

A review of Noah Wardrip-Fruin's *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*

Doug Reside <dreside_at_umd_dot_edu>, University of Maryland, College Park

Abstract

The need for greater software literacy is a pressing problem, but one still not universally acknowledged even among those working in new media and digital humanities. In *Expressive Processing*, Noah Wardrip-Fruin makes a compelling case that software studies as a field is not only an interesting avenue of research for new media specialists but also should increasingly be a basic activity of educated citizens in a 21st century democracy.

Although Noah Wardrip-Fruin's *Expressive Processing: Digital Fictions, Computer Games, and Software Studies* is probably best known at present for the author's attempt to provoke a radically open form of peer review by submitting portions of his manuscript to the blog Grand Text Auto in advance of the final submission to MIT Press, the most potentially transformative legacy of the book is in its compelling call for universal software literacy. For those of us in the humanities who find ourselves constantly making the argument that a basic understanding of algorithmic thinking is an essential literacy, not just for scholars in the digital humanities (dishearteningly even this is not always a settled question), but for all educated citizens of the 21st century, Wardrip-Fruin has given us an arsenal of rhetorical firepower and a powerful set of examples for how one might teach algorithmic literacy across the curriculum without delving into the syntax of any particular programming language. 1

This approach to studying software by looking not at the actual lines of C++, Lisp, or JavaScript code but at the general algorithm (abstracted to a level even above what computer scientists and programmers describe as pseudo-code) is what Wardrip-Fruin means by his titular phrase, "Expressive Processing" — a phrase which he derives from the multifaceted nature of his method which looks both at the user experience (the "expression" of the algorithm) as well as at the "process" (the algorithm itself) that generated it. Wardrip-Fruin builds upon the foundation laid by Lev Manovich, who, in his 2002 book *The Language of New Media*, suggested that the natural development of media studies in an age of "programmable media" should be "software studies" (a set of approaches which includes "expressive processing" but also code studies which, in Wardrip-Fruin's words, looks at "the specific text of code" written by developers). 2

Expressive Processing fulfills and extends the promise of Manovich's ideas, putting the theory into practice through a set of case studies of the artificial intelligence engines of a dozen or so software programs that might be loosely called "games." That the first real example of a software studies approach comes out of game studies is both to be expected and (somewhat) regretted. On one hand, games of the sort Wardrip-Fruin examines are a medium for storytelling and character creation, and as such are natural extensions of the work of previous literary and media studies scholars and thereby set up a convenient space for humanities scholars and teachers to consider the important cultural and technical issues raised by Wardrip-Fruin in an environment more familiar than, for instance, an analysis of the software that drives Walmart (one of Wardrip-Fruin's suggestions for another work of software studies scholarship). Unfortunately, like graphic novels and musical theater, the genre is still too easily dismissed as popular entertainment by too many of those who most need to hear Wardrip-Fruin's arguments. This is not really the fault of the book, but I do question the wisdom of selecting, as early as chapter 2, such an exceptionally "geeky" title as *Star Wars: Knights of the Old Republic* to demonstrate a relatively common problem in massively multiplayer role-playing games. 3

Even to sympathetic audiences, the transposition of the multimedia experience of a modern computer game to text can, almost inevitably, generate potentially confusing prose that spends as much time describing as it does analyzing; the small black and white screenshots do less to illustrate the author's point than they do to highlight the limitations of the print monograph for scholarly discussion in the modern age. For this reason, Wardrip-Fruin's discussion of relatively simple, text-based programs (such as the opening chapter's analysis of "Eliza," the famous electronic psychiatrist program, or the semi-randomized story generating program TaleSpin) are generally more compelling than his attempts to describe complicated, modern multimedia games like *Prince of Persia: The Sands of Time*. I fear these obstacles may cause some readers, already perhaps uncomfortable with games studies, to prematurely abort before reaching the real gold that shortly follows.

4

Still, there truly is treasure buried in this land of geekdom, and not just a few nuggets, but enough to lay the foundation of an entirely new scholarly approach for the digital humanities. Like Nick Montfort and Ian Bogost's work on Platform Studies, Wardrip-Fruin demonstrates by way of example a new way of reading new media. If Manovich drew the map, Wardrip-Fruin has opened the mine, and what may be extracted will benefit not only those working in digital humanities or new media but scholars across the curriculum. In the fifth and sixth chapters, Wardrip-Fruin makes a compelling case that software studies as a field is not only an interesting avenue of research for new media specialists but also should increasingly be a basic activity of educated citizens in a 21st century democracy. Wardrip-Fruin eloquently illustrates how the contemporary human experience is, in large part, shaped by the algorithmic processes that drive our society, algorithms that determine, as Wardrip-Fruin observes, everything from what Amazon.com recommends that one buy next to whether one is included in a terrorist watch list. To the degree that the sort of algorithmic literacy practiced by software studies is anything less than universal there will be, in the words of Ted Nelson whom Wardrip-Fruin quotes, a digital "priesthood" that rules over the rest of the populace with power that cannot be questioned or criticized. The power of *Expressive Processing* is that it not only eloquently restates the problems Nelson observed, but actually helps to solve it by offering up a truly readable generalist introduction to the field of artificial intelligence that could productively be assigned to both computer science and humanities students at both the undergraduate and graduate level. By explicating mostly inconsequential, but simple and open source, algorithmic processes such as "Eliza," Wardrip-Fruin provides what he calls "legible examples" from which the educated reader can reason by analogy to understand how larger, more powerful, and generally closed-source algorithmic processes function, and concomitantly, question them.

5

It is on these points that I find myself citing this book most frequently. The need for greater software literacy is a pressing problem for all of the reasons Wardrip-Fruin enumerates, but one still not universally acknowledged even among those working in new media and digital humanities. In too many digital humanities projects, scholars simply hire programmers to do *all* of their technical work (from outlining technical specifications and project management to coding), rather than learn, at even the most abstract level, how the processes that generate the "expression" they desire operate. Immediate consequences of this approach include rampant scope creep (because the effects changes to the "expression" have on the underlying "processes" are not well understood by the project director) and the tendency for projects to fade away after initial funding runs out. (If a scholar does not have the technical ability to maintain her own work, it is in danger of vanishing when the tithe can no longer be paid to the "priest.") If even the community of Ph.D. holding, multi-lingual digital humanities scholars is not expected to understand the technical underpinnings of the work for which they are often the leader of record, the hope of a software literate populace seems very far away indeed. The usual objections about lack of time express nothing so much as a lack of incentives, experienced as keenly by the assistant professor seeking tenure as by the English major just beginning his freshman year. Until granting agencies, hiring and tenure committees, and peer review panels reject as unqualified anyone without demonstrated software literacy, we cannot hope for a world in which the citizenry are able to evaluate whether or not the risks of false positives in a particular terrorist watch list generator outweigh the potential for increased national security.

6

At some level even the approach advocated in *Expressive Possessing* allows too much room for the technically recalcitrant. Although, for Wardrip-Fruin the particulars of variable names and code syntax are less important than a general understanding of how the algorithm works, it is not entirely clear how one comes to an understanding of the algorithm in one of Wardrip-Fruin's "legible examples" without the ability to actually read code (something Wardrip-Fruin clearly can do). Of course, one can rely on others to do this first level interpretive work, but here the metaphor of the

7

priest can be employed again, and arguably even more aptly than before. Perhaps programming knowledge is not a prerequisite for reading a monograph that follows the expressive processing approach (indeed, this book proves that it is not), but it seems that it must be a prerequisite to write one. Still, Wardrip-Fruin's approach does seem well-suited for this transitional moment before widespread software literacy is achieved, and to the extent that expressive processing (the approach) and *Expressive Processing* (the book) encourages greater literary among the scholarly community I cannot recommend both highly enough.



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.