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

Using HTML5, JavaScript, and WebGL2

Second Edition



Kelvin Sung

Jebediah Pavleas

Matthew Munson

Jason Pace

With

Original Dye Characters Design: **Nathan Evers**

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

Figures and Illustration: **Clover Wai**

C:\Users\Blitz\Documents\My Documents\Logos\Apress(r)_K.eps

Build Your Own 2D Game Engine and Create Great Web Games: Using HTML5, JavaScript, and WebGL2

Copyright © 2021 Kelvin Sung, Jebediah Pavleas, Matthew Munson, 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.

ISBN-13 (pbk): 978-1-4842-7376-0

ISBN-13 (electronic): 978-1-4842-7377-7

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, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Spandana Chatterjee

Development Editor: Laura Berendson

Coordinating Editors: Shrikant Vishwakarma

Cover designed by eStudioCalamar

Cover image by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Apress Media, LLC, 1 New York Plaza, New York, NY 10004, U.S.A. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.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 booktranslations@springernature.com; for reprint, paperback, or audio rights, please email bookpermissions@springernature.com.

Apress titles 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 Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484273760. For more detailed information, please visit http://www.apress.com/source-code.

This is the Dedication.

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

—Kelvin Sung

This page is customarily left blank.

Table of Contents

Contents at a Glance v

About the Authors xvi

Acknowledgments xvii

Introduction xix

PART I: packetC Background 1

CHAPTER 1: Dummy Text 3

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

PART II: Language Reference 53

CHAPTER 5: Variables: Dummy Text 55

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 15: Dummy Text 151

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 16: Dummy Text 159

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 17: Dummy Text 171

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 18: Dummy Text 175

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 19: Dummy Text 205

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

PART III: Developing Applications 215

CHAPTER 20: Dummy Text 217

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 21: Dummy Text 223

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 22: Dummy Text 233

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 23: Dummy Text 237

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 24: Dummy Text 245

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 25: Dummy Text 263

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 26: Dummy Text 281

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

CHAPTER 27: Dummy Text 287

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

PART IV: Industry Reprints 309

REPRINT 1: Dummy Text 311

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

REPRINT 2: A Dummy Text 319

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

REPRINT 3: Dummy Text 329

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

REPRINT 4: Dummy Text 335

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

REPRINT 5: Dummy Text 345

Dummy Text 9

Dummy Text 10

Virtual Dummy Text 11

Dummy Text 12

Dummy Text 12

Dummy Text Flow 13

Dummy Text 14

Summary 16

REPRINT 8: Dummy Text 371

REPRINT 9: Dummy Text  
Processing 377

APPENDIX A: Reference Tables 383

APPENDIX B: Dummy Text 395

INDEX 433

About the Authors

**Kelvin Sung** is a Professor with the Computing and Software Systems Division at University of Washington Bothell (UWB). He received his Ph.D. in Computer Science from the University of Illinois at Urbana‐Champaign. Kelvin's background is in computer graphics, hardware, and machine architecture. He came to UWB from Alias|Wavefront (now part of Autodesk), where he played a key role in designing and implementing the Maya Renderer, an Academy Award‐winning image generation system. At UWB, funded by Microsoft Research and the National Science Foundation, Kelvin’s work focuses on the intersection of video game mechanics, solutions to real‐world problems, and supports for remote collaboration. Together with his students and colleagues, Kelvin has co‐authored five books: one in computer graphics (Essentials of Interactive Computer Graphics: Concepts and Implementations, A.K. Peters, 2008), and the others in 2D game engines (Learn 2D Game Development with C# , APress, 2013; Build Your Own 2D Game Engine; Create Great Web Games, Apress, October 2015; Building a 2D Game Physics Engine, APress, 2016; and Basic Math for Game Development with Unity 3D, Apress 2019).

**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.

**Matthew Munson** is a graduate student in the Computer Science and Software Engineering program at the University of Washington Bothell. He received undergraduate degrees in Computer Science and Software Engineering and Mechanical Engineering at the University of Washington Bothell in 2020. Matthew is interested in operating system development, networking, and embedded systems. As a research assistant Matthew used cloud computing to analyze years of audio data recorded by hydrophones off the Oregon coast. This data was used to study the effects of climate change and shipping noise on marine mammals. Currently, Matthew is working on a networked augmented reality library that focuses on allowing users to view the same virtual scene from different perspectives.

**Jason Pace** has contributed to a wide range of games as a Producer, Designer, and Creative Director over 15 years in the interactive entertainment industry, from ultra-casual puzzlers to Halo. As a Designer Jason builds game mechanics and systems that start from a simple palette of core interactions (known as the core gameplay loop), progressively introducing variety and complexity to create interactive experiences that engage and delight players while maintaining focus on what makes each unique game fun.

About the Technical Reviewers

**Yusuf Pisan** is an Associate Teaching Professor in the School of Computing & Software Systems Division at the University of Washington Bothell. Previously he has worked at University of Technology, Sydney, and has been a visiting professor at Harvey Mudd College, University of Southern California, Worcester Polytechnic Institute (WPI), IT University of Copenhagen (ITU).

His research interests include enabling technologies for computer games, the design of virtual environments that support collaborative work, and computer science education. He founded the Australasian Conference on Interactive Entertainment conference series and helped foster the Australian games community. His list of publications can be found at Google Scholar.

Yusuf has a Ph.D. in Artificial Intelligence from Northwestern University. Before moving to Seattle in 2017, Yusuf lived in the Chicago area for 10 years and Sydney for 20 years.

For more information, see [https://pisanorg.github.io/yusuf/](https://urldefense.proofpoint.com/v2/url?u=https-3A__pisanorg.github.io_yusuf_&d=DwMFaQ&c=vh6FgFnduejNhPPD0fl_yRaSfZy8CWbWnIf4XJhSqx8&r=hBSr1R__kjUcST0tpDOpn6t2rvV4kaPHSZvoArp2MFA&m=Q7BLQIDR71VMK39agJ8JYId5LP3ScKn3urdkpitASBA&s=kY6Q0oIBeSgfyNgjspL43-UUJbRadY9wVYIBVN73Gf4&e=)

Acknowledgments

This book project was 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.

This second edition is encouraged by the many students and collaborators. In particular, students from CSS452: Game Engine Development (see https://myuwbclasses.github.io/CSS452/) at the University of Washington Bothell have been the most critical, demanding, and yet supportive. Through the many games and API extension projects (see https://html5gameenginegroup.github.io/GTCS-Engine-Student-Projects/), it became clear that updates are required of the JavaScript and WebGL versions, the bottom line synchronization mechanism, and, most significantly, the coverage of the physics engine. Fernando Arnez our co-author from first edition taught us JavaScript, Yaniv Schwartz was the first to point us towards JavaScript async/await and Promise. The discussions and collaborations with Huaming Chen and Michael Tanaya contributed directly to the chapter on game engine physics. Akilas Mebrahtom and Donald Hawkins helped constructed the extra example at the end of Chapter 9 illustrating potential presets for commonly encounter physical materials. The audio volume control was first investigated and integrated by Kyla NeSmith. Nicholas Carpenetti and Kyla NeSmith developed a user interface API for the initial game engine, which unfortunately did not make it into this edition. These and countless other feedbacks have contributed to the quality and improvements of the coverage.

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 12 was composed by David Madden and arranged by Aina Braxton. Thanks to Clover Wai for the figures and illustrations.

We also want to thank Spandana Chatterjee for believing in our ideas, her patience, and, continual efficient and effective support. A heartfelt thank-you to Mark Powers, for his diligence and lightning email responses. Mark should learn about and consider the option of sleeping some of the times. Nirmal Selvaraj organized everything and ensured proper progresses.

Finally, we would like to thank Yusuf Pisan for his insightful, effective, and above all, quick turnaround technical reviews.

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 WebGL2, 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.

# New in Second Edition

The key additions to the second edition include: JavaScript language and WebGL API update, and dedicated chapters with substantial details on physics and particle systems components.

All examples throughout the entire book are refined for the latest features in the JavaScript language. While some updates are mundane, e.g., prototype chain syntax replacements, the latest syntax allows significant improvements in overall presentation and code readability. The new and much cleaner asynchronous support facilitated a completely new resource loading architecture with a single synchronization point for the entire engine (Chapter 4). The WebGL context is updated to connect to WebGL 2.0. The dedicated chapters allow more elaborate and gradual introduction to the complex physics and particle systems components. Detailed mathematic derivation are included when appropriate.

# 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 WebGL2. 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.

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, WebGL2 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 WebGL2 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 11 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. Chapter 10 presents the basics of particle systems that are suitable for modeling explosions. Chapter 11 examines more advanced camera functionality including infinite scrolling through tiling and parallax.

Chapter 12 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 access the source code for all the projects, including the associated assets (images, audio clips, or fonts), by clicking the **Download Source Code** button located at <https://www.apress.com/9781484273760>. 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.