**Feedback**

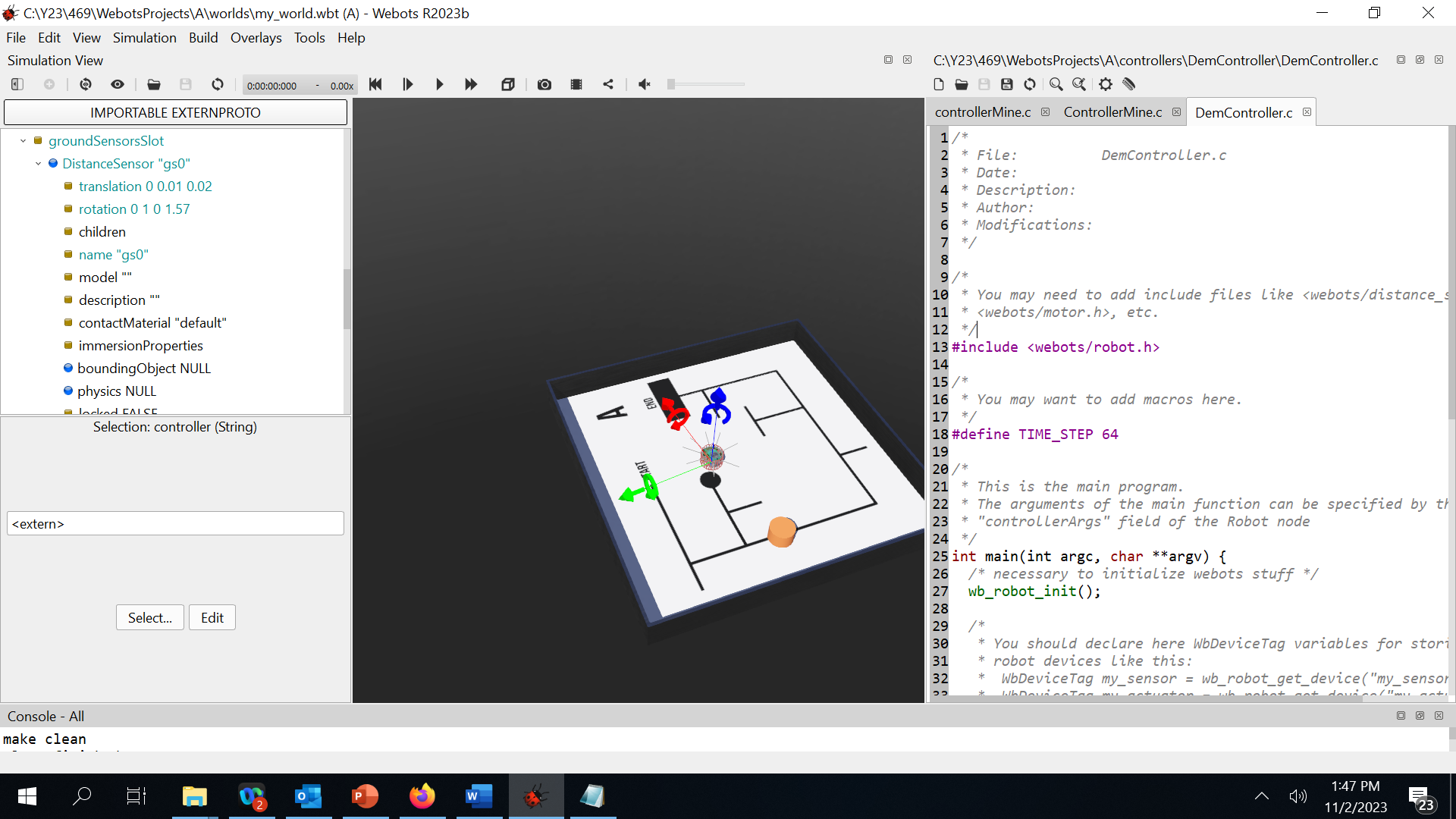
**Content:**

1. How to add your own controller program to the webots project
2. Use of the webots sample controllers
3. Use of e-puck three ground sensors proto
4. Robot motion coding tips
5. Left Hand on the Wall Algorithm
6. **How to add your own controller program:**

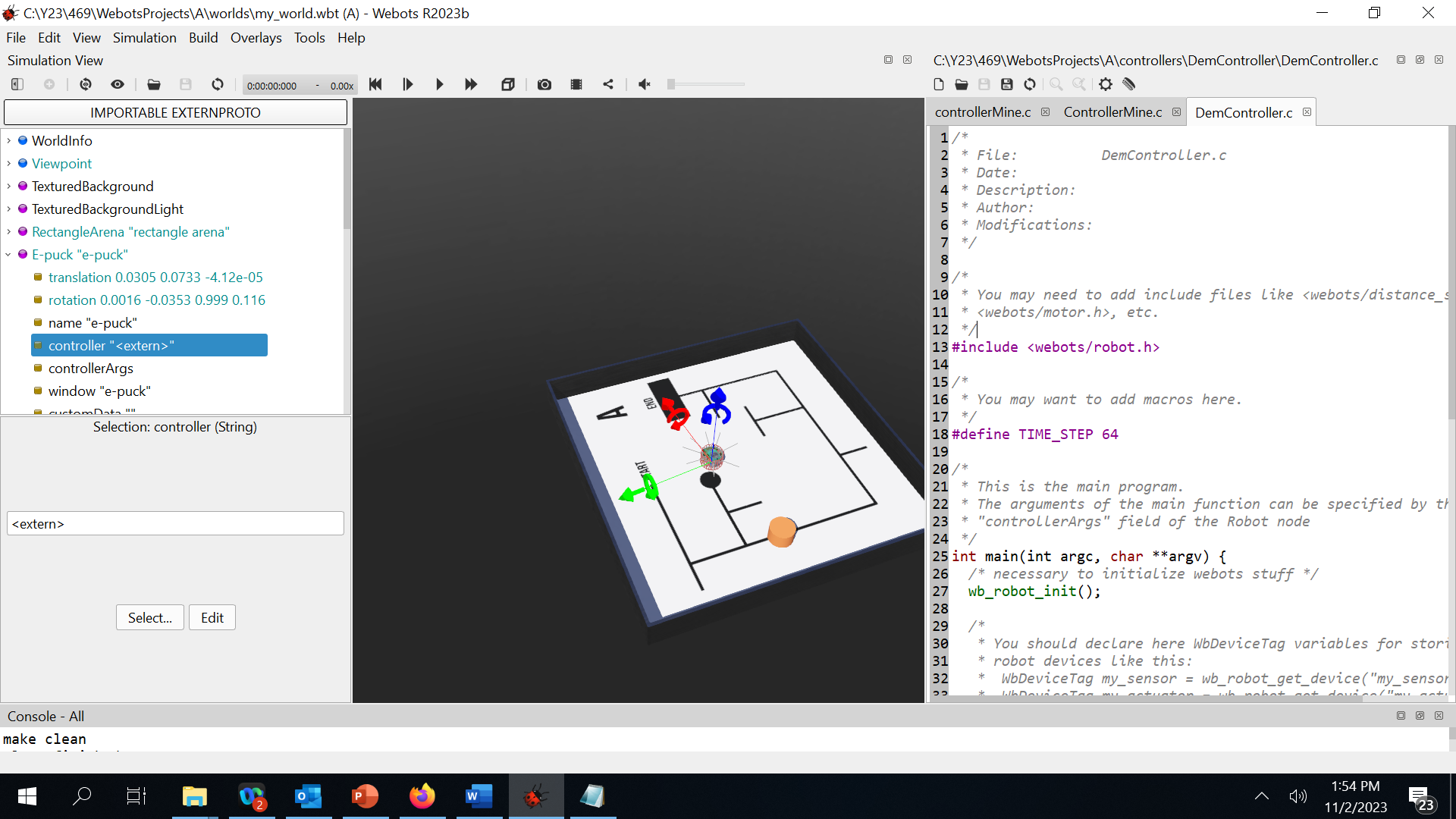
Follow the actions order below, row by row, from left to right

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

A template code will be automatically added to your named controller program. The code is displayed under a dedicated named tab at the right side of the IDE, see below.

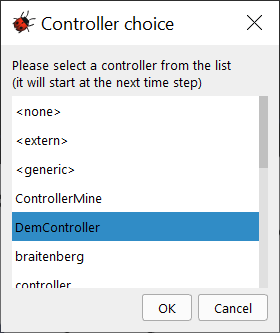


This controller program can now be added from the “Controller Choice” window. To do so, expand the e-puck/robot node from the left side of the panel and select “controller”



**Figure 1**

Click on “Select” button from the below panel in figure 1. Your controller program (DemoController.c in this case) should be available in the list of controllers (scroll down if needed):



If you select another sample controller program, click on “Edit” button in Figure 1; copy-paste the controller code in another editor (VScode, notepad, etc..) to save the program on your computer for easy future reference.

1. **Use of webots sample controllers**

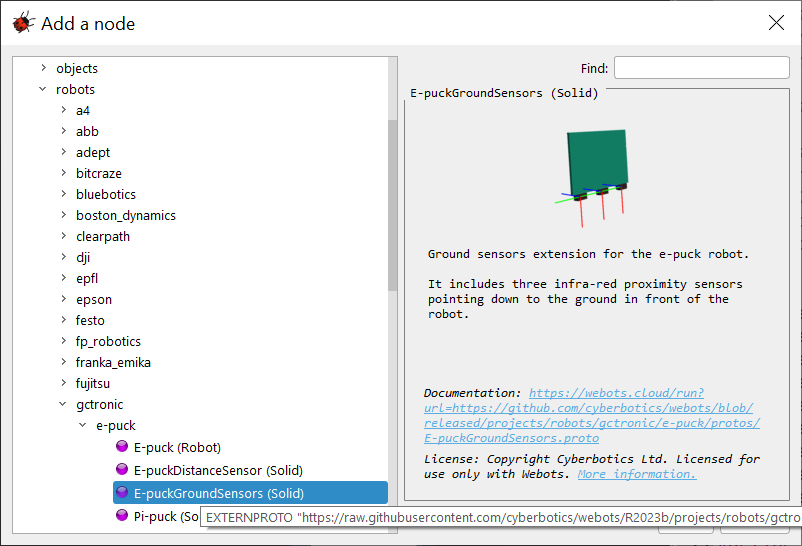
You can edit any of the webots sample controller; however, you have to acknowledge the source. Having said that, some of the sample controllers might have considerable irrelevant code to the project.

In the past weeks, I uploaded a much simpler controller program to an e-puck robot with one ground sensor. The controller prints the value of the ground sensor reading when the robot moves on white/black surface. The controller code is attached to this document although it is already part of the [webots project](https://moodle.uowplatform.edu.au/pluginfile.php/4364509/mod_folder/content/0/WebotsController.zip?forcedownload=1) available in Moodle which you should at least run.

1. **Use of e-puck three ground sensors proto**

We explained in [SensorsConfiguration.docx](https://moodle.uowplatform.edu.au/pluginfile.php/4364509/mod_folder/content/0/SensorsConfiguration.docx?forcedownload=1) how to add a ground sensor. However, as reported by a student, you can add a three ground sensors as follows.

Select “groundSensorsSlot” node from the e-puck/robot node in the left panel; right click and select “Add New”, expand “Proto nodes (webots project)”, then “robots”, then “gctornic” and select “E-puck GroundSensors(solid)”.



However, you’ll highly likely have to move the positions of the ground sensors as per the need of the width of black lane in your assigned project track; see [SensorsConfiguration.docx](https://moodle.uowplatform.edu.au/pluginfile.php/4364509/mod_folder/content/0/SensorsConfiguration.docx?forcedownload=1).

The C statements to control the motion of the robot are detailed in [RobotControllerCoding.docx](https://moodle.uowplatform.edu.au/pluginfile.php/4364509/mod_folder/content/0/RobotControllerCoding.docx?forcedownload=1) (it is the same file which was under the webotsController.zip file under the [supportFolder](https://moodle.uowplatform.edu.au/mod/folder/view.php?id=3306507) ). I would recommend to create four functions for forward, reverse, right, and left movements which you can call to drive the robot on the track.

1. **Robot motion coding tips:**

At least [RobotControllerCoding.docx](https://moodle.uowplatform.edu.au/pluginfile.php/4364509/mod_folder/content/0/RobotControllerCoding.docx?forcedownload=1) and webots sample controller programs.

1. **Left Hand on the Wall Algorithm:**

The [project specification](https://moodle.uowplatform.edu.au/mod/resource/view.php?id=3298452) gives a high level but sufficient description of the algorithm with two additional useful references. With this algorithm, at any intersection a left movement is always prioritised.

The first [Reference](https://patrickmccabemakes.com/tutorials/Maze_Solving/) explores a number of possible intersections and the corresponding robot motion as per the algorithm; the [second reference](https://core.ac.uk/download/pdf/229655734.pdf) gives an example regarding the positions of the ground sensors to solve a particular track.

You need to consider carefully the minimum ground sensors you have to add to the e-puck robots and their positions in order for the robot to move on the track, well aligned, avoiding any obstacle until it reaches the end of the track.