Foreword by
Mitchel Resnick
MIT Media Lab

Sean McManus

Scratch Programming

covers Scratch 2.0 and Scratch 1.4

in easy steps

PLAIN ENGLISH

EASY TO FOLLOW

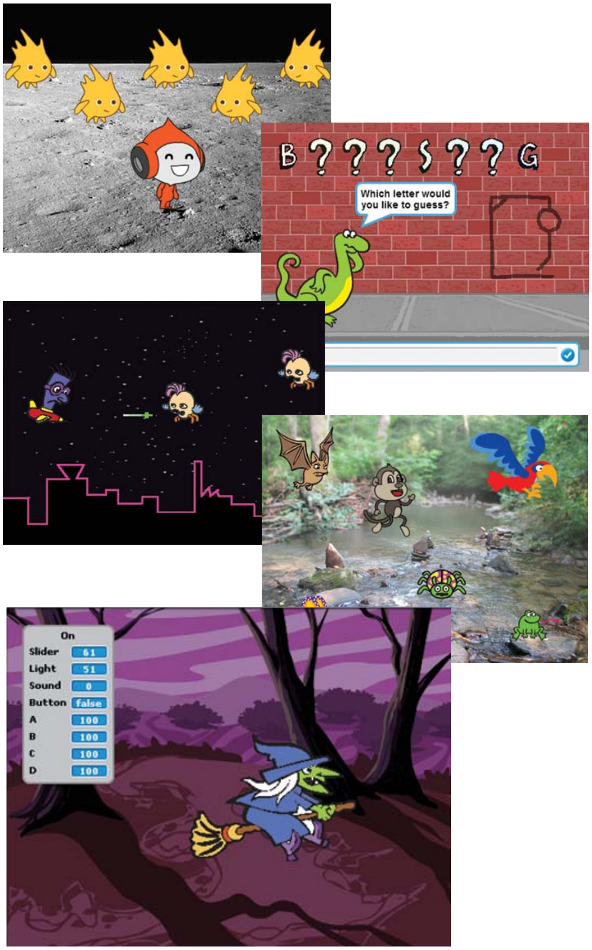
FULLY ILLUSTRATED

IN FULL COLOR

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:

- Design, build and share your own programs
- Create addictive arcade games, quizzes and word games
- Make computer-generated art
- Play your favourite 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 and the sensors on a PicoBoard



Which letter would you like to guess?

On
Slider 61
Light 51
Sound 0
Button false
A 100
B 100
C 100
D 100

This book is an in-depth guide to both the new web-based Scratch 2.0, and Scratch 1.4, as used on the Raspberry Pi. With no prior knowledge needed, this book takes you from scratch to Scratcher!

Contents

How to use this book

Welcome to the wonderful world of Scratch!

This book will show you some of the great projects you can build with Scratch, and will give you the skills and knowledge you need to start designing your own projects.

The book has been written so that each chapter teaches you something new, and builds on the previous chapters. For that reason, the best way to use this book is to work through the chapters in the right order.

Feel free to experiment as you go, though: there are lots of things you can do to customize and improve the examples in this book, and that's a big part of what Scratch is all about.

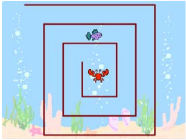
If you need advice on something that hasn't been covered yet, or you want to revise something you read earlier, consult the index for more information.

Here's what's in store:

- In Chapter 1, you'll learn the basics of Scratch. Sprites are objects you can use in your games, and you'll meet the cat sprite here. In your first program, it goes for a quick walk.

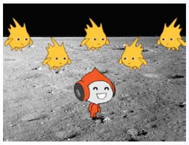
- In Chapter 2, you'll learn how to draw with Scratch, how to move sprites around the Stage, and how to make bits of your program repeat. The final project is art program Rainbow Painter, which leaves a colorful line behind your mouse.

- In Chapter 3, Spiral Rider is your first game. You'll learn how to enable keyboard control of a sprite, and how to detect when the player's character touches something else, and you'll discover how to remember information in a program.

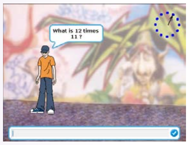


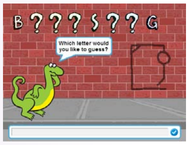
- In Chapter 4, the Dodgeball game shows you how to build a game with lots of sprites moving at once. You'll learn how to design your own sprites, how to clone them in Scratch 2.0, how to make random numbers, and how to keep score.

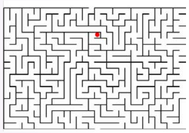
- In Chapter 5, you get to conduct the Cosmic Chorus. You'll see how to use recorded sounds and musical notes, how to get one sprite to control another, and how to make a title screen.



...cont'd







- In Chapter 6, Quiz Break tests players' maths skills against the clock. You'll learn how to use the timer, how to use the operators for maths, and how to join text together.
- In Chapter 7, the game Hangman teaches you how to use lists in Scratch, how to ask the player for information, how to structure complex programs, and how to write on the Stage.
- In Chapter 8, you'll create an arcade game and learn how to make one sprite fire on another, and how to make special effects to include in your game.
- In Chapter 9, you'll learn how to use Scratch with your webcam, microphone and the PicoBoard add-on which combines several sensors in one Scratch-friendly device. Flap your hands at the screen to wave away the bats and use your Picoboard to control a witch's night flight. You'll also see how to control a light connected to the Raspberry Pi from Scratch.
- Chapter 10 contains seven short programs you can experiment with and customize, including the games Shop Cat, Maze Mania, Photo Safari and Breakthrough.
- Chapter 11 gives you advice on creating your own programs, and resources you can use to help you get started.

Downloading the examples

The best way to use this book is to work your way through the chapters in order, creating the examples as you go. You'll learn more about programming, and you'll have a chance to customize the examples as you go.

If you can't get the examples to work, or if you want to take a shortcut, you can download them. The examples for Scratch 1.4 can be downloaded at the author's website at **www.sean.co.uk** and the publisher's website at **www.ineasysteps.com**

To find the examples for Scratch 2.0, visit the author's profile on the Scratch website. You can find it at **http://scratch.mit.edu/users/seanmcmanus/**

Foreword by Mitchel Resnick

Mitchel Resnick is Professor of Learning Research at the MIT Media Lab.

Is it important for all children to learn how to write? After all, very few children grow up to become journalists, novelists, or professional writers. So why should everyone learn to write?

Of course, such questions seem silly. People use writing in all parts of their lives: to send birthday messages to friends, to jot down shopping lists, to record personal feelings in diaries. The act of writing also engages people in new ways of thinking. As people write, they learn to organize, refine, and reflect on their ideas. It's clear that there are powerful reasons for everyone to learn to write.

I see coding (computer programming) as an extension of writing. The ability to code allows you to "write" new types of things – interactive stories, games, animations, and simulations. And, as with traditional writing, there are powerful reasons for everyone to learn to code.

The recent surge of interest in learning to code, reflected in sites like codecademy.com and code.org, has focused especially on job and career opportunities. It is easy to understand why: the number of jobs for programmers and computer scientists is growing rapidly, with demand far outpacing supply.

But I see much deeper and broader reasons for learning to code. In the process of learning to code, people learn many other things. They are not just learning to code, they are coding to learn. In addition to learning mathematical and computational ideas (such as variables and conditionals), they are also learning strategies for solving problems, designing projects, and communicating ideas. These skills are useful not just for computer scientists but for everyone, regardless of age, background, interests, or occupation.

Six years ago, my research group at the MIT Media Lab launched the Scratch programming language and online community in an effort to make coding accessible and appealing to everyone.

...cont'd

We've been amazed with the diversity and creativity of the projects. Take a look at the Scratch website and you'll find animated stories, virtual tours, science simulations, public-service announcements, multimedia art projects, dress-up games, paint editors, and even interactive tutorials and newsletters.

We find that active members of the Scratch community start to think of themselves differently. They begin to see themselves as creators and designers, as people who can make things with digital media, not just browse, chat, and play games. While many people can read digital media, Scratchers can write digital media.

Scratch community members also begin to see the world in new ways. As one 11-year-old Scratcher wrote on a public blog: "I love Scratch. Wait, let me rephrase that – Scratch is my life. I have made many projects. Now I have what I call a 'Programmer's mind.' That is where I think about how anything is programmed. This has gone from toasters, car electrical systems, and soooo much more."

It has been exciting to watch what young people are creating and learning with Scratch. But this is just the beginning. The new version of the Scratch programming language and online community moves Scratch into the cloud, enabling people to program, save, share, and remix Scratch projects directly in a web browser. The new version also adds many new features to enhance opportunities for creativity and collaboration.

But we are aware that new features and capabilities are not enough. The biggest challenges for the future are not technological but cultural and educational. Ultimately, what is needed is a shift in mindsets, so that people begin to see coding not only as a pathway to good jobs, but as a new form of expression and a new context for learning.

Mitchel Resnick

This is an edited version of an article that was originally published at Edsurge (www.edsurge.com).

Hot tip

The book has been written so that each chapter teaches you something new, and builds on the previous chapters. For that reason, the best way to use this book is to work through the chapters in the right order.

1 Introducing Scratch

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

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.

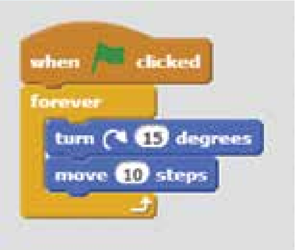
Scratch is easier to use than most other programming languages for a number of reasons:

- 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 always 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 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.

Hot tip

Scratch doesn't cost anything and it works on Windows, Mac and Linux computers.



```
when 🏴 clicked
forever
  turn ↻ 15 degrees
  move 10 steps
```

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

Which version of Scratch?

There are two versions of Scratch that are in popular use. In this book, I'll tell you about both.

Scratch 2.0

Scratch 2.0 is the latest version of Scratch, which was introduced in May 2013. I recommend you use Scratch 2.0 if you can. This version makes it easier for people to share their projects, take a look at each other's projects, and adapt them (or "remix" them).

You don't need to install any software to use Scratch 2.0: it runs inside your browser, using an Internet connection. Scratch 2.0 needs a computer that can run the Adobe Flash Player, though, so it doesn't work on some mobile devices (including the iPhone and iPad) and some lower-powered devices (including the Raspberry Pi). Most other computers run Scratch 2.0 fine.

There is also a version of Scratch 2.0 that you can download from the Scratch website and install on your computer, so you can use it without an Internet connection. There are no official plans to bring Scratch 2.0 to the Raspberry Pi, though (*correct at the time of printing*).

Scratch 1.4

Scratch 1.4 is the previous version of Scratch, and it is software you install on your computer. You can download it for Windows, Mac and Linux at **http://scratch.mit.edu/scratch_1.4/** This version of Scratch found a whole new audience with the meteoric rise of the Raspberry Pi, the stripped-down Linux-based computer for hobbyists and education (see photo).

Beware

If you're using the Scratch 2.0 website, I recommend using the Google Chrome browser with it. I experienced some bugs using Internet Explorer, which disappeared when I started using Chrome. You can download Chrome for free at **www.google.com/chrome**

Hot tip

Members of the Scratch team have said that they hope to make an iPad version of Scratch, but that it might be more limited than Scratch 2.0.

Hot tip

If you have a weak Internet connection, or none at all, you might prefer to use the downloadable version of Scratch to the browser-based version.

Starting on the Raspberry Pi

Scratch is included with the recommended version of the Linux operating system, called Raspbian. I'll assume you've got your Raspberry Pi connected up and working already:

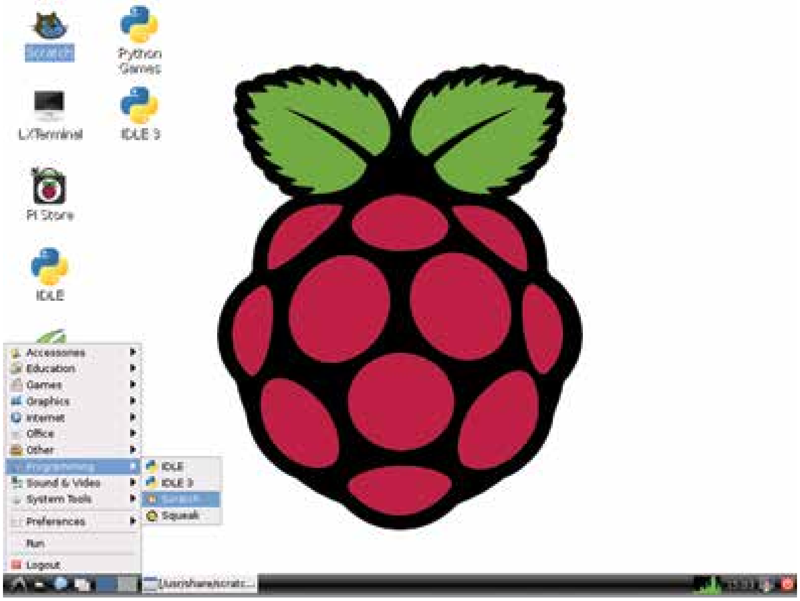
1. Switch on your Raspberry Pi and log in. The default login is **pi**, and the default password is **raspberry**

2. When you see the prompt pi@raspberrypi ~ $, enter **startx** and press the Enter or Return key

3. The LXDE desktop appears (see picture). The large Raspberry is just a background image, so you might see something different. You might also see different icons

4. Double-click the Scratch icon, which has a picture of a cat on it. In the screenshot below, it's in the top-left corner of the screen

5. You can also start Scratch using the Programs Menu, as also shown below: click the button in the bottom-left to open the menu, click the Programming folder, and finally, click Scratch



Hot tip

If you don't have Scratch installed on your Raspberry Pi, you can install it. After logging in, enter the command:

```
sudo apt-get update
&& sudo apt-get install
scratch
```

That command will also update Scratch to the latest version, if Scratch is already installed.

Beware

The commands on the Raspberry Pi are case-sensitive, so it won't work if you type in Startx (with a capital S), for example.

Creating a Scratch account

Before you begin to program with Scratch 2.0, I recommend you create an account for the Scratch website. Here's how:

1. Open a web browser, such as Google Chrome

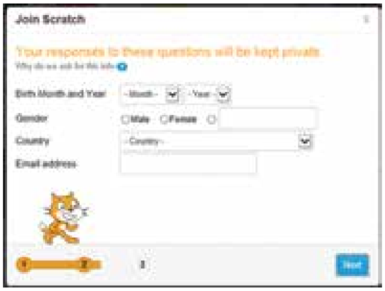
2. Enter the website address **http://scratch.mit.edu/** in your address bar, usually at the top of the screen

3. Click **Join Scratch** in the top-right

4. Make up a username. Scratch is used by young people (among others), so the site advises members to protect their privacy by not using real names. 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 upper and lower case, 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. It doesn't appear on the website anywhere. Click the **Next** button



Join Scratch

Your responses to these questions will be kept private

Why do we ask for this info?

Birth Month and Year - Month - - Year -

Gender ○ Male ○ Female ○

Country - Country -

Email address

7. Click **OK Let's Go!** and you will be logged in. Simply click **Create** on the navigation bar at the top of the screen, and you're ready to start programming!

Hot tip

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.

Hot tip

You can try Scratch 2.0 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.

Hot tip

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

Using the Scratch screen

Don't forget

If you're using a Raspberry Pi, you're using Scratch 1.4. If you're using Scratch in a browser, you're using Scratch 2.0.

Hot tip

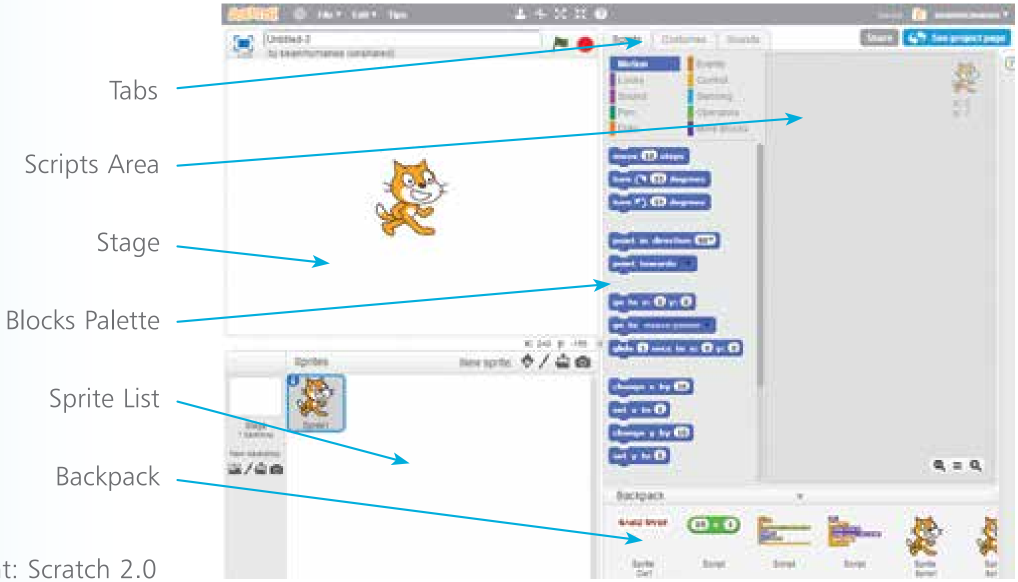
Use the tabs above the Blocks Palette (in Scratch 2.0) or above the Scripts Area (in Scratch 1.4) to switch between the scripts, costumes and sounds on a sprite. You'll learn all about how to use these later!

To start using Scratch 2.0, visit **http://scratch.mit.edu/** in your web browser and click **Create** at the top of the screen. To start using Scratch 1.4, double-click its icon on your desktop.

The most obvious difference between the two versions of Scratch is the way the screen is laid out. With the exception of the Backpack (a new feature in Scratch 2.0), all the same elements are there, but they've been juggled around a bit. The background colors have been lightened in Scratch 2.0 too. This page shows the screen layout in Scratch 2.0, and the facing page shows the screen layout in Scratch 1.4.

The main parts of the screen 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. In Scratch 2.0, the Stage is on the left, whereas in Scratch 1.4, the Stage is on the right.
- **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. In both versions of Scratch, the Sprite List is underneath the Stage.

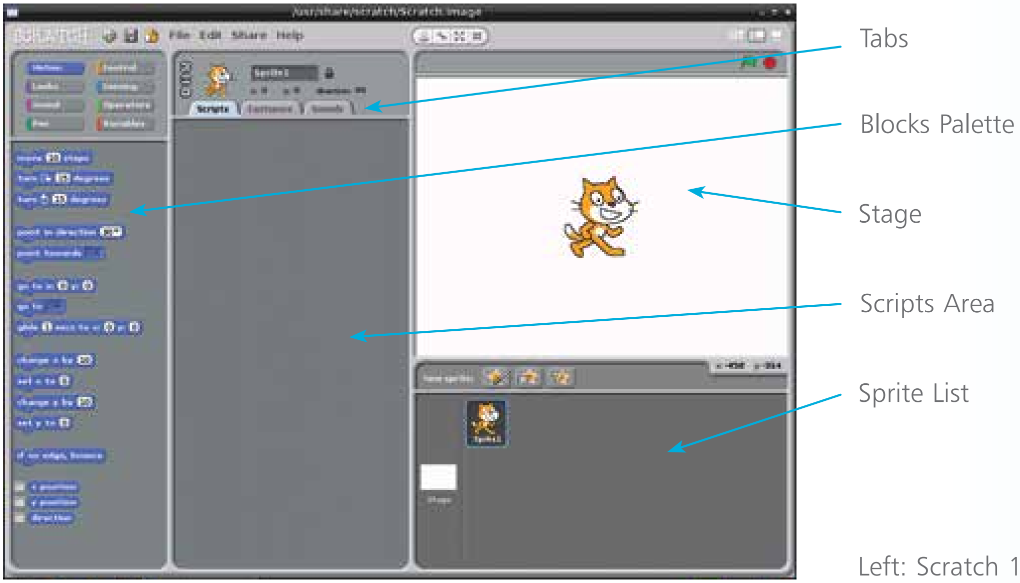


Right: Scratch 2.0

- **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 above it, such as the **Looks** button or the **Sound** button.
- **Scripts Area:** The Scripts Area is where you make your programs in Scratch, by assembling blocks there. This area expands to fill the screen space available, so if you use a larger monitor, the Scripts Area will be bigger.
- **Backpack:** The Backpack is a new feature in Scratch 2.0, which you can find underneath the Blocks Palette and Scripts Area. Click it to open it. It works a bit like a clipboard. You can copy scripts or sprites to it by dragging them there and dropping them. If you want to use them, just drag them from the Backpack back into your project. Your Backpack works across all your projects, so it's a great way to copy sprites or bits of program between different projects.

Don't forget

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 these pages are here to refer to at any time.

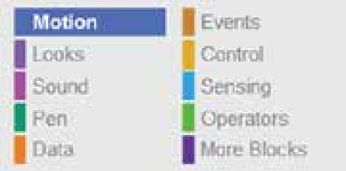


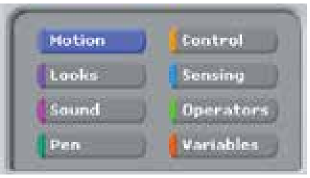
Left: Scratch 1.4

Exploring the blocks

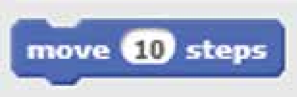
Before we start making a program, try experimenting with a few blocks to see what they do:

1 Click the **Motion** button above the Blocks Palette to show the Motion blocks. This button is selected when you first start Scratch

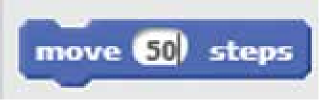




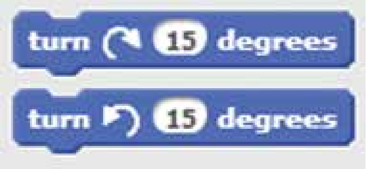
2 In the Blocks Palette, click the **move 10 steps** block. The cat on the Stage moves in the direction it's facing, to the right. Each time you click the block, the cat moves once. This block only changes the cat's position, though: you won't see its legs move



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



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



Hot tip

Sometimes I'll show you what something looks like in both Scratch 2.0 and Scratch 1.4. The picture on the left, or top, will be Scratch 2.0. The one on the right, or bottom, will be Scratch 1.4.

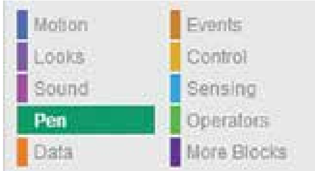
Hot tip

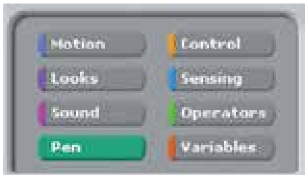
Scratch won't let the cat disappear off the screen completely. If you use numbers that are too big, the sprite will stay at the edge of the screen.

Above: The cat on the Stage, after I clicked Turn Clockwise 15 Degrees.

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 **Pen** button above the Blocks Palette

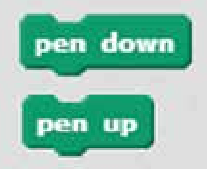




Hot tip

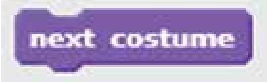
All these blocks are explained later in the book, but for now it's worth spending a few minutes exploring some of the commands you can give the cat.

7. Click the **pen down** block. Now, if you click the Motion button and click the blocks to move the cat around, it will leave a line behind it wherever it goes. There is also a **pen up** block you can use to turn this effect off again



8. If you're using Scratch 2.0, click the **Looks** button above the Blocks Palette

9. In Scratch 2.0, click the **next costume** block to see the cat's legs move, so it appears to run on the spot. Costumes are just different pictures a sprite can have, and the cat has two that show its legs in different positions (see below)



Hot tip

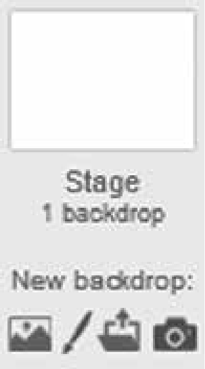
In Scratch 1.4, the cat only has one costume, but I'll show you how to add additional costumes in Chapter 5.

Changing the backdrop

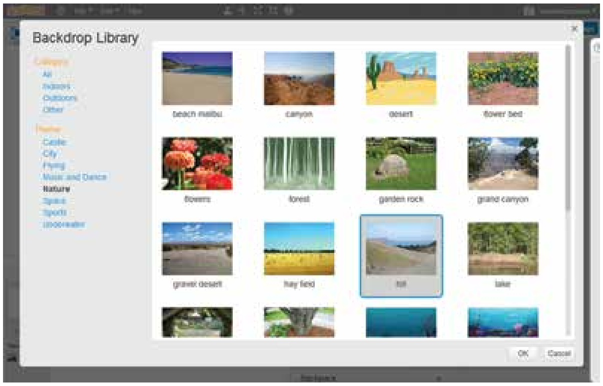
Before we make our first program, let's change the background of the Stage to something more inspiring. The way you do this is different in Scratch 2.0 and Scratch 1.4.

Choosing a backdrop in Scratch 2.0

1. To the left of the Sprites List, there is a panel for the Stage. Underneath the heading New Backdrop, click the first icon to choose a backdrop from the library. The other icons enable you to paint a backdrop, upload a picture from your computer, or use your webcam to take a photo



2. When the library opens, click the themes and categories on the left to view different backdrops available, and use the scrollbar on the right to see more designs. Click the Nature theme



3. Click the hill image, and then click the **OK** button

4. Your backdrop is added to the Stage, behind the cat, and the Paint Editor opens on the right so you can edit the background if you want to. We'll look at the Paint Editor in Chapter 4

Hot tip

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 2.0 and you can't see the File menu, click Create instead at the top of the screen.

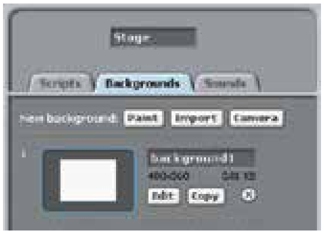
Don't forget

In Scratch 2.0, the picture on the Stage is called a backdrop. In Scratch 1.4, it's called a background.

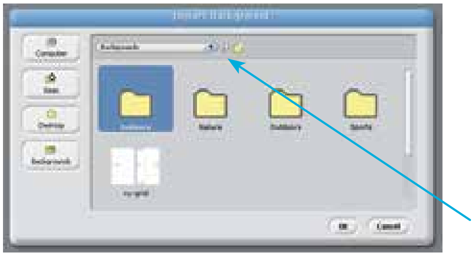
Choosing a background in Scratch 1.4

1. To the left of the Sprite List is a white icon that represents your blank Stage. Click it to start

2. Click the **Backgrounds** tab above the Scripts Area, then click the **Import** button

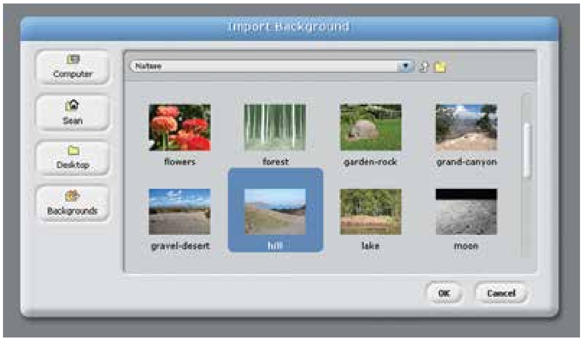


3. Use the file browser to explore the backgrounds available. Double-click a folder icon to look inside that folder. To go back to the previous folder, click the **up arrow** indicated below



Previous folder

4. Go into the Nature folder and use the scrollbar on the right to find the hill picture. Click it and then click **OK**



5. Your background is added to the Stage, behind the cat

Hot tip

You can also use the file browser in Scratch 1.4 to find your own files on your computer to use as backgrounds. In the left of the file browser, Click Computer, your username or Desktop to start browsing your computer. To get back to the Scratch backgrounds again, click Backgrounds.

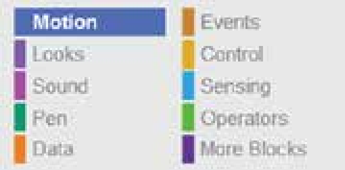
Creating your first program

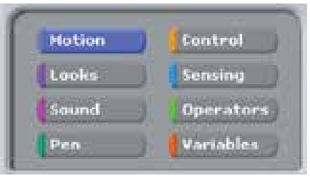
When you click blocks in the Blocks Palette, the cat moves immediately, so this is good for testing what blocks do, but not useful for making a program. A program is a set of repeatable instructions that you can store up to carry out later. For our first Scratch program, let's make the cat walk down 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 Scripts Area is open. If the Scripts Area is empty, you're ready. If it shows costumes or sounds instead, click the **Scripts** tab. In Scratch 2.0, the **Scripts** tab is above the Blocks Palette. In Scratch 1.4, it's above the Scripts Area

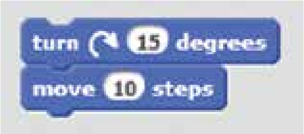
3. Click the **Motion** button above the Blocks Palette





4. You make a program by dragging blocks into the Scripts 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 Scripts 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 Scripts 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



```
turn ↻ 15 degrees
move 10 steps
```

Hot tip

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

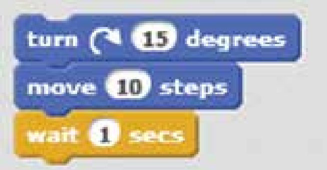
Beware

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

Don't forget

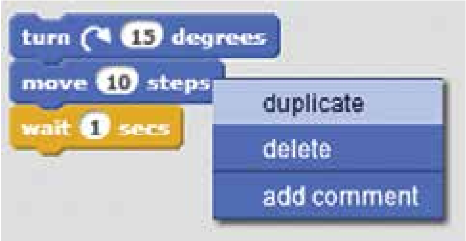
Make sure the blocks snap together, otherwise they won't work as one script. If they don't snap together, they're not close enough.

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



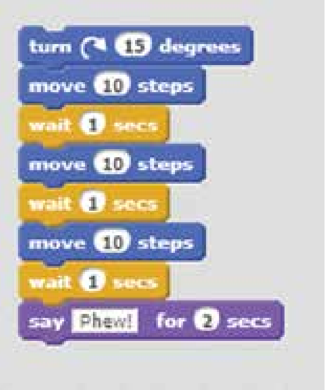
```
turn ↻ 15 degrees
move 10 steps
wait 1 secs
```

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



```
turn ↻ 15 degrees
move 10 steps
wait 1 secs
```

8 It's a long walk for a tiny cat, so let's make him finish his walk with a exclamation of 'Phew!' in a speech bubble. Click the **Looks** button above the Blocks Palette, and drag the **say Hello! for 2 secs** block into the Scripts Area and join it to your program. Click Hello! to edit what the cat says to Phew!



```
turn ↻ 15 degrees
move 10 steps
wait 1 secs
move 10 steps
wait 1 secs
move 10 steps
wait 1 secs
say Phew! for 2 secs
```

Beware

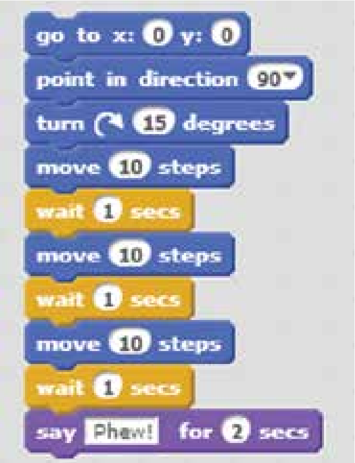
I wouldn't usually recommend you add the same blocks repeatedly, but we're right at the beginning of learning Scratch here. There is a more readable and elegant solution you'll discover in Chapter 2.

Hot tip

If right-clicking doesn't work for you (or your mouse doesn't have a second button), hold down the Control key on your keyboard and click, or try holding down the Shift key and clicking. This tip applies anywhere you're asked to right-click later in the book.

...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 Scripts Area. Scratch carries out all the joined-up instructions in order, starting at the top and working its way down the blocks



```
go to x: 0 y: 0
point in direction 90
turn ↻ 15 degrees
move 10 steps
wait 1 secs
move 10 steps
wait 1 secs
move 10 steps
wait 1 secs
say Phew! for 2 secs
```

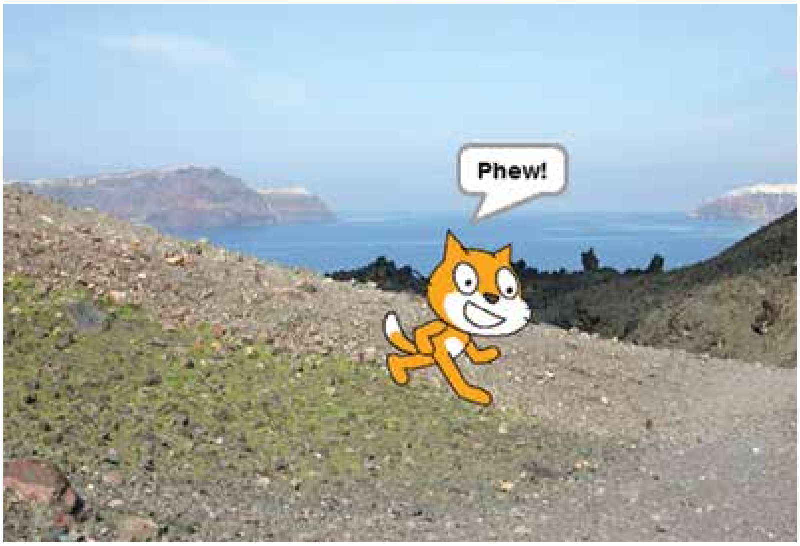
10 What happens if you click the script to run it again? The cat turns again and walks from where it finished last time. Eventually, it'll be walking on its head. Let's add some blocks to put it in the right starting position. Click the **Motion** button above the Blocks Palette and drag in the **point in direction 90** block and the **go to x:0 y:0** blocks. 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

Hot tip

You can insert blocks into an existing script. Instead of dropping a block at the end of your script, drag your block over it and a white line will show you where the block will be dropped. When it's in the right place, release the mouse button.

Hot tip

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?



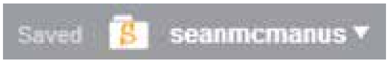
Right: A few blocks and one exhausted cat later, here's your first program.

Saving your project

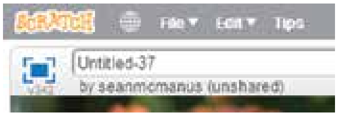
A Scratch project includes all the sprites, scripts and backgrounds that are used in it. It's a good idea to save your projects so you can come back to them later to reuse them or modify them.

Saving projects in Scratch 2.0

In Scratch 2.0 (web version), your work is automatically saved for you as you make changes to your project. In the top-right corner of the screen, you can see whether your latest changes have been saved. If they haven't, there will be a link here to **Save Now**.



Your project is saved with the name Untitled plus a number. You can choose a more useful name by editing the box above the Stage.

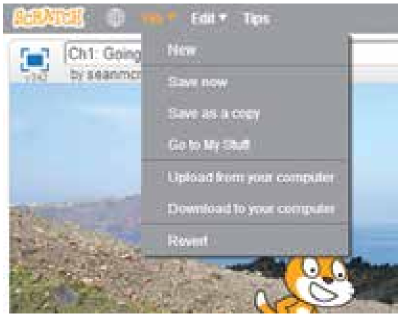


Untitled-37

by seanmcmanus (unshared)

There are additional options for saving your work in the File menu, also above the Stage, see below. These 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 program without losing a working version of it.
- **Download to your computer**: This enables you to save your project as a file on your computer. You can open it using the downloadable version of Scratch 2.0. If your Internet connection fails, use this option straight away to save your work!
- **Upload 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 2.0 website.
- **Revert**: This throws away all the changes you've made to the project since you opened it this time.



Beware

You can't use projects made using Scratch 2.0 in Scratch 1.4. You can use Scratch 1.4 projects in 2.0, though.

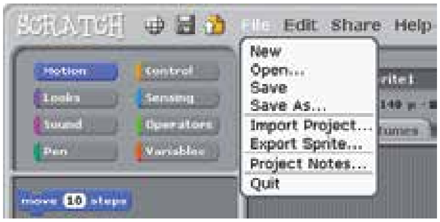
Hot tip

In the downloadable version of Scratch 2.0, your work isn't automatically saved for you. Use the **Save** option in the File menu to save your work. Use the **Save As** option to save your project with a new filename, so you don't overwrite the previous version.

...cont'd

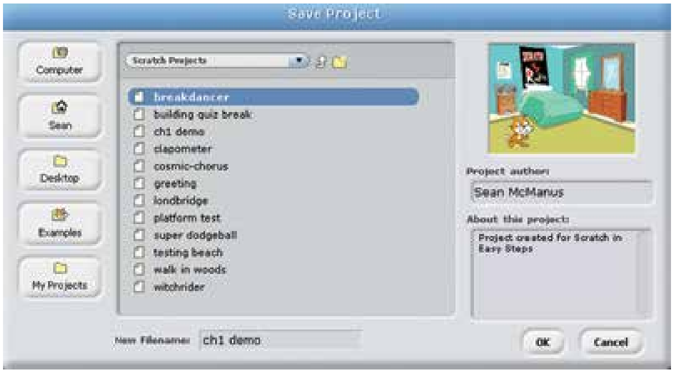
Saving and opening your project in Scratch 1.4

Click the File menu above the Blocks Palette at the top of the screen. This menu's options include:



- **Open**: Opens a previously saved file.
- **Save**: Use this to save a new project, or an old project you've been editing. If you're saving a new version of a saved project, you'll replace the old saved version.
- **Save as a copy**: If you want to keep the previous version of a project, use this option. It will save a new file containing your project, and leave the previous file untouched.
- **Import Project**: This enables you to open a project, and combine its sprites, scripts and backgrounds with the project that's currently open.
- **Export Sprite**: This option enables you to save a sprite as a costume you can use in other projects. You'll learn about costumes in Chapter 5.

When you save a new project or a new copy of a project, the file browser opens, as shown below. The buttons down the left are used to choose where to save your file. Click the folder name at the top (Scratch Projects in the screenshot) to choose another drive or folder, and click the up arrow beside it to go up a folder. Type the filename in the box at the bottom and click **OK**.



breakdancer
building quiz break
ch1 demo
clapometer
cosmic-chorus
greeting
londbridge
platform test
super dodgeball
testing beach
walk in woods
witchrider

Project author: Sean McManus

About this project: Project created for Scratch in Easy Steps

New Filename: ch1 demo

Hot tip

Click Examples on the left of the file browser to find some demo projects.

Hot tip

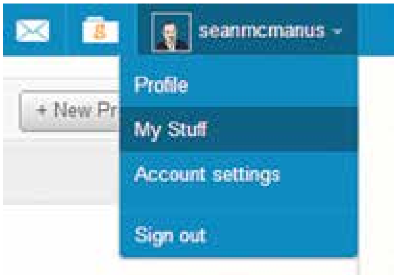
You can add the project author name and a short description of the project in the boxes on the right in the file browser.

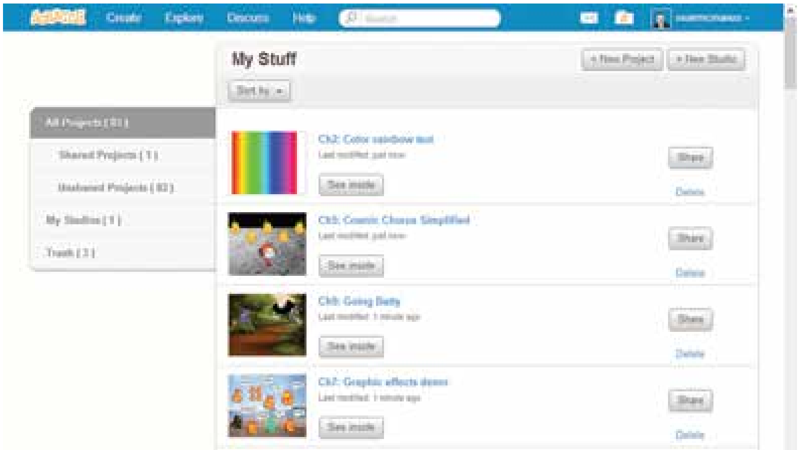
Don't forget

If you are using Scratch 1.4, remember to save regularly to make sure you don't lose any of your work.

Opening projects

In the downloadable version of Scratch 2.0, you open projects through the File menu at the top of the screen. The website is a bit more complicated. To find your projects there, click your username in the top-right of the screen 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. Take a look at my projects in the screen below:





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. If no more projects load when you click the button, that's all of them. To open a project, click its **See inside** button.

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.

Beware

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.

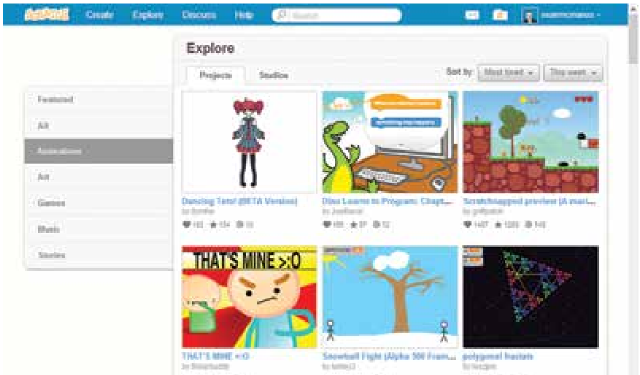
Don't forget

In Scratch 1.4, you open projects by clicking to open the File menu at the top of the screen and then choosing Open.

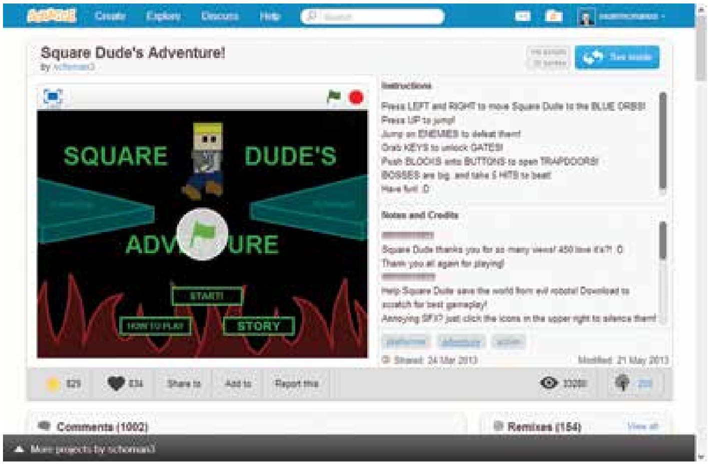
Opening shared projects

You can open the projects that other people have shared on the Scratch website too. Visit the website at **http://scratch.mit.edu** and click **Explore** at the top of the screen. Use the options on the left to choose a category.

The menus in the top right enable you to sort by the most loved, most remixed and most recent, and to choose how new you want the projects to be.



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 in the middle of the player to run it. 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 scripts, sprites and backgrounds that make it work.



Instructions

Press LEFT and RIGHT to move Square Dude to the BLUE ORBS!
Press UP to jump!
Jump on ENEMIES to defeat them!
Grab KEYS to unlock GATES!
Push BLOCKS onto BUTTONS to open TRAPDOORS!
BOSSES are big, and take 5 HITS to beat!
Have fun! :D

Notes and Credits

Square Dude thanks you for so many views! 450 love it's?! :D
Thank you all again for playing!

Help Square Dude save the world from evil robots! Download to scratch for best gameplay!
Annoying SFX? just click the icons in the upper right to silence them!

Shared: 24 Mar 2013
Modified: 21 May 2013

Hot tip

When you open somebody else's project in the editor, you can click the Remix button in the top right to create your own version of it. All the projects on the Scratch website are shared on the understanding that others can learn from them, and create adaptations of them.

Beware

At the time of writing, there isn't a way to download shared projects from the website to use in Scratch 1.4.

Index

Why choose this book?

It's written in plain English

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

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

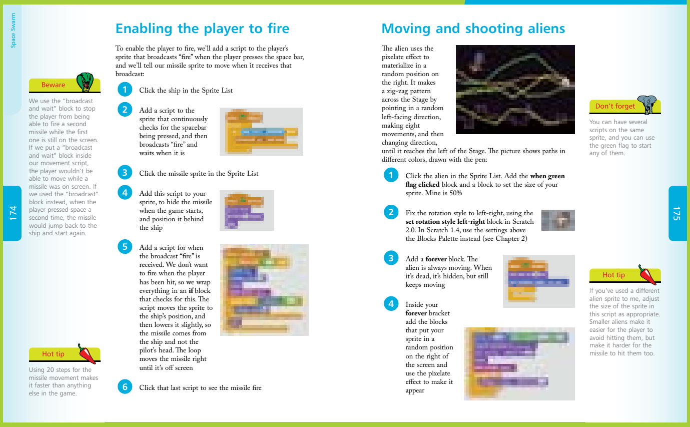
Do we need to say any more?

Scratch is a programming language that is widely used on the *Raspberry Pi* and in schools and colleges. *Scratch's* highly visual interface and drag-and-drop commands make it an ideal language for all ages to try to program. With Scratch Programming in easy steps at hand, learning programming will be a breeze.

This primer introduces you to *Scratch* fundamentals and then walks you through the commands to create games and animations. Learn to create games that require skill, knowledge or quick fingers, such as Spiral Rider, Space Swarm, or the classic Hangman game. Add music and special effects to your games, and of course keep score.

By the time you've finished, you'll be impressing your friends and family with your own computer games!

Look inside



Space Swarm

174

Enabling the player to fire

To enable the player to fire, we'll add a script to the player's sprite that broadcasts "fire" when the player presses the space bar, and we'll tell our missile sprite to move when it receives that broadcast:

1. Click the ship in the Sprite List
2. Add a script to the sprite that continuously checks for the spacebar being pressed, and then broadcasts "fire" and waits when it is
3. Click the missile sprite in the Sprite List
4. Add this script to your sprite, to hide the missile when the game starts, and position it behind the ship
5. Add a script for when the broadcast "fire" is received. We don't want to fire when the player has been hit, so we wrap everything in an **if** block that checks for this. The script moves the sprite to the ship's position, and then lowers it slightly, so the missile comes from the ship and not the pilot's head. The loop moves the missile right until it's off screen
6. Click that last script to see the missile fire

Beware

We use the "broadcast and wait" block to stop the player from being able to fire a second missile while the first one is still on the screen. If we put a "broadcast and wait" block inside our movement script, the player wouldn't be able to move while a missile was on screen. If we used the "broadcast" block instead, when the player pressed space a second time, the missile would jump back to the ship and start again.

Hot tip

Using 20 steps for the missile movement makes it faster than anything else in the game.

Moving and shooting aliens

The alien uses the pixelate effect to materialize in a random position on the right. It makes a zig-zag pattern across the Stage by pointing in a random left-facing direction, making eight movements, and then changing direction, until it reaches the left of the Stage. The picture shows paths in different colors, drawn with the pen:

1. Click the alien in the Sprite List. Add the **when green flag clicked** block and a block to set the size of your sprite. Mine is 50%
2. Fix the rotation style to left-right, using the **set rotation style left-right** block in Scratch 2.0. In Scratch 1.4, use the settings above the Blocks Palette instead (see Chapter 2)
3. Add a **forever** block. The alien is always moving. When it's dead, it's hidden, but still keeps moving
4. Inside your **forever** bracket add the blocks that put your sprite in a random position on the right of the screen and use the pixelate effect to make it appear

Don't forget

You can have several scripts on the same sprite, and you can use the green flag to start any of them.

Hot tip

If you've used a different alien sprite to me, adjust the size of the sprite in this script as appropriate. Smaller aliens make it easier for the player to avoid hitting them, but make it harder for the missile to hit them too.

175

Let these icons make it even easier

Wherever you see one of these icons you know there's a handy tip to spice up your learning, flag something to remember or ward you away from potential dangers.

> "I was extremely happy with the In Easy Steps series and have purchased several of their guides. Their "get it done" tutorials provide quick and easy reference material for when I'm programming or designing."
>
> Lowter - Webmaster ezine

£10.99 UK / $14.99 US / $16.95 CN

ISBN-13: 978-1-84078-612-5

51499

9 781840 786125

www.ineasysteps.com