**Hack-E-Bot: Making your Robot Move**

This tutorial will introduce you to the Hack-E-Bot, a crowd funded and open sourced Arduino robot aimed at kids to encourage their curiosity towards electrical engineering, electronics and programming.

Overview: We will be making your robot move in a square pattern and turning randomly if it detects an obstacle.

Disclaimer

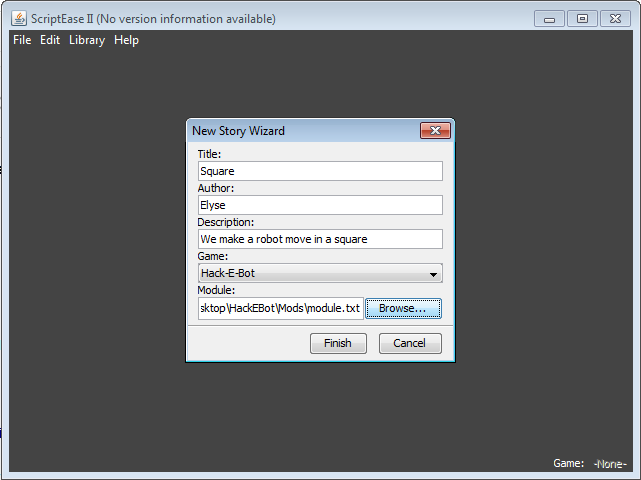
Unfortunately, as this translator is a work in progress, some problems may pop up in your experiments and may remain for the foreseeable future. You can find the current list of issues facing the Hack-E-Bot translator at (<https://github.com/UA-ScriptEase/scriptease/issues>). We wish you all the best with your robots.

Make a Square

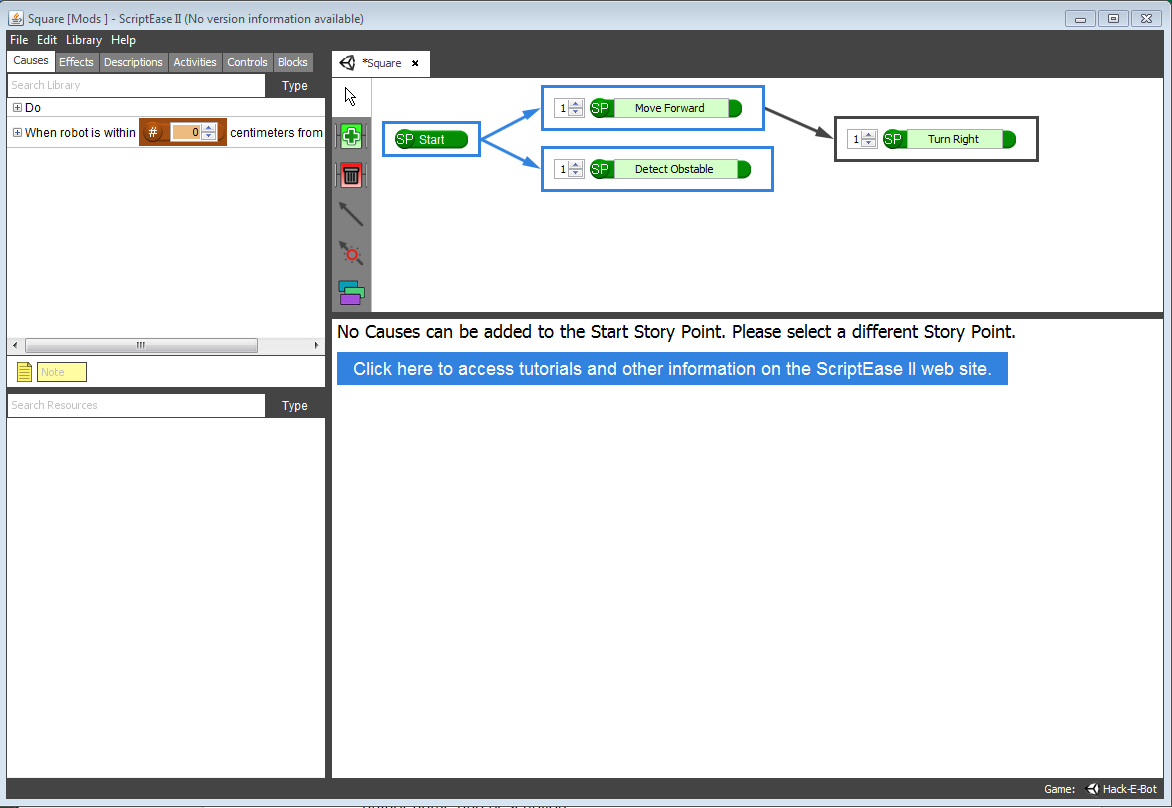
The Hack-E-Bot translator is a little different from our NeverWinter Nights and Unity translators. It is more straightforward and utilizes a global time function to keep track of where we are in the story tree. This is important to note since we will likely never want a situation where we have two parallel branches of our story active at the same time, as this will interfere with our time keeping and make the robot jerk around.

Let’s begin:

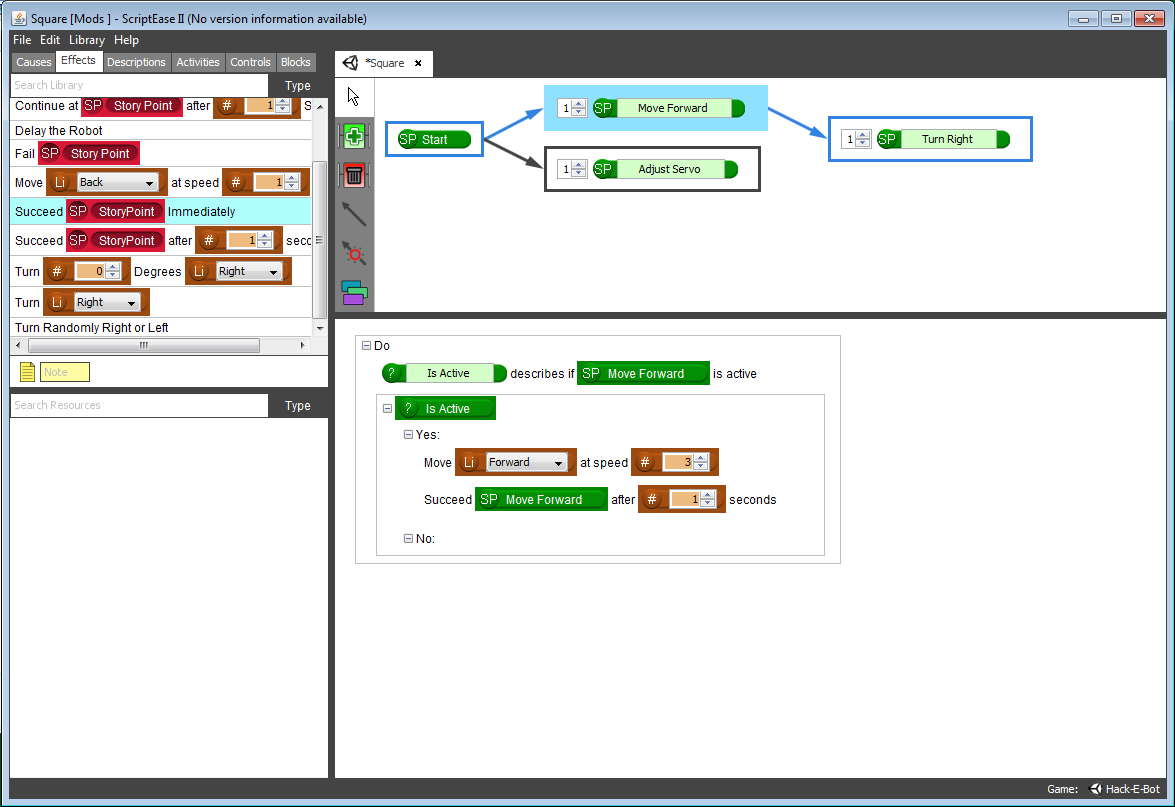
1. First, open ScriptEase and being a new story by navigating through File > New Story.
   1. Select a title, “Hack-E-Bot” from the drop down Game menu, and select a text file through the browse option for the module. Currently we save nothing to the module file so any text file will do.
   2. Author and description are optional.



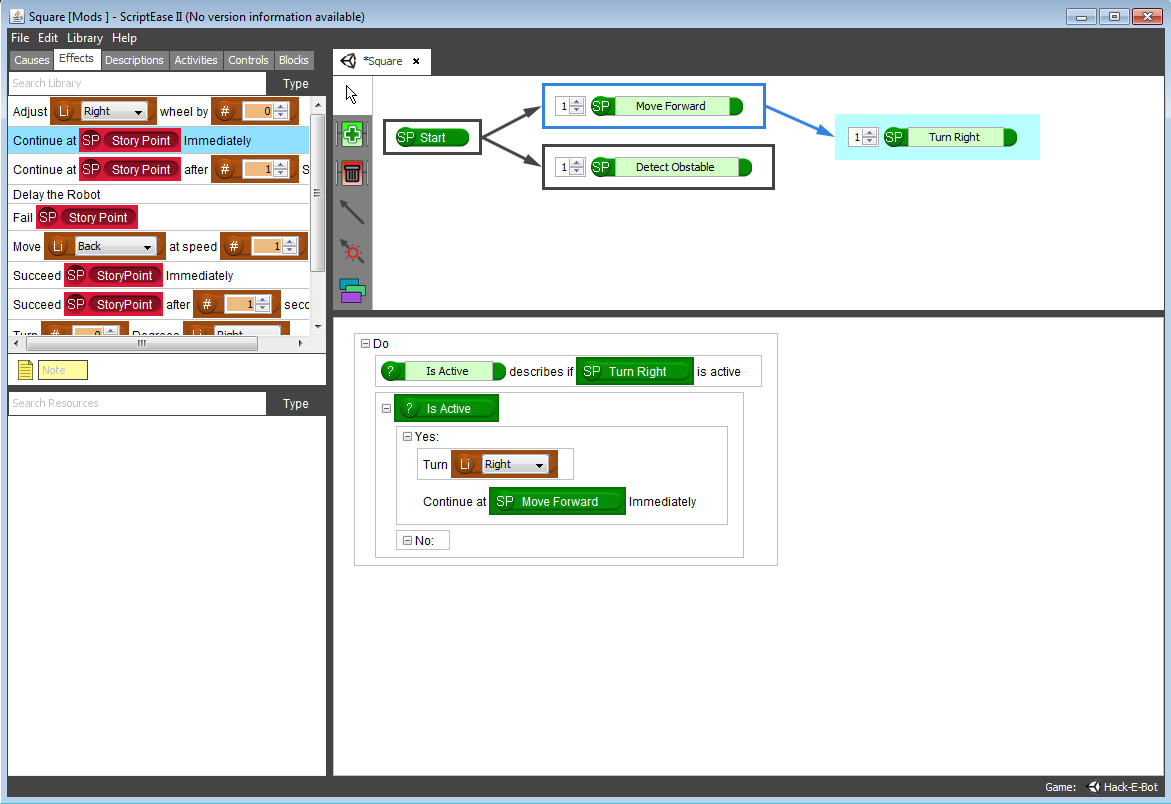
1. Let’s create three story points by selecting the  button and call them Move Forward, Turn Right and Detect Obstacle. Turn Right should be the child of Move Forward



1. Whereas with other translators, multiple actions can populate a single cause, with the Hack-E-Bot we only want one action per cause to make sure that we can utilize the timing of the robot properly.
2. Click on Move Forward to select and drag in the “Do” clause. Click on the clause and then drag in the “Move <Direction> at speed <number>” effect from the effect panel. We want to select Forward with a speed of 3
3. Next drag in the “Succeed <StoryPoint> after <Number> seconds” effect, and drag in the “Move Forward” story point name into the red (SP) field.

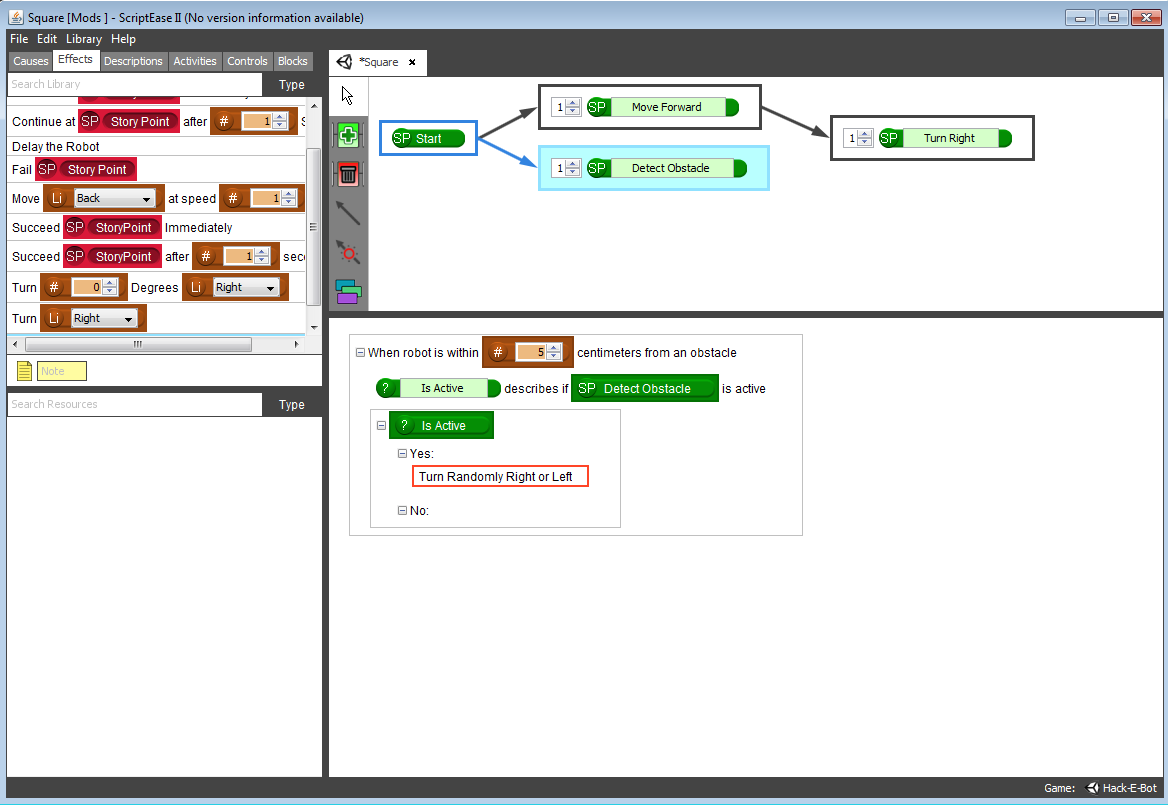


1. Next, select the Turn Right story point and drag in another “Do” clause.
2. In the Effects tab, select “Turn <direction>” and drag it into the Turn Right story point.
3. Next, drag in the “Continue at <StoryPoint> immediately” effect and populate the (SP) field with the “Move Forward” name.
   1. We want the “Continue Immediately” cause over the “Continue After <number> seconds” effect because “Turn <Direction>” already accounts for all the time it will take to turn. Conversely the reason we would use “Continue or Succeed after <number> seconds” is because “Move <Forward or Backward>” effect is internally split into 250 millisecond increments in order to use the “Detect” cause correctly.
4. “Turn Right” will now look like this:

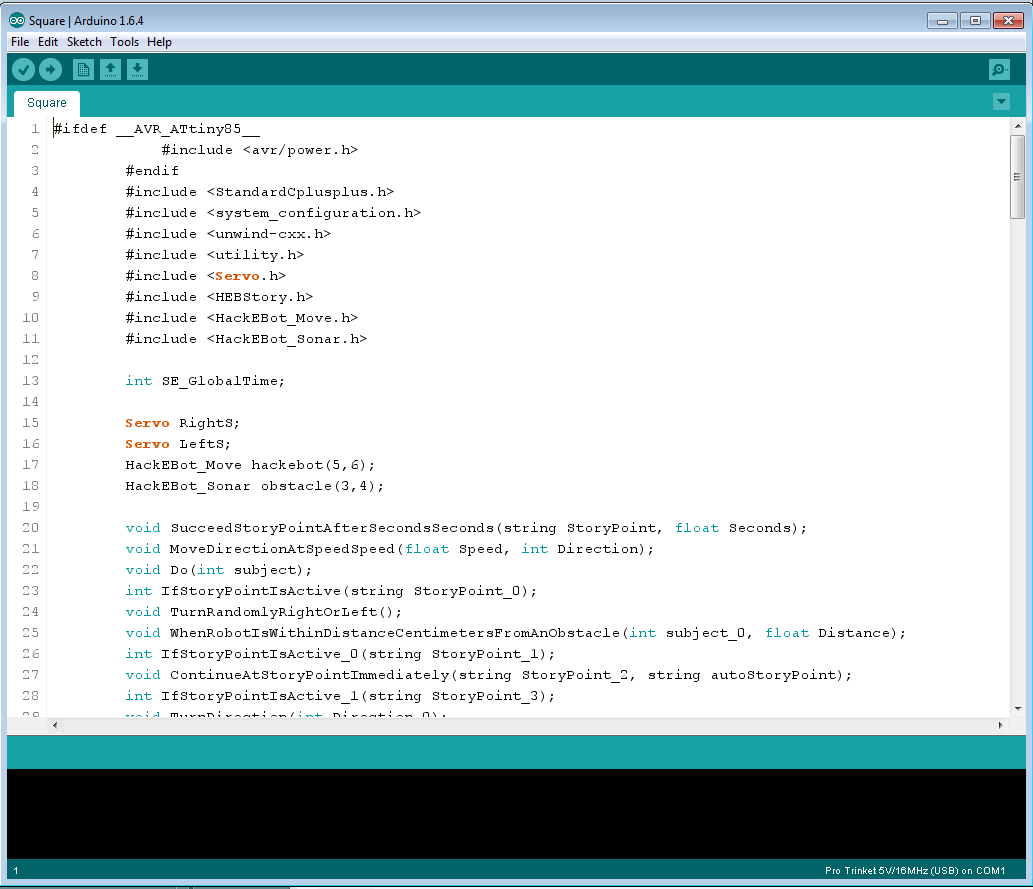


Detect an Obstable

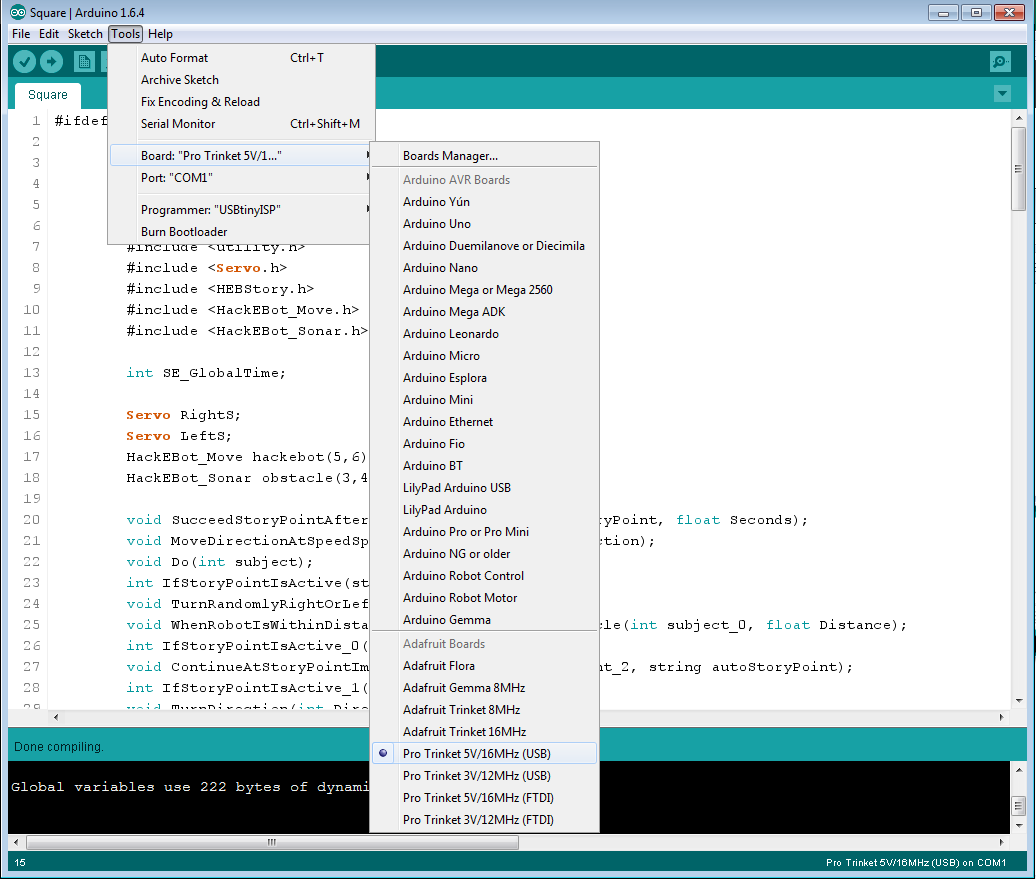
1. Next, click on the “Detect Obstacle” story point and drag in the “When robot is within <number> centimeters from an obstacle” cause. Make the distance 5 cm.
2. Drag in the “Turn randomly Right or Left” effect into the “Yes” portion of the “IsActive” question.



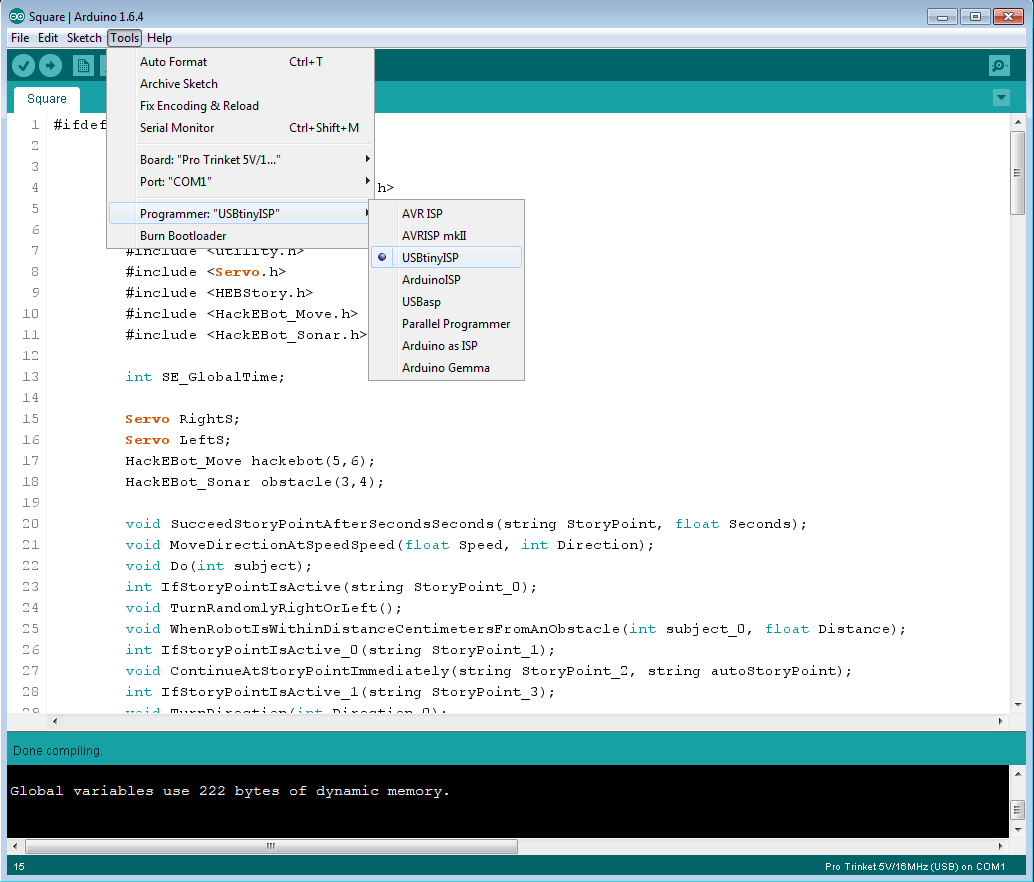
1. We don’t want to succeed or continue this story point because we always want the robot to detect if there is an obstacle in front of it.
2. Next press the F9 key and open up Arduino, the code that you just created will look like this!



1. Press the “Checkmark” button on the top left to compile your code. This might take about 20 seconds.
2. Next make sure that you have the correct board selected. Navigate through Tools > Board and select “Pro Trinket 5V/16MHz (USB)



1. Next we want to make sure that the right USB connection is selected. Navigate to Tools > Programmer and select “USBTinyISP”



1. With that done, plug in your Hack-E-Bot and press the “Arrow” upload button next to the “Checkmark” compile button. When the code is done uploading, turn on your robot and then place it on a flat surface and watch it go!

Adjust the Wheels

It is sometimes necessary to change the speed at which the wheels turn in order to make the robot move straight. If you notice that the robot pulls to the one side faster than the other, your wheels might need some calibration. In the internal programming, the wheels of the robot are set at 90 as the default speed. However since we are dealing with two physical servos, 90 might not be the what is considered ‘default’ to the wheel itself and may spin faster or slower due to its physical components. Adding or subtracting a number from 90 will likely fix this issue. Let’s do this now.

1. Delete the “Detect Obstacle” story point and add a new story point called “Adjust Wheel”
2. Click on this new story point and drag in the “Do” clause.
3. Click on the “Do” clause and then drag in the “Adjust<right or left> wheel by <number>”
4. For this exercise we’ll say that the right wheel is too fast and subtract -1 from it’s speed.
5. Next, drag in the “Succeed <StoryPoint> immediately” effect. We only want to adjust the wheels once during our story. If we left the story point always active, after a while the left wheel would be the only wheel turning turning.
6. Press F9 to open Arduino again.
7. Compile your code, and upload your code to your robot.
8. It should now move much straighter!