

## LAB 00 – Getting Set Up (0 marks)

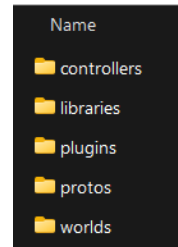
The goal of this lab session is to:

- install and run webots for JAVA, if not already done
- load up a webots world and manipulate objects in it & practicing changing the viewpoint
- compile and run code
- make use of a **switch** statement
- practice submitting your files in Brightspace

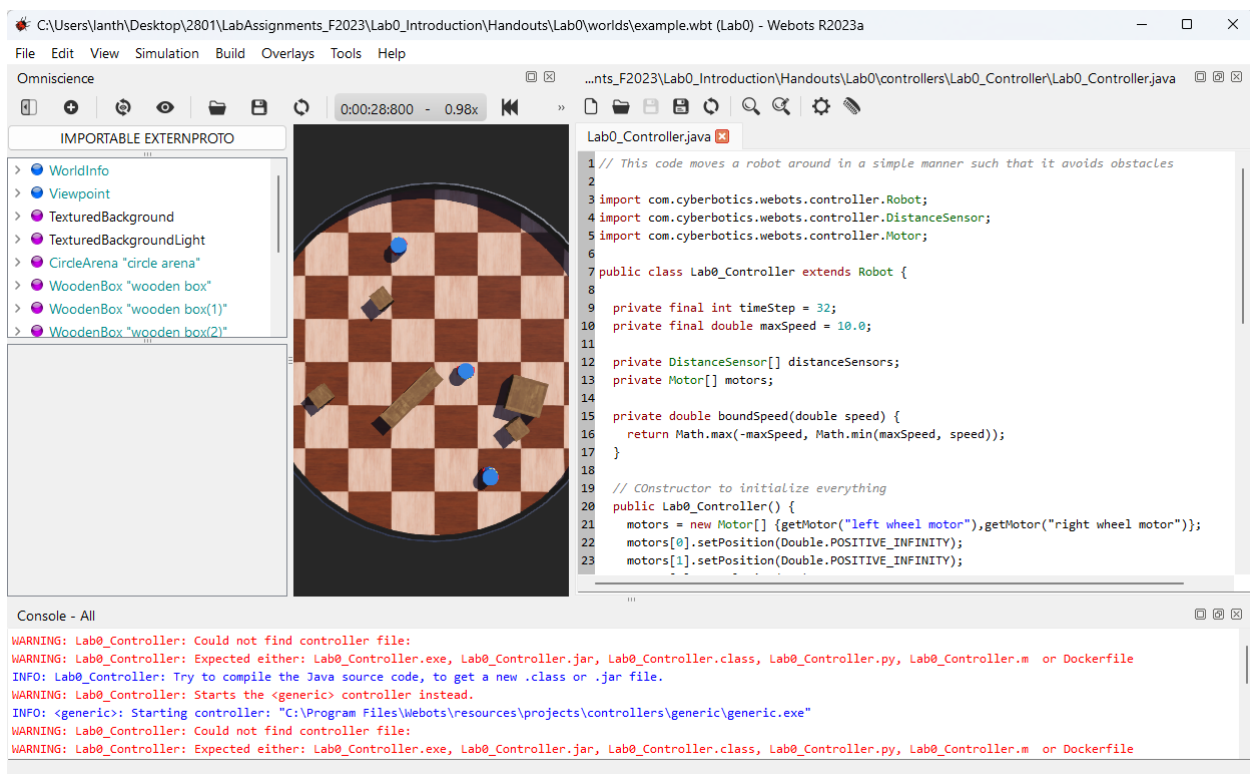
Although this is just a practice lab and is not worth any marks, it will ensure that you are able to complete the remaining labs, as they are all completed in the same manner.



Before starting this lab, you must have completed the **Installation\_Instructions** on the course website. If you have not done that yet, please do that before beginning this lab.

(1) Download the **Lab0\_GettingStarted.zip** file and unzip it. Click on the **Lab0** subfolder. You should see the 5 subfolders shown here →




Make sure that your **Webots** application is not running, close it if it is open already. Go into the **worlds** folder and double-click on the **example.wbt** world file. It should load up the world as shown below ... but there will be errors because the java code has not yet been compiled.



Compile the code with the compile button  and select **Reset** when prompted “Do you want to reset or reload the World?”. The code will likely start running (unless it is paused). You can press the **play** button  to start it. You should see the blue robots moving around. If not ... you will likely want to ask the TA for help.

**(4)** Try each of the following things now:

- pausing/resuming/restarting (see slide **7** in the notes)
- zooming in and out and viewing from different perspectives (see slide **8** in the notes)
- moving objects around manually while running or paused (see slide **9** in the notes)
- editing some of the floor options in the scene tree (see slide **10** in the notes)
- adding objects to the environment (see slides **12 & 13** in the notes)

Don't forget to save the world after you make changes, otherwise it will revert back to the previous world each time that you reset to run again. If you end up messing things up and want to get back to the start, just unzip the lab file again and copy the **example.wbt** world file into the **Lab0/worlds** directory that you were working in to overwrite the messed-up world file. Then reload the world with the reload button .

**(5)** Take a screen snapshot of the world and save it to a file called **Snapshot1.png**. (You can use the camera button). In other labs, you will be taking screen snapshots of some JAVA windows that will pop up. Make sure that you know how to do this on your computer (on a windows machine it is an **Alt-PrintScreen** key press combination which can then be pasted into **Windows Paint** and saved). To practice ... make your Webots application window smaller and take a screen snapshot of the whole window (i.e., NOT the whole screen nor desktop) and save it to a file called **Snapshot2.png**.

**(6)** Delete two of the robots by selecting them individually and pressing the delete key. Put a comment at the top of the **Lab0Controller.java** file with your name and student number like this: **// Author: JOHN DOE (SN: 100123456)**. Look at the **run()** procedure. Notice there is a **while** loop that repeats forever (or at least until the simulation is done). This "infinite" **while** loop will be VERY common in our robot code. We will insert some code to display text explaining what the robot is doing. Add the following code to the bottom of (but still inside) the **while** loop:

```
// Display what the robot is doing
int speedDiff = (int)(leftSpeed+0.5) - (int)(rightSpeed+0.5);
switch(speedDiff) {
    case 0:
        System.out.println("Moving Forward");
        break;
    case 2:
        System.out.println("Curve right");
        break;
    case -2:
        System.out.println("Curve left");
        break;
    default:
        System.out.println("Avoid Obstacle");
        break;
}
```

A **switch** statement is like choosing one of a set of **IF** statements based on a switch number (i.e., **speedDiff** variable in this case) and then evaluating it ... up to the **break** statement. In this code, we display a string based on the difference in the wheel speeds. We can actually write many lines of code for each **switch** case, but here we just wrote one. In general, all cases must end with a **break;** statement.

Compile and run the code. You should see the appropriate message appear as the robot moves around. Make sure that you FULLY understand how to use a **switch** statement, since we will use **switch** statements throughout the course.

In **Brightspace** ... submit your **Lab0Controller.java** code, your **example** world file and your **2 Snapshot.png** files when you are done. Make sure that your name and student number is in the first comment line.