

# 12

## Telling stories with maps and rules: Using the interactive fiction language “Inform 7” in a creative writing workshop

*Aaron A. Reed*

During two recent summer sessions at the University of California, Santa Cruz, I had the chance to design and teach a unique class combining a creative writing workshop with a seminar on reading and writing interactive fiction (IF).<sup>1</sup> In the course, I led an interdisciplinary mix of students from both arts and engineering backgrounds with widely varying degrees of experience in programming and creative writing through the 40-year history of digital literature, taught them some tools to create it, and challenged them to make a creative project both as fiction and process that told a story demanding participation from its readers. Both times I taught the course, students rose to the challenge, creating a variety of fascinating pieces including playable recreations of emotional traumas, philosophical explorations of the meaning of choice, and simulated conversations with dead relatives. Making interactive fiction requires a unique blend of creativity and logical thought, of writing prose and crafting code, and that duality makes it an exciting entry point into the creative writing process for a generation weaned on digital games. Crafting good interactive stories is a unique form of constrained writing that challenges students to strengthen their traditional writing muscles in unique

and surprising ways. I'd like to share some of my successes and failures in teaching the course using the language Inform 7 and why this nontraditional approach is worth considering for instructors looking to engage their students in novel ways.

## A beautiful precision of language

Why muddle programming together with writing? The two acts may seem to have little in common: the epitome of logical, orderly thought and the chaotic, freewheeling energy of raw creative process. But I've found many productive, surprising synergies between these two forms that can be explored together in IF projects. Both practices require a careful precision of language to achieve a desired effect, either perceptually (in a human reader) or mechanically (for a computer running a piece of code). Just as a misplaced word in a story can jar a reader out of a sense of flow and immersion in an imagined world, or radically alter perception of place or character, a misplaced word in a computer program can cause it to crash, misbehave, or perform in a drastically different way. Both novice writers and coders underplay the need to choose words with care. To explore this point in the course, we took a tour through progressively shorter forms of fiction, starting with microfiction of a few hundred words and moving down to "Twitter fiction" and six-word stories. These miniature forms make it clear how much the right word can convey, as in the classic Hemingway example, "For sale: baby shoes, never worn."<sup>2</sup> In a workshop session, students wrote sentences and then aggressively edited them to weed out unnecessary language, aiming for the sharpness of the six-word stories where each word contributes meaning to the whole. This precision of language is useful not only for creative writing but also for programming, where a misplaced word or punctuation mark can create not just aesthetic but functional problems.

Focusing on clarity of language is also important because IF does more than just tell a story: it creates an interface into a fictional world that the reader-player must understand and communicate with for a successful traversal of the narrative. While in theory any text can appear in an interactive story, in practice those words must also function to explain the environment and indicate how a player can interact with it. One of IF's unique charms (and archetypal frustrations) is that the verbs and nouns a given work understands are usually not explicitly revealed: instead, the player must learn them through experimenting with the story and studying its text. One of the key differences between a well- and poorly written IF is how successfully it leads the player toward productive actions (those for which the author has created a response). Like a magician subtly forcing a card on an audience member, the IF author

must do everything possible to suggest certain commands and undercut others, while still giving players the impression of free will and possibility. Consider this sentence from Eric Eve's IF *Snowquest*: "So far as you can see nothing grows here; the ground is quite bare apart from an abundance of snow lying deep and crisp and uneven."<sup>3</sup> While a seemingly casual description, the words carefully hint at the correct course of action: the player can find something buried under the snow if they dig. "Deep" implies there's enough snow to cover something; "uneven" and "so far as you can see" both suggest something might be hidden awaiting discovery. "Crisp" suggests an unbroken expanse waiting to be disturbed. The lack of any substantial noun other than the snow-covered ground focuses the player's attention; by not mentioning trees, pine cones, footprints, the sun, or any of hundreds of other things that might exist in a real-world location, the author signals that these things are not an important part of the story world, and players will probably not be successful if they try to interact with them.

At the same time, IF authors share the desire of any writer to create beautiful prose and compelling stories, so the text's functional purpose must be woven into its aesthetic goals, even if these at times seem at cross-purposes. We might productively call this a form of constrained writing, like constructing a sonnet or writing a story without using a certain letter: careful work is required to produce a result both aesthetically pleasing that also follows the rules of the form. Forcing writers to think around constraints can result in both a heightened awareness of the act of construction and an output that breaks from the writer's familiar style. By asking authors to continuously work to craft a specific mindset in the player, IF encourages the kind of intentional thinking that is just as useful in traditional writing, where helping the reader understand a character or concept can require equal care and precision.

## IF and Inform 7

It's worth contextualizing the history of this form of writing in a little more detail before moving on. Interactive fiction as a mode of storytelling has been around since the 1970s and is probably still most recognizable as the "text adventures" *Zork* and *Witness*, popular on early home computers. After a commercial heyday, the medium was rediscovered and reinvented by nostalgic hobbyists, writers interested in programming (and vice versa), and more recently, academics, all of whom have helped shepherd the form through various waves of experimentation and innovation, trying to discover how adding interactivity changes an encounter with a textual narrative. At its core, IF relies on a call-response form, where a narrator tells a story happening to a central character (usually through second-person present tense text) and

readers must frequently give instructions as to what they'd like that character to do next. IF has been placed in the tradition of the literary riddle,<sup>4</sup> a fragment of story that only becomes complete when the listener supplies the correct solution. More broadly, Umberto Eco classifies all such artworks as "open works, which are brought to their conclusion by the performer at the same time as he experiences them on an aesthetic plane . . . his comprehension of the original artifact is always modified by his particular and individual perspective."<sup>5</sup>

Having experimented with the form for years and produced some major works with it, I wanted to find a way to share the sense of discovery and exploration that diving into this unmapped territory awoke in me.

While the course began with hypertext projects created in Twine, a popular tool for making link-based fictions,<sup>6</sup> we spent the bulk of our time with a more powerful program called Inform 7.<sup>7</sup> Both a programming language and a tool for working with that language, Inform is designed specifically to create IF. Writers describe an environment (a series of connected locations containing objects), a set of actions that can happen in that environment, and rules defining behaviors between, or cause and effect relationships for, the characters and objects in the world. The uniquely literary character of these environments and rules can go beyond simply describing scenes that could be visualized graphically, potentially making use of the whole range of techniques available to traditional literature. IF has made explorable inner-emotional landscapes, subjective realities filtered through an unusual or unreliable narrator, or environments rife with wordplay. An example of this is Emily Short's *Counterfeit Monkey*,<sup>8</sup> which imagines a world where letters can be inserted or removed from items to alter them: the title references a scam involving a monkey and a "k-remover," typical of the delightful lateral thinking found throughout.

Inform is an unusual programming language as its code reads like English sentences, rather than the more opaque input of a language like C++ or Java. Writing "Al's Bistro is east of Downtown" establishes two locations and a spatial connection between them; adding "A grilled cheese sandwich and a half-empty glass of beer are on the table in Al's" decorates one of these places with incidental details. More complex sentences can create rules and behavior, such as "After insulting Al, now every person in the bistro hates the player." These sentences are still code—Inform can't understand arbitrary English—but the approach allows example programs to be understood much more easily by novice programmers and creates a pleasing symmetry between the language of input and the language of output. One of the interesting side effects of this playful intersection of words and logic is that poems can be written in Inform 7 that compile into playable simulations of themselves, such as this limerick:

The Hole Below is a dark room.  
The description is "Cavernous gloom."  
The lamp is in Seoul.  
Before going in hole,  
Instead say "You will break your neck soon."

In addition to its interesting intrinsic properties, Inform is well suited for classroom use. It's freely available and runs on any modern operating system, eliminating the need for all students to have access to the same type of computer. The Inform editor is easy to use and simplifies much of the complexities of compiling and releasing projects; it can easily output a story and its source code as a set of styled webpages a student can upload or submit. A large and active community online uses the language and answers questions about it, and a great many tutorials, references, and other resources are available online.<sup>9</sup> Finally, Inform is an established, stable platform that has been around for a decade in its current format and rests on foundations that have been largely unchanged since the 1980s, so it's not likely to disappear within a semester or two. While the full language includes a lot of powerful functionality, its core features are enough to produce playable textual environments within minutes of getting started, while still leaving headroom for more advanced students to tinker and explore more complicated features.

## Strategies for character-driven interactive stories

What kinds of stories can students tell with IF? One important way the form differs from traditional fiction is in its presentation of character. While humans excel at making up imaginary people, computers don't; it's much easier to create a convincing textual simulation of a static object or landscape than of a living, acting, and reacting person. While this is still an evolving problem for the field of interactive storytelling, authors have nevertheless managed to create meaningful, moving, and effective interactive stories that, yes, are about people anyway. Since some of their strategies can be counterintuitive from a traditional creative writing perspective, it's worth exploring a few of the more common approaches to characters in IF.

The most familiar metaphor is probably *story as tableau*, where the player controls a character investigating the physical remnants of a story that happened in the subjective past. The story of the player's character exploring this detritus becomes a frame for the interior past story, revealed perhaps

through letters, abandoned possessions, details of setting and environment, and other physical remnants. A variant is to make the interior story the main character's own past, revealed to the player through flashbacks or memories. In my IF *Sand-dancer*,<sup>10</sup> the main character plays a troubled teenager who crashes his pickup on a lonely New Mexico road while on his way to make a significant life choice. As the player explores an old ranger station trying to find a way to fix the truck, the character encounters places and objects that remind him of his troubles, through which the player slowly learns his backstory, what's at stake, and what the eventual choice will mean. Exploration of physical spaces can easily map to exploration of mental spaces, emotional spaces, or many other frames and paradigms.

Stories can also use *conversation as a map*, treating an interaction with a character as a tree-like branching structure with nodes representing points where the player-controlled character chooses what to say next. Many works using this structure are designed to unfold across several playthroughs, revealing different facets of a character as the player sees how they respond to different kinds of conversational gambits or actions. Emily Short's *Galatea*<sup>11</sup> consists entirely of a highly branching conversation with a complicated character whose full story can only be uncovered through multiple traversals. While care must be taken to minimize the scope of a conversation, this format is familiar to many game players and can be an interesting way to explore a dialogue-driven approach to story.

A more complex but potentially more rewarding model is using *social norms as rules*, where the expected social behavior in a particular story context is encoded as rules that characters (including the player) must obey. In Stephen Bond's *Rameses*,<sup>12</sup> the player controls a bullied teenager who will often refuse to obey commands not because they are technically impossible or improperly phrased, but because they violate the character's sense of his own role and agency within the constrained social order of a boarding school. The reader gains a powerful understanding of these implicit social rules through exploring the author's construction of them as a formal system. This technique touches on the concept of procedural rhetoric,<sup>13</sup> the idea that interactive systems can convey meaning through the structure of their rules and play, not only their "content" in the more traditional sense of words and pictures. Though it requires an adjustment in thinking, as a style of storytelling this can be a powerful, engaging way to tell character-driven stories by putting the reader-player in a character's shoes, in an even more direct way than a linear second-person narrative.

As another example of this last approach to character, in the course we studied an IF called '*Mid the Sagebrush and the Cactus*'<sup>14</sup> which uses a basic conflict system to drive a Wild West showdown between a wounded gunslinger and a grieving son. Various conversational strategies (such as

placate, explain, and taunt) and physical actions (such as shooting or dodging) will succeed or fail based on your past actions, the other character's changing mood, and a certain amount of randomness. The system in the short piece is just complicated enough to allow for some strategy, while remaining simple enough to get a sense of how it works with only a few playthroughs. Completing the story with a satisfactory ending requires understanding the two central characters well enough to know what their soft and hard spots are, both physically and emotionally—an understanding gained through interacting with the system abstracting their perceived options and the social rules of Western showdowns into programmatic form.

Thinking about characters in these procedural, abstracted ways can be challenging. Students might be tempted to turn the focus away from characters to abstract puzzles or story-free environments, which can work for traditional IF but isn't the ideal outcome in a creative writing workshop. But by encouraging students to tell stories about people, introducing them to techniques such as character studies when developing their story ideas, and pushing them to think about conflicts, emotional stakes, and other facets of good storytelling, an instructor can encourage students in directions that will help strengthen their writing muscles. A writer needs to think carefully through the heart of a story to convey it in a form where the characters can't be as expressive as in traditional fiction.

## Successes

Pushed outside comfort zones and easy solutions by the challenges of authoring IF, my students produced surprising and invigorating stories that engaged with the creative possibilities of an interactive medium, with prose often reflecting the greater thought and attention to detail provoked by the unique constraints of IF and structure of Inform. One standout student project, Will Lee's *Apathology*, alternated between sequences of hypertext fiction and traditional parser-based IF to tell a Kafkaesque story about an unnamed protagonist arrested for a crime he didn't commit and sentenced to death. As his freedom is stripped away, the player's freedom also changes through both the different textual media used and the shifting commands and environment in the IF sequences. The ability to make meaningful choices, and what choices are available even when choice is offered, becomes a central theme. The story's conclusion suggests that God is the only one with true choice, and this is followed by a readout of the project's source code and an invitation to edit it, literally rewriting the rules of the protagonist's universe. In another project, Zane Mariano's *Lex*, the player explores the life of a deaf synesthete who can literally see words spoken by

people around him. Exploring this character's life, the player moves through a map of sentences, adding and removing key words from them to uncover different biographical moments and facts of the character. The player's control over language echoes the complex relationship the protagonist has with it; the unique structure and the player's primal potency over the text makes the experience fundamentally different from the same story if told with traditional prose. These and other projects bore rich fruit from approaching creative writing from the unusual directions of game design and procedural practice, encouraging students to think about storytelling and writing in fresh, original ways.

## Challenges and strategies

Teaching creative writing with IF and Inform can present several significant challenges it would be remiss to overlook. While the rewards in my experience have been worth the trouble, running an ambitious course like this can be frustrating.

Learning a whole programming language in a few weeks can be a daunting prospect even for students who have coded before and especially for those who have not. I quickly learned that trying to cover too much of Inform's extensive capabilities in a five-week course was overly ambitious, especially with all the reading and writing I was asking my students to do. A revised version of the course was more thoughtful in selecting a careful subset of Inform's full capabilities, designed to give students a core set of connected skills without overwhelming them with extraneous or complex details. In the first week of using Inform, we learned how to create locations and link them together spatially, add objects to those locations, and respond to commands by players to move around and look more closely at those objects. This basic "explore and investigate" paradigm offered a step up in complexity from the hypertext module in the course's first week without being overwhelming. In the following week we focused on actions and logic; I introduced Inform's action rulebooks to adjust the behavior and responses of the built-in verbs, and we looked at how to create new actions and behaviors. This gave students enough power to start creating procedurally interesting stories that could really explore the possibility of IF while leaving more advanced material for students who were more ambitious or already had a background in programming. Most of Inform's features were never mentioned in class in favor of focusing on and reinforcing a small core skill set. This approach worked well, and the final projects demonstrated a better grasp of the core material, and thus more competent narratives than the more scattershot work from the first year.

While Inform's natural language syntax makes for elegant code, it can be misleading in that it offers the illusion that any natural language sentence should be understood. Students often came to me with programs that wouldn't compile because they were using sentences that read like English, but weren't patterns that Inform could understand. Taking care to stress that Inform only recognizes certain sentences and that sentences should be written just as they appear in the textbook and supplemental materials I'd provided for reference helped address this issue. Inform hampers itself a little in this regard by frequently offering a number of alternate ways of phrasing any given command. I found that picking a single syntax to teach in class and ignoring the others helped students recognize and learn through reinforcement the particular tools they needed to get their projects running.

While many parts of writing and coding can be productively explored in parallel, other aspects of these two processes are less easily reconciled. Consider the process for peer critique and feedback. Game design focuses on rapid prototyping to converge on a "core gameplay loop," then iterating and refining this basic skeleton into a more and more polished final version.<sup>15</sup> Feedback from play-testers is sought continuously during this process, and the conventional wisdom says it's never too early for feedback. Game design also tends to encourage collaborative processes of creation, and game programming is frequently collaborative, as in the practice of "pair programming," where two coders work together to build a single piece of code, watching to catch each others' mistakes. By contrast, writing is more often perceived as a solo craft, its output not shown to others until it's ready for feedback; a work presented at a workshop or critique group might be expected to be the author's best work, not an unfinished work-in-progress.

Reconciling these different practices and traditions raises some interesting questions. Should IF-in-progress be polished in a workshop once nearly finished, or shown as a skeletal prototype early on? Should an instructor ask for frequent milestones and encourage collaboration, or put the focus on individual effort and a final product? The answer depends to a certain extent on the context of the course and what the instructor most wants his students to take from it. In my case, I wanted students to create personal, meaningful work, and felt working in groups might stifle truer, more introspective stories in favor of "safer" subjects, so I required that all projects be individual. However, many students struggling to learn Inform said they would have benefited from being allowed to work in teams: a second pair of eyes can be immensely helpful when trying to catch a stubborn bug. Splitting the project into two halves, one collaborative and focused on mastering skills and the second solo and focused on writing, might be a good solution.

I also wanted students to have a chance to get feedback on their work, but the class size and quick timetable precluded a traditional round-table

critique format. My compromise was to assign a series of quick “weekend experiment” projects through the first few weeks of the course that students could optionally show off to each other in a “demo fair” open exhibition at the beginning of next week’s class. The hope was that students could get feedback on small-scale individual works that would be helpful for their final project, learning the kinds of things that did and didn’t work in an interactive story. While this was largely successful, some students who were working on more personal pieces didn’t feel comfortable showing their work in this sort of casual format and missed out on the sort of insightful feedback they might have gotten from a critique-group approach.

Striking the right balance between the differing creation practices of two communities means paying attention to the size and needs of a particular group of students. In my case, stealing elements from both game design and creative writing practice best led to the hybrid works my course was designed to explore, but an approach that focused more on one school or the other could also be equally successful.

Perhaps one of the biggest obstacles to using Inform in the classroom is the time investment in learning the language well enough to teach it and answer questions, as well as creating new lesson plans to cover it. Fortunately, there are several great resources to help instructors using Inform in a classroom setting. The language’s official website at [inform7.com](http://inform7.com) has a whole section on teaching, with resources broken out by grade level and subject. Inform comes with extensive free documentation and examples, and several full textbooks both online and from major publishers can help guide instructors or students through creating their first stories.<sup>16</sup> The language has a large and active community quick to offer advice and answer questions. At press time, the central hub of Inform users was the Interactive Fiction Community Forum at <http://intfiction.org>. IF author Emily Short has also compiled a list of courses with syllabi that have used Inform.<sup>17</sup> Finally, a course like this can also be a chance for cross-disciplinary collaboration. I’ve found computer science departments are often willing to engage with the arts and humanities, but rarely know where or how to begin. Bringing a group of students together from multiple backgrounds to work on projects at the boundary between engineering and art is a great way to make connections, for students and faculty alike.

All told, is teaching creative writing with Inform worth the struggle? While it took some preparation, I found the class to be intensely rewarding both times I taught it. I was able to introduce creative writing to game design students who might never have encountered it in a college context otherwise, and introduce some experienced writers to a new mode of storytelling. People have fun designing interactive stories; it’s a surprisingly beguiling activity, tapping into our natural urge to manufacture secrets and then share them

with each other. Notably, in neither class did I have a single student who tuned out and gave up; there were frustrations, there were tears, but nobody flat out quit. I took that as an encouraging sign as to the effectiveness of using IF in a writing class, and hope it will inspire others to try the same approach.

## Notes

- 1 "Literary Games: The Intersection of Writing and Play." DANM (Digital Arts and New Media) 132, Summer 2011 and 2012. Syllabus available at [gamesaslit.textories.com](http://gamesaslit.textories.com).
- 2 Another favorite, from [sixwordstories.net](http://sixwordstories.net): "Saigon Hotel. Decades later. He weeps."
- 3 Eve, Eric. *Snowquest*. Inform 7/z-code, 2009. Interactive fiction.
- 4 Montfort, Nick. *Twisty Little Passages*. Cambridge: The MIT Press, 2003: 37. Print.
- 5 Eco, Umberto. *The Open Work*. Trans. Anna Cancogni. Cambridge: Harvard University Press, 1989. Print.
- 6 You can find more about Twine online at [twinery.org](http://twinery.org).
- 7 Nelson, Graham. "Natural language, semantic analysis, and interactive fiction." *The IF Theory Reader*, eds. Kevin Jackson-Mead and J. Robinson Wheeler. Boston: Transcript On Press, 2006. 141–88. Print.
- 8 Short, Emily. *Counterfeit Monkey*. Inform 7/Glulx, 2012. Interactive fiction. Web.
- 9 The official website at [inform7.com](http://inform7.com) is the best starting point to explore this information.
- 10 Reed, Aaron A. *Sand-dancer*. Inform 7/Glulx, 2010. Interactive fiction. Web.
- 11 Short, Emily. *Galatea*. Inform 6/z-code, 2000. Interactive fiction.
- 12 Bond, Stephen. *Rameses*. Inform 6/z-code, 2000. Interactive fiction. Web.
- 13 Bogost, Ian. *Persuasive Games*. Cambridge: The MIT Press, 2007. Print.
- 14 Gijsbers, Victor. '*Mid the Sagebrush and the Cactus*'. Inform 7/Glulx, 2010. Interactive fiction. Web.
- 15 Fullerton, Tracy. *Game Design Workshop*. 2nd edn. Burlington: Elsevier Inc., 2008: 176. Print.
- 16 Aikin, Jim. *The Inform 7 Handbook*. MusicWords, 2010. Web. January 18, 2014, and Reed, Aaron A. *Creating Interactive Fiction with Inform 7*. Boston: Course Technology Press, 2010. Print.
- 17 Short, Emily. "Teaching IF." Emily Short's Interactive Storytelling, 2009. Web. January 18, 2014.

