Level 3

# High Level Player Experience

Students learn to program the Rover to explore the Martian landscape on its own, while managing a limited power supply.

## Key Features:

* Building upon what they learned in Level 2 students program the Rover for its many abilities, including movement.
* Players must explore Mars to its fullest while under the constraints of limited power
* Programming Challenges:
  + Provide many different ways to explore Mars.
  + The Student must consider Rock types, terrain, and scanning strategies to try to maximize their SV points and conserve power
  + Students must ensure that the Rover takes a photo at the end of its run, using the last of its power.

# Design Details

**Details**

**Audience:** Grades 5-8

**Time Length:** 2-4 hours; teach research, design (of terrain), features, events, and presentation skills

**Level Type:** Top-down, self-exploratory game. Kids must navigate programmatically.

**Layout:** Various

**Button layout:** Incorporate use of “inspect”, “beam”, and “scan” to recreate Rover’s experience

**Goal:** Goal is to build an experience on Mars, highlighting the following key elements:

* **Terrain types**: Different areas of terrain have different composition, which can be explained as Rovers visit this rock types. Terrain affects movement speed.
* **Formations**: Certain formations of terrain provide higher chance of sedimentary rocks.
* **Power and Paths**: What would be the optimal use of power?
* **Search Strategies**: What is the optimal way to plot a path and also program the Rover?

**Obstacles:** There are a few obstacles:

* Different terrain types cause rover to speed up/slow down which affects power consumption
* You can only inspect “10” times (to represent the number of drills on the rover). Do you wait to drill?
* There is a RED counter detailing power remaining within which to collect as many points as possible. In order to be optimal, students will need to plot a path and take into account:
  + Where the Rover needs to look to find the highest scoring rocks
  + Terrain into consideration for speed
  + Terrain features and when the Rover needs to Scan
  + Programming the Rover when to use inspect and when to use beam
* Everything the rover does costs a certain amount of power. To represent the Rover is active even when not moving, scanning, inspecting or beaming there will be a slow power drain.

**Environment:**

* Students then would open a world that has geometry that facilitates multiple routes and decision making.

**Motivation:**

The student needs to turn learning about Mars into how they choose their actions with the Rover and subsequently how to program the Rover to meet this criteria.

**Curriculum:**

* Programming the Rover to search on its own.
* Review and understanding of an existing landing / mission.
* Research into different terrain types, and impact on Rover.
* Applying principles learned in Level 2.

## Rocks

A choice of rocks is provided to give students gameplay and programming challenges as well as to teach them about Mars and the Rover mission.

All rocks begin as Unknown type. By using scan is the type of rock revealed. The type of rock is weighted toward the most common low value Lava rocks. Higher value sedimentary rocks have a greater chance of being found in the correct terrain type where those rocks are typically located on Mars.

The Rover needs to navigate a sea of unknown rocks, determine their type via a scan and then decide whether to beam them with the laser or inspect them with the drill.

## Rock types

* Sedimentary - have the highest probability for finding organics and score the most points
* Lava Rock – low chance of finding organics and score the least amount of points
* Unknown – This is a generic rock type. Unknown rocks could be Sedimentary, Hematite or Lava. Until the rover investigates the rock, its type remains unknown.

Players can reference Level 2 for point values (see SGI - KODU\_MARS\_LEVEL 2.docx).

## Revealing Rock Type

All rocks are Unknown until the Rover uses a Scan.

Once revealed, rocks are identified by color and shape. Each one scores points upon being beamed by the laser. Inspecting the rock scores double points, encouraging the player to use their limited number of inspects wisely. Players will always get at least one point from any rock.

We may add a “Discovered Evidence of Life!” rock to represent complete success of the Mars mission.  Such a rock would be very rare and worth hundreds of points.

The goal of the student will be to inspect rocks with the highest likelihood of having organics and thus the highest chance of scoring more SV points – but balancing this against the power cost of using scans, inspections, laser and getting to the rock.

## In Kodu

Revealing the rocks and having them show type will be done with bespoke tile.

## Laser

The Rover can activate the laser to beam a rock. It can be activated an unlimited number of times. Players score more points by inspecting higher value rocks (see above).

Students will need to consider whether they want the Rover to zap as many rocks as they can with the speedier laser, or prioritize high scoring rocks with the Inspect action.

There is a power cost associated with using the laser.

## In Kodu

We will need to create a Laser tile for the students to be able to implement it. The tile will need to activate the effects and give students programming hooks to create the desired results.

## Inspect

The player can activate the Drill to inspect a rock. It can be activated 5 times. Inspecting doubles the point value of the rock it is used on. Players score more points by inspecting higher value rocks.

There is a power cost associated with using the laser.

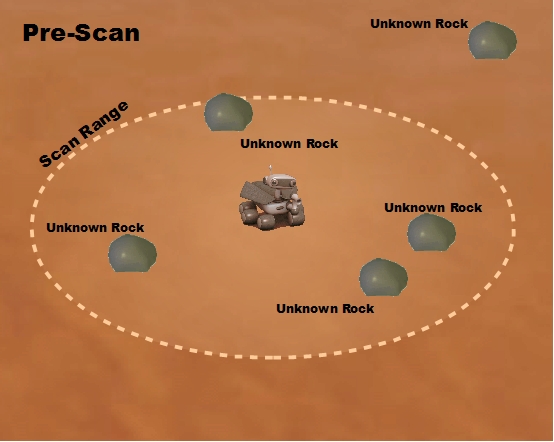
## In Kodu

We will need to create an Inspect tile for the students to be able to implement it. The tile will need to activate the effects and give students programming hooks to create the desired results.

## Scan

The Rover can use the Scan function to reveal the type of rocks that are within a specific range from the rover. The player scores points simply by Scanning, but the main advantage is revealing a large number of rock types quickly to aid in figuring out where to go to beam or inspect rocks.

Taking a Scan takes up time and power, so Scanning too often may result in a lower score.



## 

## In Kodu

Scan functionality can be achieved using Kodu’s sound and listen features (they allow us to check for things in a radius), but we would likely want to create a scan tile as using the sound Kode is not intuitive for the students.

## Picture

Because the rover has the capability of taking pictures, we will give students the ability to program the rover to take pictures, capturing what the rover can see.

This is to represent the real life actions of NASA when they begin each day by looking at what the Rover can see, and thus the photo will be take at the end of the ‘day’.

* Rover stops at a position.
* Code is executed (WHEN next rock DO picture <direction>).
* View changes briefly, and a small flash occurs, indicating picture was taken.
* Player can view pictures located on hard drive (stored on local profile space).

## In Kodu

We would need to modify Kodu’s current screen capture ability. Kodu can take screen shots in game based on the position of the camera when player hits “prt/scrn” on the keyboard. This functionality would be modified so students could access it as a tile allowing them to program the Rover to take pictures and to Kode the results of picture taking.

## Timer

The player will be constrained by the Rover’s power supply, not by a time limit.

## Layout

The terrain will be designed to give clues to the player where to look for high value rocks.

* Sedimentary rock – ridges and cliff-like formations on the terrain have a higher chance of having Phyllosilicates.
* Flat – mostly contains Lava rocks.

## In Kodu

Terrain types can be made by using Kodu’s materials. Unknown rocks in high value areas will be more likely to be high value rocks once investigated into.

# Game Flow

## Programming the Rover

Students will need to program the following:

* Rover movement along Student created path
* When Rover should perform Scan
* When Rover should activate Inspect
* When Rover should activate Laser
* When Rover should activate photo

Once playing the game, the Rover follows its program and drives across the terrain to find high value rocks to increase the student’s score.

The following actions occur:

* Scanning reveals rock types within range
* Inspecting a rock scores double points (higher value rocks score more points)
* Beaming a rock with the Laser scores points (higher value rocks score more points)
* When the Rover power supply runs out the experience ends
  + We may add “Winner” conditions for certain score thresholds