

GET CODING 2!

WRITTEN BY DAVID WHITNEY

ILLUSTRATED BY DUNCAN BEEDIE



CANDLEWICK PRESS

CONTENTS

ABOUT THIS BOOK	4
INTRODUCTION	6
HOW TO USE THIS BOOK	14
KEY CODE SKILLS	15
MISSION 1: Noughts and Crosses	18
MISSION 2: Snake	66



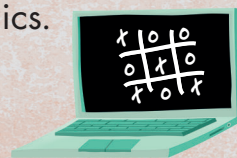


MISSION 3: Table Tennis	124
MISSION 4: Endless Runner	164
MISSION 5: Side-scrolling Platformer	208
WHAT NEXT?	220
INDEX	222

ABOUT THIS BOOK

Welcome to *Get Coding 2!*, where you're going to learn all about games and gaming. In this book, you will explore the fascinating history of computer games and learn about some of the most famous and popular games that are played all over the world. Best of all, you'll discover how to code your own games, from simple turn-based games like *Noughts and Crosses* (tic-tac-toe) to more complicated games with graphics.

Are you ready for some exciting coding missions?



What does this book teach you?

We all know how important computers are and that they can do incredible things. You can play games that are so realistic, you feel like you are right there in the world of the game. But computers can't do things by themselves. When you play your favorite computer game and there are sneaky enemies, complicated graphics, and atmospheric music, it's because someone has written a set of instructions for your computer to follow so it can create all the different elements. These instructions are known as a program, and they have to be written in a language, or code, that the computer can understand.

This book is going to teach you how to code web-based computer games using the programming languages HTML and JavaScript. HTML and JavaScript are two of the most popular languages in the world. You might not have written programs using these languages before, but every time you use the Internet, whether it is to check a website or play a game, you will be using programs created in them. HTML builds the basic structure of a web page, and JavaScript makes the page interactive so it responds to the user. Using these two languages, you can create games that will challenge the player and give endless hours of fun.

Go to
www.getcodingkids.com
for help and support

How does this book work?

There are five chapters in this book. In each chapter, there is a top-secret mission that teaches you how to code a different game. Using JavaScript, you will build different types of games, make your game respond to the player, discover how to code simple graphics, create animations, and even add artificial intelligence (AI) to your game. In each mission, you will be joining the Lucky Cat Club: Ruby, Markus, Grace, Rusty, and Scratch. They need your help to defeat SaberTooth Studios, their gaming rivals.



Are you
ready to get
coding?



We hope you enjoy this book and that it inspires you to get coding!

INTRODUCTION COMPUTER GAMES



Every day we use computers of different shapes and sizes to do a huge range of things. Computers are made up of hardware (the things you can touch such as the keyboard and screen) and software (the things you can't touch but that a computer needs to work). An important type of software is the program: a detailed set of instructions that the computer follows in order to complete a task. Programs are written in code, and writing programs is known as coding. Games are special kinds of programs that run on your computer, but instead of letting you browse the Internet or do your homework, they let you play and have fun.



If you have access to a computer, phone, tablet, or games console, you've probably played a game. They're everywhere! *Pokémon GO*, *Super Mario Bros.*, *Minecraft*, and *Angry Birds* are all popular computer games. But what is the secret to making a great game?

Why do we play?

People play games for all different reasons: some love the challenge of beating a score or obstacle; others enjoy competing with other players or the thrill of exploring new worlds and experiences. Games can be about anything—from made-up characters and universes to football games or puzzles. They often fall into categories, which game designers call genres.

Who do we play with?

Games can involve a single player playing against a computer. They can also be played over the Internet, which means you and someone thousands of miles away could be working together to defeat the same enemy. When you play against a computer—whether the computer acts as the second player in a game of chess or the enemy that has to be defeated at the end of a level—the computer isn't thinking and acting by itself. It has been programmed to have artificial intelligence (AI), so it responds in the same way a player would. But no matter what the game's genre or how many players are involved, there are certain things that all games have in common.

What makes a great game?

A great game makes you want to play it again and again. Game designers spend hours putting the different elements of a game together to give the player the best experience possible. A game needs a good character or story and a world that you believe in. Interesting **graphics** and sounds are often used to create an atmosphere. The game needs to be fun to play, but it must also contain the right level of challenge. Often a game will start off easily and then get more difficult.

One thing that all games have in common are rules that tell the player what they can and can't do. Good games have rules that surprise the player or are difficult to win against. A game also needs to have a clear goal. Does the player have to solve a puzzle, defeat an enemy, or collect objects? Often there will be a mixture of small goals, such as collecting an object, and big goals, such as defeating an enemy at the end of a level. The game needs to be designed in a way that gives the player feedback if they achieve or fail to achieve their goal. They might score points, beat the clock, or progress to the next level. Or they might lose points or lives.

A great game has the perfect balance of all these different elements.



GRAPHICS are the things you see on your computer screen that aren't text. Graphics can be photos, illustrations, diagrams, or images.

Game Genres

- **Action:** a fast-paced game in which the player gets caught up in the drama, often having to survive against different enemies or threats
- **Combat:** a game in which the player has to defeat one or several opponents
- **Platformer:** a game in which the player has to run and jump from platform to platform in levels of increasing complexity
- **Puzzle:** a game that involves solving a problem, often under time pressure
- **Role-playing:** a game in which the player becomes a character, normally in a fantasy world; the game often involves a magical storyline and the player completing a quest
- **Sandbox:** a game in which the player can explore the world at their own speed and create their own adventures
- **Simulation:** a game in which the world or characters are similar to real life
- **Sport:** a game in which the player plays a sport such as football or drives a racecar
- **Strategy:** a game that makes the player think and make choices; the player will often have to make decisions that affect the whole world and many characters

Let's take a look at the history of gaming.



A BRIEF HISTORY OF GAMING

We love to play, and for thousands of years, people have been inventing games. You've probably played a board game such as Scrabble, Monopoly, or Clue. But did you know that some board games date back thousands of years to ancient Egypt?

Computer games are a much more recent invention, but they have lots in common with simple board games. In fact, the first computer games were so basic that you probably wouldn't recognize them compared to today's games with their sophisticated graphics, music, and multiplayer capabilities.



1950s: The first computer games

For as long as we've had computers, people have wanted to play with them.

In the 1950s, computers were room-size pieces of machinery that you couldn't interact with much. But a mathematician named A. S. Douglas managed to develop a rudimentary computer tic-tac-toe game called OXO. The game's grid and progress were displayed on a small, primitive screen. In 1958, physicist William Higinbotham created a slightly more sophisticated game called *Tennis for Two*, in which each player used a control knob and a button to hit the ball over the net.

In 1961, computer scientists at the Massachusetts Institute of Technology inaugurated hacker culture by exploring what their new computer, the DEC PDP-1, could do. Although it still didn't look much like today's computers, the PDP-1 had a screen and typewriter and was programmable. Of course one of the many programs the MIT crew developed was a game.

1960s: *Spacewar!*

The first computer game for this PDP-1 at MIT was written in 1962 by three developers: Martin Graetz, Steve Russell, and Wayne Wiitanen. They called the game *Spacewar!*

It was a two-player game, with each player controlling a spaceship. The aim of the game was to shoot at the other ship. The game was immediately popular and shared with other universities for them to test on their new PDP-1 computers.

At the time *Spacewar!* was written, the few computers that existed all had their own programming languages, which meant that a game written for one computer probably would not run on another. During the 1960s, lots of small games were written for the big expensive computers in universities and schools because of how much fun everyone had with *Spacewar!*

In the 1960s, programming languages that ran on more than one computer, such as BASIC, were invented. This meant that games could be shared more widely and as a result became increasingly popular.

1970s: Early games and arcades

In the 1970s, Nolan Bushnell started a company called Atari to make games that could be played in amusement arcades and theme parks. Their second product, a table tennis game called *Pong*, was a huge hit and started the computer game industry, which is now worth billions of dollars. These early games were made from electronic circuits and didn't even use code to run.

At the same time, a company called Magnavox released its first game **console**, a home computer that plugged into a TV, which had its own version of a table tennis game included.



A game **CONSOLE** (also known as a home console) is a small computer that connects to a television or monitor for playing computer games.



Did you know?

Early home computers let you create games by typing in code. You could buy magazines full of code for different games. All you had to do was type it into your computer to run the game.

Games have changed so much throughout history!



1980s: Consoles

By the early 1980s, Nolan Bushnell's dreams had come true. Arcades were everywhere and full of 2-D games that are now regarded as classics, including *Pac-Man*, *Missile Command*, and *Donkey Kong*—the first game to feature the character Mario. During this period, Atari and Commodore continued to launch new consoles. The first consoles from Nintendo and Sega were also released.

In 1984, the IBM PC/AT and the Apple Macintosh, the two computers that are most like the ones we use today, went on sale. Games for these new computers were launched as well.

1990s: Going 3-D

Games in the 1990s became faster, more colorful and complicated, and featured iconic characters such as Sonic the Hedgehog and Mario. Sports games started to become popular, with the first version of *FIFA* released in 1993. In the late 1990s, 3-D graphics took over with games like *Doom* on the IBM PC and *Super Mario 64* on the Nintendo 64.

2000s: Blockbusters

Through the 2000s, games became as popular as films—and began to make more money than blockbusters. Sony developed the PlayStation and introduced characters such as Crash Bandicoot and games that are still popular today, including *Final Fantasy*, *Metal Gear Solid*, and *Gran Turismo*.

Microsoft joined the gaming industry, releasing their Xbox console in 2001 with bestsellers such as *Halo* and *Fable*. Game series from the 1990s, such as *Super Mario* and *The Legend of Zelda*, reinvented themselves to stay fresh.

Gaming today

In the late 2000s and early 2010s, games continued to evolve, with gamers fighting dragons in massive multiplayer games such as *World of Warcraft* and unleashing their inner rock star in *Guitar Hero*. The Nintendo Wii, released in 2006, brought games to people who previously didn't think computer games were for them.

From 2008, there was an explosion in cell-phone games. Some of the most popular are puzzle games like *Angry Birds*, endless runner games like *Canabalt*, and addictive matching games like *Candy Crush Saga*. As phones developed into tablets, the same games could be played on them too.

Today, companies such as Atari and Commodore no longer exist, but many games and characters are still the same. Mario has appeared in over 200 games since 1981 with more than 240 million copies sold worldwide.



Famous games

- *Doom*
- *FIFA*
- *Fortnite*
- *GoldenEye 007*
- *Halo*
- *Mario series*
- *Minecraft*
- *Pac-Man*
- *Rock Band*
- *Sonic the Hedgehog*
- *World of Warcraft*
- *The Legend of Zelda*

Famous characters

- *Angry Birds*
- *Crash Bandicoot*
- *Donkey Kong*
- *Lara Croft*
- *Master Chief*
- *Mario*
- *Pac-Man*
- *Sonic*

GREAT GAMES AND COOL CONSOLES

OXO by A. S. Douglas (1952)

Pong by Atari (1972)

Donkey Kong by Nintendo (1981)

Commodore 64 released (1982)

Nintendo Game Boy released (1989)

Sega Genesis released (1988)

Nintendo NES released (1983)

IBM PC/AT and Apple Macintosh released (1984)

Sony PlayStation released (1994)

Nintendo 64 released (1996)

Snake by Nokia (1998)

Sega Dreamcast released (1998)

Sony PlayStation 3 and Nintendo Wii released (2006)

Microsoft Xbox 360 released (2005)

Microsoft Xbox released (2001)

Sony PlayStation 2 released (2000)

**Apple iPhone released (2007)
Apple iPad released (2010)**

Canabalt by Semi-Secret Software (2009)

Sony PlayStation 4 and Microsoft Xbox One released (2013)

Nintendo Switch released (2017)

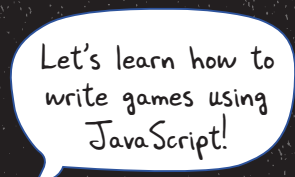
I love my console!



HOW TO CODE A GAME

Did you know that even though games have become bigger and more complicated, the way we make them is still the same as in the 1980s? Games are built by teams of people. The game designers work out what should be in a game to make it fun. Game programmers and artists then take their designs and write programs and create graphics to bring the game to life.

Like all programs, games can be written in lots of different languages, but some of the most commonly used languages to write games are C, C++, C#, and JavaScript. We're going to learn how to write games using JavaScript. This means that the game will run in a web browser.



JavaScript

JavaScript is one of the most popular programming languages in the world. It runs on lots of different types of computers, including right inside your web browser, the program that you use to access the Internet. You can use JavaScript with other programming languages such as HTML and CSS to create all kinds of web programs, apps, and games. A programming language is built from special keywords that the computer understands, along with a set of rules about how you write the code. We call the keywords and these rules the syntax of a programming language.

How to Write JavaScript

JavaScript syntax looks complicated since it is made up of a mixture of words and symbols. All you are really doing, though, is storing information, or data, so your computer can use it to do the things you want it to do. In JavaScript, we use variables to store data. You define a variable by using the variable keyword and then giving the variable a name. Then you use the assignment operator (=) to give your variable a value. Variables store three types of values: strings, numbers, and Booleans (a type that has two values—usually true or false):

```
var aString = "A string is a set of words";  
var aNumber = 123;  
var aBoolean = true;
```

variable keyword

variable name

assignment operator

variable value

We use variables with operators such as equal to (==), addition (+), and less than (<) and with statements such as the if statement to make a program that can perform actions. We can also use built-in functions to do things in the browser, such as pop up alerts. Look at this example:

```
var javascriptIsFun = true;  
var message = "JavaScript is fun!";  
if(javascriptIsFun) {  
    alert(message);  
}
```

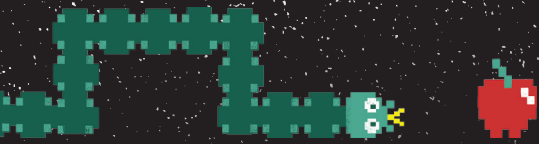
if statement

variable

built-in alert function

Run this code and an alert will pop up in your browser. Find out how to do this in the next section.





HOW TO USE THIS BOOK

The Lucky Cat Club needs your help! Ruby, Markus, Grace, and Rusty have to learn how to code five different games so they'll be ready for the Game On Hackathon. They need to beat their rivals SaberTooth Studios and show them that gaming is for everyone. Your job is to work through each mission and master all the different coding skills needed to build the games.

Mission Briefs

At the start of each chapter, you will receive a Mission Brief from a Lucky Cat Club member that will tell you all about the game they want to code. Keep an eye out for the new skills you are going to learn to code *Tic-Tac-Toe*, *Snake*, *Table Tennis*, and much more.



Game Builds

The code for each of the games has been broken down into bite-size chunks.

Follow the step-by-step instructions in each Game Build to create your game. And don't worry if you get stuck because you can always go to the *Get Coding!* website (www.getcodingkids.com), where you can find every piece of code in this book.

Games

By the end of each mission, you will have built a game of your own and picked up lots of new skills on the way. Have fun playing the game and then go to the *Get Coding!* website to find ideas for how to develop it further.

The Developer's Dictionary

The Lucky Cat Club's favorite website is the Developer's Dictionary. Learn about the history of games and gaming and use this information to help you complete the missions.

KEY CODE SKILLS

Before you receive the brief for Mission 1, there are some basic Code Skills you need to learn. You will use these skills throughout the entire book, so it's important to get a handle on them now. You can code using a PC or a Mac, but you have to create and save your HTML file in a different way depending on which system you are using.

KEY CODE SKILL 1 ► CREATING A FOLDER

You need to have a place in your computer where you can save all your HTML files. Make a folder on your desktop called **Coding 2**. It's really important that you save all your HTML files in the same place, so make sure you keep using this folder as you work through the missions.

PC

On a PC, right-click on your desktop and click *New* and then *Folder*. Call your new folder **Coding 2**.

Mac

On a Mac, hold down the Control key and click on your desktop. Then select *New Folder*. Call your new folder **Coding 2**.



Master these essential Code Skills and you'll be ready for the first mission!



KEY CODE SKILL 2

► CREATING AN HTML FILE

You need to know how to create an HTML file so you can write code. Programmers normally use specialist software to write code (see page 221 for details), but all computers come with text-editing programs that let you write HTML files. If you have a PC, you can use Notepad. If you have a Mac, you can use TextEdit.

PC

On a PC, you'll find Notepad by going into the *Start* menu and typing it in the search bar.

Mac

On a Mac, you'll find TextEdit by typing it in the *Spotlight* search magnifying glass in the top right of your screen. When you open TextEdit, you need to do the following things:

- Set up your file as a plain text (rather than rich text) file. To do this, go to *Format* in the menu bar and select *Make Plain Text*.
- Also go to *TextEdit* in the menu bar. Select *Preferences*. In the *New Document* tab in the *Format* section, make sure *Plain text* is checked. In the *Options* section, make sure *Smart quotes* is unchecked.
- In the *Open and Save* tab in *Preferences*, make sure *Display HTML files as HTML code instead of formatted text* is checked.

KEY CODE SKILL 3

► SAVING YOUR HTML FILE

When you save your HTML file, you need to make sure you save it using the file extension **.html** at the end of your file name. Your computer uses file extensions to work out how to open files. By giving your file the extension **.html**, you're telling your computer that it should open the file in a web browser.

PC

On a PC, you need to:

- Go to *File* and select *Save As*.
- Select the **Coding 2** folder as the destination to save the file to.
- Choose a name for your file, such as *Mission1*, and type it into the *File name* bar.
- After the name of the file, type **.html** so your file name reads **Mission1.html**. Click *Save*.

Mac

On a Mac, you need to:

- Go to *File* and select *Save*.
- Select your **Coding 2** folder as the destination to save your file to.
- Choose a name for your file, such as *Mission1*, and type it into the *Save As* bar.
- After the name of the file, type **.html** so your file name reads **Mission1.html**.
- Make sure the check box *If no extension is provided, use ".txt"* is unchecked. Click *Save*.

KEY CODE SKILL 4

► OPENING YOUR HTML FILE

To see your code displayed on-screen, you need to open your HTML file in a web browser. You then might want to go back into your text-editing program to make some changes to your code.

PC

On a PC, you need to:

- Save your file, as described in Key Code Skill 3.
- Open your **Coding 2** folder on your desktop. Double-click on your HTML file. It will open in your web browser.
- When you want to edit your code, right-click on the HTML file in your **Coding 2** folder. Select *Open with* and choose Notepad.

Mac

On a Mac, you need to:

- Save your file, as described in Key Code Skill 3.
- Open your **Coding 2** folder on your desktop. Double-click on your HTML file. It will open in your web browser.
- When you want to edit your code, right-click on the HTML file in your **Coding 2** folder. Select *Open with* and choose TextEdit.



KEY CODE SKILL 5

► USING DEVELOPER TOOLS

As you code the Game Builds and open your code in the browser, you may find that the code isn't displaying in the way you expected. Don't worry! You can use the developer tools that are built into your browser to help you find the bug. You'll find the developer tools in your browser settings. On a PC, try the keyboard shortcut *F12*. On a Mac, try the shortcut *Command + Option + I*. Or right-click on the page and select *Inspect*. Go to the elements tools in the console and look for where the errors are occurring.

KEY CODE SKILL 6

► USING THE *GET CODING!* WEBSITE

Don't forget that as you work through the book, you can use the *Get Coding!* website to help you with your missions. If you get stuck at any point when you're writing code, go to the website and see what your code block should look like. You can even copy and paste the code blocks from the website into your text-editing program. You will find all the files you need for Mission 5 on the website.

The *Get Coding!* URL is www.getcodingkids.com.