Robot World

What is our GOAL for this CLASS?

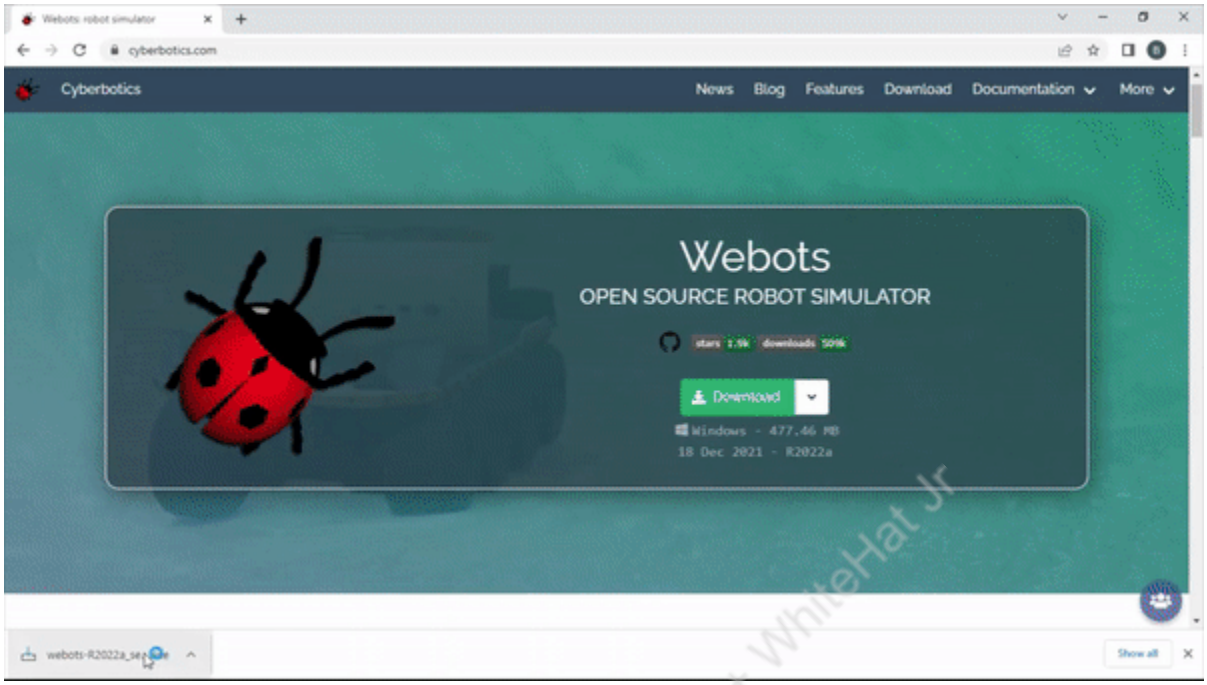
In this class, we learned about Robot World. We installed webots simulator and learned how to set a robot world using simulation. We made a rectangle arena and used an inbuilt e-puck robot to see how things work on Robot world.

What did we ACHIEVE in the class TODAY?

- We installed Webots Simulator
- We designed Rectangle Arena
- We used in_built e-puck controller
- We learned hands-on settings for rectangle arenas and Webots simulation.

How did we DO the activities?

1. We installed Webots. Follow the below video to install Webots in Windows.
    - Download the "webots-R2022a_setup.exe" installation file from our website.
    - Double click on this file.
    - Follow the installation instructions.



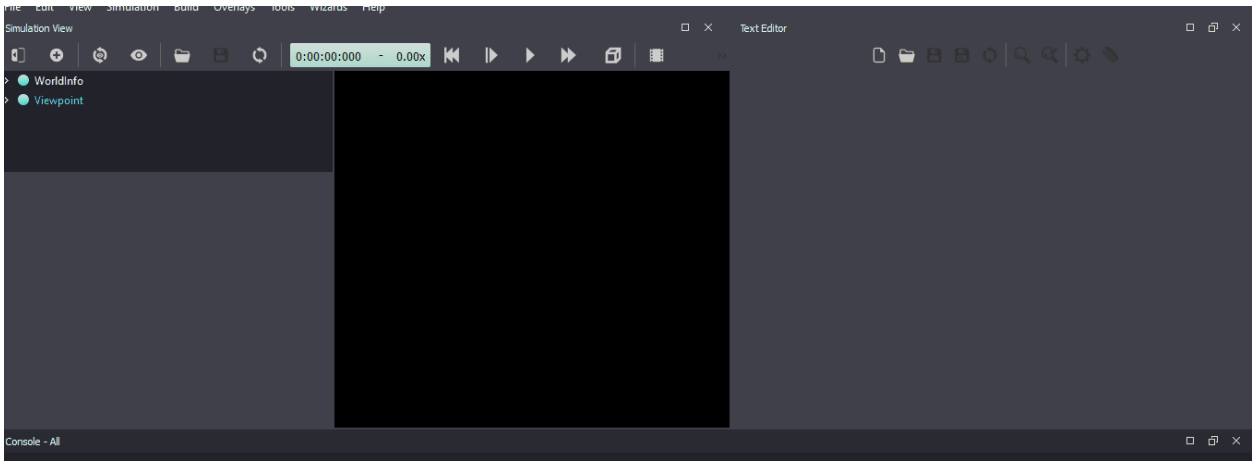
https://s3-whjr-curriculum-uploads.whjr.online/2d461a78-9391-417e-92df-91feef36c84d.gif

2. Webots GUI is composed of four principal windows:
    - **3D** window that displays and allows you to interact with the 3D simulation.
    - **Scene tree** which is a hierarchical representation of the current world
    - **Text editor** that allows you to edit source code
    - **Console Window** that displays both compilation and controller outputs.

3. **New Simulation creation:**

    - This simulation will contain a simple environment (a rectangle arena with floor and walls), one inbuilt Robot (e-puck) and a controller program that will make the robot move.

    a. **Create a new world:**
    - **World:** A World defines the initial state of a simulation. A world is stored in a file having the **".wbt" extension.**
    - Now, we would like to add some environment objects . A predefined node called RectangleArena is designed to accomplish this task quickly.

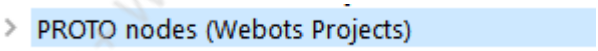


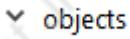
https://s3-whjr-curriculum-uploads.whjr.online/c87bad64-dc04-4ae2-a640-51b9ebeb9756.gif

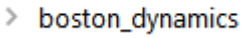
b. Set the Rectangle properties:

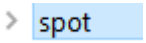
i. Go to view
ii. Go to Orthographic Projection

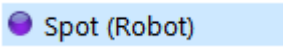
c. Add a **spot** Robot

i. Click on 

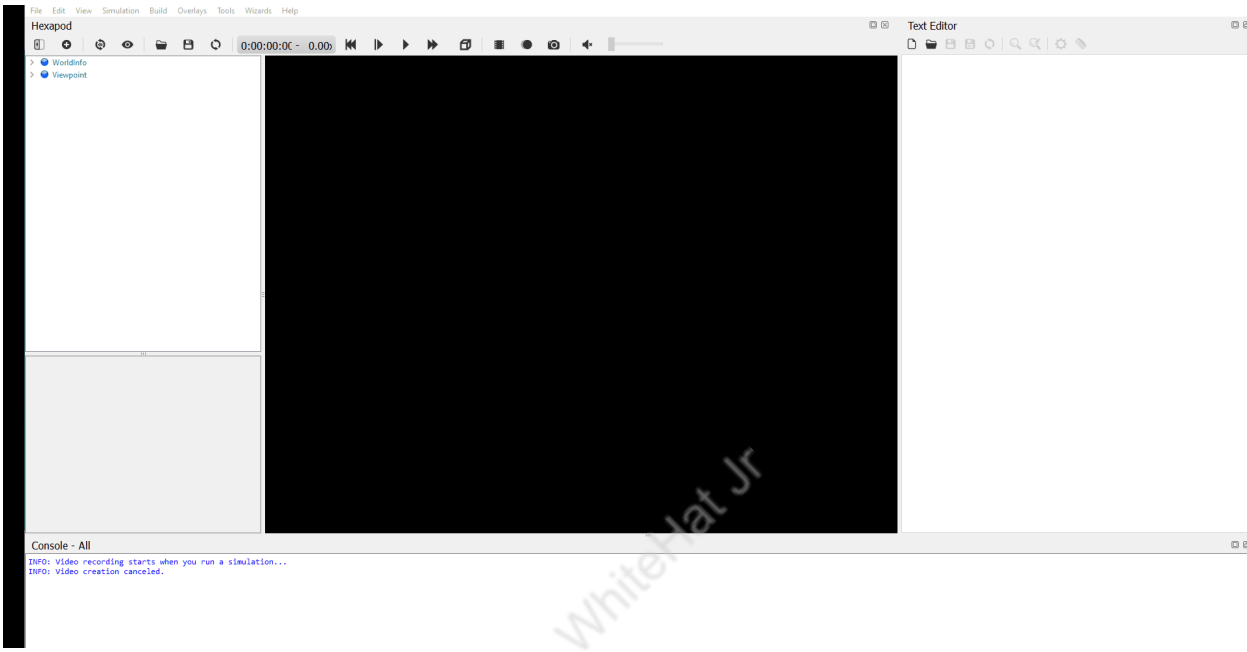
ii. Click on 

iii. Click on Robots
iv. Click on 

v. Click on 

vi. Click on 

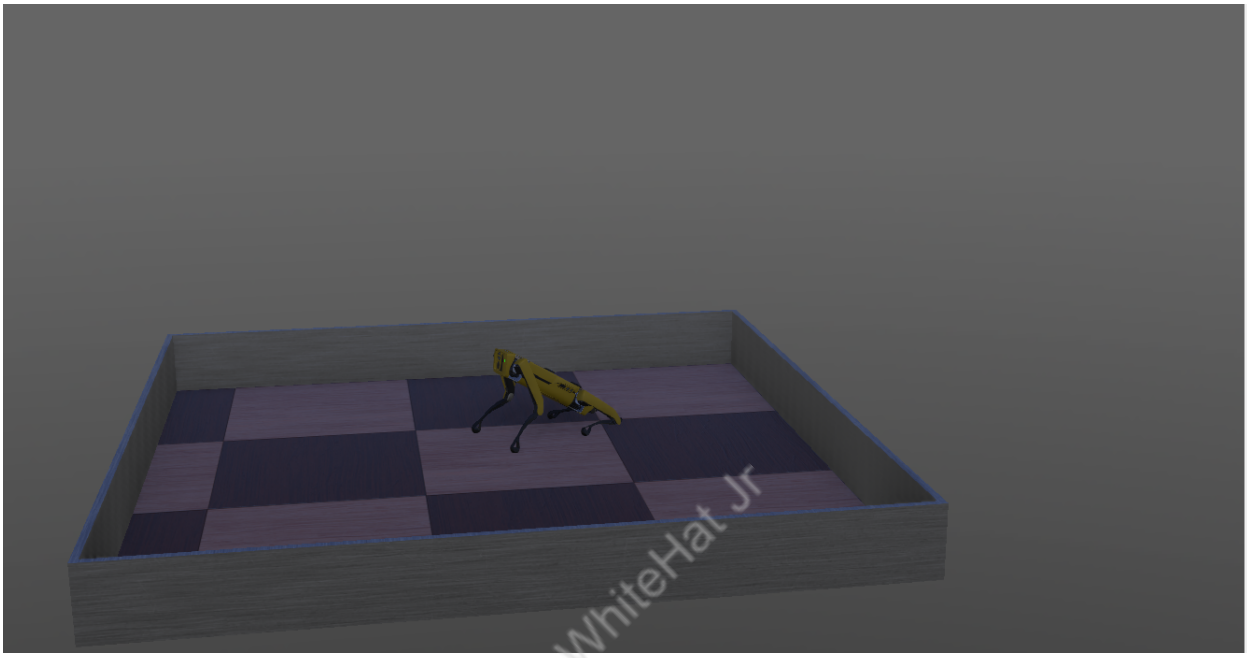
vii. Click on ADD



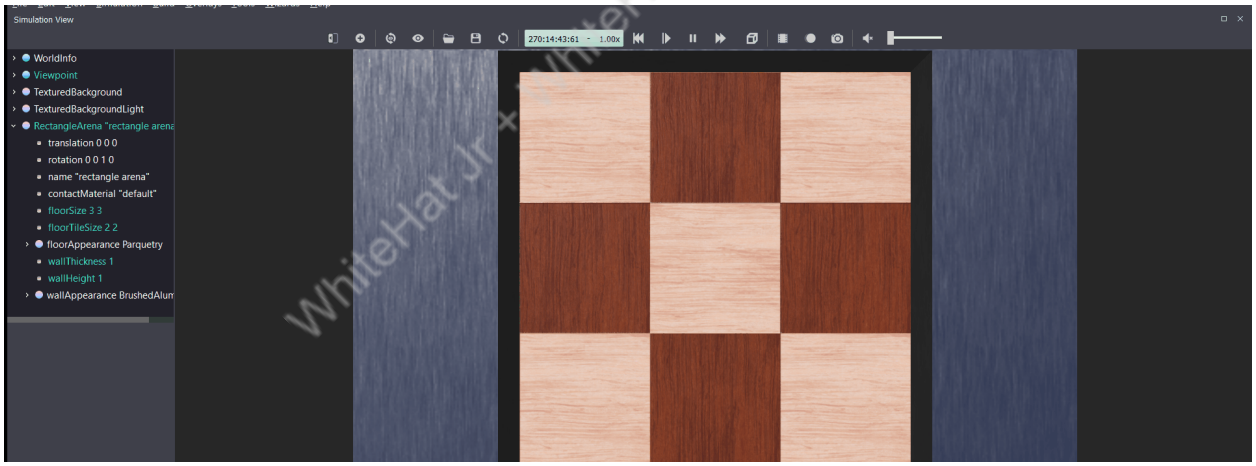
https://s3-whjr-curriculum-uploads.whjr.online/5c99bd70-def3-4dd1-9f05-2620c4272d6d.gif

Click on the cross sign to turn off the Camera.(Pink Color Cross sign on the top)

After that you will see a Boston_Dynamics Robot



d. Add an **e-puck** Robot

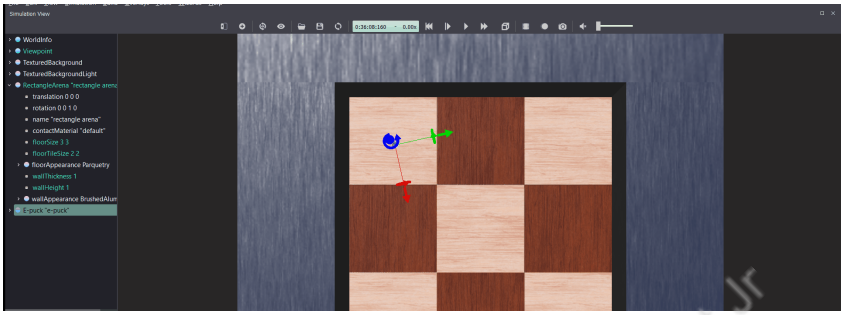


https://s3-whjr-curriculum-uploads.whjr.online/472d52d1-bbd3-451b-9adf-993f1373f123.gif

e. We can change the robot's position in the 3D view using the translation and rotation handle.

- Alternatively, the following keyboard shortcuts are available:
- ***SHIFT + left-click + drag*** to move the robot parallel to the floor;
- ***SHIFT + mouse-wheel*** to move the robot up or down.

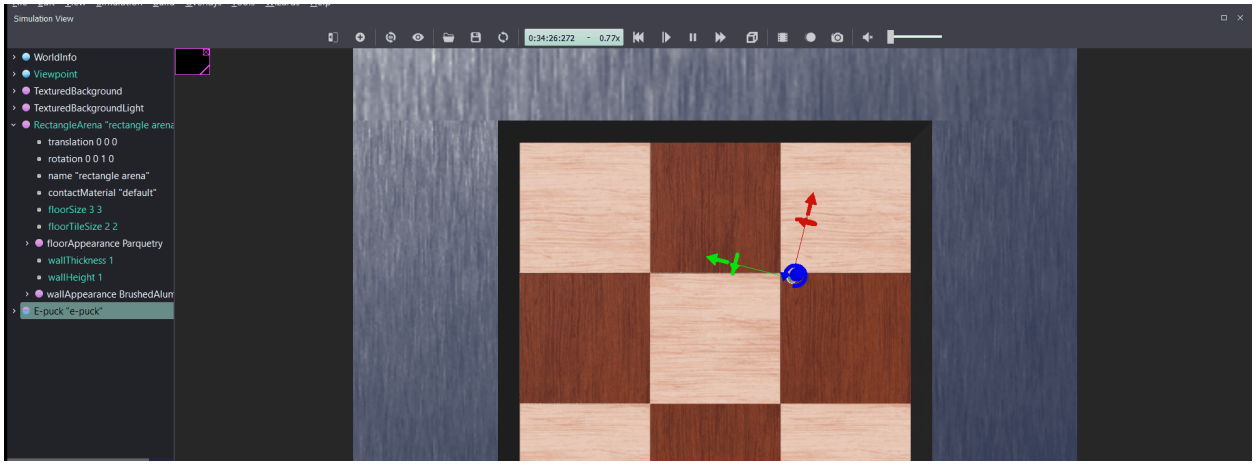
- To apply a force to the robot: ***ALT + left-click + drag.***
- On Linux, you should also press the ***CTRL*** key in addition to ***ALT + left-click + drag.***



https://s3-whjr-curriculum-uploads.whjr.online/254f305a-713f-4c4a-a192-6ee4f47ebaab.gif

f. Just after you add the E-puck node, a black window appears in the upper-left corner of the 3D view. It shows the content of Camera nodes, but it will stay black until not explicitly used during a simulation. The camera can be resized by dragging the marked corner or hidden by clicking the "x" in the top-right of the camera window.

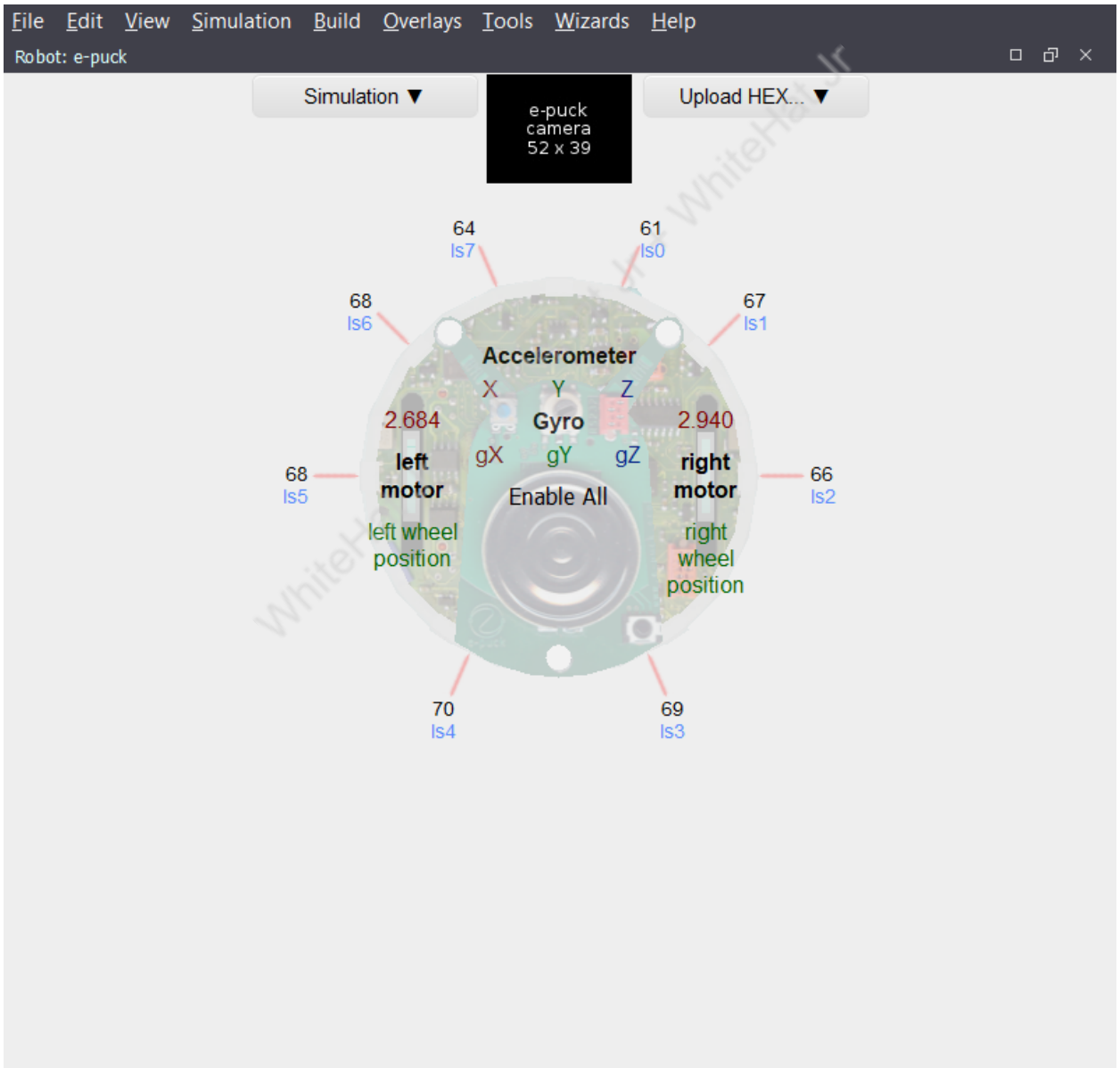
- As of now we will not use the Camera devices of the E-puck. So we can hide the window by clicking the "x" on the camera window. Don't forget to reload the world before hiding the camera and to save it after the modifications.



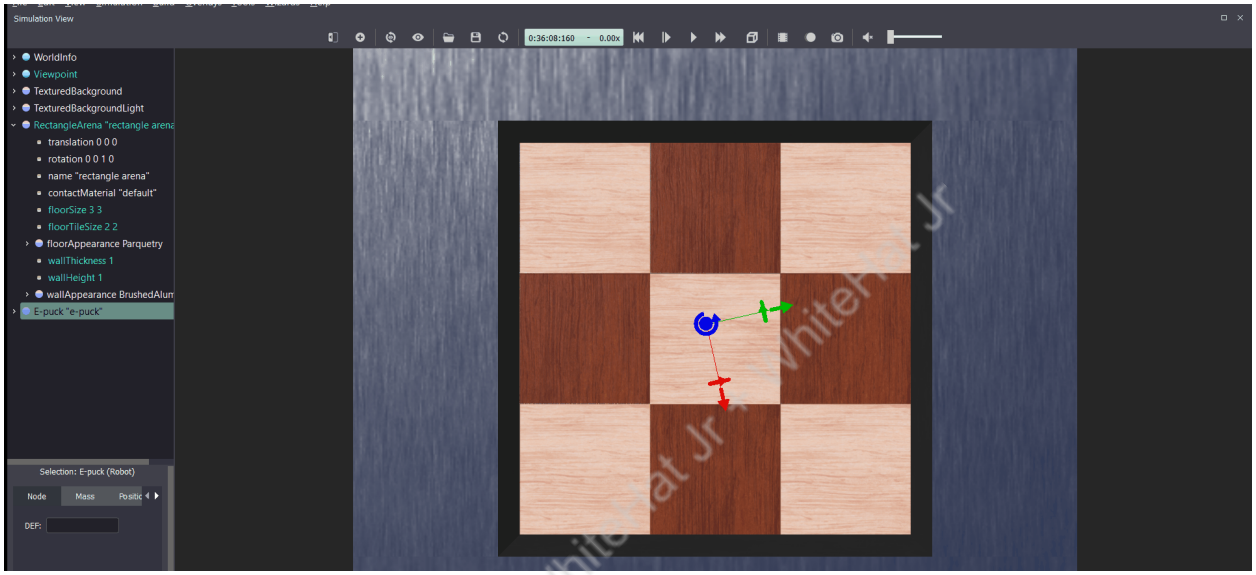
https://s3-whjr-curriculum-uploads.whjr.online/e79be729-5f10-4832-a8ba-4abe0b0

58a99.gif

4. Components which are used in **e_Puck**
    - **Double click** on your **e_Puck** a window will appear.
    - You will get all the details of components here.
    - We can see that this **e_Puck Robot** needs :
        - Accelerometer
        - Gyro
        - Motor
        - Sensors



5. Create a new controller called *e-puck_go_forward* using the Wizards / New Robot Controller.
   A controller is a program that defines the behavior of a robot.
   Steps to Create a new controller:
   1. Go to the Wizards
   2. Select New Robot Controller..
   3. Select the language and Name the controller and click on Finish



https://s3-whjr-curriculum-uploads.whjr.online/940896e8-52fe-4573-956a-d95ed134e782.gif

Default controller file in Python language looks like this:

my_controller.py

```python
"""my_controller controller."""

# You may need to import some classes of the controller module. Ex:
#  from controller import Robot, Motor, DistanceSensor
from controller import Robot

# create the Robot instance.
robot = Robot()

# get the time step of the current world.
timestep = int(robot.getBasicTimeStep())

# You should insert a getDevice-like function in order to get the
# instance of a device of the robot. Something like:
#  motor = robot.getDevice('motorname')
#  ds = robot.getDevice('dsname')
#  ds.enable(timestep)

# Main loop:
# - perform simulation steps until Webots is stopping the controller
while robot.step(timestep) != -1:
    # Read the sensors:
    # Enter here functions to read sensor data, like:
    #  val = ds.getValue()

    # Process sensor data here.

    # Enter here functions to send actuator commands, like:
    #  motor.setPosition(10.0)
    pass

# Enter here exit cleanup code.

```

Note: Make sure that the simulation is paused and that the virtual time elapsed is 0. Simulation can be paused by clicking on **Simulation** and then click on **Pause** or directly click on **Pause** Button

What's NEXT?

In the **next class**, we will learn about **first Robot design**

Expand Your Knowledge

To know more about **Webots** click here.