Skill Builder 2: Turning

THE ON-RAMP TO ROBOTICS

UNIT 1: MOTION CONTROL
TI-84 PLUS CE PYTHON

Draft

Overview:

Students will write a program on their calculator to turn the Rover left and right at various angles. They are challenged to make their Rover slowly turn like the hands of clock and to report the time of day on their calculator.

Goals:

Students will:

1. write a TI Python program to turn Rover left or right at different angles.
2. incorporate the For loop control structure into a program.
3. incorporate the print() function into a program.
4. make a model of a clock using the Rover.

Background:

The Rover turns by rotating its wheels in opposite directions at the same speed at the same time. This type of turn is called a spin because it spins in a circle that has a center at the midpoint of the two wheels. This midpoint is also the location of the marker tip when a dry-erase marker is inserted into the pen holder. When the Rover performs a turn, the program needs to inform the motors the direction and size of the spin. The direction of the turn is determined by using the rv.left() and rv.right() functions from Rover Drive menu. The direction is from the view of as if Rover had a driver's seat. The size of the turn is determined by the angle, this is a value in degrees, radians, or gradians. A full spin is 360 degrees. This number is from the base 60 sexagecimal system used by the Sumerians in ancient Babylon. Similarly, a full spin is 2π radians. This number comes from the fact that the angular width of an arc of one radius in length along the circumference of any circle is defined as one radian. Also, a full spin is 400 gradians. The gradian is defined in the metric system as 1/100 of a circle quadrant. The default angle measurement unit is degrees. The Rover can accept all three units when using rv.left(angle,"unit") and rv.right(angle,"unit") from the Rover Drive menu.

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. Sets the current position of the RV as the origin and the heading as 0 degrees measured from the x-axis. The import ti_rover as rv statement is available from the Fns>Modul>ti_rover>Drive menu.
from module_name import *	`from ti_system import *`	Imports all the functions in the **ti_system** module for use in the program. It is necessary to import the ti_system module to use sleep( ). The from ti_system import * statement is available from the Fns>Modul>ti_rover>Commands menu.

rv.right (angle)
rv.left (angle)
	`rv.right()`
	`rv.right(45)`
		Rover spins to the right 90 degrees (If no value is entered for angle_degrees, rv.right() and rv.left() use the default value of 90), followed by a spin of another 45 degrees to the right.
		For angle units other than degrees, use rv.right(angle,"unit") toward the bottom of the Rover Drive menu.

©2022 Texas Instruments Incorporated

print(value or "text string")	`n=3`
	`print(n)`
	`print("number= ")`
	`print("number= ",n)`
		Prints value of variable n, which is 3; then on the next line prints the text string "number= "; finally prints "number= " followed by the value of variable n all on the next line.
		print() is available from the Fns>I/O menu.

sleep(seconds)	`sleep(1.5)`	The calculator will wait 1.5 seconds before moving to the next line in the program. sleep() is available from the Fns>Modul>ti_rover>Commands menu.

for i in range(size):
  block
	`for i in range(10):`
	`  print(i)`
		Repeats the statements in the block ten times, printing the value of the index variable, i, as 0,1,2,...9. The index variable, i, starts at 0 and increases by 1 with each loop. If i is less than the size value, 10, the loop continues to repeat. The block starts with a colon and includes the indented lines that follow.
		The for loop statements are found on the Fns>Ctl menu.

See the Rover module section beginning on page 26 of the Python Programming for the TI-84 Plus CE Python Graphing Calculator Guidebook for more programming information.

* The LEFT and RIGHT turns are made with a frame of reference from Rover's driver's seat.
** Radians is an angular unit of measure used in mathematics. There are 2π RADIANS in 360˚ DEGREES.
*** Gradians is an angular unit of measure also used in mathematics. There are 100 GRADIANS in a quarter circle; hence 400 grads in a full circle.

Setup:

Students may work in groups of two or three. Choose an area to work that has at least 2 meters of clear uniform floor space. Carpeted flooring is less desirable than tile. If needed, driving mats may be used as a driving surface.

Materials:

- Miniature traffic cones
- Masking tape
- Drive mats or hard, flat, clean surfaces
- TI-84 Plus CE graphing calculator with On-Ramp to Robotics Unit 1 Student TI-84 Plus CE Python program files loaded for use in the Python app.
- Calculator unit-to-unit cable (USB mini A to USB mini B cable)
- TI-Innovator™ Hub
- TI-Innovator™ Rover.
- Student "Challenge" cards have been provided and can be printed/ cut into individual task cards, and distributed to students (optional)

Draft

Student Activity:

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

Teacher Activity:

Review and introduce the calculator, Hub, and Rover commands needed for this activity. In preparation for the coding on this activity, refer to Meet the Rover with Geometry Challenges (download the TI-84 Plus CE Python PDF), Unit 1 Getting Started with Python Skill Builder 1 of the 10 minutes of code activities and Unit 4 Rover's Driving Features Skill Builder 1 of the 10 minutes of code with TI-Innovator activities.

- Download the student and teacher files from the project web page.
- Use CE Connect to download the student Python files to the handhelds. The student files include the necessary module import statements and some additional scaffolding of the activity.
- Open the student Python program file for each challenge in the Python Editor. Have the students enter the necessary statements to drive the challenge.
- Attach Rover.

Challenge 1: Write a program that uses right turns to spin Rover a total of 360 degrees.

Guidance during challenge 1:

- If necessary, review
  - the TI Python editor and how to run a program to students.
  - Basic navigation on the calculator.
  - Saving and opening files.
  - Editing new and existing programs.
  - Running programs.
  - Editing program features.
- Do the first challenge just after introducing the rv.right() and rv.left() functions. Do not yet, inform the students about the option to enter a value for the turn angle.
- As the students explore using the rv.right() and rv.left() functions without inputting an angle value, challenge them to determine what angle the Rover is turning. Next, ask how many angle turns are needed to spin a circle?
- Discuss and challenge students that are ahead how they could do the same program turning to the left.

Program: ORSB2C1T

```
import ti_rover as rv
rv.right()
rv.right()
rv.right()
rv.right()
```

Challenge 2: Write a program to turn Rover in a circle using the rv.right() or rv.left() functions. After each turn, display the total angle turned from the starting point on the calculator screen.

Guidance during challenge 2:

Review the usage of the print() function on the Built-ins I/O (Inputs/Outputs).

- print() is available from the Fns>I/O menu.
- Enter the display string in quotes “ “
- An example: print(”turn angle is 90”)

Review the usage of the sleep() function.

- sleep() is available from the Fns>Modul>ti_rover>Commands menu.
- Enter the time in seconds to wait.
- An example: sleep(2)
- The Rover and the calculator perform at different speeds. When a drive command is sent to the Rover, it may require several seconds for the Rover to drive that command. To keep in synch, the program must wait for the Rover to finish before executing another statement on the calculator. The sleep function is used to tell the calculator to wait. Estimate how long it takes Rover to turn 90 degrees. Use the time estimate as the time to sleep the program.
- Have the students explore the effect of using the sleep function in different places in their program and also with different times.
- Discuss and challenge students who are ahead, how they could do the same program in radians or gradians?

Draft

Program: ORSB2C2T

```
import ti_rover as rv
from ti_system import *
rv.right(90)
print(“turn to 90 degrees”)
sleep(3)
rv.right(90)
print(“turn to 180 degrees”)
sleep(3)
rv.right(90)
print(“turn to 270 degrees”)
sleep(3)
rv.right(90)
print(“turn to 360 degrees”)
sleep(3)
```

Challenge 3: Write a program using a For loop to turn three circles to the right and then three circles to the left in steps of 90 degrees.

Display the total of degrees turned by the Rover at each step on the calculator screen.

Guidance during challenge 3:

Review the usage of the for i in range() control function.

- The for loop statements are found on the Fns>Ctl menu.
- The for loop control structure enables the program a repeat a set of statements a fixed number of times.
- The for loop is defined by four values: an index variable to count the number of iterations, a start value for the index variable, an end value for the index variable and a step value for the index variable. The for i in range() statement assumes a start value of 0 and an index step value of 1. In Python, the statements to be repeated are defined by a beginning colon and statements that are indented from the row of the statement setting up the for control structure.

```
for i in range(5):
  set of statements to repeat (note: indented from the for loop statement)
statement that is not part of the loop (note: at same indent level as for loop statement)
```

- Help the students to understand the difference between using the previous program which explicitly calls out each statement and the use of the for loop which reuses one set of statements repeatedly.
- Also, inform students of the use of the index variable. This variable may be used to calculate the total angle turned by the Rover using the expression (i+1)*90, where i has a start value of 0 and then is incremented by 1 each time the for loop is iterated.

Review the usage of the print() function to combine text and calculation results with variables.

- print() is available from the Fns>I/O menu.
- Enter the display string in quotes “ “
- Include more items to print by adding a comma after and item.
- An example combining a text string and a calculation: print(”turn angle is ”,(i+1)*90)

Program: ORSB2C3T

```
import ti_rover as rv
from ti_system import *
for i in range(12)
  rv.right(90)
  print(“right turn angle= “,(i+1)*90)
  sleep(3)
for i in range(12)
  rv.left(90)
  print(“left turn angle= “,(i+1)*90)
  sleep(3)
```

Challenge 4: Write a program using Rover to model the hour hand on a clock. Turn the Rover to stop at each hour of the clock. Display the value of the current hour on the calculator display.

Guidance during challenge 4:

This challenge is the final challenge of this activity and requires that students incorporate all the skills learned so far. Encourage student to look back through the previous programs as references for creating this new program.

Students may be curious of the fact that there are 24 hours in a day. This number, like 360 degrees in a circle, comes from the Babylonian sexagecimal system since 360/15=24.

Program: ORSB2C4T

```
import ti_rover as rv
from ti_system import *
for i in range(12)
  rv.right(30)
  print(“right turn angle= “,i+1)
  sleep(2)
```