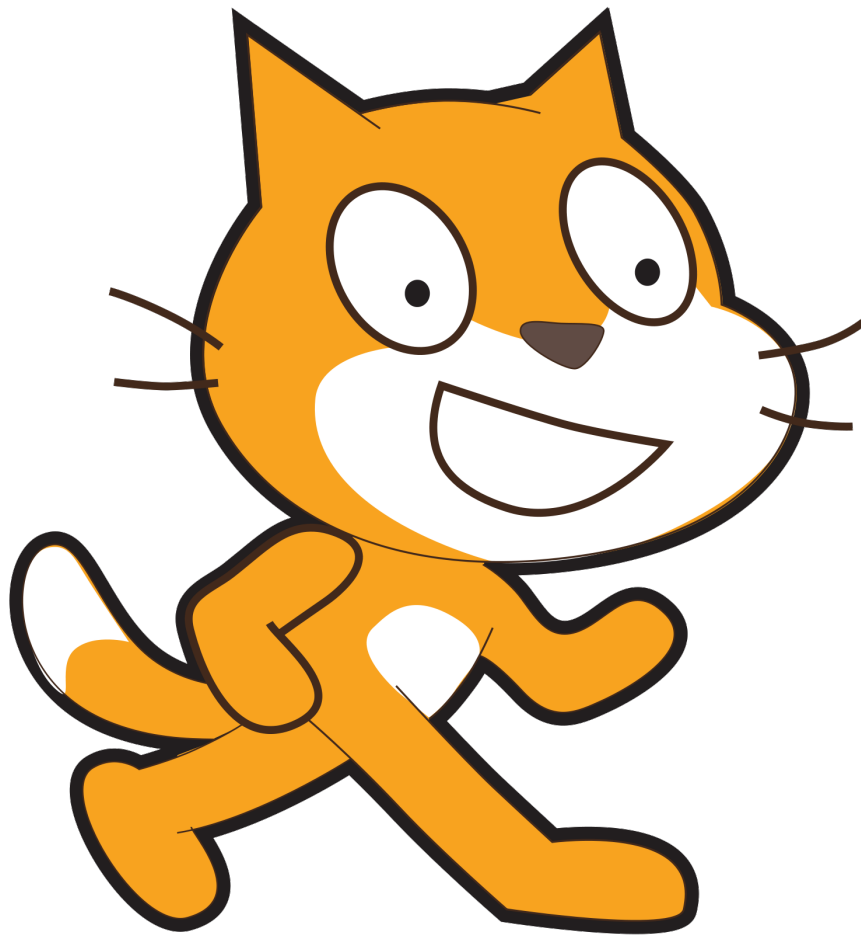# technoteach

technocamps

Advanced Scratch Workbook

Scratch

If you have not used Scratch before, please refer to our "Scratch for Beginners" workbook.

In this workshop you will be using a software called Scratch, a drag and drop environment you can use to build your own games easily. You can learn fundamental programming principles without having to worry about any spelling or typing errors and enjoy creating and playing your very own game.
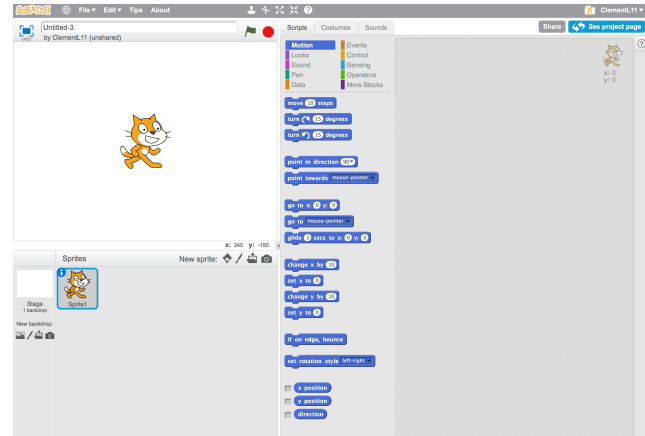
During this workbook you will be looking at how to advance your Scratch skills applied within the previous Scratch workbook. So far you should be able to comfortably add sprites, create a basic animation, code movement, collision and even implement variables such as scores or timers. Within this workbook you will be creating a maze with a difference! You will be developing a game to use as a teaching resource. As your character progresses around the maze, the user will be asked various questions and can only proceed if the user gets the answers right!

> **Note: You can use a past game for this workbook.**

1. Adding Sprites and creating a background

Create a basic maze game, following the instructions from the previous workbook. To progress you will need a main character, some collectables and a maze.
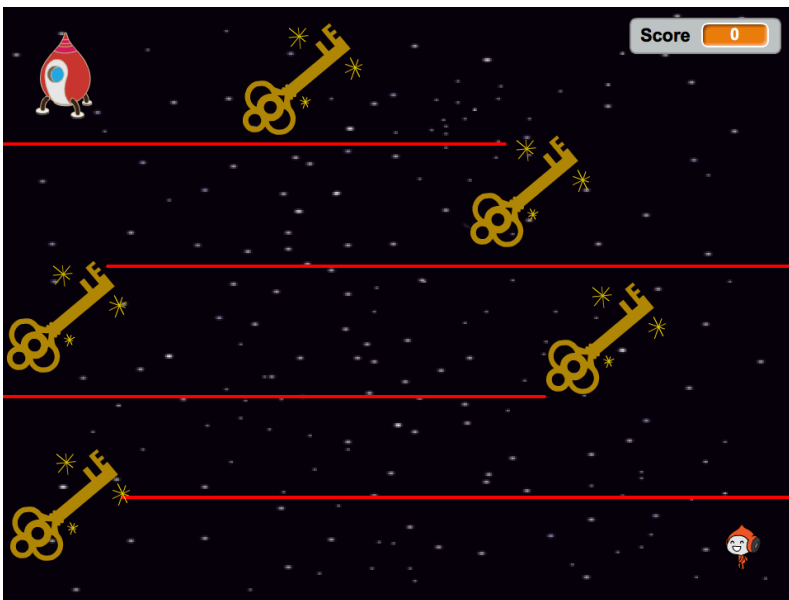
On the left, you can see a basic game created. The theme of this game will be a teaching resource testing student's knowledge about "Space".

As you can see, the main character is the sprite in the bottom left hand corner, needing to manoeuvre around the maze and get past each key. To achieve a good score and get past the key the user needs to answer correctly questions about "Space"! The more right answers, the better the score!

Scratch

Questions	Answers
What is the closest planet to the sun?	Mercury
Which planet is home to the tallest mountain in the solar system?	Mars
Who was the first man on the moon?	Neil Armstrong
Rounded up, how many trillion miles is 1 light-year equal to?	6
How many moons does Pluto have?	5

3. Creating the foundations

Following your understanding from the previous workbook, ensure the following:

- Your main character goes back to a starting position when the green flag is clicked (x y)
- Each collectable is set to "show" when the green flag is pressed (Looks)
- A variable of "score" has been created (Data)
- The score variable is set to 0 each time the green flag has been pressed (Events)

You can incorporate some of the techniques you learnt from the previous Scratch workbook, in particular "conditional statements". In our quiz game, you will need to code the following:
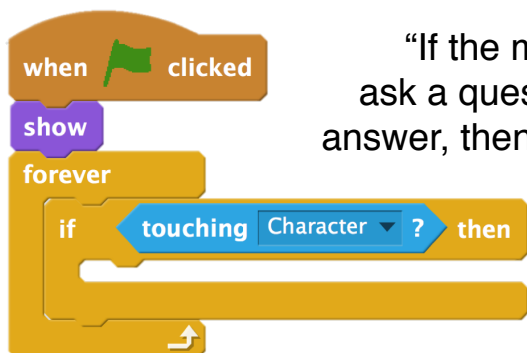
"If the main character collides with a collectable (a key) then ask a question and wait. If the answer typed in is the correct answer, then add points to the score and hide the key. The main character can now advance through the game."

Add a "Forever" and an "If then" statement, with the condition that if a sprite touches the main character, something will happen as a result.

Scratch

In the "Sensing" category, there is this code: `ask [What's your name?] and wait`

4. Asking a question

Using the above segment of code to ask a question. You can see within the sensing category there is also code called "answer". This allows you to implement a conditional statement, saying "If the *answer* entered is the same as...then *DO SOMETHING*". Your code should look like something similar to the left:

```
when [flag] clicked
show
forever
  if < touching [Character ▼] ? > then
    ask [Which planet is closest to the Sun?] and wait
    if < answer = [Mercury] > then
      change [Score ▼] by (5)
```

Hint: This code `< ▯ = ▯ >` can be found in the operators category.

You could also replace the embedded "IF" with an "IF ELSE" statement. This can be used to tell the game how to react if anything except the correct answer is entered. For instance:

IF "answer = Mercury" then
  change score by +5
ELSE
  change score by -2
  go to x = … y = …

5. IF and ELSE

Try implementing your own "IF ELSE" statements for each of your collectable sprites.

```
when [flag] clicked
show
forever
  if < touching [Character ▼] ? > then
    ask [Which planet is closest to the Sun?] and wait
    if < answer = [Mercury] > then
      change [Score ▼] by (5)
      hide
    else
      change [Score ▼] by (-2)
      hide
      broadcast [Go Back! ▼]
      wait (1) secs
      show
```

Hint: The code on the left includes a "broadcast" statement. When the main character receives a broadcasted message it will return to it's starting position for getting the answer to the question wrong.

Broadcasting is used in Scratch to send messages behind the scenes of a game/animation created, e.g. you can use it to signal sprites to appear/disappear (commonly used for "Game Over"). On the left we are using the broadcast to send a message called "Go Back!" (the name of the message is meaningless to Scratch). What is most important is what happens when the message is received.

```
when I receive [Go Back! ▼]
go to x: (210) y: (-147)
```

technocamps

3

Scratch

6. Collectables

Ensure a variety of collectables ask your chosen questions and the main character can only proceed each time if the answer is correct. Test your game, checking to see if your code does anything unexpected or doesn't quite work how you entered.

As Scratch doesn't feedback regarding any errors or mistakes in the code, keep an eye out for any code that looks out of place or if you feel it doesn't achieve what you had planned.

Score 10

Who was the first man on the moon?

Neil Armstrong
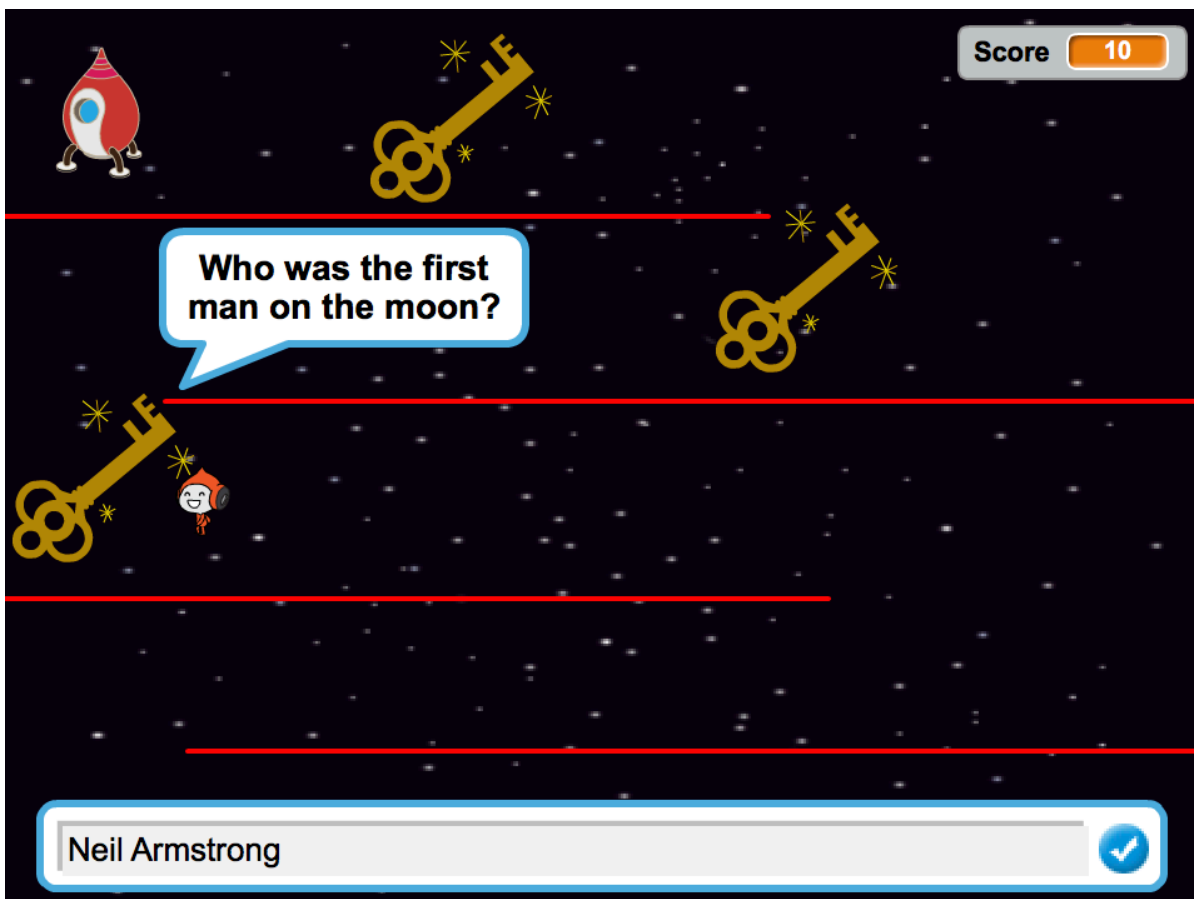
7. Game Over

What is going to happen when you reach the space ship at the end of the game? After practicing the use of conditional statements, add a statement for when the Main Character collides with the space ship.

Hint: You could also create a sprite that is just text "game over. With the use of broadcasting you can hide and show the sprite when the Main Character collides with the space ship.

technoteach

Scratch

Typing each question individually for each sprite can be time-consuming. It would be easier to have a list of questions and answers saved somewhere in the game that you can quickly add/edit and call throughout the game. To create a list on Scratch, click on the variables category and select "Make a list". Make sure that "for all sprites" is selected.
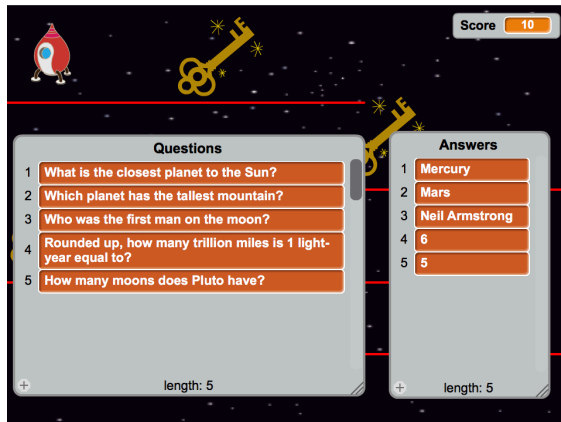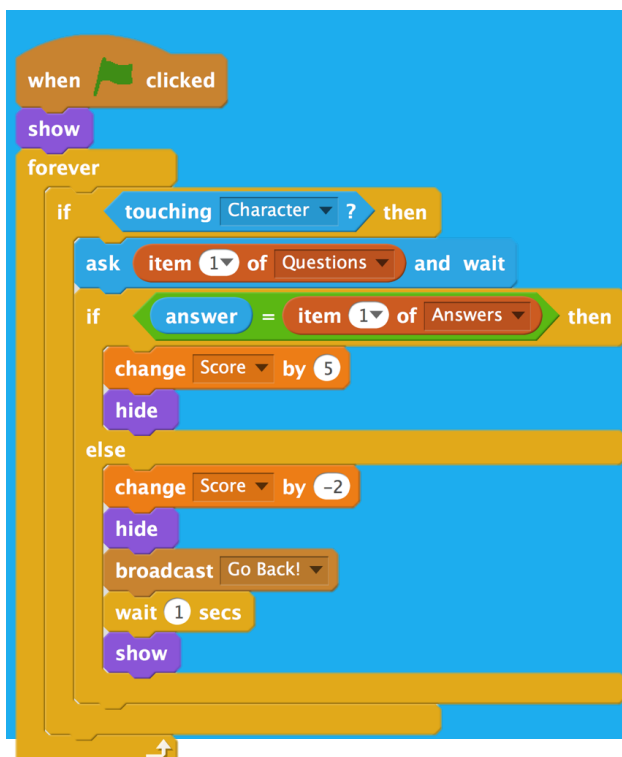

8. Lists

Create 2 lists on your game, one for your questions (these can be added using the + button) and one for the correct answers.

At the moment your lists are in the middle of the screen, covering your game. You can choose to show/hide the lists, so they exist in the game but cannot be seen. This can be done by unchecking the tick boxes on the left in the Data category.

You can specify which item to refer to within a list, for instance item 1 from the questions array is "What is the closest planet to the sun?" and item 3 from the answers array is "Neil Armstrong". Similar to how it has been implemented previously, you can refer to the items within a list and compare the entered answers to the specified items. The code to use to implement the use of lists can also be found within the data category

9. Application

Apply this method of using lists (arrays) to all of your sprites, try editing some of the items in the lists. There is a way to call them at random.

What is the problem here?

What if you just wanted to open the game without having to add any questions?

What if you wanted the game to call out the items from the list at random?

What if once you had got a question right you didn't want to have to answer it again?

Scratch

In the operators category, you can see a piece of code called "Pick random 1 to 10", this can be manipulated to search randomly for a number between 1 and the length of the list (e.g. if there are 6 items in the list, the length will be 6). Firstly you need to create a new variable, called random_number, this variable will be the random number that is picked out between 1 and the length of the list.   Now that you have the "random_number" variable ready to use, you can apply this number to the item numbers within each list. E.g If the random_number = 3, then it will output the question

`ask item random of Questions ▾ and wait`

"Who was the first man on the moon?"
when added to the statement:

10.Random questions

Now that you are picking an item from our list at random, you need to compare this item against the user input and the item within the "answers" array you have previously created. To do this, as before you create a conditional statement (IF) comparing answer and the item (random_number) from within the answers array.

Next, you need to ensure that if your user answers a question correctly that the question is no longer in the list so can not be repeated throughout the game. To do this you can add "delete *random_number* from the answers array" and "from the questions array also".

This can be applied to each collectable sprite.

```
when [green flag] clicked
show
forever
  if < touching Character ▾ ? > then
    set random ▾ to (pick random 1 to length of Answers ▾)
    ask (item random of Questions ▾) and wait
    if < answer = item random of Answers ▾ > then
      change Score ▾ by 5
      hide
      delete random of Questions ▾
      delete random of Answers ▾
    else
      change Score ▾ by -2
      hide
      broadcast Go Back! ▾
      wait 1 secs
      show
```

11. Resetting the game

You can also add code to your game populating both of the arrays whenever the green flag is pressed, resetting both the game and the arrays. As the user is progressing through the game, if a question is answered correctly it is removed from the array, so repopulating the array is important. To repopulate the array, begin the script (in any sprite's code) "When *green flag* is clicked" and repeat the following code, adding your items to each array:

```
when [green flag] clicked
delete all ▾ of Questions ▾
insert What is the closest planet to the Sun? at 1 ▾ of Questions ▾
insert Which planet has the tallest mountain? at 2 ▾ of Questions ▾
```

```
when [green flag] clicked
delete all ▾ of Answers ▾
insert Mercury at 1 ▾ of Answers ▾
insert Mars at 2 ▾ of Answers ▾
```

technoteach

@Technocamps

Find us on **Facebook**