Exploring Mars with SCRATCH

A game based learning activity

Author: Ntovolou Iliana

DiSTARS

ELLINOGERMANIKI AGOGI

Contents

Co-funded by the
Erasmus+ Programme
of the European Union

Table of figures

1 Introduction

1.1 Summary

In this activity, students create a Mars exploration game, using the Scratch programming language. Using their computer keyboard, they drive their exploration rover to the surface of the planet Mars and collect spectacles from its surface. Depending on the number of rocks they will collect and their avoidance from target traps, they win or lose points in their final score.
Students will engage in programming, analytical, and computational thinking and use mathematics and appropriate programming commands to plan their game.

1.2 Cognitive areas involved and prerequisites of students' knowledge

Simple computer skills, internet search knowledge, simple number comparison knowledge, simple math knowledge

1.3 Link to school curriculum

Connection with the faculty of informatics and the skills are needed in the fifth grade of primary school

1.4 Age of children

Fifth grade of Primary School, 10-11 years old

1.5 Estimated duration

4 teaching hours of 45 minutes each

1.6 Teaching resources (materials & technological tools)

1. Computer
2. Internet access/Google connection
3. SCRATCH programming language
4. Mars surface images
5. Mars rover image

1.7 Skills 21st century

This educational scenario will enhance students with the following skills, defined as 21st century skills:

- Computer handling and internet search skills
- Development of programming thinking
- Development of analytical and computational thinking
- Critical thinking/ decision making skills

1.8 Teaching approaches and learning strategies/theories

This activity is divided into multiple sections/ steps.
The first section is an introduction to the creation of a "mars rover" game using the scratch programming language. It can be completed as a stand-alone lesson or in combination with any of the following sections, depending on the students' and teachers' familiarity with Scratch and the programming languages based on "blocks" commands.
The game's creation is described in detail below in the order in which commands are created. As students create code, they test different blocks and explore what's going on. There are many ways in which this game can be created, depending on each student's creativity and resulting in creating their own characteristic game.
Throughout the development of the game, students should rotate other students' games to see if their game is operating as planned. The student-players should provide feedback on what happened when they played, as well as what they thought should happen. The game can also be tested and played on an interactive whiteboard.

2 Description of the scenario activities

2.1 Step 1– Create a folder with the images needed (10')

1. Right-click on the desktop of computer and create a new folder.
2. Save it on the desktop and enter as a document title, your name.

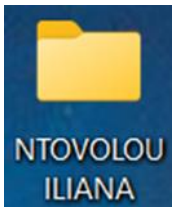


Figure 2-1: Create a folder and give your name as title

3. On the website: https://www.jpl.nasa.gov/ Type in the search bar "**Alba Patera Mars Images"** and select the image you like and save it as **"Mars_background"**, in your folder.
4. On the website: https://www.pngall.com/mars-rover-png choose whichever image you like and save it as **"Mars_rover"** in you folder.

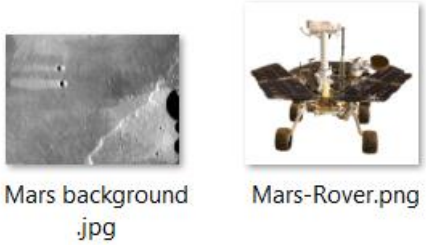


Figure 2-2: Download the background and rover images

of the European Union

2.2 Step 2 – Creating a background (10')

1. Log in SCRATCH's website: https://scratch.mit.edu/
2. Click on the **"Start Creating"** option, to start a new project.

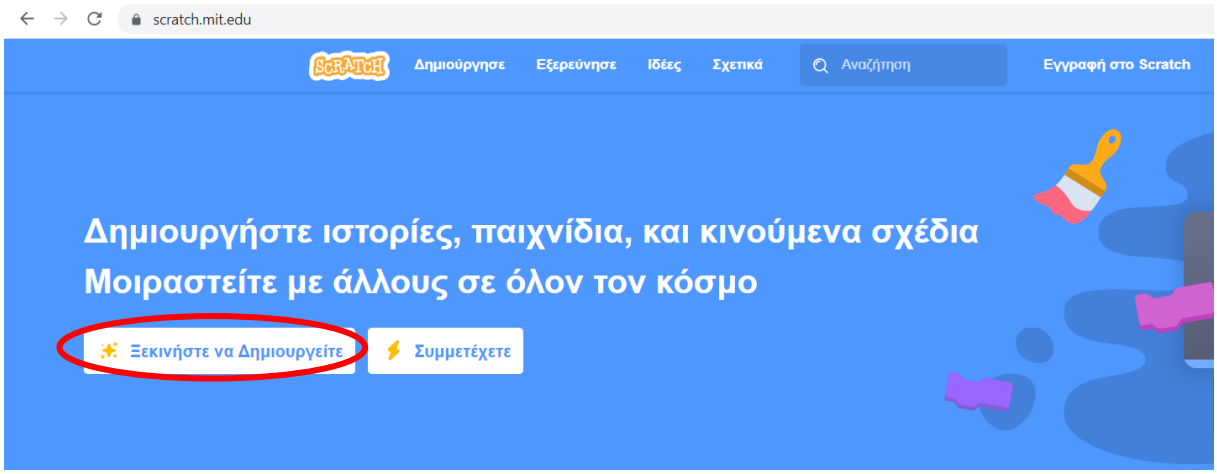


Figure 2-3: Log in SCRATCH's website and select the option "Create a new project"

3. Choose **English language** from the selection:

Figure 2-4: Choose English language

4. Close all the pop-up messages:

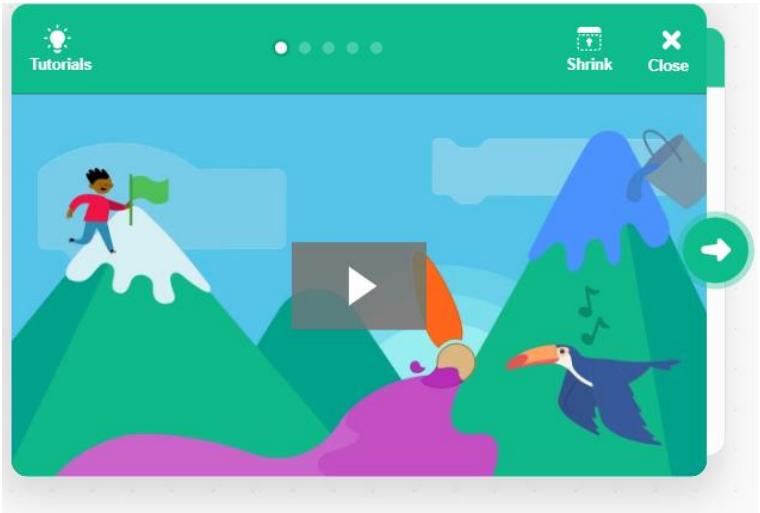


Figure 2-5: Close all the pop-up messages

5. The environment we see is:

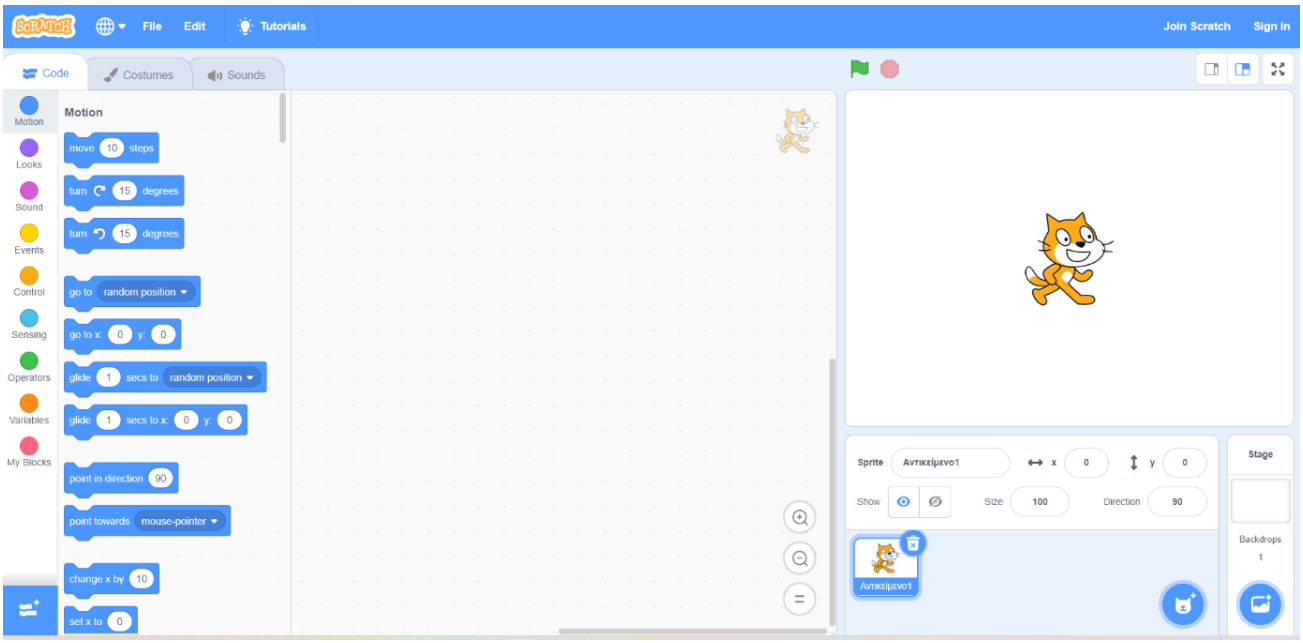


Figure 2-6: The environnent of SCRATCH

6. Create a new background by moving the mouse pointer over the option **Choose a Backdrop**:

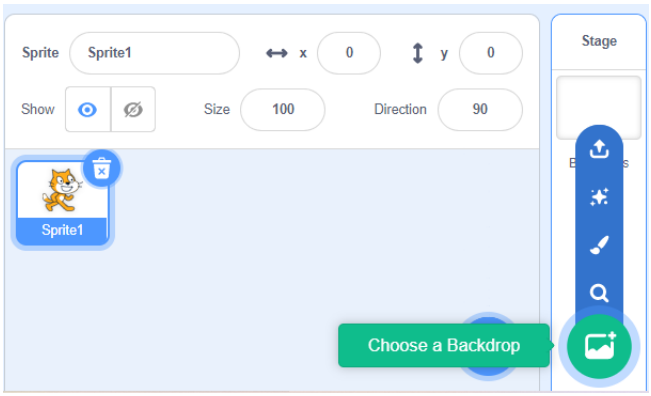


Figure 2-7: Move the mouse pointer over the Choose a Backdrop option

7. Click on the option **Upload Backdrop**:

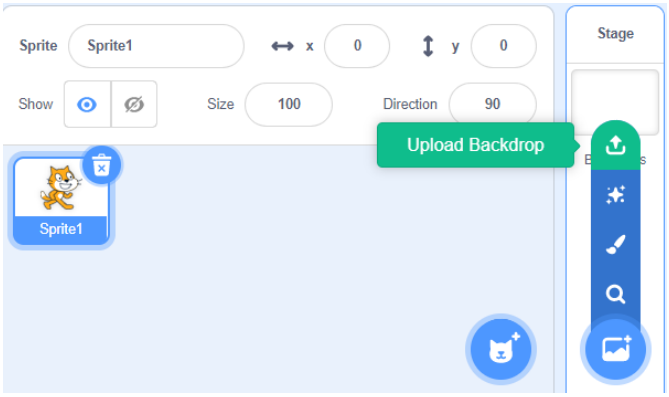


Figure 2-8: Click on the option Upload Backdrop

8. Select the image **Mars background** from your folder

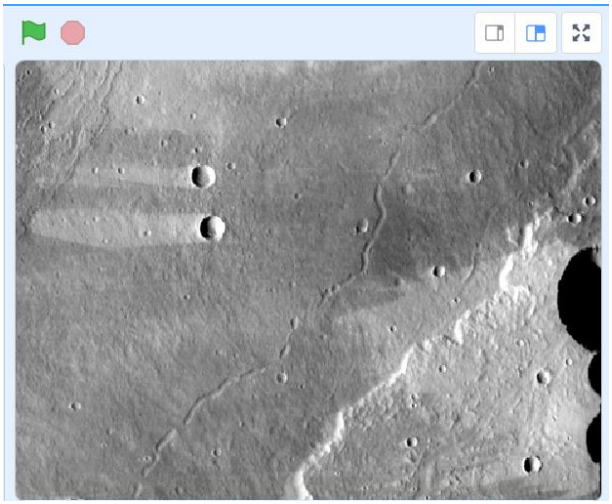


Figure 2-9: Select the image Mars background from your folder

9. Now, the environment is:

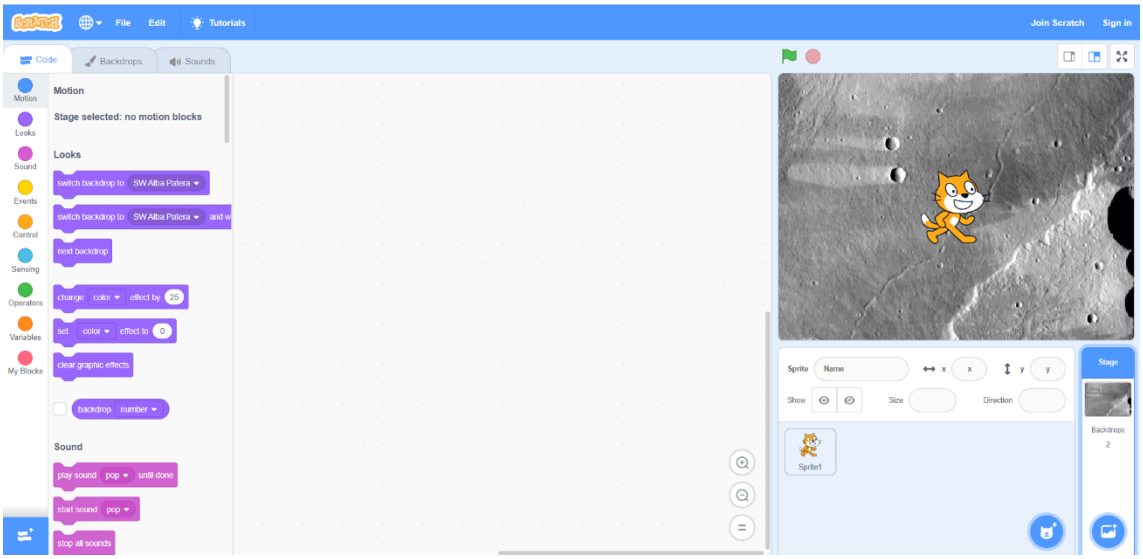


Figure 2-10: The environment of SCRATCH after uploading the background image

2.3 Step 3 – Upload the Mars rover sprite (10')

1. Delete the cat object by clicking the bin option, so you can add the items of your choice:

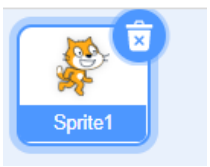


Figure 2-11: Delete the cat object

2. Add the **Mars rover** object by moving the mouse pointer over the option **Choose a sprite**:

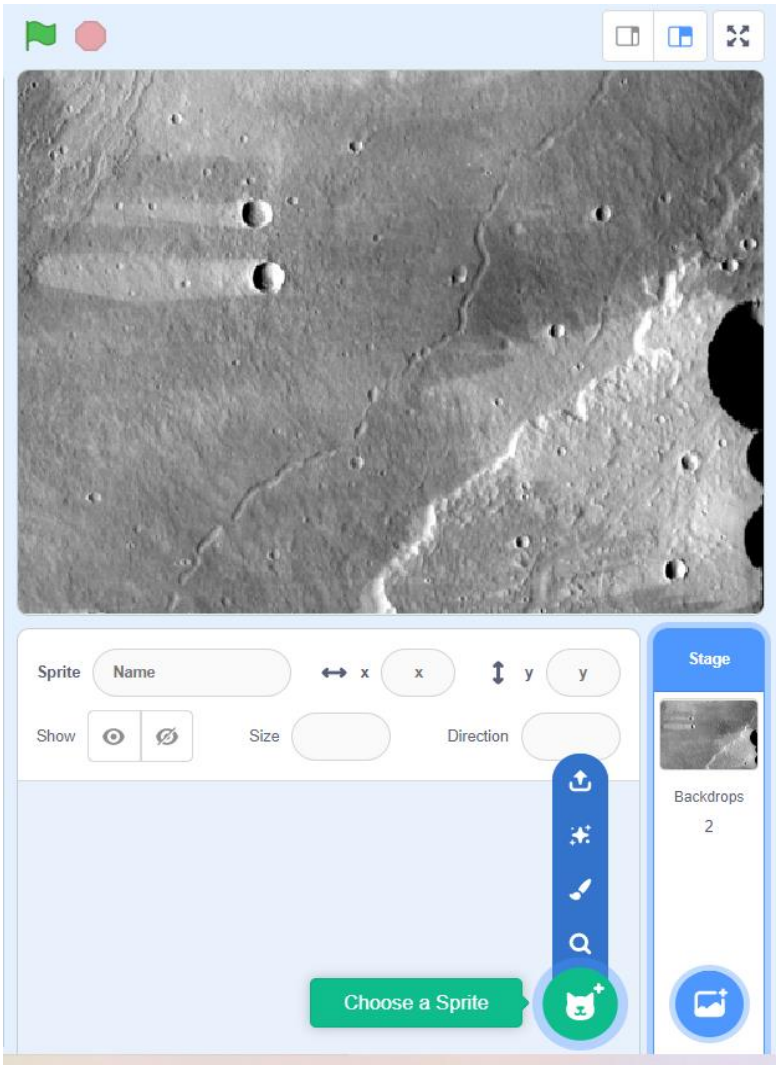


Figure 2-12: Choose a sprite

3. Select the **Upload sprite** option

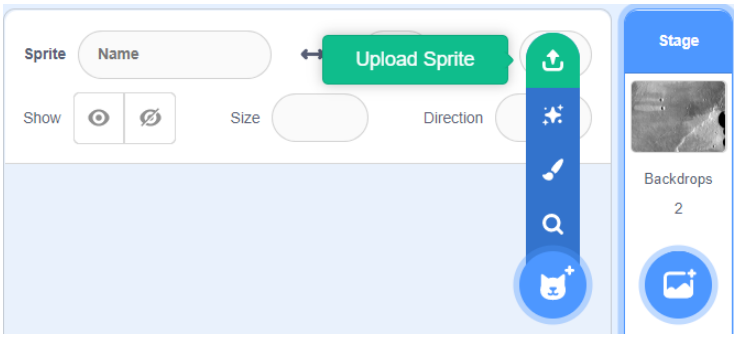


Figure 2-13: Upload sprite

4. Select the sprite **Mars rover** from your folder **and set the size up to 20.**

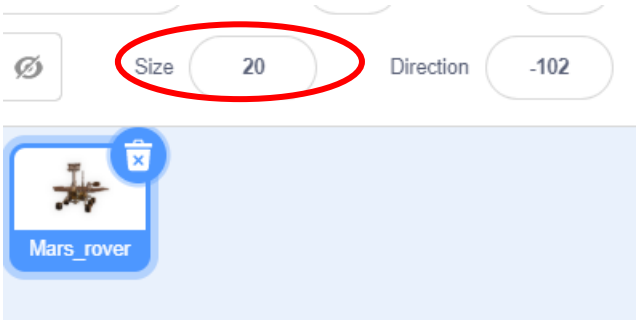


Figure 2-14: Select the Mars_rover sprite and set size up to 20

2.4 Step 4 – Upload the Rocks sprite (10')

1. Then, in the same way, add the object **Rocks** by moving the mouse over the option **Choose a Sprite:**

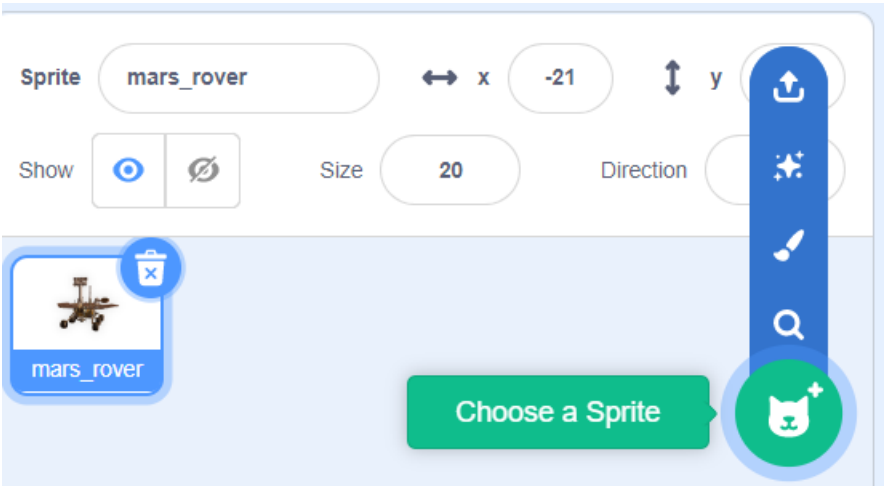


Figure 2-15: Choose sprite Rocks

2. Write on the search bar **"Rocks":**

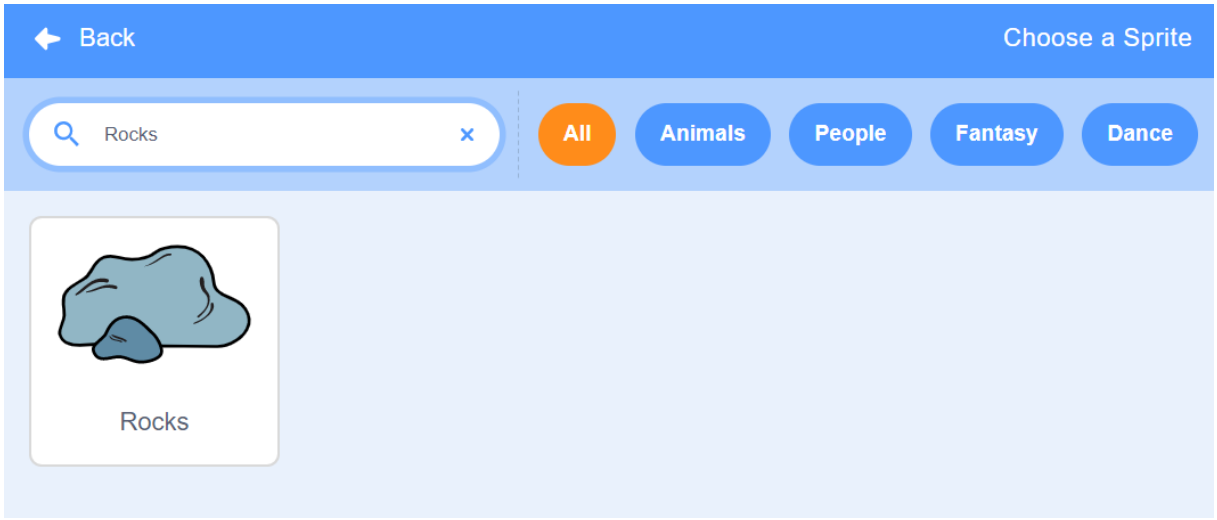


Figure 2-16: Searching the sprite « Rocks »

3. Set the size to 30:

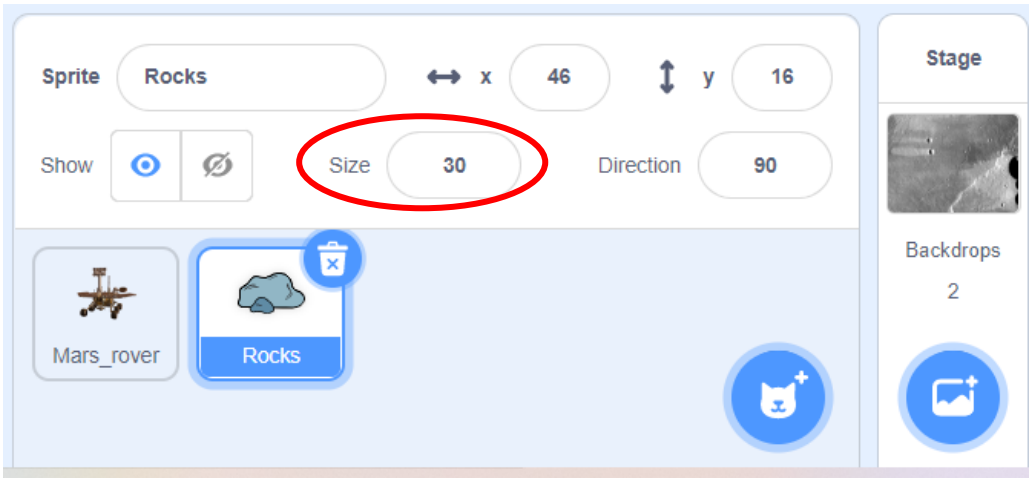


Figure 2-17: Upload sprite "Rocks" and set size to 30

2.5 Step 5 – Programming the Mars rover object (10')

1. Start programming the game by giving the first commands to the object **Mars rover**. For this reason, select the object **Mars rover**, by clicking it. Notice that it turns blue:

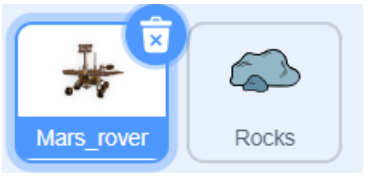


Figure 2-18: Start coding the Mars rover sprite by clicking on it

2. Click on the code environment to give the commands on the sprite

Figure 2-19: Click on the code environment

3. Our game will start every time the green flag is clicked:

Figure 2-20: When green flag is clicked, the game starts

4. When the flag is clicked, we want the **Mars rover** sprite to appear. Therefore, from the **Events** section, we select the command:

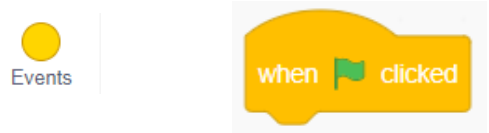


Figure 2-21: The "When green flag is clicked" command

5. We want the rover to show when the green flag is clicked, so choose from the **Looks** section the command **show**:

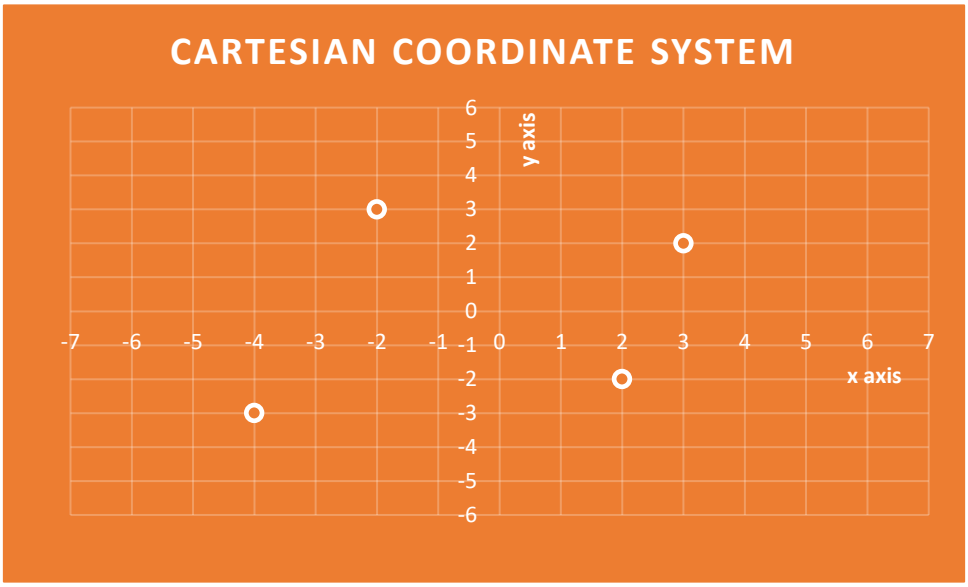
Figure 2-22: The Mars rover shows when the flag is clicked

6. Every time the game starts, we want **the Mars rover** to appear in a random position. So, from the **Motion** section, select the command **go to a random position**. It is needed to go to a random position first and then appear:

Figure 2-23: Mars rover appears in a random position

2.6 Step 6 – Directional commands of the Mars rover (20')

The **Mars rover** will be directed by pressing the **up-down-left-right** arrow keys of the computer keyboard.
Explain to students that the movement of an object in the **Scratch** environment is divided into the **x and y axis**. Click on the **Mars rover** in the game environment to see that the **values of x and y** can be both positive and negative, depending on the position of the **Mars_rover**.



In addition, we can show them the Cartesian coordinate system:

Figure 2-24: The Cartesian coordinate system

Is needed to add a command with which when the arrow is pressed then the **Mars rover** will make a specific movement.
From the Cartesian coordinate system, we see that when the **Mars rover** moves in specific direction, it follows specific axis, at the positive or negative direction:

Direction	Axis	Sign

right	x	Positive (+)
left	x	Negative (-)
up	y	Positive (+)
down	y	Negative (-)

Figure 2-25: Directions of the Mars rover

1. From the **Events** section, choose the command **when space key pressed** and select the right **arrow** option:

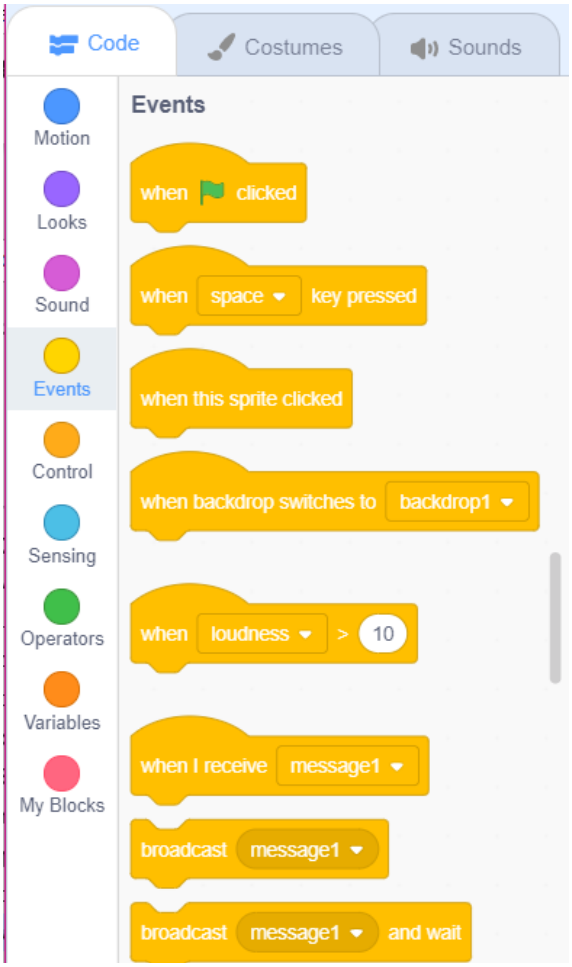


Figure 2-27: The commands of the Events section

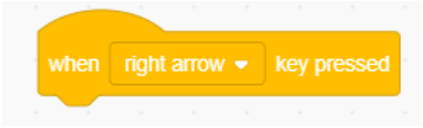


Figure 2-26: When right arrow key pressed command

2. Repeat the process for the other directions (left, up and down arrows):

Figure 2-28: When a specific arrow key pressed

3. Notice that when the right arrow is presses, the **Mars rover** is not moving yet. Give students the opportunity to process their commands to decide which command to add. Lead them to the conclusion that they need to add a command from the **Motion** section and explain to them that they can move their rover by using various commands. However, a command that moves the rover to the right is the command **change x by 10**. Select **step 5**, for a small move.

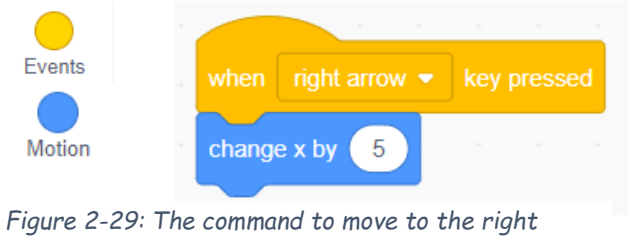


Figure 2-29: The command to move to the right

Correspondingly, to move to the left will be:

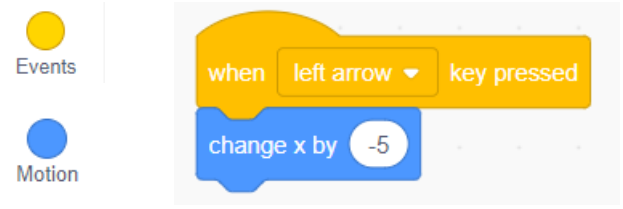


Figure 2-30: The command to move to the left

To move upwards will be:

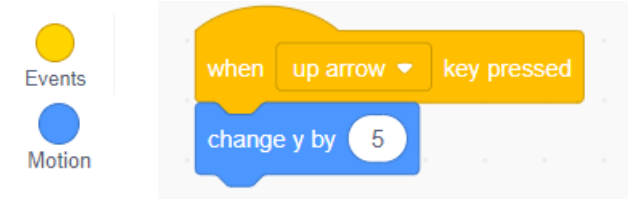


Figure 2-31: The command to move upwards

Finally, to move down:

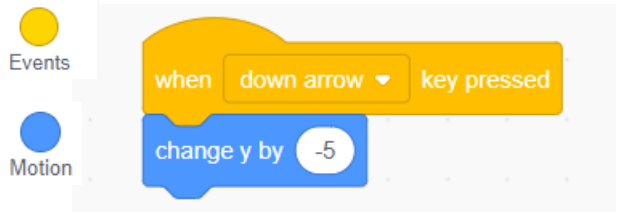
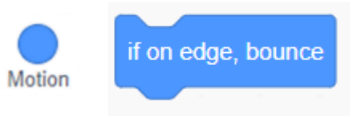


Figure 2-32: The command to move down

4. At this point, by pressing the computer arrow keys, **Mars_Rover** moves in all directions.
However, notice that when the rover reaches the limits of the surface of the game, it disappears, which is why we choose from the **Motion** section, the option **if on edge, bounce:**



5. Therefore, the program converts by far, as follows:

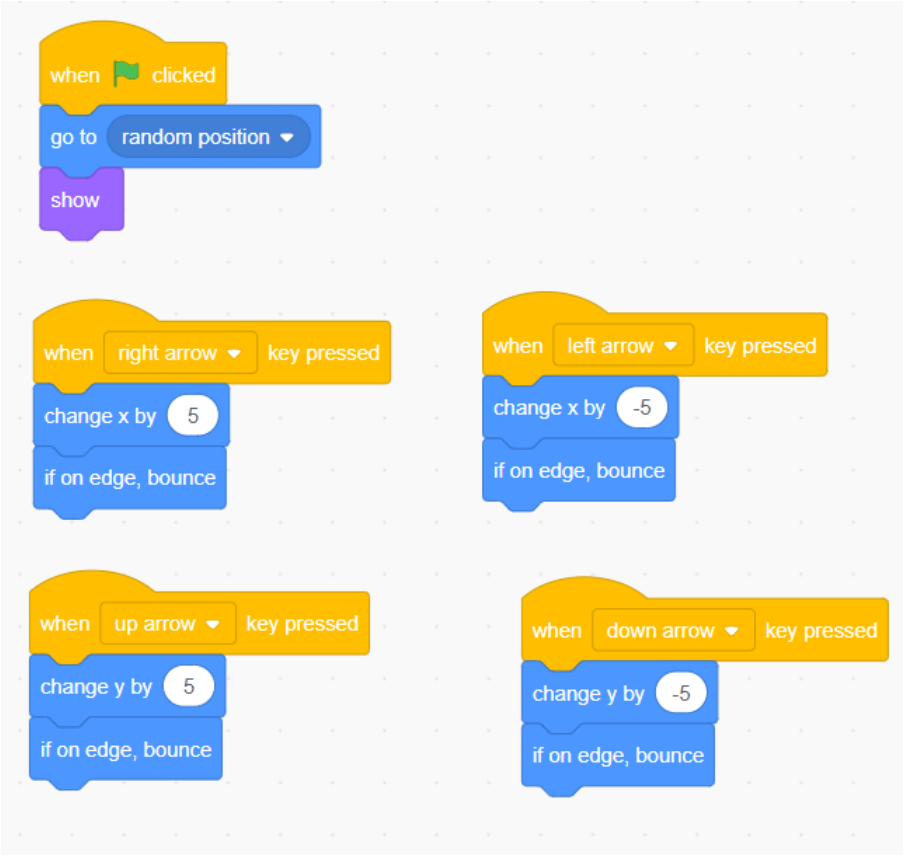


Figure 2-33: The programm until step 6

6. Notice that **Mars rover** changes direction when it touches the limit. Therefore, select click in the **Direction** box and select **do not rotate**, so it does not rotate when it touches the limit:

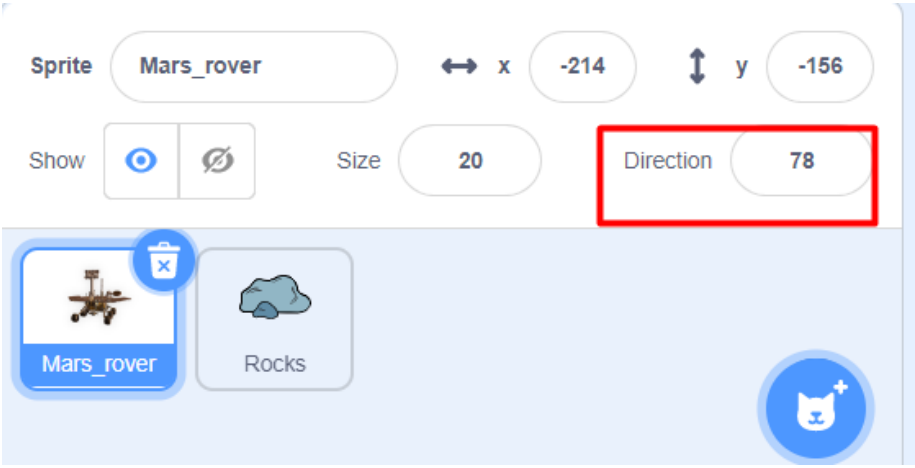


Figure 2-35: Click in the direction box

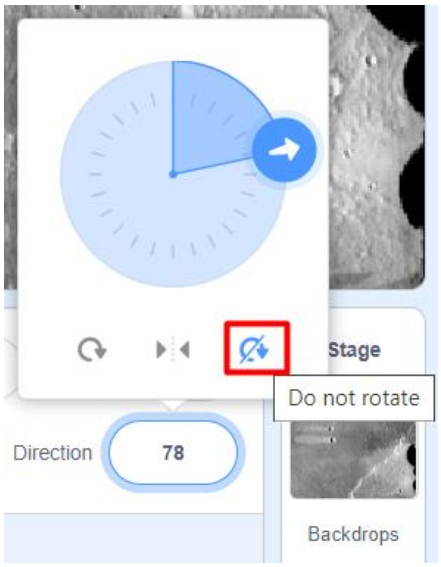


Figure 2-34: Select to not rotate

2.7 Step 7 – Programming the Rocks object (10')

The aim of the game is that when the **Mars rover** touches the **Rocks** object, disappears and then **appears in a random position** so that a new mission for the rover can begin.

1. At this point, select the subject of **"Rocks"** and select the Code environment in order to program it.

Figure 2-37: Select Rocks to give commands to it

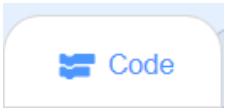


Figure 2-36: Click the Code option

2. As in the case of **Mars rover**, program the **Rocks** object every time the game starts (every time the green flag is clicked) to appear in a random location:

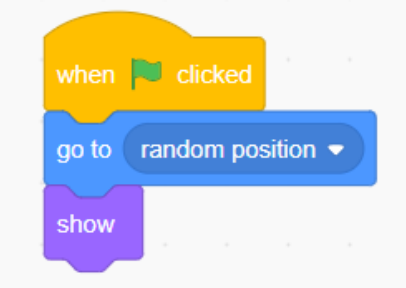


Figure 2-38: As the game begins, Rocks objectappear in random location

3. In addition, select the **if on edge, bounce** command so if it goes to the limit, it does not disappear:

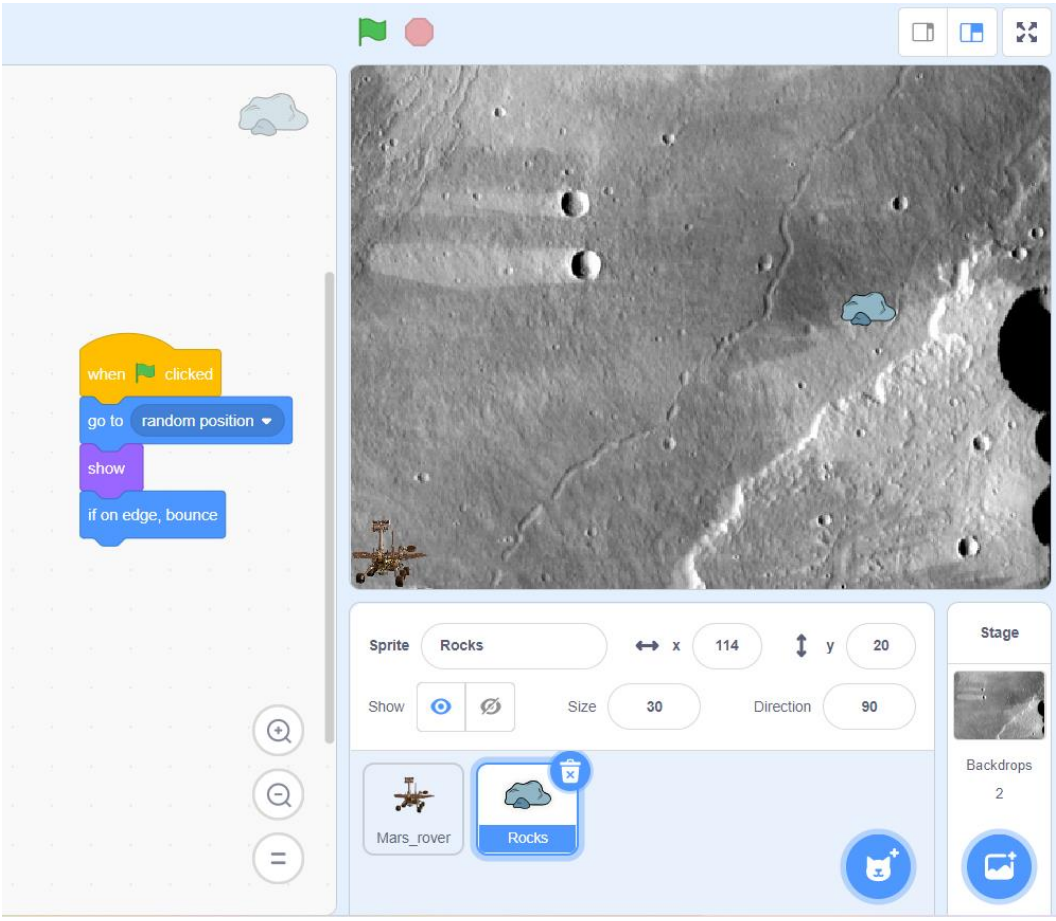


Figure 2-39: Giving the first commands to Rocks object

2.8 Step 8 – The "if" Control Command (10')

At this point, program the **Rocks** object so that when **Mars rover** touches **Rocks** ,it disappears and appears again in a random position.

1. The first case is: **"If the Mars rover touches the Rocks"**

Is needed to be checked whether this condition is being fulfilled.
At the **Control** section select command **if...then**:

Figure 2-40: The « If ...then » command

2. The condition is "if **Rocks** touches **Mars rover**". It is an energy that is **"felt"** by the rover. Also, notice that the hole at the command, has a diamond shape.
The command is located in the **Sensing** section.

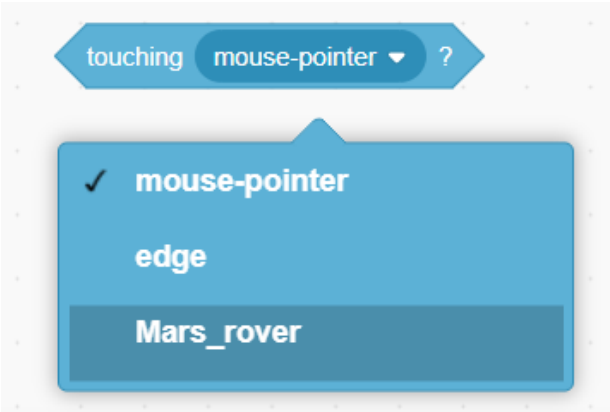


Figure 2-41: Touching Mars rover

If it touches the **Mars_rover** give commands to:

1. Play a sound
2. Disappears
3. Goes to a random location
4. And show up again

So, the program is now:

Figure 2-42: The program after the if command and the 4 situations that are fulfilled

2.9 Step 9 – Insert a time variable and time countdown (10')

At this point, add a time countdown to the game. Set the duration of the game. In addition, set time countdown, so that the time will be decreased by 1 second, until it becomes 0 and stops the game.

Time is a variable of the game because we can alter and choose its value as we want.

Program the time variable at the environment of **Mars rover**. So, choose the object of **Mars rover**, so it becomes blue and code it.

1. At the **Variables** section, select **Make a Variable** and call it **TIME**:

Figure 2-44: Make a variable

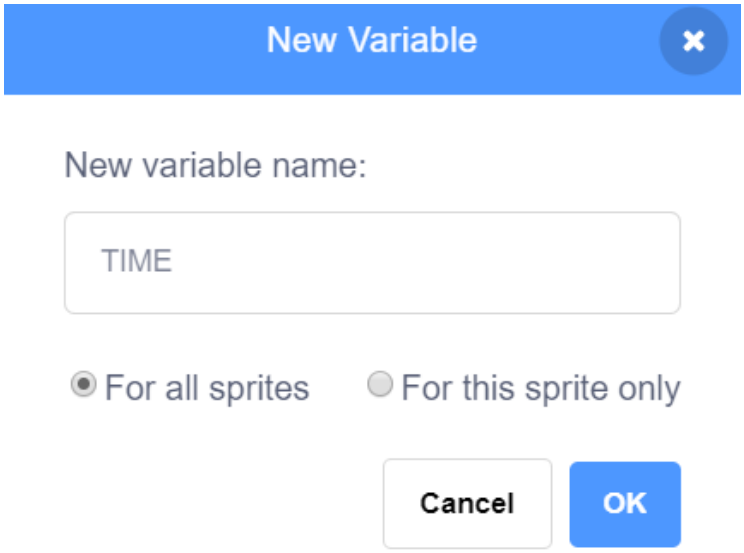


Figure 2-43: Name the new variable as Time

2. Start the countdown program by selecting the command **When the flag is clicked:**

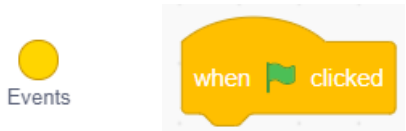


Figure 2-45: Begin a new program

3. Set the initial time of the game up to 60 seconds

   From the **Variables** section select the command **set time up to 60**

Figure 2-46: Set time up to 60

4. As **a second passes** the time variable **decreases by 1**. So, we select from the **Control** and **Variables** section, the commands above:

Figure 2-47: Decreasing time by 1 second

5. Time is constantly reduced, repeatedly by 1, until it turns **0** and stops the game. Therefore, select from the **Control** section, the command **repeated until:**

Figure 2-48; The repeat until command

6. It must be repeated until time becomes zero. Therefore, choose from the section **Operations**, the operator of equality and on the left of the equal sign, enter the **variable TIME** and to the right the number **0**. Therefore:

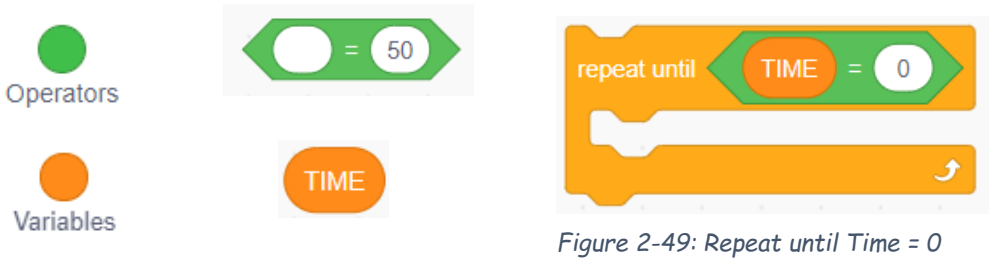


Figure 2-49: Repeat until Time = 0

7. In the repetition put **only** the commands about the decreasing of time because what is repeated is **only** the decrease of time by 1.
So, our overall countdown program is:

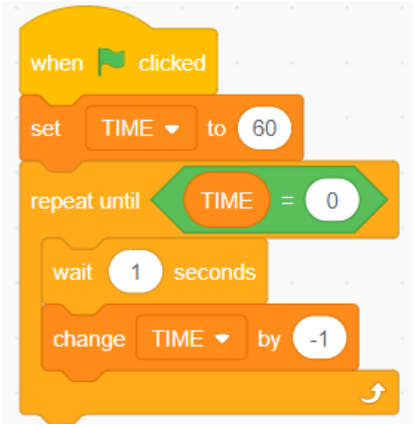


Figure 2-50: The time countdown program

2.10 Step 10 – Enter a Score Variable with step 10 (10')

Similarly, as the time counter, we will also insert a score counter.
Every time touches **Rocks**, score increases by **10**.

1. Continue the program have made for the **Rocks** object, so click on the **Rocks** code environment
2. From the section **Variables** select **Make a Variable** and name it **SCORE**

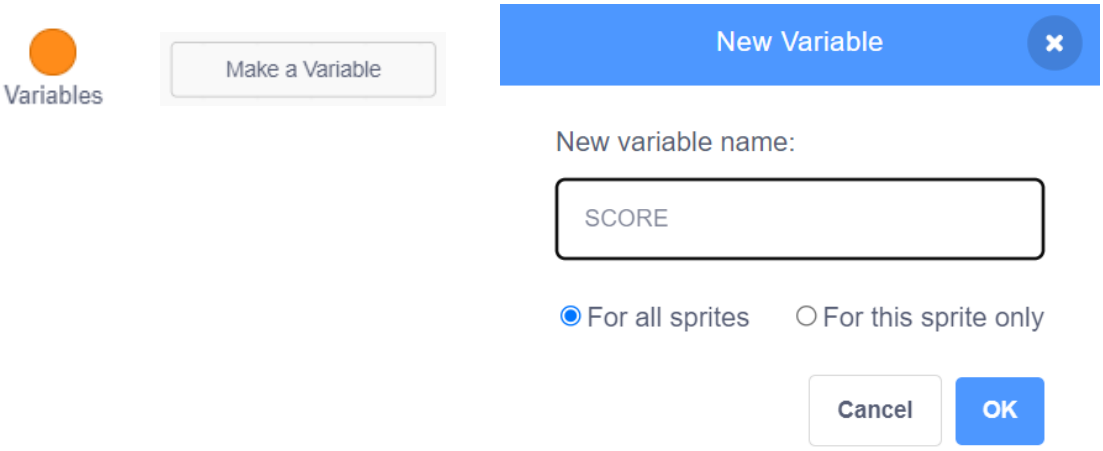


Figure 2-51: Make a Score Variable

3. At the beginning of the game the **SCORE** is 0, so set the variable score equal to 0, using the command **set Score to** from the section of **Variables**.

Figure 2-53: Set Score to 0

Figure 2-52: The program so far

4. When **Mars rover** touches the **Rocks**, the **SCORE increases by 10**. So, add after the **Control command "if"**, the command **changed score by 10:**

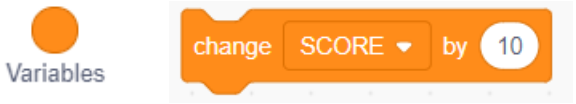


Figure 2-54: Change SCORE by 10

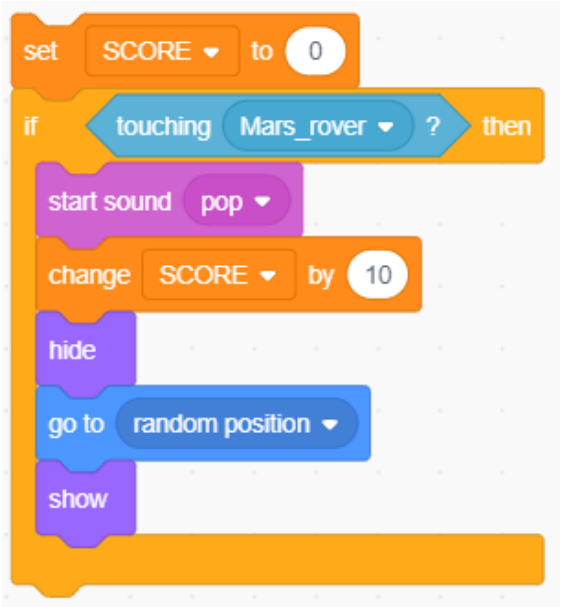
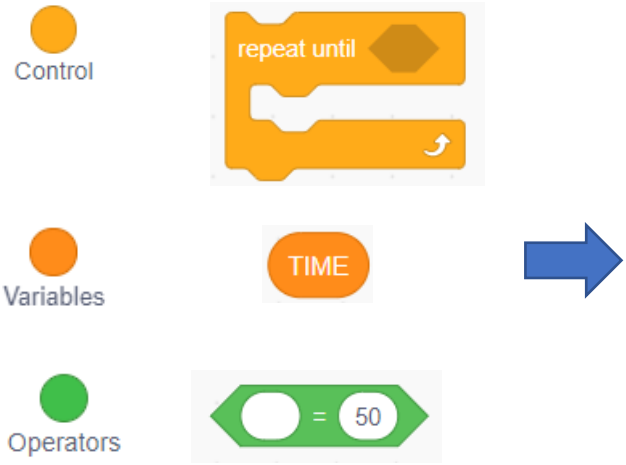


Figure 2-55: The program after step 10

2.11 Step 11 – Continuation of the program of the Rocks (10')

1. The game will end when the time becomes zero, so all commands should be repeated until the time is 0. So put the **If** command, inside a **repeat until** command, that you will find at the **Control** section:



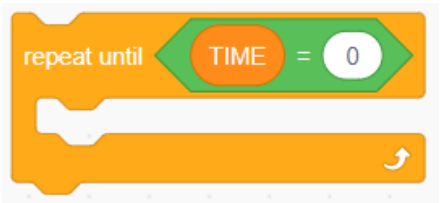


Figure 2-56: Repeat until TIME = 0

Therefore:

Figure 2-57: The program after the repeat until command

When time becomes zero, the object **Rocks** disappears. So, put from the **Looks** section the command **hide**, **out from the repeat until** command:

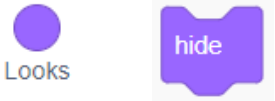
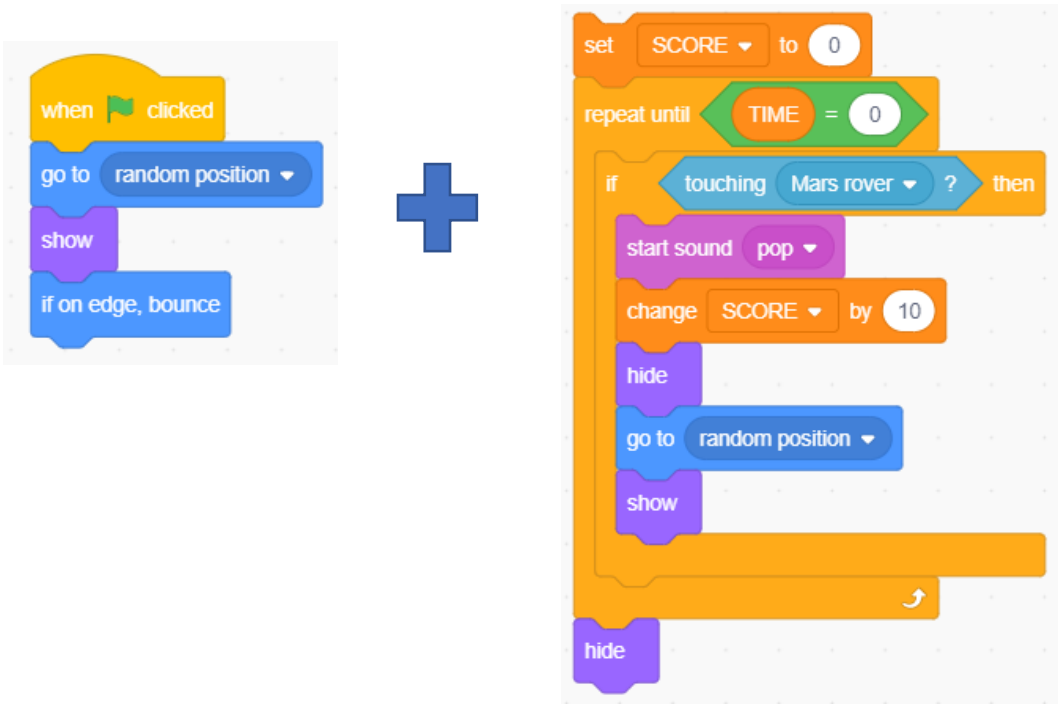


Figure 2-58: After the command hide

The program of **Rocks** object was at step 7:
It becomes:



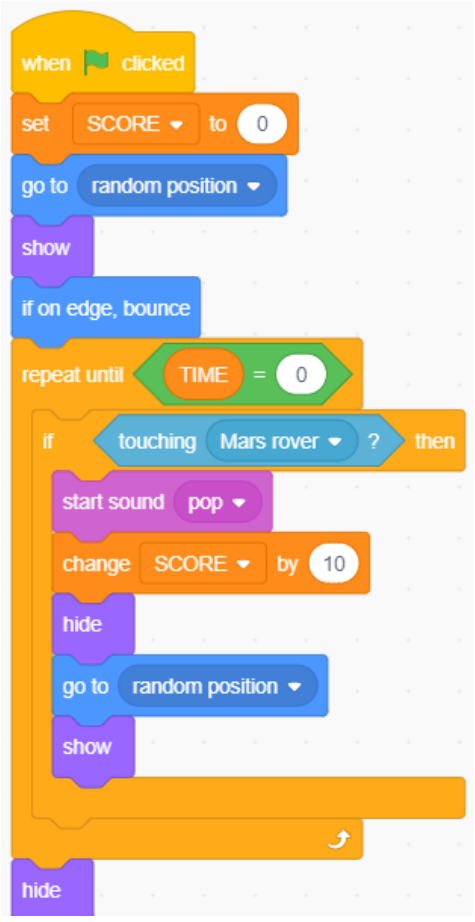


Figure 2-59: Total program of Rocks sprite

2.12 Step 12 – Continuation of the program of Mars rover (10')

From **Step 10**, the program of Mars rover is:

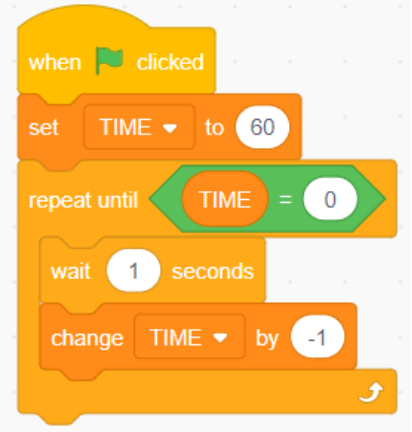


Figure 2-60: Program from step 10

1. **Mars rover** appears when the game starts. So, we enter the command **show** from the section **Looks**:

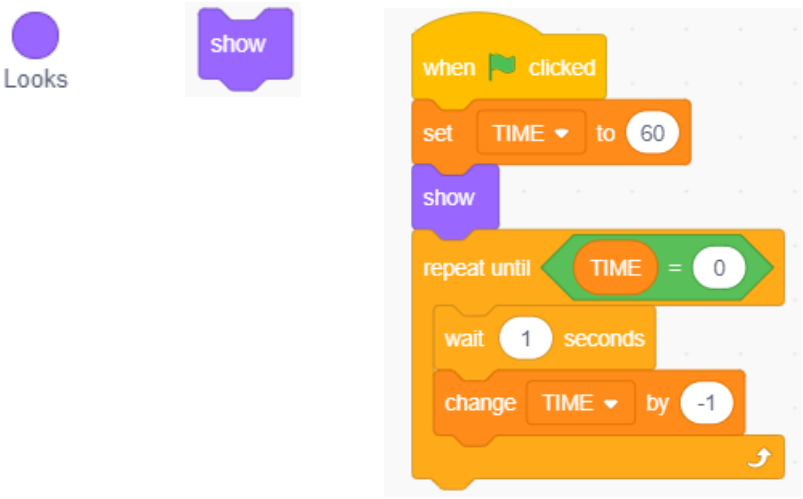


Figure 2-61: After the command show

2. When time turns 0, pop-up a message that says:

"GAME OVER!! THE SCORE =" and at the same time display the variable **SCORE**, the points that the player has collected.

To achieve this, from the Section **Looks**, choose the command **say Hello for 2 seconds**. The message will be displayed for 2 seconds so that the player has the time to read it.

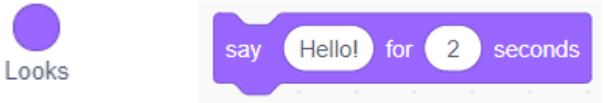


Figure 2-62: The text command

3. The text will appear in join with the variable of **SCORE**. Therefore, is needed a command that joins them. This command is in section **Operations**, and it is the command **join**:

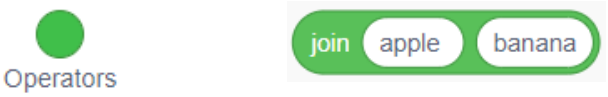


Figure 2-63: The command join

4. Put the join command inside the Hello hole:

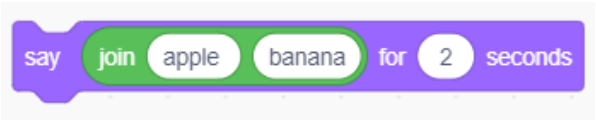


Figure 2-64: Join the text with the SCORE Variable

5. Where is the word apple, write **"GAME OVER!! THE SCORE ="** and where is the word banana put the variable **SCORE**:

6. Then, after GAME OVER, the **Mars rover hide and stop the game:**

Figure 2-65: Mars rover hides and the game stops

7. The program of Mars rover is now:

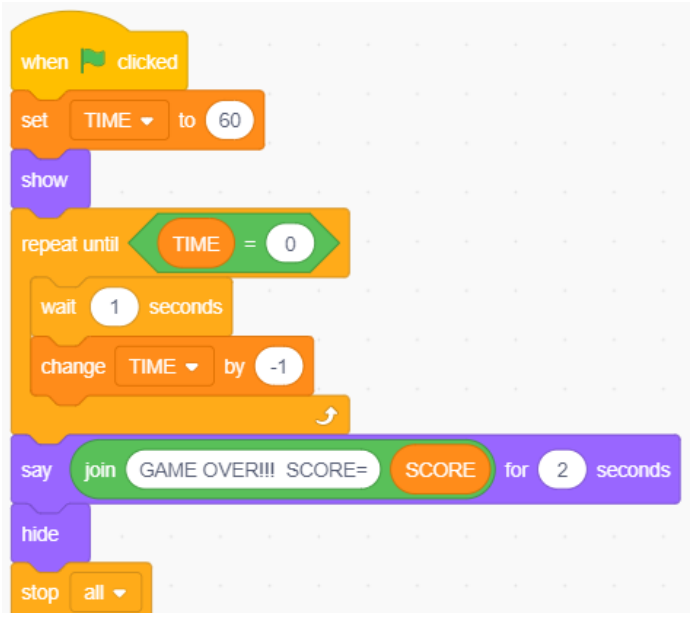


Figure 2-66: The program of the Mars rover sprite until step 12

2.13 Step 13 – Enrichment of the programm (20')

At this point, enrich your game with additional commands.

Draw **areas of different colors in the background image** so that when **Mars rover** passes through them, will trigger commands- traps.

1. Paint 4 areas of your choice at the background, with a different color. To achieve this, choose the **Stage** option and then the **Backdrops** option and then paint either by inserting a **circle or line or your own design**:

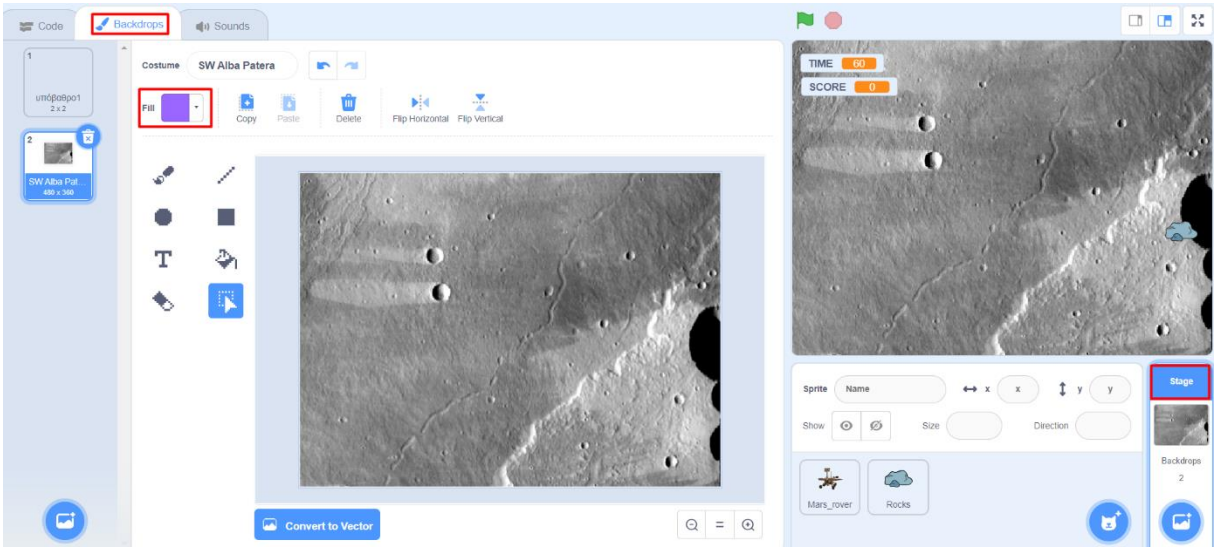


Figure 2-67: Paint 4 areas of the background

In this case, the following colors and areas have been selected:

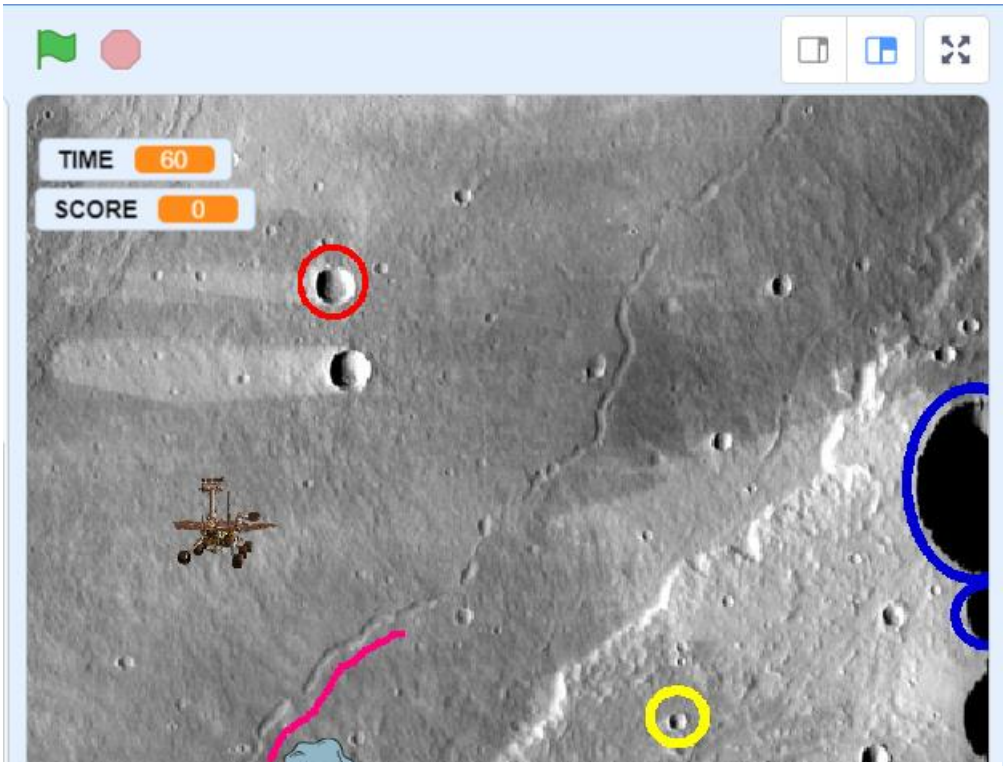


Figure 2-68: The colours and areas which have been selected

2. Program the specific areas as follows:

A. When it touches **red** color, the **SCORE** will be **reduced by 5**

B. When it touches **yellow** color, the **TIME** will be **reduced by 1 second**

C. When it touches **blue** color, **it loses and the game stops**

D. When it touches the **pink** color, **it will pop-up an information message** about the planet Mars or the specific background

For all four cases we choose from the **Control** section, the command **if then**. This control command is needed to be **forever** checked if it is fulfilled. So, in addition select the command forever from the **Control** section:

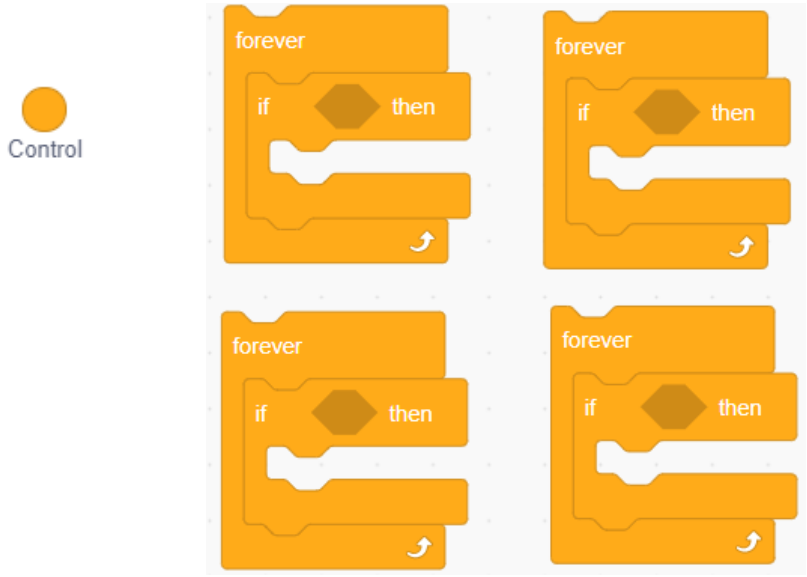


Figure 2-69: Forever and if then commands

3. For each of these structures, insert from the **Sensing** section, the command **touching color**:

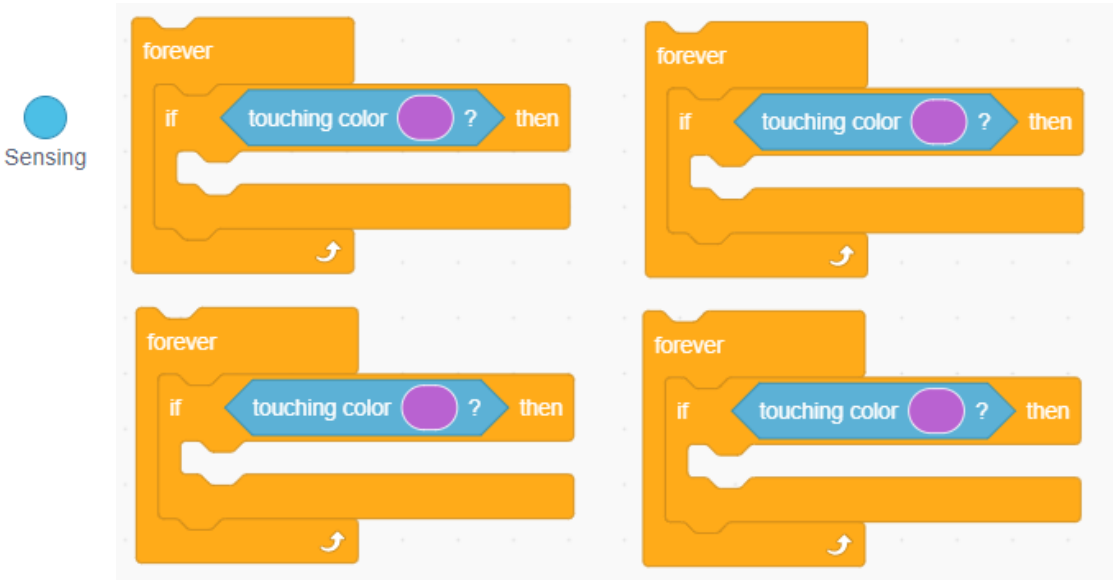


Figure 2-70: The touching colour commands

4. For each loop, choose the specific color you have paint. Click in the color choice and then the icon  and zoom in the background:

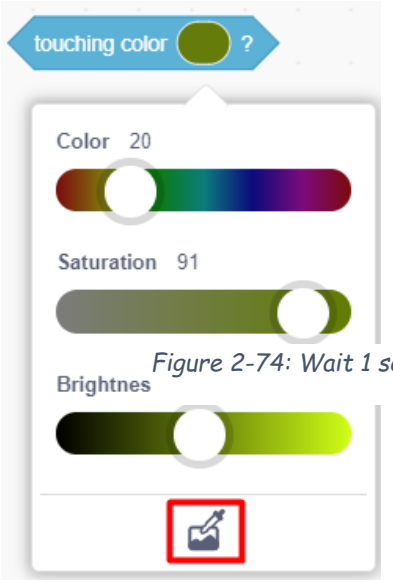


Figure 2-72: Select your specific colour from the background

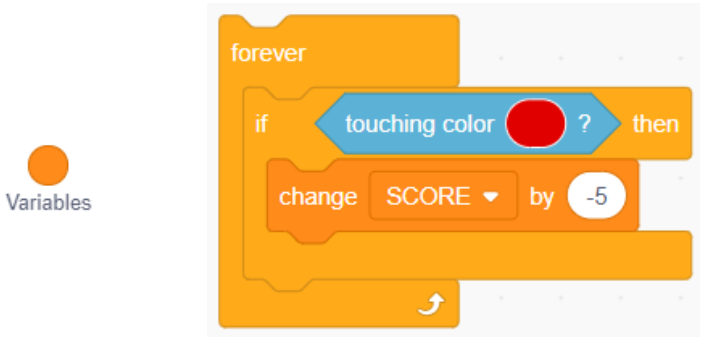
Figure 2-71: Zoom in the background

Figure 2-74: Wait 1 second and then change SCORE

5. Then, it is analyzed specifically for each case, the commands that are needed to be entered:

A. When Mars rover touches red color, the SCORE will be reduced by 5

Therefore, within the loop, enter the command **change SCORE by -5,** from the section **Variables:**



Notice that, **SCORE** decreases **continuously and quickly**, for this reason we enter the command to **wait 1 second** and then change the **SCORE**, so that it has time to leave the red area and then change the **SCORE**:

Figure 2-73: When red is touched, SCORE decreases by 5

B. When it touches the yellow color, TIME will be reduced by 1 second.
Therefore, insert in the loop the commands for the time countdown as described in a previous chapter:

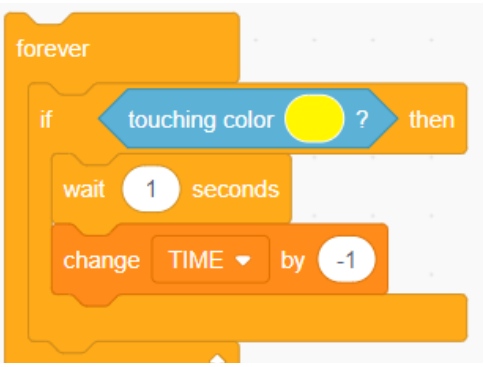


Figure 2-75: When it touches yellow color, TIME decreases

C. When it touches the blue color, a pop- up message emerges "IT'S A TRAP!!! GAME OVER!! SCORE =" and display the SCORE Variable.
To achieve this, **join** the **"say Hello"** command with the variable of **SCORE**.
First, select from the **Looks** section the command **Say Hello for 2 seconds**.

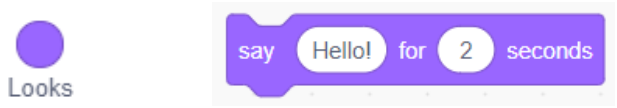


Figure 2-76: Say Hello command

Then, from the **Operators** section, sellect the command **join**:

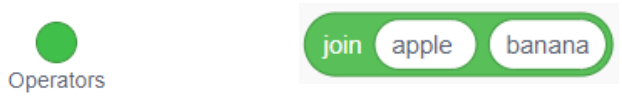


Figure 2-79: The join command

After, insert the **join** command **inside the Hello**:

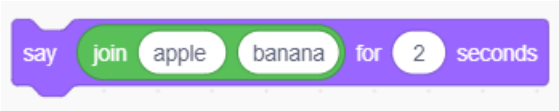


Figure 2-78: Join Say Hello command with Join command

Then, replace the word apple with the text **"IT'S A TRAP!!! GAME OVER!!! SCORE ="** and inside the "banana" gap insert the Variable **SCORE**:

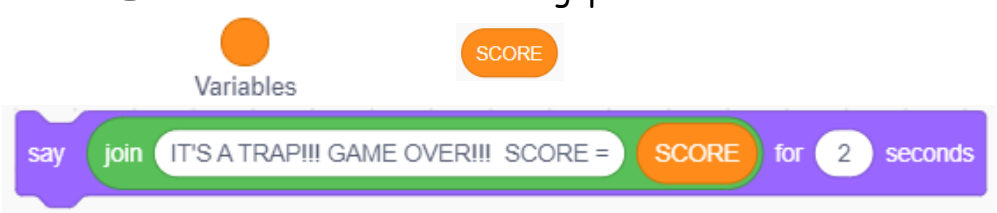


Figure 2-77: Insert Text and Variable SCORE

Erasmus+ Programme of the European Union

After hiting the trap area, the **Mars rover** disappears and **TIME** stops, so from the section Looks, sellect the hide command and from the section Control the stop all command.
So the program becomes:

D.When it touches the pink color, an information pop-up message about the background we selected will emerge.
The message is **"YOU JUST TOUCHED TRACES FROM LAVA FLOWS"** and will pop-up **for 1 second**.

Therefore, the program becomes:

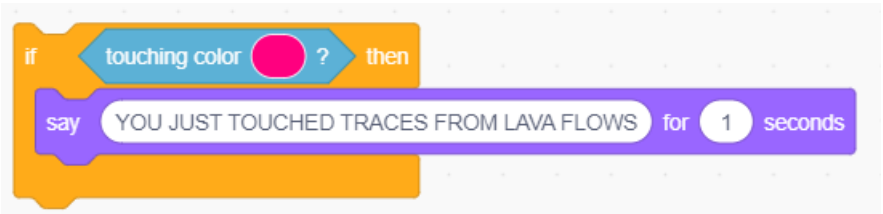


Figure 2-80: "You just touched traces"

The full program for the 4 cases is:

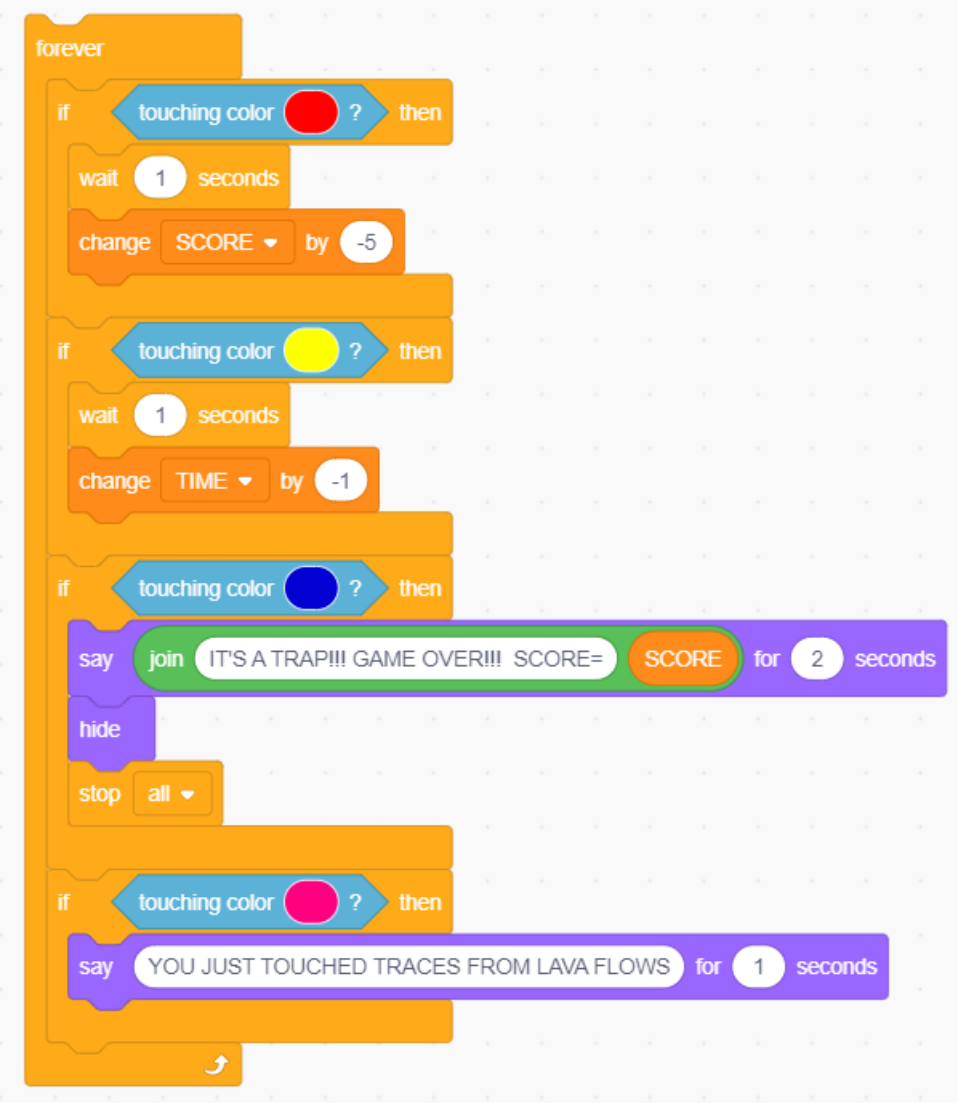


Figure 2-81: The program with the 4 color cases

Connect this program with the program of Mars rover at step 5 and the total game commands are:

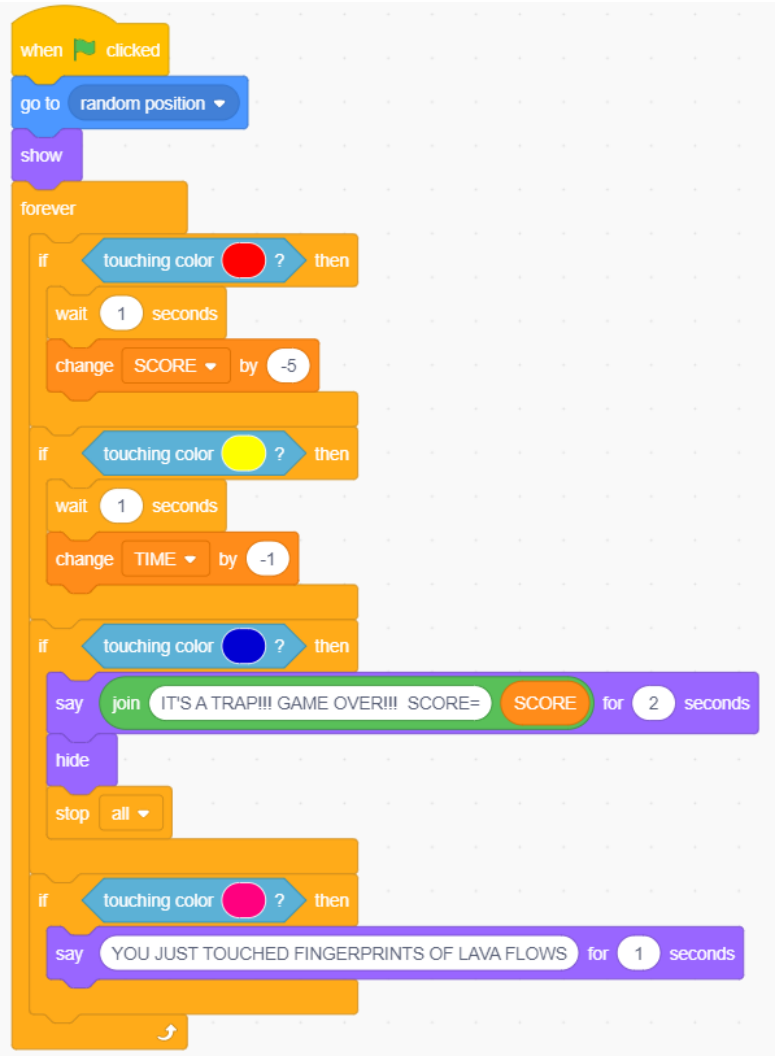
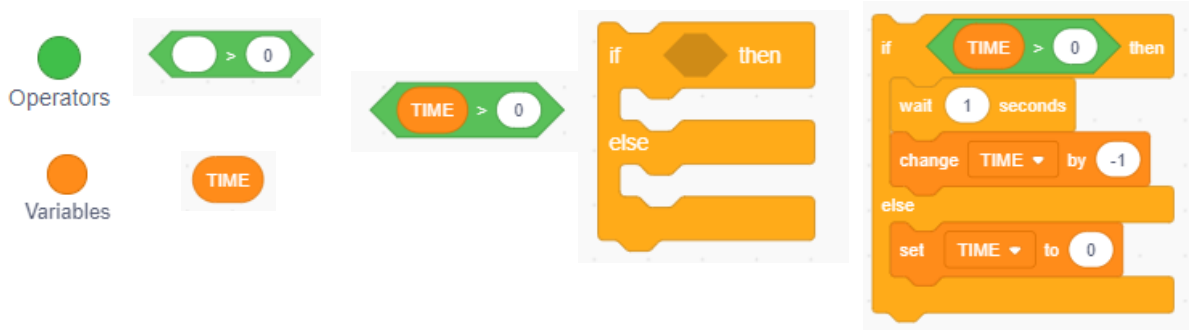


Figure 2-82: Program of Mars rover

2.14 Step 14 – Non negative value of TIME variable (10')

Notice that after a run of the game, **TIME variable** becomes **negative** and in some cases **the negative TIME does not stop increasing**. For this reason, add the last commands to ensure positive or equal to zero **TIME**, in the case of yellow and blue colors.

For the **yellow** control loop, from the **Control** section chooce the **if then else** command and from **Operators** section chooce the …>… command. **If TIME>0, decrease TIME by 1 second, else (TIME becomes negative) set TIME to 0.**



So the program becomes:

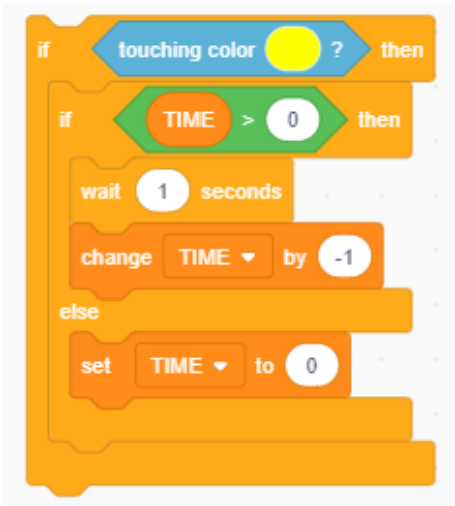


Figure 2-83: Non negative TIME for the yellow loop

For the **blue** control loop, from the **Variable** section, choose **set TIME to 0**:

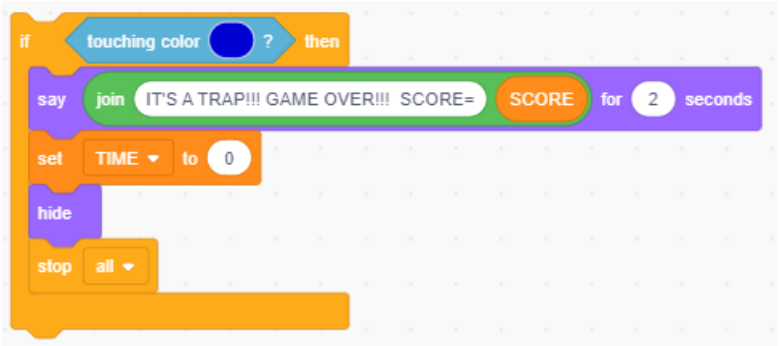


Figure 2-84: Non negative TIME for the blue loop

The total program becomes:

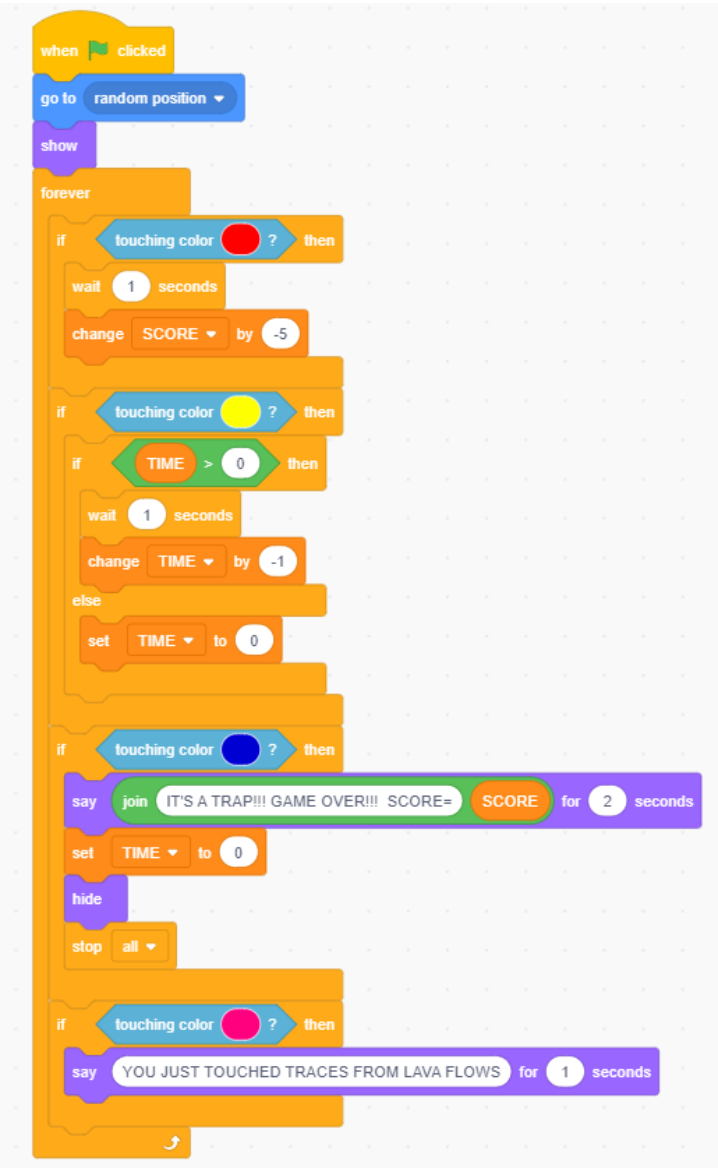
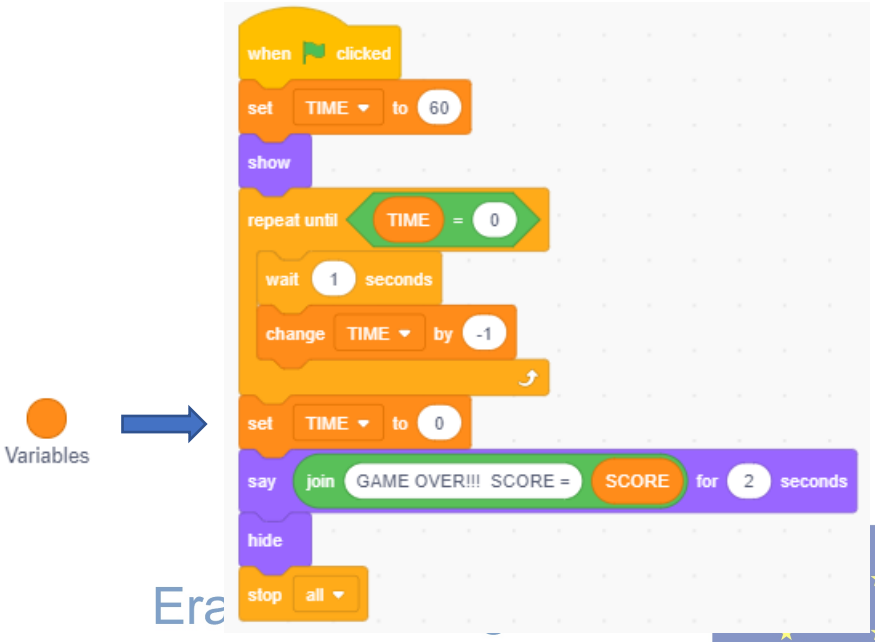


Figure 2-85: Total program after step 14

Moreover, **set TIME to 0** at the total program of step 12 of **Mars rover**:



2.15 Step 15 – Total game commands

The total game commands for the Rocks code environment are:

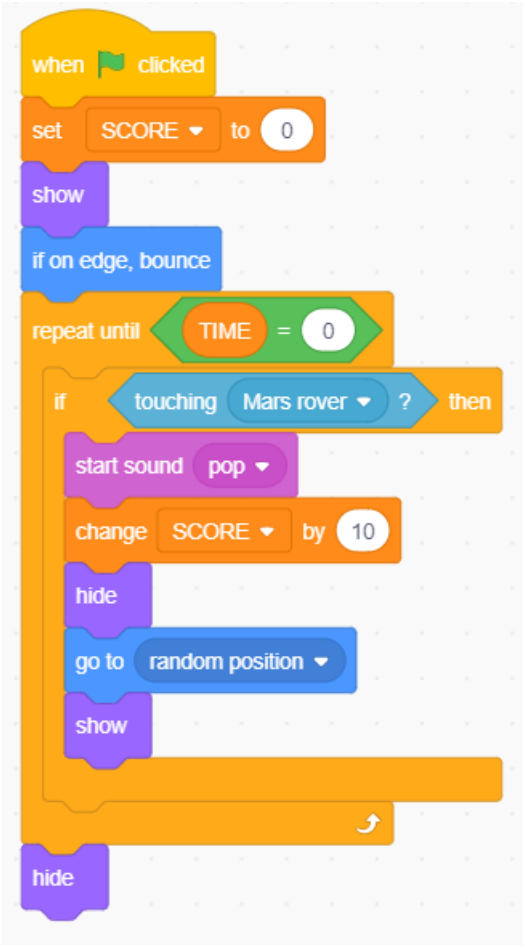


Figure 2-86: Total game commands for Rocks

The total game commands for the Mars rover code environment are:

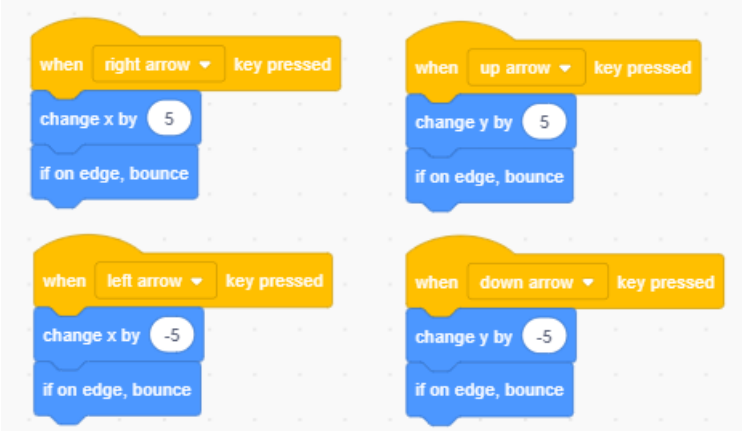


Figure 2-87: Total game commands for Mars rover, part 1

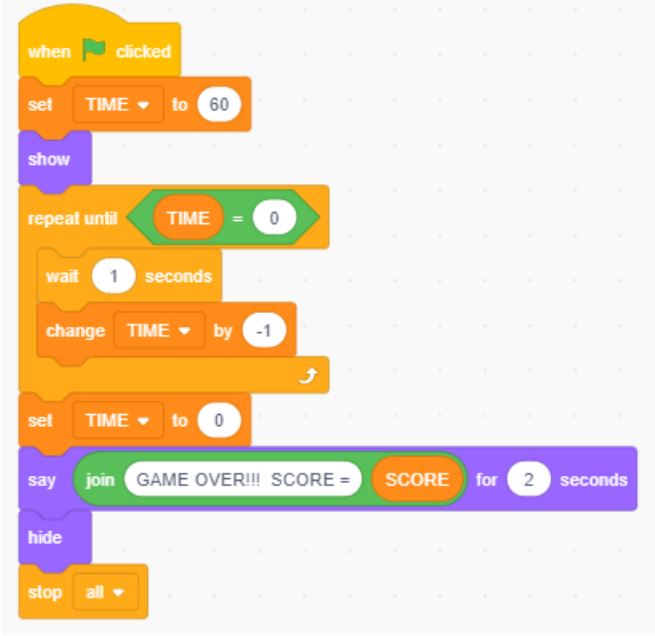


Figure 2-88: Total game commands for Mars rover, part 2

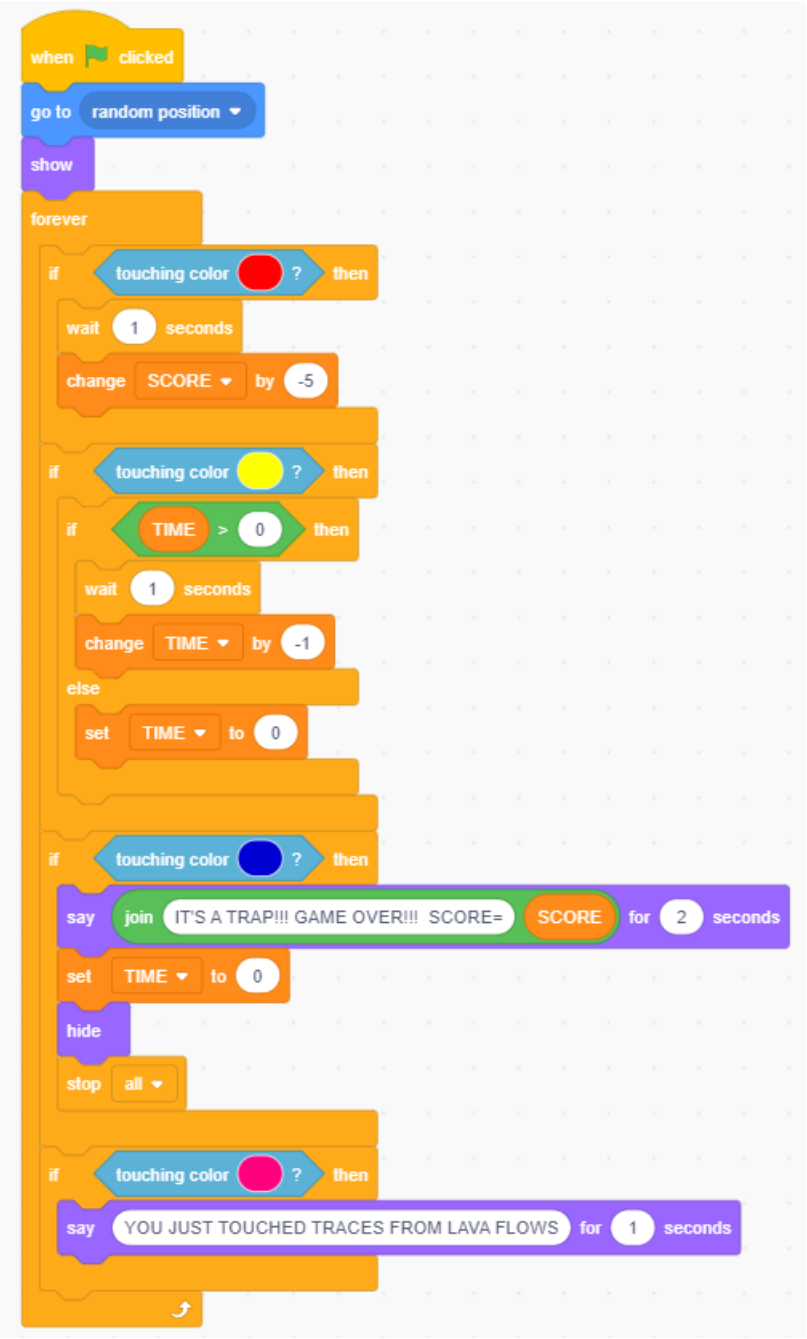


Figure 2-89: Total game commands for Mars rover, part 3