Topic	SELF DRIVING CAR 2	
Class Description	Students will learn how to attach a camera to the car and create an algorithm for the “self driving car”.	
Class	PRO C299	
Class time	50 mins	
Goal	• Attaching a camera node to the car. • Algorithm for the self driving car.	
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	5 mins 20 mins 20 mins 5 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

<table>
<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" align="center">WARM-UP QUIZ
Click on In-Class Quiz</td>
</tr>
<tr>
<td colspan="2">Activity Details

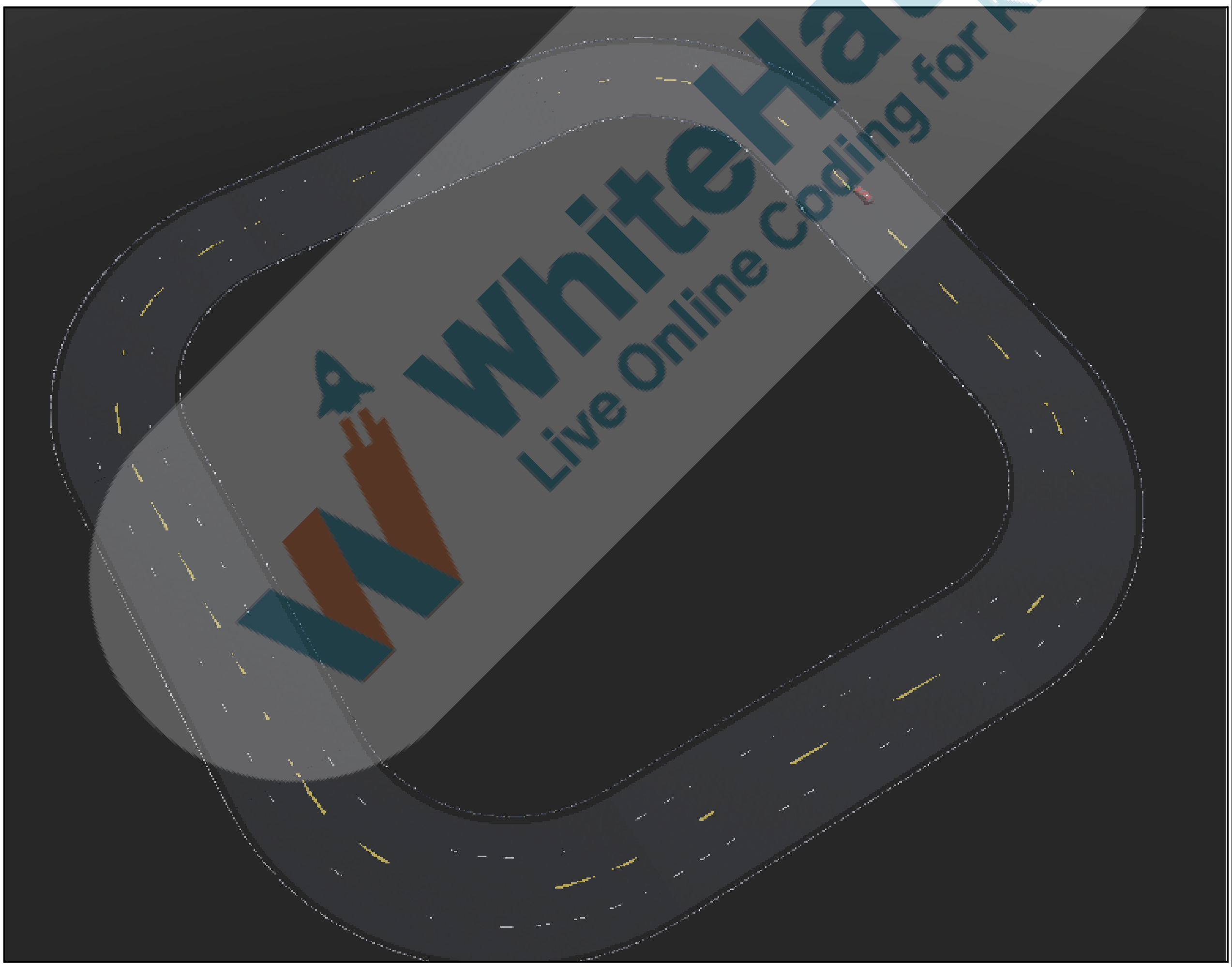
Following are the session deliverables:
● Appreciate the student.
● Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.</td>
</tr>
<tr>
<td colspan="2" align="center">TEACHER-LED ACTIVITY - 15 mins</td>
</tr>
<tr>
<td colspan="2" align="center">Teacher Initiates Screen Share</td>
</tr>
<tr>
<td colspan="2"> <u>ACTIVITY</u> ● Adding a camera node to the BMW X5 car node.</td>
</tr>
<tr>
<td align="center">Teacher Action</td>
<td align="center">Student Action</td>
</tr>
<tr>
<td>Do you remember what we did in the last class?

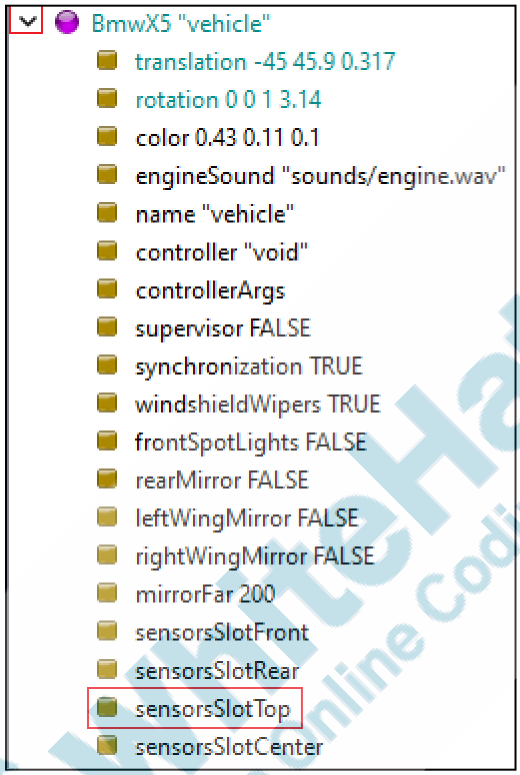
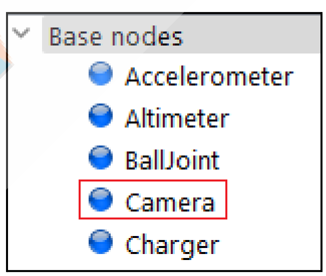
Great, if you have any doubts from the last class, please ask.

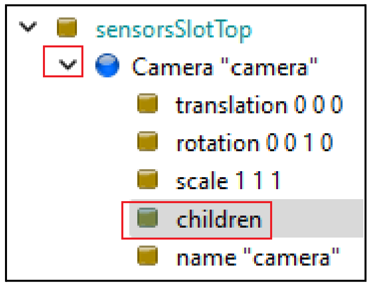
Note : Teacher will clear the doubts, if students have any.

Now that you don’t have any questions from the previous classes, let’s learn something new today.</td>
<td>ESR : Yes, we started creating a self driving car.</td>
</tr>
</table>

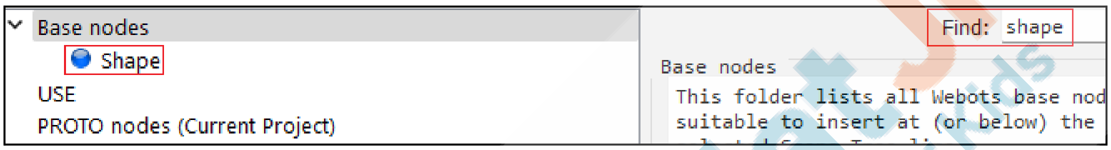
In the previous class, we were able to successfully move the car, but there was one problem. Can you tell me what the problem was? Great, if we want to make a **self driving car**, let's first attach a **camera node** with it, so that it can get information from its external surroundings. For that, let's first open the Teacher Activity 1 and download all the files from here. *Note : The boilerplate link has the file with a bigger and a modified track.*	ESR : Yes, the car was just moving straight.



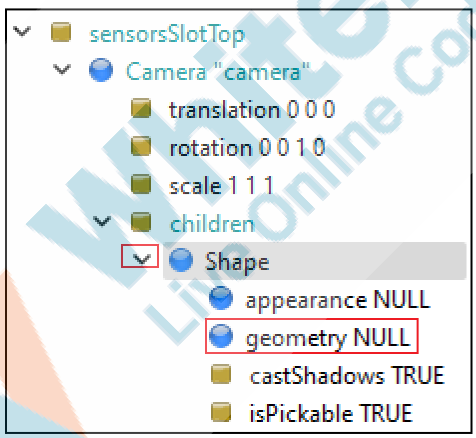
Once you have opened the downloaded files in the webots software, **expand** the **BMW X5 node** and **double click** on the **sensorSlotTop** property.	
	
A window will open which will ask you to add a new node. **Expand** the **Base nodes** and add a **camera** node.	
	
To give our camera a **definite shape** and **appearance, expand** the **camera node** and **double click** on the **children** property.	



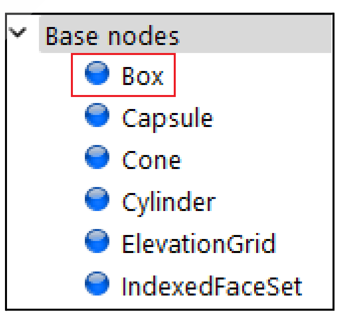
Search for the **Shape** node and add it.



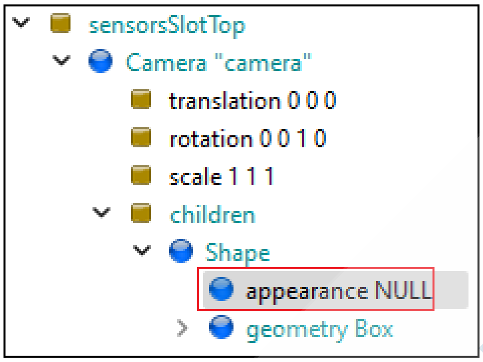
To add a **definite geometry**, **expand** the **Shape node** and **double click** on the **geometry** property.



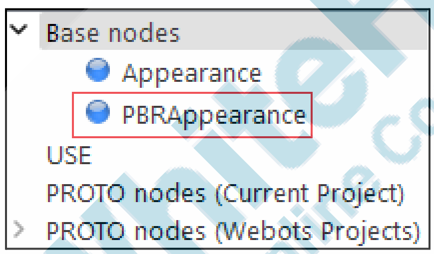
Add the **Box** node and don't change its **size**.



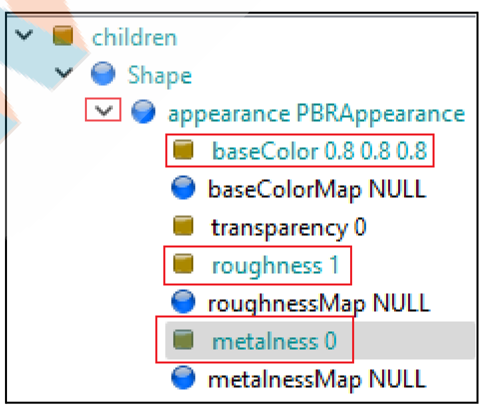
Next, let's give our box an appearance. For that, **double click** on the **appearance** property.



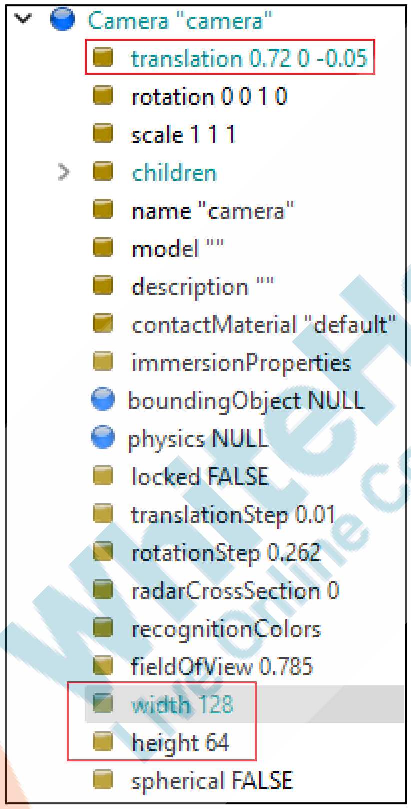
Expand the **Base nodes** and add the **PBR Appearance** node.



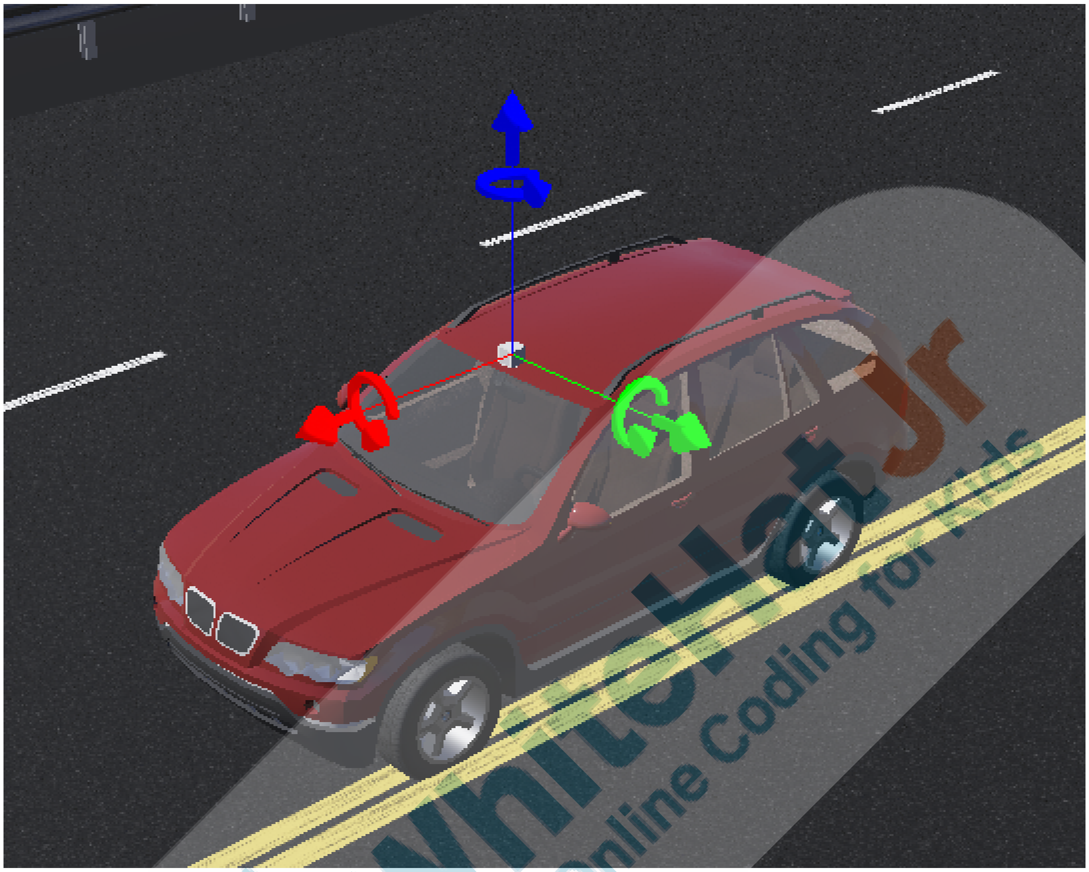
To change the appearance property, let's first **expand** it. Next, change the **base color** to **0.8** in **red, green** and **blue**. Also change the **roughness** to **1** and **metalness** to **0**.



Next, let's change the camera **translation** to **0.72** in the **X direction** and **-0.05** in the **Z direction**. Also, for better image processing, let's change the **resolution** of the camera to **128 pixels** in **width** and **64 pixels** in **height**.	



Finally we are all set with our camera. Save your work till here.	



So now it's your turn.

Please share your screen with me.

STUDENT-LED ACTIVITY 15 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

- **Writing an algorithm for driving a car autonomously.**

Teacher Action	Student Action
Once we have added the camera, let's write an appropriate code so that we can drive the vehicle autonomously. But before writing the algorithm, let's discuss what you think could be the algorithm for a self driving car. There are multiple algorithms for creating autonomous vehicles. We are going to use a simple one. We know we have **yellow colored track** on the **road**, we will write an algorithm where the car will **move** or **turn** in such a way, such that the yellow track is always in the **center** of the **camera vision**. To understand it better, let's break the algorithm into its child steps, ● Get the **image** from the **camera**. ● Look for all the **yellow pixels** in the **image** and **store** the **x coordinates** of all the **yellow pixels** in a **list**. ● Find the **average x coordinate** of all the **yellow pixels**. ● Find the **center x coordinate** of the **image**. ● **Steer** the **car** in such a way, so that the **average x coordinate** calculated in **step 3**, remains in the **center** of the image. For that, let's open the student boilerplate link, and download all the files from it. Open the downloaded files in webots software. In the **sdc1.py** file, ● Fetch the device named **"camera"** using the **.getDevice()** method. ● **Enable** is using the **.enable()** method.	 **ESR** : Varied

```python
from controller import Robot

bot = Robot()

timestep = 64

# getting devices
cam = bot.getDevice('camera')
left_wheel = bot.getDevice('left_front_wheel')
right_wheel = bot.getDevice('right_front_wheel')
l_steer = bot.getDevice('left_steer')
r_steer = bot.getDevice('right_steer')

# initialisations
cam.enable(timestep)
left_wheel.setPosition(float('inf'))
right_wheel.setPosition(float('inf'))
l_steer.setPosition(0)
r_steer.setPosition(0)
left_wheel.setVelocity(0)
right_wheel.setVelocity(0)
```

Next, in the main loop,

- Get an **image** from the camera using the **.getImage()** method.
- Get the **image width** using the **.getWidth()** method.
- Get the **image height** using the **.getHeight()** method.

```python
# main loop
while bot.step(timestep) != -1:

    # image data
    img = cam.getImage()
    image_width = cam.getWidth()
    image_height = cam.getHeight()
```

Next,

- Let's take an **empty list** and name it as **x_yellow**.
- Let’s traverse through each and every pixel of our image and calculate the **red** content in the pixel color using the **.imageGetRed()** method.

• Similarly, find the **green** content in the pixel color using the **.imageGetGreen()** method. • Similarly, find the **blue** content in the pixel color using the **.imageGetBlue()** method. • Once you have the **red, green** and **blue** content of the pixel color, check if it is a **yellow colored pixel** or not by using the **if** statement. • If it's a **yellow** pixel, add its **x coordinate** in the **x_yellow list** using the **.append()** method. *Note : The red, green and blue content in yellow color is 190, 180 and 90 respectively.*	

```
x_yellow = []
for x in range(0, image_width):
    for y in range(0, image_height):
        red_val = cam.imageGetRed(img, image_width, x, y)
        green_val = cam.imageGetGreen(img, image_width, x, y)
        blue_val = cam.imageGetBlue(img, image_width, x, y)
        if red_val > 190 and green_val > 180 and blue_val > 90:
            x_yellow.append(x)
```

Next, • Let's check if the **x_yellow** list is **empty** or **not**. • If it's **not empty**, let's traverse through the list and calculate the **average x coordinate** for all the yellow pixels in the list.	

```
if x_yellow: # if there are any yellow pixels
    x_total = 0
    for x in x_yellow:
        x_total = x_total + x
    x_average = x_total / len(x_yellow)
```

Next, let's find the **x coordinate** of the **center pixel** of the image by simply **dividing** the **image width** by **2**.	

```
x_center = image_width / 2
```

Next,	

- Let's compare the **average x coordinate** and the **center x coordinate** using the **conditional statements**. - If **x_average** is **less than x_center**, which means maximum yellow pixels are on the **left** side of the image, which means we have to take a **left** turn. - To take a **left** turn, set the positions of the **steering motors** to **-0.1**. - If **x_average** is **more than x_center**, which means maximum yellow pixels are on the **right** side of the image, which means we have to take a **right** turn. - To take a **right** turn, set the positions of the **steering motors** to **0.1**.	

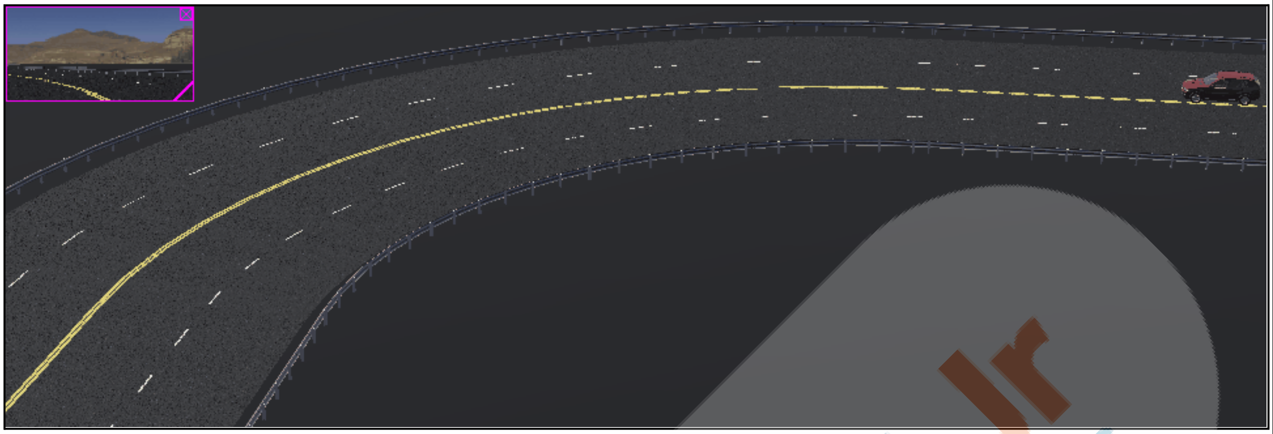
```
if x_average < x_center: # max pixels are on the left, take a left turn
    l_steer.setPosition(-0.1)
    r_steer.setPosition(-0.1)
elif x_average > x_center: # max pixels are on the right, take right turn
    l_steer.setPosition(0.1)
    r_steer.setPosition(0.1)
```

Finally, let's move our car in the **forward** direction by giving a **velocity** of **10 rad/s** to the **wheels**.	

```
# move forward
left_wheel.setVelocity(10)
right_wheel.setVelocity(10)
```

Save your work till here and run the simulation.	

Reference Output:



Click here to view the reference video.

Teacher Guides Student to Stop Screen Share

WRAP-UP SESSION - 5 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

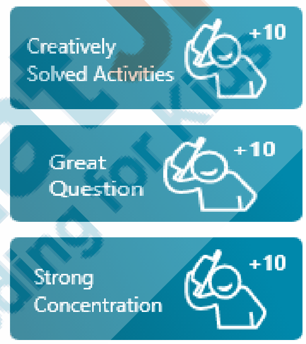
Following are the session deliverables:

- 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 complete our self-driving car by adding obstacle detection abilities to it.	*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	Teacher Boilerplate Code	https://github.com/procodingclass/PRO-C299-Teacher-Boilerplate
Teacher Reference 1	Project	
Teacher Reference 2	Project Solution	

Teacher Reference 3	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/66130515-6618-490f-bb46-d076d83efcff.pdf
Teacher Reference 4	Reference code	https://github.com/procodingclass/PRO-C299-Reference-Code.git
Teacher Reference 5	Final output gif	https://s3-whjr-curriculum-uploads.whjr.online/94234701-bb62-4a99-ab52-ecfe7637590f.gif
Student Activity 1	Student Boilerplate Code	https://github.com/procodingclass/PRO-C299-Student-Boilerplate