



# Mission 1: Welcome to Mars!

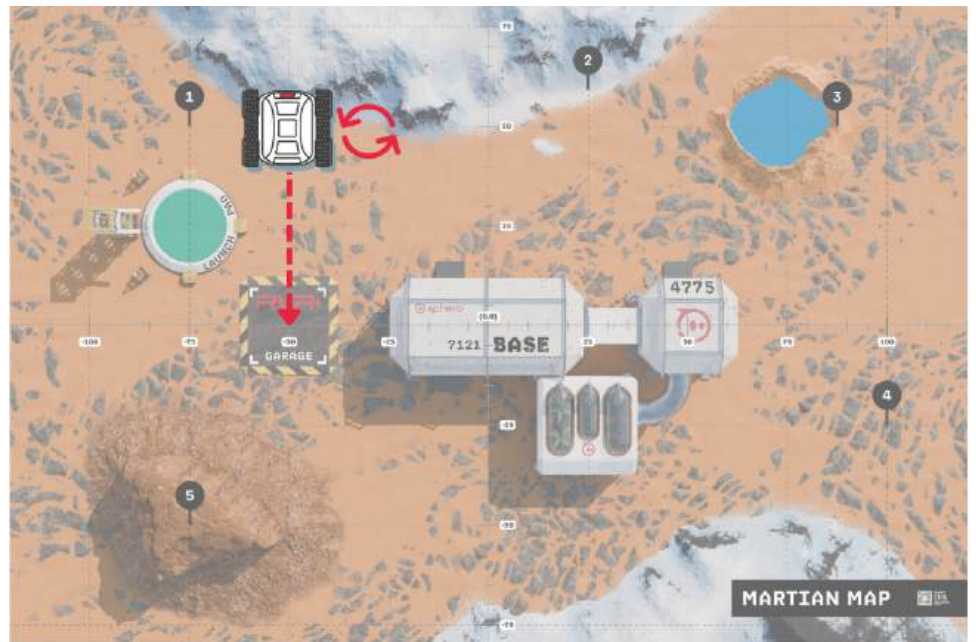
These Missions require the RVR+ Code Mat. Get it here: <https://sphero.cc/buy-rvr-mat>

<b>Program Type:</b>	<b>Concepts:</b>	<b>Duration:</b>
beginning block	heading, sound, roll to distance	10 minutes

## MISSION BRIEF

Welcome to Mars! Your team of Martian explorers and scientists has landed at the landing zone, coordinates (-50, 50). You're here to set up a sustainable base on Mars to further humanity's understanding of our neighboring planet.

Your first objective is to drive RVR+ to the garage for charging. You must change the inputs on the heading and roll to distance blocks to rotate RVR+ 180°, power up the drive system, and drive to the garage, coordinates (-50, 0).



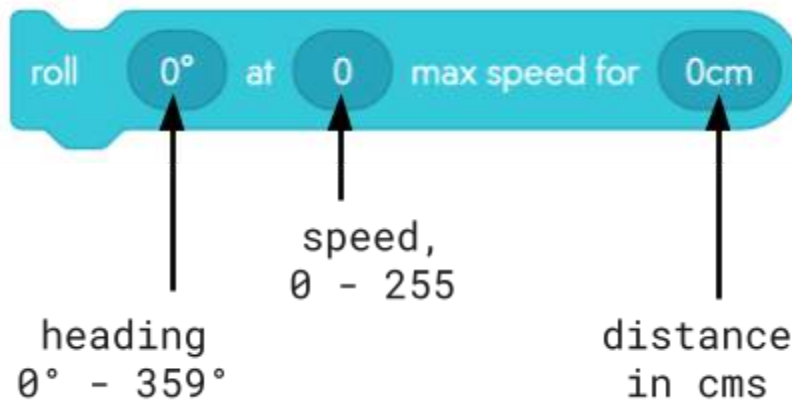
## MISSION LAUNCH PROGRAM

Start by building the program:



## MISSION CONTROL SUPPORT

- Go to [edu.sphero.com/d](https://edu.sphero.com/d) to install the Sphero Edu App or open the Sphero Edu Web App. Open the app and choose to create a new block program for RVR+.
- The **heading block** tells RVR+ which direction it is pointed between  $0^{\circ}$  and  $359^{\circ}$ .
- The power up sound is in the 8-bit sound category.
- The **roll to distance block** has three inputs: heading, speed, and distance.



## MISSION COMPLETION CHECKLIST

- ☐ RVR+ starts at the landing zone  $(-50, 50)$ .
- ☐ RVR+ parks in the garage  $(-50, 0)$ .
- ☐ Communicate your solution to your Mission Commander (teacher).
- ☐ Prepare for your next mission.



## Mission 2: Security Check

**Program Type:**  
beginning Python

**Concepts:**  
`roll_to_distance()`, `exit_program()`

**Duration:**  
10 minutes

### MISSION BRIEF

You've made it to base. It is time for your first trip around the perimeter to check the structural integrity of your new home. Start in the garage (-50, 0), travel all the way around the base modules, and return to the garage to finish your security check.

Your objective is to refine the Python program below so that RVR+ drives as closely as possible to the modules. The launch program only makes a rough rectangle.



### MISSION LAUNCH PROGRAM

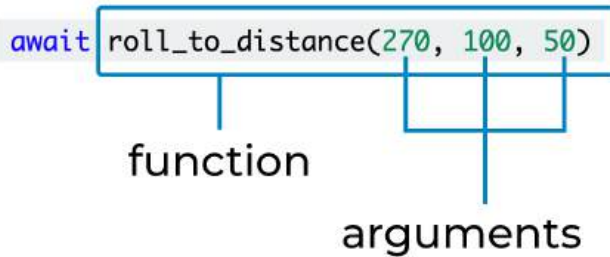
Start by building the program:

```
async def start_program():  
    await roll_to_distance(heading = 0, speed = 100, distance = 50)  
    await roll_to_distance(heading = 270, speed = 100, distance = 125)  
    await roll_to_distance(heading = 180, speed = 100, distance = 75)  
    await roll_to_distance(heading = 90, speed = 100, distance = 125)  
    await roll_to_distance(heading = 0, speed = 100, distance = 25)  
    await exit_program()
```



## MISSION CONTROL SUPPORT

- Each line of code in the program is a function. The **roll\_to\_distance()** function has three arguments: heading, speed, and distance.



- The functions in the starter program include keywords for heading, speed, and distance. These aren't necessary. For example, `await roll_to_distance(heading = 0, speed = 100, distance = 50)` is the same as `await roll_to_distance(0, 100, 50)`.
- The word `await` appears before many functions. This tells RVR+ to fully execute the function before moving on to the next function in the program.

## MISSION COMPLETION CHECKLIST

- ☐ RVR+ starts in the garage (-50, 0).
- ☐ RVR+ traveled around the perimeter of the base modules.
- ☐ RVR+ ends in the garage (-50, 0). The base is secure.
- ☐ Communicate your solution to your Mission Commander (teacher).
- ☐ Prepare for your next mission.



# Mission 3: Search for Life

**Program Type:**  
beginning block

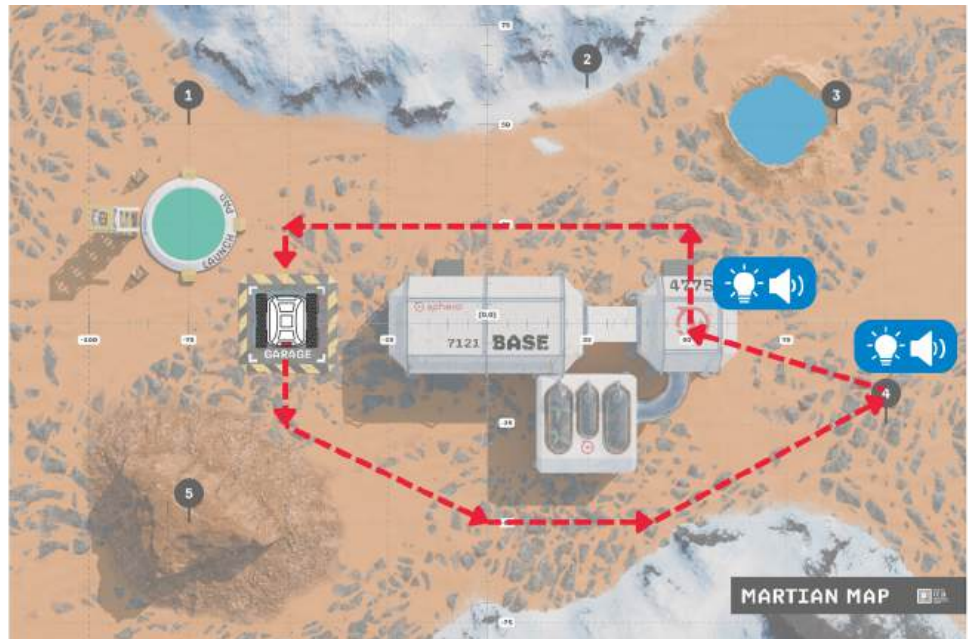
**Concepts:**  
lights, sounds

**Duration:**  
20 minutes

## MISSION BRIEF

Now that you're safely established in the base, it's time to start your scientific research. Humans have long wondered whether there is or was life on Mars. Start in the garage (-50, 0), collect a soil sample at position #4 (100, -25), deliver it to module 4775, and return to the garage. The Mission Launch Program provides the code for RVR+'s path.

Your objective is to add lights and sounds to represent collecting the soil sample and delivering the sample to the base.



## MISSION LAUNCH PROGRAM

Start by building the program. When you are working with a lot of the same block, right click or long press on the block to duplicate blocks to speed up the process.



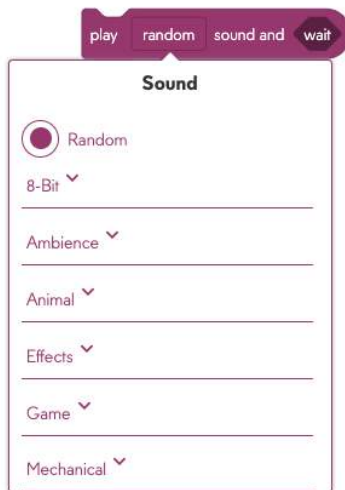


## MISSION CONTROL SUPPORT

- The Sphero Edu app has lots of different light blocks for RVR+. Explore some of the following to learn how they work.



- There are over a hundred sounds in the Sphero Edu app sound library. Try some of the sounds in the 8-bit and Science Fiction category to fit the Martian Map theme.



- When a sound block is set to "wait," the program will play the entire sound before moving on to the next block in the program. Executing blocks one at a time is called **synchronous programming**.
  - When a sound block is set to "continue," the program will start playing the sound and then start executing the next block in the program. Executing multiple blocks at the same time is called **asynchronous programming**.
- For this challenge, you can use synchronous or asynchronous programming.

## MISSION COMPLETION CHECKLIST

- ☐ RVR+ starts in the garage (-50, 0).
- ☐ RVR+ travels to position #4 (100, -25) to collect a soil sample. RVR+ uses lights and sounds to represent collecting a sample.
- ☐ RVR+ travels to Module 4775 to deliver the soil sample. RVR+ uses lights and sounds to represent delivering the sample.
- ☐ RVR+ returns to the garage (-50, 0).
- ☐ Communicate your solution to your Mission Commander (teacher).
- ☐ Prepare for your next mission.



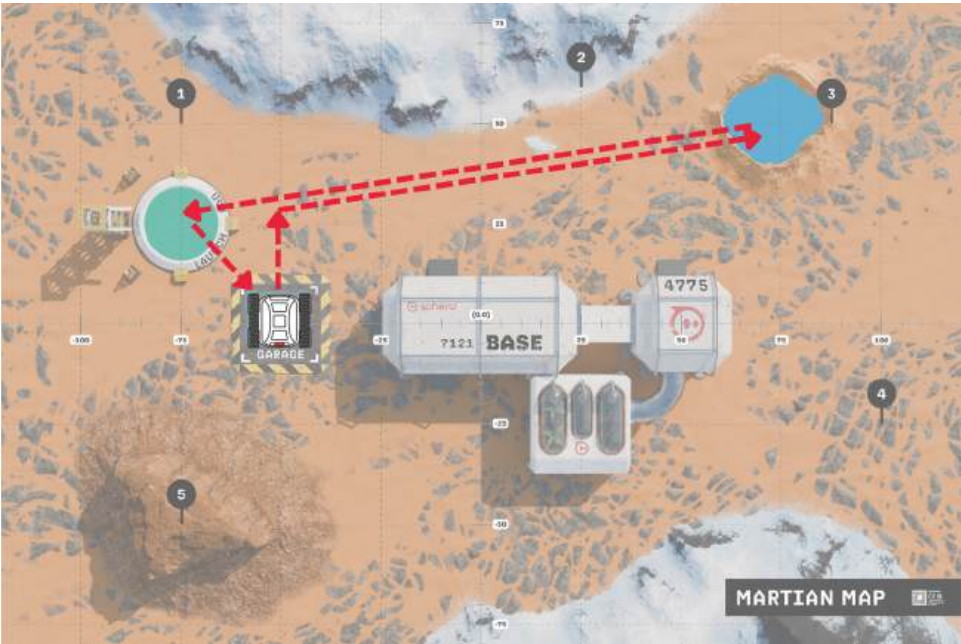
# Mission 4: The Hunt for Water

<b>Program Type:</b>	<b>Concepts:</b>	<b>Duration:</b>
intermediate block	on color, speed, stop	15 minutes

## MISSION BRIEF

For life to exist on Mars, scientists believe there must be a source of water. The most likely place is an underground lake in the crater of Olympus Mons, the solar system's biggest volcano. RVR+ needs to scan this area for water using its color sensor.

Your objective is to modify the program so that RVR+ navigates from the garage (-50, 0) to the crater (75, 50) and detects water using its color sensor. RVR+ should take a sample of the water, navigate to the launch pad (-75, 25), and launch the sample back to Earth for testing. Lastly, RVR+ should return to the garage (-50, 0) for charging.



## MISSION LAUNCH PROGRAM

Start with the program below. Then connect each group of blocks to the appropriate event block. Adjust the inputs of each block to complete the the mission objective.

on start program

stop

roll 0° at 30 max speed for 0cm

exit program

on color

roll 0° at 100 max speed for 25cm

heading 0°

speed 30

on color

stop

heading 0°

speed 30



## MISSION CONTROL SUPPORT

- The **on color blocks** are called events. They allow RVR+ to perform actions when its color sensor detects a specific color. To train RVR+ to look for a color:
  - Turn on RVR+ and connect it to your device.
  - In the app, select the color square of the **on color block**.
  - Set RVR+ on top of the color you want it to detect.
  - In the app, the color circle will change to indicate the color detected by the color sensor. Select Set Color to store this color for the **on color block**.
  - Click the checkmark in the top left corner to close the color sensor box.
- **Decomposition** is the process of breaking down a problem into smaller tasks or steps. This mission is decomposed into three steps: finding water, launching the sample back to Earth, and returning to the garage. Decomposing a problem into smaller tasks allows you to test and refine each section of code, one at a time.
- The **speed block** causes RVR+ to start driving at the designated speed (0 to 255). This speed can be either forward or backward.
- The **stop block** stops RVR+ from moving.
- Add blocks for lights and sounds to indicate RVR+ is collecting water, launching a water sample back to Earth, and returning to the garage to charge.



## MISSION COMPLETION CHECKLIST

- ☐ RVR+ starts in the garage (-50, 0).
- ☐ RVR+ uses its color sensor to detect water in the crater (75, 50).
- ☐ RVR+ uses lights and sounds to indicate that it has collected a water sample.
- ☐ RVR+ uses its color sensor to detect the launch pad (-75, 25).
- ☐ RVR+ uses lights and sounds to indicate that the water sample was launched back to Earth.
- ☐ RVR+ returns to the garage for charging (-50, 0).
- ☐ Communicate your solution to your Mission Commander (teacher).
- ☐ Prepare for your next mission.





# Mission 5: Ice Cap Expedition

**Program Type:**

intermediate block

**Concepts:**

drive, on button, on color, stop block

**Duration:**

15 minutes

## MISSION BRIEF

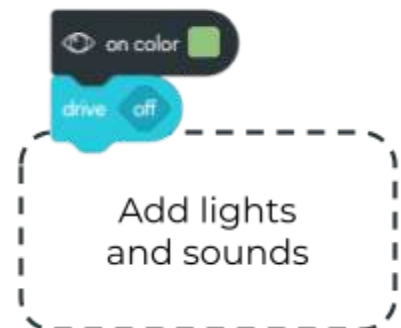
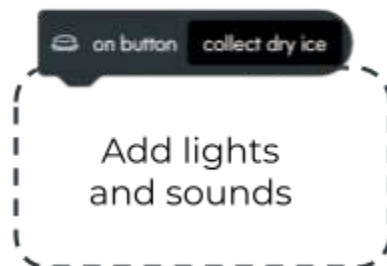
Exciting news!!! Looks like there may be signs of life in that soil sample collected at position 4. You need to package it in dry ice (frozen CO2) collected from the southern ice cap and then launch the sample back to Earth for further study.

Your objective is to drive from the garage (0, -50) to the southern ice cap (25, -75). Represent the collection of dry ice with lights and sounds before driving to the launch pad (-75, 25). At the launch pad, use more lights and sounds to represent launching the sample back to Earth.



## MISSION LAUNCH PROGRAM

Start by building the program. Set the **on color block** to the green color on the launch pad.



## MISSION CONTROL SUPPORT

- The **drive block** allows you to turn manual drive on and off. When turned on, you'll be able to use drive controls to control RVR+'s movement. When turned off, you'll need to control RVR+ movement with blocks in the movement category.

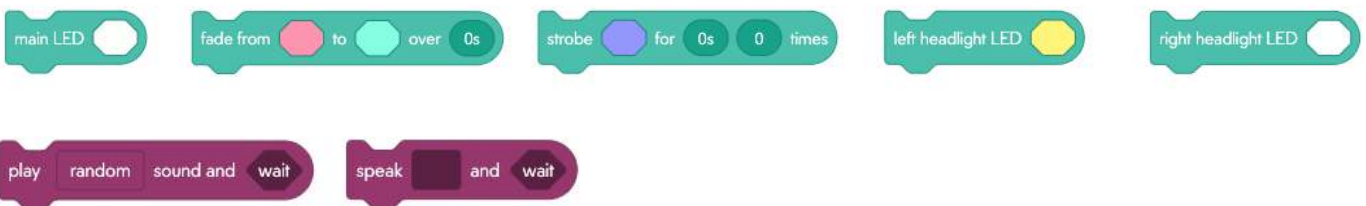


- The **on button event** allows you to trigger any blocks connected to the event to execute when a button in the app is pressed. The button will appear below the live sensor data. You can add up to three buttons to any RVR+ program.

### Buttons

1 COLLECT DRY ICE

- You can use your creativity when adding blocks to the **on button** and **on color** events. You may want to try some of the following.



## MISSION COMPLETION CHECKLIST

- ☐ RVR+ starts in the garage (-50, 0).
- ☐ RVR+ is driven to the southern ice cap (25, -75).
- ☐ RVR+ uses lights and sounds to indicate that it has collected dry ice.
- ☐ RVR+ is driven to the launch pad (-75, 25), and the color sensor is used to detect the green color.
- ☐ RVR+ uses lights and sounds to indicate that the sample was launched back to Earth.
- ☐ Communicate your solution to your Mission Commander (teacher).
- ☐ Prepare for your next mission.





# Mission 6: Resupply Delivery

**Program Type:**  
intermediate Python

**Concepts:**  
set\_main\_led(), for loop

**Duration:**  
20 minutes

## MISSION BRIEF

A rocket has landed at the launch pad to resupply the station. These supplies must be collected and brought inside before a dangerous dust storm hits.

Your objective is to program RVR+ to drive to the launch pad (-75, 25) to collect supplies and deliver the supplies to the station's living quarters (0, 0). There are too many supplies for RVR+ to carry in one trip. RVR+ will need to make four total trips. When RVR+ is loaded with supplies, its main LED should be red, and when it isn't loaded with supplies, its main LED should be green. After making all the deliveries, RVR+ needs to return to the garage.



## MISSION LAUNCH PROGRAM

Start by building the program:

```
async def start_program():
    await roll_to_distance(315, 200, 35)      # drive to launch pad
    set_main_led({'r': 0, 'g': 0, 'b': 0})    # set main LED to black
    await roll_to_distance(108, 50, 79)       # drive to living quarters
    set_main_led({'r': 255, 'g': 255, 'b': 255}) # set main LED to white
    await roll_to_distance(288, 200, 79)      # drive to launch pad
    await roll_to_distance(135, 200, 35)      # drive to garage
    await exit_program()
```





## MISSION CONTROL SUPPORT

- The provided program has RVR+ make one delivery from the launch pad to the living quarters, return to the launch pad, then return to the garage. However, RVR+ needs to make four delivery trips. One way to accomplish this would be to repeat the lines of code for the delivery trip three more times. However, this is not very efficient programming. Instead, you can use a **for loop**. A **for loop** repeats certain lines of code a designated number of times.

- In this example:

```
variable      # of loops
for trips in range(4):
    await roll_to_distance(108, 100, 79)
```

indent

- trips is the name of a variable that keeps track of the number of times through the loop. You can name this variable anything that isn't already a keyword in Python.
  - range(4) is the number of times the loop should repeat. RVR+ needs to make four trips.
  - All the commands that need to be a part of the loop should be indented.
- The function to change RVR+'s main LED is `set_main_led({'r': 255, 'g': 0, 'b': 0})`
  - One way colors can be represented in code is by their RGB values. RGB stands for red, green, and blue.
  - Each of these three main colors can have a value from 0 to 255.
  - Millions of colors can be created through different RGB combinations.

**Red**

```
set_main_led({'r': 255, 'g': 0, 'b': 0})
```

red = 255   green = 0   blue = 0

**Green**

```
set_main_led({'r': 0, 'g': 255, 'b': 0})
```

red = 0   green = 255   blue = 0

- You don't need to use the word `await` before the `set_main_led()` function.
- In Python, a `#` symbol indicates a code comment. Everything after the `#` symbol is ignored by the program. Code comments are used to help explain lines of code in normal language.

```
await roll_to_distance(315, 200, 35) # drive to launch pad ← -- code comment
```

## MISSION COMPLETION CHECKLIST

- ☐ RVR+ starts in the garage (-50, 0).
- ☐ RVR+ drives to the launch pad (-75, 25).
- ☐ RVR+ changes its lights to red to indicate it has a load of supplies.
- ☐ RVR+ drives to the living quarters (0, 0).
- ☐ RVR+ changes its lights to green to indicate it has delivered the supplies.
- ☐ RVR+ repeats the delivery trip three more times (four trips total).
- ☐ RVR+ returns to the garage(-50, 0) before the dust storm arrives.
- ☐ Communicate your solution to your Mission Commander (teacher).
- ☐ Prepare for your next mission.

# Mission 7: Decompression Alert

**Program Type:**  
intermediate block

**Concepts:**  
on time elapsed, spin

**Duration:**  
20 minutes

## MISSION BRIEF

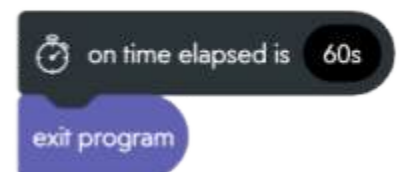
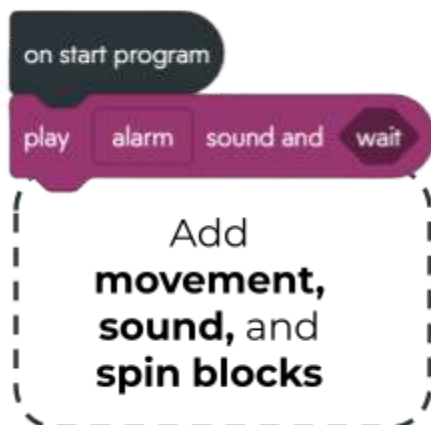
Alert, alert! The dust storm was stronger than expected and has damaged the node (tube) connecting the greenhouse to the base. The module is decompressing and venting oxygen. This will kill all the plants, the station's only renewable food supply.

Your objective is to travel from the garage (-50, 0) to the node (50, -25) and make repairs. RVR+ needs to spin in a circle six times and play the three adventure sounds. All repairs must be completed in 60 seconds. The program will automatically shut off after 60 seconds. Will you make it in time?



## MISSION LAUNCH PROGRAM

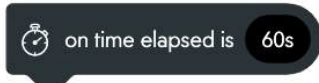
Start by building the program:



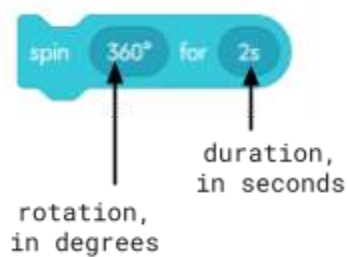


## MISSION CONTROL SUPPORT

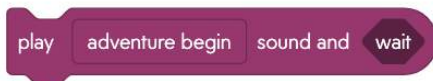
- The **on time elapsed event block** will trigger the blocks attached to execute when the number of seconds entered into the block have passed.



- The **spin block** will rotate RVR+ in place a specified number of degrees over a specified duration. You can use degrees greater than 360° to make RVR+ spin more than one rotation.



- The “adventure begin,” “adventure middle,” and “adventure end” sounds are located in the storytelling category.



By toggling wait to continue, you can use synchronous programming to program RVR+ to execute multiple blocks at the same time. However, be careful. If the program is currently playing a sound, it will skip subsequent sound blocks.

## MISSION COMPLETION CHECKLIST

- ☐ RVR+ starts in the garage (-50, 0).
- ☐ RVR+ drives to the node near the greenhouse (50, -25).
- ☐ RVR+ plays the “adventure begin”, “adventure middle”, and “adventure end” sound.
- ☐ RVR+ spins in place six times.
- ☐ RVR+ repairs the greenhouse before 60 seconds have elapsed.
- ☐ RVR+ returns to the garage (-50, 0).
- ☐ Communicate your solution to your Mission Commander (teacher).
- ☐ Prepare for your next mission.







# Mission 8: Mining for Minerals

**Program Type:**  
advanced block

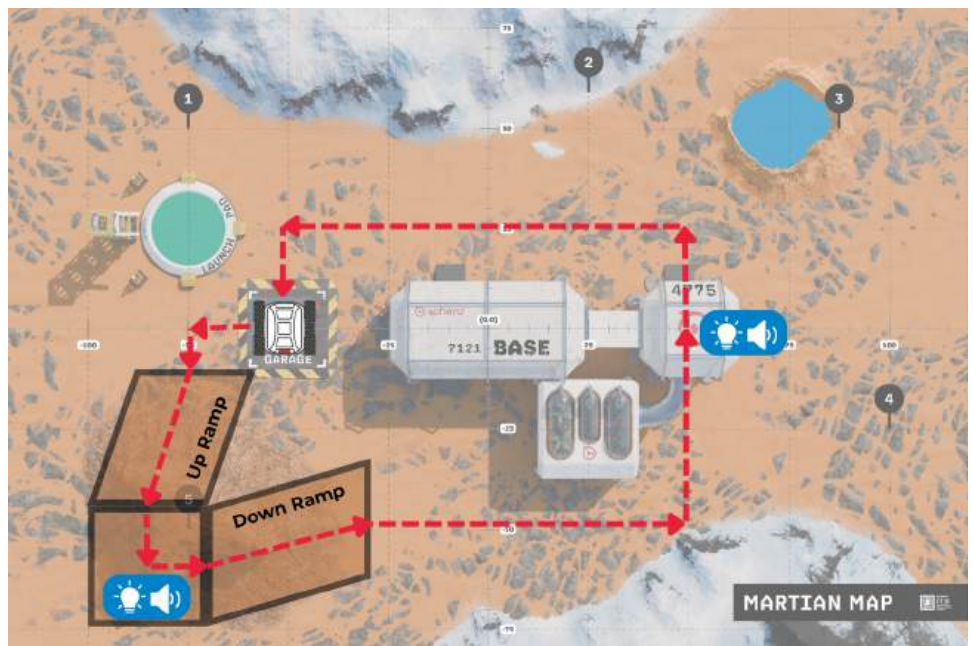
**Concepts:**  
if, comparator, and orientation blocks

**Duration:**  
30 minutes

## MISSION BRIEF

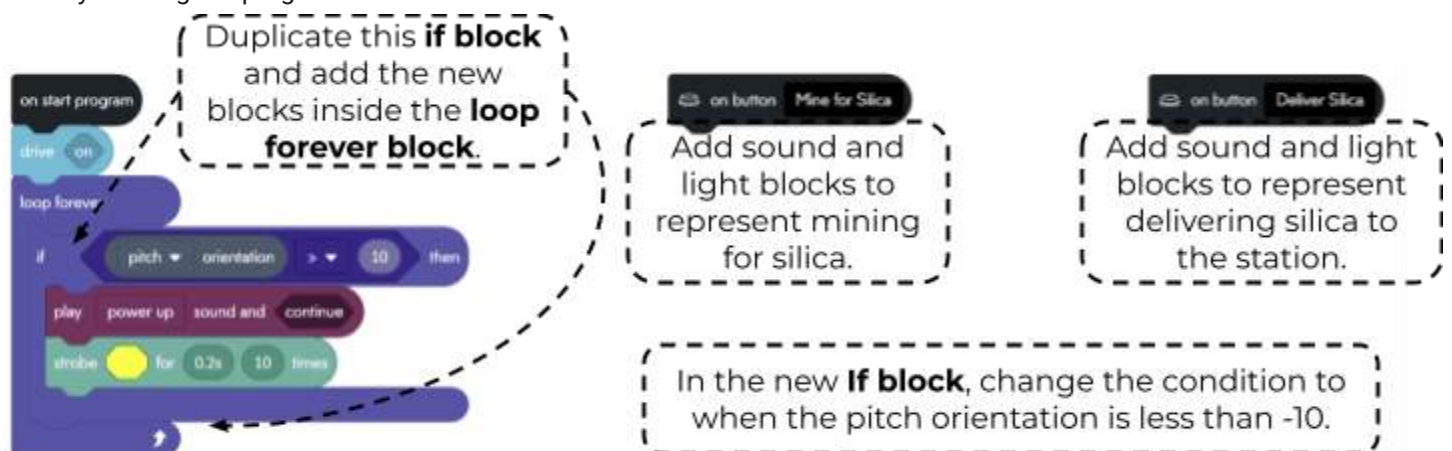
The dust storm is over, but several of the solar panels that power the station have been damaged. The station is on life support until more panels can be made. Solar panels are made from glass, and glass is made from silica. Thankfully, silica is abundant on Mars.

Your objective is to drive RVR+ to the mountain at position 5, (-75, -50), and mine for silica. As RVR+ drives up or down the mountain, a tip alarm with lights should indicate that RVR+ is on uneven ground and could tip over. After collecting a large supply of silica, you'll need to deliver the load to module 4775 (50, 0) and then return to the garage (-50, 0).



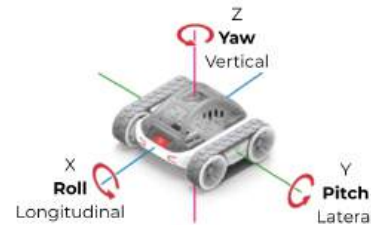
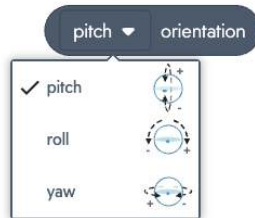
## MISSION LAUNCH PROGRAM

Start by building the program:



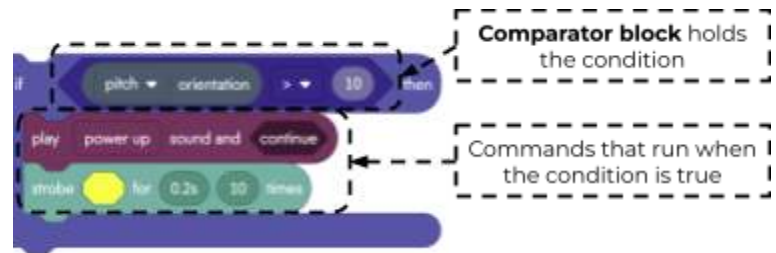
## MISSION CONTROL SUPPORT

- RVR+'s gyroscope sensor detects its orientation. Pitch is when RVR+ tilts forward or backward, roll is when RVR+ tilts side to side, and yaw is when RVR+ spins. The **orientation sensor block** in the sensors category reports the tilt in degrees ( $-180^{\circ}$  to  $180^{\circ}$ ) in the pitch, roll, or yaw direction.



- The **if block** allows RVR+ to do specific things when a condition is true. Connected to the **if block** is a **comparator block** that compares two values. If the condition is true, then the blocks inside the **if block** will run. If the condition is false, then the blocks inside the **if block** are skipped.

- In the provided program, the condition is "if the pitch orientation is greater than 10." In other words, if RVR+ tilts up more than 10 degrees.
- If RVR+ goes uphill and the gyroscope measures the pitch to be more than 10 degrees, then the "power up" sound will play and RVR+'s lights will strobe yellow.



- Duplicate the **if block** and place the new block inside the **forever loop block**. Change the condition to when the pitch orientation is less than  $-10^{\circ}$  (RVR+ is going downhill).
- Use the **on button** events to simulate mining and delivering silica with sound and light blocks.
- Use whatever sounds and colors you think best represent what RVR+ is supposed to do for each task.
- On the map, create a ramp that goes up to the mountain, an elevated platform where RVR can mine for silica, and a ramp coming down the mountain.

## MISSION COMPLETION CHECKLIST

- ☐ Create simple ramps and uneven terrain at the mountain for RVR+ to navigate. Ramps can be made from cardboard taped to a stack of books.
- ☐ RVR+ starts in the garage ( $-50, 0$ ).
- ☐ RVR+ drives up the ramp to the mountain at position 5 ( $-75, -50$ ) with pitch alarms going off.
- ☐ RVR+ uses lights and sounds at the mountain to indicate it is mining for silica.
- ☐ RVR+ drives down the ramp from the mountain with pitch alarms going off.
- ☐ RVR+ drives over uneven terrain with roll alarms going off to module 4775 ( $50, 0$ ).
- ☐ RVR+ uses lights and sounds at the station to indicate it is delivering the silica.
- ☐ RVR+ returns to the garage ( $-50, 0$ ).
- ☐ Communicate your solution to your Mission Commander (teacher).
- ☐ Prepare for your next mission.



# Mission 9: Terraforming Mars

**Program Type:**  
advanced block

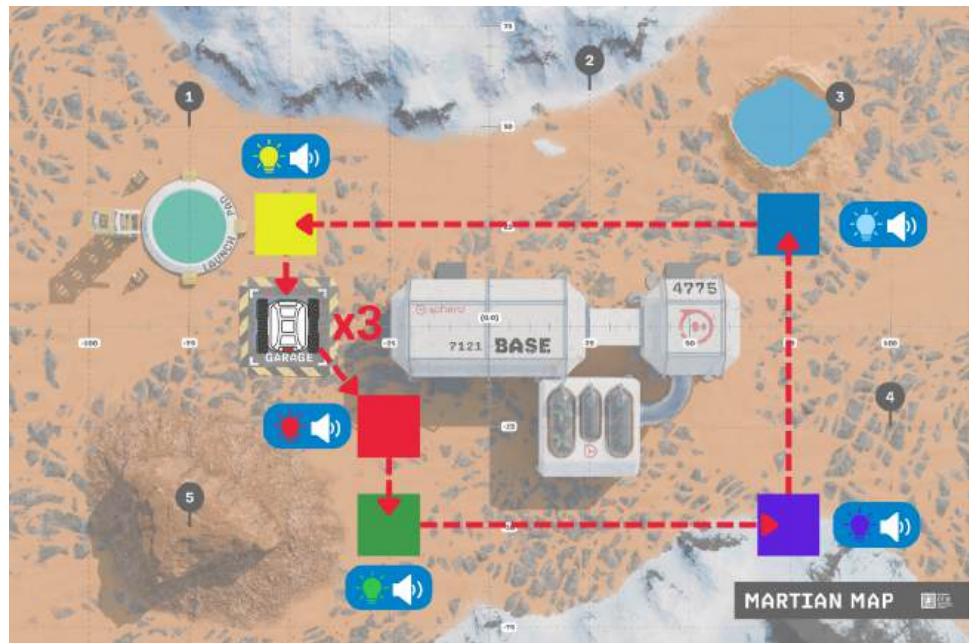
**Concepts:**  
number variables

**Duration:**  
30 minutes

## MISSION BRIEF

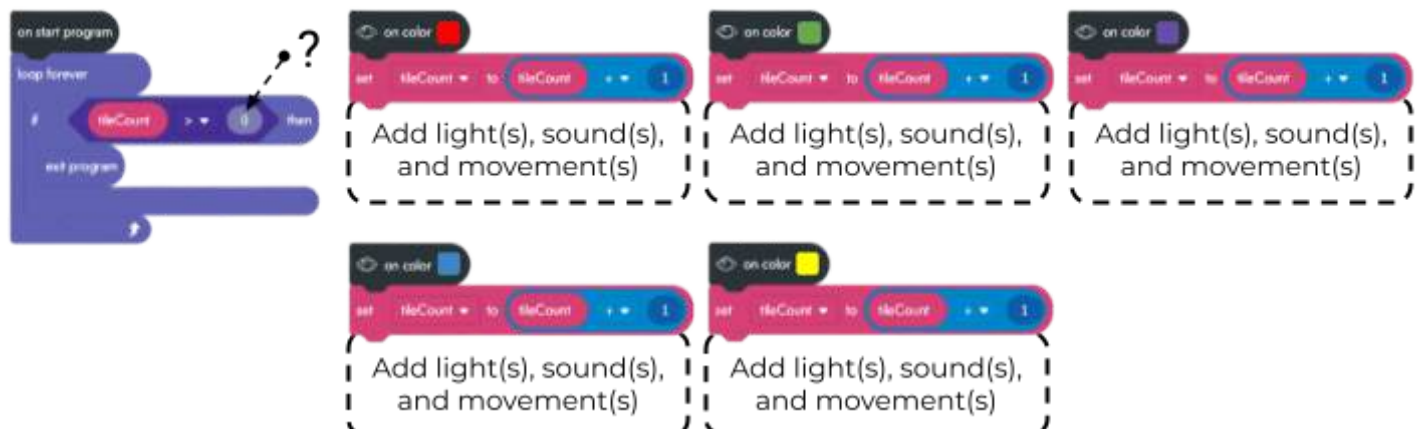
In order to make Mars habitable for plants and humans, the planet must be terraformed. This includes modifying Mars's atmosphere and geology. Each color tile represents an important modification zone: red = climate; green, yellow, purple = agriculture; and blue = water. You'll need to keep track of the zones to track your progress.

Your objective is to use RVR+'s color sensor to travel around the map three times. On each color, RVR+ should show the color and make a unique sound. You must use the **tileCount** variable to keep track of the zones.



## MISSION LAUNCH PROGRAM

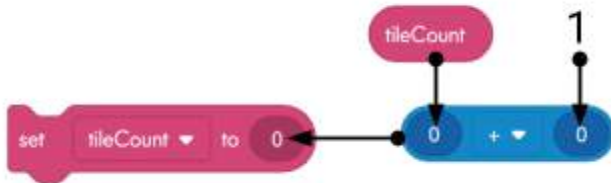
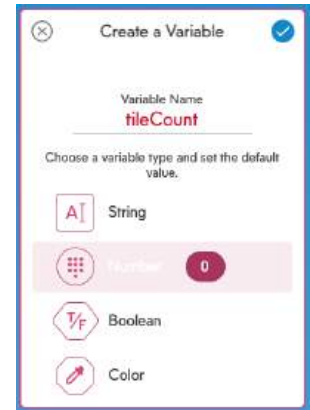
Start by building the program:





## MISSION CONTROL SUPPORT

- This challenge uses all five color tiles. Use the instructions in Challenge #4, to set the color in each **on color event** to match a tile. Additionally, each event must include movement blocks to get to the next tile.
- This program includes a number variable, **tileCount**. Follow these steps to create a number variable:
  - From the Variables category of the block library, choose "Create a Variable."
- Each time RVR+ recognizes a color tile, 1 is added to the **tileCount** variable. This is called incrementing the variable. Use an **addition operator** to construct the block underneath each **on color event**.



- Under the **on start program block**, use an **if then block** inside of a **loop forever block** to determine when to exit the program. Which number should be put in the **comparator** to stop the program after RVR+ has made three complete cycles around the map?

## MISSION COMPLETION CHECKLIST

- ☐ RVR+ starts in the garage (-50, 0).
- ☐ RVR+ moves to the red color tile (-25, -25), shows red LEDs, and plays a sound.
- ☐ RVR+ moves to the green color tile (-25, -50), shows green LEDs, and plays a sound.
- ☐ RVR+ moves to the purple color tile (75, -50), shows purple LEDs, and makes a sound.
- ☐ RVR+ moves to the blue color tile (75, 25), shows blue LEDs, and makes a sound.
- ☐ RVR+ moves to the yellow color tile (-50, 25), shows yellow LEDs, and makes a sound.
- ☐ RVR+ travels around the map to all five sites three times.
- ☐ RVR+ exits the program when it visits the yellow tile for the last time.
- ☐ Communicate your solution to your Mission Commander (teacher).
- ☐ Prepare for your next mission.



# Mission 10: Mars Tourism

**Program Type:**  
advanced block

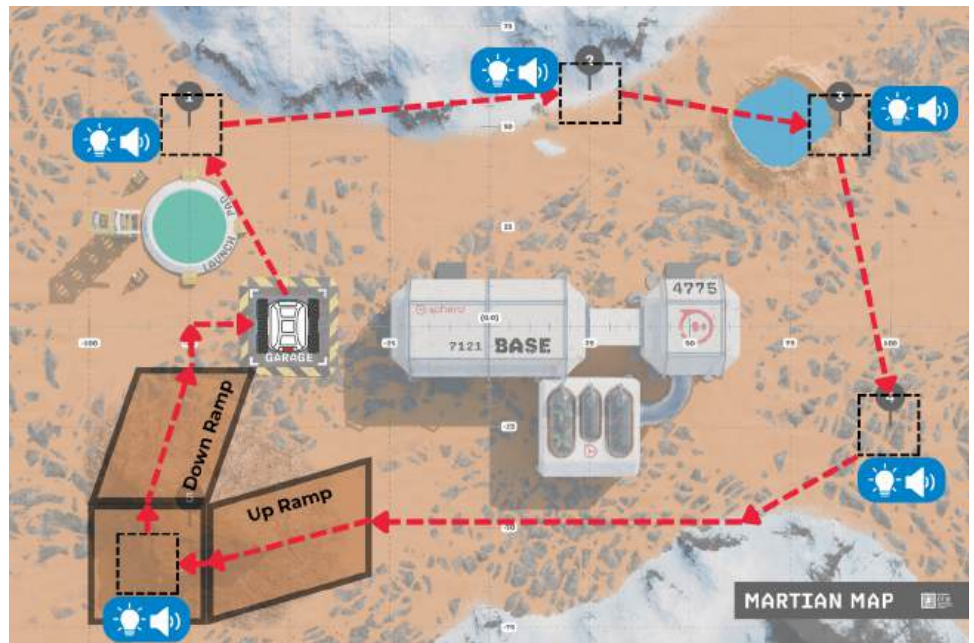
**Concepts:**  
on ambient light block, speak block

**Duration:**  
30 minutes

## MISSION BRIEF

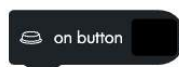
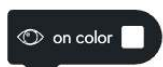
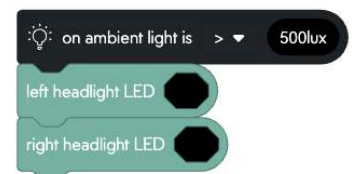
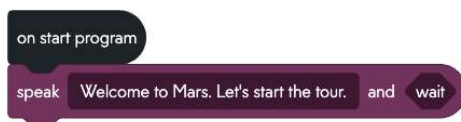
Terraforming Mars is going well and the colony is ready to expand. Tourists have begun to travel to the colony to see what life on Mars is like.

Your objective is to turn RVR+ into a tour guide visiting the five marked locations on the map, including up and down the mountain. At each site, RVR+ must state a Mars fact related to that location. The program needs at least one **on button event** and one **on color event**. RVR+ should use its ambient light sensor to activate headlights in the dark and deactivate them in light. Enhance the tour using other blocks and features from previous challenges.



## MISSION LAUNCH PROGRAM

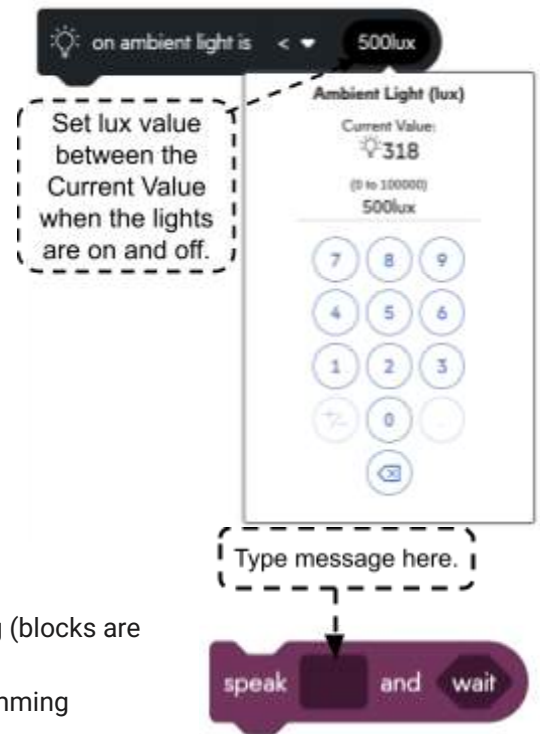
Start by building the program:





## MISSION CONTROL SUPPORT

- There are two options for your tour:
  - **Autonomous:** RVR+ can be programmed to navigate to each location on the tour.
  - **Guided:** RVR+ can be driven to each location using the **drive on block**.
- The **on ambient light event block** allows RVR+ to react to light changes.
  - Light intensity, or brightness, is measured in lux. Most classrooms range between 300-500lux.
  - To set the lux value:
    - Connect RVR+.
    - Click the value in the **on ambient light event block**. Note the Current Value in the dialog box.
    - Turn off the lights and note the new value.
    - Set the lux value between these two readings.
- The **speak block** uses a text-to-speech engine to speak a message from your programming device.
  - Type a message in the box.
  - Set the **speak block** to “wait” for synchronous programming (blocks are executed one at a time).
  - Set the **speak block** to “continue” for asynchronous programming (multiple blocks are executed at the same time).
  - Use a **speak block** to say a fact about each tour location.
- Use any combination of **on color** and **on button event blocks** to program each stop on the tour. You must have at least one of each in your program.
  - Use the color tiles with the **on color event blocks**.
- Build ramps for the mountain as you did in Mission 8.



## MISSION COMPLETION CHECKLIST

- ☐ Create ramps and uneven terrain at the mountain for RVR+ to navigate.
- ☐ RVR+ automatically turns on its headlights at any point on the tour if it gets dark.
- ☐ RVR+ starts in the garage (-50, 0).
- ☐ RVR+ drives to Position 1 (-75, 50) to describe the importance of the launch pad.
- ☐ RVR+ drives to Position 2 (25, 60) to explain Mars's polar ice caps.
- ☐ RVR+ drives to Position 3 (82, 50) and talks about the search for water on Mars.
- ☐ RVR+ drives to Position 4 (100, -25) to describe the search for life on Mars.
- ☐ RVR+ drives up the mountain to Position 5 (-75, -50) to talk about mining for minerals on Mars.
- ☐ RVR+ returns to the garage (-50, 0) to pick up a second tour group.
- ☐ Enhance RVR+'s tour guide capabilities by adding features from other challenges.
- ☐ Communicate your solution to your Mission Commander (teacher).
- ☐ Well done! RVR+ has completed its Mars mission. What new mission ideas do you have?