Topic	Robot World	
Class Description	**Students will be introduced to Robots and their world. Students will also learn about some basic functions of robots.**	
Class	**PRO C278**	
Class time	**45 mins**	
Goal	• Introduction to Webots • Installation of Webots • Explore Robot World	

Resources Required	• Teacher Resources:
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen
	  ○ Smartphone
	• Student Resources:
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen	

Class structure	**Warm-Up** **Teacher-Led Activity 1** **Student-Led Activity 1** **Wrap-Up**	**10 mins** **10 mins** **20 mins** **05 mins**
Credit & Permissions:	**This project uses Webots, an open-source mobile robot simulation software developed by Cyberbotics Ltd.** **This project uses Webots (http://www.cyberbotics.com), an open-source mobile robot simulation software developed by Cyberbotics Ltd.**	

<table>
<tr><th colspan="2">WARM-UP SESSION - 10 mins</th></tr>
<tr><td colspan="2">Teacher Starts Slideshow 
Slide # to #
<Note: Only Applicable for Classes with VA>
Refer to speaker notes and follow the instructions on each slide.</td></tr>
<tr><th>Teacher Action</th><th>Student Action</th></tr>
<tr><td>Hey <student's name>. How are you? It’s great to see you!
Are you excited to learn something new today?

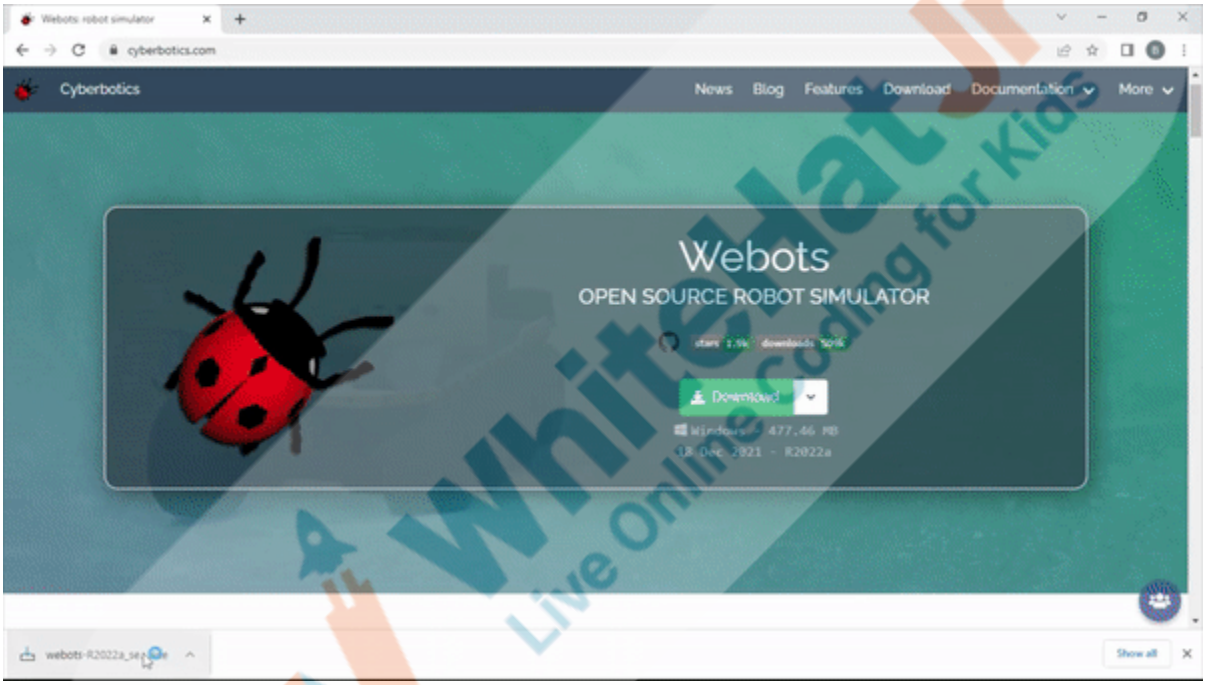
Following are the WARM-UP session deliverables:
• Greet the student.
• Revision of previous class activities.
• Quizzes.</td><td>ESR: Hi, thanks!
Yes I am excited about it!

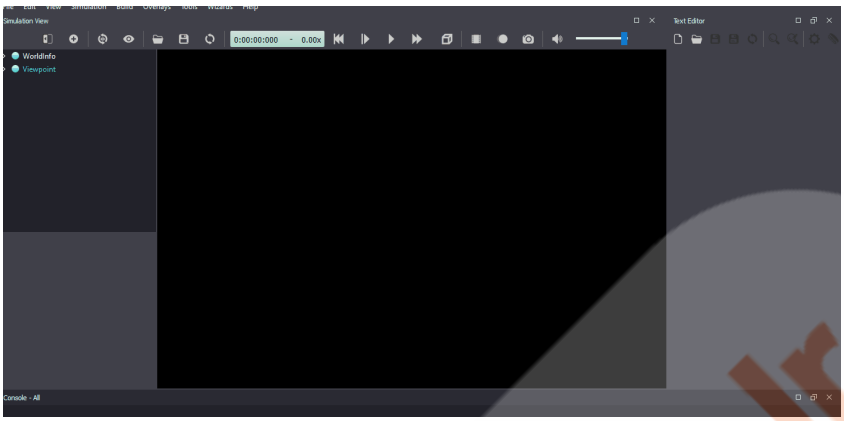
Click on the slide show tab and present the slides</td></tr>
<tr><td colspan="2">WARM-UP QUIZ
Click on In-Class Quiz</td></tr>
<tr><td colspan="2">Activity Details

Following are the session deliverables:
• Appreciate the student.
• Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.</td></tr>
<tr><th colspan="2">TEACHER-LED ACTIVITY - 10 mins</th></tr>
<tr><td colspan="2">Teacher Initiates Screen Share</td></tr>
<tr><td colspan="2">• Introduction to Robots
• Installation of Webots</td></tr>
<tr><th>Teacher Action</th><th>Student Action</th></tr>
<tr><td>Any doubts from the last class!</td><td></td></tr>
</table>

The teacher will clarify if there are any doubts! How's your experience with ESP32 and Arduino Controller? So what type of applications can we develop using ESP32 and Arduino? In a few classes we learned about a lot of hardware and controllers. We developed a lot of applications for the same. It was fun! But where we can use these controllers and hardware components together. Although we have seen a lot of applications, what if we want to use all the components in one application only? Do you think it would be possible? Suppose, I want to use servo, LEDs, position sensor, Ultrasonic sensor, PIR sensor, and displays in one application. Can you tell me one application? What about Robots? Have you seen any live robots? Of course yes, we can see small or big Robots depending upon the application. But my next question is:	**ESR:** Varied! **ESR:** Varied!

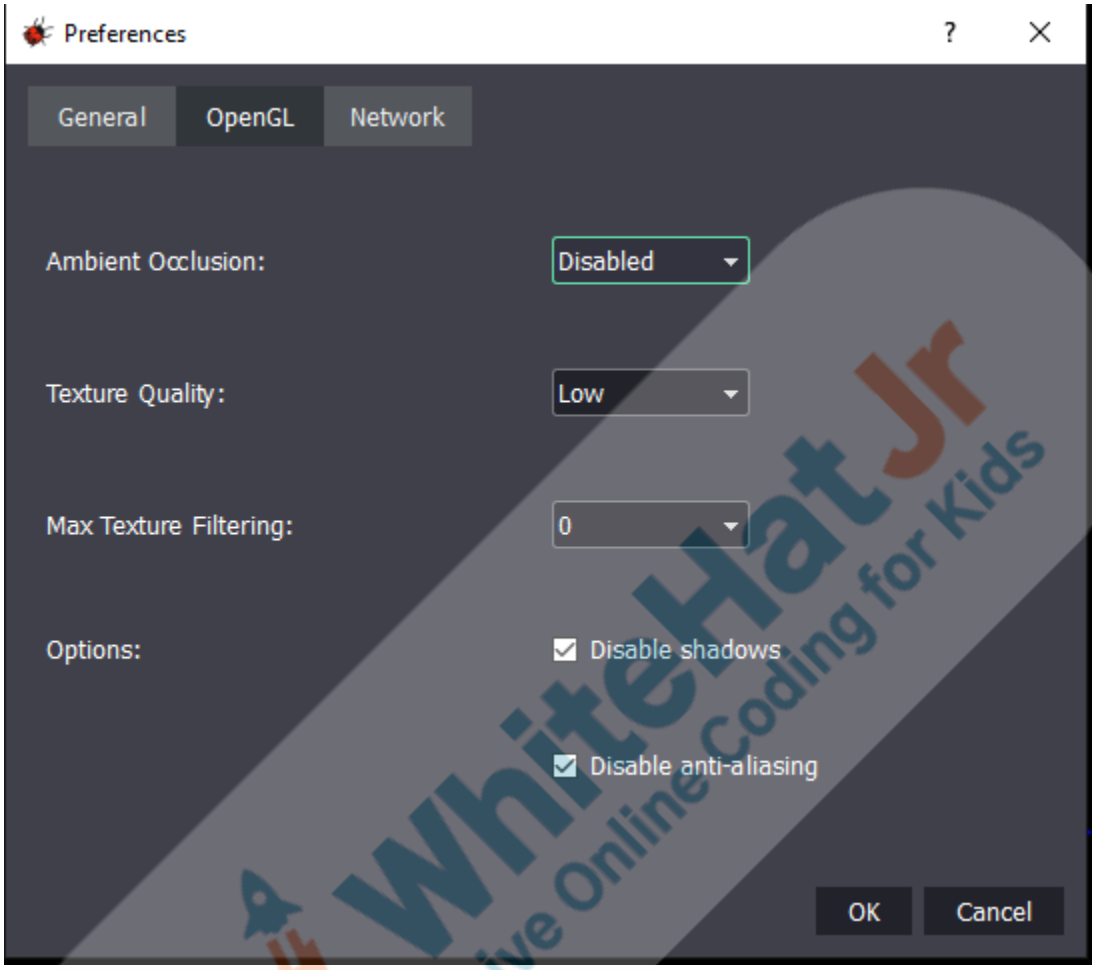
• Who made the robots? • How do they do it? • What types of components do they use to make Robots? • How do they train Robots? I know you must have the same questions. No worries!! We will get all the answers when we enter the world of Robots. So Let's deep dive into the Robots World! Robots can do anything, we just need to train them accordingly. You must know this as we have seen so many Robo-centric movies.	
Let's Learn about Webots: Webots is a professional mobile robot simulation software package. It offers a rapid prototyping environment that allows the user to create a 3D virtual Robots world.Webots contains a large number of robot models and controller program examples to help users get started. A Webots simulation is composed of following items: • A Webots world file (.wbt) that defines one or several robots and their environment. The .wbt file does sometimes depend on external PROTO files (.proto) and textures. • and controller programs for the Robots	
Teacher will click on Teacher Activity 1	Student will click on Student

	Activity 1
Click on the **Download link** and **install webots.** Follow the below video to install Webots in Windows. 1. Download the **"webots-R2022a_setup.exe"** installation file from our website. 2. **Double click** on this file. 3. Follow the **installation instructions.**	
 https://s3-whjr-curriculum-uploads.whjr.online/d5c0f974-9e8e-49fb-8ff0-9f911796c5f1.gif	
Webots Installation Procedure for Mac. You must need administrator privileges to be able to install Webots. You may also change your macOS security settings to open Webots anyway (System Preferences / Security & Privacy / General / Allow apps downloaded from)	
Once the installation is done. We need to do some more settings. Go to the **Tools> Preferences>>Open GL**	



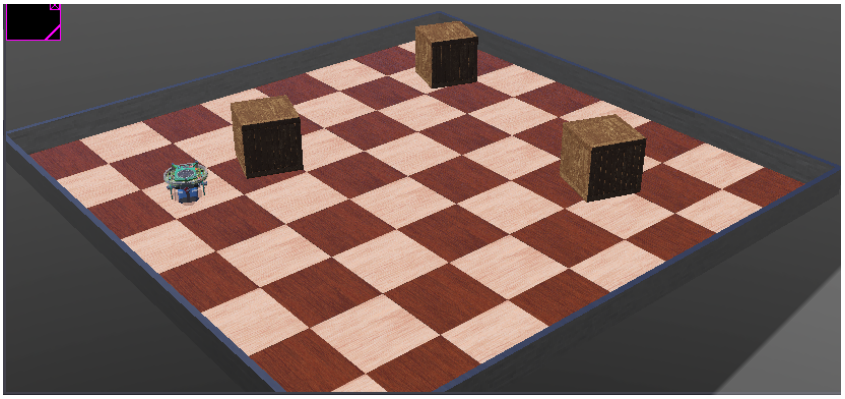
https://s3-whjr-curriculum-uploads.whjr.online/626bdd81-a6e4-4fa2-b10c-99dbab467fbf.gif

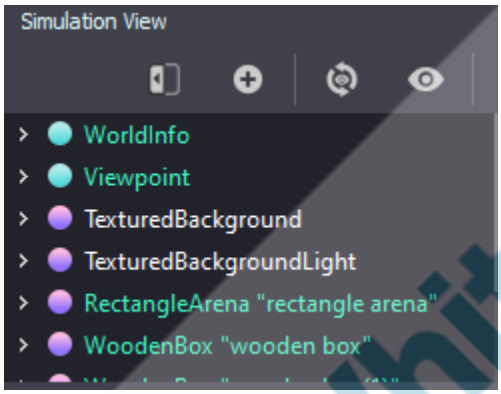
Do **OpenGL** settings as shown below:
Click on **OK.**



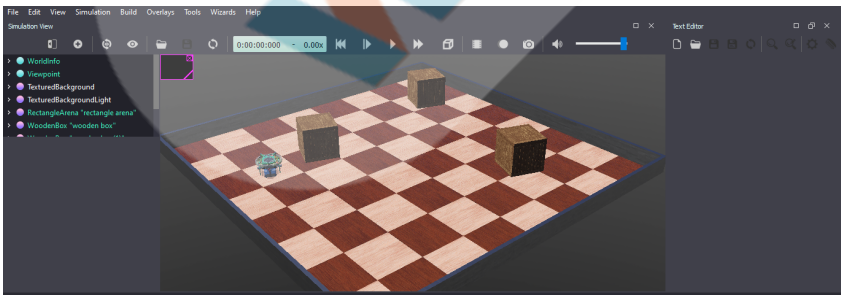
Our installation is complete. Let's explore webots. The first time you start Webots, it will open the "Welcome to Webots!" menu with a list of possible starting points. You will see a lot of options in Webots.	
The File menu allows you to perform usual file operations: loading, saving, etc. • The **New World** menu item (and button) opens a new world in the simulation window containing	

a chessboard of 10 x 10 squares on a surface of 1 m x 1 m. •  The **Open World**... menu item (and button) opens a file selection dialog that allows you to choose a ".wbt" file to load. • The **Open Recent World** menu item gives the possibility of reopening a ".wbt" file that was opened recently by choosing it from the list displayed in the submenu. •  The **Save World** menu item (and button) saves the current world using the current filename (the filename that appears at the top of the main window). On each Save the content of the ".wbt" file is overwritten and no backup copies are created by Webots, therefore you should use this button carefully and eventually do safety copies manually. • The **Save World As...** menu item (and button) saves the current world with a new filename entered by the user. Note that a ".wbt" file should always be saved in a Webots project directory, and in the "worlds" subdirectory, otherwise it will not be possible to reopen the file. •  The **Reload World** menu item (and button) reloads the current world from the saved version and restarts the simulation from the beginning.	
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. The scene tree contains the information that describes a simulated world, including robots and environment, and its graphical representation.This will be on screen left side where you will see **World** and **View Info**.



Text editor that allows you to edit source code.

Console Window that displays both compilation and controller outputs and errors if any.

Console Window:

Webots GUI will contain a lot of options. We will explore it while doing it.

But to start, pause, reset the Simulation, we must be aware of control features of the simulation menu.

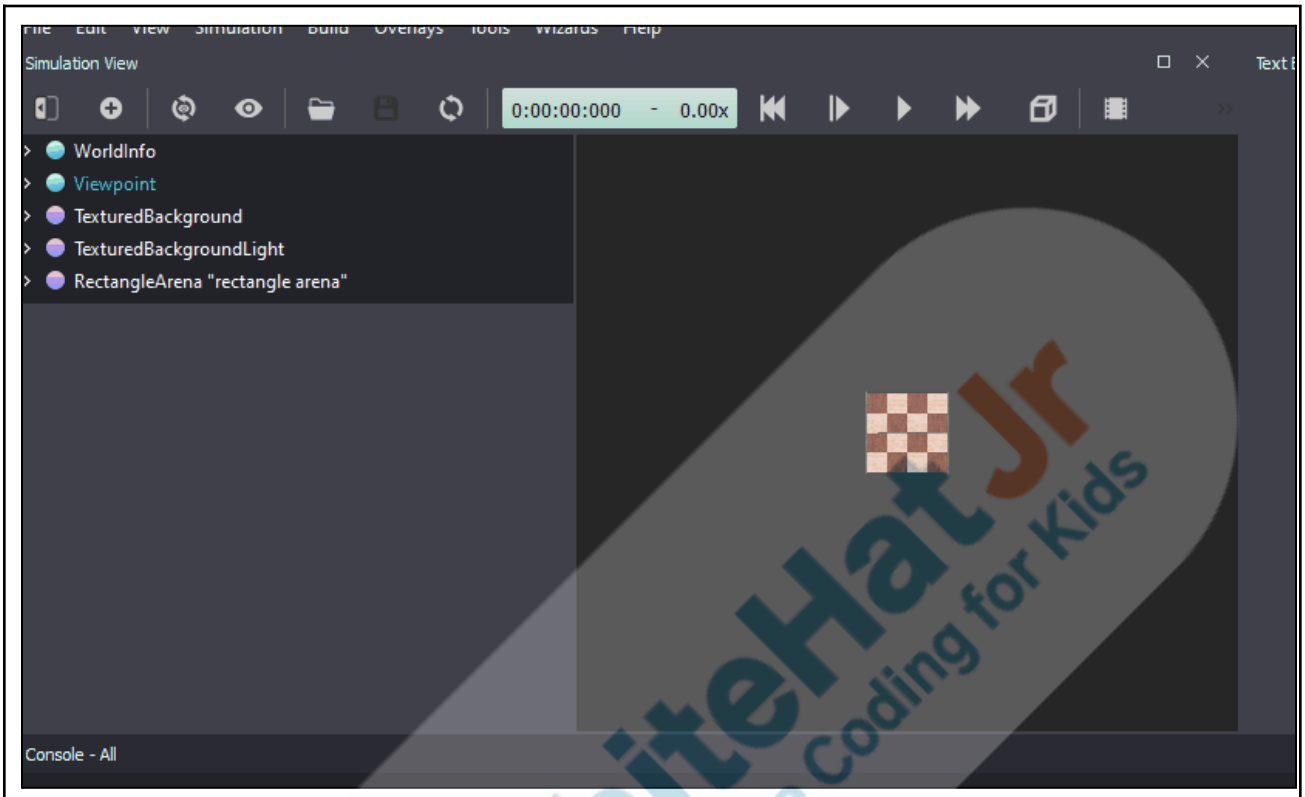
-  The Pause menu item (and button) pauses the simulation.
-  The Step menu item (and button) executes one basic time step of simulation.
-  The Real-time menu item (and button) runs the simulation in real-time until it is interrupted by Pause or Step.
-  The Fast menu item (and button) is like Real-time, except that it runs as fast as possible.In fast mode, the 3D display of the scene is refreshed every n basic time steps.

Always use these buttons to control the simulation.

Note: To make changes always reset the simulation first.

Save the simulation

 Sign is used to save the simulation. At every step use the save button to save the simulation.

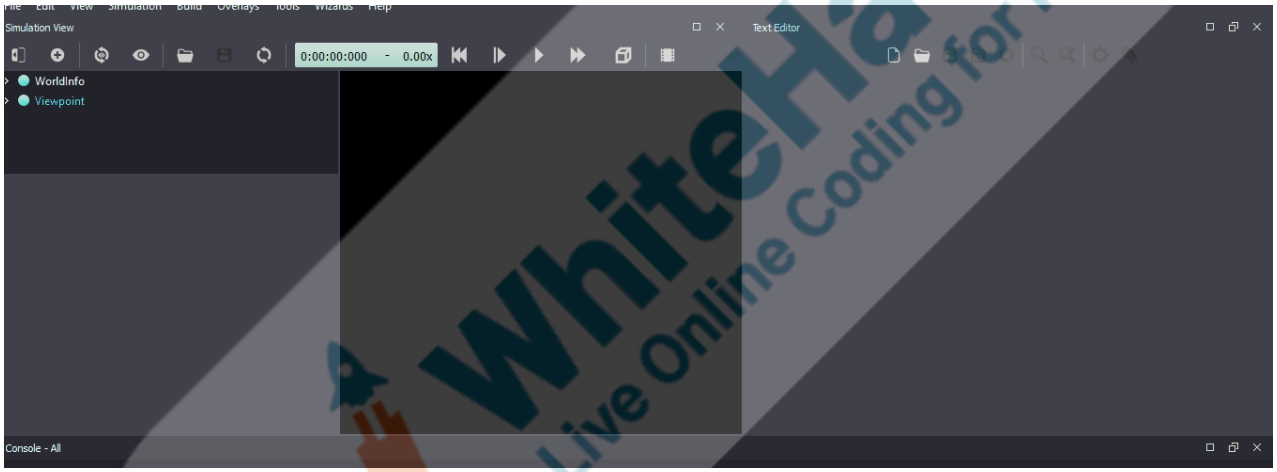


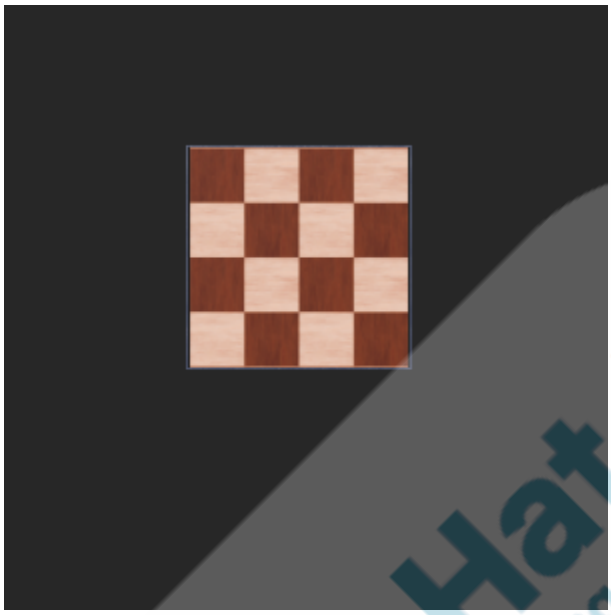
Click here to view the reference video.

So basically this rectangle arena is your playground where our robots will work.	
Now you must be waiting to meet with Mr Robot. Am I right?	**ESR :** Yes!
Teacher Stops Screen Share	
So now it's your turn. Please share your screen with me.	

Teacher Starts Slideshow
Slide # to #
<**Note**: Only Applicable for Classes with VA>

Refer to speaker notes and follow the instructions on each slide.	
We have one more class challenge for you. Can you solve it? Let's try. I will guide you through it.	
Teacher Ends Slideshow 	
STUDENT-LED ACTIVITY - 20 mins	
• **Ask the student to press the ESC key to come back to the panel.** • **Guide the student to start Screen Share.** • **The teacher gets into Full Screen.**	
Student Initiates Screen Share	
ACTIVITY • **Robot Installation**	
Teacher Action	**Student Action**
So today we will explore Robots and their worlds In webots we can develop any type of graphics simulation. The first objective is to familiarize with the user interface and with the basic concepts of Webots.	
A new Simulation: This simulation will contain a simple environment (a rectangle arena with floor and walls), one inbuilt Robot and a controller program that will make the robot move.	
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.**	
Go to Wizards 1. Select new ProjectDirectory 2. Name the Project 3. Click on Next 4. Write the name of your World 5. Tick mark on Rectangle Arena 6. Click on Next 7. Click on Finish ***Note: Follow the GiF***	
 Click here to view the reference video.	
You will see output like this:	



If you will see on your left you will see the following options are already added.

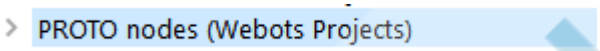
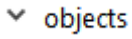
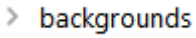
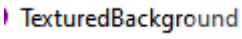
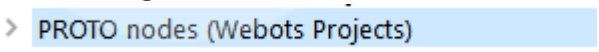
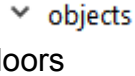
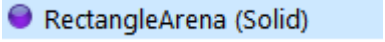
1. Textured background
2. Textured background & Light
3. Rectangle Arena

Note: Teacher can delete above three options by clicking on option and make them understand how to add objects in webots .

Adding things to make an environment is very easy.

Procedure to add Rectangle Arena and Lights:

Note: Teachers can delete the Lights and Rectangle arena option to make students practice how to add things by

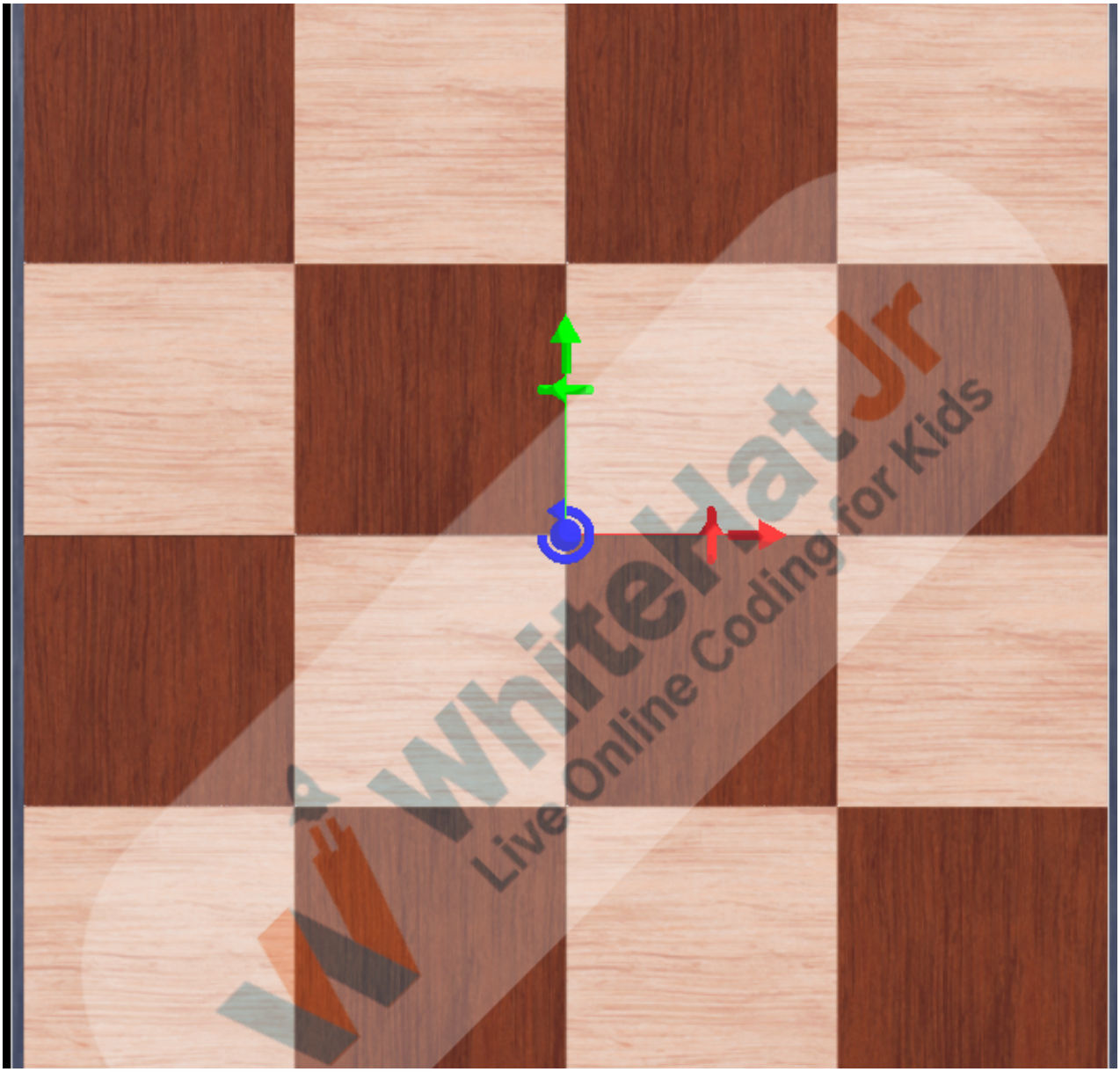
following the options below: This is important as they will understand how to add objects . *Click on Texture Light and delete it. Do the same for others too.* To add objects just we need to click on  sign on the left side just above WorldInfo Click on **+** to add objects. We will see a lot of options: 1. Click on  2. Click on  3. Click on  4. Click on  5. Click on  So we added Backgrounds and Light.	
Now, we would like to add some environment objects .A predefined node called RectangleArena is designed to accomplish this task quickly. Note: By default rectangle arena will be present. In case if it is not present then follow the below steps: Click on  sign Click on  Click on  Click on floors Click on 	

But that rectangle arena will be very small. So we need to increase the size of the rectangle Arena. To make it large either we can change size or we can choose enlarge option: Go to **view** Click on **Orthographic Projection or Press F10** Rectangle Arena will look like this:	

In case it's not visible. Just go to Eye symbol  and check options to make it visible.

Select **Top view** to make it visible.

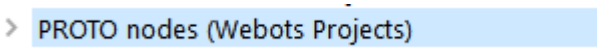
Not let's see how we can move rectangle arena, Three arrows (Green, Blue, Red) will be visible on the screen. If you click on the arrow and explore the arrows, With arrows we can change or move the rectangle arena. Or **Left click + scrolling** can also move rectangle arena **Right click + scrolling** can change the direction of the rectangle arena.	

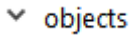


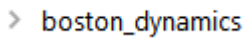
Add Some objects on Rectangle Arene

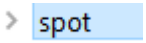
Note:
Make sure that the simulation is paused and that the virtual time elapsed is 0.Simulation can be paused by clicking on

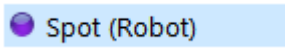
Simulation *and then click on* ***Pause*** *or directly click on* ***Pause*** *Button* 

1. Click on 

2. Click on 

3. Click on Robots
4. Click on 

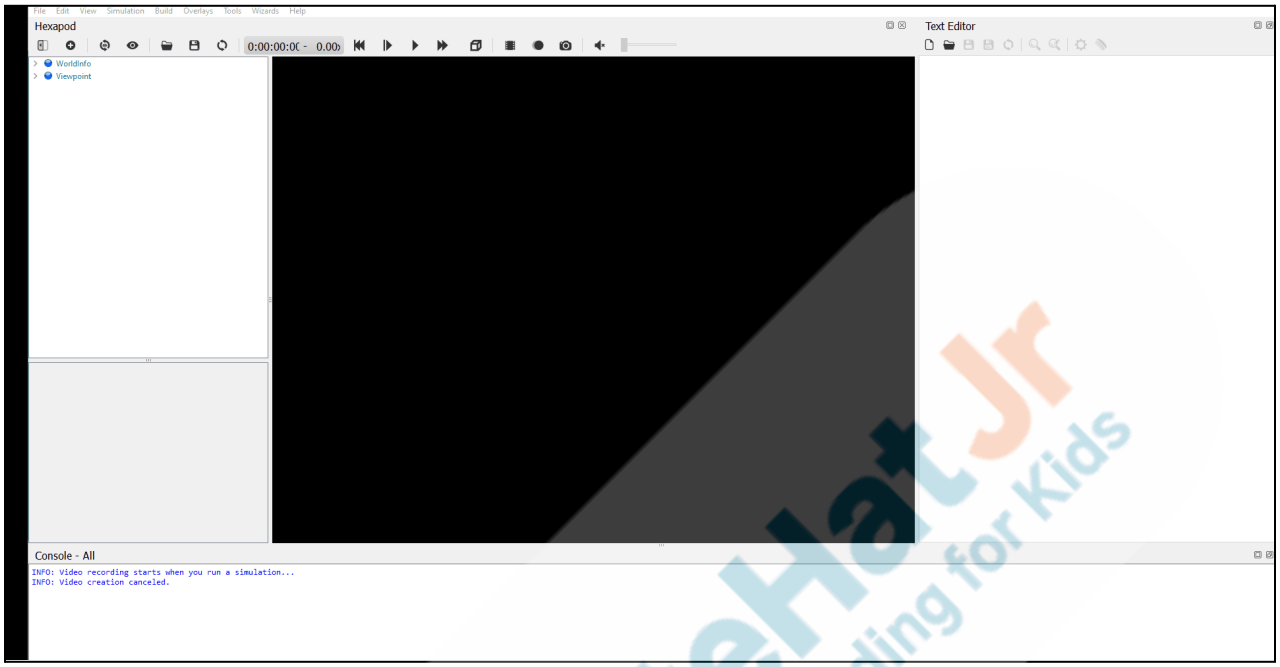
5. Click on 

6. Click on 

7. Click on **ADD**

Boston_Dynamic spot is a very famous Industrial Robot.
When you first add your Robot , You will see an invisible camera that will show you all surroundings .

Just after you add the Boston_spot Robot, 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.

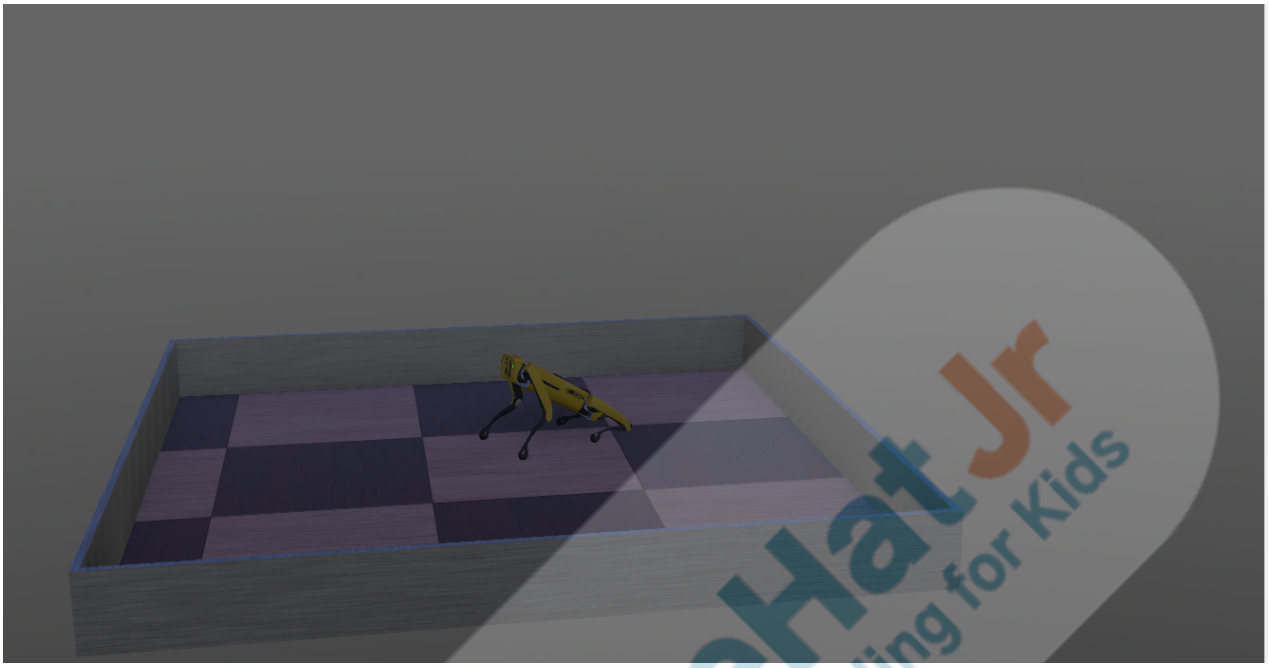


Click here to view the reference video.

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

Robot will look like this:



Note: Now if you run the simulation, the robot moves: that's because the robot uses a default controller with that behavior. Please pause and reload the world before going on.

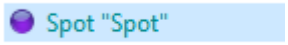
The Simulation menu is used to control the simulation mode. 

- The Pause menu item (and button) pauses the simulation.
- The Step menu item (and button) executes one basic time step of simulation.
- The Real-time menu item (and button) runs the simulation in real-time until it is interrupted by Pause or Step.

•  The Fast menu item (and button) is like Real-time, except that it runs as fast as possible. Starting the simulation by pressing the Run button will make Webots run the simulation as fast as possible. In order to obtain a real-time simulation speed, the Real-Time button needs to be pressed	
We can change the robot's position in the 3D view using the translation and rotation handle. Alternatively, the following keyboard shortcuts are available: 1. ***SHIFT + left-click + drag*** to move the robot parallel to the floor; 2. ***SHIFT + mouse-wheel*** to move the robot up or down. 3. To apply a force to the robot: ***ALT + left-click + drag.*** 4. On Linux, you should also press the ***CTRL*** key in addition to ***ALT + left-click + drag.***	
To check fast **simulation** of the Robot. Go to **simulation and** select the Fast Simulation to check the behavior of Simulation	
Note: If still have time in the class, Repeat the same process and select another Robot. Make sure you will delete old Robot **How to delete Previous Environment**	

Just click on the environment under the scene tree and then select delete.

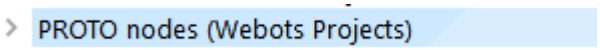
Suppose you want to delete Boston_Dynamic Spot Robot

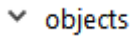
Just select on 

>> **Right Click** and select **Delete.**

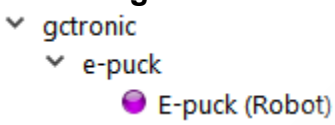
Now Let's try with another Robot.

These all Robots are in_Built Robots. But Later on we will learn how to develop Robots from Scratch.

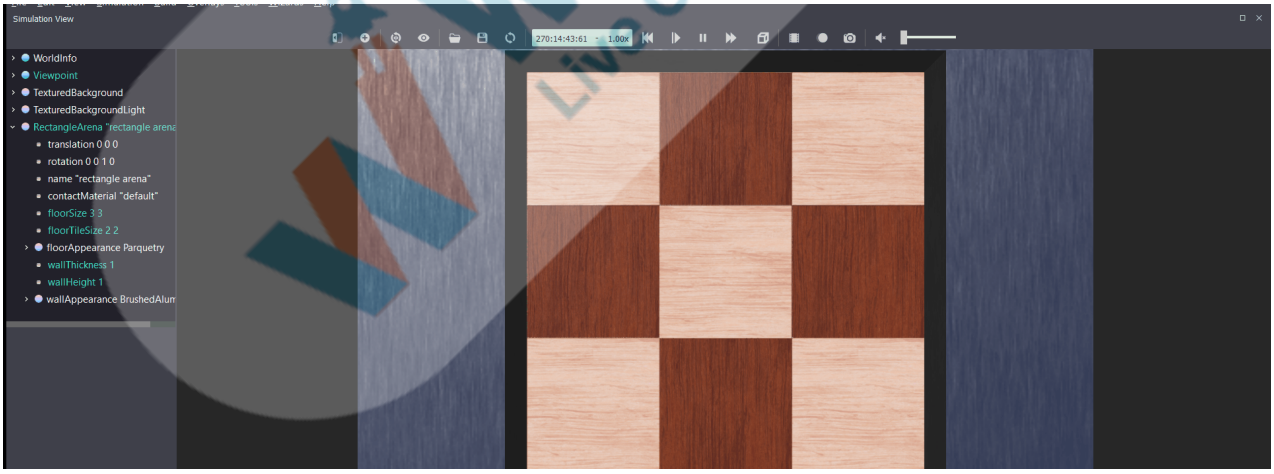
1. Click on 

2. Click on 

3. Click on Robots
4. Click on **gctronic**

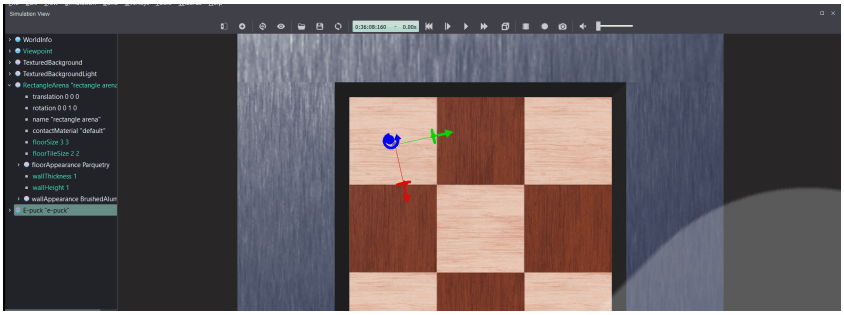


5. Click on e-puck
6. Click on E-puck (Robot)
7. Click on **ADD**



[Click here](#) to view the reference video.

E-Puck is another famous educational Robot.

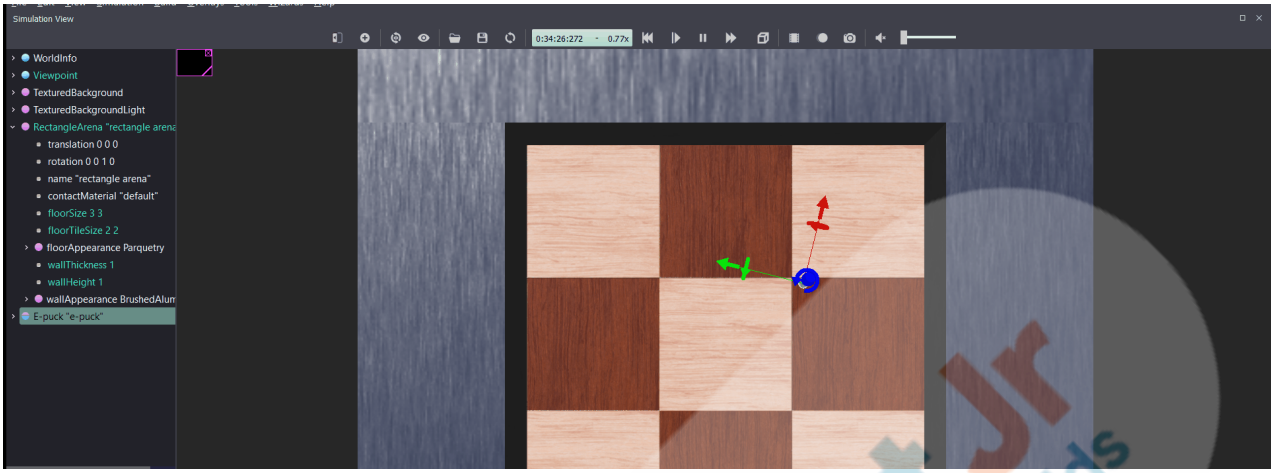


Click here to view the reference video.

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.

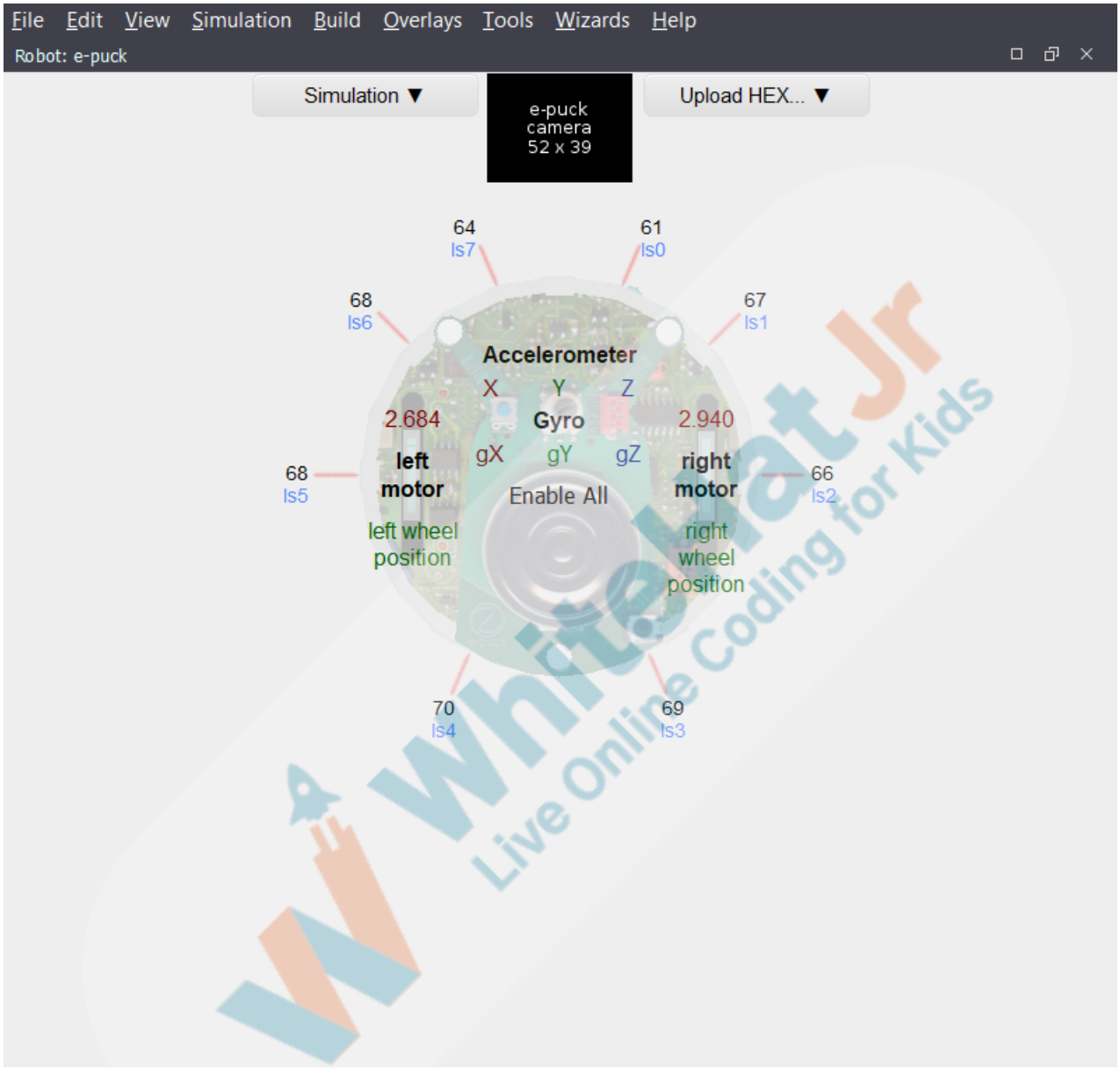
- save the simulation. 



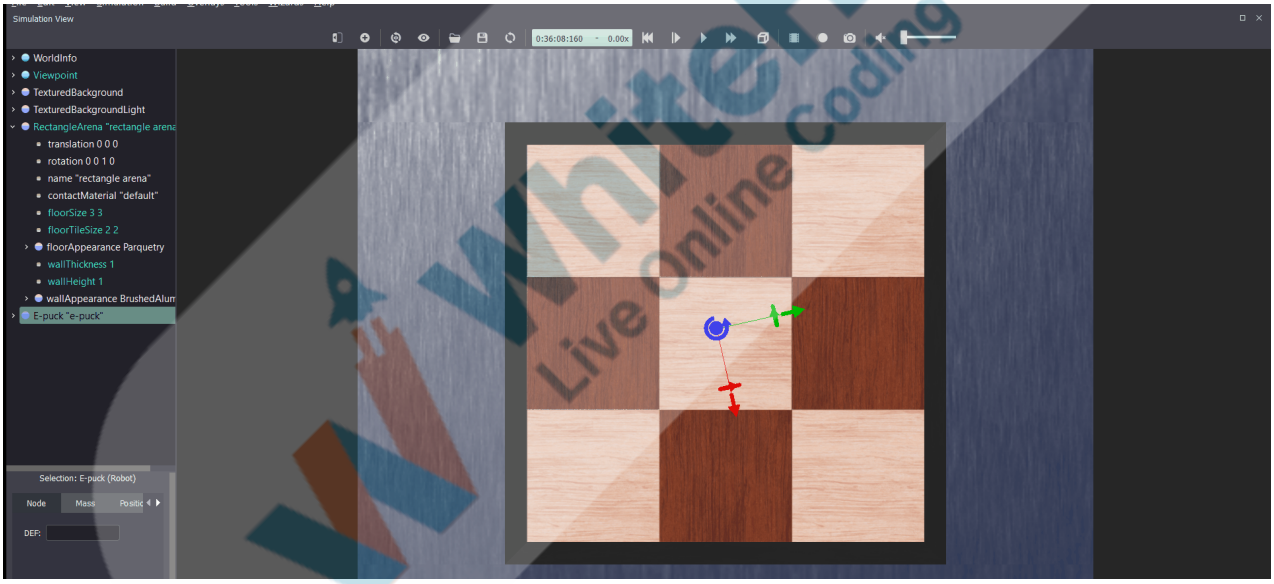
Click here to view the reference video.

Making a robot is not an easy one! Making simple Robots requires a lot of things. Can you guess what type of components we need to make a Robot? In some last classes, we learned about a lot of hardware components.	
Now you can imagine making of Robots needs all such hardware components. Depending upon application, we need to use the required components like sensors, Motors, Camera's , Accelerometers and some Physics too. Let's see about 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 *Note: Teachers will ask the student about the above components one by one as already used in previous classes.* **Sensors:** Sensors convert physical quantities like pressure, Temperature, Movement, Heat into electrical signals. **Motors:** The purpose of an electric motor is **to convert electrical energy into mechanical energy**. This mechanical energy can then be used to power everything from heavy, industrial machinery to everyday tools and appliances.	



So today we explored webots and how robots work. In the next class we will explore more about webots.	
We know to make things work we must need some type of controller, same case is with our Robots.	

No Robot works without a Controller. Let's learn how we can add controllers and how the basic control file looks like.	
Create a New Controller: A controller is a program that defines the behavior of a robot. 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	
 Click here to view the reference video.	
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.
```

We will program a simple controller that will just make the robot move forwards. As there is no obstacle, the robot will move forwards forever. Firstly we will create and edit the controller, then we will link it to the robot.

We will learn more about webots and controllers in the next session.

Teacher Guides Student to Stop Screen Share

WRAP-UP SESSION - 05 mins

Activity details

Following are the WRAP-UP session deliverables:

- Appreciate the student.
- Revise the current class activities.
- Discuss the quizzes.

WRAP-UP QUIZ
Click on In-Class Quiz

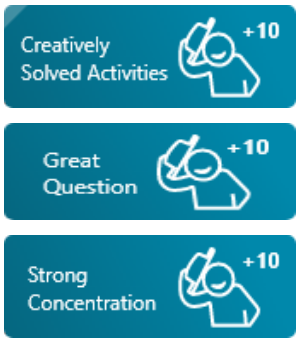
Activity Details

Following are the session deliverables:

- Explain the facts and trivia
- Next class challenge
- Project for the day
- Additional Activity (Optional)

FEEDBACK

- **Appreciate and compliment the student for trying to learn a difficult concept.**
- **Get to know how they are feeling after the session.**
- **Review and check their understanding.**

Teacher Action	Student Action
You get “hats-off” for your excellent work! In the next class, we will learn how to make our own objects in Webots.	*Make sure you have given at least 2 hats-off during the class for:* 

PROJECT OVERVIEW DISCUSSION

Refer the document below in Activity Links Sections

Teacher Clicks ✖ End Class

ACTIVITY LINKS		
Activity Name	**Description**	**Links**
Teacher Activity 1	Webots software download	https://cyberbotics.com/
Teacher Reference 1	Project	https://s3-whjr-curriculum-uploads.whjr.online/172c287b-794e-4fbc-a66e-dba8cd7a8ce8.pdf
Teacher Reference 2	Project Solution	NA
Teacher Reference 4	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/eab2d959-2a04-4b04-81a4-c62825068d4c.pdf
Student Activity 1	Webots software download	https://cyberbotics.com/