technocamps

Inspiring | Creative | Fun
Ysbrydoledig | Creadigol | Hwyl

Scratch:
Eye of the Storm

Aims

So far, you have already learned some of the basics of Scratch and programming. We will revisit these skills and apply them in today's workshop.

By the end of today, you will:

- Make a new fun and interactive game.
- Improve your skills in Scratch.
- Raise your awareness about renewable energy.

Overview

A tornado was created for a renewable energy project but something has gone wrong! It is picking up lots of renewable energy items which is making it go out of control. It is your job to remove these items and stop the tornado!!!

This will be achieved by correctly answering questions about renewable energy.

Every question answered will remove an item from the tornado.

When all the items are cleaned up, you win the game!

Make sure you remember to test your game after each change you make.

Keywords

Sprites – They are the characters in our game that contain blocks of code in order to interact with one another.

Stage – It is the backdrop for our game, it can also contain blocks.

Blocks – They are the building blocks or code of Scratch which fit together like pieces of a jigsaw puzzle.

Sequence- The process of running your code in order.

Variables – They are used to store information and their values can be changed.

Loops – It allows sections of blocks to be repeated.

IF Statements – They check if a condition is met and executes blocks based on whether the condition is true or false.

Questions & Answers

Question	Answer

Basics of Scratch

Motion

Motion blocks deal with the movement of sprites.

Looks

Looks blocks are related to the appearance of sprites and the stage.

Data

Data blocks contain variables. Variables are used to store information such as the answers to our questions.

Events

Event blocks trigger things to happen in our game. For example, if the green flag is clicked, the game will start.

Control

Control blocks organise how the program will run. This is done through loops (repeating code multiple times) and IF statements.

Sensing

Sensing blocks are used to detect when a condition has been met. An example is checking whether a sprite is touching another sprite.

Operators

Operator blocks consist of many mathematical functions such as addition and checking if something is equal.

Task 1: The TORNADO

Firstly, we need to add our tornado to the screen. The sprite for this will be provided to you from the Technocamps website. If the sprite is not available, why not make your own!

Once added, we now need to make our tornado repeatedly move between two points. We need to use coordinates for this. The centre of the game is (0, 0) meaning x = 0 and y = 0.

Hints:

The Events section has a block that will cause something to happen when the game is started.

The Motion section will help you find a block that moves the tornado to specific coordinates.

The Control section has a block that will help you repeat the movement forever.

y

(0,0)

X
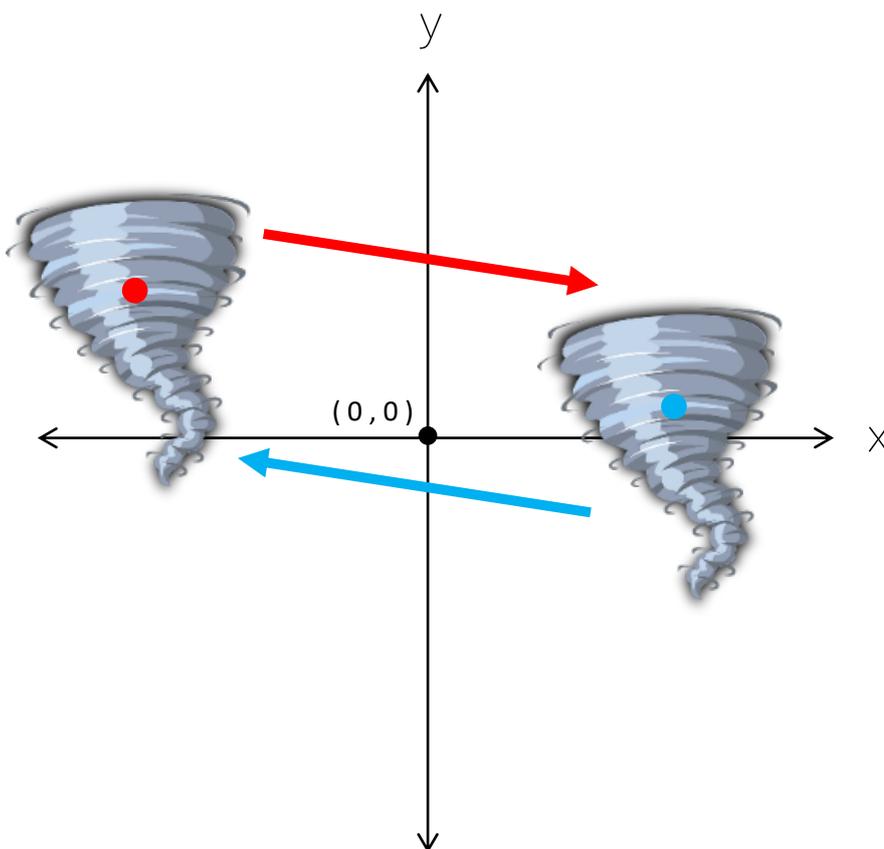
Note: We can get the current coordinate of a sprite by hovering your mouse over the stage. They will appear in the bottom right of the stage window.

Co-ordinates

● Start = (___ , ___)

● End = (___ , ___)

Task 2: Renewable Energy

This is similar to Task 1; we are going to add renewable energy items to the screen using more images from the Technocamps website. Like Task 1, We also want to move them between two points but it's going to be a bit harder this time. To make our game look more realistic, we need to make the items go behind the tornado at certain points and in front of it at others.

Hints:

Use the **Motion** section to be able to move to a certain coordinate.

Use the **Looks** section to be able to make the items go behind the tornado and appear in front again.

You will need to write the code in the **Sprite** itself.

Behind
In front

y

A

A

(0,0)

X
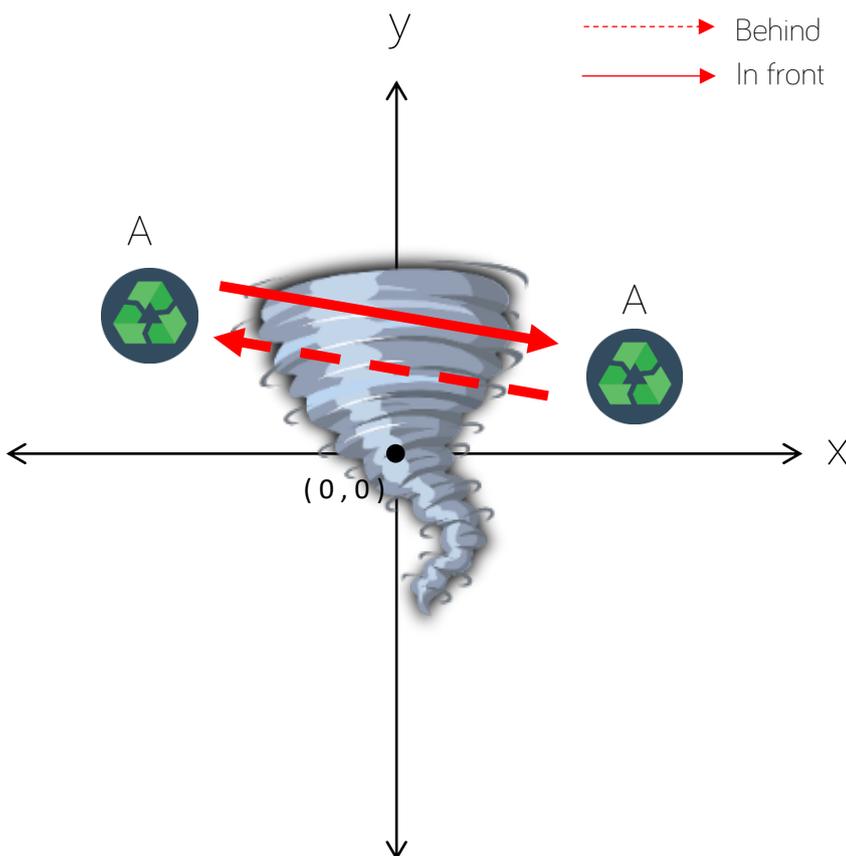
Co-ordinates

A   Start = (__ , __)
     End = (__ , __)

B   Start = (__ , __)
     End = (__ , __)

C   Start = (__ , __)
     End = (__ , __)

D   Start = (__ , __)
     End = (__ , __)

E   Start = (__ , __)
     End = (__ , __)

Task 3: Asking Questions

Using the questions we created at the start, you are going to add a question to each item so you will be writing code in the same place as Task 2. When an item has been clicked we want to ask a question then check if it is the correct answer. If it is the correct answer we want to hide the item.

Hints:

You will need to use the Sensing section to ask questions and receive answers.

You will need an IF statement for each question to check if it is correct.

You will need to stop the movement code for the item that is asking the question.

If the answer is correct you need to hide the item.

Fill in the Blanks!

Sentences

- If answer is incorrect- continue the program
- Hide the item and stop all other actions
- Say Correct!
- Check if answer is correct
- Ask a question based on renewable energy

When the renewable energy item is clicked:

- _____
- _____
- _____
- _____
- _____

Try running through this **sequence** of instructions with a partner.

Task 4: Winning the game

Now we have our game we want to add score to see how good you are at the game. If we have the score we can also check if the player wins or not, by answering all the questions correctly.

You will need to create a variable for all sprites to keep track of the score. When a question has been answered correctly we want to add 1 to the score. Inside your tornado, you will need to check if the score is equal to the amount of questions you have in order to see if the player has won the game.

Hints:

The data section will be used a lot as we have to create the score variable, set the score and change it by 1.

Fill in the Blanks!

Correct!

Incorrect!

Score
0 ⟶ **Score**
___
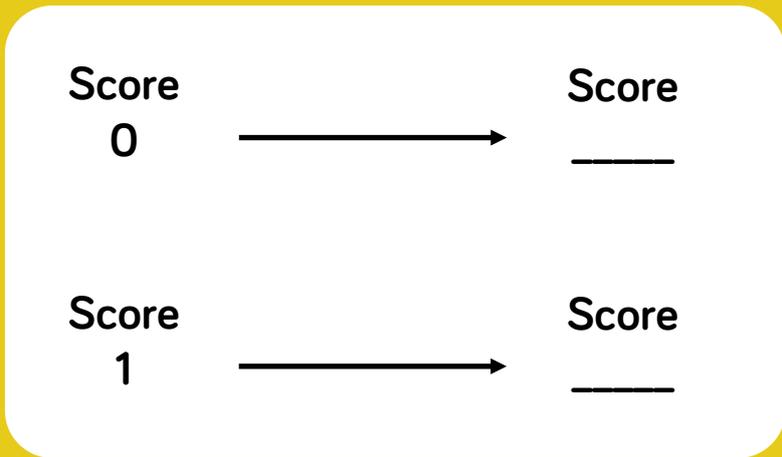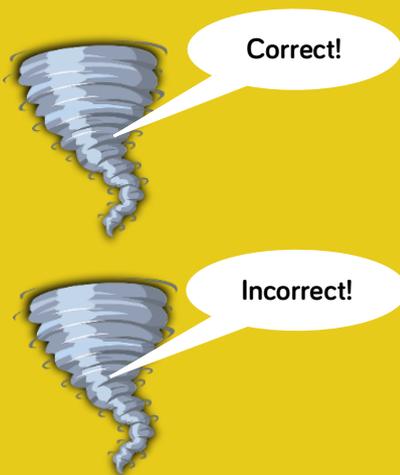
Score
1 ⟶ **Score**
___

When a question is answered correctly we increase the score by _____.

When the score hits _____ then

- Stop all scripts.
- Show "You have Won!".

Task 5: Making Items Move Faster

If you have made it this far, well done! You have a completed game but you can still make it better. The first challenge is to make your items move faster. This may sound like an easy task but it requires some thinking. You don't need any extra blocks; it only requires changing an existing block.

Task 6: Adding Sounds to the Game

The second challenge requires adding some sounds to your game at an appropriate time. We suggest that your game should make a sound when a question is answered correctly. However, you can make your game unique and play sounds at a different time. You will need to use a new block category to achieve this.

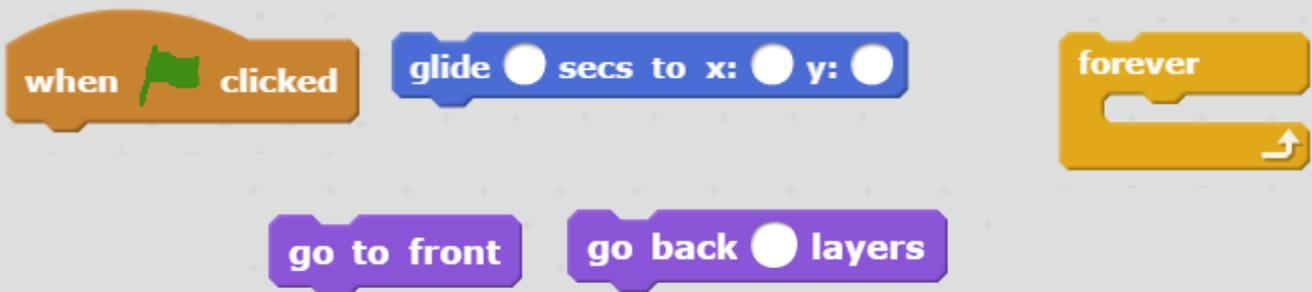## Task 7: Multiple Answers to Questions

The final challenge is changing your questions so that they have more than one answer. You could possibly change "Is coal a type of fossil fuel" to "Name a type of fossil fuel". In order to achieve this, you will need to have multiple IF Statements for your questions.

If you've managed to complete all the tasks- Well done! You are a great scratch programmer!
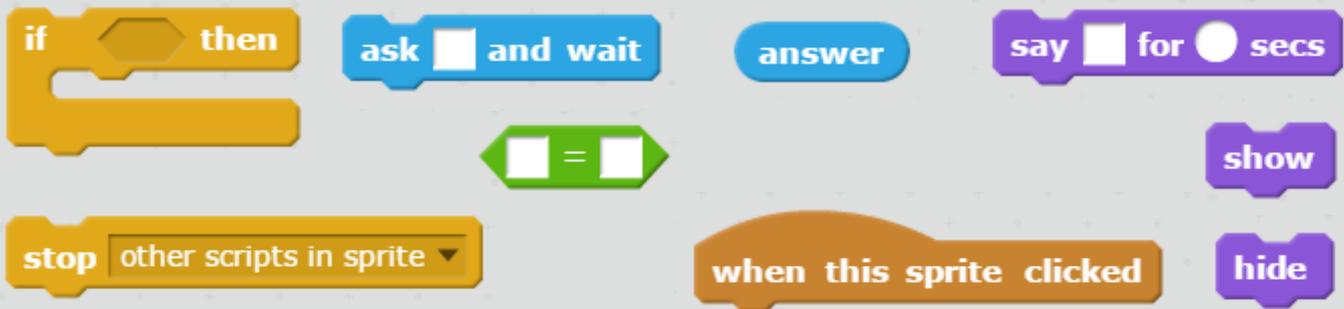
Task 1

when [🚩] clicked

glide ◯ secs to x: ◯ y: ◯

forever

Task 2

when [🚩] clicked

glide ◯ secs to x: ◯ y: ◯

forever

go to front

go back ◯ layers

Task 3

if ⬡ then

ask ☐ and wait

answer

say ☐ for ◯ secs

☐ = ☐

show

stop other scripts in sprite ▾

when this sprite clicked

hide

Task 4

stop all ▾

☐ = ☐

score

if ⬡ then

say ☐ for ◯ secs

change score ▾ by ◯