

The logo features a blue computer mouse with a cord, positioned between two yellow arrows. One arrow points upwards and the other points to the right.

# Game Control



## Student Workbook



Name:

Class:

## Assessment



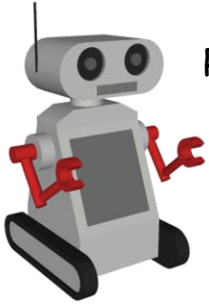
### Self Assessment

Date	Learning Objective(s)	Achievement 😊 or 😞

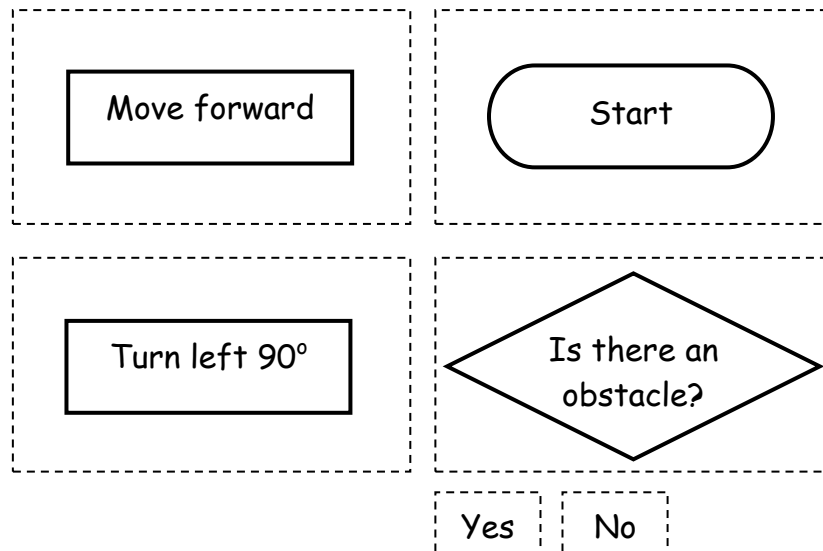
### Teacher Assessment

	NC Level: <input style="width: 80%;" type="text"/>
--	--

## Lesson 1 - Cutouts

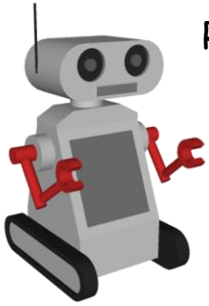


R.O.B.B.O the Robot is not working properly as all his instructions have been mixed up! Cut out the flowchart symbols below around the dotted lines ----- and stick them down **in order** on the next page to make Robbie work properly.



This page is intentionally left blank.

## Lesson 1 - Task 1



R.O.B.B.O should work like this:

- Continuously move forward
- Stop moving and turn left  $90^\circ$  when he meets an obstacle then continue moving again

Stick the cut-out instructions in the spaces below in the correct order.



Now **join the instructions together** by drawing **arrows** in the correct places.

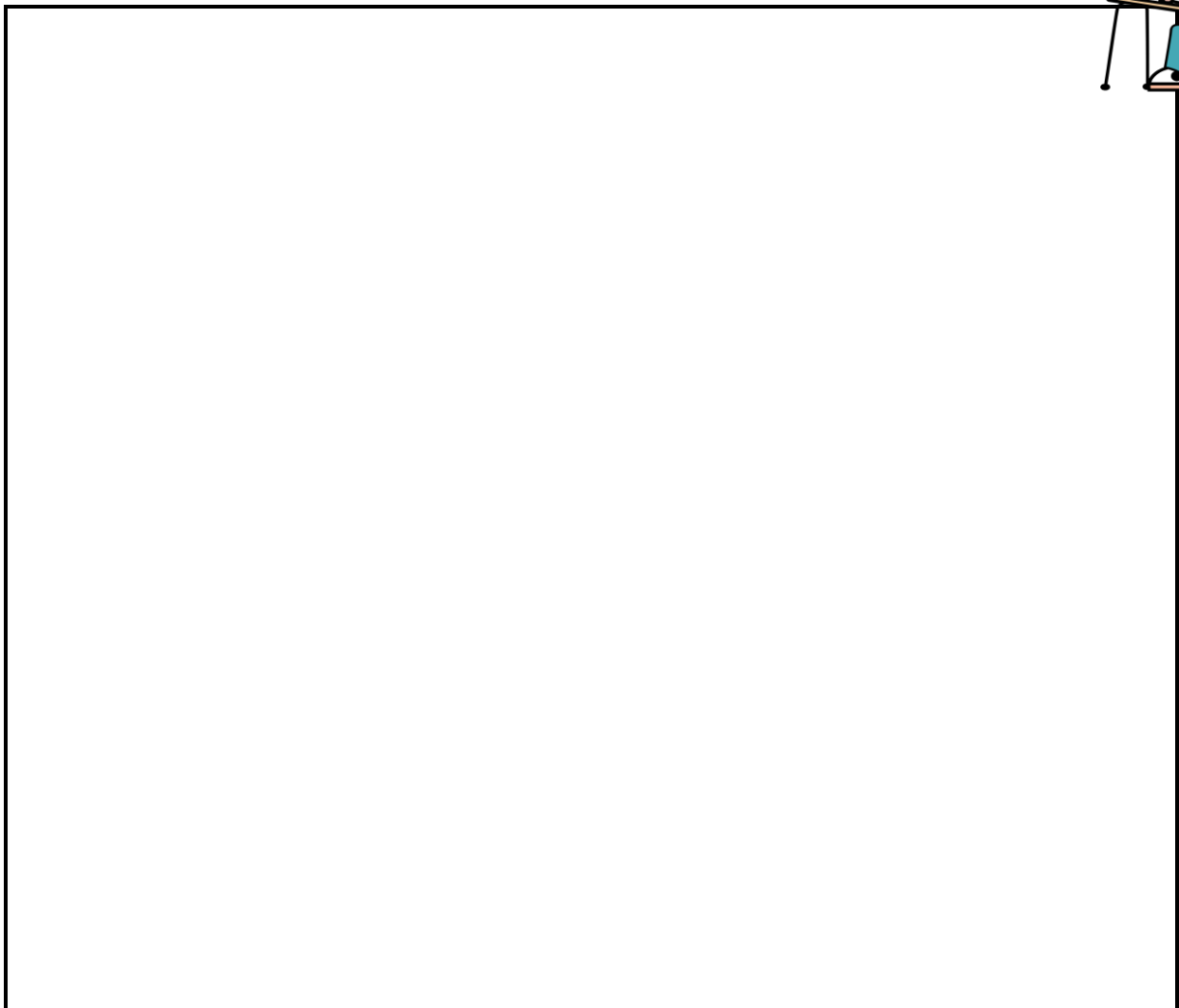



## Lesson 1 - Task 2

How do you make a cup of coffee? These are the instructions of how to make a cup of black coffee:

1. Turn kettle on
2. Add coffee to cup
3. Add sugar to cup
4. Check if kettle has boiled
5. Pour water into cup
6. Stir and enjoy!

Draw a flowchart below to represent these instructions:



## Lesson 2 - Starter

Test what you have learnt so far by filling-in the blanks using the words below:

symbols

input/output

sequence

process

decision

flowcharts

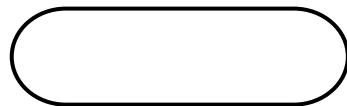
arrows

start/stop

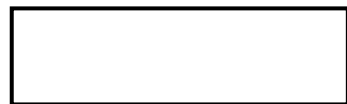
Instructions for computers can be summarised in  
\_\_\_\_\_, using special \_\_\_\_\_ and language.

The instructions in flowcharts are displayed in  
\_\_\_\_\_, this sequence is shown by the direction of the  
\_\_\_\_\_ connecting the symbols.

The pictures below show the symbols used in flowcharts, label them:



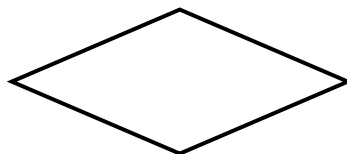
\_\_\_\_\_



\_\_\_\_\_



\_\_\_\_\_

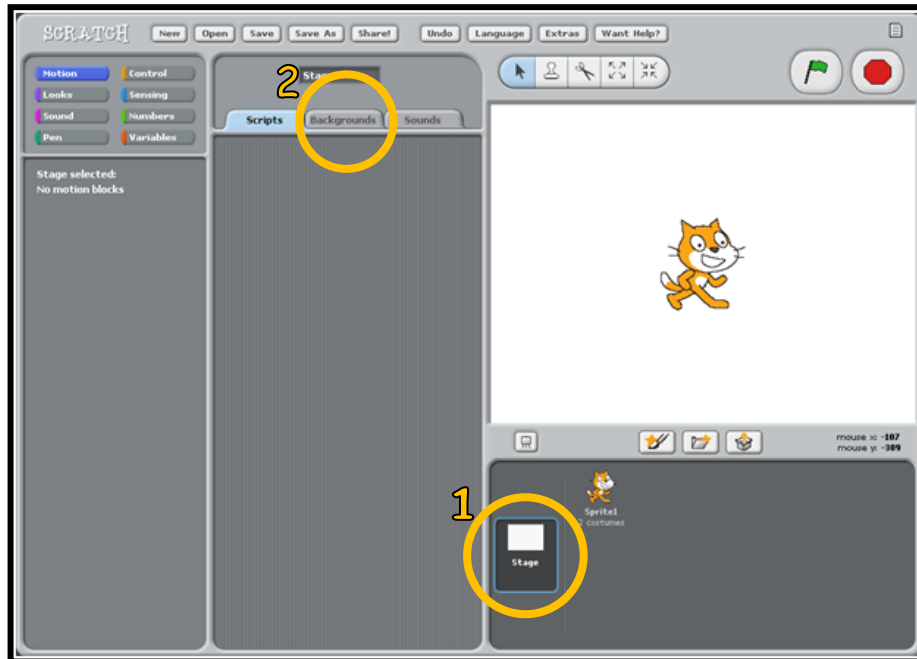


\_\_\_\_\_

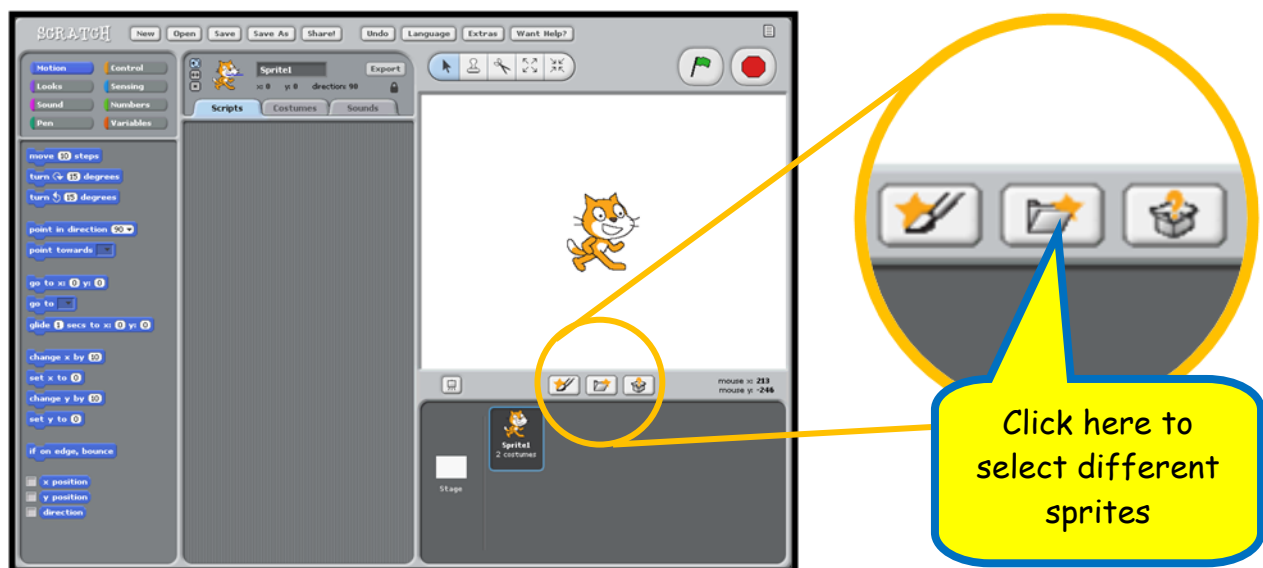
## Lesson 2 - Introduction to SCRATCH

### Adding a Stage Background

First (1) Select the **Stage** in the bottom right. Next (2) Select the **Backgrounds** tab and **Import** a different background for the stage.



### Selecting New Sprites



## Changing Direction and Moving

Select the drop-down arrow to choose which direction the sprite should face:

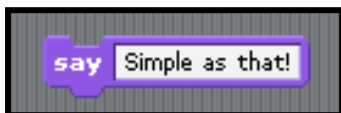


Select the move instruction and type in the number of steps forward to move:



## Talking Sprites

You can make sprites talk by using the "Say" instruction:



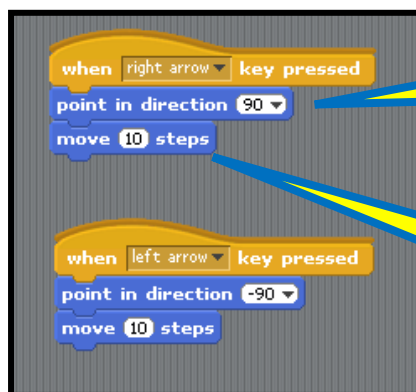
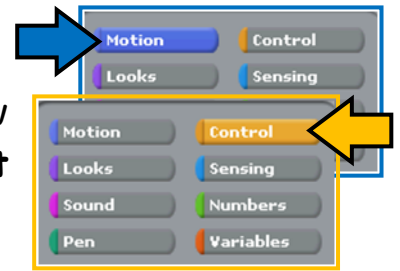
You can also make your sprite talk for a set length of time like this:



## Lesson 3 - Loopy Motion Control

### Movement with the Cursor Keys

You can use any keys to control the sprite, below shows it being controlled using the **left** and **right arrow keys** on the keyboard:



This instruction tells the sprite what direction to face

This states how many steps forward to move at once

### Movement in a Loop

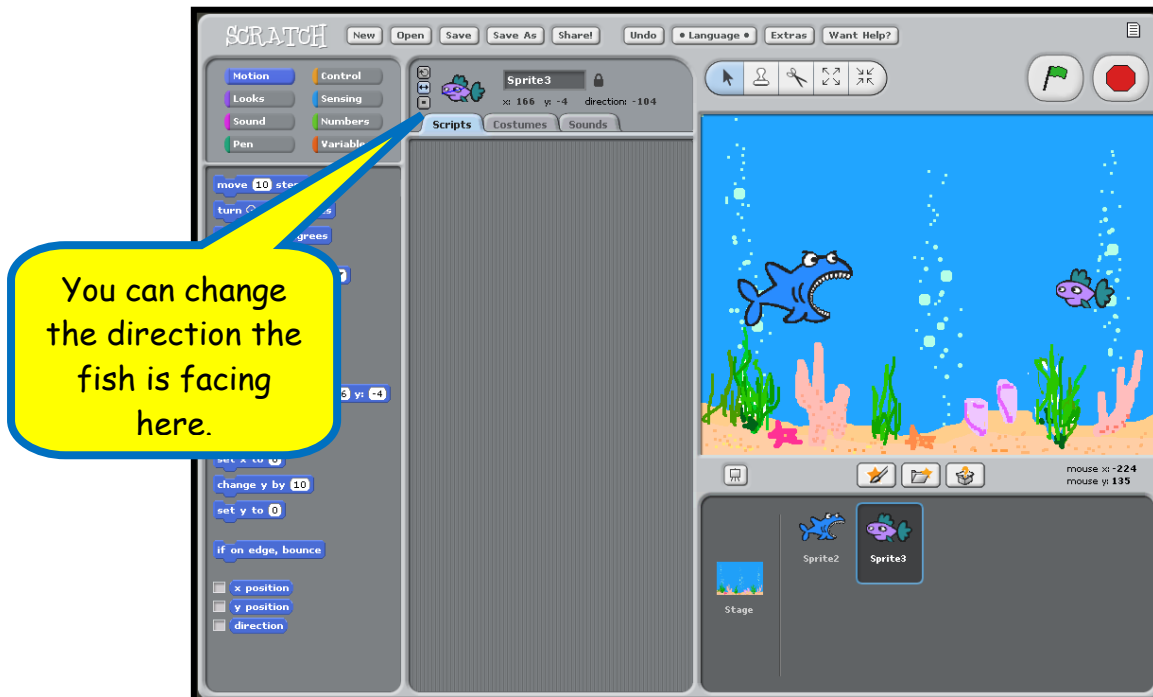
Using a **loop** is a clever way to make the sprite **repeat** instructions over and over again. The instructions below will make the sprite follow the mouse cursor continuously once you've pressed the green flag (🚩) to start.



## Lesson 4 - Shark Attack Game

### The Stage and Sprites

Select a suitable background for the stage (underwater would be a good idea). Then add a shark and a fish to the game screen so it looks like this:



I've done this (✓ when done): ☐

### Making the Shark Move

Add these scripts for the shark to make it move up and down when you press the arrow keys. Make sure you double-click on the shark first.



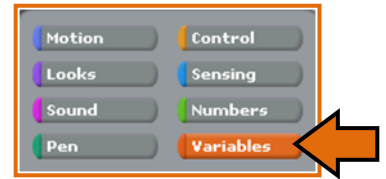
Y is the **vertical**(↑↓) position of a sprite on the screen. X is the **horizontal**(←→) position.

I've done this (✓ when done): ☐

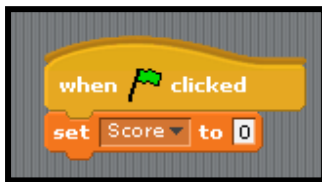
# Game Control

## Adding a Score Variable

A variable is what is used to **store** some data. In this case the **score**.



**Make a variable** and call it **Score**.



We then need to make sure that when we start a new game the score is reset to zero.

---

I've done this (✓ when done): ☐

## The Loopy Bit

Now we are going to make the fish keep appearing in different positions and move across the screen. To do this we need to use a **Loop**. Add this script to the fish sprite (double-click) on the fish.



The loop won't do anything yet because there's nothing inside it.

---

I've done this (✓ when done): ☐

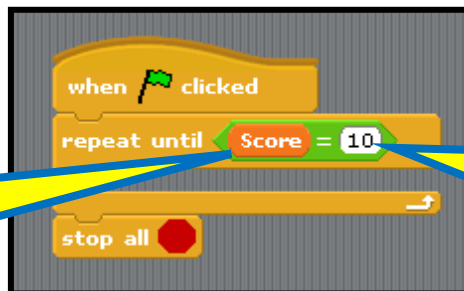
# Game Control

## Using Numbers

We're now going to add a number into our loop so when the score reaches ten the game will stop. Add these bits to your loop script:



This is the **Score** variable you created in the last task.



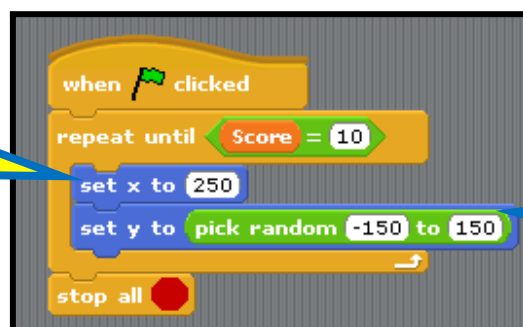
Drag and drop the numbers control into the loop and type "10" into the end box.

I've done this (✓ when done): ☐

## Setting the Fish's Position

We are now going to make it so that when the game starts the fish is just off the screen on the right-hand side (X position = 250) and it starts in a random position up or down the screen (Y position from -150 to +150). Add this script to the loop on the fish sprite:

250 means the fish will be just off the screen.



This picks a **random** position between -150 and +150

I've done this (✓ when done): ☐

## Naming the Sprites

Now would be a good time to give our sprites names. This means we can identify them later within our scripts.



Rename the  
sprites to:  
"Shark" and  
"Fish".

I've done this (✓ when done): ☐

## Making the Fish Move

We now need to add another loop **inside** our current one to make the fish keep moving. Like this:

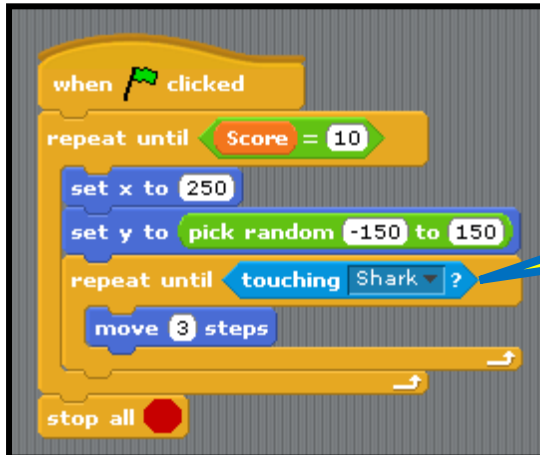
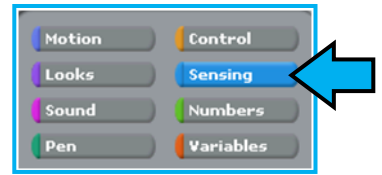


I've done this (✓ when done): ☐

# Game Control

## Eating the Fish

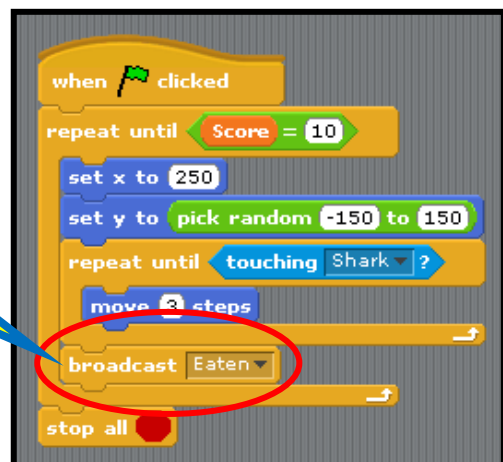
Now we need to **sense** whether the shark has caught the fish.



The fish will keep moving forward until it gets caught by the shark.

The fish now needs to "tell" the shark that it's been "eaten". We are going to use the **broadcast** command to do this here. This basically sends a message to the other sprites in the game (in this case there is only one).

Make sure you put the **broadcast** command **after** the second loop but still **inside** the first one.



I've done this (✓ when done): ☐

## Last Bit Now

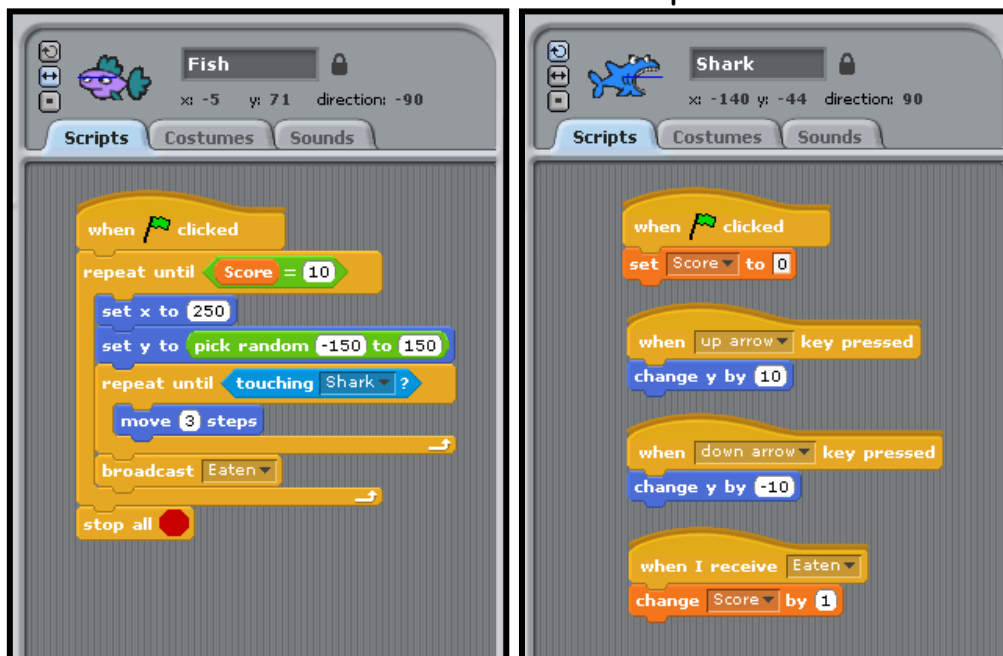
The only thing left to do now is increase the score every time the fish is "eaten". Add a final script to the **shark sprite** to increase the score when it receives the fish's broadcast of "eaten".



I've done this (✓ when done): ☐



Check it's all correct. This is what both scripts should look like:



## Extension Task

Make your game better by adding the following:

- Change the sharks costume so it closes it's mouth to eat the fish.
- Add a **GAME OVER** stage to be displayed if the fish doesn't get eaten.

## Lesson 5 - A Virtual Pet



Today you are going to make your own virtual pet. These are the things your game will need:

1. A variable (starting at 0) to store **Hunger**. This will go **up** every 30 seconds.
2. If **Hunger** gets up to 10 the pet will die.

You can copy these first scripts to get you started:



To feed your pet you'll need to add some food which you can click on to feed it. This will reset hunger back to zero.



### Extension Task

Try adding these features:

1. Aging - make your pet get bigger as time passes.
2. Make the pet change into a ghost and float off the screen when it dies.
3. Add a variable for health which will go down over time and if the pet's not fed often enough.

## Lesson 6 - Finishing the Virtual Pet

You now need to make your virtual pet into a real game! Try to add some of the following features on your own:

1. **Health**

This should go down gradually over time but could be boosted by giving him medicine.

2. **Happiness**

This should go down gradually but can be boosted by clicking on your pet (giving him attention).

3. **Appearance**

Change the pets costume at different times:

- a. Make the pet go green when he gets ill
- b. Change the pets facial expression between happy and sad.

4. **Intelligence**

Add a variable for intelligence which you can boost by clicking a picture of a book.

Can you think of any other features to add?

There will be a prize for the best and most imaginative virtual pet in the class!



## Lesson 6 - Peer Assessment



THIS PAGE MUST BE COMPLETED BY ANOTHER MEMBER OF THE CLASS!

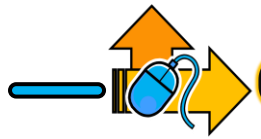
Assessor Name:

Give the following a rating (circle ☹ or 😐 or 😊):

Quality of game play	☹	😐	😊
Graphics used (pictures)	☹	😐	😊
Ease of use	☹	😐	😊
How well does it work?	☹	😐	😊

What else should be included to make the game better?

Which parts of the game did you like best?



## Lesson 7 - Self Assessment



Complete the following statements honestly.

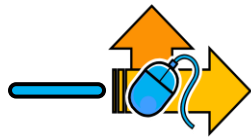
1. The parts of this topic I have enjoyed the most are . . .

2. The bits I have not enjoyed are . . .

3. I have downloaded and used Scratch at home:

☐ Yes      ☐ No

*If no, why not?*



# Game Control

4. I would rate the effort I put into this unit as:

- |  |  |
|--|--|
| <input type="checkbox"/> A - Excellent                 | <input type="checkbox"/> B - Good              |
| <input type="checkbox"/> C - Satisfactory              | <input type="checkbox"/> D - Cause for concern |
| <input type="checkbox"/> E - Serious cause for concern |  |

*Why?*



## National Curriculum Levels

I can do the following (tick the ones that apply):

Level 3:

- ☐ Use sequences of instructions to control a sprite in Scratch

Level 4:

- ☐ Create my own scripts in Scratch
- ☐ Change scripts to make them better
- ☐ Use inputs from the keyboard or mouse to control a sprite
- ☐ Display an output on the screen (using variables)

Level 5:

- ☐ Make my scripts work because I understand the need to be precise when sequencing instructions
- ☐ Change variables in a game and know what will happen as a result

I would grade myself with a National Curriculum level of:

## Want to Practice at Home?



It is **free** to download Scratch so you can use it at home by going to this website:

<http://scratch.mit.edu>

The Scratch website also has games to play made by others and you can even upload your own games for other people to play!



## Want a BIGGER challenge?

There are lots of other game making packages you can download that are also free (and more complicated than Scratch). Here are some examples:



[www.alice.org](http://www.alice.org)



[www.greenfoot.org](http://www.greenfoot.org)



[www.yoyogames.com/make](http://www.yoyogames.com/make)



