

S e a n M c M a n u s

2nd  
Edition

# Scratch Programming

updated for Scratch 3

in  
easy steps



PLAIN ENGLISH

EASY TO FOLLOW

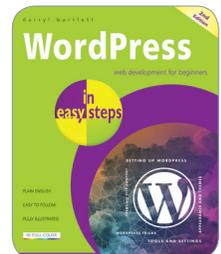
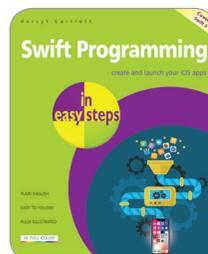
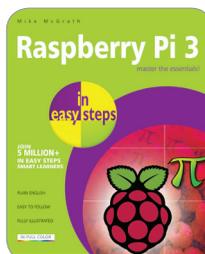
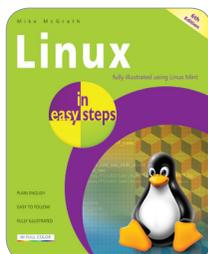
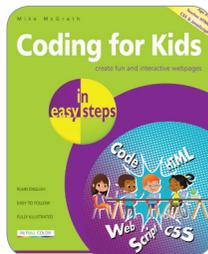
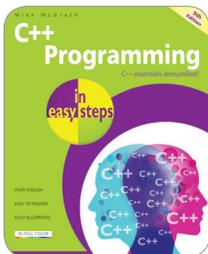
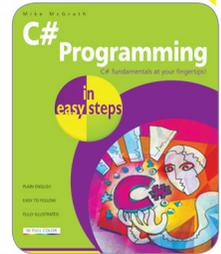
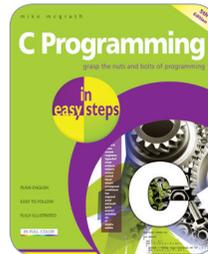
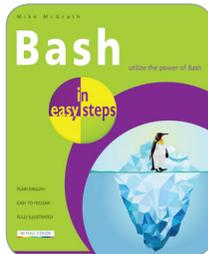
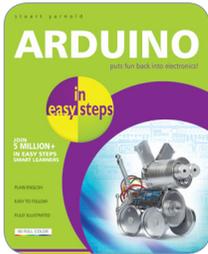
FULLY ILLUSTRATED

IN FULL COLOR

DESIGN & SHARE YOUR OWN PROGRAMS  
CREATE ARCADE GAMES & QUIZZES  
MAKE COMPUTER-GENERATED ART  
LEARN TIPS AND TRICKS

# Liked this book? Learn more in easy steps

Age  
9+



**In Easy Steps ebooks**  
now available to download onto your  
computer, tablet and smartphone from

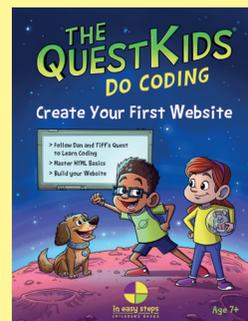
[www.ineasysteps.com](http://www.ineasysteps.com)



amazon kindle



## For children aged 7+



With a captivating storyline, this book will teach aspiring young coders how to build their first website in a fun way, and in easy steps.

visit [www.ineasysteps.com](http://www.ineasysteps.com) for more information on titles and updates

Sean McManus

# Scratch Programming



Second Edition  
updated for Scratch 3

In easy steps is an imprint of In Easy Steps Limited  
16 Hamilton Terrace · Holly Walk · Leamington Spa  
Warwickshire · United Kingdom · CV32 4LY  
[www.ineasysteps.com](http://www.ineasysteps.com)

Content Copyright © 2019 by Sean McManus

In Easy Steps Learning Approach, Series and Format Copyright © 2019  
by In Easy Steps Limited.

All rights reserved. No part of this book may be reproduced or  
transmitted in any form or by any means, electronic or mechanical,  
including photocopying, recording, or by any information storage or  
retrieval system, without prior written permission from the publisher.

#### Notice of Liability

Every effort has been made to ensure that this book contains accurate  
and current information. However, In Easy Steps Limited and the  
author shall not be liable for any loss or damage suffered by readers  
as a result of any information contained herein.

#### Trademarks

All trademarks are acknowledged as belonging to their respective  
companies.

In Easy Steps Limited supports The Forest Stewardship Council (FSC),  
the leading international forest certification organization. All our titles  
that are printed on Greenpeace approved FSC certified paper carry the  
FSC logo.



Printed and bound in the United Kingdom

ISBN 978-1-84078-859-4

## 1

### Introducing Scratch

7

What is Scratch?	8
Creating a Scratch account	9
Meet Scratch 3	10
Using the Scratch screen	11
Using Scratch on tablets	12
Exploring the blocks	14
Changing the backdrop	16
Creating your first program	17
Saving your project	20
Opening projects	21
Opening shared projects	22

## 2

### Drawing with Scratch

23

Understanding coordinates	24
Changing a sprite's position	26
Using the pen	27
Drawing a house in Scratch	29
Using directions to move	31
Keeping sprites upright	32
Drawing using directions	34
Making shapes using repeat	35
Putting loops inside loops	36
Creating Rainbow Painter	37

## 3

### Spiral Rider

39

Introducing Spiral Rider	40
Using the green flag	41
Creating variables	42
Drawing a spiral	44
Changing the backdrop	47
Adding sprites	48
Animating the crab	49
Enabling keyboard control	51
Enabling touch control	53
Making the fish move	54
Adding Game Over messages	57

# 4

## Super Dodgeball

59

Introducing Super Dodgeball	60
Setting up the variables	61
Preparing for the game start	63
Using coordinates to move	64
Adding more images	66
Making random numbers	67
Moving the ball	68
Copying and cloning sprites	70
Adding the energy meter	72
Painting in Scratch	74
Creating the ice cream	77
Making the ice cream appear	78
Enabling the player to score	80
Tweaking the gameplay	82

# 5

## Space Opera

83

Introducing Space Opera	84
Playing sounds	85
Adding the electric guitar	86
Using costumes	87
Adding graphic effects	88
Adding the drum	89
Adding the singer	90
Playing music in Scratch	92
Adding the synthesizer	93
Making your own tunes	94
London Bridge	96

# 6

## Quiz Break

97

Introducing Quiz Break	98
Preparing for Quiz Break	99
Drawing the timer	101
Using broadcasts	103
Testing out broadcasts	104
Moving the clock hand	105
Adding a rising sound effect	108
Asking questions	109
Joining text to greet players	110
Preparing the quiz	111
Making the questions	112
Checking the answers	114
Reporting the scores	116
Adding the victory dance	117

## 7

**Evil Robot****119**

Introducing Evil Robot	120
Importing the images	121
Making a block for speech	122
Making the alphabet sprite	124
Writing on the Stage	125
Designing the Evil Robot	128
Creating the variables	130
Animating the robot	132
Adding the main game code	134
Creating lists	135
Using lists	136
Setting up the variables	138
Creating the game board	139
Showing the game board	140
Asking for the player's guess	141
Checking the letter	143

## 8

**Space Swarm****147**

Introducing Space Swarm	148
Creating the images	149
Adding sound effects	151
Creating the variables	152
Programming the hero	153
Enabling the player to fire	156
Moving and shooting aliens	157
Finishing touches	160
Adding the high score	162

## 9

**Physical Computing with Scratch****163**

What is physical computing?	164
Using a webcam	165
Using video direction	166
Going Batty	168
Making a clap-o-meter	170
Using a Sense HAT	171
Introducing Feeding Time	174
Creating the cat's script	175
Adding the donut scripts	176
Adding the dinosaur scripts	178
Setting up a micro:bit	180
Using a micro:bit	181
Introducing Balloon Floater	182
Drawing the cave	183

Coding the balloon	185
Coding the buttons	188
Using the Raspberry Pi GPIO	189

## 10

### Seven shorties

191

Keepy-Uppy	192
Shop Cat	194
Penguin Patter	197
Abstract Artist	198
Maze Mania	199
Photo Safari	200
Gran Ama's Anagrams Quiz	202

## 11

### Making and sharing projects

205

Making your own programs	206
Fixing common errors	207
Sharing your projects	209
Resources & Acknowledgements	210

### Index

211

# 1

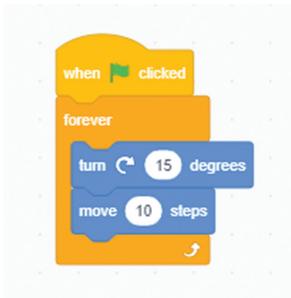
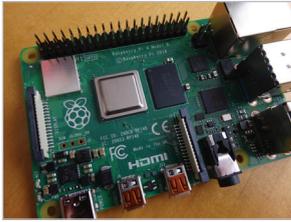
# Introducing Scratch

*In this chapter, you'll get started with Scratch, including meeting some of the blocks used to give instructions, and creating your first program. You'll also learn how to save your work, and load projects created by others.*

- 8** What is Scratch?
- 9** Creating a Scratch account
- 10** Meet Scratch 3
- 11** Using the Scratch screen
- 12** Using Scratch on tablets
- 14** Exploring the blocks
- 16** Changing the backdrop
- 17** Creating your first program
- 20** Saving your project
- 21** Opening projects
- 22** Opening shared projects



Scratch doesn't cost anything and it works on Windows, Mac and Linux computers, as well as tablet devices. For the latest version of Scratch, the Raspberry Pi Foundation recommends a Raspberry Pi 4 computer (pictured below) with at least 2GB of RAM.



Above: A simple Scratch program, showing how the color-coded commands lock together.

# What is Scratch?

Programming is the art of writing instructions to tell a computer what to do. A set of instructions is called a program. The instructions are written in what's known as a programming language, and there are thousands to choose from.

Scratch is a programming language that is perfect for making games, animations, interactive stories and other visually rich programs. It provides a great introduction to programming for people of all ages. It's widely used in schools and colleges, but Harvard University has also used it in higher education at its Summer School. I've led workshops for adults where Scratch provided a friendly introduction to the kind of creative problem solving that programmers do all the time.

Scratch is easier to use than most other programming languages for a number of reasons:

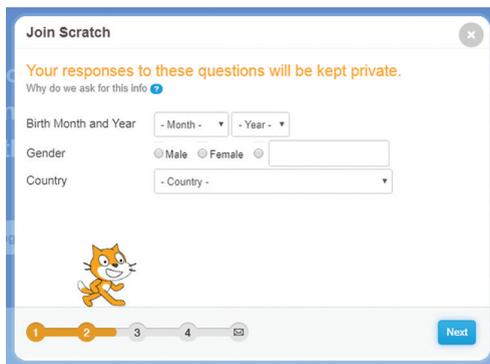
- You don't have to remember or type any commands: they're all on screen, so you can just drag and drop them.
- Commands fit together like jigsaw pieces, so there are strong visual hints about how you can combine them.
- Error messages are rare. Because Scratch commands lock together, programs usually make some kind of sense. It is possible to still write programs with logical errors in, if they don't do what you expected, but Scratch guides you to write things that work, rather than nagging you when they don't.
- The commands are color-coded and categorized, so you can easily find a command when you need it.
- The commands in Scratch simplify common activities in games, such as testing whether a missile has hit an alien (collision detection), or rotating a character on screen.

In short, Scratch is designed for your success. It enables you to quickly see results from your work, and even includes graphics and sounds you can use so you can get started right now. Many other programming languages require you to learn text commands, and have strict rules about how you can use them. Scratch doesn't. That means you can focus your energy instead on the fun stuff: dreaming up ideas for new programs, working out how to build them, designing them, and sharing them with friends.

# Creating a Scratch account

Before you begin to program with Scratch, I recommend you create an account for the Scratch website. Here's how:

- 1 Open a web browser, such as Google Chrome
- 2 Enter the website address **https://scratch.mit.edu/** in your address bar, usually at the top of the screen
- 3 Click **Join Scratch** in the top right
- 4 Make up a username. The site advises you to protect your privacy by not using your real name. You can't change your username later, so choose wisely
- 5 Pick a password and enter it twice. The second time is to make sure you've typed it correctly. Use a mixture of uppercase and lowercase, numbers and symbols to make it more secure. Click the **Next** button
- 6 Enter your date of birth, gender, country and email address. This personal information is used to help you recover your password if you forget it, and is used by the Scratch team to understand who uses Scratch. Click **Next**



- 7 Enter an email address so they can confirm your account before you can share projects or post comments. Click **Next** and you will be logged in



The design of websites can change from time to time, so don't worry if you see variations in the sign-up process when you do it.



You can try Scratch by just going to the website and clicking Create at the top of the screen. If you use an account, though, the website will automatically save your work for you.



When you return to the site next time, you can click "Sign in" in the top right to get back to your projects.



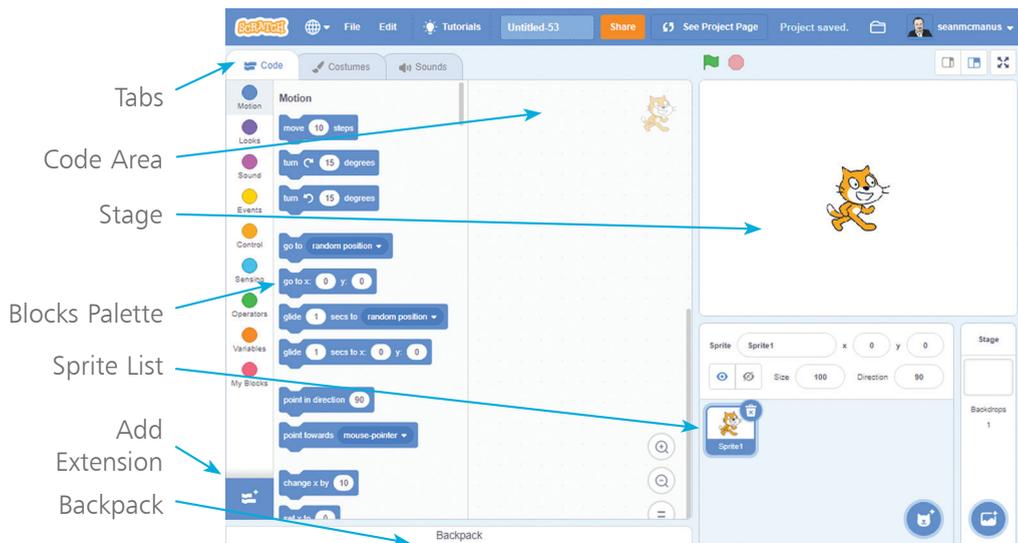
Scratch 3 adds new features for making projects, a streamlined user interface, and compatibility with more devices.

# Meet Scratch 3

Scratch 3 is the latest version of Scratch, launched in January 2019. It runs in your web browser and is based on standard web technologies (HTML5). That means it is compatible with many different types of computer. Scratch 3 introduced several new features, including:

- **Tablet support.** You can now build and use projects on a tablet device, and can play (but not build) projects on a mobile phone. Games that require keyboard controls to move a character on screen may not work. You can use your tablet's onscreen keyboard to type information into a project, though.
- **New extensions.** There are new capabilities to work with the micro:bit and Sense HAT devices, to speak out loud, to translate text, and to work with the Lego WeDo 2.0 and Lego Mindstorms EV3 kits. To add an extension, click the button in the bottom left of the screen.
- **Simplified Blocks Palette.** The menu of instructions has been streamlined by moving the music and drawing (Pen) blocks into the extensions section. If you're familiar with Scratch and are wondering where they went, you can find them there!

Older versions of Scratch (Scratch 2, and Scratch 1.4) are still available to download. There's an offline version of Scratch 3 too. Find them all at <https://scratch.mit.edu/download>



# Using the Scratch screen

The main parts of the Scratch screen, as shown on the facing page, are:

- **Stage:** This is where you can see your animations and games in action. When Scratch starts, there's a large orange cat in the middle of the Stage.
- **Sprite List:** The cat is a "sprite", which is like a character or object in a game. Your project might include lots of sprites, such as the player's spaceship, invading aliens and a missile. In the Sprite List, you can see all the sprites that are in your project, and click them to switch between them. The Sprite List is underneath the Stage.
- **Blocks Palette:** In Scratch, you give the computer commands by using blocks, which are instructions that fit together like jigsaw pieces. The Blocks Palette presents you with all the blocks you can use. When you start Scratch, you can see the Motion blocks, which are color-coded in dark blue, and are used for moving sprites around the Stage. You can browse a different set of blocks in the Blocks Palette by clicking one of the buttons, such as the **Looks** button or the **Sound** button. You can also scroll through all the blocks in the different categories in the Blocks Palette.
- **Code Area:** The Code Area (previously known as the Scripts Area in older versions of Scratch) is where you make your programs, by joining instruction blocks together. When you join blocks together, the result is called a script. The Code Area expands to fill the screen space available, so if you use a larger monitor, the Code Area will be bigger.
- **Backpack:** The Backpack at the bottom of the screen is handy for copying sprites and scripts between different projects. Click the Backpack at the bottom of the screen to open it. You can then drag sprites or scripts into it, or drag them out into your current project.

The blue stripe across the top of the screen is used for managing your projects. You can use the File menu to save your work, including downloading it to your computer. The online version of Scratch saves your updates automatically, so you won't need this often. The box containing "Untitled" is where you rename your project. Click the folder in the top right to find all your projects.



You'll see all these elements in action soon, so don't worry about memorizing the screen layout. This section is just to help you get your bearings. Remember that you can refer to the screenshot on the facing page at any time.



Use the tabs above the Blocks Palette to switch between the scripts, costumes and sounds on a sprite.



With the launch of Scratch 3, touch devices are more widely supported. The previous version of Scratch was based on Flash, which didn't work on iPads. The latest version has redesigned blocks that are easier to move by hand.



If you have a PC with a touchscreen, you might want to use the keyboard and touchscreen together to create your projects. Some people find it easier to use the touchscreen than the mouse.

# Using Scratch on tablets

You can now use Scratch on a tablet, such as an Apple iPad. Simply visit the Scratch website in your browser. You may find that Scratch runs more slowly than it does on a PC, but you should still be able to enjoy building and using many Scratch projects. If you're using a tablet, make sure you log in to the Scratch website, because you can't save projects to your device.

You can't use projects that require the keyboard, such as games that use keys to move a sprite. If the project includes **key [space] pressed?** or **when [space] key pressed** blocks (to detect any key), the project is unlikely to work well without a real keyboard. From this book, you can't play games like Super Dodgeball or Shop Cat (although they are fine to build). But you can play Spiral Rider, Space Opera, Quiz Break, Evil Robot, Photo Safari and more. I'll warn you at the start of a chapter if a project won't work on a touch-based device.

## Using gestures

You'll find the touch gestures feel natural and are easy to remember. Here's how to use a touch-based device instead of a mouse:

- To “click” something, **tap** it on the screen. For example, tap to use a button or start a script.
- To “right-click” something, **tap and hold** it on the screen – for example, to duplicate a sprite in the Sprite List.
- To “hover” over a button, simply **tap** it – for example, to open the menu to add a sprite.
- You can “drag” blocks and sprites by touching them and **dragging your finger** across the screen.
- To scroll through the Blocks Palette, **swipe** it up or down.
- To move around the Code Area, **drag** its background up, down, left or right.
- To scroll in the Paint Editor (see Chapter 4), **drag the scrollbars** at the bottom and side of the canvas.
- To change the values in a block, or type something in response to an **ask** block on the Stage, **tap the box to make your virtual keyboard appear**.

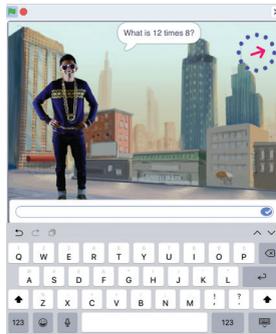
...cont'd

## Scrolling around the Scratch screen

If Scratch doesn't fit in your tablet's screen, you can scroll left and right by dragging the blue bar at the top. To scroll up and down, drag part of the light blue background – for example, an empty space in the Sprite List.

## Using Scratch in portrait mode

If you want to see all of the Stage and the keyboard at the same time – for example, to play Quiz Break in this book – use your tablet in portrait orientation.



## Using Scratch apps

At the time of writing, MIT is creating apps for Scratch 3. There is an unofficial app based on Scratch 1.4, called Pyonkee. It's a fun way to tinker with Scratch, but it doesn't have the cat or certain other sprites, and doesn't have newer Scratch features such as sprite cloning, extensions, and sound effect blocks.



In the rest of this book, I'll mostly assume readers are using a keyboard and mouse, so refer back here if you need a reminder of how to use your touchscreen.



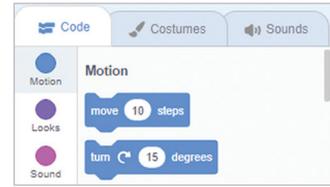
MIT has also created ScratchJr, a reinvention of Scratch for younger children, designed for iPads and Android tablets. For tips on using ScratchJr, see our book [Cool Scratch Projects in easy steps](#).



# Exploring the blocks

Try experimenting with a few blocks to see what they do:

- 1 Click the round, blue **Motion** button to the left of the Blocks Palette to show the Motion blocks. This button is selected when you first start Scratch



- 2 In the Blocks Palette, click (or tap) the **move 10 steps** block. The cat on the Stage moves in the direction it's facing, to the right. Each time you click the block, the cat moves once. This block only changes the cat's position, though: you won't see its legs move



Scratch won't let the cat disappear off the screen completely. If you use numbers that are too big, the sprite will stay at the edge of the screen.

- 3 The number of steps is how far across the screen you want the cat to move. Click the number 10 and change it to something else. Try 50 and when you click the block, the cat moves five times as far. Whenever you see a white hole in a block, you can change what's in it



Above: The cat on the Stage, after I clicked "turn clockwise 15 degrees".

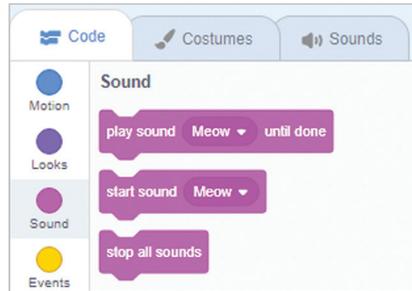
- 4 Rotate the cat by clicking the **turn clockwise 15 degrees** block. To change the angle of the turn, change the number. Remember to click the block to actually make the cat turn. When you click the **move 10 steps** block next time, the cat walks in its new direction



...cont'd

5 If the cat gets to the edge of the Stage, drag it back again with your mouse pointer. Click the cat, hold the mouse button down, move the cat, and then release the mouse button to drop it in place

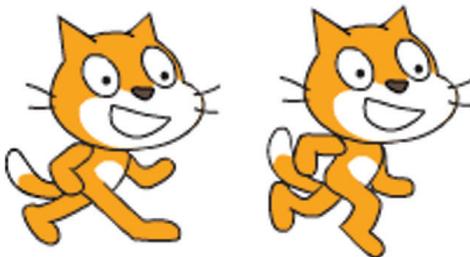
6 Click the round, pink **Sound** button beside the Blocks Palette



7 Click the **play sound Meow until done** block. You should hear the cat make a meow sound if you have speakers on

8 Click the round, purple **Looks** button. It's beside the Blocks Palette. You can see it in the picture above

9 Click the **next costume** block to see the cat's legs move, so it appears to run on the spot. Costumes are different pictures a sprite can have, and the cat has two that show its legs in different positions (see below). In Scratch, the sprites are like the actors; they wear the costumes (different images), and they move on the Stage



All these blocks are explained later in the book, but for now it's worth spending a few minutes exploring some of the commands you can give the cat.



Scratch 3 includes new blocks to change the pitch of the sound. Try changing the numbers in the "change [pitch] effect by 10" block, clicking it, and then clicking the "play sound Meow until done" block again.



If you're using a tablet, you can tap things instead of clicking them.

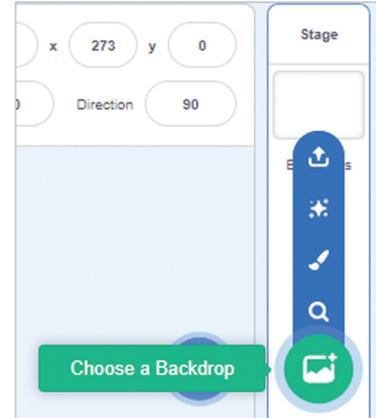


Before you begin building each program in this book, it's a good idea to start a new project. Click the File menu and then click New. If you're using Scratch 3 and you can't see the File menu, click Create instead at the top of the screen.

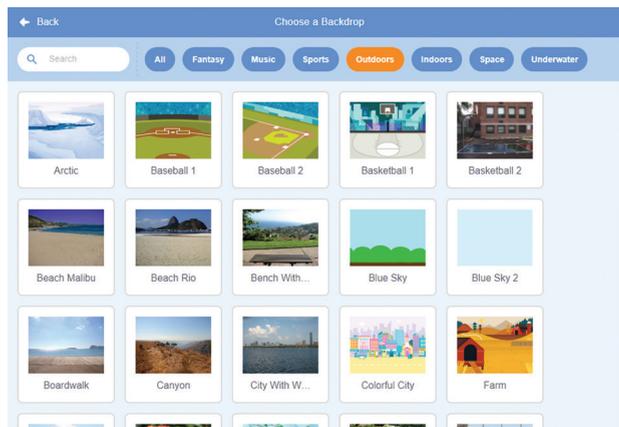
# Changing the backdrop

Before we make our first program, let's change the background of the Stage to something more inspiring. The background is called a backdrop in Scratch.

- 1 To the right of the Sprite List, there is a panel for the Stage. At the bottom, hover over the round button to open the backdrop options. Use the magnifying glass icon to choose a backdrop from the library. (You can also just click the round button to choose from the library.) The other options enable you to paint a backdrop, get a surprise backdrop, or upload a picture from your computer



- 2 When the library opens, click the categories at the top to view the different backdrops available, and use the scrollbar on the right to see more designs. Select the Outdoors category



- 3 Scroll to find, and then select, the Hill image



Scratch 3 adds some new backdrops, including some great space cities, illustrated sports fields, and a Jurassic skyline.

# Creating your first program

When you click blocks in the Blocks Palette, the cat moves immediately, so this is good for testing what blocks do, but not useful for making a program. A program is a set of repeatable instructions that you can store up to carry out later. For our first Scratch program, let's make the cat walk down the hill:

- 1 We're going to write a program for the cat, so click the cat in the Sprite List
- 2 Next, check that the Code Area is open. If the middle pane shows costumes or sounds instead, click the **Code** tab above the Blocks Palette

- 3 Click the **Motion** button to the left of the Blocks Palette (see screenshot, right)



- 4 You make a program by dragging blocks into the Code Area from the Blocks Palette. To do this for your first block, click the **turn clockwise 15 degrees** block in the Blocks Palette, hold the mouse button down, move the mouse pointer into the Code Area and then release the mouse button. This first block will point our cat downhill, ready for its walk

- 5 Click the **move 10 steps** block in the Blocks Palette, drag it into the Code Area and drop it underneath the **turn clockwise 15 degrees** block. They will snap together. When blocks are joined like this they make what's known as a "script". A sprite can have more than one script, and a program might include lots of sprites with several scripts



If you click and drag a block that's joined to other blocks above it in the Code Area, it will break away from them, and carry all the blocks underneath it with it.



You can move blocks around the Code Area, but if you drag them into the Blocks Palette, they'll be deleted.



Make sure the blocks snap together, otherwise they won't work as one script. If they don't snap together, they're not close enough.

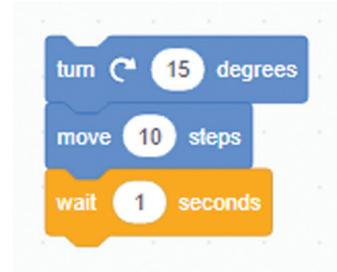


I wouldn't usually recommend you add the same blocks repeatedly, but we're right at the beginning of learning Scratch here. There is a more readable and elegant solution you'll discover in Chapter 2.

...cont'd

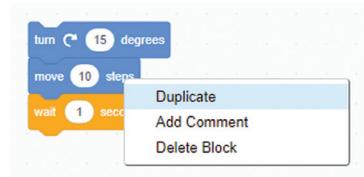
6

Click the **Control** button beside the Blocks Palette. Control blocks are used to decide when things should happen. Drag the **wait 1 seconds** block into the Code Area and snap it underneath the other two blocks. This block adds a 1-second delay. Without it, our cat would move so fast, it would appear to just jump from the start of its walk to the end. Slowing it down enables us to see what's going on. You can make it walk a bit faster by changing the delay from 1 second to 0.5 seconds



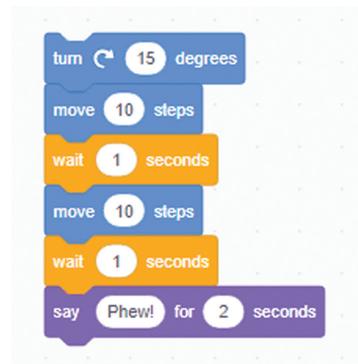
7

Right-click the **move 10 steps** block, and when the menu opens, choose **Duplicate**. This copies the block plus any blocks underneath it in your script. In our example, it copies the **move** and the **wait** blocks. Move the copy to the bottom of your program, and click to place the blocks there. You can repeat this step several times to make the cat walk further



8

It's a long walk for a tiny cat, so let's make it finish its walk with an exclamation of "Phew!" in a speech bubble. Click the **Looks** button beside the Blocks Palette. Drag the **say Hello! for 2 seconds** block into the Code Area and join it to your program. Click Hello! to edit what the cat says to Phew!



To right-click on a touchscreen device, tap and hold. When you duplicate a script, the copy is placed in the Code Area. You can drag it to the bottom of your script.

...cont'd

**9** When you start a script's commands, it's called "running" the script. To run your script, click any of the joined-up blocks in the Code Area. Scratch carries out all the joined-up instructions in order, starting at the top and working its way down the blocks

**10** What happens if you click the script to run it again? The cat turns again and walks from where it finished last time. Eventually, it'll be walking on its head. Let's add some blocks to put it in the right starting position. Click the **Motion** button beside the Blocks Palette and drag in the **point in direction 90** block and the **go to x:0 y:0** block. If the **go to** block has different numbers in it, edit them both to make them zero. Add these blocks to the top of your script. Can you make the cat's walk longer?



You can insert blocks into an existing script. Instead of dropping a block at the end of your script, drag your block over it. The blocks in the script will make room to let the new blocks in. When the blocks are in the right place, release the mouse button or your finger.



Experiment! If you use a negative number of steps in the "move 10 steps" block, the sprite moves backwards. Why not see if you can make the cat walk backwards up the hill again?

Left: A few blocks and one exhausted cat later, here's your first program.



You can't use your projects in an older version of Scratch than you used to create them. You can use projects with a newer version, though.

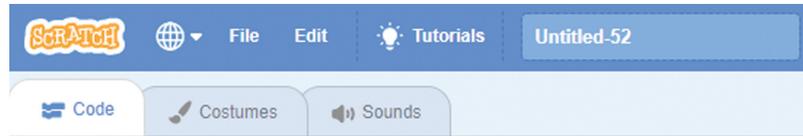
# Saving your project

A Scratch project includes all the sprites, scripts and backdrops that are used in it. It's a good idea to save your projects so you can come back to them later to reuse them or modify them.

In Scratch 3 online, your work is automatically saved for you. If your latest changes have not been saved, there will be a link to **Save Now** in the blue bar at the top of the screen, beside the folder icon that gives you access to your projects.



Your project is saved with the name Untitled plus a number. You can choose a more useful name by editing the box in the blue bar, above the Code Area, and to the right of the tabs.



There are additional options for saving your work in the File menu. You can find it above the Blocks Palette, as shown above. These options include:

- **Save as a copy:** This makes a copy of your project with a new name. The previously saved version of your project is left untouched. Use this if you want to experiment with your project without losing a working version of it.
- **Save to your computer:** This enables you to save your project as a file on your computer. If your internet connection fails, use this option straight away to save your work!
- **Load from your computer:** If you previously downloaded a Scratch project to your PC, or used a downloadable version of Scratch to create it, use this option to upload it to the Scratch website, so you can edit and share it there.



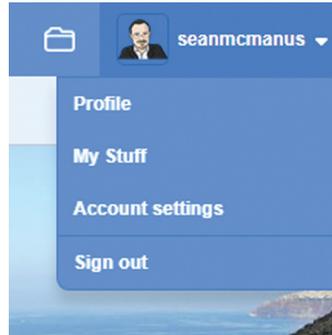
In the downloadable versions of Scratch, your work isn't automatically saved for you. Use the **Save to your computer** option in the File menu to save your work.

# Opening projects

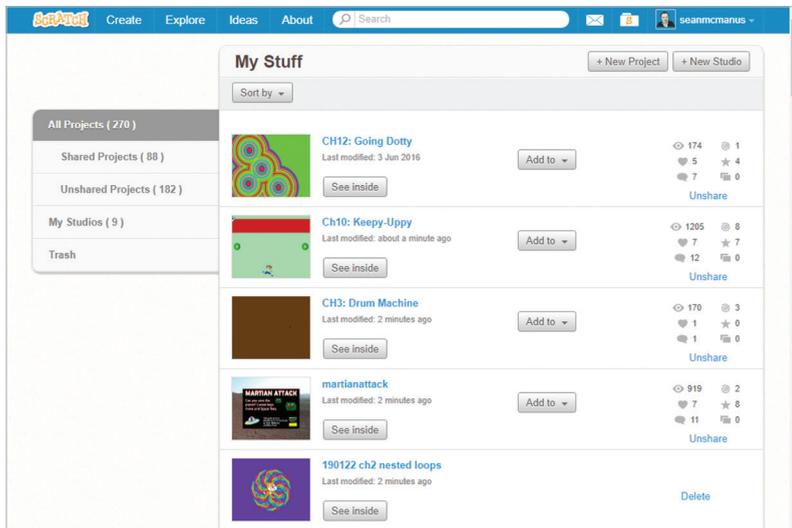
To find your projects on the Scratch 3 website, click the folder icon beside your username in the top right of the screen. You can also click your username and then click **My Stuff**.

If it says “Sign in” in place of your username, click it to sign in first.

The My Stuff section shows all your projects, with those you most recently edited nearer the top. You can see some of my projects in the screenshot below:



If you click the Share button on a project, it will be available for anyone to see, use and reuse. See Chapter 11 for more on sharing your projects.



Use the scrollbar at the right edge of your browser window to find more projects and click the **Load More** button when you get to the bottom of the list. To open a project for editing, click its **See inside** button. To simply run the project, click its screenshot.

Because projects are saved automatically, your My Stuff area quickly fills up with untitled projects. To tidy up, delete unwanted projects by clicking their **Delete** links on the right. If you delete a project by mistake, click the **Trash** folder on the left, and then click the **Put back** button to recover the project. You can't delete a shared project without unsharing it first (see page 209 in Chapter 11).



In downloaded versions of Scratch, you open projects by clicking to open the File menu at the top of the screen and then choosing Open.

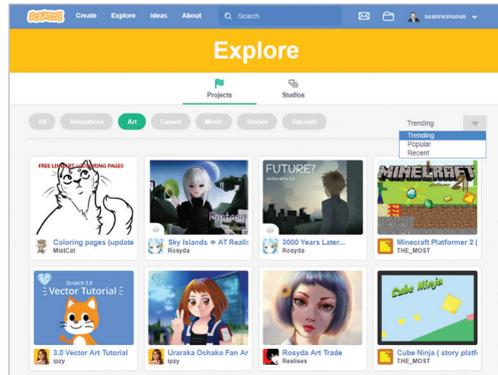


If you are logged in, you can click the Remix button to create your own version of a project shared on the Scratch website. All the projects on the Scratch website are shared on the understanding that others can learn from them, and create adaptations of them. You can find a Remix button on the project's page, and another in the blue bar at the top of the screen when you look inside the project.

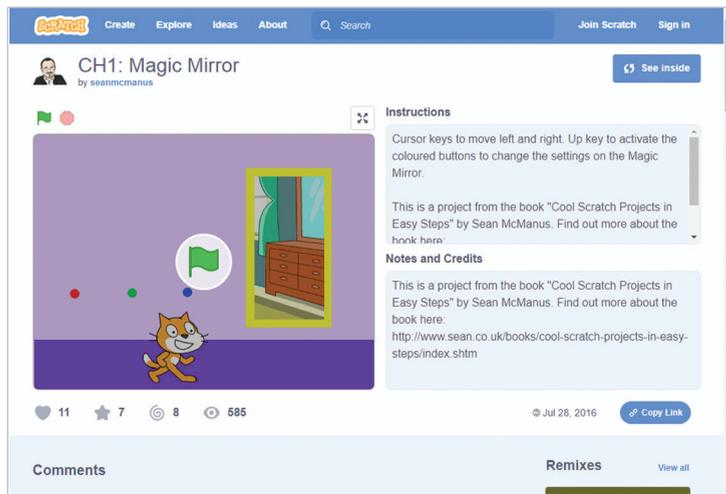
# Opening shared projects

You can open the projects that other people have shared on the Scratch website too. Visit the website at <https://scratch.mit.edu> and click **Explore** at the top of the screen.

Use the buttons to choose a category of projects to explore. The triangle to the right of the project categories opens a menu that you can use to find projects that are trending, popular, or recent.



Click a project to go to its page. The instructions on the right tell you how to use the project, and you click the **green flag** button to run it, either the one in the middle of the project or the one above the project's display. If you like what you see, and you want to know how it was done, click the **See inside** button in the top right to go into the editor and see the code.



# 2

# Drawing with Scratch

*Find your bearings on the Scratch Stage, learn how to move sprites and draw with them, and discover how loops make it easy to repeat parts of your program. You'll also make an interactive art program called Rainbow Painter.*

- 24** Understanding coordinates
- 26** Changing a sprite's position
- 27** Using the pen
- 29** Drawing a house in Scratch
- 31** Using directions to move
- 32** Keeping sprites upright
- 34** Drawing using directions
- 35** Making shapes using repeat
- 36** Putting loops inside loops
- 37** Creating Rainbow Painter



You can click and drag your sprites around the Stage to reposition them while you're writing your programs. It's a good idea to use the "go to x:0 y:0" block to position them in the program, though, otherwise they might start off somewhere unexpected. If your sprite moves in the program, it won't be in the same position the second time the program is used. Other people can experiment with your program too, and might move sprites around before it starts.



An easy way to remember which way around x and y should be is that "x is a cross" (and "across").

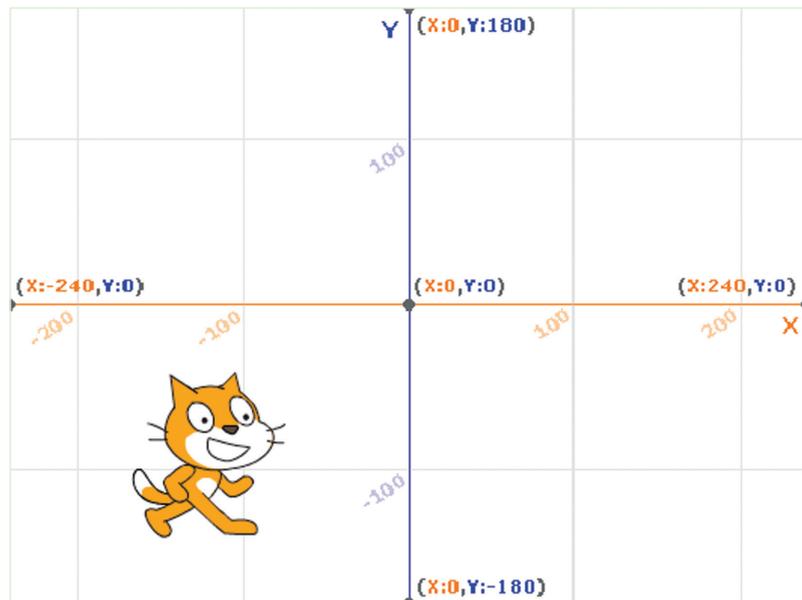
# Understanding coordinates

You can use the sprites in Scratch to draw on the Stage. It's a great way to familiarize yourself with how to move sprites, and the technique can be used to create designs that your sprites can interact with, as you'll see when we make a game in Chapter 3.

First, let's take a look at how you position sprites on the Stage. Each position on the Stage has a grid reference, similar to those used on maps. The position across the Stage horizontally is called x, and the position up or down the Stage is called y.

When you start a new project, the cat is in the middle of the Stage, and this position has the grid reference  $x=0$  and  $y=0$ .

Here's a map of the grid references on the Stage, after I've moved the cat to the bottom-left quarter of the Stage:



As you can see, the numbers for the y coordinate run from -180 at the bottom of the Stage, up to +180 at the top of the Stage.

The numbers for the x coordinate run from -240 at the left edge of the Stage, up to +240 at the right edge of the Stage.

# 3

## Spiral Rider

*Learn how to make your first game, including using the green flag to start scripts, using variables to store a number, moving a sprite with the keyboard or touchscreen, and letting Scratch make simple decisions about which blocks should run.*

- 40** Introducing Spiral Rider
- 41** Using the green flag
- 42** Creating variables
- 44** Drawing a spiral
- 47** Changing the backdrop
- 48** Adding sprites
- 49** Animating the crab
- 51** Enabling keyboard control
- 53** Enabling touch control
- 54** Making the fish move
- 57** Adding Game Over messages

# Introducing Spiral Rider

Poor Freddy Fish has bought a new T-shirt and the tag in the back is really itching. Chris the Crab has claws that could clip the tag off, if only the pair could meet up. Chris has crawled through a maze of passages in the rock. Can you help Freddy get there?

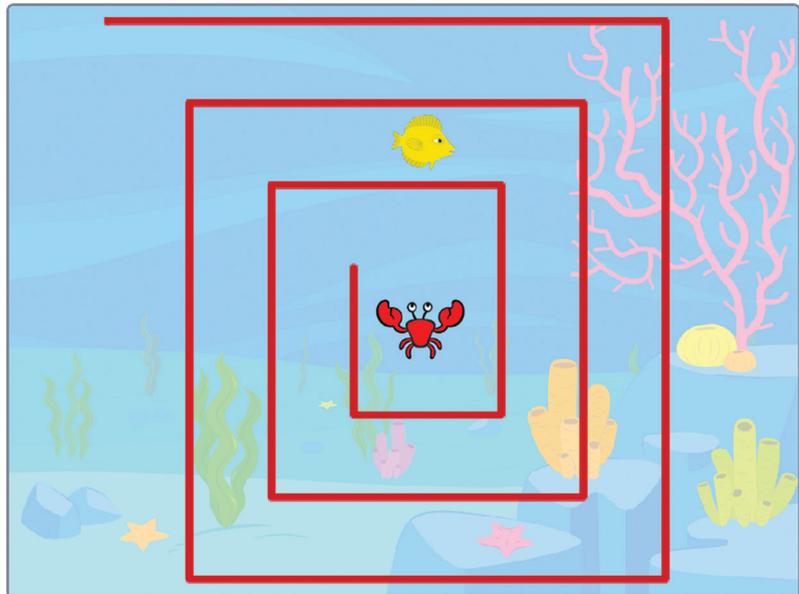
Behind this silly premise lies a game that will use what you have already learned about Scratch and take it further to help you make your first real game. You'll learn how to:

- Keep track of a number, so you can make a square spiral with ever decreasing line lengths.
- Use multiple sprites, and use the green flag to synchronize them.
- Move sprites with the keyboard or using a touch control on a tablet device.
- Make a sprite move automatically.
- Detect when a sprite hits something.

In this game, you control the fish using the arrow keys or by tapping it to make it turn. You must turn at the right time on each corner to navigate the spiral successfully.



If you can't get the examples to work, or if you want to take a shortcut, you can find them online. Visit the author's profile on the Scratch website at <https://scratch.mit.edu/users/seanmcmanus/> or follow the link at <https://www.sean.co.uk/scratch/> or at [www.ineasysteps.com/resources/download](http://www.ineasysteps.com/resources/download)



You can play this game on a tablet. We'll add a special touch control so you can turn by tapping the fish.

# 4

# Super Dodgeball

*In this chapter, you'll make a game where you have to dodge bouncing balls and collect ice creams. As you build the game, you'll learn a new way to move sprites, how to make random numbers and use them to add suspense to a game, how to paint your own sprites, and how to clone sprites.*

- 60** Introducing Super Dodgeball
- 61** Setting up the variables
- 63** Preparing for the game start
- 64** Using coordinates to move
- 66** Adding more images
- 67** Making random numbers
- 68** Moving the ball
- 70** Copying and cloning sprites
- 72** Adding the energy meter
- 74** Painting in Scratch
- 77** Creating the ice cream
- 78** Making the ice cream appear
- 80** Enabling the player to score
- 82** Tweaking the gameplay



This game requires keyboard controls to play, so you can't play it on tablet devices.



If you need more room in the Code Area, reduce the size of the Stage. Above the Stage are three buttons. The first makes the Stage small, the second makes it normal size, and the third one is for full screen.



# Introducing Super Dodgeball

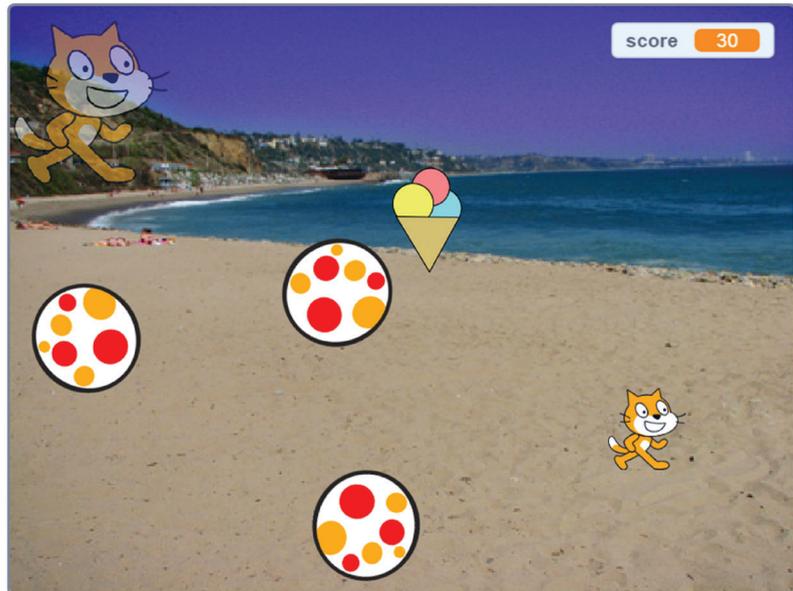
It can be hard to relax at the seaside with so many other people around, playing games and getting in the way. In Super Dodgeball, you have to avoid all the beachballs bouncing around.

Each time a ball hits you, your strength is sapped. Your energy is shown with a semi-transparent picture of your cat character, which shrinks as your strength decreases.

You score by collecting ice creams that pop up on the screen, but they soon disappear again if you're not quick enough. Ice cream also gives you a small dose of energy, to help you recover from being hit by the beachballs.

In this chapter, you'll learn how to:

- Move sprites in a new way, under keyboard control.
- Design your own sprites.
- Add randomness to your game.
- Copy and clone sprites.
- Add sound effects.
- Adjust the game's difficulty.



# 5

## Space Opera

*In this chapter, you get to conduct an alien opera performance, triggering samples and musical notes.*

*You'll learn how to add sound effects and musical notes to your projects. You'll also learn how you can use Scratch to compose music or play your favorite tunes.*

- 84** Introducing Space Opera
- 85** Playing sounds
- 86** Adding the electric guitar
- 87** Using costumes
- 88** Adding graphic effects
- 89** Adding the drum
- 90** Adding the singer
- 92** Playing music in Scratch
- 93** Adding the synthesizer
- 94** Making your own tunes
- 96** London Bridge



This project mostly works fine on touch devices. The guitar needs a computer keyboard, though, so you might prefer to leave the guitar out if you don't have a computer keyboard.

# Introducing Space Opera

In Space Opera, you conduct an operatic alien singer, and can jam along on the keyboards, guitar and drum. You use the top row of letter keys (from Q to I) to control which notes the guitar plays, and click the other instruments to play them. The keyboard note varies depending on which part of the keyboard you click, with low notes on the left and high notes on the right. The singer sings a lower note than the previous one if you click them on their left, and a higher note than the previous one if you click them on their right. You can make up tunes by clicking on their left and right hands in different patterns.

This project teaches you some important Scratch techniques:

- Two different ways to make music, by playing short recordings, and by playing notes using simulated instruments.
- How to use graphic effects to change a sprite's appearance.
- How to animate a sprite using multiple costumes.

Now that you've completed several Scratch projects, you're an expert on the basics. In this chapter, I'll assume that you know how to add blocks to a script, and where to find the blocks you've used often before, such as the **when green flag clicked** block, and the **go to x:0 y:0** block.



# 6

## Quiz Break

- Can you beat the clock?*
- In Quiz Break, you are challenged to answer 10 maths questions in an average of three seconds each. If you do it, the quizmaster dances. In this project, you'll learn how to use the timer, ask the player questions, and join text in speech bubbles.*
- 98** Introducing Quiz Break
  - 99** Preparing for Quiz Break
  - 101** Drawing the timer
  - 103** Using broadcasts
  - 104** Testing out broadcasts
  - 105** Moving the clock hand
  - 108** Adding a rising sound effect
  - 109** Asking questions
  - 110** Joining text to greet players
  - 111** Preparing the quiz
  - 112** Making the questions
  - 114** Checking the answers
  - 116** Reporting the scores
  - 117** Adding the victory dance



You can play this game using a tablet. I recommend using portrait orientation. Tap the answer box to see your keyboard. You can adjust the difficulty if you find it too hard to answer quickly on a tablet.

# Introducing Quiz Break

In recent years, games based on speed mathematics have become popular on handheld consoles and phones. They're a good warm-up for your brain, as well as being itchy, frustratingly good fun.

In the game Quiz Break, you're challenged to answer simple mathematics questions against the clock. If you can answer 10 questions quickly enough, your host will be so impressed he'll dance for you. You don't get that on TV quiz shows!

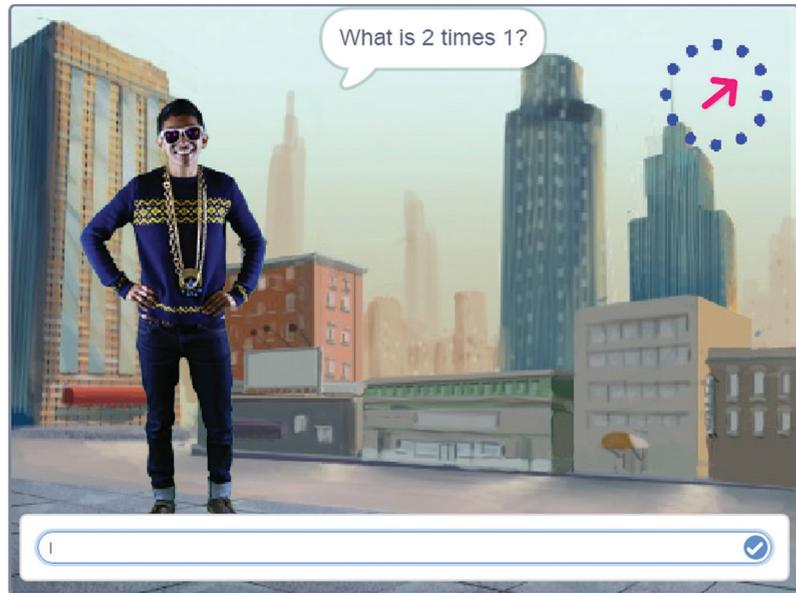
In this chapter, you'll learn how to:

- Ask the player questions.
- Use a timer to measure how long the player is taking.
- Use broadcasts to coordinate actions between sprites.
- Use the Operators blocks to perform calculations.
- Join pieces of text together so you can make sprites say more complex sentences in their speech bubbles.

To complete this project, you'll draw on what you've learned in previous chapters, including using the pen (see Chapter 2), variables (see Chapter 3), random numbers (see Chapter 4), and multiple costumes (see Chapter 5).



This game is set up to test your multiplication tables, but you can easily make it test addition, subtraction or division by changing the Operator blocks used.



# 7

# Evil Robot

- Create a version of the classic word game Hangman, and you'll learn how to structure more sophisticated projects using your own blocks, how to use lists to manage information, and how to write on the Stage with a sprite. You can personalize this game with your own word lists and your choice of cartoon host.*
- 120** Introducing Evil Robot
  - 121** Importing the images
  - 122** Making a block for speech
  - 124** Making the alphabet sprite
  - 125** Writing on the Stage
  - 128** Designing the Evil Robot
  - 130** Creating the variables
  - 132** Animating the robot
  - 134** Adding the main game code
  - 135** Creating lists
  - 136** Using lists
  - 138** Setting up the variables
  - 139** Creating the game board
  - 140** Showing the game board
  - 141** Asking for the player's guess
  - 143** Checking the letter



You can play this game using a tablet. I recommend using portrait orientation.

# Introducing Evil Robot

In this traditional game, commonly known as Hangman, you have to guess the letters that could be in a mystery word. If you guess correctly, you're shown where your guessed letter appears in the word. If you guess wrongly, another part of the evil robot is assembled. When the robot is complete, it's game over.

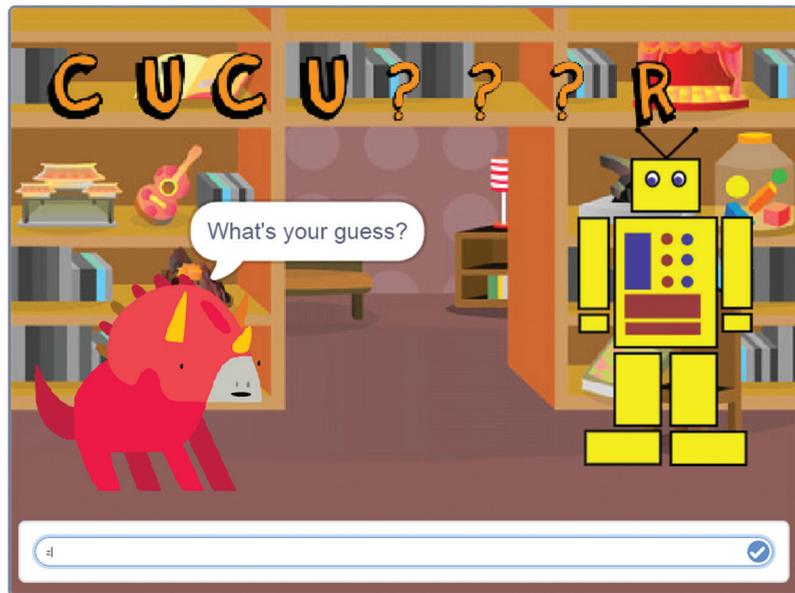
This is a one-player game, with the computer randomly picking a word from a list, and telling you whether your guesses are right or wrong. The game idea might seem simple, but implementing it in Scratch will draw on most of your skills so far.

You'll learn how to:

- Make Scratch talk out loud using the Text to Speech extension, a new feature in Scratch 3.
- Store lists of information, such as lists of letters or words, and use blocks to add, find and remove items in the list.
- Write text on the Stage using a sprite with costumes for each letter of the alphabet.
- Look at individual letters in a piece of text.
- Use functions to create scripts that are easier to understand, and easier to write.



It's okay to experiment with your programs to see what works and what doesn't. I tried making this game another way at first, using a different sprite for each letter in the word. It was clumsy synchronizing between the sprites, so I worked out the simpler solution you'll see in this chapter.



# 8

## Space Swarm

*Use what you've learned so far and build on it to make an arcade game. You'll learn how to use loops to create special effects, how to enable a sprite to fire at another sprite, and how to make a high score table.*

- 148** Introducing Space Swarm
- 149** Creating the images
- 151** Adding sound effects
- 152** Creating the variables
- 153** Programming the hero
- 156** Enabling the player to fire
- 157** Moving and shooting aliens
- 160** Finishing touches
- 162** Adding the high score



This game requires keyboard controls, so you can't play it on tablet devices.

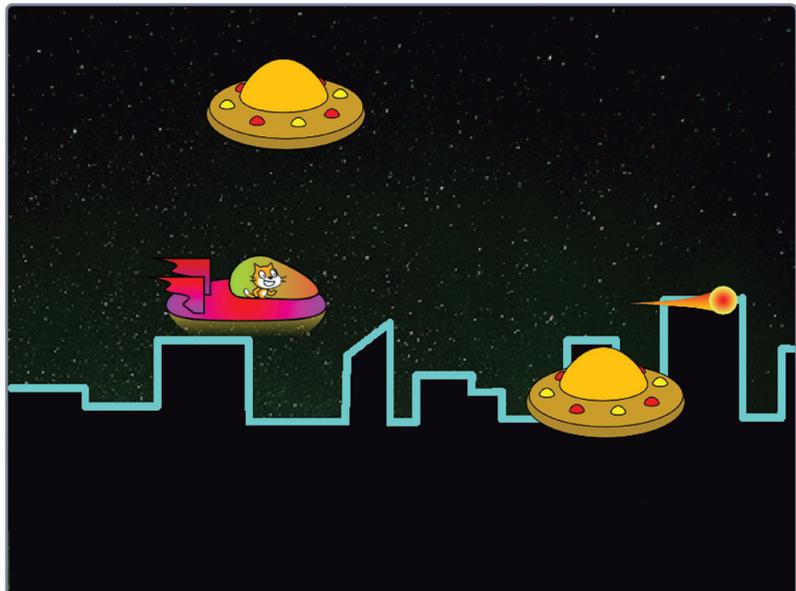
# Introducing Space Swarm

We're under attack! As the aliens swarm around the planet, eager to plunder all of its resources, only one person can save us. You!

Space Swarm, in the spirit of classic arcade games, sees you firing on enemies and dodging their invasion. The aliens rush in towards you from the right of the screen, zig-zagging in random directions. To play the game, move your character with the arrow keys and press the spacebar to fire. You lose one of your three lives each time an alien hits you.

This game brings together much of what you've learned in previous chapters, and will also show you how to:

- Add special effects to make sprites materialize and evaporate.
- Enable one sprite to fire upon another.
- Use flags to swap information between sprites.
- Add looping music to your game.
- Make a high score table that remembers the best score anyone has achieved playing your game on the Scratch website.
- Give the player three lives.



# 9

# Physical Computing with Scratch

- Design new ways to interact with Scratch using hardware devices such as webcams, microphones, the Sense HAT add-on for the Raspberry Pi, and the micro:bit. This chapter includes two games that use tilt sensors and instructions for controlling a light connected to a Raspberry Pi. You'll also see how to make a clap-o-meter and games that respond to your actions on camera.*
- 164** What is physical computing?
  - 165** Using a webcam
  - 166** Using video direction
  - 168** Going Batty
  - 170** Making a clap-o-meter
  - 171** Using a Sense HAT
  - 174** Introducing Feeding Time
  - 175** Creating the cat's script
  - 176** Adding the donut scripts
  - 178** Adding the dinosaur scripts
  - 180** Setting up a micro:bit
  - 181** Using a micro:bit
  - 182** Introducing Balloon Floater
  - 183** Drawing the cave
  - 185** Coding the balloon
  - 188** Coding the buttons
  - 189** Using the Raspberry Pi GPIO



The new extensions feature in Scratch 3 has been used to add support for the Sense HAT and the micro:bit.



Check out our companion titles, [Raspberry Pi in easy steps](#), [Raspberry Pi 3 in easy steps](#), and [Electronics in easy steps](#) at [www.ineasysteps.com](http://www.ineasysteps.com), for more information on these topics.

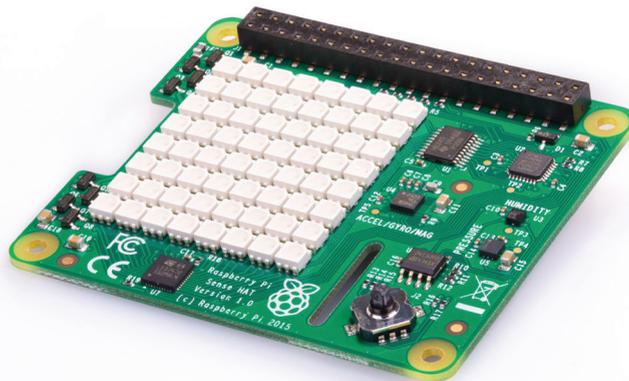
Right: The Sense HAT adds sensors to a Raspberry Pi computer. In this chapter, you'll see how to use it as a game controller.

# What is physical computing?

Physical computing is all about building computer projects that interact with the real world. In this chapter, we'll explore how different sensors can be used to control Scratch projects. We'll also introduce the simple electronics extension for the Raspberry Pi.

You'll see:

- How Scratch can detect motion on a webcam, and how you can use that to control action on the Stage.
- How Scratch can detect volume using a microphone connected to your computer, and use that to control scripts. We'll make a clap-o-meter, as used on TV talent shows to measure which act gets most audience applause.
- How the Sense HAT add-on for the Raspberry Pi can be used to detect movement, and to display messages using an 8x8 grid of multicolored lights. We'll make a game where you tilt the device to move around the screen, and use the tiny joystick to fire in four directions.
- How the micro:bit device can be used to detect movement and button pushes. We'll make a game that demands physical precision as you tilt the device to control a floating balloon.
- How to connect up an LED to the Raspberry Pi and turn it on and off using the simple electronics extension.



# 10

## Seven shorties

*To finish up, here are seven short programs you can build and experiment with, including five games, a chat program and an art generator.*

- 192** Keepy-Uppy
- 194** Shop Cat
- 197** Penguin Patter
- 198** Abstract Artist
- 199** Maze Mania
- 200** Photo Safari
- 202** Gran Ama's Anagrams Quiz

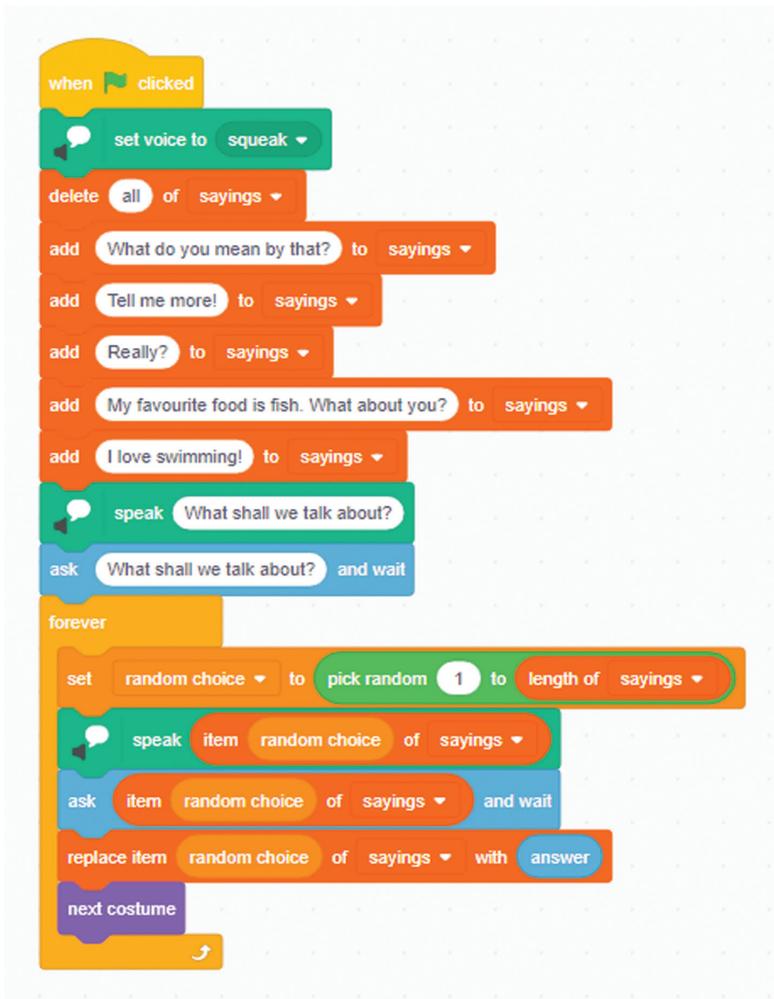
# Penguin Patter

In Penguin Patter, you can chat to a friendly penguin:

- 1 Delete the cat. Add a backdrop and sprite. I've used the sprite Penguin 2 and Beach Malibu backdrop
- 2 Make the variable *random choice* and the list *sayings*. They can be for all sprites. Uncheck the boxes in the Blocks Palette to hide them
- 3 Add this script to your sprite. You'll need to add the Text to Speech extension for the **speak** block



Above: You type in what you want to say. The penguin's replies appear on screen and are also spoken out loud.



I used a variable to remember which saying was randomly chosen, so the program can replace it with what the player enters. As a result, the penguin learns what you say and copies you.



Customize this game with your own personality. You can add lots more sayings at the start, and the program works better the more you add.

# 11

## Making and sharing projects

*Now it's time to make your own projects! Discover tips and resources to help here.*

- 206** Making your own programs
- 207** Fixing common errors
- 209** Sharing your projects
- 210** Resources & Acknowledgements

# Making your own programs

Now you've seen how programs are built in Scratch, you can start to design and share your own creations. Here are some tips:



Scripts glow in the Code Area as they run.

**1** It's rare to program something that works first time unless it's a really simple program. Programming is all about trying something out, testing it, and then changing your scripts. Keep testing your program as you build it. You can click a script in the Code Area to run it straight away without running your whole program

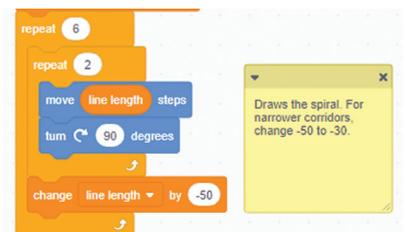
**2** If you need to reset the position or look of a sprite for testing, remember you can click blocks in the Blocks Palette to use them on your sprite straight away

**3** If you're having problems with a variable, click the checkbox beside it in the Variables part of the Blocks Palette. That will show the variable's value on the Stage so you can see what's going on. You can also show the x and y coordinates and direction of a sprite by checking the boxes beside them in the Motion blocks

**4** If you can't see all the blocks, check whether you need to add an extension. Extension blocks have an icon on them and can be added using the **Add Extension** button in the bottom left of the Blocks Palette

**5** To tidy up the Code Area, right-click on it and then choose Clean up Blocks from the menu. This lines up all your scripts and stops them overlapping

**6** Add comments to your script to help you to remember what it's doing, and what you need to do with it next. Right-click on the Code Area to add a comment box. Click the bottom-right corner to resize it, and drag it into the Blocks Palette to delete it



A great way to start making your own projects is by modifying existing projects. How can you build on the projects in this book?



Also available in this series by the same author: **Cool Scratch Projects in easy steps**. Build a drum machine, make projects with 3D glasses (included in the print edition), animate cartoon frogs, code maze games, and more!



If you enjoyed this book, please blog about it or write a review of it on your favorite online store! Thank you!

# Resources & Thank yous

The Scratch website  
<https://scratch.mit.edu/>

The author's website, including examples, links to find the projects in this book online, and bonus content  
<https://www.sean.co.uk/scratch>

Scratch Wiki: a reference guide to Scratch  
[https://en.scratch-wiki.info/wiki/Scratch\\_Wiki](https://en.scratch-wiki.info/wiki/Scratch_Wiki)

Raspberry Pi forums, including one dedicated to Scratch  
<http://www.raspberrypi.org/forums/>

To download the examples in this book, visit [www.ineasysteps.com](http://www.ineasysteps.com) Select **Free Resources** and then choose **Downloads**.

## Acknowledgements

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <https://scratch.mit.edu>

Special thanks to Mai T. Vu, J.D. and the students in her programming class at Casey Middle School in Boulder, Colorado; Darrell Little; and Laurence Molloy, who all helped with testing projects for this updated second edition.

Thank you to everyone who read the first edition and sent such lovely feedback! It's been amazing to see your projects!

Sense HAT photos are courtesy of Raspberry Pi. Image of micro:bit courtesy of micro:bit Educational Foundation at [microbit.org](http://microbit.org). Thank you to those who helped with permissions and research queries, including Mitchel Resnick; Liz Upton, Eben Upton, Lucy Hattersley and Helen Lynn of Raspberry Pi; Jan Boström; Tim Benson; and Mike Cook.

Extra special thanks to Karen McManus, who was a fantastic help throughout this project, in particular with indexing and layout into the In Easy Steps style; and to star playtester Leo.

## About the author

Sean McManus is an expert technology writer and Scratch enthusiast. He writes for The MagPi magazine and his other books include **Cool Scratch Projects in easy steps**, **Coder Academy**, **Raspberry Pi For Dummies**, and **Web Design in easy steps**. Visit his website at [www.sean.co.uk](http://www.sean.co.uk) for additional resources.

## A

Add Extensions	27, 93
adding sprites	48-49
adding scripts to sprites	17
alphabet sprite	124
anagrams	202
animation	49-50, 118, 129, 132-133
apple banana.	
See blocks, Operators, join apple banana	
arrow keys	52
asking questions	109
average	116

## B

backdrops	16, 47
background music	151, 160
Backpack	10-11
bitmap art	76
blocking paths	199
blocks	11, 14
color of	37
Control	18
create clone of myself	70
delete this clone	70
forever	37-38, 50, 80-81
if	64, 81
if... then... else	57-58, 114-115, 117-118
repeat 10	35-36, 45, 96, 111, 126, 207
repeat until	54-56, 105-108, 126-127
stop all	73, 146, 161
wait 1 seconds	18, 29, 50, 68, 79, 118, 201
wait until	80, 201
when green flag clicked	41, 44, 47, 63
when I receive [message]	103-104, 132
when I start as a clone	70-71
when space key pressed	51
Events	
broadcast [message]	103, 155
broadcast [message] and wait	103-104
when green flag clicked	41, 44, 47
when I receive [message]	103
when loudness > 10	200
when [space] key pressed	51
when this sprite clicked	53
Looks	11, 15, 18
change [color] effect by 25	88-89, 133, 154-155
clear graphic effects	88
costume number	132

go forward 1 layers	78, 93
go to front layer	68, 78, 93
hide	37, 44, 71, 200
next costume	87, 118
say Hello! for 2 seconds	18, 58, 115
set color effect to 0	47, 72, 88, 154
set size to 100%	49, 63, 73, 82
show	68, 200
switch costume to [costume name]	54, 87, 118, 121
micro:bit	
A button pressed?	181
clear display	181
display [pattern]	181, 188
display text Hello!	181, 185
tilt angle front	181
tilted any?	181
when A button pressed	181, 188
when moved	181
when pin 0 connected	181
when tilted [direction]	181
Motion	11, 14, 26, 34
change x by 10	26, 50, 65
change y by 10	26, 65
glide 1 secs to x:0 y:0	26, 198
go to [random position]	38, 67, 170
go to x:0 y:0	19, 24-25, 29, 63
if on edge, bounce	68, 169
move 10 steps	14, 17-19, 31, 34-35, 46, 82
point in direction 90	19, 31, 44, 107
point towards	31
set rotation style	33
set x to 0	26
set y to 0	26
turn clockwise 15 degrees	14, 17, 31, 34-35
turn counter-clockwise 15 degrees	31
Music	11, 85, 108
change tempo by 20	92
play drum 1 for 0.25 beats	92
play note 60 for 0.25 beats	92-96, 108
rest for 0.25 beats	92
set instrument to [name]	92, 108
set tempo to 60	92, 108
tempo	92
My Blocks	122-123, 125, 183, 193, 203
Operators	
*	107
/	116, 208
<	72
=	114, 126
>	81, 106, 162

- and 55
- apple contains a? 143
- join apple banana 110, 112-113, 162
- length of apple 141
- letter 1 of apple 139
- not 141, 204
- or 55
- pick random 1 to 10 67, 112, 121
- Pen 27-30, 34, 44, 192-193, 198
  - change pen color by 10 28, 36, 38, 198
  - change pen size by 1 28
  - erase all 27
  - pen down 27, 29
  - pen up 27, 29
  - set pen color to 0 28
  - set pen color to [color] 28-29, 193
  - set pen size to 1 28-29, 193
  - stamp 27, 127
- Sense HAT
  - clear display: 171
  - display character A 171
  - display [pattern] 171
  - display text Hello! 171
  - humidity 173
  - joystick pushed up 172
  - pitch 173
  - pressure 173
  - roll 173
  - set background to 171
  - set colour to [color] 171
  - set pixel x:0 y:0 to [color] 171
  - set rotation to 0 degrees 171
  - temperature 173
  - when joystick pushed up 172
  - when shaken 172
  - when tilted forward 172
  - yaw 173
- Sensing 165
  - answer 109-110
  - ask [What's your name?] and wait 109-110, 113
  - backdrop# of Stage 170
  - key [space] pressed? 64
  - loudness 170
  - mouse x 91
  - mouse y 91
  - reset timer 105-107
  - timer 101-102, 105-107, 116, 200
  - touching? 69, 177
  - touching color? 56-58, 193, 199
  - username 162
  - x position of Sprite1 170
  - y position of Sprite1 170
- Sound
  - change [pitch] effect by 10 15, 85
  - change volume by -10 85
  - clear sound effects 85
  - play sound [sound name] until done 15, 85
  - set [pitch] effect to 100 85
  - set volume to 100% 85
  - start sound [sound name] 85-86, 161
  - stop all sounds 85, 161
  - volume 85
- Variables 43
  - add thing to [list name] 136
  - change [variable name] by 1 43, 46, 80
  - delete 1 of [list name] 136
  - delete all of [list name] 136
  - hide list [list name] 137
  - hide variable [variable name] 43
  - insert thing at 1 of [list name] 136
  - item 1 of [list name] 137
  - item # of thing in [list name] 137, 139-141, 143, 145
  - length of [list name] 137, 197
  - [list name] contains thing 137, 141-142
  - my variable 99
  - replace item 1 of [list name] with thing 136
  - set [variable name] to 0 43, 63, 109, 116
  - show list [list name] 137
  - show variable [variable name] 43
- Video Sensing
  - set video transparency to 50% 165
  - turn video on 165, 168
  - video motion on sprite 165-166, 169
  - When video motion > 10 165
- Blocks Palette 10-12, 17-18, 37, 206
- bouncing 68
- bracket 58
- brightness 88
- broadcasts 103, 159, 208
- browser 9
- bugs 207-208

## C

- cat 11, 29
- chat program 197
- checking for movement 169
- checking input 141
- cleanup 206
- cloning sprites. See sprites, cloning
- Cloud Data Log 152
- cloud variables 43, 152, 162
- Code Area 10-11, 17-19
- Code tab 29
- collision detection 8, 58, 69, 80, 159, 177, 179
- color 28
- graphic effect 88

comments	206, 209
community	209
Control. See blocks, Control	
coordinates	24-26, 64-65, 93
copying blocks. See duplicating scripts	
copying costumes. See duplicating costumes	
copying scripts. See duplicating scripts	
copying sprites. See duplicating sprites	
costumes	15, 17
adding	87, 124
copying. See duplicating costumes	
drawing	128-129
Costumes Area	87, 128
Costumes tab	74, 87

## D

dance	117-118
death sequence	155
debugging	206-208
deleting	
lists	135
projects	21
scripts	65
sprites	99
variables	135
detecting	
sprite position	195
walls	199
difficulty, adjusting	82, 153, 201
direction	31, 66
division	116
dragging blocks	17
drawing	
a clock	101-102
a hexagon	35-36
a house	29-30
an ice cream	77
a robot	128-129
a spiral	44-46
a square	30, 34-35
on the Stage	27
patterns	36
sprites	74-76, 128-129, 149-150
using directions	34
drum	89
duplicating	
costumes	128
scripts	18, 57, 65
sprites	70

## E

electronics	181, 189
energy meter	72-73
errors	8, 206-208
Events. See blocks, Events	
examples	
Abstract Artist	198
Anagrams Quiz	202-204
Balloon Floater	182
blinking LED	189
clap-o-meter	170
Donut Chaser	166-167
drawing a house	29-30
drawing a spiral	42
drum	89
Evil Robot	120
Feeding Time	174
ghost (broadcasts)	104
Going Batty	168-169
guitar	86
Hangman	120
hexagons pattern	36
Keepy-Uppy	192-193
London Bridge	96
loop counters	131
Maze Mania	199
Penguin Patter	197
Photo Safari	200-201
Quiz Break	98-100
Rainbow Painter	37-38
Shop Cat	68, 194-196
simple Scratch program	8
singer	90-91
Space Opera	84
Space Swarm	148
Spiral Rider	40
Super Dodgeball	60, 194
synthesizer	93
teleport	154
victory dance	117-118
video direction	166-167
walking cat	17
writing on the Stage	125-127
Explore	209
extensions	10, 27, 206
Music	92, 108
Pen	27-28, 44, 183, 192
Sense HAT	171-174
Text to Speech	122-123
Video Sensing	165-169
direction	166

## F

feedback	209
File menu	11, 20
firing	157-159
fisheye	88
flying	168
football	192-193

## G

Game Over	57-58, 73, 146, 150, 160-161, 201
General Purpose Input/Output	189-190
gestures	12
ghost effect	47, 72, 88, 155
Google Chrome	9
GPIO	189-190
graphic effects	47, 88
green flag	22, 41, 46, 72. <i>See also</i> blocks, Events, when green flag clicked
grid reference	24

## H

hat blocks	41, 51, 53, 70
hide a sprite	37
high score	152, 162

## I

insert blocks	19
iPad. <i>See</i> tablet devices	

## J

joining text	110
Join Scratch	9
joystick	172, 176

## K

keyboard	control 10, 12, 51-52, 64-65, 154, 187, 192, 195
----------	--

## L

left-right	32
Lego Mindstorms EV3	10
Lego WeDo 2.0	10
light emitting diodes (LEDs)	171, 179, 189
limiting variables	81
lists	135-137, 197, 202
deleting	135
importing and exporting	137, 203
lives	153
Looks. <i>See</i> blocks, Looks	
loop counters	131
loops	35-36, 45
different verses	96
loudness	200
lowercase	126

## M

Make a List	135, 137
Make a Variable	42
maximum value	81
maze	199
Maze Generator	199
menus in blocks	43
micro:bit	10, 164, 180-183, 185, 187-188
buttons	188
LEDs	188
tilting	186
tilt readings	182
microphone	165, 170
missile	150, 156-157, 159
MIT Media Lab	210
mobile phone	10
Morse code	190
mosaic effect	88, 154
Motion. <i>See</i> blocks, Motion	
mouse pointer	38
moving sprites. <i>See</i> blocks, Motion	
multiplication	107, 112
music	92-96
background	151, 160
beats	95
notes	94-96
rests	95
sheet music	94
My Blocks. <i>See</i> blocks, My Blocks	
My Stuff	209
my variable	99

## N

nested loop 36  
new sprite 48

## O

opening projects 21-22  
Operators. See blocks, Operators

## P

Paint Editor 74-77, 129, 150, 192  
paint new sprite 74-76  
password 9  
Pen. See blocks, Pen  
physical computing 164  
pipette 56, 149, 193  
pixel art 76  
pixelate 88  
Print Screen key 201  
privacy 9, 152, 165  
program 8, 17-19  
programming 8  
programming language 8  
project 20-22  
  deleting 21  
  saving 20-21  
  sharing 209  
Put back button 21

## Q

questions, asking 109

## R

random  
  direction 67, 69, 158, 169  
  list items 197  
  numbers 67, 112, 155  
  pattern 198  
  pause 79, 200  
  screen position 67, 69, 78  
  size 169  
Raspberry Pi 8, 164, 171-172, 189-190, 210  
  pins 189-190  
recovering deleted projects 21  
remix 22

removing scripts 65  
reporting boxes 62  
resistors 189-190  
rotating sprites 32-33  
rotation style 32-33  
running a script 19

## S

saving 9, 12  
  as a copy 20  
  projects 20-21  
  Save Now 20  
  Save to your computer 20  
score 61, 63, 80, 116, 152, 159, 194, 201  
  high score 152, 162  
Scratch  
  definition 8  
Scratch 1.4 10  
Scratch 2 10  
Scratch 3 10, 12, 15-16, 33, 67, 74, 85,  
  99, 106, 110, 122, 137, 143, 145, 164, 171  
Scratch account 9  
ScratchJr 13  
scripts  
  creating 17  
  defined 17  
  deleting. See deleting scripts  
  duplicating. See duplicating scripts  
See inside button 21-22  
Sense HAT 10, 164, 171, 174  
Sensing. See blocks, Sensing  
sensors 171  
Share 21  
shared projects 22, 41, 152  
sharing projects 209  
Sign in 9, 21  
soccer 192-193  
Sound. See blocks, Sound  
sound effects 151, 156, 161  
  rising 108  
sound library 90  
sounds 17  
  adding 90, 151  
  detecting 165  
  editing 91  
  playing 85  
Sounds Area 91  
Sounds tab 90  
sprite library 48-49  
Sprite List 10-11, 17, 56

sprites	17
adding	48-49
cloning	70-71, 168, 177
copying	70-71
costumes. See costumes	
customizing	74
defined	11
deleting	99
detecting position	195
drawing	74-76, 128, 149-150
hiding	200
random	48
revealing	78-79
touching	179
uploading	48
Stage	10-11, 16, 24, 27, 47, 107
edge, positioning sprites	178
reporting boxes	62
stamp. See blocks, Pen	
starting a script. See running a script	
stop a program	38
Stop button	38, 105
strength	61, 72, 80-81
strings, joining	110
surprise sprite	48

## T

tablet devices	8, 10, 12, 53, 84, 98, 111, 120, 148, 192
teleport effect	154
testing	17, 69
text, joining	110
Text to Speech	122-123
tilt sensors	175, 186
timer	101-102, 200
timing	82
random	79
touch control	53
touchscreen. See tablet devices	
transparency	47, 72
video	165-168
Turbo mode	208

## U

undeleting	
projects	21
Undo	91
Untitled projects	11, 20

uploading	
projects	20
sprites	48
uppercase	126
username	9, 162

## V

validating input	141
variables	42-
43, 61-62, 96, 99, 109, 130-131, 152, 206	
cloud	43, 152, 162
deleting	135
hiding	100, 131
in sound effects	108
limiting	81
naming	43
reporting boxes	62, 69
Variables blocks. See blocks, Variables	
vector art	74-75
vehicles	194
velocity	186
video	165-169
video direction	166-167
volume	85

## W

wait, random	79
walls	199
webcam	165-169
website	22, 162, 209-210
whirl	88

## X

x	24, 26, 64
x position, finding	170

## Y

y	24, 26, 64
y position, finding	170

## Z

zig-zag	157
---------	-----

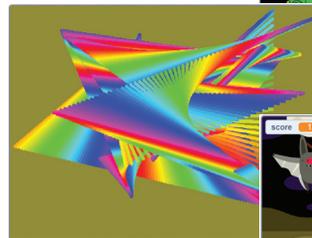
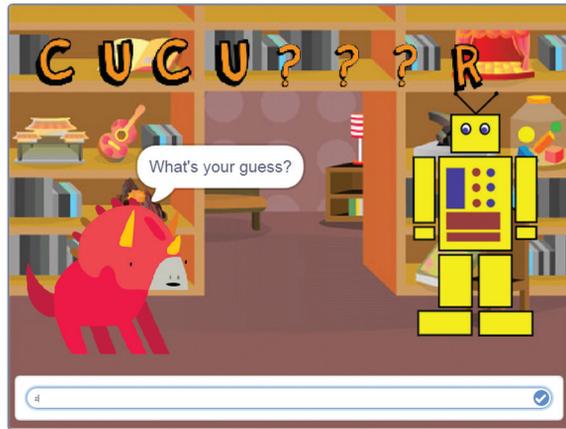
# Learn to program with Scratch

Scratch makes it easy to create your own games, animations, music, art or applications. It's the perfect way to learn programming because it takes away a lot of the complexity. That means you can focus on having great ideas and bringing them to life. With this book as your companion, you'll learn how to:

- Build and share exciting Scratch projects.
- Create addictive arcade games, quizzes and word games.
- Make computer-generated art.
- Play music and compose your own tunes.
- Use variables, lists, loops, broadcasts, and operators to create sophisticated software.
- Avoid common programming pitfalls and bugs.
- Interact with webcam video, the Raspberry Pi Sense HAT, and the micro:bit.

This popular book has now been updated to cover Scratch 3 so you can use all the latest features.

With no prior knowledge needed, we'll take you from scratch to Scratch!



# Why choose this book?

## It's written in plain English

Put off by the jargon? Don't worry, we keep things straightforward.

## It's easy to follow

Clear, step-by-step instructions make learning simple.

## It's fully illustrated

We don't just tell you how to do it, we also **show** you how.

## It's in full color

This book's printed in color to make it simpler to use and easier on the eye.

## And it's fantastic value

Do we need to say any more?

£10.99 UK / \$15.99 US



The Scratch programming language is widely used in schools and on the Raspberry Pi. Its drag-and-drop commands make it an ideal language for all ages to learn to program. And this popular book, **Scratch Programming in easy steps**, now fully updated for Scratch 3, is packed with ideas and games that illustrate what's possible with Scratch. This book introduces you to Scratch essentials and then walks you through the commands to create games and animations that'll impress. Learn to:

- Build games that require skill, knowledge or quick fingers
- Add music
- Create eye-catching visual effects
- Keep score
- Avoid common pitfalls and learn how to fix bugs

**Use Scratch Programming in easy steps to get creative and become a super Scratcher!**

## Let these icons make it even easier



to spice up your learning



highlights something worth remembering



wards you off potential danger!



indicates a new or changed feature

Categories: Children's>Hobbies>Computers  
Children's>Education>Technology  
Computers> Programming

[www.ineasysteps.com](http://www.ineasysteps.com)