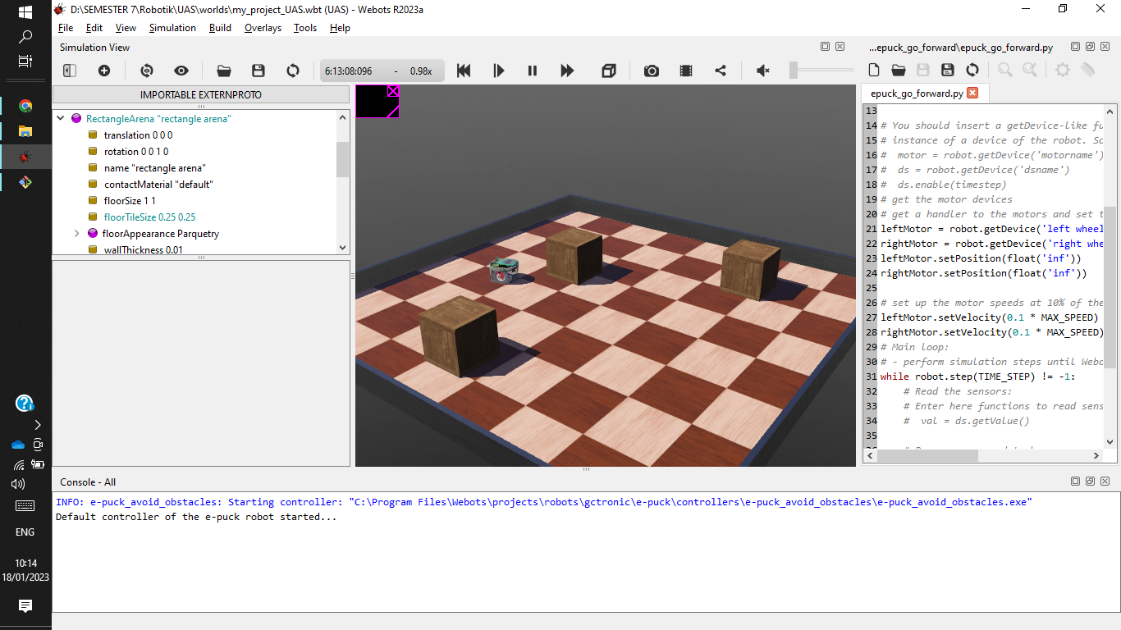
DOCUMENTATION WEBOT TUTORIALS & MICROMOUSE

NAMA:AGUSTIO

NIM:1103194102

1. **Tutorial 1: Your first Simulation in Webots**

* create robot objects and boxes and can move them
* create first simulation containing a simple environment: an arena with floor and walls, a few boxes, an e-puck robot and a controller program that will make the robot move.



1. **Tutorial 2: Modification of the Environment**

* in this step learn how to make a ball, by combining 2 shapes
* at this stage learn the concept of nodes, how to create nodes and node inheritance

A screenshot of a game

Description automatically generated with medium confidence

Graphical user interface, application, PowerPoint

Description automatically generated

1. **Tutorial 3: Appearance**

* at this stage learn the graphic display well. starting from coloring the objects and providing textures that can be imported from local images

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

1. **Tutorial 4: More about Controllers**

* This tutorial will introduce to the basics of robot programming in Webots. At the end of this chapter, you should understand what is the link between the scene tree nodes and the controller API, how the robot controller has to be initialized and cleaned up, how to initialize the robot devices, how to get the sensor values, how to command the actuators, and how to program a simple feedback loop.



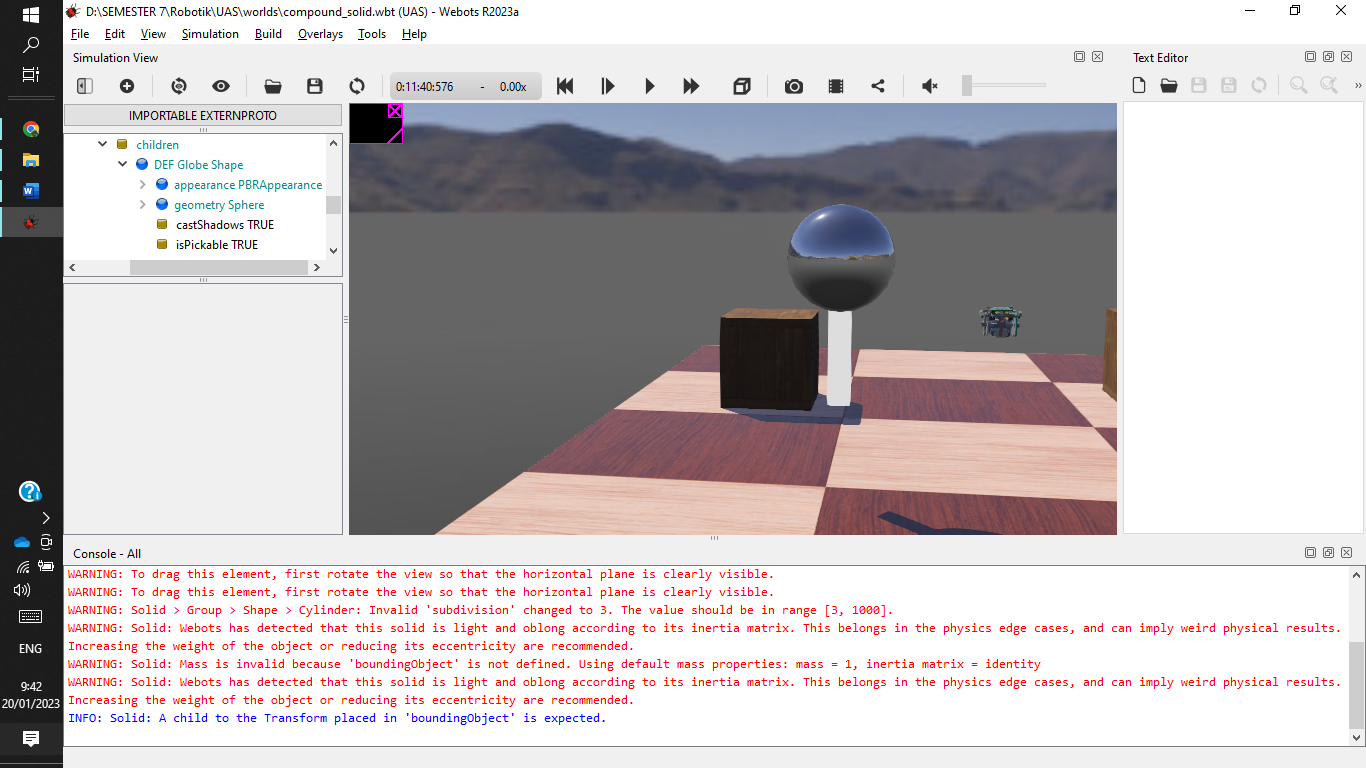
1. **Tutorial 5: Compound Solid and Physics Attributes**

* In the following simulation, learn more about physics, by combining several objects so that they become dumbbells
* there are three children who have their own inheritance, there are two balls and a tube which has a ball at each end
* step 1 create node cylinder, setup size, radius, and position

Graphical user interface

Description automatically generated

* the next step, transform the object by adding a transform node, and create a new sphere object node by inheriting the shape. Set up transformation x, y, z-axis until position at the end of the cylinder.



* copy the previous node and set the transform to be at the position of the other end of the tube

Graphical user interface

Description automatically generated

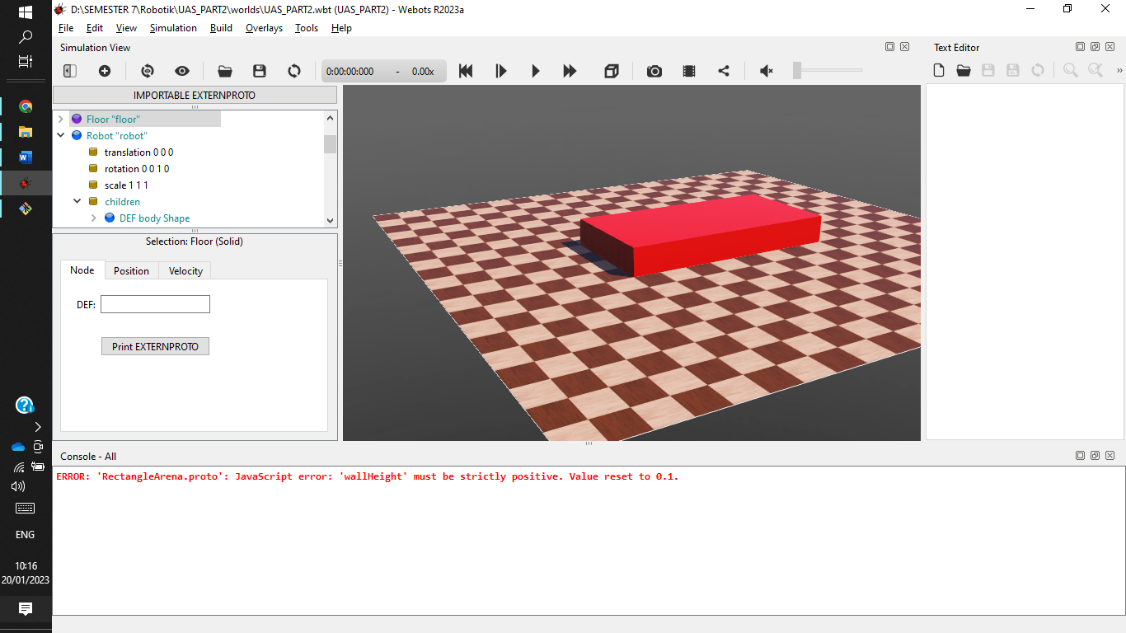
* finally , give a texture at the ball and cylinder so that beautifully object

Calendar

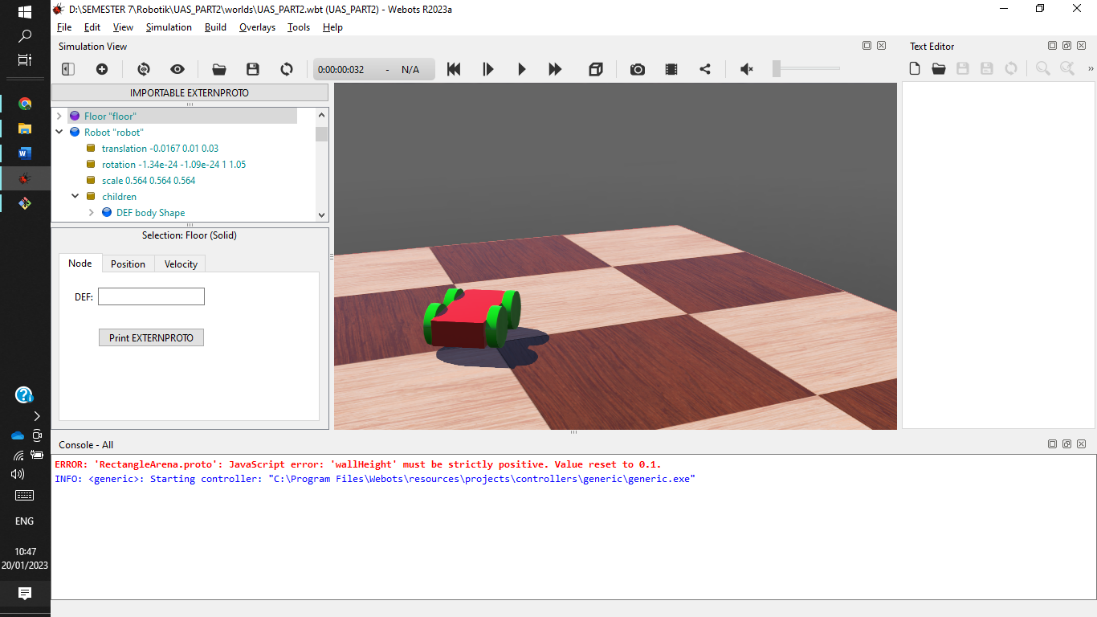
Description automatically generated

1. **Tutorial 6: 4-Wheels Robot**

* at this time will make a robot with a sensor
* first create node solid nodes in the form of a rectangular block



* the next step, make wheels on the robot using the hingejoint node and set the size and rotation, then define it as wheel1. so is done on the next 4 wheels

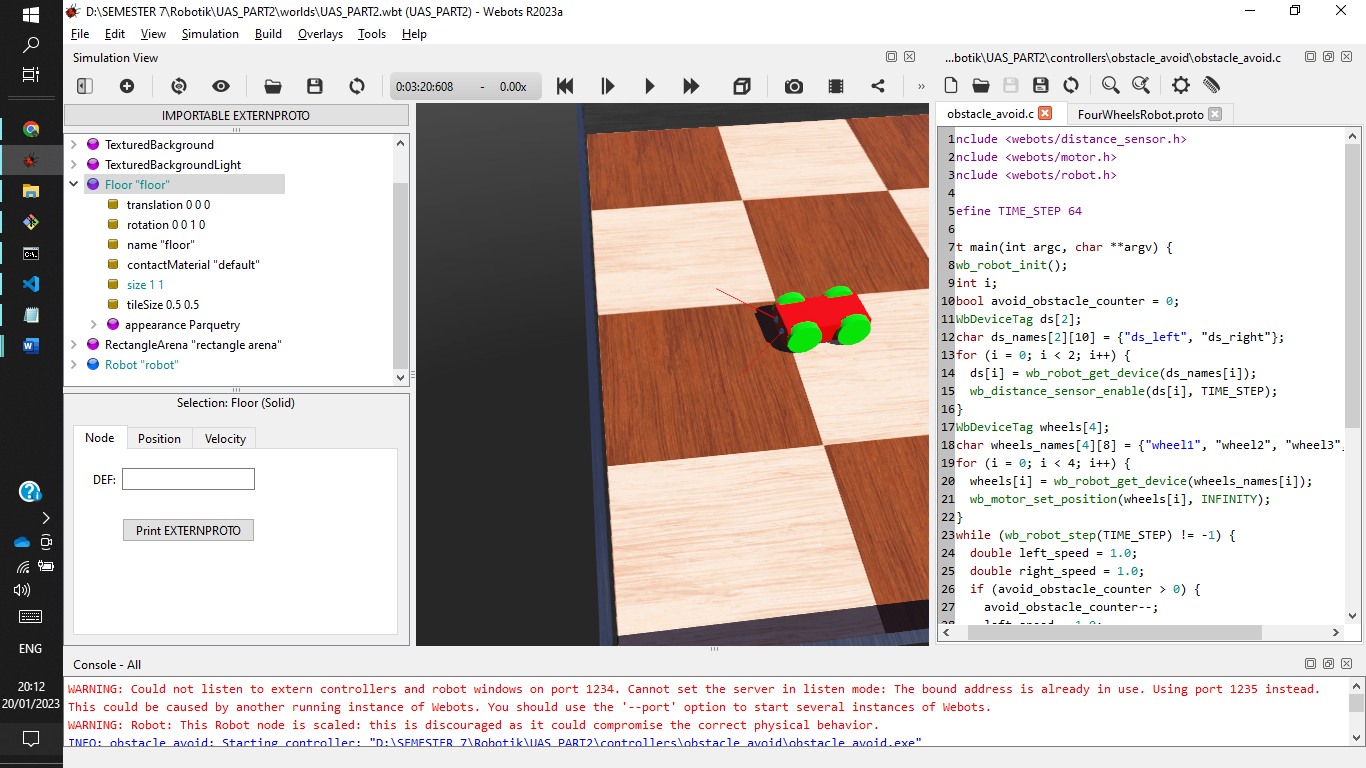


* then add a proximity sensor on the front of the robot

Graphical user interface, application

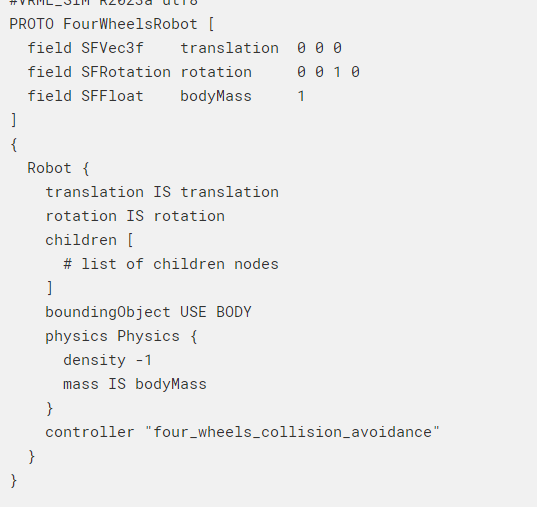
Description automatically generated

* then give the controller to the robot so it can walk



1. **Tutorial 7: Your First PROTO**

* create a new proto then initialize the following code



* after that copy the code in the .wbt file about the robot definition

Text, application

Description automatically generated

* adding the new node the new proto. Walla , now me have two same robots

Graphical user interface, application, PowerPoint

Description automatically generated

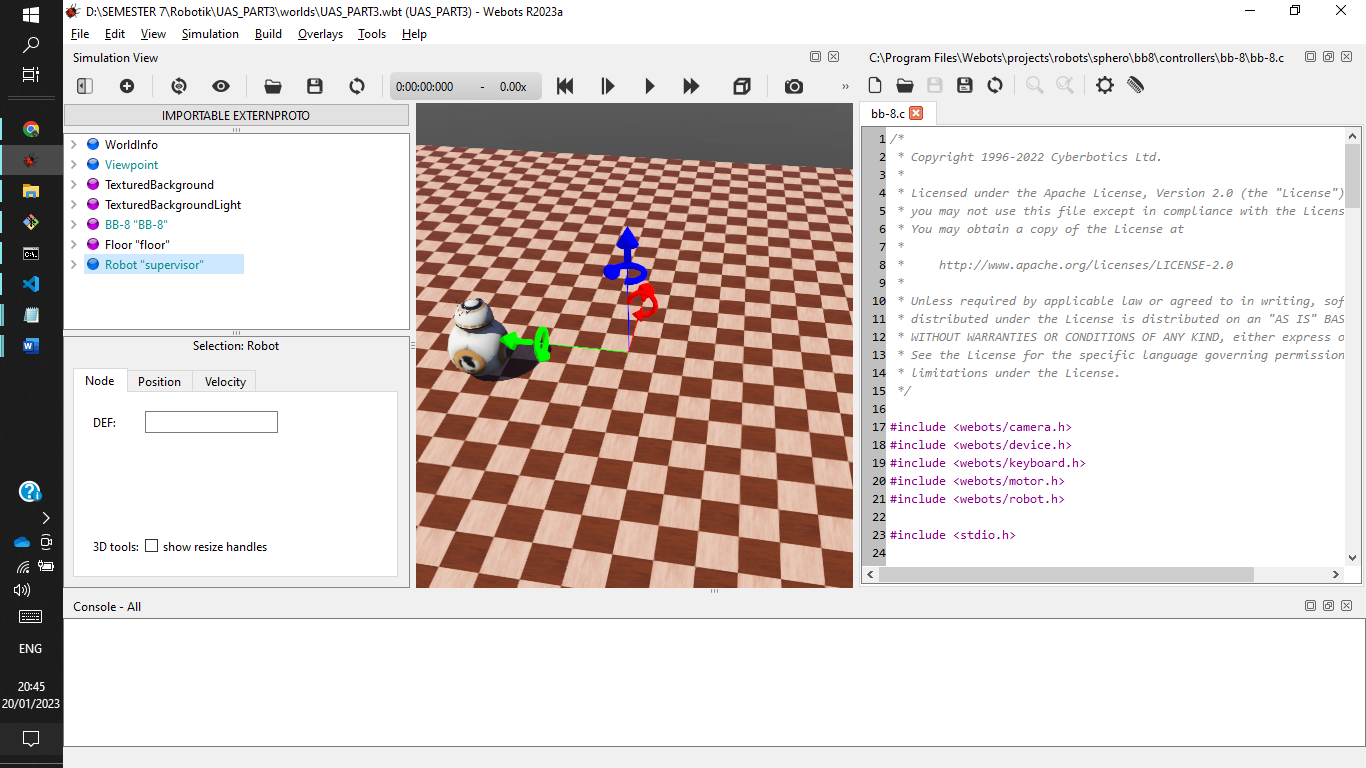
1. **Tutorial 8: The Supervisor**

* Create the new directory and add node robot bb-8

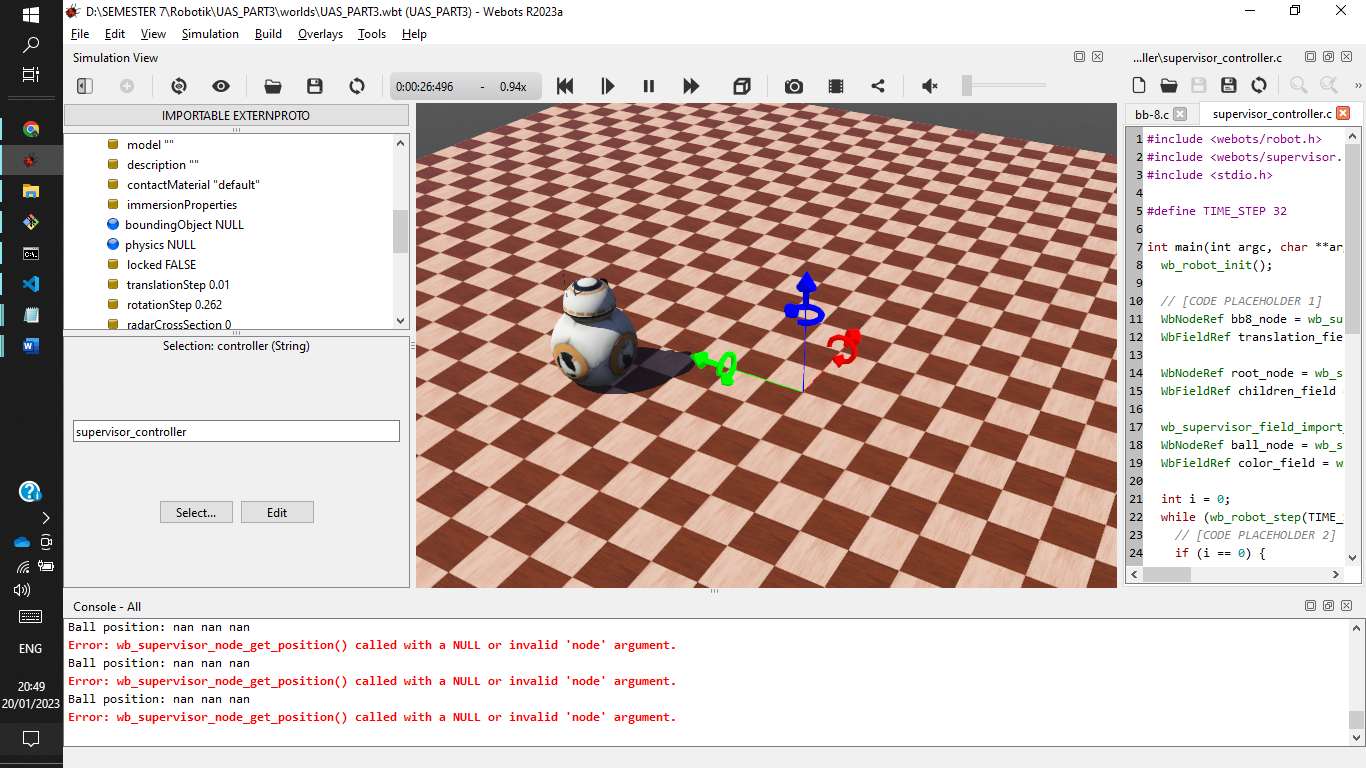
A screenshot of a computer

Description automatically generated with low confidence

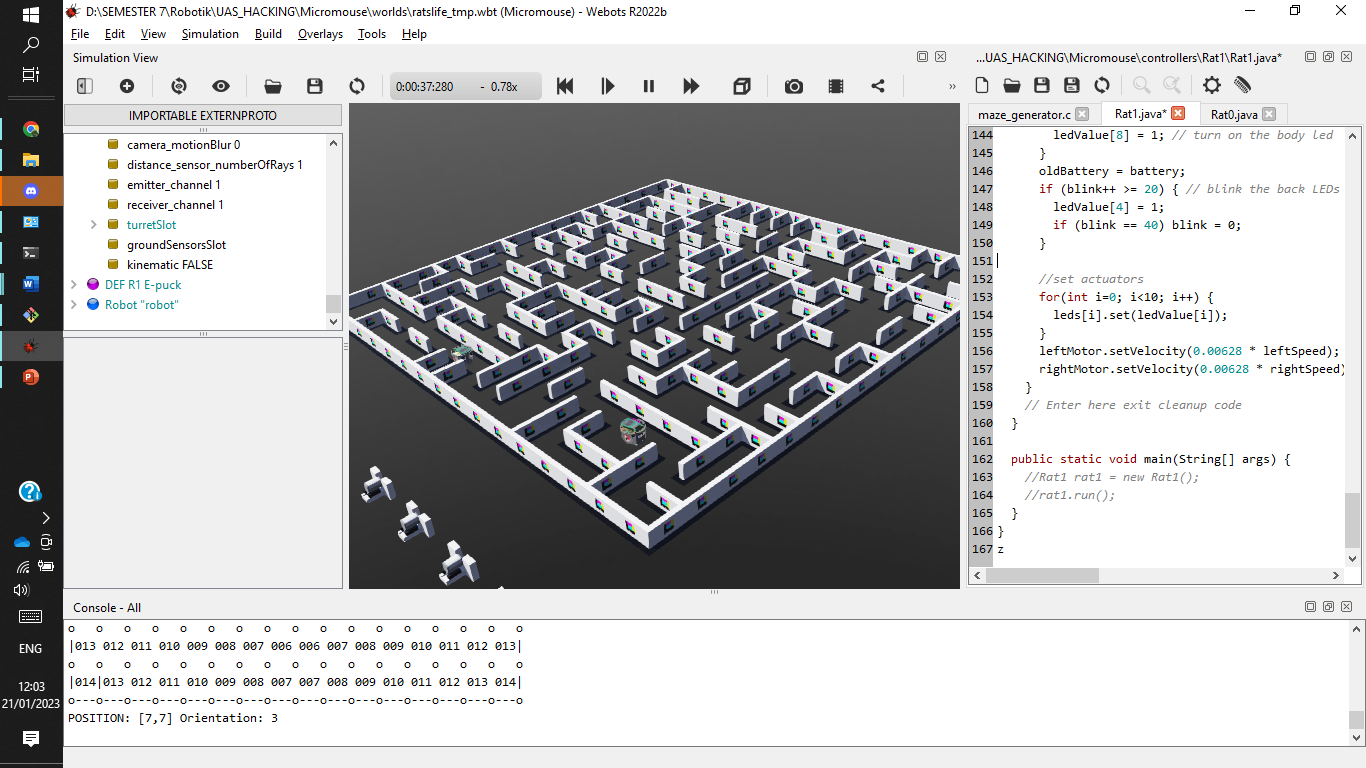
* Create new robot at base node and rename become “supervisor”

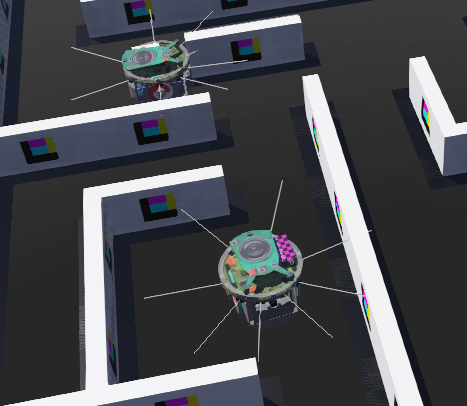


* Finally create new controller rename as “supervisor\_controller” and copy code web that. The result!



**HACKING MICROMOUSE**



This is a micromouse simulation, where there is a labyrinth and in it there is a robot that is programmed to go through the maze to its destination. the e-puck robot is programmed in 2 controllers, the first is rat0 and rat1. of the two controllers, only controller rat0 can work past the labyrinth path obstacles. 

For the robot used is the e-punk robot, there are 8 sensors located on each side of the robot, these sensors function to detect maze obstacles so that the robot can not hit obstacles