



11 Awesome Projects

written especially for young people!

Adventures in Coding

The image shows a Scratch script for a drum set. It consists of a large orange 'else' block containing three nested 'if' blocks. Each 'if' block checks for a specific key press and then triggers a 'play drum' block with a specific drum number and a duration of 0.25 beats. To the right of the script are three yellow callout boxes, each connected to a 'play drum' block by a yellow line. The first callout is connected to 'play drum 2', the second to 'play drum 6', and the third to 'play drum 11'.

```
else
  if key left arrow pressed? then
    play drum 2 for 0.25 beats
  else
    if key right arrow pressed? then
      play drum 6 for 0.25 beats
    else
      if key down arrow pressed? then
        play drum 11 for 0.25 beats
```

- up arrow, play snare
- left arrow, play bass drum
- right arrow, play hi-hat
- down arrow, play cowbell

Eva Holland
Chris Minnick

WILEY

Adventures in Coding

Adventures in Coding

by **Eva Holland and Chris Minnick**

WILEY

Adventures in Coding

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2016 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-119-23268-1
ISBN: 978-1-119-23269-8 (ebk)
ISBN: 978-1-119-23275-9 (ebk)

Manufactured in the United States of America

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEB SITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEB SITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEB SITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Library of Congress Control Number: 2016931381

Trademarks: Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

This book is dedicated to the brave souls who have inspired in us a sense of adventure, especially David J. Holland, Patricia Minnick, Mary Ellen Holland, and Patrick Minnick.

Publisher's Acknowledgments

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

Editorial

Series Creator: Carrie Anne Philbin

Professional Technology & Strategy Director: Barry Pruett

Acquisitions Editor: Aaron Black

Project Editor: Charlotte Kughen

Copy Editor: Kezia Endsley

Technical Editor: Mike Machado

Editorial Manager: Mary Beth Wakefield

Marketing

Marketing Manager: Lorna Mein

About the Authors

EVA HOLLAND is an accomplished author, trainer, and co-founder of WatzThis?, a company aimed at teaching technical topics in a fun and approachable manner. In addition to being co-author of this book, Eva is the co-author *JavaScript For Kids For Dummies* and *Coding with JavaScript For Dummies*. She enjoys tennis, music, reading, and the outdoors.

CHRIS MINNICK is a prolific published author, trainer, web developer, and co-founder of WatzThis?. Chris loves sharing his knowledge with others and has trained thousands of adults and kids in computer programming. As an author, his published books include *JavaScript For Kids For Dummies*, *Coding with JavaScript For Dummies*, *Beginning HTML5*, *CSS3 For Dummies*, and *Webkit For Dummies*. Chris is an avid reader, writer, swimmer, and musician.

Authors' Acknowledgments

This book is the result of a team effort, not only by your authors, but also by a talented crew of editors and other professionals who are credited on an earlier page.

Thank you to everyone at Wiley, including our Project Editor Charlotte Kughen, Acquisitions Editor Aaron Black, Copy Editor Kezia Endsley, Illustrator Sarah Wright, and Senior Editorial Assistant Cherie Case. A very special thank you to our Technical Editors Gavin Machado and Mike Machado. Thank you to Jay Silver for advising us in the initial stages of the book and for his role in the creation of Scratch.

Thank you also to our agent, Carole Jelen of Waterside Productions, as well as to our friends and families.

Contents

Introduction. 1

What Is Scratch?	1
Who Should Read This Book?	1
What You Will Learn	1
How This Book Is Structured	2
What You Need to Use This Book	2
Conventions	2
Companion Website	3
Contact Us.	3

Adventure 1

Scratching the Surface. 5

Coding Is Everywhere	5
Speaking the Language of Machines	6
Knowing Your Coding Lingo	7
Writing Your First Scratch Program	8
Joining Scratch	8
Meeting Scratch the Cat	11
Moving Scratch Around	12
Connecting Blocks	13
Looping Movements	14
Starting at the Green Flag	16
Bouncing Off the Walls	17
Creating a Sprite	18
Handling Collisions on the Dance Floor	19
Slowing It Down	22
Learning the Scratch Environment	23
Exploring the Scratch Project Editor	24
The Toolbar	24
The Stage	31
The Sprite Pane	32
The Sprite Info Pane	33
The Scripts Area	33
The Block Palette	34
The Costume Pane	35
The Sound Pane	36
Working with Colors and Shapes in Scratch	37
Further Adventures in Coding	37

Adventure 2

Where in the World Is Scratch? 39

Setting the Stage	39
Interacting with the Stage	41
Customizing the Backdrop	42
Using a Picture Backdrop	46
Taking a Picture of the Stage	46
Understanding Coordinates on the Stage	47
Taking Center Stage	48
Moving Left, Right, Up, and Down	49
Knowing Your Directions	50
Working with Degrees of Rotation	51
Spinning Sprites	51
Moving a Sprite	54
Stepping	55
Gliding	55
Jumping	55
Creating Random Scratch Art	56
Randomizing Movements	56
Drawing Random Lines	58
Further Adventures in Coding	59

Adventure 3

Using Control Blocks 61

Understanding Code Nesting	61
Branching Programs in Scratch	64
The if () then Block	64
The if () then, else Block	65
Boolean Blocks	67
Making Comments in Scratch	69
Standalone Comments	69
Block Comments	71
Looping in Scratch	72
Looping Forever	72
Stopping Loops with the stop () Block	72
Creating a Counting Loop	72
Looping Until a Condition Is Met	73
Waiting	74

Building the Fly Catcher Game	74
Setting the Stage	75
Placing the Frog	76
Adding the Hippo-Fly	77
Scripting the Hippo-Fly	78
Making the Tongue	78
Further Adventures in Coding	81

Adventure 4

Using Sensing Blocks

Learning the Sensing Blocks	83
Getting and Using Text Input	84
Detecting Key Presses	89
Watching for Mouse Moves	91
Using Timing	94
Sensing Touching and Distance	97
Building the Apple Patrol Game	102
Further Adventures in Coding	105

Adventure 5

Using Event Blocks

Understanding the Role of Events	107
Working with Key Press Events	109
Using Backdrop Change Events	114
Implementing Sensing and Timing Events	118
Detecting Video Motion	118
Measuring Loudness	118
Waiting for the Right Time	118
Getting Your Message Across	118
Putting on the Big Event	123
Setting the Stages	124
Programming the Ringleader	126
Programming the Ballerina Show	128
Programming the Dinosaur Show	130
Programming the Stage	131
Showing and Hiding Sprites	133
Further Adventures in Coding	136

Adventure 6

Variables and Lists137

Understanding Variable Blocks	138
Variables Have Names	140
Variables Can Display on the Stage	142
Variables Can Change	144
Scratch Variables Are Persistent	146
Scratch Variables Are Big	147
Making Lists	148
Making a List	148
The Ultimate Party Host	151
Introducing the Lists and Variables	152
Asking for Names and Ages	152
Recalling Names and Ages	154
Further Adventures in Coding	158

Adventure 7

Using Operators in Scratch 159

Saying Hello to the Operators	159
Doing the Math	161
Addition	161
Subtraction	164
Multiplication	165
Division	166
Coding Logically	168
Operating on Text	170
Combining Text with join () ()	170
Finding Letters	172
Getting Text Length	173
Understanding Other Operations	174
() mod ()	174
round ()	174
() of ()	175
Make a Math Practice Game	176
Programming Different Paths	177
Making the Addition Quiz	178
Making the Multiplication Quiz	181
Further Adventures in Coding	182

Adventure 8

Using Graphics and the Pen185

Creating Art with the Paint Editor	185
Using Bitmap and Vector Graphics	189
Understanding the Bitmap Tools	189
Drawing with the Vector Tools	190
Making Slides	192
Use the Pen to Make a Skywriting Plane	198
Further Adventures in Coding	203

Adventure 9

Building Your Own Blocks 205

Making Your Own Blocks	205
Splitting Programs into Custom Blocks	206
Changing Custom Blocks	210
Borrowing Blocks with the Backpack	215
Putting on a Fashion Show	217
Further Adventures in Coding	222

Adventure 10

Making and Using Sounds and Music223

Using Sounds	223
The Sound Library	224
Using the Sound Editor	227
Editing Sounds	228
Recording Sounds	229
Importing Sounds	231
Forming the Scratch Jazz Band	232
Tuning Up Your Instruments	232
Finding a Drummer	233
Playing the Melody	234
Playing Together	237
Singing Along!	240
Further Adventures in Coding	242

Adventure 11

Exploring the Scratch Universe 243

Visiting scratch.mit.edu	243
Sharing Your Projects	245
The Scratch Community Rules	249
Remixing Projects	252
Interfacing with the Real World.	253
Understanding Electricity	255
Understanding Makey Makey.	256
Understanding Conductivity.	257
Sensing with the PicoBoard	259
Further Adventures in Coding	261

Appendix A

Installing the Scratch Offline Editor 263

Installing the Scratch Offline Editor on Windows	263
Installing the Scratch Offline Editor to Your Mac Operating System	268
Other Resources.	275

Appendix B

Testing Your Programs 277

Practice Proper Planning.	277
Ask Other People to Test	278
Look for Possible Invalid Input	279
Use Comments Often	280
Test Early and Often	280
Keep Track of Tests and What Breaks Things	280
Use Custom Blocks	281
Use Sliders for Numbers	281
Keep Learning.	282
Keep Practicing.	283

Glossary 285

Index 289

Introduction

ARE YOU A fearless adventurer? Do you like to set sail on new ventures and learn new skills? Do you want to learn how to use technology to turn your ideas into reality? Are you curious about computer programming, but aren't sure where to start? If your answer to these questions is a confident “yes!” then this book is for you!

Like people, computers can be talked to in a variety of languages. Computer programming, or *coding*, is a way for people to talk to computers. Many computer languages are similar to one another, so once you know one programming language, it's much easier to learn another one.

Adventures in Coding introduces you to the world of programming, using Scratch.

What Is Scratch?

Scratch is for anyone new to coding. It's a great place to begin your lifelong adventures in the world of computer programming. Scratch introduces programming concepts in a fun and approachable way. Using simple drag-and-drop features, you can create real computer programs while learning the fundamentals of coding.

Who Should Read This Book?

Adventures in Coding is a great starting point for any young person who is interested in learning how to create games, apps, and art on a computer.

What You Will Learn

Adventures in Coding introduces and guides you through the world of coding using Scratch. You learn the ins and outs of the Scratch universe—from learning about the features of the Project Editor, to connecting with fellow “Scratchers” and sharing projects.

Adventures in Coding teaches you how to create fun games, animate characters, build interactive projects, and more!

How This Book Is Structured

Each chapter of this book is an adventure of its own. With each adventure, you learn a new aspect of Scratch while building on what you already learned. Each adventure finishes with a completed project.

What You Need to Use This Book

The Scratch programming interface lives on the web. All you need to get started and complete the adventures in this book is a computer with a web browser (such as Chrome, Safari, Firefox, or Internet Explorer) and an Internet connection. No experience is expected or necessary. There is nothing you need to purchase or install. Scratch is always free, to anyone!

Conventions

Throughout the book, there are some special boxes to guide and support you. They use the following key:



These boxes explain new concepts or terms.



These boxes provide tips to make your life easier.



These boxes include important things to watch out for.

These boxes further explain the inner workings of programs.



These boxes provide explanations or additional information about the topic at hand.



These boxes point you to videos on the companion website.



You will also find two sets of sidebars in the book. *Challenge* sidebars give you extra suggestions for expanding on the projects in the book. *Digging into the Code* sidebars further explain how some of the more complex programs work.

Companion Website

To download the videos mentioned in this book, visit the companion website at www.wiley.com/go/adventuresincoding.

Contact Us

Your authors, Chris and Eva, would love to hear about your progress in coding. You can reach out to us with questions, or to show us a cool project you've created, by visiting www.watzthis.com or emailing us at info@watzthis.com.



Adventure 1

Scratching the Surface

PROGRAMMING COMPUTERS IS a lot of fun. It's also a skill that many people see as mysterious and even magical. This chapter unmask programming to show you just how easy it can be to start your own coding adventure.

Coding Is Everywhere

Computer programming, also known as **coding**, is how people tell computers what to do. What sorts of things can you do after you learn to program? For starters, you can write your own computer games, create modifications (or “mods”) for existing games, program robots to do your bidding, create beautiful computer art and animations, and instruct your computer to play songs! The best part is that the whole time you're doing all these fun things, you're learning a valuable skill that is in sky-high demand!

Coding is a common name for computer programming. When you code, you're using a computer language to tell computers what to do.



Can you think of other things that computers can do? Think of all the things that programmers can tell computers to do. There are hundreds, or thousands, of things.

Think about all the things you see computers do every day—and not just the fun things. Computer programs are used to create new medicines, design buildings, do complex mathematics, control cars, and so much more.

This is the amazing world in which computer programmers live; we get to solve interesting problems every day and do things that other people see as magic.

Speaking the Language of Machines

All sorts of different people are **programmers**. Programmers come from different places and countries, with different experiences and different training. They speak different languages, have many different interests, and program for different reasons. What they have in common is that they've learned to speak at least one language that is understood by computers.



A **programmer** is a person who writes computer programs.

Computers don't speak the same languages that people do. People speak languages such as English, French, Spanish, Portuguese, Japanese, and many others. Computers speak machine language. Machine language is a difficult-to-read (for us) language that uses numbers to provide instructions to computers.

If machine language were the only way people could talk to computers, coding would be difficult. Fortunately, people have invented languages, called **programming languages**, which make it easier for people to talk to computers. Here are some examples of programming languages:

- JavaScript
- BASIC
- Perl
- PHP
- Python
- Java
- Visual Basic
- C

- C++
- Scratch

These languages all have one thing in common: They take words and symbols that people understand and translate them into words and symbols that computers understand.

A **programming language** is a language used for giving instructions to computers.



The examples in this book use Scratch. Scratch is a language that was invented at the Massachusetts Institute of Technology (MIT). It was designed to be easy for beginners to learn while using (and teaching) all of the most important things that programmers need to know.

Knowing Your Coding Lingo

You already know some of the lingo of coding. You know that “coding” is just another name for “computer programming,” and you know that people who do computer programming—or coding—are called computer programmers (or coders).

Programming languages, like human languages, are made up of different parts. In English, we have nouns, verbs, adjectives, pronouns, and other parts of speech, not to mention punctuation, and they form sentences and paragraphs. In programming languages, you combine different statements (also known as **commands**) to make computer programs, which are also known as **applications** (or *apps*).

A **command** is an instruction, written in a programming language, that tells a computer to do a task.



An **application** is a set of programming commands that follow each other in a particular order to accomplish tasks. Application is another name for a computer program.



Scratch, and certain other programming languages, use the term **script**. Script is just another name for a program.



A **script** is another term for a computer program that is smaller and more limited than an application.



There are a lot of specialized words in coding that all have very specific meanings, and you'll find that sometimes there are many different words for the same thing. For this reason, we've included a glossary at the back of this book where you can look up or remind yourself of the meaning of terms you're not familiar with.

One of the greatest things about Scratch is that it's easy to dive right into! To get started, you don't need to learn a lot of new concepts or vocabulary. So enough talk! Let's begin!

Writing Your First Scratch Program

When we were growing up, kids didn't care about learning a specific style of dance. Instead of trying to learn complicated dance moves like the cha-cha or the hustle, we ran around like lunatics, jumping off of things, and we sometimes got hurt in the process. Your first Scratch program will be a simulator of an old-fashioned punk rock mosh pit.

Figure 1-1 shows what the finished product will look like. If you imagine the two characters in the figure bouncing off the walls and off each other while a drumbeat plays, you'll have a good idea of the program you'll be learning to make.

Joining Scratch

In order to create, save, and share your programs on the Scratch website, you need to use your favorite web browser to visit <http://scratch.mit.edu>, where you can sign up for a free account. When you go to the website, you see a screen that looks similar to Figure 1-2.

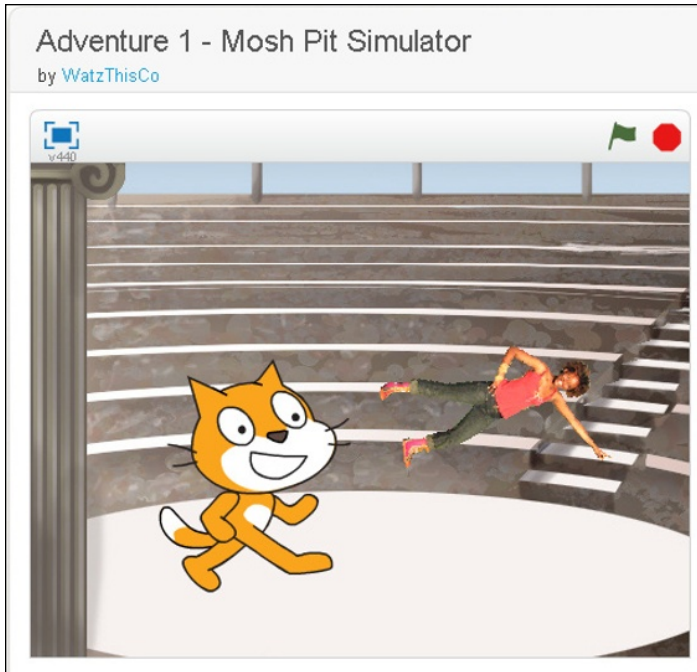
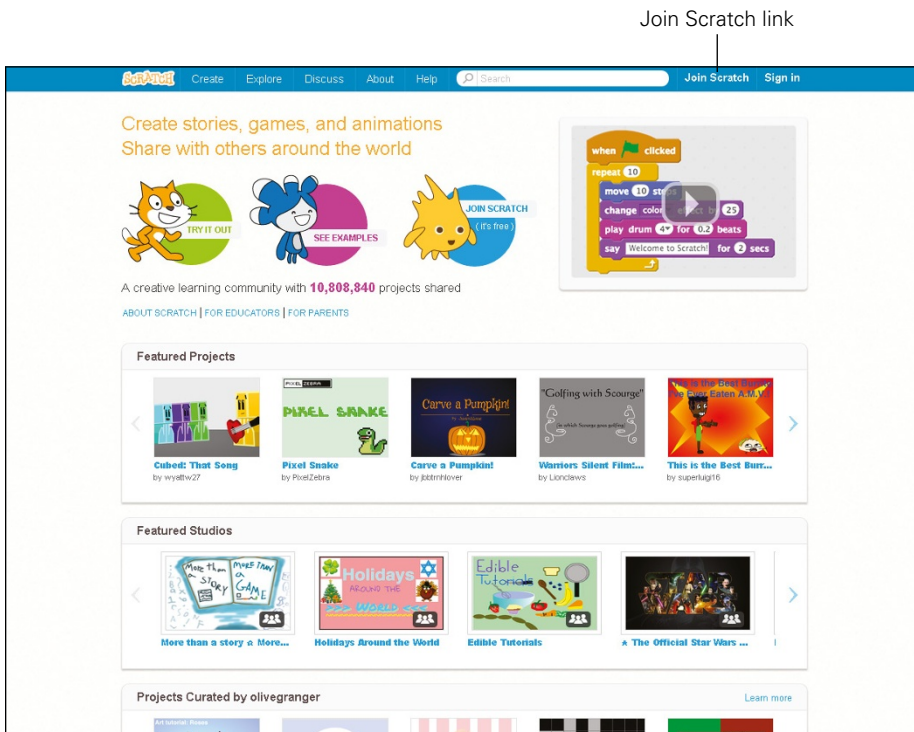


FIGURE 1-1 Your first Scratch program



Join Scratch link

FIGURE 1-2 The Scratch website

Follow these steps to create your free account:

1. Click the Join Scratch link in the upper-right corner or center of the screen.
The Join Scratch window opens.
2. Type a username into the field labeled Choose a Scratch Username.



Your username is how Scratch will know you and how other users will see you when you start to share programs. Be creative! Choosing a username can be fun! For everyone's safety, a good username shouldn't reveal any personal information, such as your full name, age, gender, or address. Try personalizing your username by including the name of your favorite sports team or musical group.

3. Choose a password and enter it into the Password and Confirm Password fields.



When creating your password, avoid using information that other people are likely to know, such as your address or birthday. Your password should be something you can remember, but it should also contain numbers or punctuation to make it more secure.

4. Click Next.
You see the second screen of the signup form.
5. Enter your birthday, gender, and country and then click Next.
6. When you're asked for an email address, enter your email address in both the Email Address and Confirm Email Address fields and click Next.
Scratch sends a confirmation email to the address you provided.
7. Click the OK Let's Go! button.
8. Check your email. When you get the email from Scratch, click the link in it to confirm your account.

Now you're ready to go. The next section tells you how to start coding!

Meeting Scratch the Cat

After you've joined Scratch and you're ready to start coding, click the Create tab in the top menu of the screen. When the new page loads, you see the scratch Project Editor, which looks like Figure 1-3.

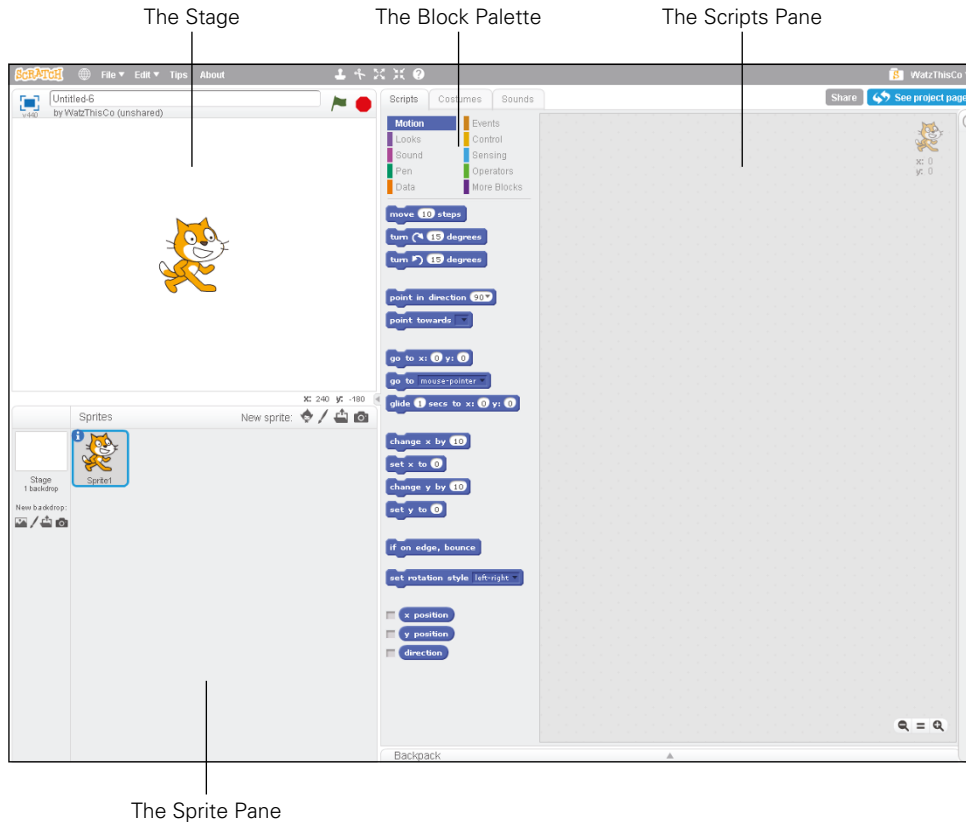


FIGURE 1-3 The Scratch Project Editor

Don't worry too much about what the things on this screen do. We'll be talking in detail about each part later. For now, let's build something!

See that cool cat in the middle of the screen? Her name is Scratch the Cat. Every new Scratch program starts with her sitting right there, waiting for instructions from you. The area where she lives is called the Stage. This is where all the action of your program takes place.

Below the Stage is an area called the Sprite Pane. The Sprite Pane shows small images of each of the characters (also known as sprites) in your program.

To the right of the Stage is a rectangle containing differently shaped blocks. This is called the Block Palette. Think of it like a painter's palette, where a painter selects the colors that she wants to paint with before combining those colors to make a painting on a canvas.

To the right of Block Palette is the Scripts Area. This is your canvas, where you put together the blocks selected from the Block Palette to make your sprites do things.

Moving Scratch Around

To get a better idea of how all the pieces fit together, follow these steps to make Scratch the Cat do something:

1. Find the block that says `move 10 steps`. This is the `move ()` block. It looks like Figure 1-4.



FIGURE 1-4 The `move` block

2. Click the `move ()` block and drag it into the Scripts Area. Your Scripts Area should now look like Figure 1-5.

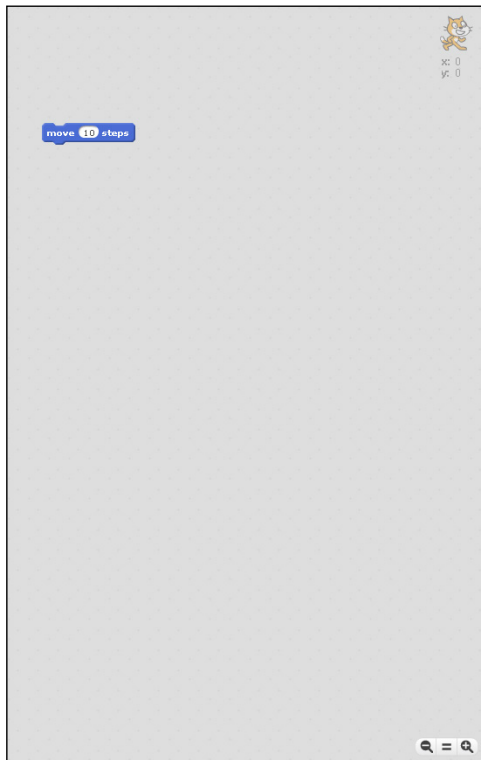


FIGURE 1-5 The Scripts Area, with one block

3. Double-click the `move` () block and watch what Scratch the Cat does. Did you see it? She moved slightly to the right.
4. Click the number 10 inside the `move` block to highlight it and change the 10 to 20.
5. Double-click the block. Now Scratch moves twice as far as she did before.
6. Try changing the value in the `move` () block to an even larger number and see what happens.

Connecting Blocks

Notice that the `move` block looks a bit like a puzzle piece. It works a bit like a puzzle piece too! Any time you see a block shaped like this, you know that you can attach it to another block.

Follow these steps to put some blocks together to make Scratch the Cat do something more complicated:

1. Find the `turn clockwise` block in the Block Palette. Figure 1-6 shows what the block looks like.

Notice that it has the same shape as the `move` block.



FIGURE 1-6 The `turn clockwise` block

2. Drag the `turn clockwise` block to the Stage and snap it to the bottom of the `move` block. When the blocks are connected, they look as shown in Figure 1-7.



FIGURE 1-7 Your first connected blocks

Double-click the combination of blocks and watch what happens on the Stage. The instructions on the first block happen, and then the instructions on the second block happen!

By right-clicking the combination of blocks and selecting Duplicate, from the menu you can create an exact copy of the block combination. Try it out! Then snap the two combinations of blocks together so that they look like Figure 1-8.



FIGURE 1-8 Duplicating blocks

Now click the whole thing and watch what Scratch the Cat does.

Looping Movements

If you want to make Scratch the Cat do this turning and moving thing over and over, you could keep making as many duplicates as you want of these same blocks, or you could create what's called a **loop**.



A **loop** is a block that causes the commands contained within it to repeat one or more times.

Use the following steps to create a loop:

1. Separate the second two blocks from the first two by clicking the second **move** block and dragging downward.
2. Right-click the second set of blocks and then select Delete from the context menu to remove the block set from the Stage.
3. Click the word Control in the color-coded menu in the Block Palette. Then find the **forever** block (see Figure 1-9) inside the Control Block Palette.