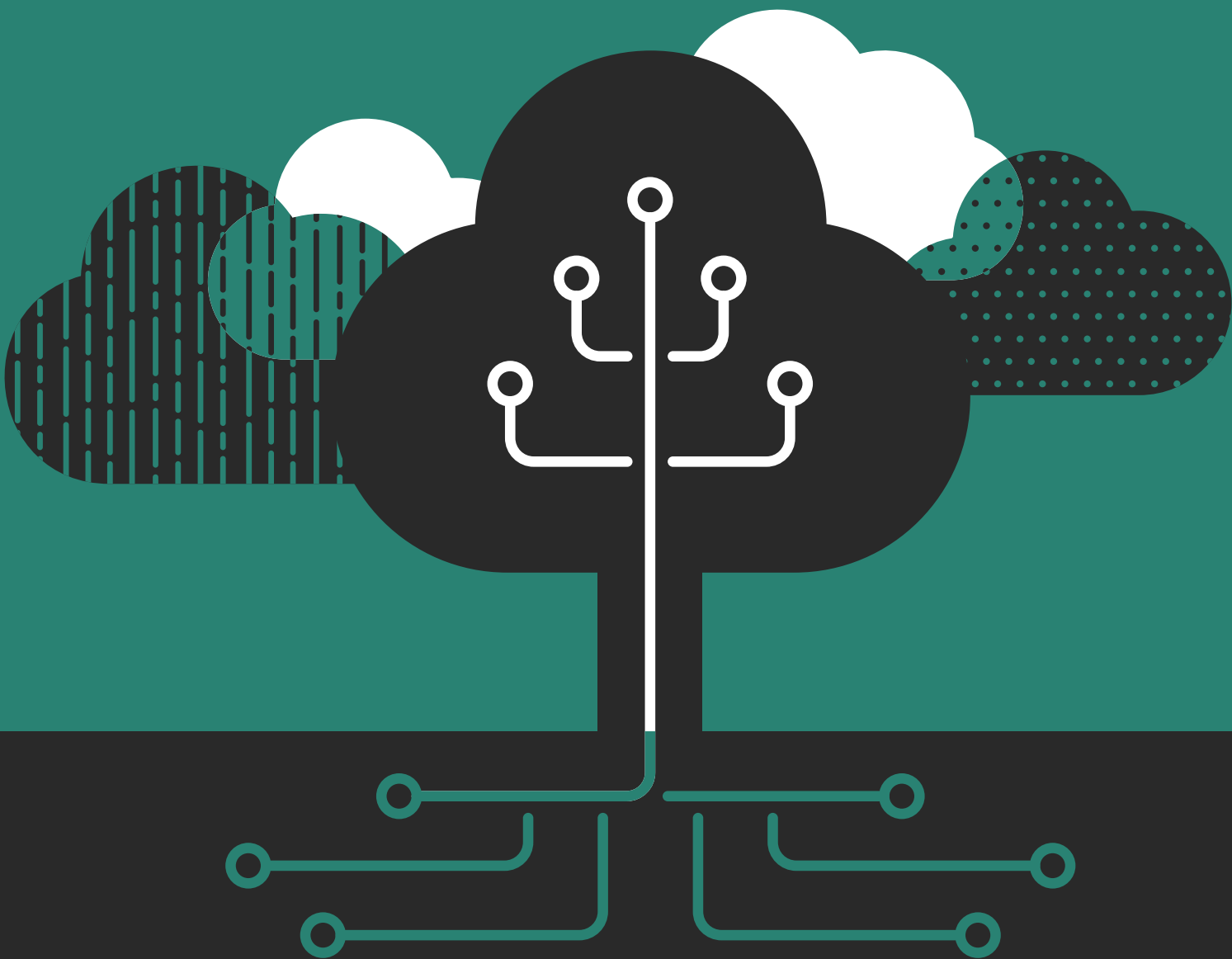


# Solutions



# MakeCode Arcade activity

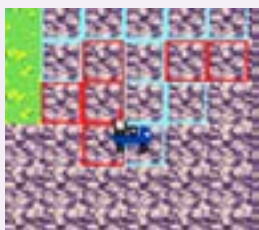
## Solutions

The following solutions are suggestions, you may have come up with a different working solution.

## Section B - Play the Game

### Solution: Task 1 - Learn how the full solution game currently works

Our game has a LiDAR van (sprite) which we can drive around the screen with



Driving the van to the volcano clouds on the right will collect the data (information). We can see where we have collected data as the boxes will have a border – red for dangerous concentrations of ash and blue for not dangerous.



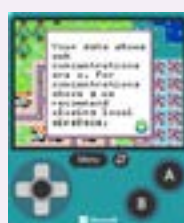
There is a score in the top right, this tells us how many more **data points** we have left to collect. As we drive over the grid boxes that contain ash cloud (data points) and take an observation, the score decreases. We must drive over all the data points for this score to go down to zero. Only then are we able to send our data back to the Met Office.



We submit data by driving to the data receiver:



If our counter is not at zero we will see a “We need more data to make a decision” message. We must take our LiDAR van back out to drive over more ash cloud.



If we are at zero and drive to the data receiver, it will make a recommendation and we must decide whether to close the airspace or not.

We can now press `A` and decide whether to close the airspace, based on the information given to us in the message.



## Section D – Bug Hunting



The ash levels are not reporting correctly at the end of the game and the message needs fixing.



## Section D – Bug Hunting

You may have spotted extra things that you would like to fix, we will be focussing on fixing these two missing features of our code.

### Solution: Task 2 – What's broken? - Deeper Discovery

The data points are not being collected properly. The tiles do not change colour as the LiDAR van drives over them. The counter (top right) is not counting down and the high/low concentration ash levels are not being counted correctly, resulting in the wrong values being displayed when we deliver the data.



## Section E – Get Coding

### Solution: Task 4 - Counting the ash level measurements - First Exploration and Deeper Discovery

After completing the task, your code should look like this:

We have adapted our if...else block.

If we go over a red/high ash level tile, then we need to add one to the number\_high\_ash\_levels **variable**. And the same for the low ash level tiles.



### Solution: Task 5 - Reporting the result of the game - First Exploration and Deeper Discovery

We have adapted this `nested if`. A `nested if` is an `if` within an `if`!

Our messages are now in the correct place and our game is now complete. Well done for all your efforts so far, you are super coders!



## Section F - Stretch and Extension Task (Suitable for all games)

### Solution: Task 6 – How to add new code



When our game starts, we need to set a countdown of 40 seconds. You can search for `start countdown` in the search box and drag it into the `On Start` code.

We now need an event – what happens when our timer runs out?

In the search box, type `on countdown end` and drag it to a fresh spot on our code grid.

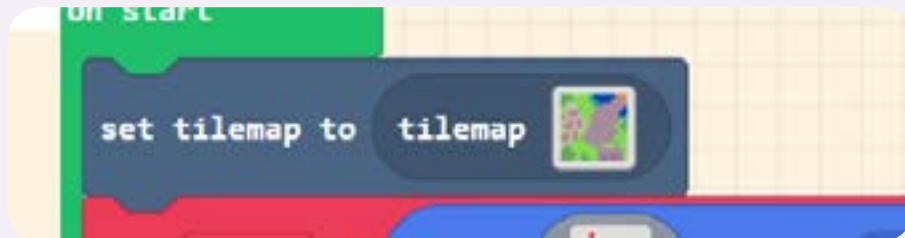


We have added a message, `show long text` and reset the game within this countdown end block. It must be inside the event so the game knows to run that code. If it was outside the block, it would not be run, and our game would have a bug in it.

## Section F - Stretch and Extension Task (Suitable for all games)

### Solution: Task 7 – Add walls by editing the `tilemap`

Double click on the `tilemap` which is the background of our game –



This will bring up the editor:



Add a wall by clicking on the wall icon in the left menu.



You can now draw along the edge of the volcano and create a wall. The wall should appear in red. Click Done.

Our van can no longer drive into the volcano – phew!

### Solution: Task 8 – Investigation

This is an open task – hope you had fun exploring differences between block coding and Python!



## Objectives

### Technology/Computing

- I can design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- I can use sequence, selection, and repetition in programs, work with variables and various forms of input and output
- I can use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- I can design, use and evaluate computational abstractions that model the state and
- I can understand several key algorithms that reflect computational thinking
- I can explore and use the features of a range of digital technologies, integrated software and online resources to determine the most appropriate to solve problems.
- I can describe different fundamental information processes and how they communicate, and can identify their use in solving different problems

### Geography

- I can describe and understand key aspects of volcanoes