9 Awesome Projects
written especially for young people!

Adventures in Minecraft®

```
     ointOnCircle function to fi
hourHandX, hourHandY = findPointOnCircl

# draw hour hand
mcdrawing.drawLine(clockMiddle.x, clo
                   hourHandX, hourHandY
                   block.DIRT.id)

#minute hand
# what angle would a minute hand poi
minHandAngle = (360 / 60) * minutes
# use findPointOnCircle function to f
minHandX, minHandY = findPointOnCircle(

# draw minute hand
mcdrawing.drawLine(clockMiddle.x, clockM
                   minHandX, minHandY, cl
                   block.WOOD_PLANKS.id

#second hand
# what angle would a second hand point to
secHandAngle = (360 / 60) * seconds
# use findPointOnCircle function to find the
secHandX, secHandY = findPointOnCircle(clockM
                                       SEC_HAND
```

Martin O'Hanlon
David Whale

WILEY

Adventures in Minecraft®

Adventures in Minecraft®

Martin O'Hanlon and David Whale

WILEY

This edition first published 2015

Registered office

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

All photographs used in this work courtesy of S K Pang, 2014.

A catalogue record for this book is available from the British Library.

ISBN 978-1-118-94691-6 (paperback); ISBN 978-1-118-94685-5 (ePub); 978-1-118-94684-8 (ePDF)

Set in Chaparral Pro 10\12.5 by TCS\SPS

Printed and bound in Great Britain by Bell & Bain

For my wife Leonie, without you, this would never have been.

—Martin.

For my wife Gail, for putting up with me while I constantly played Minecraft.

—David.

Publisher's Acknowledgements

Some of the people who helped bring this book to market include the following:

Editorial

Publisher: Barry Pruett

Associate Publisher: Jim Minatel

Executive Commissioning Editor: Craig Smith

Acquisitions Editor: Aaron Black

Project Editor: Sydney Argenta

Copy Editor: Grace Fairley

Technical Editor: Cliff O'Reilly

Editorial Manager: Mary Beth Wakefield

Senior Project Editor: Sara Shlaer

Editorial Assistant: Jessie Phelps

Illustrator: Sarah Wright

Marketing

Marketing Manager: Lorna Mein

Marketing Assistant: Polly Thomas

Minecraft Consultants

Zachary Igielman

Lauren Trussler

Sam Whale

Ben Foden

Ben Ramachandra

Ria Parish

About the Authors

MARTIN O'HANLON has been designing and programming computer systems for all of his adult life. His passion for programming and helping others to learn led him to create the blog <Stuff about="code" /> (www.stuffaboutcode.com) where he shares his experiences, skills and ideas. Martin regularly delivers presentations and workshops on programming Minecraft to coders, teachers and young people with the aim of inspiring them to try something new and making programming fun.

DAVID WHALE writes computer programs for devices you wouldn't imagine have computers inside them. He was bitten by the computer programming bug aged 11 when he was at school, and still thoroughly enjoys writing software and helping others to learn programming. He runs a software consultancy business in Essex, but also regularly volunteers for The Institution of Engineering and Technology (The IET) helping in schools, running weekend computing clubs, judging schools competitions, and running programming workshops for young people at community events all around the UK. You can follow his adventures on his blog at http://blog.whaleygeek.co.uk.

Acknowledgments

Many people are involved in producing a book, too many to mention in this small space. We would both like to give our special thanks to the following people:

- The staff at Mojang, for designing such a great game, and their genius and insight in making the game programmable. Without this insight, this book would not have been possible.
- The Raspberry Pi Foundation and the open source community, without which there wouldn't be a Raspberry Pi or a Bukkit server, both of which are vital platforms that enabled this book to be written for a wide audience.
- Our testers and young Minecraft experts, Zachary Igielman, Lauren Trussler, Sam Whale, Ben Foden, Ria Parish, who tried our programs and provided really useful feedback, without which we would never have known if we were pitching the book correctly to the target age group.
- Mr S.K.Pang, for all his advice and help with selecting the right electronic components for our projects, and for helping us to make it possible to easily and cheaply control electronic circuits from the PC and the Mac.
- Cliff O'Reilly, for making sure everything was technically right, and for testing everything 3 times for us (once each on the 3 different computer platforms).
- Sarah Wright, for the truly amazing illustrations throughout this book. They are beautiful pieces of visual artwork, and cleverly and perfectly capture the concepts being presented in each adventure.
- Ben Ramachandra, the young lad at the Christmas 2013 Fire Tech Camp event at Imperial College, London: You were so determined to follow the Python course entirely in Minecraft, which was the moment that caused the idea for this book to spark into existence!
- Roma Agrawal, structural engineer for The Shard, UK: for her suggestions and links to inspiring tall buildings in Adventure 4 and the Bonus Adventure – let's hope we see some amazing creations from our readers!
- Last, but not least, we would like to thank Carrie-Anne Philbin, for having the vision and determination to write her first book *Adventures in Raspberry Pi*, without which the Adventures series of books would not exist—now, see what you've started, Carrie-Anne?!

Contents

Adventure 5

Interacting with Electronic Circuits

Adventure 6

Using Data Files

Adventure 7

Adventure 8

Adventure 9

The Big Adventure: Crafty Crossing 237

Appendix A

Where to Go from Here 273

Index 279

FOREWORD

It was another busy IT lunch club at school. I've always been happy to give up my lunch break to allow the pupils an opportunity to catch up on homework, research on the web, email their teachers and print out assignments. As I walked along the rows of computers I saw the "hard work" in progress: multi-coloured birds being catapulted into towers, aliens being zapped into submission and one pupil trying to reverse a car round a car park—at least that one might have some educational benefit!

My philosophy has always been that the more time kids spend on the computers, the more natural the computers become to them; even a game will improve their hand/eye coordination and familiarisation with the keyboard. They become adept at logging in, starting the browser, and finding the shortest number of key terms possible for the search engine. They can be into the room and onto a game in the time it takes me to prop the door to the IT lab open.

As I patrolled the aisles, ever eager to find the one child in a dozen that was actually doing something productive, I spotted a pupil diligently stacking what looked like little 3D cubes on top of each other in their game window. This pupil seemed enthralled, and as I paused, curious as to why this comparatively basic endeavour had captured this child's attention, a house emerged from the little collection of blocks that had been stacked.

This was my introduction to Minecraft: a little architect in the making, a pupil who could hide himself away in the chaos of an IT room at lunch and disappear into a world of his own creation: a virtual home he designed, created and explored. Kids' games are infectious little things and once it reaches a critical mass, once enough have bought into an idea, they're all at it. Once it gains momentum, they need to get on board to keep up with the rest of the crowd and by summer they were all on Minecraft.

However, unlike a lot of the passing fads I soon discovered that there was a lot more to this game than the rest. I found out students can automate their constructions. By wiring up their houses they can create their own lighting systems and elevators. They can devise hidden staircases that appear from inside walls. In fact they can create practically any contraption they can imagine. A history lesson overspills into lunch club as they build the castles they just learned about, biology class means they're all competing to build the best virtual skeleton, and a geography lesson destroys all that hard work when their simulated volcano erupts lava over both.

When our school purchased a MinecraftEdu account, my pupils could work collaboratively. They've built giant replicas of the school logo, they devised a maze, and held a school contest to escape it. They measured the classrooms and built the replica school inside Minecraft.

More importantly though, this has given me a virtual classroom; a strange state of affairs where I fly above the avatars of my class, instructing them on the finer details of logic. I start by building an exclusive-OR gate using the blocks of this world, and then set them a team-work task of building a calculator by wiring several of these circuits together.

To my delight I learned that Minecraft is also available on the Raspberry Pi. But more importantly, much more importantly, it can be programmed in the Python programming language. My pupils began work on it immediately. Through for loops and block placement we created platforms and trampolines, fireballs and drag races—games within a game that did things even the original game designers had not imagined—they had made the world their own.

One of my pupils, left to his own devices, created an auto-house script that allows the user to generate a home around themselves; another created a mine-sweeper puzzle. Writing computer programs was suddenly something they were all desperate to learn, driven by the need to build and automate larger and more complex tasks inside the Minecraft world, and to compete and keep up with what all their friends were inventing.

Our school now has a purpose-fitted Minecraft network with its own server, built by the pupils, for the pupils. With the help of their network they can submit work for grading in exactly the same way that their coursework is assessed. They can compete against each other to code individual parts of a larger world and then explore it together.

This book is a first step into that amazing world. It teaches coding, and it teaches it using one of the most popular games on the planet! Children are gradually drawn outwards from the Minecraft world they are used to, and into the world of computer coding; they have a real purpose to learn coding, to achieve things that they are desperate to learn quicker than their friends can.

It is the ability to see an instant visualisation of the results of their code that makes it so powerful. Programming can be dry and uninteresting to pupils, unable to relate to the outcome of their code when the results are often just text on a screen. Minecraft enables them to see the outcome of their code in a context they understand and enjoy.

I genuinely believe that Minecraft has the power to influence an entire generation; to encourage coding for what it should be—fun and hugely rewarding. So, don't waste any time, get started on your own Adventures in Minecraft with Martin and David to guide you along the way. Be creative, stay excited, build something awesome, and learn how to code along the way!

Ben Smith BSc (Hons)

Head of Computing,

ArnoldKeqms, Lytham St. Annes.

Introduction

ARE YOU AN adventurer? Do you like to try new things and learn new skills? Are you a huge fan of Minecraft? And would you like to push the boundaries of what you can do in Minecraft by learning how to write computer programs that interact with your game, and amaze your friends with your creativity and magic? If the answer is a resounding "Yes!" then this is the book for you.

What Is Minecraft?

Minecraft is a sandbox indie game, where you build structures, collect items, mine minerals and fight monsters in order to survive. It appears to you as a 3D virtual world made of different types of blocks, each block having its own place inside the grid layout of the 3D virtual world. Figure 1 shows an example of the Minecraft world.

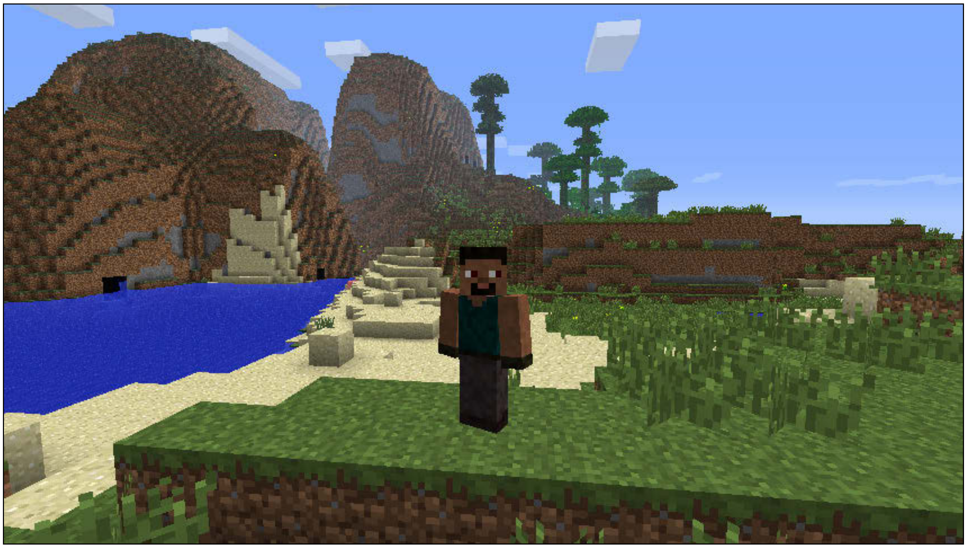

FIGURE 1 The Minecraft world

The Virtual World

In a sandbox game, you are a player inside a virtual world (a sandbox with very distant edges, like a playpen filled with sand). Instead of being offered levels in a pre-set order, you roam around the virtual world and make your own choices about what goals you want to achieve and how to set about them. Because you are making your own choices right from the start, sandbox games have limitless possibilities. You make up your own stories and move through the 3D world, learning new skills and features by discovering them by chance and experimentation.

In Minecraft your player, or avatar, is called Steve. You direct Steve through the sandbox virtual world to achieve whatever mission you decide. If you are successful in surviving your first night against the monsters, you can follow your own enthralling missions to interact with other participants of the game and build huge structures limited only by your imagination.

A sandbox game allows you, the player, to make your own decisions about playing the game, rather than being forced down a specific route by the game designers. You can read more about this type of game design here: http://en.wikipedia.org/wiki/Open_world. There is a little bit of mystery about why the player is called Steve, but you can read more about it here: http://minecraft.gamepedia.com/The_Player.

How Did Minecraft Come About?

Indie games are "independent video games", created by individuals or small teams. They are often developed without any funding or support from a games publisher. As a result of their independent nature, indie games are often more innovative than other, more mainstream games. According to Wikipedia, Minecraft was created by the Swedish computer programmer Markus Persson, who is known by the gamer tag "Notch". He first demonstrated Minecraft as an early version in 2009, and the first official release of the game took place in 2011. Notch founded a Swedish company called Mojang AB, which continues to develop the Minecraft game on many computer platforms, including PC, Mac, Raspberry Pi, Linux, iOS, Android, Xbox 360, Playstation and Wii.

You can find out more about the fascinating Minecraft story in a documentary film called *Minecraft: The Story of Mojang* (http://en.wikipedia.org/wiki/Minecraft:_The_Story_of_Mojang).

What Is Minecraft Programming?

This a book about computer programming—it uses Minecraft as a way to teach you about computer programming. If you are looking for some helpful tips on how to build structures and fight combat, there are some other great books on the market listed in Appendix A that will help.

By programming Minecraft, you make your gaming experiences even more exciting, creative, and individual. As you play the normal game, you follow the basic rules of the Minecraft game as set out by the game designers. By writing programs that interact with the Minecraft game world, you can make complex and repetitive tasks—like building huge streets of houses and large structures—automatic. You can make the game and the objects inside it behave in new ways, and invent new things that even the original creators of the game didn't think of. But most of all, you will learn a general skill—how to program using the Python programming language. You will then be able to apply this to all sorts of other things, not just Minecraft. Figure 2 shows a huge street of houses that was built automatically by a short Python program.

In a recent video about why all children should learn programming (www.youtube.com/watch?v=nKIu9yen5nc), Will.i.am is quoted as saying "great coders are today's rock stars". The new skills you learn while following the adventures in this book will make your Minecraft experiences more personal, more creative, more ambitious. Your new wizardry with programming will amaze your friends and fellow gamers and inspire them to ask you what magic you used to achieve such amazing feats. The answer, of course, is the magic of computer programming.

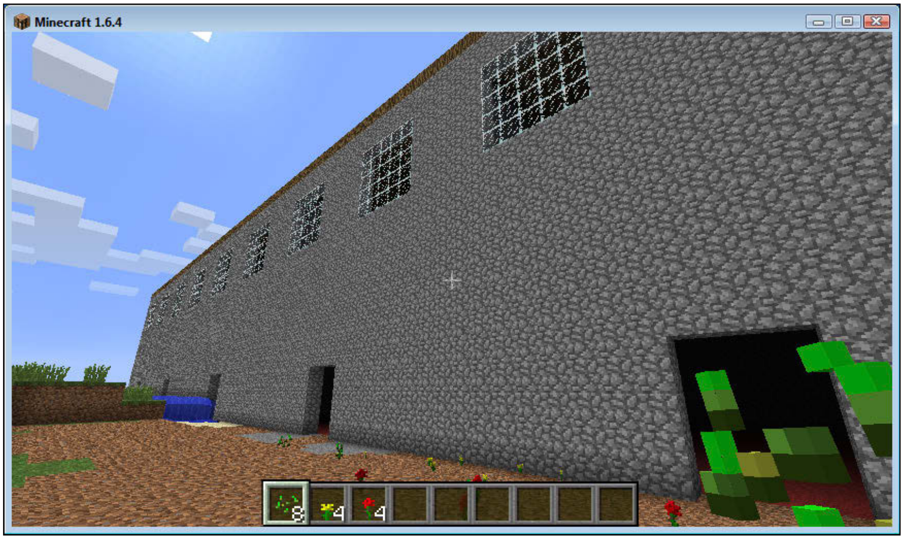

FIGURE 2 A huge street of houses, built by a 20-line Python program

Who Should Read This Book?

Adventures in Minecraft is for any young person who loves playing Minecraft, and would like to learn to program and do more with it. The Adventures series of books is aimed at readers in the age range 11–15, but some of the more challenging later adventures might be appropriate for older readers too. The earlier chapters have also been tested with readers as young as 8.

You might already be an expert in playing the game but find yourself getting frustrated by the length of time it takes to build new structures. Or you might want to find ways to extend the game by adding some additional intelligence and automation to the world. Whatever your reasons, this book will be your guide for a journey through Minecraft programming; and as every adventurer knows, your guidebook is the most important item in your backpack. Your trek will take you from simple beginnings, such as posting messages to the Minecraft chat, through learning the basics of programming Minecraft using the Python programming language, to discovering how to use your new computer programming skills to program your own exciting game inside Minecraft. By the end of your adventures you will have learned the skills you need to become a pioneer in Minecraft programming!

What You Will Learn

You will learn about many aspects of the Minecraft game and how to interact with Minecraft features through the Python programming language. You will discover how blocks are addressed in the 3D world using coordinates, how to sense the position of your player, how to create and delete blocks in the Minecraft world, and how to sense that a block has been hit by the player.

If you are using a Raspberry Pi, you will learn how to install and use the Minecraft programming interface that comes bundled with Minecraft Pi edition. If you are using a PC or a Mac, you will learn how to set up and run your own local Minecraft server using the community developed craft-bukkit server, and how to program it using the same Minecraft programming interface as the Raspberry Pi through the Raspberry Juice plug-in.

You will learn how to write programs in the Python programming language, from the very beginnings of a Hello Minecraft World program to the creation of and interaction with huge 3D objects that, thanks to your new Python programming skills, you can stamp with your own personality.

Using the free MinecraftStuff module of pre-written Python helper code, you will be able to enhance your ability to create both 2D and 3D objects out of blocks, lines, polygons and text.

Your adventures will not be limited to the virtual world of Minecraft though! We will introduce you to ways to connect Minecraft to electronic components, meaning that your Minecraft world will be able to sense and control objects in the real world. Thus, we give you a valuable secret: how to break out of the boundaries of the virtual sandbox world!

Minecraft has two main modes of working: Survival mode and Creative mode. You will be using Creative mode throughout this book. We won't be covering Survival mode (mainly because it's extremely frustrating when a creeper kills you just as you are watching your program running). There are many good books already on the market that explain how to survive the night in Minecraft, and we give links to those and other resources in Appendix A at the back of this book. However, any programs you create in Creative mode will also work in Survival mode.

What We Assume You Already Know

Because this is a book about programming with Minecraft and we want to focus on learning the programming aspects of Minecraft, we have to assume a few things about you the reader and what you already know:

1. You have a computer (a Raspberry Pi running Raspbian, a PC running Microsoft Windows, or an Apple Mac running Mac OS X), which meets the minimum requirements for running Minecraft, and is already set up and working.
2. You have a basic understanding of how to use your computer, such as using a keyboard and a mouse, using the menu system to start programs, and using application menus like File➪New➪Save.
3. You have a working connection to the Internet, and you know how to use a web browser to download files from the Internet.
4. If you are using a PC or a Mac, you already have a Minecraft user ID and a working copy of Minecraft installed.
5. You know how to play the Minecraft game, such as how to start it, how to move around, how to choose items from the inventory, and how to create and delete blocks in the world.

Because this is a book about programming Minecraft, we don't assume you have any prior knowledge about how to program. As you progress through your adventures, we will lead you through the steps needed to learn programming.

What You Will Need for the Projects

We have written this book to work on three commonly available computers: the Raspberry Pi running Raspbian, a PC running Microsoft Windows, and an Apple Mac running MacOS X. Minecraft is supported on other platforms too, such as a PC running various flavours of Linux, but we don't cover the set-up of those platforms in this book.

To make the set-up of the various parts simpler, we have prepared three starter kits, one for each of the supported computer platforms. You can download the correct starter kit for your computer from the Wiley website, and in your first adventure we provide step-by-step instructions about how to download and install these and get everything working. These starter kits include everything you need, except the actual Minecraft game itself. You'll be up and running in no time!

You will need an Internet connection on your computer in order to download the starter kits. Almost everything you need for the adventures is included in the starter kits. A few of the adventures have special requirements and we note these at the start of the adventure so you can get everything prepared before you start.

In Adventures 5 and 9, we show you how to connect small electronic circuits together to link the Minecraft virtual world to the real world. For this you will need to buy a small collection of electronic components, which are available from most electronics components stockists. (We provide some links to these in Appendix A.)

The Raspberry Pi has built-in input/output pins, so you can connect your electronic components directly to these. Because PCs and Macs don't include input/output pins, we have chosen a small affordable plug-in board that works via the USB connection of your computer for these projects. Again, there are links in Appendix A to outlets where you can buy this.

The most important things you need on this journey are your own excitement and enthusiasm for Minecraft, and some curiosity and willingness to experiment with your own ideas and push the boundaries of what you already know!

A Note for Parents and Teachers

We have split this book into separate self-contained adventures that you can treat as individual standalone projects, each of which focuses on one specific feature of Minecraft programming. The Python language is introduced gradually and progressively throughout each adventure; the early adventures are aimed chiefly at beginners, with the later adventures becoming more challenging and introducing more Python, stretching the reader a bit more.

Each adventure presents a practical project with step-by-step instructions (that readers can tick off as they complete them), delivered in a descriptive style, very much like a well-commented program listing. Detailed explanations appear in Digging into the Code sidebars that students can read later, meaning that they are not distracted from the progress of typing in and trying the programs.

Each adventure will probably take more than one session to complete, but they are all split into sections, with subheadings at logical points that could be used to provide a goal for an individual lesson, or an activity to be stretched over a number of sessions.

The Python language uses indents on the left-hand side of the program to represent code structure, and it is a case-sensitive language. Extra guidance from an adult may be useful sometimes with very young readers, to make sure they are being careful to use case and indents correctly, thus avoiding the possibility of them introducing errors into their programs. All of the programs are downloadable from the companion website, so if you have problems with indentation you can check our versions of the programs to see where you might have gone wrong.

Many schools have Python version 3 installed. At the time of writing, Mojang have not released a version of the Minecraft programming interface that works with Python version 3, so you should use Python version 2 as explained in Adventure 1.

How This Book Is Organised

Every chapter of the book is a separate adventure, teaching you new skills and concepts as you program and test the projects. The book is organised so that each adventure is a standalone project, but you might find it easier to work through them in order, as we build up your understanding of the programming concepts gradually throughout the book.

It is vital that you do Adventure 1 before doing anything else. This is because it shows you how to download and install everything you need, and to check that it all works properly. We introduce some basic steps in this adventure that you need to know how to do in all the other adventures, but will give you some reminders in the earlier adventures as you get started.

The first three adventures are written for beginners who have little or no programming knowledge, and we explain all the jargon and concepts as you work through them. In Adventures 2, 3 and 4, you cover the key parts of any good Minecraft game. These include *sensing* things that happen in the Minecraft world, doing some *calculations* with some simple maths, and making your programs *behave* differently, for example by displaying a message on the chat or automatically creating blocks in the world. You will use these three concepts of *sensing*, *calculating* and *behaving* throughout the book to build bigger and more exciting Minecraft programs!

Adventures 5 and 6 build on what you learned in the earlier adventures and explore some exciting ways of linking the Minecraft virtual world to the real world. You will experiment with the exciting topic of physical computing by building some small electronic circuits that cause things to happen inside Minecraft and respond to things that happen in Minecraft. There is an abundance of exciting ideas and games you could create using this as a basis! Adventure 6 looks at ways you can bring in large amounts of data from data files to save and duplicate large structures with a 3D "duplicating machine".

Adventures 7 and 8 introduce the free MinecraftStuff module, which makes it possible to use blocks to build lines, circles and other 2D shapes, and also some fantastic 3D spheres and pyramids. These can form the beginnings of huge structures that would be very hard to build by hand. Adventure 8 shows how you can add personalities to moving objects to give them their own intelligence. With these techniques, you can write some exciting "games inside a game" that will amaze your friends.

Adventure 9 draws on all the programming concepts and skills from the earlier adventures to create one final big project—an awesome game with scoring, and moving objects that you have to avoid or carry around with you. In this adventure, you also have the option to experiment with physical computing by using electronic components, allowing you to do things in the game by pressing buttons in the real world.

Appendix A ("Where Next") suggests a whole range of resources that you can use to extend and enhance your adventures, learn more about programming in Python and create even more awesome Minecraft programs based on what you have learnt throughout this book.

In Appendix B ("Quick Reference") we have included a comprehensive reference guide to the programming features used throughout the book, along with a reference to the programming statements that are specific to Minecraft, and a table of block types that you can build with. You'll find this is an invaluable reference section to help with all your own projects and inventions as well!

The glossary provides a handy quick reference to all the jargon and terminology we have introduced throughout the book, and is a collection of all definitions from each adventure.

The Companion Website

Throughout this book you'll find references to the Adventures in Minecraft companion website at www.wiley.com/go/adventuresinminecraft. The website is where you'll find the starter kits you will need to start programming in Minecraft, together with a collection of video tutorials we have put together to help you if you get stuck. Code files for some of the bigger projects can also be found on the website.

You will also find on the companion website a complete extra bonus adventure! With this adventure you will build a fully functional passenger lift simulation inside Minecraft allowing you to whizz up and down through the world. This bonus adventure is quite challenging and should exercise all of the skills that you have learned and practiced throughout this book, including controlling your lift with electronic circuits.

Our publisher (Wiley) has kindly allowed us to provide a reference appendix in a PDF format that you can download from the companion website. Keep it by your side as a handy reference as you work through these Minecraft adventures. You can also use it in any programming projects you embark on in the future. The Wiley website also includes a glossary. Although definitions are included in the adventures themselves, anytime you want to look up a word, you can go online.

Other Sources of Help

Computers are complex devices, and operating systems and software are changing all the time. We have tried to protect you and your adventures from future changes as much as possible by providing a downloadable starter kit in Adventure 1 that should give you most of what you need. However, if you run into problems or need specific help, here are some useful places to go:

Sign up for a user ID and downloading and installing Minecraft: http://minecraft.net

Play the Minecraft game: http://minecraft.gamepedia.com/Minecraft_Wiki

Raspberry Pi: www.raspberrypi.org

Microsoft Windows: http://support.microsoft.com

Apple Mac and Mac OS X: www.apple.com/support

The Python language: www.python.org

The IDLE programming IDE: https://docs.python.org/2/library/idle.html

Minecraft Pi edition: http://pi.minecraft.net

Craft-bukkit server: http://wiki.bukkit.org

Raspberry Juice bukkit plug in: http://dev.bukkit.org/bukkit-plugins/raspberryjuice

Conventions

You'll notice that there are special boxes throughout this book, to guide and support you. Here is what they look like:

DEFINITIONS

These boxes explain concepts or terms you might not be familiar with.

These boxes give you hints to make your computer-programming life easier.

These boxes contain important warnings to keep you and your computer safe when completing a step or a project.

These boxes feature quick quizzes for you to test your understanding or make you think more about a topic.

These boxes allow us to explain things or give you extra information we think you'll find useful.

These boxes point you to videos on the companion website that will take you through the tasks, step by step.

You will also find two sets of sidebars in the book. Challenge sidebars give you extra tasks you can accept if you want to take the project a bit further, perhaps by making changes or adding new features. Digging into the Code sidebars explain in a bit more detail some concept or feature of the program, to give you a better understanding of the programming language Python. These sidebars mean you can focus on getting the programs working first, and then read in more detail about how they work and ways you can extend them further once they are working.

When you are following our steps or instructions using code, you should type the code in exactly as we have described it in the instructions. Python is a language where the amount of space at the start of the line (the indent) is important to the meaning of the program, so take extra special care to make sure you put enough spaces at the left of each line. We have coloured the code listing boxes for you so that it makes it easier to see how much each line needs to be indented. Don't worry too much about it—we explain indenting in the early adventures when you first need to use it.

Sometimes you need to type a very long line of code, longer than will fit on a single line in this book. If you see ↩ at the end of a line of code, it means that line and the following line are part of a single line of code, so you should type them as one line, not separate lines. For example, the following code should be typed on one line, not two:

```
print("Welcome to Adventures in Minecraft by ↩
Martin O'Hanlon and David Whale")
```

If you are viewing this book on an e-reader, to make sure that the programs you type in are correctly laid out, please set your e-reader font size smaller. This is so that the program listings are not unnecessarily wrapped around the page margins, and to prevent errors being introduced into your programs.

Most adventures include a Quick Reference Table at the end to sum up the main programming statements or concepts. You can refer to these guides when you need a refresher. There is also a reference section in Appendix B, which shows you the most important programming statements for Minecraft and Python. We hope you will find this handy to refer to as you progress through your adventures.

Whenever you complete an adventure, you unlock an achievement and collect a new badge. You can collect the badges to represent these achievements from the Adventures in Minecraft companion website (www.wiley.com/go/adventuresinminecraft).

Reaching Out

In Appendix A you will find ways to take your Minecraft programming knowledge further, with lists of websites, organisations, videos and other resources. Many of these resources include forums where you can ask questions or get in touch with other Minecraft programmers.

You can also contact the authors by sending us a message through our websites:

Martin: www.stuffaboutcode.com

David: http://blog.whaleygeek.co.uk

Time to start your adventures!