A Programmers To Sound

Offering the most comprehensive, up-to-date coverage available, MODERN RADIO AND AUDIO PRODUCTION: PROGRAMMING AND PERFORMANCE, 10e combines the latest trends and technologies with explanations of traditional equipment and practices. The authors' clear writing style, excellent descriptions and explanations, and attention to detail make the text extremely reader friendly. In addition to new examples, illustrations, and photos throughout, the text's three all-new chapters focus on writing, ethics, and mobile radio. Important Notice: Media content referenced within the product description or the product text may not be available in the ebook version.

One CD-ROM disc in pocket.

New to this second edition are the following: evolutionary computing and its relevance to sound design, PSOLA techniques, granular and pulsar synthesis, artificial intelligence, humanoid singing and the use of supercomputers in sound synthesis.

An encyclopedic handbook on audio programming for students and professionals, with many cross-platform open source examples and a DVD covering advanced topics. This comprehensive handbook of mathematical and programming techniques for audio signal processing will be an essential reference for all computer musicians, computer scientists, engineers, and anyone interested in audio. Designed to be used by readers with varying levels of programming expertise, it not only provides the foundations for music and audio development but also tackles issues that sometimes remain mysterious even to experienced software designers. Exercises and copious examples (all cross-platform and based on free or open source software) make the book ideal for classroom use. Fifteen chapters and eight appendixes cover such topics as programming basics for C and C++ (with music-oriented examples), audio programming basics and more advanced topics, spectral audio programming; programming Csound opcodes, and algorithmic synthesis and music programming. Appendixes cover topics in compiling, audio and MIDI, computing, and math. An accompanying DVD provides an additional 40 chapters, covering musical and audio programs with micro-controllers, alternate MIDI controllers, video controllers, developing Apple Audio Unit plug-ins from Csound opcodes, and audio programming for the iPhone. The sections and chapters of the book are arranged progressively and topics can be followed from chapter to chapter and from section to section. At the same time, each section can stand alone as a self-contained unit. Readers will find The Audio Programming Book a trustworthy companion on their journey through making music and programming audio on modern computers.

The Ultimate Intermediate Guide to Learn Arduino Programming Step by Step
Beep to Boom
Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming
An Introduction to Computer Programming and Digital Signal Processing in MATLAB
Csound
A Sound and Music Computing System

The Programmer's Brain unlocks the way we think about code. It offers scientifically sound techniques that can radically improve the way you master new technology, comprehend code, and

memorize syntax. You'll learn how to benefit from productive struggle and turn confusion into a learning tool. Along the way, you'll discover how to create study resources as you become an expert at teaching yourself and bringing new colleagues up to speed.

Taking programmers through the complete development process for a game audio engine, this practical handbook offers detailed explanations of basic WAV sound effect playback, as well as the techniques of audio scripts and ambient sound, and demonstrates the use of MP3, Windows Media, S3M/IT/MOD, CD audio, and Ogg. Original. (Beginner)

This book is a standard tutorial targeted at game developers which aims to help them incorporate audio programming techniques to enhance their gameplay experience.This book is perfect for C++ game developers who have no experience with audio programming and who would like a quick introduction to the most important topics required to integrate audio into a game.

This comprehensive introduction to software synthesis techniques and programming is intended for students, researchers, musicians, sound artists and enthusiasts in the field of music technology. The art of sound synthesis is as important for the electronic musician as the art of orchestration is important for symphonic music composers. Those who wish to create their own virtual orchestra of electronic instruments and produce original sounds will find this book invaluable. It examines a variety of synthesis techniques and illustrates how to turn a personal computer into a powerful and flexible sound synthesiser. The book also discusses a number of ongoing developments that may play an important role in the future of electronic music making. Previously published as Computer Sound Synthesis for the Electronic Musician, this second edition features a foreword by Jean-Claude Risset and provides new information on: · the latest directions in digital sound representation · advances in physical modelling techniques · granular and pulsar synthesis · PSOLA technique · humanoid voice synthesis · artificial intelligence · evolutionary computing The accompanying CD-ROM contains examples, complementary tutorials and a number of synthesis systems for PC and Macintosh platforms, ranging from low level synthesis programming languages to graphic front-ends for instrument and sound design. These include fully working packages, demonstration versions of commercial software and experimental programs from top research centres in Europe, North and South America.

Arduino Programming

The Development of Advanced Runtime Sound Systems for Games and Extended Reality

How to Make a Noise
Spectral Music Design
Making Music with Java
Programming Sound for DOS and Windows
A Programmer's Guide to SoundAddison-Wesley Professional
An all-in-one introduction to implementing sound, this guide provides
a comprehensive practical resource for programmers. Tim Kientzle,
technical editor of "Dr. Dobb's Journal", presents the basic principles
of sound and sound processing, together with concrete
implementation details for a variety of sound file formats and
algorithms. The CD-ROM includes royalty-free sound libraries and a
rich collection of utilities.
Martin Wilde's cutting-edge exploration of the creative potential of
game audio systems addresses the latest working methods of those
involved in creating and programming immersive, interactive and non-
linear audio for games. The book demonstrates how the game
programmer can create an software system which enables the audio
content provider (composer/sound designer) to maintain direct control
over the composition and presentation of an interactive game
soundtrack. This system (which is platform independent) is described
step-by-step in Wilde's approachable style with illuminating case
studies, all source codes for which are provided on the accompanying
CD-Rom which readers can use to develop their own audio engines. As
a programmer with experience of developing sound and music
software engines for computer game titles on a multitude of platforms
who is also an experienced musician, Martin Wilde is uniquely placed
to address individuals approaching game audio from various levels
and areas of expertise. Game audio programmers will learn how to
achieve enable even better audio soundtracks and effects, while
musicians who want to capitalise on this shift in roles will gain a
greater appreciation of the technical issues involved, so enhancing
their employment prospects. Students of game design can practice
these skills by building their own game audio engines based on the
source code provided.
A practitioner's guide to the basic principles of creating sound effects
using easily accessed free software. Designing Sound teaches
students and professional sound designers to understand and create
sound effects starting from nothing. Its thesis is that any sound can be
generated from first principles, guided by analysis and synthesis. The
text takes a practitioner's perspective, exploring the basic principles
of making ordinary, everyday sounds using an easily accessed free
software. Readers use the Pure Data (Pd) language to construct sound
objects, which are more flexible and useful than recordings. Sound is

considered as a process, rather than as data—an approach sometimes known as "procedural audio." Procedural sound is a living sound effect that can run as computer code and be changed in real time according to unpredictable events. Applications include video games, film, animation, and media in which sound is part of an interactive process. The book takes a practical, systematic approach to the subject, teaching by example and providing background information that offers a firm theoretical context for its pragmatic stance. [Many of the examples follow a pattern, beginning with a discussion of the nature and physics of a sound, proceeding through the development of models and the implementation of examples, to the final step of producing a Pure Data program for the desired sound. Different synthesis methods are discussed, analyzed, and refined throughout.] After mastering the techniques presented in Designing Sound, students will be able to build their own sound objects for use in interactive applications and other projects
Exploring Visual Representation of Sound in Computer Music
Software Through Programming and Composition
Embedded Programming with Microcontrollers and Python
The Csound Book
A Programmer's Guide to Sound
A Computational Approach
The Ultimate Reference for Sound Design

This book is your how-to guide for programming audio and sound for the Open Source Linux Operating System. Readers will learn how to utilize DSPs, sampled audio, MIDI, karaoke, streaming audio and more. Linux is a major operating system that can not only do what every other operating system can do, but can also do a lot more. But because of its size and complexity it can be hard to learn how to do any particular task. The Linux sound system is a major example of this: there is a large variety of tools and approaches for almost every aspect of sound. This ranges from audio codecs, to audio players, to audio support both within and outside of the Linux kernel. What you'll learn: How to do sampled audio What is and how to handle Digital Signal Processing (DSP) How to do MIDI How to build Karaoke like application How to handle streaming audio and more Who is this book for:Experienced Linux users and programmers interested in doing multimedia with Linux and perhaps even game development./div Summary Programming for Musicians and Digital Artists: Creating Music with ChucK offers a complete introduction to programming in the open source music language ChucK. In it, you'll learn the basics of digital sound creation and manipulation while you discover the ChucK language. As you move example-by-example through this easy-to-follow book, you'll create meaningful and rewarding digital compositions and "instruments" that

make sound and music in direct response to program logic, scores, gestures, and other systems connected via MIDI or the network. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About this Book A digital musician must manipulate sound precisely. ChucK is an audio-centric programming language that provides precise control over time, audio computation, and user interface elements like track pads and joysticks. Because it uses the vocabulary of sound, ChucK is easy to learn even for artists with little or no exposure to computer programming. Programming for Musicians and Digital Artists offers a complete introduction to music programming. In it, you'll learn the basics of digital sound manipulation while you learn to program using ChucK. Example-by-example, you'll create meaningful digital compositions and "instruments" that respond to program logic, scores, gestures, and other systems connected via MIDI or the network. You'll also experience how ChucK enables the on-the-fly musical improvisation practiced by communities of "live music coders" around the world. Written for readers familiar with the vocabulary of sound and music. No experience with computer programming is required. What's Inside Learn ChucK and digital music creation side-by-side Invent new sounds, instruments, and modes of performance Written by the creators of the ChucK language About the Authors Perry Cook, Ajay Kapur, Spencer Salazar, and Ge Wang are pioneers in the area of teaching and programming digital music. Ge is the creator and chief architect of the ChucK language. Table of Contents Introduction: ChucK programming for artistsPART 1 INTRODUCTION TO PROGRAMMING IN CHUCK Basics: sound, waves, and ChucK programming Libraries: ChucK's built-in tools Arrays: arranging and accessing your compositional data Sound files and sound manipulation Functions: making your own tools PART 2 NOW IT GETS REALLY INTERESTING! Unit generators: ChucK objects for sound synthesis and processing Synthesis ToolKit instruments Multithreading and concurrency: running many programs at once Objects and classes: making your own ChucK power tools Events: signaling between shreds and syncing to the outside world Integrating with other systems via MIDI, OSC, serial, and more

Created in 1985 by Barry Vercoe, Csound is one of the most widely used software sound synthesis systems. Because it is so powerful, mastering Csound can take a good deal of time and effort. But this long-awaited guide will dramatically straighten the learning curve and enable musicians to take advantage of this rich computer technology available for creating music. Written by the world's leading educators, programmers, sound designers, and composers, this comprehensive guide covers both the basics of Csound and the theoretical and musical concepts necessary to use the program effectively. The thirty-two tutorial chapters cover:

additive, subtractive, FM, AM, FOF, granular, wavetable, waveguide, vector, LA, and other hybrid methods; analysis and resynthesis using ADSYN, LP, and the Phase Vocoder; sample processing; mathematical and physical modeling; and digital signal processing, including room simulation and 3D modeling. CDs for this book are no longer produced. To request files, please email digitalproducts-cs@mit.edu.

Learn how to program JavaScript while creating interactive audio applications with JavaScript for Sound Artists: Learn to Code With the Web Audio API! William Turner and Steve Leonard showcase the basics of JavaScript language programing so that readers can learn how to build browser based audio applications, such as music synthesizers and drum machines. The companion website offers further opportunity for growth. Web Audio API instruction includes oscillators, audio file loading and playback, basic audio manipulation, panning and time. This book encompasses all of the basic features of JavaScript with aspects of the Web Audio API to heighten the capability of any browser. Key Features Uses the readers existing knowledge of audio technology to facilitate learning how to program using JavaScript. The teaching will be done through a series of annotated examples and explanations. Downloadable code examples and links to additional reference material included on the books companion website. This book makes learning programming more approachable to nonprofessional programmers The context of teaching JavaScript for the creative audio community in this manner does not exist anywhere else in the market and uses example-based teaching

Designing Sound

The Audio Programming Book

Raspberry Pi GPU Audio Video Programming

JavaScript for Sound Artists

Synthesis Techniques and Programming

Learn to Code with the Web Audio API

The aim of this thesis is to investigate techniques for, and applications of automatic sound synthesizer programming. An automatic sound synthesizer programmer is a system which removes the requirement to explicitly specify parameter settings for a sound synthesis algorithm from the user. Two forms of these systems are discussed in this thesis: tone matching programmers and synthesis space explorers. A tone matching programmer takes at its input a sound synthesis algorithm and a desired target sound. At its output it produces a configuration for the sound synthesis algorithm which causes it to emit a similar sound to the target. The techniques for achieving this that are investigated are genetic algorithms, neural networks, hill climbers and data driven approaches. A synthesis space explorer provides a user with a representation of the space of possible sounds that a synthesizer can produce and allows them to interactively explore this space. The applications of automatic sound synthesizer programming that are investigated include studio tools, an autonomous musical agent and a self-reprogramming drum machine. The research employs several methodologies: the development of novel software frameworks and tools, the examination of existing software at the source code and performance levels and user trials of the tools and software. The main

contributions made are: a method for visualisation of sound synthesis space and low dimensional control of sound synthesizers; a general purpose framework for the deployment and testing of sound synthesis and optimisation algorithms in the SuperCollider language sclang; a comparison of a variety of optimisation techniques for sound synthesizer programming; an analysis of sound synthesizer error surfaces; a general purpose sound synthesizer programmer compatible with industry standard tools; an automatic improviser which passes a loose equivalent of the Turing test for Jazz musicians, i.e. being half of a man-machine duet which was rated as one of the best sessions of 2009 on the BBC's 'Jazz on 3' programme.

Processing audio in the spectral domain has become a practical proposition for a variety of applications in computer music, composition, and sound design, making it an area of significant interest for musicians, programmers, sound designers, and researchers. While spectral processing has been explored already from a variety of perspectives, previous approaches tended to be piecemeal: some dealt with signal processing details, others with a high-level music technology discussion of techniques, some more compositionally focused, and others at music/audio programming concerns. As author Victor Lazzarini argues, the existing literature has made a good footprint in the area but has failed to integrate these various approaches within spectral audio. In Spectral Sound Design: A Computational Approach, Lazzarini provides an antidote. Spectral Sound Design: A Computational Approach gives authors a set of practical tools to implement processing techniques and algorithms in a balanced way, covering application aspects as well the fundamental theory that underpins them, within the context of contemporary and electronic music practice. The book employs a mix of Python for prototyping and Csound for deployment and music programming. The tight integration of these three languages as well as the wide scope offered by the combination (going from embedded to supercomputing, and including web-based and mobile applications) makes it the go-to resource to deal with the practical aspects of the subject.

Making Music with Java is an introduction to music making through software development in the Java programming language using the jMusic library. It explains musical and programming concepts in a coordinated way. The book is written for the musician who wishes to learn about Java programming and computer music concepts, and for the programmer who is interested in music and sound design with Java. It assumes little musical or programming experience and introduces topics and issues as they arise. Sections on computer music and programming are interlaced throughout, but kept separate enough so that those with experience in either area can skip ahead as required.

Shows How Programmers Can Achieve Sophisticated Graphics & Sound Effects on the Commodore 64

Learning Core Audio

Game Audio Programming 2

Getting Started with C++ Audio Programming for Game Development

Programming for Musicians and Digital Artists

Occupational Outlook Quarterly

Realtime and Object-Oriented Audio

Manuals

Program audio and sound for Linux using this practical, how-to guide. You will learn how to use DSPs, sampled audio, MIDI, karaoke, streaming audio, and more. Linux Sound Programming takes you through the layers of complexity involved in programming the Linux sound system. You'll see the large variety of tools and approaches that apply to almost every aspect of sound. This ranges from audio codecs, to audio players, to audio support both within and outside of the

Linux kernel. What You'll Learn Work with sampled audio Handle Digital
Signal Processing (DSP) Gain knowledge of MIDI Build a Karaoke-like
application Handle streaming audio Who This Book Is For Experienced
Linux users and programmers interested in doing multimedia with Linux.
For intermediate programmers, beginning sound designers. Sound gives
your native, web, or mobile apps that extra dimension, and it's
essential for games. Rather than using canned samples from a sample
library, learn how to build sounds from the ground up and produce them
for web projects using the Pure Data programming language. Even
better, you'll be able to integrate dynamic sound environments into
your native apps or games--sound that reacts to the app, instead of
sounding the same every time. Start your journey as a sound designer,
and get the power to craft the sound you put into your digital
experiences. Add sound effects or music to your web, Android, and iOS
apps and games--sound that can react to changing environments or user
input dynamically (at least in the native apps). You can do all this
with Pure Data, a visual programming language for digital sound
processing. Programming Sound with Pure Data introduces and explores
Pure Data, building understanding of sound design concepts along the
way. You'll start by learning Pure Data fundamentals and applying
them, creating realistic sound effects. Then you'll see how to analyze
sound and re-create what you hear in a recorded sample. You'll apply
multiple synthesis methods to sound design problems. You'll finish
with two chapters of real-world projects, one for the web, and one for
an iOS and Android app. You'll design the sound, build the app, and
integrate effects using the libpd library. Whether you've had some
experience with sound synthesis, or are new to sound design, this book
is for you. These techniques are perfect for independent developers,
small shops specializing in apps or games, and developers interested
in exploring musical apps.
This rigorous book is a complete and up-to-date reference for the
Csound system from the perspective of its main developers and power
users. It explains the system, including the basic modes of operation
and its programming language; it explores the many ways users can
interact with the system, including the latest features; and it
describes key applications such as instrument design, signal
processing, and creative electronic music composition. The Csound
system has been adopted by many educational institutions as part of
their undergraduate and graduate teaching programs, and it is used by
practitioners worldwide. This book is suitable for students,
lecturers, composers, sound designers, programmers, and researchers in
the areas of music, sound, and audio signal processing.
Power Tools for Synthesizer Programming
A Comprehensive Guide to Synthesizer Programming
Instant Audio Processing with Web Audio
Techniques and Applications
Principles and Practices
Programming Sound with Pure Data
**Computers are at the center of almost everything related to audio. Whether for
synthesis in music production, recording in the studio, or mixing in live sound, the**

computer plays an essential part. Audio effects plug-ins and virtual instruments are implemented as software computer code. Music apps are computer programs run on a mobile device. All these tools are created by programming a computer. Hack Audio: An Introduction to Computer Programming and Digital Signal Processing in MATLAB provides an introduction for musicians and audio engineers interested in computer programming. It is intended for a range of readers including those with years of programming experience and those ready to write their first line of code. In the book, computer programming is used to create audio effects using digital signal processing. By the end of the book, readers implement the following effects: signal gain change, digital summing, tremolo, auto-pan, mid/side processing, stereo widening, distortion, echo, filtering, equalization, multi-band processing, vibrato, chorus, flanger, phaser, pitch shifter, auto-wah, convolution and algorithmic reverb, vocoder, transient designer, compressor, expander, and de-esser. Throughout the book, several types of test signals are synthesized, including: sine wave, square wave, sawtooth wave, triangle wave, impulse train, white noise, and pink noise. Common visualizations for signals and audio effects are created including: waveform, characteristic curve, goniometer, impulse response, step response, frequency spectrum, and spectrogram. In total, over 200 examples are provided with completed code demonstrations.

Drawing on decades of experience, Beep to Boom: The Development of Advanced Runtime Sound Systems for Games and Extended Reality is a rigorous, comprehensive guide to interactive audio runtime systems. Packed with practical examples and insights, the book explains each component of these complex geometries of sound. Using practical, lowest-common-denominator techniques, Goodwin covers soundfield creation across a range of platforms from phones to VR gaming consoles. Whether creating an audio system from scratch or building on existing frameworks, the book also explains costs, benefits and priorities. In the dynamic simulated world of games and extended reality, interactive audio can now consider every intricacy of real-world sound. This book explains how and why to tame it enjoyably.

Delve into the Broadcom VideoCore GPU used on the Raspberry Pi and master topics such as OpenGL ES and OpenMAX. Along the way, you'll also learn some Dispmanx, OpenVG, and GPGPU programming. The author, Jan Newmarch bumped into a need to do this kind of programming while trying to turn the RPi into a karaoke machine: with the CPU busting its gut rendering MIDI files, there was nothing left for showing images such as karaoke lyrics except for the GPU, and nothing really to tell him how to do it. Raspberry Pi GPU Audio Video Programming scratches his itch and since he had to learn a lot about RPi GPU programming, he might as well share it with you. What started as a side issue turned into a full-blown project of its own; and this stuff is hard. What You'll Learn Use Dispmanx and EGL on Raspberry Pi Work with OpenMAX and its components, state, IL Client Library, * * Buffers, and more on RPi Process images and video on RPi Handle audio on RPi Render OpenMAX to OpenGL on the RPi Play multimedia files on the RPi Use OpenVG for text processing and more Master overlays "br>Who This Book Is For You should be comfortable with C programming and at least some concurrency and thread programming using it. This book is for experienced programmers who are new or learning about Raspberry Pi.

How To Make A Noise-perhaps the most widely read book about synthesizer

programming-is a comprehensive, practical guide to sound design and synthesizer programming techniques using subtractive (analog) synthesis, frequency modulation synthesis, additive synthesis, wave-sequencing, and sample-based synthesis. The book looks at programming using examples from six software synthesizers: Cameleon 5000 from Camel Audio, Rhino 2 from BigTick, Surge from Vember Audio, Vanguard from reFX, Wusikstation from Wusik dot com, and Z3TA+ from Cakewalk. Simon Cann is a musician and writer based in London. He is author of Cakewalk Synthesizers: From Presets to Power User, Building a Successful 21st Century Music Career, and Sample This!! (with Klaus P Rausch). You can contact Simon through his website: www.noisesculpture.com.
Programming with MicroPython
A Hands-on Guide to Audio Programming for Mac and IOS
Computer Music Instruments II
Linux Sound Programming
The SuperCollider Book
Automatic Sound Synthesizer Programming

Are you ready to take your programming to the next level? Are you already using Arduino and like the way it works? Do you want to be able to build more powerful projects and applications? Arduino is one of those programming languages that offers you much more than you expected at every stage you reach. It provides effective tools that really work and can have you doing complex tasks quickly with its hands-on approach. Now, with Arduino Programming: The Ultimate Intermediate Guide to Learn Arduino Programming Step by Step, you can take the next step on your Arduino journey and increase your knowledge and skills further, with chapters on: • Getting the most from Arduino • Functions, calculations and tables • Linking the physical to the virtual • Coupling and multiplexing • How to digitalize sound • Advanced techniques • Networking • And more… With its combination of theory and practical advice, Arduino Programming is the stand-out book when it comes to building on your basic understanding of this fantastic programming resource. Get a copy today and enhance your knowledge while building ever more complex applications for your computer!

The essential reference to SuperCollider, a powerful, flexible, open-source, cross-platform audio programming language. SuperCollider is one of the most important domain-specific audio programming languages, with potential applications that include real-time interaction, installations, electroacoustic pieces, generative music, and audiovisuals. The SuperCollider Book is the essential reference to this powerful and flexible language, offering students and professionals a collection of tutorials, essays, and projects. With contributions from top academics, artists, and technologists that cover topics at levels from the introductory to the specialized, it will be a valuable sourcebook both for beginners and for advanced users. SuperCollider, first developed by James McCartney, is an accessible blend of Smalltalk, C, and further ideas from a number of programming languages. Free, open-source, cross-platform, and with a diverse and supportive developer community, it is often the first programming language sound artists and computer musicians learn. The

SuperCollider Book is the long-awaited guide to the design, syntax, and use of the SuperCollider language. The first chapters offer an introduction to the basics, including a friendly tutorial for absolute beginners, providing the reader with skills that can serve as a foundation for further learning. Later chapters cover more advanced topics and particular topics in computer music, including programming, sonification, spatialization, microsound, GUIs, machine listening, alternative tunings, and non-real-time synthesis; practical applications and philosophical insights from the composer's and artist's perspectives; and "under the hood," developer's-eye views of SuperCollider's inner workings. A Web site accompanying the book offers code, links to the application itself and its source code, and a variety of third-party extras, extensions, libraries, and examples. Welcome to the second volume of Game Audio Programming: Principles and Practices – the first series of its kind dedicated to the art of game audio programming! This volume features more than 20 chapters containing advanced techniques from some of the top game audio programmers and sound designers in the industry. This book continues the tradition of collecting more knowledge and wisdom about game audio programming than any other volume in history. Both audio programming beginners and seasoned veterans will find content in this book that is valuable, with topics ranging from extreme low-level mixing to high-level game integration. Each chapter contains techniques that were used in games that have shipped, and there is a plethora of code samples and diagrams. There are chapters on threading, DSP implementation, advanced middleware techniques in FMOD Studio and Audiokinetic Wwise, ambiences, mixing, music, and more. This book has something for everyone who is programming audio for a game: programmers new to the art of audio programming, experienced audio programmers, and those souls who just got assigned the audio code. This book is for you!
Filled with practical, step-by-step instructions and clear explanations for the most important and useful tasks. A concise, recipe-based approach to use Web Audio's automation functionality to produce interesting audio effects such as audio stitching and ducking.This book is designed for developers with some HTML and JavaScript programming experience who are seeking to learn about Web Audio. Experience with AJAX and web server installation/configuration is a plus but is not a necessity in order to follow the content of the book.
Synthesis techniques and programming
Computer Sound Design
Make Your Apps Come Alive with Dynamic Audio
What Every Programmer Needs to Know about Cognition
Hack Audio
The Programmer's Brain

Describes the Core Audio framework, covering such topics as recording, playback, format conversion, MIDI conectivity, and audio units.
Welcome to the third volume of Game Audio Programming: Principles and Practices—the first series of its kind dedicated to the art and science of

game audio programming. This volume contains 14 chapters from some of the top game audio programmers and sound designers in the industry. Topics range across game genres (ARPG, RTS, FPS, etc.), and from low-level topics such as DSP to high-level topics like using influence maps for audio. The techniques in this book are targeted at game audio programmers of all abilities, from newbies who are just getting into audio programming to seasoned veterans. All of the principles and practices in this book have been used in real shipping games, so they are all very practical and immediately applicable. There are chapters about split-screen audio, dynamic music improvisation, dynamic mixing, ambiences, DSPs, and more. This book continues the tradition of collecting modern, up-to-date knowledge and wisdom about game audio programming. So, whether you've been a game audio programmer for one year or ten years, or even if you've just been assigned the task and are trying to figure out what it's all about, this book is for you! Key Features Cutting-edge advanced game audio programming concepts with examples from real game audio engines Includes both high-level and low-level topics Practical code examples, math, and diagrams that you can apply directly to your game audio engine. Guy Somberg has been programming audio engines for his entire career. From humble beginnings writing a low-level audio mixer for slot machines, he quickly transitioned to writing game audio engines for all manner of games. He has written audio engines that shipped AAA games like Hellgate: London, Bioshock 2, The Sims 4, and Torchlight 3, as well as smaller titles like Minion Master, Tales from the Borderlands, and Game of Thrones. Guy has also given several talks at the Game Developer Conference, the Audio Developer Conference, and CppCon. When he's not programming or writing game audio programming books, he can be found at home reading, playing video games, and playing the flute.

Mozart on your PC? You bet! Give your computer personality with this dynamic book/disk set - add speech and music to your C applications for DOS and Windows. Learn to write programs, applications, demos, and utilities that speak in your own voice. Then, catch a passerby's attention with text and graphics that move in synchronization with the playback of sound. With this book you'll be able to write programs that support .WAV files, .VOC files, .SND files, .TS files, .S files, and MIDI files. To incorporate sound into your applications you need: the Microsoft or Borland C compiler, the Windows operating system for Windows programs, the MS-DOS operating system for DOS programs. All the program examples are explained step by step in easy-to-follow language.

It's an exciting time to get involved with MicroPython, the re-implementation of Python 3 for microcontrollers and embedded systems. This practical guide delivers the knowledge you need to roll up your sleeves and create exceptional embedded projects with this lean and efficient programming language. If you're familiar with Python as a programmer, educator, or maker, you're ready to learn—and have fun along the way. Author Nicholas Tollervey takes you on a journey from first steps to advanced projects. You'll

explore the types of devices that run MicroPython, and examine how the language uses and interacts with hardware to process input, connect to the outside world, communicate wirelessly, make sounds and music, and drive robotics projects. Work with MicroPython on four typical devices: PyBoard, the micro:bit, Adafruit's Circuit Playground Express, and ESP8266/ESP32 boards Explore a framework that helps you generate, evaluate, and evolve embedded projects that solve real problems Dive into practical MicroPython examples: visual feedback, input and sensing, GPIO, networking, sound and music, and robotics Learn how idiomatic MicroPython helps you express a lot with the minimum of resources Take the next step by getting involved with the Python community

Fundamentals of Audio and Video Programming for Games
An Introduction to Computer Music, Java Programming and the JMusic Library
Commodore 64 Graphics & Sound Programming
Creating music with ChucK
Audio Programming for Interactive Games
Beginning Game Audio Programming

This book is divided into two parts. The chapters in Part I offer a comprehensive introduction to the C language and to fundamental programming concepts, followed by an explanation of realtime audio programming, including audio synthesis and processing. The chapters in Part II demonstrate how the object-oriented programming paradigm is useful in the modelling of computer music instruments, each chapter shows a set of instrument components that are paired with key C++ programming concepts. Ultimately the author discusses the development of a fully-fledged object-oriented library. Together with its companion volume, Computer Music Instruments: Foundations, Design and Development, this book provides a comprehensive treatment of computational instruments for sound and music. It is suitable for advanced undergraduate and postgraduate students in music and signal processing, and for practitioners and researchers. Some understanding of acoustics and electronic music would be helpful to understand some applications, but it's not strictly necessary to have prior knowledge of audio DSP or programming, while C / C++ programmers with no experience of audio may be able to start reading the chapters that deal with sound and music computing.
Modern Radio and Audio Production: Programming and Performance

Commodore 64 Music Book
Game Audio Programming 3: Principles and Practices