



000

0. Introduction 5

001

1. A beginner's guide to Scratch..... 7

1.1 Scratch basics 9

1.2 Moving sprites..... 15

1.3 Animation (loops)..... 18

1.4 Maths cat..... 23

1.5 Artificial intelligence 29

1.6 Control..... 35

1.7 Scratch games..... 44

What next? 50

010

2. Greenfoot on the Raspberry Pi..... 52

2.1 Interacting with Greenfoot 53

2.2 Movement and key control..... 58

2.3 Detecting and removing actors, and making methods.. 62

2.4 Saving the world and playing with sound..... 66

2.5 Adding a randomly moving enemy 69

Useful links..... 71

011

3. Experiments in Python 72

3.1 Getting to grips with Python 73

3.2 MasterPy..... 86

3.3 Roman Numerals & data manipulation..... 89

3.4 Getting artistic..... 94

3.5 Simulations and games 100

3.6 Limited resources - memory & storage 106

3.7 Accessing the web - providing a weather forecast..... 108

This is only the beginning - where do we go from here? 111

100

4. Human-computer interfacing	113
4.1 Twitter	115
4.2 Email application	116
4.3 Remote Procedure Call	118
4.4 Web applications.....	120
4.5 General Purpose Input/Output (GPIO)	125

101

5. GeoGebra: fun with maths!	136
5.1 Getting to know GeoGebra with René Descartes	137
5.2 A tale of two bridges	143
5.3 Pistons and cranks - linkages in action.....	150

110

6. The Linux Command Line	152
6.1 Commands are just programs	153
6.2 Command syntax and file structure	155
6.3 The superuser	161
6.4 Creating and destroying files and directories.....	163
6.5 Remote access to the Raspberry Pi	166

111

7. What next?	169
----------------------------	------------

This manual was brought to you by...

This manual is a bit different. It was written entirely by unpaid volunteers, all of whom are keen to share their expertise and enthusiasm for computing with as many people as possible.

What all of these contributors have in common, apart from a youth spent mainly indoors in front of ZX Spectrums and Commodore 64s, is that they're all members of the organisation **Computing at School (CAS)**. To find out more about CAS and its work promoting the teaching of computer science, head over to <http://www.computingschool.org.uk>

Manual Contributors

Introduction by Andrew Hague
A beginner's guide to Scratch by Graham Hastings
Greenfoot on the Raspberry Pi by Michael Kölling
Experiments in Python by Andrew Hague
Human-computer interfacing by Ben Croston
GeoGebra: fun with maths! by Adrian Oldknow
The Linux Command Line by Brian Lockwood
Where next? by Clive Beale

Manual Production

Karl Wright, Robert Cruse and Paul Kingett of Publicis Blueprint

Digital Contributors

The following people offered contributions not covered in the manual, but available online and on your SD card.

Scratch Pong by Bruce Nightingale
Caesar Cipher by Brian Starkey
Fly by Alan Holt

Special Thanks

Martin Richards (University of Cambridge)
Simon Humphreys (Computing at Schools)
Alex Bradbury (University of Cambridge/Raspberry Pi Foundation)
Liz Upton (Raspberry Pi Foundation)
Eben Upton (Raspberry Pi Foundation)

Hello, Raspberry Pi users

Chapter 0

Congratulations! You have in your possession a Raspberry Pi. A small but powerful computer designed to help you understand and explore the almost-magical world of computing. Use it wisely; it's an object of great power.

What is the Raspberry Pi?

The Raspberry Pi is a computer, very like the computers with which you're already familiar. It uses a different kind of processor, so you can't install Microsoft Windows on it. But you can install several versions of the Linux operating system that look and feel very much like Windows. If you want to, you can use the Raspberry Pi to surf the internet, send an email or write a letter using a word processor. But you can also do so much more.

Easy to use but powerful, affordable and (as long as you're careful) difficult to break, the Raspberry Pi is the perfect tool for aspiring computer scientists. What do we mean by computer science? We mean learning how computers work so you can make them do what you want them to do, not what someone else thinks you should do with them.

And who do we mean by computer scientists? We mean you. You may finish this manual and decide you want to be next Tim Berners Lee, but even if you don't, we hope you have fun, learn something new and get a feel for how computers work. Because no matter what you do in life, computers are bound to be part of it.

Notes:

What am I going to learn?

This user manual is different. Don't expect a dry-as-dust description of how to plug things in or where to find your serial number. And you certainly won't learn how to create a spreadsheet or a presentation. That's really not computer science, it's something else entirely.

Instead, think of this manual, along with your Raspberry Pi, as a "computer science set". Have you ever been given a chemistry set? With a chemistry set, you can make lots of bangs, smells and odd-coloured goop to learn all about elements, molecules and compounds.

We're not going to make odd-coloured goop, but we will use experiments to discover how to program a computer to create your own games and animations, how to make graphics appear on screen just by typing in the right code (just like the developers of your favourite games do), how to get a cat to do your maths homework for you, and much more.

By doing all this, you will learn the basic principles of computer science. And that's your first step on the journey to becoming a real computer programmer, a games developer, an über-hacker just like in the movies (only cooler and staying strictly within the law) and many other things besides. Exactly what, depends on you.

Who is this manual for?

When we wrote this manual, our aim was for it to be suitable for most people of eight years and older. But that doesn't mean it's for eight year olds. This book is for anyone and everyone who is curious to know more about computing and creating computer programs. If you don't have computer-programming experience but you want to get some and you're looking for a place to start, this is it.

We begin the manual with some relatively easy experiments in computer science. Things then get progressively more challenging with each successive exercise. Try to spend time with each experiment and, once you've got an exercise doing what the manual says it should, feel free to change the code to see what happens: it's one of the best ways to learn.

Will I break it?

You can't break your Raspberry Pi by doing any of the experiments in this book, but you might just surprise yourself with what you can achieve. You will be working through and learning genuinely difficult but exciting concepts, and laying the foundations for even more exciting discoveries in the future.

So, without further delay, have everyone in the room stand back: we're going to do computer science!