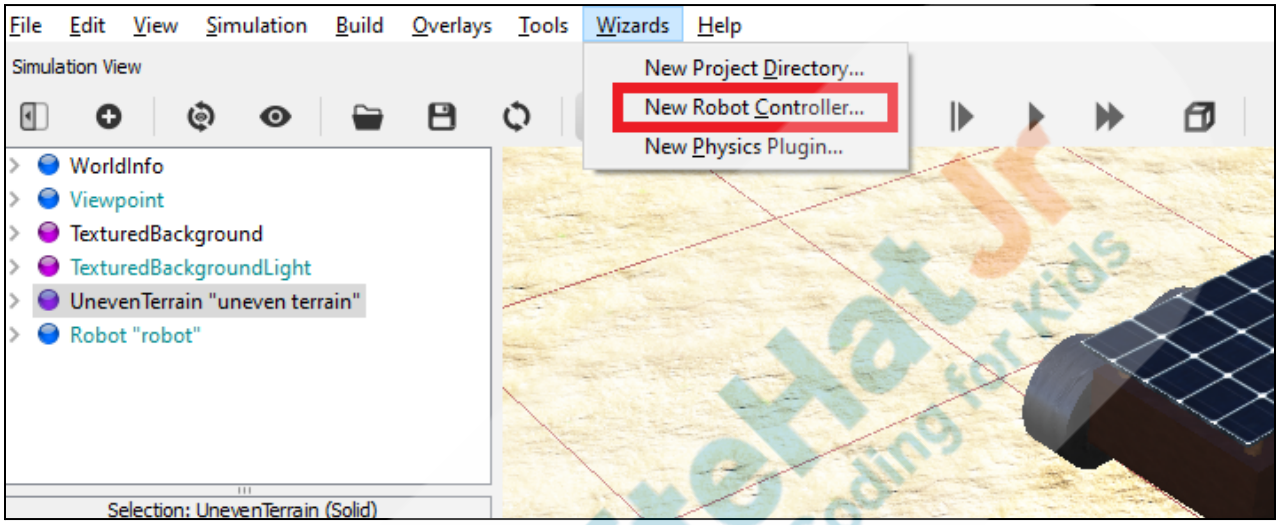
Topic	Planet Exploration Robot - 2	
Class Description	**Students will be introduced to the keyboard controls in Webots. They will learn to move the rover forward, backward left and right when the WSAD keys are pressed.**	
Class	**PRO C285**	
Class time	**45 mins**	
Goal	• Adding a controller to the Planet Exploration Rover. • Move the rover using the keyboard.	

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	

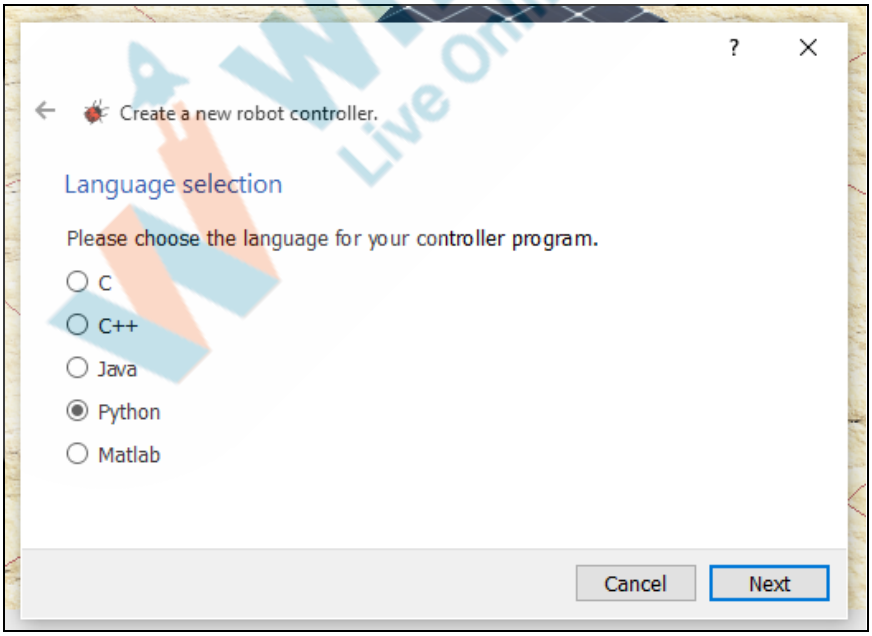
Class structure	**Warm-Up** **Teacher-Led Activity 1** **Student-Led Activity 1** **Wrap-Up**	**10 mins** **10 mins** **20 mins** **05 mins**
Credit & Permissions:	**This project uses Webots, an open-source mobile robot simulation software developed by Cyberbotics Ltd.** **License**	
WARM-UP SESSION - 10 mins		

Teacher Action	Student Action
Hey <student's name>. How are you? Did you look up other rovers on the internet? What did you learn? **Following are the WARM-UP session deliverables:** • Greet the student. • Revision of previous class activities. • Quizzes.	**ESR**: Hi, varied **ESR:** Varied. 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	
• **Adding Robot Controller to Planet Exploration Robot**	
Teacher Action	**Student Action**
Today we will add a controller to the planet exploration Rover. We will program it so that the rover can be controlled using the keyboard. *Teacher opens Teacher Activity 1 and starts creating.* Do you remember how to add a controller to the robot?	**ESR:** By clicking on **Wizards > New Robot**

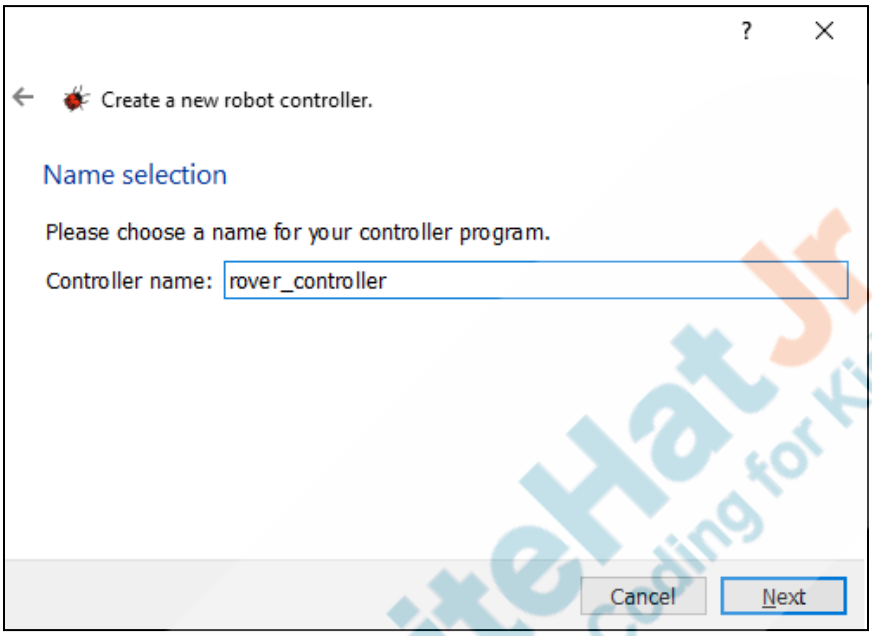
That is correct, we can add a controller using the **Wizards > New Controller** menu.	**Controller**



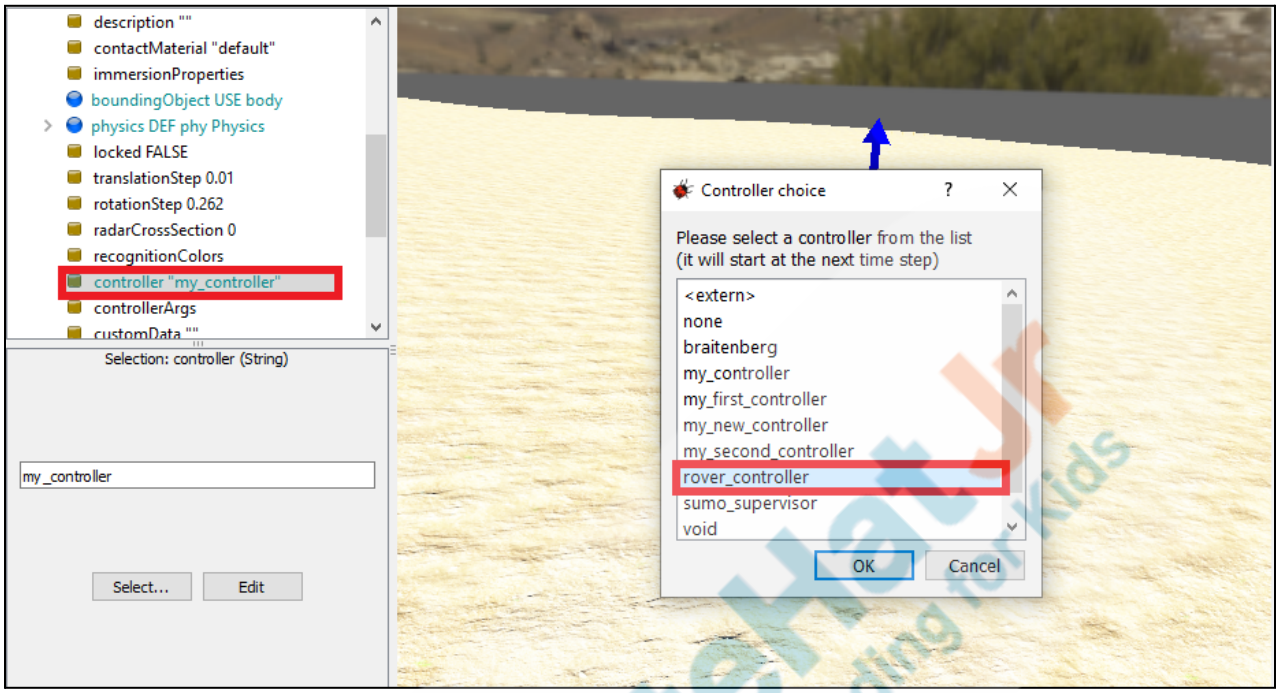
Remember to select **Python**



The name that you give is important since the same name has to be selected for the Robot.



Select the controller that you just created under **Robot - controller**



Now that the controller is ready, let's start coding. What do we have to import first in the controller file? Let us import the Robot from the controller. We will also import Keyboard since we need to use Keyboard functions. Keyboard is a set of functions that are available for the Robot to by default for each Robot node to read the keyboard input. We will create an instance of the Robot called robot and of the Keyboard called keyboard	**ESR:** import the Robot

```
from controller import Robot
from controller import Keyboard

robot=Robot()
keyboard = Keyboard()
```

The Robot has a function called getDevice(name) that returns the unique identifier device with the given name.

wheel1_left=robot.getDevice("wheel1_left")

Will return a unique identifier for the device of the first left wheel, that is it will return the Motor in the left wheel.

Once we get the motor device, it can be used to control velocity. This is done with two function calls:
setPosition(inf) and **setVelocity(speed)**. This will initiate a continuous motor motion at the desired speed

inf stands for INFINITY and is passed as the position parameter.
We will set the initial velocity as zero.

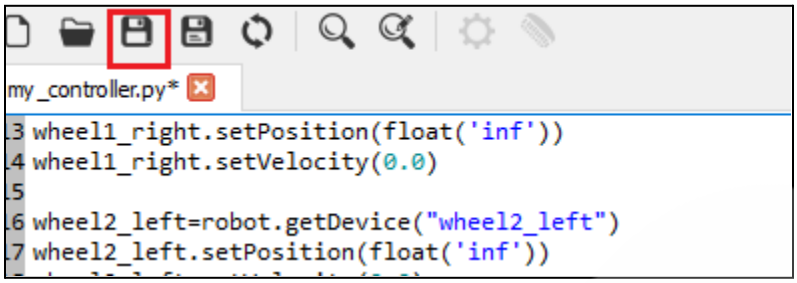
```
wheel1_left=robot.getDevice("wheel1_left")
wheel1_left.setPosition(float('inf'))
wheel1_left.setVelocity(0.0)

wheel1_right=robot.getDevice("wheel1_right")
wheel1_right.setPosition(float('inf'))
wheel1_right.setVelocity(0.0)

wheel2_left=robot.getDevice("wheel2_left")
wheel2_left.setPosition(float('inf'))
wheel2_left.setVelocity(0.0)

wheel2_right=robot.getDevice("wheel2_right")
wheel2_right.setPosition(float('inf'))
wheel2_right.setVelocity(0.0)
```

In Webots, **basicTimeStep** indicates the duration of one step of simulation, that is. the time interval between two computations of the position, speed, collisions, etc. This is used to synchronize our program and the simulator condition. **basicTimeStep** is equal to 32 milliseconds. We can set the timeStep to be multiples of 32. Today we will set it as 64.

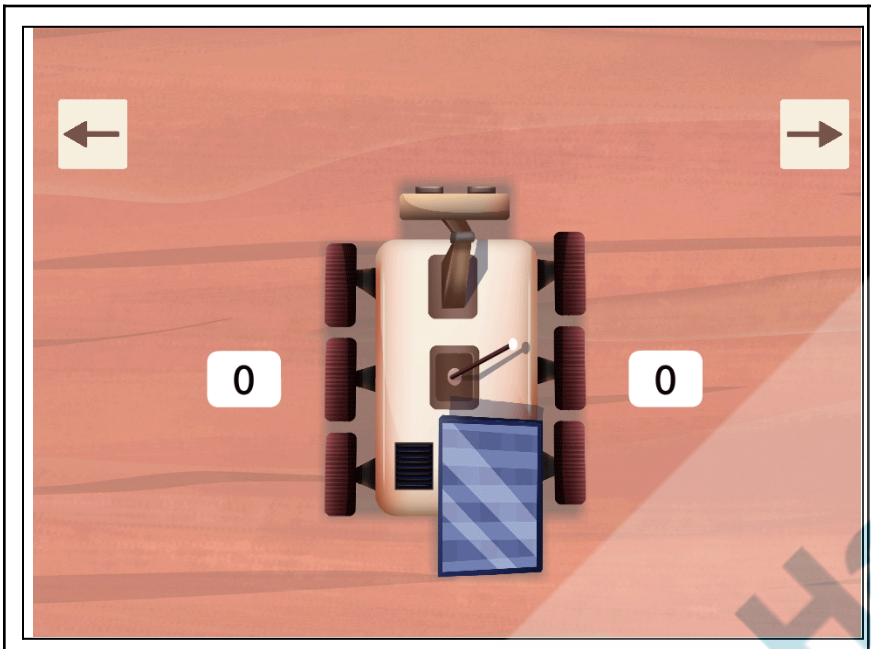
Let us press the save button to save the work done so far.

```
timestep=64
```

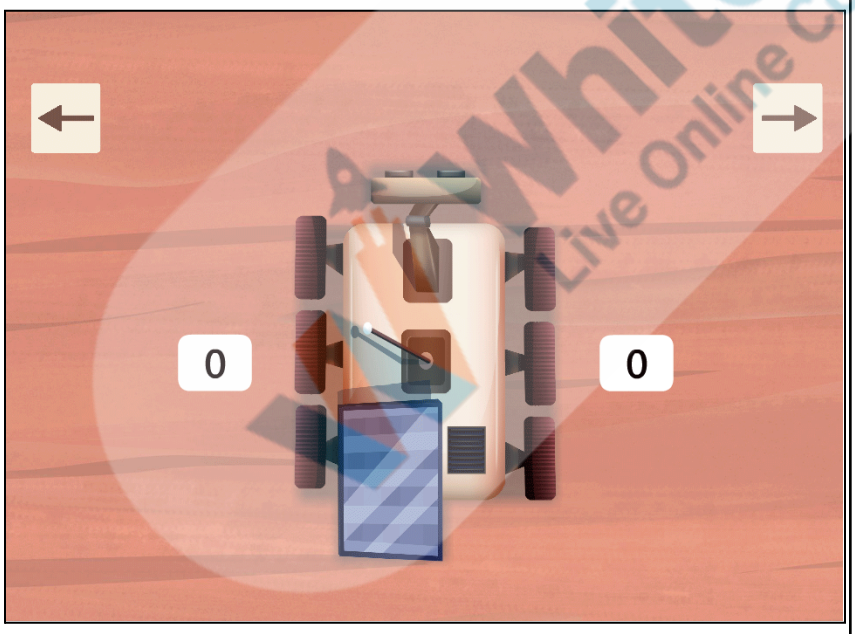
Save button



Now that the motor and the timeStep is ready it is ready to bring in the keyboard controls. We will use the **WSAD** keys to move the rover . Can you tell me what these keys are used for?	**ESR: WSAD** is used to move forward, backward, left and right. This is used instead of arrow keys.
That is correct, these keys can be used to move the rover to all sides. We will use the arrow keys for something else in the upcoming classes. Let us first understand how vehicles with four or more wheels make a turn. A turn of a vehicle can be implemented in 3 ways- A vehicle turns when there is a difference between the speed of its left wheels and the right wheels.	



Click here to view the video.

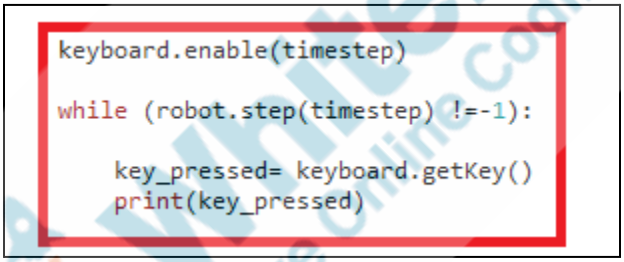


Click here to view the video.

1. In the first kind of turn, the speed of the wheels on one side is 0. And we assign some forward motion to the other side of the wheels. Let's say if we want to turn right, we will assign a positive value to the **left wheels** and 0 to the **right wheels.**. 2. In the second kind of turns, you assign a positive value to one wheel and negative to the other. Let's say if I want to turn left, I will assign a positive value to the **right wheels** and negative to the **left wheels**. 3. In the third kind of turns, you assign a positive value to both the wheels but not the same value. If we want to turn left, we will assign a positive value for **right wheels** and for the **left wheels**, we will assign another positive value but lesser. Notice that in all three cases there is a difference between the speeds of the **left wheels** and the **right wheels.** Today we will use the second type of turn where we assign positive velicity to one side and negative velocity to the other.	
Teacher Stops Screen Share	
So now it's your turn. Please share your screen with me.	
Can you make the rover turn based on the key pressed? Let's try. I will guide you through it.	**ESR:** Yes!
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 • **Move the rover depending on the key pressed.**

Teacher Action	Student Action
Teacher helps the student to download the boilerplate. *Guide the student to create a controller. Student has to delete the contents of the newly created controller and paste the code given in the downloaded kb_controller.py file.*	*Student downloads **Student Activity 1** in **Webots***
We have already imported the keyboard and created an instance of the keyboard. The next step is to enable the keyboard input. This can be done by the **enable()** function. The parameter passed to the **enable()** function sets how frequently readings are updated. It is expressed in milliseconds. The Robot has a lot of sensors that can be enabled. To update the sensors in the controller, we have to call the function, **robot.step(timestep).** This function synchronizes the sensor and data between Webots and the controllers. The parameter passed is again the duration between two updates. We have to call the **robot.step(timestep)** continuously. When the simulation terminates, the Webots returns a value of -1. So we need the code block to run as long as	

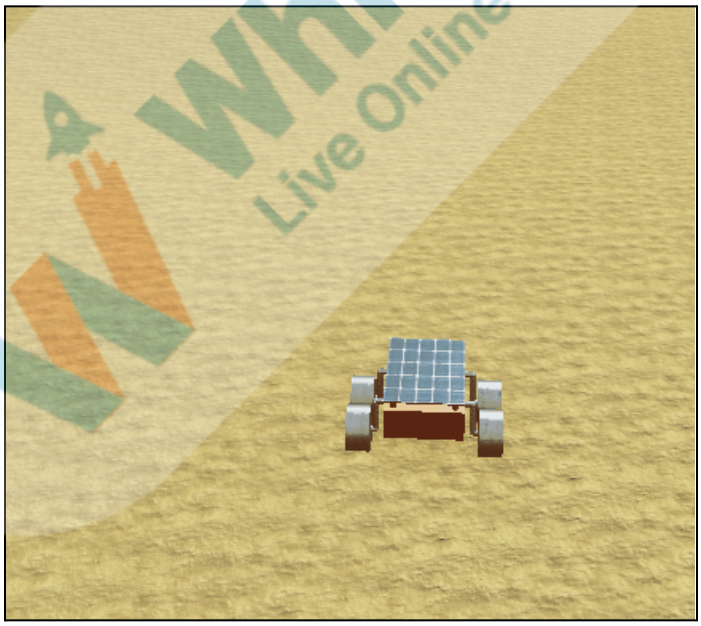
the simulation is running or in other words, so long as the return value is not -1. Which code block do we use so that it runs as long as a condition is met? That's correct, we will use a while loop to check which key is pressed and move the robot according to that. To understand which key is pressed, the function **getKey()** is used. The ASCII value of the key pressed is returned by the **getKey()** function. When no key is pressed, the value returned is -1. By printing the value of **keyPressed**, we can get the ASCII value of each key.	**ESR:** while block
	
Can you tell me the ASCII value of the key "w" by looking at the console window? What do we do when we press the **w** key? That is right. We will give a positive velocity to all four wheels to move the rover forward.	**ESR: 87** *Let the student explore ASCII values of other keys too.* **ESR:** We have to move the rover forward.

Let's declare a variable speed and assign it a value 4.	

CODE:

```
speed=4

keyboard.enable(timestep)

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

    key_pressed= keyboard.getKey()
    print(key_pressed)

    if(key_pressed== 83):
        wheel1_left.setVelocity(speed)
        wheel1_right.setVelocity(speed)
        wheel2_left.setVelocity(speed)
        wheel2_right.setVelocity(speed)
```

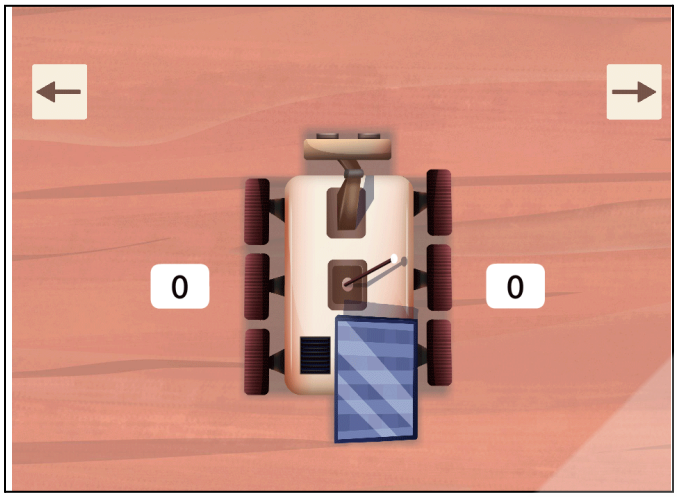
OUTPUT:



That's great. The rover is moving forward. The rover should move backwards when we press the "**s**" key. What velocity should we set for it to move backwards? Can you check the ASCII value of s and write code to move the rover backwards ? *Note: Guide the students and check for errors while the student is writing the code.*	**ESR:** Negative velocity **ESR:** Yes, I will try.


`while (robot.step(timestep) !=-1):`
 	
`key_pressed= keyboard.getKey()`
`print(key_pressed)`
 	
`if(key_pressed== 83):`
`wheel1_left.setVelocity(speed)`
`wheel1_right.setVelocity(speed)`
`wheel2_left.setVelocity(speed)`
`wheel2_right.setVelocity(speed)`
 	
`if(key_pressed== 87):`
`wheel1_left.setVelocity(-speed)`
`wheel1_right.setVelocity(-speed)`
`wheel2_left.setVelocity(-speed)`
`wheel2_right.setVelocity(-speed)`	

Now let us turn the rover left. Do you remember what velocities we have to set to turn it left?	**ESR:** Negative velocity for left wheel and positive velocity for the right wheel



That is correct. Similarly to make a right turn, we have to set negative velocity to the right wheels and positive velocity to the left wheels.

When no key is pressed, the rover should be stationary. To achieve this, we will write an if condition that checks if the keyPressed is -1. When -1 is returned it means that no key is pressed. In this case we will set the velocity to zero.

Do you think you can write the code on your own?

Encourage students to give different values to variables, timestep and speed and check the changes in the output.

ESR: Definitely.

CODE:

```
if(key_pressed== 87):
    wheel1_left.setVelocity(-speed)
    wheel1_right.setVelocity(-speed)
    wheel2_left.setVelocity(-speed)
    wheel2_right.setVelocity(-speed)

if(key_pressed== 65):
    wheel1_left.setVelocity(-speed)
    wheel1_right.setVelocity(speed)
    wheel2_left.setVelocity(-speed)
    wheel2_right.setVelocity(speed)

if(key_pressed== 68):
    wheel1_left.setVelocity(speed)
    wheel1_right.setVelocity(-speed)
    wheel2_left.setVelocity(speed)
    wheel2_right.setVelocity(-speed)

if(key_pressed== -1):
    wheel1_left.setVelocity(0)
    wheel1_right.setVelocity(0)
    wheel2_left.setVelocity(0)
    wheel2_right.setVelocity(0)
```

OUTPUT:

Click here to view the output video.

Wonderful, now the rover is moving in all four directions. You did a great job in class today. Well done!	

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 to move the rover automatically whenever it detects an obstacle.	*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-C284-Reference-Code
Teacher Activity 2	Reference Code	https://github.com/procodingclass/PRO-C285-Reference-Code
Teacher Reference 1	Project	https://s3-whjr-curriculum-uploads.whjr.online/d41259de-287a-47b5-a394-57594d46891c.pdf
Teacher Reference 2	Project Solution	https://github.com/procodingclass/PRO-C285-Project-Solution
Teacher Reference 4	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/b496a6d2-66d2-4c1f-b1e5-49ca18c2c58f.pdf
Student Activity 1	Boilerplate Code	https://github.com/procodingclass/PRO-C285-student-boilerplate