Editors' Introduction to "Computational Fictions"

[Pat Harrigan](#)

2008-03-15

Editors' Introduction to "Computational Fictions"

What makes computer games different from other games? What makes *Pong* different from tennis, video poker different from traditional poker, or one of EA's *Lord of the Rings* computer role-playing games different from *Dungeons & Dragons*?

Of course, the differences are many, when considering particular games - for example, whether referees are employed, cards are used, or hard cover rulebooks of hundreds of pages are regularly published. But another type of difference runs consistently through all of these examples.

If we play tennis, we expect that all players understand the structure of play (e.g., who serves, from which side, in which order); it is we, or an umpire, who judges balls in or out of bounds and keeps score. In *Pong*, however, this is all determined by computational processes. At our weekly poker nights, we shuffle the cards into a random order, deal them out as required by the game variant being played, and judge each player's hand according to the rules; but a video poker machine takes on all these tasks itself. Playing *Dungeons & Dragons*, we utilize maps, character documents, rulebooks, dice, and ongoing discussion to simulate a virtual world and adventure within it; EA's *Lord of the Rings* games simulate everything within the software.

This is to say, non-computer games require that people understand them as a system, make judgments, and keep track of the game state. They require that players operate the game mechanics; therefore, the game mechanics are an important part of player experience. By contrast, computer games take on at least part of the mechanics themselves. Many computer games continue to function appropriately as a system even if the players completely fail to understand them - it's possible to play some computer games successfully based on an entirely mistaken understanding of how they function internally, even while the games themselves continue to enforce their own rules and structures. But players who significantly misunderstand non-computer games will almost invariably cause the game systems to operate incorrectly.

There are limits to what this proceduralism can accomplish. A competent group of *Dungeons & Dragons* players can simulate any eventuality and deal with any action or communication attempted by the players, while even the best computer RPG can barely prevent the dialogue of computer-controlled characters from becoming painfully repetitious. Any action too far outside those already imagined by a computer game's design team is usually either met with an uninteresting response or is simply impossible to carry out.

But this is only to say that given our current state of technological advancement, there are things that computers do poorly and things they do well. Take for example, Avalon Hill's *Squad Leader*, which simulates a WWII battle through a notoriously elaborate rules system.

Editors Pat Harrigan and Noah Wardrip-Fruin introduce the essays of the "Computational Fictions" section of *Second Person*, focusing on the conversion of human ludic interaction into computational processes - a necessary condition for computer games.

Tags:
[quantification](#)

Although the game has its adherents, most would admit that it pushes the bounds of what human memory and patience can accommodate.

A computer, on the other hand, can easily handle Squad Leader's complex mechanics (simulating terrain, movement, weapons accuracy and penetration, morale, line of sight, prep fire, defensive fire, advancing fire, close combat) and much more. A game like Gearbox's *Brothers in Arms* provides both a more complex simulation than *Squad Leader* and a player experience that requires no knowledge of how the simulation operates. Computer games can also surprise their players: rules can change, contents can be revealed, and other operations can take place without the players making them happen. The variety of contents can also be much greater than with non-computer games, as contents can be recombined (or even generated) algorithmically.

Of course, the real question is not, "Can computers reproduce or replace tabletop games?" Certainly computers can speed certain processes - *Diplomacy* may be easier to play by e-mail than by post - but that is not the limit of what can be done. Similarly, computer games may seem, at first blush, to be operating at a handicap: they lack the apparent access to human creativity and the context of social relations that contribute much of the power of non-computer games. But this is only as true as it is of other art forms - such as literature and film - which generally must create an audience experience on their own, rather than in combination with the efforts of a group of human players.

Rather than focus on comparisons between tabletop and computer games, the contributors to this section ask, "What types of new playable media are made possible through use of the computer?" In particular, they focus on playable experiences that develop characters and stories - undertakings that are usually considered the province of other primarily non-social art forms (again, such as literature and film). And this raises the necessity of seeing these works in more traditional aesthetic terms, rather than purely ludic ones.

That is to say, in addition to game design, in addition to interaction, in addition to the elegant design of their internal procedures, the works in this section demand we be prepared to consider works for their characters, fictional worlds, use of language, and so on. Helen Thorington's *Solitaire* may look like a card game, but it can't be played to win. Talan Memmott procedurally "re-authors" Rene Magritte, but not as a game license. And Stuart Moulthrop, in calling his *Pax* a "textual *instrument*," specifically points us toward an aesthetics of playing that isn't game-centric.

Of course, we can't ignore the fact that there are tabletop games that demand to be seen in these terms as well. The works discussed in this book's first section are examples, and there are many more. A game such as *Betrayal at the House on the Hill* isn't enjoyable because of its elegant gameplay. In fact, its many horror scenarios include some that are remarkably poorly balanced. Rather, the enjoyment comes from the way that its structures, as a work of playable media, are so well integrated with its horror genre content. It's not a finely structured game with horror "spray-painted" on, any more than *The Maltese Falcon* is a novel with noir detection lathered over it.

For some of the works presented here, the most effective way to understand their procedural, interactive, and aesthetic dimensions may be to see them as hybrids between computer and non-computer forms, or as extensions of aesthetic traditions into the digital realm. *Juvenate*, discussed here by Marie-Laure Ryan, can be seen as bringing together the forms of the artist's book and the computer-driven flowchart or maze. Mark Marino's *Twelve Easy Lessons to Better Time Travel* brings together a *Tristram Shandy*-style narrative with the forms of cross-media learning modules. Lev Manovich utilizes digital procedures to

combine and display traditional cinematic elements in new ways. D. Fox Harrell's GRIOT performances take place in both the digital and real worlds, using computational processes modeled on human conceptual blending to produce narrative poetry in response to user input.

An important component of almost any experience of story or role-playing, with or without a computer, is the presence of characters. As discussed in the introduction to the previous section, in a tabletop RPG, character is not a procedural issue. There are PCs, acted by players, and NPCs, acted by the gamemaster. On-the-fly interactions between these fictional characters is accomplished effortlessly, in what is for most of us the most user-friendly medium possible: human speech. Designers of computer playable media have to look for different solutions.

A brief example of one such approach is given here by Steve Meretzky. Meretzky has designed games for most conceivable media, but here revisits one of his most famous early creations, Floyd the robot, in Infocom's text adventure *Planetfall*. Floyd is arguably the first fully-realized NPC in a computer game; the nature of the character's scripting lent Floyd a perceived depth that had been lacking up to that point.

Lee Sheldon discusses the computer game adaptation of Agatha Christie's mystery novel *And Then There Were None*. In particular, he addresses the problem of how much information the suspects know, and how and why they might give it to the PC investigator. Sheldon's "suspicion meter" model is one method of addressing this need.

A longer discussion of a game scripting process can be found in Jordan Mechner's chapter on Ubisoft's story-rich platformer *Prince of Persia: The Sands of Time*. In scripting the events, dialogue, and characters for the game, Mechner drew on his experience as a screenwriter, and the result is, in fact, highly cinematic.

Michael Mateas and Andrew Stern, in their work *Façade*, have taken a different approach, attempting to be more "dramatic" than cinematic. *Façade* places the player in an awkward dinner party with a married couple, Grace and Trip, on the verge of a breakup. [1] What Grace and Trip say, and how they say it, is dependent on their "perception" of the player's attitude and what has already gone before in the conversation. This, in turn, feeds back into the continuing conversation and ultimately affects the outcome of the scenario; that is, whether Grace and Trip stay together or break up, and what they might learn about themselves and the relationship in the process. From the player's point of view (ideally), they are participating in a natural, if somewhat tense, conversation.

Another potential breakup is discussed in Robert Zubek's chapter on his *The Breakup Conversation*. Here, the user takes the role of one half of a relationship - the half that wants to break up, in fact. The game system plays the user's partner, and tries to prevent the breakup. Success and failure are determined by the outcome of the conversation.

Chris Crawford also discusses an approach to programming conversation - along with other story-determining actions - into a playable, digital form. His solution is to create a "toy language," with which players and NPCs can interact. He also demonstrates how this stripped-down language can be used to enable story creation, providing one method of reconciling the fundamental concerns of this book: role-playing and story.

Two long chapters, by Nick Montfort and Jeremy Douglass, point to the continuing interest in the Interactive Fiction (IF) form. With its roots in early text adventures like *Colossal Cave/Adventure* and the pioneering work of the Infocom writers, IF has evolved in a wide variety of directions. Montfort and Douglass provide close examinations of two works, and

also address wider issues with this form. The questions they raise, of simulation and of playing the "player character," resonate with many other topics discussed in Second Person. Emily Short is also present, to discuss her celebrated work of IF, *Savoir-Faire*.

Our final contributor to this section, Mark Keavney, has created the "City of IF" web site, although his definition of IF is quite different from that of Montfort and Douglass; stories at City of IF are generated through a community of authors and web visitors. [2]

Notes

[1] *In fact, the explicit model for Façade is Edward Albee's play Who's Afraid of Virginia Woolf?*

[2] *Astute readers will notice that "City of IF" lacks the sort of procedurality that unites the other works in this section. As such, it could be argued that this chapter more properly belongs in part III of this book. But Keavney describes his project as IF, and so we have placed his chapter among those of the other IF writers. Happily, Second Person is not an interactive work, procedural or otherwise, and the authors of this introduction reserve for themselves this measure of editorial fiat.*