**Overview:**

Students will explore multiple representations of position, velocity and time to write code to navigate a set of challenges. Students will apply their knowledge of the relationships among position vs. time graphs, velocity vs. time graphs, and their verbal descriptions to write programs for Rover to carry out the motion described. This activity is appropriate for students familiar with modeling and solving relationships involving distance, rate, and time.

**Goals:**

Students will:

- Use their knowledge of position vs. time and velocity vs. time in the context of Rover motion challenge problems.
- Create and edit Python programs that include several commonly used Rover and calculator functions.

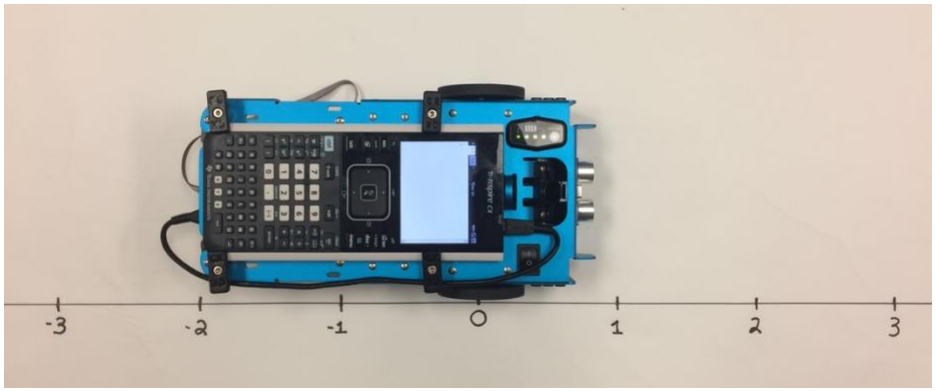
**Python Syntax Reference:**

| Statement | Example | Behavior |
|---|---|---|
| import module_name as name_space | import ti_rover as rv | Required for all TI Rover Python programs. Imports the ti_rover module into the Python program. The module provides the methods for controlling the Rover.<br>Sets the current position of the RV as the origin and the heading as 0 degrees measured from the x-axis. |
| # text comment | **# Press ctrl-R to run the program** | # at the beginning of a line denotes a comment. Comments are a “best practice” by programmers to annotate their code. Comment statements are ignored when the program is run. In the TI-Nspire CXII Python editor, [ctrl]+[T] toggles the statement of the current cursor location from a comment to a statement that will be run. |
| rv.forward_time(time,speed,"unit")<br>rv.backward_time(time,speed,"unit") | rv.forward_time(3,2.3,"units/s") | Rover drives forward at a rate of 2.3 grid units/sec for a time of 3 seconds. Note that the rate at which rover drives will vary for different surfaces which affects how far it travels. The SPEED specification accepts 1.4 to 2.3 units per second. The default speed is 2 units per second.<br>rv.forward_time() and rv.backward_time () are available from the Rover> Drive> Drive with Options menu. |
| rv.stay(time in seconds) | rv.stay(3) | Rover will stay at its current position 3 seconds. |
| rv.position(x,y) | rv.position(2,0) | Sets the current physical location of Rover to be x=2 and y=0 on the coordinate plane. A use of rv.position is to start Rover at a different position than the default of x=0 and y=0.<br>rv.position(x,y) is available from the Rover>Commands menu. |

For more on programming Rover with TI-Nspire CXII follow the links to the TI Rover Menu Map: TI-Nspire™ Python Programming > Python Menu Map > TI Rover Menu.

**Setup:**

Students may work in groups of two or three. An example number line layout is below. Use this for the position of Rover. Mark with 10 cm units to match with Rover units, from -10 units to +10 units.




**Materials::**

- TI-Nspire™ CXII graphing calculator with Drive_the_Line Student TI-Nspire CXII Python.tns file loaded
- Calculator unit-to-unit cable (USB mini-A to USB mini-B cable)
- TI-Innovator™ Hub
- TI-Innovator™ Rover with TI-Innovator™ Hub connected inside the Rover.
- Number line for position. Use a 2-meter course, from -10 units to +10 units. Rover uses a 10 cm unit.
- Student “Challenge” cards have been provided and can be printed/ cut into individual task cards, and distributed to students (optional)

**Student Activity:**

Sit in small groups with your calculator and supplies for this activity. Practice the guidance modeled by your teacher.

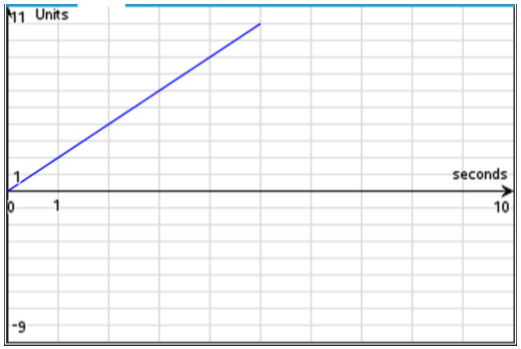
**Teacher Notes:**

Review and introduce the calculator, Hub, and Rover commands needed for this activity. In preparation for the coding on this activity, refer to Unit 1 Skill Builder 1 of the 10 minutes of code activities and Unit 4 Skill Builder 1 of the 10 minutes of code with TI-Innovator activities.

- Start a new program
- Attach Rover

**Challenge 1:**

Use rv.forward_time(time,speed,"unit") to have Rover drive the path described by the graph below and on page 1.4:



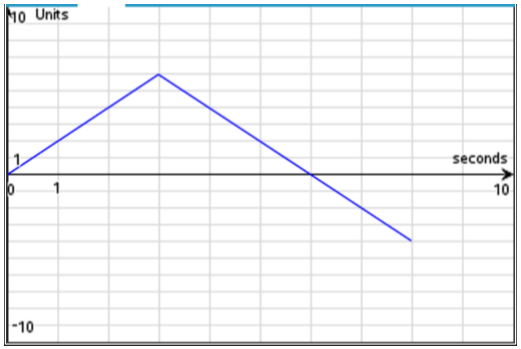

**Teacher Guidance during Challenge 1:**

- Students can have Rover drive the graph by determining the rate and time and then using the forward(distance,"unit",speed,"unit") function. See the Rover> Drive> Drive with Options menu.
- A Rover unit is 10 centimeters. Speed is in units per second. In each challenge of this activity, we will be using units per second. The Speed option takes arguments of 1.4 units per second to 2.3 units per second. Speed is affected by the surface. This may require some adjustments to the challenges based upon the surface. In most cases the rates may be slower than what is entered.
- After the students drive Challenge 1, have them go back to the graph on page 1.4 to compare their Rover's actual path in red to the original challenge graph in blue. There is code already in the .tns file at the bottom of the program to capture the time and position of Rover as it drives. The drive path data is stored to lists for plotting. NOTE: The red graph may not match exactly with the blue graph. Have students discuss possible sources for the variation.
- Example Program: **challenge1.py**

```
import ti_rover as rv
rv.forward_time(5,2,"units/s")
```

- *Discussion Starters*
  - The following are suggested discussion starters to engage your students with the mathematics inherent in Challenge 1:
    - How much time did Rover's trip take?
    - How far is Rover from zero at time t=5 seconds?
    - What is Rover's rate of change for this path?
    - Write a sentence describing Rover's path. Give the beginning position and the rate and time of the drive and draw Rover's trip on a number line and make a table of values for position and time for each second.

**Challenge 2:** Use rv.forward_time() and rv.backward_time() or rv.forward() and rv.backward() with distance and speed options to have Rover drive the path described by the graph below and on page 2.2:




**Teacher Guidance during Challenge 2:**

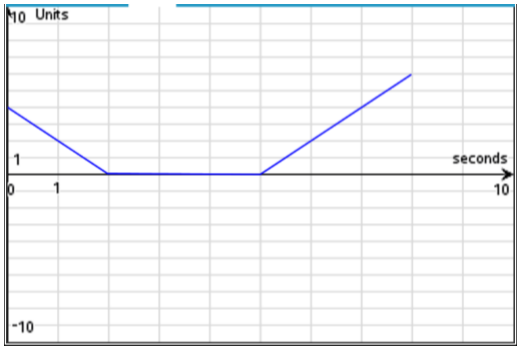
- There are several possible solutions to this challenge. Make sure students are making connections among the code, the path, and the description.
- It may be helpful to have students answer some of the discussion starters prior to writing the code.
- After students drive this challenge, have them compare the plot of position vs. time for Rover against the challenge graph on page 2.2.
- Example Program: **challenge2.py**

```
import ti_rover as rv
rv.forward_time(3,2,"units/s")
rv.backward_time(5,2,"units/s")
```

*Discussion Starters*

- The following are suggested discussion starters to engage your students with the mathematics inherent in Challenge 2:
  - What is the total distance traveled for Rover in this drive? How do you know?
  - Is there a time when Rover is two units from zero? Explain your answer.
  - What is Rover's maximum distance from zero in this drive? When does this occur? How do you know this is the maximum distance from zero?
  - At what times is Rover at position zero?
  - What is Rover's velocity from 0 seconds to 3 seconds?
  - What is Rover's velocity in units per second from 3 seconds to 6 seconds?
  - What is Rover's average velocity in units per second on the interval $0 \le t \le 6$ seconds? Explain why this makes sense.

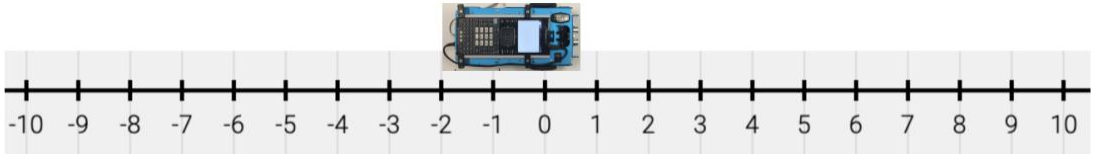
**Challenge 3:** Use the rv.forward_time() , rv.backward_time(), rv.stay(), and rv.position() functions to have Rover drive the path described by the graph below and on page 3.2:



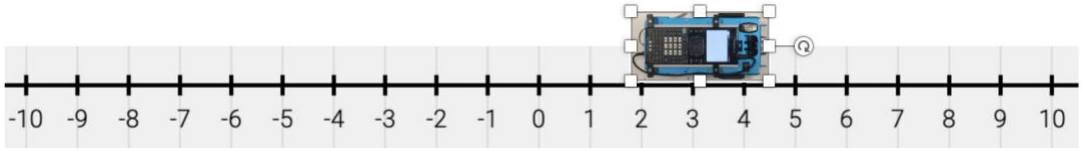

**Teacher Guidance during Challenge 3:**

- The rv.stay() function is found under the Rover Drive menu. It takes a single input of seconds. Rover will stay at its current position for *n* seconds and then begin the next drive function.
- It may be helpful to have students answer some of the discussion starters prior to writing the code.
- Use the rv.position() function to set Rover's initial position to x=4 on the number line. The rv.position() function takes inputs of x- and y-coordinates separated by a comma. rv.position() is found on the Rover. Commands menu.

**Physical Location of Rover at x=0**
**Default coordinate value of x=0 and y=0**




**Physical Location of Rover at x=0**
**rv.position(4,0) sets coordinate value to x=4 and y=0**




- **Note:** The default is to set Rover's location at the beginning of a program to coordinate values of x=0 and y=0. If you place your Rover on the number line that you created for this activity at x=0 then the physical movements of the Rover will match the coordinate values that Rover measures.

The rv.position() function can be used to set the coordinate value of Rover's location at the beginning of a program to something other than x=0 and y=0. For example, to set Rover's beginning location to x=4 use rv.position(4,0) and place your Rover at the appropriate location on the physical number line.

- After students drive this challenge, have them compare the plot of position vs. time for Rover against the challenge graph on page 3.2.

- Example Program: **challenge3.py**

```
import ti_rover as rv
rv.position(4,0)
rv.backward_time(2,2,"units/s")
rv.stay(3)
rv.forward_time(3,2,"units/s")
```

- *Discussion Starters*
  - The following are suggested discussion starters to engage your students with the mathematics inherent in Challenge 3:
    - What is the initial position of Rover?
    - Suppose someone says that Rover is standing still on an interval, do you agree or disagree? Explain why.
    - Does Rover travel farther between $t = 0$ and $t = 2$ or between $t = 2$ and $t = 5$ seconds. Explain your response.
    - Suppose that Rover continues moving after 8 seconds. Between 8 and 10 seconds, Rover's distance from zero decreases at a rate of 2 units per second, complete the graph. Explain how to determine Rover's final position.

**Challenge 4:** Use the rv.forward_time() , rv.backward_time(), rv.stay(), and rv.position() functions to have Rover drive the path described below.

Rover starts at 3 units to the right of zero. Rover drives backward at 2 units per second for 3 seconds. Rover stops and stays for 2 seconds, then drives forward at 2 units per second for 1.5 seconds.

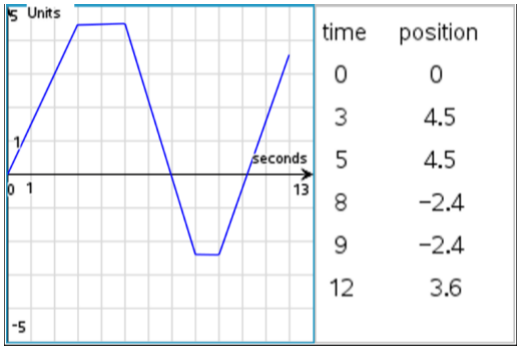
**Teacher Guidance during Challenge 4:**

- It may be helpful to have students draw the path of Rover prior to coding the challenge.
- It may also be helpful to have students answers some of the discussion starters prior to coding the challenge.
- Use rv.position() to set Rover's initial position to x=3.
- After students drive this challenge, have them compare the plot of position vs. time for Rover against the challenge graph on page 4.4.
- Example Program: **challenge4.py**

```
import ti_rover as rv
rv.position(3,0)
rv.backward_time(3,2,"units/s")
rv.stay(2)
rv.forward_time(1.5,2,"units/s")
```

- *Discussion Starters*
  - The following are suggested discussion starters to engage your students with the mathematics inherent in Challenge 4:
    - Is there a time when Rover is at position zero? If so, how do you know and at what time?
    - Suppose someone says Rover finishes its drive 3 units to the left of where it started, would you agree with them? Why or why not?
    - What is the time interval for when Rover is stopped?
    - Suppose from 6.5 seconds to 10 seconds Rover drives forward at 2 units per second, what is its final distance from zero?

**Challenge 5:** Have Rover drive the path described by the graph and table below and on page 5.2.
Use the rv.forward_time(), rv.backward_time() and rv.stay() functions.




| time | position |
|---|---|
| 0 | 0 |
| 3 | 4.5 |
| 5 | 4.5 |
| 8 | −2.4 |
| 9 | −2.4 |
| 12 | 3.6 |

**Teacher Guidance during Challenge 5:**

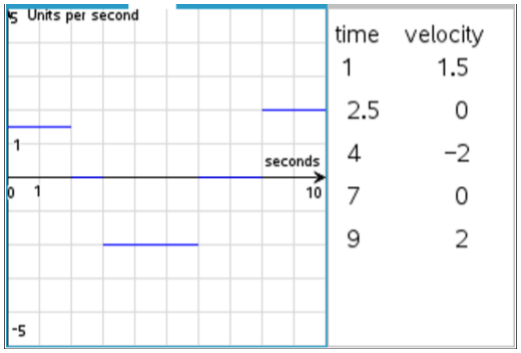
- Your students will need to change the values for speed in each segment.
- After students drive this challenge, have them compare the plot of position vs. time for Rover against the challenge graph on page 5.2.
- Example Program: **challenge5.py**

```
import ti_rover as rv
rv.forward_time(3,1.5,"units/s")
rv.stay(2)
rv.backward_time(3,2.3,"units/s")
rv.stay(1)
rv.forward_time(3,2,"units/s")
```

- *Discussion Starters*
  - The following are suggested discussion starters to engage your students with the mathematics inherent in Challenge 5:
    - Give an interval where Rover's velocity is the greatest. Explain
    - Suppose someone says that Rover is 1 unit to the right of zero only twice, how would you respond?
    - Explain how to determine how far Rover travels on the interval $5 \le x \le 8$, where x is the time in seconds since Rover started moving?
    - How fast is Rover moving at time x=4 seconds? Justify your answer
    - How fast is Rover moving at time x=7 seconds? Justify your answer
    - What is Rover's average rate of change of distance on the intervals: $0 \le x \le 3$, $3 \le x \le 5$, $5 \le x \le 8$, $8 \le x \le 9$, $9 \le x \le 12$, $0 \le x \le 12$?

**Challenge 6:** Have Rover drive the path described by the graph and table below and on page 6.2. Use the rv.forward_time() , rv.backward_time() and rv.stay() functions. Assume Rover starts at position zero.

Note: The graph is of velocity vs. time.




| time | velocity |
|---|---|
| 1 | 1.5 |
| 2.5 | 0 |
| 4 | −2 |
| 7 | 0 |
| 9 | 2 |

**Teacher Guidance during Challenge 6:**

- Make sure students are paying close attention to the axes. This is a graph of rate and time. Students may struggle interpreting this graph. It may be helpful to have students interpret the graph before writing the code. For example, have students describe how the car is moving on the interval 0 seconds to 2 seconds. Students should say “moving forward at 1.5 units per second”.
- Students may need help determining how far Rover moves in each segment and connecting area with distance traveled in a given time period
- After students drive this challenge, have them compare the plot of position vs. time for Rover against the challenge graph on page 6.2.
- Note: the challenge graph is a rate vs. time graph and so students will be able to connect the rate graph with the position graph from the Rover data. It will make a nice discussion about the meaning of the values on the rate graph and connections to the area between the rate graph and the x-axis (time axis) can be made.
- Example program: **challenge6.py**

```
import ti_rover as rv
rv.forward_time(2,1.5,"units/s")
rv.stay(1)
rv.backward_time(3,2,"units/s")
rv.stay(2)
rv.forward_time(2,2,"units/s")
```

- *Discussion Starters*
  - The following are suggested discussion starters to engage your students with the mathematics inherent in Challenge 6:
    - How is writing the program for this graph different than the others?
    - What does the value of the graph at time t=1 mean?
    - How far did Rover travel in the first segment? Explain how you calculated this.
    - What does the area between the curve and the x-axis represent?
    - Explain how to determine Rover’s position at any time ,t, using this graph.
    - Explain how to determine Rover’s final position from this graph.
    - What is Rover’s position at time 6 seconds? Justify your answer
    - Suppose someone says Rover is moving backward at time 4 seconds, do you agree or disagree? Explain

- Is there an interval where Rover isn't moving? If so, what is it, and explain how you know?
- Is Rover ever to the left of zero? Explain how you know.
- Give the time intervals where Rovers position is negative and Rover's velocity is positive.
- At what times is Rover at position zero?

- *Extension challenge:*
  - Given the Rover code below, have students draw the graph of Rover's position vs. time. Then have them write the code and see if the action matches what they drew.

```
import ti_rover as rv
rv.position(3,0)
rv.forward_time(3,2,"units/s")
rv.stay(3)
rv.backward_time(5,2.3,"units/s")
rv.stay(2)
rv.forward_time(5,1.5,"units/s")
```