

# Build Your Own 2D Game Engine and Create Great Web Games

Using HTML5, JavaScript, and WebGL



**Kelvin Sung**  
**Jebediah Pavleas**  
**Fernando Arnez**  
**Jason Pace**

With

Original Dye Characters Design: **Nathan Evers**

Other Game Character design and game arts: **Kasey Quevedo**

Figures and Illustration: **Clover Wai**

**Apress®**

## **Build Your Own 2D Game Engine and Create Great Web Games: Using HTML5, JavaScript, and WebGL**

Copyright © 2015 by Kelvin Sung, Jebediah Pavleas, Fernando Arnez, and Jason Pace

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

All visual and audio assets included with the sample projects in this book are protected by the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 license (CC BY-NC-SA 3.0) <http://creativecommons.org/licenses/by-nc-sa/3.0/>. You may adapt and share the materials and create derivative works, but you may not use the material for commercial purposes and all derivative works must be distributed under the CC BY-NC-SA 3.0 license.

ISBN-13 (pbk): 978-1-4842-0953-0

ISBN-13 (electronic): 978-1-4842-0952-3

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr

Lead Editor: Ben Renow-Clarke

Technical Reviewer: Jason Sturges

Editorial Board: Steve Anglin, Mark Beckner, Gary Cornell, Louise Corrigan, Jim DeWolf,

Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham,

Susan McDermott, Matthew Moodie, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke,

Gwenan Spearing, Matt Wade, Steve Weiss

Coordinating Editor: Melissa Maldonado

Copy Editor: Kim Wimpsett

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springer.com](http://www.springer.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit [www.apress.com](http://www.apress.com).

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at [www.apress.com/bulk-sales](http://www.apress.com/bulk-sales).

Any source code or other supplementary material referenced by the author in this text is available to readers at [www.apress.com](http://www.apress.com). For detailed information about how to locate your book's source code, go to [www.apress.com/source-code/](http://www.apress.com/source-code/).

*To my wife, Clover, and our girls, Jean and Ruth, for completing my life.*

*—Kelvin Sung*

*To my nieces, Marley and Monroe, for the utter joy and inspiration they bring to those around them.*

*—Jebediah Pavleas*

*To my parents whose support carried me so many times, and my grandmother for being forever understanding.*

*—Fernando Arnez*

*To my husband, Craig, and my mother, Linda, for their constant patience, support, and encouragement.*

*—Jason Pace*



# Contents at a Glance

<b>About the Authors.....</b>	<b>xv</b>
<b>About the Technical Reviewer .....</b>	<b>xvii</b>
<b>Acknowledgments .....</b>	<b>xix</b>
<b>Introduction .....</b>	<b>xxi</b>
<b>■ Chapter 1: Introducing 2D Game Engine Development with JavaScript.....</b>	<b>1</b>
<b>■ Chapter 2: Working with HTML5 and WebGL .....</b>	<b>15</b>
<b>■ Chapter 3: Drawing Objects in the World .....</b>	<b>47</b>
<b>■ Chapter 4: Implementing Common Components of Video Games .....</b>	<b>77</b>
<b>■ Chapter 5: Working with Textures, Sprites, and Fonts .....</b>	<b>127</b>
<b>■ Chapter 6: Defining Behaviors and Detecting Collisions .....</b>	<b>187</b>
<b>■ Chapter 7: Manipulating the Camera.....</b>	<b>233</b>
<b>■ Chapter 8: Implementing Illumination and Shadow .....</b>	<b>273</b>
<b>■ Chapter 9: Integrating Physics and Particles .....</b>	<b>365</b>
<b>■ Chapter 10: Supporting Camera Background .....</b>	<b>419</b>
<b>■ Chapter 11: Building a Sample Game: From Design to Completion.....</b>	<b>441</b>
<b>Index.....</b>	<b>457</b>



# Contents

**About the Authors.....xv**

**About the Technical Reviewer .....xvii**

**Acknowledgments .....xix**

**Introduction .....xxi**

**■ Chapter 1: Introducing 2D Game Engine Development with JavaScript..... 1**

    The Technologies..... 2

    Setting Up Your Development Environment..... 3

        Downloading and Installing JavaScript Syntax Checker..... 3

    Working in the NetBeans Development Environment..... 4

        Creating an HTML5 Project in NetBeans..... 5

        The Relationship Between the Project Files and the File System..... 8

    How to Use This Book..... 9

    How Do You Make a Great Video Game? ..... 9

    References ..... 13

        Technologies..... 13

**■ Chapter 2: Working with HTML5 and WebGL ..... 15**

    Introduction ..... 15

    Canvas for Drawing ..... 15

        The HTML5 Canvas Project..... 15

    Separating HTML and JavaScript ..... 18

        The JavaScript Source File Project..... 18

        Observations..... 23

<b>Elementary Drawing with WebGL .....</b>	<b>23</b>
The Draw One Square Project.....	23
Observations.....	30
<b>Abstraction with JavaScript Objects .....</b>	<b>30</b>
The JavaScript Objects Project.....	30
Observations.....	39
<b>Separating GLSL from HTML .....</b>	<b>39</b>
The Shader Source Files Project.....	39
Source Code Organization .....	43
<b>Changing the Shader and Controlling the Color .....</b>	<b>43</b>
The Parameterized Fragment Shader Project.....	43
<b>Summary.....</b>	<b>45</b>
<b>■ Chapter 3: Drawing Objects in the World .....</b>	<b>47</b>
Introduction .....	47
<b>Encapsulating Drawing .....</b>	<b>48</b>
The Renderable Objects Project .....	48
Observations.....	50
<b>Transforming a Renderable Object .....</b>	<b>51</b>
Matrices as Transform Operators .....	51
The glMatrix Library .....	53
The Matrix Transform Project .....	54
Observations.....	58
<b>Encapsulating the Transform Operator.....</b>	<b>58</b>
The Transform Objects Project.....	58
The Transform Object.....	59
<b>View, Projection, and Viewports .....</b>	<b>61</b>
Coordinate Systems and Transformations .....	61
The View Projection and Viewport Project.....	64



The Camera .....	71
The Camera Objects Project .....	71
Summary .....	76
■ <b>Chapter 4: Implementing Common Components of Video Games</b> .....	<b>77</b>
Introduction .....	77
The Game Loop .....	78
The Game Loop Project .....	79
Keyboard Input .....	86
The Keyboard Support Project .....	86
Resource Management and Asynchronous Loading .....	93
The Resource Map and Shader Loader Project .....	94
Game Level from a Scene File .....	103
The Scene File Project .....	103
Scene Object: Client Interface to the Game Engine .....	110
The Scene Objects Project .....	110
Audio .....	116
The Audio Support Project .....	117
Summary .....	125
Game Design Considerations .....	125
■ <b>Chapter 5: Working with Textures, Sprites, and Fonts</b> .....	<b>127</b>
Introduction .....	127
Texture Mapping and Texture Coordinates .....	128
The Texture Shaders Project .....	130
Drawing with Sprite Sheets .....	150
The Sprite Shaders Project .....	152
Sprite Animations .....	161
Overview of Animated Sprite Sheets .....	162
The Sprite Animation Project .....	163

Fonts and Drawing of Text.....	170
Bitmap Fonts .....	170
The Font Support Project.....	171
Summary.....	184
Game Design Considerations.....	184
<b>■ Chapter 6: Defining Behaviors and Detecting Collisions .....</b>	<b>187</b>
Introduction .....	187
Game Objects .....	188
The Game Objects Project .....	188
Chasing of a GameObject.....	196
Vectors Review .....	196
The Front and Chase Project .....	200
Collisions Between GameObjects.....	207
Bounding Box .....	207
The Bounding Box and Collisions Project .....	208
Per-Pixel Collisions.....	213
The Per-Pixel Collisions Project.....	213
Generalized Per-Pixel Collisions.....	221
Vector Review: Components and Decomposition .....	221
The General Pixel Collisions Project .....	224
Per-Pixel Collisions for Sprites .....	227
The Sprite Pixel Collisions Project.....	228
Summary.....	231
Game Design Considerations.....	231
<b>■ Chapter 7: Manipulating the Camera.....</b>	<b>233</b>
Introduction .....	233
Camera Manipulations .....	234
The Camera Manipulations Project.....	234
Interpolation .....	240
The Camera Interpolations Project .....	241

Camera Shake Effect.....	246
The Camera Shake Project.....	247
Multiple Cameras .....	253
The Multiple Cameras Project.....	254
Mouse Input Through Cameras .....	259
The Mouse Input Project.....	260
Summary .....	266
Game Design Considerations.....	267
<b>■ Chapter 8: Implementing Illumination and Shadow .....</b>	<b>273</b>
Introduction .....	273
Overview of Illumination and GLSL Implementation .....	274
Ambient Light .....	274
The Global Ambient Project .....	275
Light Source .....	281
GLSL Implementation and Integration into the Game Engine .....	282
The Simple Light Shader Project .....	283
Multiple Light Sources and Distance Attenuation .....	293
The Multiple Lights Project.....	294
Diffuse Reflection and Normal Mapping .....	302
The Normal Maps and Illumination Shaders Project .....	305
Specular Reflection and Materials .....	315
Integration of Material in the Game Engine and GLSL Shaders.....	318
The Material and Specularity Project .....	318
Light Source Types .....	327
The Directional and Spot Lights Project .....	328
Shadow Simulation .....	337
The Shadow Simulation Algorithm .....	339
The Shadow Shaders Project.....	340
Summary .....	356
Game Design Considerations.....	356

■ **Chapter 9: Integrating Physics and Particles** ..... 365

    Introduction ..... 365

    Physics Overview ..... 366

        Movement..... 366

        Collision Detection ..... 368

        Collision Resolution ..... 369

    Detecting Collisions..... 369

        The Rigid Shape Bounds Project ..... 370

    Resolving Collisions ..... 381

        The Rigid Shape Impulse Project..... 381

    Particles and Particle Systems..... 397

        The Particles Project..... 397

    Particle Emitters..... 408

        The Particle Emitters Project ..... 408

    Summary..... 412

        Game Design Considerations..... 413

■ **Chapter 10: Supporting Camera Background** ..... 419

    Introduction ..... 419

    Tiling of the Background ..... 420

        The Tiled Objects Project ..... 421

    Simulating Motion Parallax with Parallax Scrolling..... 426

        The ParallaxObjects Project..... 428

    Layer Management ..... 432

        The Layer Manager Project..... 432

    Summary..... 438

        Game Design Considerations..... 439

<b>■ Chapter 11: Building a Sample Game: From Design to Completion.....</b>	<b>441</b>
Part 1: Refining the Concept.....	441
Part 2: Integrating a Setting .....	444
Contextual Images Bring the Setting to Life .....	444
Defining the Playable Space .....	445
Adding Layout to the Playable Space .....	446
Tuning the Challenge and Adding Fun .....	447
Further Tuning: Introducing Enemies .....	448
General Considerations.....	448
Part 3: Integrating Additional Design Elements .....	449
Visual Design .....	449
Game Audio .....	450
Interaction Model.....	451
Game Systems and Meta Game.....	451
User Interface (UI) Design .....	452
Game Narrative.....	453
Bonus Content: Adding a Second Stage to the Level.....	454
Summary .....	455
<b>Index.....</b>	<b>457</b>



# About the Authors

**Kelvin Sung** is a professor with the Computing and Software Systems Division and the principal investigator of the Game-Themed Research Group at University of Washington Bothell (UWB). He received his Ph.D. in computer science from the University of Illinois at Urbana-Champaign in 1992. His background is in computer graphics, hardware, and machine architecture. He came to UWB from Alias|Wavefront (now part of Autodesk) in Toronto, where he played a key role in designing and implementing the Maya renderer, an Academy Award-winning image generation system. Before joining Alias|Wavefront, Kelvin was an assistant professor with the School of Computing, National University of Singapore. Kelvin's research interests are in studying the role of technology in supporting human communication. Funded by Microsoft Research and the National Science Foundation, Kelvin's recent work focused on the intersection of video game mechanics, real-world problems, and mobile technologies. Kelvin teaches both undergraduate and graduate classes in computer graphics, game development, and mobile computing.

**Jebediah Pavleas** is a graduate student in the Computer Science and Software Engineering program at the University of Washington Bothell (UWB) as well as an intern on the NExT Enable team at Microsoft Research. He is also the coauthor of the book *Learn 2D Game Development with C#*. He received a bachelor's of science degree in 2012 and was the recipient of the Chancellor's Medal for his class. During his time as an undergraduate, he took great interest in both computer graphics and games. His projects included an interactive math application that utilizes Microsoft's Kinect sensor to teach algebra, a 2D role-playing game designed to teach students introductory programming concepts, and a web site where students can compete in various mini-games to control checkpoints around campus. Relating to these projects, he coauthored publications in IEEE Computers and The Journal of Computing Sciences in Colleges (CCSC). When not working toward his graduate degree, he enjoys designing, building, and playing games of all kinds as well as adapting technology for improved accessibility.

**Fernando Arnez** is an undergraduate student in the Computing and Software Systems Division at the University of Washington Bothell (UWB) working toward his bachelor's degree in computer science and software engineering. He is a member of the Game-Themed Research Group and has participated in projects that built casual games for teaching introductory programming concepts. He coauthored an article in IEEE Computers discussing his work and the results from those projects.

**Jason Pace** directs the University of Washington Bothell's Digital Future Lab ([www.bothell.washington.edu/digitalfuture/about](http://www.bothell.washington.edu/digitalfuture/about)), an interactive media research and development studio modeling startup culture for a diverse group of student developers, designers, artists, musicians, and producers. He started the lab after spending 16 years at Microsoft leading user experience and product development teams on a number of Microsoft's key consumer products, including serving as a creative director and lead producer on the Halo team at 343 Industries and Design Manager for the Microsoft Casual Games group. His work in the lab focuses on exploring how radically diverse teams that seek to maximize differences among contributors can lead to unexpected insights and new directions in design and development. The lab brings undergraduate students together from across majors and schools to create high-performance creative teams engaged in both commercial game development and design research.





# About the Technical Reviewer

**Jason Sturges** is a cutting edge technologist focused in ubiquitous delivery of immersive user experiences. Coming from a visualization background, he's always pushing boundaries of computer graphics to the widest reach cross platform while maintaining natural and intuitive usability per device. From interactivity, motion, animations, and creative design, he has worked with numerous creative agencies on projects from kiosks to video walls to Microsoft Kinect games. Most recently the core of his work has been mobile apps.

Committed to the open source community, he is also a frequent contributor at GitHub and Stack Overflow as a community resource leveraging modern standards, solid design patterns, and best practices in multiple developer tool chains for web, mobile, desktop, and beyond.



# Acknowledgments

This book project is a direct result of the authors learning from building games for the Game-Themed CS1/2: Empowering the Faculty project, funded by the Transforming Undergraduate Education in Science Technology Engineering and Mathematics (TUES) Program, National Science Foundation (NSF) (award number DUE-1140410). We would like to thank NSF officers Suzanne Westbrook for believing in our project and Jane Prey, Valerie Bar, and Paul Tymann for their encouragements.

The invaluable collaboration between the technical team in the Game-Themed Research Group (<https://depts.washington.edu/cmmr/GTCS/>) and the design team in the Digital Future Lab (<http://www.bothell.washington.edu/digitalfuture>) at the University of Washington Bothell, where much of our learning occurred during the production of the many casual games for teaching introductory programming concepts, formed the foundation that allowed the development of this book. Thank you to all the participants of this research undertaking, especially to Mike Panitz, Rob Nash, Brian Hecox, Emmett Scout, Nathan Evers, Cora Walker, and Aina Braxton for working with us throughout all these years. The authors would also like to thank the students at the University of Washington Bothell for the games they built from the course CSS385: Introduction to Game Development (see <http://courses.washington.edu/css385>). Their interest and passion for games has provided us with the ideal deployment vehicle and are a source of continuous inspiration. They have tested, retested, contributed to, and assisted in the formation and organization of the contents of this book.

Jebediah Pavleas would like to thank the Computing and Software Systems Division at the University of Washington Bothell for the generous tuition scholarships that funded his education throughout his participation with this book project.

The hero character Dye and many of the visual and audio assets used throughout the example projects of the book are based on the Dye Hard game, designed for teaching concepts of objects and object-oriented hierarchy. The original Dye Hard development team members included Matthew Kipps, Rodelle Ladia, Chuan Wang, Brian Hecox, Charles Chiou, John Louie, Emmett Scout, Daniel Ly, Elliott White, Christina Jugovic, Rachel Harris, Nathan Evers, Kasey Quevedo, Kaylin Norman-Slack, David Madden, Kyle Kraus, Suzi Zuber, Aina Braxton, Kelvin Sung, Jason Pace, and Rob Nash. Kyle Kraus composed the background music used in the Audio Support project from Chapter 4, originally for the Linx game, which was designed to teach loops. The background audio for the game in Chapter 11 was composed by David Madden and arranged by Aina Braxton. Thanks to Clover Wai for the figures and illustrations.

We also want to thank Gwenan Spearing at Apress for connecting us to our editor Ben Renow-Clarke. A heartfelt thank-you to Kevin Walter for his patient and diligent organization skills in guiding us, to Melissa Maldonado for tolerating and working with our constantly behind schedule frenzy, and to Kim Wimpsett for the tireless and excellent edits that make much of this book actually readable. Finally, we would like to thank Jason Sturges for his insightful technical feedback.

All opinions, findings, conclusions, and recommendations in this work are those of the authors and do not necessarily reflect the views of the sponsors.



# Introduction

Welcome to *Build Your Own 2D Game Engine and Create Web Games*. Because you have picked up this book, you are likely interested in the details of a game engine and the creation of your own games to be played over the Internet. This book teaches you how to build a 2D game engine by covering the involved technical concepts, demonstrating sample implementations, and showing you how to organize the large number of source code and asset files to support game development. This book also discusses how each covered technical topic area relates to elements of game design so that you can build, play, analyze, and learn about the development of 2D game engines and games. The sample implementations in this book are based on HTML5, JavaScript, and WebGL, which are technologies that are freely available and supported by virtually all web browsers. After reading this book, the game engine you develop and the associated games will be playable through a web browser from anywhere on the Internet.

This book presents relevant concepts from software engineering, computer graphics, mathematics, physics, game development, and game design—all in the context of building a 2D game engine. The presentations are tightly integrated with the analysis and development of source code; you'll spend much of the book building game like concept projects that demonstrate the functionality of game engine components. By building on source code introduced early on, the book leads you on a journey through which you will master the basic concepts behind a 2D game engine while simultaneously gaining hands-on experience developing simple but working 2D games. Beginning from Chapter 4, a "Design Considerations" section is included at the end of each chapter to relate the covered technical concepts to elements of game design. By the end of the book, you will be familiar with the concepts and technical details of 2D game engines, feel competent in implementing functionality in a 2D game engine to support commonly encountered 2D game requirements, and capable of considering game engine technical topics in the context of game design elements in building fun and engaging games.

## Who Should Read This Book

This book is targeted toward programmers who are familiar with basic object-oriented programming concepts and have a basic to intermediate knowledge of an object-oriented programming language such as Java or C#. For example, if you are a student who has taken a few introductory programming courses, an experienced developer who is new to games and graphics programming, or a self-taught programming enthusiast, you will be able to follow the concepts and code presented in this book with little trouble. If you're new to programming in general, it is suggested that you first become comfortable with the JavaScript programming language and concepts in object-oriented programming before tackling the content provided in this book.

## Assumptions

You should be experienced with programming in an object-oriented programming language, such as Java or C#. Knowledge and expertise in JavaScript would be a plus but are not necessary. The examples in this book were created with the assumption that you understand data encapsulation and inheritance. In addition, you should be familiar with basic data structures such as linked lists and dictionaries and be comfortable working with the fundamentals of algebra and geometry, particularly linear equations and coordinate systems.

## Who Should Not Read This Book

This book is not designed to teach readers how to program, nor does it attempt to explain the intricate details of HTML5, JavaScript, or WebGL. If you have no prior experience developing software with an object-oriented programming language, you will probably find the examples in this book difficult to follow.

On the other hand, if you have an extensive background in game engine development based on other platforms, the content in this book will be too basic; this is a book intended for developers without 2D game engine development experience. However, you might still pick up a few useful tips about 2D game engine and 2D game development for the platforms covered in this book.

## Organization of This Book

This book teaches how to develop a game engine by describing the foundational infrastructure, graphics system, game object behaviors, camera manipulations, and a sample game creation based on the engine.

This book teaches how to develop a game engine by describing the foundational infrastructure, graphics system, game object behaviors, camera manipulations, and a sample game creation based on the engine.

Chapters 2 to 4 construct the foundational infrastructure of the game engine. Chapter 2 establishes the initial infrastructure by separating the source code system into folders and files that contain the following: JavaScript-specific core engine logics, WebGL GLSL-specific shader programs, and HTML5-specific web page contents. This organization allows ongoing engine functionality expansion while maintaining localized source code system changes. For example, only JavaScript source code files need to be modified when introducing enhancements to game object behaviors. Chapter 3 builds the drawing framework to encapsulate and hide the WebGL drawing specifics from the rest of the engine. This drawing framework allows the development of game object behaviors without being distracted by how they are drawn. Chapter 4 introduces and integrates core game engine functional components including game loop, keyboard input, efficient resource and game level loading, and audio support.

Chapters 5 to 7 present basic functionality of a game engine: drawing system, behavior and interactions, and camera manipulation. Chapter 5 focuses on working with texture mapping, including sprite sheets, animation with sprite sheets, and the drawing of bitmap fonts. Chapter 6 puts forward abstractions for game objects and their behaviors including per-pixel accurate collision detection. Chapter 7 details the manipulation and interactions with the camera including programming with multiple cameras and supporting mouse input.

Chapters 8 to 10 elevate the introduced functionality to more advanced levels. Chapter 8 covers the simulation of 3D illumination effects in 2D game scenes. Chapter 9 discusses physically based behavior simulations and particle systems that are suitable for modeling explosions. Chapter 10 examines more advanced camera functionality including infinite scrolling through tiling and parallax.

Chapter 11 summarizes the book by leading you through the design of a complete game based on the game engine you have developed.

## Code Samples

Every chapter in this book includes examples that let you interactively experiment with and learn the new materials. You can download the source code for all the projects, including the associated assets (images, audio clips, or fonts), from the following page: [www.apress.com/9781484209530](http://www.apress.com/9781484209530).

Follow the instructions to download the `9781484209530.zip` file. To install the code samples, unzip the `9781484209530.zip` file. You should see a folder structure that is organized by chapter numbers. Within each folder are subfolders containing NetBeans projects that correspond to sections of this book.

