Topic	ROOM CLEANING ROBOT- II	
Class Description	**Students will use their knowledge about IMU sensors and complete the code for the room cleaning robot.**	
Class	**PRO C297**	
Class time	**45 mins**	
Goal	• Building the algorithm for automated cleaning robot	

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.** License	
WARM-UP SESSION - 10 mins		
Teacher Action		**Student Action**

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.	**ESR**: Hi, thanks! Yes I am excited about it! Click on the slide show tab and present the slides
WARM-UP QUIZ Click on In-Class Quiz	
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.	
TEACHER-LED ACTIVITY - 10 mins	
Teacher Initiates Screen Share	
• **Writing algorithm to find a corner position in the room from where the robot will start cleaning.**	
Teacher Action	**Student Action**
So what did we learn in our last class? *The teacher will clarify if there are any doubts!* So let's get started with today's class.	**ESR:** Varied!
Note: Make sure you save the project after every major step. Otherwise, you might	

lose your work.

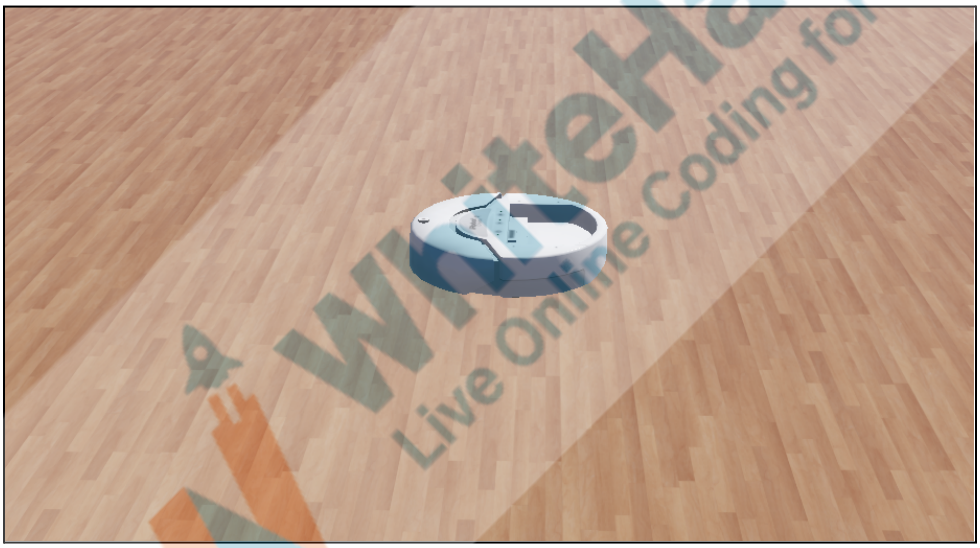
We were working on the room cleaning robot in the last class and we learnt about the inertial unit.

Let's complete our project today.

The teacher downloads the previous class code from Teacher Activity 1.

Let's get started with a new project.

1. Open **Webots** Robot Simulator.
2. Open the downloaded project.



Here, we had one **Inertial unit** which helped us understand the orientation of the robot at any point. We had also added 3 distance sensors in 3 directions - front, left and right.

We had written the code to turn the robot to a certain angle with the help of an IMU sensor. We have also written the code to move the robot forward until it detects an obstacle.

Today, we will write the rest of the code where the robot will move back and forth sweeping the room. 	
But to do this, the robot must find a corner of the room first. It will make sure that we start from one corner and cover the whole room. For that we will write a method named **find_corner().** This will be a step by step movement. First, the robot will orient itself towards 180 degrees using the value returned by the Inertial unit. Then, it will move forward until it reaches a wall. Once it reaches the wall, it turns towards its right and moves forward until it reaches a wall again. This way it will be in the corner. Let's write the code.	
1. As it is a step by step movement, we will need a variable which will keep track of the steps. Let's	

initiate a variable **movement** to **0**.

Also, initiate a variable named **robo_orientation** to **180**. This variable will keep track of the initial orientation of the robot.

```
21 ds_left.enable(timestep)
22 ds_right = robot.getDevice("ds_right")
23 ds_right.enable(timestep)
24
25 movement=0
26 robo_orientation=180
27
28 def move(left_speed,right_speed):
29     left_motor.setVelocity(left_speed)
```

2. Let's define the **find_corner()** method and it will require one parameter which will be **movement**.
3. Now, what should be the first movement?

ESR: Turn towards 180 degrees.

Let's write the code for it.

```
47 def find_corner(movement):
48     if(movement==0):
49         turn_towards_angle(robo_orientation)
```

4. Now, we want to increase the value of the **movement** variable by one once the robot has turned to that certain angle.

   To do this, we will change the code for the **turn_towards_angle()** method a little.

   a. We will add **movement** as a parameter for the **turn_towards_angle()** method.
   b. Once the robot is oriented towards that given angle, we will increase movement by 1. Return this value so that the next movement happens.

```
34 def turn_towards_angle(target_angle,movement):
35     if(target_angle != yaw_current):
36         move(-1,1)
37     else:
38         move(0,0)
39         movement+=1
40     return movement
```

Also, modify the **find_corner()** method accordingly.

```
def find_corner(movement):
    if(movement==0):
        movement=turn_towards_angle(robo_orientation%360, movement)
```

5. Now, we need to write code for the next movement where we want to go forward till a wall is detected.

   But we need to make changes in the **move_forward_till_wall_detection()** method. We need to increase the **movement** variable by 1 once the wall is detected.

   a. Pass a parameter named **movement** through **move_forward_till_wall_detection()** method.
   b. Increase the **movement** variable by 1 once the wall is detected and return the value.

```
43 def move_forward_till_wall_detection(movement):
44     if(ds_front_value>300):
45         move(15,15)
46     else:
47         movement+=1
48     return movement
49
```

6. Now, let's add the second movement in the **find_corner()** method. Here, we want to go straight

till the wall is detected.

```
51 def find_corner(movement):
52     if(movement==0):
53         movement=turn_towards_angle(robo_orientation%360, movement)
54     if(movement==1):
55         movement=move_forward_till_wall_detection(movement)
56
57
```

7. Now, the robot is close to one wall (even for other initial positions of the robot). But not in the corner.

To move it to the corner, we first turn it to **robo_orientation-90** degrees i.e. 90 degrees.

```
if(movement==2):
    movement=turn_towards_angle((robo_orientation-90)%360, movement)
```

8. Again, we want to go straight till the wall is detected.

```
if(movement==3):
    movement=move_forward_till_wall_detection(movement)
```

9. To move it to the corner, we turn it to **robo_orientation-180** degrees i.e. 0 degrees.

```
if(movement==4):
    movement=turn_towards_angle((robo_orientation-180)%360, movement)
```

10. Also, return the **movement** variable at the end of the **find_corner()** method.

```
50
51 def find_corner(movement):
52     if(movement==0):
53         movement=turn_towards_angle(robo_orientation%360, movement)
54     if(movement==1):
55         movement=move_forward_till_wall_detection(movement)
56     if(movement==2):
57         movement=turn_towards_angle((robo_orientation-90)%360, movement)
58     if(movement==3):
59         movement=move_forward_till_wall_detection(movement)
60     if(movement==4):
61         movement=turn_towards_angle((robo_orientation-180)%360, movement)
62
63     return movement
64
```

11. Let's call the **find_corner()** method inside the main while loop.

```
while robot.step(timestep) != -1:

    ds_front_value= ds_front.getValue()
    ds_left_value= ds_left.getValue()
    ds_right_value= ds_right.getValue()

    angle=imu.getRollPitchYaw()

    # the range was -180 to 180 , we change it to 0 to 360
    yaw_current=round(math.degrees(angle[2]))+180

    movement = find_corner(movement)

```

Reference Code:

```
def turn_towards_angle(target_angle,movement):
    if(target_angle != yaw_current):
        move(-1,1)
    else:
        move(0,0)
        movement+=1
    return movement


def move_forward_till_wall_detection(movement):
    if(ds_front_value>300):
        move(15,15)
    else:
        movement+=1
    return movement


def find_corner(movement):
    if(movement==0):
        movement=turn_towards_angle(robo_orientation%360, movement)
    if(movement==1):
        movement=move_forward_till_wall_detection(movement)
    if(movement==2):
        movement=turn_towards_angle((robo_orientation-90)%360, movement)
    if(movement==3):
        movement=move_forward_till_wall_detection(movement)
    if(movement==4):
        movement=turn_towards_angle((robo_orientation-180)%360, movement)

    return movement
```

```
89
90 while robot.step(timestep) != -1:
91
92     ds_front_value= ds_front.getValue()
93     ds_left_value= ds_left.getValue()
94     ds_right_value= ds_right.getValue()
95
96     angle=imu.getRollPitchYaw()
97
98     # the range was -180 to 180 , we change it to 0 to 360
99     yaw_current=round(math.degrees(angle[2]))+180
100
101    movement = find_corner(movement)
102
```

Reference Output:

Click here to view the reference output.

Now, we have the code to move our robot to one corner.	

Let's code the back and forth movement.	
Are you ready?	**ESR:** Yes.
Teacher Stops Screen Share	
So now it's your turn. Please share your screen with me.	
We have one more class challenge for you. Can you solve it? Let's try. I will guide you through it.	
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 • **The student writes code for the back and forth movement of the cleaning robot.**	
Teacher Action	**Student Action**
Teacher guides the student to download the boilerplate code from Student Activity 1.	*Student downloads the boilerplate code from Student Activity 1.*
The teacher helps the student to write the code. The student takes the lead and tries to figure out how to solve this problem.	
Once the robot finds a corner, we want to write the code to move it in a back and forth motion.	

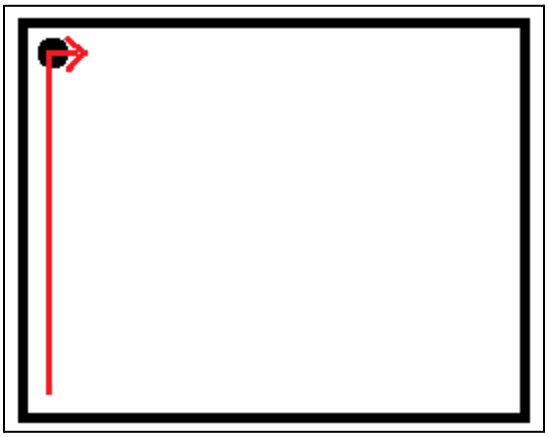
Let's write the code for it.

1. Once the **find_corner()** method is executed, the robot is now in the corner facing at an angle 0 zero with respect to a global coordinate system. The default global coordinate system used in Webots is called ENU. "ENU" means East along the X-positive axis, North along the Y-positive axis, and Up along the Z-positive axis.

   Now, we want to move it forward till the wall is detected. This will be movement number 5.

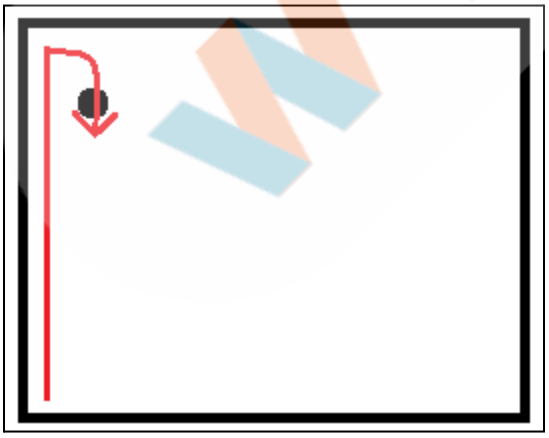
```
64
65 def back_and_forth_movement(movement):
66     if(movement==5):
67         movement=move_forward_till_wall_detection(movement)
68
```

2. As we have reached a wall, the robot needs to turn towards its right hand side now. This will be a point turn and we will call the **turn_towards_angle()** method to this.



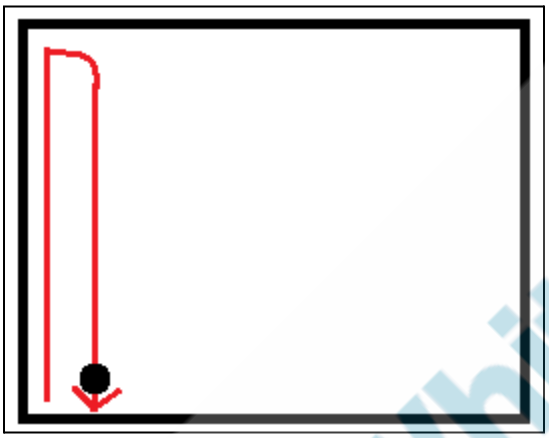
```
64
65 def back_and_forth_movement(movement):
66     if(movement==5):
67         movement=move_forward_till_wall_detection(movement)
68     if(movement==6):
69         movement=turn_towards_angle((robo_orientation+90)%360, movement)
70
71
```

3. We are facing the right hand side now. After this, we want to take a differential turn and face the angle of 180 degrees with respect to a global coordinate system.



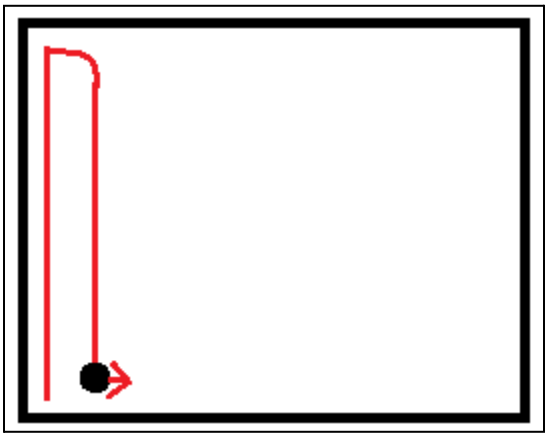
```
if(movement==7):
    if(robo_orientation%360 != yaw_current):
        move(3,1)
    else:
        movement+=1
```

4. After that, move forward again till the robot reaches the next wall.



```
if(movement==8):
    movement=move_forward_till_wall_detection(movement)
```

5. Turn the robot towards its right hand side i.e. 270 degrees with respect to a global coordinate system.



```
if(movement==9):
    movement=turn_towards_angle((robo_orientation+90)%360, movement)
```

6. After this, we want to take a differential turn and face the angle of 0 degree with respect to a global coordinate system.

   At the end, we want to go back to step 5 to continue this motion.

```
if(movement==10):
    if((robo_orientation-180)%360 != yaw_current):
        move(1,3)
    else:
        move(0,0)
        movement=5
```

Reference code:

```
def back_and_forth_movement(movement):

    if(movement==5):
        movement=move_forward_till_wall_detection(movement)
    if(movement==6):
        movement=turn_towards_angle((robo_orientation+90)%360, movement)

    if(movement==7):
        if(robo_orientation%360 != yaw_current):
            move(3,1)
        else:
            movement+=1
    if(movement==8):
        movement=move_forward_till_wall_detection(movement)
    if(movement==9):
        movement=turn_towards_angle((robo_orientation+90)%360, movement)
    if(movement==10):
        if((robo_orientation-180)%360 != yaw_current):
            move(1,3)
        else:
            move(0,0)
            movement=5
    return movement
```

7. Let's call the **back_and_forth_movement()** method in the main while loop.

```
while robot.step(timestep) != -1:

    ds_front_value= ds_front.getValue()
    ds_left_value= ds_left.getValue()
    ds_right_value= ds_right.getValue()

    angle=imu.getRollPitchYaw()

    # the range was -180 to 180 , we change it to 0 to 360
    yaw_current=round(math.degrees(angle[2]))+180

    movement = find_corner(movement)

    movement=back_and_forth_movement(movement)
```

8. Now, the robot will reach the other end of the room with the back and forth motion. After that we want to change the orientation of the robot and make it go back and forth sideways. So, we will write the code for that.
    a. For this we will initiate a new variable named **reached_end** to **0.**
    b. Then, we will check the condition to reach the other end.

```
"""
when it reaches the other corner for the first time, we change the robo_orientation to 270.
this will make it sweep the floor sideways now.
"""
if(reached_end==0 and ds_front_value<500 and ds_left_value<1000 and (yaw_current==180 or yaw_current==270)):
    movement=0
    robo_orientation=270
    reached_end=1
```

9. When it reaches the corner again after sweeping sideways, we stop the process by changing the value of **reached_end** variable to 2.

```
"""
when it reaches the corner again after sweeping sideways, we stop the process by changing
the value of reached_end variable to 2
"""
if(ds_front_value<500 and ds_left_value<500 and (yaw_current==360 or yaw_current==0)  and reached_end==1):
    reached_end=2
```

Reference Code:

```
while robot.step(timestep) != -1:
    ds_front_value= ds_front.getValue()
    ds_left_value= ds_left.getValue()
    ds_right_value= ds_right.getValue()

    angle=imu.getRollPitchYaw()

    # the range was -180 to 180 , we change it to 0 to 360
    yaw_current=round(math.degrees(angle[2]))+180

    movement = find_corner(movement)

    movement=back_and_forth_movement(movement)

    """
    when it reaches the other corner for the first time, we change the robo_orientation to 270.
    this will make it sweep the floor sideways now.
    """
    if(reached_end==0 and ds_front_value<500 and ds_left_value<1000 and (yaw_current==180 or yaw_current==270)):
        movement=0
        robo_orientation=270
        reached_end=1

    """
    when it reaches the corner again after sweeping sideways, we stop the process by changing
    the value of reached_end variable to 2
    """
    if(ds_front_value<500 and ds_left_value<500 and (yaw_current==360 or yaw_current==0)  and reached_end==1):
        reached_end=2
```

Reference Output:

Click here to view the complete output video.

Great work!	

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 start working on a self-driving car.	*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 	

ACTIVITY LINKS		
Activity Name	**Description**	**Links**
Teacher Activity 1	Previous Class Code	https://github.com/procodingclass/PRO-C296-Reference-Code
Teacher Reference 1	Reference Code	https://github.com/procodingclass/PRO-C297-Reference-Code
Teacher Reference 2	Project	
Teacher Reference 3	Project Solution	
Teacher Reference 4	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/a7fa223c-b7ed-43c2-acdb-0639a0731dcf.pdf
Student Activity 1	Boilerplate Code	https://github.com/procodingclass/PRO-C297-Student-Boilerplate