Webots-Blockly Tutorial

Jeffrey Cheng, Victor Hu, and Julian Lee

Contents

1 Introduction

Webots is a simulation environment in which users can program robots without having a physical robot (see https://cyberbotics.com/doc/guide/index for more information). Webots-Blockly is an extension where users can use drag-and-drop programming within Webots to program the robots. This is a guide on how to install and use Webots-Blockly on Windows. Note that some of the steps and/or images may look slightly different depending on when you are working with Webots-Blockly.

2 Installation

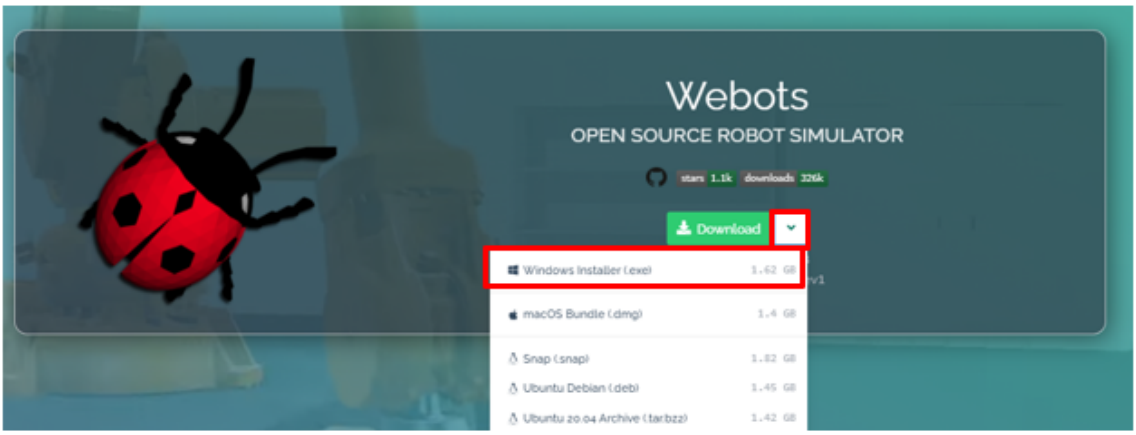
To run Webots-Blockly, you will need to install/download three items: Webots R2020b, Python 3.8, and the Webots-Blockly GitHub Repository. This section will guide you in the installation process, but there is also a video tutorial on installing Webots-Blockly: .

2.1 System Requirements

Visit the following link to make sure your computer satisfies the system requirements for installing Webots: https://cyberbotics.com/doc/guide/system-requirements. Just about any computer should be able to install Python and download the GitHub Repository - it is mainly Webots that may require concern. To install the three items listed above, you will need at least 5 GB of free disk space. It is also recommended that you plug in your computer directly to the router to shorten the installation time for Webots.

2.2 Installing Webots

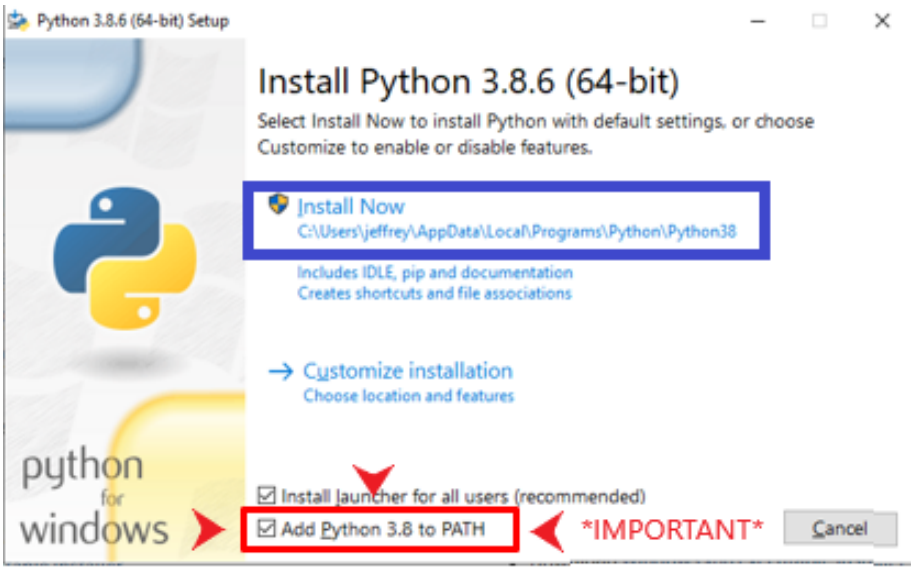
To install Webots, visit https://cyberbotics.com/. Select the drop-down arrow in the middle of the page and select the option according to your operating system. This automatically downloads the latest version of Webots.



Run the .exe file that was downloaded and a new window should appear. Select "Next" -> "Next" -> "Install." Press "Finish" after the install is completed, and you may exit from Webots for now.

2.3 Installing Python

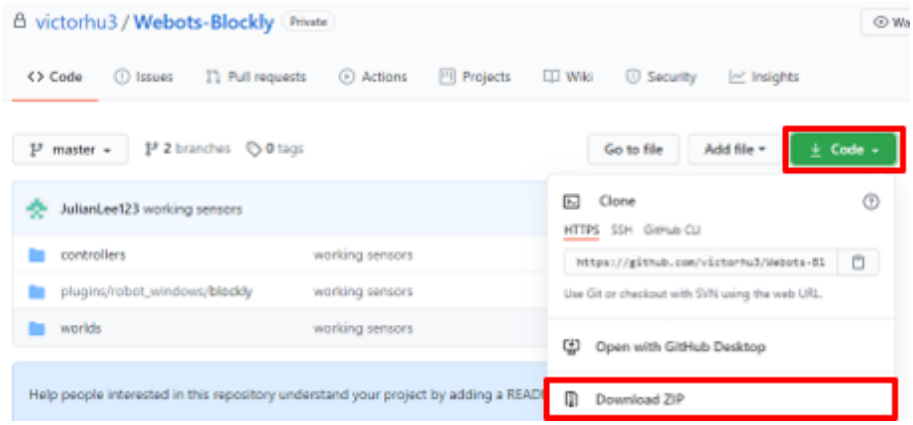
Find out the latest version of Python that your Webots supports at https://cyberbotics.com/doc/guide/using-python – the Python versions should be listed in the second paragraph of the website's introduction section. Note that this guide's version numbers may be outdated based on when you are performing this installation. Visit https://www.python.org/downloads/windows/ and download the appropriate version of Python for your computer – select the "Windows x86-64 executable installer" if you are unsure of which option to choose. Open the .exe that is downloaded. IMPORTANT: When the installer opens, check the box for "Add Python 3.8 to PATH." Then, select "Install Now." If you did not check this box by accident or you did not check it when you previously installed Python, see Error #1 in the "Common Errors" document.



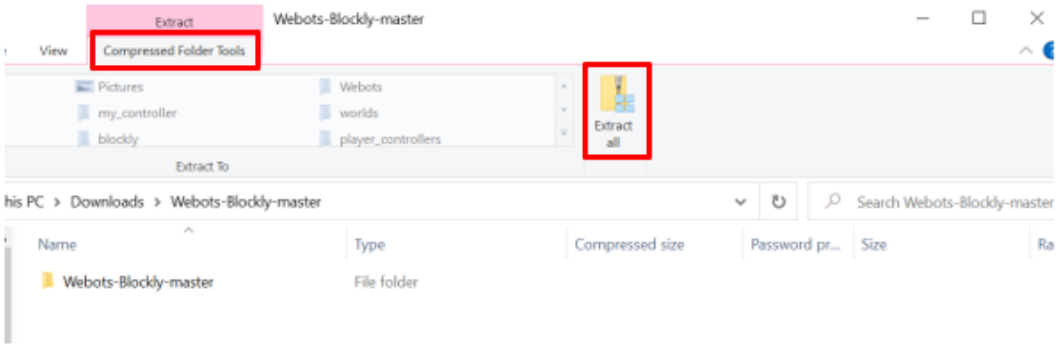
2.4 Downloading the Webots-Blockly GitHub Repository

The Webots-Blockly GitHub Repository is where you can find the drag-and-drop extension. **Note:** If you do not use the Webots world within this repository, you will not have access to the drag-and-drop

programming feature. Open the following link to find the GitHub repository: https://github.com/victorhu3/Webots-Blockly. Select the green "Code" button and press "Download ZIP" to download the project folder.

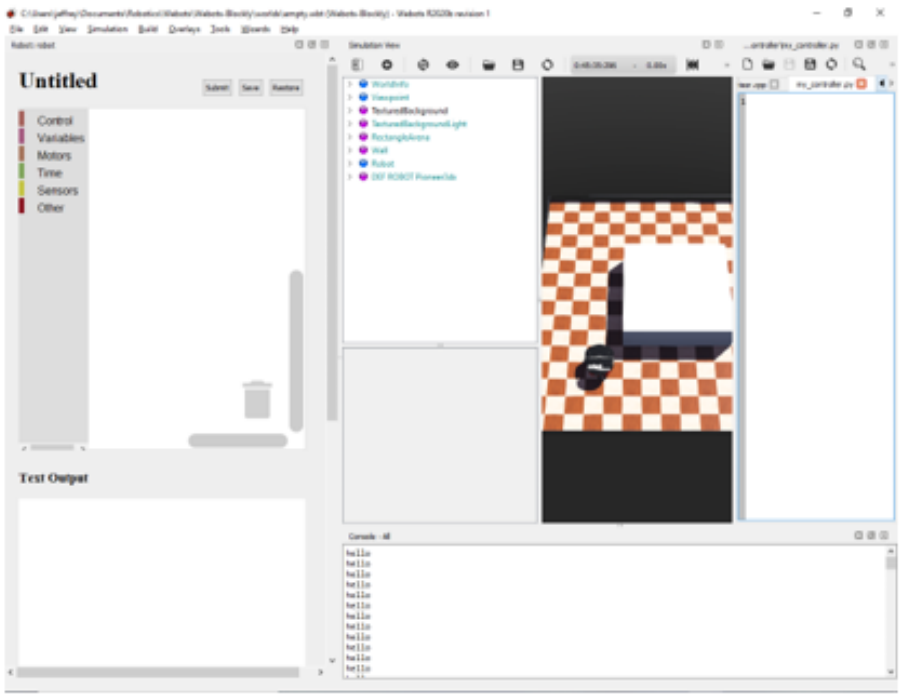


Click on the .zip that is downloaded. When it opens, click on "Compressed Folder Tools" at the top, select "Extract all," and select "Browse" to select a location for the project folder to be saved. **Remember where you save your project folder**.

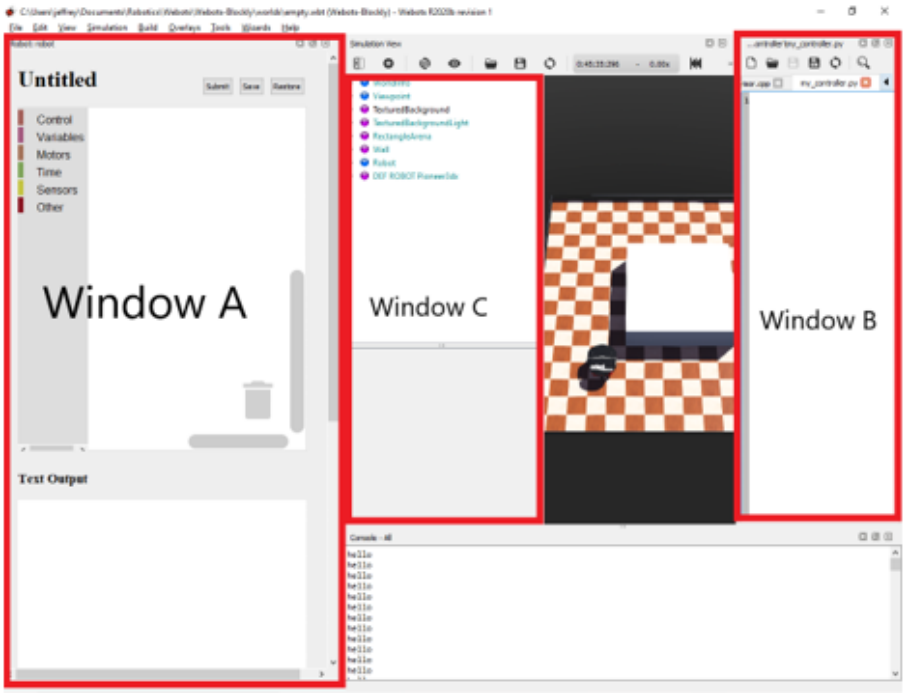


Navigate to where you saved the project folder and go into the "Webots-Blockly" folder and then the "worlds" folder. Double click on the file named "empty.wbt" (it should have a red ladybug as its icon), and that should open Webots to a simple world. **Note: Whenever you want to use Webots-Blockly, you must open Webots by opening a world file from this "worlds" folder. Otherwise, it will not contain Webots-Blockly.**

2.5 Optimal Window Layout



When Webots opens, your screen should look somewhat like the image above. Follow the steps below to achieve the optimal window layout for Webots-Blockly, which use the diagram below. Note: It is perfectly normal to see the error message "SyntaxError: Unexpected token '<' (blockly.html?175_0:1)" and "Blockly.Constants.Logic.HUE is deprecated and unused" outputted in your console window. Any red errors, however, should draw your attention. See the "Common Errors" document if you encounter any unmentioned errors.



2.5.1 The Webots-Blockly Window

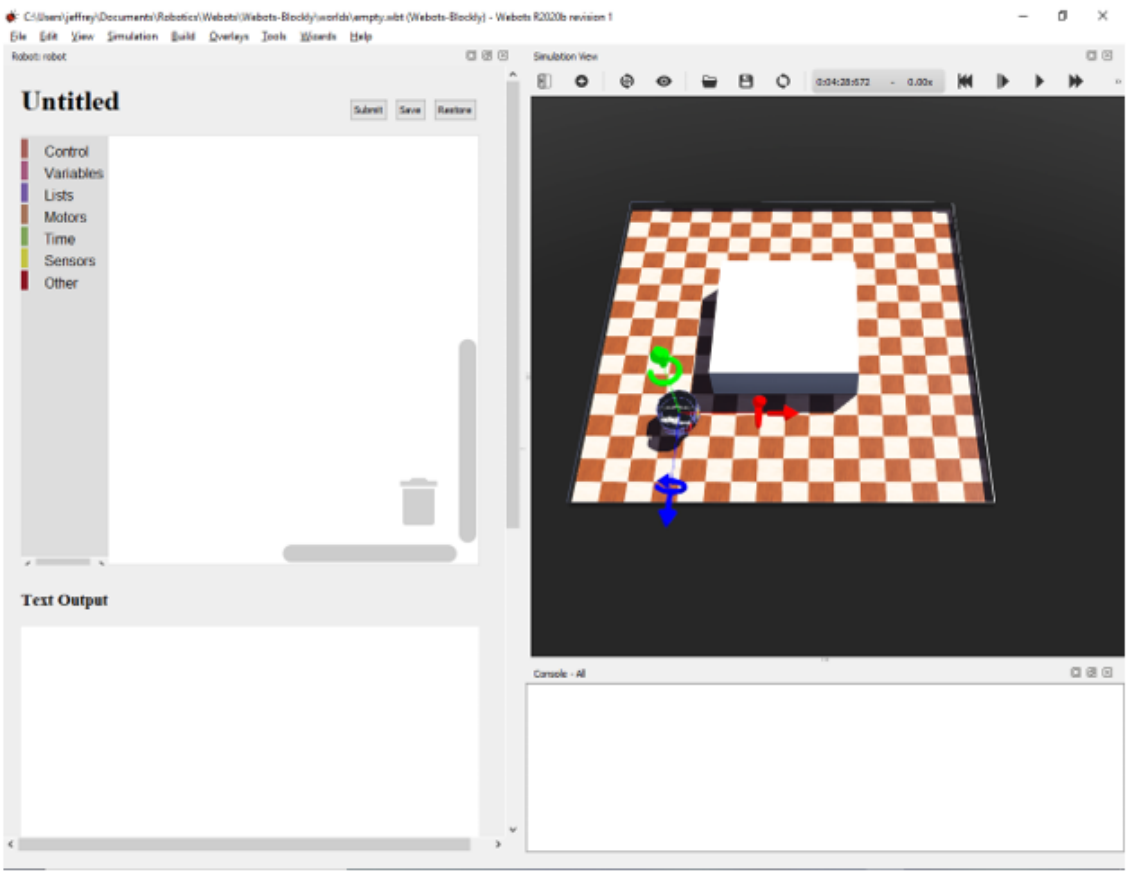
The Webots-Blockly Window is Window A from the diagram. This is your workspace for working with Webots-Blockly. If you do not see this window, go to Window C, right click on the "Robot" bullet point, and select "Show Robot Window." If you do not see Window C, press "Ctrl + t." You can resize this window by clicking and dragging its right border. You may need to make this window smaller if you cannot see the pause/play button mentioned later.

2.5.2 The Text Editor

The text editor is Window B from the diagram. If you do not see this window, you may skip this step. Since you are not typing your code, you do not need this window cluttering your simulation view. First, close all the files within the text editor – there should be three named "my_controller.cpp," "MAKEFILE," and "supervisor.cpp." **Do not edit these files**. Next, close out of the text editor itself with the "x" button at the top right corner.

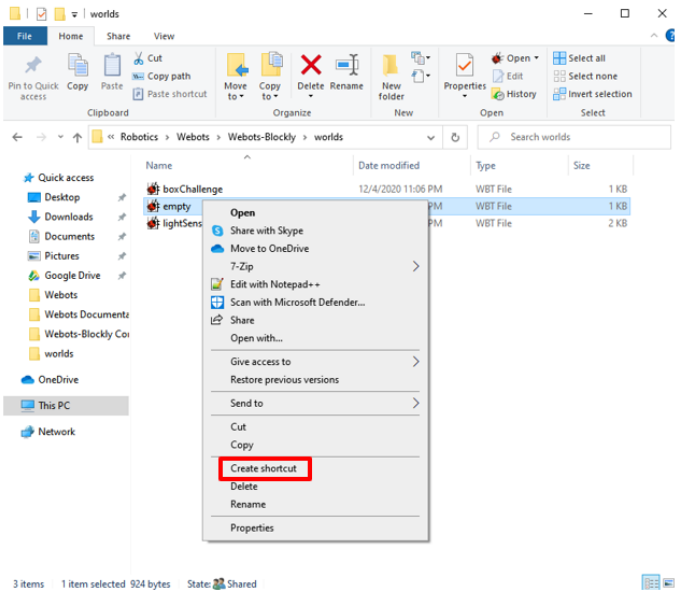
2.5.3 The Scene Tree

The Scene Tree is Window C from the diagram. If you do not see this window, you may skip this step. This window contains all the properties of the simulation world you are looking at. However, you do not need this window for now, and it only blocks your view. Click and drag the window's right border to the left until the entire column collapses. You should now see a screen similar to the image below, which is the optimal view for using Webots-Blockly.



2.6 Webots-Blockly Shortcut - Optional

As mentioned before, you have to open Webots worlds from the folder you downloaded of GitHub in order to use Webots-Blockly. If you want to avoid the process of opening the folder and finding the worlds every time, you can create a shortcut. Find the world file you want to work with whenever you use Webots-Blockly (e.g. empty.wbt). Right click on the world file and select "Create shortcut."



Drag the shortcut file into the background of your home screen. Alternatively, copy and paste the shortcut file that is created, navigate to your "Desktop" folder, and paste the file into "Desktop." On your home screen, you should now see a shortcut to open up the desired Webots-Blockly world file.

3 Using Webots

Note that this guide is not a full tutorial for using Webots and is only meant for you to get started with Webots-Blockly. For an in-depth explanation of all the features of Webots, visit https://cyberbotics.com/doc/guide/index.

You should see several windows in front of you. The Webots-Blockly Window to the left is where you will program, the Console window at the bottom right is where all print statements are displayed, and the world view to the upper right is a 3D visual of the world your robot is in. Left click and drag to rotate the world and right click and drag to shift the world.

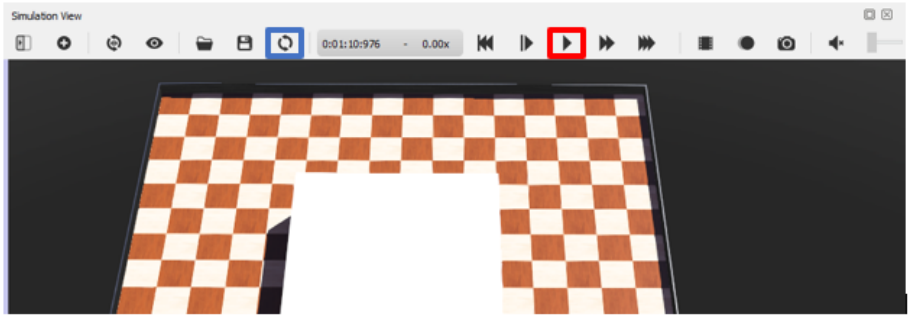
4 Important Buttons

4.1 Simulation Buttons

To interact with the simulation, there are two buttons you need to know about. The first is the pause/play button in the red box below. Use this button to pause and play the simulation. When the simulation is playing, the timer to the left of the pause/play button will begin counting.

The second button you need to know about is the reload world button in the blue box below. This button reloads the entire Webots world, and you will use it to reset the simulation for a new run. Make

sure the simulation is paused before reloading the world, or else the simulation will automatically start after the world is reloaded.



4.2 Webots-Blockly Buttons

The window on your left is the Webots-Blockly window. At the top, you can change the name of your file from "Untitled" by simply clicking on the word. To the right, you will see the submit, save, and restore button.

- **Submit Button:** Uploads your code to the robot for you to run.
- **Save Button:** Saves your code. You can find the file by going into your project folder, then your "controllers" folder, and then the "Blockly_Programs" folder. Note that your code will not be automatically saved when you run your program.
- **Restore Button:** Allows you to open any file you have previously saved.

The box labeled "Text Output" is where you can see your block program translated into Python. You do not have to interact with this box to run your program but watching it can be quite interesting and insightful.

5 Programming with Webots-Blockly

Programming with Webots-Blockly works just like any other drag-and-drop programming application. Click the categories on the left and drag the desired blocks into the white space, connecting them in sequential order.

5.1 Running a Webots-Blockly Program

When you want to run your program, follow these steps:

1. Save your code using the ”Save” button at the top.
2. Press the ”Submit” button to upload your code to the robot.
3. Make sure the simulation is paused so that the simulation does not automatically run after being reset.
4. Reload the world. This resets the simulation so you are ready for a new run.
5. Press the pause/play button to run your simulation.

6 Tutorial

This is a quick tutorial to introduce you to programming in Webots-Blockly. For a complete documentation on all the features of Webots-Blockly, see the “API Documentation” document.

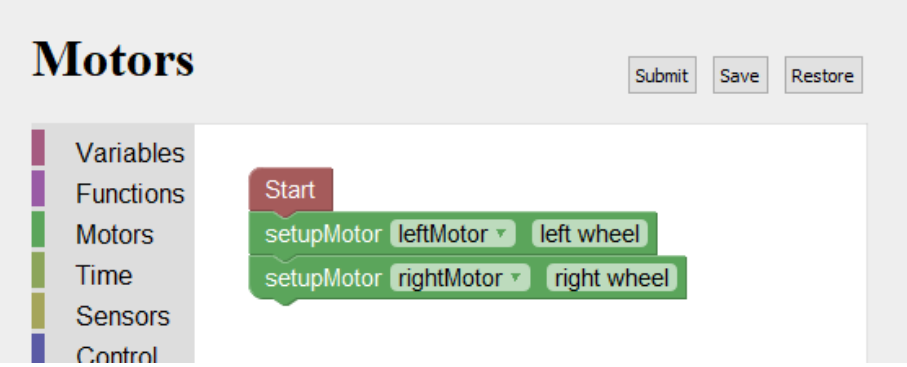
6.1 Basics

Every Webots-Blockly program must start with a *Start* block, which can be found in the “Other” category. You can also find the *print* block in the “Other” category. Attach values to the right of the *print* block to display to the Console window.

6.2 Motors

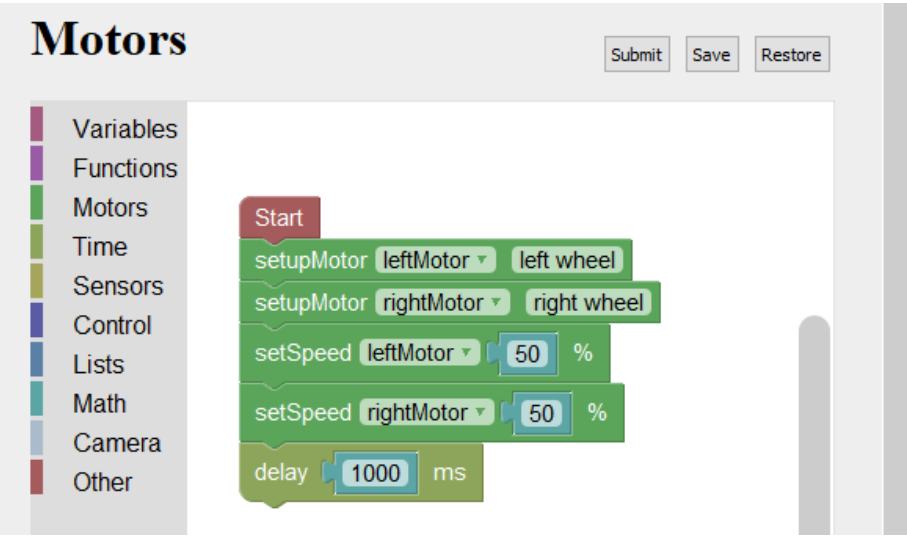
To get started with motors, follow these steps:

1. Create two variables called *leftMotor* and *rightMotor*.
2. Drag out 2 *setupMotor* blocks from the “Motors” category and attach them to the *Start* block.
3. For the first *setupMotor* block, use the drop-down menu to select the *leftMotor* variable and type in “left wheel” into the second box (omit the quotes).
4. Repeat step 3 for the second *setupMotor* block, but use the *rightMotor* variable and “right wheel” (again, omit the quotes).



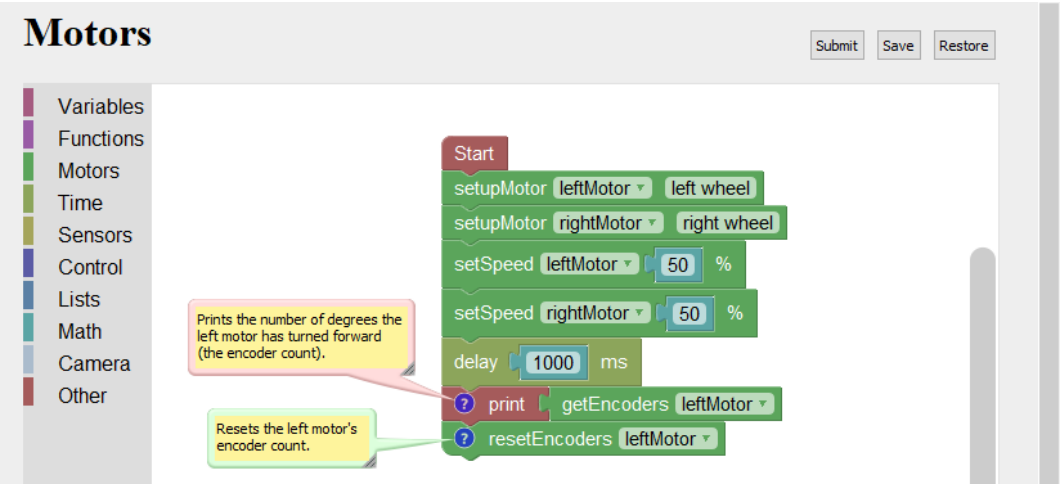
This initializes the motors and allows you to use the *leftMotor* and *rightMotor* variables for other blocks. To move the motors, follow these steps:

1. Drag out two *setSpeed* blocks from the "Motors" category.
2. For the first *setSpeed* block, use the drop-down menu to select *leftMotor*. Then, drag a number block from the "Math" category and use it to fill the hole in the *setSpeed* block. Type 50 into the number block.
3. Repeat step 2 for the second *setSpeed* block, but select *rightMotor* using the drop-down menu.
4. Drag out a *delay* block from the "Time" category. Drag a number block from the "Math" category and use it to fill the hole in the *delay* block. Type 1000 into the number block.



This program will make your robot move forward at 50% of its maximum speed for 1000 milliseconds. Run your program to see for yourself (see Section 5 if unsure how to).

As the motors rotate, they will keep track of the number of degrees they have rotated using an encoder count. This count increases by 1 for every degree the motor has turned forward and decreases by 1 for every degree the motor has turned backward.



6.3 Sensors

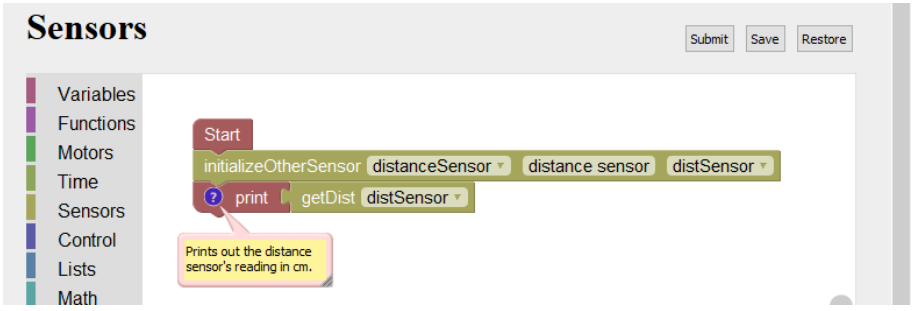
The robot in the world comes with 5 sensors by default:

- Right-facing distance sensor
- Downward-facing light sensor
- Gyro sensor
- GPS
- Forward-facing camera

6.3.1 Distance Sensor

To use the distance sensor, follow these steps:

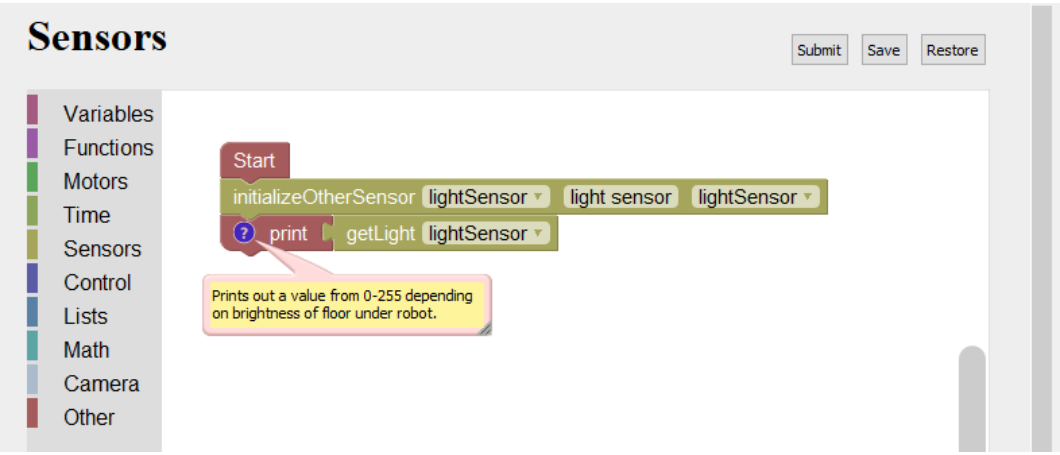
1. Create a variable called *distSensor*.
2. Drag in an *initializeOtherSensor* block from the "Sensors" category. Use the first drop-down to select "distanceSensor". Type "distance sensor" into the second box (omit the quotes). Use the last drop-down menu to select *distSensor*.
3. Drag out a *getDist* block from the "Sensors" category and use the drop-down menu to select *distSensor* to get the distance sensor's reading in centimeters.



6.3.2 Light Sensor

To use the light sensor, follow these steps:

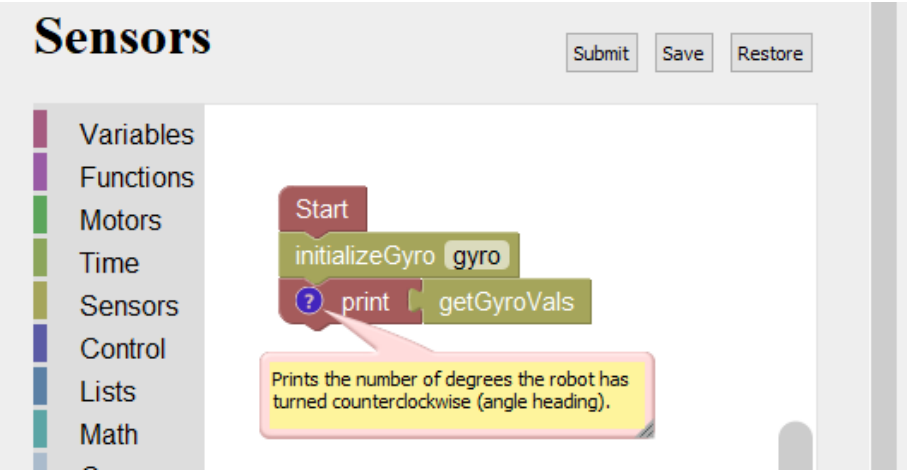
1. Create a variable called *lightSensor*.
2. Drag in an *initializeOtherSensor* block from the "Sensors" category. Use the first drop-down to select "lightSensor". Type "light sensor" into the second box (omit the quotes). Use the last drop-down menu to select *lightSensor*.
3. Drag out a *getLight* block from the "Sensors" category and use the drop-down menu to select *lightSensor* to get the light sensor's reading. The light sensor returns a value from 0 to 255 based on the brightness of the floor below the robot (0 = dark, 255 = light).



6.3.3 Gyro Sensor

To use the gyro sensor, follow these steps:

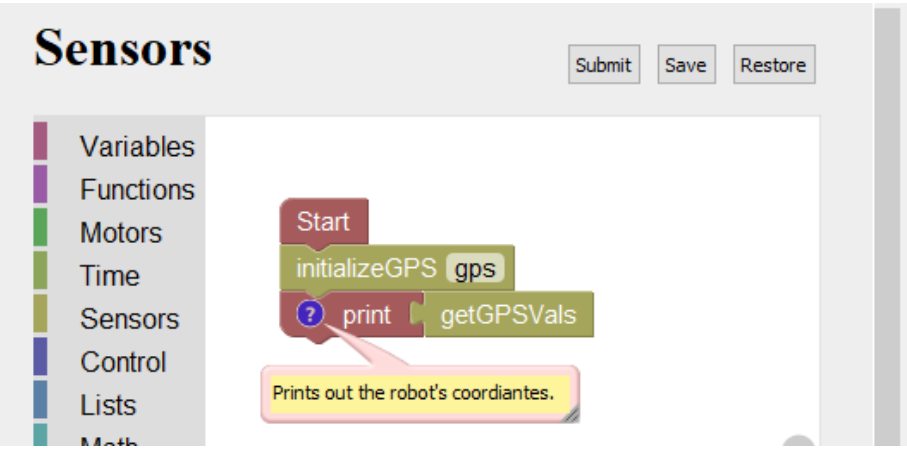
1. Drag out an *initializeGyro* block from the "Sensors" section. Type "gyro" into the space (omit the quotes).
2. Use the *getGyroVals* block to retrieve the robot's angle heading from the gyro sensor. The angle heading begins at 0 and is increased by 1 for every degree the robot turns counterclockwise and decreased by 1 for every degree the robot turns clockwise.



6.3.4 GPS

To use the GPS, follow these steps:

1. Drag out an *initializeGPS* block from the "Sensors" section. Type "gps" into the space (omit the quotes).
2. Use the *getGPSVals* block to retrieve the robot's position from the GPS. The GPS returns a 3-element list containing the absolute XYZ coordinates of the robot in meters.

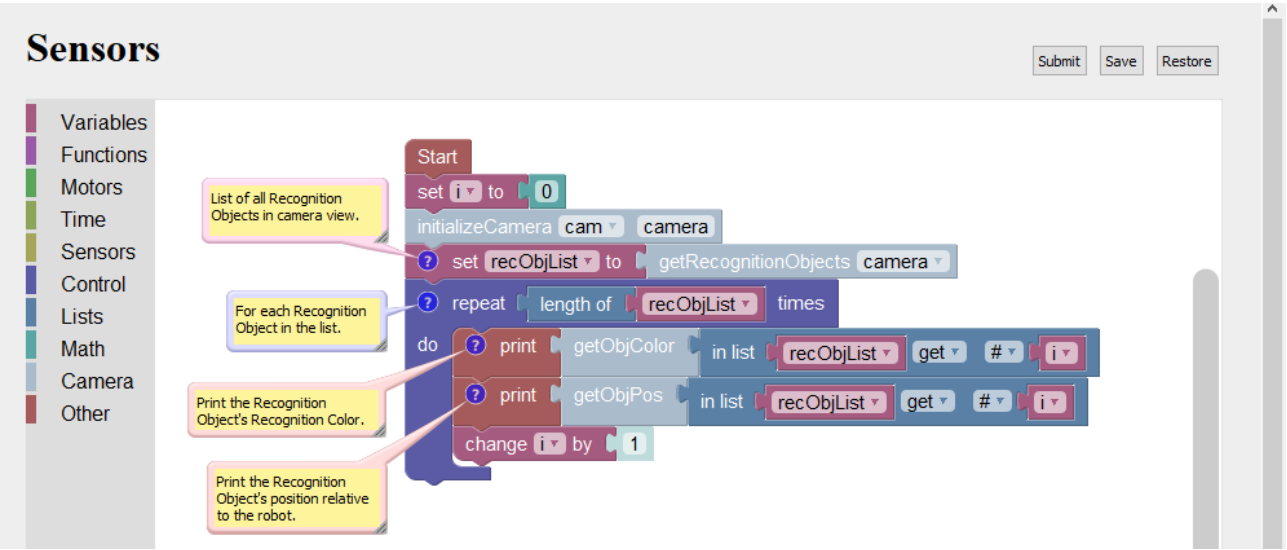


6.3.5 Camera

To simplify computer vision tasks, Webots created a system called Recognition Objects, which is what Webots-Blockly uses. See the full documentation for more information on Recognition Objects. To use the camera, follow these steps:

1. Create a variable called *cam*.
2. Drag out an *initializeCamera* block from the "Camera" category. Use the drop-down menu to select *cam* and type "camera" into the second box (omit the quotes).

3. Use the *getRecognitionObjects* block to retrieve a list containing all the Recognition Objects in the camera's view. Iterate through this list and use the other *get* blocks to retrieve information on each Recognition Object.



7 Sample Code

This tutorial was meant to get you started with Webots-Blockly. For more detailed explanations and additional features of Webots-Blockly, be sure to see the full documentation. In addition, the folder that you downloaded from GitHub comes with several files of sample code. Press the "Restore" button, and any file you did not save yourself is a piece of sample code. They provide excellent demonstrations of using multiple features from Webots-Blockly for a box challenge.

8 Helpful Webots Features - Optional

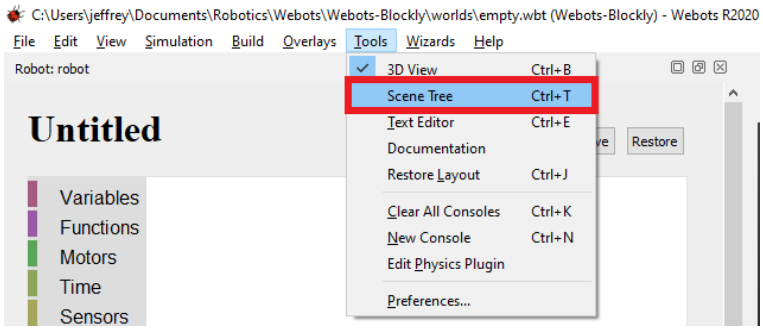
8.1 Moving Objects in the World

To move or rotate objects in the world, first click on them, and red, green, and blue axis lines should appear. To shift the objects, use the arrows at the end of each axis line. You can also hold shift, click the object, and drag it around to move freely horizontally. To rotate the objects, use the circular arrows to rotate about an axis. You can also hold shift, right click the object, and drag to rotate horizontally.

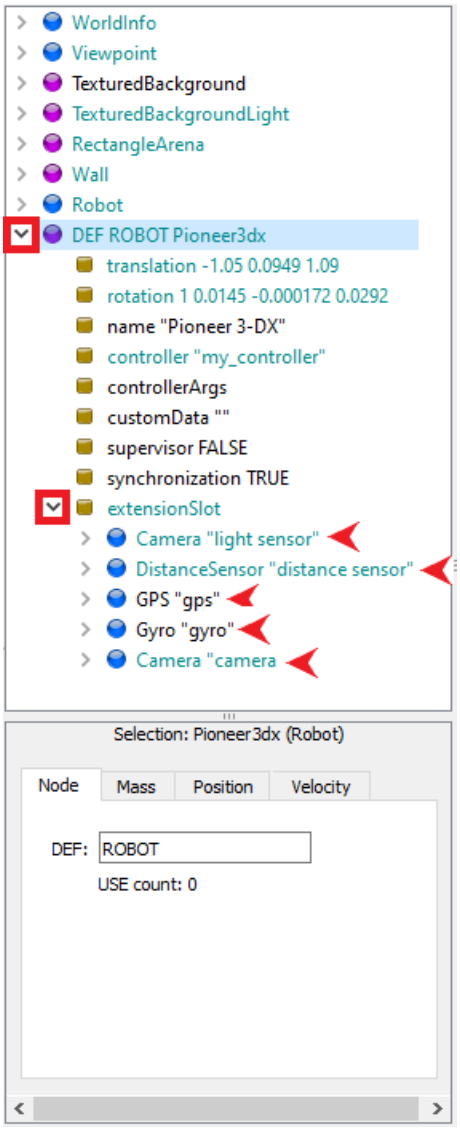
8.2 Sensors

8.2.1 Accessing Sensor Attributes

The sensors on your robot have many modifiable attributes. To access these attributes, first click on the "Tools" option in the top toolbar and click on "Scene Tree".



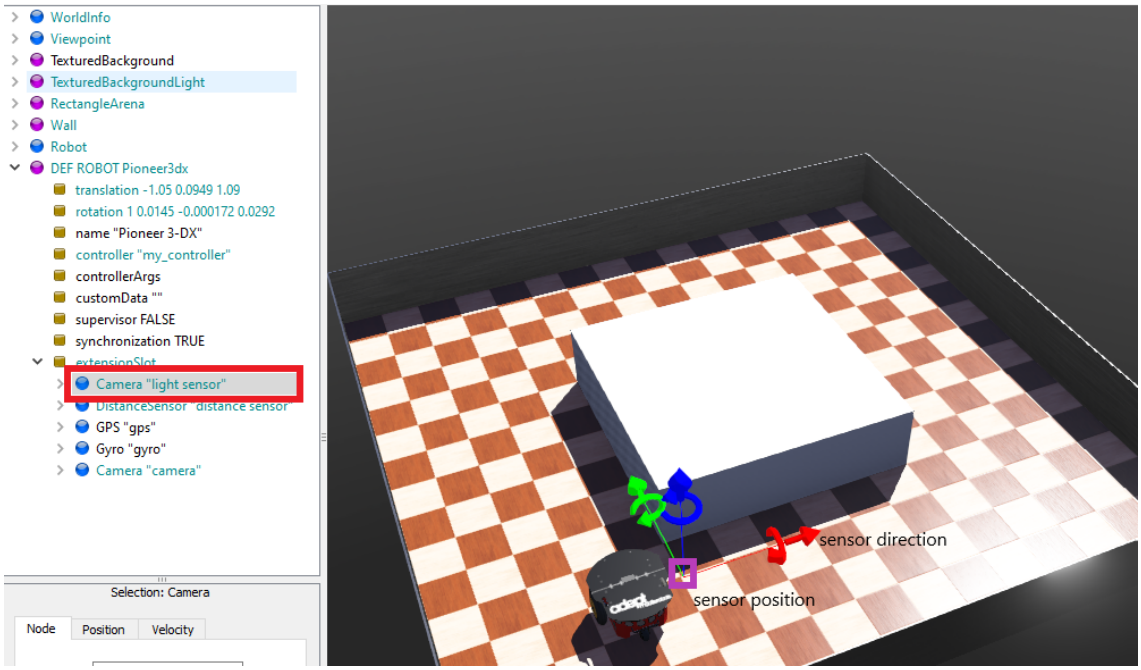
Then, click on the drop-down arrow for the bullet point in the Scene Tree that says "DEF ROBOT Pioneer3dx" (or whatever robot you are using). Next, click on the drop-down arrow for the bullet point that says "extensionSlot." The sensors and their names (in quotes) will be listed under the extensionSlot bullet.



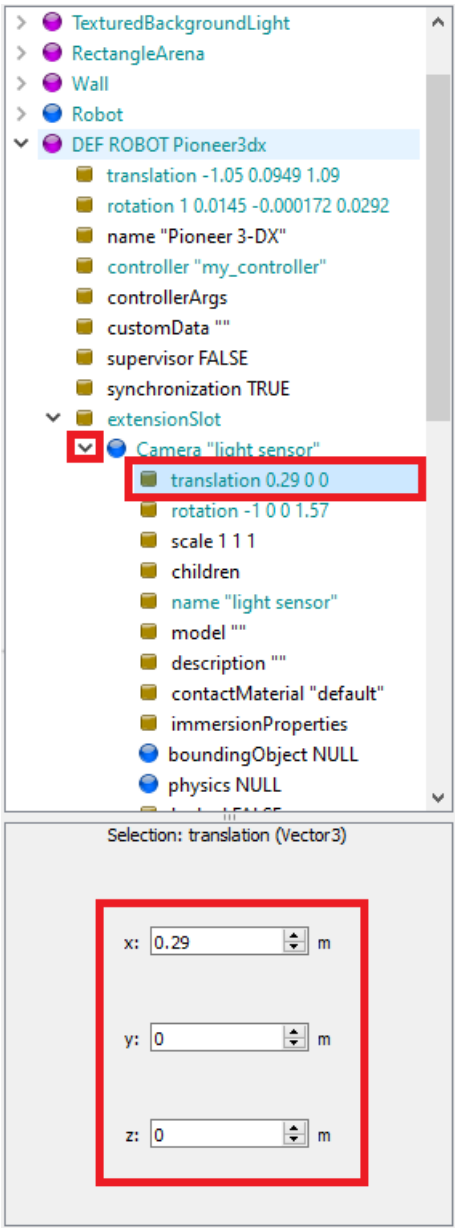
8.2.2 Changing the Position and Orientation of a Sensor

There are two ways to change the position of a sensor or which direction it is located. Note that there are no graphics for the sensors, so you will not be able to see any change in the sensor as you adjust it. Just remember that the sensor is positioned where its axis lines intersect and points in the direction of the red axis line (see next paragraph).

For the easier method, first click on the sensor you want to adjust. In the world, red, green, and blue axis lines should appear. Use the arrows to shift or rotate the sensor (see “Moving Objects in the World” section above).

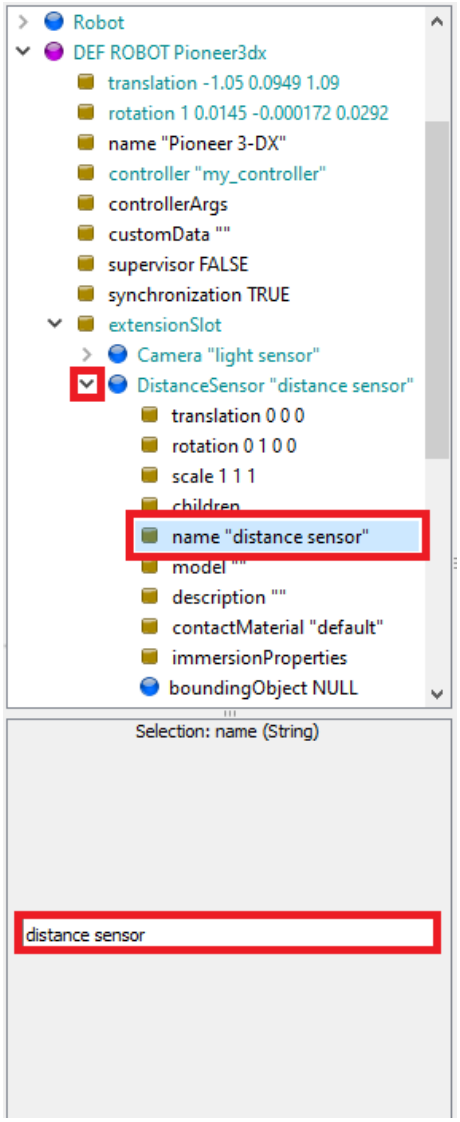


For the more precise method, press the drop-down menu for the desired sensor and click on the proper attribute (translation for position, rotation for direction). Use the window at the bottom to adjust the values.



8.2.3 Changing the Name of a Sensor

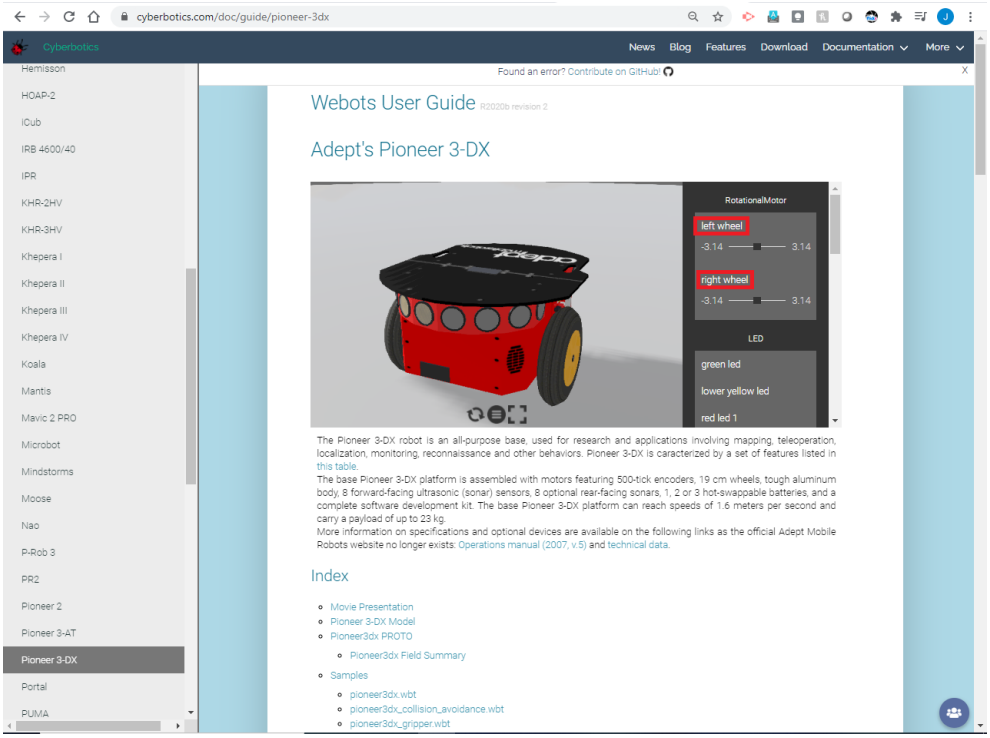
To change the name of a sensor, first press the drop-down arrow for that sensor. Then, click on the bullet point that says “name.” Change the name as you desire using the gray window that appears below.



These are some features that may be useful to you as you become more advanced in Webots-Blockly. However, they are not necessities to the success of your program.

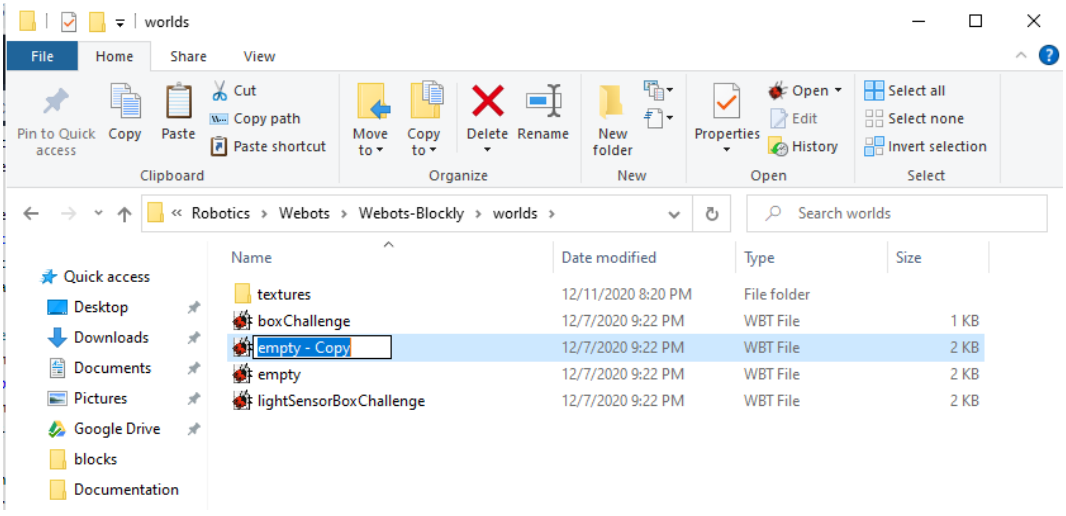
8.3 Finding Name of Motors

The name of the robot's motors can be found by visiting https://cyberbotics.com/doc/guide/robots, clicking on the robot in your world (Pioneer 3dx by default), and the names of the robot's motors will be listed in the first window you see.

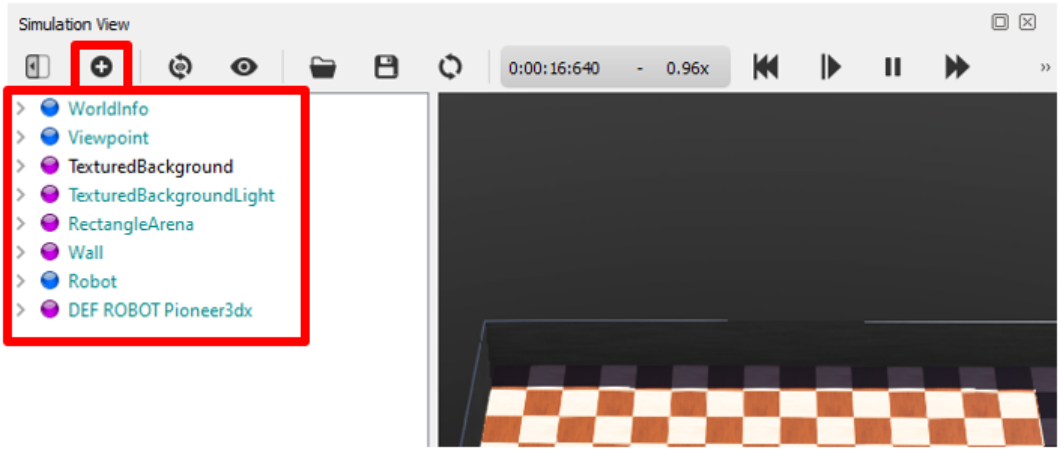


8.4 Creating a New World

To create a new world, navigate to your Webots-Blockly project folder and go into the "worlds" folder. Pick a world file and make a duplicate of that world by copying and re-pasting it into the "worlds" folder. Rename the world as you so choose.



Since this world is a duplicate of a previous one, you may want to remove or add items to your world. Use the "Tools" menu at the top or press "Ctrl+t" to open up the world's Scene Tree. From there, delete objects by deleting bullets from the Scene Tree and add objects using the plus sign above the Scene Tree.


Simulation View
0:00:16:640
0.96x
WorldInfo
Viewpoint
TexturedBackground
TexturedBackgroundLight
RectangleArena
Wall
Robot
DEF ROBOT Pioneer3dx