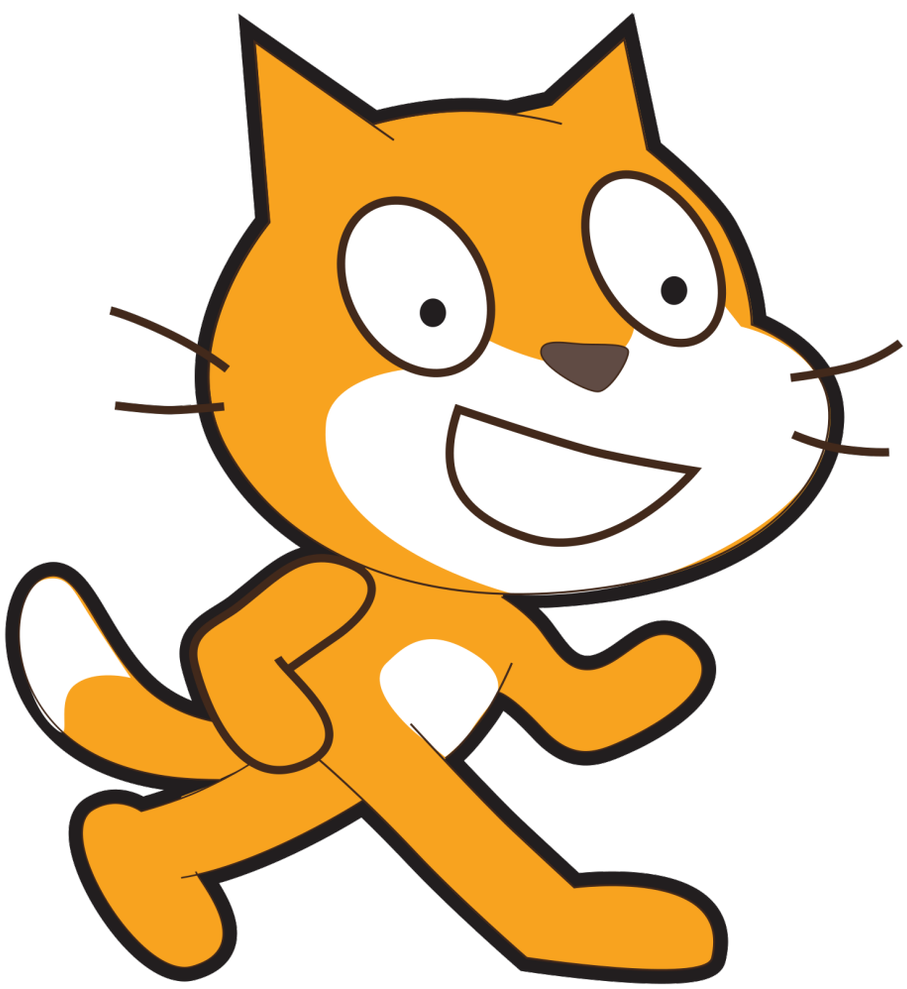
technoteach

technocamps

Scratch for Beginners Workbook



Swansea University
Prifysgol Abertawe

Cardiff University
Prifysgol Caerdydd

Prifysgol Bangor University

Cardiff Metropolitan University | Prifysgol Metropolitan Caerdydd

i.t.wales
www.itwales.com

Prifysgol Aberystwyth University

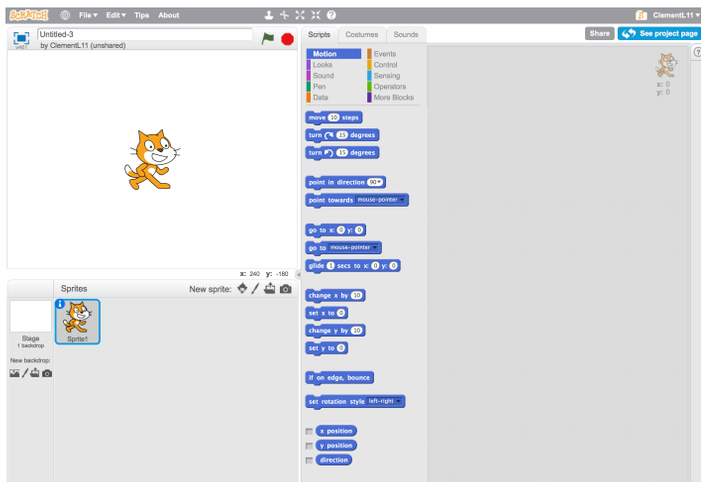
Prifysgol Glyndŵr University

University of South Wales
Prifysgol De Cymru

Ariennir yn Rhannol gan **Lywodraeth Cymru**
Part Funded by **Welsh Government**

Scratch

In this workshop you will be using a software called Scratch, a drag-and-drop style software you can use to build your own games. 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.



Let's have a look at the Scratch software - you can open it by visiting https://scratch.mit.edu using Google Chrome (Safari can have some bugs) or by finding the offline editor in Applications. You may have a shortcut already available on your desktop.

Scratch has 5 main sections, firstly starting from the centre of the screen you have a list of categories to choose from. As you select each one you can see in the column below different instructions appear (colour co-ordinated), these can be used to create your game by dragging them across to the right where the scripts can be used for your game. On the left of the screen you can see the white box with the Scratch character inside, this is your game. As you begin developing your game, you can play and stop it using the green flag and red stop-sign buttons above. There is also one last section, this is your library below your game. As you begin adding Sprites (another name for characters) you can see them all being stored in your game library along the bottom; you can select these to bring up their scripts and make changes.

1. Adding Sprites

You can add new sprites (characters) into the game in 4 different ways, using the 4 buttons just below your game library.

The first enables you to import a ready-made character from a range of options such as animals, fantasy and transportation.

The second one opens up a paint-style editor, allowing you to draw and create your own character.

The third button, once selected, will allow you to upload a picture to use as a sprite.

The fourth allows you to use a camera on the device to take a picture to use as a sprite.

Delete your Scratch the Cat by right-clicking and selecting 'delete'. Try adding in your own characters, or you could even draw your own!

Scratch

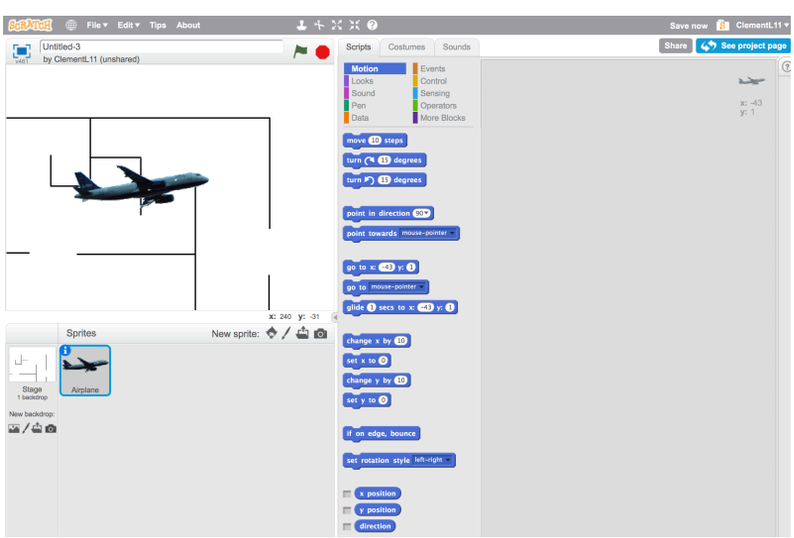
Now that you have your characters we can also change the background of our game to something a bit more exciting than a white background!
If you click on ‘Stage’ within your library, you can see a blue line around the icon, this means it is selected properly - always check which sprite is selected to ensure you are giving code to the right sprite. Now above where you add scripts you can see 3 tabs: Scripts, Backgrounds, Sounds. Click on ‘Backgrounds’ and you can either Paint, Import or take a picture of a new background to use on your game.

2. Maze Background

To create a game similar to Pac Man with a maze to move around, you need to draw one on our Background. If you click ‘Paint’ you can use the line tool to draw up your own maze.

Hint: If you hold down the SHIFT key on your keyboard as you draw your maze, you can get perfect straight lines.

Once you’ve drawn your maze, click OK.



What’s the problem with the picture on the left?

Have a look at the buttons.

You can use the 2 on the right to change the size of your sprite to make it fit around your maze without touching any walls!

When you move you do it sometimes without thinking, especially walking and running. But our brains are telling you to firstly face the direction you wish to move and to then start walking in that direction. That is the same structure of instructions you need to give our character to get it to move.
Have a look on the motion category (dark blue) and you can see all the different types of instructions you can use to make your character move properly. Can you see any which you can use to make it move the way you do? How can you get your characters to move, using which keys on your keyboards?

Scratch

3. Movement

Use the commands in Motion and Control to get your characters to move.

Firstly drag on the command “When ‘space’ key is pressed”, you can change ‘space’ to a key of your choice, e.g. right arrow key. Now underneath you can put instructions of what will happen when that particular key is pressed.

In Motion, use 

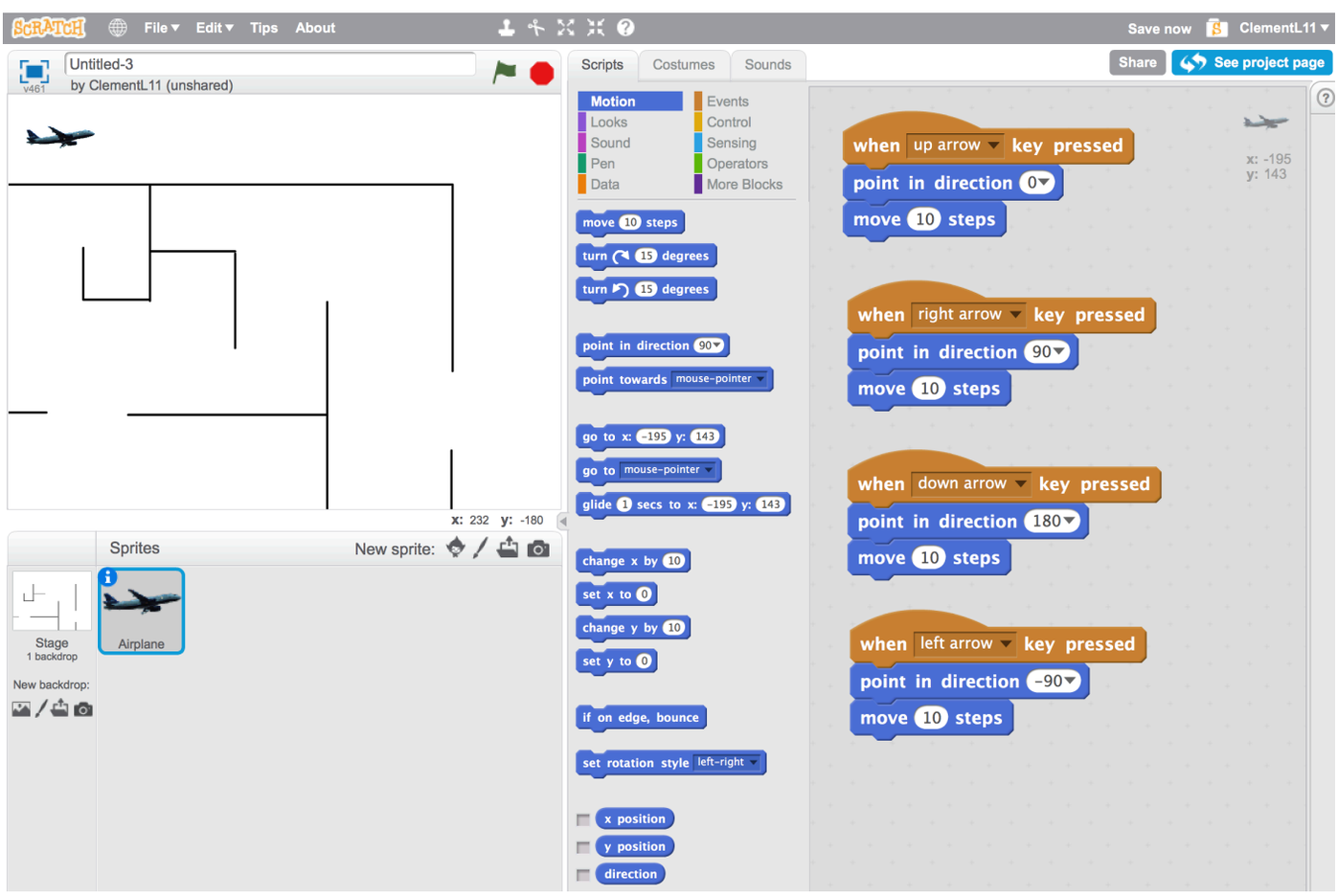
and 

to start getting your character to move. You can click on the drop down menus and change the script.

Remember: You must point in the direction you want to go before moving.

Once you’ve done one command, you can right-click and select ‘duplicate‘. This will copy the selected code, don’t forget to change some of the commands.

Make sure you have told your character how to move up, down, left and right.



Don't forget to keep testing your game to see if it works properly!

So far, you have a character, an obstacle (the maze) and you have also successfully implemented movement into your game using the keys on your keyboards.
The only problem with your obstacle is that your character does not know yet that it cannot go through the lines of the maze. At the moment your character should be able to move anywhere on the screen. There are 2 different ways you could approach this issue:

- You could get your character to bounce off the walls of the maze by telling the character to move back 2 steps when it comes in contact with the wall, creating the bouncing effect.
- You could return your character to the beginning when it comes in contact with the wall, but first you need to tell it where that is.

4. Be careful you don't touch the walls of the maze!

See if you can use the "Forever" and "If Then" commands in the 'Control' category. First you need to use 'When *green flag* is clicked'. The Green flag button is your reset button, so the command means 'When I reset my game...'. Underneath you can add "Forever" and an "If Then" inside the Forever block.

Now it says 'When I reset my game, if *whatever we put in the diamond shaped box* happens, then do...'
Go into the 'Sensing' category, here we can select 'touching colour' and add it to your "If Then" command. You can change the colour in the box by clicking it and using the eye dropped clicking on a wall of your maze.

Now it says 'When I reset my game, if I touch the selected colour then do...'. Depending on how you want your game to work here is where you can either enter 'move -2 steps' or 'go to x: y:' (choosing which x and y co-ordinates to use).

You can also use the code 'When *green flag* is clicked' followed by setting 'go to x: y:' and setting the x and y co-ordinates. This would be your starting position every time the green flag is clicked (your game is reset).

5. Adding an Enemy

Try adding a new sprite, just as you did with the first one. This one can be used as an enemy of your first sprite. Ensure you give it a starting position:

'When *green flag* is clicked...go to x: y:'

In the ‘Sensing‘ category, you previously used ‘touching colour‘ this time you can use ‘touching *drop down menu allowing user to select a sprite*’. Here you can give your main sprite instructions, that if it touches your enemy, it must go back to the beginning. You have already done the same code previously, but using a different type of sensing command.

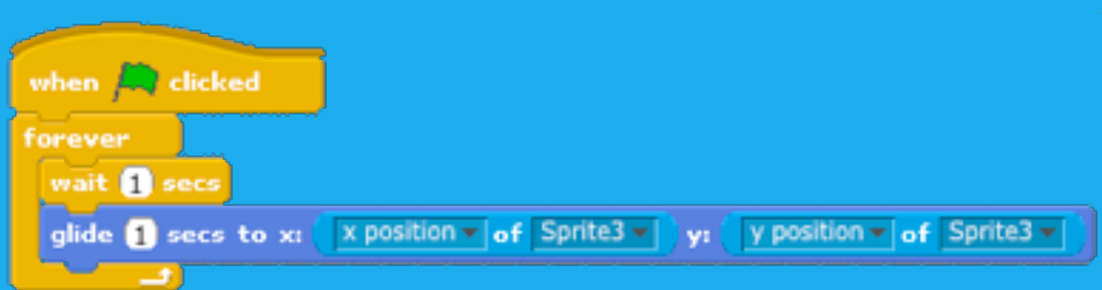
6. Enemy movement

You can either make your enemy chase your character, create a 2 player game (using W, A, S and D keys instead of the arrows) or get your enemy moving back and forth in a loop between 2 sets of x and y co-ordinates.

To chase:

Here you will be using another ‘When *green flag* is clicked’ followed by a ‘forever’ loop. Whatever you put inside the ‘forever’ loop will be completed repeatedly as the game is running. You need to tell our enemy to ‘glide 1 secs to x: y:’ but instead of entering a co-ordinate, you can go onto ‘Sensing’ and use ‘x position of SpriteX’. You can change the amount of seconds and speed it takes to get to that location by altering the glide for x secs.

If you leave the code like that, the enemy is going to go straight to the position of your sprite, so you need to add a ‘wait for x secs’ which can be found in the ‘Control’ category.



To move back and forth:

As above you need to use ‘wait’ to break up your commands for your enemy, so to move to one position, wait, move back to previous position, wait, etc...you need to use a combination of the ‘wait’ command and ‘go to x: y:’ command inside a forever loop.

2 Player Game:

With a 2 player game, instead of coding your 2nd character to move also to ‘arrow keys’ you can use the keys W, A, S and D. These will enable 2 people to play the game, commanding 2 separate characters around the created maze.

Extended Exercises

Variables can be used to achieve a Score or a Timer.
To implement a Timer, you can click onto 'Variable' category and click 'Make a variable'. You can name your variable in the pop-up box.
Ensure "For all sprites" is selected and press 'Okay', now you can see some command options have appeared in orange below. You can use "Set Timer to 0" underneath a "When *green flag* is clicked", this will set your Timer to 0 each time you reset your game. You can change the Timer to start at any number you wish, giving your character enough time to get around the maze.

7. Timer

Our Timer won't automatically count down, it is only called a 'Timer' (but doesn't really know it is one yet) so you have to tell it how to count down. You can put this code anywhere as it does not affect any sprites, only the Timer variable in the background of the game.

You can see on the left that we have set the Timer to 60, and after waiting a second the Timer value is being deducted by 1 each time.

Have a go at making your own timer.

Unfortunately, you are missing out some key code here.
What happens when your timer hits 0?
It keeps going, -1 -2 -3 -4 -5 ...

You need to give a command telling our game that forever throughout your game, if TIMER = 0 then you need to send a message to the game alerting it that the game should end here. You can do this using 'Broadcasting', these commands can be found in the 'Events' category. You need to tell your game "If TIMER= 0 (you can find the = sign in 'Operators' category) then BROADCAST a message.

8. Broadcasting

See if you can get your game to broadcast a message when the timer runs out.

What happens next?

You need to create a sprite or backdrop that consists of the text 'Game Over', we can tell it to hide and then only appear when it receives the broadcasted message.

We have used "If" statements quite a bit already, see if you can work out how to do this task.

Hint: You can tell your sprite to 'show' and 'hide' using commands from the 'Looks' category. You can also find a "Switch Backdrop To" command in "Looks"