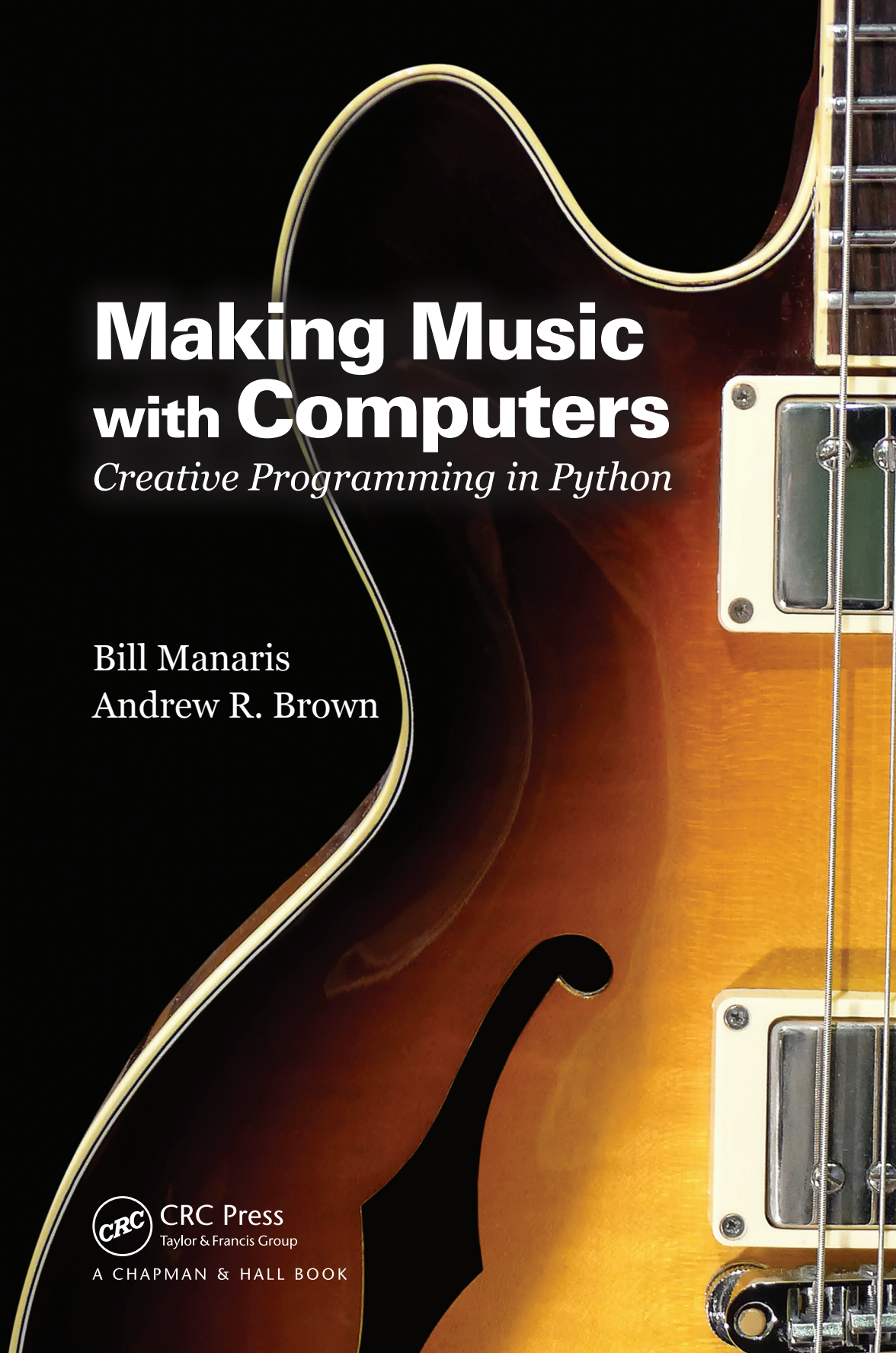

Making Music with Computers

Creative Programming in Python

Bill Manaris
Andrew R. Brown

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Making Music with Computers

Creative Programming in Python

CHAPMAN & HALL/CRC TEXTBOOKS IN COMPUTING

Series Editors

John Impagliazzo
Professor Emeritus, Hofstra University

Andrew McGettrick
Department of Computer and Information Sciences
University of Strathclyde

Aims and Scope

This series covers traditional areas of computing, as well as related technical areas, such as software engineering, artificial intelligence, computer engineering, information systems, and information technology. The series will accommodate textbooks for undergraduate and graduate students, generally adhering to worldwide curriculum standards from professional societies. The editors wish to encourage new and imaginative ideas and proposals, and are keen to help and encourage new authors. The editors welcome proposals that: provide groundbreaking and imaginative perspectives on aspects of computing; present topics in a new and exciting context; open up opportunities for emerging areas, such as multi-media, security, and mobile systems; capture new developments and applications in emerging fields of computing; and address topics that provide support for computing, such as mathematics, statistics, life and physical sciences, and business.

Published Titles

Paul Anderson, Web 2.0 and Beyond: Principles and Technologies

Henrik Bærbak Christensen, Flexible, Reliable Software: Using Patterns and Agile Development

John S. Conery, Explorations in Computing: An Introduction to Computer Science

Ted Herman, A Functional Start to Computing with Python

Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph, Foundations of Semantic Web Technologies

Mark J. Johnson, A Concise Introduction to Data Structures using Java Uvais Qidwai and C.H. Chen, Digital Image Processing: An Algorithmic Approach with MATLAB®

Mark J. Johnson, A Concise Introduction to Programming in Python

Lisa C. Kaczmarczyk, Computers and Society: Computing for Good

Mark C. Lewis, Introduction to the Art of Programming Using Scala

Bill Manaris and Andrew R. Brown, Making Music with Computers: Creative Programming in Python

Henry M. Walker, The Tao of Computing, Second Edition

CHAPMAN & HALL/CRC
TEXTBOOKS IN COMPUTING

Making Music with Computers

Creative Programming in Python

Bill Manaris
College of Charleston
South Carolina, USA

Andrew R. Brown
Queensland University of Technology
Keperra, Australia

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Version Date: 20140402

International Standard Book Number-13: 978-1-4822-2221-0 (eBook - PDF)

Contents

Foreword

The human desire to express and communicate has influenced computing almost as long as there have been computers. ENIAC was first turned on in 1947. The first computer music was generated in 1957.

The desire to *say* more with a computer has driven many advances in computer science. Ivan Sutherland invented interactive computer graphics in 1963, and his creation inspired the idea of classes in object-oriented programming. Alan Kay and Adele Goldberg described the computer as human's first *meta-medium*, the first creative medium that could encompass all previous media. Their research group at Xerox's Palo Alto Research Center (PARC) worked in the 1970s to answer the question, "What would a computer used for creative expression look like?" That's what led them to invent the desktop user interface as we know it today. In a real sense, the menus and windows that we use today to access Facebook were invented in order to make the most powerful tool ever for human expression.

Making music on a computer is a natural way to learn more about mathematics, computer science, and music. Bill Manaris and Andrew Brown have created this marvelous book that will engage and inspire you to learn more about the science and art of creating music through computation. They lead us through exploration of fascinating questions. How does music draw on both mathematical patterns and randomness? How did Bach use algorithms to generate canons? How can we turn data about proteins and planets into music? What kinds of new interfaces can you create to make it easier for you and others to make music?

Bill and Andrew offer an accessible path into a wonderful world that is both as modern as your new laptop and as ancient as Plato. In that world of music and mathematics, they constructed a sandbox of computational

tools. They encourage you to create, to compose music, and to play with patterns and data. They invite you to continue in the traditions of Ivan Sutherland and Alan Kay to use computing to explore powerful and creative ideas.

Mark Guzdial
Georgia Institute of Technology
July 2013

Preface

The book in your hands is the result of more than a decade of independent and collaborative effort by the two authors and their computer music associates. Combining computers and music has a long and fruitful heritage. The ideas which underpin the connection between calculating and composing date back centuries. In the 21st century, computers and music are more closely aligned than ever before. In particular, computers have become indispensable in music making, distribution, performance, and consumption.

This book introduces important concepts and skills necessary to make music with computers. It interweaves computing pedagogy with musical concepts and creative activities. It does this while maintaining a natural, steady increase in computational skills that are motivated by creative musical contexts.

This book is intended primarily for introductory computer science courses and for courses in the intersection of computing and the arts. However, it is naturally suited for self-study. It assumes little musical and programming experience; it introduces topics and concepts as they arise through motivating, and hopefully inspiring examples.

CREATIVE PROGRAMMING

"Making Music with Computers" is an introduction to creative software development in the Python programming language. It uses music-making as a vehicle to introduce computer programming and computational thinking to non-traditional audiences. This book helps computer science educators teach students how to synthesize the creativity and design of the arts with the mathematical rigor and formality of computer science.

Initially inspired by Randy Pausch's "head-fake" approach*, we utilize exciting and innovative music-creation activities to ultimately teach

* See Randy Pausch's "Last Lecture" (readily available online).

introductory computer science concepts. Our goal is to keep this "game" going throughout the book, just long enough so that the students learn to express themselves algorithmically.

The book covers all concepts found in a traditional "Intro to Computer Programming" (CS1) course. These concepts include data types, variables, assignment, arithmetic operators, input/output, algorithms, selection (`if` statements), relational operators, logical operators, iteration (loops), lists (arrays), functions, modularization (functions), classes (object-oriented programming). Additionally, the book covers graphical user interfaces (GUIs), event-driven programming, big data, MIDI programming, client-server programming (via OSC messages), recursion, fractals, and complex system dynamics (boids).

TARGET AUDIENCE

This book addresses two trends in computing education: (1) the growing use of the Python language for teaching introductory programming, and (2) the increasing infusion of computational thinking into liberal arts courses, especially interdisciplinary offerings in computing and the arts. It does so by presenting computer music topics in an accessible way for our two main target audiences:

- First- and second-year university students, as well as advanced high school students, who are interested in computer music and wish to learn computer programming in a creative context; and
- Musicians of all levels and backgrounds who wish to expand their creative horizons by modeling musical processes through computer programming, and by applying these processes to create novel and intriguing musical material for composition and live performance.

NAVIGATING THE BOOK

The book may be navigated using one of three narratives, *objects first, procedures first,* or *à la carte:*

- **Objects first** (chapters 1–3, followed by chapters 8–11, with just-in-time introduction of `for` loops, functions, and `if` statements). This approach works well with inexperienced students, as it is creatively rich. It includes building graphical user interfaces (GUIs) and interactive musical instruments, and thus motivates hard-to-grasp

programming concepts (such as loops, functions, and if statements). To quote one of our students, "students want to do the work, because it is fun." In particular, chapters 1-3 introduce object-based programming (using Notes, Phrases, Parts, and Scores). Chapter 8 introduces GUI objects, event-driven programming, and important human-computer interaction (HCI) ideas, such as how to develop usable interfaces through paper prototyping, usability testing, and iterative refinement. Chapter 9 introduces MIDI and OSC (open sound control) input/output objects, thus enabling programs to connect to traditional musical instruments (e.g., guitars, pianos, etc.)* and physical controllers (e.g., MIDI control surfaces, smartphones, touch-sensitive tablets, etc.). Finally, chapters 10 and 11 introduce Python classes, music from math equations, the harmonograph (a way to visualize and sonify complex, yet beautiful repetitive patterns found in nature, such as planetary orbits), animation, fractals, the golden ratio, recursion, Zipf's law, and chaotic systems (boids). This material is full of musical and other creative possibilities.

- **Procedures first** (chapters 1–11). This is a traditional narrative, which interweaves computational and musical concepts incrementally, from beginning to intermediate level of expertise. In addition to the above topics, it includes randomness and creativity, data sonification, image processing, musical canons (musical puzzles, of which JS Bach was a master), minimalism, and stochastic music, to name a few. It also provides a thorough introduction to programming in Python, including data types, variables, assignment, arithmetic operators, input/output, algorithms, selection (if statements), iteration (loops), lists (arrays), file input/output, modularization (functions), event-driven programming (callback functions), and object-oriented programming (classes).

- **À la carte** (explore topics as desired/needed). This approach is best suited for self-learners, and for musicians (and programmers) who already know some Python (and music), and who wish to explore techniques to enhance their potential for creative expression. If you belong in this group, then the table of contents is your best friend. Study it carefully, looking for items that seem attractive, and go from

* Any MIDI-enabled instrument will work, opening the door for some powerful creative possibilities, for building hybrid computer music instruments.

there. If you are lacking some Python background (to fully appreciate the provided examples, without having to read the whole book up to that point), either you can look up Python topics in the book index, or search the Internet (the latter being full of great Python reference material). Please enjoy weaving your own path through this material - we certainly did!

All three narratives are supported by the website provided at http://jythonMusic.org. There you will find additional resources (including more code examples) to enhance your creative exploration and learning. Enjoy!

PEDAGOGY

From the point of view of pedagogy, our primary audience is educators interested in teaching computing and computational thinking in a media-rich context, in conjunction with guidelines such as the CS Principles and Big Ideas.* In particular, this book supports teaching of the 7 CS Big Ideas:

- **Big Idea I**—Creativity. Computing is a creative human activity that engenders innovation and promotes exploration (whole book, and in particular chapters 1, 6, 7, 8, 9, 10, and 11).
- **Big Idea II**—Abstraction. Abstraction reduces information and detail to focus on concepts relevant to understanding and solving problems (whole book, and in particular chapters 2, 3, 7, and 11).
- **Big Idea III**—Data. Data and information facilitate the creation of knowledge (chapters 3, 4, 7, 10, and 11).
- **Big Idea IV**—Algorithms. Algorithms are tools for developing and expressing solutions to computational problems (chapters 4, 5, 6, 7, 10, and 11).
- **Big Idea V—**Programming. Programming is a creative process that produces computational artifacts (chapters 3, 4, 5, 6, 7, 8, 9, 10, and 11).
- **Big Idea VI**—Internet. Digital devices, systems, and the networks that interconnect them enable and foster computational approaches to solving problems (chapter 9).

* The College Board, "Computer Science: Principles, Big Ideas and Key Concepts", 2012.

- **Big Idea VII**—Impact. Computing enables innovation in other fields including mathematics, science, humanities, and arts, among others (chapters 1, 8, 9, 10, and 11).

SOFTWARE LIBRARIES

The book comes with a Jython environment and a collection of software examples and libraries, including music, image, graphical user interface, MIDI, audio, and Open Sound Control. The music library is an extension of the jMusic library and incorporates other cross-platform programming tools. This software is available for download on the website associated with this book, http://jythonMusic.org.

We hope that this book will enhance the educational experience of students in entry-level courses in computing and computational thinking. We also hope that it may serve as a reference and text for computer music courses, such as those offered by music technology programs. Finally, we hope this book may serve as a reference and tutorial resource for digital music enthusiasts who wish to expand their creative horizons and learn how to write music software and create algorithmic music compositions.

The Authors

Bill Manaris is a computer science educator, researcher, and musician. He holds a Ph.D. in computer science and is professor of computer science and director of computing in the arts program at the College of Charleston, South Carolina. He has studied music theory, and classical and jazz guitar, and performs live occasionally. He has been active in curriculum development in human–computer interaction, artificial intelligence, and computing in the arts. His teaching experience spans 30 years. His research interests include statistical, connectionist, and evolutionary techniques for modeling human aesthetics and creativity in music and art. He has developed several systems for computer-aided music analysis, composition, and performance, including NEvMuse, Armonique, and Monterey Mirror. For more information visit http://www.cs.cofc.edu/~manaris.

Andrew R. Brown is an educator, musician, digital artist, and computer programmer. He holds a Ph.D. in music and is professor of digital arts at Griffith University, in Brisbane, Australia, where his work explores the aesthetics of process and regularly involves programming software as part of the creative process. In addition to a history of computer-assisted composition and audio-visual installations, Andrew has in recent years focused on real-time artworks using generative processes and musical live-coding. The latter is a practice where the software to generate a work is written as part of the performance. He has been invited to perform live coding in many international venues. His digital media artworks have been shown in galleries across Australia and in China. For more information visit http://andrewrbrown.net.au.

Acknowledgments

We would like to thank John Impagliazzo for recommending and encouraging the writing of this book; Randi Cohen for patiently working through the various stages and ever-extending timeframes that this book required; and our various collaborators and students who assisted in various invaluable ways to complete this project (via code development, API design and review, and testing) including Dana Hughes (GUI library), David Johnson (OSC and MIDI libraries), Kenneth Hanson, J.R. Armstrong, Thomas Zalonis, Patrick Roos, Luca Pellicore, Timothy Hirzel, Brian Muller, William Daugherty, Dallas Vaughan, Christopher Wagner, Semmy Purewal, and Valerie Sessions (Zipf library and metrics). We are particularly indebted to Mark Guzdial for opening the door to introducing computer science concepts via Media Computation. Also of particular importance to the first author was attending the 15-day Workshop in Algorithmic Music Composition (WACM), in 2010, and his interactions with David Cope, Peter Elsea, Paul Nauert, and Daniel Brown, as well as the various workshop participants. The second author is particularly indebted to those who contributed to the development of the jMusic library and tutorials upon which this book builds. These include Andrew Sorensen, Rene Wooller, Tim Opie, Andrew Troedson and Adam Kirby.

We owe a debt of gratitude to the reviewers of this book, especially William Greene and Maximos Kaliakatsos for reading every chapter word-for-word and every code example, and for offering numerous suggestions and improvements. Additional comments and support were provided by David Cope, Daniel Brown, Yiorgos Vassilandonakis, Walter Pharr, and Blake Stevens.

Finally, we want to thank our families for their patience and support as we worked the long nights, weekends, and holidays this book required.

This book has been partially supported through funding by the US National Science Foundation (including DUE-1044861, IIS-0736480, IIS-0849499 and IIS-1049554).

CHAPTER 1

Introduction and History

Topics: Pythagoras (music, nature, and number), the Antikythera mechanism, Kepler' s harmony of the world, cymatics, fractals, electronic music, computers and programming, the computer as a musical instrument, running Python programs.

1.1 OVERVIEW

This chapter provides a quick tour of some of the major technological landmarks in Western music history and computer science. When we think of computer music, we usually imagine electronic technologies, particularly the synthesizer, computer, and sound recording devices. These devices are products of the information age in which we live. This age focuses on computational thinking, that is, using computers in creative ways to manipulate data and perform various tasks, usually involving some form of programming. The introduction of computers and, in particular, computer programming has also expanded the sonic and structural boundaries of music composition and performance.

In the 20th century, the fundamental education of an individual consisted of the three R's—**r**eading, w**r**iting, and a**r**ithmetic. In the 21st century, with the proliferation of computing devices, this list now consists of four R's, that is, **r**eading, w**r**iting, a**r**ithmetic, and p**r**ogramming.

Once computer programming is mastered, new vistas of creative expression become available. This new expressive capability is not confined only to computer music—it is available in every area of the arts as well as the sciences. Accordingly, the programming skills you will acquire in this book are not specific only to making music. They may be applied to creative endeavors in all areas of human knowledge and expression.

1.2 CONNECTING MUSIC, NATURE, AND NUMBER

The development of music and mathematics is connected to humanity's early observations of nature, and attempts to explain and formulate aspects

of the human experience. The ancient Babylonians, Egyptians, and Greeks investigated the origin of sound and resonance, and developed the notion of musical scale in terms of integer ratios. To them nature was a harmonious artifact, in which humans found themselves exploring and creating. Mathematics was created around that time, and in its early phases, it was intricately connected to nature and music. Even more recent concepts of the golden ratio, Fibonacci numbers, Zipf's law, cymatics, and fractals are all based on this ancient theme. In this book, we let this ancient theme guide us, as we interweave music, number, and computer programming.

1.2.1 Pythagoras—Harmonic Series

The ancient Babylonians, Egyptians, and Greeks were fascinated with the technological nature of music—perhaps even more than we are today. For instance, Pythagoras (c. 570–c. 495 BCE) discovered that musical pitch intervals could be described by numbers. He and his students are credited with the discovery of mathematics, a term which they coined. Pythagoras left Greece at a young age to be educated in Egypt. There he associated himself with Egyptian priests, who at the time studied astronomy, geometry, and religion (all at once, without the divisions we have today). Pythagoras spent close to 20 years in Egypt, but then was captured during a war and was transferred as a slave to Babylon (an area now part of Iraq). There, through his knowledge and intellect, he gained access again to the educated elite and continued his studies in astronomy, religion, geometry, and music.

Pythagoras's contributions helped shape the ideas of subsequent philosophers, mathematicians, and scientists, including Plato, Aristotle, and many more. Aristotle tells us the Pythagoreans discovered that musical harmony can be explained by numbers; they took up mathematics, and "thought its principles were the principles of all things. Since, of these principles, numbers are by nature the first, and in numbers they seemed to see many resemblances to the things that exist and come into being" (Aristotle 1992, pp. 70–71). This observation suggests that everything we experience through our senses can be described (e.g., measured and represented) by numbers, in some way or another, and then it can be turned into music. For instance, consider the music stored on your digital music player (inside the machine, this music is represented by numbers).*

* This applies to all information (e.g., text, images) stored on a computer, or the Internet—the term *digital* refers to representing information using numbers (i.e., converting information to data).

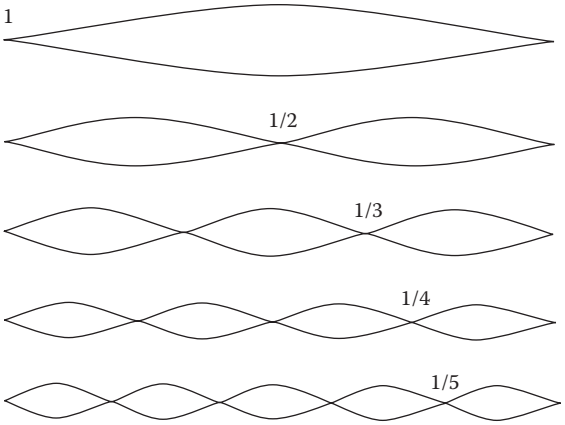

FIGURE 1.1 String resonating at integer ratios.

Also, consider the concept of sonification, that is, the conversion of arbitrary data to sounds, so that they may be perceived more easily.* In other words, music and numbers are interchangeable.

One of the major discoveries contributed by the Pythagoreans, which helped shape the nature of music theory many centuries later, is the observation that strings resonate in simple ratios. In particular, they observed that strings exhibit harmonic proportions—they vibrate at integer ratios of their length, that is, 1/1, 1/2, 1/3, 1/4, 1/5, etc. (see Figure 1.1). The instruments of the era, the *lyra* and the *kithara* (the latter etymologically related to the modern guitar), were most probably used in their experimentations.

This was a major discovery, since it demonstrated that integers emerge from the natural properties (or geometry) of a string. Accordingly, the 19th century mathematician Leopold Kronecker said, "God made the integers; all else is the work of man" (Bell 1986: 477). He argued that arithmetic and mathematical analysis must be founded on integers. The Pythagorean discovery is even more profound when considering the implications of string theory in physics, which states that the universe consists of subatomic particles resembling one-dimensional resonating strings. These ideas are related to the fields of *cymatics* and *fractal geometry* (see the following sections).

Finally, Pythagoras and his students worked on a theory of numbers and explored the *harmony of the spheres*. The harmony of the spheres (or *musica universalis*—music of the spheres) is the philosophical belief that

* See Chapter 7 for a more in-depth discussion of sonification.

the planets and stars moved according to mathematical equations. Since numbers are connected to musical notes, the orderly movement of planets was said to create an astronomical symphony. According to different religious/philosophical traditions, this music could be heard only by the most enlightened individuals. However, with the advent of the modern computer (and the knowledge you will accumulate in this book), this music is now accessible to everyone.

One of the major discoveries of this era (first described by Plato, in Timaeus, and then by Euclid, in his Elements) was the golden ratio (or golden) mean. This special proportion, which humans find aesthetically very pleasing, is found in natural or human-made artifacts (Beer 2008; Calter 2008, pp. 46–57; Hemenway 2005, pp. 91–132; Livio 2002). It is also found in the human body (e.g., the bones of our hands, the cochlea in our ears, etc.). The golden ratio reflects a place of balance in the structural interplay of opposites.

1.2.2 The Antikythera Mechanism—The First Known Computer

Ancient astronomical models were well established. They were used to construct the first computing machines approximately 2,100 years ago (Vallianatos 2012). Of these early computational machines, only one survives, in the National Archeological Museum of Greece, in Athens. Interestingly, these machines would have been unknown to us had it not been for the early 20th century discovery of fragments of a working model on a 2,000-old shipwreck near the island of Antikythera (see Figure 1.2).

The Antikythera mechanism uses the same design principles (i.e., employing gear ratios to implement mathematical relations) as the much later (19th century) Difference and Analytical Engines designed by Charles Babbage and Lady Ada Lovelace (see Figure 1.3). The connection between these machines and modern computers is indisputable.

1.2.3 Johannes Kepler—Harmony of the World

The Pythagorean ideas and theories inspired many in the centuries that followed, including Johannes Kepler. In 1619 Kepler wrote his seminal work *Harmonices Mundi* (*Harmony of the World*). In this book, Kepler describes physical harmonies in planetary motion. His work contributed significantly to the scientific revolution that brought us out of the dark ages.

In this book Kepler presents his third law of planetary motion, that the distance of a planet from the sun is inversely proportional to its speed. Based on this result, he also discusses the harmony found in the motions

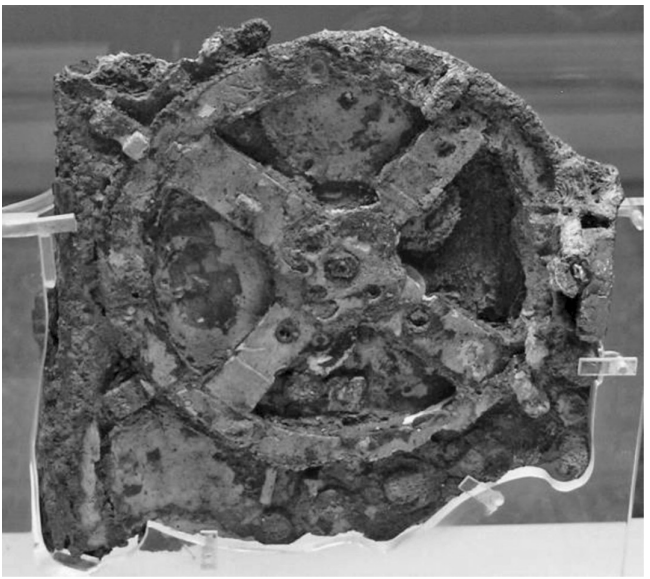

FIGURE 1.2 Fragment from the Antikythera mechanism.

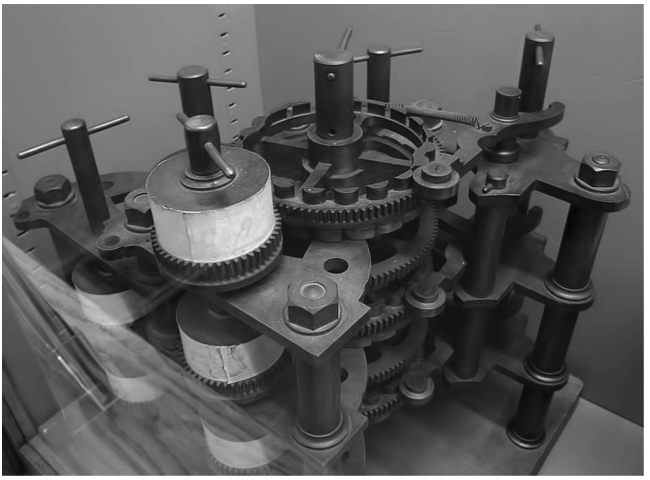

FIGURE 1.3 Part of Babbage's difference engine.

of the planets. In particular, he discovered that the speeds of consecutive planets approximate musical harmonies. The only exceptions are Mars and Jupiter. However, we now know that this is the result of a missing planet, whose mass is found in the asteroid belt between Mars and Jupiter. This belt was discovered 150 years after Kepler's death.

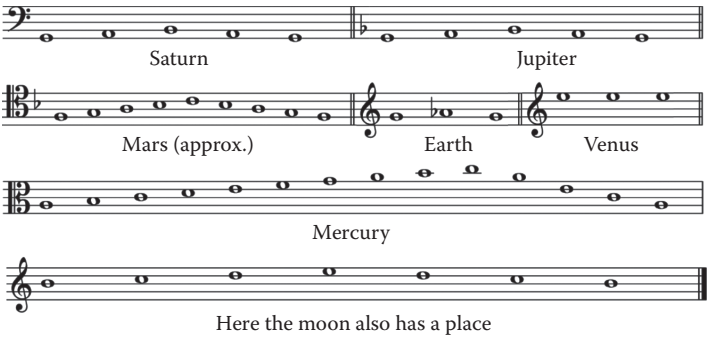

FIGURE 1.4 Kepler's study of musical notes representing the motion of the known planets (capturing changes in speed as planets traverse their elliptical orbits around the sun).

Kepler argued that planets can be thought of as "singing" together in near harmony. This harmony fluctuates as planets slow down and speed up (i.e., each has a minimum and maximum angular speed). Only rarely do planets "sing" in perfect concord.

This kind of *sonification* (i.e., turning data into music) has been applied to many natural and human-made phenomena to generate sounds that are not too foreign to our ears, as might initially be imagined (see Figure 1.4). Later in the book, we explore this idea of sonification, so you too can create your own experiments related to the Pythagorean ideas. Recently, geologist John Rodgers and jazz musician Willie Ruff helped materialize Kepler's *Harmonices Mundi* by sonifying actual orbital data of planets in our solar system. This recording can be easily found on the Internet and is very inspiring to listen to.

1.2.4 Cymatics

Cymatics (from the Greek κύμα, "wave") is the study of visible (visualized) sound and vibration in 1-, 2-, and 3-dimensional artifacts. It was influenced significantly by the work of the 18th century physicist and musician Ernst Chladni, who developed a technique to visualize modes of vibration on mechanical surfaces, known as *Chladni plates* (see Figure 1.5).

When drums or gongs are struck, they vibrate in similar ways. That similar modes of vibration relate to musical pitch, rhythmic subdivisions, and sound timbre is interesting and suggests that many aspects of music and sound can be described computationally and controlled through software. Cymatics is an inspiring young field of exploration—for more

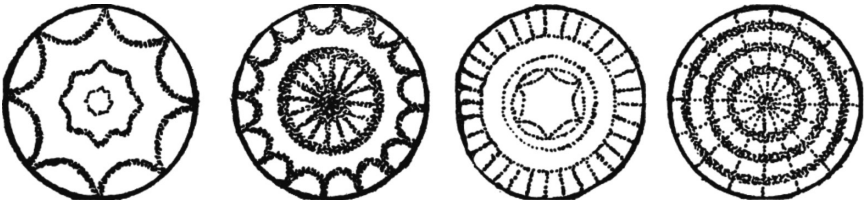

FIGURE 1.5 Chladni plates, vintage engraving. Old engraved illustration of Chladni plates isolated on a white background. (From Charton, É. and Cazeaux, E., eds. (1874), Magasin Pittoresque.)

information, see Evan Grant's TED Talk, which demonstrates the science and art of cymatics, through beautiful visualizations of soundwaves (Grant 2009).

1.2.5 Fractals

In the spirit of Pythagoras, mathematical descriptions for musical organization continue to be pursued. The hierarchical nature of music has led many to consider fractal geometry as an interesting candidate for such descriptions. Fractals are *self-similar* objects (or phenomena), that is, objects consisting of multiple parts, with the property that the smaller parts are the same shape as the larger parts, but of a smaller size. Fractals were developed by Benoit Mandelbrot to study harmonic proportions in nature (Mandelbrot 1982). Figure 1.6 displays a fractal tree (also known as a Golden Tree, since it incorporates golden ratio proportions). This fractal is constructed by dividing a line into two branches, each rotated by 60° (clockwise and counter-clockwise), with a length reduction factor equal to the golden ratio (0.61803399…). These smaller lines, again, are each subdivided into two lines following the same procedure. This repetition/subdivision continues on and on (theoretically) to infinity. Interestingly, similar patterns appear extensively in nature (as they maximize the amount of matter that can fit in a limited space, that is, touching but not overlapping). Such artifacts are very easy to construct using a computer.

The Harvard linguist George Kingsley Zipf (1902–1950) was a great influence on the development of fractals. In his seminal book, *Human Behavior and the Principle of Least Effort*, Zipf reports the amazing observation that word proportions in textbooks, as well as notes in musical pieces (among other phenomena), follow the same harmonic proportions (i.e., 1/1, 1/2, 1/3, 1/4, 1/5, etc.) first discovered by Pythagoreans on strings. Zipf proportions have been discovered in a wide range of natural and

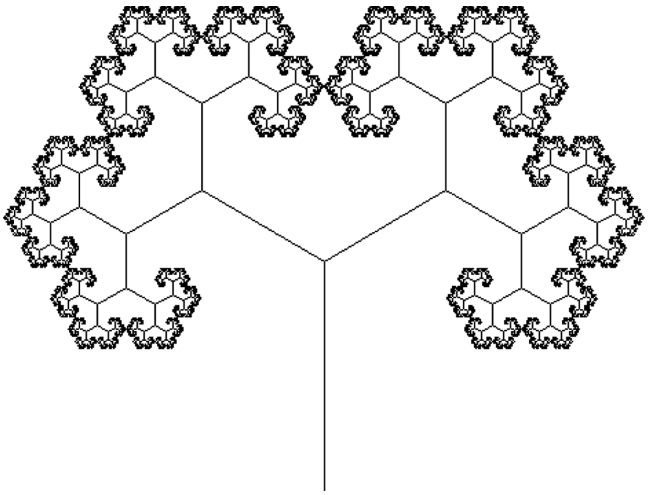

FIGURE 1.6 Fractal tree.

human-made phenomena, including music, city sizes, salaries, subroutine calls, earthquake magnitudes, thicknesses of sediment depositions, extinctions of species, traffic jams, and visits to websites, among many others.

Zipf proportions are also known as *pink-noise*, *harmonic*, and *1/f* proportions and can be considered to be measures of variety or interest. At one extreme is a random probability of occurrence (i.e., chaos or white noise, such as radio static) where events are unpredictable and seemingly unorganized. In the mid range lie fractal and brown-noise that have some discernable organization. At the other extreme are very monotonous phenomenon (aka black-noise proportions), such as a musical piece consisting mostly of one note. In physics, white-noise, pink-noise, brown-noise, and black-noise proportions are known as *power laws*. Psychologists have shown that people prefer music, and other experiences, that have a balance of predictability and surprise, and so having a computable measure of this likelihood can be useful in computer music making.

Many interesting attempts have been made to generate music from fractal artifacts. Conceptually, the process is relatively straightforward—it involves converting aspects of a fractal object to aspects of a musical artifact. For instance, the placement and size of a line in Figure 1.6 could be converted to the pitch and duration of a note. As the fractal object is being visually generated through a computer program, that same program could output the corresponding musical notes to a MIDI file, thus

generating a fractal music piece. The process of mapping visual elements to audio elements is called *sonification*. Sonification is an art in itself, as there are many possible ways of converting between visual elements and audio elements. (For instance, consider how you might sonify Figure 1.6.) The trick is to identify which visual elements to select, and how to map them to audio, so as to generate the most aesthetically pleasing (or scientifically interesting) audio artifacts. Sonification and fractals are explored later in the book.

1.3 COMPUTER MUSIC HISTORY

Throughout human history, technologies have consistently influenced our societal development, with periods of accelerated influence occurring at times such as the Renaissance, the Industrial Revolution, and the Information Age. This is paralleled by a relatively similar pattern of music technology developments. The earliest harps, horns, and drums are clearly technologies and their development and usage relied on new technologies of their day, very similar to the way computers are applied to music production in our age.

Landmarks in the history of music technologies include the use of written notation from around mid 9th century CE, the development of polyphony in the centuries that followed, and organ building improvements and equal temperament in the Middle Ages.

The Renaissance and Baroque periods saw an obsession with music of the spheres resulting from the newly developed field of astronomy (see above), and a peaking of craftsmanship in the violins of Stradivarius and in compositional technique in the fugues of Bach. The study of alchemy led to 19th century chemistry and physics, which provided new metals and efficient methods to improve instrument fabrication. This surge in instrument development went hand in hand with increases in orchestra size. Also, industrialization was a common underscoring theme in music, such as Wagner's "Der Ring des Nibelungen." Early 20th century landmarks include the automation of music via the player piano, the technological abstractions of electronic and recorded sound in the music of Schaeffer and Stockhausen, and parallel abstractions in the musical structures and notations of Debussy, Stravinsky, Schoenberg, Xenakis, Cage, and others.

This history is continuous in its highlighting of human curiosity and creativity. However, the developments in knowledge and technology are not deterministic and did not follow a simple evolutionary path (i.e., a path of increasing complexity). For example, the interests of Pythagoras

resurfaced and inspired (more than one thousand years later) Kepler's explorations of musical patterns in astrological movements and Fourier's investigations into sonic spectra in the later 18th century. In between these investigations were centuries of explorations that followed other technological paths. The path of technological development is in no way straight or predictable in advance, even if such developments appear as a logical sequence with hindsight.

1.3.1 Automated Music

One of the characteristics of the computer as a music machine is that it can be automated by programming. Automatic instruments have existed for a long time, probably since antiquity. One possibility is the *hydraulis* which is attributed to Ctesibius of Alexandria (3rd century BCE). The hyrdraulis, about which little is known (due to the loss of ancient knowledge mentioned earlier), used water pressure to drive air through pipes, thus producing sounds (similarly to later ecclesiastical organs). Another possibility is the wind organ developed by Heron also of Alexandria (1st century CE), which was driven by a wind wheel. These and later designs were passed through Byzantine and later Arab scholars to Italy around the Renaissance period.

These designs contributed to later automated instruments, such as the barrel organ of Henry VIII built in 1502. It was manually driven, but the course of the following century led to fully autonomous instruments driven by clockwork mechanisms (similar to the Antikythera mechanism, whose design principles were also passed through Byzantine and later Arab scholars).

In order to increase the repertoire used in automated music machines, an alternative was sought to barrel organs that used replaceable barrels, which were expensive to produce and on which playing time was limited. A solution to both these problems presented itself in the 18th century in the form of the punch card technologies employed by Jacquard weaving looms. Scores were made in the form of holes punched in paper tape or cards. The cards could be strung together to create long sequences. This became a new form of musical notation, which was not efficient for human reading but quite efficient for machine reading. The machine became the interpreter of these machine-specific scores. Such instruments constitute more than an amusement even if their quality of performance was quite low. They enabled musical performances to be captured and transported, to be reproduced on demand, and replayed time and again for closer inspection.

Perhaps the most sophisticated (and certainly the most popular) automated instrument was the *player piano*. Although its development historically paralleled the gramophone, its sonic quality was far superior for quite a while and brought music on demand into many homes in the first half of the 20th century. The availability of automated musical performances in the home changed the role of the audience, affecting (not always detrimentally) concert attendance and the social status of musical performance skills. The player piano, more than electronic recording technologies, was the parent of MIDI sequencing in choosing to capture pitch, duration, and force (*velocity*) for each note. The piano rolls were editable and so "near perfect" performances could be created, and composers were not slow to realize that piano rolls could produce music beyond that humanly performable. In this way the composer first became a nonperforming producer, involved in all the steps from conception to final sounding.

1.3.2 Early Computer Music

The first public performance of computer music was programmed by Geoff Hill and Trevor Pearcey and generated by CSIRAC (Council for Scientific and Industrial Research Automatic Computer) at the Australian Computer Conference in August 1951. At this time, computer music was little more than a computational barrel organ playing popular tunes of the time; however, to do so at that time was no easy task, especially given the fickleness of the valve components, the timing constraints of the memory using mercury delay lines, and awkward punched paper tapes for describing programs. CSIRAC was the first computer in Australia and a machine intended purely for scientific research, so the achievement is a remarkable example of how quickly people turn any technology to musical purposes.

Computer-based music composition had its start in the mid-1950s when Lejaren Hillier and Leonard Isaacson did their first experiments with computer-generated music on the ILLIAC computer at the University of Illinois. They employed both a rule-based system utilizing strict counterpoint and also a probabilistic method based on Markov chains (also employed by Iannis Xenakis around the same time). These procedures were applied variously to pitch and rhythm, resulting in "The ILLIAC Suite," a series of four pieces for string quartet published in 1957.

The recent history of automated music and computers is densely populated with examples based on various theoretical rules from music theory and mathematics. While The ILLIAC Suite used known examples of these, developments in such theories have added to the repertoire of intellectual

technologies applicable to the computer. Among these are the Serial music techniques, the application of music grammars (notably the Generative Theory of Tonal Music by Fred Lerdahl and Ray Jackendoff), sonification of fractals and chaos equations, and connectionist pattern recognition techniques based on work in neuropsychology and artificial intelligence.

Arguably, the most comprehensive of the automated computer music programs is David Cope's Experiments in Music Intelligence (EMI), which performs a feature analysis on a database of a particular composer's works (Cope 2004). Using this analysis, EMI can then compose any number of pieces in that composer's style (e.g., J.S. Bach, Chopin, etc.). The term *style*, here, is a function of many musical aspects, including melody and harmony.

In terms of melody, EMI captures repeated ideas that run through the works in its database, that is, common melodic material that a composer tends to use. Finding such repeated ideas is a complicated task, as the same melodic idea can be presented in a variety of different ways within a single piece: notes can be added to it or removed, it can be sped up or slowed down, and it can be played over different harmonies. This requires sophisticated pattern recognition within a complicated context and is one of the major accomplishments of Cope's research.

In terms of harmony, EMI extracts chords from the works in its database and then constructs its own chord progressions. These progressions do not replicate the ones in the database's works; they are novel. However, they are stylistically similar to (i.e., follow similar construction rules as) the analyzed works. In this way, the composer's "harmonic style" is also replicated in EMI's new compositions.

EMI's database can, actually, be loaded with the works of more than one composer. When this is done, its resulting compositions blend the styles of those composers. The results are sometimes odd, but quite often surprisingly clever and beautiful.

1.3.3 Electronic Music

After Thaddeus Cahill's relatively unsuccessful attempt at creating a massive organ-like device using early American telephone technologies called the *Telharmonium*, one of the first electronic performance instruments was Leon Thérémin's device invented in the 1920s in Moscow. The *Theremin*, as it was known, was played by positioning each hand at a varying distance from two antennae. The location of the hands changed the electromagnetic fields generated by electricity passing through

the antennae, one controlling volume, the other the pitch of a constant and relatively pure tone. The Theremin made quite an impact, with pieces being written for it by Aaron Copeland and Percy Grainger, although the most popularly known example is in the opening of the Beach Boys' hit "Good Vibrations."

The first popular electric keyboard instrument was the Hammond organ, invented in 1935 by Laurens Hammond using electromagnetic components to generate sinusoidal waveforms which could be combined in various combinations using drawbars. The drawbars acted similarly to pipe organ stops, but rather than simply turning on or off oscillators, they controlled their degrees of loudness. The B3 model, first produced in 1936, has become legendary in gospel, jazz, and rock music. It provided a relatively affordable and portable keyboard instrument for music performance, and the timbral variety "synthesized" through drawbar settings gave to keyboard players a taste of customizable timbre that would later be expanded by the synthesizer.

The solid body electric guitar was developed after some initial production of semiacoustic electric models in the 1930s. Following early experiments by Adolf Rickenbacker and Les Paul, the first production models appeared in the early 1950s from the Gibson and Fender companies. The major technical hurdle was the refinement of the pickups to eliminate noise and provide a clear signal, which was solved largely by the development of the twin-coil "humbucking" pickup.

The early development of recording technologies by Thomas Edison was done with mechanical technologies around the turn of the 20th century. It was not until electronic amplifiers became available in the form of vacuum tubes that the minute etchings of the recording process could be played back with any fidelity. Even then the making of recorded cylinders was tedious and specialized. Building on this research, the first commercial magnetic tape recorder was introduced in 1948. The ability to record, not only play back, was the shift necessary to motivate musicians to use this technology creatively.

In Paris in the late 1940s after World War II, Pierre Schaeffer developed a compositional use for the previously reproduction-focused tape recorder. The compositional technique Musique Concrète, as it became known, used recorded sounds of both instrumental and environmental origin, manipulated them through variations in timbre, pitch, duration, and amplitude, then collaged these sounds into a polyphonic musical form.

Tape-based compositional works were produced by Karlheinz Stockhausen in Cologne from the mid-1950s, which he called Elektronische Musik (Electronic Music). As well as treating recorded sounds, Stockhausen and contemporaries such as Edgard Varèse focused on synthesizing new timbres using oscillators, filters, and amplifiers.

The successful commercialization of synthesizers came with the release in 1964 of the Moog synthesizers. The technical breakthrough that made these instruments possible was the use of transistors instead of vacuum tubes, which dramatically reduced the instrument's size and increased the stability of voltage control. One of the more popular early recordings using the Moog synthesizers was Wendy Carols's "Switched-on Bach," which was a notable achievement at the time, but created a legacy of imitative thinking which still haunts synthesizer usage, as more recently reinforced in the *General MIDI* specification. The most popular of Robert Moog's synthesizers was the Mini Moog, one of the first portable all-in-one synthesizers, still highly regarded 50 years after its release (see Figure 1.7).

The use of recording as a compositional and synthesis tool did not change much from the days of musique concrète until the late 1970s, when the development in Australia of the Quasar and M8 digital synthesizers by Tony Furse influenced the commercially successful *Fairlight* CMI developed by Peter Vogel and Kim Ryrie, and at the same time the New England Digital *Synclavier* was developed in New Hampshire by

FIGURE 1.7 The Mini Moog synthesizer.

Sydney Alonso, Jon Appleton, and Cameron Jones. The *Fairlight* and the *Synclavier* introduced sampling technologies (short-duration digital recording) to commercial music making in 1979. Both instruments were also capable of sound synthesis processes and used keyboard controllers for performance, attached to computer systems for storage, display, and editing of waveforms. A version of the *Fairlight* is now available for the Apple iPad, which highlights how much computing power and expense has changed in the last half a century or so.

Digital technologies made their way into synthesizers first as memory banks for presets, most famously in the Sequential Circuits *Prophet V*, and later in the sound synthesis engine itself, notably with the Yamaha DX7 (see Figure 1.8). The release of the DX7 in 1983 coincided with another significant event in electronic music history: the introduction of the Musical Instrument Digital Interface (MIDI) standard.

Developed by Dave Smith of Sequential Circuits, and with input from other major manufactures of the time, notably Roland and Yamaha, the MIDI standard replaced the plethora of interconnecting standards such that equipment from different manufacturers could communicate. MIDI began as a note-based live performance protocol, intellectually indebted to music notation and player-piano technologies. The MIDI standard has expanded over the years to include file formats, sample transfer protocols, the General MIDI standard sound set, a music XML format, and a range of other musical and operational parameters.

The synthesizer, in its keyboard form, has remained quite stable since the 1980s, with some controller extensions modeled on other instruments including guitar, woodwind, and percussion. Research continues into new instrument designs, as it always has, with STEIM in the Netherlands and the HyperInstrument group at MIT's Media Lab contributing significantly during the 1990s, but with developments expanding quite broadly since then. Many of the latest research developments are evident

FIGURE 1.8 The Yamaha DX7 synthesizer.

in the proceedings of the annual New Interfaces for Musical Expression (NIME) conference.

Along with advances in MIDI and the digital synthesizers, the 1980s also saw an accelerating increase in personal computer ownership and with it the expansion of music software. Most significant from a commercial aspect was the rise of the MIDI sequencer software. Building on the techniques of earlier electronic sequencers to repeat short series of notes, software sequencers continue to provide more comprehensive musical transformations.

Alongside sequencing, music notation programs were also appearing at this time, although it took the desktop publishing revolution of the early 1990s for all the appropriate technologies to fall into place, notably the postscript font-description language and laser printing. Computer music publishing is now the norm rather than the exception. The first program to successfully combine both sequencing and notation was C-Lab's *Notator* on the Atari computer, which proved the rule that you only need one "must have" program to sell a computing platform. Over the years, this program has transformed into Apple's Logic Pro software.

As personal computer power increased in the late 1990s, synthesis software (long the domain of expensive systems such as the *Fairlight* or computer workstations) became accessible. This is evident in the current popularity of hard disk recording systems, such as *Pro Tools*, as well as real-time signal processing systems, which are becoming practical on mobile computers for reverb and equalization, and even real-time synthesis as complex as frequency modulation, granular, and physical modeling.

The integration of many of these technical threads in computer-based composing, recording, publishing, and multimedia occurred around the late 1990s, and now digital music systems provide rich and expressive tools for the musician. Around the turn of the 21st century the increases in computing power reached a threshold where personal computers were powerful enough to manage most audio and some video processes in real time. This saw the concentration of computer music systems in software or "virtual" versions of what had been previously separate hardware components. The laptop computer had become the one-stop digital music workspace and an instrument for live performance. This process of concentrated computing power continues, with mobile devices such as smartphones and tablet computers increasingly becoming the site for computer music practices.

1.3.3.1 Reflection Questions

1. What were the dominant technological drivers of the past few centuries?
2. Where do you think the current borders of technical innovation are that will affect music making?
3. Given that new materials such as iron and aluminum have shaped the development of acoustic instruments, what changes have driven electronic/computer instrument development?
4. What have been the major developments in automated music described in this chapter?
5. The use of electronics has shaped music making over the past 100 years. Who were some of the musicians to first pioneer the use of electronic devices for music?
6. What has been the impact of audio recording on music making?
7. What was the basis of the compositional technique known as musique concrète?
8. What changes occurring during the 1990s are described in this chapter?

1.4 ALGORITHMS AND PROGRAMMING

Computers have been traditionally programmed to calculate solutions to numerical problems (the name "computer" itself reflects this—modern computers were viewed as a replacement for human computers in the military). This view, of course, is very restrictive, as any normal computer user can attest. Computers are wonderful for playing games, searching the Internet, and for making music. In this book, we introduce computer programming in the context of connecting number, music, and nature.

One of the more significant advantages of the computer for music making is its ability to be programmed: its ability to automatically do a series of tasks and to do them quickly. This is, of course, the basis for all software development but can also be the basis for a music making practice. Algorithmic music using a computer takes advantage of this ability to automate a series of instructions (an algorithm) to musical ends.

Definition: An algorithm is a series of steps (or instructions) for performing a task.

Examples of algorithms include instructions for assembling a bookshelf (assembly instructions sheet), steps for making spaghetti sauce (a recipe), and instructions for performing a musical piece (a musical score).

Computers can be programmed to follow such series of instructions using a programming language. When programming computer music, the challenge is to write instructions that lead to interesting and expressive music. Musical algorithms can describe how each of the musical elements is specified and varied as the piece proceeds. This can include control over the pitch, duration and loudness of notes, the timbre of sounds, the use of structural features such as repetition and variation, as well as tempo, volume, balance and so on.

The ability of computers to run algorithmic processes (programs, or sequences of steps) gives the impression that computers have autonomy and are possibly "smart." At its most advanced levels this autonomy is referred to as Artificial Intelligence (AI), most well known through systems such as IBM's Deep Blue for playing chess, and popularized through science fiction systems such as *Hal* in the science fiction film *2001—A Space Odyssey* and androids such as *R2D2* in *Star Wars* or robots such as *Walle* in the film of the same name.

In algorithmic music systems the intention and possibilities are generally far more modest, even though some comprehensive systems, such as Experiments in Musical Intelligence by David Cope, can construct complex and complete pieces. Generally, algorithmic composition systems are used for more mundane purposes, such as generating a tonal melody of a few bars, creating valid variations in a 12 tone row, suggesting possible chorale harmonization, or sonifying mathematical structures such as fractals or artificial life worlds by converting the numbers generated by formulae into pitches, rhythms, and form.

Many algorithmic systems deal with music at the note level, specifying or manipulating attributes such as pitch, duration, and dynamic. This is historically the most prevalent way of thinking about music and is the basis for common practice notation, so it is not surprising that note-based generative systems are common. Algorithmic processes can be applied in many ways to notes. Small pitch changes at the frequency level can be used for microtonal music, or loudness may be controlled by a function introducing a kind of jitter or instability to the note which, if subtle, may add some life to an otherwise static electronic performance. Similarly, subtle changes can be applied to the dynamic levels of a repeated phrase in order to provide variety which masks the machine-like repetition to