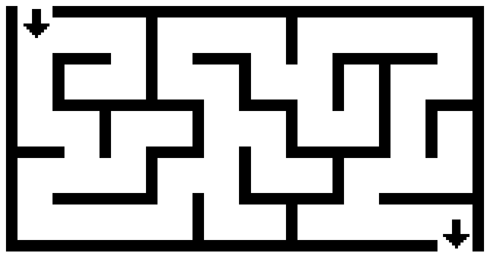
Mini-Activity: Programming as an Algorithm (10 minutes)

- Using a roll of tape, make a short maze on the floor in the corner of the room resembling this diagram ➘
- Blindfold a volunteer "robot" and have him/her stand at the start of the maze, hands at his/her sides.
- Have another student—the "programmer"—give a series of commands (such as, go forward 2 steps, turn right, etc.) to instruct the volunteer to complete the maze.



Mini-Activity: Programming as an Algorithm (continued)

Observations and Questions:

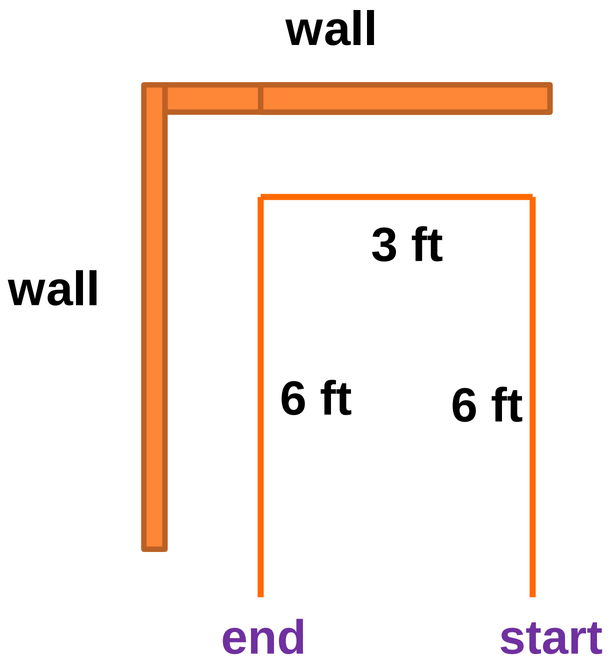


- **Did the "programmer" get the blindfolded "robot" through the maze?**
- **Did the "commander" always tell the "robot" to go the correct number of steps?**
- **Often, it is easier for the "programmer" to give instructions if s/he can tell the "robot" to go forward until s/he senses something. *Why?***
- **This way, the "programmer" does not have to worry about telling the "robot" exactly how many steps to move forward.**

Mini-Activity: Programming as an Algorithm (continued)

- Choose a new volunteer to be blindfolded.
- This time, permit the volunteer "robot" to stretch out his hands in front of his body to sense when he is approaching a wall.

- Have another student give commands to instruct the volunteer to get through the maze.



- Now, commands such as "go forward until you hit a wall" are permitted.

Mini-Activity: Programming as an Algorithm (continued)

- Was it easier to give instructions the first time or second time?

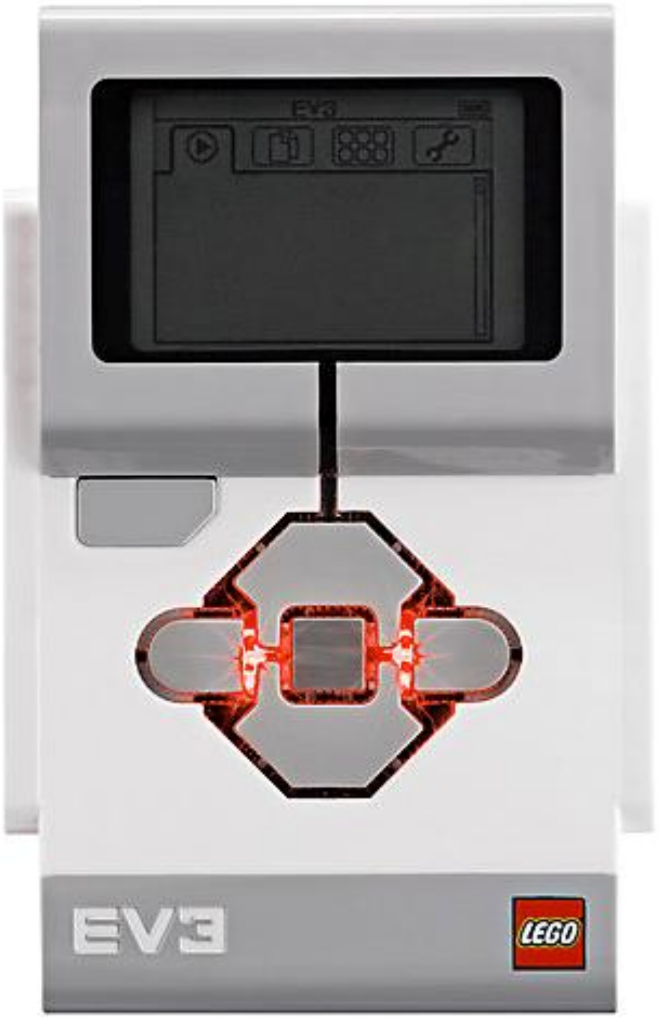
- It is generally easier to tell a robot to go forward until it senses something (such as a wall) than to tell it exactly how far it should go before it turns.

Mini-Activity: Programming as an Algorithm (continued)

- Note that the volunteer had to be told everything, that is, which direction s/he should move in, how long to move in that direction, how much to turn, etc.
- The LEGO EV3 robot needs to be given exactly the same information. It knows nothing and is "blind" and will do exactly what you tell it to do. If you make an error in programming, the EV3 won't do what you want.

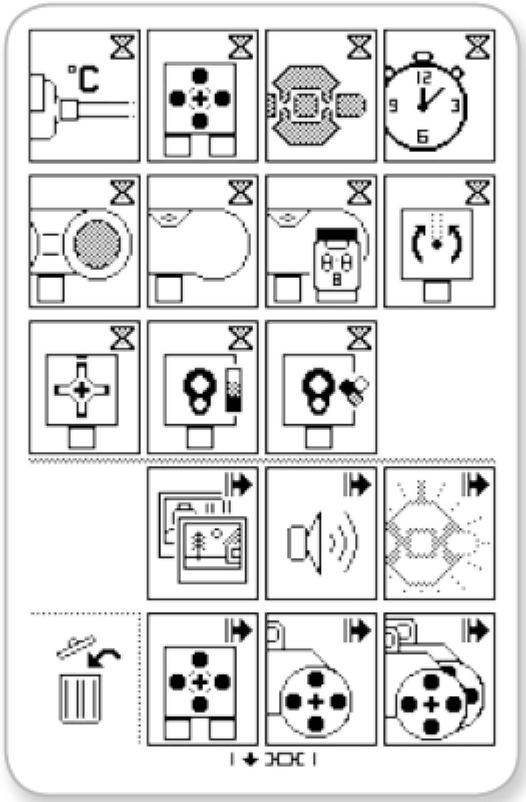
Quick Programming Using the EV3 Brick

- We can program a robot to do some simple tasks just by using the LEGO EV3 brick because the brick is a small computer.
- To turn on the brick, press the center button
- Then select Right button twice to the Brick Apps screen and select Brick Program App
- To choose block, use the Up button to see the Block Palette



Quick Programming Using the EV3 Brick (continued)

Do This: Let's program the robot to move forward for 5 seconds, then wait 2 seconds, turn right for 2 seconds, wait for 5 seconds, and then stop.


Full Block Palette

- **Select the "Forward 5" block.**
- **Then use the arrows to scroll to the "Wait 2" block and select it.**
- **Now scroll over to the "Right 2" block and select it.**
- **Scroll over to the "Wait 5" block and select it.**
- **Scroll over to the "Stop" block and select it.**
- **Scroll over to Start block at the very beginning of the program and press the center button.**

Quick Programming Using the EV3 Brick (continued)

Do This:

- **Press the up button to the block palette until you have cleared the 5 slots of the EV3 Program.**
- **Select different blocks and make your own quick programs on the EV3 brick.**

Quick Programming Using the EV3

Drawbacks

Programming directly on the EV3 brick is a convenient way to make simple programs and it is helpful to know the different features of the EV3 brick.

However, this method of programming is not very efficient and won't work for everything we want to do, for the following the reasons:

- The EV3 brick only provides a small set of commands and durations (timing choices) for programming.
- The EV3 brick can only make programs that are a maximum of 5 blocks long.

Quick Programming Using the EV3

Drawbacks (continued)

It is not even possible to complete the maze you worked through in the activity by programming on the EV3 brick.

So, from this point forward, we will use LEGO MINSTORMS EV3 software.

This software enables us to:

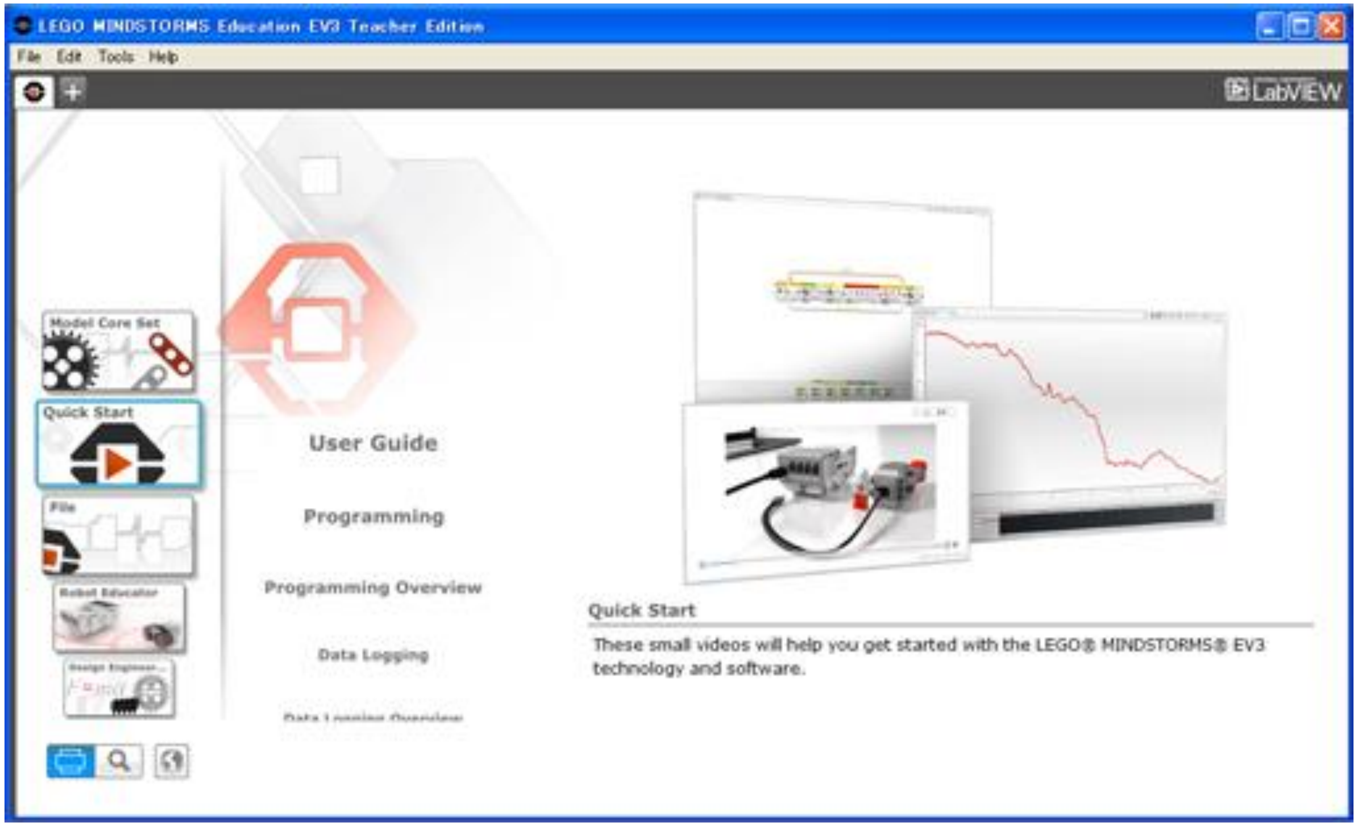
- make longer programs
- use more sophisticated commands
- have control over the fine details of the program

DAY 2

Getting Started with MINDSTORMS

Open the MINDSTORMS software by clicking on the start menu.

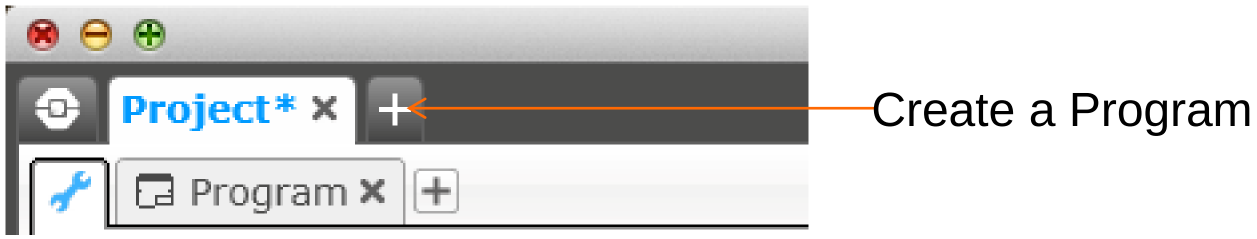
A window should open, looking like the one below. ↓



These small videos will help you get started with the LEGO® MINDSTORMS® EV3 technology and software.

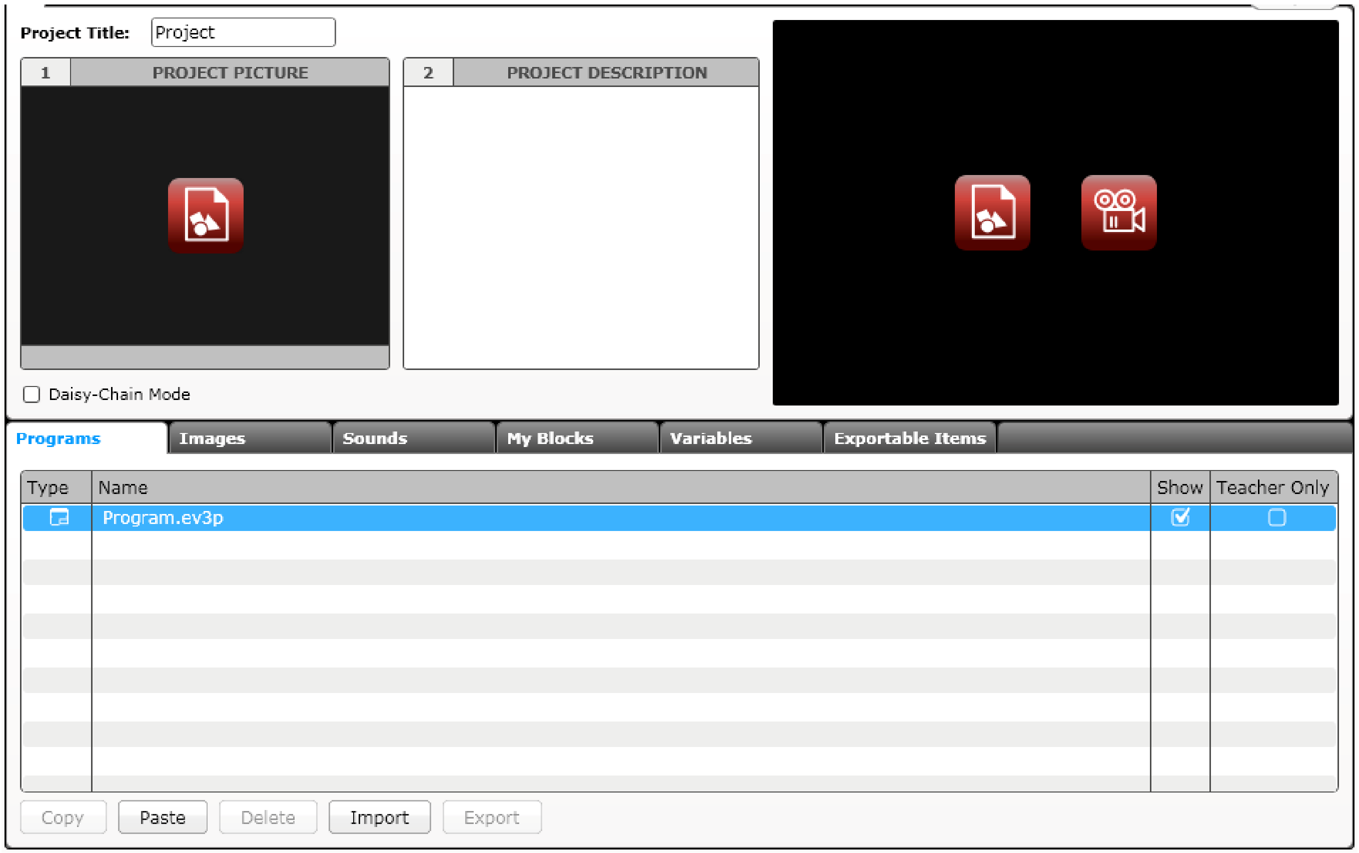
Getting Started with MINDSTORMS (continued)

Move the cursor inside the dialogue box next to "+" and create a new program.



Getting Started with MINDSTORMS (continued)

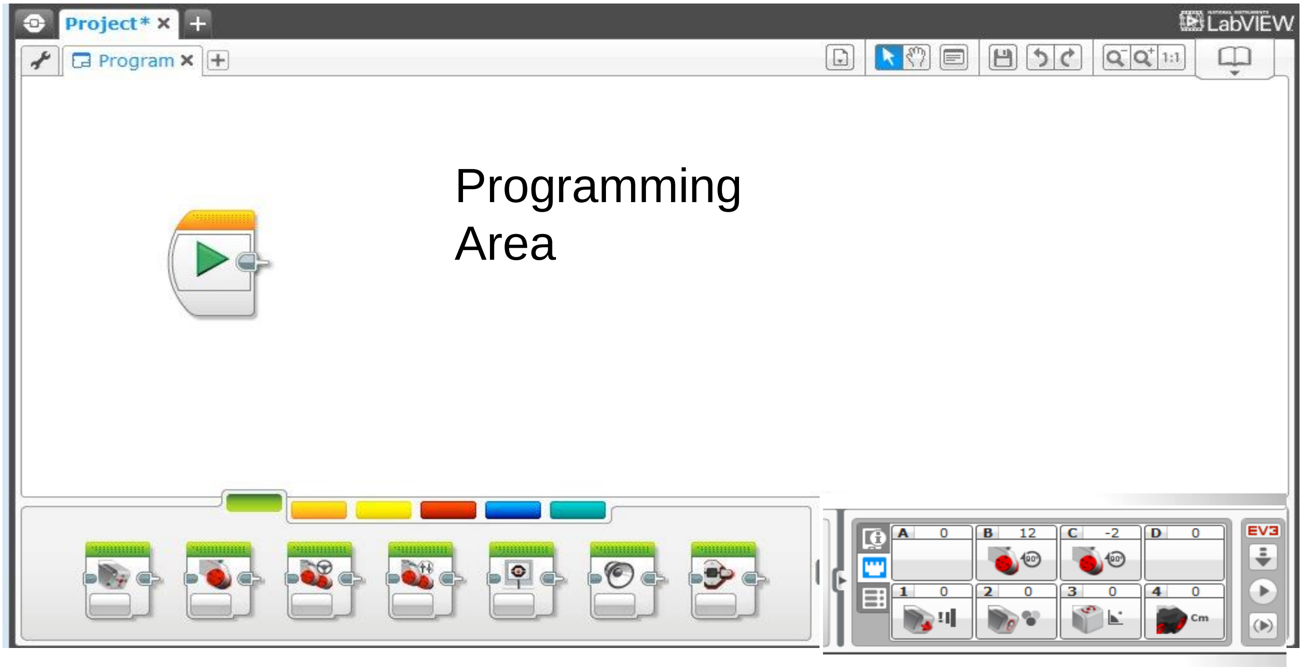
Your screen should now resemble the image below. ↓
This is the workspace you will use to create your programs.



Type	Name	Show	Teacher Only
	Program.ev3p	✓	☐

Getting Started with MINDSTORMS (continued)

- The square icons on the bottom of the screen represent different blocks that can be used for programming.
- In this lesson, we will be using only the move block.

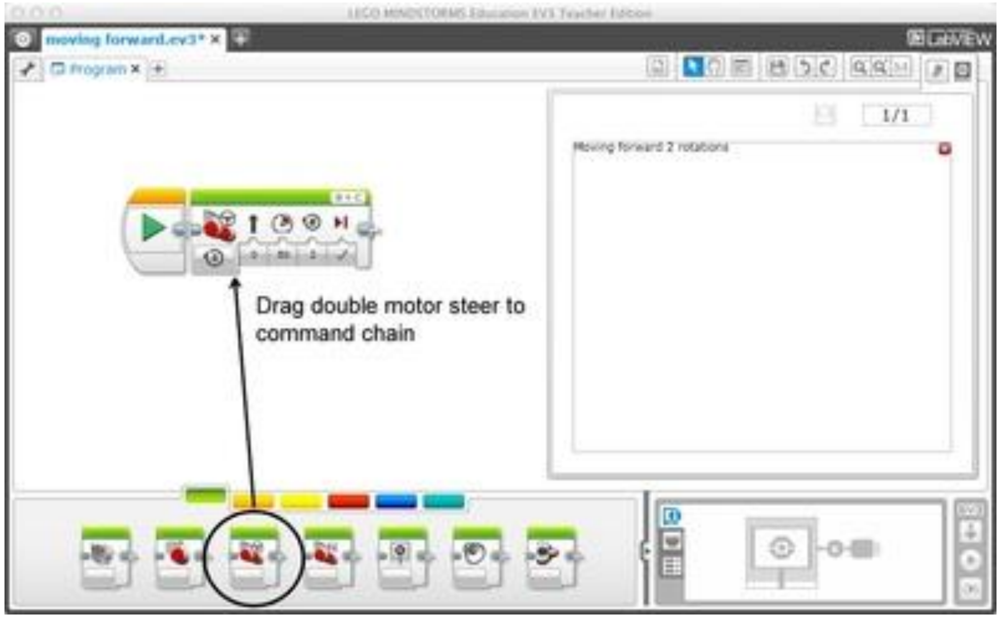


Programming Blocks in 6 Colored Tabs

Brick Status & Downloading

The Move Block

- Move your cursor over the move block icon.
- Click and drag the move block into the “start square”
- When you are done, the top left quarter of your screen should resemble the image shown below
- The move block in your program should be highlighted in a thin light blue outline. If it is not, highlight the block by clicking on it.

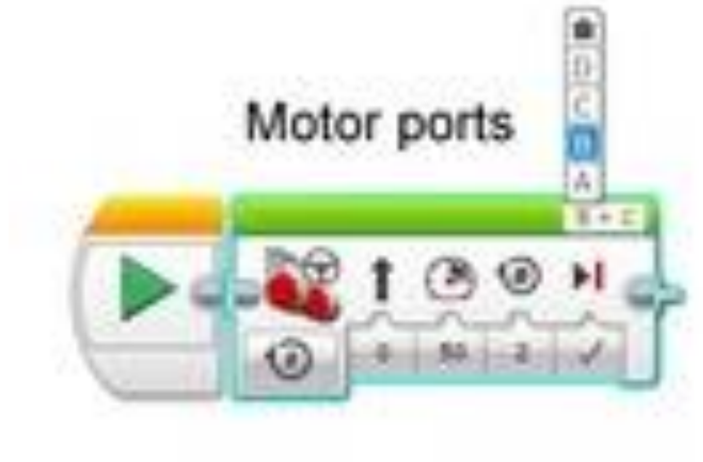


Moving forward 2 rotations

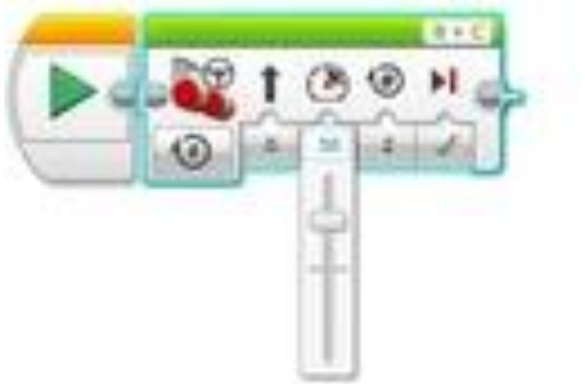
Drag double motor steer to command chain

The Move Block: Port

- A move block is used to control one or more motors.
- You will rarely need to change the “Port” settings.

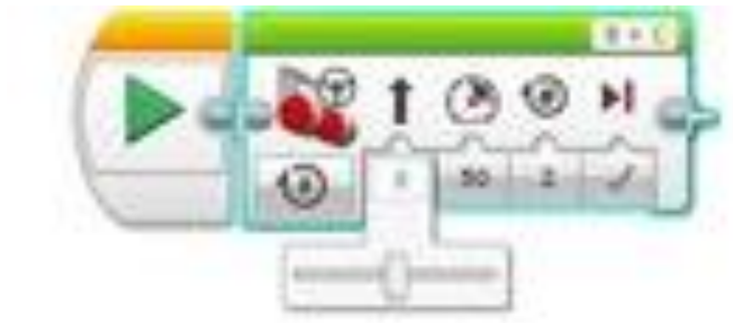


The Move Block: Power



- "Power" controls how fast the motors turn and which way the motors are set to rotate.
- The positive sign means forwards, the negative sign means backwards.
- Increasing or decreasing the power too much tends to decrease the consistency of the motor movement, so it is recommended that "75" is used, unless there is a good reason to change it.

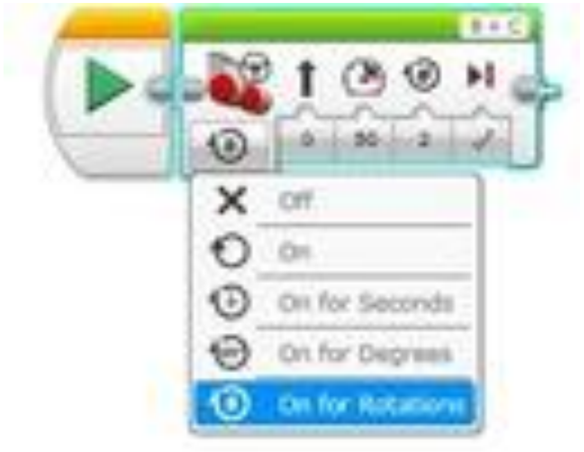
The Move Block: Steering



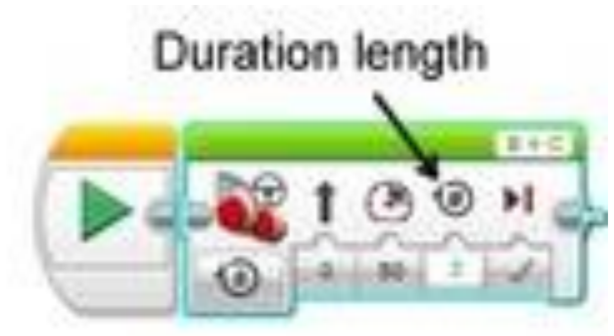
- Steering allows us to easily make the robot turn.
  - Dragging the pointer to the left makes the robot turn left
  - Dragging it to the right makes the robot turn right
  - The farther to one side the pointer is dragged, the sharper the turn will be
  - If the pointer is dragged all the way to one side, the robot should turn in place

The Move Block: Duration

- Duration controls how long the motors rotate.
- Clicking the dropdown arrow, reveals several options. ↓
- "Seconds" allows you to control the length of time the motors turn for.
- "Degrees" allows you to control precisely how many degrees the motor spins, where 360° is one rotation of the motor.
- "Rotations" allows you to control how many times the motor spins. (1 rotation = 1 full spin)

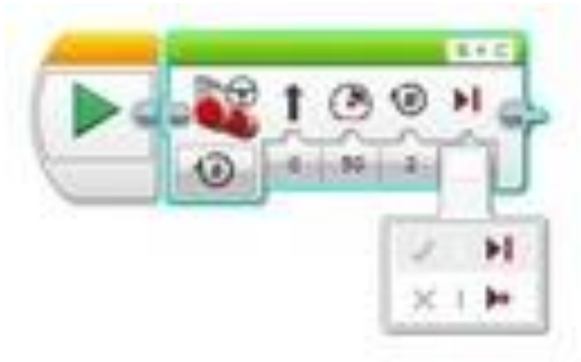


The Move Block: Duration (continued)



- Rotations is generally preferred over Degrees/Seconds because:
  - We generally need to make the robot move a few rotations in a direction and 3 rotations is much less tedious to type than 3*360 degrees = 1080 degrees.
  - Depending on factors such as the power setting and the amount of battery charge, the motors can spin different amounts of time in a second. But a motor spinning 1 rotation should always spin exactly the same amount.

The Move Block: Next Action



- You will almost never need to use the Next Action option
- It simply controls whether the motors are locked after performing a command or not
- This option matters if the robot is on a steep incline.
  - "Brake" prevents the robot from rolling back down after it executes a command
  - "Coast" allows the motors to be turned by gravity

Move Block: Helpful Hints

The robots are not perfectly consistent or alike but generally:

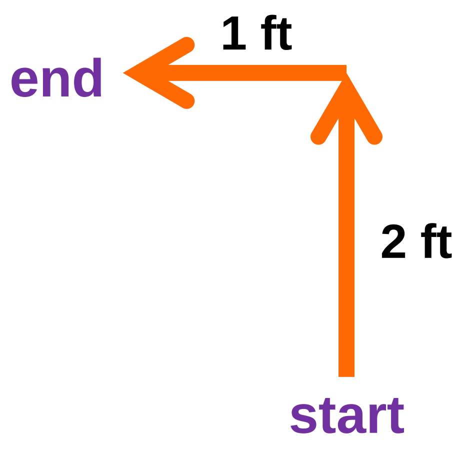
- **For every 1 rotation forward, the robot moves about 7 inches.**
- **Pulling the steering pointer all the way in one direction and setting duration to ".5 rotations" produces a turn that is close to 90°**

Example Program

Program the robot to follow the track shown below. ↓

Start by breaking this track down into steps:

1. Move forward for 2 feet
2. Turn left
3. Move forward for 1 foot



Part 1: Move forward for 2 feet

- First, we drag a move block into the program.
- Now we know we are moving straight forward. So this means, that we want to keep the upwards arrow selected.
- Next, we have to decide what to do about the steering. Since the robot is moving straight forward, we do not want the robot to turn at all, so we do not need to change the steering.
- Finally, we need to determine the duration:

  In Helpful Hints, we learned that 1 rotation is about 7 inches

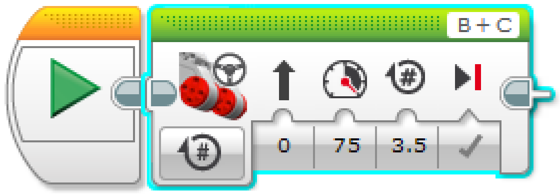
  1 ft = 12 inches, so 2 ft = 24 inches

  24 inches divided by 7 inches ≈ 3.5

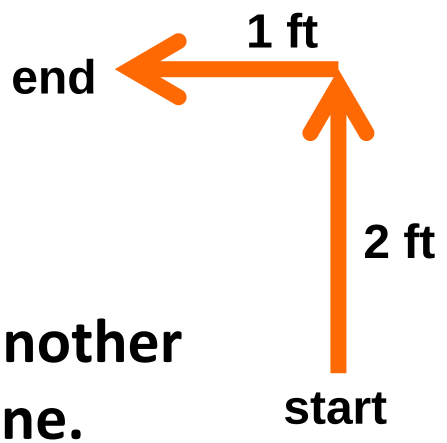
  So to go 2 ft, we need to go about 3.5 rotations

Part 1: Move Forward for 2 feet (continued)

So far, the program should look like this: ➔



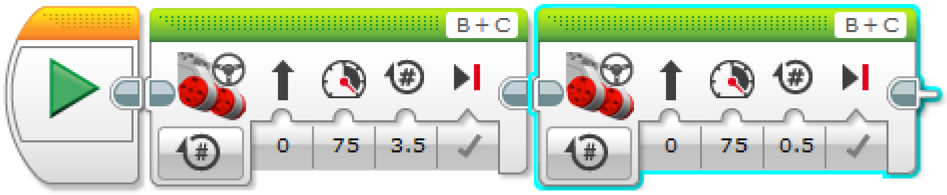
Part 2: Turn Left



- We want the robot to turn left next, so drag another move block down and set it next to the first one.
- Since the turn forms a rectangular corner, it is a 90° turn.
- Recall from "helpful hints" that we said moving the steering pointer all the way to one direction for a duration of .5 rotations gets us close to a 90° turn.
- So keeping the direction forward (the robot is moving forward as it turns), pull the steering pointer all the way to the left, and set duration to 0.5 rotations.

Part 2: Turn Left (continued)

**So far, the program should look like this.
The second move block is highlighted, and the details are shown below. ➜**



Part 3: Move Forward for 1 Foot

- Just as in Part 1, the robot is moving straight forward, so we do not need to change direction or steering.
- We do need to figure out the duration again.

This time, the robot should move 1 foot, or 12 inches forward.

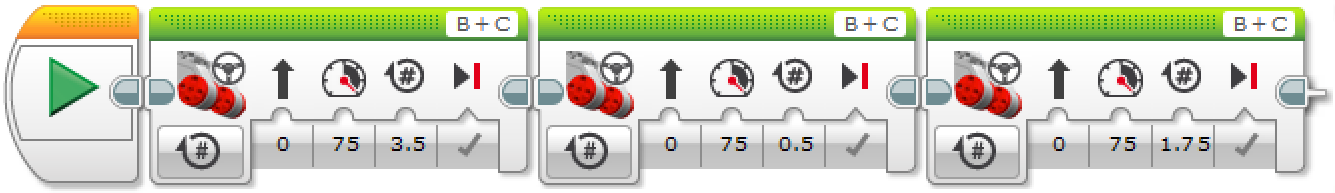
Remember 1 rotation is about 7 inches.

So 12 inches divided by 7 inches ≈ 1.75

So let’s set the duration to 1.75 rotations

Part 3: Move Forward for 1 Foot (continued)

The complete program should look like this, with the details of the highlighted third block shown below. ➜

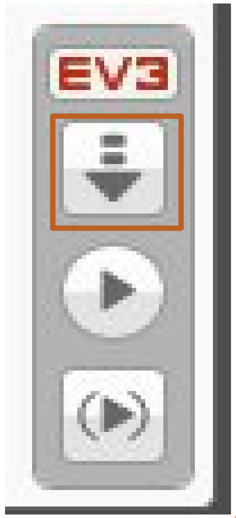


Downloading the Program onto the EV3

Now it's time to download the program onto your EV3:

Do This:

- **Plug one end of the USB connecting cable into the computer, the other into the EV3**
- **Make sure both ends are fully inserted**
- **Turn on the EV3 by pressing the center button**
- **Look for this set of buttons in the bottom right corner of your computer screen ➔**
- **To download the program, click on the down arrow**
- **Then press the center button until the program starts running**



What Is a Program? Post-Quiz

1. **What is a program?**

2. **What is an algorithm? Give an example.**

What Is a Program? Post-Quiz **Answers**

1. **What is a program?**

   A program is a sequence of instructions written to direct a computer to perform a task.

2. **What is an algorithm? Give an example.**

   An algorithm is a clear and specific procedure for solving a problem in a finite number of steps.

   ***Example*: The "addition algorithm" is a procedure for how to add any two numbers.**

Vocabulary

program: **A sequence of instructions written to direct a computer to perform a task.**

algorithm: **A clear and specific procedure for solving a problem in a finite number of steps.**

Images Sources

Slide 1: Girl looking at laptop monitor; source: Microsoft® clipart: http://office.microsoft.com/en-us/images/results.aspx?qu=computer&ex=1#ai:MP900430487|

Slide 8: Girl blindfolded in trust game; source: 2005 Tracee L. Jackson, U.S. Marine Corps via Wikimedia Commons: http://commons.wikimedia.org/wiki/File:Blindfoldlowres.gif

Slides 10, 11: Blindfolded kids in trees; source: North Carolina State Parks: http://www.ncparks.gov/Visit/parks/hari/pics/harp_maze.jpg

Slide 9: Maze line drawing; source: 2005 ZeroOne, Wikimedia Commons: http://commons.wikimedia.org/wiki/File:Maze01-01.png

Slide 15, 39: Girl on floor with laptop and hands raised; http://office.microsoft.com/en-us/images/results.aspx?qu=computer&ex=1#ai:MP900448599|

Slide 18: men's feet stepping: source: Microsoft® clipart: http://office.microsoft.com/en-us/images/results.aspx?qu=feet&ex=1#ai:MP900425511|

Device and programming images from LEGO MINDSTORM EV3 User's Guide http://goo.gl/wuhSUA

Screen captures and diagrams by author