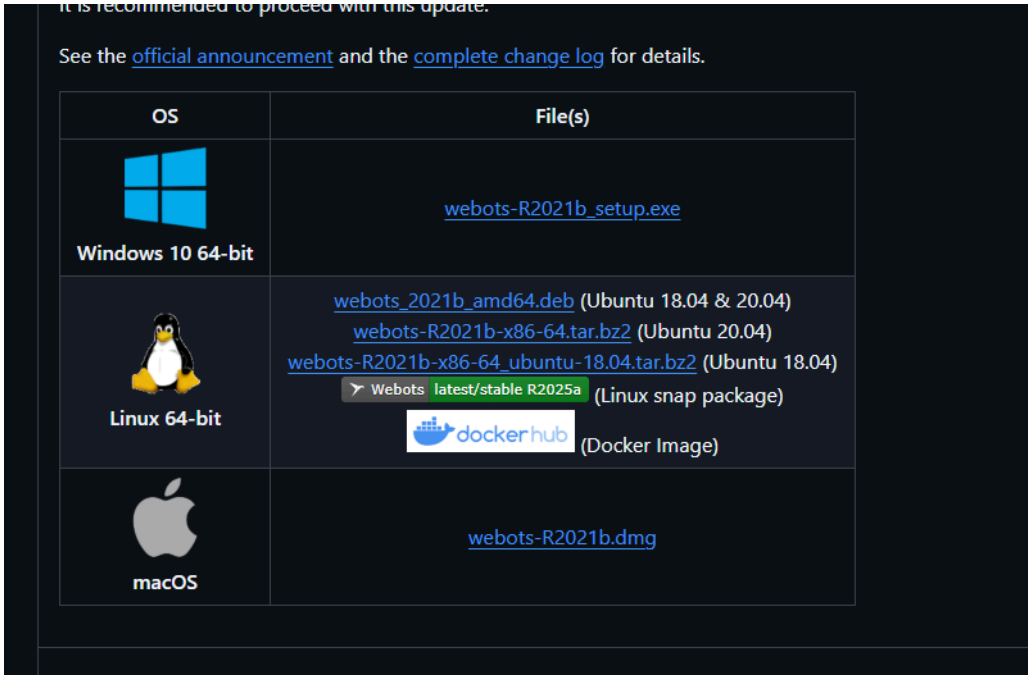
Giovanni Argueta

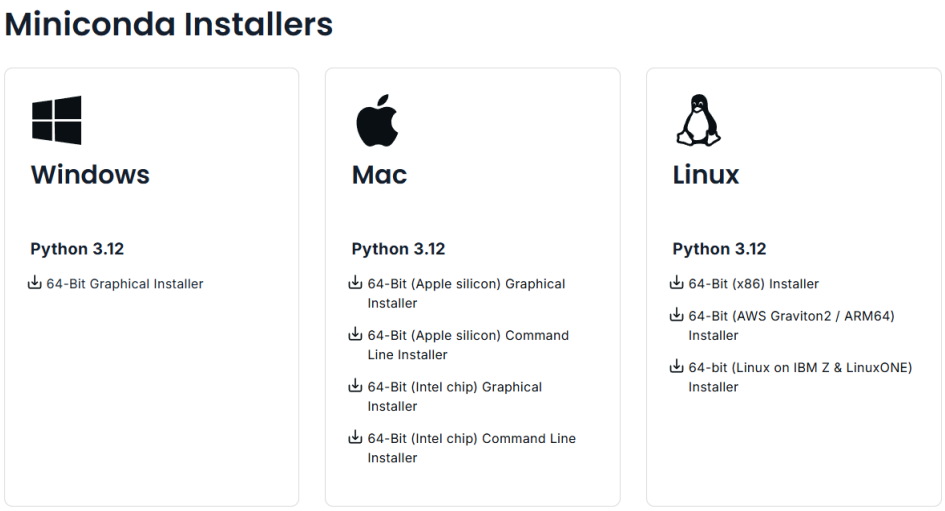
Running Webots simulation

github : https://github.com/giovanniargueta1/webots_roomba_sim/tree/main

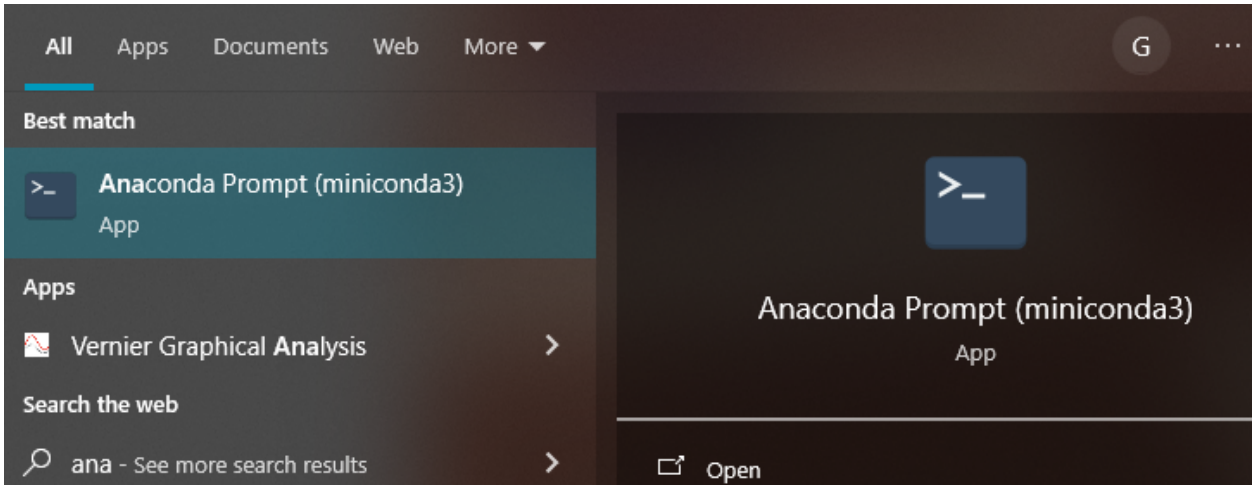
1. Download the Webots simulator version R2021B, found here(scroll down)- https://github.com/cyberbotics/webots/releases, for Windows (or your OS)



2. Download miniconda found here (might have to sign up with email), this will provide you a terminal to create a python virtual environment- https://www.anaconda.com/download/success



3. After successfully installing miniconda, you should have access to their terminal as seen here called “Anaconda prompt”; Open it



4. In order to link our python installation to webots, we will create our own virtual environment with 3.8 with open cv. First type the following command into the terminal (It does not matter where your terminal is pointing to) (-n opencv2 names this new environment, call it whatever you like)

```
(base) C:\Users\Geo Argueta>conda create -n opencv2 python=3.8
```

5. Then **after waiting** for it to install, confirm it was create by typing the following command, and seeing it on your list, then activate it by doing conda activate (the name), you will see the (base) tag turn into the name of the new environment

```
(base) C:\Users\Geo Argueta>conda env list
# conda environments:
#
base                  *  E:\miniconda3
cortex_api               E:\miniconda3\envs\cortex_api
opencv                   E:\miniconda3\envs\opencv
opencv2                  E:\miniconda3\envs\opencv2
spacy-llm                E:\miniconda3\envs\spacy-llm


(base) C:\Users\Geo Argueta>conda activate opencv2

(opencv2) C:\Users\Geo Argueta>
```

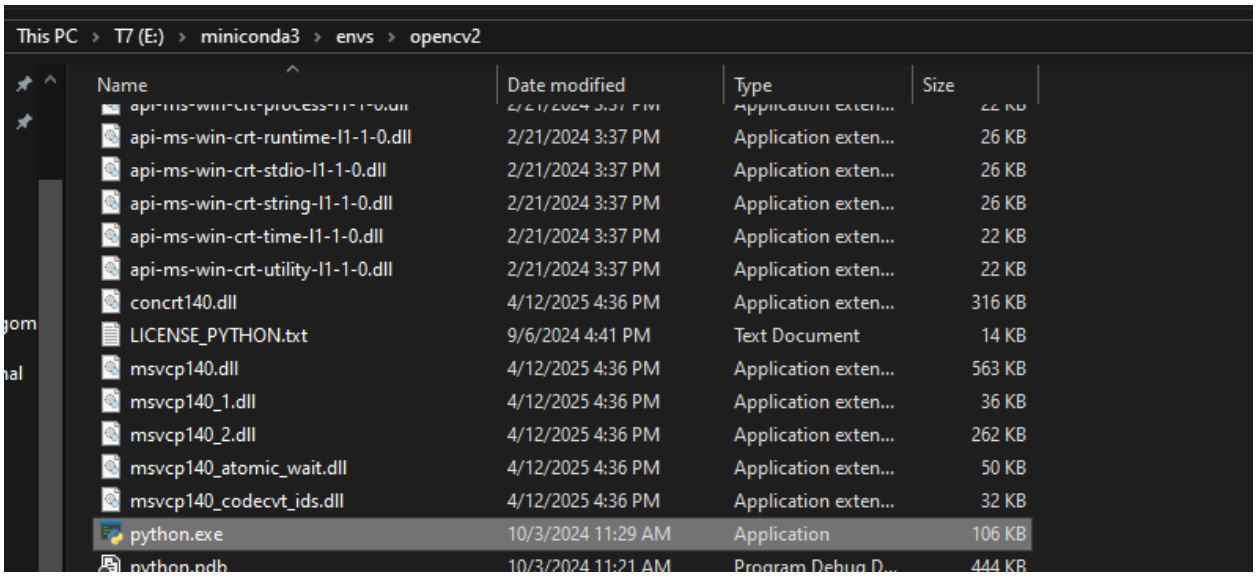
6. Now once the environment is activated type in the following command to install opencv-

```
(opencv2) C:\Users\Geo Argueta>pip install opencv-contrib-python
```

Then after installation(**wait**), type conda list and confirm you have the same libraries as this:

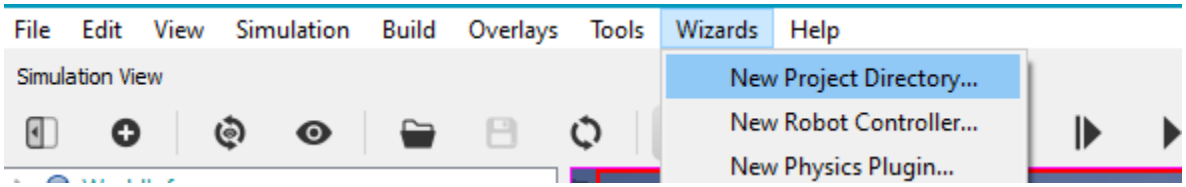
```
(opencv2) C:\Users\Geo Argueta>conda list
# packages in environment at E:\miniconda3\envs\opencv2:
#
# Name                    Version                   Build  Channel
ca-certificates           2025.2.25            haa95532_0
libffi                    3.4.4                hd77b12b_1
numpy                     1.24.4                   pypi_0    pypi
opencv-contrib-python     4.11.0.86                pypi_0    pypi
openssl                   3.0.16               h3f729d1_0
pip                       24.2              py38haa95532_0
python                    3.8.20               h8205438_0
setuptools                75.1.0            py38haa95532_0
sqlite                    3.45.3               h2bbff1b_0
vc                        14.42                haa95532_5
vs2015_runtime            14.42.34433          hbfb602d_5
wheel                     0.44.0            py38haa95532_0
```

7. You can now exit the terminal, now we find the location of this new python executable. Look under the drive you installed miniconda (most likely C: drive), and find the folder called “miniconda3”, now head over to “envs”, and then the name of the environment you just created. Under this folder, there will be a python.exe, keep the location of this file on a notepad or remember its location, we will use it shortly. For example in this case I would record E:\miniconda3\envs\opencv2\python.exe as my location for this env.

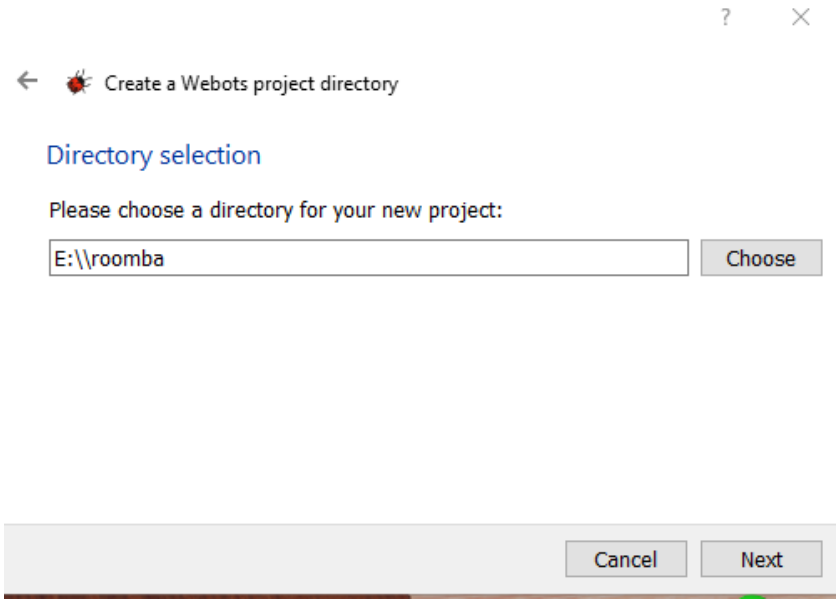


8. Now download the “Object Rocognition Robot.wbt” file from my folder-SIM_ENV/world/ — this will be the world file that loads in the environment in Webots (DO NOT OPEN yet in Webots)

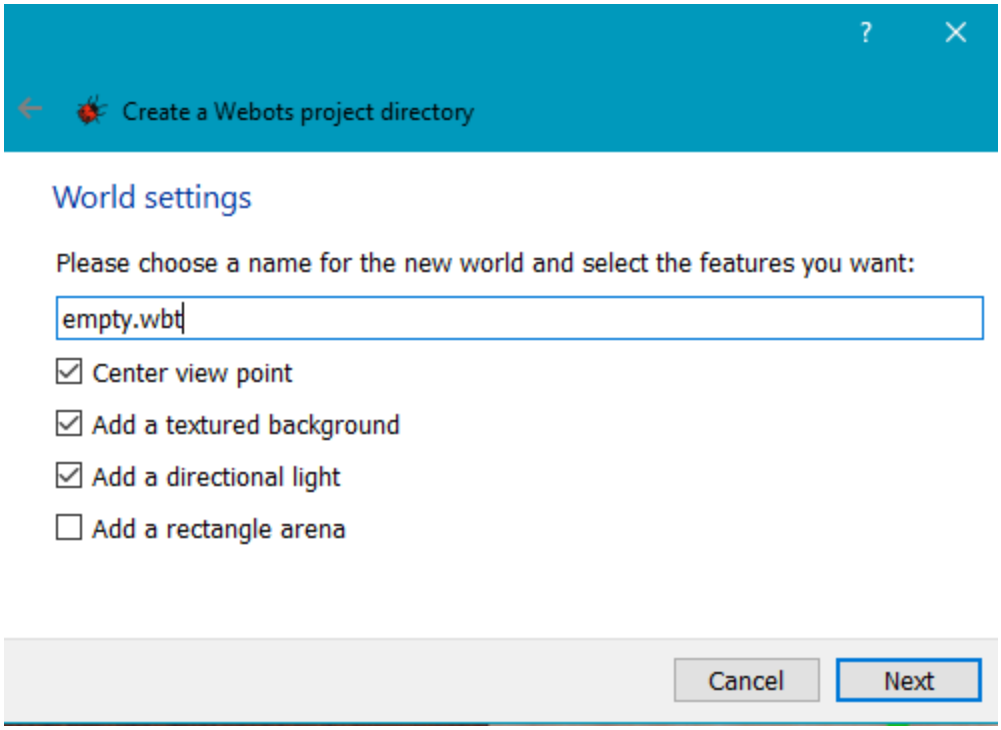
9. Now open the webots application, you may be prompted to start an empty world but ignore it or any tutorial. Simply head over to the “Wizards” tab found in the top left corner of the screen and click on new project directory.



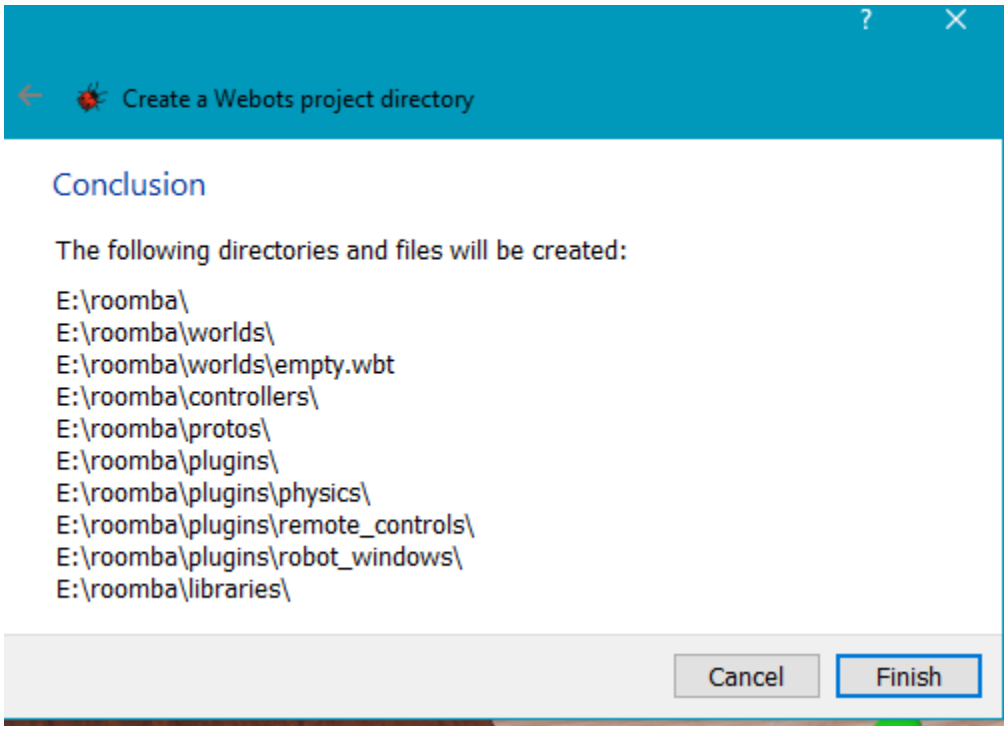
Then choose a location you can find (ideally in the same drive as where you installed webots) and call it what you like, (here I called it roomba but later I will show you I actually called it capstone)



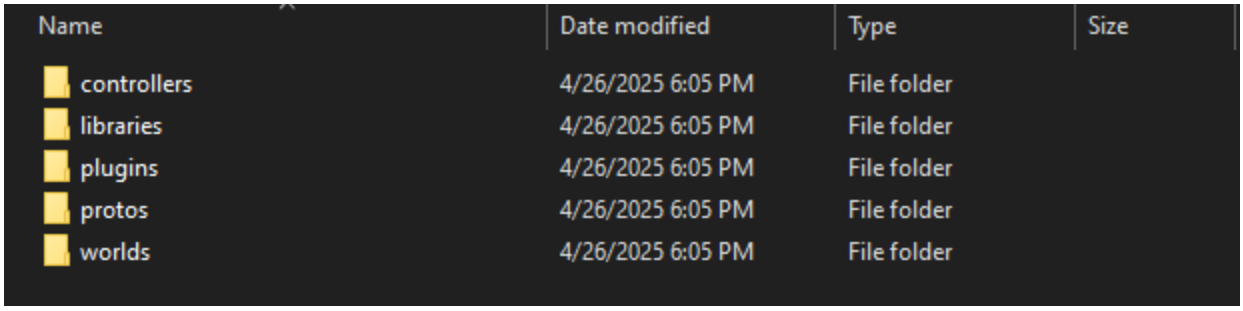
After hitting next , you will see the following screen please ignore as we will use another .wbt file, just hit next.



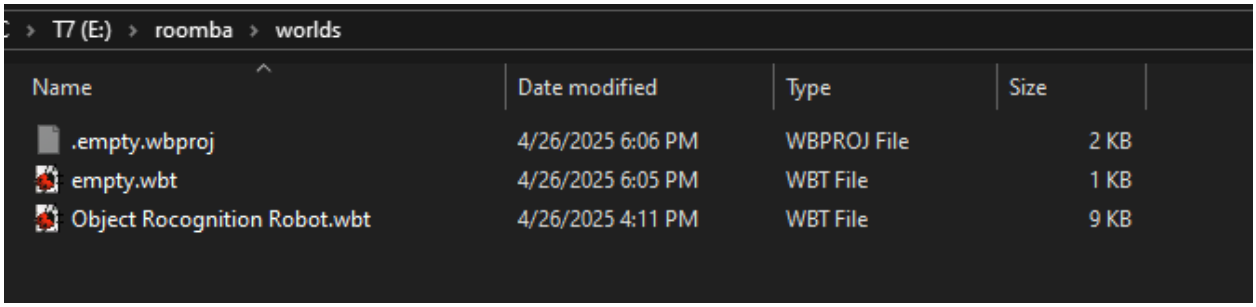
You should see this screen, hit finish.



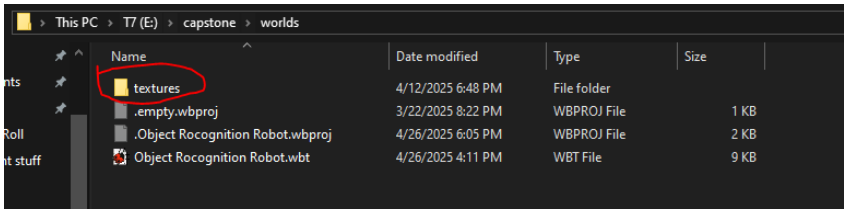
10. After hitting finish, exit the application. Find that new parent folder you created that has the following structure:

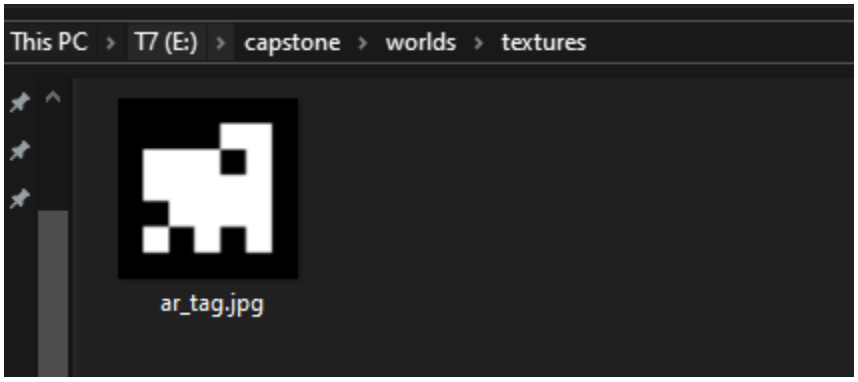


Then head over to worlds, where you will see the empty.wbt you created. You can delete it for now, or simply ignore it. Now head over to the “worlds” folder and copy and paste that “Object Rocognition Robot.wbt” file on the github you downloaded earlier so your world folder looks like this:

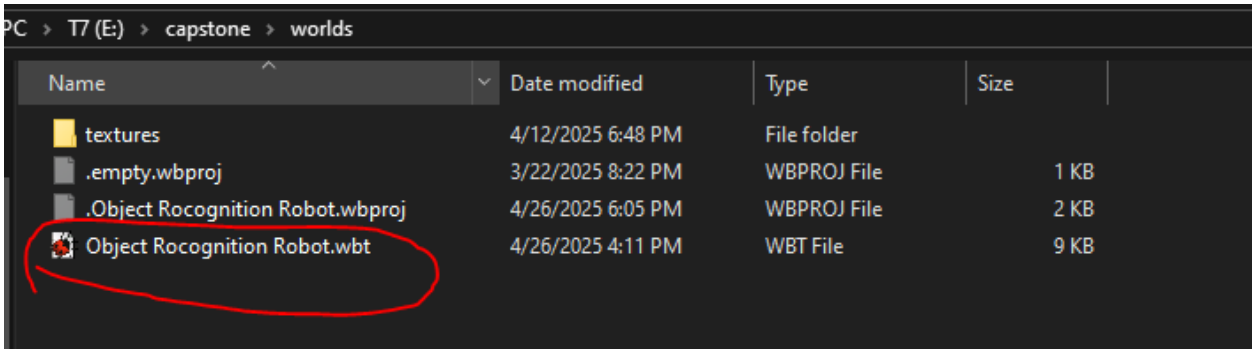


11. Now in that same folder, create a folder called “textures”. In that folder, please download the “ar_tag.jpg” file found at SIM_ENV/world so the folder looks like this

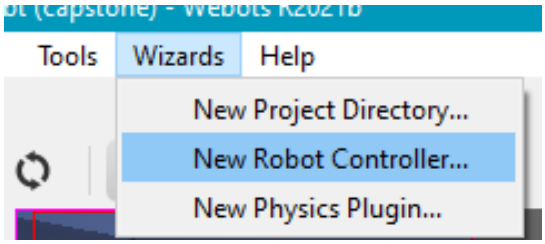




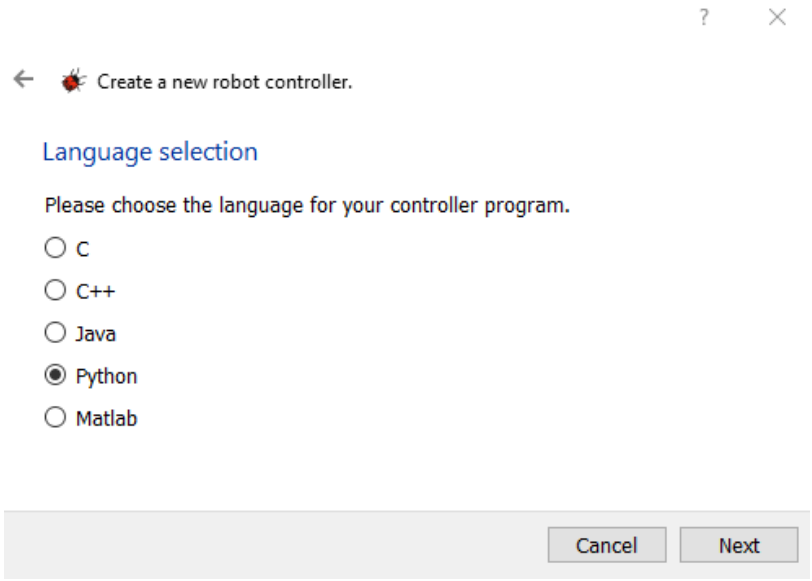
12. Now to open up our environment, simply click on the “Object Recognition Robot.wbt” you pasted in the worlds folder.



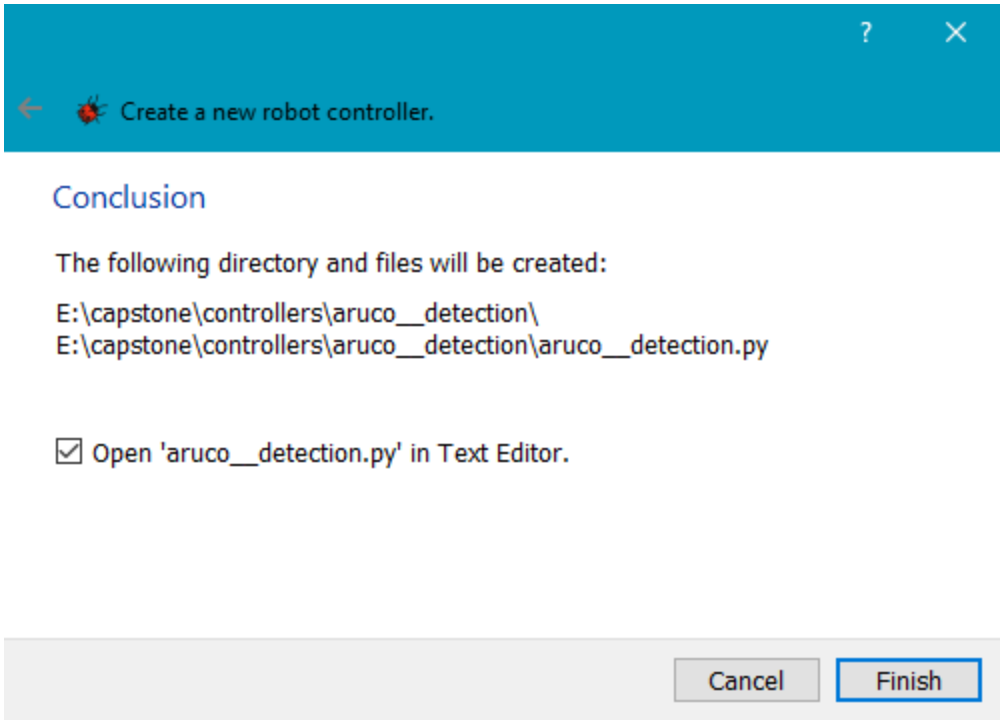
13. You will most likely see in the console (bottom of the screen) an error message saying there is no controller. We will have to create one, first find this under wizards-



After clicking on “New Robot Controller”, hit next and you should be met with this screen, please choose “Python”.

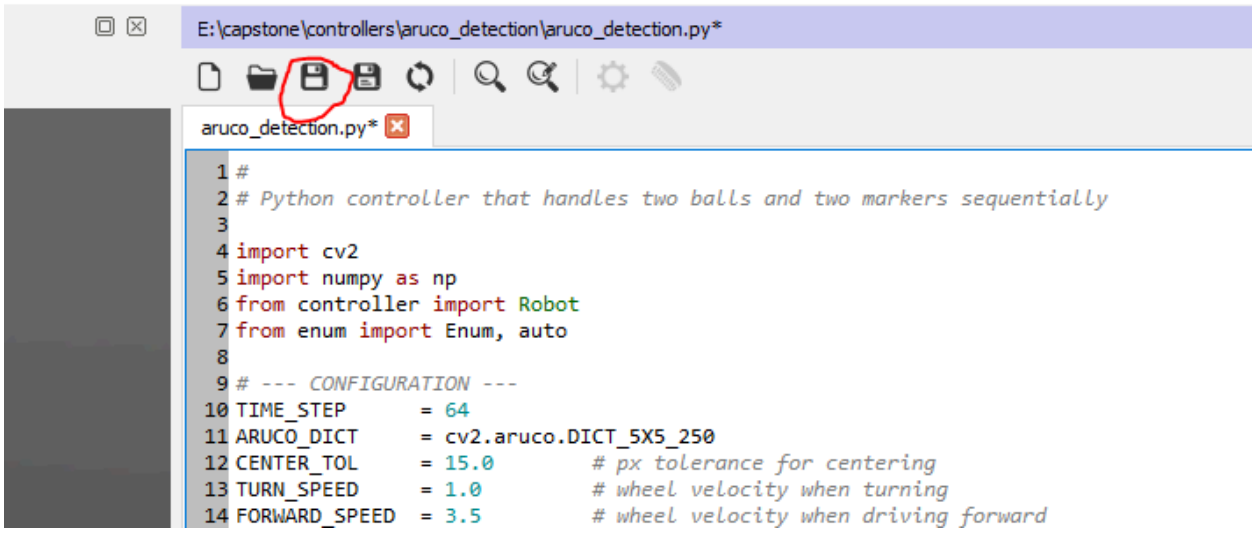


After hitting next, we have to name our controller. We will test my non-pid/non-q learning controller, please call it “aruco_detection”. After hitting next, you should see this screen-

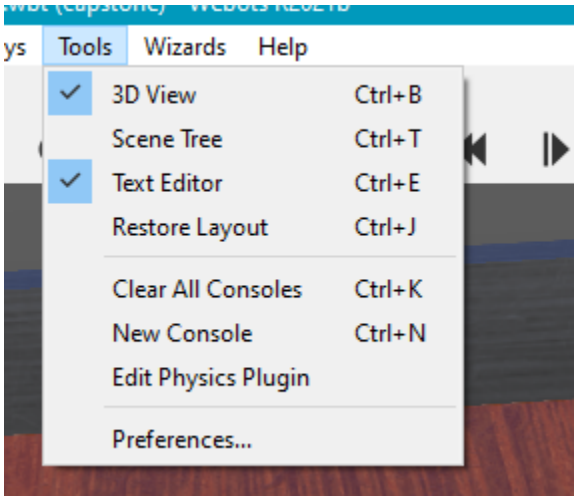


Click finish and notice how the text editor opens on the right side.

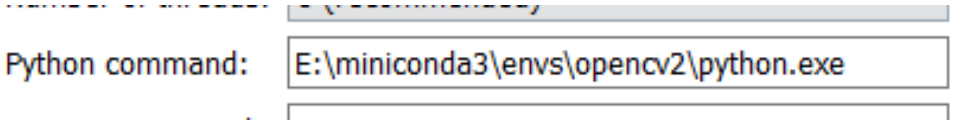
14. Now head over to SIM_ENV/code, open up the file named “aruco_detection.py”, you could download it to that folder but it is easier if you copy its raw content (aka the code) and paste it in the text editor in Webots, then click save (as seen by the red circle).



15. The simulation is almost ready, now under “Tools” click on the “Preferences” option,

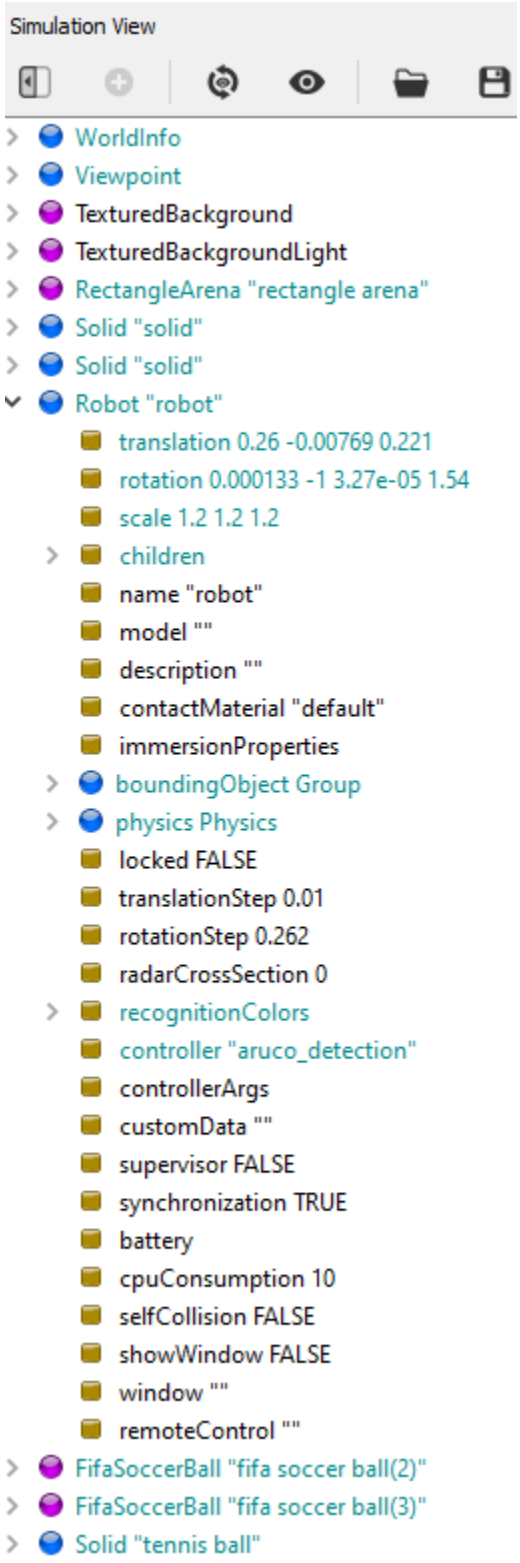


Under preferences please refer back to your python.exe location and paste it in the “Python command” section.

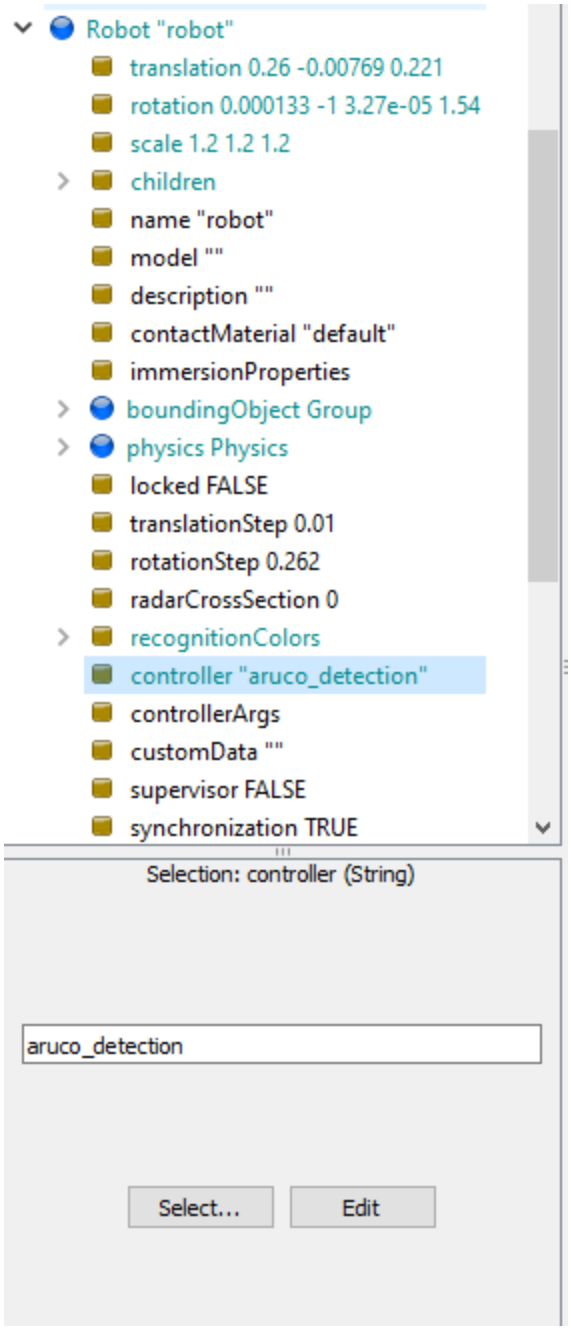


16. Final step, now we equip the robot with said controller. Please head over to the left side of the screen under the Simulation view. Take notice to the blue node called

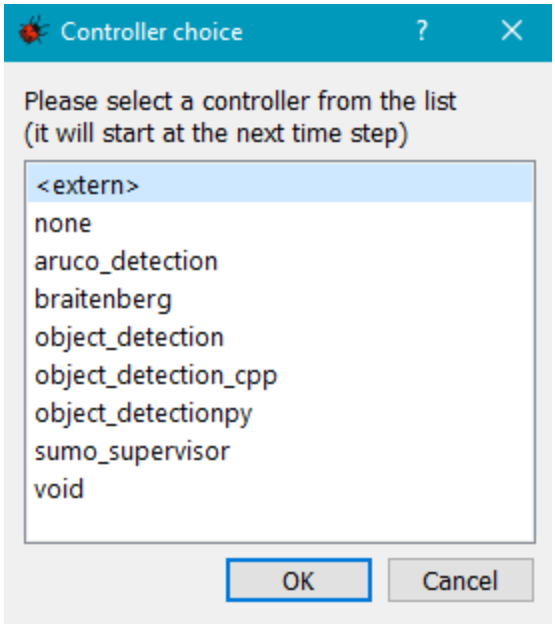
“Robot”, please click on the arrow to expose its underlying contents-



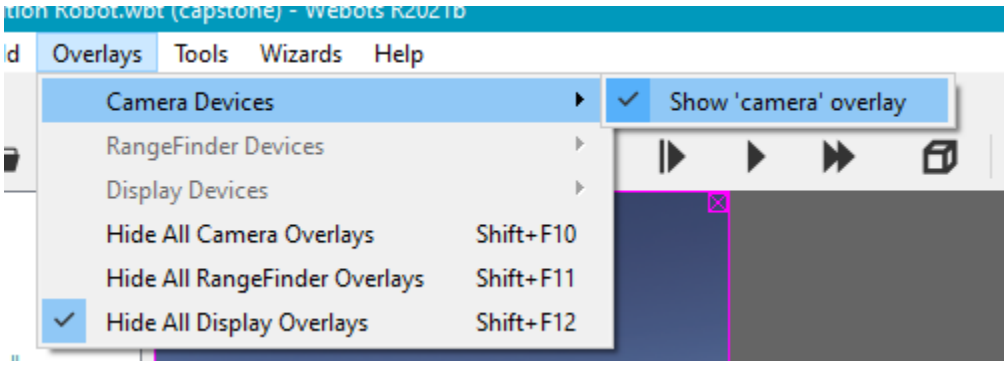
Now click on the tab called “ controller “..” “ right under recognition colors



Select your controller you created (in this case aruco_detection) and hit ok.



17. Now you can finally play the simulation(*please exit the application after completing the previous step and reopen the wbt file*) Before starting if you want the camera overlay on head over to “Overlays” and click the following option:



Please take note of the play buttons seen here:

The left most button restarts the simulation. The third button plays and pauses the simulation.