

hands-on game development with webassembly pdf

hands-on game development with webassembly pdf is a valuable resource for developers eager to harness the power of WebAssembly in creating high-performance, browser-based games. This technology revolutionizes how games are built and executed on the web, offering near-native speed and efficiency. The PDF format provides a structured, accessible guide that covers essential concepts, practical coding examples, and optimization techniques. It is designed to help programmers transition from traditional game development to leveraging WebAssembly's capabilities effectively. This article explores the key facets covered in the hands-on game development with webassembly pdf, including foundational knowledge, development workflows, integration strategies, and performance tuning. Readers will gain insight into the practical application of WebAssembly in game programming, supported by detailed explanations and best practices. The following sections outline the main topics discussed in the guide to facilitate a comprehensive understanding of this evolving technology.

- Understanding WebAssembly in Game Development
- Setting Up the Development Environment
- Core Concepts and Tools in Hands-On Game Development
- Building and Optimizing WebAssembly Games
- Debugging and Performance Tuning
- Advanced Techniques and Future Trends

Understanding WebAssembly in Game Development

WebAssembly (Wasm) is a low-level binary format designed to enable high-performance applications on web platforms. Its introduction has significantly impacted game development by allowing developers to run complex game engines and logic directly in the browser with near-native execution speed. The hands-on game development with webassembly pdf thoroughly explains the role of WebAssembly as a complement to JavaScript, highlighting its advantages in computationally intensive tasks common in gaming such as physics simulations, graphics rendering, and AI processing.

What is WebAssembly?

WebAssembly is a portable, size- and load-time-efficient format that browsers can execute quickly. It acts as a compilation target for languages like C, C++, and Rust, enabling developers to write performance-critical code that runs seamlessly in web environments. This capability makes WebAssembly particularly suited for game development, where performance and responsiveness are crucial.

Benefits of Using WebAssembly for Games

The hands-on game development with webassembly pdf highlights several benefits, including:

- **Improved Performance:** Faster execution of game logic compared to traditional JavaScript.
- **Cross-Platform Compatibility:** Games run consistently across all modern browsers and devices.
- **Access to Existing Codebases:** Ability to reuse existing C/C++ game engines and libraries.
- **Enhanced Security:** Sandboxed execution environment reducing vulnerabilities.

Setting Up the Development Environment

Before diving into coding, the hands-on game development with webassembly pdf provides detailed instructions on preparing the development environment. This setup is critical to ensure efficient workflow and smooth integration of WebAssembly modules into game projects.

Required Tools and Software

Developing games with WebAssembly requires a combination of compilers, build tools, and editors. Essential tools include:

- **Emscripten:** A compiler toolchain that converts C/C++ code into WebAssembly modules.
- **Rust and wasm-pack:** For developers using Rust, wasm-pack streamlines building and packaging WebAssembly modules.
- **Text Editors and IDEs:** Popular choices such as Visual Studio Code or

JetBrains CLion provide syntax highlighting and debugging support.

- **Web Browsers with Developer Tools:** Chrome, Firefox, and Edge for testing and debugging WebAssembly applications.

Configuring Build Systems

The PDF guide covers how to configure build systems like CMake or custom scripts to automate the compilation of source code into optimized WebAssembly binaries. Proper configuration ensures efficient builds and integration with JavaScript frontends.

Core Concepts and Tools in Hands-On Game Development

The hands-on game development with webassembly pdf emphasizes understanding foundational concepts and mastering essential tools that facilitate WebAssembly-based game creation. This knowledge is crucial for crafting games that are both performant and maintainable.

Memory Management and Data Structures

Efficient memory handling is vital for real-time games. The guide explains how WebAssembly manages memory using linear memory and how developers can allocate, free, and manipulate memory buffers. It also discusses common data structures optimized for use with WebAssembly.

Interfacing WebAssembly with JavaScript

Since WebAssembly modules rely on JavaScript for interaction with the DOM and browser APIs, understanding the communication between these two environments is essential. The PDF details how to call WebAssembly functions from JavaScript and vice versa, manage data passing, and handle asynchronous operations.

Game Engines and Frameworks

The resource explores popular engines and frameworks that support WebAssembly, such as Unity, Unreal Engine, and Godot. It illustrates their integration processes and highlights how developers can export WebAssembly builds to leverage existing game development ecosystems.

Building and Optimizing WebAssembly Games

Creating performant WebAssembly games requires a combination of sound coding practices and optimization strategies. The hands-on game development with webassembly pdf offers comprehensive guidance on these aspects to maximize game efficiency and user experience.

Compiling and Linking Strategies

The guide details compilation flags and linker options that influence the size and speed of the WebAssembly module. It explains how to balance between debug information and optimization levels to suit development or production builds.

Graphics and Audio Integration

Rendering graphics and managing audio are core to immersive gaming. The PDF explains how to use WebGL and Web Audio APIs alongside WebAssembly modules, ensuring smooth visual and auditory performance.

Optimization Techniques

Performance tuning tips include:

1. Minimizing JavaScript-Wasm boundary crossings to reduce overhead.
2. Using SIMD and multi-threading where supported to accelerate computations.
3. Applying code size optimizations for faster loading times.
4. Leveraging caching strategies for resource management.

Debugging and Performance Tuning

Effective debugging and profiling are critical to developing stable, high-quality games. The hands-on game development with webassembly pdf outlines methods and tools designed for WebAssembly game debugging and performance analysis.

Using Browser Developer Tools

Modern browsers provide tools to inspect WebAssembly modules, set breakpoints, and monitor memory usage. The guide offers step-by-step instructions on utilizing these features to identify and fix bugs.

Profiling and Benchmarking

Profiling tools help measure CPU usage, frame rates, and load times. The PDF explains how to benchmark different parts of the game code and interpret results to guide optimization efforts.

Advanced Techniques and Future Trends

The hands-on game development with webassembly pdf also explores advanced topics and emerging trends to prepare developers for the evolving landscape of web-based game development.

Multi-threading and Parallelism

As WebAssembly gains support for multi-threading via Web Workers and shared memory, developers can implement parallel algorithms to boost game performance. The guide discusses how to design thread-safe code and manage synchronization.

WebAssembly System Interface (WASI)

WASI extends WebAssembly capabilities beyond the browser, enabling games to run on various platforms with standard system calls. The PDF outlines potential uses of WASI in expanding game portability and functionality.

Future of Game Development with WebAssembly

Looking ahead, the integration of WebAssembly with emerging web technologies like WebGPU and improved tooling promises to further enhance game graphics and interactivity. The resource highlights ongoing developments and encourages developers to stay informed about innovations.

Frequently Asked Questions

What is the book 'Hands-On Game Development with WebAssembly' about?

The book 'Hands-On Game Development with WebAssembly' provides practical guidance on building games using WebAssembly technology, focusing on performance optimization and leveraging modern web capabilities for game development.

Who is the target audience for 'Hands-On Game Development with WebAssembly'?

The book is aimed at game developers, web developers, and programmers interested in learning how to create high-performance games on the web using WebAssembly and related tools.

Does 'Hands-On Game Development with WebAssembly' include example projects or source code?

Yes, the book typically includes example projects and source code to help readers understand and implement game development concepts using WebAssembly in a practical manner.

Where can I download the PDF version of 'Hands-On Game Development with WebAssembly'?

The PDF version of the book can usually be obtained through official publishers like Packt Publishing or authorized ebook retailers. It is recommended to access the book through legitimate channels to respect copyright.

What programming languages are covered in 'Hands-On Game Development with WebAssembly'?

The book often covers languages that compile to WebAssembly, such as C, C++, and Rust, showing how to integrate them into web-based game development workflows.

How does WebAssembly improve game development compared to traditional JavaScript?

WebAssembly provides near-native performance and faster execution times compared to JavaScript, enabling more complex and resource-intensive games to run smoothly in web browsers.

Additional Resources

1. *Hands-On Game Development with WebAssembly*

This book provides a practical introduction to building high-performance games using WebAssembly. It covers the fundamentals of WebAssembly, integration with JavaScript, and how to leverage it for real-time game graphics and physics. Readers will learn through hands-on projects, including creating a simple 2D game from scratch.

2. *WebAssembly for Game Developers: A Practical Guide*

Focused on game developers looking to enhance their web games, this guide explains how to optimize game performance using WebAssembly. It includes tutorials on compiling C++ and Rust code to WebAssembly and integrating with popular web game frameworks. The book also explores debugging and performance profiling techniques.

3. *Building High-Performance Web Games with WebAssembly*

This title dives into advanced techniques for creating fast and efficient browser games using WebAssembly. It covers memory management, interoperability with JavaScript, and graphics rendering using WebGL. Real-world examples demonstrate how to build complex game mechanics while maintaining smooth gameplay.

4. *Rust and WebAssembly Game Development*

Targeting developers interested in using Rust for web games, this book explores the synergy between Rust's performance and WebAssembly's portability. It walks through setting up a Rust development environment, compiling to WebAssembly, and creating interactive web games. The book also highlights best practices for game architecture and asset management.

5. *Interactive Web Game Programming with WebAssembly*

This book offers a hands-on approach to creating interactive web games powered by WebAssembly. It covers the basics of WebAssembly modules, integrating with HTML5 canvas, and handling user input efficiently. Readers will build multiple projects, including puzzle and platformer games, to solidify their understanding.

6. *Game Engine Development Using WebAssembly*

Focusing on the underlying engine architecture, this book guides readers through developing a custom game engine using WebAssembly. It discusses core engine components like rendering pipelines, physics simulation, and audio processing. The book is ideal for developers interested in the low-level details of game development.

7. *WebAssembly and JavaScript for Game Programmers*

This resource bridges the gap between JavaScript and WebAssembly for game development, showing how to combine both for optimal performance. It includes examples of game loops, asset loading, and real-time networking in multiplayer games. Readers will gain insight into leveraging WebAssembly to overcome JavaScript performance bottlenecks.

8. *Cross-Platform Game Development with WebAssembly*

This book explores strategies for creating games that run seamlessly across desktop and mobile browsers using WebAssembly. It highlights platform-specific considerations, performance tuning, and responsive design for diverse devices. Practical projects demonstrate how to maintain code portability and maximize user experience.

9. *Practical WebAssembly for Game Developers*

Designed as a practical guide, this book walks through the entire process of developing games with WebAssembly from concept to deployment. It covers language choices, toolchains, debugging, and browser compatibility issues. The book includes case studies and tips to help developers avoid common pitfalls in WebAssembly game development.

Hands On Game Development With Webassembly Pdf

Find other PDF articles:

<https://a.comtex-nj.com/wwu9/files?docid=WqU75-5316&title=instructional-fair-inc-chemistry-if8766.pdf>

Hands-On Game Development with WebAssembly PDF

Ebook Title: WebAssembly Game Development: A Hands-On Guide

Ebook Outline:

Introduction: What is WebAssembly? Why use it for game development? Setting up your development environment.

Chapter 1: WebAssembly Fundamentals: Understanding the WASM binary format. Working with JavaScript APIs. Memory management in WASM.

Chapter 2: Building a Simple Game: A step-by-step tutorial creating a basic 2D game using WASM and a JavaScript framework (e.g., Phaser). Including game loop implementation, rendering, and input handling.

Chapter 3: Advanced Game Mechanics: Implementing more complex game features like physics, collision detection, and animations. Exploring different libraries and tools.

Chapter 4: Optimizing for Performance: Techniques for improving game performance, including code optimization and asset management.

Chapter 5: Deployment and Distribution: Publishing your WebAssembly game online. Considerations for different platforms and browsers.

Chapter 6: Case Studies and Advanced Topics: Examples of successful WebAssembly games. Exploring more advanced topics like 3D rendering and multiplayer games.

Conclusion: Recap of key concepts and future trends in WebAssembly game development. Resources for continued learning.

Hands-On Game Development with WebAssembly: Unleashing the Power of the Web

The web is no longer just a place for static content and simple applications. With the advent of WebAssembly (Wasm), it's become a powerful platform for high-performance applications, including complex games. This comprehensive guide dives into the world of WebAssembly game development, providing a practical, hands-on approach to building your own games using this revolutionary technology. This article will explore the key aspects of developing games with WebAssembly, from setting up your environment to deploying your finished product.

What is WebAssembly and Why Use It for Games?

WebAssembly (Wasm) is a binary instruction format designed for execution in modern web browsers and other environments. It's not a programming language itself, but rather a compilation target for languages like C++, C#, Rust, and Go. This allows developers to write performance-critical code in languages they're comfortable with and compile it to a highly efficient format that runs near-natively in the browser.

Why is this significant for game development? Traditional web game development often relied heavily on JavaScript, which, while versatile, can sometimes struggle with the demanding performance requirements of complex games. WebAssembly changes the game (pun intended!) by offering performance comparable to native applications, opening up the possibility of creating sophisticated, visually stunning games that run smoothly in the browser.

Setting Up Your Development Environment: A Smooth Start

Before diving into code, you need a solid development environment. This involves:

Choosing a Text Editor or IDE: VS Code, Sublime Text, Atom, or WebStorm are popular choices offering excellent support for various programming languages used in WebAssembly development.

Installing a WebAssembly Compiler: The compiler you choose depends on your preferred source language (e.g., Emscripten for C++, various compilers for Rust and C#).

Selecting a JavaScript Framework (Optional but Recommended): Frameworks like Phaser, PixiJS, or Babylon.js simplify game development by providing pre-built functionalities for rendering, input handling, and physics. They handle the complexities of interacting with the browser's canvas, freeing you to focus on game logic.

Setting up a Local Web Server: This is crucial for testing your game in a browser environment, as directly opening `HTML` files can lead to issues with resource loading.

WebAssembly Fundamentals: Understanding the Engine

This section delves into the core concepts of WebAssembly:

The WASM Binary Format: Understanding how the compiled code is structured and interacts with the browser. While you won't be writing WASM directly (unless you're working with a low-level assembly language), understanding the underlying format helps in debugging and optimizing your code.

Interacting with JavaScript APIs: WASM doesn't operate in isolation. It communicates with the browser and JavaScript code through a well-defined interface. Learning how to effectively exchange data and call JavaScript functions from your WASM modules is crucial for game development.

Memory Management in WASM: WASM has its own memory model, different from JavaScript's garbage collection. Understanding memory allocation, deallocation, and potential pitfalls is vital to prevent memory leaks and crashes.

Building a Simple Game: A Practical Example

This section provides a step-by-step guide to building a simple 2D game using WebAssembly and a JavaScript framework. This will likely involve:

Project Setup: Creating a basic project structure with necessary files (HTML, JavaScript, WASM module).

Game Loop Implementation: Designing the main game loop that updates the game state and renders the game visually.

Rendering Graphics: Using the chosen framework (e.g., Phaser) to render sprites, backgrounds, and other visual elements.

Implementing Game Logic: Writing the core game logic in WASM, including player movement, collision detection (if applicable), and scoring.

Handling User Input: Integrating user input (keyboard, mouse) into the game loop to control the game.

Advanced Game Mechanics: Leveling Up

Once comfortable with the basics, this section introduces more advanced game development concepts:

Implementing Physics Engines: Integrating physics engines (like Box2D or Matter.js) to add realistic movement and interactions to game objects.

Collision Detection: Implementing robust collision detection to manage interactions between game objects.

Animations: Creating smooth and engaging animations to bring your game to life.

Using External Libraries: Exploring and integrating pre-built libraries to handle tasks like sound effects, particle effects, and advanced rendering techniques.

Optimizing for Performance: Speed and Efficiency

Performance is crucial for a positive user experience. This section covers optimization techniques:

Code Optimization: Techniques for writing more efficient WASM code.

Asset Management: Optimizing the size and loading time of game assets (images, sounds).

Profiling and Debugging: Using browser developer tools and profiling tools to identify performance bottlenecks.

Deployment and Distribution: Sharing Your Creation

Getting your game to players is the final, crucial step:

Hosting Options: Exploring different hosting options (e.g., GitHub Pages, Netlify, custom servers).

Building for Different Browsers: Ensuring compatibility across different browsers and devices.

Using a Build System: Streamlining the build process using tools like Webpack or Parcel.

Case Studies and Advanced Topics: Exploring the Frontier

This section showcases successful WebAssembly games and explores future directions:

Advanced Rendering Techniques: Investigating advanced 3D rendering techniques.

Multiplayer Game Development: Exploring the challenges and techniques involved in creating multiplayer WebAssembly games.

Emerging Technologies: Discussing future trends and technologies that will impact WebAssembly game development.

Conclusion: The Future of Web Gaming

WebAssembly is revolutionizing web game development, enabling the creation of high-performance games that were previously impossible. This guide provides a solid foundation for building your own games, but the possibilities are endless. Continue exploring, experimenting, and pushing the

boundaries of what's possible with WebAssembly!

FAQs

1. What programming languages can I use with WebAssembly? C++, C#, Rust, Go, and others are commonly used.
2. Is WebAssembly suitable for all types of games? Yes, from simple 2D games to complex 3D titles, WebAssembly's performance is impressive.
3. How does WebAssembly compare to JavaScript for game development? WebAssembly offers significantly better performance for computationally intensive tasks.
4. What are the common challenges in WebAssembly game development? Debugging and memory management can be more complex than in JavaScript development.
5. What frameworks are recommended for WebAssembly game development? Phaser, PixiJS, and Babylon.js are popular choices.
6. Can I use existing game engines with WebAssembly? Some game engines offer WebAssembly support, or you can adapt existing code.
7. How can I optimize the loading time of my WebAssembly game? Code splitting, asset optimization, and efficient loading strategies are crucial.
8. Where can I find more resources for WebAssembly game development? The WebAssembly community website and various online tutorials are excellent resources.
9. What's the future of WebAssembly in gaming? WebAssembly is expected to become increasingly important for high-performance web gaming.

Related Articles:

1. Optimizing WebAssembly for Game Performance: Techniques for improving the speed and efficiency of your WebAssembly games.
2. Building Multiplayer Games with WebAssembly: Strategies for creating online multiplayer games using WebAssembly.
3. Integrating Physics Engines with WebAssembly: How to use physics engines like Box2D or Matter.js in your WebAssembly games.
4. Advanced Animation Techniques in WebAssembly Games: Creating high-quality animations for your WebAssembly game.
5. Deploying WebAssembly Games to Different Platforms: Strategies for publishing your

WebAssembly game on various platforms.

6. Using Web Workers with WebAssembly for Parallel Processing: Utilizing Web Workers to enhance game performance through parallel processing.

7. Debugging and Profiling WebAssembly Code: Tools and techniques for identifying and fixing issues in your WebAssembly code.

8. A Comparison of WebAssembly Game Development Frameworks: An analysis of various frameworks suitable for WebAssembly game development.

9. The Future of WebAssembly and its Impact on the Gaming Industry: An exploration of the potential of WebAssembly in shaping the future of web gaming.

hands on game development with webassembly pdf: Hands-On Game Development with WebAssembly Rick Battagline, 2019-05-31 Make your WebAssembly journey fun while making a game with it Key FeaturesCreate a WebAssembly game that implements sprites, animations, physics, particle systems, and other game development fundamentalsGet to grips with advanced game mechanics in WebAssemblyLearn to use WebAssembly and WebGL to render to the HTML5 canvas elementBook Description Within the next few years, WebAssembly will change the web as we know it. It promises a world where you can write an application for the web in any language, and compile it for native platforms as well as the web. This book is designed to introduce web developers and game developers to the world of WebAssembly by walking through the development of a retro arcade game. You will learn how to build a WebAssembly application using C++, Emscripten, JavaScript, WebGL, SDL, and HTML5. This book covers a lot of ground in both game development and web application development. When creating a game or application that targets WebAssembly, developers need to learn a plethora of skills and tools. This book is a sample platter of those tools and skills. It covers topics including Emscripten, C/C++, WebGL, OpenGL, JavaScript, HTML5, and CSS. The reader will also learn basic techniques for game development, including 2D sprite animation, particle systems, 2D camera design, sound effects, 2D game physics, user interface design, shaders, debugging, and optimization. By the end of the book, you will be able to create simple web games and web applications targeting WebAssembly. What you will learnBuild web applications with near-native performance using WebAssemblyBecome familiar with how web applications can be used to create games using HTML5 Canvas, WebGL, and SDLBecome well versed with game development concepts such as sprites, animation, particle systems, AI, physics, camera design, sound effects, and shadersDeploy C/C++ applications to the browser using WebAssembly and EmscriptenUnderstand how Emscripten HTML shell templates, JavaScript glue code, and a WebAssembly module interactDebug and performance tune your WebAssembly applicationWho this book is for Web developers and game developers interested in creating applications for the web using WebAssembly. Game developers interested in deploying their games to the web Web developers interested in creating applications that are potentially orders of magnitude faster than their existing JavaScript web apps C/C++ developers interested in using their existing skills to deploy applications to the web

hands on game development with webassembly pdf: Hands-on Rust Herbert Wolverson, 2021-06-30 Rust is an exciting new programming language combining the power of C with memory safety, fearless concurrency, and productivity boosters - and what better way to learn than by making games. Each chapter in this book presents hands-on, practical projects ranging from Hello, World to building a full dungeon crawler game. With this book, you'll learn game development skills applicable to other engines, including Unity and Unreal. Rust is an exciting programming language combining the power of C with memory safety, fearless concurrency, and productivity boosters. With Rust, you have a shiny new playground where your game ideas can flourish. Each chapter in this book presents hands-on, practical projects that take you on a journey from Hello, World to building a full dungeon crawler game. Start by setting up Rust and getting comfortable with your development environment. Learn the language basics with practical examples as you make your own version of

Flappy Bird. Discover what it takes to randomly generate dungeons and populate them with monsters as you build a complete dungeon crawl game. Run game systems concurrently for high-performance and fast game-play, while retaining the ability to debug your program. Unleash your creativity with magical items, tougher monsters, and intricate dungeon design. Add layered graphics and polish your game with style. What You Need: A computer running Windows 10, Linux, or Mac OS X. A text editor, such as Visual Studio Code. A video card and drivers capable of running OpenGL 3.2.

hands on game development with webassembly pdf: Programming WebAssembly with Rust Kevin Hoffman, 2019-05-31 WebAssembly fulfills the long-awaited promise of web technologies: fast code, type-safe at compile time, execution in the browser, on embedded devices, or anywhere else. Rust delivers the power of C in a language that strictly enforces type safety. Combine both languages and you can write for the web like never before! Learn how to integrate with JavaScript, run code on platforms other than the browser, and take a step into IoT. Discover the easy way to build cross-platform applications without sacrificing power, and change the way you write code for the web. WebAssembly is more than just a revolutionary new technology. It's reshaping how we build applications for the web and beyond. Where technologies like ActiveX and Flash have failed, you can now write code in whatever language you prefer and compile to WebAssembly for fast, type-safe code that runs in the browser, on mobile devices, embedded devices, and more. Combining WebAssembly's portable, high-performance modules with Rust's safety and power is a perfect development combination. Learn how WebAssembly's stack machine architecture works, install low-level wasm tools, and discover the dark art of writing raw wast code. Build on that foundation and learn how to compile WebAssembly modules from Rust by implementing the logic for a checkers game. Create wasm modules in Rust to interoperate with JavaScript in many compelling ways. Apply your new skills to the world of non-web hosts, and create everything from an app running on a Raspberry Pi that controls a lighting system, to a fully-functioning online multiplayer game engine where developers upload their own arena-bound WebAssembly combat modules. Get started with WebAssembly today, and change the way you think about the web. What You Need: You'll need a Linux, Mac, or Windows workstation with an Internet connection. You'll need an up-to-date web browser that supports WebAssembly. To work with the sample code, you can use your favorite text editor or IDE. The book will guide you through installing the Rust and WebAssembly tools needed for each chapter.

hands on game development with webassembly pdf: The Art of WebAssembly Rick Battagline, 2021-06-01 A thorough, practice-based introduction to WebAssembly. Learn how to create high-performing, lightning-fast websites and applications. WebAssembly is the fast, compact, portable technology that optimizes the performance of resource-intensive web applications and programs. The Art of WebAssembly is designed to give web developers a solid understanding of how it works, when to use it (and when not to), and how to develop and deploy WebAssembly apps. First you'll learn how to optimize and compile low-level code, debug and evaluate WebAssembly, and represent WebAssembly in the human-readable WebAssembly Text (WAT) format. Once you have the basics down, you'll build a browser-based collision detection program, work with browser rendering technologies to create graphics and animations, and see how WebAssembly interacts with other web languages. You'll also learn how to: Embed WebAssembly applications in web browsers and Node.js Use browser debuggers to evaluate your WebAssembly code Format variables, loops, functions, strings, data structures, and conditional logic in WAT Manipulate memory Build a program that generates graphical objects and detects when they collide Evaluate the output of a WebAssembly compiler The Art of WebAssembly will help you make sense of this powerful technology to boost the performance of your web applications.

hands on game development with webassembly pdf: WebAssembly in Action Gerard Gallant, 2019-11-06 Summary WebAssembly in Action introduces the WebAssembly stack and walks you through the process of writing and running browser-based applications. Expert developer Gerard Gallant gives you a firm foundation of the structure of a module, HTML basics, JavaScript

Promises, and the WebAssembly JavaScript API. About the technology Write high-performance browser-based applications without relying only on JavaScript! By compiling to the WebAssembly binary format, your C, C++, or Rust code runs at near-native speed in the browser. WebAssembly delivers greater speed, opportunities to reuse existing code, and access to newer and faster libraries. Plus, you can easily interact with JavaScript when you need to. About the book WebAssembly in Action teaches you how to write and run high-performance browser-based applications using C++ and other languages supported by WebAssembly. In it, you'll learn to create native WebAssembly modules, interact with JavaScript components, and maximize performance with web workers and pthreads. And you'll love how the clearly organized sections make it a breeze to find the important details about every function, feature, and technique. What's inside Dynamic linking of multiple modules at runtime Communicating between modules and JavaScript Debugging with WebAssembly Text Format Threading with web workers and pthreads About the reader Written for developers with a basic understanding of C/C++, JavaScript, and HTML. About the author Gerard Gallant is a Microsoft Certified Professional and a Senior Software Developer at Dovico Software. He blogs regularly on Blogger.com and DZone.com.

hands on game development with webassembly pdf: Learn WebAssembly Mike Rourke, 2018-09-25 In the first definitive guide on WebAssembly, you'll learn how you can wield this new technology to break through the current barriers of web development and build an entirely new class of performant applications . Key FeaturesGenerate WebAssembly modules from C and C++ using Emscripten and interact with these modules in the browser Learn how to use WebAssembly outside of the browser and load modules using Node.js Build a high-performance application using C and WebAssembly and port an existing C++ game to WebAssembly using Emscripten Book Description WebAssembly is a brand-new technology that represents a paradigm shift in web development. This book teaches programmers to leverage this technology to write high-performance applications that run in the browser. This book introduces you to powerful WebAssembly concepts to help you write lean and powerful web applications with native performance. You start with the evolution of web programming, the state of things today, and what can be done with the advent and release of WebAssembly. We take a look at the journey from JavaScript to asm.js to WebAssembly. We then move on to analyze the anatomy of a WebAssembly module and the relationship between binary and text formats, along with the corresponding JavaScript API. Further on, you'll implement all the techniques you've learned to build a high-performance application using C and WebAssembly, and then port an existing game written in C++ to WebAssembly using Emscripten. By the end of this book, you will be well-equipped to create high-performance applications and games for the web using WebAssembly. What you will learnLearn how WebAssembly came to be and its associated elements (text format, module, and JavaScript API)Create, load, and debug a WebAssembly module (editor and compiler/toolchain)Build a high-performance application using C and WebAssemblyExtend WebAssembly's feature set using Emscripten by porting a game written in C++Explore upcoming features of WebAssembly, Node.js integration, and alternative compilation methodsWho this book is for If you are a web developer or C/C++ programmer keen to leverage the powerful technology of WebAssembly to build high-performance web applications, then this book is for you.

hands on game development with webassembly pdf: Creative Projects for Rust Programmers Carlo Milanese, 2020-06-19 A practical guide to understanding the latest features of the Rust programming language, useful libraries, and frameworks that will help you design and develop interesting projects Key FeaturesWork through projects that will help you build high-performance applications with RustDelve into concepts such as error handling, memory management, concurrency, generics, and macros with RustImprove business productivity by choosing the right libraries and frameworks for your applicationsBook Description Rust is a community-built language that solves pain points present in many other languages, thus improving performance and safety. In this book, you will explore the latest features of Rust by building robust applications across different domains and platforms. The book gets you up and running with

high-quality open source libraries and frameworks available in the Rust ecosystem that can help you to develop efficient applications with Rust. You'll learn how to build projects in domains such as data access, RESTful web services, web applications, 2D games for web and desktop, interpreters and compilers, emulators, and Linux Kernel modules. For each of these application types, you'll use frameworks such as Actix, Tera, Yew, Quicksilver, ggez, and nom. This book will not only help you to build on your knowledge of Rust but also help you to choose an appropriate framework for building your project. By the end of this Rust book, you will have learned how to build fast and safe applications with Rust and have the real-world experience you need to advance in your career. What you will learn Access TOML, JSON, and XML files and SQLite, PostgreSQL, and Redis databases Develop a RESTful web service using JSON payloads Create a web application using HTML templates and JavaScript and a frontend web application or web game using WebAssembly Build desktop 2D games Develop an interpreter and a compiler for a programming language Create a machine language emulator Extend the Linux Kernel with loadable modules Who this book is for This Rust programming book is for developers who want to get hands-on experience with implementing their knowledge of Rust programming, and are looking for expert advice on which libraries and frameworks they can adopt to develop software that typically uses the Rust language.

hands on game development with webassembly pdf: C++ Game Development By Example Siddharth Shekar, 2019-05-03 Explore modern game programming and rendering techniques to build games using C++ programming language and its popular libraries Key Features Learn how you can build basic 2D and complex 3D games with C++ Understand shadows, texturing, lighting, and rendering in 3D game development using OpenGL Uncover modern graphics programming techniques and GPU compute methods using the Vulkan API Book Description Although numerous languages are currently being used to develop games, C++ remains the standard for fabricating expert libraries and tool chains for game development. This book introduces you to the world of game development with C++. C++ Game Development By Example starts by touching upon the basic concepts of math, programming, and computer graphics and creating a simple side-scrolling action 2D game. You'll build a solid foundation by studying basic game concepts such as creating game loops, rendering 2D game scenes using SFML, 2D sprite creation and animation, and collision detection. The book will help you advance to creating a 3D physics puzzle game using modern OpenGL and the Bullet physics engine. You'll understand the graphics pipeline, which entails creating 3D objects using vertex and index buffers and rendering them to the scene using vertex and fragment shaders. Finally, you'll create a basic project using the Vulkan library that'll help you get to grips with creating swap chains, image views, render passes, and frame buffers for building high-performance graphics in your games. By the end of this book, you'll be ready with 3 compelling projects created with SFML, the Vulkan API, and OpenGL, and you'll be able take your game and graphics programming skills to the next level. What you will learn Understand shaders and how to write a basic vertex and fragment shader Build a Visual Studio project and add SFML to it Discover how to create sprite animations and a game character class Add sound effects and background music to your game Grasp how to integrate Vulkan into Visual Studio Create shaders and convert them to the SPIR-V binary format Who this book is for If you're a developer keen to learn game development with C++ or get up to date with game development, this book is for you. Some knowledge of C++ programming is assumed.

hands on game development with webassembly pdf: Rust Brain Teasers Herbert Wolverson, 2022-02-25 The Rust programming language is consistent and does its best to avoid surprising the programmer. Like all languages, though, Rust still has its quirks. But these quirks present a teaching opportunity. In this book, you'll work through a series of brain teasers that will challenge your understanding of Rust. By understanding the gaps in your knowledge, you can become better at what you do and avoid mistakes. Many of the teasers in this book come from the author's own experience creating software. Others derive from commonly asked questions in the Rust community. Regardless of their origin, these brain teasers are fun, and let's face it: who doesn't love a good puzzle, right? What better way to exercise your brain and increase your Rust

programming knowledge than with a collection of dynamic brain teasers? As you read through each of these puzzles and try to work out the answers, you'll not only learn about Rust's unique quirks and peculiarities, you'll also have loads of fun along the way. Dive right in and get started with example code and sample problems that cover numbers and text, shadowing and memory, and everything in between. Try to figure out why a particular program won't compile, why it produces unexpected output, or why it panics and terminates with an error message. Once you've run the code and read the answer, it's time to get to the heart of the matter with a detailed explanation. Learn why a program produced the result it did, and discover how similar issues might affect the code you write in your own programs, even in production. Sourced from engaging discussions within the Rust community, real-world problems, and even reader feedback, these challenges will certainly surprise, enlighten, and entertain you. Are you ready to experience Rust like never before? Then sharpen your brain and get ready for a challenge! What You Need: This book assumes you have some knowledge of the Rust programming language. To work through the brain teasers in this book, you'll need a working Rust environment on any platform. You can install Rust by visiting <https://rustup.rs/>. You'll also need a text editor or Rust-friendly IDE.

hands on game development with webassembly pdf: The Art of Assembly Language, 2nd Edition Randall Hyde, 2010-03-01 Assembly is a low-level programming language that's one step above a computer's native machine language. Although assembly language is commonly used for writing device drivers, emulators, and video games, many programmers find its somewhat unfriendly syntax intimidating to learn and use. Since 1996, Randall Hyde's The Art of Assembly Language has provided a comprehensive, plain-English, and patient introduction to 32-bit x86 assembly for non-assembly programmers. Hyde's primary teaching tool, High Level Assembler (or HLA), incorporates many of the features found in high-level languages (like C, C++, and Java) to help you quickly grasp basic assembly concepts. HLA lets you write true low-level code while enjoying the benefits of high-level language programming. As you read The Art of Assembly Language, you'll learn the low-level theory fundamental to computer science and turn that understanding into real, functional code. You'll learn how to: -Edit, compile, and run HLA programs -Declare and use constants, scalar variables, pointers, arrays, structures, unions, and namespaces -Translate arithmetic expressions (integer and floating point) -Convert high-level control structures This much anticipated second edition of The Art of Assembly Language has been updated to reflect recent changes to HLA and to support Linux, Mac OS X, and FreeBSD. Whether you're new to programming or you have experience with high-level languages, The Art of Assembly Language, 2nd Edition is your essential guide to learning this complex, low-level language.

hands on game development with webassembly pdf: How To Code in Go Mark Bates, Cory LaNou, Tim Raymond, 2020-06-11

hands on game development with webassembly pdf: Web Development with Blazor Jimmy Engström Engström, 2021-06-18 Develop modern web UIs quickly with server-side Blazor and Blazor WebAssembly Key Features Create and deploy a production-ready Blazor application from start to finish Learn Blazor fundamentals, gain actionable insights, and discover best practices Find out how, when, and why to use server-side Blazor and Blazor WebAssembly Book DescriptionBlazor is an essential tool if you want to build interactive web apps without JS, but it comes with its own learning curve. Web Development with Blazor will help you overcome most common challenges developers face when getting started with Blazor and teach you the best coding practices. You'll start by learning how to leverage the power of Blazor and explore the full capabilities of both Blazor Server and Blazor WebAssembly. Then you'll move on to the practical part, which is centred around a sample project - a blog engine. This is where you'll apply all your newfound knowledge about creating Blazor Server and Blazor WebAssembly projects, the inner working of Razor syntax, and validating forms, as well as creating your own components. You'll learn all the key concepts involved in web development with Blazor, which you'll also be able to put into practice straight away. By showing you how all the components work together practically, this book will help you avoid some of the common roadblocks that novice Blazor developers face and inspire you to start experimenting

with Blazor on your other projects. When you reach the end of this Blazor book, you'll have gained the confidence you need to create and deploy production-ready Blazor applications. What you will learn Understand the different technologies that can be used with Blazor, such as Blazor Server and Blazor WebAssembly Find out how to build simple and advanced Blazor components Explore the differences between Blazor Server and Blazor WebAssembly projects Discover how Entity Framework works and build a simple API Get up to speed with components and find out how to create basic and advanced components Explore existing JavaScript libraries in Blazor Use techniques to debug your Blazor Server and Blazor WebAssembly applications Test Blazor components using bUnit Who this book is for If you're a .NET web or software developer who wants to build web UIs using C#, then this book is for you. You'll need intermediate-level web-development skills and basic knowledge of C# before you get started; the book will guide you through the rest.

hands on game development with webassembly pdf: Python Basics Dan Bader, Joanna Jablonski, Fletcher Heisler, 2021-03-16 Make the Leap From Beginner to Intermediate in Python... Python Basics: A Practical Introduction to Python 3 Your Complete Python Curriculum-With Exercises, Interactive Quizzes, and Sample Projects What should you learn about Python in the beginning to get a strong foundation? With Python Basics, you'll not only cover the core concepts you really need to know, but you'll also learn them in the most efficient order with the help of practical exercises and interactive quizzes. You'll know enough to be dangerous with Python, fast! Who Should Read This Book If you're new to Python, you'll get a practical, step-by-step roadmap on developing your foundational skills. You'll be introduced to each concept and language feature in a logical order. Every step in this curriculum is explained and illustrated with short, clear code samples. Our goal with this book is to educate, not to impress or intimidate. If you're familiar with some basic programming concepts, you'll get a clear and well-tested introduction to Python. This is a practical introduction to Python that jumps right into the meat and potatoes without sacrificing substance. If you have prior experience with languages like VBA, PowerShell, R, Perl, C, C++, C#, Java, or Swift the numerous exercises within each chapter will fast-track your progress. If you're a seasoned developer, you'll get a Python 3 crash course that brings you up to speed with modern Python programming. Mix and match the chapters that interest you the most and use the interactive quizzes and review exercises to check your learning progress as you go along. If you're a self-starter completely new to coding, you'll get practical and motivating examples. You'll begin by installing Python and setting up a coding environment on your computer from scratch, and then continue from there. We'll get you coding right away so that you become competent and knowledgeable enough to solve real-world problems, fast. Develop a passion for programming by solving interesting problems with Python every day! If you're looking to break into a coding or data-science career, you'll pick up the practical foundations with this book. We won't just dump a boat load of theoretical information on you so you can sink or swim-instead you'll learn from hands-on, practical examples one step at a time. Each concept is broken down for you so you'll always know what you can do with it in practical terms. If you're interested in teaching others how to Python, this will be your guidebook. If you're looking to stoke the coding flame in your coworkers, kids, or relatives-use our material to teach them. All the sequencing has been done for you so you'll always know what to cover next and how to explain it. What Python Developers Say About The Book: Go forth and learn this amazing language using this great book. - Michael Kennedy, Talk Python The wording is casual, easy to understand, and makes the information flow well. - Thomas Wong, Pythonista I floundered for a long time trying to teach myself. I slogged through dozens of incomplete online tutorials. I snoozed through hours of boring screencasts. I gave up on countless cruffy books from big-time publishers. And then I found Real Python. The easy-to-follow, step-by-step instructions break the big concepts down into bite-sized chunks written in plain English. The authors never forget their audience and are consistently thorough and detailed in their explanations. I'm up and running now, but I constantly refer to the material for guidance. - Jared Nielsen, Pythonista

hands on game development with webassembly pdf: Head First C# Andrew Stellman, Jennifer Greene, 2020-12-04 What will you learn from this book? Dive into C# and create apps, user

interfaces, games, and more using this fun and highly visual introduction to C#, .NET Core, and Visual Studio. With this completely updated guide, which covers C# 8.0 and Visual Studio 2019, beginning programmers like you will build a fully functional game in the opening chapter. Then you'll learn how to use classes and object-oriented programming, create 3D games in Unity, and query data with LINQ. And you'll do it all by solving puzzles, doing hands-on exercises, and building real-world applications. By the time you're done, you'll be a solid C# programmer--and you'll have a great time along the way! What's so special about this book? Based on the latest research in cognitive science and learning theory, Head First C# uses a visually rich format to engage your mind rather than a text-heavy approach that puts you to sleep. Why waste your time struggling with new concepts? This multisensory learning experience is designed for the way your brain really works.

hands on game development with webassembly pdf: Creative DIY Microcontroller Projects with TinyGo and WebAssembly Tobias Theel, 2021-05-14 Explore embedded programming, and get hands-on with real-world embedded projects relating to IoT, low-powered devices, and other complex systems using TinyGo and WebAssembly Key FeaturesBuild creative embedded apps with TinyGo using low-powered devices and microcontrollersUnderstand the practicality involved in integrating hardware and sensors while programming them using TinyGoUse TinyGo in modern browsers to display embedded applications' statistics on WebAssembly dashboardsBook Description While often considered a fast and compact programming language, Go usually creates large executables that are difficult to run on low-memory or low-powered devices such as microcontrollers or IoT. TinyGo is a new compiler that allows developers to compile their programs for such low-powered devices. As TinyGo supports all the standard features of the Go programming language, you won't have to tweak the code to fit on the microcontroller. This book is a hands-on guide packed full of interesting DIY projects that will show you how to build embedded applications. You will learn how to program sensors and work with microcontrollers such as Arduino UNO and Arduino Nano IoT 33. The chapters that follow will show you how to develop multiple real-world embedded projects using a variety of popular devices such as LEDs, 7-segment displays, and timers. Next, you will progress to build interactive prototypes such as a traffic lights system, touchless hand wash timer, and more. As you advance, you'll create an IoT prototype of a weather alert system and display those alerts on the TinyGo WASM dashboard. Finally, you will build a home automation project that displays stats on the TinyGo WASM dashboard. By the end of this microcontroller book, you will be equipped with the skills you need to build real-world embedded projects using the power of TinyGo. What you will learnDiscover a variety of TinyGo features and capabilities while programming your embedded devicesExplore how to use display devices to present your dataFocus on how to make TinyGo interact with multiple sensors for sensing temperature, humidity, and pressureProgram hardware devices such as Arduino Uno and Arduino Nano IoT 33 using TinyGoUnderstand how TinyGo works with GPIO, ADC, I2C, SPI, and MQTT network protocolsBuild your first TinyGo IoT and home automation prototypesIntegrate TinyGo in modern browsers using WebAssemblyWho this book is for If you are a Go developer who wants to program low-powered devices and hardware such as Arduino UNO and Arduino Nano IoT 33, or if you are a Go developer who wants to extend your knowledge of using Go with WebAssembly while programming Go in the browser, then this book is for you. Go hobbyist programmers who are interested in learning more about TinyGo by working through the DIY projects covered in the book will also find this hands-on guide useful.

hands on game development with webassembly pdf: Rust in Action Tim McNamara, 2021-09-07 This well-written book will help you make the most of what Rust has to offer. - Ramnivas Laddad, author of AspectJ in Action Rust in Action is a hands-on guide to systems programming with Rust. Written for inquisitive programmers, it presents real-world use cases that go far beyond syntax and structure. Summary Rust in Action introduces the Rust programming language by exploring numerous systems programming concepts and techniques. You'll be learning Rust by delving into how computers work under the hood. You'll find yourself playing with persistent storage, memory, networking and even tinkering with CPU instructions. The book takes you through using Rust to

extend other applications and teaches you tricks to write blindingly fast code. You'll also discover parallel and concurrent programming. Filled to the brim with real-life use cases and scenarios, you'll go beyond the Rust syntax and see what Rust has to offer in real-world use cases. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology Rust is the perfect language for systems programming. It delivers the low-level power of C along with rock-solid safety features that let you code fearlessly. Ideal for applications requiring concurrency, Rust programs are compact, readable, and blazingly fast. Best of all, Rust's famously smart compiler helps you avoid even subtle coding errors. About the book Rust in Action is a hands-on guide to systems programming with Rust. Written for inquisitive programmers, it presents real-world use cases that go far beyond syntax and structure. You'll explore Rust implementations for file manipulation, networking, and kernel-level programming and discover awesome techniques for parallelism and concurrency. Along the way, you'll master Rust's unique borrow checker model for memory management without a garbage collector. What's inside Elementary to advanced Rust programming Practical examples from systems programming Command-line, graphical and networked applications About the reader For intermediate programmers. No previous experience with Rust required. About the author Tim McNamara uses Rust to build data processing pipelines and generative art. He is an expert in natural language processing and data engineering. Table of Contents 1 Introducing Rust PART 1 RUST LANGUAGE DISTINCTIVES 2 Language foundations 3 Compound data types 4 Lifetimes, ownership, and borrowing PART 2 DEMYSTIFYING SYSTEMS PROGRAMMING 5 Data in depth 6 Memory 7 Files and storage 8 Networking 9 Time and timekeeping 10 Processes, threads, and containers 11 Kernel 12 Signals, interrupts, and exceptions

hands on game development with webassembly pdf: Modern Web Development Dino Esposito, 2016-02-22 Master powerful new approaches to web architecture, design, and user experience This book presents a pragmatic, problem-driven, user-focused approach to planning, designing, and building dynamic web solutions. You'll learn how to gain maximum value from Domain-Driven Design (DDD), define optimal supporting architecture, and succeed with modern UX-first design approaches. The author guides you through choosing and implementing specific technologies and addresses key user-experience topics, including mobile-friendly and responsive design. You'll learn how to gain more value from existing Microsoft technologies such as ASP.NET MVC and SignalR by using them alongside other technologies such as Bootstrap, AJAX, JSON, and JQuery. By using these techniques and understanding the new ASP.NET Core 1.0, you can quickly build advanced web solutions that solve today's problems and deliver an outstanding user experience. Microsoft MVP Dino Esposito shows you how to: Plan websites and web apps to mirror real-world social and business processes Use DDD to dissect and master the complexity of business domains Use UX-Driven Design to reduce costs and give customers what they want Realistically compare server-side and client-side web paradigms Get started with the new ASP.NET Core 1.0 Simplify modern visual webpage construction with Bootstrap Master practical, efficient techniques for running ASP.NET MVC projects Consider new options for implementing persistence and working with data models Understand Responsive Web Design's pros, cons, and tradeoffs Build truly mobile-friendly, mobile-optimized websites About This Book For experienced developers and solution architects who want to plan and develop web solutions more effectively Assumes basic familiarity with the Microsoft web development stack

hands on game development with webassembly pdf: Speed Up Your Python with Rust Maxwell Flitton, 2022-01-21 Discover how to inject your code with highly performant Rust features to develop fast and memory-safe applications Key Features Learn to implement Rust in a Python system without altering the entire system Write safe and efficient Rust code as a Python developer by understanding the essential features of Rust Build Python extensions in Rust by using Python NumPy modules in your Rust code Book Description Python has made software development easier, but it falls short in several areas including memory management that lead to poor performance and security. Rust, on the other hand, provides memory safety without using a garbage collector, which

means that with its low memory footprint, you can build high-performant and secure apps relatively easily. However, rewriting everything in Rust can be expensive and risky as there might not be package support in Rust for the problem being solved. This is where Python bindings and pip come in. This book will help you, as a Python developer, to start using Rust in your Python projects without having to manage a separate Rust server or application. Seeing as you'll already understand concepts like functions and loops, this book covers the quirks of Rust such as memory management to code Rust in a productive and structured manner. You'll explore the PyO3 crate to fuse Rust code with Python, learn how to package your fused Rust code in a pip package, and then deploy a Python Flask application in Docker that uses a private Rust pip module. Finally, you'll get to grips with advanced Rust binding topics such as inspecting Python objects and modules in Rust. By the end of this Rust book, you'll be able to develop safe and high-performant applications with better concurrency support. What you will learnExplore the quirks of the Rust programming language that a Python developer needs to understand to code in RustUnderstand the trade-offs for multiprocessing and thread safety to write concurrent codeBuild and manage a software project with cargo and cratesFuse Rust code with Python so that Python can import and run Rust codeDeploy a Python Flask application in Docker that utilizes a private Rust pip moduleInspect and create your own Python objects in RustWho this book is for This book is for Python developers who want to speed up their Python code with Rust and implement Rust in a Python system without altering the entire system. You'll be able to learn about all topics relating to Rust programming. Basic knowledge of Python is required to get the most out of this book.

hands on game development with webassembly pdf: Practical Haskell Alejandro Serrano Mena, 2019-04-27 Get a practical, hands-on introduction to the Haskell language, its libraries and environment, and to the functional programming paradigm that is fast growing in importance in the software industry. This book contains excellent coverage of the Haskell ecosystem and supporting tools, include Cabal and Stack for managing projects, HUnit and QuickCheck for software testing, the Spock framework for developing web applications, Persistent and Esqueleto for database access, and parallel and distributed programming libraries. You'll see how functional programming is gathering momentum, allowing you to express yourself in a more concise way, reducing boilerplate, and increasing the safety of your code. Haskell is an elegant and noise-free pure functional language with a long history, having a huge number of library contributors and an active community. This makes Haskell the best tool for both learning and applying functional programming, and Practical Haskell takes advantage of this to show off the language and what it can do. What You Will Learn Get started programming with Haskell Examine the different parts of the language Gain an overview of the most important libraries and tools in the Haskell ecosystem Apply functional patterns in real-world scenarios Understand monads and monad transformers Proficiently use laziness and resource management Who This Book Is For Experienced programmers who may be new to the Haskell programming language. However, some prior exposure to Haskell is recommended.

hands on game development with webassembly pdf: Hands-On Full-Stack Web Development with ASP.NET Core Tamir Dresher, Amir Zuker, Shay Friedman, 2018-10-31 Become a full-stack developer by learning popular Microsoft technologies and platforms such as .NET Core, ASP.NET Core, Entity Framework, and Azure Key FeaturesBring static typing to web development with features compatible in TypeScript 3Implement a slim marketplace single page application (SPA) in Angular, React, and VueModernize your web apps with Microsoft Azure, Visual Studio, and GitBook Description Today, full-stack development is the name of the game. Developers who can build complete solutions, including both backend and frontend products, are in great demand in the industry, hence being able to do so a desirable skill. However, embarking on the path to becoming a modern full-stack developer can be overwhelmingly difficult, so the key purpose of this book is to simplify and ease the process. This comprehensive guide will take you through the journey of becoming a full-stack developer in the realm of the web and .NET. It begins by implementing data-oriented RESTful APIs, leveraging ASP.NET Core and Entity Framework. Afterward, it describes the web development field, including its history and future horizons. Then, you'll build

webbased Single-Page Applications (SPAs) by learning about numerous popular technologies, namely TypeScript, Angular, React, and Vue. After that, you'll learn about additional related concerns involving deployment, hosting, and monitoring by leveraging the cloud; specifically, Azure. By the end of this book, you'll be able to build, deploy, and monitor cloud-based, data-oriented, RESTful APIs, as well as modern web apps, using the most popular frameworks and technologies. What you will learn Build RESTful APIs in C# with ASP.NET Core, web APIs, and Entity Framework See the history and future horizons of the web development field Bring static-typing to web apps using TypeScript Build web applications using Angular, React, and Vue Deploy your application to the cloud Write web applications that scale, can adapt to changes, and are easy to maintain Discover best practices and real-world tips and tricks Secure your backend server with Authentication and Authorization using OAuth 2.0 Who this book is for This book is for developers who are keen on strengthening their skills in the field of cloud-based full-stack web development. You need basic knowledge of web-related pillars, including HTML, CSS, and JavaScript, as well as C# and REST. This book targets novice developers in the realm of Web development and ASP.NET who desire to advance to modern Web and ASP.NET Core development and leverage the Cloud to manage and bring everything together.

hands on game development with webassembly pdf: Rust Web Development Bastian Gruber, 2023-03-07 Create bulletproof, high-performance web apps and servers with Rust. In Rust Web Development you will learn: Handling the borrow checker in an asynchronous environment Learning the ingredients of an asynchronous Rust stack Creating web APIs and using JSON in Rust Graceful error handling Testing, tracing, logging, and debugging Deploying Rust applications Efficient database access Rust Web Development is a pragmatic, hands-on guide to creating server-based web applications with Rust. If you've designed web servers using Java, NodeJS, or PHP, you'll instantly fall in love with the performance and development experience Rust delivers. Hit the ground running! Author Bastian Gruber's sage advice makes it easy to start tackling complex problems with Rust. You'll learn how to work efficiently using pure Rust, along with important Rust libraries such as tokio for async runtimes, warp for web servers and APIs, and reqwest to run external HTTP requests. About the technology If you're sick of cookie-cutter web development tools that are slow, resource hungry, and unstable, Rust is the solution. Rust services deliver rock-solid safety guarantees, an amazing developer experience, and even a compiler that automatically prevents common mistakes! About the book Rust Web Development, teaches you to build server-side web apps using Rust, along with important Rust libraries like tokio for async runtimes, warp for web servers and APIs, and reqwest to run external HTTP requests. The book is packed full of examples, code samples, and pro tips for setting up your projects and organizing your code. As you go, you'll build a complete Q&A web service and iterate on your code chapter-by-chapter, just like a real development project. What's inside Handle the borrow checker in an asynchronous environment Build web APIs and handle JSON Compose a tech stack for asynchronous Rust development Handle errors gracefully Test, trace, log, and debug Deploy Rust applications to multiple environments About the reader This book is for web developers familiar with Java, Node, or Go, and the absolute basics of Rust. About the author Bastian Gruber was part of the official Rust Async Working Group, and founded the Rust and Tell Berlin MeetUp group. Table of Contents PART 1 INTRODUCTION TO RUST 1 Why Rust? 2 Laying the foundation PART 2 GETTING STARTED 3 Create your first route handler 4 Implement a RESTful API 5 Clean up your codebase 6 Logging, tracing, and debugging 7 Add a database to your application 8 Integrate third-party APIs PART 3 BRING IT INTO PRODUCTION 9 Add authentication and authorization 10 Deploy your application 11 Testing your Rust application

hands on game development with webassembly pdf: Mobile Applications Development Tarkeshwar Barua, Ruchi Doshi, Kamal Kant Hiran, 2020-12-16 The book covers the concepts of Python programming language along with mobile application development. Starting from fundamentals, the book continues with the explanation of mobile app development using Kivy framework. All the chapters offer questions and exercises for to better understanding of the subject.

At the end of the book some hands-on projects are given to help the readers to improve their programming and project development skills.

hands on game development with webassembly pdf: *Mastering Ethereum* Andreas M. Antonopoulos, Gavin Wood Ph.D., 2018-11-13 Ethereum represents the gateway to a worldwide, decentralized computing paradigm. This platform enables you to run decentralized applications (DApps) and smart contracts that have no central points of failure or control, integrate with a payment network, and operate on an open blockchain. With this practical guide, Andreas M. Antonopoulos and Gavin Wood provide everything you need to know about building smart contracts and DApps on Ethereum and other virtual-machine blockchains. Discover why IBM, Microsoft, NASDAQ, and hundreds of other organizations are experimenting with Ethereum. This essential guide shows you how to develop the skills necessary to be an innovator in this growing and exciting new industry. Run an Ethereum client, create and transmit basic transactions, and program smart contracts Learn the essentials of public key cryptography, hashes, and digital signatures Understand how wallets hold digital keys that control funds and smart contracts Interact with Ethereum clients programmatically using JavaScript libraries and Remote Procedure Call interfaces Learn security best practices, design patterns, and anti-patterns with real-world examples Create tokens that represent assets, shares, votes, or access control rights Build decentralized applications using multiple peer-to-peer (P2P) components

hands on game development with webassembly pdf: *97 Things Every Java Programmer Should Know* Kevlin Henney, Trisha Gee, 2020-05-15 If you want to push your Java skills to the next level, this book provides expert advice from Java leaders and practitioners. You'll be encouraged to look at problems in new ways, take broader responsibility for your work, stretch yourself by learning new techniques, and become as good at the entire craft of development as you possibly can. Edited by Kevlin Henney and Trisha Gee, *97 Things Every Java Programmer Should Know* reflects lifetimes of experience writing Java software and living with the process of software development. Great programmers share their collected wisdom to help you rethink Java practices, whether working with legacy code or incorporating changes since Java 8. A few of the 97 things you should know: Behavior Is Easy, State Is Hard—Edson Yanaga “Learn Java Idioms and Cache in Your Brain”—Jeanne Boyarsky “Java Programming from a JVM Performance Perspective”—Monica Beckwith Garbage Collection Is Your Friend—Holly K Cummins “Java's Unspeakable Types”—Ben Evans The Rebirth of Java—Sander Mak “Do You Know What Time It Is?”—Christin Gorman

hands on game development with webassembly pdf: *Professional Visual Studio 2017* Bruce Johnson, 2017-10-18 Skip the basics and delve right into Visual Studio 2017 advanced features and tools Professional Visual Studio 2017 is the industry-favorite guide to getting the most out of Microsoft's primary programming technology. From touring the new UI to exploiting advanced functionality, this book is designed to help professional developers become more productive. A unique IDE-centric approach provides a clear path through the typical workflow while exploring the nooks and crannies that can make your job easier. Visual Studio 2017 includes a host of features aimed at improving developer productivity and UI, and this book covers them all with clear explanation, new figures, and expert insight. Whether you're new to VS or just upgrading, this all-inclusive guide is an essential resource to keep within arm's reach. Visual Studio 2017 fixes the crucial issues that kept professionals from adopting VS 2015, and includes new features and tools that streamline the developer's job. This book provides the straightforward answers you need so you can get up to speed quickly and get back to work. Master the core functionality of Visual Studio 2017 Dig into the tools that make writing code easier Tailor the environment to your workflow, not the other way around Work your way through configuration, debugging, building, deployment, customizing, and more Microsoft is changing their release cadence—it's only been about two years since the last release—so developers need to quickly get a handle on new tools and features if they hope to remain productive. The 2017 release is designed specifically to help you get more done, in less time, with greater accuracy and attention to detail. If you're ready to get acquainted, Professional Visual Studio 2017 is your ideal guide.

hands on game development with webassembly pdf: Beginning Rust Programming Ric Messier, 2021-03-09 Quickly learn the ropes with the Rust programming language using this practical, step-by-step guide In Beginning Rust Programming, accomplished programmer and author Ric Messier delivers a highly practical, real-world guide to coding with Rust. Avoiding dry, theoretical content and “Hello, world”-type tutorials of questionable utility, the book dives immediately into functional Rust programming that takes advantage of the language’s blazing speed and memory efficiency. Designed from the ground up to give you a running start to using the multiparadigm system programming language, this book will teach you to: Solve real-world computer science problems of practical importance Use Rust’s rich type system and ownership model to guarantee memory-safety and thread-safety Integrate Rust with other programming languages and use it for embedded devices Perfect for programmers with some experience in other languages, like C or C++, Beginning Rust Programming is also a great pick for students new to programming and seeking a user-friendly and robust language with which to start their coding career.

hands on game development with webassembly pdf: Hands-On Smart Contract Development with Hyperledger Fabric V2 Matt Zand, Xun (Brian) Wu, Mark Anthony Morris, 2021-09-09 Blockchain technology continues to disrupt a wide variety of organizations, from small businesses to the Fortune 500. Today hundreds of blockchain networks are in production, including many built with Hyperledger Fabric. This practical guide shows developers how the latest version of this blockchain infrastructure provides an ideal foundation for developing enterprise blockchain applications or solutions. Authors Matt Zand, Xun Wu, and Mark Anthony Morris demonstrate how the versatile design of Hyperledger Fabric 2.0 satisfies a broad range of industry use cases. Developers with or without previous Hyperledger experience will discover why no other distributed ledger technology framework enjoys such wide adoption by cloud service providers such as Amazon, Alibaba, IBM, Google, and Oracle. Walk through the architecture and components of Hyperledger Fabric 2.0 Migrate your current Hyperledger Fabric projects to version 2.0 Develop blockchain applications on the Hyperledger platform with Node.js Deploy and integrate Hyperledger on Amazon Managed Blockchain, IBM Cloud, and Oracle Cloud Develop blockchain applications with Hyperledger Aries, Avalon, Besu, and Grid Build end-to-end blockchain supply chain applications with Hyperledger

hands on game development with webassembly pdf: Practical Rust Projects Shing Lyu, 2020-02-27 Go beyond the basics and build complete applications using the Rust programming language. The applications in this book include a high-performance web client, a microcontroller (for a robot, for example), a game, an app that runs on Android, and an application that incorporates AI and machine learning. Each chapter will be organized in the following format: what this kind of application looks like; requirements and user stories of our example program; an introduction to the Rust libraries used; the actual implementation of the example program, including common pitfalls and their solutions; and a brief comparison of libraries for building each application, if there is no clear winner. Practical Rust Projects will open your eyes to the world of practical applications of Rust. After reading the book, you will be able to apply your Rust knowledge to build your own projects. What You Will Learn Write Rust code that runs on microcontrollers Build a 2D game Create Rust-based mobile Android applications Use Rust to build AI and machine learning applications Who This Book Is For Someone with basic Rust knowledge, wishing to learn more about how to apply Rust in a real-world scenario.

hands on game development with webassembly pdf: Learning ASP.NET Core 3.0 -Second Edition Kenneth Yamikani Fukizi, Jason de Oliveira, Michel Bruchet, 2019-12-27 A beginner's guide to building fully functioning web applications from scratch using the latest features of ASP.NET Core 3 and C# 8 Key Features Get to grips with the new features and APIs in ASP.NET Core 3, EF Core 3, and Blazor Create web APIs that integrate your applications with other systems and services Learn to deploy your web applications in new environments such as the cloud and Docker containers Book Description ASP.NET Core is an open source framework from Microsoft that

makes it easy to build highly efficient and dynamic cross-platform web applications. Updated for the latest features of ASP.NET Core 3, this second edition will equip you with the skills you need to build powerful web applications. The book starts with an introduction to ASP.NET Core and its features, giving you a complete understanding of the framework. You will also learn how to set up your development environment with Visual Studio 2019 and build a fully functioning application from scratch. You'll then understand core concepts for building web applications such as Model View Controller (MVC), dependency injection, and WebSockets. As you advance, you'll discover how to use Entity Framework Core 3 to automate all database-related activities for your application. You will then build and document secure web APIs using security best practices to protect your web applications from threats and vulnerabilities. Finally, you will learn how to use Azure DevOps as a CI/CD tool to deploy and monitor your applications using Microsoft Azure, Amazon Web Services (AWS), and Docker. By the end of this book, you'll have the skills you need to develop efficient and robust web applications in ASP.NET Core 3. What you will learn

- Build an MVC web application and use Entity Framework Core 3 to access data
- Add web APIs to your web applications using RPC, REST, and HATEOAS
- Create a fully automated continuous integration and continuous delivery (CI/CD) pipeline using Azure DevOps
- Use Azure, Amazon Web Services, and Docker to deploy and monitor your applications
- Secure your web application from common attacks such as Cross-Site Scripting and SQL injection
- Explore client-side development using C# Razor components

Who this book is for This book is for developers who want to build modern web applications with ASP.NET Core. The book will also be helpful for anyone working in infrastructure engineering and operations to monitor and diagnose problems during the runtime of ASP.NET Core 3.0 web applications. Although no prior understanding of ASP.NET or .NET Core is required, basic C# programming knowledge is assumed.

hands on game development with webassembly pdf: Working Effectively with Legacy Code Michael Feathers, 2004-09-22 Get more out of your legacy systems: more performance, functionality, reliability, and manageability Is your code easy to change? Can you get nearly instantaneous feedback when you do change it? Do you understand it? If the answer to any of these questions is no, you have legacy code, and it is draining time and money away from your development efforts. In this book, Michael Feathers offers start-to-finish strategies for working more effectively with large, untested legacy code bases. This book draws on material Michael created for his renowned Object Mentor seminars: techniques Michael has used in mentoring to help hundreds of developers, technical managers, and testers bring their legacy systems under control. The topics covered include Understanding the mechanics of software change: adding features, fixing bugs, improving design, optimizing performance Getting legacy code into a test harness Writing tests that protect you against introducing new problems Techniques that can be used with any language or platform—with examples in Java, C++, C, and C# Accurately identifying where code changes need to be made Coping with legacy systems that aren't object-oriented Handling applications that don't seem to have any structure This book also includes a catalog of twenty-four dependency-breaking techniques that help you work with program elements in isolation and make safer changes.

hands on game development with webassembly pdf: C# 9 and .NET 5 - Modern Cross-Platform Development Mark J. Price, 2020-11-10 Publisher's Note: Microsoft stopped supporting .NET 5 in May 2022. The newer 8th edition of the book is available that covers .NET 8 (end-of-life November 2026) with C# 12 and EF Core 8. Purchase of the print or Kindle book includes a free PDF eBook Key Features Explore the newest additions to C# 9, the .NET 5 class library, Entity Framework Core and Blazor Strengthen your command of ASP.NET Core 5.0 and create professional websites and services Build cross-platform apps for Windows, macOS, Linux, iOS, and Android Book DescriptionIn C# 9 and .NET 5 - Modern Cross-Platform Development, Fifth Edition, expert teacher Mark J. Price gives you everything you need to start programming C# applications. This latest edition uses the popular Visual Studio Code editor to work across all major operating systems. It is fully updated and expanded with a new chapter on the Microsoft Blazor framework. The book's first part teaches the fundamentals of C#, including object-oriented

programming and new C# 9 features such as top-level programs, target-typed new object instantiation, and immutable types using the record keyword. Part 2 covers the .NET APIs, for performing tasks like managing and querying data, monitoring and improving performance, and working with the file system, async streams, serialization, and encryption. Part 3 provides examples of cross-platform apps you can build and deploy, such as websites and services using ASP.NET Core or mobile apps using Xamarin.Forms. The best type of application for learning the C# language constructs and many of the .NET libraries is one that does not distract with unnecessary application code. For that reason, the C# and .NET topics covered in Chapters 1 to 13 feature console applications. In Chapters 14 to 20, having mastered the basics of the language and libraries, you will build practical applications using ASP.NET Core, Model-View-Controller (MVC), and Blazor. By the end of the book, you will have acquired the understanding and skills you need to use C# 9 and .NET 5 to create websites, services, and mobile apps. What you will learn Build your own types with object-oriented programming Query and manipulate data using LINQ Build websites and services using ASP.NET Core 5 Create intelligent apps using machine learning Use Entity Framework Core and work with relational databases Discover Windows app development using the Universal Windows Platform and XAML Build rich web experiences using the Blazor framework Build mobile applications for iOS and Android using Xamarin.Forms Who this book is for This book is best for C# and .NET beginners, or programmers who have worked with C# in the past but feel left behind by the changes in the past few years. This book doesn't expect you to have any C# or .NET experience; however, you should have a general understanding of programming. Students and professionals with a science, technology, engineering, or mathematics (STEM) background can certainly benefit from this book.

hands on game development with webassembly pdf: [Programming Rust](#) Jim Blandy, Jason Orendorff, Leonora F.S. Tindall, 2021-06-11 Systems programming provides the foundation for the world's computation. Writing performance-sensitive code requires a programming language that puts programmers in control of how memory, processor time, and other system resources are used. The Rust systems programming language combines that control with a modern type system that catches broad classes of common mistakes, from memory management errors to data races between threads. With this practical guide, experienced systems programmers will learn how to successfully bridge the gap between performance and safety using Rust. Jim Blandy, Jason Orendorff, and Leonora Tindall demonstrate how Rust's features put programmers in control over memory consumption and processor use by combining predictable performance with memory safety and trustworthy concurrency. You'll learn: Rust's fundamental data types and the core concepts of ownership and borrowing How to write flexible, efficient code with traits and generics How to write fast, multithreaded code without data races Rust's key power tools: closures, iterators, and asynchronous programming Collections, strings and text, input and output, macros, unsafe code, and foreign function interfaces This revised, updated edition covers the Rust 2021 Edition.

hands on game development with webassembly pdf: [A Software Engineer Learns HTML5, JavaScript and JQuery](#) Dane Cameron, 2013-11-25 A Software Engineer Learns HTML5, JavaScript and jQuery guides you through the process I went through as an experienced software engineer, writing a large-scale, standards based web-application for the first time. It is intended to teach you the fundamentals of HTML5, JavaScript and jQuery - without presenting you with long lists of APIs, or intricate details of every feature (these can be found in reference manuals). -- Amazon.com viewed July 29, 2020.

hands on game development with webassembly pdf: [Command-Line Rust](#) Ken Youens-Clark, 2022-01-13 Updated in 2024: A new version has been released that simplifies the programs used in the book, based on changes in the Rust language and crates since original publication. The code has been updated to reflect version 4 of the clap crate. For several consecutive years, Rust has been voted most loved programming language in Stack Overflow's annual developer survey. This open source systems programming language is now used for everything from game engines and operating systems to browser components and virtual reality simulation engines. But Rust is also an incredibly

complex language with a notoriously difficult learning curve. Rather than focusing on the language as a whole, this guide teaches Rust using a single small, complete, focused program in each chapter. Author Ken Youens-Clark shows you how to start, write, and test each of these programs to create a finished product. You'll learn how to handle errors in Rust, read and write files, and use regular expressions, Rust types, structs, and more. Discover how to: Use Rust's standard libraries and data types such as numbers, strings, vectors, structs, Options, and Results to create command-line programs Write and test Rust programs and functions Read and write files, including stdin, stdout, and stderr Document and validate command-line arguments Write programs that fail gracefully Parse raw and delimited text manually, using regular expressions and Rust crates Use and control randomness

hands on game development with webassembly pdf: 3D Graphics Rendering Cookbook Sergey Kosarevsky, Viktor Latypov, 2021-08-25 Build a 3D rendering engine from scratch while solving problems in a step-by-step way with the help of useful recipes Key FeaturesLearn to integrate modern rendering techniques into a single performant 3D rendering engineLeverage Vulkan to render 3D content, use AZDO in OpenGL applications, and understand modern real-time rendering methodsImplement a physically based rendering pipeline from scratch in Vulkan and OpenGLBook Description OpenGL is a popular cross-language, cross-platform application programming interface (API) used for rendering 2D and 3D graphics, while Vulkan is a low-overhead, cross-platform 3D graphics API that targets high-performance applications. 3D Graphics Rendering Cookbook helps you learn about modern graphics rendering algorithms and techniques using C++ programming along with OpenGL and Vulkan APIs. The book begins by setting up a development environment and takes you through the steps involved in building a 3D rendering engine with the help of basic, yet self-contained, recipes. Each recipe will enable you to incrementally add features to your codebase and show you how to integrate different 3D rendering techniques and algorithms into one large project. You'll also get to grips with core techniques such as physically based rendering, image-based rendering, and CPU/GPU geometry culling, to name a few. As you advance, you'll explore common techniques and solutions that will help you to work with large datasets for 2D and 3D rendering. Finally, you'll discover how to apply optimization techniques to build performant and feature-rich graphics applications. By the end of this 3D rendering book, you'll have gained an improved understanding of best practices used in modern graphics APIs and be able to create fast and versatile 3D rendering frameworks. What you will learnImprove the performance of legacy OpenGL applicationsManage a substantial amount of content in real-time 3D rendering enginesDiscover how to debug and profile graphics applicationsUnderstand how to use the Approaching Zero Driver Overhead (AZDO) philosophy in OpenGLIntegrate various rendering techniques into a single applicationFind out how to develop Vulkan applicationsImplement a physically based rendering pipeline from scratchIntegrate a physics library with your rendering engineWho this book is for This book is for 3D graphics developers who are familiar with the mathematical fundamentals of 3D rendering and want to gain expertise in writing fast rendering engines with advanced techniques using C++ libraries and APIs. A solid understanding of C++ and basic linear algebra, as well as experience in creating custom 3D applications without using premade rendering engines is required.

hands on game development with webassembly pdf: JavaScript Frameworks for Modern Web Development Sufyan bin Uzayr, Nicholas Cloud, Tim Ambler, 2019-10-31 Enrich your software design skills and take a guided tour of the wild, vast, and untamed frontier that is JavaScript development. Especially useful for frontend developers, this revision includes specific chapters on React and VueJS, as well as an updated one on Angular. To help you get the most of your new skills, each chapter also has a further reading section. This book will serve as an introduction to both new and well established libraries and frameworks, such as Angular, VueJS, React, Grunt, Yeoman, RequireJS, Browserify, Knockout, Kraken, Async.js, Underscore, and Lodash. It also covers utilities that have gained popular traction and support from seasoned developers and tools applicable to the entire development stack, both client- and server-side. While no single book

can possibly cover every JavaScript library of value, JavaScript Frameworks for Modern Web Development focuses on incredibly useful libraries and frameworks that production software uses. You will be treated to detailed analyses and sample code for tools that manage dependencies, structure code in a modular fashion, automate repetitive build tasks, create specialized servers, structure client side applications, facilitate horizontal scaling, and interacting with disparate data stores. What You'll Learn Work with a variety of JavaScript frameworks, such as Angular, Vue, React, RequireJS, Knockout, and more Choose the right framework for different types of projects Employ the appropriate libraries and tools in your projects Discover useful JavaScript development tools such as Grunt, Yeoman, Lodash, etc. Who This Book Is For Web developers of all levels of ability; particularly relevant for front-end developers, server-side coders, and developers interested in learning JavaScript.

hands on game development with webassembly pdf: Software Architecture with C# 9 and .NET 5 Gabriel Baptista, Francesco Abbruzzese, 2020-12-28 Design scalable and high-performance enterprise applications using the latest features of C# 9 and .NET 5 Key Features Gain fundamental and comprehensive software architecture knowledge and the skillset to create fully modular apps Design high-performance software systems using the latest features of .NET 5 and C# 9 Solve scalability problems in web apps using enterprise architecture patterns Book Description Software architecture is the practice of implementing structures and systems that streamline the software development process and improve the quality of an app. This fully revised and expanded second edition, featuring the latest features of .NET 5 and C# 9, enables you to acquire the key skills, knowledge, and best practices required to become an effective software architect. This second edition features additional explanation of the principles of Software architecture, including new chapters on Azure Service Fabric, Kubernetes, and Blazor. It also includes more discussion on security, microservices, and DevOps, including GitHub deployments for the software development cycle. You will begin by understanding how to transform user requirements into architectural needs and exploring the differences between functional and non-functional requirements. Next, you will explore how to carefully choose a cloud solution for your infrastructure, along with the factors that will help you manage your app in a cloud-based environment. Finally, you will discover software design patterns and various software approaches that will allow you to solve common problems faced during development. By the end of this book, you will be able to build and deliver highly scalable enterprise-ready apps that meet your organization's business requirements. What you will learn Use different techniques to overcome real-world architectural challenges and solve design consideration issues Apply architectural approaches such as layered architecture, service-oriented architecture (SOA), and microservices Leverage tools such as containers, Docker, Kubernetes, and Blazor to manage microservices effectively Get up to speed with Azure tools and features for delivering global solutions Program and maintain Azure Functions using C# 9 and its latest features Understand when it is best to use test-driven development (TDD) as an approach for software development Write automated functional test cases Get the best of DevOps principles to enable CI/CD environments Who this book is for This book is for engineers and senior software developers aspiring to become architects or looking to build enterprise applications with the .NET Stack. Basic familiarity with C# and .NET is required to get the most out of this book.

hands on game development with webassembly pdf: Visual Studio 2019 In Depth Du Preez Ockert J., 2019-09-20 Step-by-step guide to all the tools and extensions in the Visual Studio 2019 IDE Key features a- Create and use custom IDE extensions a- Find, download, and use the best IDE extensions for web, mobile, Azure, and Windows a- Enhance programming experience and time with debugging tools a- Enhance coding capabilities with coding tools a- Test projects proactively a- Create powerful web, mobile, and Azure solutions for the real world Description This book peeks into every corner of the Visual Studio IDE and will help you get started with the latest 2019 version. Right from installation, you'll discover new features within the tool and the optimal way to use the features you may already know. You'll learn, for example, how to extend Visual Studio with your own

customizations, so that you can make it perform the way you want. You will then explore everything about NuGet package, test applications using Live Unit Testing, and learn how to make code templates using the T4 code generation tool. You'll get to grips with the richer JavaScript IntelliSense, which will help you focus more on coding. Moving on, you'll learn to work with the dedicated workloads for data storage and data science. You will also review the more advanced architecture tools concealed within the IDE and finally create cloud-first applications powered by Microsoft Azure using the built-in suite of Azure tools. What will you learn By the end of the book, you will be able to tackle any solution for any platform head-on. You will create real-world solutions from start to finish. By using the tools and extensions outlined in this book, you will be able to code better and faster, debug better, share your code with more peers, test your code better, and install or publish your apps quicker and without issues. Who this book is for The book is intended for any .NET developer. You can be a seasoned developer or a newbie just starting out. This book will play a pivotal role in presenting all the tools you need to become a better developer. Table of contents 1. Getting started with Visual Studio 2. Digging in the Visual Studio IDE 3. IntelliSense 4. Language & coding changes in C# 5. What's new in .Net core 6. Built-in tools 7. Debugging tools 8. Testing tools 9. ASP.NET tools 10. Mobile tools 11. Azure tools 12. IDE extensions 13. ASP.NET extensions 14. Mobile extensions 15. Azure DevOps extensions About the author Ockert du Preez is a self-taught developer who started learning programming since the days of QBasic. He has written several articles over the years detailing his programming quests and adventures. .NET is his second love, just after his wife and kid. He has always been an avid supporter of .NET since the beginning, and is an expert in VB and C#. He was given the Microsoft Most Valuable Professional Award for .NET (2008-2017). He has worked as a moderator and an article reviewer and currently writes articles for CodeGuru, Developer.com, DevX, and the Database journal. His blog: <https://www.codeguru.com/member.php/Hannes+DuPreez/>

hands on game development with webassembly pdf: Learn OpenGL ES Prateek Mehta, 2013-08-29 Want to create sophisticated games and graphics-intensive apps? Learn OpenGL ES gets you started immediately with OpenGL ES. After mastering the basics of OpenGL ES itself, you will quickly find yourself writing and building game apps, without having to learn about object oriented programming techniques. This book demonstrates the use of a powerful open-source modeling tool, Blender. You will be guided, step by step, through the development of Tank Fence, a dynamic, interactive 3D game. Along the way you'll gain skills in building apps with Eclipse and the Android SDK or NDK, rendering graphics using hardware acceleration, and multithreading for performance and responsiveness. iOS developers will also find this book's information invaluable when writing their apps. You'll learn everything you need to know about: Creating simple, efficient game UIs Designing the basic building blocks of an exciting, interactive 3D game Pulling all the elements together with Blender, a powerful open-source tool for modeling, animation, rendering, compositing, video editing, and game creation Taking the next big step using custom and inbuilt functions, texturing, shading, light sources, and more Refining your mobile game app through collision detection, player-room-obstacle classes, and storage classes Doing all this efficiently on mobile devices with limited resources and processing What you'll learn How to install and use OpenGL ES 2.0 on Android GLSL ES Fundamentals State Management Modeling 3D Objects Using Blender Using the Perl Mesh Parser Vertex Buffer Objects Using Color Masks sampler2D and samplerCube Uniforms Multi-Texturing Lambert Illumination Model Implementing the Lighting Equation Design, write, and build Tank Fence, an interactive 3D game Who this book is for Learn OpenGL ES is ideal for mobile game and interactive app developers who want to know more about the OpenGL ES engine and use it to build more sophisticated, graphically-rich games and other apps. While the code is developed on Android, iOS developers will also find this book invaluable. Table of Contents 1. Why OpenGL ES? 2. UI for games: Keep it simple 3. First Steps: Mobile Game App Development 4. 3D Modeling 5. Functions, Shading, Light Source and Objects 6. Carrying Further: Collision Detection

hands on game development with webassembly pdf: Mastering Rust Rahul Sharma, Vesa

Kaihlaivirta, 2019-01-31 Become proficient in designing, developing and deploying effective software systems using the advanced constructs of Rust Key FeaturesImprove your productivity using the latest version of Rust and write simpler and easier codeUnderstand Rust's immutability and ownership principle, expressive type system, safe concurrencyDeep dive into the new domains of Rust like WebAssembly, Networking and Command line toolsBook Description Rust is an empowering language that provides a rare combination of safety, speed, and zero-cost abstractions. Mastering Rust - Second Edition is filled with clear and simple explanations of the language features along with real-world examples, showing you how you can build robust, scalable, and reliable programs. This second edition of the book improves upon the previous one and touches on all aspects that make Rust a great language. We have included the features from latest Rust 2018 edition such as the new module system, the smarter compiler, helpful error messages, and the stable procedural macros. You'll learn how Rust can be used for systems programming, network programming, and even on the web. You'll also learn techniques such as writing memory-safe code, building idiomatic Rust libraries, writing efficient asynchronous networking code, and advanced macros. The book contains a mix of theory and hands-on tasks so you acquire the skills as well as the knowledge, and it also provides exercises to hammer the concepts in. After reading this book, you will be able to implement Rust for your enterprise projects, write better tests and documentation, design for performance, and write idiomatic Rust code. What you will learnWrite generic and type-safe code by using Rust's powerful type system How memory safety works without garbage collection Know the different strategies in error handling and when to use themLearn how to use concurrency primitives such as threads and channels Use advanced macros to reduce boilerplate code Create efficient web applications with the Actix-web frameworkUse Diesel for type-safe database interactions in your web applicationWho this book is for The book is aimed at beginner and intermediate programmers who already have familiarity with any imperative language and have only heard of Rust as a new language. If you are a developer who wants to write robust, efficient and maintainable software systems and want to become proficient with Rust, this book is for you. It starts by giving a whirlwind tour of the important concepts of Rust and covers advanced features of the language in subsequent chapters using code examples that readers will find useful to advance their knowledge.

Back to Home: <https://a.comtex-nj.com>