

Extracted from:

# 3D Game Programming for Kids, Second Edition

Create Interactive Worlds with JavaScript

This PDF file contains pages extracted from *3D Game Programming for Kids, Second Edition*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2018 The Pragmatic Programmers, LLC.

All rights reserved.

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, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Raleigh, North Carolina

The  
Pragmatic  
Programmers

# 3D Game Programming for Kids

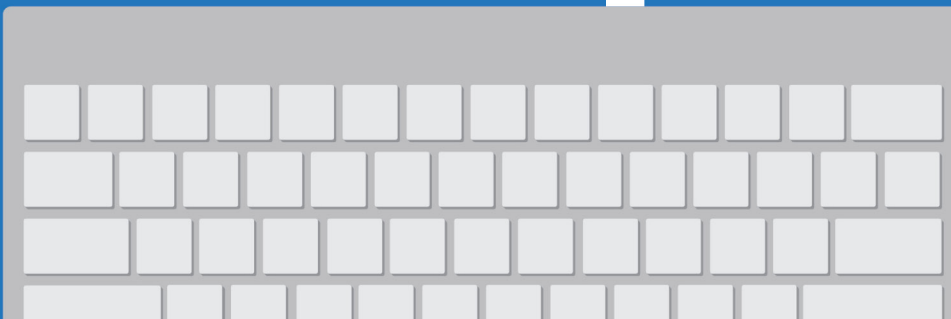
Second Edition



Create Interactive  
Worlds with JavaScript

Chris Strom

*edited by Adaobi Obi Tulton*



# 3D Game Programming for Kids, Second Edition

Create Interactive Worlds with JavaScript

Chris Strom

The Pragmatic Bookshelf

Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at <https://pragprog.com>.

The team that produced this book includes:

Publisher: Andy Hunt

VP of Operations: Janet Furlow

Managing Editor: Brian MacDonald

Supervising Editor: Jacquelyn Carter

Development Editor: Adaobi Obi Tulton

Copy Editor: Paula Robertson

Indexing: Potomac Indexing, LLC

Layout: Gilson Graphics

For sales, volume licensing, and support, please contact [support@pragprog.com](mailto:support@pragprog.com).

For international rights, please contact [rights@pragprog.com](mailto:rights@pragprog.com).

Copyright © 2018 The Pragmatic Programmers, LLC.

All rights reserved. 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, or otherwise, without the prior consent of the publisher.

ISBN-13: 978-1-68050-270-1

Book version: P1.0—July 2018

# Introduction

---

Welcome to the world of programming!

I won't lie; it can be a frustrating world (it makes me cry at least once a week). But it's totally worth the pain. You get to make this world do whatever you want. You can share your world with others. You can build things that really make a difference.

This book that you hold in your hands is a great way to get started with programming. Why? Because this book believes the best way to learn programming is by *playing*. Oh, there are a few chapters that describe fundamentals, but we only get to those after having fun. So we jump right into some pretty cool 3D animation in Chapter 1.

This is going to be a blast!

## How I Learned to Program (and Why That Matters to You)

When I was a kid, I copied computer game programs out of books. This was a long time ago, so I bought books with nothing but computer code, and typed it into my computer. It took a while, and I had no idea what I was doing at first.

Eventually, I started to recognize things. And then I started to change some of the programs—little things at first—to see what happened. Then I started making bigger changes. Eventually I got pretty good at it and could even write my own programs.

I hope that this book will let you do the same, but with one important difference: I'll explain what's going on so you won't have to guess quite as much.

## How YOU Can Learn to Program

Everybody is different. Everybody learns differently.

Because of that, there are at least three different ways that you can learn from this book:

1. Play with cool stuff, then read an occasional chapter on fundamentals.
2. Learn the basics, then make some cool stuff with what you know.
3. Just type the code (like I did when I was a kid).

You choose which works best for you!

If you want to play first (the first option), start with Chapter 1 and then go in order through the rest of the book. You'll mostly work on games and simulations in project chapters—chapters whose titles start with the word “Project”—followed by occasional chapters on fundamental skills. If you're not sure which option is best, choose this one. This is the way I wish I could have learned.

If you're the kind of person who likes to understand fundamentals before building bigger things (the second option), then read the “essentials” chapters first—any chapter without “Project” in the title. There's still plenty of coding in them and 3D fun in some of them. Compared to other programming languages, the core of JavaScript is fairly small. You can learn 80–90 percent of JavaScript's core just by reading those chapters. Being able to use it well is another thing—which is where the “Project” chapters come in!

If you just want to code (the third option), then flip to [Appendix 1, Project Code, on page ?](#). All the code from all the games is there. If you get stuck with some of the coding, then flip to the chapter that the code came from for a deeper explanation. This is how I learned, and I turned out mostly OK!

No matter which option is best for you, there's one rule: *always type the code in by hand*. This is much slower than copying and pasting. You're much more likely to make mistakes typing code in yourself.

Going slow and making mistakes is the point!

Typing code by hand makes you think more about what you're typing. You might believe that you can learn just as well by reading and then copying and pasting, but this is 100 percent not true. Even if you've been programming for 50 years, you should never copy and paste code. Typing gives you time to think and to understand the code you're adding. That is *far* more important than getting it done quickly.

Making mistakes is part of programming. Being able to fix broken code is just as important as being able to program. So make mistakes. It will be hard and frustrating. And it will be worth it.

This means that we've reached the first tip in the book! Especially important tips and guidelines are highlighted throughout the book. Pay attention to them because they can really make a difference as you explore this world.

---

#### Always Type Code / Never Copy and Paste!

---



Good programmers always type code by hand—they never copy and paste code. Writing good computer programs is much more important than writing them quickly. Good programming means thinking about the code that you create. It means understanding code. So type it out!

---

One more thing: if you get stuck, you can always get help!

## Getting Help

Every programmer needs help. If you've never programmed in your life, you will need help. If you've been programming for 50 years, you'll still need help.

The two rules for asking for help:

1. Try to figure it out yourself at first.
2. Don't be afraid to ask for help.

Knowing a programming language really well is *not* the most important skill a programmer can have. The most important skill is problem-solving. Knowing everything about a computer language does not mean that you'll create 100-percent problem-free code. It helps, but problems still happen. A lot. So try to figure out problems on your own. Even if you can't solve a problem, you're still improving your problem-solving skills.

Speaking of problem-solving skills, [Chapter 2, Debugging: Fixing Code When Things Go Wrong, on page ?](#) is the place to get started. That chapter describes common mistakes, how to solve problems using the code editor and the browser, and how to recover when things go really, really wrong.

---

#### Don't Skip Chapter 2!

---

Problem-solving is crazy important, so be sure to read Chapter 2.



No matter how you learn, the debugging skills described in Chapter 2 are *must reading*. It might seem like you can skip that stuff, but skipping that chapter is like playing sports without pads or driving without a seat belt. Things might seem fine without the safety of debugging skills. Right up until the point that something goes really wrong.

---

It is 100 percent OK if you can't figure out a problem on your own. I am happy to help. This book has its own web page where you can find all the code for this book, as well as the book's community forum.<sup>1</sup> If you get stuck, post a question to the forum. I almost always answer within 24 hours. Just be sure to tell me how you've already tried to solve the problem—otherwise that's going to be my first question to you.

But please, I want you to do well. Ask questions. Use the forum. Let's code!

## What You Need for This Book

You'll need the Google Chrome web browser and a relatively new computer. That's it!

Not all web browsers can generate the cool 3D-gaming objects that we'll build in this book. To get the most out of the book, you should install the Google Chrome web browser on your computer.<sup>2</sup> Other web browsers will work, but some of the exercises in this book rely on features available only in Google Chrome.

Any modern computer (from the past five years) with Google Chrome installed will be sufficient. To be sure, you can test your computer's capabilities by visiting the Get WebGL site.<sup>3</sup> If that site doesn't tell you that your browser supports WebGL, then you may have to try another computer.

## What Is JavaScript?

There are many, many programming languages. Some programmers enjoy arguing over which is the *best*, but the truth is that all languages offer unique and worthwhile things.

In this book we use the JavaScript programming language. We program in JavaScript because it's the language of the web. It's the only programming language that all web browsers understand. If you can program in JavaScript, not only can you make the kinds of games that you'll learn in this book, but you can also program just about every website there is.

- 
1. <https://talk.code3Dgames.com/>
  2. <https://www.google.com/chrome/>
  3. <http://get.webgl.org/>



## What's New in the Second Edition?

This is the second edition of *3D Game Programming for Kids*. The first edition was *awesome*. I've been told that I'm biased, but I don't see it. I'm pretty sure the first edition really was close to perfect.

Right, Chris, if it was perfect, why make a second edition?

Well, first, a lot has happened since the first edition of the book. The programming world is always changing. There's always new stuff coming out. Most of that new stuff won't help us learn, but every now and then, some of it really makes a difference. Three years after the first edition came out, there was enough helpful new stuff that I believed I could make the second edition even better than the first.

The other reason for a second edition is that I've changed a lot since the first edition. As a programmer you can never stop learning. I've been programming for 15 years and am always working to learn as many new things as I can. I've also worked hard to become a better programmer, teacher, and writer. This learning helps me to write even better books!

This is a very different book than the first edition. I removed a few chapters and added some completely new ones. All of the remaining chapters and code have been significantly rewritten. Many of the changes add cool new features (flying controls in [Chapter 5, Functions: Use and Use Again, on page ?](#) or fire special effects). The new features are fun, but they're not the main reason for the changes.

The changes are for you. They make the book more fun. They make learning easier. They better help you become a programmer.

So what's new? Pretty much everything. Except for the parts that really were already perfect.

## What This Book Is Not

Just to be clear:

*We're not going to become experts in JavaScript.*

And...

*We're not going to become experts in 3D Game Programming.*

We'll cover a ton in this book, but we won't be ready to create the next Facebook when we're done. And we won't be able to recreate MarioKart. Both of those require hundreds of programmers working for hundreds of hours—often with more advanced skills than we will cover in this book.

But we will learn the most important parts of JavaScript. And we'll learn lots of 3D skills. We can do some amazing things with these skills. And we'll be ready to get started on even bigger, even more amazing things.

## Let's Get Started!

Enough introduction—let's jump into programming!