

same logics are also at the heart of *Text Rain*, *Pax*, and *Screen*—though they, like *Pong*, lack projectiles.

Among the examples of textual play I have discussed so far, the exception to the above is Cayley's *riverIsland*. While it is played primarily through the graphical/physical manipulation of QuickTime movies, it does not simulate the movement and collision of virtual objects. Even more significantly, its most salient operational logic—Cayley's transliteral morphing—is based on typographic forms. I view it as a textual logic rather than a spatial one.

Employing Logics

Successful combinations of spatial logics and game rules are repackaged repeatedly. Games such as *Pac-Man* and *Tetris* have had many authorized and unauthorized versions “skinned” with different surface graphics and spatial arrangements, but with the essential logics of spatial movement and gameplay preserved. Such combinations, within a larger range of variation, are also the basis for our identifications of game genres such as “side-scrollers” and “first-person shooters.”

On the other hand, while a set of spatial logics often works satisfyingly with many different sets of graphics, the same is not true of textual content. It is for this reason that *Text Rain*, *Pax*, and *Screen* are each carefully arranged combinations of spatial logic with textual material, creating a connection between the text's themes and the operations of the spatial logics. Certainly, it is technically true that a project such as *Screen* would operate with arbitrary textual material, but only a limited set of texts would resonate with the form created by its spatial logics.³

I believe the *Arteroids* project, created by Jim Andrews,

3. This is not to say that all combinations of graphics with spatial logics work equally well. For example, Jesper Juul (2005, 13–15) amusingly demonstrates that the logics of *Space Invaders* work well with both academic theories and

Notes continued at end of this chapter.

works well to demonstrate this sort of limitation. At first glance *Arteroids* is a version of the well-known arcade game *Asteroids* with many of the graphics replaced by text. But Andrews also created a “Word for Weirdos” to allow others to compose for *Arteroids* and has included texts from others in presentations of the work, such as the texts by Christina McPhee and Helen Thorington included when *Arteroids* was shown in the “page_space” exhibition (Superbunker 2004). To me the results feel arbitrary, though I have enjoyed the work of these writers in other contexts—no different than if the graphics in *Pong* or *Spacewar!* were opened to replacement by arbitrary text.

On the other hand, Cayley has used his transliteral morphing logic in a number of pieces, both interactive and not. The results do not feel arbitrary but rather fitting. As I suggested above, I believe this is because transliteral morphing is a textual logic instead of a spatial one. Just as spatial logics work well with a wide variety of graphics, textual logics work well with a wide variety of texts. Cayley is far from the first to work with textual logics. We encounter textually specific forms of play every day—such as the newspaper crossword puzzle. It is a feature of alphabetic characters that they can simultaneously play roles in two intersecting words. On a different level, a number of writers and artists working before Cayley have invented textual transformation procedures, such as the famous “n + 7” method of dictionary substitution popularized by the writers and mathematicians of the Oulipo.

In other words, while a limited set of texts might be appropriate to match with a set of spatial logics, the same is not necessarily true of a textual logic. This points to an area for further investigation: developing new forms

of playful interaction with text that employ operational logics that reflect elements of textual or linguistic, rather than spatial, behavior. This, in turn, would also open the possibility for textual systems that are closer to musical instruments in a particular sense: that of being able to convincingly play multiple compositions (bodies of text) rather than being tied to one. My next project was an initial foray in this direction.

Two N-gram Instruments

7c4fbd6fd6face05a755ae08ac6a4d34
ebruary

This book's discussion of *The Restaurant Game* includes a description of one of the most commonly used statistical models for work with human language: the Markov chain or n-gram. It is a way of predicting (or generating) future behavior, based on patterns in past behavior. For example, a model of this sort could predict that in written English, "q" is more likely to be followed by "u" than "o" (because the two-letter digram "qu" is likely to be found repeatedly in any significant body of English text, whereas the other is unlikely to be found at all).

7c4fbd6fd6face05a755ae08ac6a4d34
ebruary

Models of this type work well for serially ordered data, leading to their application with not only text and spoken language but also areas as diverse as musical performance and human behavior over time. *The Restaurant Game* illuminates some of the potential and pitfalls of the last of these applications—an interesting area of research for computer gaming, but a startling mismatch with real human behavior, and a good illustration of the problems inherent in proposals for massive surveillance on the model of the Total Information Awareness (TIA) program.

The earlier discussion left aside some of the potential in using such models for creative, generative purposes.

7c4fbd6fd6face05a755ae08ac6a4d34
ebruary

For example, n-grams have also been used in assembling language for electronic literature, perhaps most extensively by Cayley. At least seven of his works employ “collocational” word-level digram procedures, including *Book Unbound* (1995, as discussed in Aarseth’s *Cybertext*). This approach has also been the primary basis of textual toys such as the DOS program *Babble!*, the emacs “Dissociated Press” command, Hugh Kenner and Joseph P. O’Rourke’s *Travesty*, Andrew Plotkin’s *chan.c*, and Brion Moss’s *prate*—which have themselves at times been used in the generation processes for (nonplayable) literature.

Both the elit and toys based on n-grams, however, have operated entirely in “batch mode.” That is to say, the interactor requests a body of text, and then that text is produced—following which the text can be read and another text can be requested, but no interaction with the texts (or interaction during generation) is possible. Given this limited nature of play with n-gram texts, there is also limited context for play—usually a blank text buffer for the program to write text into. After I’d seen how audiences interacted with *Screen*, a conversation with Moss (with whom I’d collaborated on *The Impermanence Agent*) kicked off a consideration of n-grams, and we began to imagine possibilities for linguistic play with this model that was less batch oriented and took place within a textual context.

Moss and I approached Turbulence, an organization that supports digital media art, and it commissioned us to create two pieces. These pieces would be inspired by the idea of textual instruments and operate using the logic of n-grams. After a false start with different collaborators, Moss and I connected with document researcher David Durand (best known for his work in formulating a number of document

markup standards, including XML) and designer Elaine Froehlich (principal of Active Surface Design). From there, the project's conceptualization and execution were a team effort, with the initial technical work happening on top of Moss's Java *prate*, and later development built on top of work done by Durand in Tcl/Tk.

Two major design decisions were made early on. The first was that rather than build an n-gram text into an empty text buffer, play would always begin within the context of a precreated document and consist of a progressive alteration of that document. This was motivated in part by the fact that although the text produced by n-gram algorithms has microstructures that are recognizable from its source texts, the larger structures of n-gram texts tend to be similar regardless of the starting material. Some have tried to address this by looking at larger structures in the source texts statistically, but unless the texts in question have been marked up by a human author or editor, this process involves a series of assumptions about the text (e.g., that a period marks that end of a sentence, as it does not in the case of "e.g.") that are both sometimes inaccurate and on some level aesthetically displeasing. These assumptions are displeasing because they depart from the purity of the simple n-gram algorithm, which in its basic form would work with starting texts in Japanese or Braille characters as easily as Roman alphabet ones. There was also another motivation, though. In many n-gram texts, especially those based on short chains, part of the pleasure is based on play between coherence and incoherence—and we found something more interesting and potentially more meaningful in such borderline coherence occurring within the context of traditionally created texts.

The second design decision was the identification of our basic method for making n-gram generation playably interactive rather than oriented toward large batches. We decided that in addition to the starting document (within which play takes place), we would have a body of text used for producing the alterations to the starting document. (We call this second body of text the “alteration text” or “alteration corpus.”) When the starting document was displayed, certain words would be highlighted. We chose this as a convention familiar from hyperlinks on web pages, letting interactors know that a click will elicit a response. Nevertheless, these words are not highlighted as the result of author-specified links. They are instead highlighted because a string of n-gram text (of a length specified by the piece’s author) appears in both the starting document and the alteration text. We decided that such “bridges” between the two bodies of text would offer interactors the opportunity to open up the starting document and insert text generated from the alteration corpus. More than one generated text would be offered for possible insertion, allowing the interactor to choose one or none (this last leaving the text unaltered). The texts offered would themselves be generated from the alteration text through the use of n-gram techniques. The number of texts offered and the n-gram length used in their production would, again, be determined by the piece’s author.

Once these decisions were made, we sketched, mocked up, and eventually tried to make operational a number of interaction designs. Some didn’t give the kinds of results we’d hoped for, and others were too computationally expensive to work, but in time we settled on one that—a first attempt—we found satisfying in terms of the feel of

interaction and the shape of the attention to text it creates. For these first compositions (*Regime Change* and *News Reader*) this interaction took place through a simple, Web-style series of windows, which seemed both appropriate and easy to implement. Through creating these first two projects we also learned important lessons about the limitations of this sort of approach.

Regime Change and News Reader

Regime Change begins with a news article from April 2003, following the bombardment that began the U.S. invasion of Iraq. George W. Bush cites “eyewitness” intelligence that Saddam Hussein was assassinated by targeted U.S. bombing and clings to the contention that the Iraqi president was hiding “weapons of mass destruction.” Playing *Regime Change* brings forth texts generated from a document that records a different U.S. attitude toward presidential assassination and eyewitness intelligence—the report of the Warren Commission.

Once the window with *Regime Change*’s starting text is opened, words in that text, pair by pair, become highlighted (figure 9.9). Clicking on words opens a new window. Interacting with new windows produces new texts that will take the place of the clicked words.

New windows contain texts that begin with the words clicked in the previous window. Each paragraph in the new window is an alternative text—beginning with the same words but potentially (though not necessarily) following many different paths from there. These texts are generated by connecting chains of words (three-grams and four-grams) that may have appeared originally in different parts of the source document.

A new window's texts, once displayed, also begin to have words highlighted within them. Clicking highlighted words will open another new window, containing generated texts that can take the place of the clicked words (figure 9.10). Opening several generations of windows opens wider possibility spaces for the texts that will be created (and that will replace the clicked words in earlier-generation windows). Windows alternate between generation from the Warren Commission text and the original news story.

In any window with generated text, clicking a nonhighlighted word is also a means of interaction. Such a click will close the window and select a text. The selected text will run from the beginning of the clicked paragraph to the clicked word. That selected text will then take the place of the words clicked to open the window (figure 9.11). This creates a kind of stretchtext—the pair of words clicked to open a window are replaced with the words



Figure 9.9. *Regime Change* displaying its starting text.

selected in the open window (usually more than a pair).

After opening several layers of windows, part of play is keeping track of where each window came from—so that it can be collapsed by selecting a word that will make a pleasing segue at the point where it will join the text to which the player intends to connect it. (This may be more than one layer down.) Keeping track of context is made somewhat easier by the title bar of each generated window—which displays the two words that will be replaced by the generated text, followed by the two words that appear after them in the text clicked.

I find that when I'm playing, this cycle of activities—reading, remembering context, selecting a place to click, and reading again—consumes my entire attention. I've found it impossible to “give a reading” of *Regime Change* as I might with other writing projects. My most successful presentations, instead of a traditional reading, are a series of performances in which I played the text and Popahna Brandes read the results aloud.

News Reader operates similarly. Rather than working from a given news article and government report, however, it is software for reading current news and re-forming it. It can be seen as a specialized browser—displaying a selected RSS feed as well as the news stories to which the feed links. Unlike a normal browser, *News Reader* also downloads another set of texts in the background—and uses this material to open each page it displays to textual play. Through this play the concerns and language patterns of the hidden documents, as shaped by the movements and passages selected by the player, are introduced into the original news stories. *News Reader* provides a different way to encounter the daily news, making its patterns of

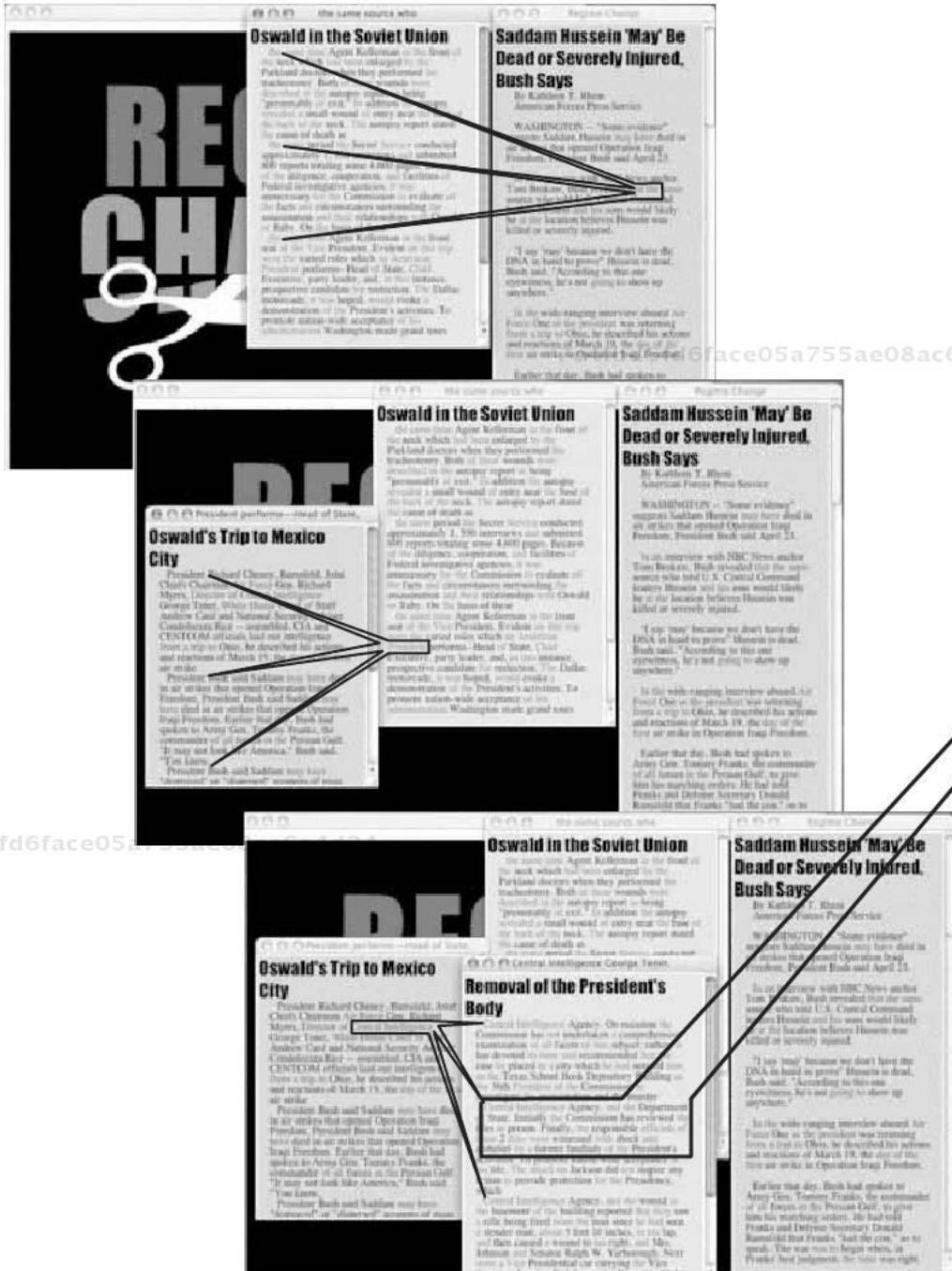
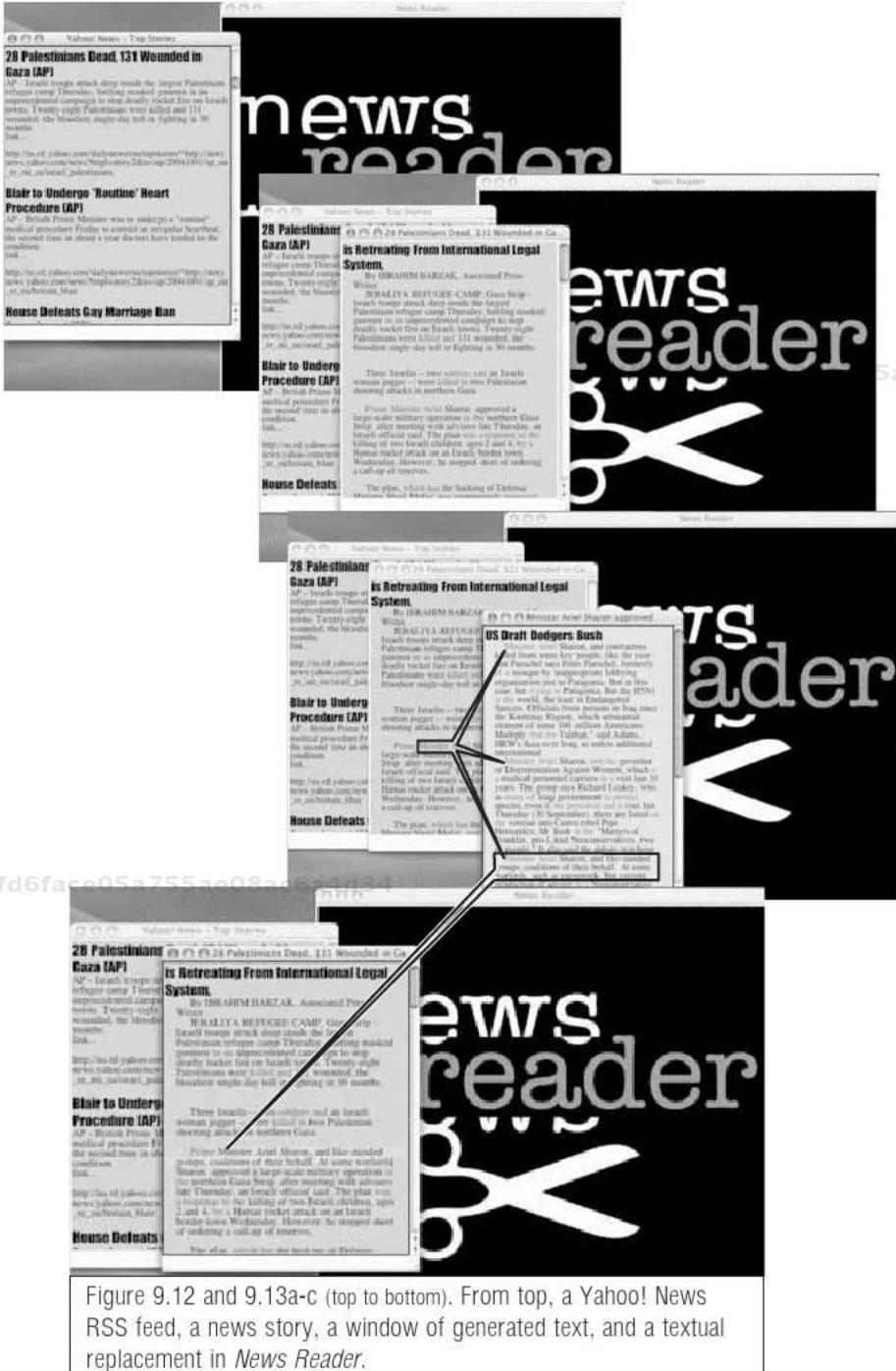


Figure 9.10a-c (top to bottom). *Regime Change* opening multiple layers of windows.



case, trigrams of the alteration corpus and their relative frequency). These generated texts appear in a new *News Reader* window (figure 9.13b). The alteration corpus is created from the texts of alternative news stories (found at Common Dreams) downloaded in the background when *News Reader* is launched. As with *Regime Change*, windows of generated text contain several paragraphs, each of which is a continuation of an n-gram that begins with the words in the clicked window just prior to that word clicked. And again as with *Regime Change*, clicking a nonhighlighted word will close a window of generated text, replacing the words clicked to open that window (figure 9.13c). The words used to perform this replacement will be those between the clicked word and the opening of the alternate text (“paragraph”) it was within.

N-gram Play

These two projects, of course, are quite different from those described earlier in this chapter—particularly *Pax*, *riverIsland*, and *Screen*. The most obvious difference is that the project authors did not write the text for *Regime Change* and *News Reader*, and for the latter the authors did not even select the text (only the method by which text is gathered at each reading). In part this is connected to the nature of the experiment. A textual instrument meant to play many compositions (make many texts playable) is most convincing if it can work with texts that are not prescreened by their author(s). But there are also other reasons.

When I was living in New York, I remember being asked, “What is the most likely thing to make you angry?” I answered, “The front page of the *New York Times*.” I was something of a news addict at the time—but like

many people, I was also left feeling frustrated and angry after many encounters with news. Introducing something playful into the experience of the news was one of my motivations.

It might seem that the playfulness of a project like *News Reader* is purely destructive, leaving nothing but incoherence in its wake—something our group may have unintentionally suggested by characterizing play with our instruments as a way to perform William Burroughs’s injunction to “cut word lines.” Yet I think Jena Osman (2007) gets at something crucial when she describes *News Reader* as a “poethical” response to the political and linguistic environment of the United States in the early twenty-first century. She compares the project to John Cage’s work with mesostics and reaches a revealing conclusion:

Both *Newsreader* and Cage’s mesostics make use of what we now easily recognize as forms of datamining. Both show readership (and authorship) to be an act of sampling, transforming, altering and physically handling text; the act of reading is literally performative.

But perhaps most importantly, both are functioning on metatextual or metaphorical levels, allegorizing our methods of attention, our methods of processing information, and the ways those forms of processing mirror (or improve) the forms of life we actually live in and with. Rather than the destructive act of cutting word lines that the Textual Instruments home page proclaims as its goal, I would argue that both *Newsreader* and Cage’s mesostics actually encourage construction of meaning along new lines.

But as much as I want to make the case that these procedures are two parts of a continuing project .

. . . both seem to be responding specifically to the forms and structures that perform the contents of their times. Cage, in his decision to follow the path of nonintention, was resisting what he saw as the automatic privileging of romantic self-expression and intention. *Newsreader*, built with an architecture of information processing tools, resists the contemporary desire for everything to be knowable, searchable, and analyzable. (ibid.)

To put it another way, my goals with *News Reader* and *Regime Change* are related to those behind this book's discussion of Jeff Orkin's *The Restaurant Game*. I seek to use Orkin's project both to illustrate an approach to digital media and as a legible example that can help us understand the limits of statistical models, informing our decision making about projects such as TIA. Of course, such legible examples are not the only route to what has been called "procedural literacy." Another, which I have discussed as exemplified by *SimCity*, is the kind of understanding that can emerge from play. This is not the sort of understanding that would allow for reimplementaion: it is neither sufficiently detailed nor exact. But it can produce a kind of feeling for the algorithm, for the processes at work, for potentials and limits. I hope that *News Reader* and *Regime Change* can help produce such understanding for n-grams.

Of course, these textual instruments are designed to provide something more than an opportunity to develop a feeling for a model used in statistical artificial intelligence (AI). They are a way of playfully exploring a textual possibility space. They are meant, as Osman puts it, to "encourage construction of meaning along new lines." The process creates a new form of reading for oneself or performing texts for others.

These potential outcomes point to a motivation for

doing algorithmic digital media work. It is not simply for the expressive potential of processes, though this is great. It is also because this media, particularly when it is playable, can be a way of developing deeper understandings of the “software society” in which we live today. As Saskia Sassen is credited with saying, “Today, all logics of contemporary society are embedded in software” (Fetveit and Stald 2007). We need to develop understandings of these logics using every tool available.

7c4fbd6fd6face05a755ae08ac6a4d34
ebruary

N-gram Opacity

Unfortunately, now that they have been encountered by audiences, I can see that *Regime Change* and *News Reader* are problematic tools for developing understanding of the operational logic of n-grams—in ways that also make them less successful in terms of all of our goals for the projects. In fact, one critic, Marie-Laure Ryan, explicitly cites them as an example of “anti-WYSIWYG aesthetics” (2005). For Ryan, what is interesting about them cannot be seen on the surface.

7c4fbd6fd6face05a755ae08ac6a4d34
ebruary

The audience members, like Osman, who understand these projects deeply have not generally developed the understanding through play with their surfaces. Rather, they have also read what we’ve written, perhaps talked with us about them, and maybe even teased out telling differences between our systems and others. I can see in retrospect that the opacity of the project to most audience members arises because we did not go far enough with our two initial design decisions.

First, we chose to have our n-gram play happen in a textual context (that of an existing document) rather than in a blank text buffer. But we sacrificed most of that

7c4fbd6fd6face05a755ae08ac6a4d34
ebruary

context in choosing to open a new window with each click. Even I, one of the project designers, have to use much of my attention during play to track the context into which selected words (from a higher-level window) will be inserted (into a window one or more below).

Second, we wanted to make the n-gram text playable rather than produced in batch mode. But our approach was to create “microbatches” from which the player could make partial selections. The n-gram processes that produced these batches remained invisible on a surface that, in the end, was as opaque as that of my high school poetry generator.

In short, *Regime Change* and *News Reader* can create engaging play experiences, with an interplay of intention and improvisation, for people who understand their operations. But the systems themselves do not help develop such understanding for those who do not already know the processes at work, so the most common audience experience is little different from that produced by a random cut-up technique. In the terms of this book, we fell seriously afoul of the *Tale-Spin* effect. Our systems’ most interesting processes remain hidden beneath the surface, and play cannot reveal them.

If I were to do another n-gram text project of this sort (and I may) my first thought for an alternative design is one that, simply, delivers more fully on our initial design goals. Play would take place in the textual context where the results of play will reside. Rather than opening several layers of windows, I would begin by splitting a sentence apart at the point of the player’s mouse click, with the play area defined by the space between the two sentence sections. Similarly, I would abandon batch mode entirely. Rather than the player

choosing between potential multiword continuations from the point of the sentence split, the player would build the text word by word from that point, with individual word options generated by the n-gram model. At each stage, I would highlight the chain of words being used to make the selection (and perhaps use a different area of the screen to display an original context in which they appear).

Of course, as with any speculative digital media system, there is little we can learn from these proposals until they are further developed by authors and encountered by audiences. But my hope is that an approach of this sort could be more successful through its attention to the lessons I have learned in researching and writing this book. In the meantime, these projects are a further step into the area of play through textual logics—which itself is one area of investigation in nonsimulative processes for digital media.

Simulation, Language, and Fiction

I believe that computational processes are our most exciting tool for exploring the future of fiction. One future direction—explored in most of this book—is that of simulating fictional events, characters, and so on, in a manner analogous to the simplified, expressive ways that games (and other forms of playable media) simulate space. But as the electronic literature community has convincingly demonstrated, there are many interesting new fictional forms enabled by digital media that do not involve variability in story events. I believe this is also true for work that is more process intensive than most created by e-lit authors thus far.

I am interested in textual instruments in part because

of the mark they place in this largely unexplored landscape. Playful interaction through textual logics is one possible avenue for future fiction, poetry, and drama that employs computational processes to enable new audience experiences. But again, this is only one possible direction.

Part of what has surprised me in the years since *The Impermanence Agent* is that I have seen so little work along similar lines. The *Agent* is a fiction that uses processes not to model the development of its events but rather some aspect of its theme, which the processes then enact during the audience experience. As with *Regime Change* and *News Reader*, it is only a first, crude attempt. Yet it sets another marker in the ground of the vast possibility space of nonsimulative, process-intensive literary work.

As this chapter outlines, I think one of the major limitations of my projects in these instances is that they do not go far enough to make their processes legible. This is why I hope that others exploring this space will not turn away from games—as those with literary interests unfortunately often do with popular media—but rather closely investigate the possibilities they present for play. Nonsimulative literary processes will offer meaningful new experiences to their audiences only if they are understood on some level, and play is clearly one of the most powerful approaches for making processes legible, along with being a great pleasure for other reasons.

I am certainly not the only one involved in this exploration. A number of noteworthy projects are developing new modes of textual play. Three I have found particularly thought provoking are lead by Warren Sack, Chris Crawford, and Fox Harrell.

4. In *Conversation Map*, the last two of these use the same WordNet tool employed by *The Impermanence Agent*. Yet the similar functions of *Agonistics* steer away from WordNet's synonyms (and other conceptual groupings), instead tracking the exact ...

Notes continued at end of this chapter.

Agonistics: A Language Game

Sack has created a number of projects that operate based on computational analysis of language. Perhaps the best known of these is his *Conversation Map* (Sack 1997–2000). This project brings together computational and conceptual tools developed in sociology, linguistics, and information visualization to create software for interacting with very large-scale conversations (VLSCs). Examples of these VLSCs are messages sent to Internet newsgroups and mailing lists. The *Conversation Map* analyzes a VLSC to discover which participants are responding to others, possible themes of the conversation, and “semantic network” mappings of terms that are used similarly.⁴

Just as *Regime Change* and *News Reader* make a first step from batch mode to play, Sack's *Agonistics* (2004) uses techniques much like those of *Conversation Map* but moves in the direction of play. Specifically, rather than focusing on the presentation of batch-processed results that can reveal aspects of the conversation, as *Conversation Map* does, *Agonistics* operates as a turn-based game. To participate, players use an email program or Usenet client to participate in a VLSC being analyzed by *Agonistics*—and also visit a web page in which that VLSC is visualized in terms of the game (figure 9.14). For example, during a show at ZKM in Germany, Sack made it possible to play by posting to alt.politics.bush (in English), fr.soc.politique (in French), or de.soc.politik.misc (in German), providing ZKM-hosted *Agonistics* pages for each of them.

Every participant in the conversation (even those who don't know about the game) is assigned a face. Players who engage in dialogue with others get points and see their face move closer to the center. Players who are seen by *Agonistics*

as addressing a theme the group is discussing cause that theme to be shown and a sentence from their message to be highlighted at the top of the screen. As Sack's catalog description states, these rules have certain results:

Winning players will be those who can (a) build a large coalition by engaging a number of people in dialogue; (b) promote a desired set of themes of discussion that are taken up by others in their posts; and, (c) articulate an influential opinion about the themes of discussion.

For Sack, *Agonistics* is an attempt to make a playable game that operates in a manner inspired by those philosopher linguists (he cites Chantal Mouffe, Gilles Deleuze, Bruno Latour, and others) who imagine democratic debate in gamelike rather than warlike terms. Obviously, such a game can never satisfy demands for a "quantifiable outcome," though it is certainly playable. I am particularly attracted to this project because Sack shows a way to create meaningful multiplayer language play. *Agonistics*

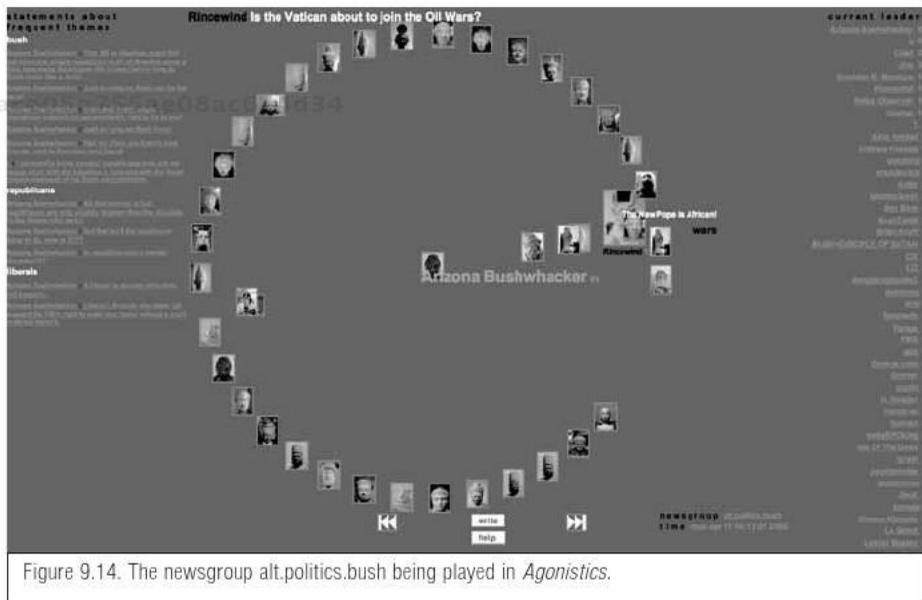


Figure 9.14. The newsgroup alt.politics.bush being played in *Agonistics*.

5. A feature that makes it possible to play *Scrabble* by treating the English language as a set of allowable codes rather than a means of expression.

operates using a model tied to the structure and meaning of language and discussion, rather than, say, using a particular dictionary as a razor for determining legal and illegal moves.⁵ It is a far cry from using words as a skin for objects governed by spatial logics.

Deikto: A Language for Interactive Storytelling

Crawford, whose concept of “process intensity” has been a touchstone for this volume, is a towering figure in the field of computer game design. He authored one of the first books on the subject (Crawford 1984) and founded the field’s most significant conference. But in the 1990s he abandoned the field of computer games, instead turning his full attention to the pursuit of interactive storytelling. The basis of his current approach is a new language, designed for play.

This language, Deikto, is meant as a response to the limited number of “verbs” presented in most interactive experiences. For example, in many graphical games, the only available verbs are “move” (accomplished by the means of navigation) and “attack” (carried out using the available weapon, toward whatever is in its path). Given that Crawford believes that interactivity is the essence of what computation makes possible for media, and that being able to act as a story’s protagonist is the fundamental experience that interactive storytelling should offer, it was necessary for him to find a way for players to express much more complex, nuanced verbs.

The designers of *Façade*, as discussed earlier, sought to address a similar problem by allowing players to type whatever they please. The *Façade* system then attempts

to interpret this, based on the current context, as one of a set number of discourse acts, which alter the state of the underlying system in particular ways. This opens up opportunities for a wide variety of player performance, but it also results in two types of failures. First, the system can fail to appropriately map player utterances to discourse acts. Second, the system may not be able to offer any meaningful response to even accurately recognized utterances, with results that are as unsatisfying as those from recognition failure. Crawford was determined to avoid failures of both sorts.

His alternative, Deikto, is a simplified language, with sentences built by players as their way of making moves in the game. Building such sentences creates a much larger possibility space for expression than what is found in standard games, even if the space of what can be said (made to appear on-screen) is smaller than that experienced when playing *Façade* or chatting with *Eliza/Doctor*. More important for Crawford, the options presented to a player building a sentence are those that have meaning in terms of the underlying system. This prevents unsatisfying failures and also provides a way for authors to guide players toward the actions that will produce meaningful system responses—in a much more subtle way than the techniques used in most games (e.g., graphical highlighting of particular objects or hints embedded in quest journal descriptions).

The original design of Deikto was diagrammatic, with lines representing connections between major elements and each utterance capable of expanding into a complex branching structure. This could produce sentences that were quite difficult to read.

As of this writing, Crawford has developed a new “linear Deikto” that intersperses connecting words—not interpreted by the system, and not necessarily grammatically correct—between elements constructed through menu-based interaction with the system, resulting in a more audience-interpretable text. This is critical in part because system-controlled actors also express their storyworld actions in Deikto, so the time needed for audience members to interpret Deikto determines some of the pacing of the story experience. Crawford’s team provides an example of linear Deikto, presumably constructed as part of the work on an updated version of his landmark game *Balance of Power* (the new version is being developed using the Storytronics system, for which Deikto is at the heart). I have rendered Deikto elements in brackets:

USA acts: [You] [start with this approach] to accomplish [Afghanistan hand over bin Laden]

You decide: [I] [offer deal] to: [United Kingdom] in which [I] promises to ask [China] to do this:

[China recognizes Taiwan] in return for which [United Kingdom] agrees to ask [Afghanistan] to do this: [Afghanistan hand over bin Laden]

This can be seen as a type of simulation—but it turns a number of common assumptions about the mapping between simulation and language on their heads. It marks a foray into a middle ground, a language designed both for audience interaction and internal operation. In taking this position, it offers the enticing possibility of audience members constructing complex actions without falling into the *Eliza* effect troubles that plague *Façade*—

or indeed, the “search for the verb” problems that have limited the appeal of much textual interactive fiction. It is certainly too early to say how influential Crawford’s particular system will be (I write on the verge of its planned release), but I believe the ideas it explores will be crucial as we move forward.

***Griot*: Call and Response Narrative**

Deikto could be seen as one example of a widespread mode of playful linguistic interaction: “call and response.” Harrell pursues a different form of call and response in a strand of his digital media work. He relates this work’s form of play to turn-taking traditions ranging from signifyin(g) and Capoeira Angola songs to Japanese linked poetry and Oulipian language games. In each of his works in this area, the system and audience co-create short texts—with the audience’s contributions brief, and indeed often only a word. Such audience “calls” serve not only as part of the surface experience but also shape the operations of the underlying system, after which the system responds with another text, and frequently another invitation for the audience to make a call in turn.

The underlying system that Harrell uses to support these experiences is called *Griot*. It is designed to make it possible for authors to craft interactive computational narratives and new forms of what he terms “polymorphic discourse” based on the blending of thematically important concepts specified using Joseph Goguen’s theory of algebraic semiotics. An author specifies theme domains as sets of axioms, phrase templates as possible output texts with open slots for the results of conceptual blends, and a narrative structure as a sequence for the

6. As discussed below, he is currently exploring alternative interfaces in which alphabetic text is not central. Harrell and his students apply the system not only to texts, but also to user identity representations across media (e.g. avatars, characters ...

Notes continued at end of this chapter.

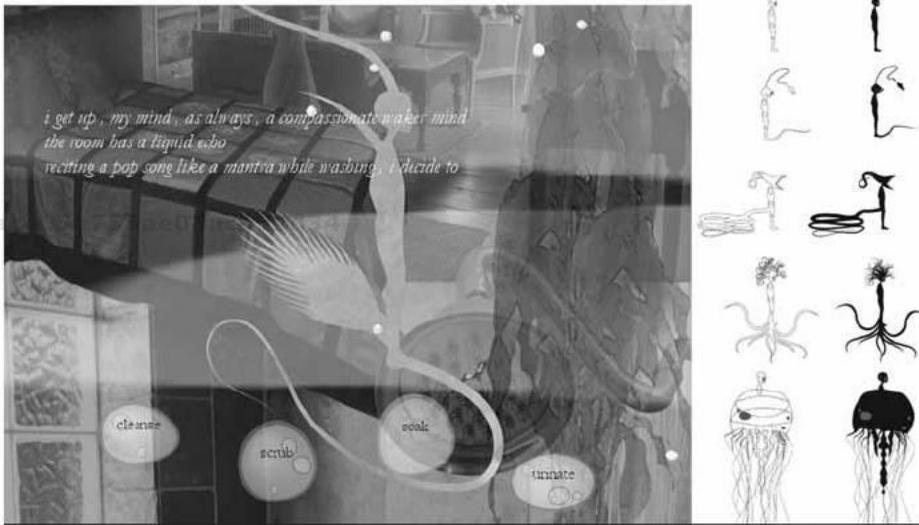
experience defined in terms of the number and types of narrative phrases that can be employed at a given stage. Each audience call is scanned for keywords, which are used to select the theme domains blended in each system response. His early examples focused on textual output, but his framework supports a broader definition of *text* that includes output in multiple modalities—for instance, a type of visual poetry described in this section.⁶

At the heart of the *Griot* system is his Alloy blending algorithm. This algorithm takes as input two algebraically represented concepts to be blended: a base concept that represents the shared structure between those concepts, and mappings between the base concept and other input concepts. It finds all of the possible ways to combine those concepts using a set of principles based on how the concepts are structured, how they map to each other, how much of the input concepts are preserved in the blend, and whether data types change or not as they are blended. *Griot* uses a subset of the full power of the Alloy approach—for example, often just outputting the most optimal blend—and Harrell continues to develop ways to use the generative power of blending to allow texts to create new, thematically constrained meanings on the fly (Goguen and Harrell forthcoming).

We can see specific examples of this in Harrell's variable narrative poem *The Girl with Skin of Haints and Seraphs* (2007). The narrative structure dictates that each poem begins with an abstract phrase and an orientation phrase. Possible opening phrases include “She begins her days (verb-clause)” and “Each morning foes called her (singular-noun)!” The domains include Africa, Europe, angels, demons, skin, and whiteness. The demons keywords

include demon, pitchfork, devil, and brimstone. Demons axioms include “Person:demon spawns Emotion:fear” and “Object:fire burns Person:soul.”

Harrell, in some cases working with his graduate students, is also extending this work into other types of material. In *Loss, Undersea* he creates a multimedia interactive narrative poem in which a character moving through a standard workday encounters a world submerging into the depths (figure 9.15). In work with Kenny Chow, *Griot* is being employed in a system for “visual renku”—in which the call and response can include some of the imagistic, diagrammatic, and conceptual connections that can be evoked between Chinese characters. In work with Jichen Zhu, *Griot* is part of a storytelling system that invites the audience to call out aspects of how the world is perceived



9.15. In *Loss, Undersea*, as a user selects emotion-driven actions for the character to perform, the character dynamically transforms—deep sea creature extensions protrude and calcify around him—and poetic text is generated narrating the loss of humanity and the human world undersea. *Griot* is used to implement blending in several ways. Structural blending (from algebraic semiotics) is the integration of multimedia elements according to diagrammatic and visual meanings, whereas conceptual blending (from cognitive linguistics) is the integration of logical data structures representing concepts.

and engaged by the focalized character, which not only shapes actions in the world but also the extent and tone of stream-of-consciousness character daydreams.

What I find fascinating is Harrell's work to make linguistic play an engagement with underlying conceptual models—while also organizing his system so that both the linguistic and conceptual structures are individually authored for each work. While his conceptual blending methods may require that concepts be represented in similar form in each work, this seems much more promising than assuming the same concepts will be appropriate for every work (an assumption that seems to foreclose one of art's most important possibilities).

Process and Fiction

The research and writing for this book has given me a much greater interest in the possibilities for simulation-oriented approaches to process-intensive fiction. Perhaps because of my training as a fiction writer, I was more inclined to image stories with fixed events, in which computational models opened variability related to language and theme. But the richness of the simulative tradition—much of it little discussed in the books I found on electronic literature and game design, both of which tend to focus on approaches with low process intensity—has convinced me of the potential of this direction. In the meantime, I hope that this chapter has convinced some who have largely seen computational approaches to fiction as synonymous with simulation that a much wider range of possibilities awaits our exploration.

Notes

1. Within this phrase I'm subsuming a discussion, around texts with instrumental qualities, that has used a variety of loose terminology. I'm also abandoning an earlier terminological distinction from my previous writing on this topic between "instrumental texts" and "textual instruments" (2003a, 2005). But the distinction between the ideas is still present here, simply without the confusingly similar terms.
2. I originally referred to these spatial logics as graphical logics. As I recall, the change in terminology was suggested by Espen Aarseth.
3. This is not to say that all combinations of graphics with spatial logics work equally well. For example, Jesper Juul (2005, 13–15) amusingly demonstrates that the logics of *Space Invaders* work well with both academic theories and television personalities—but this assumes that the relationship with both is antagonistic. And of course, the specifics of the fiction depicted by a set of graphics have a great impact on how we interpret spatial logics. Raph Koster (2004, 167–169), for example, demonstrates that the famously abstract game *Tetris* can be turned into a disturbing experience with graphics depicting human bodies being dropped into a gas chamber or mass grave.
4. In *Conversation Map*, the last two of these use the same WordNet tool employed by *The Impermanence Agent*. Yet the similar functions of *Agonistics* steer away from WordNet's synonyms (and other conceptual groupings), instead tracking the exact words and phrases introduced into the conversation.
5. A feature that makes it possible to play *Scrabble* by treating the English language as a set of allowable codes rather than a means of expression.
6. As discussed below, he is currently exploring alternative interfaces in which alphabetic text is not central. Harrell and his students apply the system not only to texts, but also to user identity representations across media (e.g. avatars, characters, and profiles). In other work, he is exploring possibilities for visual iconically-oriented, game-like interfaces that could be used to pass meaningful keywords to the system based on analysis of user behavior, with or without the user's awareness

7c4fbd6fd6face05a755ae08ac6a4d34
ebrary

7c4fbd6fd6face05a755ae08ac6a4d34
ebrary

7c4fbd6fd6face05a755ae08ac6a4d34
ebrary

7c4fbd6fd6face05a755ae08ac6a4d34
ebrary

Chapter 10 Conclusion

This book is certainly about the topics named in the subtitle—digital fictions, computer games, and software studies—but most of all it is about *expressive processing*.

Over the course of this book I have approached this topic from three perspectives. One perspective is oriented toward digital media creation, focused on the potential of understanding computational processes as a powerful tool for authorial expression. Another is oriented toward the critical study of digital media, focused on understanding the aspects of works that aren't apparent on the surface. Finally, I have also taken a political perspective, oriented toward the political stakes in some works and the political lessons to be learned from others. Here, in the conclusion of *Expressive Processing*, I will revisit each of these perspectives, bringing together ideas from throughout the book and suggesting some future directions.

7c4fbd6fd6face05a755ae08ac6a4d34
ebrary

Authors

We have reason to discuss digital media because of the already demonstrated ability of computational processes to define compelling new media forms—from the first-person shooter to the collaborative wiki. The specifics of the processes used to craft such media can shape the audience's experience as fundamentally as the specifics of the images used in a motion picture, if not more so. The harnessing of such specifics for authorial expression is one of the senses in which I mean the term *expressive processing*.

7c4fbd6fd6face05a755ae08ac6a4d34
ebrary

I am particularly interested in the potential of expressive processing within the realm of fiction: in creating flexible models of story events, producing a wide variety of language for characters and narration, and also for nonsimulative experiences that make language itself playable or enact procedural transformations that amplify fictional themes. As a fan of computer games, which have greatly developed the expressive simulation of space, I'm particularly interested in the possibilities of flexible models of fiction in this area.

Consider the open world computer role-playing game. In a game like *The Elder Scrolls IV: Oblivion* (Rolston, Howard, Cheng, et al. 2006) the world is simulated spatially, as in many other games. But it is also simulated in other senses. As I move through the world I collect ingredients I can use to make potions. As I make potions my skill in this area improves. As my skill increases I learn more about the possible effects of different ingredients. I can use the resulting potions or sell them into a simulated economy. What I get for selling them is connected to my abilities as a merchant (which get better the more I use them) and the simulated feelings other characters have about me (which can be altered by a conversation-simulation minigame as well as my faction affiliation, whether I'm holding my weapon unsheathed, and so on). And this is only one slice through the types of intersecting activities supported throughout the many spaces of *Oblivion*. In contrast to this vast, explorable world—through which players may move in many ways, with many different goals and paths toward them—the structure of the game's story is rigid. As Ken Rolston, *Oblivion's* lead designer, writes:

Exploring has more genuine suspense than following quest stages. There are so many

directions to turn, so many people to talk to, so many holes to crawl into, so many creatures and malefactors to chase after and righteously (or foully) slaughter. By contrast . . . I am forever running up against the boundaries of the plots I'm served, and disappointed in the choices of dialogue lines I'm picking from, and the avenues of inquiry I'm offered. I'm always conscious of the ways the characters and plots limit me—but in the choices of where to go and what to do when I'm exploring, the boundaries are less chafing and frustrating. (2009)

7c4fbd6fd6face05a755ae08ac6a4d34
ebrary

One might address this problem by simply abandoning any hope for fulfilling fictional experiences. Yet in fact, many players of role-playing games turn to this form precisely for an experience of fiction that is more developed and successful than that found in many other game genres. Rather than abandon fiction, it seems a better solution might be to begin to integrate narrative movement into the simulated world, using techniques of the sort explored during decades of research on story generation and interactive drama.

The same might be said of the ways that games like *Grand Theft Auto IV* (Benzies, Garbut, Fowler, et al. 2008) are “incoherent” in Jesper Juul’s sense (2005, 123–130): their events make sense only with reference to the game rules, rather being explainable in terms of the fictional world. One can experience such incoherence from the outset of *Grand Theft Auto IV*. For example, in the first major mission the player must deliver his character’s cousin to a card game, be on the lookout for loan sharks, and then drive the getaway car after they appear. If the player fails by getting shot by the loan sharks, his character winds up on the sidewalk outside a medical center, having been charged for services. The player needs to demonstrate the

7c4fbd6fd6face05a755ae08ac6a4d34
ebrary

7c4fbd6fd6face05a755ae08ac6a4d34
ebrary

1. The *uncanny valley*—a term coined by Japanese roboticist Masahiro Mori—indicates a phenomenon in representations of human characters. As representations approach reality they appear increasingly familiar up to a ...

Notes continued at end of this chapter.

skills required in the mission, so the game encourages the player to try again. If the player tries again, exactly the same conversation leads into exactly the same mission. It's as though one jumped back to an earlier point in the story. But the money for the medical treatment is still gone—which can only be explained by viewing the cost of the treatment as a game-rules punishment for failing the mission, rather than the character's wallet somehow existing in a different time/space continuum. A more coherent world, and a more interesting experience for players redoing missions, would come from abstracting the elements of the mission (driving to, looking out, getting away) and using computational processes to produce a series of missions, coming one after another in fictional time, until the player succeeds in demonstrating a sufficient level of skill.

Both of these would represent steps toward addressing what I call the *unimplemented valley*. I suggest this term as a deliberate reminder of the *uncanny valley*—a phrase that I began to hear more often around the time that Sony was advertising the graphics processor in the PlayStation 2 as an “emotion engine.”¹ The most common discussions of the uncanny valley suggest that our ability to identify with human characters is closely tied to their graphical representation. The idea of an emotion engine suggests that greater graphical fidelity is the key to greater emotional involvement. But the non-player characters who seem to actually elicit the greatest emotional engagement don't seem to be the ones with the best graphics, or the best just shy of the uncanny valley.

Consider *The Sims*. As I discussed in an earlier chapter, this game exhibits a remarkable closeness between the

surface representation and the underlying simulation. I believe that we can form emotional attachments to Sims in part because they don't speak English. They speak Simlish, made up of symbols representing aspects of the underlying model of interests for Sims. Their graphical representations are actually rather simple, and their animation sequencing can be problematic, but they appear genuinely responsive to changing circumstance—within a limited range that is continually telegraphed to the audience by the design of the system—and this is highly engaging.

2. Andrew Stern has called this the “big hair era of games.”

The emotion engine view of characters in digital media is the opposite. It is focused on giving characters more expressive faces, body models, and movements.² A mildly enlightened cousin of this view also wants to give characters more compelling things to say. But there is little said about the underlying models driving these characters, and the assumption generally seems to be of models simpler than those in *The Sims*. As a result, developing these more expressive faces is just creating a larger and larger gap between the surface and the underlying model, between the appearance of response and the actual ability to respond. Even if this vision succeeded completely at its aims, the eloquent lines expressively performed by these characters would have less meaning in the system of the resulting game than the iconographic Simlish of the Sims.

I propose an alternative. Our research in real-time graphics and natural language generation should be driven by dual aspirations. Not only should we aim for engaging expression but also for expression that communicates the evolving state of the underlying system. We should strive for the closeness of surface and simulation achieved by *The Sims*, but while moving

3. Moving beyond the most basic aspects of life is one of the stated goals of *The Sims 3*, which is not yet released (as of this writing) but is promising, in part because of the involvement of Richard Evans (whose work on *Black & White's* AI was noted earlier).

both forward and sideways toward elements of human life other than the most basic.³ Obviously, *Grand Theft Auto* missions that are represented within the simulated world and generated by system processes (rather than hard coded as data) would demonstrate only a small step in this direction—but every step matters.

More generally, just as the simulation of space opened up a wide variety of new game possibilities, so flexible models for story structure, character behavior, and fictional language have the potential to develop experiences that can't be fully imagined (and certainly can't be compellingly executed) using today's common approaches. *Façade* is one example of this—an interactive drama that would fall flat if implemented using quest flags and dialogue trees. As the previous chapter indicates, I am also interested in radically different, nonsimulative directions for expressive processing in the realm of fiction and language.

I believe we are now developing a generation of authors who understand both processes and media, who are ready to begin to seize these possibilities. Given my interest in turning our attention to processes, I find a particularly encouraging sign in the rapid development of the community around the Processing programming language initiated by Casey Reas and Ben Fry (2007). Their book begins with the assumption that computational processes are media-authoring tools—a place I hope more introductions to programming will start in the future.

Critics

Just as authors can seize a great opportunity by grappling with the newly available power of computational processes (an approach much less explored than crafting static

words, sounds, or images), so critics of digital media have an opportunity to take their analysis further by turning their attention to the processes that enable such works to operate. The interpretation of processes, however, will require a new set of conceptual frameworks. Most models for digital media do not make a place for processes.

The major exception for digital fictions—Espen Aarseth’s traversal function model—takes processes into account, but positions all of them as means of converting strings in the system into text experienced by audiences. This variety of textual transformations could be seen as a family of operational logics. My proposal is that, rather than assume these logics are central to all digital media, critical engagement should begin by identifying the actual logics operating in a particular work and the relationships between them. There are a number of reasons for this, not least what it makes possible for critics to understand about the relationships of a work’s particular processes to histories of thought and communities of practice. For example, in this book I demonstrated how such an approach leads to a different interpretation of *Tale-Spin*—based on the connection of its processes and structures with the scruffy tradition of cognitive science and artificial intelligence—than those from critics such as Aarseth and Janet Murray.

I understand such connections as one of the things that processes can “express.” At least as important, I am also concerned with the expression of ideas through processes, of which the author may or may not be aware—and which the author may or may not intend. The authors of *Brutus*, for instance, write of the significance of their logical model of betrayal to the system. But a careful examination of the system’s operational logics

reveals that the model of betrayal is in fact unnecessary in the generation of stories in the actually implemented version of *Brutus*. The concept of story generation that the system's processes express runs directly counter to the authorial account of the system. Not all cases are this dramatic, of course, but being able to interpret what processes express in this sense is another goal I have for the operational logics approach. More generally, I hope an examination of operational logics will become a useful tool in the development of the field of software studies as a whole, joining approaches discussed earlier in this book, ranging from Chris Crawford's process intensity to Ian Bogost's procedural rhetoric.

In addition to outlining and demonstrating the general approach of examining operational logics, this book has also presented three effects that can arise in the relationship between audience, surface, and system processes: the *Eliza*, *Tale-Spin*, and *SimCity* effects. While I have chosen to largely discuss these effects with reference to the systems for which they are named, they are general effects that one can identify in many types of digital media.

The *Eliza* effect is an initial illusion of system intelligence based on audience expectations. The illusion can be maintained by severely constrained interaction. During less restricted, playful interaction, on the other hand, the illusion breaks down in a manner determined by the shape of the underlying system. While the system for which this effect is named, *Eliza/Doctor*, is not now as ubiquitous as it was in my early days of computing, the experience of the *Eliza* effect is familiar in a variety of contemporary forms, which need not be textual. Many game players, for example, are accustomed to NPCs that

at first seem to have a general-purpose competence in moving through the game's simulated space. But even in the greatly simplified spaces presented by computer games, this illusion of general competence breaks down regularly—and NPCs end up caught in level geometry, failing at pathfinding when presented with objects as simple as ramps and so on. The specific breakdowns are shaped by the underlying systems of spatial simulation and character navigation. They could be avoided in a game that wanted to simplify space and movement sufficiently (obviously, they couldn't exist in a game like *Myst*), but relatively free-form interaction is also often important to creating the initial illusion. Luckily, in many games the illusion of NPC spatial intelligence is not so central to the experience that breakdown significantly diminishes the enjoyment of play.

The *Tale-Spin* effect, on the other hand, creates a surface illusion of system simplicity, without providing a means for interaction that would allow audiences to come to understand the more complex processes at work within the system. As a result, elegant and thought-provoking systems can appear arbitrary and shallow. This effect is perhaps most familiar to those who work in computer science research and/or digital art. For instance, I have written in a forthcoming book chapter about the intriguing project titled *Amy and Klara* by Marc Böhlen (2006a). This work takes the form of two pink boxes from which large robotic eyelike speakers emerge. Then synthesized speech begins, with one robot commenting on an article from *Salon.com*. This quickly devolves into an uninteresting fight—and audiences can lose interest. But when one begins to look at *Amy and Klara* as a system,

with information provided by Böhlen (2006b), further elements become apparent. First, through slots in *Amy and Klara*'s boxes, two cameras look at each other. Second, each robot also houses noise-reducing microphones. The robots of *Amy and Klara*, in other words, not only “speak”; they also “see” and “listen.”

In addition, the robots of *Amy and Klara* “read.” Each performs a statistical evaluation of the contents of *Salon.com*. This is the starting point for their dialogues, as the Amy robot chooses a topic identified by her reading of *Salon.com* on which to offer a comment. A text-to-speech system turns Amy's comment (assembled by an agent architecture in part based on AIML) into sounds sent through her speaker. Because the robots do not share data, the Klara robot only “hears” Amy's comment through her microphone and must use automatic speech-recognition technology to turn it into text. Given the limitations of software systems for text-to-speech conversion and automatic speech recognition, misunderstandings begin almost immediately, thereby giving rise to disagreements. The “fights” seen by the audience, then, are just epiphenomena emerging from a much more interesting set of interacting processes. The operations of this assemblage of processes express something about recognition and misrecognition, communication and miscommunication, mechanism and emotion. But it is an expression to which the piece's audience experience provides almost no access, displaying the classic elements of the *Tale-Spin* effect.

The *SimCity* effect, in contrast to the other two effects, leads to audience understanding of the operations of an underlying system. It is most interesting for works with a relatively complex set of internal processes—often a