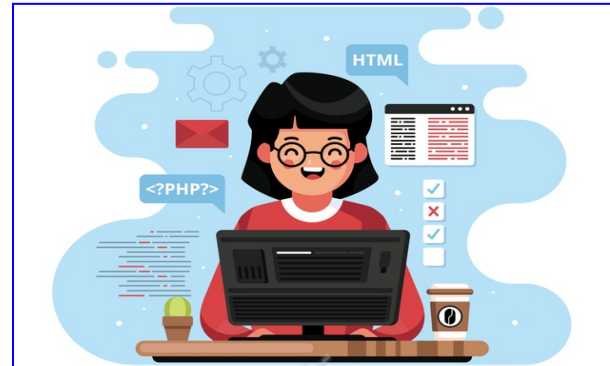


Ramp Follower Robot - IV



What is our GOAL for this CLASS?

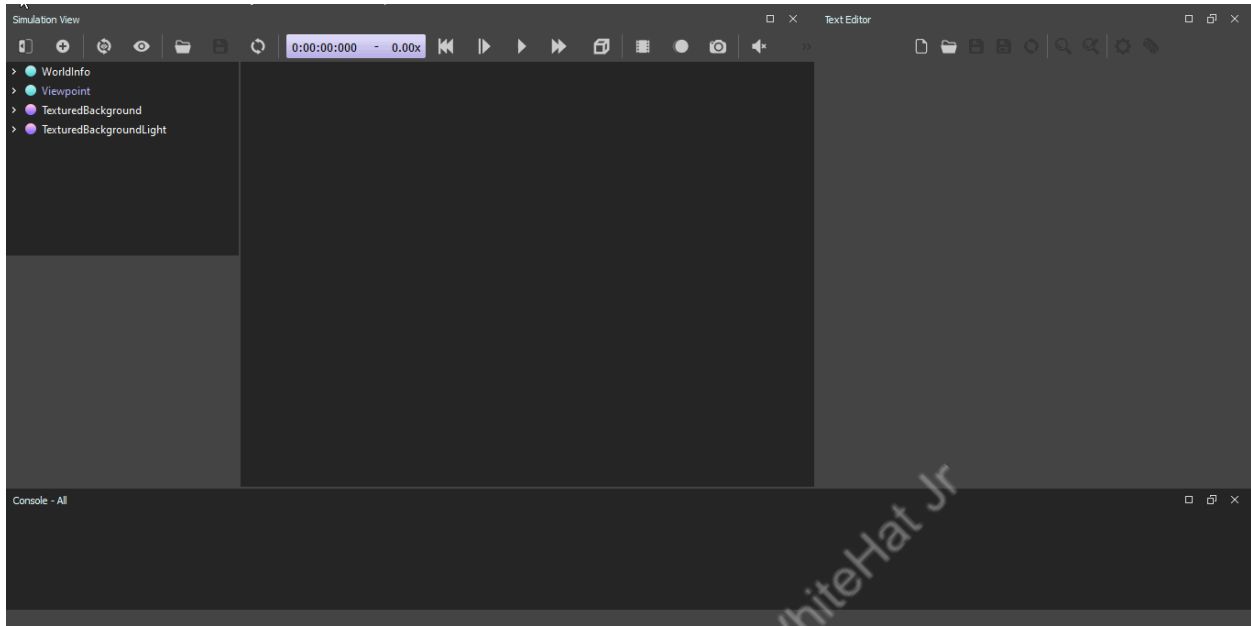
In this class, we designed the controller of the Robot.

What did we ACHIEVE in the class TODAY?

- We learnt about controllers.
- We learnt about robot programming to create a controller.
- We learnt to integrate the controller to a robot device.

How did we DO the activities?

1. **We learnt about controllers.**
 - A controller is a program that defines the behavior of a robot.
2. **We learnt about Robot programming and how to create a robot controller using Python:**
 - a. Create a new Python controller using the Wizards / New Robot Controller...



<https://s3-whjr-curriculum-uploads.whjr.online/0424480f-a401-4052-9cc8-dbb7ce737a4c.gif>

b. Steps to code:

- i. At the beginning of the controller file, we imported the libraries corresponding to the Robot.

Reference Code:

```
from controller import Robot
from controller import Motor
from controller import Altimeter
from controller import LED
import math
```

- ii. **Main function:**

The main function is where the controller program starts execution. We created our own class MyController and created an object of it and used its run method where we will implement the code to start the execution.

Reference Code:

```
controller = MyController()
controller.run()
```

The Webots API has to be initialized using the `__init__` function

Reference Code:

```
super(MyController, self).__init__()
```

The basic time step is the time step increment used by Webots to advance the virtual time and perform simulation. This duration is specified in milliseconds.

Reference Code:

```
self.timeStep = 32 # set the control time step
```

We also define the basic structure of our class MyController as shown below:

Reference Code:

```
1 from controller import Robot
2 from controller import Motor
3 from controller import Altimeter
4 from controller import LED
5 import math
6
7
8
9 class MyController(Robot):
10     def __init__(self):
11
12
13     def run(self):
14
15
16
17 # main Python program
18 controller = MyController()
19 controller.run()
20
```

Next, we create and define the variables that will store information about devices used by our robot.

- iii. Code to work with an altimeter.

Reference Code:

```
self.altimeter=self.getDevice("altimeter")
self.altimeter.enable(self.timeStep)
```

The **getDevice function of Robot** will look for the actual device name.

Each sensor must be enabled before it can be used. If a sensor is not enabled it returns undefined values. Enabling a sensor is achieved by using the corresponding *.enable function, where

the star (*) stands for the sensor type.

- iv. Code to do wheel rotation .

Reference Code:

```
self.left_motor = self.getDevice("left wheel motor")
self.right_motor = self.getDevice("right wheel motor")
self.left_motor.setPosition(math.inf)
self.right_motor.setPosition(math.inf)
```

- v. Continuously keep on moving the robot:

Reference Code:

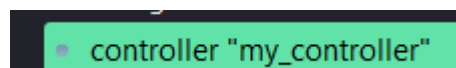
```
def run(self):

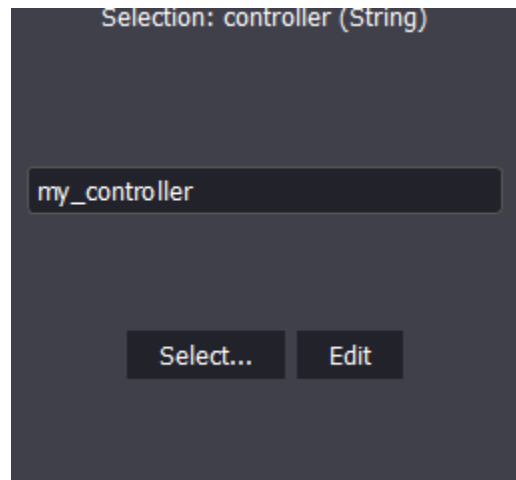
    while self.step(self.timeStep) != -1:
        # get the time step of the current world.

        altitude = self.altimeter.getValue()
        # print(altitude)
        if (not self.direction_switch):
            self.left_motor.setVelocity(2.0)
            self.right_motor.setVelocity(2.0)
            if (altitude <= 0.05):
                self.direction_switch = True
        else:
            self.left_motor.setVelocity(-2.0)
            self.right_motor.setVelocity(-2.0)
            if (altitude >= 0.25):
                self.direction_switch = False
```

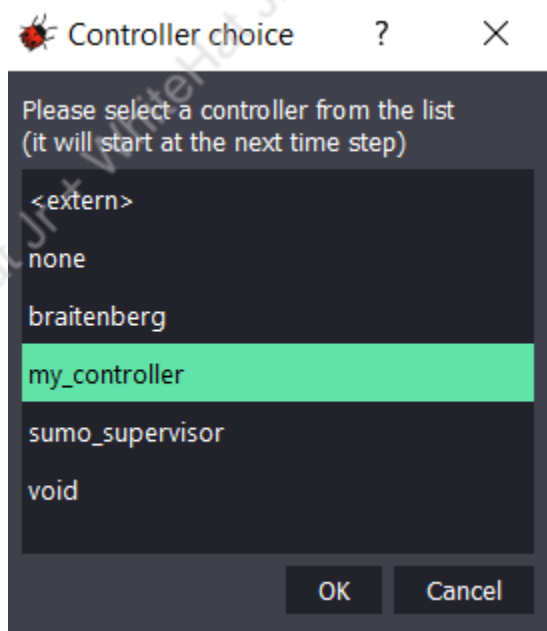
- c. Next, we added controller name under scene tree too:

- Go to your Scene tree
- Click on Robot
- Check Controller name
- Click on controller

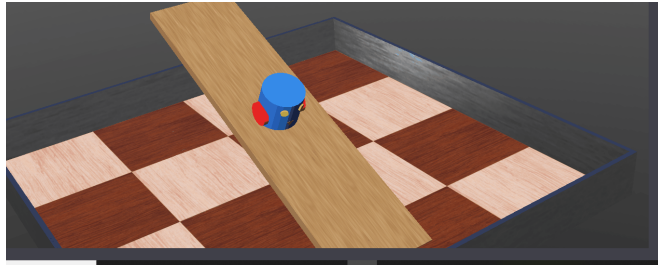




- Click on **Select**
- Select the name of your controller file which we mentioned while creating the controller.
- In this case the controller name is **my_controller**.



<https://s3-whjr-curriculum-uploads.whjr.online/670292b9-fa0f-494a-9fda-6d458f9d8d25.gif>

Reference Output:

<https://s3-whjr-curriculum-uploads.whjr.online/63ada1e8-797a-4697-afd3-df67adf96c5a.gif>

What's NEXT?

In the **next class**, we will learn to design obstacle detection.

Expand Your Knowledge

To know more about **Controller Programming** [click here](#).