



NOAH WARD RIP-FRUIIN
EXPRESSIVE
PROCESSING

Digital Fictions, Computer Games, and Software Studies

Expressive Processing

Software Studies

Matthew Fuller, Lev Manovich, and Noah Wardrip-Fruin, editors

Expressive Processing: Digital Fictions, Computer Games, and Software Studies,
Noah Wardrip-Fruin, 2009

Expressive Processing
Digital Fictions, Computer Games,
and Software Studies

Noah Wardrip-Fruin

The MIT Press
Cambridge, Massachusetts
London, England

© 2009 Noah Wardrip-Fruin

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

All images and screenshots were either created by the author or are reproduced with permission.

The image of *Pong* in chapter 1 is courtesy of The International Arcade Museum.

For information about special quantity discounts, please email special_sales@mitpress.mit.edu

This book was set in Adobe Garamond, Helvetica Neue, and Arena by Michael Crumpton.

Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Wardrip-Fruin, Noah.

Expressive processing : digital fictions, computer games and software studies / Noah Wardrip-Fruin

p. cm.

includes bibliographical references and index.

ISBN 978-0-262-01343-7 (hardcover : alk. paper).

1. Interactive multimedia—Social aspects. 2. Computer games—Social Aspects. 3. Digital media. 4. Digital computer simulation. I. Title.

QA76.76.159.W37 2009

006.7—dc22

2009000144

10 9 8 7 6 5 4 3 2 1

Chapter 1

Introduction

Media Machines

A computer is a strange type of machine. While most machines are developed for particular purposes—washing machines, forklifts, movie projectors, typewriters—modern computers are designed specifically to be able to *simulate* the operations of many different types of machines, depending on the computer’s current instructions (and its available peripherals).

This is why a computer can simulate a movie projector: showing a set of image frames in quick succession. It’s also why a computer can act like a tape player: reading and amplifying a stream of sound data.¹

And it is for this same reason that computers can be instructed to act like previously impossible types of machines. A computer can simulate a typewriter—getting input from the keyboard and arranging pixels on the screen to shape the corresponding letters—but it can also go far beyond a typewriter, offering many fonts, automatic spelling correction, the painless movement of manuscript sections (through simulations of “cut” and “paste”), programmable transformations (such as “find and replace”), and even collaborative authoring by large, dispersed groups (as with projects like Wikipedia). This is what modern computers (more lengthily called “stored-program electronic digital computers”) are designed to make possible: the continual creation of new machines, opening new possibilities, through the definition of new sets of computational processes.

1. During the blog-based peer review of this book’s manuscript on Grand Text Auto (discussed in the afterword) Mark Marino suggested that the ability of computers to simulate movie projectors and tape players is “the result of a set of cultural ...

Notes continued at end of this chapter.

“Digital media” are the media enabled by this possibility. This includes Web projects like Wikipedia and also all computer games. The first modern computer games were created on early stored-program computers, and since then we have seen a major cultural impact from the fact that a computer can not only simulate a pinball machine but also act like game machines never seen before: a *Tetris* machine, a *Doom* machine, a *SimCity* machine, and more.

Personally, I am fascinated by the possibilities that digital media open for fiction. A blossoming of new models of character, story, and language is being enabled by computational processes. From computer games with epic structures to experimental interactive films, digital fictions are providing diverse experiences for a wide range of audiences. From ambitious artificial intelligence (AI) experiments to straightforward uses of weblogs and email, authors are creating digital fictions at a wide range of technical complexity. The field is already too vast to cover in a single book.

Luckily, quite a number of books have already been written about digital literature, and many more have been written about digital media more generally. Almost all of these, however, have focused on what the machines of digital media look like from the outside: their output. Sometimes the output is considered as an artifact and interpreted in ways we associate with literary scholarship and art history. Sometimes the output is seen in relation to the audience and the wider culture, using approaches from fields like education and sociology. And there are, of course, a variety of other perspectives. But regardless of perspective, writings on digital media almost all

ignore something crucial: the actual processes that make digital media work, the computational machines that make digital media possible.

On the one hand, there is nothing wrong with this. Output-focused approaches have brought many valuable insights for those who seek to understand and create digital media. Yet, on the other hand, it leaves a big gap.

This book is my attempt to help bridge the gap. As far as I know, it is the first book focused on computational processes that comes from the perspective of media, games, and fiction (rather than software engineering or computer science). It is a first passage across the gap, and we will want to move much more weight across over time. But hopefully it demonstrates that there is something to be gained by being able to move between the gap's two sides, being able to see the inside and outside of digital media's machines.

Expressive Processing

Bridging the gap requires talking about processes in new ways and connecting them with broader issues. In this book I work to do this through a notion of *expressive processing*. This term is meant to evoke, in the forthcoming discussions, two important things about processes.

First, computational processes are an increasingly significant means of expression for authors. Rather than defining the sequence of words for a book or images for a film, today's authors are increasingly defining the rules for system behavior. When I play a computer role-playing game (RPG), author-crafted processes determine how I can speak with the non-player characters (NPCs). When I play a simulation game, author-crafted processes

2. For more on this project, *Screen*, see chapter 9.

determine the operations of the virtual economy. There is authorial expression in what these rules make possible—and also in what they leave out, as compared with what we see in the everyday world of human conversation and economic transactions.

Computational processes can also be used to craft possibilities that aren't simplified models of phenomena from our everyday world. For example, in my own collaborative work I take advantage of processes supporting room-size virtual reality displays in order to create the illusion of words (from short fictions about memory) peeling loose from paragraphs, flocking around the audience, and flying back (or breaking apart) when hit by an audience member's hand.² The exact details of this experience are different every time, but it always unfolds within parameters determined by authored processes, and we can see authorial expression in these processes as surely as in those meant to evoke elements of the everyday world.

Second, I use the term *expressive processing* to talk about what processes express in their design—which may not be visible to audiences. Just as when opening the back of a watch from the 1970s one might see a distinctive Swiss mechanism or Japanese quartz assembly, so the shapes of computational processes are distinctive—and connected to histories, economies, and schools of thought. Further, because digital media's processes often engage subjects more complex than timekeeping (such as human language and motivation), they can be seen as “operationalized” models of these subjects, expressing a position through their shapes and workings. Processes, when examined, may also express a very different set of priorities or

capabilities than one might assume from authorial or scholarly descriptions of the system.

These possibilities are explored further in the coming pages, in discussing particular works of digital media. This second sense of *expressive processing*—what processes express through their designs and histories—is important to me because I think it is central to understanding digital media. I also believe that, from this perspective, digital media provides particularly legible examples of things that we need to understand about software in general. For instance, when we understand the capabilities and histories of AI techniques in the context of a relatively easy to evaluate area such as computer games (in which, say, NPCs may act in obviously inappropriate manners), we can use that understanding to judge proposals for using similar techniques in higher-stakes social contexts (e.g., areas such as surveillance).

The main body of *Expressive Processing* discusses these issues in more detail. Rather than theoretical discussion, though, most of the rest of this book is dedicated to a close examination of a set of influential examples. These examples are key to the history of process-oriented innovation in fiction and games. I find it a fascinating history, which I hope will prove thought provoking for fiction writers, game designers, AI researchers, and anyone with an interest in the history of technology or the future of fiction.

It is also a history almost never told. Despite increasing interest in computer games, digital fictions, and digital media more generally, only bits and pieces of the movements traced in this book have appeared elsewhere. Instead, most books on related topics have focused on

other traditions—ones with relatively stable processes that have served as platforms for the creativity of writers and game designers, who have used these relatively stable forms to produce a compelling variety of output for audiences. Good examples of such work include Nick Montfort’s book on interactive fiction (*Twisty Little Passages*, 2003) and Matt Barton’s book on computer RPGs (*Dungeons & Desktops*, 2008). There is also a significant body of work that specifically looks at what writers can accomplish with digital media, useful introductions to which are *Electronic Literature: New Horizons for the Literary* by N. Katherine Hayles (2008) and *Les Basiques: La Littérature numérique* by Phillippe Bootz (2006). And of course there are helpful practice-oriented guides to what can be accomplished with the game industry’s common processes, such as Lee Sheldon’s *Character Development and Storytelling for Games* (2004) and Chris Bateman’s *Game Writing: Narrative Skills for Videogames* (2006).

I do write about some of the work considered by these authors in *Expressive Processing*, but largely as a way of talking about where we are today. Most of my attention, instead, is on the past and the future. I draw many of my examples from AI research projects because this is where some of the most revealing processes for character and fiction have been crafted. A number of these are quite famous within certain circles, such as the simulated therapist *Eliza*; the first story-generation program, *TaleSpin*; and the widely discussed experimental game *Façade*. Others are less well-known. In this book I move along a conceptual trail that connects them—and also compare them with mainstream games such as *Star Wars: Knights of the Old Republic*, *F.E.A.R.*, and *The Sims*. In doing so,

I hope to provide a parallel history and set of concepts focused on the development of processes that can complement the ongoing discussion that concentrates on the nonprocess work of writers and designers.

Tracing this history, and making these connections, I also draw out a number of ideas that may be of use for digital authors, scholars, and members of the public. For instance, I consider projects from this history as iconic examples of three “effects” that appear throughout digital media, including in today’s systems, and from which we can learn important design lessons. I will say more about them toward the end of this chapter. First, however, it will be useful to introduce the perspective on digital media that underlies my work here.

A View of Digital Media

Let me return to my first conception of expressive processing: the possibility of creating new simulated machines, of defining new computational behaviors, as the great authoring opportunity that digital media offers. Seizing this opportunity requires a bit of a shift. It is common to think of the work of authoring, the work of creating media, as the work of writing text, composing images, arranging sound, and so on. But now one must think of *authoring new processes* as an important element of media creation.

In undertaking this shift, it may be helpful to think of the creation of a piece of digital media as being organized like figure 1.1. The work is made up of data and process, with a somewhat fuzzy line between them.³ The data elements are mostly precreated media (text, still images, video and animation, and sound and music) and the sorts

3. Though the concepts of data and process seem clear enough as ideas, in practice any element of a system may be a mixture between the two. For example, the text handled by a Web application is generally thought of as data. However, this ...

Notes continued at end of this chapter.

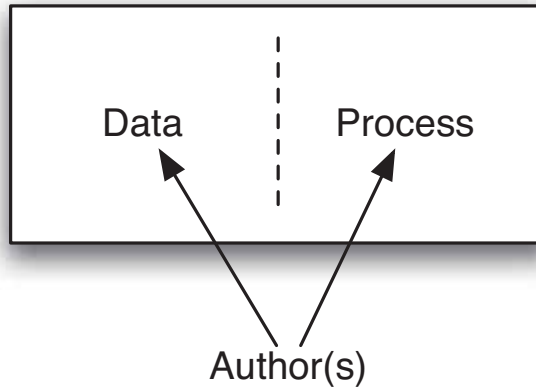


Figure 1.1. Authoring data and process.

of things that are stored in spreadsheets (lists and tables of information, with varying degrees of structure).

The processes, on the other hand, are the working parts of the simulated machine. Some are dedicated to tasks with simple structures, such as displaying a series of video images on a screen. But many of digital media's tasks are more complex in structure, requiring processes capable of performing in a range of different ways. Even a simple piece of digital media such as *Pong* (figure 1.2) has processes that define behaviors much more complex than showing a series of images in quick succession. The processes of *Pong* define and calculate simple rules of physics (how the ball bounces off the paddles and walls) as well as simple game rules (who receives each serve, how points are scored, and how winning is achieved) that, when well tuned, can combine to create a compelling experience of gameplay—even in the face of remarkably primitive graphics.

Of course, the idea of creating media through the authoring of novel processes is not new. Tristan Tzara's Dada cut-up technique was presented in the wake of World War I as a process for turning a chosen newspaper

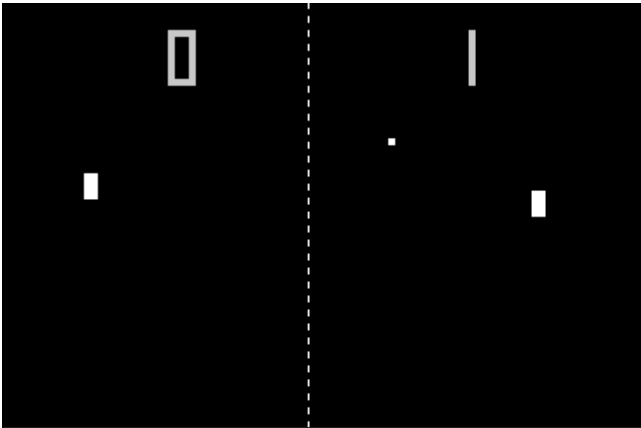


Figure 1.2. The iconic early video game *Pong* gives players a simple goal: to use their simulated paddles to knock back a simulated ball—keeping it in play until one player misses, causing the other player to score.

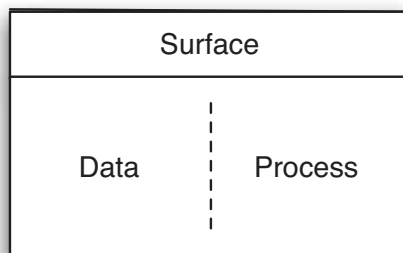
article into a poem. On a more technological level, the pioneers of early cinema had to develop novel processes (embodied in physical machinery) to capture and display their sets of image data. And on a longer-term level, the creation of board and card games has always primarily been the development of process definitions—embodied in game rules—that determine how play moves forward.

In important ways these noncomputational media processes are like the processes of digital media: they are defined previously, but (at least in part) carried out during the time of audience experience. This is true as Tzara pulls a paper scrap from his sack, as the Zoetrope image flickers, as the poker hand goes through another round of betting, and as the image of a *Pong* ball bounces off the image of a *Pong* paddle. The processes of digital media, however, are separated from noncomputational media processes by their potential numerousness, repetition, and complexity. For example, we might play a game of tennis using the rules of *Pong*—since they're simpler than the

normal rules of tennis. But we wouldn't want to play *Pong* as a board game, having to hand execute all the processes involved even in its (extremely simplified) modeling of physics. It is the computer's ability to carry out processes of significant magnitude (at least in part during the time of audience experience) that enables digital media that create a wide variety of possible experiences, respond to context, evolve over time, and interact with audiences.

Given the importance of such audience experiences, the author's view is not the only one I consider in this book. To reflect that, here is a slightly more complex figure (1.3) that adds a layer called "surface" over the initial data and process. In this book, the surface of a work of digital media is what the audience experiences: the output of the processes operating on the data, in the context of the physical hardware and setting, through which any audience interaction takes place. When playing a console game, for example, the surface includes the console and any indicator lights or other information it provides, the television or monitor and any image it displays, the sound hardware (e.g., television speakers, stereo, or headphones) and any sound produced, and the controller(s) with their

Audience(s)



Author(s)

Figure 1.3. Adding surface to data and process.

buttons, lights, and perhaps vibrations.⁴ The audience experience of digital media is that of being connected to, and in some cases through, the surface.

I find this a more satisfying view of digital media. Yet this figure, like my discussion so far, doesn't account for something quite important about digital media processes: the fact that they generally don't operate on their own. From Web-based knowledge repositories to console-based video games, the operations of digital media are, in crucial ways, only truly realized in contact with audiences. A wiki's processes mean little if the audience doesn't use them to add data, edit it, and follow the connections embedded in it. Similarly, many of a game's processes never come into operation if the game has no player.⁵

None of this is any surprise. But as I discussed earlier, we generally understand this situation from the audience's perspective, looking at both the audience's actions and the work's behavior as though the work is a proverbial black box. I believe it is also essential to understand this situation more reciprocally: to think about the *relationship* between the audience's experience and the system's internal operations.

Figure 1.4 adds a representation of interaction to my diagram of digital media. While *interaction* is certainly a contested term, for the purposes of this book I am defining it as a change to the state of the work—for which the work was designed—that comes from outside the work. Interaction takes place through the surface of the work, resulting in change to its internal data and/or processes. In many cases, some trace of interaction is immediately apparent on the surface (e.g., an audience member types and the letters appear as they are typed,

4. Many console games also have more complicated surfaces, often in the form of additional controllers such as dance mats, simulated musical instruments, or cameras. And digital fictions and games more generally may have much more ...

5. At the same time that processes come into operation through audience actions at the surface, process and surface also define some of the possibility space for audience actions, as well as the feedback produced, and thereby shape the audience. For ...

Notes continued at end of this chapter.

or an audience member moves her hand and a video image of her hand moves simultaneously), but this is not required. Interaction, while it always changes the state of the work, can be approached with the primary goal of communication between audience members—as when communicating through a shared virtual world such as *World of Warcraft* or *Second Life*. Finally, given the definition of interaction that I am using, it also becomes clear that digital media works interact with more than audiences—which is why the revised diagram also notes the possibility of interaction with outside processes and data sources.

Obviously, the diagram has now become rather complicated. But I believe that keeping all these elements in mind is important to our initial thinking about a work of digital media, as a creator, scholar, or audience member. And of course this diagram is still immensely simplified, compared with the details of the actual components and connections of most works of digital media. This simplification is a first step toward consideration of digital media works at a yet more abstract level.

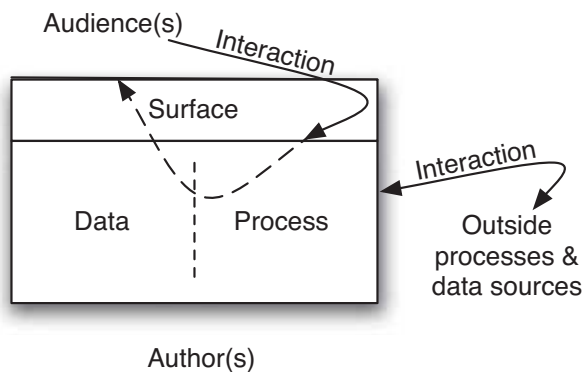


Figure 1.4. Adding interaction to the diagram of digital media.

Operational Logics

The primary goal of the model of digital media developed above is not found in its individual components. My main hope is not that readers will come away with an understanding of every nuance of what I mean by *data* or *surface*. Rather, my hope is that a basic understanding of these components will provide the foundation for a new approach to thinking about digital media (and computational systems more generally). I use the term *operational logics*, described further below, to name a new type of element specific to procedural systems that this type of thinking can help identify and analyze.

When a work of digital media operates, this can be seen as an interplay between the elements of the model discussed so far: data, process, surface, interaction, author, and audience. Observing the specifics of this interplay can be informative. Is the system actually doing what it is described as doing? What unspoken assumptions are built into the ways in which operations proceed?

At a higher level of abstraction, however, we can also notice patterns in this interplay. I call these patterns *operational logics*. In my first discussion of them (Wardrip-Fruin 2005), I talked about common spatial or graphical logics, such as *collision detection*. This is a term for when a system notes the virtual “touch” of two simulated objects. When the *Pong* ball bounces off the paddles and walls, this is collision detection. It is also collision detection when the *Doom* engine prevents players from walking through walls—and when it determines that a bullet (or chain saw) has hit a target.⁶

From these examples it is probably clear that the same

6. I will return to the discussion of spatial logics in chapter 9.

operational logic can be implemented in a wide variety of ways on a continually expanding set of platforms. Once we have identified a logic, it can be informative to make comparisons between different instances. These comparisons can delve deep into the specifics of how the logic is implemented for different works or operate at a higher level, looking at how the same logic can contribute quite differently to the experience of a set of works. This is true not only of spatial logics but also of many operational elements of systems. Later chapters in this book will consider operational logics ranging from the quest logics implemented in many computer games to the assertion and inference logics implemented in many symbolic AI systems.

Although operational logics will, I hope, prove powerful for comparative studies, in this book they are primarily used within the examination of individual works. More specifically, while an operational logic can be seen as a pattern that arises in the interplay of the elements of a digital media system, I am interested in the examination of the interplay of a system's operational logics—and in this as a starting point for critical interpretation. I pursue this method most explicitly in this book's chapter on *Tale-Spin*, which identifies planbox-based planning as the work's dominant logic, driving the others (a fact invisible when examining only the work's surface). Though not as explicit in other chapters, this approach underlies much of this book's analysis—and it is my hope that these examples will encourage others to further develop their own approaches of this sort (see sidebar, "Terms for Thinking about Processes").

Three Effects

At a higher conceptual level, this book also identifies three *effects* that can arise in the relationship between system processes and audience experiences, which may occur with a wide variety of operational logics. These will serve as the major waypoints for the remainder of this volume. The first—*the Eliza effect*—is the well-known phenomenon in which audience expectations allow a digital media system to appear much more complex on its surface than is supported by its underlying structure. I will also consider what most authors have ignored, however: during playful interaction with the simulated therapist for which the *Eliza* effect is named, the illusion breaks down rapidly. One alternative to breakdown with a system of this sort is to severely restrict interaction. Another is the one pursued by many modern games: never building up the *Eliza* illusion and instead clearly representing the operations of a simple system on the work's surface. But these simple systems prove too limited for the fictional experiences that games seek to make available to their players, resulting in breakdowns of a different type. This leaves only one option for those seeking to create ambitious playable fictions: more developed system models of story and character.

Expressive Processing's next waypoint is an effect named for the first major story-generation system, *Tale-Spin*. Though created decades before modern games, it represents an initial step toward a more flexible model of fiction. Its characters have goals, make plans to satisfy them, examine their relationships with other characters, speculate about the plans of others, and so on. But little of this is visible to audiences in the resulting story, and

little can even be deduced through repeated interaction. It is this that leads me to coin a new term—*the Tale-Spin effect*—for works that fail to represent their internal system richness on their surfaces, in an inversion of the *Eliza* effect that is not uncommon in digital media. From there *Expressive Processing* looks at a different model of character planning (in the first-person shooter game *F.E.A.R.*), the alternative of statistical AI approaches, and the development of systems for story generation over the next several decades. While the examined story-generation systems demonstrate a variety of interesting models of fiction, most focus on this to the exclusion of a rich audience experience, resulting in the continued presence of the *Tale-Spin* effect in even the most advanced work in this area.

The book's third waypoint is named after a work famous for making a relatively complex system into a rich, enjoyable experience of play: the city planning game *SimCity*. *The SimCity effect* is my term for systems that shape their surface experience to enable the audience to build up an understanding of their internal structure, especially a relatively complex one. This approach underlies the most popular and influential game built around simulated human characters: *The Sims*. It also provides a way of understanding the development of systems for believable animated characters and interactive drama, which eschew the most direct representations of internal system state. Finally, the *SimCity* effect provides an important design guideline for those who seek to make another powerful technology for fiction—language—playable in new ways.

Terms for Thinking about Processes

The idea of operational logics is my way of talking about aspects of digital media with which other authors and designers are also concerned. Given this, it should be no surprise that operational logics are related to some of the ideas others have proposed for thinking about digital media systems. Ian Bogost's *Persuasive Games* (2007), for example, positions my notion of operational logics as the “tropes” of procedural rhetoric (which seems appropriate, at least for those logics that clearly structure audience experience). The concept of operational logics is also not unrelated to Bogost's figure of the “unit operation”—though Bogost's aim is in some ways more general in that his work creates a foundation for “any medium—poetic, literary, cinematic, computational” to be “read as a configurative system, an arrangement of discrete, interlocking units of expressive meaning” (2006, ix). At the same time, Bogost's aims are also more specific. For example, he writes, “Unit operations are modes of meaning-making that privilege discrete, disconnected actions over deterministic, progressive systems. . . . In software technology, object technology exploits unit operations; structured programming exhibits system operations” (3). In my current thinking, I use the concept of operational logics only in reference to computational systems (not, for instance, traditional cinema) and in a manner that encompasses both what Bogost calls unit operations and system operations.

Game design is certainly one area for which it is particularly important to consider the parts of a system that operate and connect, and that can be combined and adjusted to create successful audience experiences. One influential framework for thinking in these terms is called *MDA*, standing for Mechanics, Dynamics, and Aesthetics (Hunicke, LeBlanc, and Zubeck 2004). In the language of game design, especially for those familiar with the *MDA* framework, what I call operational logics might often be termed *mechanics*—except that *MDA* mechanics are framed as being limited to operational logics experienced by the audience. As Robin Hunicke, Marc LeBlanc, and Robert Zubeck write, “Mechanics are the various actions, behaviors and control mechanisms afforded to the player within a game context.”

Coming from a rather different direction, computer science has also thought quite a bit about the question of system operations. A general computer science term for a more abstract level of consideration of the operations of process and data is *algorithms*. This concept certainly shares the relatively implementation-independent nature of operational logics (an introduction to algorithms class may implement the “bubble sort” algorithm in many languages and on many platforms). Yet, for an operational logic to be useful in interpretation, it need not have the characteristics that D. E. Knuth requires of algorithms (1968). For example, it need not be *definite* (the terms employed can require human knowledge rather than being specified at a computable level of detail). More fundamentally there is little emphasis in computer science education on considering algorithms and their relationships critically or aesthetically—rather than in terms such as efficiency.

Looking Forward

If I am successful, *Expressive Processing* is two books in one. One book makes an argument that we need to pay more attention to the processes of digital media. This book is an example of the expanding area of software studies (discussed in more detail in chapter 5). It makes its case on authorial, critical, and political levels. It outlines my two primary meanings for *expressive processing* along with the three general effects I have identified in the relationship between system processes and audience experiences. It presents a way of thinking about software and media that grows from my interactions with writers, artists, game designers, computer scientists, and humanities scholars—and I hope it will be of use to members of all these groups. A book like this must be supported with examples, but I believe that such a book could have been written by an author with a passion for any one of a variety of types of digital media, from computer music to generative visual art.

The other book within these covers is the one shaped by my passion for digital fictions and games. It tells a history of process-oriented innovations in these areas—a history that can provide inspiration (and cautionary guidance) as we create the projects that will shape the future of storytelling and play. We are in the midst of a creative explosion in these areas, and that is an exciting place to be. As our work moves forward, we are also coming to see that the exciting possibilities aren't necessarily those assumed earlier. We aren't finding greater authorial power in closer approximations of everyday reality. Rather, we're finding it in models that are deliberately artificial, emphasizing elements that capture the imaginations of

authors, expressing a view of the world that can matter to audiences—just as traditional fictions have left out most of their characters’ lives in favor of those elements central to that particular work. Similarly, we aren’t finding greater potential in obviously “natural,” “immersive,” or “invisible” interfaces. It is instead those that expose the evolving state of the underlying system and the opportunities for audience action in connection with their fictions that are creating exciting new roles for us (no longer simply visual “viewers,” textual “readers,” or formal system “players”) as well as helping us develop new modes of understanding for fictional worlds. This in turn provides the strongest connection for the two books between these covers. Coming to understand fictional worlds as systems—and exploring their potential through play—is also a powerful means of coming to understand our evolving society, in which (often hidden) software models structure much of how we live now.

I assume most readers are drawn to *Expressive Processing* primarily for one of these two intertwined books. My hope is that whatever your starting interests, you will come away having found something of value in both.