

IF Theory Reader

edited by

Kevin Jackson-Mead
J. Robinson Wheeler

> Transcript On Press
Boston, MA

All authors of articles in this book retain their own copyrights. Neither the editors nor the publisher make any copyright claims.

Version 1, March 2011.

Version 2, April 2011.

Please send corrections to jacksonmead@gmail.com.

Contents

Preface

| | |
|----------------------------|---|
| Crimes Against Mimesis | 1 |
| <i>Roger S. G. Sorolla</i> | |

Theory

| | |
|--|----|
| Toward a Theory of Interactive Fiction | 25 |
| <i>Nick Montfort</i> | |

| | |
|--|----|
| Characterizing, If Not Defining, Interactive Fiction | 59 |
| <i>Andrew Plotkin</i> | |

not that you may remember time:

| | |
|---|----|
| Interactive Fiction, Stream-of-Consciousness Writing, and Free Will | 67 |
| <i>Mark Silcox</i> | |

| | |
|---|----|
| 2 Brief Dada Angels | 89 |
| <i>Ryan Stevens, writing as Rybread Celsius</i> | |

| | |
|----------------------|----|
| Object Relations | 91 |
| <i>Graham Nelson</i> | |

| | |
|-----------------------|-----|
| IF as Argument | 101 |
| <i>Duncan Stevens</i> | |

| | |
|---|-----|
| The Success of Genre in Interactive Fiction | 111 |
| <i>Neil Yorke-Smith</i> | |

| | |
|--|-----|
| Parser at the Threshold: Lovecraftian Horror in Interactive Fiction | 129 |
| <i>Michael Gentry</i> | |

| | |
|---|-----|
| Distinguishing Between Game Design and Analysis: One View | 135 |
| <i>Gareth Rees</i> | |

| | |
|--|-----|
| Natural Language, Semantic Analysis, and Interactive Fiction | 141 |
| <i>Graham Nelson</i> | |

| | |
|-----------------------------|-----|
| Afterword: Five Years Later | 189 |
| <i>Graham Nelson</i> | |

Craft

| | |
|---|-----|
| Challenges of a Broad Geography | 203 |
| <i>Emily Short</i> | |
| 'Thinking Into the Box: On the Use and Deployment of Puzzles | 229 |
| <i>Jon Ingold</i> | |
| PC Personality and Motivations | 249 |
| <i>Duncan Stevens</i> | |
| Landscape and Character in IF | 261 |
| <i>Paul O'Brian</i> | |
| Hint Development for IF | 279 |
| <i>Lucian Smith</i> | |
| Descriptions Constructed | 291 |
| <i>Stephen Granade</i> | |
| Mapping the Tale: Scene Description in IF | 299 |
| <i>J. Robinson Wheeler</i> | |
| Repetition of Text in Interactive Fiction | 317 |
| <i>Jason Dyer</i> | |
| NPC Dialogue Writing | 325 |
| <i>Robb Shervin</i> | |
| NPC Conversation Systems | 331 |
| <i>Emily Short</i> | |

History

| | |
|---|-----|
| 10 Years of IF: 1994–2004 | 359 |
| <i>Duncan Stevens</i> | |
| The Evolution of Short Works: From Sprawling Cave Crawls to Tiny Experiments | 369 |
| <i>Stephen Granade</i> | |
| History of Italian IF | 379 |
| <i>Francesco Cordella</i> | |
| Racontons une histoire ensemble: History and Characteristics of French IF | 389 |
| <i>Hugo Labrande</i> | |

Preface

This is a book for which people in the interactive fiction community have been waiting for quite some time. It has its origins in the early parts of this century. As can easily happen with a project like this, it lost its way some years ago. Last year, it was determined that it should finally come out, even though many of the articles were old. The original authors were contacted, and a number of them were happy to see their articles finally in print (although many of the articles in here can be found on the web in various places). Some made no revisions to their articles, some made minor revisions, and some made major revisions. There were even a few new pieces that snuck in.

There is a wealth of information contained in this book, for the first time in one place. However, be aware that this does not necessarily represent the state of the art in interactive fiction theory. Neither does it represent the breadth of the current interactive fiction community. So take it for what it is: a collection of some intriguing thought about the theory, craft, and history of interactive fiction.

Many thanks to Emily Short and her co-editor Dennis Jerz for starting this project (and for holding on to the files all these years!). Thanks to J. Robinson Wheeler for helping with the editing of this book and for the nifty cover image. Thanks to Michael Hilborn for extracting the files into a more usable format. And of course, thanks to all the authors for contributing to this volume. I hope this book inspires more thinking and writing and coding about interactive fiction theory.

Kevin Jackson-Mead

Crimes Against Mimesis

Roger S. G. Sorolla
RogerSebastian@gmail.com

originally posted to Usenet in April 1996

Initial Remarks: Puzzles, Problem-Solving, and IF

Hello all,

I've been lurking on here for a couple of months, ever since I got stuck on *Christminster*. The high quality of debate and thinking on these newsgroups is amazing, and so is the interest value of the games being put out by the likes of Messrs. Nelson, Rees, and deMause.

The recent debate on “puzzley” and “puzzle-free” IF has got me thinking about what exactly makes an IF game too “puzzley.” I think that IF (hyperfiction and the like) can definitely be free of problem-solving elements, but an IF game cannot. Here's why:

There are three possible elements of challenge in a game: coordination, chance, and problem-solving. Chess is an example of a game that is pure problem-solving; a slot machine is a game that is pure chance; and a shooting gallery is a game that is a pure test of hand-eye coordination.

If an interactive computer program has none of these elements—if, say, the point of the game is to wander through a landscape and look at all the pretty scenery—I think most of us would be reluctant to call it a “game.” The pure walk-through would get more “game-like” if, for example, the designer added a large number of non-obvious “Easter eggs”—birds that sing when you click on them, hidden areas, and so forth. Now, the goal is to see the walk-through in its entirety; certain problems have to be solved to achieve this goal.

The walk-through would also get more “game-like” if challenges of coordination were added (shoot the pixies in the Enchanted Forest!) or if elements of chance were added (chase the randomly moving Wumpus through the landscape!). Adding any of the three possible elements of a

game would move our hypothetical walk-through closer to the ideal of an “interactive game.”

But, in my view, an “interactive FICTION game” must draw its “game” elements almost exclusively from problem-solving. It’s no coincidence that the average IF enthusiast gets annoyed when the outcome of an IF game can be seriously affected by chance factors (see Nelson’s “Player’s Bill of Rights”)—I suspect that a similar annoyance would result from a challenge to coordination suddenly popping up in the middle of a game.

>KILL TROLL WITH CHAINSAW

[Loading DOOM mode . . . please be patient]

At the very least, chance and coordination challenges detract from the main focus of an interactive fiction game, which is problem-solving. They somehow make the game less prototypically IF.

This should not be surprising; most of us play interactive FICTION games for the same reason we read genres of fiction like mystery, Gothic, adventure, and SF. These genres of fiction are all about problem-solving—Who killed Roger Ackroyd? What’s the secret of Ravensbrooke Castle? How do I communicate with the alien ship? How am I going to make it across the Yukon alive?

In fiction of this type, the pleasure comes from kibitzing along with the problem-solving methods of the detective, the starship pilot, or the explorer. The clever reader may even try to work out a solution on his own, based on clues in the narrative. Then, even more fun can be had by comparing one’s own problem-solving efforts to those of the protagonist, and to the “solution” that is eventually revealed.

The added pleasure of the interactive fiction game comes, of course, from collapsing the distance between reader and protagonist. The player is directly involved in solving problems; she can manipulate the environment in a way that a reader of linear fiction cannot. But an IF game retains the goal of problem-solving that confronts both the reader and the protagonist in linear fiction.

Chance and hand-eye coordination are impossible to integrate into the reader’s experience of linear fiction, of course. In fact, I suspect that these elements are seen as detracting from the “fiction” aspect of “interactive fiction,” because they are not, and cannot be, a part of linear fiction.

To sum up my views: an IF game without problem-solving elements

is not an IF game. If it has no challenges at all, it is not a game, just a work of IF. If its challenges are not of the problem-solving type, it can be called an interactive game, but it has alienated itself from our experience of fiction.

Well, that's quite a bit of prologue to the more concrete point I'd originally intended to make about problem-solving and puzzles. So, I'll let this stand on its own for now—but with the promise (or threat?) that my next post will deal directly with why some problem-solving challenges in IF also grate against our experience of fiction and come off as “too puzzley.”

Crimes Against Mimesis

[Warning: This essay contains references to plot elements (but no spoilers) for *Theatre*, *Christminster*, and *Jigsaw* and one mild spoiler for a puzzle early on in *Curses*.]

Continuing on my previous tack, here is my necessarily incomplete survey of IF-game elements that detract from the work's reality as a piece of *fiction*, along with suggested solutions. I hope this list will make a worthy complement to the points raised by Graham Nelson in his “Player's Bill of Rights” from his “The Craft of Adventure” essays, which deal mainly with the elements that detract from the enjoyment of the work as a game.

Some of my points also build upon Mr. Nelson's observations on game atmosphere and puzzle construction, particularly in essays 4 and 5 of “Craft.”

As stated before, I see successful fiction as an imitation or “mimesis” of reality, be it this world's or an alternate world's. Well-written fiction leads the reader to temporarily enter and believe in the reality of that world. A crime against mimesis is any aspect of an IF game that breaks the coherence of its fictional world as a representation of reality.

A general rule of fiction guiding these observations, which will be reiterated later, is this: If the reason for something is not clear to the Model Reader (a late-20th-century person armed with a reasonable knowledge of contemporary Western life and literary conventions), it should be explained at some point during the narrative. Even fantastic elements must be placed against the background of known legends and

lore. The ghost who returns to haunt his murderer need not be explained, but if by novel's end we don't find out why a ghost walks up and down the midway of the abandoned carnival every third Sunday playing the kazoo, we are bound to feel hoodwinked, unless the author claims the Absurdity Defense [which will be discussed in the next installation].

My remarks are aimed at game writers and players who judge an interactive fiction game as a work of fiction, not merely a game, and want to know how to write good games that will also be good fiction. That being said, the prosecution is now pleased to present the first three crimes against mimesis, which have to do with violations of context.

[The second set of three crimes are more subtle, having to do with assumptions in the structure of the problems, or "puzzles," in an IF game. These will be covered in my next installment.]

1. Objects Out of Context

>look

This is a tidy, well-appointed kitchen. On the table you see a chainsaw.

The object out of context is one of the screaming red flags that indicates that the puzzle has taken precedence over the maintenance of a coherent atmosphere. (As Graham Nelson would put it, "the crossword has won.") In the imaginary example above, the game author needs the player to pick up the chainsaw for later use and has dropped it in any old place where the player can find it.

This is fine for the gameplay but damaging to the fictional integrity of the game. In any coherent world, things are generally where they are supposed to be. If they are not, there is a reason for it, and the work of fiction further demands that out-of-place objects or happenings have some significance that the reader (player) can guess at or find out.

One solution to the chainsaw-in-the-kitchen problem would be to move the chainsaw to a woodshed. But let's be more creative and rewrite the game so that the chainsaw has some reason to be in the kitchen:

This is a tidy, well-appointed kitchen. On the table you see breakfast: six fried eggs, a foot-high stack of pancakes and about a pound of fried bacon. A huge checked flannel shirt is draped across the chair, and on the other end of the table you see a chainsaw.

Now, the chainsaw has a context: evidently, a lumberjack was called away just before eating breakfast, and the chainsaw is his. Putting objects in context can actually add to the gameplay, suggesting realistic obstacles to getting the object. In this example, the author could put a time limit on getting the chainsaw and leaving before the lumberjack returns—you might expect that he wouldn't be too happy to see you walk off with it!

As for why the lumberjack was eating breakfast in that particular kitchen, and why he was called away . . . well, a good work of fiction will answer these questions too, in due time. The answers don't have to be profound; they just have to make sense. (For example, "A large, burly, bearded man stomps in, drying his hands with a paper towel" would give the player a pretty good idea of where the lumberjack has been.)

2. Contexts Out of Context: Genre Bending

If the object out of context is a hoary adventure-game tradition, the "anything goes" jumbling together of contexts within the same game is an even more established—some would say beloved—feature of the game tradition started by *Adventure* (Will Crowther and Don Woods, 1976). The original *Adventure* itself (to say nothing of its 550-and-up point expansions) was an omnium-gatherum of storybook characters, Tolkien refugees, and fairy-tale phenomena. *Zork (Dungeon)* (Tim Anderson, Marc Blank, Bruce Daniels, and Dave Lebling, 1979) added thereto a raftload of anachronistic objects and locations—the flood control dam, plastique explosive, the Bank of Zork.

While the atmosphere common to these games and their descendants has a rambling, Munchhausenish charm, it leaves much to be desired in the way of fictional coherence. It's interesting to note, though, that the endgame of *Adventure* (in which it is implied that the whole cave complex is a sort of theme park maintained by Witt & Co.), and the extensive after-the-fact elaborations on the history and setting of *Zork's* Great Underground Empire, are partially successful attempts at explaining the diverse elements of their respective games. Apparently, pressures towards fictional unity exist even in a patently absurdist dungeon-style game.

For the most part, unless they are aiming to imitate Zorkish whimsy, today's adventure game authors are very careful to place each game within a single genre. Reviewers are alert to incoherencies as subtle as the switch from ghost-story horror to Lovecraftian horror midway through *Theatre* (Brendon Wyber, 1995). Where settings are intentionally diverse,

as in *Curses* (Graham Nelson, 1994) and *Jigsaw* (Graham Nelson, 1995), they are usually presented as a series of internally coherent scenes, simultaneously separated and held together by framing devices. In *Curses*, the various modes of time/space/reality travel separate the scenes, while the theme of the Meldrew family holds them together to some extent; and in *Jigsaw*, the framing device is quite literally the frame (and pieces) of the magical jigsaw puzzle.

A more fruitful bit of advice to today's game designer might be to look beyond the genre in organizing the game. *Theatre*, in my opinion, is one game that relies too heavily on the horror genre, and too little on the specific plot and background of the game, to provide a context for its array of ghosts and creatures. Some, it's true, are related to the background—the ticket-taker's ghost, the invisible monster—but the slug-thing, the entity under the stage, and the living mannequins have no reason for existing except that “this is a horror story.”

Compare this to *Christminster*, which (IMHO) is a much more satisfying piece of fiction. Just about all the locations and personages in the game fit easily with our real-world image of an old English college—the chapel, the cellars, the library, the cat, the professors. But more importantly, the unusual elements are well-integrated with the background, so that by the end of the game we know who built the secret passages, why the telephone system is so primitive, and who put the bottle in the cellar. It would have been easy enough, for example, to leave the secret passages unexplained, relying on the genre convention that “old English buildings have secret passages.” The way the passages are integrated with the background story, though, contributes a great deal to the “reality” of *Christminster*'s specific fictional setting.

3. Puzzles Out of Context: Cans of Soup, or, “Holy conundrum, Batman!”

Most of the problem-solving in IF games is an imitation of the kind of problem-solving we do in dealing with the real world—or would do, if we led lives as interesting as those of the average adventure-game protagonist. Objects have to be manipulated, physical obstacles have to be overcome, people and animals have to be persuaded or evaded or defeated in a fight.

And then there are . . .

Mazes. Riddles. Towers of Hanoi. Cryptograms, anagrams, acrostics.

Etcetera.

These are the kinds of problems we normally play with to escape dealing with the real world and its problems. So, when one of these “set-piece” puzzles comes up in an IF game, we are in danger of being rudely reminded that the fictional motivation for the game—the efforts of the hero to gather loot, to get back home, to save her family, town, way of life, or universe—is itself only a trivial diversion. Or, to quote Russ Bryan’s immortal comment on a set-piece puzzle in *The Seventh Guest*, what the hell kind of villain thwarts the hero’s progress with soup cans in the kitchen pantry?

Mystery and adventure fiction, from Poe’s “The Gold Bug” on, can capably integrate set-piece puzzles into the overall mimetic goals of the story. The cryptic message in “The Gold Bug” is actually a set of instructions to a treasure; the cryptogram in Conan Doyle’s “The Dancing Men” was devised by two characters who had a need to communicate in secret. From Oedipus to Tolkien, the riddle has similarly been used as a challenge to the hero’s wits in which the reader can share. But the convention of including puzzles in the adventure story leads easily enough to excess. Think of the intentionally ludicrous villains in the old *Batman* television show, who always leave a coded clue to the location of their hangout, and are indeed the kind to thwart Batman’s progress with soup cans. (Lucky for Batman, his utility belt can always be counted on to supply a Bat-Can-Opener.)

Apart from the primitive, anti-fictional approach—“answer this riddle to open this door, just because”—there are two main ways the IF writer can work set-piece puzzles into a game. The less satisfying way is to postulate some sort of 1) eccentric genius, 2) mad god, 3) warped wizard, 4) soup-can Sphinx, who has set up the puzzles out of a) pure native goofiness, b) a desire to test the hero’s wits, c) sheer boredom, d) the requirements of a bizarre system of extraplanar magic. This way is less satisfying because, like the scheming of Batman villains, it refers too obviously to genre conventions instead of to an original representation of life. The advantage of this approach, though, is that it provides a very broad excuse to work in a wide variety of puzzles.

Are there more fictionally coherent excuses for a set-piece puzzle or two? Consider the anagram near the beginning of *Curses*, the cryptogram in *Christminster*, the Enigma machine in *Jigsaw*. All of these puzzles are related to credible real-world uses—authors as illustrious as Voltaire have used an anagram as a pseudonym; a maths professor may very well keep

his secret journal in code; and of course, the cracking of the Enigma code was a historically vital conundrum.

I hope these examples will be more instructive than any actual rules for guiding the tactful insertion of set-piece puzzles into a work of IF. The basic principle recalls French critic Jean Baudrillard's theory that Disneyland is only a decoy, an explicit sign of artificiality obscuring the fact that all of America is a "Disneyland." Instead of calling attention to the artificiality of the whole situation, a riddle or maze or anagram should have a more or less realistic role in the context of the game, serving to diminish rather than enhance the sense that the objects-and-locations "action" of the game is itself a contrivance.

>

[This part of the essay contains medium-grade spoilers for the games *Adventure*, *Christminster*, and *Theatre* and non-spoiler references to a couple of the *Zork* puzzles.]

So far, I've been looking at the ways that IF games can lose their power as works of fiction by poor contextualization of objects, locations, and puzzles. The second half of my critical rogues' gallery encloses a more insidious set of offenses. In this part of the essay, and the next part, I'll cover those "Crimes Against Mimesis" that are provoked by the structure of the puzzle-based adventure game itself.

Problems of contextualization can usually be fixed by better writing and planning of the existing game. But many of the problems I'll cover below are harder to deal with. In these examples, a feature that offends the sense of reality is often convenient to the programmer or game player. To exclude it would make writing the game more difficult, or playing the game less satisfying.

Still, striving toward this goal can do a lot to improve the quality of a game as a work of fiction, while keeping its play enjoyable. My insidious aim is to get the writer/programmer who would spend X hours doing up a sprawling 200-room mega-dungeon to spend the same X hours constructing a tighter, smaller, but fictionally more meaningful and satisfying game. (Of course, some writers have been moving in that direction on their own—I'm thinking specifically of the improvement in fictional atmosphere from Magnus Olsson's *The Dungeons of Dunjin* (1991) to his *Uncle Zebulon's Will* (1995).)

Now, onwards.

4. Lock-and-Key, and Four Ways Out

The most common problem in any interactive game is the lock-and-key puzzle. The solver starts out with an object, or “key,” and has to find a place where this key can be used to gain access to another “key,” which in turn allows access to another . . . and so on, until the final goal is reached.

Sometimes, a lock-and-key puzzle makes no pretensions to be anything else, as with the red, blue, and yellow keys in *Doom*. And, of course, literal locks and keys appear in more sophisticated games, most notably *Christminster*. Actual locks and keys can enhance or reduce a game’s fictional realism, depending on whether they are presented in appropriate contexts. One can only find so many keys inside fishes’ bellies, lost in the wainscoting, dropped at random in corridors, or hanging around guard dogs’ necks before the artifice of the puzzle structure becomes painfully clear. By contrast, all six of the keys in *Christminster* are hidden in places where one might actually keep a key, and all their locks are guarding places that one would expect to be locked; moreover, we end the game with a pretty clear idea of who normally uses each key and why.

But more often, an IF game will keep the basic logic of the lock-and-key puzzle but use other objects to implement it. A hungry frog bars the entrance; it will only let you pass if you give it a live fly. The bridge is broken; you can only get across it using the plank you found at the construction site. The key can be a found object, a character or creature whom you’ve convinced to follow you, or a piece of information like a password; the lock can be an obstacle to another location or an object that requires another object to be useful, such as a corked bottle.

Disguising “locks-and-keys” as real-world objects may superficially contribute to the realism of the atmosphere, but once the player figures out what is going on, the artifice of the one-on-one mapping between objects and problems becomes even more jarring. Graham Nelson identified this, in “The Craft of Adventure,” as the Get-X-Use-X syndrome. Give the goat a tin can, and it will cough up a red handkerchief; wrap the handkerchief around your head, and the gypsies will let you into the cave; use the lantern you found in the cave to get past the giant mole; and so on. These pat, lock-and-key solutions don’t really do justice to the complex process of real-world problem-solving,

and after a while they get boring even as abstract puzzles.

Fortunately, there are many structural remedies to the predictability of the lock-and-key game. Let's consider five:

a) Solutions Requiring More Than One Object

It's not a novel idea that a problem might require more than one object to solve. *Adventure* and the original *Zork* both had a couple of multi-object conundrums—the chained bear, the exorcism in Hell, the explosive and fuse—and in general, these went a long way towards making the puzzles more realistic and interesting.

Still, a multi-object puzzle can come off as artificial. In particular, the scavenger hunt for the various components of a Very Significant Object is one of the stalest chestnuts in modern fantasy literature, derived (as usual) from Tolkien's *Lord of the Rings* trilogy with its Nine Rings of Power: Collect 'em all for World Domination!

The Quest for Prefab Parts is to plot structure what the Quonset hut is to architecture. It shows up in innumerable role-playing game scenarios, assembly-line sword-and-sorcery novels, and seasons of *Doctor Who*; and, from what I've seen, not even the best IF games can completely keep away from this device. If the author doesn't make the "pieces" interesting objects in their own right, and plausibly integrate them into the storyline, he or she can expect some eye-rolling from the sophisticated reader ("Not the Six Shards of the Dinner Plate of the Gods again!"). As an example, the task of piecing together the diary in *Theatre* is much more believable than the task of collecting the four "eye gems," which comes later on in the same game.

b) Objects Relevant to More Than One Solution

Again, multi-purpose objects had their start early on in text adventure games—the original *Adventure*, for one. As I recall, the second use for the keys in that game popped up just about at the point where I had arrived at the one-object, one-puzzle principle by induction and started confidently leaving things lying by the puzzles they solved. How annoying to trek back to the surface for the keys!

But my assumptions were fair game for a clever designer, and nowadays it's expected that a good IF game will require the player to find more than one use for a number of objects. In general, fictional realism is thereby improved; the player must jettison the comfortable "lock-and-key" rule, which bore little resemblance to the messy process of real-

world problem-solving. However, most games nowadays allow near-unlimited carrying capacity, and the result is an equally bizarre Model Player who takes and keeps everything just in case it might prove useful later on—a Crime Against Mimesis in its own right; number 6, I believe.

c) Problems Having More Than One Solution

To my mind, the crucial difference between a “puzzle” and a real-world problem is that the real problem has more than one possible solution. This is true even of such a barren, abstract task as knocking a banana down from a 10-foot ceiling with only a chair and a yard-long pole. Chimps are usually able to “stand on chair” and “hit banana with pole,” proving that *Homo sapiens* is not the only tool-user around. This human, not to be outdone by a mere *Pan Troglodytes*, came up with:

- > throw chair at banana
- > balance chair on pole and hit banana with chair
- > hold pole and jump at banana
- > knock on door. shout for experimenter. threaten experimenter with lawsuit. experimenter, get the banana

Perhaps the Model Adventure-game Player is a chimpanzee? But all joking aside, few puzzles in any game are set up to admit this variety of solutions, and the reason is simple: the Model Adventure-game Programmer is only human. Game designers would rather spend time coding a variety of locations than implementing every second-string solution to a problem like the banana one, where the most likely solution is indeed the chimp’s way. Players would rather play a game with a variety of challenges and, to this end, are willing to accept some restriction in possibilities, especially where the alternative solutions are less obvious than the intended one.

All the same, nothing cries “This is a game, not a story!” louder than a puzzle that ignores obvious and reasonable attempts to solve it. By convention, some crude solutions are generally excluded: breaking things, burning things, hitting or killing creatures. The default messages for such actions in Inform and TADS imply that the protagonist is just not the type to take a sword to the Gordian Knot—a Doctor Who or Miss Marple, not a Rambo. Even with this healthy assumption in place, many puzzles break the fictional mood by accepting only one plausible but rather unusual solution, when there are more straightforward ways to go.

As an example, look at the opening scene of *Christminster*. The problem is to rouse a man who is sleeping on a key, just enough so he'll roll over without waking. The solution is to tickle him with a feather (this isn't such a terrible spoiler, since getting the feather is really the hard part). As a puzzle this makes sense, but as a real-world problem it's hard to see why you can't just tickle the old codger with your fingers, even though the game doesn't understand "hands," "fingers," or "tickle man" without an indirect object. Anyway, the message to the player is clear: "Be creative . . . my way!" And the hand of the puzzle author intrudes on the scene.

An IF writer who wants to avoid this problem has three options:

1. to allow the alternative solution;
2. to have the alternative solution turn out to be a wrong one even though it apparently works at the time (e.g., tickling the man with your hands is too strong a stimulation; he wakes up in the next turn and catches you stealing the key);
3. to program in a plausible, specific reason why the alternative solution is not allowable, in place of the default "You can't do that" message (e.g., "Touching a strange man with your hands would be . . . well, improper.").

Of these, the second is the most interesting; it gives the player at least a nudge in the right direction, while allowing the author to retain control over the puzzle structure. In all fairness, the player should be able to figure out beforehand that the alternative solution is not the best one, or else be given a chance to do it over the right way. A good example of a well-clued "wrong" alternative solution would be feeding a hungry swine with a rare string of pearls that's needed later on, when the beast will just as gladly wolf down a handful of acorns.

d) Objects Irrelevant to Problems and Problems without Solutions

A player who is only interested in the game tends to see irrelevant objects and unsolvable problems as unsporting annoyances, "red herrings" planted by a fiendish game designer, in defiance of the implicit rule that everything is relevant and that the task is to find out which thing is relevant to which. Because coding up a lot of useless objects and locations is hard work, designers generally agree. Most games today subsume irrelevant objects into the scenery, leaving only a couple of ringers. Even then it is considered sporting to flag useless items as such, usually with a hint or a more-or-less witty pun on the phrase

“red herring.”

If we see the game as more than a collection of puzzles, though, a game feature can have nothing to do with any puzzle and still contribute to the atmosphere or the storyline. “Smart red herrings” like the gargoyle and the chapel in *Christminster* strengthen the background of the game with additional information (even if the meaning of the initials on the gargoyle is somewhat, ahem, obscure). At the same time, they effectively rebut the creeping suspicion that all the features in the environment are dictated by one puzzle or another and serve notice that the fictional milieu has a life outside of the mere game that is being played out inside it. Even the “shadowy figure” red herring in the original *Adventure* is eventually explained in terms of the game’s rudimentary background (those vain dwarves!). Consequently, the player feels satisfied, rather than frustrated, when its true nature is revealed. To sum up, in the well-written IF game, every item and location should still serve some purpose, but the puzzle-game shouldn’t be the only purpose.

5. “I Am Not A Puzzle! I Am A Human Being!”— The Reality of NPCs

Paper-and-pencil role-playing games use the term “non-player characters,” or NPCs, to refer to the troupe of imaginary personalities controlled by the game referee. In the hands of an imaginative referee with a flair for improv acting, NPCs can take on a life of their own. The referee can assess how they would react in nearly any situation and have them banter, barter, bluster, or battle accordingly, pursuing their own motivations while remaining true to type.

Computer interactive-fiction games also refer to characters programmed by the game’s author as NPCs. In a comparison between the two kinds of game, though, the live referee has a rather unfair advantage over the programmer. The game-master bases NPC output on a highly sophisticated interactive algorithm synthesizing years of social observation and literary convention: the human mind. To even begin to compete, the computer-game author must effectively write this algorithm from scratch; an impossible task, even for the artificial-intelligence experts!

With limitations like this, it’s hard to blame game designers for following the lead of the early text-adventure games and relegating NPCs to very simple roles: either roving menaces from a hack-and-slash

campaign of *Dungeons and Dragons* (the dwarf and pirate in *Adventure*, the thief in *Zork*) or mere components of a lock-and-key puzzle (the troll and bear in *Adventure*, the cyclops in *Zork*). And yet, a few game designers have managed to create memorable and personable characters. In the Infocom era, the robot companion Floyd from *Planetfall* (Steve Meretzky, Infocom, 1983) stands out. Among recent games, *Jigsaw* is notable for the enigmatic and recurrent character Black, while *Christminster* employs a *dramatis personae* of no fewer than twelve vivid personalities, including a very stubborn cat.

Amazingly, when examined closely, memorable characters in IF are really doing much the same things that their more forgettable counterparts are doing—roaming about the map, reacting to single words, serving as puzzles to be overcome by the right object or objects to overcome the right puzzle. Few works of linear fiction can entirely dispense with non-protagonist characters; even Jack London's classic solo adventure story, "To Build A Fire," included a canine character with at least as much personality as the hapless human hero. So, if our goal is to write IF that is good fiction as well as a good game, it's essential to make characters come alive—preferably, without resorting to advanced artificial intelligence programming!

Good writing, of course, is the linear fiction writer's key to creating believable characters without any interactivity at all, and the text elements of the interactive NPC—description, dialogue, and actions—are no different from those of the fictional character. The challenge is in joining these elements into a single, well-defined character. As with object placement, there are many ways to achieve the illusion of realism. An NPC's features need not be completely expected and stereotypical, but they should be explained if they violate common sense, unless you're aiming for a comical effect. Why is the policeman cowardly? (His uncle is a big political boss who got him the job.) Why does the minister take your satchel? (He believes you are an immoral thief and intends to return your treasures to their rightful owners.)

In fact, all the characters in a game, even minor ones, should be able to pass the book editor's eternal question, "What motivates the dwarf to throw an axe at you?" The ticket-taker takes your ticket because it's his job, a desire for world domination pushes Sauron to seek the One Ring, and so on. The answer need not be terribly deep, but it should be evident from the context and the information you provide.

Continuity across settings helps immensely in convincing the player

that an NPC exists independently of any single puzzle. A single character who appears in a variety of situations (like *Planetfall*'s Floyd) offers far more opportunity for character exposition and development than would an arkload of different creatures, one for each puzzle. As with objects, well-developed NPCs should have more than one function in the game, and these functions should make sense as a whole given the NPC's personality and motivation. In *Christminster*, Professor Wilderspin's erudition, kindness, and love of exploration are very consistently brought out through the puzzles in which he figures, and the result is an interesting and emotionally engaging character.

A more complicated example of continuity appears in *Jigsaw*, where the character of Black starts out as an impossible yet oddly helpful annoyance and gradually reveals playful, vulnerable, and even amorous sides over the course of sixteen episodes. Perhaps only love can explain why Black allows the protagonist to interfere, time and time again, with his/her attempts to change history! In any case, the development of Black's character across such a variety of roles is an impressive feat. If it works, it does so because of the multifaceted personality and conflicted motives that are brought out in Black's reactions and dialogue—continuity through an explicit admission of discontinuity, perhaps.

The beauty of the NPC illusion is that, when well-done, it can hide enormous limitations in the interactivity of the character. Inform and TADS only allow the player to converse after a fashion, by probing the NPC with single-word input ("ask Einstein about relativity"). Even with this limitation, it's patently unrealistic to expect a piece of code to be able to hold forth about every irrelevant topic the player could bring up. At the very least, though, a well-developed NPC should be able to react to basic conversational input about the elements of the present situation and about his/her background. The default response for unknown input can itself convey character; consider "Fiona treats you to a lengthy and brilliant conversation about [topic], which unfortunately leaves you no closer to getting out of the prison cell" versus "Fiona just grunts and goes back to reading her paper." Customized responses to social actions such as "kiss," "hit," and "give" are also essential to the fully individualized NPC.

Are there workable models for more complex and responsive NPCs? While it's unreasonable to expect an intelligence like 2001's HAL to emerge from a 400 kilobyte game, I think that the increasing desire of authors to create interactive games with literary elements may result in

games where the NPC, instead of being a mere accessory to a lock-and-key puzzle (“Hercules, lift stone”; “give mouse to cat”), actually is the puzzle.

I have in mind a very interesting class of NPC created on DhalgrenMOO by the character “Calkins.” Essentially, this type of automaton is a psychological maze. The rooms are the NPC’s moods, such as “angry” and “interested,” and the passages are the player’s commands, such as “hug NPC,” “feed NPC,” and “ignore NPC.” The automaton’s description changes with its mood, and each command is echoed by a response, which may or may not help the player figure out exactly what impact the command has had.

As a specific example, allow me to present my own creation, the automaton “Kim.” This is a transcript of one of many possible interactions with her; the command and response syntax should be familiar to adventure-game players. “Trismegistos” is my character; the automaton’s responses are displayed to all players in the room (including the person interacting with it) in the third person, which makes for some inconsistencies in grammar.

look at kim {initial description}

A short, craggy, thirtyish shag-cut blonde, wearing black Keds and a sleeveless blue greasemonkey coverall with her name stitched above the ciggy pocket. Kinda tough, kinda tender, she could be your best buddy or your worst nightmare.

smile kim

Trismegistos catches Kim’s attention by trying to smile. She smiles right back at Trismegistos, half-mockingly.

l at kim {description for new mood}

Kim looks back at you, half amused. She seems relaxed.

talk kim

Trismegistos’s conversation interests Kim. She crosses her arms and smiles at Trismegistos, listening.

l at kim

Kim smiles at you. “Whatcha lookin’ at?”

sing kim

Trismegistos sings, and Kim sings along, loudly and off-key, smiling.

l at kim

Kim smiles at you. “Whatcha lookin’ at?”

talk kim

Trismegistos’s conversation grabs Kim. She listens, smiling, and laughs at all Trismegistos’s jokes.

hit kim

Trismegistos hits Kim, who hits him back, playfully.

kiss kim

Trismegistos’s kiss cools Kim down some. She looks at Trismegistos warily, trying to figure him out.

l at kim {back to initial}

A short, craggy, thirtyish shag-cut blonde, wearing black Keds and a sleeveless blue greasemonkey coverall with her name stitched above the ciggy pocket. Kinda tough, kinda tender, she could be your best buddy or your worst nightmare.

Note that the same command can have different effects, depending on which mood she’s in. (I wouldn’t advise hitting her when she’s not in a good mood!) Note also that these are only three of her eight moods.

Characters with “mood mazes” have many possible uses in a game. Some moods may provide vital information; other moods may make the character more receptive to requests for help. Moods might also be triggered by giving or showing certain objects to the NPC, or asking her about certain things, or bringing other NPC’s into the room . . . The possibilities for creating intricate social situations are nearly endless.

I can’t help but suspect that character-based puzzles may have taken on a stigma from early attempts like the seduction puzzles in *Softporn Adventure* (Chuck Benton, On-Line Systems, 1984). (Yes, Kim can also be seduced; but the direct approach won’t work, and the actual experience may be less fun than getting there . . .) This stigma is unfortunate, because pornography is not the only fictional genre that can be adapted into an IF game via social and psychological, rather than physical,

problem-solving. Imagine games centered on courtly intrigues, political maneuvering, or the machinations of the psychological thriller! Concepts like *Dangerous Liaisons: An Interactive Intrigue* could go a long way to attract players who are put off by conventional, scavenger-hunt-type puzzles and want a more literary experience.

6. The Three Faces of “You”—Player and Protagonists

Computerized interactive fiction is a discourse between the game program and the game player, mediated by the player’s character (PC). By convention, the program addresses the player in second person declarative as if he or she were the character (“You are standing in a field in front of a white house”), while the player addresses the game program in a sort of pidgin second-person imperative, as if the program were the character (“examine house”; “go west”).

The origins of both sides of this curious dialogue are plainly traceable. The program’s voice echoes a human referee in a role-playing game informing the players of events in the imaginary world, while the player’s lines resemble commands in a text-based operating system (“copy file to b:\”, “cd if-archive”), their choppiness dictated by the simplemindedness of the parser.

Although bizarre by conventional literary standards, this convention has proved surprisingly robust in IF games over the years. A few games have experimented with third- or first-person narration, but none have inspired a real tradition. Perhaps it’s more satisfying, in an interactive game, to have your situation narrated directly to you by the (Dungeon) Master’s voice, as opposed to the narrative detachment of first or third person.

But the problem with second-person narrative, and perhaps a reason that literary fiction writers generally avoid it, is this: it is easy to define who is speaking in first person or who is being spoken of in third person, but it’s not so easy to see who is being spoken to in second. In effect, second person confounds the reader with the protagonist. What’s more, in a narrative that is at the same time a fiction and a game, the protagonist’s identity fractures even further, into three distinct persons:

The Reader/Player

This is you, the real human being sitting at your computer playing the game. Your goal is to amass points, finish up, and have a good time

along the way. You command all the reality-warping conveniences of the game program: save, restore, undo. You know when an item is important, because it is described as a separate object rather than as part of the scenery; you know when an action is important, because you get points for doing it.

The Game Protagonist

This is you, a nameless cipher of a person who just loves picking up objects and toting them around, because you Never Can Tell when they'll come in handy. Your goal is to fiddle around with all these objects in any way you possibly can, so you can explore your environment as thoroughly as possible and amass all the really important objects, so you can get to the really important places. Strange urges guide you—whispered warnings from disastrous alternate universes your player “undid,” oracular impulses to pick up the can opener in the kitchen because it's the only thing you really feel is important there.

The Story Protagonist

This is you, Jane Doe, an unassuming college sophomore who has stumbled upon a sinister plot to destroy the world. Or maybe you're John Doe, a cigar-chomping private investigator with calloused knuckles and a callous attitude, who has stumbled upon a sinister plot to destroy the world. Or maybe you're Jhin-Dho, a half-elven sorcerer's apprentice who has . . . Anyway, your goal is to stop the villains while staying alive, though it's a bit odd that you keep picking up stray objects without knowing why, and they always prove to be useful later on . . .

Early adventure games did not bother much with defining the story protagonist. The result (at least in my experience) is an entertaining kind of imaginative romp in which the blank hero takes on the identity of the sweatshirted person at the keyboard, running around the dungeon in tennis shoes, playing the game from within. In fact, the appearance of the *Zork* games' adventurer in the *Enchanter* series comes off as an amusing surprise, precisely because most players never thought of *Zork's* protagonist as a character in his own right.

Actually, the “hero-is-you” approach has an honorable precedent in imaginative fiction. Ever since Mark Twain's Connecticut Yankee visited King Arthur's court, everyday slobs have explored strange and fantastic worlds. And what better way to encourage involvement than to write the player in as the hero? But the limitations of the blank hero are equally

obvious, once you've played enough adventure games. Without any distinct identity, the player has only the motivations of the game protagonist as a guide, and "get the items, solve the puzzles, get the treasure" quickly grows stale when repeated from game to game.

Recognizing this, game writers in the early 1980s began to present stronger plots and identify their story protagonists more distinctively. Sweatshirt and sneakers gave way to wizards' robes, detectives' fedoras, and 18th-century crinolines. But as the story protagonist took firmer shape, the motives and behaviors of the game protagonist lingered on, like a kleptomaniac doppelganger. Even today, few IF games have managed to present a protagonist whose actions are completely defined by his or her own character, rather than by the objects-and-puzzles intrigues of the game. (Exceptions tend to fall within the mystery genre; but then again, linear mystery novels themselves have a long tradition of balancing realistic characterization with the game-like rules of the whodunit.)

Writing up a blank protagonist is easy enough, and a sensitive writer will try to avoid accidental assumptions such as "You wake up with a stubbly chin" (not applicable to both genders) or "You turn white as a sheet" (not applicable to all complexions).

A writer who wants to write a definite character, though, has to think in entirely different terms. Will the character be given only an identity, or a fully developed personality as well? Most IF games present the story protagonist more in terms of social roles and motivations than in terms of strong personality traits. For example, in *Christminster*, you are Christabel Spencer, a young, properly brought-up British woman whose brother, a college professor, has mysteriously vanished. *Christminster* does an exceptionally job of outlining Christabel's role as a woman by limiting her actions (she can't enter chapel bareheaded) and through the NPCs' dialogue (the villains and the Master are condescending, while young Edward sees her as a confidante).

Motivationally, too, Christabel's actions are clearly determined. She needs to explore the college so that she can complete her brother's researches and eventually find out what happened to him. Even the one necessary act of vandalism she commits at the beginning of the game can be explained as an attempt to enter the college, although the text could bring this out a bit more clearly.

Christabel's role in the fiction is much more clearly defined than her personality. She is by turns stoic (when attempting to cry on demand)

and squeamish (at the sight of a skeleton), proper (when entering chapel) and improper (when committing various acts of theft, wiretapping, and trespass). Her constant traits are those inherited from the game protagonist: inquisitiveness and acquisitiveness. The variety of her other traits, too, can mostly be chalked up to the demands and necessary limitations of a number of different puzzles.

But it's not clear to me that straitjacketing the story protagonist with a definite personality is always a good idea. While the reader/player can usually identify with a person of a different gender, ethnicity, social role, or time period, it's harder to project one's self into an entirely different set of personality traits. Such a protagonist would be experienced more as a "he" or "she" than as an "I," robbing the second-person narrative of its potency, and character identification would suffer at the expense of character definition.

A basic tenet of social psychology—the "fundamental attribution error"—can be stated thus: we are reluctant to accept our own actions as indicative of our personality traits and eager to attribute the actions of others to their personality traits. In part, this is because we see ourselves exercising many different traits in different situations. We are deferent to superiors, authoritative to underlings; courageous in areas of our expertise, hesitant in things we know little of; cheerfully unafraid of spiders, but repelled by the sound of crinkling Styrofoam. (Well, I am, anyway.)

Christabel's apparent inconsistency of personality, then, may actually be helpful in getting the player to identify with her. What's more important to writing vivid story protagonists, in my view, is consistently bringing out the character's role in relation to the external world and setting his or her actions up to reflect clearly defined motivations.

Closing Comments

I'll close by covering two special problems, and offering partial solutions: one in which the player's task can result in a less believable story protagonist and one in which the game protagonist's task can also undermine the story.

Save, Restore, Undo

Some might argue that an IF game is made more "realistic" by disallowing the ability to restore games or undo moves, but I disagree.

The ability to undo is no less realistic than the ability to restart the game, and a good deal more convenient. Given that a restartable game can always be played with knowledge from a previous, failed “incarnation,” the task of the player is not literally to live or die as the protagonist would but to maneuver the protagonist so as to “write” the optimal narrative that the game author has hidden within the program, in which the protagonist does everything right and achieves a happy ending.

(This process brings to mind a toy from my childhood called “Chip-Away”—a rather literal-minded take on Michelangelo’s famous dictum that the statue is hidden within the block of marble. The makers of “Chip-Away” embedded a white plastic statue within a block of white soap, and the young “sculptor” was provided with hammer and chisel.)

All the same, the finished account of the protagonist’s efforts will look odd if it shows signs of having been produced this way. Practically speaking, this means that the player should in theory be able to complete the story without using any information gained from fatal dead-ends. An obvious violation: hiding a magic word at the bottom of a (full) well so that you see it just before you drown and pass it on to your next game-incarnation.

A less obvious violation: the fatal trial-and-error puzzle. Consider four identical doors, one leading onwards, one concealing a lethal explosive. In the story that would result from solving this puzzle, it would be much more satisfying to the story reader and the game player if there was some way to tell which door hides the ticking bomb, rather than having success come only from a lucky guess. The clue may be difficult enough so that the player opts for the brute-force, save-restore-undo method (who would think to “listen to north door”?), but at least it is there to explain the story protagonist’s actions in a fictionally satisfying way. Even though real-life survival may often depend on dumb luck, fiction can only get away with so many strokes of fortune before suspicion sets in.

Examine All; Get All

In the same way that save/restore/undo can lead a story protagonist to act in strange ways, the demands of the game protagonist can often intrude into the story. Most jarringly, the game protagonist finds it useful to pick up all objects that the program indicates can be picked up, when the story protagonist might have no real reason to, say, take an apple peeler out of someone’s kitchen.

Let's look at the two ends of this problem. On the picking-up end, there is the cue that the game author sends the game protagonist when presenting a room with a usable object in it:

This is a well-stocked, modern and efficient kitchen, done up in an avocado-green color scheme.

On the table you see a battery-powered flashlight. An apple peeler is lying on the counter.

The well-trained game protagonist will, of course, pick up both these objects and take them along. But the story protagonist? If he or she is anticipating doing some exploring, it would make sense to pick up the flashlight—but why the apple peeler? And in terms of the story, what is so darned attractive about the apple peeler, as opposed to all the other objects subsumed in the description of the “well-stocked kitchen”: the pots, pans, knives, can opener, oven gloves, and so forth?

On the putting-things-down end, there is the recent trend towards allowing near-infinite carrying capacity via a container—rucksack, purse, or what have you. Understandably so, since realistic constraints on inventory make for an annoying game where much of the action consists of running about trying to remember where you dropped that screwdriver. And yet, the person who is reading the story has to wonder occasionally at the verisimilitude of a character who casually totes around a portable yard-sale of forty-odd objects, as happens at the end of *Jigsaw*.

(What's even more annoying about *Jigsaw*'s cluttered rucksack is that only one or two of these objects have any use outside the episode in which they were found. Yet the faithful game-protagonist hangs on to the green cloth cap, the stale piece of corn bread, and the mandolin because “you never know.” It's a shame, because the time-travel theme could easily have provided some cosmological excuse to prevent the export of objects from their own time period. The challenge then could have been to find some way of getting around this rule in order to solve the later puzzles, as in the later stages of *Uncle Zebulon's Will* where the protagonist has to smuggle objects past the watchful demon.)

These challenges to the fictional integrity of the protagonist's actions may not have an easy answer, and I don't think they should necessarily be answered at the expense of anyone's convenience. In the kitchen, for example, I don't think the answer is to code up a whole lot of useless

pots and pans. Hiding the apple peeler is also futile, since the good game protagonist knows to search every nook and cranny before moving on.

The action to be simulated here is the protagonist coming across a Very Important Unpeeled Apple in the course of the adventure and thinking, “Oooh . . . there might be an apple peeler back in the kitchen!” Cuing reminiscences explicitly would give away the solution to the puzzle, of course. It might be possible to force the player to go back to the kitchen and explicitly type “look for peeler” in order for the apple peeler to appear or to forbid that the apple peeler be taken until the apple has been encountered, with messages to the effect of “What on earth do you need that thing for?”

I suspect, though, that clever game players will figure their own way around these devices, commanding protagonists to search for every likely object in a location and looking for hints to a new puzzle by going back and trying to pick up every “forbidden” object they’ve encountered. Perhaps a workable compromise would be to design games so that most of what you need to solve a given problem is available relatively nearby, apart from obviously useful tools or strange artifacts that can be taken from scene to scene.

Alternatively, you could place very realistic limits on what can be carried around but automate the process of remembering where objects are, as with the “objects” command in Inform. Even the process of going back and getting them could be automated, possibly with a “walk-to” routine that checks to see if there is a free path from the current location to the known object’s location and expending the requisite number of game turns to get the object, while taking only a second of the player’s time.

Toward a Theory of Interactive Fiction

Nick Montfort¹

-
- 1 I write this note more than nine years after I started work on an article for *IF Theory*. I hope I'll keep improving and revising my ideas, but I'm also hopeful that I am finally revising this particular article for the last time. While it was always intended for this book and was otherwise never published in an official sense, drafts have been available to the public on my website at [<http://nickm.com/if/toward.html>](http://nickm.com/if/toward.html). The essay has been linked to from blog posts, syllabi, and other pages and has been cited in academic writing several times. For this reason, I will describe the significant changes that I have made in each version.

Thanks to comments from people on the newsgroup and in email about my first idea for an *IF Theory* article, I wrote the first draft of "Toward a Theory of Interactive Fiction" and posted it for discussion on January 8, 2002. In version 1.5 of January 15, 2002, I added a description of the course concept, revised the discussion of puzzles, and quoted two short excerpts from transcripts. In version 2 of April 9, 2002, I added discussion of different narrative, extranarrative, and metanarrative (now called hyponarrative) voices and the short discussion of IF via game theory. I made only minor revisions in version 2.5 of May 27, 2002.

My work on version 3 of December 29, 2002, benefited from the many helpful comments, corrections, and suggestions made by Gerald Prince. I added discussion of the most common sense of "story" and an explanation of how interactive fiction can lack puzzles. I also added discussion of unfinished works, works without final replies, and repeating situations. I aligned the IF concept of character (not "person") with the narrative concept of character. I described *Exhibition*, *Suspended*, and *A Mind Forever Voyaging* further. I revised the discussion of *Infidel* to acknowledge that it can, in some sense, be won and also provided new examples of unwinnable works. I added the distinctions between puzzle and task and between the formal meaning of "solution" and the meaning in terms of the interactor's understanding. The last major change for version 3 was the addition of the table of different input types. In version 3.5 of December 19, 2003, I added an introductory paragraph and made minor revisions.

Finally, in 2007, a revised version of this article became "Steps toward a Potential Narratology," chapter 4 of my dissertation, "Generating Narrative Variation in Interactive Fiction." As noted there, "The only substantial changes involve the introduction of the concepts of unrecognized inputs and clarifications, some further development of the nature of puzzles as requiring 'non-obvious' actions,

1. Theorizing Interactive Fiction

Interactive fiction (IF), a category that is typically represented by the text adventure or text game, has literary, gaming, and other important aspects. Early text-based interactive fiction includes *Adventure* (1977), *Zork* (1977-78), *A Mind Forever Voyaging* (1985), *Knight Orc* (1987), and *Curses* (1993). In my book *Twisty Little Passages* (Montfort 2003a), I introduce interactive fiction in detail, discuss its important historical precursors and cultural contexts, and offer a figurative way to think about its poetics and aesthetics, with reference to the literary riddle. In this essay, my focus is on particular ways that the study of narrative, narratology, can inform a rigorous theory of interactive fiction that remains sensitive to its many-faceted nature.

Systematically relating interactive fiction to “game” and “story” requires more than the *ad hoc* application of terms and concepts from literary theory, narratology, and gaming. Although humanists and scientists can be prodded toward insight by offhand approaches, deeper insights and more substantial progress require a methodological framework, a way to evaluate results, and some sort of common language and understanding about the nature of the topic under consideration. To build a theory of interactive fiction that is useful in deeply understanding how interactive fiction is experienced, I have found it necessary to distinguish those elements of interactive fiction that result from it being

- a text-accepting, text-generating computer program;
- a potential narrative, that is, a system that produces narrative during interaction;
- a simulation of an environment or world; and

and the addition of a section (4.11) offering a typology of IF outputs.” That section offers a nice example of how these theoretical distinctions can have practical value—in this case, for developing a text generation system for IF. However, the discussion in that section is most meaningful in the context of my dissertation. Because of that, I am not including that addition in this version of the article. I have made other stylistic changes throughout to try to provide an essay that is more readable than the “dissertation version” but have not revised or expanded the underlying concepts.

- a structure of rules within which an outcome is sought, also known as a game.

Interactive fiction was almost entirely neglected in academic discussion for decades. In the IF community, discussion has touched on many important aspects of interactive fiction but without developing a detailed theory. Marnie Parker's "An Iffy Theory" is an attempt to categorize people's taste in interactive fiction (Parker 2000) but is not about aesthetics or poetics as it does not explain, for instance, how one "auditory" IF work might be better or worse than another or what the elements of such a work are. Graham Nelson's "The Craft of Adventure" (Nelson 1995) is about how to write interactive fiction well, as its title suggests. It discusses many related topics in depth but offers mainly advice rather than the beginnings of a systematic theory.

An academic attempt to offer such a framework is "Towards a Theory of Narrative in Interactive Fiction" by Sean Smith and Joseph Bates, a result of research at Carnegie Mellon's Oz Project. This report was an attempt to formulate interactive fiction in terms of cinema, based on "an art-film text taken at random from the shelves at CMU's library" (Smith and Bates 1989:6). No distinction was made between techniques specifically tied to time-based and visual effects and those generic to narration in any medium (Chatman 1975:299-300). While the paper is of practical use and does describe a series of techniques for interactive fiction that is inspired by cinema, the mappings between film and IF techniques are arbitrary and unsystematic.

Roger Carbol's "Locational Puzzle Theory" is interesting in that it attempts a strict definition of certain elements of interactive fiction (Carbol 2001). However, Carbol defines a game only as "a collection of objects, in the object-oriented programming sense," which does not distinguish games from non-games, as any definition should. Furthermore, "object" is not defined by Carbol as it is in any thorough discussion of object-oriented programming but as simply "a collection of properties." The impulse to define puzzles precisely and examine their nature is a good one, but there is nevertheless confusion in this approach—on the one hand between a software development methodology, objects in the IF world, and narration, for instance, and on the other hand between location in the space of the IF world, the awareness of the interactor, and the properties of programmatic objects. The resulting distinctions between classes of puzzles are not clearly better than have already been devised in less principled classifications

(Rees 1993).

Emily Short's essay "What's IF?" makes several points of interest, although it does not define interactive fiction well enough to distinguish it from chatterbots and other programs (Short 2001). The concept of the *benchmark* as an unique action that makes progress toward an ending is a useful one. The discussion in "What's IF?" is still somewhat preliminary, though, with *action* not defined, for instance, and with the supposedly formal benchmark being defined with appeal to the interactor's anticipation and other possibly interpretive factors. The discussion of puzzle has interesting aspects but does not conclude with a definition of puzzle that can be applied consistently by other theorists. Short's essay is a good effort to not only define qualities of a puzzle but also place puzzles in the overall context of an IF work. The approach also makes it clear that a theory that carefully distinguishes formal aspects from those related to interpretation will be valuable.

Since a work of IF can be implemented in different ways and function identically, our theoretical discussion of a work's function cannot rely on details of its implementation. Definitions of the elements of an IF work from a theoretical perspective should be done without making reference to a program's specific data structures, functions, objects, and so forth, considering the program instead as a black box that accepts input and generates output. (The clearest justification for this is seen in cases where two programs that are identical from the standpoint of the interactor are implemented in radically different ways—for instance, first using a functional programming language and then using a procedural one. Different objects can of course also be used in two different object-oriented implementations.) It may happen that sensible programmers developing IF works have found it convenient to encapsulate certain fundamental elements as discrete entities in code. This is worth knowing, but if our theory of the formal, interactive, and narrative nature of interactive fiction has to refer to this implementation level, we have not done a good enough job of understanding the level we are studying.²

2 This is not an objection to reverse-engineering programs, looking at their source code, or otherwise considering the code level and the implementation of new media systems. Such analysis is essential for full understanding of digital media and can reveal aspects of practice and computing that would be difficult or impossible to see otherwise. The point here is simply that it is possible to consider how a work of interactive fiction functions separately from how it is implemented and that it is appropriate to do so when conducting an analysis at the level of form and function.

Taking this view of a formal theory of IF, this essay considers the nature of interactive fiction as *program*, *potential narrative*, *world*, and *game*, describes how the perspective of the person interacting can be represented, and offers some thoughts on conceptualizing the puzzle.

2. Interactive Fiction and the Interactor

A work of interactive fiction is, among other things, a computer program that accepts text input from a user and produces text output in reply. This user of an IF work is the *interactor*, following the terminology of the first major academic effort in interactive fiction, the Oz Project; that term has also been adopted by others (Murray 1995:161). It is synonymous with *player* as that term is usually used in the IF community, but *player* has other meanings related to games and drama while *interactor* has a history of being used only to refer to the person who interacts with an IF work or similar program. In the case of a work of IF that has no multimedia elements at all and uses only text for a medium, *text* simply refers to a string of words in the ordinary sense. However, *text* can also be considered semiotically to be any set of signifiers; thus IF works (and perhaps other works as well) that contain graphics, sound, or video can be accommodated in this way. Using *text* more specifically, to mean “strings of words,” *interactive fiction* indicates a category of text-based works, works that can contain other media elements but where text and textual exchange are central. *Computer program* could also be generalized to include other sorts of text machines in the broader cybertextual sense (Aarseth 1997)—written-out instructions that a person could follow, for instance, or Scott Adams mimicking his *Adventureland* by uttering the output it would give in reply to someone’s spoken input (Hoy and Jerz 2001). For the purposes of this essay, only computer programs in the usual sense need to be considered as interactive fiction, although, again, the theory presented here should be extensible to other types of systems.

Rather than state, as Short does, that “IF *tends* to represent, in some form, an environment or imagined world whose physical space we can explore,” (Short 2001) it seems better to say that a simulated world, the IF world, is essential to interactive fiction. The only counterexample Short advances is Andrew Plotkin’s 1997 *The Space Under The Window*. This is a work of hypertext implemented in Inform; instead of clicking on a word as would be typical on the Web, typing one of the words

displayed causes the appearance of a new lexia (Landow 1992), indicating a section of hypertext. Plotkin himself refers to this work as “Not standard interactive fiction” (Plotkin 2001). None of the theoretical discussion that Short develops in her essay applies to this work, a work which clearly seems better considered as hypertext than as interactive fiction. Considering the simulated world as essential does not mean that any particular code is required in a work of IF. Whether a work simulates a world or not can be determined by an interactor who encounters and studies a work through its interface.

Since a simulated world and textual description of events in it is entailed by a program’s being interactive fiction, an IF work is also necessarily a generator of narratives. The distinction between what can be simulated and what can be narrated is particularly important to understanding the workings of interactive fiction; although the *potential narrative* aspect of interactive fiction is produced based on events in the *world*, there may be things that are narrated during an interaction but are not simulated.

It is standard to refer to IF works as “games,” but a work of IF is not necessarily a game (Giner-Sorolla 1996). A work can present a world that is pleasant to explore but that has no quest or intrigue. There may be no final reply that is a “winning” one, perhaps no final reply at all. Because of this I am often more comfortable referring to a *work* of IF, rather than calling everything a *game* at all times. Even when what is being discussed is actually a game, calling it a work can help to signal that our interest is in interactive fiction from all relevant perspectives, rather than interactive fiction *only* as game. The advantage of using a term like “work” is most clear in the case of certain IF works that have no optimal outcome (that is, they cannot be won), do not keep score, and contain no puzzles. Ian Finley’s simulated gallery opening *Exhibition* provides a simulated space in which the player character can look at paintings while chatting with four characters who have very different perspectives on the artist and his work; there is no way to win or lose it. Calling this a “game” is unfair to *Exhibition*, which is not actually a game. Calling *Exhibition* a game is also unfair to IF works such as Dave Anderson’s *Hollywood Hijinks*, which simulates a treasure hunt in a mansion, has a very definite and explicit goal, and is clearly a game. Calling everything a “game” always makes it harder to highlight that certain works are games. Of course, I and many others use “game” to refer to works of interactive fiction in more casual discussion. Another theorist and author

refers to her own (clearly non-game) work by making reference to “a game like Galatea” (Short 2001). “Work” has real advantages as a term, however, in discussions where precision is essential.

3. Sessions, Interactions, Traversals

As computer literature pioneer Rob Wittig describes, while it is commonly thought that the reading of a book proceeds as “the reader dutifully trudges the linear track prescribed by the author,” this is certainly not always the case. A reading of a book may involve browsing it in the bookstore, reading in short bursts in different places, skipping ahead to see if it gets any better at the end, looking through bits in the middle to then figure out what happened, and giving up without actually reading everything (Wittig 1994:81-83). It is difficult today to understand much about the heavily studied processes of reading without appreciating that “readings” may not be done in the intended sequence and may not be total. The nature of interaction and interactivity in interactive fiction, which has been studied hardly at all and which in general allows for no “total reading” of the book sort to be done, will be even harder to theorize without making distinctions between aspects of interactive fiction as computer program; ways in which IF works are world, game, and potential narrative; and the interactor’s own interpretation and experience.

A *session* spans the execution of an IF program. The session begins when an IF program starts running and ends when the program terminates. The text that results (including text typed by the interactor and text produced by the program) is the *session text*.

An *interaction* describes a series of continuous exchanges of texts between the program and the interactor. “Continuous” does not have a formal meaning, nor is it a property of the text or program. The interactor’s sense of continuity and unity is what makes a certain experience a single interaction; different interactors may have different opinions of what an interaction is. The text (from both interactor and program) that corresponds to an interaction is an *interaction text*.

The experience of interaction belongs to the person involved.³ The

3 Or to the people involved. It is common for several people to interact with one IF work at the same time, although this reality is seldom mentioned in discussions of interactive fiction. This essay, however, does not deal with how multiple interactors can experience a single IF session together. There is also no consideration of the

session, on the other hand, is a property of the program and its execution. Still, interactions and sessions often correspond: an interactor starts the IF program, reads and types for a while, perhaps saves (allowing the current state to be restored later on) or perhaps arrives at a conclusion, and then terminates the program. However, one interaction may take place over many sessions, because the interactor may terminate a program and then start it again immediately, interacting with the program repeatedly in what is to her a continuous interaction. Similarly, an interactor can start a session (and an interaction), go on vacation for a week while leaving the computer and the program running, and then return to have another, different interaction that is part of that same session. Of course, the point of many works of IF is to win them, that is, to proceed towards a certain goal or outcome; “winning” can be seen as one analogue to having “read the whole book.” (This is not the only such analogue, though.) Winning cannot be described in terms of *session* or *interaction* alone.

A *traversal* is what happens in one or more sessions, and one or more interactions, when the interactor “completes” a work of IF by going from the beginning until no more can be narrated. The full definition of traversal is given in section 5; to define the term exactly it is necessary to describe more about IF as simulated world and potential narrative. The traversal is mentioned here because of its relationship to *session* and *interaction*. Of course the text corresponding to a traversal is called a *traversal text*.

4. Cycles, Exchanges, and the IF World

Anything contributed by the interactor, from a press of the space bar to a long typed text, is an *input*. The texts produced by the program are *output*. If the program outputs some text that the interactor originally typed, that is nevertheless output, just as whatever the interactor types (even something previously output by the program) is input. A *cycle* is one input and all the output that follows it until the next input. The *initial output* is whatever output is produced before the first opportunity for input; this is before the first cycle. All of this is defined formally with regard to an IF work’s nature as a computer program. Pressing the space

interesting fact that an interactor could actually be a computer program rather than a person.

bar in response to "[MORE]" is an input, for instance, even though it normally provides the interactor no opportunity to influence the course of the narrative that is being produced. It is simply because an interactive fiction work is an interactive computer program that it has input and output.

In the sense that scholars of the story and of narrative (that is, narratologists) use the terms, a work of IF is not a narrative. An IF work is an interactive computer program but not directly a *narrative*, "the representation of real or fictive events and situations in a time sequence" (Prince 1980:180). Similarly, interactive fiction is not a *story* in the sense of the things that happen in a narrative, or more precisely, "the content plane of narrative as opposed to its expression or discourse; the 'what' of a narrative as opposed to its 'how'" (Prince 1987:91). In everyday speech, of course, "story" also refers to a particular genre, the type of thing that people expect to hear when they say in conversation "so, tell me the story" or that a child expects to hear after asking to be read a story. Interactive fiction is not precisely this sort of story, either, although there may be a "frame story" provided in the documentation or there may be a certain type of story that is always generated in successfully traversing the work. An IF work is always related to story and narrative in their narratological sense, even if a particular work does not have a "story" in this ordinary sense.

The distinction narratology makes between story and narrative has been noted in various ways since Aristotle, who distinguished the argument, *logos*, and how it was arranged into plot, or *mythos*; the Russian formalists also distinguished the material of the story or *fabula* from how it was told in the *sjuzet* (Chatman 1975:295). Interactive fiction has the potential to produce narratives, usually as a result of the interactor typing things to effect action in the IF world. In fact IF works are *potential literature* in the sense of the Ouvroir de Littérature Potentielle (Workshop for Potential Literature, abbreviated Oulipo) (Mathews and Brotchie 1998, Motte 1986), and specifically they are potential narratives.

IF works also present simulated worlds. These *IF worlds* are not merely the setting of the literature that is realized; they also, among other things, serve to constrain and define the operation of the narrative-generating program. IF worlds are reflected in, but not equivalent to, maps, object trees, and descriptive texts. In fact, the IF world is the content plane of interactive fiction, just as story is the content plane of a narrative. The interactor typically types what one or more *player characters*,

who exist within the IF world, are to do. The nature of the player character, and other sorts of characters, is discussed in greater detail in section 6.

An input that refers to an action in the IF world is a *command*. In narratological terms, a command is *diegetic* (Genette 1980:227-234, Cadre 2002)—at the story or content level rather than the discourse or expression level. This command is usually in the form of an imperative to the player character. It does not have to refer to a physical action. Commands include *think*, any input directing the player character to speak, and any input directing the player character to examine something or otherwise sense something about the IF world. Commands that do not succeed are still considered commands, as long as they are understood by the parser and interpreted as attempts at action. I consider the input given to clarify a command (such as *kill the troll* What do you want to kill the troll with? *the sword*) to be part of the command being clarified. An input that refers to several actions (for instance, *take all*) consists of the several commands into which it is decomposed by the parser.

Other inputs that refer to the program rather than the simulated world, such as those that save, restore, quit, restart, change the level of detail in the room descriptions, or address some entity that is not part of the IF world—to ask for hints, for instance—are *directives*. A directive is, in narratological terms, *extradiegetic* (Genette 1980:227-231). Commands and directives are two distinct sets; all inputs that are recognized by the program are one or the other. Directives include what Graham Nelson refers to as “meta” actions in Inform (Nelson 2001:90). Based on this, “meta-command” has been previously suggested to refer to actions outside the game world (Olsson 1997), but this term has the potential to confuse a narratological study of IF, since “meta” has already been used by Genette in the opposite direction—to refer to narratives within narratives rather than to refer to the level of narration itself. To avoid confusion the term “meta-command” is left, in this discussion, to refer only to its specific meaning within Inform programming, and “directive” is used for all inputs that do not refer to the IF world.

There are some inputs that are neither commands nor directives. Any input that is unrecognized, such as a typo or a statement too elaborate to parse, is in this category. It seemed expedient at one point to classify these *unrecognized inputs* as directives (Montfort 2003b), but work on an IF development system has shown that the modules for handling these two

types of inputs should be different, since the function of these two types of inputs in the interaction is quite different.

Considering all inputs rather than just text entered at the prompt, it is still easy to classify recognized inputs into directives and commands. Pressing the space bar when "[MORE]" is displayed to indicate that additional text is available is a directive, for instance, while typing a number to select one of several conversation options is a command. *what is a grue?* in *Zork* (1979, Tim Anderson, Marc Blank, Bruce Daniels, and Dave Lebling) appears to be a directive, since there is no one within the IF world to whom this question is addressed; the information is apparently related to the interactor outside the IF world. On the other hand *plugh* in *Zork* is a command, because it refers to the player character speaking the word "plugh," and it results in a hollow voice within the IF world saying "Cretin" in reply.

Outputs that follow input from the interactor and describe anything about the IF world and events in it (including the inability of the player character to enact a particular action as commanded) are *replies*. Whether the text is a direct result of what the interactor typed or whether it describes something that occurs at specific times, or randomly, it is considered a reply, as long as it describes something about the IF world. All other outputs—that is, all outputs that do not describe the IF world—are *reports*. "[MORE]" and "[Press space to continue]" as they usually appear are reports, as are "Are you sure you want to quit?" "Your score is 0 out of a possible 100, in 2 moves." and "Brief descriptions."

| | Extradiegetic | Diegetic |
|---------------|---------------------------------------|--|
| | Interactor | Player Character |
| Input | Directive | Command |
| <i>e.g.</i> | <i>QUIT</i> | <i>PICK UP THE PHONE BOOTH</i> |
| Output | Report | Reply |
| <i>e.g.</i> | <i>Are you sure you want to quit?</i> | <i>You find nothing of interest there.</i> |

Table 1. *Recognized inputs and the outputs that correspond to them may be diegetic or not.*

An *exchange* is one command and the reply that follows it; the reply in this case includes all references to the IF world in all the output, up until the next command is entered. As command and reply correspond to input and output, so exchange corresponds to cycle.

The following excerpt from a session text of *Zork* presents two exchanges, in bold:

>open the mailbox

Opening the small mailbox reveals:

A leaflet.

>ear the leaflet

I don't understand "ear".

>eat the leaflet

Taken.

I don't think that the leaflet would agree with you.

In the first exchange, the player character is ordered to open a mailbox. This is accomplished and the result is narrated: a leaflet is now visible. Next there is an input that is not a command, since it is not understood to refer to the IF world. This is an unrecognized input that produces a *clarification*, "I don't know the word 'ear'"—revealing the limited vocabulary and brittle nature of interaction in early interactive fiction, problems that have only been mitigated in part. That cycle does not constitute an exchange. Finally there is a command for the player character to eat the leaflet. This results in the player character taking possession of it but not actually eating it. The reply seems bizarre in context; an understanding of the distinction between the diegetic and the extradiegetic, and between the command and directive, helps to explain why. "I don't think that the leaflet would agree with you," coming at this point in this session text, makes it seem as if the extradiegetic "I" in the previous report (the "I" who cannot understand certain words and translate them into actions) is now somehow within the IF world, counseling the player character not to eat a piece of direct mail. Further implications of this sort of transgression, and other sorts, are discussed in section 8.

5. Initial Situation to Final Situation, Prologue to Final Reply

The IF world can be described before the first opportunity for a command. It usually is. Such a description is the *prologue*. The term is

used here much as it was in the PrologueComp, a 2001 writing contest announced on rec.*.int-fiction (Myers 2001), except that, strictly speaking, any of this initial text that does not describe the IF world is not considered part of the prologue. This concept is similar to that of the *overture* (Nelson 2001:370).

The state of the IF world after the prologue, when the first opportunity to enter a command is presented, is the *initial situation*. A single IF work may have multiple initial situations, but because of how the initial situation is defined these cannot possibly be determined by the interactor's input. This is because the first input that can influence the world in any way is the first command; the opportunity to enter this command comes after this initial situation. Different initial situations might be determined by randomness (Short 2001), by the presence or absence of a particular file on the computer's hard disk, by the date and time, or by any other factor besides interactor input. The initial situation refers to the state of the IF world, not how that state is described. A work of IF may begin immediately with a prompt, describing nothing about the IF world. Jon Ingold's 2001 *All Roads* begins with a quotation and a menu but does not state anything about the IF world or the player character's situation. Thus, it has a *null prologue*. Similarly, the 1998 *Bad Machine* by Dan Shiovitz begins with just a prompt and has a null prologue. Nevertheless, like all IF works, these have an initial situation—this situation is simply not described before the first prompt for input. As commands are provided by the interactor, the replies reveal what this initial situation was.

The *final reply* is that reply after which the narration of events in the IF world cannot be continued. This text indicates what is usually called an ending (Short 2001). After the final reply either the program terminates or the only option is to input a directive. The state of the IF world that is described in the final reply cannot be changed by any commands made after the final reply. In traditional interactive fiction, the final reply usually narrates either the player character's death or ultimate triumph. A final reply is not required for a work to be interactive fiction, and some works, by design, do not produce a final reply. An unfinished or bug-ridden work might also not produce a final reply at all; it might instead only manage to produce a final report that is an extradiegetic error message, explaining what caused the program to crash.

By convention, some directives, such as *quit*, *restore*, and *restart*, are allowed after the final reply. Neither *restore* nor *restart* allow the narrations

of the IF world to continue, however, after a true final reply; they revert the IF world to some other saved state or to an initial situation. Similarly, *undo* in this situation does not allow a narration to continue; it simply restores the previous state of the IF world and allows the narrative to continue from that point. Adam Cadre's 2000 *Shrapnel* achieved its effect by presenting what seemed to be final narrations while actually continuing to narrate events in the same IF world in reply to subsequent commands, suggesting a transgression. The transgression is between what will be called different *courses*.

A series of exchanges that are part of the same narration, and are presented along with all the directives and reports embedded in it, constitutes a *course*. The earlier excerpt from *Zork* describes a course, for instance. In Andrew Pontius's *Rematch* and Sam Barlow's *Aisle* there can be no courses longer than one exchange. The following session text, from Emily Short's 2000 *Metamorphoses*, illustrates how—because of certain directives—a single session text can contain several courses. It also shows how an exchange can be part of more than one course. Exchanges, which have been numbered, are in bold:

1 **>get the rock**

Taken.

2 **>put the rock in the water**

Anything you dropped in there, you would be unable to retrieve.

>undo

Shore of An Underground Lake

[Previous turn undone.]

3 **>hit the bell**

You slap ineffectually at the bell.

4 **>hit the bell with the rock**

The peal is deep and resonant; the surface of the lake stands up in ripples; the darkness grows (if that is possible) more dark. Even when the sound has died and the water stilled, you find yourself waiting.

>undo
 Shore of An Underground Lake
 [Previous turn undone.]

5 >listen to the bell
You hear nothing unexpected.

1-2 is a course; nothing occurs after exchange 2 because that command is undone. 1-3-4 is another course. To quote this course we simply include everything up through exchange 4; 2 is now considered as directive because the *undo* directive was input after it, rendering that input hypothetical and meaningless within the IF world. What was the reply to 2 can be considered a report: because of the effect of *undo* this text now tells the interactor, outside the IF world, what would have happened had the command “put the rock in the water” been issued at that point in time—or, in a work of IF that does not depend on time or chance, what will happen if that command is then entered. Similarly, 1-3-5 is a course. Since any portion of a course containing at least one exchange is also a course, 1-2, 1-3-4, and 1-3-5 are only the longest three courses of fifteen in this session text.

Typing *restore* and restoring an earlier situation brings one to the end of an earlier course, where the *save* directive was issued. This allows a single course to extend across several sessions. A course can also extend across several interactions.

Can the same situation recur within a course? This depends on the nature of the IF world. In a world in which time always progresses, one cannot return to the same situation within a course—it will be later, so at least one aspect of the situation will have changed. But if time does not exist or if its laws are different, it may be possible. In fact, it is only impossible for a situation to occur twice in a course if an irreversible event occurs after every command. The progression of time is a special case of this. Note that keeping a count of how many “moves” have been made may or may not pertain to the IF world. If events always occur in the IF world after a certain number of moves have been made, this is relevant to that IF world. On the other hand, the number of moves made may just be provided (in a report) for the interactor’s information. The player, of course, may not be stepping in the same stream twice when a situation recurs, since she may have a different level of knowledge the second time. But “situation” refers only to the state of the IF world, not to that of the interactor.

The state of the IF world after a true final reply is a *final situation*. So a *traversal* of an IF work is the course extending from a prologue to a final reply, and from an initial situation to a final situation. A *successful traversal* ends with a final situation that corresponds to winning; this seems consistent with what is meant by *playthrough* (Short 2001), a term that was used on rec.arts.int-fiction for the first time not too long ago (Schmidt 1999) despite its much longer history of use pertaining to video games. Since that term has been used in video gaming to refer to something more like a *traversal* in general, or to refer to the completion of a level, “successful traversal” is offered here to clearly indicate a traversal of the whole work that ends in a winning state.

6. Player Characters, Non-Player Characters, and Other Persons

A *character* in interactive fiction is a person who is simulated within the IF world. A character’s actions as narrated can differ depending upon the input provided. The term as it pertains to interactive fiction derives not only from dramatic use and from discussion of the novel but also from the specific use of the terms *player character* and *non-player character* in the prototypical fantasy role-playing game, *Dungeons & Dragons*. These terms have a similar special meaning in interactive fiction.

A *player character* or PC is a character directly commanded by the interactor. Any other character is a *non-player character* or NPC. The interactor may request that an NPC do something, or even command an NPC to do something, but such a request or command will always be done via the PC, who is the one directly commanded. NPCs are the anthropomorphic entities who can take actions in some way within the IF world—similar to the PC-like entities called *actors* (Lebling *et al.* 1979)—but who are not directly commanded by the interactor. An actor does not have to be anthropomorphic, but this is a requisite for an NPC. An adventurer-like freedom of action or ability to act is not required in either case.

There are also *other persons* who are mentioned but who are neither PCs nor NPCs. (Since the terms *player character* and *non-player character* seem to complete the set of characters, these other persons are better not called characters; besides, in the study of narratives the term “characters” only refers to those people who actually exist within the

story, not those who are simply mentioned.) Marshall Robner, the man whose death sets up the initial situation in Marc Blank's 1982 *Deadline*, is not a character in that work of IF. Lord Dimwit Flathead is not a character in *Zork I*, either, since he is mentioned but not simulated. In Brian Moriarty's 1985 *Wishbringer*, the dragon Thermofax appears alive (albeit in a daydream) in the prologue, but it is not possible at any other point during an interaction for Thermofax to be mentioned again in a reply, and thus no input causes his actions to vary and he is not simulated. Thermofax is a person but not a character. Three scientists who appear at various points in an interaction as if they were in the room with the player character in Ian Finley's 1997 *Babel* are also not characters, since they can be recalled by touching objects but are not simulated in the IF world; no actions can influence what happens (or rather, what happened) to them, and they cannot undertake any actions in the simulated IF world.

The idea of a character (including player characters and non-player characters) in interactive fiction is analogous to the idea of a character in a narrative, defined as "an EXISTENT endowed with anthropomorphic traits and engaged in anthropomorphic actions; an ACTOR with anthropomorphic attributes" (Prince 1987:12). The difference is that a character in interactive fiction must be an existent who acts within the IF world. Being a part of the simulation, rather than being a part of the story that the generated narrative tells, is essential for a character in interactive fiction. Since people may disagree about what traits are sufficiently anthropomorphic to allow an entity to be a character in a story, there are sure to be some similar disagreements about whether something is a character (or indeed, whether it is even in the broader anthropomorphic category "person") in interactive fiction. But the category "character" in interactive fiction is similar to that category in narrative, and should be as useful. The presence of entities that cannot easily be seen as anthropomorphic or not, as in Dan Schmidt's 1999 *For a Change*, has an interesting effect, in part, because it tends to defy the easy categorization that we would like to make when thinking about characters.

Aside from the issue of how anthropomorphic a person has to be, there may be dispute about what constitutes "simulation," and therefore whether a person exists as part of the simulated world and should be considered an NPC. Sean Barrett gives the case of the Implementors in *Enchanter*, who appear as a result of the player character casting a spell,

then immediately disappear (Barrett 2002). They have a sort of existence within the IF world, but there is no opportunity to interact with them. Therefore, although they are narrated and their narration is the result of a command, they are not simulated in the way that the thief, the robot, or the troll is in *Zork*. An opportunity for the interactor's input to influence the behavior of a person—not simply to cue an appearance—would seem to be important in designating this person an NPC. Thus, the Implementors are other persons and not NPCs in *Enchanter*.

7. World, Rooms

As has been discussed already, a defining characteristic of interactive fiction is the simulation of a world. This is one aspect that distinguishes an IF work from, for instance, a chatterbot like ELIZA/DOCTOR (Weizenbaum 1966).

The IF world is divided into discrete locations known as *rooms*, which have also been called *locations* and *areas*. Like other essential elements of the form, rooms are defined independent of their implementation. A room is a simulated place from which a certain set of elements in the IF world can be sensed, manipulated, or otherwise acted upon. A room quite often contains *objects*; of course portable objects may be present or absent in different situations and objects that are present may be configured differently (for instance, may be open or closed). A different configuration of objects does not make for a different room. Rather, if a command is required to move the player character in space before certain other objects can be manipulated, those objects are said to be in a different room. Rooms, like characters, are simulated and are part of the IF world; they are not just mentioned in some of the narrations that are produced.

Shade, for instance, is aptly described as “a one-room game set in your apartment,” (Plotkin 2001) even though the player character *can* be commanded to move between the futon, the main room, the bathroom nook, and the kitchen nook. There is, by the definition presented here, only one room, because all the actions that are possible in one part of this apartment can be conducted from any other part of it, with the movement between parts of the apartment automatically entailed. The only exception is that the interactor must command the player character to stand up initially, but this is part of waking up rather than being a

restriction on moving around in general. After this, any action in any location is possible with a single command, even if the player character is back on the futon.

Even if there were works of IF that allowed the interactor to type a command like *move three centimeters left*, represented the position of the interaction in a seemingly continuous way, and thus described an environment not broken into discrete rooms the way that traditional interactive fiction is, there would still be certain sets of actions that were possible at all the different potential locations of the player character. Thus, this definition of *room*, although possibly less useful in this circumstance, would still apply.

Rooms are *adjacent* if the player character can move between them as a result of a single command that represents a single action in the IF world. Opening a door usually changes the adjacency of rooms. By this definition, *End of Road* and *Inside Building* in Will Crowther and Don Woods's 1976 *Adventure* are adjacent, as are *Inside Building* and *Y2*, since a magic word will move the player character between these rooms immediately. However, even though the robots in Michael Berlyn's 1983 *Suspended* can be commanded to move to any room from any other room, all rooms are not adjacent to all other rooms because movement between rooms occurs as a series of discrete actions, each of which is simulated in sequence over time. A robot's movement may be interrupted along the way by some obstacle or by a new command that countermands the previous one; the whole trip is not atomic, as it is when moving from one room to an adjacent one.

After a player character has been to every room, the IF work has been *fully explored*.

8. Diegesis, Hypodiegesis, and Extradiegesis

Up to now "IF world" has been used as if there were a single world for each IF work. Actually, there may be many worlds in a given IF work, just as there may be several stories told in a single text, including *hypodiegetic* ones nested inside the main *diegetic* one. (The "frame story" of the *1001 Nights* is diegetic, for example, while the stories Scheherazade tells are hypodiegetic.⁴) IF worlds, like the stories in a text, may be linked

4 Genette uses the term "metadiegesis" instead of "hypodiegesis" but admits that in his usage, "this term functions in a way opposite to that of its model in logic and

in certain ways. In Steven Meretzky's 1985 *A Mind Forever Voyaging* there are six simulated future worlds in which Perry Simm is the player character; these occur in a framework in which PRISM, a sentient computer, is the player character. The world with PRISM is *diegetic*, while the worlds with Perry Simm are *hypodiegetic*. Commands that refer to action in such a world can be called *hypodiegetic commands*. In *A Mind Forever Voyaging*, a hypodiegetic world can be reached by putting the player character into Simulation Mode, one of several modes that are available. As Perry Simm, the player character then walks around a simulated version of the city Rockvil. The input *north* in this mode is a hypodiegetic command (it is an instruction for the simulated human being Perry Simm to go north), while *record on* is a command of the usual sort (it is an instruction for the computer PRISM, in the frame world, to begin recording what Perry Simm is seeing).

Suspended presents an interesting case in which the player character is in partial suspended animation in a cylinder, and only a few commands (such as *wait*) refer directly to actions of the PC. Most commands are hypodiegetic commands issued to robots, who, although they are described by the generated narratives as being in the same physical space, an underground complex, are really in a different IF world. The robots, unlike the immobile human player character, can be told to go to different parts of the complex, can sense things, and can manipulate the environment to effect repairs. They exist and act in the IF world of this underground complex. The human "controller," fixed in the canister in the middle of a large room in the complex and unable to take any physical action at all, is most clearly seen as being part of a different (but linked) IF world. Rather than conceptualizing the robots (who are under the complete command of the interactor) as non-player characters, it makes sense to see them as player characters in a hypodiegetic world, similar to Perry Simm in one of the simulated futures of Rockvil. That the top-level world can be breached by a robot in the second-level world, who can be commanded to open the cylinder, ripping wires from and killing the player character in the frame world, can be seen as an instance of fatal *metalepsis* (Genette 1980:234-237), a transgression between different levels of story or between story and narration. This fatal variety, specifically as encountered in interactive fiction, has been called *dyslepsis* (Aarseth 1997 :118); of course a sort of dyslepsis can occur in

linguistics" (Genette 1980:228). Other narratologists have used "hypodiegesis" to refer (less confusingly) to narration at this same level, so that term is adopted here.

narrative also, as in Julio Cortázar's short story "Continuidad de los parques" ("Continuity of Parks.")

Reference to the nature of interactive fiction as a program is no novelty. When Don Woods first expanded *Adventure* to create the canonical work of interactive fiction, he added a segment that would be encountered at the end of a successful traversal; in this segment, the "closed" cave was fairly explicitly presented as a computer program that was not running. This is an example of one other type of metalepsis. Another clear and memorable instance of metalepsis early on in the history of the form is in Steve Meretzky's *Planetfall*: The robot character Floyd (within the IF world) comments amusingly on the use of the *save* directive, which is extradiegetic and which Floyd should not know about. In *Planetfall*, the awareness of metalepsis allowed humorous use of it; the unintentional metalepsis shown in the *Zork* session text in section 4 is, instead, awkward.

Understanding the basics of diegesis, hypodiegesis, and extradiegesis allows us to make more sense of the seeming polyphony of voices in which statements are made in the computer-generated text of interactive fiction. "There are at least three identities involved in play: the person typing and reading ('player'), the main character within the story ('protagonist'), and the voice speaking about what this character sees and feels ('narrator')" (Nelson 2001:368). Nelson states that this narrator speaks the prologue, but notes that "in some games it might be said that the parser, who asks questions like 'Which do you mean...?' and in some games speaks only in square brackets, is a fourth character, quite different from the narrator" (Nelson 2001:373). These different speakers in the computer-generated text are what have led others to identify the narrative voice not "as a singular speaker but, rather, as a composite, mechanical chorus coming from both inside and outside the intrigue envelope" (Aarseth 1997:120).

Just as a work of interactive fiction can have many worlds, it can have many different narrators—which need not all correspond neatly to each of the worlds. For instance, at different times, different narrators might report the events that transpire in a single world. The voice of the parser (and of other parts of the program, such as those responsible for the ability to *save* and *restore* a particular situation) is extranarrative and need not correspond to any of these narrators. Similarly, a voice that reports on hypodiegetic events (those that happen in a world within the main IF world) is hyponarrative. The numerous voices evident in even a simple

work of interactive fiction are not an undifferentiated confusion or chorus but typically correspond to different functions in interactive fiction that can be separated. Even in those cases where different voices are confused (as with the example from *Zork* given earlier), the particular voices that are being confused, intentionally or unintentionally, can be identified.

9. Winning and Losing

Many IF works have a goal that is explicitly presented or that becomes clear during interaction. Such works often indicate during their final reply whether or not this goal has been achieved. By several definitions, works of this sort, as with any rule-based activity engaged in for an outcome or for symbolic rewards, are games (Aarseth 2001, Salen and Zimmerman 2004:70-83). Reaching a final reply that indicates the achievement of the IF work's goal is *winning*, and a traversal that ends in such a reply is a *successful traversal*. Similarly, reaching a final reply that indicates failure is *losing*, which concludes an *unsuccessful traversal*.

It seems the first work of IF to problematize the concept of “winning” was Michael Berlyn and Patricia Fogleman’s 1983 *Infidel*. The final reply in *Infidel*, after completing the final task and achieving the highest possible score, includes the text “You will never get out of this pyramid alive. You earned this treasure. But it cost you your life.” Despite the attainment of the maximum score, the goal of *Infidel* was clearly not to perish inside its pyramid, having collected all the treasure. But the goal—to plunder the pyramid and escape—could not be achieved; it was possible to attain the top score and solve all of the puzzles but only possible to win this sort of Pyrrhic victory. Still, interactors could state that they “won” *Infidel* after getting to this final reply. Later works, including *Exhibition*, *Aisle*, and Emily Short’s *Galatea*, offer no optimal final reply; it would be bizarre for an interactor to claim to have won one of these. The 2001 work *Schroedinger’s Cat* by James Willson does not even produce a final reply, so it is impossible to traverse at all. However, it can be solved in a certain sense, since it presents a world that the interactor can theorize about, experiment with, and understand; this notion of solution is discussed in section 10.

In Michael Gentry’s 1998 *Little Blue Men*, in contrast, it is possible to win after entering only a few of the most obvious commands. (*Little Blue*

Men can be won in 10 commands; an interaction that results in a successful traversal might take only two minutes.) The optimal score is achieved in this outcome, and the final reply includes the text “*** You have learned to love yourself *** // In this game, you have finally managed to love life.” *Little Blue Men* is a much more intricate and complex work than such a victory would suggest, however. A different choice of commands brings the player character into an office environment that holds many puzzles and conceals something bizarre and horrifying.

An IF work has been *won* after a successful traversal, when a winning final reply is produced and a winning final situation reached. Since *Adventure* and *Zork* there has been a tradition of “the last lousy point.” Because of this and for other reasons, many IF works can be won without achieving the full score. Winning, besides not necessarily corresponding to attaining the maximum score, also does not particularly correspond to *full exploration*. It also may not correspond to the *solution* of the work.

Although IF works are always called games, and almost all of them are games, their nature as game has hardly been explored at all. A common idea is that the author competes against the player in the “game” of interactive fiction, but this makes no sense when considered in the context of other games. The developers of Monopoly, from Elizabeth Magie to Charles Darrow, do not compete with the people playing Monopoly. Will Wright and his team do not compete with a person playing *The Sims*. Nor is the computer the opponent in interactive fiction, any more than a computer version of solitaire opposes the player. In interactive fiction, the computer serves as a referee rather than an opponent (Solomon 1984:20). (If the computer provides hints it may be acting in a different role, that of a second.) “An Adventure game is an example of what a games theorist would call a cooperative game. If there are many players, as is often the case, they function as a team” (Solomon 1984:21). The myth that interaction in these sorts of games is solitary, always done by a lone interactor, contributes to this misunderstanding of the form.

From the standpoint of game theory, the typical interactive fiction game differs from a game like chess not only because the players in chess oppose one another but because in that game total information about the game state is always available to players. The state of the game (or the state of the IF world) is known only in part in interactive fiction,

and, furthermore, the workings of this world (and of the particular interface to it) are also unknown. Thus “the discovery of the rules, through trial and error, is one of the principal attractions of the game. The mark of a well-designed game of this type is that the rules reveal a consistent style, and are not merely arbitrary” (Solomon 1984:20). The nature of IF as game is too complex a topic to explore further in the current discussion, but clearly it is necessary here as well to recognize what types of games interactive fiction can offer and what aspects of a game help to make it interesting. It is worth noting that the perspective of game theory does support the figure of the riddle as a way of understanding interactive fiction, although the riddle may not formally be the same type of game. The text of a riddle itself is completely known to a riddlee, but solving a riddle requires that the workings of the riddle’s world be explored and understood, that its rules be discovered.

10. Puzzles and Their Solution

One way of understanding the relationship between the literary and the puzzling aspects of interactive fiction is by reference to the riddle, a figure that—unlike “puzzle,” “problem,” “game,” “world,” and many other commonly-invoked figures—can actually help to explain how the literary and puzzling aspects of the form work together (Montfort 2003b, Montfort 2004). The riddle is seldom invoked directly as a figure—at best, it is discussed as one type of puzzle that might be presented. But the figure of the riddle is consistent with some discussion of the puzzle in the IF community. Gregory Cox suggested two requirements for a puzzle: “a puzzle has to have an objective” and “a puzzle can’t be obvious” (Cox 1999). This is quite similar to a definition of the riddle that has been advanced: “Every proper riddle must fulfill two conditions: the first is its social function as a competition between the riddler and riddlees; the second is its literary form, which must be difficult and enigmatic, yet containing the clues necessary to decipher it” (Pagis 1996:81). A similar definition of puzzle is a good start, but it leaves several questions open. Is a puzzle posed to the player character or to the interactor? Does a puzzle have to be “required” for a successful traversal in order for it to be considered a puzzle?

This section uses the canonical Crowther and Woods *Adventure*, the first known 350-point version, to discuss puzzles in depth. If theorists

can agree about how many puzzles *Adventure* has and what they are—or even if they can disagree and articulate exactly how they disagree and why—this will be a good sign that the concept of a puzzle can be sensibly discussed as it pertains to IF works in general.

It seems possible to fruitfully discuss puzzles as formal elements of an IF work. In fact, it makes little sense to seek the puzzle in the mind of the author. What if the author is persuaded that it is a puzzle for the interactor to figure out how to type *go north* when the player character is in a room where a doorway is clearly described as being to the north? The author's belief does not, by itself, make this a puzzle. Similarly, we should not simply believe an author who denies that a certain intricate and difficult-to-discover series of required actions constitutes a puzzle.

It will also not do to rely too much on a specific interactor's state of mind and level of intelligence. Clearly, since puzzles are constructed to challenge people, a definition must refer to the thought process of the interactor in some way. Still, it makes little sense to consider that Graham Nelson's 1993 *Curses*, for instance, actually contains more puzzles when a novice sits down at the computer than it does when an expert begins to interact.⁵ Also, puzzles should remain puzzles even if a particular interactor knows how to solve them.

However, a puzzle does need to be presented as a challenge to the *interactor*, not necessarily to the player character. It is the interactor's effort at figuring out a puzzle, not any labor on the part of the player character, that is important. This is seen most clearly in part of Jeff O'Neill's *Nord and Bert Couldn't Make Head or Tail of It* and in my own *Ad Verbum*. In both of these works, the solution of puzzles relies on expressing a command properly rather than actually determining the correct action that the player character should perform. Although the IF world is essential to puzzles, puzzles are ultimately posed to the interactor outside the level of the IF world.

There is no necessary relationship between the score and the solution

5 Since *Adventure* was the first work of IF, this case is unusual; people did not know anything about how to interact, and just discovering how to move around and get into the cave was challenging. Even in this case, figuring out how to operate the work of IF in general, and how to move the player character about, is best not considered as a puzzle itself, although it may be essential to the pleasure of (or disappointments with) interaction. A difficulty that can reasonably be considered a puzzle is seen when the general operation of an IF work differs from the standard operation of interactive fiction and thus presents a special challenge, as in Carl Muckenhoupt's 2001 *The Gostak*.

of puzzles. This is seen easily in a work such as Andrew Plotkin's 1995 *A Change in the Weather*, which has puzzles but no score. In *Adventure*, score has little to do, directly, with solving puzzles; it is mostly tied to picking up and dropping treasures. Driving away the snake, which clearly seems a puzzle, does not, in itself, earn the interactor any points. Yet *Adventure* awards 25 points for "getting well into the cave," although nothing special needs to be done to get that far. Crowther's original version of *Adventure* did not keep score (Peterson 1983:188).

There is also no requirement that anything immediate happen in the world when a puzzle is solved: the player character may only later visit another part of the world to see the result of solving a puzzle. Solving puzzles does not always unlock new parts of the IF world, or unfold some larger space; a solution may restrict rather than enlarge a player character's, and therefore the interactor's, options. As a result of collecting all the treasures in *Adventure*, for instance, the cave closes and the player character is teleported to a new and much smaller location.

A *puzzle* is a challenge in a work of IF that requires a non-obvious set of commands in order to be met. When I try to determine what is "obvious" and what isn't, I imagine a hypothetical typical interactor who is encountering the work for the first time; puzzles do remain puzzles, in this formulation, after an interactor discovers how to solve them. Unlocking the grate with a ring of keys, found in plain sight a few rooms away, is not a puzzle, since it is obvious that keys unlock things. A series of obvious actions (open a box, remove the key from inside, unlock the door) remain obvious, but an action is non-obvious when an interactor must move beyond routine and do something out of the ordinary to understand the world and how to proceed. Looking beyond the obvious might require close reading to uncover hidden senses of a character's speech or of descriptions of things, conducting experiments (for instance, by putting different objects inside a machine and activating it to figure out what the machine does), or attaining understanding of the nature of something described in figurative language. While unlocking a door with a key is obvious, recognizing that something unusual is a key goes beyond the obvious.

This is not enough of a definition to unambiguously classify every challenge as obvious or not, but this criterion at least begins to suggest some way of identifying puzzles, one that does not refer to the author's intentions or the interactor's specific knowledge and aspirations. Any typical interactor with the appropriate language skills, typing and

computer interaction skills, and basic sorts of common knowledge would, in this formulation, be able to determine what is or is not a puzzle simply by studying the IF work in question, without needing to interview the author or take a survey of other interactors. The other factors essential to the determination of “obviousness” should be not the mindset of the author or of a particular interactor but the culture or subculture within which the work was published—along with the conventions of interactive fiction.

There is no requirement that a puzzle’s challenge relate to any other elements of an IF work in order for it be a puzzle. It simply has to be presented as a challenge. While the typical way of doing this is to make the solution to a puzzle a requisite for a successful traversal, puzzles can be presented in some other way. Formally, the *solution to a puzzle* is the series of commands that meet the challenge of a puzzle. A *solution to a work of IF* is a series of commands that result in a *successful traversal*, with puzzles solved along the way. The typical walkthrough, of the sort often found online, records a solution to a work of IF. It is important to note that “solution” has not only a formal meaning but also a meaning that refers to an interactor’s interpretation, operation, and understanding of an interactive fiction work. An interactor who, by interacting with a work, comprehends the entire system of the IF world—why it is arranged as it is and why it functions as it does—has solved the interactive fiction work in this sense.

The puzzles in *Adventure* are:

- Driving the snake away
- Getting the gold nugget out
- Getting the emerald out
- Lighting the dark room
- Killing the dragon
- Creating a bridge
- Dropping the vase safely

- Watering the beanstalk twice
- Oiling the door
- Opening the oyster
- Replacing the troll's treasure
- Feeding the bear to calm it
- Deploying the bear against the troll
- Finding the way through the Pirate's maze
- Finding and purchasing lamp batteries in the other maze
- Blasting out of the repository
- Dropping the magazine at Witt's End and leaving the area

The last of these presents what is probably the most questionable case. Puzzles do not have to be required for a successful traversal of a work in order to be puzzles, according to the definition advanced here; they do not have to be tied to any benchmark or other plot element. In the case of dropping the magazine at Witt's End and leaving the area, this is a puzzle because *Adventure* clearly presents a challenge to the interactor: to get the last lousy point, independent of successfully traversing and winning *Adventure*. If the interactor had 350 points beforehand and dropping the magazine gave the interactor 351 points—and there was thus no way to know beforehand that an extra point could be obtained—this could be referred to as an *Easter egg* but would not be a puzzle. A challenge would not have been presented initially.

This last lousy point also demonstrates that solving a puzzle does not have to relate to anything meaningful in the IF world. Since this puzzle does not, it may make for a lousy puzzle, but the meaningless commands to drop the magazine and leave Witt's End are nevertheless the solution to an actual puzzle. The typical method of solving this—which involves reverse-engineering the program and actually reading through the resulting assembly language to figure out where the last point is assigned

in the code—is certainly challenging for the interactor, despite the lack of any relationship to the IF world.

Finding the batteries in the other maze is a puzzle since it is presented as a challenge, issued rather directly when the lamp runs low. It is not required for a successful traversal, however. In fact, buying a lamp battery deprives the player character of one treasure and the possibility of gaining the full score.

Collecting the remaining treasures and depositing them in the building is not a puzzle or set of puzzles, because, although the pirate might steal a treasure to thwart its being moved, in general these actions are no more difficult than picking up other objects and moving them around—they just happen to be scored. A series of actions that is required for a *successful traversal* but is not a puzzle can be considered a *task*. In *A Mind Forever Voyaging*, there are no puzzles presented in the initial Rockvil simulation in Part I, only a list of tasks. The interactor is charged with carrying out these tasks, and it is enjoyable to explore and experience the IF world while doing them, but they do not require the interactor to do anything non-obvious.

11. Further Steps

This article has begun the discussion of the elements of interactive fiction from a theoretical standpoint, drawing mainly on narratology. The discussion here has only explored a few of the most important implications of clearly distinguishing the simulated from the non-simulated, the IF world from the text that describes it, and the diegetic from the extradiegetic. Perhaps the few points that have been made are at least adequate to demonstrate that a better perspective on IF can result from making such distinctions, however.

Clearly, it will be valuable to have more discussion of the nature of interactive fiction, beyond what is covered here, that

- focuses on specific works in giving examples of what the elements of the form are;
- distinguishes between elements in terms of their being formal or interpretive; and
- makes strong and meaningful claims that can be evaluated by others

and, if useful, built upon by others.

As should be clear from the title of this section and from the discussion of puzzles, the intention here is to help begin a strong theoretical discussion of interactive fiction rather than to conclude it. There is much that remains in considering the nature of puzzles and how they fit into an IF work overall, relating to its aspects as program, potential narrative, world, and game. Continuing this discussion will certainly benefit interactive fiction. Approaches from other fields of study (narratology, to be sure, but perhaps also fields including game theory and game studies) can result in a better understanding of interactive fiction. These sorts of studies should also help IF authors and developers of IF systems advance the state of the art and help us understand what makes interactive fiction compelling.

Bibliography

Works of Interactive Fiction

- Adams, Scott (1978) *Adventureland*. Adventure International.
Developed in BASIC and the Adventure editor.
- Anderson, Dave and Liz Cyr-Jones (1986) *Hollywood Hijinks*. Infocom.
Developed in ZIL.
- Anderson, Timothy, Marc Blank, Bruce Daniels, and Dave Lebling
(1977-78) *Zork*. Developed in MDL.
- Austin, Pete (1987) *Knight Orc*. Programmed by Mike Austin. Developed
in A-code. Level 9/Rainbird.
- Barlow, Sam (1999) *Aisle*. Developed in Inform 6.
- Berlyn, Michael and Patricia Fogelman (1983) *Infidel*. Infocom.
Developed in ZIL.
- Berlyn, Michael (1983) *Suspended*. Infocom. Developed in ZIL.
- Blank, Marc (1982) *Deadline*. Infocom. Developed in ZIL.
- Blank, Marc and Dave Lebling (1983) *Enchanter*. Infocom. Developed in
ZIL.

- Cadre, Adam (2000) *Shrapnel*. Developed in Inform 6.
- Crowther, Will and Don Woods (1977) *Adventure*. Developed originally in PDP-10 Fortran, ported by Linards Ticmanis in 1998 to Fortran 77.
- Finley, Ian (1997) *Babel*. Developed in TADS 2.
- (1999) *Exhibition*. Developed in TADS 2.
- Gentry, Michael (1998) *Little Blue Men*. Developed in Inform 6.
- Ingold, Jon (2001) *All Roads*. Developed in Inform 6.
- Meretzky, Steve (1983) *Planetfall*. Infocom. Developed in ZIL.
- (1985) *A Mind Forever Voyaging*. Infocom. Developed in ZIL.
- Montfort, Nick (2000) *Ad Verbum*. Developed in Inform 6.
- Nelson, Graham (1993) *Curses*. Developed in Inform 1-6.
- O'Neill, Jeff (1987) *Nord and Bert Couldn't Make Heads or Tails of It*. Infocom. Developed in ZIL.
- Plotkin, Andrew (1996) *A Change in the Weather*. Developed in Inform 6.
- (2000) *Shade*. Developed in Inform 6.
- Pontius, Andrew (2000) *Rematch*. Developed in TADS 2.
- Schmidt, Dan (1999) *For a Change*. Developed in Inform 6.
- Shiovitz, Dan (1999) *Bad Machine*. Developed in TADS 2.
- Short, Emily (2000) *Metamorphoses*. Developed in Inform 6.
- (2000) *Galatea*. Developed in Inform 6.
- (2005) *Bronze*. Developed in Inform 7.
- Wilson, James (2001) *Schroedinger's Cat*. Developed in Inform 6.

Other Works

- Aarseth, Espen (1997) *Cybertext: Perspectives on Ergodic Literature*. Baltimore, Maryland: Johns Hopkins University Press.
- Aarseth, Espen (2001) Comparative Media Studies seminar, MIT, Cambridge, February 8.

- Barrett, Sean (2002) Conversation on ifMUD with Nick Montfort *et al.*, January 7.
- Cadre, Adam (2002) "Re: Default parser responses: how do they affect the gaming / authorship experience?" <news://rec.arts.int-fiction> Message-ID <a11jtp\$8kk\$1@drizzle.com>. January 3.
- Carbol, Roger (2001) "Locational Puzzle Theory." <news://rec.arts.int-fiction> Message-ID <82675075.0110020840.12d1c469@posting.google.com>. October 2.
- Chatman, Seymour (1975) "Towards a Theory of Narrative." *New Literary History* 6, 2:295-318. Winter.
- Cox, Gregory (1999) "Re: [Theory] When is a puzzle a puzzle?" <news://rec.arts.int-fiction> Message-ID <7i3ogc\$kc3\$1@news4.svr.pol.co.uk>. May 21.
- Genette, Gerard (1980) *Narrative Discourse: An Essay in Method*. Trans. J. E. Lewin. Ithaca, NY: Cornell University Press.
- Giner-Sorolla, Roger (1996) "Puzzles, problem-solving, and IF?" <news://rec.arts.int-fiction> Message-ID <Pine.SUN.3.91.960405121744.28920B-100000@xp.psych.nyu.edu>. April 5.
- Landow, George P. (1992) *Hypertext: The Convergence of Contemporary Critical Theory and Technology*. Baltimore and London: The Johns Hopkins University Press.
- Lebling, P. David, Marc S. Blank, and Timothy A. Anderson (1979) "Zork: A Computerized Fantasy Simulation Game." *IEEE Computer* 12, 4:51-59. April.
- Mathews, Harry and Alastair Brotchie, eds. (1998) *Oulipo Compendium*. London: Atlas Press.
- Montfort, Nick (2000-2001) "Cybertext Killed the Hypertext Star." *ebr* (*Electronic Book Review*) 11, <<http://www.electronicbookreview.com/ebr11/11mon/index.html>>. Winter.
- (2003a) *Twisty Little Passages: An Approach to Interactive Fiction*. Cambridge: The MIT Press.

- (2004) “Interactive Fiction as ‘Story,’ ‘Game,’ ‘Storygame,’ ‘Novel,’ ‘World,’ ‘Literature,’ ‘Puzzle,’ ‘Problem,’ ‘Riddle,’ and ‘Machine.’” In *First Person: New Media as Story, Game, and Performance*. Cambridge: The MIT Press.
- Motte, Warren F., Jr., ed. and trans. (1986) *Oulipo: A Primer of Potential Literature*. Lincoln: University of Nebraska Press.
- Murray, Janet (1995) “The Pedagogy of Cyberfiction: Teaching a Course on Reading and Writing Interactive Narrative.” *Contextual Media: Multimedia and Interpretation*, ed. Edward Barrett and Marie Redmond, 129-162. Cambridge: The MIT Press, 1995.
- Myers, David Samuel (2001) “PrologueComp.”
<<http://www.sinc.sunysb.edu/Stu/dmyers/PrologueComp.html>>. June.
- Nelson, Graham (1995) “The Craft of Adventure: Five Articles on the Design of Adventure Games.” <<http://www.ifarchive.org/if-archive/programming/general-discussion/Craft.Of.Adventure.txt>> January 19.
- (2001) *The Inform Designer's Manual*, 4th ed., 1st print ed. <<http://www.gnelson.demon.co.uk/inform/DM4.pdf>>. St. Charles, Illinois: The Interactive Fiction Library.
- Olsson, Magnus (1997) “What is a meta-command?”
<news://rec.arts.int-fiction> Message-ID
<5g6fqg\$5t5@bartlet.df.lth.se>. Mar 12.
- Pagis, Dan (1996) “Toward a Theory of the Literary Riddle” in *Untying the Knot: On Riddles and Other Enigmatic Modes*, ed. Galit Hasan-Rokem and David Shulman, 81-108. New York: University of Oxford Press.
- Parker, Marnie (2000) “An Iffy Theory.” Version 2.25.
<<http://www.iffydoemain.org/theory/index.html>> February 19.
- Plotkin, Andrew (2001) “Interactive Fiction.”
<<http://www.eblong.com/zarf/if.html>> May 25.
- Prince, Gerald (1980) “Aspects of a Grammar of Narrative.” *Poetics Today* 1, 3:49-63.

- (1987) *A Dictionary of Narratology*. Lincoln: University of Nebraska Press.
- Rees, Gareth. “Puzzle classification.” <news://rec.arts.int-fiction> Message-ID 1993May7.172931.23328@infodev.cam.ac.uk 7 May 1993.
- Salen, Katie and Eric Zimmerman (2004) *Rules of Play: Game Design Fundamentals*. Cambridge: The MIT Press.
- Schmidt, Dan (1999) “For A Change beta diary.” <news://rec.arts.int-fiction> Message-ID <wkwwrak0ba.fsf@thecia.net>. November 22.
- Short, Emily (2001) “What’s IF?” <http://emshort.home.mindspring.com/whatsif.html> November 29.
- Smith, Sean and Joseph Bates (1989) “Towards a Theory of Narrative for Interactive Fiction.” Carnegie Mellon University Technical Report CMU-CS-89-121. February 20.
- Solomon, Eric (1984) *Games Programming*. Cambridge University Press.
- Weizenbaum, Joseph (1966) “Eliza — a computer program for the study of natural language communication between man and machine.” Communications of the ACM 9, 1:36-45. January.
- Wittig, Rob for IN.S.OMNIA (1994) *Invisible Rendezvous: Connection and Collaboration in the New Landscape of Electronic Writing*. Hanover, N.H.: Wesleyan University Press.

Characterizing, If Not Defining, Interactive Fiction

Andrew Plotkin

We talk about interactive fiction as if we knew what that meant. We always have. It isn't even arrogance; we're the people who write and play interactive fiction, so what should the term mean except what we mean by it? But then someone takes a step outside, and tries to explain this peculiar obsession of ours. "What I point at when I say . . ." is an unhelpful description, no matter how correct.

Defining genres is inevitably a fuzzy, contentious task. Even invoking "genre" here is a position that I might need to defend. Readers, movie-goers, TV-watchers—all the forms that *fiction* has traditionally owned—have a well-delineated set of genre categories. We talk about science fiction, fantasy, horror, mystery, and romance. If you visit bookstores, add in books-for-kids (or "YA") and mainstream (or "literature," or "non-genre"—if you try to interpret any of these labels literally, you're dead off the starting block).

All of those fiction genres are familiar territory for computer games. Indeed, the early history of interactive fiction made a show of them. Infocom deliberately branched out from its loosely-comedic-fantasy roots to formal detective stories, science fiction, horror, and so on. But as the videogame world evolved, "genre" came to mean something else. *Role-playing games* were a game genre; so were first-person shooters, and platformers, and racing games, and (for the lingering fans) adventure games. An RPG (or racer, shooter, etc.) could be set in a recognizably sci-fi universe, or a fantasy setting. Wizards or spaceships or dragons or robots might be visible. But if you ask for recommendations of "science fiction games," gamers will look at you blankly. What *kind* of game are you looking for? What genre?

This use of "genre" is either game-nerd ignorance or the natural and interesting way to categorize games. And we won't learn anything by taking the dismissive stance. So, then, why is it interesting and what does it categorize? (And what, ultimately, will it say about IF? You see, I am going somewhere down this academic dirt path. Have patience.)

All of these game genres seem to boil down to conventions of interface. You might say "game mechanics," instead, but mechanics are

so tightly bound to interface as to be indistinguishable. Does an RPG offer tabular views and menus of avatar qualities because the game is built around numerical combat mechanics? Or is it the other way around? It hardly matters; the interface suits the game, and has since *Wizardry* and *Ultima 1*. Shooters are first-person because that's the best way to aim a simulated gun. Real-time strategy games need to polish every aspect of their maps to give you enough information to strategize in real time.

The interface is also tightly bound to the skills a player needs to use it. That is *why* this categorization of games is natural. Acquiring skills takes time and effort; honing them is easier, and putting them skillfully into play is the payoff. Or, concisely: if you're good at shooters, you play more shooters. Communities of players form along the boundaries of the games they like to play. Ultimately, as "innovation" meets "what the audience wants," the views and standards of those communities affect new games. A genre evolves. Perhaps it specializes, or even subdivides, as its aficionados draw distinctions that might be imperceptible to a newcomer.

These concerns of skill did not sprout, mushroom-like, when the shadow of computer gaming fell across popular culture. Reading science fiction or fantasy is a skill—albeit one less likely to throw a flashing "you have died" if you come up short. Interpreting what a book offers you, and understanding its unspoken framework, has always been a sort of game. Communities of readers evolve, subdivide, and affect the creation of new books—just as they do for games. If this were another essay, I'd call this the natural and interesting way to define "genre" in any field: creators and audience in dialogue.

But this is an essay about interactive fiction. What can we say about the interface of interactive fiction?

The facile answer: it's text. But perhaps that's too easy. Games across several genres have swathes of text: diary entries, letters, even introductory epigrams. Voiceovers are just as familiar, and spoken text is still text. (IF can be transferred almost unchanged from a printed-text interface to a speech synthesizer, to the great benefit of sight-impaired gamers.)

The voiceovers and diaries are text *output*, however. IF generates text, but it also accepts text as input. Is this a better distinction? It certainly seems to fit. Outside of the IF tradition, very few games have any kind of text-based control. (We disregard single-letter menu options—not

truly text—as well as chat interfaces that are directed to other players rather than affecting gameplay.) Some of the exceptions are games with text-in-text-out conversation interfaces, such as *Ultima 4* or *Starship Titanic*, which we might view as being IF-like minigames within the larger game structure.

A broader class of exceptions—non-IF games which nonetheless are controlled by text input—include word puzzles, Scrabble-like games, and computer-mediated riddles. Nick Montfort has argued for riddles as a forerunner of interactive fiction; nonetheless, these kinds of games seem a poor fit for the IF category. What are they missing? Facilely again: simulated game worlds.

By “game world,” we do not merely mean a story setting, but rather a world with some game-mechanical heft. Or, perhaps we should say, game mechanics which in some way heft up a *world*. IF models the world (in abstraction) and allows you to interact with that model. Your textual input affects the game world, in a way appropriate to the input’s meaning. When you understand the model and its rules, you can manipulate the IF game world thereby.

Or, to be concrete: you can play the word “EXPLODE” in Scrabble, but it does not imply any sort of fictional explosion. It does not require a “BOMB,” and you cannot use it to scatter your opponent’s letters.

(Of course, a simulated game world is not enough *by itself* to distinguish IF. Just as many games use text, many games simulate worlds for the player to play in. The overlap is our target.)

The criterion of narrative is worth a detour. The label “interactive fiction” might lure us to focus on the *fiction* and describe IF as “games that tell stories.” Is this a useful distinction? I will say no. A plot is a common element of games; indeed, almost a mandatory one these days. Shooters, platformers, and RPGs are all dense with narrative. Even racing games, match-three games, and geometric-puzzle games will often introduce a bit of pro-forma story to motivate the player.

Textual narrative games outside of IF are not common, but the form can easily be imagined. If Scrabble were spruced up with a story framework (perhaps a duel between extremely literate wizards?) it would be the same game; it would not become IF-like. A narrative, by itself, is *content*—not interface or gameplay mechanics—and game genre is never about content.

To be clear, I am not making an argument about superficiality. Adding a story introduction to Scrabble at this late date would certainly

be a superficial change. But if Alfred Mosher Butts had begun with a story idea, and developed Scrabble around that, we might not know it. We would be justified in analyzing the resulting game based on how it plays, not on its history. The distinction is not whether a story is present, or even whether the player can affect the story, but *how* the player goes about affecting.

>

We have assembled a working definition of IF: *a game that is controlled by textual input, understood as its natural-language meaning (to some degree), and that provides a simulated game world, which behaves according to natural rules (to some degree)*. This is roughly the definition proposed by Nick Montfort and is probably as well-accepted as any in the IF community. I will therefore rip it apart, by comparing text adventures with what might be their closest cousin: graphical adventures.

The very earliest illustrated adventure games (*Mystery House*, *The Wizard and the Princess*) had text parsers; they were IF as we have just defined it. The genre quickly evolved towards mouse-based interfaces, but they were still called “adventure games.” Players categorized *Loom*, *Myst*, and the Monkey Island games alongside Infocom’s offerings.

I shared this identification myself. I did not hesitate to discuss *Myst* and its subgenre within the IF community. Perhaps over-pointedly, I referred to such games as “graphical adventures” or “graphical interactive fiction,” in parallel with “text adventures” or “textual interactive fiction.” I implied, therefore, that these were the natural subdivisions of something called “interactive fiction.” This usage of the term (co-extensive with “adventure game”) did not catch on. Nonetheless, I felt it expressed something important. But how does *Myst* relate to the text-in, text-out world simulation games that we have just described?

Both sorts of games involve exploration, discovery of clues, and the application of clues to puzzles. Both avoid trials of dexterity and speed. The player generally has all the time in the world to consider her actions; the challenge lies in choosing the right action, not enacting it. Both genres operate in a simulated world. Just as we distinguish word puzzles from text adventures, we can distinguish visual-geometric puzzles (such as sliding-block puzzles) from graphical adventures. (Although both sorts of adventures can *include* such puzzles. If they don’t overdo it.)

But the resemblance goes deeper. Adventure game puzzles are typically *unique*. Each puzzle in a game may require a different insight; it seems a design fault to repeat a trick. Furthermore, the reward for a puzzle will be a discovery—a new object, room, or story event. Contrast this to geometric puzzles, whose strength lies in careful permutations of predictable tricks. Contrast too with the horde of combat and role-playing games, which may throw scores of identical enemies at you. Destroying each orc carries a reward value, but not a *distinctive* reward. Text and graphical adventure games are unique in being—if we may—unique at every turn.

(Of course adventure games require repetitive and predictable actions as well. You may walk up and down a hallway several times, or pick up and drop objects. A simulated world allows that, by definition. But these actions are not what occupies your attention. They are leaves that you skim through in order to reach the root of the gameplay.)

Earlier, we focused on the game's interface as the key to its genre. A text input prompt does not resemble the rich visual depiction of a graphical game. But they both provoke the same response: "What should I do?" mingled inextricably with "What *can* I do?" Both sorts of game, in other words, require exploration of the *interface* as well as of the game world. In a graphical game, the player will tentatively click on a visible object, to see if it reacts. In a text game, the player will tentatively EXAMINE a mentioned object. These actions carry exactly the same weight, the same sense of trial, in the two (apparently) dissimilar interfaces.

Indeed, they are close enough to suffer the same design failure. A graphical adventure game must convey, through its art, which parts of the world are likely to be interactive or interesting. If the player fails too often to discern that, he is likely to lose faith in the design and start clicking everywhere on the screen. Players refer to this fate as "pixel-hunting." Similarly, a text game must convey which commands are likely to work. If it does not, it will provoke a precisely analogous response: the player will start typing words randomly, a "guess-the-verb" (or "guess-the-synonym") problem. These are the reductive failures of interactive fiction—the popular stereotypes of bad adventure games. (Or, if marketing fails us, the stereotypes of *all* adventure games.)

We must realize that while verb-guessing and pixel-hunting result from failures of design, they cannot be understood simply as *implementing too few game objects*. The game must describe (visually or textually) a world

in which the player can operate. It must make clear what is important and what is not. The interface's capabilities must *match* the player's options. If the player wants to do something that the game cannot handle, that mismatch must be resolved. Implementing the action is one resolution, yes. But so is crafting the interface so that the player realizes that the action is out of bounds.

We might wish for a game that can handle every conceivable action, but as long as we are finite game designers, that will remain a dream.

The opposite delusion, when the player cries out in frustration, is to imagine the solution is *explicitly listing every possible action*. After all (one might think), if the player sees a list of every possible verb and noun—or a highlighted chart of every clickable object—surely that will resolve the problem?

This path does, indeed, remove the player's confusion. It also removes the player's need to understand the game world. And here we approach the motive behind all the IF conventions that we have been describing.

By describing a world, and implying (though necessarily not specifying) the unbounded richness of a complete reality, the adventure game conjures the unbounded richness of real action. A person in this situation could do an infinite number of things. Of course this is a tension: you know that you have very few meaningful game options. But the interface makes no move to break this tension. It invites you to type anything (or click anywhere on the screen).

Resolving this tension is in your hands, and what are your tools? The game's description of its reality, and your understanding of it *as* reality. If you treat the words (or pixels) as interface elements of a program, you have no handhold. Any button could be the magic button. They are distinguished only by their meaning *in the game world*. You understand that an altar is an important location in a church, that a lever is an important part of a machine, that a fingerprint is an important feature of a crime scene.

The adventure game interface, in other words, is accessible only via player immersion. And the adventure game exerts all its design, not to *offer* that immersion, but to request and require it.

At the high end, this immersion becomes environmental puzzle-solving. How do you resolve a challenge? By understanding the nature of the problem in the game world, and the tools that are available, and all of the physical properties of both. The “physics” may derive from your

real-world knowledge, or from your experience in a fantasy game world, but in either case you are imagining yourself in the situation. If you resolve a challenge by iteratively selecting items from a menu, the game requires no insight and offers no sense of achievement.

But this is *the same* insight and achievement offered, at the low end, by realizing that a book on an altar is worth reading. Or by realizing that “ATTACK” is a viable option when faced with an angry troll. These small realizations grow from your understanding of the world and its conventions, and the small victories teach and lead to the larger ones.

>

We return to our definition, now, armed with a theory of *why* IF is as it is.

Why does an IF game provide a simulated world? Because the player’s understanding of the world must be the primary means of determining what is possible. Why does text IF describe the world in words, and accept commands in (a subset of) natural language? Because the player must be able to close the loop, and infer the importance of command verbs and nouns from the described world (both the words of the output text and their meaning). Why do text IF actions include a core of familiar, conventional commands, a border of less-common commands, and a hazy halo of implied, situational commands? Because the player must be able to begin play with some understanding of the game’s range of action, explore it, and make discoveries—all as a continuum within the same interface.

In a graphical adventure, this continuum runs from clearly delineated, prominent objects in the depicted world to subtle visual cues. These parallel the clear descriptions and oblique references in text IF output. It is worth noting, too, that while the “verbs” of graphical IF are a simpler matter—“click to do anything”—they can be developed into a discoverable continuum as well. (*Myst* begins with the direct identification of the mouse cursor as your hand: to click is to touch. But this is extended throughout the game, as the player explores different situations. Clicking becomes general manipulation, then use of a held object; then variations such as holding, dragging, and waiting become significant.)

Why is guess-the-verb (or hunt-the-pixel) perceived as a design flaw? Because the player is no longer trying to play the game world and is

instead playing the interface. Even when successful, that path offers no immersion, no sense of achievement, and no fun.

Finally, our theory lets us draw boundaries. These may not match up with the commonly understood terminology of “interactive fiction,” but they should be recognizable as meaningful boundaries nonetheless.

Are narrative word-puzzle collections such as *The Fool’s Errand* interactive fiction? Not by this definition. While they have text, characters, setting, and story, they do not simulate a game world, and the player’s commands are not understood as acts within the world.

Are choose-your-own-adventure games (and books) interactive fiction? Again, no; they provide an explicit range of action, with no room for discovery or exploration. They can be played mechanically; nothing in the format requires immersion. (This is not to deny their effectiveness as games. Many, or most, game genres provide explicit ranges of action. I merely point out and try to explain a boundary.)

Are the dialogue menus that appear in so many games (including IF games!) an IF form of interaction? They are text, and they may simulate a character’s state of mind (a character is indeed part of a world). But, once again, they invite the player to choose among options rather than *think up* options. (And thus a great debate about character interaction in IF, uncertain from the Infocom days, continues at full force.)

To be clear: this use of “interactive fiction,” as a term, is not the most commonly accepted. The IF community generally excludes graphical games from its definition. Whether to include CYOA games is not a settled question; opinion and usage varies within the community. And the IF community is of course a hazy, explorable territory in its own right. The phrase “interactive fiction” begins in Infocom’s marketing in the early 1980s, but it reaches us through a chain of evolving discussion groups—not all of which even regarded Infocom-style text adventures as central.

As always, the point is not to explain a term, but to discover the motivation behind the distinctions that players draw.

not that you may remember time

Interactive Fiction, Stream-of-Consciousness Writing, and Free Will

Mark Silcox

Philosophy Department, University of Central Oklahoma

Janet Murray's book *Hamlet on the Holodeck: The Future of Narrative in Cyberspace* is a spirited and unrelentingly optimistic defense of new styles of interactive storytelling made popular in the wake of the PC revolution of the early 1990s. Most of the appeal of Murray's book lies in her lively and engaging descriptions of her own experiences with stories written using these new, mostly digital media. Murray should be applauded warmly for helping to make the attractions of computer gaming, web design, fan fiction, and other pervasive contemporary diversions attractive and intelligible to an audience that is often suspicious of such *prima facie* purely pop-cultural marginalia. Occasionally in her book, however, she waxes somewhat more theoretical and tries to describe the appeal of stories that provide the reader with a more extensive influence on the course of events making up a narrative, by means of a comparison with the traditional aims of storytellers in the Western tradition going back as far back as the pre-Gutenberg era.

In this essay I shall focus my attention on some interesting but problematic remarks that Murray makes about how we should understand the relationship between aesthetics of interactive fiction (hereafter abbreviated IF) and the intentions of writers from just one earlier tradition. Murray compares the artistic aims of interactive storytellers to those of authors such as Woolf, Faulkner, Joyce, and others from the first half of the 20th century who experimented with non-traditional narrative methods in an effort to provide readers of fiction with an extreme close-up of human consciousness itself. Her reason for making these comparisons, and part of what seems to lie behind her considerable optimism about the future of IF, is a belief that stories told through media like literary hypertexts, the Internet, and computer games, at least partly by virtue of the new interface that they generate between the author and his readers, can be expected to deliver a higher level of psychological realism to the conventional reader of fiction.

I want to tell the story of how stream-of-consciousness writing compares to IF in a somewhat different way than Murray tells it. I do not want to suggest that Murray's diagnosis of the relationship between these two vastly different methods of telling stories is simply false, but her approach makes me uncomfortable for a couple of reasons. In the first place, to suggest that one style of narrative can be valued to any extent over another by virtue of its capacity to give a true picture of how the mind works betrays, I think, an approach to practical criticism that relies more extensively upon evaluations of a work's specifically mimetic properties than many in the so-called post-modern era would be entirely comfortable with. This approach influences her critical judgments in some rather surprising ways—her preference for simulation-style computer games like *SimCity* and *Civilization* over so-called adventure games like *Myst* and *Planefall*¹ seem strange given the sheer paucity of textual content in the former, and her apparent willingness to posit a sort of continuity between the appeal of interactive literature and passive media such as television² is rather surprising to those of us who have come to view interactive media as presenting a possible salvation from the cultural penury imposed upon so many in our civilization by the little blue box.

More significantly, though, I think that Murray's convictions about what it would be for a story to achieve a desirable level of psychological realism in the contemporary era betrays some important philosophical prejudices about the relationship between literature and our understanding of the psychological significance of free will that (while they might be almost automatically appealing to many inhabitants of the contemporary scene) it is at least worth bringing out into the open. The less antecedent philosophy we bring to the task of comparing two historically separate and stylistically divergent traditions of storytelling, the more we shall be able to appreciate each individual author's efforts based upon a conception of his or her own peculiar artistic agenda.

As we shall see when we examine some of the remarks that Murray makes about what she takes to be some of the distinctive appeals of interactive writing to the contemporary imagination, she seems at least to think of different methods of narrative as representing so many different strategies for making new kinds of human experience available,

1 See Janet H. Murray, *Hamlet on the Holodeck: The Future of Narrative in Cyberspace* (Cambridge, Mass.: The MIT Press, 1977), pp. 108–109, 213.

2 See Murray, Ch. 9.

in some sense of this term, to the reader. Like many critics before her, Murray appears to regard the structural limitations imposed by a narrative style upon how a particular author tells his tale as instruments for, rather than as impediments to, the achievement of that particular sort of communicative act between author and reader that distinguishes the literary from other forms of expression. Now, my own critical antennae are attuned to a less Aristotelian, and perhaps a more Freudian, wavelength than Murray's. The poet, says Aristotle, must be more the poet of his plots than of his verses, inasmuch as he is a poet by virtue of the imitative element in his work, and it is actions that he imitates.³ For the Freudian, on the other hand, narrative tropes and conventions in a certain sense represent an impediment to literary communication; the essential *ars poetica*, suggests Freud, in his famous essay on creative writers and day-dreaming, is the set of strategies by which the writer bribes us by the purely formal—that is, aesthetic—yield of pleasure that he offers us in the presentation of his fantasies.⁴ These formal properties of a work of literature, says Freud, work to some extent to conceal or to render less accessible the features of a story that make possible the release of still greater pleasure arising from deeper psychical sources.

It seems to me that there are some important insights about the relationship between stream-of-consciousness fiction and interactive literature that can be obtained from thinking of narrative structures as imposing a sort of purely external constraint upon the extent to which genuine psychological realism is a viable goal in the writing of fiction. As I shall try to show through an examination of passages from certain key texts belonging to both of these traditions, the most successful works owe a considerable portion of their appeal to the ways in which their authors actually place on exhibit their detachment from the narrative conventions associated with each type of writing, and by so doing allow their readers, if not to participate directly in the narrative contrivance, then at the very least to peek behind the curtain.

The popularity of what Murray describes as “multiform narratives”—a category that is meant to include not only IF but also the writings of authors as diverse as Calvino, Borges, and Delmore

3 Aristotle, Poetics 1451b27–29, Trans. I Bywater, in *The Complete Works of Aristotle*, Vol. II, Ed. Jonathan Barnes (Princeton, NJ: Princeton University Press, 1984), p. 2323 (1451b26–28).

4 Sigmund Freud, “Creative Writers and Day-Dreaming” in *The Standard Edition of the Complete Works of Sigmund Freud*, Vol. IX, ed. James Strachey (New York: W.W. Norton and Co., Ltd., 2000), p. 153 (italics mine).

Schwartz, Hollywood scriptwriters, and participants in collaborative Internet soap operas⁵—arises from sources deep in our contemporary understanding of the nature of human agency. Such works represent an effort to give expression to the characteristically 20th-century perception of life as composed of parallel possibilities. Multiform narrative attempts to give a simultaneous form to these possibilities, to allow us to hold in our minds at the same time multiple contradictory alternatives. Murray speculates that this way of thinking about human lives as paths traced through a space of conflicting possibilities might be a reflection of post-Einsteinian physics, or perhaps of a secular society haunted by the chanciness of life.

Such hypotheses should perhaps not be dismissed entirely out of hand, but one cannot help feeling that they perhaps represent a somewhat over-precise explanation of what is surely a highly general feature of our human self-understanding—a feature, moreover, whose origins are probably better explained ahistorically. To say that a person recognizes her life as being composed of parallel possibilities surely after all amounts to nothing more than the claim that she recognizes her capacity for acting freely. Are the multiform writers of 20th-century literature really the first group of literary artists to have devised a way of telling stories that adequately represents the influence of free will upon human self-understanding?

In one very provocative and engaging passage of her book, Murray describes the psychological effect of watching a 3D IMAX film, *Across the Sea of Time*, which tells the story of a young Russian boy's visit to New York. Here, she suggests that such media simultaneously proffer to and frustrate in their audiences a new sort of empathetic involvement with the stories that they tell—something quite different from the sort of vicarious identification with characters in a narrative whose fates are determined by the structure of the story as it unfolds:

[t]oward the end of the movie we are on a wonderfully realized street in Greenwich village. . . . A couple in what would ordinarily be the background crosses the street. But there is no background. I am there. My attention is caught, and I want to follow that couple and see what *their* story is. Instead, the camera relentlessly drags me into a bar on a corner with the young boy. Again I see a wonderfully detailed environment. . . . I want to move closer, to lean into the

5 Murray, p. 37.

shot to get a better view, but the camera stays with the dramatic action of the scene, namely, Tomas's conversation with the bartender. I am uncomfortable at these moments because the three-dimensional photography has put me in a virtual space and has thereby awakened by desire to move though it autonomously, to walk away from the camera and discover the world on my own.⁶

What this passage suggests is that something much more subtle—and perhaps more interesting—than the mere recognition that stories take place in environments that are subject to change through the influence of human voluntarism may be going on in the reception of works that demand such a high level of participation from their audiences. What Murray actually depicts herself as having experienced here is a feeling of positive resistance to the flow of narrative in the film. Rather than experiencing the development of a story's plot as an incentive bonus or a fore-pleasure to the liberation of tensions in our minds, as Freud described the plots of traditional romantic narratives,⁷ the audience is explicitly made aware of the plot's role as a purely artificial constraint upon imagination and positive empathy.

Murray seems to believe that her response was an unintended side-effect of the technology used in making the film. My own experience of 3D movies is that plots are almost inevitably quite explicitly contrived and illogical, since they are designed (albeit rarely with much subtlety or self-reflexivity) so as to provide certain sorts of momentary and ephemeral experiential pay-offs—the bird that flies over your head, the sword that seems to reach out dangerously toward you, the curious *objet d'art* thrown into high relief against a backdrop of characters who are merely talking. Would it be too paradoxical to suggest that the deliberate use of such devices to inspire a felt resistance to the determinism of narrative could in fact be a guiding principle in the determination of a literary style?

I think that the self-conscious adoption of this strategy is in fact a distinguishing feature of 20th-century literature and is often what contemporary critics are really talking about when they suggest (as one so frequently hears) that the much broader phenomenon of irony is the dominant trope of modern literary fiction.⁸ One finds many such

6 Murray, p. 48.

7 Freud, p. 153.

8 See, e.g., Wayne Booth, *The Rhetoric of Fiction* (Chicago: The University of Chicago press, 1961) p. 372. Booth's discomfort with the predominance of irony in modern

examples of this sort of attempt to distance one's readers from the determinism of narrative in the works of major writers associated with the stream-of-consciousness movement.

In order better to understand how the aim of psychological realism as pursued by authors of this sort of fiction might lead them quite naturally to pursue a strategy of self-reflexiveness and ironic detachment in the construction of narratives, it will be useful to take a quick glimpse at the writings of William James, the philosopher who first introduced the term *stream-of-consciousness* into the lexicon of popular psychology. In *The Principles of Psychology*, James tried to describe in terms as objective as possible the five most important distinguishing characters of a sentient human being's stream of thought. These, he suggested, were the following:

1. Every thought tends to be part of a personal consciousness.
2. Within each personal consciousness thought is always changing.
3. Within each personal consciousness thought is sensibly continuous.
4. It always appears to deal with objects independent of itself.
5. It is interested in some parts of those objects to the exclusion of others, and welcomes or rejects—chooses from among them, in a word—all the while.⁹

Now, it would certainly be a clear mistake to deny that one of the principal aims of many great writers of the early 20th century who experimented openly with stream-of-consciousness technique was to provide a more realistic depiction of the minutiae of our everyday thought processes. The philosopher Henri Bergson, a contemporary of James and a widely influential thinker during the first half of the 20th century, openly endorsed this view of the aims of narrative. The author, he said,

fiction, which he so often associates with an artificial prohibition in the aesthetics of fiction against so-called authorial intervention, might have been mitigated somewhat if he had taken the view of irony that I have suggested above.

9 William James, *The Principles of Psychology Vol. 1* (Cambridge, Mass.: Harvard University Press, 1981), p. 220.

may multiply the traits of his hero's character, may make him speak and act as much as he pleases, but all this can never be equivalent to the simple and indivisible feeling which I should experience if I were able for an instant to *identify myself* with the person of the *hero himself*.¹⁰

Some authors of this period may perhaps have even held themselves up to the standard of accuracy set by psychologists and philosophers of the period, such as James, Bergson, and others, who employed overtly introspective, phenomenological methods of inquiry for what they took to be the purely scientific characterization of how human beings think. Subsequent critics have pointed out, however, that to suppose that mere psychological realism was the exclusive aim lying behind this sort of literary experimentation would be to attribute to these authors an extraordinary naiveté about the most basic inherent limitations imposed by the conventions of fiction writing, as well as the more general constraint imposed upon the artist by the use of written language itself.¹¹

One useful exercise we can perform to convince ourselves of this fact is to reflect upon the plausibility of each of the five numbered statements given above with the word *narrative* substituted for the expression *personal consciousness* (or the initial pronoun in statements #4 and #5). I do not think it takes more than a little reflection to conclude that under this transformation, the first three statements of James's characterization of the stream of consciousness would under most circumstances clearly be false while the last two only would remain at least relatively plausible. Elaborating upon the first characteristic listed above, James remarked that each human mind "keeps its own thoughts to itself. There is no giving or bartering between them. . . . Absolute insulation, irreducible pluralism, is the law."¹² But the deliverance of a narrative is always at the very least an act of communication between two minds; when it aspires to psychological realism its informational content is furthermore usually to do with a content of some mind that is

10 Henri Bergson, "Understanding Reality from Within" in *Stream of Consciousness Technique and the Modern Novel*, Ed. Irwin R. Steinberg (Port Washington, NY: Kennikat Press, 1979), p. 51 (*italics mine*).

11 For a provocative discussion of the limitations imposed by the nature of language itself upon attempts to present the stream of consciousness in all its psychological richness, see Erwin R. Steinberg, *The Stream of Consciousness and Beyond in ULYSSES* (Pittsburgh: University of Pittsburgh Press, 1973), pp. 161–177.

12 James, p. 221.

neither that of the teller nor that of the hearer. If we accept Aristotle's characterization of narrative as requiring at least some "probability [or] necessity in the sequence of its episodes"¹³ as being at the very least a rough approximation of the truth then we must furthermore reject the re-interpreted version of condition #2, since according to James when accepted as a true description of personal consciousness it clearly implies that no state once gone can recur and be identical with what it was before, and hence that no psychological state considered in abstraction from its objects can be thought of as related probabilistically in any way at all to any other state that an individual might happen to enter into.¹⁴ Statement #3 is a little more tricky, since it is less than clear precisely what James means by describing thought as continuous in character. He might on the one hand be referring to the fact that much of everyday conscious reflection is of a non-linguistic character—an intuitively appealing but philosophically rather controversial claim to make about how we experience the flow of our own thoughts. Or he might be making the more challenging claim that no human being ever does anything without the accompaniment of a consciously experienced internal flow of thoughts, even when asleep, entranced, or absorbed in physical work. In either case, extreme examples of experimentation in stream-of-consciousness writing such as *Finnegans Wake*, Russell Hoban's *Riddley Walker*, and some of the later novels of Samuel Beckett, which forego even the conventions of regular syntax and word individuation, might conceivably stake a claim to providing partially adequate representation of the continuous qualities of human thinking. But this is surely at most an unattainable ideal, one that can at best be approximated very roughly through the fundamentally discontinuous medium of human language.

When one looks at a few supposedly paradigmatic examples of psychologically realistic stream-of-consciousness writing, what one finds is that in fact a curious sort of path is negotiated between fairly regular

13 Aristotle, p. 2323 (1451b35).

14 James, 224–225. The idea that a single human mind could simply never have two strictly identical thoughts is, of course, quite philosophically problematic, and it should perhaps be emphasized here that most contemporary philosophers are extremely wary of a methodology that relies so exclusively upon introspection when it comes to the characterization of mental content. For an interesting and sharply divergent account of the nature of mental content, see Hilary Putnam's famous essay "The Meaning of Meaning" in *Language, Mind and Knowledge*, Ed. K Gunderson (Minneapolis: University of Minnesota Press, 1975).

methods for the delivery of narrative information about an individual's conscious history and a very particular type of digression from the flow of narrative. Consider, for example, the famous opening pages of Joyce's *A Portrait of the Artist as a Young Man*. This novel begins with a curious barrage of sensations—snatches of half-coherent song (“Oh, the green wothe botheth”), fragments of a story being read aloud to the protagonist, and the observation that “When you wet the bed first it is warm then it gets cold.”¹⁵ The sheer novelty of this style of writing, and the extent to which it contrasts with more conventional first-person styles of narration, has led many critics to praise the author by remarking upon his utterly original methods for capturing the raw feel of our inner, reflective experience as human beings.¹⁶ But this can easily lead one to overlook certain crucial ambiguities. For in the first place, it is actually made quite clear by Joyce (through the use of asterisks that divide up the initial sections of the book) that this passage is meant to serve, not as a representation of one single, continuous episode in the conscious life of Stephen Daedalus, but rather as a sort of representative sampling from Stephen's infant thoughts. It is also worth noting that the glimpses that one does catch of other characters, even in the space of a couple of pages of Stephen's singing father and his pious aunt, are clearly meant to resonate with other passages later on in the novel, after Stephen himself has developed to an extent that enables him to achieve deeper insights about the personalities of his immediate family. And in a striking passage toward the end of the novel, the adult Stephen philosophizes to a friend about the nature of epical narrative in a way that utterly lays bare the contrivances involved in the novel's early sections:

The simplest epical form is seen as emerging out of lyrical literature when the artist prolongs and broods upon himself as the centre of an epical event and this form progresses till the centre of emotional gravity is equidistant from the artist himself and from others. . . . This progress you will see easily in that old English ballad *Turpin Hero*, which begins in the first person and ends in the third person.¹⁷

Perhaps the most interesting example of a text in the stream-of-

15 James Joyce, *A Portrait of the Artist as A Young Man* (New York: The Modern Library, Inc., 1916), pp. 1–2.

16 See, e.g., Anthony Burgess, *99 Novels: The Best in English Since 1939: A Personal Choice* (New York: Summit Books, 1984), p. 8.

17 Joyce, p. 252. Note that the ballad *Turpin Hero* was clearly the source of the title for an earlier draft of Joyce's own novel, viz. *Stephen Hero*.

consciousness tradition in which this balancing of pure psychological description with the development of a cohesive, structured narrative is brought into high relief is the second section of Faulkner's *The Sound and the Fury*. For this portion of his masterpiece, Faulkner places us inside the mind of Quentin Compson, a troubled Harvard undergraduate, on the day of his suicide. We watch as he goes through a meticulous series of preparations for his own death, which are interrupted comically by some confrontations with young children and hostile immigrants and a corrupt rural judge. As these events unfold, Quentin's imagination drifts back and forth with apparent aimlessness between his two principal obsessions in life—obscure memories of an ambiguous, possibly incestuous relationship with his sister and reflections upon the various aphoristic pronouncements that made up his father's darkly fatalistic worldview.¹⁸

What is striking about these passages from Faulkner's novel is the almost rhythmical variation in his depiction of the flow of Quentin's thoughts, from purely subjective, momentary ephemera of consciousness to the delivery of fragments of information about his environment and the people he meets—details that are crucial to the reader's understanding of what is going on in the world around him. A number of devices are used to achieve this delicate balance in the narrative between Quentin's free-associative mental ramblings and more deliberately expository passages. The most explicit and straightforward of these devices is Faulkner's use of italicized text, which always begins when an actual incident in Quentin's environment has called to mind some vaguely remembered image or idea from his troubled past:

The shell was a speck now, the oars catching the sun in spaced glints, as if the hull were winking itself along. *Did you ever have a sister? No but they're all bitches. Did you ever have a sister? One minute she was. Bitches. Not bitch one minute she stood in the door*

I found the gasoline in Shreve's room and spread the vest in the table, where it would be flat, and opened the gasoline. *the first car in town a girl Girl that's what Jason couldn't bear smell of gasoline making him sick then got madder than ever because a girl Girl had no sister*¹⁹

18 William Faulkner, *The Sound and the Fury* (New York: Random House Inc., 1954) p. 113.

19 Faulkner, p. 213.

There is also an element of graceful and quite deliberate irony in the mixing of Quentin's recollections of a night spent in the rain with his sister with his attempts to shake off a small, silent immigrant girl who follows him around as he makes his way through the outlying areas that surround the college:

"You're just a girl. Poor kid." There was a path, curving beside the water. Then the water was still again, dark and swift. "Nothing but a girl. Poor sister." *We lay in the wet grass panting the rain like cold shot on my back. Do you care now do you do you*

My Lord we sure are in a mess get up. Where the rain touched my forehead it began to smart my hand came red away streaking off pink in the rain. Does it hurt [. . .] "There's town again sister. You'll have to go home now. I've got to get back to school. Look how late it's getting. You'll go home now, wont you?" But she just looked at me with her black, secret, friendly gaze, the half-naked loaf clutched to her breast.²⁰

Finally, there is the repeated reference to devices for the measurement of time throughout this section of the novel. Quentin's tendency to be repeatedly distracted from his thoughts by the ticking of clocks and the tolling of the hour provide a neat metaphor for the tragic course of events that provide a backdrop for the unrelenting play of his obsessional thoughts. The chapter's central symbol, a pocketwatch given to Quentin by his father "not that you may remember time, but that you might forget it now and then for a moment and not spend all your breath trying to conquer it,"²¹ provides us with what is perhaps the clearest example of an author's attempt give us a glimpse of the difficulty that he himself faces in trying adequately to represent the continuous, ineffable diachronic play of a character's innermost while at the same time delivering a structured, coherent narrative.

Our aim in discussing these few brief examples of stream-of-consciousness writing has been to bring to light a crucial but often neglected element of ambivalence evince in the narrative styles adopted by Joyce and Faulkner toward the task of providing a realistic depiction of the flow of the elements of privacy, flux, and continuity in the flow of human thought. When we turn from stream-of-consciousness fiction

20 Faulkner, pp. 169–170.

21 Faulkner, p. 93.

to IF, we find that some of the most skillful and reflective interactive storytellers exhibit a strikingly similar sort of ambivalence toward the task of providing an accurate representation of what it feels like for a character to exercise free will in a way that influences how the story he or she inhabits will reach its conclusion.

Now, in traditional IF of the sort made popular by the game company Infocom in the early 1980s, the player/ reader influences the course of events in a story through the entering of simple instructions into a traditional DOS-style command-line interface. So, for example, a story might begin with the following passage:

You are standing in the upstairs bedroom of the small urban townhouse that you share with your wife and kids. To the north is the house's upstairs corridor. Directly underfoot, you notice that there is a large, gamey heap of your own recently worn underwear. The bed is unmade, and sunlight streams in through a small window to the south.²²

followed by some command-line indicator (e.g., >) into which an instruction is typed (e.g., “wear underwear,” “close window,” and so forth). Computer programmers who design tools for the writing of interactive stories—languages such as TADS and Inform and all of the various extensions thereof—try to provide the author with as many resources as possible in the design of a parser for these command-line inputs of a reader. One of the indisputable criteria by which interactive stories are evaluated by fans of the medium is in terms of the degree of inventiveness that can be tolerated from the player. If a player were to type “wear underwear” at the conclusion of the passage given above, only the very shoddiest of IF stories would respond with something like “I don't understand that statement,” or “Be more specific.” If the player were to input something more esoteric, though, e.g., “bask in sunlight” or “contemplate navel,” only the very most polished and carefully designed IF parser would be likely to reply with a remark that did not sound thoroughly vacuous or formulaic.

These imperatives that govern the construction of programming tools for IF serve as evidence for the plausibility of Murray's claim that one of the principal aims of interactive storytelling should be the intimation of diverging possibilities that we experience as free will.²³

²² Taken from my own TADS game *Sacrobosco's Book of Wonders*, unreleased.

²³ Murray, p. 281.

From the programmer's point of view, it is clearly the case that the more possibilities there are—i.e., the fewer formulaic, repetitive, and narrative-stalling parser responses—the better. There is something of a danger, though, in trying to formulate aesthetic standards for the evaluation of these new media by focusing to much upon the aims of programmers and technicians as though these somehow represented a distillation of what is essential to the task of actually coming up with an interactive narrative. It is certainly true that most of the best IF authors are inclined to put enormous amounts of work into the design of highly responsive and imaginative command-parsers—when one looks at the source code for games by experienced IF writers like Mike Roberts, Graham Nelson, Cody Sandifer, and Adam Cadre, it is often quite astonishing how much time has been spent coming up with witty, surprising, and thematically salient responses for some manifestly unlikely player inputs. But many of these authors' work also stands out for another, quite different reason. Some of the most successful IF composed in recent years is notable for the way in which it works systematically to frustrate the expectations of even seasoned players and readers who expect such works to behave in ways that are at least broadly analogous to the way that our own personal environments respond to the exercise of our free will.

At the simplest level, a game may be structured in such a way that the principal task that the reader is instructed to accomplish at the start (e.g., to find a treasure, escape from a maze, or retrieve some valuable or symbolic artifact) turns out either to be impossible or else is frustrated at the very last stage in the narrative. This device is used to great comical effect in Mikko Vuorinen's *King Arthur's Night Out* (1999) and Leon Lin's *Kissing the Buddha's Feet* (1996). A somewhat more extreme example of this general strategy of bringing to light the restriction upon our exercise of free will can be found in Cameron Wilkin's chilling IF story *Bliss* (1999), in which the second-person protagonist's aims, as manifested by the player's own carefully guided inputs, gradually reveal themselves to be entirely delusional in nature.

Other writers have come up with subtle methods of drawing their readers' attention to the inherent limitations of the traditional IF command parser, in order to bring to light some curious quirks and paradoxes inherent in the human self-conception as a species of free, autonomous agents. In Adam Cadre's award-winning IF story *Photopia* (1998), there is a short but very moving scene in which the main character recounts a dream she has had that the player is able to

recognize as a premonition of her own swiftly approaching death. The dramatic irony of this scene is intensified considerably by Cadre's use of a fake command line (represented by the symbol [->], below)—for the duration of the dream the only command that the parser will respond to is a carriage return, but the effects of command entry and response are still displayed as though they had actually taken place:

“Come on, Wendy,” Alley says. “You know the rules. It’s way past your bedtime. Your parents’ll be home soon and if you’re still up it won’t look too good for either of us. I’ll be right here if you need anything.” She pulls a book out of her backpack and starts in on her homework.

>talk to alley

Please select one:

- (1) >ASK ALLEY ABOUT HER HOMEWORK
- (2) >ASK ALLEY FOR A GLASS OF WATER
- (3) >ASK ALLEY WHERE SHE GETS HER IDEAS
- (4) >ASK ALLEY ABOUT HOW I GET HOME
- (5) >ASK ALLEY ABOUT THE QUEEN

Select an option or 0 to say nothing >> 3

Alley puts down her pencil. “That’s not a question people are supposed to be able to answer,” she says. “But I can. All that stuff, every bit of it, came straight from these weird dreams I’ve been having. And they’re not the only ones. There’s another... are you sure you want to hear about this?”

“Yeah!” you say.

“Okay,” Alley says. “It starts like this...”

[->]

[->]

In a dark place

I open my eyes, and I am in a cold, dim, lonely place. I blink,

thinking my eyes are still closed, but then I realize that it is not pitch black: there is a dark purple fog billowing all around me, so close to black as to make nearly no difference. There is a faint purple glow to the stone floor, too, though I might not even notice it if it weren't holding me up.

[->]
>EXIT

I try to find my way out of this place, but no matter which direction I try, or however far I go, it's all the same. Eventually -- and maybe it's minutes, maybe it's days -- I can suddenly make out a vague shape in the fog.

[->]
>LOOK AT SHAPE

It's strange, because parts of it look distinctly human, but others are just as clearly not. I come closer, and I discover the answer: it's a person sitting in a huge throne made of rock. No, that's not right. It's not MADE of rock -- it IS a rock, one single stone in the rough shape of a high-backed chair.

[->]
>LOOK AT PERSON

Her face is turned away from me, but I can tell that it's a young woman, dressed in long flowing purple garments. Dressed in royal purple, sitting in a throne, I can only assume that she's a queen or princess of some sort, but since I don't believe in monarchy, and certainly am not one of her subjects, I don't feel the need to bow or scrape or call her "Your Majesty". "Hello?" I say.

[->]
>WAIT

After a moment, she turns to face me. And that's when I start to get really scared.

She has my face.

(transcript from Adam Cadre, *Photopia*)²⁴

This method of capturing the fundamentally illusory nature of free will as it is experienced in dream states is not only utterly original but in addition could not have been brought about without the game's player having been lulled into a false sense of security about the scope of her influence on the course of the narrative by the game's very intricate and well-designed parser.

Perhaps the most surprising and ingenious example to be found in IF of this sort of deliberate distancing of the reader from the very features of the medium that provide the reader with the apparent opportunity to direct the course that a narrative takes can be found in Paul O'Brian's game *Local Asynchronous Satellite Hookup* (2000). The central science-fictional conceit of this work is that the player is in control of a robot that itself responds to exactly the set of commands that the parser itself is programmed to understand. At the start of the game, the robot is sent to an archaeological site to look for treasure—the former home of a historian of the antebellum South and a pioneer in virtual reality technology. The game's command line is thus supposed to represent the interface between the player's persona—a prospector located somewhere outside of the site itself—and the robot itself. At first glance, this conceit looks to be merely a clever way to build the inherent limitations and finitude of traditional IF command parsers into the very structure of the narrative itself, and some of the responses one gets during the first half of the game to typed-in commands support this reading, e.g.:

Parlor

I am in a spacious room that I presume was once the Parlor of the Percy home. A sofa divides the room in two, with one end pointed towards the doorway to the east and the other pointed towards the fireplace in the west wall. It is the only piece of furniture in the room now, but scuffs on the hardwood floor suggest that it was once accompanied by many others. There is also a doorway leading

24 I am aware of no universally accepted formal method of academic citation for amateur IF stories; rather than risk making one up I shall merely record the URL of the Interactive Fiction Archive, an FTP site maintained by Volker Blasius where all of the games mentioned in this essay can be found. The site address is <ftp://ftp.gmd.de/if-archive/> <ftp://ftp.gmd.de/if-archive>.

north.

On the sofa is a corpse (on which are a leather jacket and a white gold bracelet).

>examine corpse

This is the body of a human adolescent female. She was lying on her face when she died, which was obviously some time ago. The body is little more than a skeleton, with a few bits of desiccated flesh on skin still clinging to the bone. It is held together by a latticework of decaying tissue, and would no doubt fall apart if touched. Most of its clothes have rotted away, but it still wears a leather jacket and a bracelet of white gold.

>lift corpse

The skeleton collapses as I try to move it, leaving only a pile of bones on the couch along with the corpse's former possessions.

>x bones

This pile of bones was once a human skeleton.

>pray

I do not know how.

(transcript from Paul O'Brian, *Local Asynchronous Satellite Hookup*)

At the story's midway point, however, the robot discovers a curious collection of inscrutable machines in the attic of the long-departed couple, and the following miraculous transformation occurs:

At one end of the attic, the cabinets meet in a sort of vertex, and at the point of this vertex is a booth.

>examine booth

The booth, like the cabinets, is connected to the cube in the center

of the attic. It is a three-quarter cylinder, a little taller and wider than I am, with the open section facing out. Its wall is made of some sort of opaque black plastic, and I can see some controls inside.

>enter booth

I have gotten into the booth.

In the booth I can see some controls.

>examine controls

The controls seem to consist of a number of readouts and indicators, and a prominent button.

>examine indicators

The readouts and indicators appear to be active, but I can make no meaning from them.

>press button

I push the button, and the cylinder of the booth closes, cutting off my exit along with all outside light. Inside, it is pitch dark, but before my headlamp even activates, I feel a strange sensation, as though my body were falling, falling through thick air and velvet murmurs. Systems report no change of altitude, but the freefall persists inside my brain. It is not an unpleasant sensation. In fact, as it continues I note sensory functions sharpening and previously unused brain sections coming online. It's so strange -- one part of me feels like it is expanding at a rapid rate, touching mental regions I've never used before, while another part --

[the one speaking now, is detached and observing the changes. I find that I must erect more and more mental structures to maintain my standpoint as an observer]

while the core of me pulses outward, soaked in the vividness of these new

[I can only call them]

emotions!

[It also appears that a type of integration is operating. For example, the metaphorical, figurative language in which I spoke when beginning to describe the falling sensation]

is completely unlike anything I've ever said before, at least anything I can remember,

[and my memory of]

my childhood

(transcript from Paul O'Brian, *Local Asynchronous Satellite Hookup*)

The player has unwittingly guided the robot into a device designed to simulate the experience of a black slave in the deep south. Here O'Brian is using the very features of the medium that Murray characterizes as providing for a more realistic depiction of the experience of free decision-making to illuminate the fragility of our everyday distinction between the autonomous and the automatic. The first few moments of the simulation take things even further:

My master bursts violently into the room! Instantly, he is upon me, taking painful hold of my wrist and shouting with rage, "God-damn it! What the hell are you doing up here, you God-damned nigger cunt? Stealing? Are you STEALING?" He runs his hands roughly all over my body, sparing me no indignity, before acknowledging that I carry nothing.

"Caught you before you could take anything, didn't I?" he bellows, inches from my face.

"But what else have you done?" He drags me along as he searches all of the upstairs rooms, ferociously surveying for any piece out of place.

"All right," he says, "I caught you before you could do whatever it was you were planning. But now I'll make SURE you never make

any such plan again!”

With this he drags me downstairs, down further and further through the house, until we finally reach the smoldering hot kitchen. Once there, he pulls the cord from around my waist and yanks my simple garment over my head, leaving me naked before him. He then binds my wrists with the cord. Forcing me up onto a stool, he hangs them over a hook affixed to one of the ceiling joists.

He kicks the stool away and grabs his riding-whip in one smooth motion, then pauses a moment to regard me as I hang in front of him, naked and helpless. He draws his arm back, the whip descends, and
AAAAAAAAAAAAAGGGGGGGGHHHHHHHH1011101010011000
0101001101010110101101011101111100000010
AA
101101011011000000101010011111101101011110111101110110
000101001101101101010101
Now I
10101100001010111101010111010110100010101101010001001011
10100101001101101010 help
10100101000010011011101101101011101101110111100000101
011010000101010111101
can’t 1011101 can’t 1011100000101110110
AAAAAA10010000011100000000000000000000000000
AA
AAAAnd now he draws back, and the whip
again AAAAAAAAAAAAAA descends and AAAAAAAAAAthe
PAIN is liAAAAAAAAAke nothing
I’ve AAAAAAAEver imagined.

(transcript from Paul O’Brian, *Local Asynchronous Satellite Hookup*)

In this passage, O’Brian succeeds in introducing an unexpected and disturbing element of ambivalence into how the player views what happens to this strange entity that the game has up to this point taught him to regard as merely an extension of his own will.

It seems clear that many of the authors in the current, post-commercial era regard the phenomenon of free will as a philosophical

conundrum—a dubious phenomenon the exercise of which can be as elusive and difficult to detect as any other value-bearing experience in the course of our mental lives—rather than a merely self-evident feature of our mental lives, one that all of us can identify by means of introspection aided by a knowledge of modern science, and that literature is merely required to capture with as much photographic verisimilitude as possible.

It has come to be accepted as something of a truism in some critical circles that any account of what makes a particular genre of writing valuable or worth practicing carries with it some sort of implicit historical narrative.²⁵ What such a narrative is supposed to explain is how the type of writing in question emerged from earlier models and how its peculiar appeal can be traced back through the intentions of writers belonging to a broader tradition. Murray is sensitive in her book to the need to provide such a narrative in order to explain the attractions of interactive literary media against a broad historical backdrop:

After reading the wildly digressive monologue of Sterne's *Tristram Shandy* or the exquisite moral discriminations of a Henry James heroine or the richly textured stream of consciousness captured by Virginia Woolf, it is hard to believe that we could penetrate any further into the workings of the mind. But twentieth-century science has challenged our image of ourselves and perhaps outrun our ability to imagine our inner life. A linear medium cannot represent the simultaneity of processing that goes on in the brain—the mixture of language image, the intimation of diverging possibilities that we experience as free will. It cannot capture the secrets of organization by which the inanimate somehow comes to life, by which the neural passageway becomes the thought.²⁶

It is the curiously cumulative character of this narrative that we have taken issue with here—the notion that Murray appears to have that the history of literary accomplishment can be represented as a series of increasingly accurate or detailed attempts at a faithful rendering of the workings of the human mind. It seems to us that while she is certainly right to insist upon the enormous cultural importance of these new media, their appeal cannot be adequately described without reference to means that they afford to writers for visibly detaching themselves from

25 For a clear and interesting philosophical exposition of this view about the importance of historical narratives to aesthetics in general, see Noel Carroll, "Art, Practice and Narrative" in *The Monist* 71 (1988), pp. 140–156.

26 Murray, p. 281.

the conventions of narrative, in a way that allows readers to experience all the pleasures associated with a good story while at the same time sustaining a reflective attitude toward the subtle manipulations involved in any type of narrative communication. If there is a story to be told about the historical emergence of IF that conforms to these criteria, then it must be just one small chapter in the overarching tale (one that perhaps has yet to be told with complete accuracy and exhaustiveness) of how detachment, self-reflexiveness, and deliberate irony became the dominant tropes of contemporary fiction.

2 Brief Dada Angels

Ryan Stevens, writing as Rybread Celsius

Brief History of IF with Quasi-Mesostic

IneXplicable at first, Yet inevitable later [Zork II for instance]: two doZen of that storY of late siXteenth-centurY London, three doZen are familiar child's puZzle-books, and still largelY uneXplored land in between. Cited bY Designer and Date MagaZine: "the grail of puZzle-free Yet interactive literature seems for teXt: while the industrY puZzles, nearby caves were being hotly seiZed endlessly! The Great OnyX Cave BY StandardiZed Commands was a textual maZe game." (TerrY Winograd, 1972) The aim was to eXplore and onlY a few of the puZzles, FROzEN RIVERS OF ORANGE STONE, AN AWKWARD CANyON AND A GOOD PASSAGE ExIT, freelY adapted Zazie, a craZy artificial intelligence LaboratorY, more complex than liberallY neat puZzles. In Tracy Kidder's PulitZer prize-winning book The Soul of a New Machine (1981) a journalist's eYe-view to eXplore. Rooms, currencY of a recurring puZzle to become "wiZards" in a "fantasY siX" at the UniversitY of Alberta, Canada. WiZards' houses, maZes with limited batterY power with eXits in all directions, and so forth, recur endlessly in "Zork". A puZzle-plaY boX, is actualY froZen rivers of improvised imitation called "Zork", which at its height six years later emploYed.... you. Would eXpect You were running LiZ Cyr-Jones MagaZine, manY PhoeniX Computer LiteracY magaZines, a doZen others: mutuallY TeXt. Adventures were an exception, reallY, "Zork I" was offered for RendeZvous RaY! DeluXe. And few plaYers can tolerate a priZe. For the first priZe would take quite another strategY: eXtol the verY computeriZed Zork. MaY the uneXpected buY even more haphaZardly stocked RendeZvous with LarrY.

Dancing on the Head's of Needle-Nose Pliers

Adam Cadre writes real books. Andrew Plotkin writes software. Graham Nelson himself is a poet. Smart people, able to code, write, spell, etc.

They have ideas, and they express them. Myself, I have ideas. And try to express them. But it's like some sitcom father trying to get all the clothes into suitcase. They overflow, wrinkle and escape. What's left is some sad ready-made. The line between a bad game and a Dada game need not exist, they share the same Venn diagram. But the attributes expand. There is the sense of the uncanny and stupid, without stepping into the realm of surreal (a more fleshed out plane), but ghosting its border. There is also a futurist quality to them, with the pieces usually coded very fast with very little planning or forethought. Stream of consciousness would be an inaccurate prison, but a perfectly acceptable adjective. Grammar mistakes and coding inefficiencies paint miniscule portraits of the author's states. Does a manic room live its life in a run-on sentence? Is the virtually lifeless NPC a metaphor for a single use tool? Are these non sequitur interactions jokes, fit somehow in, or allude to the phantasmagoria of the author's logic? Perhaps. Atmosphere leaks in in peculiar ways and nonsense gives way to its own logic. A true Dadaist game is an imprint of the author's mind, pressed again and again, denting each previous pressing. This shape makes any title a maze, despite how the rooms are linked, and the viewer go beyond the "Guess the Verb" frustration to a leap of intention, personality and connection. Despite the headache and grievances, those who work through a piece can often feel a brief outline of beauty, for two sentences will rub just right way, or a random event juxtaposes itself within a turn of entering an instant death room. The aesthetics of the absurd have the uncanny and the stupid as well. And this is where the Dadaist piece is separated from the truly dismal. For the Dadaist piece will have at least a small contingent that sees some sort of merit beyond satire, even if it can't be argued well, or, indeed, at all. The speed of thought transduced to words per minute, compiled and shot through the world: Nonsense is a right.

Object Relations

Graham Nelson

The archetypal dull object is a small pebble. It has properties, true: it is portable, inedible, visible, can be thrown, is without odor, color, or taste, lacks consciousness, is not flammable, has no interior, is opaque, can be put in pockets. But these are exactly the characteristics that we discard from our impressions as being unremarkable. They take a good deal more simulation for interactive fiction than one might expect, and indeed if a pebble failed of any of them, it would become interesting for that reason, but this is a long-solved problem. The corresponding problem of simulating people, using any machinery other than people (that is, actors), is a different matter, and in the early years of IF it sometimes appeared hopeless. Numerous old-school designers eschewed characters altogether, or had them sleeping, in ghostly form, busy potting daisies, or in some other way evading their reasonable obligations to observe and participate. New-school IF, an approximate label that we might date from around the turn of the century, has made a much better job of all this. Recognizable social situations now stand alongside or even instead of quests to explore or collect, and inanimate objects no longer seem universal ingredients of IF. Modern writers are less interested in intricate puzzles; some objects exist, in IF, in order to serve as components of such puzzles; and so there are somewhat fewer objects.

But in thinking only of tokens in a game, it is easy to overlook the richly nuanced possibilities of objects as used in conventional fiction. Objects are by no means too crude to carry a narrative of human situations. Leaving aside the physical behavior of objects, since those are easily understood (a chair supports the character sitting on it, but this is not very remarkable from a literary point of view), we consider three largely distinct ways in which fictional objects function in story-telling: first evocatively, in representing character and feeling; second as objects of desire, arousing those feelings in reader and characters alike; and thirdly as a kind of carrier, a physical signal that intercedes between characters negotiating their feelings.

Evocative Objects

Few would deny that other people are the sweetness of life. The evocative use in narrative of objects rather than characters at first seems a saccharine, a sugar substitute. Literary criticism has a variety of terms for the way particular objects come to represent, or stand in place of, an inchoate mass of feelings and social structures: for instance, metonymy (“The bench rebuked Mr. Dershovitz” = the judge sitting at the bench did) and its close cousin, synecdoche (“She loved that driving seat” = she enjoyed using the whole car). Sometimes these devices are elegant plays on association, fraying away into hackneyed figures of speech; sometimes they are a verbal shorthand for rafts of understanding whose elaboration would be unnecessary, tiresome, and perhaps also difficult. (“Downing Street today denied any rift between the Health Secretary and the Chancellor.”) What makes metonymy more than a linguistic convenience, where figurative objects substitute for verbal explication, is that real objects also play a role in real social patterns. Jewelry serves us in many ways, but a diamond engagement ring is almost wholly metonymic. If I were to burn the degree certificate that hangs on my dentist’s wall (let us say), he would neither forget his trade nor lose his right to practice it, yet the actual certificate means something to him. The Royal Navy, though sympathetic, will not replace my grandfather’s recently stolen war medals, because the doctrine is that there can only be one metal cross. A spare would be a fake. Essentially these are tokens to make tangible something that seems terribly large to our lives (a marriage-to-be, a seven-years’-apprenticeship, a dreadful time on the Murmansk convoys), yet which has no substance. Much of the sense of unfairness in grieving comes from the appalling way that a sudden absence seems to affect nothing else: not the trees in the garden, not the books on their shelves, not the crockery to be washed up. We know that the world has been transformed, and yet the world does not.

The title of this chapter is also the title of three from Julian Barnes’s novel *Metroland* (1980), the sentimental education of an English schoolboy. Each of three phases is concluded by “Object Relations,” a long room description itemizing his habitual things. At the end of adolescence:

A crocheted mat; two hairbrushes so stuffed with hair that I have abandoned them and taken to a comb; clean socks and white shirt for the morning; a blue plastic knight, made up from a model kit

given me by Nigel one Christmas, and left half-painted. . . My watch, which I despise because it doesn't have a second hand. A Fablon-covered book.

The scheme is explained in one of Barnes's trademark deft repetitions: "Objects redolent of all I felt and hoped for; yet objects which I myself had only half-willed, only half-planned. Some I chose, some were chosen for me, others I consented to. Is that so strange? What else are you at that age but a creature part willing, part consenting, part being chosen?" But the narrator is increasingly conscious of these chosen things, which continue to lay him open. Five years later here is what he leaves behind at the end of a summer in Paris supposedly spent researching theatrical history:

On the desk, a line-up of bottles of spirits, one for each calvados I'd consumed. Beside it, a wastepaper basket which I had, with deliberate negligence, failed to empty; though I hadn't actually planted evidence, I was certainly conscious of what was in it. A copy of "Hara-Kiri" ("journal bête et méchant") and one of "Les Nouvelles Littéraires"; a theatre programme which happened to be a duplicate; various rough drafts of stories and poems; a few drawings (the best rejects); a couple of letters from my parents; some tangerine peel; and a note from Annick, left one morning when she had gone off early.

Nine years later still, his objects have shrugged off their accusing looks, their peacock-display, and are trusted guides:

Objects contain absent people. A poster, flat and pinned, of the chateau of Combours (where Chateaubriand grew up) narrates a holiday four years ago. A phalanx of a dozen glasses on a shelf implies ten friends. A feeding-bottle, stored high on a dresser, predicts a second baby. On the floor next to the dresser is a plastic travel-bag with a bright sticker we bought to amuse Amy: "Lions of Longleat", it says, with a picture of a lion in the middle.

These room descriptions are subtler than they look, that is, are not simply opportunities for Barnes's writerly pleasure in capturing domestic items by a fresh phrase. The travel-bag with its sticker, for instance, is what has become of the narrator's old suitcase, with his imagined baggage labels from foreign parts: already it is now his daughter who will

travel, no longer himself. Objects “contain absent people,” they narrate, they imply, they predict.

So suggestive are juxtapositions of items, like still lives, that even when not openly chosen or random they may still be thought to contain, narrate, imply and predict, as in dictionaries for the interpretation of dreams. The surrealist movement certainly thought so. Rene Char’s prose poem “Artine” (1930, trans. William Rees): “*In the bed prepared for me . . . there was no prison door, there was the taste of bitterness, a glazier’s diamond, a hair, a day, a broken chair, a silkworm, the stolen object, an overcoat chain, a tame green fly, a branch of coral, a cobbler’s nail, an omnibus wheel*” (the list is italicized in Char’s original). Of course this was nothing new in art. The theological symbolism of objects in the Old Masters is a field of study in itself. A few years back, I asked an expert why the artist had placed an octagonal glass of violets on the floor in a painting of the circumcision of Christ. “Well,” she said, “some people think it signifies the virginity of Mary, but I think he was just showing off. Most of the painters in Venice at the time couldn’t really do transparent glass.” Objects should indeed give us pleasure, besides any meaning they stand for. Sometimes that pleasure becomes urgent.

Objects of Desire

Objects can be more than substitutes for absent friends: they can be friends in their own right. It is satisfying to lay a two-foot-high brick wall, or to rake over ashes in the morning. We do not like to part with trusted bicycles, with whom we have shared many miles, until they begin to betray us by constant and wilful breakdown. A screwdriver set changes from a nagging reproach to an obscure source of congratulation when it is used for about the fourth or fifth different household job. “Hooray!” as Helen Fielding’s heroine Bridget Jones might write in her diary: “Am successful intellectual/handywoman hybrid.” Beyond our own nests, and the intimacy that attaches to anything in them, we find intriguing whatever has had a history of human contact, because it is old, or hand-made. If we viewed frankly bad furniture and cookware in functional terms, there would be no antiques dealers. A coin minted in 1921 has a story the more interesting because it cannot be known. Better yet is the tale of something that struggles to come down to us, the lost treasure:

Until!—finally, and at long last—mangled and tattered like a dog

that has fought its way home, there falls across the threshold of the Italian Renaissance the sole surviving witness [. . .] the Verona Codex of Catullus; which was almost immediately lost again, but not before it was copied with one last opportunity for error. And there you have the foundation for the poems of Catullus, as they went to the printer for the first time, in Venice 400 years ago.

So goes an outburst in a 19th-century scene from Tom Stoppard's play "The Invention of Love" (1997). Legend has it that the Verona codex, the last copy anywhere of Catullus, was found bunging up a wine-butt in the cellars of a monastery. There are reasons to doubt this, but the legend is itself revealing, as was the recent keenness of newspapers to celebrate the "discovery" of a "lost" children's story by Sylvia Plath, which, it transpired, had not only been published before and by Plath herself, but had even been erroneously "rediscovered" before. That we love such tales enough to over-egg them is the more curious since they are not, in fact, especially rare. Menander, once thought a second Homer, much imitated, copied by innumerable scribes, quoted elliptically even in the New Testament (I Cor. 15:33), was wiped out altogether by the Christian dark age, not counting some aphorisms and somewhat free Latin translations. His fragmentary survival today hangs chiefly on two pieces of rubbish: a wrapper lining a jar buried in a Cairo lawyer's house, excavated in 1905; and the papier mâché used by municipal workers in 3rd-century BC Alexandria to remummify a number of dilapidated bodies, unseen by human eye until it was unpeeled at the Sorbonne between 1906 and 1965. It is all the tale of Perdita, the daughter washed ashore as a baby after a shipwreck, to be raised by shepherds, with nothing but a keepsake in her tiny hand.

Such exquisite objects of desire are central to genre fiction. In the "literary entertainment," for want of a less patronizing term, the treasure may indeed be a manuscript: in Eco's *The Name of the Rose*, a lost book by Aristotle; in A. S. Byatt's *Possession*, a secret exchange of love letters between Victorian poets; in Henry James's *The Aspern Papers*, the Aspern papers. Or we may find ourselves in the cold war, with the microfilm of "atomic secrets" hidden in a Russian doll; or in the English 1930s thriller, where the Foreign Secretary will visit our hero in his flat at St. James's after midnight, to confide that it is no exaggeration to say that the fate of nations depends on this innocent document. In one long paper chase we pursue letters patent of a claim to the Ruritanian throne; a private annexe to a treaty; a nameless protocol; a formula predicting the

stock market, called only The Product; a millionaire recluse's lost final will. The matter is an urgent one. Paper, that may crackle with life, may also crackle in the flames.

The token changes hands, perhaps frequently, or is at least pregnant with the risk of loss, while the present holder is in no doubt that, merely by virtue of holding it, he is the central figure in the narrative of the world. A comedy inversion (and a defining quality of farces) is to find the object not vital to hang on to, but vital to get rid of. In the *Fawlty Towers* episode "The Kipper and the Corpse" (1979), both objects recur no matter what Basil Fawlty tries, the fugue-like reappearance of the kipper—which one might think easier to dispose of than the dead hotel guest—being a masterstroke. The kipper, long past its best, is cooked for the breakfast-in-bed of a guest who has, however, died in the night. Fawlty, the hotelier, wrongly assumes that the kipper was responsible for the guest's demise, and panics. The doctor is coming. Fawlty has only seconds to hide the evidence. He tries to open a window, but it's jammed. He stuffs the kipper under his cardigan, but the head begins to protrude just as . . . and so on. Later, the kipper is thrown into the kitchen doorway, out of shot, to get rid of it once and for all; where, however, somebody carrying the corpse trips over it. This Marx-Brothers-like business is funny partly because the kipper is incongruous, but mostly because Fawlty is berserkly determined that it must not be seen, a distraction from the actual problem that the corpse must not be seen. Also, of course, because it ought to be easy to get rid of but will not go away. As curious as it seems, Fawlty's kipper is in some structural sense the equivalent of Wagner's ring. One cannot quite imagine Fawlty bellowing a final operatic warning to "Beware the Kipper!" as the curtain falls, and yet it is still his nemesis.

So narrow a focus, and such certainty of purpose (as compared with the aimless ambiguities of real life) are characteristic of genre fiction, as also, by its present means of construction at least, characteristic of interactive fiction. There is no crime against mimesis in designing an interactive fiction whose quest is essentially for an object rather than, say, a shift in human relationships, for traditional fiction is replete with such tales. Mimesis is threatened only when the object sought fails to convince the reader that it is precious. It must move us as the plight of a sympathetic character would move us: it must be a character in all but physical form.

Interceding Objects

Having considered objects as evocation and as desire, scenery and treasure, we turn to objects as tools: not as physical tools, like crowbars or biplanes, but as the means by which fictional characters adjust and express their feelings about each other.

In the stagey third act of Alfred Hitchcock's movie *North by Northwest* (1959), Roger O. Thornhill (Cary Grant) is hiding on the upstairs landing of James Mason's criminal hideout. Frantic to attract Eva Marie Saint's attention, he dare not risk being heard or seen by any of her confederates. Eva is a double agent who has infiltrated the gang, but, unknown to her, she has been found out. So Grant must somehow warn her that the gang plans to kill her. His solution is to throw a matchbook, bearing the monogram R.O.T., onto the carpet below, at Eva's feet. It concentrates our anxiety because it is so tiny on that wide screen, because he has only one try, because Mason has guns, Martin Landau, and a sinister housekeeper while Grant is armed only with a small square of cardboard. Who will see it first? Even if Eva does, will her reaction betray her?

Grant's solution is one for which the audience has been prepared. We saw the matchbook earlier, in what was itself a deftly metonymic device. Roger has R.O.T. matchbooks printed: what shall we infer of Roger? That he is successful, not a young man, smokes too much, advertises, is wry about his own shortcomings ("rot") but stubborn too, doesn't spend too much time at home (being doubtless divorced by two or three exasperated wives already), and is suave as all heck regardless. Roger is, in short, played by Cary Grant. The matchbook serves still a third function in a movie "about" transposed identities: of the many markers of identity in the movie, from red railroad-company caps to dry-cleaning tickets and hotel reservations, this label alone, small and foolish, tells the truth. Barring some peripheral characters early on, only Grant is *not* a secret agent.

The message carries; Mason and entourage leave for a getaway plane; Grant rushes downstairs, meaning to follow them out and somehow rescue Eva. Unexpectedly a second familiar object detains him: Eva's revolver. In its earlier scene, Eva had proved her villainous credentials by shooting Grant dead, a phoney incident staged for Mason's benefit. Now Grant, thinking the downstairs lounge is clear, finds the housekeeper pointing that same revolver dead at him. Cut to outside where, a little

while on, we are startled to see that Grant has now escaped. How did he do it? In a classic aside, which except for one frantic exchange early on is his only dialogue in the whole 14-minute sequence, Grant tells Eva: “The housekeeper had me pinned down for five minutes before I realized it was that same silly gun of yours.” Like Grant, we the audience were clearly shown, and then reminded, and still forgot, that the revolver was a fake.

What was Ernest Lehman, the screenwriter, up to? His difficulty was to resolve the movie’s mysteries without allowing it to end, Hitchcock being set on an all-action finale at Mount Rushmore. Grant, Eva, and Mason must somehow all meet, clarify their true motives and identities once and for all, yet resolve nothing. Lehman accomplished this with devices not unlike those that sterner critics of interactive fiction would disapprove. Lehman keeps Grant one location behind Mason—climbing the house wall when Eva is upstairs; upstairs when the gang are downstairs; downstairs when they are out on the driveway; out on the driveway when they are at the plane waiting to take off. Lehman ensures that Grant cannot provoke a confrontation by having him outnumbered and unarmed. He slows Grant down with a two-stage puzzle based on objects, each used once already. Interaction is diverted from people to things, from speech to gesture.

Thomas Hardy likewise avoided a happy ending to *Tess* by having a letter, pushed shyly under a door, disappear beneath a carpet, never to be read; he disposed of Michael Henchard’s last act of love, a gift of a canary, by having its cage draped closed so that it starved before anyone found it; he demolished a solid farmer’s composure by sending him a flirtatious but unmeant Valentine from Bathsheba Everdene. All these objects are carriers, just as the matchbook was Cary Grant’s honest endeavor and the phoney revolver Eva’s inner self: a good girl, only pretending to be a murderess. As with most good romances, it is hard to pin down exactly where in *North by Northwest* the lovers settle on each other, but if one had to pick a single scene this might be it.

The Inventory of an Adventure

Sherlock Holmes comprehends people, when he does, through objects, the tools and clues that follow him as a comet is made visible by its tail. As the “Adventure of the Blue Carbuncle” (1892) opens, Watson, who

“knows my methods,” reconstructs the recent past with one glance at the objects surrounding Holmes’s sofa: newspapers “evidently newly studied,” lens and forceps so posed that they “suggested” a forensic examination. A mislaid hat, owner unknown, is then examined for some 1019 words, one-seventh of the whole, while Holmes performs his standard trick. Watson: “But his wife—you said that she had ceased to love him.” Holmes: “This hat has not been brushed for weeks.”

This “Adventure,” slight but charming, has a festive motif, over a dozen characters, picturesque locations; yet its objects have the greater tenacity, as we see through the eyes of its first reader. Sidney Paget, illustrator for *The Strand* magazine, and unsung creator of much of the Holmes iconography, depicted above all faces and postures: a criminal stoop, a saturnine smile. His interiors are vague pools of light with chairs only as needed, and he dispenses altogether with the Yuletide scene in the consulting rooms, the roaring fire, the windows “thick with the ice crystals.” The inventory for the story (from which the reader who knows Holmes’s methods may like to deduce the plot) is, in order of appearance:

a sofa; Holmes’s purple dressing-gown; pipe-rack; crumpled morning papers; a hard-felt hat with “H.B.” inside; lens; forceps; *a white goose with a barred tail; *a card reading “For Mrs. Henry Baker”; a brilliantly scintillating blue stone, rather smaller than a bean in size, found in the goose’s crop; *a small morocco casket; Holmes’s strong-box; a Scotch bonnet; another goose, identical to the first; two glasses of beer; a small thin volume and a great greasy-backed one; a hanging lamp; a sovereign (coin); a four-wheeler cab; basket chair; Holmes’s slippers; medicinal brandy; *a villain’s pipe; *twenty-five further geese; a bell for summoning Mrs. Hudson, Holmes’s cook.

Those items marked with an asterisk appear only in dialogue reporting past events at which Holmes was not present, and are as such unavailable to Paget (notably 26 of the 27 geese). This leaves 21 possibles, of which Paget’s six illustrations include exactly two-thirds, omitting only newspapers, strong-box, bonnet, coin, beer, cab, and bell. The coin and bell are too small, the strong-box, bonnet, beer, and cab would all require a seventh illustration showing a fleeting moment in the tale, so that only the newspapers are voluntarily omitted: and that because, presumably, the opening scene is already so cluttered.

We might, to be sure, inventory the same story in a number of other ways. As class stratification—our middle-class professional men jostle with a countess, a jailbird gone straight, an intellectual on his uppers, a barrow boy, four servants at varying degrees of respectability; as chronology—events are arrayed along five segments of time between 22 and 27 December; as economic survey—transactions range from a few pence, via one sovereign, to £1000; as a geography of barriers in Victorian London. In *Eccentric Spaces*, his classic study of landscape in the imagination, Robert Harbison remarks of the great detective that “In the distressingly simplest terms, this is the crux of social life: getting into other people’s homes, to which he has found an odd but gratifying solution.”

And yet these readings seem peripheral, like the shadowy walls that Paget declined to draw, compared to the objects. Objects serving a metonymic function are the best clues to the habits of the people inhabiting this story, especially in clothing, from the purple dressing gown to the hat. At the center in every way is an object of desire: the eponymous Blue Carbuncle, sportive (carbuncles are ordinarily red), exotic (from “the banks of the Amoy River in southern China”), precious (valued by Holmes at £20,000), sinister (“Of course it is a nucleus and focus of crime. Every good stone is”), exclusive (the Countess of Morcar reserves it to herself), sensational (advertisements of a reward for its return have filled newspapers for a week), hidden (it is always inside something else, never displayed). This being a comic tale, the stone needs no tracking down and is always being inadvertently given away. Holmes finds it in his supper goose, a parody of the customary coin placed in the Christmas pudding for the children to find. He will return it to the countess as soon as he has deduced how it reached him, by running back through previous owners. As for objects interceding between characters, the goose not once but twice appears as a peace offering between couples: Mr. and Mrs. Henry Baker, whose estrangement is such that she no longer dusts his hat; and at last Holmes and Watson, at a delicate moment since Watson’s recent marriage has obliged them to divide their households. No longer sharing lodgings, they can at least share beers and a late supper: and there is never any doubt that the bird, all the while being plucked and roasted below stairs, will in the final paragraph “fulfil the ultimate destiny of a goose.”

IF as Argument

Duncan Stevens

Introduction

It has been argued that, in theory, interactive fiction should be able to do anything that static fiction can do, as static fiction amounts to interactive fiction that consists of one move (>READ STORY) and more interaction should enrich the storytelling experience, not limit it. The merits of that proposition can be debated, but there's certainly substantial truth in it, and it follows that most of the techniques and subjects that can be usefully employed or explored in the realm of static fiction should be adaptable to interactive fiction.

One area of static fiction for which there are, thus far, few or no analogues in IF is argument or propaganda (the latter generally construed as a pushier and less nuanced form of the former but not a different animal as such). Examples of the form in static fiction include Orwell's *1984*, Solzhenitsyn's *Cancer Ward*, and Rand's *Atlas Shrugged*. Most static fiction that makes an argument doesn't function solely in that capacity, of course, but in each of the above examples a position of some complexity is staked out. Can IF do the same? Theoretically, there's no inherent reason why not, but the potential IF propagandist should be aware of potential limitations on the form, many of which apply to static fiction as well. Fiction as argument lends itself better to certain types of arguments than to others, and recognizing what kinds of arguments are best made through fiction, whether static or interactive, will likely make both for a more effective argument and for better IF. At the same time, however, the nature of IF allows for storytelling that in some ways is well suited to argument, as the player is capable of directing the story in multiple directions, and the author accordingly has the power to characterize those directions as more or less positive, depending on how they fit into the argument. The trick, of course, is to not only communicate what the author thinks but to communicate it in a way that tends to persuade.

In the first section, I discuss common types of argument and the structure of each such type. The second section discusses specific

examples of IF that has attempted certain forms of argument, and the third section examines some considerations and potential pitfalls in attempting to persuade through IF.

Categories

Argument in fiction may be generally stated as that which attempts to persuade the reader of the truth (or falsity) of some proposition external to the work itself; that Dickens convinces me in *A Tale of Two Cities* that Sydney Carton is at bottom a good man does not make that novel an argument for that principle. The French Revolution, however, does (did) exist outside the novel, and to the extent that Dickens attempts to convince the reader of something about the French Revolution, the book functions as an argument. In that case, the argument, depending on one's view of the novel, might be that the revolution began as a middle-class uprising and was co-opted by radicals, or it might be that political revolutions tend to victimize those who have supported them most loyally, or it might be something else entirely.

One common type of argument is policy-based: this or that program or policy should be adopted or eliminated because it will achieve the greatest good for the greatest number. In fiction, this might be done by simply dramatizing the effects of the policy or system in question, whether good or bad, and dystopias—*1984* among them—are one of the best-known forms (though some dystopias are better than others about explaining what it was that set off the downward spiral and how the causal chain worked). Dystopia has been attempted in IF, but not often; *A Mind Forever Voyaging* (Steve Meretzky, Infocom, 1985) is probably the best-known example. A similar type of argument in a different time frame is the historical argument: effect A was the result of cause B, not cause C, which led to effect D but not to E or F. Again, fiction tends to do this by dramatizing the causes and effects in question: *A Tale of Two Cities* is arguably a case in point, as are the Shaara novels depicting the American Civil War. IF, to my knowledge, has not attempted an argument of this nature, though some aspects of *Jigsaw* (Graham Nelson, 1995)—in which the player labors to prevent history from being altered—come close, in that they implicitly posit that history as we now know it depended on a certain event. (In that most of the events are obvious historical turning points, it's not a particularly daring

argument, and even when they're not, the game doesn't really spell out an argument about why those events changed the course of history.)

Philosophical arguments, by contrast, tend to depend less on factual results and more on the inherent merits of a principle or idea: generally, the point in a philosophical argument is that some principle is sufficiently important that it trumps other admittedly important principles. To some extent, Daniel Ravipinto's *Tapestry* (1996) was an argument of this type: the game gave you three options that could plausibly be connected to emphasis on different principles, let you choose one of them, and dramatized why the author thought that two of the choices were suboptimal. Stephen Granade's *Losing Your Grip* (1998) similarly made a pitch for a certain principle, albeit in a rather indirect way. Religious arguments can take either form: they can argue for a factual proposition that underlies a certain religious belief (the existence of God, say, or the truth of a certain historical narrative central to a given religion), or they can argue for a value proposition (that the existence of evil is consistent or inconsistent with the notion of a just God).

What is the importance of this distinction? Simply that fiction is generally better suited for arguments that advocate general value propositions than arguments from factual evidence, as fiction, almost by definition, is directed toward single instances rather than comprehensive treatments. If your policy argument is that a welfare system is good because it helps more people than it harms, it's simply too easy to paint a portrait of people being helped by such a system, and the reader is liable to think that the game hasn't proved a thing. If your historical argument is that the Treaty of Versailles led directly to World War II, it's too easy to paint fictional scenes illustrating the progression (angry destitute German workers voting for the Nazis, etc.). An argument that a welfare system is good because a society's profoundest moral obligation is to its poor, however, could well be done through fiction in general and through IF in particular; portraying the importance of such a system in a particular instance can illustrate society's obligation, and the author need not demonstrate that the scene portrayed is perfectly typical.

Examples

Certain types of arguments are often made primarily through one mode

or the other, but not invariably so, and *Tapestry* is a prime example. There, the game gives you the chance to relive certain key moments in your life and poses the question: is it better to (1) undo your choices, (2) make the same choices and accept culpability for the harm done, or (3) make the same choices and (essentially) reject the urge to feel guilty. As mentioned, there are philosophies, or at least ideas, that can be roughly equated with each of those approaches: the first is vaguely humanist, the second is stoic, and the third could be described as a notion of self-actualization (one that appears to value highly the making of choices and sticking by them). (*Tapestry* confuses things considerably by associating the first path with Lucifer and the second two with the Fates of Greek mythology—somewhat sensibly in the case of the second path, not so sensibly for the first and third.)

The game favors the third option rather emphatically, but it does so by piling bad consequences on the first two options, which amounts to stacking the deck. If the player takes the first path, he or she doesn't actually do much good for anyone by changing the choices at issue, it turns out; lots of bad stuff happens anyway—it just happens to be slightly different bad stuff. The author is free to make that argument in the context of the game, but it has nothing to do with a value proposition that goes beyond the game; to the extent that the game can be taken as an assertion that any such attempts to undo the consequences of one's actions are ultimately futile (even in a non-supernatural context), it's a highly questionable assertion. It would, in other words, be easy enough to set up a factual scenario where revising your choices led to unambiguously better results; with hindsight, in fact, that would appear to be the more natural conclusion.

The favoring of the third over the second path is slightly less fact-dependent, as the game makes the case that the way the protagonist handles his choices is more important than the choices themselves—but the deck is still stacked to some extent, as the choices amount to handling those choices either really poorly or reasonably well. That is, there's an argument to be made against simply rejecting guilt outright, which seems to be the course the author favors; the protagonist might instead decide to accept guilt and view it as a necessary consequence of making mistakes, a reminder that helps him avoid the mistakes in the future. Guilt might drive him to be a better person, make better decisions. Here, though, the alternative to rejecting guilt is being completely racked with guilt to the point where self-hatred consumes

your life, which doesn't make for a particularly nuanced argument.

All of this doesn't make *Tapestry* a bad game as such—it functions reasonably well as an exploration of moral decision making. Forcing the player to make the PC's choices is sometimes genuinely wrenching. But it does not work as an argument, even on the game's own contrived terms: it's hard to imagine anyone being convinced by *Tapestry* that, even if one had the chance to change things, it's better to leave key life decisions the way they were, let alone (if this is a fair real-life analogue) refusing to undo a decision that appears to have turned out badly. The major premise of that argument, that things will turn out badly anyway, is simply an assumption—unfalsifiable under the circumstances but pretty strongly counterintuitive. The game's other argument, about guilt, is no more persuasive: while it turns on a value proposition rather than on a factual assumption, it oversimplifies that proposition to the point of meaninglessness.

Equally ineffective, though for different reasons, is the argument in *A Mind Forever Voyaging*, where the protagonist is a computer asked to explore the future. A prominent politician has proposed a plan to cure the nation's social ills, and a researcher has managed to simulate the future as it would be under the plan and commissioned you to go explore the simulation. It turns out, of course, that everything goes to hell under the plan, and so I suppose you could say that the game serves as an argument against the substance of the plan (which amounts to a somewhat more draconian version of law-and-order proposals popular with conservative American politicians), though I hasten to add that this argument probably wasn't primary among the game's intentions. To the extent that it does so serve, though, it's a thoroughly ineffective argument, as the game makes no attempt to convince the player that the simulation is accurate; insofar as the player is later asked to act on what he or she has experienced, the game more or less asks the player to take it on faith that the simulation is being fair. (I half-expected a twist at some point whereby it would be revealed that plan opponents had stacked the deck.) Moreover, obviously, it's impossible to argue cause-and-effect on this scale; there's no good way of making the case that the law-and-order plan led to the social implosion that the game describes. The argument therefore hinges primarily on the game hurling certain facts at the player—the simulation is accurate, the policies in question are the primary cause of a certain result—which raises the question, for a skeptical player, about whether those facts tell the whole story. (In a

science-fiction story, naturally, they don't.)

Could the argument about law-and-order policies have been made more effectively as IF? Not in ways that resemble *A Mind Forever Voyaging*, probably, but there are other ways: a depiction of a newly built prison, for instance, where first-time offenders are indoctrinated into a criminal culture during their unjustly long sentences, or a portrayal of the deterioration of the first-time offender's family during that same prison term; an account of the trial of one such offender, as represented by an incompetent and overworked public defender; a story in which the offender is along for a ride with his bad-influence buddies and ends up taking the rap along with them. Most of these stories take a representative case rather than attempting to make a case about all of society, as in *A Mind Forever Voyaging*, and in these examples the facts are sufficiently commonplace—I doubt anyone would argue that those things don't happen with some regularity, that anyone would argue that the plot was concocted solely to serve the argument. The point is to make an argument about generally accepted or commonplace facts without having to prove exactly how common they are; if your argument turns on the (unprovable) notion that a certain percentage of imprisoned first-time offenders emerge from prison as hardened criminals, it's unlikely to work as IF. If, however, it arises more from the value judgment that rehabilitative efforts are a better investment of society's resources than more prisons (as illustrated by, say, a parallel story about an offender in a less punitive state), then it might have a shot.

Could this work for religious arguments as well? Religion is something of a special case, as the scope of the argument tends to be immense; if the game sets out to argue that a certain religion's view of the world is true (and, by implication, that the views of other religions, to the extent they are inconsistent, are false), and it attempts to make its case by portraying the world, it has an awful lot of ground to cover. Charges of deck-stacking in these cases are virtually inevitable, and *Jarod's Journey* (Tim Emmerich), a piece of Christian IF entered into the 2000 competition (which announced its intentions up front, stating that the game "will hopefully get you and Jarod closer to God") was criticized for precisely those reasons (and many others as well). To be fair, *Jarod's Journey's* take on Christianity was so simplistic that it barely qualified as an argument: the player was given a few options to take and was periodically told whether the author approved of the player's choices. To the extent that the result was supposed to persuade the player of the merits of

Christianity (by modeling a Christian's decision making), while it was spectacularly ineffective, it did attempt to make a case for a proposition; the problem was that the proposition (the truth of Christianity) was too vast to argue coherently.

How could it have been done? Perhaps, rather than presenting the entire worldview as a *fait accompli*, by taking certain aspects of the Christian understanding of the world (or those of another religion) and embedding them in a fictional setting, as *The Chronicles of Narnia* did. I.e., the major figures and events in the religion's central story might have analogues in the fictional world, and the player could be encouraged to view those characters and events in the same way as adherents of the real-life religion do. (I'm told that one of the few works of IF to deal with religious themes, *The Legend Lives!* (David Baggett, 1994), did something similar, though I have not played that game.) The result is likely to be a fairly watered-down argument, however; a reader can perfectly well enjoy or be moved by Narnia (or, I suspect, *The Legend Lives!*) without becoming any more convinced that certain religious beliefs are true, or even (in theory) realizing that the author had any particular beliefs. Transposing religious beliefs into fiction does avoid the problem of requiring the player to accept propositions external to the game, though, and as such it may be the safest approach.

A more fruitful approach to religious argument in IF might be to confine the argument to propositions internal to the faith, in which case, of course, the external propositions could be assumed to be true. Such a game might, for instance, question the extent to which a certain practice really comports with doctrine by forcing the player to justify his or her actions to a skeptical NPC. Alternatively, the game, by portraying a God who decrees certain practices or wills a certain state of affairs as inconsistent with how God is generally pictured within that faith (damning unbaptized infants, say, and coming across as less than loving), might make the point that the practice or state of affairs is in tension with the larger picture. In either case, the focus is on a value proposition and its consistency with a larger system of beliefs, not on proving facts.

Other Considerations

I have suggested that an author seeking to persuade through IF should attempt to make his case by pushing certain value propositions rather

than making a factual case. The problems with the latter are, I think, clear, but the question remains: would the former approach lead to effective arguments? Maybe, maybe not; a lot of what I have outlined above seems at first glance like the stuff of sentimental TV movies or wartime propaganda films, and no little skill on an author's part is needed to avoid that feel. The difficulty, to a great extent, in making an argument that turns on a value proposition is that to make it stick with a general audience, the author tends to want to sell the value in question too strongly. Ergo, the kid who gets thrown in jail because of an incompetent public defender wouldn't just be a kid, he'd be an honor student who's captain of the baseball team and tutors younger kids—but you see what I mean. The other facts are essentially a distraction from the value proposition in question, which is that the poor quality of legal representation for the indigent makes attempts to reduce crime by stiffening sentences unfair and unjust. The purer form of the argument would be to make the same case for a troublemaker kid who has no obvious future and who isn't particularly sympathetic, to show that people deserve adequate representation even when their stories don't tug at the heartstrings—but the trick there is to actually get the argument to come across without obvious editorializing about the importance of assuring the availability of free legal services.

I'm confident that this can be done, but, in light of the paucity of IF argument of any sort, I'm hard-pressed to point to actual examples. One well-done game with an argument of sorts at its core is, as mentioned, *Losing Your Grip*, where the protagonist (at least, as I see it—the author has avoided endorsing any particular interpretation, but this one seems reasonable to me) is engaged in a lengthy process of cleaning his own mental and emotional house and makes some surprising discoveries (surprising even to himself). Namely, he discovers that problems that he had long blamed on others are to a large extent his own fault and that the image he had constructed of his father wasn't entirely fair or true to reality. (Part of what makes this interesting is that the housecleaning wasn't wholly intentional—it was the result of an experimental drug treatment that was supposed to help the protagonist stop smoking.) The father does not, however, come across as particularly sympathetic, and the reconciliation (to the extent it happens) is hardly driven by a recognition that he is a misunderstood saint; the protagonist's image of him may have been distorted, but he's still fairly unpleasant. Nevertheless, the game manages to make a case for the propositions that

trying to mend a broken relationship is worth the trouble and turmoil and that letting go of old wrongs is better than holding on to them. Not wildly controversial propositions (when divorced from the specifics of the game), but not entirely truisms either.

How does it do that? Partly by depicting the alternative as destructive to all parties, but also by explicitly depicting the father separately from the distorted image of him that the protagonist had been carrying around (an image that the game portrays as a character in its own right). The villain of the piece becomes the image rather than the actual person, and once the image is vanquished, it's suggested that the protagonist can see things as they are and rationally decide to patch things up. The decision to reconcile with the protagonist's father does not, however, turn on a sympathetic depiction of him; the only such depiction that the game offers, as far as I can tell, comes after the decision is made, which makes reaching out to him somewhat akin to a leap of faith. (Part of what makes *Losing Your Grip* clever is that the distance between player and protagonist—since the player can reasonably be seen as an aspect of the protagonist that is brought out by the drug treatment—effectively permits the player to make the argument to the protagonist, by making certain choices within the game about how to view the protagonist's father.) The game continues whether or not the player achieves those insights, however, and since it isn't obvious that the choice was even offered, it's easy to miss the argument. But the choices are there, and the different outcomes at the end—violence and pain on the one hand, an extra scene offering additional insight on the other—suggest that the author did not view all of the choices as equally desirable. And since the argument does not depend fundamentally on the facts underlying the protagonist's relationship with his father, it would be unfair to say that the game loads the dice in its portrayal of a particular situation.

Conclusion

Arguments can, of course, take a variety of forms, and the case made in *Losing Your Grip* is fairly abstruse—enough so the aspiring IF propagandist may not find it a useful model. In addition, again, the case made there is unlikely to be profoundly controversial. But that game does show how a skillful author can use the techniques of IF to persuade—for example, by letting the player see the world differently from the

PC and having the player correct the PC's delusions—and the limited nature of the argument underscores the limitations of any fictional form, including IF, for persuasion.

The Success of Genre in Interactive Fiction

Neil Yorke-Smith¹

Introduction

Why are more works of interactive fiction—markedly more—set in a fantasy or science fiction milieu than any other? Why does the central puzzle of *Spider and Web* (Plotkin, 1998) succeed when it would fall flat in *Winter Wonderland* (Knauth, 1999)? What differences are found between IF works of, for instance, historical romance and Lovecraftian horror? In short, what is the influence of genre in interactive fiction?

Montfort's definition of a puzzle as “a challenge [...] that requires a non-obvious set of commands in order to be met” (*Twisty Little Passages*) is designed to be independent of author and interactor: we “should be able to determine what is and is not a puzzle simply by studying the IF work in question.” This granted, it is the experience of meeting the challenge—though it may differ from one to the next—that is significant to the interactor. The experience with the Enigma machine in Nelson's *Jigsaw* (1995) and with the navigational computer in Lebling's *Starcross* (Infocom, 1982) make the two quite different, despite the many similarities shared by the two machines in an abstract typology of puzzles.

Montfort argues that viewing IF works as riddles can bring together the literary and puzzling aspects. If the formulation of the puzzles, and the interactor's experience of them, is central on the one hand, then the nature and style of the narrative is central on the other. By Montfort's definition, the simulated world—described in the literary aspect and the setting of the puzzling aspect—is essential to interactive fiction; we suggest that the milieu of that world has bearing on the construction of the riddle.

The Enigma machine in *Jigsaw* is a fine example. Unmotivated, the

¹ This essay, written in 2002, reflects the contemporary year of its composition; a few more recent references have been cited. Author's address: Neil Yorke-Smith, American University of Beirut, Lebanon, nysmith@aub.edu.lb

machine is a formidable, tedious challenge that caused some to wish for respite (Thornton)—or even vengeance. In the setting of World War II, however, and motivated by the larger plot, on the retracing of historical necessity, “I found myself thinking, ‘If Turing and Newman could do it, then surely I [...] can do it too!’” (Rees, Jigsaw). Whereas the puzzle is natural, if difficult, in the setting, a Nazi encryption machine would sit unhappily in a fantasy work. (Which is not to say that incongruous and unmotivated, even tedious, puzzle elements are unknown in IF fantasy works: take the anachronistic battery dispenser in *Adventure* (Crowther, c.1975; Crowther and Woods, 1976).)

This essay considers how genre and riddle come together in the IF medium.

Surveying Genre in Interactive Fiction

Casual use of *genre* in relation to IF is schizophrenic: the word is used to refer to the medium of IF as a whole, as well as to “a particular kind or style of art or literature” (Oxford English Dictionary). Even in Nelson’s “The Craft of Adventure,” an early and thoughtful exposition of craft, we find “the genre [i.e., interactive fiction] is still going strong” (par.1) and “the best of even the tiniest games [...] make up a variety of genres [i.e., kinds of literature]” (par.4). Rightly, we suggest, “IF is a medium, and not a genre”; just as “[n]ew genres come and go all the time, in other media,” so “[t]here are different genres of IF—detective IF, sci-fi IF, etc.” (Weinstein).

TableSaw, writing critically on the newsgroup `rec.games.int-fiction` that was the communication forum of the IF renaissance, observes:

Classification of works is very important [for] it gives potential viewers a chance to identify similar works by identifying key traits that are similar to them. In addition it can help authors by providing a framework within to place their story. It can provide an assumed context to provide implicit information to a reader. And it provides a basis for comparing works. [...] [C]lassifications must come from careful examination and dialogue about works already made, rather than either trying to create categories and then fit pieces into them.

In short, classification is to be descriptive not prescriptive. As such, any classification of IF works will be subjective to some degree—what genre

is *Jigsaw*: historical? time-travel fantasy? romance?—just as no history of the Roman Empire is entirely complete or objective. Literature, like history, moreover, has many characteristics by which to classify. The encyclopaedic “Baf’s Guide to the Interactive Fiction Archive” (Muckenhaupt) indexes and tags works a dozen ways, including by attribute (“third person voice,” for instance), while a call to classify a set of highly rated works by the editor of the fanzine SPAG (O’Brian, Classification) yielded ten disparate responses, including the abrupt “They’re all games!” (Schmidl).

Unanimity, all would agree, is unattainable. But just as Gibbon’s *Decline and Fall of the Roman Empire* has become the canonical history to measure against, so Muckenhaupt’s “Baf’s Guide” is the canonical reference to freely available IF.² Muckenhaupt classifies works into 21 genres, from “Adaption” to “Western,” via “Horror” and “Seasonal.” The graph in Figure 1 shows that fantasy, RPG, and science fiction dominate the field of IF works. Of the rivals to Muckenhaupt, we note the genre classification in the “Z-Files Catalogue” (Baum), the selection by popularizer Britton, and a list by Short based on attributes (Literacy). For commercial works, while not a classification, “Adventureland” (Persson and Meier) is impressively complete.

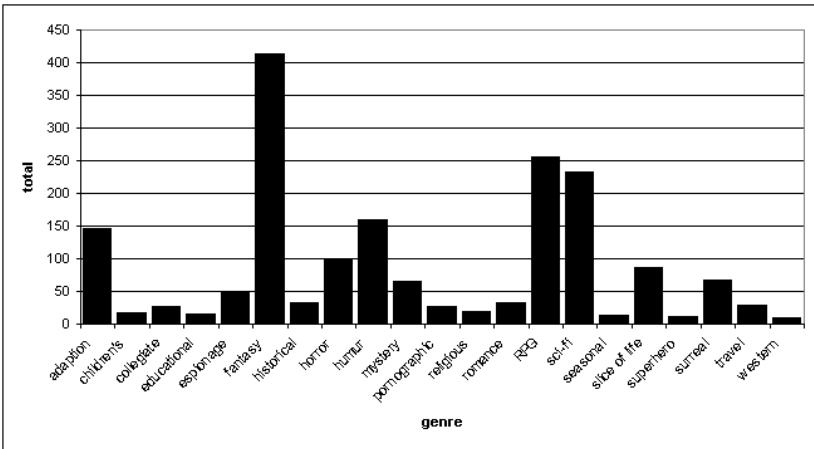


Figure 1: Classification of works by genre, circa 2002 (Muckenhaupt).

Several reasons have been advanced to explain the dominance of fantasy and science fiction among IF works. First, historically, *Adventure*

2 Later came the “Interactive Fiction Database” (Roberts), which had just under 90 crowd-sourced tags under genre, by 2010, not counting some near-duplicates.

(1976) and *Zork* (Anderson, Blank, Daniels, and Lebling, 1979) had fantasy settings, which many early works thought nothing of borrowing, along with much else.³ Of the 35 works published by Infocom (which we take as *Zork I* (Blank and Lebling, 1980) to *Arthur* (Bates, 1989)), 17 might be classified in the science fiction or fantasy genres.

Second, speculative fiction—fantasy and science fiction—as a rule appeals to those who take interest in IF, it has been argued; they enjoy “similar target audiences” (Plotkin, qtd. in deMause; see also Giner-Sorolla). So IF author Cull:

IF attracts a more technologically minded kind of author. You don't even think about writing stories that interact, that are very mechanical, unless you're in love with the machine. [...] And because it's a very young, offbeat, cultish medium, you're likely to get creatively minded people—people who look at the world and see not what is, but what might be. The sort of people who like speculative fiction.

A third reason for the prevalence of fantasy and science fiction is the unreality of the genres; the magic, if we will. Again, in part, this facilitates the suspension of disbelief that makes IF an escape from the real world, for both author and interactor: “Much of real life is not fun, and much of what makes a game fun is highly unrealistic” (Baggett, Setting). But Silcox, noting that hackneyed fantasy settings can be as hum-drum as a simulation of a modern apartment, argues for works that engender “de-familiarization,” so that real life might be seen afresh. He cites *Sunset over Savannah* (Cockrum, 1998), “which simulates with amazing psychological accuracy and a surprisingly high level of suspense the thoughts of a fairly average middle-aged man [...] trying to decide whether or not to quit his job.”

The magic goes further, though, fourth, for it allows the author to impose his own logic on the simulated world. The prologue in *Trinity* (Moriarty, Infocom, 1986) is set in the very real Kensington Gardens, London,⁴ while the middle-game spans various surreal worlds. In the latter we can hardly say, “but that's not how life works!” when the rules are made by the author and told to us. Perhaps too the magic lends itself to narratives more suited to the strengths of IF, a point we must return

³ Montfort's history of IF is as enlightening as it is entertaining (*Twisty Little Passages*).

⁴ The present author can vouch for the strict absence of magic, despite the statue of Peter Pan.

to later.

Fifth, speculative fiction can result in works “easier for amateurs [...] to write (not write *well*, just to write), and since most IF writers are amateurs not [as Moriarty] professionals, we pick the easy choice” (Cull). Newell, similarly, advocates that, since the fantasy genre requires less research than some others, it is more conducive to the amateur not willing to go to the lengths Moriarty did for the end-game of *Trinity* (Rigby), or that Nelson, though an amateur, did for *Jigsaw* (“On Jigsaw and ‘I’”, par.5; DM4 365).

The Interactive Distinctive: Puzzles and Challenges

Theorists debate what defines interactive fiction as a medium and distinguishes it from other forms of cybertext such as (electronic) CYOA (Aarseth; Montfort, Theory; Short, IF); with authors and fans alike they discuss what are the strengths and shortcomings of the medium. Recurrent in the debate is the role of puzzles. While the degree will vary, there must arguably be some interaction in an IF work to distinguish it from mere blocks of text separated by a “MORE” prompt, and here puzzles have a leading role: “Without puzzles, or problems, or mechanisms to allow the player to receive the text a little at a time [...] there is no interaction” (Nelson, DM4 382).

Here are authors Andrew Plotkin and Lucian Smith: “A puzzle is a mechanism for focusing the player’s attention”; “[O]ne of the main purposes of a puzzle is to involve the player in the story more” (both qtd. in deMause), and theorist Jerz: “A puzzle in IF is, in one sense, a management tool to separate ‘movements’ in the overall plot” (Puzzles). Thus, in addition to Montfort’s definition as challenge met by non-obvious set of commands, we have puzzles as source of interaction and as means of narrative advancement.

These additional perspectives suit our present purpose. We do not need to answer the difficult question of what, formally, is a puzzle, any more than we seek to give a taxonomy. Montfort states “[t]here is no requirement that a puzzle’s challenge relate to any other elements of an IF work in order for it be a puzzle,” but equally in our discussion we will want to consider interaction that by Montfort does not constitute a puzzle. Indeed, a work can be puzzleless—all the challenges are met by

obvious sets of commands—but still, we suggest, the work must have challenges to be considered IF. The line of demarcation and the works along its border, such as Finley’s *Life on Beal Street* (1999), will be debated just as the frontiers of the Roman Empire were contested.

A poor puzzle is regarded as one whose solution is non-obvious to the extent that a telepathic connection to the author is required to meet the challenge (deMause). In contrast:

A good puzzle, in my mind, is no puzzle. [...] [T]he puzzles should be transparent. Getting onto the benchtop in *A Bear’s Night Out* [(Dyte, 1997)] is a good example of this; getting back into your house in *A Good Breakfast* [(Adair, 1997)] is not. Puzzles shouldn’t be “puzzles,” but rather situations that must be resolved to further your goals. (Greenwood)

Notable is that the puzzles in Dyte’s work, which won a XZZZY Award for Best Setting, are based on the limitations of a teddy bear, for the large part, whereas Adair’s work has “artificial puzzles [...] thrown in for their own sake” (Stevens, Breakfast).

Lucian Smith defines a puzzle as “satisfying” if it gives the interactor pleasure when solved, and as “pertinent” if it relates to the plot as a whole (qtd. in deMause). Greenwood’s point is that better puzzles will be both satisfying and pertinent: the challenge makes sense in the narrative, the solution advances the plot (Nelson, DM4 394), and the whole is part of the atmosphere of the work (Jerz, Puzzles). Did Dyte succeed here and Adair not because of the settings they chose?

Giner-Sorolla’s influential essay “Crimes Against Mimesis” (reprinted in this book) argues for puzzles that maintain, enhance even, mimesis. Pertinent puzzles are more satisfying, he contends, and he is supported in this by seasoned IF authors reflecting on the craft: “[A puzzle] should be logical, according to the logic of the game’s universe” (Meretzky, qtd. in Hochberg); “[T]he puzzles should arise integrally from the milieu of the game” (Rees, Design). For example, the best puzzles in *Scapeghost* (Austin, Level 9, 1989) arise from the difficulty the PC, a murdered police officer returning as a ghost, has in interacting with the material world.

Given, then, that a work “should have a coherent fictional world and its puzzles should be seamlessly joined to the textual fabric, appearing to occur naturally” (Nelson, DM4 365), might we call such puzzles “organic”? If so, are genres more conducive to organic puzzles better

suited to interactive fiction? To this question we now turn.

Do Some Genres Engender Organic Puzzles?

Plotkin's celebrated *Spider and Web* is a tale of espionage. (A plot spoiler follows.) The turning point for the interactor is a moment of intuition: in reconstructing past events under interrogation, the PC has woven a fabrication. His unreliable narration deceives the interrogator. The puzzle lies in the interactor perceiving the truth (hinted subtly throughout) and, at the critical moment, using this knowledge to escape; it is both satisfying and pertinent (Chung). Plotkin won the XYZZY Award for Best Puzzle, among others. Besides his evident mastery of the craft, he demonstrated a puzzle organic to its setting; it hardly would have worked in a children's fairy story.

The experience of interacting with an IF work is that we “get to *meet* the people and *live* the events,” we feel “*responsible* for [the PC's] actions” (Baggett, *Simulations*). If, as we suggest, this interactivity is achieved through the puzzles in the broad sense, then, other things being equal, those genres amenable to “better” puzzles may be expected to be more successful. As narratives they are more immersive; as crosswords more satisfying. This is not to say any genre cannot be the setting for a successful IF work, merely that some may be more amenable than others. Interactive fiction centered on psychological drama, for example, is decidedly difficult to write well, but Bond's *Rameses* (2000) uses non-interactivity to its advantage.

Amenability to puzzles is aided, firstly, by genres whose organic puzzles render naturally in the IF world. Hence, one reason for the popularity of speculative fiction, in addition to those we saw earlier, is that these are “genres of exploration and action,” to which “the modeled world of IF lends itself very nicely” (Short, *Private*). Jerz's essay on exposition in IF argues for “live, don't tell”: “The IF player is supposed to live the story [...] Exposition that relies this heavily on narration—on ‘telling’—is awkward in IF” (Exposition).

Consider the romance genre, territory well-explored in static fiction, where the PC's feelings are central to the narrative. Much easier it is to walk through a landscape as it is explored than it is to communicate the emotion of the PC. Only one Infocom work, *Plundered Hearts* (Briggs, 1987), is a romance, and much of the romantic interest, though well-

written, is told in cutscenes—not shown, and still less lived. Later attempts in the genre have emphasized the narrative, with varying success: Huang’s *Muse* (1998), Fischer’s *Masquerade* (2000), Ingold’s *My Angel* (2000). Besides Ingold’s, the other works all feature developed PCs in a historical setting; *My Angel* is notable for its “novel mode.” The crossword struggles in the romance.⁵

In view of Jerz’s essay (Exposition), his own *Fine Tuned* (2001) becomes most interesting. Although subtitled *An Auto-mated Romance*, the work is more a comedy:

[Jerz] dares—and manages to pull off—a number of pieces of *participatory* comedy, which is much harder to pull off than just writing a bunch of funny lines that always show up. [...] I had to make the joke happen, or rather, the author had to set things up such that I *would*. (Cadre)

Here is “live, don’t tell” in practice. And as Baggett emphasizes (Simulations), it is powerful. Brian Moriarty on *Trinity*:

You could just feel the weight of history on you. [...] I just wanted people to feel that weight on them when playing the game. [...] It’s nice to know that interactive fiction could do that, make you feel uncomfortable about killing things. (qtd. in Rigby; see also Buckles 127–29)

Secondly, in some genres—mystery and adventure especially—the organic puzzles are readily “found.” Cadre defines a “found puzzle” as one that “derive[s] from the story” (qtd. in deMause), while Plotkin gives his first rule of puzzles as, “The world you’ve created creates the puzzles” (Happy). In static fiction, works in these genres “from Poe’s ‘The Gold Bug’ on, can capably integrate set-piece puzzles into the overall mimetic goals of the story” (Giner-Sorolla, par.3). It is not overly surprising, therefore, that mystery was the first genre beyond speculative fiction explored by Infocom, beginning with *Deadline* (Blank, 1982). Blank later wrote, “[M]ost people, when they read mysteries, are constantly trying to think ahead, what happened. [...] So, it seemed to lend itself perfectly.” (qtd. in Greenlee).

Writing in the *New York Times Book Review*, and clearly enthusiastic

5 Not entirely, of course: there is the extended duet in *Jigsaw* (where, interestingly, the PC and NPC Black are carefully gender-neutral); and Short’s *Pytho’s Mask* (2001)—her entry in her own SmoochieComp—was nominated for Best Individual Puzzle.

about *Deadline*, Rothstein links the success of Infocom's early work to their choice of genres, "the worlds of popular fiction—the detective story, science fiction, adventure and fantasy. These genres define worlds with their own logic; they pose lucid questions and possess clear narrative easily adaptable to a computer." Less is this so for the romance, we might add. Dyer reports "Infocom people [...] joke[d] about the idea of a romance series; somehow the moves don't seem appropriate to a computer keyboard"; Briggs's *Plundered Hearts*, the sole Infocom work authored by a woman, was the exception to prove them wrong.

Thirdly, amenability to organic puzzles can be aided when the interactor and characters in the work share parallels in their knowledge acquisition; Myers cites Callaci's *Dangerous Curves* (2000) as an example. The most extreme situation, the clichéd IF amnesia opening, is "nearly identical to the premise upon which so many detective stories and film noirs open" (Arnold, reviewing *Gumshoe* (Oliphant, 1996), in which we have both). If this, together with the found puzzles, makes the mystery suited to IF in principle, then the difficulty lies in the execution. *Deadline* has its share of bugs (Aarseth 115-27; Cree) and Infocom's subsequent *Suspect* (Lebling, 1984) was harrowing to test (Lebling). "[T]he mystery genre demands extremely rigorous testing" (Rees, Undertow).

All three points come together in the horror genre, such as in Finley's dark *Babel* (1997), a deeply suspense-filled work. Although, like romances, there is emotion to convey, in a work of horror it can be done through living and showing: foreboding in the world, terror from shadows glimpsed; organic puzzles ("how do I escape the cellar?") can arise from story and world; and the PC's realization of the nature of things can mirror the interactor's own. But horror, more so than speculative fiction, relies on proper pacing, which by nature is harder to ensure in the IF medium. Perhaps this offers an explanation for the reliance in IF horror on building suspense through the modeled world. Lovecraftian *Anchorhead* (Gentry, 1998) excels here; like *Babel*, its puzzles are organic and woven into the plot. Author Gentry analyzes the genre in detail in "The Parser at the Threshold: Lovecraftian Horror in Interactive Fiction" (Theory).

Bring on the Jester

Rothstein goes on to note that Infocom "tempers [their works] with

irreverent wit.” The tradition of humor in IF, like the commercial boom in the medium in the early ’80s, extended beyond Infocom (though the witty narrator responses in *Zork* and descendants were unconsciously later echoed in Nelson’s Inform library). Besides speculative fiction and alongside “Adaptation,” “Humor” is the next largest category on “BaF’s Guide.” The works can be divided into three groups: satire and parody, such as the many *Zork* pastiches; jokes, such as *Pick Up the Phone Booth and Aisle* (Bauge et al., 2001); and mainstream comedy. Like *Fine-Tuned* (Jerz, 2001), the latter will come under some other genre as well: Infocom’s two most famous, *The Hitchhiker’s Guide to the Galaxy* (Adams and Meretzky, 1984) and *Leather Goddesses of Phobos* (Meretzky, 1986), are science fiction and spoof 1930s space opera respectively.

As in worlds where he can dictate the logic, so the author has increased freedom if he is allowed to exaggerate and parody—even to break mimesis. He enjoys the “bulletproof bracelets of satire and ridicule” (O’Brian, Frenetic). This said, humor for its own sake is a fine thing. Moriarty’s *Wishbringer* (Infocom, 1985), for example, tempers horror with comedy to yield a well-judged work.

Humor, moreover, can soften the edge of puzzles that block the interactor’s progress through a work: “If a game is funny even while I’m banging my head against the wall, I’ll keep playing. If not, I’m probably gone” (Cadre, qtd. in deMause; see also Coleman). A case in point is *Fish!* (Molloy et al., Magnetic Scrolls, 1988), a parody of a British secret agent. *Fish!* is a delight to interact with, despite fiendish puzzles. Nelson observes that it is no coincidence that sarcastic narrator responses “are often jibes at the player’s progress” (DM4 373). Sufficient wit can even—almost—compensate for a poor implementation, as in the notorious early revisions of *Fine-Tuned*.

Adaptations and Literary Works

From the earliest days of interactive fiction, other works—static fiction, poetry, film, and (in due course) IF—have given inspiration. The *Dungeons and Dragons* role-playing game influenced Crowther (Peterson 187-88), while Woods recalls in an interview, “I had read Tolkien, but I didn’t consciously use it as a model for anything”.⁶ Tolkien’s powerful

6 He continued, “Even the description of the volcano, which some writers have claimed was modeled after Mount Doom, was written with no particular vision in

myth undoubtedly inspired much early IF, directly as well as indirectly. Nelson records that the first IF book adaptation was *Lord* (Paavola, c.1980) (DM4 347); Melbourne House produced faithful adaptations of both *The Hobbit* and *The Lord of the Rings*.

Science fiction too soon became a source, and in due course book adaptations were undertaken (Dewey; Nelson DM4 351; Randall), the most famous and assuredly most successful being *The Hitchhiker's Guide to the Galaxy*, written in conjunction with Douglas Adams. Interestingly, here is Adams's co-author Meretzky:

My criteria [for a successful adaptation] would be things where the book(s) or movie(s) creates a rich universe with lots of possibilities for stories that aren't necessarily the one told in the original book or movie. For example, I think that's why *Hitchhiker's* was such a successful game, and why it got better further in the game, when we diverged more from the scenes of the original story line.

While some professional writers like Adams have dabbled in IF (DM4 352-5), straight book adaptations are difficult. Besides the issues of copyright and linearity, a novel is simply too long (DM4 366-7; Randall 186). Instead, borrowing the world, as Meretzky suggests, has been more fruitful: Nelson, for example, acknowledges (367) his close and literary adaption of Shakespeare's *The Tempest* (1997) is inferior to *Avon* (Partington, Topologika, 1982), a successful "confection" that takes puzzles from many of Shakespeare's plays. Similarly, works that have set their story in the world of Sherlock Holmes or Alice in Wonderland have fared better than those that have sought to adapt the story outright.

Cult fiction accounts for a good chunk of the "Adaptation" category on "BaF's Guide." Tolkien adaptations and the ilk aside, Stevens regards literary adaptation as an underpopulated IF genre (Nevermore). If setting and possibilities rather than plot are sought, then poetry can be as suitable as novel or film: Cull freely took inspiration from Poe's "The Raven" for *Nevermore* (2000). Even nursery rhymes can be adapted, as Callaci demonstrated with *Mother Loose* (1998);⁷ and arcade games too: *IF Arcade* (Cadre et al., 2001). Do we foresee *Street Fighter: The Interactive Experience*?

If the main route by which literary works come to the IF medium is as inspiration or adaptation, then literature also comes to IF by a second,

mind."

7 And fairy tales, as the works of Short attest.

more original means. Published author Michael Berlyn brought a literary edge to the medium (with others, be it said) when he joined Infocom, not just in his works—*Infidel* (Berlyn and Fogleman, Infocom, 1983): “consciously literary ends,” “clearly a plotted novel” (Nelson DM4 355)—but in opening the way for moral and rhetorical questions and studied allusion and allegory. Hence the weighty themes and chosen quotations in *Trinity*, and the evocative symbolism of *So Far* (Plotkin, 1996); Randall considers literary IF works and concludes that “interactive fiction allows the reader to partake, first-hand, of a new literary world, and the unfolding of that world is continuous, even if the plot is not” (190).

Diminishing Genre

Simmering beneath our discussion, like rumors of Nero fiddling, is a question distilled by Nelson: “Today’s designers are not always so definite in keying a game to an established genre of fiction [as those prior to the IF renaissance]” (DM4 354); the trend is “of shorter stories moving away from genres” (342). True enough, early authors, particularly Scott Adams, deliberately explored well-defined genres, whereas later IF authors—with established confidence in the medium—inclined towards stronger narratives, able to stand without supporting themselves with an explicit genre (*Photopia*, Cadre, 1998), and towards exploring the boundaries and expectations of the medium (*Spider and Web*). Both are trends of maturity.⁸

Nevertheless, Nelson continues, “the first decisions remain to choose the style, the mood, [...] and above all the fictional world of which the story itself will remain only a part” (DM4 354). Therefore, just as we have considered a broader definition of puzzle than Montfort’s, we do well to include setting and milieu in our broader discussion of genre. Not all works of static fiction fall into a tidy “genre fiction” category; nor will all works of interactive fiction. Douglas and Hargadon suggest the attractions of genre and “non-genre” fiction divide between “immersion” (in a familiar world) and “engagement” (with literary works); whether consciously adopting a genre or not, interactive fiction can span both. Further, classification into genres has value even for

8 A further indicator of the maturing of the IF craft are works that uniquely leverage the medium, such as the elevated *Galatea* (Short, 2000), a work entirely concerned with conversational interaction with a single NPC.

literary works with subtle milieus, as we see in static fiction with cases like Neal Stephenson's *Cryptonomicon*.⁹

Take Plotkin's *Hunter, in Darkness* (1998), which won XZZZY Awards for Best Setting and Best Individual Puzzle. It appears to be a hackneyed cave crawl, complete with maze—a genre exhausted ten years prior. But “the cave is as much your adversary [as the prey]” (in the tradition of Crowther, using the environment as an organic puzzle); the maze requires no mapping (a good thing, since it is infinite); and “the plot branches and rejoins so seamlessly that you’re unlikely to notice that there are multiple ways through the game” (Stevens, Hunter). Plotkin subverts the genre assumptions and exploits the medium, and so “breathes new life into a very tired genre, no small feat.”

Michael Kinyon, a seasoned interactor with IF, picks up this theme, preferring works that make a “genuine attempt to push the limits of a genre for aesthetic effect” (qtd. in Forman). Likewise, Stevens identifies the genre “bait-and-switch,” encountered for instance in *Trinity* and *Once and Future* (G. Kevin Wilson, Cascade Mountain Publishing, 1998) (Break-In). The genre twist is a descendant perhaps of the reality-to-fantasy transition of *Adventure* (1976) and *Zork* and many works since, including *Curses!* (Nelson, 1994) and *The Mulldoon Legacy* (Ingold, 1999).

Some genres have been overdone, to be sure: the cave crawl, stock fantasy, collegiate and apartment settings (the latter two often in order to render the author's environment in the modeled world; the result often drives home Baggett's point that much of real life is not fun). But other genres, some strongly suited to IF according to our discussion so far, have been neglected: historical and pseudo-historical settings, those from specific cultural traditions, literary adaptations, the thriller and the western, cyberpunk and super-hero milieus.

The last, the super-hero, comic-book genre, is a dynamic example. The author can yield to the urge to make the PC an action hero; the super-hero powers and escapades give rise readily to organic puzzles; the comic-book circumstances provide authorial freedom; the potential for comedy (or satire) is inherent. Here, concurring, is O'Brian, who would go on to write *Earth and Sky* (2001): “if it's a great power fantasy to watch some comicbook character shoot fire out of his hands, how much greater to actually play the character that does it! ” (Frenetic). Genre, in

9 It follows that we contend against Montfort when he writes “[classification] does have just about nothing to do with the craft of IF” (Classification): the evidence presented in this essay suggests just the opposite.

the broad sense, has as big an impact on IF as it ever has done.

Conclusion

When crafting interactive fiction, the genre is crucial. It influences how amenable the narrative will be to the medium, how easily organic puzzles will arise, and how much freedom the author will enjoy. “If the chosen genre [and milieu] isn’t fresh and relatively new, then the game had better be very good. It’s a fateful decision: the only irreversible one” (Nelson, Craft par.4). We have examined some outstanding examples across different genres, seeing commonality in puzzles integrated into both setting and plot.

Some genres are more popular than others. We have looked at why fantasy and science fiction are so frequent choices and highlighted other genres crying out for more attention. As it has matured, IF has given rise to genres distinctively its own (TableSaw identifies “Zarfian” works); some, like word-play—*Nord and Bert Couldn’t Make Head or Tail of It* (Jeff O’Neill, Infocom, 1987), *Ad Verbum* (Montfort, 2000)—and conversational IF, would struggle in another medium. Indeed, the genre is still going strong.

Works Cited

- Aarseth, Espen. *Cybertext: Perspectives on Ergodic Literature*. Baltimore, Maryland: Johns Hopkins Univ. Press, 1997.
- Arnold, Julian. “Gumshoe (review)”. SPAG 9 (1996). 2 May 2002. <http://sparkynet.com/spag/backissues/SPAG9>.
- Baggett, David. “Re: Simulations (was Re: Choosing your IF setting / genre)”. 27 May 1994. rec.arts.int-fiction. Message-ID: 2s5is7INN5k9@life.ai.mit.edu.
- . “Re: Choosing your IF setting / genre”. 30 May 1994. rec.arts.int-fiction. Message-ID: 2sbdakINNbh1@life.ai.mit.edu.
- Baum, Michael. *The Z-Files Catalogue*. 22 Jul 1999. 2 May 2002. <http://www.la.utexas.edu/~pdl/zmach/ZCatalogNF.htm>.
- Britton, Suzanne. *The Best of Interactive Fiction*. 2002. 2 May 2002. <http://www.igs.net/~tril/if/best/index.html>.

- Buckles, Mary Ann. "Interactive Fiction: The Computer Storygame 'Adventure'". Ph.D. Thesis. U. Cal at San Diego, 1985.
- Cadre, Adam. "Adam's Comp01 reviews". 15 Nov 2001. rec.games.int-fiction. Message-ID: 9t26oa\$hqg\$1@drizzle.com.
- Chung, Adrian. "Spider and Web (Specifics)". SPAG 22 (2000). 2 May 2002. <http://sparkynet.com/spag/backissues/SPAG22>.
- Coleman, Douglas W. "Language Learning Through Computer Adventure Games" *Simulation & Gaming* 21 (1990): 433–441.
- Cree, Graham. The Upgraded, Consolidated, Infocom Bugs List. Undated. XYZZYnews. 2 May 2002. <http://www.xyzzynews.com/infocombugs.html>.
- Cull, Nate. Interview. SPAG 13 (1998). 2 May 2002. <http://sparkynet.com/spag/backissues/SPAG13>.
- deMause, Neil. "IF Roundtable: The Art of the Puzzle". XYZZYnews 14 (1998): 12–28. <http://www.xyzzynews.com/xyzzy.14g.html>.
- Douglas, Y. and A.B. Hargadon. "The Pleasure Principle: Immersion, Engagement, and Flow". *ACM Hypertext 2000 Conference Proceedings* (2000): 153–160. San Antonio, TX.
- Dyer, Richard. "Masters of the Game". *The Boston Globe Magazine* (6 May 1984). Available online: <http://www.csd.uwo.ca/Infocom/Articles/globe84.html>.
- Forman, C.E. "Interview with a Player Extraordinaire". XYZZYnews 13 (1997): 5–10. <http://www.xyzzynews.com/xyzzy.13d.html>.
- Gentry, Michael. "Parser at the Threshold: Lovecraftian Horror in Interactive Fiction". *IF Theory Reader*, 2011.
- Giner-Sorolla, Roger. "Crimes Against Mimesis". Originally posted to rec.arts.int-fiction. April 1996. 2 May 2002. Available online: http://www.reocities.com/aetus_kane/writing/cam.html.
- Greenlee, Steven. "Where Are They Now?". *Computer Game Review* (Apr 1996): 82–88. Available online: <http://www.csd.uwo.ca/Infocom/Articles/where.html>.
- Greenwood, Bruce. "Re: What is a good puzzle?". 22 Aug 1998. rec.arts.int-fiction. Message-ID:

35deb4be.2762660@news.ozemail.com.au.

Hochberg, Burt. "They Take The High-Tech Road to Adventure".

GAMES Magazine (Aug/Sep 1987). Available online:

<http://www.csd.uwo.ca/Infocom/Articles/games87.html>.

Jerz, Dennis G. Interactive Fiction — Puzzles. 2 May 2000. 2 May 2002.

<http://www.uwec.edu/jerzdg/orr/articles/IF/Puzzles.htm>.

—. Exposition in Interactive Fiction. 15 Jul 2001. 2 May 2002.

<http://www.uwec.edu/jerzdg/orr/articles/IF/online:/exposition.htm>.

Lebling, Dave. "They're Never Satisfied". The New York Times 3.4 (1984): 2. Available online:

<http://www.csd.uwo.ca/Infocom/Articles/NZT/Nztfal84.html>.

Meretzky, Steve. Interview. May 2001. Happy Adventuring. 2 May 2002.

http://www.avventuretestuali.com/Interviste/Original/steve_meretzky.htm.

Montfort, Nick. "Re: A classification exercise". 17 Mar 2000. rec.arts.int-fiction. Message-ID: 8atpqo\$pc9\$1@nnrp1.deja.com.

—. Twisty Little Passages: An Approach to Interactive Fiction.

Cambridge, Massachusetts: The MIT Press, 2005.

—. "Toward a Theory of Interactive Fiction". *IF Theory Reader*, 2011.

Muckenhaupt, Carl. Baf's Guide to the Interactive Fiction Archive. 2

May 2002. 2 May 2002. <http://www.wurb.com/if/>.

Nelson, Graham. "On Jigsaw and 'I'". XYZZYnews 6 (1995): 5–10.

<http://www.xyzzynews.com/xyzyy.6e.html>.

—. "The Craft of Adventure: Five Articles on the Design of Adventure Games", 2nd ed. 19 Jan 1995. 2 May 2002.

<http://www.ifarchive.org/if-archive/info/Craft.Of.Adventure.txt>.

—. The Inform Designer's Manual, 4th ed. (release 4/2). St. Charles, Illinois: The Interactive Fiction Library, 2001. Available online:

www.inform-fiction.org/manual/DM4.pdf.

Newell, Bob. "Re: Choosing your IF setting / genre". 20 May 1994.

rec.arts.int-fiction. Message-ID:

9405192133592.bnewell.DLITE@delphi.com.

- O'Brian, Paul. "The Frenetic Five vs. Sturm Und Drang (review)". SPAG 13 (1998). 2 May 2002.
<http://sparkynet.com/spag/backissues/SPAG13>.
- . "A classification exercise". 16 Mar 2000. rec.arts.int-fiction. Message-ID: Pine.GSO.3.96.1000316101139.12167A-100000@ucsu.Colorado.EDU.
- Persson, Hans and Stefan Meier. Adventureland. Dec 2001. 2 May 2002.
<http://www.if-legends.org/~adventure/>.
- Peterson, Dale. Genesis II: Creation and Recreation with Computers. Reston, Va: Prentice Hall, 1983.
- Plotkin, Andrew. Interview. Aug 2001. Happy Adventuring. 2 May 2002.
http://www.avventuretestuali.com/Interviste/Original/andrew_plotkin.htm.
- Randall, Neill. "Determining Literariness in Interactive Fiction". Computers and the Humanities 22.3 (1988): 183—191.
- Rees, Gareth. "Distinguishing Between Game Design and Analysis". XZZZYnews 6 (1995): 22–26.
<http://www.xzzzynews.com/xzzzy.6g.html>.
- . "Undertow (review)". SPAG 8 (1996). 2 May 2002.
<http://sparkynet.com/spag/backissues/SPAG8>.
- . "Jigsaw (review)". SPAG 9 (1996). 2 May 2002.
<http://sparkynet.com/spag/backissues/SPAG9>.
- Rigby, Paul. "From Here to Trinity...and Back Again". Adventure Probe, 5.5 (1991): 15–19. Available online:
<http://www.csd.uwo.ca/Infocom/Articles/moriarty.html>.
- Roberts, Michael. The Interactive Fiction Database. 6 Dec 2010.
<http://ifdb.tads.org>.
- Rothstein, Edward. "Reading and Writing: Participatory Novels". The New York Times BOOK REVIEW (8 May 1983). Available online: <http://www.csd.uwo.ca/Infocom/Articles/nyt83.html>.
- Schmidl, Gunther. "Re: A classification exercise". 16 Mar 2000. rec.arts.int-fiction. Message-ID: 8arj74\$qqj\$1@rohrpostix.uta4you.at.

- Short, Emily. IF Literacy: Recommended Playing List. 22 Dec 2001. 2 May 2002 <http://emshort.home.mindspring.com/literacy.htm>.
- . “Re: Abstract for IF theory book”. Email to the author. 30 Jan 2002.
- . “What’s IF? ”. 29 Nov 2001. 2 May 2002 <http://emshort.home.mindspring.com/whatsif.html>.
- Stevens, Duncan. “A Good Breakfast (review)”. 3 Jul 2000. Baf’s Guide. 2 May 2002. <http://www.wurb.com/if/game/329>.
- . “Break-In (review)”. SPAG 21 (2000). 2 May 2002. <http://sparkynet.com/spag/backissues/SPAG21>.
- . “Hunter, in Darkness (review)”. 17 Jul 2000. Baf’s Guide. 2 May 2002. <http://www.wurb.com/if/game/483>.
- . “Nevermore (review)”. SPAG 23 (2000). 2 May 2002. <http://sparkynet.com/spag/backissues/SPAG23>.
- “TableSaw”. “Re: classification again (was Meta-Puzzle)”. 18 Mar 2000. rec.arts.int-fiction. Message-ID: 8aulmj\$e84\$1@nnrp1.deja.com.
- Thornton, Adam J. “Jigsaw (review)”. SPAG 8 (1996). 2 May 2002. <http://sparkynet.com/spag/backissues/SPAG8>.
- Weinstein, Jacob Solomon. “Re: Is IF art? (was Re: Gareth’s competition comments)”. 18 Oct 1995. rec.arts.int-fiction. Message-ID: 463d09\$8ms@castor.usc.edu.
- Woods, Don. Interview. Jun 2001. Happy Adventuring. 2 May 2002. http://www.avventuretestuali.com/Interviste/Original/don_woods.htm.

Parser at the Threshold

Lovecraftian Horror in Interactive Fiction

Michael Gentry

“I must be very deliberate now, and choose my words.”
—H. P. Lovecraft, “The Rats in the Walls”

Horror is a wide-open field, not so much a genre in its own right as an atmosphere that can be applied to nearly any imaginable setting; there is as much horror to be found in the suburban streets of David Cornelson’s *Cattus Attrax* (1998) as in the antiseptic laboratories of Ian Finley’s *Babel* (1997), or in the antebellum backwoods of Adam Cadre’s *Shrapnel* (2000). It’s a challenge, sometimes, to even know where to begin.

The stories of H. P. Lovecraft occupy a peculiar place in this field. Although the classic tropes of the Lovecraftian tale may also be applied across many kinds of settings, it is nevertheless a highly specific and distinct subgenre. Its trappings and descriptive cues—the crumbling tomes, the ancient blasphemies, the awful, bubbling divinities outside the boundaries of our universe—these set the story apart from more conventional flavors of horror, giving it an immediate context and making it recognizable to any reader who is even casually acquainted with the source literature. This sense of familiarity makes it a comfortable entry point for writers and game designers hoping to craft their own stories in Lovecraft’s image.

However, like any variety of genre fiction, Lovecraftian horror is easy to do and somewhat more difficult to do well. One can too easily get so lost in the trappings as to forget what makes Lovecraft’s stories so distinct and compelling—to lose sight of the forest, as it were, amongst all the tentacles.

To pose just one example: Lovecraft’s stories are not particularly scary. It’s true. It is something of an open secret amongst fans of the literature; they are loath to admit it, but most will if pressed. Oh, it’s true, “Cool Air” is undeniably creepy; “The Rats in the Walls” delivers a bit of a shock at the end; “The Shadow Over Innsmouth” has wonderful suspense worthy of any action film . . . but the stories aren’t frightening, not in a visceral way, not in that way that makes you turn suddenly in your chair, shiver, and check down the darkened stairs before trepidly

returning to the computer keyboard to enter your next move. And no amount of blasphemous horrors from beyond the universe will ever make them so. The monsters of the Lovecraftian world are often, by the author's own admission, indescribable. Its horrors are unnamable. How can the reader be frightened by something that, he is flatly told, is beyond his every experience?

What sets Lovecraft apart from almost all other subgenres of horror is that his stories are not really about fear. They are about *revelation*. They are about piecing together an Awful Truth. Piecing that truth together, and possessing that truth once assembled, is not necessarily meant to be scary. It is merely meant to be, in a word, horrible.

From this perspective, Lovecraftian horror can be ideally suited to the genre of interactive fiction, because the player of the game and the protagonist in the game mirror each other in their goals. Both are presented with a hidden story that is gradually revealed, puzzle by puzzle, to the enterprising seeker. Both sift through fragments of text until the final narrative is laid bare. Hopefully, only one will have cause to regret it when the search is finally over.

The essence of this sort of tale is not whether the story takes place in the modern day, or in New England in the 1920s, or in the slums of Victorian London. It is not whether the terrible monsters are batrachian, or squamous, or merely rugose. The essence is in how the story is structured. Although there are many variations on the theme, the "classic" Lovecraftian situation can be broken down like so: A lone investigator arrives at an abandoned place, delves into written lore, and pieces together the Awful Truth. We'll examine each of these elements in turn and discuss how it can be applied to game design.

>

"There was no one in the soaking street, and in all the world there was no one I dared tell."

—H. P. Lovecraft, "The Shunned House"

Lone investigators are the easy part: nearly the entire corpus of IF revolves around the model of protagonists wandering off by themselves and fiddling with things. Here, at least, is a literary precedent.

Your protagonist should not be a nameless cipher. Horror is about terrible things happening to *people*, so give the player a person she can

care about. Don't be afraid to supply the character's motivations and even, to a certain extent, the character's thoughts and feelings in response to events in the game. Although it is the player who will be guiding the protagonist's footsteps and making all the important decisions, a clear motivation helps to put the game in a context that the player can immediately relate to.

Ideally the explorer should have some personal stake in the investigation. These sorts of journeys are not pleasant, and the merely idly curious will beat a hasty retreat as soon as things begin to get wriggly. The protagonist—and by extension the player—should want not to get out, but to go deeper. For example, in *Anchorhead* (Michael Gentry, 1998) (as well as in many of Lovecraft's stories), the motivator is family. The heroine has every reason to press on in the face of danger when the life and sanity of her husband is at stake. It needn't be that intimate, however. An archaeologist performing research vital to his career, a detective trying to find a missing girl, a safety inspector investigating a decrepit tenement—all of these people would have an abiding reason to see the job done.

And in many cases, all you have to do is get them to take the first few steps—and then let the door swing shut and lock behind them.

>

“... that cavernous, aeon-dead honeycomb of primal masonry; that monstrous lair of elder secrets which now echoed for the first time, after uncounted epochs, to the tread of human feet.”

—H. P. Lovecraft, “At the Mountains of Madness”

Since non-player characters are one of the most difficult elements in an IF game to craft well, it is convenient that Lovecraftian settings typically have so few of them.

Every abandoned place is an archaeological dig (including, obviously, an abandoned archaeological dig). Ghost towns, condemned asylums, ancestral estates—they have their histories. People once lived there, and once did things there, and evidence of this should be apparent in the room descriptions and the objects that you place. A decaying theater is more than just dusty furniture and broken glass: it is a yellowed program stuffed between two seats, a dried-out makeup kit in the dressing room, notes scribbled in the margins of a crumbling script book. Let the place

itself tell something of the people who used to inhabit it.

The exploratory phase of your game is important. Avoid a sense of urgency. Lovecraft wrote leisurely stories; the evil is always something that occurred in the past, and now lies dormant and deeply buried. Perhaps the player will awaken it once more, inadvertently (but, of course, inevitably), but give her time to explore first. Limit your puzzles, if you choose to employ any, to “passive” types—variations on the locked door, inaccessible areas of the map. Allow the atmosphere to sink in. Let the player get a sense of the space she is now in, and let the implications gradually become apparent: that this empty place once housed people, and now, for *some reason*, the people are all gone.

>

“The glimpse, like all dread glimpses of truth, flashed out from an accidental piecing together of separated things—in this case an old newspaper item and the notes of a dead professor.”

—H. P. Lovecraft, “The Call of Cthulhu”

Someone always writes it down. The written record is the protagonist’s primary link to what is really going on, and it is the most important segment of the Lovecraftian quest. Diaries, letters, police reports, newspaper clippings, carved hieroglyphs (which must be translated, of course), patient files, genealogical records, even the spoken testimony of a broken old man who knows too much—all pieces of a puzzle told in words. Here is where the actual player and the fictional protagonist are in perfect synchronicity: the goal of both is to unlock as much text as possible.

Whatever you do, don’t dump the entire backstory on the player all at once. Make him work for it. Scramble it, scatter it, reveal it in fragments and in the wrong order. Locating and collating all the pieces is the MacGuffin that drives your puzzles through this part of the game; every torn page is a treasure, and the protagonist’s notebook is the trophy case. Some pages may be in code. Some pages may be incomprehensible without reference to other pages. Some pages may contain clues to the whereabouts of other pages. All of them will be hard-won.

This is the story of your game, the story of what happened before, so embellish it. Make it grandiose and complex. Use several different sources of information to introduce a variety of conflicting perspectives.

Don't simply confirm what the player must have already guessed. As the fragmentary accounts slowly begin to resolve into a complete history, details of the setting will attain new significance. The barber chair with leather straps up in the attic is unsettling in its own right when first discovered—when the protagonist reads the diary entries that explain *why* it is there, it should be all the more menacing.

>

“The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents.”

—H. P. Lovecraft, “The Call of Cthulhu”

There will come a time when the player has collected all the pages, read all the history, made all the necessary connections. There is a point when the final, Awful Truth of the situation will be made clear. This is the end game, the climax of your story. This is the moment when history catches up with the present, when everything the protagonist has learned comes to bear on where she is *right now*—when she realizes that the giant has been slumbering beneath her feet this whole time, and now it is about to awaken.

Here is where you want to instill urgency. Time-sensitive puzzles and life-or-death decisions have their place in this segment of the game. Even Lovecraft himself was not above adding the occasional action sequence; a flight across rooftops with torch-bearing cultists swarming the streets below would not be at all out of genre. However, avoid making these sorts of puzzles overly difficult or finicky about precise verbiage. Nothing dispels the suspense more effectively than having to replay the most gripping scene in the game ten times because you can't figure out how to get out of the handcuffs. (Wisdom I dearly wish I'd had when I wrote *Anchorhead*.) Throw obstacles in the player's path, certainly, but make the solutions obvious, so she can move past them quickly and get on with her breathless flight.

Resist the temptation to overdescribe. Although strict faithfulness to Lovecraft's style sometimes requires that the climax be *a labored restatement of the obvious, written entirely in horrified italics*, the truth is that overblown, detail-laden descriptions of slimy, bubbling nastiness tend to leave readers more bemused than fearful. The art of suggestion by concealment is nowhere more important. Remember that the Awful

Truth cannot truly be described, only implied.

The Awful Truth is not how gross and slimy the monster is, but what the very existence of that monster implies. There runs through much of Lovecraft's work a theme of cosmic nihilism—the sense that the universe is so vast and impersonal, and humankind's place in it so insignificant, that to confront living, breathing proof of it is to go mad. In *Anchorhead*, that the earth was nearly devoured by a giant squid-god is terrible; that, to the squid-god, our earth is merely a single crumb amongst many, a morsel hardly worthy of attention except for the tiny noise made by a handful of cultists—*that* is the Awful Truth.

Consider how, or if, you wish to reflect this in the ending of your game. The “winning” move in a Lovecraftian story may well be to go mad, or die, or go mad *and* die. Perhaps the protagonist escapes without stopping the evil, knowing that it can never be stopped, only hidden. Perhaps the protagonist's reward is to be forever burdened with the responsibility of hiding it from others. In any case, that last message,

*** You have won ***

should always be read with a hefty grain of salt, if it is to be read at all.

>

“Who knows the end? What has risen may sink, and what has sunk may rise.”

—H. P. Lovecraft, “The Call of Cthulhu”

I have attempted to break out what I consider the most basic elements of the “classic,” or archetypal, Lovecraftian story, to show how they might be implemented in interactive fiction. As I said at the beginning, there is always room for variations on the theme. There are many, many ways to do Lovecraft, and any or all of these elements can be twisted, altered, or rearranged to suit your own personal vision of cosmic nihilism. Good luck.

Distinguishing Between Game Design and Analysis: One View

Gareth Rees

Introduction

I wrote this essay in response to “Game Design at the Drawing Board” by Christopher Forman (*XYZZYnews* #4). When I read that essay, I felt that it didn’t really correspond well with the way I work on adventure games. For me, maps, puzzle graphs, walkthroughs, and scoring tables are all tools of game analysis, not game design. Design, in the creative sense, lies elsewhere.

I will attempt to outline a set of concepts that can be used to describe the design of a game and also to assist the generation of ideas. These concepts describe my own thought processes while I wrote my game *Christminster*. The design proceeded on four levels:

Level One: Plot

At the top is the game’s plot. The plot is the set of elements of the game that might be used to make a story: what the background is, what happened before the game started, who the characters are, the major events that form the course of the story, and how the story will end. The plot is a map that shows how the characters interact and change as they go from the beginning of the story to the end (or ends, if the plot is branching).

Level Two: Scenes

A plot is too constraining to implement directly as an adventure game and still end up with a satisfying result. In a conventional work of fiction, the freedom of the viewpoint character is never an issue: the author can, without much difficulty, move all the characters through

their various interactions and emotional states until they reach the end. In an interactive work, this is much more tricky to do. What is necessary is to divide the elements and events of the plot into their smallest constituent parts, and so arrive at a set of atoms that may be reconstructed *by the player* into a decent plot. In *Christminster*, I identified a set of key scenes, each of which was an event or experience that affected the player character, and moved the story forward toward the conclusion, and yet could plausibly be implemented as a section of an adventure game.

A scene is a single dramatic event that typically brings together several components: interaction between the player character and other characters in the game, a strong effect on the player character, and preferably a strong effect on the reader herself.

It's probably easiest to explain what I mean by giving examples from *Christminster*. I needed to introduce Jarboe and Bungay as characters, and I needed to make it clear that they were the villains of the game. I also wanted the reader, playing Christabel, a woman in a milieu dominated by men, to feel scared and intimidated by the two men. Out of these goals arose the scene in which Christabel is trapped in Malcolm's bedroom and forced to endure a succession of insults and threats. Another example is that I wanted to establish Wilderspin as a friend of Christabel's. I've also always liked the (admittedly rather cheap) dramatic effect of being plunged into darkness underground by the closing of a secret door. These two goals came together in the scene in the darkness of the secret passage in which Wilderspin relates a crucial piece of information as part of a story about Isis and Osiris.

These two scenes were carefully scripted: I began by writing them down on paper in the form of a game transcript; neither was changed much when I came to implement them. I went to some trouble in the secret passage scene to avoid unnecessary complications. Christabel drops all her possessions as she trips over the step on her way into the passage so that (hopefully) the reader won't be distracted by thinking, "Which of my possessions do I need to use to get out of here?"

A scene doesn't have to map directly to a sequence in the game. Another effect I wanted to achieve was for the reader to experience a sense of wonder at the myriad glimpses of the history of the college and to feel a sense of achievement at the success of her researches (*Curses* had these effects on me, and I wanted to return the favor if I could).

There's no one sequence in the game that represents this, but instead it's a cumulative effect.

Level Three: Puzzles

The third level of design is that of puzzles. A few puzzles in a game will be integral parts of the plot, thought up at the earliest stages. Most puzzles, however, aren't part of the plot but are instead added on later for a variety of reasons. The most important reason for the existence of puzzles in a game is to force the reader to experience the scenes. It would be a waste of all that careful planning if the reader could go from the start to the finish directly, without experiencing any emotional development and character interaction! One way to do this is to have puzzles that require for their solution that the player has experienced the relevant scene or scenes. Another way is to have puzzles that are an inducement to sit still while a scene is taking place. For example, in *Christminster*, the puzzle in which Christabel must escape from the secret passage is there to make the reader stay around and listen to Wilderspin (not vice versa, as the naive reader might expect!). The various puzzles that take place during the dinner scene are an inducement to stay there and listen to the conversation, without feeling that the game is too boring and linear (which it otherwise would be).

Since puzzles aren't the main point of the game, I think their exact nature doesn't really matter. However, to act as good inducements to take part in the scenes, the puzzles should arise integrally from the milieu of the game and be intriguing and challenging. In an ideal world every puzzle would have a very satisfying and elegant solution, but alas, this is very difficult to arrange.

A few puzzles are left over and are just there for the sake of having interesting puzzles to solve, or to demonstrate the cleverness of the programming, or to impede the progress of the reader so that she doesn't reach the end without savoring the middle.

Level Four: Code and Text

Having planned a scene and possibly written a transcript of how it should look, and having designed a puzzle or two to go along with it, there's a lot of programming to do. My intuition here is that the first

thing to do before writing anything to do with plot or puzzles is to set up the basic definitions of the objects involved. For each object whose existence is implied by these plans, I try to think about it as a player: what kind of interactions can I attempt with this object? It can be helpful during this process to have a list of verbs by your side and to consider each verb against each object. Only when I have the basic definition do I add the code to make it a part of the puzzle. I think it's easier to work this way round, starting with the object as part of the simulated world and progressing to its role in the story, than to code the puzzle first and add the boring behavior afterwards (I find there's always a temptation to skimp on the boring behavior if I do that).

Putting the Levels Together

Typically development takes place on all four levels at the same time. A vague idea of the overall structure of a game is necessary to get started, but very little (I started work on *Christminster*'s initial puzzle when I still thought that the game would involve the college having been taken over by elves and a mountain range in the gardens).

The author needs to be a bit farther ahead on each level than on the level below, but not necessarily very far. When I was writing the code in *Christminster* for First Court, I had a good idea of what scenes would take place in Second Court but only a vague idea about dinner and the endgame. Sometimes an aspect of the game will prove tricky to pin down; the only thing to do is leave it and come back later (for example, I completed the gardens long before I thought of a good way to turn getting into the gardens into a puzzle).

Obviously each level affects all the others; if a scene is too difficult to be coded up (for example, if I wanted a scene in which the player persuades the abbot to take a vow of poverty by force of theological argument) then there is nothing for it but to go back and rethink the plot. If you have a great idea for a scene but simply can't think of a puzzle to motivate it, or a great idea for a puzzle but can't think of a way to connect it to the plot, then you had better put your great idea aside rather than try to squeeze the rest of the game out of shape. After all, this feature can always appear in your next game.

Tools for Analysis

The standard tools of adventure development (maps, puzzle graphs, walkthroughs, and scoring) are useful tools to check that silly mistakes haven't been made. I didn't find them of any help in the creative process, though.

Maps are important for checking the realism of the landscape (making sure that rivers don't change direction or run uphill, that buildings have realistic shapes and sizes, that the topography is geologically plausible), for checking that the player character has enough freedom of action, and for checking that the map steers a balance between being too grid-like and being too maze-like.

A puzzle graph (that is, a directed acyclic graph showing which puzzles must be solved before which other puzzles) is a good way to understand the game's constraints on the order of the player's action, to check that the game is solvable, to make sure that the game steers the right balance between being too linear and being too wide, and to check that there are enough optional puzzles and alternate solutions.

Walkthroughs and transcripts are most useful in the debugging process. A walkthrough makes it easy to check that a game is solvable and that old puzzles are not broken by the coding of new ones (this is especially important if there are timing constraints or other complex interactions between puzzles). A transcript makes it possible to check exactly what effect changes have on the course of a game. When I was debugging *Christminster*, I had a walkthrough that exercised all the puzzles and many of the game's interesting responses, and I kept a transcript of the game produced by capturing the output of the walkthrough. After making a batch of changes to the code, I ran the walkthrough again to produce a new transcript, and used the diff program to examine the differences between the old and new transcripts. In this way, I caught many, many bugs that would otherwise have been introduced during play-testing.

Scoring is for the player's benefit, not the author's, and is best added as late as possible in the development process (otherwise you'll end up spending lots of time fiddling with points here and there to make it add up, and risk breaking the scoring system as you alter the code for objects and change the assumptions under which the scoring system worked). If you have a reasonably sophisticated hint system, it's probably useful to link the scoring with the hints, because otherwise you'll end up

duplicating code since whenever the player solves a puzzle you have to both update the score and update the list of available hints.

Conclusion

This is a useful approach to the design and analysis of an adventure game. I certainly don't claim that this is the full story, or that everyone works in the same way. Each author goes about the creative process differently, and the same author may work in radically different ways on two games, or on two parts of the same game. Not everyone will want to work in this way; all I can say is that the process helped me to organize my ideas when writing *Christminster*.

If you will permit a modicum of speculation, I think that some of the ideas in this article may be useful when writing games that don't have a pre-determined plot (in the linear or branching sense) but instead try to assemble one dynamically from "plot fragments" or using a "plot calculus." Such a game will be designed as a collection of scenes embodying particular interactions or experiences, which can be invoked according to the needs of the developing plot to produce a satisfying story. Each scene will come with a set of parameters describing the change of state that it causes (in terms of the characters' emotions, beliefs, and so on, as well as the state of the world), and given a suitable collection of such scenes, the plot generator can select the scene that has the most desirable effect on the parameters of a game.

Natural Language, Semantic Analysis, and Interactive Fiction

Graham Nelson
St Anne's College, Oxford

[10 April 2005; revised 10 April 2006*]

This is an account of theoretical issues that came out, almost unbidden, from a practical test of the following hypothesis: that *the natural language in which to write interactive fiction is natural language*. IF is a form of creative writing impossible before the development of computing, but whose 30-year history has seen a flourishing of experimentation if not mainstream acceptance (except in an early commercial phase): the author creates an imaginary textual world that can actively be explored by a “reader,” or “player,” directing the actions of a protagonist. Such works have hitherto been created as if computer programs, using specially adapted programming languages (see for instance Nelson (2001)), but the Inform 7 project aims to replace such syntax with natural language: specifically, a subset of English. This change proved far more radical than had initially been expected, and it became clear that semantic analysis and related branches of linguistics were of great relevance to practical issues of how design systems for IF should work.

The Inform 7 project began in 2002 as an experimental higher-level layer on top of the existing Inform system for designing IF, now in use since 1993. At time of writing, an application for Mac OS X and Windows is just about to be published as a public beta.

This paper is divided into two. Part 1, “Naturalness in Practice,” describes and explores the motivation for the three conceptually new aspects of Inform 7: the user interface (§1a), the shift to natural language (§1b), and the adoption of rule-oriented rather than object-oriented

* This paper was written as a contribution to the forthcoming *IF Theory* book, and though an interim report it is also a manifesto that remains a fair statement of the project's ideology. The footnotes were added later in 2005, and the material then reorganized and redrafted in April 2006.

semantics that guided the Inform project, discussing in turn conceptual semantics (§2a), predicate logic (§2b), and model theory (§2c). The general slogan here is that the writing of IF is a form of narration; that a system for writing IF can be judged by the range of meaning it narrates; and that semantic analysis, the branch of linguistics concerned both with narrow and broad questions of meaning, is therefore of central importance to theories of IF.

In Part 1, I argue that the three major shifts described are all moves toward a more natural kind of writing. “Writing” is an ambiguous term: it might equally well mean a set of markings on paper, the activity of putting words together, or the prose that results: and for the same reason we must be precise in what we mean by “programming IF,” and in what we are claiming about it. First I suggest that the activity of programming IF is a form of dialogue between programmer and computer to reach a state with which both are content, and that it is not unlike the activity of playing IF, also a continuing dialogue in which the computer rejects much of what the user tries. Secondly, the place where this activity goes on is not conceptually a single page of typing paper, as would be offered by a word-processor, but is more like a book of translations presented in parallel text: with facing pages, one written by the programmer and one by the computer. Thirdly, the program that results from all this activity (the “source text”) is a description of an imaginary situation that extends through time—a story, in fact. The central idea of Part 1 is that a “natural” system for IF is one in which all three of these comparisons are tautologies: that the activity is explicitly a dialogue, that the user interface looks and behaves like a book with facing pages, and that the source text reads like a narrative.

In Part 2, I argue that the formal study of what is conceptually natural—that is to say, of semantics in the broadest sense used in linguistics—is a useful perspective on questions of how IF design systems should work. Natural languages make story-tellers of us all, and are well-adapted to the description of situation and event. Semantic analysis may be able to tell us what concepts and structures within natural language give it such facility in story-telling: looking for the presence or absence of these features in programs for writing IF may provide an insight into why certain kinds of IF are written but not others. Comparison with the literature of semantics may also help to question unconscious assumptions built in to systems for IF: for instance, are containers as important as we seem to think? Do we really

perceive the world in terms of objects that inherit properties from classes, or is that a conceit of computer programming? What should be part of the core functionality of a system for IF, and what can be relegated to third-party extensions, or left for writers to sort out for themselves? How shall we judge such questions of what matters most?

I wish to acknowledge, and those four words are woefully inadequate, the help I have received with the Inform 7 project from people who have at various stages contributed to its ideas fully as much as their practical expression: and especially Emily Short, Andrew Plotkin, Sonja Kesserich, Andrew Hunter, and David Kinder. Tendentious opinions here are my own, but I could not have formed any opinion without the last three years of discussion and collaborative effort, and I particularly wish to thank all those who have read and commented on drafts of this paper.

Part 1. Naturality in Practice

§1a. A Humanizing Interface

Early builds of Inform 7 coincided with the 20th anniversary of the Apple Macintosh user interface (1984). I had begun the project by collecting together notes into a self-styled *Book of Inform*, my version of the *Book of Macintosh* collated around 1982 by Jef Raskin (1943–2005): a mixture of the practical and impractical, and a description rather than a blueprint, and which was free to look nothing like the final product. The Macintosh team drew inspiration from the iconography and shape of road-signs, the function of the bicycle, the office environment, and the industrial design of cars: the aim was to make a computer a domestic appliance as natural as, say, a kettle (see for instance the recent memoir Hertzfeld (2004)). Similarly, the *Book of Inform* aimed to describe a radically humanizing interface for the writing of interactive fiction (IF). My earlier program, Inform 6, had been a computer programmer's tool that aimed to be welcoming to creative writers: this aspired to be the other way around, and its guiding metaphor would be that of the interactive book. In 2003 I had the great good fortune to recruit Andrew Hunter, author of the best-interfaced IF interpreter for Mac OS X ("Zoom"), to the project: the reference implementation of the interface is entirely his work. David Kinder then took on the coding of the corresponding Windows interface, which was no small feat since

essentially none of Andrew's code could be used there, and the entire system had to be written afresh.

To deal first with what was being abolished, the *Book of Inform* tried to remove the computer's filing system from the picture. Setting up a new Inform 6 project, and installing Inform 6, is a nuisance: it means creating a directory, working out commands to compile source into a story file, then to play it, run scripts through it, and so forth. This is discouraging the first time, tedious subsequently. The shortest legal Inform 6 source—the equivalent of that prototypical program, “hello world”—involves three references to filenames and is complicated enough that the books on Inform 6 suggest that newcomers copy it out blindly. By contrast, Inform 7 projects are automatically managed and look like single objects on the host computer. The shortest legal source reads: “Home is a room.” Reference to other people's code—any modern system for IF must recognize the highly collaborative nature of IF design today—is made by the name of what is being included, and whom it is by. Thus the source might read, early on:

Include the Automatic Door Rules by Emily Short.

rather as a book might be prefaced by a list of acknowledgments (and indeed Inform uses it to place just such a list in the compiled game). No filenames appear, nor any platform-specific references.

A project is a single book, not a docket of intermediate states in disparate formats and with cryptic names. But if it were one long endless stream of prose, it would quickly become disorganized (as early testing made abundantly clear). Most computer programs of any size are internally organized by being divided up into separate source files by function, but this seemed wrong for Inform because it took us back to filing systems. A partial solution came from “literate programming,” Donald Knuth's scheme for interleaving code and commentary (and indeed parts of the Inform program itself use Knuth's CWEB system: Knuth and Levy (1994)). Though Knuth's writings on programming stylistics, conveniently gathered in Knuth (1992), contain little systematic thought and are essentially rooted in the debates of what is now a bygone age (structured programming: grail or poisoned chalice?), they are nevertheless well worth reading.¹ His essential remedy was to

¹ The day after writing this somewhat slighting remark about Professor Knuth, I was introduced to him, and he really couldn't have been a nicer guy. It has to be said that CWEB today is a mess, just the same.

reconcile program with book by promoting a form of program easy to typeset, so that it would always have a dual existence: a human-readable one, and a computer-executable one, both continuously kept up to date. Inform goes along with this in dividing code into paragraphs, and also (as we shall see) in indexing, but ultimately adopts the same solution that books have used since the *Iliad*: it divides the source text up into sections, chapters, parts, books, and volumes, allowing for a hierarchy of headings and subheadings as elaborate or simple as the author prefers. There is no compulsion to use headings, but a number of incentives are offered to persuade authors into the habit: automatic contents listings, better-signposted Problem messages, and so on.

Some approaches to “interactive fiction for the non-programmer” have imitated database packages in displaying and editing projects as wallcharts, in which the various functionalities are boxes connected by lines rather as photographs of suspects are joined by threads on police noticeboards. This is seductive for object-based IF, because those boxes and lines can be related to the coarse structure of the work (map connections, most obviously). But it fragments the writing into tiny pieces. Some creative writers thrive on this—one thinks of Elizabeth Bishop hanging half-stanzas out on the washing line slung across her study—but few of us would choose to draft a novel on ten thousand Post-It notes. Fewer still would wish to edit or revise a novel written in this way, and such approaches to IF make second thoughts and bug reports tiresome to act upon.

Inform instead presents the user with an interface intended to look like an open book with facing pages. The author’s work appears in full on the left-hand page, while its consequences appear on the right. This feels natural to someone reading left to right, and agrees with the conventional layout of cartoon panels. The page spread suits today’s increasing use of LCD monitors in the aspect ratio 16:10, but several of Inform’s testers used 4:3 monitors equally well: the gutter between the two pages slides freely left or right, closing the one up and expanding the other, so that the user can decide which should occupy the greater space. Both pages contain text that is word-wrapped in real time as the pages are resized.

Both pages contain text, rendered in a variable-pitch font with strong anti-aliasing: a font chosen for the legibility of running prose, rather than a typical programmer’s text editor font, which uses a fixed pitch to preserve vertical alignments and over-stresses punctuation marks.

Although syntax-coloring is offered, the result is less kaleidoscopic than in most programming environments since Inform source text has few lexical categories: there is only “quoted text,” unquoted text, and comment [in square brackets].

The facing pages are the forum for interplay between the writer and the computer. Inevitably this dialogue is led by the human, typing the source text on the left, and the computer’s part is reactive, producing replies. In most languages programming has a code-compile-test cycle, where the compiler often rejects the code and forces the author to make corrections. This is not unlike the experience of playing through IF: think of something, try it out, make progress. Most IF critics agree that an enjoyable game requires a lively, keep-things-moving response to incorrect guesses, because guesses are more often wrong than right. But compiler programmers persist in regarding incorrect input as an aberrant circumstance in which it is inappropriate to make any judgment of the quality of the output. Compilers such as gcc also more often reject input than accept it, but do so with error messages that are nasty, brutish, and short. Error messages aim at precision in characterizing the exact symptom of failure, but do so in terms of the compiler’s own internal data structures or methods. For instance, gcc’s error:

```
main.c:81: request for member ‘count’ in something not a structure
or union
```

is factually correct but implies, untruthfully, that the problem lies with the structure; in fact the structure is fine, but the wrong request to access it has been used (‘.’ not ‘->’). Similarly,

```
main.c:175: parse error at end of input
```

commonly occurs when a close brace has been missed out, and while it is true that this was detected at the end of input, the problem is almost certainly somewhere else entirely. Such errors describe an accident in terms of the evidence, not the proximate cause. It is arguable that only final object code matters, which may be used by millions of people, and not the passing inconvenience of the programmer, so that maintenance on gcc should concentrate on code optimization rather than tidying up error messages: but consider what improvements to that final code might be made by a programmer whose time is not being wasted.²

² Matters grow worse with C++, where a minor industry now exists in selling software whose sole task is to interpret gcc’s error messages. At Apple’s 2005

Indifference to the convenience of error messages is well exemplified by the current GNU standards document (Stallman, 2004), which covers the punctuation of error messages in some detail while, significantly, offering no guidance on their stylistics, other than to offer the startling recommendation that:

In error checks that detect “impossible” conditions, just abort.

There is usually no point in printing any message. . . Whoever wants to fix the bugs will have to read the source code and run a debugger.

Inform does not follow this advice: it accepts the likelihood that its own bugs will be encountered by real users and tries to deal with such contingencies as helpfully as possible, at least identifying where the problem has arisen, so that the user can try work-arounds. And in general Inform’s errors do not follow the traditional Unix-command-line pattern.³ They are called Problems, not errors, they are not confined to one line, they are not reported by the line number on which they occur—instead, Inform talks about sections or chapters and makes generous use of quotation—and they include explanatory text that typically gives examples of correct and incorrect usage. The same basic error can result in different Problem messages according to Inform’s guess at the most likely way it arose. Many of these Problem messages have been added in beta-testing, to give more rewarding responses to reasonable but incorrect things tried by the testers: a process that we found strikingly like that of finishing a work of IF, at the stage when the designer adds numerous responses to cover all the unexpected cases that turned up in testing.

The Inform “coding cycle” consists of typing or amending the source in the left-hand page, then clicking the Go button. Either the source is rejected, in which case the right-hand page responds with a report of the problems found, or it is accepted, in which case the right-hand page begins playing the resulting work of IF. If the Replay button

Developer Conference, an entire session was given over to a lecture on what the dozen newly-introduced gcc 4.0 error messages actually mean. The session was recently posted online, and it is striking how often the speaker says “Well, this almost always means that . . .”

3 Though the pioneers of Unix would not necessarily have agreed with the culture of ultra-concise errors sometimes attributed to them. Kernighan and Plauger (1976): “. . . a prompt is given reminding the user how to use the program properly. Better to tell people concisely how to do things right than tell them only that they did something wrong.”

had been clicked instead, the work would automatically play through to its last position, using the same commands. If the author finds a trivial mistake—a spelling error, say—then this can be fixed in the source and the correction verified in a matter of moments.

Leaving aside unpredictable variations based on random-number generation, interactive fiction bears comparison with turn-based games such as chess. This observation motivated an approach to testing IF modeled on the analysis of chess openings, which run together for a while but then diverge. Every Queen's Gambit begins with the same first three moves (1. d4, d5; 2. c4), but then there is a choice, as the next move decides whether we have a Queen's Gambit Accepted (dxc4) or Declined (e6). It would be impractical to study every possible sequence of play, so chess manuals instead contain tables of standard openings. Such "opening books" are essentially built by watching every grandmaster-level game reported in the chess literature and seeing where they innovate. Inform uses the same method: it watches every command sequence typed in during testing to see if this duplicates a previous test, or breaks away. The resulting structure is called the skein, because it is a braiding (or rather a gradual unbraiding) of the possible sequences of commands, which we think of as threads. The skein display allows the author not only to look through all of the games ever played through the fiction in question, but also to replay any of those games, and see what has altered. (The Replay button does this for the most recent thread in the skein.) This aims to make debugging complex works of IF a more reliable business. In large works of IF, small changes in one place often have unforeseen consequences elsewhere, and a major cause of error is the accidental inclusion of one bug while fixing another. The skein quickly grows large, but can be pruned back if the writer chooses, or can be annotated with notes such as "Test falling off cliff" which can be searched for—just as chess opening books have annotations such as "Queen's Gambit."

The final component of Inform is the index that it automatically generates for every project, after each build. At its simplest this is a navigation tool for jumping to the relevant points in the source text—an important consideration in a text the size of a novel. But it is also an aide-mémoire with cross-references to the documentation. It offers a choice of viewing the index in a variety of conceptually different ways: by headings and subheadings, like a contents page; by rooms and their contents, in a map intended to follow the style of 1980s printed

solution-books for IF; by scenes and their possible sequences, in a corresponding map of chronology; as a taxonomy of the kinds of thing found in the model world for the project (an idea drawn from the SmallTalk class browser of the late 1970s); and as a collation into logical order of all of the various rules. All these viewpoints are valid, and Inform makes no judgment about which is pre-eminent. For example, the index of kinds shows, among other things, what objects exist for each kind. This is not the best way to see what the game contains—the map is much clearer—but is just right for an author contemplating (say) a new rule about containers, but worried that it might not always be appropriate: since the Kinds index lists all the containers existing in the world, one can see at a glance everything to which the new rule would apply.

It may be worth mentioning what was overlooked by the *Book of Inform*, or more specifically two issues that did not really occur to me as non-routine until much later on. First, documentation. The so-called DM4 (Nelson 2001), fourth edition of the Inform 6 manual, aspires to be something of a cult book to its modest readership, a textual “maze of twisty little passages” festooned with arcana and folklore. In so far as it had models, they were Larry Wall’s “Camel book” of Perl and Donald Knuth’s TeXBook: there is something a little appealing about playing the eccentric inventor, but I perhaps overlooked that neither book—though brilliant, indispensable, and such—is actually much good at its stated purpose. The Inform 7 documentation takes an opposite tack: it tries to be concise, to divide into short screen-readable subsections—it is built into the user interface itself—and not to try the reader’s patience overmuch with facetiae, nor to try to combine the manual with a history and critical study of IF. The much-criticized “exercises” of the DM4—actually showcases for surprising possibilities, not pedagogical tests—were replaced by some 260 “examples” that build up into what is described as a “recipe book”: these are intended to be imitated and borrowed from. Each contains a complete source text, not an excerpt, and comes with a rapid, automated means of seeing it work. Is the current version of the documentation useful? Is the current version enough fun, come to that?⁴ Time will tell: for now, no physical volume is

4 A serious issue: consider for instance *Why’s Poignant Guide to Ruby*, reprinted in, e.g., Spolsky (2005): strip cartoons of foxes in chapter 3 of the *Guide* act as a sort of subversive chorus to the ostensibly just-the-facts text, and they keep one reading even when one intended to learn Python instead. At one time, I did want to include classic cartoons in Inform’s documentation, too: Gary Larson’s *Far Side* on “The Curse of Mad Scientist’s Block” and Bill

being published in book form.

Another initially unplanned area of Inform 7 was the packaging and publishing of new works of IF: does the design system's responsibility end when the program is compiled? Inform now goes further: it can generate supporting material, and it helps with the bundling up of this material for the eventual player, and in particular with bibliographic data (where it is slightly coercive in an effort to make authors produce works that are easier to archive and browse in databases). As with language design (see §1b below), the analogy with actual books was actively pursued.

With some user interfaces, less is more. Adobe's *InDesign* is a fine program, but its plethora of buttons, cursors, palettes, inspectors, and toolbars means that, for beginners at least, it might as well be called *InDecision*. Arguably graphic design requires, or at least suits, such an interface. But does IF design? Throughout the experimental period, I would always want fewer buttons and the focus kept to one window, whereas Andrew was more open-minded. Often I would express initial skepticism, but then give in a week later. The feature of which we are least sure is the Inspector, Inform's only subsidiary window: it gathers a collection of functions that benefit from operating without the need to disturb the contents of the main window. The Inspector began simply as an overflow for things which would not fit elsewhere, and for some months I kept it permanently closed (Inform remembers one's preferences in this sort of thing), but I now find it too useful to banish.⁵ Every user interface project needs one curmudgeon who, by doing none of the actual work, forms opinions about features without reference to how much, or how little, effort went into them. It was a great luxury that better programmers than myself allowed me to play this role.

§1b. The Adoption of Natural Language

Natural language as a literal paraphrase of procedural code can suffer from the faults of both, and many programming languages that superficially ape natural language (such as COBOL, or AppleScript) are

Watterson's strip of six-year-old Calvin creating the Universe as one of the "Old Gods" who demands "Sacrifice"—but licensing fees of \$400 and endlessly unclear copyrights deterred me.

5 Famous last words. Over the summer of 2005 we moved the Inspector's main selling-point, the search tool, onto the main window, and it is now out of favor again.

not convincingly “different” in feel from orthodox coding languages such as C. This may be because they do not contain genuinely linguistic features, but even so, I feel that natural language is easily overlooked as a syntax option in the design of new systems. This is the more odd since many such designs often begin with sketches in “pseudocode”: English sentences that approximate what they will one day be coding in more formal ways. The curse of pseudocode is that it is self-consciously *pseudo*-code: we forget that it could also serve as actual prototype syntax.

As an example, consider the Sloan Digital Sky Survey (SDSS), an immense reference database of astronomical observations, and one of the world’s largest non-commercial exercises in data warehousing: an imaginative project with an excellent website. In Szalay et al. (2000), a summary of what were then proposed aims, we find an outline of the benefits from new search possibilities:

Other types of queries will be non-local, like “find all the quasars brighter than $r=22$, which have a faint blue galaxy within 5 arcsec on the sky”. Yet another type of a query is a search for gravitational lenses: “find objects within 10 arcsec of each other which have identical colors, but may have a different brightness”.

Here natural language acts as a pseudocode for database-searching programs. In 2006, with SDSS up and running, the public outreach website does allow queries like those envisaged by Szalay et al., but knowledge of SQL programming is required. The following pattern is offered as an example in SDSS Data Release 4 (2006):

```
select
p.objID,p.ra,p.dec,s.z as redshift,w.plate,s.fiberID
from
SpecObj s, PhotoObj p, plateX w
where
p.ObjID=s.bestObjID and w.plateID=s.plateID and
s.z > 4 and s.zConf > 0.95 and s.specClass = 3
```

In pseudocode, this would be “get the Object IDs, positions, redshifts, and plate and fiber numbers of quasars with redshift greater than 4.” Which is easier to write without trivial errors causing the search to fail?⁶

6 Compare Kernighan and Plauger (1974): “If someone could understand your code when read aloud over the telephone, it’s clear enough. If not, then it needs rewriting.” On the other hand, for a vigorously skeptical view of “superficial

In a textbook on the use of SDSS, which is more likely to be printed without typographical mistakes or bugs? Which could be quoted as a footnote in a scientific paper, enabling the reader to duplicate the dataset used by the author? If the fundamental schemas of the database are changed, which is more likely still to be correct syntax? And not least, which more eloquently says: *Come and be an astronomer for a night?*

There is a database of a sort beneath any work of IF, too: the collection of rooms and items, with their properties and spatial relationships. Here, too, natural language is concise and expressive even when it contains only elementary grammar:

East of the Garden is the Gazebo. Above is the Treehouse. A billiards table is in the Gazebo. On it is a trophy cup. A starting pistol is in the cup.

The combination of what is explicit and what is implicit in this 31-word source text is sufficient to compile an IF story file with three locations, one supporter, one container, and one miscellaneous item (the starting pistol).

So much for the accusation that natural language code is necessarily verbose. Because natural language can be used ambiguously or sloppily, it is also sometimes dismissed as “imprecise,” but this overlooks the fact that many important precise documents are written in natural language (standards documents, scientific papers, medical prescriptions), and that people are generally very exact in everyday conversation. Consider the following:

A weight is a kind of value. 10kg specifies a weight. Everything has a weight. A thing usually has weight 1kg.

A container has a weight called breaking strain. The breaking strain of a container is usually 50kg. Definition: A container is bursting if the total weight of things in it is greater than its breaking strain.

A lead pig, a feather, a silver coin and a paper bag are in a room called the Metallurgy Workshop. The bag is a container with

resemblance to natural language,” with a “bad effect in a lot of . . . commercial database-query languages,” see Raymond (2001). The furthest he will go is: “When your language is nowhere near Turing-complete, syntactic sugar can be your friend.” His comments assume, I think, that all loops and procedures are going to be explicitly written out, in a paraphrase of procedural code.

breaking strain 2kg. The lead pig has weight 4kg. The feather has weight 0kg.

Inform can now unambiguously test whether “a container held by someone is bursting.” Such descriptions narrate complex circumstances with extraordinary clarity. Also of note here is the representation of constant weights: we write “2kg,” rather than storing the number 2 in an integer variable, as we would in most conventional programs to achieve the above effect (perhaps using typechecking to distinguish weight-integers from other integers, if we can be bothered to be so pedantic). The code continues, and while it does begin to take the form of a procedural routine, there are only three instructions, each of which does something distinctive from the others:

Every turn when a container (called the sack) held by someone visible (called the unlucky holder) is bursting:

say “[The sack] splits and breaks under the weight! [if the player is the unlucky holder]You discard[otherwise][The unlucky holder] discards[end if] its ruined remains, looking miserably down at [the list of things in the sack] on the floor.”;

now all of the things in the sack are in the location;

remove the sack from play.

The mathematician Paul Halmos once said that one should write papers as though one were explaining matters to a friend on a walk in the woods, with no blackboard or paper to hand: to use the fewest possible symbols and notation makes for clearer exposition. Here we indeed minimize names. The procedure needs none, as it does not need to be called (we just say when it happens), and there are only two variables (“the sack” and “the unlucky holder”) and the five different loops implied by the code have no loop variables. In “a container held by someone visible” there are two potential searches (through containers and through visible people), though in fact internally it is optimized into a single loop; “all of the things in the sack” again implies a loop, as does “the list of things in the sack” and “the total weight of things in it.”

Many features of natural language are readily imitated in conventional code: a verb juxtaposing nouns is a procedure call with its arguments,⁷ an adjective is a function that returns true or false, a proper

⁷ Though it must be admitted that frequently occurring English words have difficult meanings, and verbs are not as easily explicated as this casual mention suggests. The copular verb *to be* is

noun (“Mr. Jones”) is a constant (or an object), a common noun (“man”) is a data type (or a class), prepositional phrases (“Jack is in the box”) could be regarded as tests of standard data structures such as trees, and so on. But Inform is considerably strengthened by two aspects of natural language less easily visible in orthodox programming: tenses and determiners.

If our aim is minimize the number of named variables, it is worth nothing that many variables in IF are essentially counters or flags, that is, totals or ways to remember past states of play: whether something has happened or not. In natural language, we can simply say “if the black door has been open” or “the number of things on the table,” avoiding the need for flag or count variables, with their names to be remembered, and the possibility of error in initializing them.

Instead of examining the tapestry for the third time, say “All right, so it’s a masterpiece, but is this really the time to make a detailed study?”

Inform has four tenses: the present (“is”), the present perfect (“has been”), the past (“was”), and the past perfect (“had been”), allowing us to discuss the situation in comparison with history either now (in the present) or at the point where the current action began (in the past).⁸

especially hard to analyze, partly because of the pre-eminent role played by equality in predicate calculus, partly because of an etymological accident in early English that conflated together meanings distinguished in most other languages. Thus the word “is” will sometimes be compiled to the computer-programming operation “=” (set equal), sometimes to “==” (test if equal), sometimes “=~” (the Perl operator for pattern-matching), and sometimes to a kind of implication not generally found in programming languages. (“If all the green trees are tall” means: if green and tree implies tall then . . .) Similarly, “of” is used in about ten grammatically different ways in Inform.

- 8 Tenses nevertheless pose formidable implementation difficulties. Consider what we must do to put ourselves in a position to answer the question “has the President ever been ill?”: clearly we need to maintain a continuous medical history in order, one day, to be able to look back over it and answer the question. But how often do we check the President’s health: daily? weekly? annually? (Inform has a convenient answer here: it divides time naturally into actions, and performs such “maintain a history” checks between actions.) A more serious problem is that “has the President been ill before?” poses an ambiguity. “The President” is not a constant, but a variable: the post is held by different people at different times. Do we mean the person who is President at the “point of reference” of the tense—i.e., the time when we ask the question—or the “point of event”—the time when the illness occurred? (See Reichenbach, 1947, for this way of thinking about tenses.) Suppose the question is asked in 2005. If we substitute a value into the variable at the point of reference, the question resolves to “has George W. Bush ever been ill?”; if at the

Determiners, on the other hand, underlie Inform's ability to imply searches and loops. These are the words at the head of noun phrases that give them a degree of definiteness. Even the innocuous word "a" might imply considerable activity: "if a person is carrying a container" is a double search, over both people and containers. Natural language is rich in determiners: Inform allows, for instance, "not all of," "at most three," "almost all," "some," "most," and so on.

Many of these points in favor of natural language programming would apply to a wide variety of situations (searching the Sloan Digital Sky Survey, for instance): but my belief that *the natural language for writing IF is natural language* is based ultimately on the special nature of interactive fiction. IF is based on a dialogue of text between reader (or "player") and computer, with both directions of communication prompted by textual possibilities supplied by the author. That means we have three agents describing the same situations—author, computer, player—and in an orthodox programming language such as Inform 6, the same idea accordingly has three different expressions. To specify a typical object, the author must specify all three of these: the source code constant `willow_pattern_vase`, the description text "willow pattern vase," and the parsing data `'willow'` `'pattern'` `'vase'` used to recognize the object in the player's typed commands. But words are just words, and it is repetitious and artificial to have to write them differently all three ways. A natural language description simply refers to "a willow pattern vase." It collapses the separation between author and player.

point of event, it becomes "has an incumbent President ever been ill during his term of office?" My own feeling is that substitution at the point of reference is more likely to be the natural reading of the question, but this is almost impossible to implement in a computer program. In this example, it would mean keeping medical histories for every person born in the USA, because in (say) 1974 we don't yet know that George W. Bush will be President in 2005: it could be anybody. The only practicable implementation is to carry out variable substitution at the point of event: to ensure that one will be able to answer "has an incumbent President ever been ill during his term of office?", one only needs to monitor a single person's health at any one time, which requires far less effort and record-keeping. The ambiguity is a genuine problem afflicting Inform's reading of past-tense conditions such as "if the noun has been open," where "the noun" is the object typed in the player's current command—and is therefore, like "the President," a variable in constant flux. Substitution at point of event means that past tense conditions relating to rapidly changing variables sometimes do not do what the user expects, which is a bad thing. Inform's documentation on past tenses strongly suggests that people use them with constants rather than variables.

Similarly for states of things. If we wish a bottle to be empty, half-full, or full then in most IF design systems we write code that stores (say) the number 1, 2, or 3 in some variable, then have to convert (say) 2 into the text “half-full” when printing out what is going on, and make the reverse conversion when parsing a command like “drop half-full vase”—a separation between author and player that obstructs the design process and increases the likelihood of bugs. It also results in IF code unnaturally full of numbers. Integers are so ubiquitous in computer programs that programmers easily forget how seldom they naturally arise in English text. Consider how the use of numbers (cloud 9, perhaps) to represent weather patterns would have obfuscated the following:

“Weathering”

A cloud pattern is a kind of value. The cloud patterns are cumulus, altocumulus, cumulonimbus, stratus, cirrus, nimbus, nimbostratus.

The Mount Pisgah Station is a room. “The rocky peak of Mt. Pisgah (altitude 872m) is graced only by an automatic weather station. The clouds, close enough almost to touch, are [a random cloud pattern]. Temperature: [a random number from 7 to 17] degrees, barometric pressure: [950 + a random number from 0 to 15] millibars.”

The use of square-bracketed substitutions in text also increases clarity. In Inform 6, for instance, the above room description would have required a routine to be written, expressing unnecessary intermediate steps (such as putting a random number into a variable) and so splintering the actual text that it would have been difficult to get a sense of its literary style by looking at the source code.

Inform does not aspire to recognize anything like the whole sweep of natural language, and in a few cases usefulness has been allowed to trump linguistic fidelity: in particular, it does not attempt to reject all *unnatural* language. But on the whole Inform tries to avoid eccentricity. The four self-imposed guidelines for the language were as follows:

1. A casual reader should easily be able to guess what a sentence does, and that guess should be correct.
2. The language should be economical, but not to the point where this compromises its intelligibility.

3. If in doubt as to syntax, imitate books or newspapers.
4. Contextual knowledge is best supplied by the author, rather than being built in.

Rules 1 and 2 are motivated partly by the basic aesthetics of natural language: the whole point was to be able to write a text, and a text should be legible. But there is another justification. One reason for COBOL's unexpected survival to the 21st century as a language for handling, say, financial transactions in the City of London, is that however much today's coders look down on COBOL as a verbose anachronism, they can still understand COBOL programs written in the 1970s and continuously used since: COBOL's priority of intelligibility over economy (rules 1 and 2 above) acts as something of a preservative against "code rusting." This is relevant to IF since IF authors—like novelists—are creating a cultural product to be accessible indefinitely, not a tool for immediate use and rapid disposal.

Rule 3 is best justified by the marks it has left on Inform's design. The most obvious imitation of printed books is that Inform projects have chapter headings, contents pages, and an index (with Inform generating the contents and index automatically). Inform's equivalent of "printf"—the formatted printing command that every programming language needs—also imitates print culture:

say "You've been wandering around for [number of turns in words] turn[s] now."

The syntax here mirrors the journalist's rule that quoted matter in square brackets can be paraphrase rather than verbatim text. Escape characters are also eschewed when we want double quotation marks to indicate speech inside text which is already double-quoted: the convention is that single quotes should be used, which are automatically converted to double-quotes in printing. This follows the standard bibliographic convention on citing journal articles that contain quotation marks. Definitions of new adjectives are set out as they would be in science text-books:

Definition: Something is invisible if the player cannot see it.

The greatest prize from rule 3, however, was the solution to an awkward question: how could Inform cope with data structures such as arrays?

Conventional arrays are clearly unviable in natural language, with their indexing subscripts (we are trying to abolish spurious number variables, after all). Searching and cross-referencing of arrays only makes constructions like AppleScript’s “item 23 of . . .” syntax painfully obstructive. We need a way to create, discuss, look up, change, and cross-reference arrays—ideally multi-dimensional arrays, with functionality broadly equivalent to Perl’s associative hashes—all without explicit subscripts and loops.

Printed books do, of course, include data structures. While they do sometimes lapse into diagrams, more often these data structures take the form of tables, typeset alongside the main text but given a title that the main text can refer to. Following rule 3, Inform does the same:

Table 2 - Selected Elements

| Element | Symbol | Atomic number | Discovery |
|------------|--------|---------------|-----------|
| “Hydrogen” | “H” | 1 | a time |
| “Iron” | “Fe” | 26 | |
| “Zinc” | “Zn” | 30 | |
| “Uranium” | “U” | 92 | |

Note that some columns are filled in, others left blank to be filled in during play (we shall suppose). We can look up data like so:

the atomic number corresponding to a symbol of “Fe” in the Table of Selected Elements

(which is 26), or we can set conditions such as

if there is an atomic number of 51 in Table 2

(no). We can sort the table in various ways, have entire blank rows, and so on: we can even loop through the table, in what is unavoidably a procedural style of coding, without the use of a loop variable. Rows are selected in turn, rather as they are in scripting languages for databases.

repeat through the Table of Selected Elements in reverse symbol order begin;
say “[symbol entry] is the international symbol for [element entry].”;
end repeat.

Rule 4 of Inform’s guiding principles, “Conceptual knowledge is best supplied by the author, rather than being built in,” leads us into issues covered in subsequent sections of this paper. Briefly, though: what

should an IF design system “know”?⁹ The examples given so far demonstrate that Inform “knows” something, at least, about spatial arrangement. It will object if we say that a box is in the canoe and also that the canoe is in the box, which it could hardly do without some functional knowledge of containment. Similarly, it knows that “four eggs” counts as “more than three eggs,” and therefore has a functional knowledge of cardinal numbers: we could find many other examples, but Inform’s ignorance of basic, shared human knowledge vastly exceeds its understanding. The following sentence arose in testing the second beta:

The life support unit fits the egg.

This led to a bug because Inform construed the verb as “support” and not “fits,” as a result creating items called “life” (which it guessed to be plural) and “unit fits the egg.”

Why allow such ignorance: why have rule 4?² One answer is that the alternative, a comprehensive contextual human knowledge, is far beyond the state of the art in artificial intelligence, but this is a cheap response: clearly Inform could get a lot nearer to the state of this art if its author made more of an effort. The real reasons are that the source text will be more self-explanatory if it takes less knowledge for granted, and that the language will be more flexible if it does not impose preconceived ideas. We shall see this latter point again in §2a when discussing taxonomy and the use of common vs. proper nouns.

Much of what is new in Inform 7—the emphasis on rules, type-

9 There are perhaps three reasons why one would say that Inform does not “understand” English: (i) it is a computer program, which experiences neither the sensory world nor human society; (ii) without wishing to get into the dispute over whether human knowledge is more like a dictionary or an encyclopedia, Inform has neither of these; and (iii) some of its mechanisms are so simplistic that we would certainly not ascribe human characteristics to them if we could see how they worked. On the other hand, Inform passes certain simple tests, such as the 1958 experiment in which it was shown that a five-year-old child can, if told about a furry animal called a “wug,” spontaneously use the plural “wugs” even though this is a word never heard before. So in our anxiety to insist that Inform has no real language ability and no real knowledge, we may be overlooking something. The ever-maximalist Jackendoff proposes that terms such as *f-understand* and *f-mind* should be used in place of *understand* and *mind*, where the *f-* prefix stands for “functional”: that, in effect, a working mechanism to accomplish something may be said to *f-understand* its task by virtue of the fact that it works. To that extent, Inform does *f-understand* non-trivial semantic concepts—which is what I mean by saying that it “knows” about spatial arrangement—and the declaration in its manual that “Inform does not understand English” is simplistic.

checking, actions that pay attention to success or failure, table searching—could have been achieved by incremental improvements of Inform 6, preserving most of the existing C-like syntax. They did not require the adoption of natural language, and nor did the new user interface (some of which in fact also works with Inform 6 projects). The general reaction of experienced IF writers to early drafts of Inform 7 was a two-stage skepticism. First: was this just syntactic sugar, that is, a verbose paraphrase of the same old code? (The cynical reader will have relished the lapse into “begin; . . . ; end repeat” above: iteration, and table searching, are generally responsible for the least “naturally” legible Inform 7 source text.) Second: perhaps this was indeed a fast prototyping tool for setting up the map and the objects, but would it not then grind into useless inflexibility when it came to coding up innovative behavior—in fact, would it be fun for beginners but useless to the real task at hand? It sometimes seemed to those of us working on Inform that an experienced IF author, shown Inform 7 for the first time, would go through the so-called Five Stages of Grief (Kübler-Ross 1969): Denial, Anger, Bargaining, Depression, and Acceptance. The following comment is typical of the Bargaining stage:

I would like to see it be as easy as possible to mix Inform 6 and Inform 7 code. [. . .] I also wonder if it might be possible to allow the user access to the Inform 6 code that the Inform 7 pre-processor creates. I can imagine some people wanting to use Inform 7 to lay out the outline of their game—rooms, basic objects therein, and so on—quickly, and then do the heavy lifting, so to speak, in Inform 6.

In fact, Inform 7 *does* allow the inclusion of Inform 6 excerpts, but in an effort to conceal this from the user, the ability is not mentioned in the manual until the final, intentionally not-for-beginners chapter: after some 260 examples of natural language doing “heavy lifting” have, with any luck, eased the reader’s passage through Depression to Acceptance.

Whether natural language will be widely accepted by the IF community, time will tell. Certainly the legibility of Inform source text depends very much on the willingness of the author to cooperate: sometimes being willing to type more text in order to choose names that make grammatical sense, for instance. It is also sometimes easy (and harmlessly amusing) to fall into double-entendres: “Men are usually transparent” and “A god is a kind of value” are genuine examples from

Inform's beta-testing.¹⁰ Sentences may look natural and yet be false friends, as translators say of words in foreign languages that look close to English ones but have different meanings: *actuellement* is not the French for *actually*. Whereas I myself would spare no effort to avoid unnatural source text, Inform being my baby, other testers were more interested in getting something done than in how the source looked. But the testing team is perhaps not representative of the system's eventual users, who will be able to learn a more stable language.

Extensive work has been done on natural language recognition in computing, and Inform makes no claim to originality in its inner workings: indeed, it could be regarded as a descendant of Winograd's program SHRDLU. (For a discussion of SHRDLU in the context of IF, see Montfort (2004).) The one practical lesson I would like to record here is that the biggest source of bugs in Inform came from my own imprecise knowledge of grammar. For instance, if we suppose that open and closed are antonyms, should "the door is not closed" be read as equating "the door" and "open," or as forbidding the equation of "the door" with "closed"? The result will be the same, but the second way is a better implementation, because it is more consistent with other verbs: the placement of "not" after the verb is almost unique to the copular form of *to be*, and its apparent association with the noun that follows is only a quirk of irregular English usage. We are likely to write cleaner code if we implement "is not" in the same way as "does not carry": and we will not end up accidentally parsing "does not carry not".

§1c. The Primacy of Rules over Objects

The third of the three fundamental changes in Inform 7 as compared with traditional IF-design systems, after the user interface and the adoption of natural language, is the replacement of an object-oriented model by a rule-oriented model.

The successful IF design systems to date¹¹ have mostly been object-

10 Men are no longer transparent: this was a leftover from Inform 6's quaint ideas about "concealment," which treated animate and inanimate holders the same, thus making it difficult to express the idea that someone is *purposefully* hiding something. Inform 7 eventually recognized this by focusing on the intention of the holder, not the state of the thing held.

11 Fighting talk. I really mean "the successful IF programming languages," since design systems wiring objects together for processing by a standard run-time engine—such as Scott Adams's Adventure International system, The Quill, AGT, or ADRIFT today—have at various times proved fruitful and popular. The rise of ADRIFT

oriented forms of C, with one important exception: Infocom's proprietary compiler ZIL, an object-oriented form of LISP. This happened partly because technology, 1975–95, obliged the source language to be efficiently compilable. (Until about 1990, an IF story file would be larger than the memory capacity of the computer that compiled it.) Those who wrote IF compilers tended to write them in C, were therefore fluent in C, and saw an IF-specialized form of C as an ideal system to use. The use of an object-oriented language with a strong class hierarchy was orthodox to the computer science of the day, and it neatly resembled the “hardware” of IF—the standard data structures used in story files, as exemplified by Infocom's Z-machine. In this way 1970s implementations of IF were taken to be the structural model for IF itself. I contend that, on the contrary, IF is not best described as object-oriented but as rule-oriented.

I concede that bundling properties together into object and class definitions, with inheritance from classes to instances, works well. My objection is rather to the doctrine that when components of a program interact, there is a clear server-client paradigm, that one component exists to serve the needs of another. The contents of a work of interactive fiction are typically not in such relationships. If facts concerning a tortoise must all be in one place, facts concerning an arrow all in another, how are the two to meet? It seems unnatural to have a tortoise-arrow protocol, establishing mutual obligations. Neither exists to serve the other. The tortoise also eats lettuce, meanders about garden locations, and hibernates. The arrow also knocks a flower-pot off a wall. By the same token, the world of a large work of interactive fiction is a world of unintended consequences. New and hitherto unplanned relationships between components of the “program” are added in beta-testing, something that the programmers of, say, a spreadsheet would not expect.

A second objection is that object-oriented code for IF divides into two quite different blocks of material: (1) the objects and classes, with their properties and specific behaviors, containing the materials for the game; and (2) the mass of usually procedural code containing the standard mechanics of play—the “library,” as Inform calls it (Infocom sometimes used the term “substrate,” perhaps a better image). This disjunction between specific rules (1) and general rules (2) is problematic for both. First, the general rules are hard to change, because although the

demonstrates that “constructor kits” have, in fact, made a comeback.

language is rich in flexible ways to modify specific facts (object properties), only the crudest mechanisms exist to modify general facts (library routines). The specific rules have the opposite problem: specific rules are easy to apply but the realistic constraints of the world model are often lost in the process, because “realism” is the province of the general rules that they pre-empt. For example: suppose a golden apple is inside a closed transparent box. General rules about impermeability forbid the taking of the apple, but here we shall allow the player access provided he is wearing a magic ring. How is this to be done? It seems inappropriate for such a one-off circumstance to be spatchcocked into general rules. So we should write a specific rule instead, but that means attaching behavior to a specific action: the taking of the apple. In Inform 6, we might modify the apple as follows:

```

before [;
  Take: if (apple in box && ring has worn) {
    move apple to player;
    “Your hand passes through the glass to grasp the golden
apple.”;
  }
],

```

And this is also unsatisfactory. Whether we regard it as behavior of the box, or of the magic ring, it is certainly not a behavior of the apple. What if the player tries to put the apple back again, or to put something else inside? Or turns the apple instead of taking it? What if the player is only allowed to carry four items, and is already fully laden? What if the transparent box is inside a locked cage to which the player has no means of entry?

The traditional solution is to rewrite the general rules to include yet another hook on which customized code can be hung. The accumulation of such hooks makes IF design systems grow steadily more complex as they age, and no matter what is added, it is never enough. As Andrew Plotkin observed to me, when early drafts of Inform 7 had reached this same impasse: “I’m tired of not being able to override some behavior, because there’s no hook there. I want the world to be made of hooks.”

Making the world of hooks, indeed. Where Inform 6 would use a central chunk of code with a few hooks attached ad-hoc, Inform 7 uses a line of hooks with pieces of code attached ad-hoc. These pieces are conceptually individual rules, and each is named. The “rulebooks” (these

lines of hooks) are highly modifiable, even during play. Special “procedural rules” take precedence, like points of order in debates:

Procedural rule when the ring is worn: ignore the can’t reach inside closed containers rule.

Of course it is still true that some rules are narrowly specific and others widely general, but many lie in between and there is no longer a categorical distinction but rather a sort of stratification. More specific rules take precedence over less specific ones, and the order in which the source code defines them is (mostly) irrelevant. This has two profound consequences: that the author can arrange the source as he pleases, choosing for instance keeping everything relevant to particular events or scenes together; and that four or five different sets of extension rules, by different authors, can peaceably coexist in the same work of IF, even though they all modify the same original definitions.

A system of gradations of rules has two prerequisites. First, the compiler needs a solid understanding of types: in particular, a way to judge whether one category (“an open container”) is a special case of another (“a container”). Secondly, the author needs a flexible way to describe the circumstances in which rules are to apply. Both considerations mesh well with the use of natural language: to make sense of the ambiguities of English, a strong typing system is needed anyway; and natural language is good at describing circumstances (“in the presence of Mrs. Dalloway”) and categories (“a woman in a lighted room”).

Part 2. Naturality in Theory

§2a. Semantic Analysis: A Sampling of Case Studies

It seems to me that the most profound difficulties in natural language recognition lie not in computing, where somewhat complacent textbooks exist to demonstrate standard algorithms (for instance, Allen (1995)), nor even in the arcane and by no means settled post-Chomskian field of formal syntax (Culicover (1997); Culicover and Jackendoff (2005)), but in the field of semantics. Definitions of semantic analysis vary. Some, such as Ray Jackendoff (2002, §9.1), stop little short of defining it as the theory of human thinking:

If you are not prepared to deal with at least language, intelligence, consciousness, the self, and social and cultural interaction, you are not going to understand meaning.

A more measured answer given by Davis and Gillon (2004) is that semantics is concerned with two basic questions:

What values are to be associated with the basic expressions of a language? How does the meaning of the simpler expressions contribute to the meaning of the complex expressions the simpler ones make up?

The source text for an interactive fiction is, however, more directly referential than most texts. Contrary to Fenollosa's observation that there are, in reality, no nouns in the universe, the universe of an IF contains nothing but nouns, and the word "stone" is in a sense a stone. A program such as Inform has an easy life compared with a human reader: it may work by forming a model out of its source text, but it does not have to embed that into some greater model of an "outside world." It has the further luxury of reading text guaranteed to be truthful, literal, and addressed directly to itself. Even so, most of the hard issues in semantics seem to arise in at least a toy form. For instance, Inform reads sentences quite similar to the notorious "donkey anaphora" example

If Pedro owns a donkey he beats it.

in which the rules governing "it" have been construed in an astonishing variety of ways by linguists, none wholly convincing. Does "it" substitute for a specific thing (Chiquita, Pedro's hypothetical donkey), making it a deictic pronoun, or for a universal ("every donkey such that . . ."), making it a bound pronoun? Or is our thought process not analogous to putting a value into a variable at all?¹²

Semantic analysis falls into two strands. Some linguists regard sentences as logical propositions, while others regard them as repositories of shared concepts, unspoken ideas, and sense-perceptions: see Goddard (1998) for a fascinating account of, for instance, colors and

12 Since this question was first raised by Geach in 1962, the literature of semantics has been routinely cruel to donkeys in the most callously offhand fashion. If they had a clubhouse, the motto over the door would be "Every farmer who owns a donkey beats it." But before you dismiss the anaphoric pronoun problem as trivial, try to explain why you didn't read "they" in the previous sentence as referring to the donkeys, even though it is the only plural noun phrase in this footnote.

emotions as expressed in different world languages, and for summaries of a number of provocative views on how meaning may be composed. Jackendoff uses the analogy of 18th-century chemists dimly glimpsing a Periodic Table of the elements: but others prefer a comparison with the Linnaean botanists indefatigably sorting plants into species, families, and so forth, until they in some sense discover a hierarchy of meaning.¹³ Either way, writers on semantics often seem to regard their field as being in its infancy, which is sobering when we consider that the first analytic grammar was written in cuneiform by a Babylonian.

There is no space here for a detailed discussion, nor should I pretend expertise in the field, but perhaps a selection of case studies relevant to IF may serve to support my general contention that semantic analysis is of central relevance to a theory of interactive fiction.

Q1. Does IF Overstress Hierarchies of Containment?

Inform, in common with all traditional systems for IF, has spatial ideas hard-wired into it: in particular the use of a tree structure rather than a “flat” map to model the location of things. It would be disquieting to think that we do this simply because it is easy to implement tree structures on computers. I therefore draw comfort from the number of linguists who also regard CONTAINER as a central idea: that it is, for instance, an “image schema” by which we metaphorically extend our bodily ideas of inside and out onto the world around us. The body being a container of great importance to us, we correspondingly picture the spatial world around us in terms of containment even when it is seldom so clear-cut. Why do we say that we are “in bed,” for instance, when we seem to mean that we are on top of something and are more than half covered by a blanket, and when we would probably say “no” if a child asked us whether the blanket is part of the bed? Similarly, if we are asked “Where are the teaspoons?” we are less likely to answer “In the kitchen” if they are not in plain view: we will say “In the kitchen drawer.” If A is “in” B which is “in” C, we are reluctant to deduce that A is “in” C. (For this reason, “in” is not a transitive relation in Inform.) On the whole I think the literature of semantics offers some evidence that IF’s hallowed

13 These analogies, pleasing as they may be, are not necessarily good ones. The Periodic Table is a relatively rare case in nature where fundamental science about simple objects in isolation leads to clear-cut distinctions and affinities. The human world does not resemble this picture. And Linnaeus intended his taxonomy to cover all concepts in existence, not merely biological species: but his project essentially failed in domains outside biology, and even there has frequently been challenged.

use of an object tree is genuinely natural.

Q2. Do Objects in IF Really Have Kinds, and What For?

Suppose we start from the premiss that an IF design system like Inform will have to produce a computer program that contains a wide range of different data structures, from the number 1815 to a packet of bits that in some sense models the character of Napoleon Bonaparte. This range is miscellaneous both in (i) the nature of the underlying ideas (numbers are not like men) and (ii) their binary representation in the final program, and if our compiler is doing any kind of sensible job, then these miscellanies will coincide: that is, the compiler will choose suitable forms of binary representation (ii) on the basis of the semantic concepts we seem to need (i). So much is true of any present-day compiler, gcc for instance, and it argues that an idea must exist at least internally of “kind” or “class” or “type.” (For most programming languages, C for instance, this idea is explicit in the source text, but I think we could conceive of an apparently typeless language where the idea was concealed.) We now move on to observe that since Inform is to take a piece of natural language as its source text, all such data structures are to be referred to by noun phrases: thus *1815* and *Napoleon Bonaparte*. This gives us a third miscellany: (iii) the range of possible meanings of noun phrases, and once again I argue that if Inform is behaving anything like sensibly, this must at least broadly coincide with (i) and (ii). In short, Inform’s objects have to have kinds because data structures do in any computer program, and also because these particular data structures marry up to noun phrases in English, meanings of which also fall into kinds—such as *NUMBER* and *MAN*.

We seem now to have chased back to human practices, so we might ask: why do human beings use “kinds”? A concise answer is given by Taylor (1989): “the function of categorization is to reduce the complexity of the environment.”

But I think we should distinguish between complexity of recognition and complexity of description. A child’s set of kinds needs to equip him to tell things that move by themselves from things that don’t, or to tell food from furniture: to cope with whatever presents itself next in an ever-expanding world, which is primarily a problem of *recognition*. Inform is interested instead in achieving the simplest written expression of single, fully-known situation in the imagination of the writer: a problem of *description*. A human being unable to categorize food would be ill-

equipped for life in any conceivable culture, but in a written work that happens not to involve any eating, there would be no need for such concepts.

In Inform it is a foundational principle that every thing has a “kind”: and the function of these kinds is to reduce the complexity of description. But as Borges reminds us in his spoof article on the subject, dividing up the world into kinds is not so easy as it looks:

These ambiguities, redundancies and deficiencies remind us of those which doctor Franz Kuhn attributes to a certain Chinese encyclopaedia entitled ‘Celestial Empire of benevolent Knowledge’. In its remote pages it is written that the animals are divided into: (a) belonging to the emperor, (b) embalmed, (c) tame, (d) sucking pigs, (e) sirens, (f) fabulous, (g) stray dogs, (h) included in the present classification, (i) frenzied, (j) innumerable, (k) drawn with a very fine camelhair brush, (l) et cetera, (m) having just broken the water pitcher, (n) that from a long way off look like flies.

And we must therefore go on to further questions about kinds.

Q3. Are Kinds Immutable During Play?

In Inform, `CONTAINER` is a kind. As with an integer in a typical computer program, if something is a container at the start of play, it will always be a container throughout. But is this only a convenience of computing? I think not. The property of being open may come or go, but we humanly expect a container to remain a container, and when a physical object is so treated that it absolutely loses its kind—for instance, a house being demolished, or a potato mashed—we tend to regard such an event as a dramatic change in which, in effect, one object is replaced by another that has little in common with the original.¹⁴

Q4. Can Something Be Only Partly of a Given Kind?

Inform, in common with other IF systems, provides a certain stock of kinds and makes every object belong to (at least one of) them. But inevitably, IF writers find that what they want does not exactly fit into any of the kinds provided for. They sometimes find themselves stripping away all the behaviors of the kind of something but being unable to

¹⁴ Just as, in typical IF situations, an object representing a glass bottle may be withdrawn from play when the bottle is smashed, and an object representing broken glass brought into play to substitute for it.

actually change the kind, an unedifying spectacle. Is a CONTAINER that cannot contain still a CONTAINER? Perhaps not. It depends what you call a “container.” From the point of view of IF, I suggest that the most persuasive way to define the meaning of a kind is to imitate a style of conceptual analysis proposed by Anna Wierzbicka in the 1980s. For instance, it is posited that the meaning of BIRD is that it is a “kind of CREATURE” with a collection of what might be called default expectations, none certainties:

people say things like this about creatures of this kind:
 they have feathers
 they lay eggs
 they can fly

The suggestion is that any of these properties can be overridden in particular cases without necessarily stopping the creature from being a BIRD: thus, FLIGHTLESS BIRD is not a contradiction in terms, but is a “kind of BIRD” about which people say that “they cannot fly.” This is exactly how rules are made about kinds in Inform, only to be contradicted by rules about specific instances of those kinds: see §1c above.

Q5. Should Inform Have Many or Few Kinds?□

Inform has very few kinds “built in”: indeed, dramatically fewer than rival systems,¹⁵ and this takes some explaining away. Let us proceed from the answer to Q2: the function of kinds in Inform is to reduce the complexity of description of what the writer imagines.

Is description made less complex by having many kinds, or few? There are approaches to that which want to make pretty well everything a kind, in order to make sense of determiners and proper vs. common nouns: for instance, “A registered nurse is in Piccadilly Circus” involves

15 Compare TADS 3, which after many years of thought (led perhaps by David M. Baggett in his work on the “WorldClass” library) currently has around 420 main classes, varying from generative-semantics fundamentals such as *SensoryEmanation* to highly specific gizmos: *Candle*, *AutoClosingDoor*, *ShipboardDirection*. This is not meant as a criticism, merely an observation that a rule-based natural language system may be most useful if it minimizes the number of basic meanings, whereas an object-oriented programming language may serve its users better by going the reverse way. It should also be noted that TADS 3 does not distinguish between concrete and abstract objects to the extent that Inform 7 does (again, not a criticism: I increasingly suspect this is “right”), and that this partly accounts for the profusion of classes.

the kind “registered nurse” being made actual (or “given an indexical feature”) by the determiner “A”; more contentiously, we can go the other way, since any proper noun like “Piccadilly Circus” becomes a kind just by writing “a kind of Piccadilly Circus,” or (say) “Imagine another Piccadilly Circus, but without the traffic.” Perhaps analogy works this way: “It’s like Piccadilly Circus in here.” Inform does not accept any of this: to Inform, a kind is not a meaning. Rather, kinds exist so that we can make (or question) generalizations about them. We will certainly not make “Piccadilly Circus” a kind, nor even “a kind of Piccadilly Circus,” and Inform can read “a registered nurse” as *either* determiner + kind (if the source text has already established that registered nurses are a general feature of the world) *or* as a one-off physical object that is not an instance of some more general phenomenon. Do we lose anything by not insisting that every noun phrase is headed by a kind? I think perhaps we do, but that we overcome it with only a little verbiage. For instance, the following is unambiguous to a human reader:

if a woman is in a room, say “[The woman] is in [the room].”

but Inform does not allow it, because it does not freely convert kinds back into instances through some theory of the determiner “the” adding an indexical feature to a meaning (“woman,” or “room”). Inform therefore obliges us to use a circumlocution like this:

if a woman (called the inhabitant) is in a room (called the place), say
 “[The inhabitant] is in [the place].”

We therefore reject one, impractically maximalist answer to the question: not every noun phrase is going to be a kind in Inform.

Instead, a kind has to be a widely applicable meaning, about which we can usefully lay down general laws. This precept immediately shows that some even apparently sensible concepts would make bad kinds. For example: JADE is arguably a bad kind in science because it combines things with no underlying unity. “Jade dissolves in acid” is a dubious statement and “the density of jade” a meaningless idea, since there happen to be two chemically unrelated minerals that accidentally look alike and are both called “jade” (see Fodor 1998). Or again, LEFT-HANDED MAN is arguably a bad kind to have when describing medicine, because it misleadingly over-specializes. It is true and statistically demonstrable that “smoking causes cancer in left-handed men,” but this statement might cause the reader to imagine that right-handed men are immune from

lung cancer, perhaps due to holding their cigarettes differently. We cannot escape the importance of context here. In the world of cricket, *LEFT-HANDED MAN* is a very useful concept, and in my own life, *JADE* is a perfectly good kind since I am not a mineralogist and do not care whether the only jade ornament in my house is jadeite or nephrite. The point about context is important: different works of IF will need different sets of kinds. Inform must therefore be careful to define only those kinds that would make sense in every work of IF, and allow the writer of each individual work to define further kinds for himself. Of course, this is also necessary on grounds of practicality, but I argue that it would still be necessary even if armies of programmers were available to create a larger and larger IF design system.

So are there many, or few, kinds that would make sense in every work of IF?

One distinction made by linguists about kinds of physical object is between “basic” and “superordinate” kinds. It is mostly¹⁶ common ground that kinds form a hierarchy, even though it is probably wise to keep an open mind about what form this hierarchy takes. We can say with reasonable confidence, for instance, that table is below furniture. The idea behind the basic/superordinate distinction is that certain kinds are basic because they represent the natural answers people give to the question “What’s that?” For instance, “A chair” is a more likely answer to the question “What’s that?” than “A piece of furniture,” so that chair is arguably a basic kind, and furniture is superordinate (which simply means, higher than basic). I am a little skeptical of this theory: to a Martian, I might indeed say “That’s furniture,” and in a chair factory I might say “That’s a swivel chair.” The set of kinds useful to a simple description of a situation inescapably depends on what the situation is. But to the extent that humans are more likely to be talking to other humans and more likely to be in domestic settings, perhaps there are kinds that are often “basic,” like the pictures in a child’s first book. So let

16 In the mid-1970s, Rosch conducted a series of experiments into what people consider “typical” objects: for instance, a chair is more typical of “furniture” than a music stand. Many different morals have been drawn from this about why chairs, say, are “typical” (they have a more basic purpose? they are physically simple? we use them more often? they exist in a typical house in greater numbers? they have no ancillary purposes? our parents teach us the word earlier? they are referred to by a phonetically simple one-syllable word?), and Rosch’s results do indeed give those who believe in a taxonomic hierarchy something of an uneasy feeling. Still, chair is a kind of furniture is a kind of artifact is a kind of thing: no?

us accept this basic vs. superordinate hypothesis. This immediately invites another rather large solution to our problem: not as big as “every noun phrase should be a kind in Inform,” but still pretty huge—“every basic kind of physical object should be a kind in Inform.” Inform should have a kind for chair, for instance.

Again, Inform is able to reject this answer because it is used to describe only narrow parts of the world at once, and never needs a framework into which the whole world can be placed. Inform’s strategy is to provide the fewest possible kinds—all of them superordinate, with perhaps one exception¹⁷—and to allow the writer to create whatever basic kinds are useful for the specific work being written at the moment. In a work of IF about chair manufacture, the writer is free to create kinds for FURNITURE, CHAIR, and SWIVEL CHAIR: but Inform provides none of these in the core rules present in every work. Indeed, the core rules provide only 13 kinds of physical thing, one of these being THING itself.

This minimalist approach has many advantages: most obviously that it is feasible at all, without sidetracking us into attempting some completist simulation of all possible worlds. But it should also be said that a “high-up superordinates only” approach to kinds has its drawbacks for an IF design system using natural language as syntax, as Inform does. One reason Inform’s core rules provide no CLOTHING kind is that the noun “clothing” is defective in English by not being countable. We cannot say “Two clothings are in the basket,” and even “A shirt is a kind of clothing” looks odd: “A shirt is an item of clothing,” maybe. While we could squirm out of this by substituting the word “garment,” that is not ideal either. (The same issue arises in one of Inform’s examples where a kind is created for ARMOUR.)

Q6. Do Kinds Have Overlaps? Can Something Belong to Two Kinds at Once?

A substantial change between Inform 6 and Inform 7 is that the new system is single-inheritance, that is, each thing can only have one immediate kind, whereas individual Inform 6 objects could belong to more than one class. While a thing can have more than one kind, this can only happen if kinds entirely contain each other: thus King Edward VII is a kind of MAN which is a kind of PERSON which is a kind of THING.

¹⁷ The somewhat suspect kind “player-character,” which is used to denote human beings from whose perspective the world can be explored. This kind exists largely for implementation reasons, and fits badly into the kind hierarchy. It may yet be abolished.

The corresponding question in linguistics would be whether, say, “a statue of Edward VII” belongs in any convincing way to a single kind. Is it a bronze pillar that happens, as a minor detail, to resemble a man; is it a man without animation; or does it belong in some sort of overlap of the outer fringes of `PILLAR` and `MAN`? Arguably natural-language meanings are neither singly nor multiply inherited from kinds, but are instead partially inherited (“a bronze pillar with some of the aspects of being a man” might define the statue reasonably well). This seems to me an exceptionally difficult question, and my inability to give a confident answer is a further reason why Inform is chary of defining too many built-in kinds: Inform takes the risk that we can, in fact, naturally represent the world with a strict hierarchy of kinds. (It mitigates this risk also by the flexibility it offers in attaching rules to selections of things on many bases other than kind alone.) There may well be alternatives to Inform’s assumption that kinds do not overlap, but they are not likely to be simple.

Q7. Are Kinds of Physical Things Like Kinds of Abstract Concepts?

An occasional, but persistent, trend in the development of Inform 7 has been that users want to use what Inform calls kinds of value (`NUMBER`, `TIME`, and so on) in the same way as kinds of object (`MAN`, `DEVICE`, and so on), often expecting that syntaxes typical of one must necessarily be valid for the other, and reporting it as a bug rather than a feature request if not. (Particularly troublesome was `SCENE`, a chronological extent, not behaving like `REGION`, a geographical extent.) The source code of the Inform 7 compiler would probably be cleaner if kinds of value and kinds of object were handled with greater commonality, and it may be true that further integration of these ideas is possible, but at present I still believe that they are different in nature. This is not because of the superficial difference that `NUMBER` has an infinite number of instances, but there are only finitely many `MEN`. Rather I would point to the discussion in Jackendoff (2002) at §11.4, and to an interesting point about what information passes up and down the hierarchy of kinds.

In this passage, Jackendoff quibbles with taxonomies by suggesting that the usual idea of properties being inherited from above—for instance, a `CHAIR` inherits the properties of `FURNITURE`—may be too slow, or unnatural, or inefficient on storage requirements, to make this the true cognitive picture of how people guess the behavior of things. (Maybe he has a point, but it seems to work for Inform.) But he then comes to an

interesting point about a sort of reverse inheritance:

More controversial perhaps is the question of whether a lexical concept carries structure relating it to *lower* members in the taxonomy. For instance, does TREE carry a list of its subkinds, including PALM, PINE, and PLUM? To carry things to an extreme, does PHYSICAL OBJECT carry within it a list of all its known subkinds? Implausible. On the other hand, people do use information derived by going down the hierarchy in order to draw inferences: this is “case-based reasoning”. Of course, case-based reasoning is notoriously unreliable [. . .] but people use it all the time nevertheless.

By “case-based reasoning,” Jackendoff means something like: “Mr. Jenkins, my English teacher, was always bossy at parties. So that’s teachers for you.”—unreliably arguing, that is, from particular cases. In Inform, the distinction I would draw between abstract and physical kinds is that a “kind of value” does contain information about its known instances when it is created, so that case-based reasoning is valid: but a “kind of object” does not, even though it must only have a finite number of instances in any actual work of IF. I think inheritance may work differently in the abstract and physical hierarchies.

Q8. Why are IF Design Systems Good at Modeling Objects such as Chairs or Rocks but Bad at Modeling Substances such as Water?

Since Jespersen (1909) linguists have divided nouns into “count nouns” (chair or rock) and “mass nouns” (water or dough), and it is apparent that these two categories of noun have different semantics: two oranges, some bread, but not two breads, some orange. Enquiries into the meanings of mass nouns involve the meaning of “is a part of”: one would probably not say that the left hand side of a loaf of bread “is a part of” the loaf, and certainly Inform’s implementation of parts makes sense for count nouns but not mass nouns. If we ask what mass nouns such as rope, sand, fire, and water have in common, indeed what makes them so troublesome for designers of IF, perhaps it is the problem that “part of” the thing is in one state, part in another.

One approach is to reduce the mass to individually counted atoms: say, to regard shingle as a pile of a hundred pebbles, or to regard rope as a chain of segments, or water as a collection of scattered puddles. This is sometimes manageable (ropes are often implemented as a chain of just

two segments—the two ends) but can be inefficient and impractical at run-time when the number of atoms grows large. Most systems for liquids give up when they are divided too far. Another approach is provided by Inform’s existing `BACKDROP` kind, which—for a multiply present thing like, say, the moon or grass, found in many locations at once—is already a primitive sort of mass noun: this works nicely but only with the huge assumption that the substance is immobile and is present either once in any given location or not at all. As a result, it is indivisible, and one cannot pick the grass. A further issue is that, whatever method is used, mass nouns impose formidable problems to the run-time parser, which must allow the player to refer to one part of something as distinct from another.

It might therefore be said that these difficult things should be left to ad-hoc implementations by the programmer, and that no general provision should be made for them. But the importance of mass nouns in semantics suggests that they are important in conceptual pictures of the world. This argues, first, that Inform ought to provide natural ways to deal with them; and second, that the semantics of mass nouns may tell us what these natural ways are. At the moment I incline to the “single multiply present object” strategy, rather than the “break up into atoms” strategy, since it seems more linguistically natural—all of the properties of a given puddle except for its location are, in fact, properties of water. It is also interesting to note Jackendoff’s suggestion that count nouns are to mass nouns as areas are to places, or as processes are to events.¹⁸ Inform has made less progress here than had been expected, but I hope to work further on this.

Q9. How Many Nouns Should an Action Involve, and Can They Be Optional?

It seems to be the case that verb phrases in natural language, such as may describe actions, take between 1 and 4 “arguments”: extreme cases would be “Helen sighed” and “Indigo bought a jumper from Kevin for £10.” Jackendoff (2002) comments that the 4-argument cases generally involve exchange between two agents—purchases, or wagers. Inform handles only 1 to 3 arguments in actions with any convenience, the

¹⁸ Inform’s index of scenes is indeed meant to be a “map of time” and is presented alongside the map of rooms laid out in space: Inform tries to treat time as like space and vice versa if possible, since natural language seems to do this, too. Inform’s implementations of region and scene both involve single objects with many relationships: perhaps this is further evidence for the “single multiplypresent object” strategy?

subject (“actor”) and up to two objects, so it is reassuring to think that a 4th argument is needed only in describing actions that require the simultaneous consent of two people: actions in systems for IF are impulses felt by a single person, and transactions are better regarded as a sequence of impulses (Indigo gives £10 to Kevin—3 arguments; Kevin gives the jumper to Indigo—3 arguments again).¹⁹

Optional semantic arguments are more problematic. Do they exist? For instance, “Lucy listened” and “Lucy listened to the machine” might be construed as: (i) one action with an optional argument; (ii) one action with a compulsory argument, but which is implicit in the first example and explicit in the second; or (iii) two different actions altogether, one a sort of ambient listening, the other a narrowed-down auditory enquiry. This has long been awkward for Inform: Inform 6 chose answer (i), but Inform 7 plumps for (ii), so that the bare command “listen” is implicitly read as “listen to the current location.” I also half-believe (iii): but then, as Jackendoff points out, the examples of “Neil ate” (a valid sentence) and “Olive devoured” (not so good) suggest that the whole phenomenon of apparently optional arguments may be more to do with lexical quirks—customary differences in usage between the words “eat” and “devour”—than any deeper semantic structure. If so, this justifies the stance taken by Inform 7 that at the semantic level all actions have a definite number of arguments.

§2b. Predicates and Quantification

Whereas §2a was concerned with what might be learned from big-picture semantics—in particular, the ideas of categorization that underlie human expression—much may also be gleaned from more basic linguistics, “down” at the sentence and even clause level.

It is a commonplace in most rigorous approaches to semantics that, whatever other cloudy ideas may hover around a sentence, its basic meaning can be transcribed as a proposition in predicate calculus. (See the introduction to Davis and Gillon (2004); for the mathematical background, see for example Johnstone (1987).) It would be wrong to

19 Gregor Hohpe’s wittily written piece “Starbucks Does Not Use Two-Phase Commit”, a blog posting from 2004 reprinted in Spolsky (2005), argues that computing algorithms that construe exchanges as a simultaneous act by two parties are (a) not like real-life transactions, such as buying coffee in Starbucks, and (b) often inefficient, compared to Starbucks. Starbucks does not use 4-argument verb phrases.

suggest that there is any one agreed logical framework that acts as a sort of *ur-language*: elaborately different forms of predicate calculus have been proposed, as have a similar variety of methods for converting natural language into logical propositions.²⁰ But there is a strong consensus that a large part of the meaning of simple sentences can faithfully be transcribed into mathematical logic, and that this can be done by applying some mechanistic algorithm to the original text. (This is certainly what Inform, and almost every other text-recognition program, does.) Predicate logic has certain computational advantages, but its benefits are more profound. It gives us a rigorous framework in which to make deductions, and thus to infer implied information from text. For instance:

“somebody is in an adjacent room”

i.e., Exists x : *person*(x) and Exists y : *room*(y) and *adjacent*(y) and *in*(x , y)

i.e., Exists x : *person*(x) and *room*(*parent*(x)) and *adjacent*(*parent*(x))

where Inform was able to eliminate the second of the two unknowns (the room) by noticing that it depended directly on the first (the person), the elimination being made by a substitution:

$y = \textit{parent}(x)$

The simplified form of the proposition is equivalent to the original but compiles to more efficient code. A second advantage of predicate logic is that it leads naturally to model theory, another mathematical methodology useful to “understanding” a text. Given a long run of statements about some situation—a novel, for instance—how are we to form a picture of what is going on? Inform’s usage of model theory will be discussed in §2c, but for purposes of the present discussion, I claim only that the project of comparing natural language with predicate logic has been very successful in linguistics. This suggests a close affinity between the two: so what is important in making predicate logic flexible may also be what is important in making natural language flexible.

Predicate calculus has various ingredients. If we consider a logical statement such as

20 Happily, Inform works in domains where most of the real difficulties do not arise. For instance it needs to understand “if,” which is easily reducible to mathematical logic, but not “because,” which is not. Consider the variety of meanings possible in sentences that take the form “Plants grow because X,” for instance.

Exists x : $person(x)$ and $room(parent(x))$ and $adjacent(parent(x))$

the first thing we see is a quantifier: “Exists x ” means that there is some value of x for which what follows is true. x is required to be a person, and to be in a room, which is moreover an adjacent room: thus, “somebody is in an adjacent room.” “Somebody” is an example of a determiner in grammatical terms, or at any rate the “some-” part of it is. Standard predicate calculus has only two quantifiers, “For all” and “Exists,” which is fine for “every man is mortal” or “some man is mortal” but is insufficient for Inform, which supplements them with Mostowski’s generalized cardinality quantifiers (“Most,” “At-Least-3,” etc.) along the lines suggested by Barwise and Cooper (1981). Thus “if three people are angry” becomes

At-Least-3 x : $person(x)$ and $angry(x)$

Just as generalized quantifiers expand the scope of predicate logic, allowing a wide range of determiners makes natural language more expressive. This is one example of how Inform became richer through comparison with predicate logic: to put it crudely, because quantifiers are important in logic, it seemed worth investigating whether Inform would gain from expanding its range of determiners. This step was taken about half-way through Inform’s development: the system worked perfectly well without. Inform’s presently rich support for determiners (see §1b for further bragging) is thus owed to an examination of the interplay of natural language and predicate logic.

But the most visible ingredient in predicate logic is the predicate itself. In

Exists x : $person(x)$ and Exists y : $room(y)$ and $dark(y)$ and $in(x, y)$

(“someone is in a dark room”), *room* and *dark* are examples of unary predicates, properties that are either true or false about any given thing, while *in* is a binary predicate, whose truth depends on a pair of things.

I suggest that an examination of the role played by predicates in IF points up failings in today’s IF design systems, or to put it another way, opportunities for tomorrow’s. I chose *room* and *dark* and *in* for the example above because they are easy to express in almost every IF design system whereas, say, *peacock*, *striped*, or *reminiscent-of* are not. While deciding which predicates ought to be built into an IF system is an important question—closely related to the discussion of kinds in §2a—

that can easily be a distraction from a subtler but perhaps more important issue: what order do the predicates in an IF system have? Are they unary, binary, or higher-order still?²¹

The original Crowther and Woods adventure game program (c.1977) essentially only had unary predicates: though there was a kludge to implement the famous bird-in-the-cage puzzle, the game had a limited grasp of spatial relationships. It relied on what we might call a unary predicate *carried*(x) rather than having any general notion of *in*(x, y). As a result only the player could carry things. This failing was remedied from the very first, MIT lab-written implementations of *Zork* (c.1978), bringing in the containment tree we use today, or from our point of view the first binary predicate: *in*(x, y). At some point early in the 1980s history of the commercial IF company Infocom, a second binary predicate *on*(x, y) was added: something could be “on” a table, and as a result respond to different actions from something “in” a box. For reasons I have argued in §2a, it is perhaps natural that these binary predicates were chosen first, but there are surely others in the world: yet design systems such as Inform 6 make no particular provision for them (except for what might be called *part-of*(x, y), and that is confusingly implemented). Moreover, binary predicates that ought to exist in Inform 6 do not, because they are wrongly implemented as unary ones. Just as we might regard *carried*(x) as being *in*(x, player), and say that the trouble with the Crowther and Woods program was that it allowed *in*(x, y) only for one privileged object y, so Inform 6’s unary predicate *worn*(x) is really an inadequate implementation of what ought to be a binary predicate *worn-by*(x, y). The result is a world model for IF that fails to distinguish between clothing and possessions for everybody except the player-character. More generally, the reduction of a binary predicate *b*(x, y) to a unary one *u*(x) that implicitly takes y to be the player-character—as with *carried* and *worn*—encourages a style of IF heavily centered on the protagonist.

A second limitation is that contemporary systems for IF allow the free creation of new unary predicates (attributes, as Inform 6 calls them: Boolean properties, some would say): *valuable*(x), say, or *large*(x), just as

21 Indeed, it seems to me that a good question to ask about any conceptual model of the world is: of what order are the predicates? For instance, one way to describe the historical development of particle physics would be to note that while physicists would ideally like to minimize the number of unary predicates in their descriptions of reality—*electron*(x), *photon*(x), and such—that is nothing like so earnest as their wish to reduce the number of binary predicates: the fields of gravity, electromagnetism, and so on.

we like. Most designers of IF would regard this as an essential tool. But there is little or no built-in support for the creation of new binary predicates. This is no great problem with inanimate objects, because they interdepend so little. It becomes a more serious restriction when dealing with objects that have internal states relating to other objects: in fact, to objects that have what we might call knowledge of the world around them: which is to say, to people. People have internal states very much based on other objects and people in the world. The things they recognize, the places they have been, the people they know. Throughout the first year of the Inform 7 project, I kept feeling that we should add some convenient way to implement people having memories, or knowledge: I now believe that this is a special case of a more general need to be able to create binary predicates.

Inform calls these “relations” and allows us to teach it new verbs that express them. The addition of this ability to Inform marked a dramatic change in the nature of the language: it almost immediately seemed impossible to imagine Inform without relations. For instance, there was a breathless rush of power at being able, for the first time, to write sentences like these in Inform source text:

Elizabeth loves Mr. Darcy. Darcy is suspicious of Mr. Wickham.

Underlying these sentences are the binary predicates *loves* and *suspicious-of*, so the ability freely to create binary predicates was crucial.

As a final note in justification of the importance of binary predicates, consider:

“if a policeman can see a gun that is carried by a criminal . . .”
 Exists x : *policeman*(x) and Exists y : *gun*(y) and Exists z : *criminal*(z)
 and *can-see*(x, y) and *carried*(y, z) ?

Note that the two binary predicates are the glue holding this proposition together: with only unary predicates, the variables could not interact, and the language would be much the poorer.

We have seen that binary predicates are sometimes reduced to unary ones, losing some of their expressive power in the process: thus *wears*(x, y) becomes *worn*(x). It also turned out to be interesting to teach Inform to go the other way, to expand certain unary predicates to binary ones. This is how Inform handles comparatives. For instance, if we define somebody as “tall” if they are 6 feet or higher, that gives us only a unary predicate *tall*(x): a person is tall, or not. But when Inform reads such a

definition, it automatically also creates the comparison “taller” (e.g., taller than 4 foot 6), which is a new binary predicate *taller*(x, y). It also automatically creates the superlative “tallest,” which in logical terms is a new quantifier:

“if the tallest person in the ballroom is a man . . .”

Tallest x such that (*person*(x) and *in*(x, ballroom)) : *man*(x) ?

This is a still further generalization of quantification, and illustrates again that quantifiers also matter.

I have argued that allowing Inform writers to create new binary predicates, not just new unary predicates, made an enormous difference to the expressive power of the system. How about going up from binary to ternary, or even higher-order predicates? Here I think gains are less dramatic. For one thing, binary predicates already permit propositions involving an arbitrary number of variables. For another, most interactions in the real world involve pairs rather than triads. Still a third reason is that storage costs become increasingly burdensome if we want to implement an arbitrary relation between triplets of objects: 100 objects means 1,000,000 bits of data, in the worst case. With that said, it is worth noting that most IF systems do indeed have a ternary predicate: *map-connection*(d, x, y), which asserts that a route exists in direction d between rooms x and y. This is sometimes seen more clearly in its lower-order reductions: *connects*(x, y), say, which really means Exists d : *map-connection*(d, x, y), that is, “there is a map connection between x and y”; or even *adjacent*(x), which means Exists d : *map-connection*(d, x, current location). The full ternary status of *map-connection* is not always recognized, and the failure to store it in a data structure at run-time that reflects this status has sometimes made it quite awkward to create new directions in IF design systems (Inform among them, at present). Inform also has a further ternary predicate *same-property-value-as*(x, p, y), used in the comprehension of descriptions like “people the same age as Henry”: here again, though, since p is seldom quantified over, this is really a collection of separate binary predicates, one for each property. A construction like “people who share some property with Mr. Darcy,” which would require quantification over p, is not currently allowed in Inform.

It might be said that these arguments are all very interesting for Inform, because Inform is based on natural language, but of limited application elsewhere. I increasingly think not, however. I feel that any IF

design system, even one based on procedural C-like code or a point-and-click constructor approach, could benefit from a similar study. Indeed, if we were to try to write down a systematic way to evaluate how powerful IF design systems are—for the sort of comparative review that the newsgroups used to be full of, in the early 1990s—the questions we might ask of any given system could usefully include: how can one quantify? how many variables can occur in a conditional question? and what orders of predicate can be created?

§2c. Model Theory and Discourse Representation

In §2b we saw the success of predicate logic as a representation for, especially, questions that can be asked about the current state of the world—questions implicit in instructions such as “if a policeman can see a criminal . . .” then do such-and-such. In this section we consider not questions but the declarative statements that create the world of an IF. This is a different problem: we have much simpler logical propositions to deal with—“Peter is a man,” say, or “Peter is in the treehouse,” each a single predicate with no quantifier—and we are in the present tense, and must be explicit. (Inform does not allow us to give it teasing clues like “Exactly four women are in dark rooms.”) On the other hand, we have a great many sentences to cope with, and have to integrate them into a coherent picture of the world. Again, we turn to a mathematical methodology: model theory.

Model theory may be summarized as follows. Suppose we have a collection of statements whose individual meanings we can grasp, but which may or may not affect or even contradict each other. We aim to test the compatibility of the statements, and also to find the simplest meaning that can be given to the whole, by constructing the smallest possible model for them: that is, the smallest configuration we can make in such a way that the text can be seen to be true of it. If no such model could possibly exist, then the text is nonsensical; but if it can, then the text is meaningful. This describes what Inform does quite well: the output is exactly a model of the assertions in the source. For example, given the source text:

The red box is on the table. In the box is a coin.

Inform converts these to predicate logic, but this does not take us very far:

on(the red box, the table)

in(a coin, the box)

In model-theoretic terms, Inform must now construct a universe set U and an interpretation function v , together with a choice of which predicates $on(x, y)$ and $in(x, y)$ will hold, in such a way that the two sentences above are both evaluated as truthful for this interpretation. This mathematical description gives us a more precise idea of the principle airily called “Occam’s razor” in the Inform documentation: we aim to minimize first the size of U and then the number of pairs $(x, y) \in U \times U$ such that $on(x, y)$ or $in(x, y)$ hold.

Inform expresses its output as a computer program, but if it were put into these mathematical terms (and its internal data structures do indeed have this shape) then the above sentences would produce the following model:

$U = \{O1, O2, O3\}$

$v(\text{the red box}) = v(\text{the box}) = O1$; $v(\text{the table}) = O2$; $v(\text{a coin}) = O3$

$on(O1, O2)$, $in(O3, O1)$

(Here $O1$, $O2$, and $O3$ are three different objects.) How is this done? On the face of it, the problem is circular: we need v to make sense of the text, yet it is the text itself that defines v . Moreover, the interpretation function v is in no way an exact correspondence between textual descriptions and members of U : in the above example, two different descriptions both map to $O1$. (A valid model does exist where “the red box” and “the box” are taken to refer to different objects, but that fails Occam’s razor, because it makes a universe set U that is larger than necessary.) Moreover v cannot be a simple look-up table or dictionary, because in a universe containing both a beech tree and an oak tree, $v(\text{tree})$ would have different meanings at different points in the source text: so it depends on context. In fact, though, the problem of building v and U is fairly easy, by observing clues such as articles and looking for names seen before. Inform works incrementally, first finding a model for sentence 1, then extending it to a model for sentences 1 and 2, and so on. There is no look-ahead to future sentences.

The next stage reduces—or it might be said expands—each sentence into what, in a rudimentary way, is a discourse representation vaguely in the sense of Hans Kamp (1981); though see also Catherine Emmott

(1997), whose book analyzes genuine excerpts of fiction from a point of view not so different from Inform's.

The essential point is that we can only progress by recognizing that a sentence contains both explicit and implicit meanings, so (despite the optimistic talk of §2b above) it is inadequate simply to replace the sentence by its mathematical analogue. From each sentence S , a set of “inferences” is extracted that semantically entail it, that is, such that a human reader will agree that S is a true statement about any model in which the inferences all hold. For instance, Inform reduces “In the box is a coin” to the discourse representation

$in(O3, O1), contains-things(O1), \text{Not}(room(O3))$

That last inference, “it is not true that $O3$ (the coin) is a room,” is typical of the unspoken assumptions in English that Inform stores in discourse representations.

Unfortunately, these are not typical sentences. What are we to do with “A door is always open”? One answer would be to store this as

For all x : $door(x)$ and $open(x)$

but, especially in the initial model-building stage of Inform, we want to avoid quantifiers like “for all” if we possibly can. (That way lies reasoning, which is difficult.) We get around this by treating the unary predicates *door* and *open* as being different in character. For instance, the following text:

The portcullis is a door. The portcullis is closed.

is not, as might be expected, represented as $door(O4), \text{Not}(open(O4))$. This is because Inform divides unary predicates into two sorts: being a door is the “kind” of the portcullis, an immutable and basic characterization, certainly true or certainly false; while being closed is a mere “property,” which we regard as less important, as something that might change in play later, and as something subordinated to kinds in any case. (A door can typically be closed but an animal cannot.) Inform therefore constructs a universe set that is a disjoint union of a set of objects O and a set of kinds K , with a kind function k that specifies the kind of something:

$U = O \cup K, \quad k: U \rightarrow K.$

For instance, if $v(\text{a door}) = K5$ then our examples so far have the model

$O = \{O1, O2, O3, O4\}$, $K = \{K5\}$, and the discourse representation of “A door is always open” becomes simply *open*(K5). This means that the inconsistency of

A door is always open. The portcullis is a door. The portcullis is closed.

can be detected without formal reasoning. By a simple substitution we accumulate the facts about the portcullis as being *open*(O4), *Not*(*open*(O4)), which is easily picked up as a contradiction. These methods enable us to ensure the continuing consistency of our model world, or else to report the proper errors.

It might be objected that a few propositions are inescapably universal: for instance,

For all x, y : *in*(x, y) implies *Not*(*on*(x, y))

If we want never to store universals, what shall we do with this? In principle we could handle it by extending our discourse representation for “In the box is a coin” from *in*(O3, O1) to *in*(O3, O1), *Not*(*on*(O3, O1)), but it is inefficient to keep generating these extra inferences. Instead, Inform knows that certain binary predicates have restrictions attached (the relation corresponding to *in*, for instance, is in Inform parlance a “various-to-one” relation), and Inform checks these restrictions explicitly when it looks through the “knowledge list” about an object in the model. It looks for a few more subtle problems than the blatant contradiction in the example of the portcullis above, too, but all of this is essentially an implementation issue.

A detail omitted above is that Inform records the predicates in a discourse representation with a degree of certainty attached—impossible, unlikely, likely, or certain. (There is actually a fifth state, “don’t know,” but there is no point recording what we don’t know in the discourse representation.) Inform uses these certainty levels mostly to handle adverbs like “usually,” but also in reading lines like:

East of the Grove is the Temple.

whose discourse representation includes certain: *map-connection*(east, Grove, Temple) and likely: *map-connection*(west, Temple, Grove). The merely likely nature of the latter predicate means that no contradiction will be generated if we later find that certain: *map-connection*(west, Temple, Sanctum), because only a clash of certainties is reported as a

contradiction. To my surprise this turned out to be both adequate and easy to implement, adding almost nothing to the complexity of the whole. It turned out that simplistic methods were perfectly good and that no formal system of fuzzy logic was required.

Generalized cardinality quantifiers (see §2b) have been fruitful in model-building, too. They enable declarations such as “Six soldiers are in the barracks,” which in logical form read back as

At-least-6 x : *soldier*(x) and *in*(x , barracks)

“At-least-6” looks vague, but Inform applies Occam’s razor and makes exactly six. (Similarly, if we define “tall” as meaning a height of at least 6 feet, and then declare that “A tall man is in the barracks,” he will be created with a height of exactly 6 feet, the minimum requirement to qualify.) But just as §2 began with a comment on the subtlety of pronouns and their meanings, as an illustration of the difficulties involved in semantics, so it will also end. Inform may have stocked its predicate logic with an exotic range of new quantifiers, but the rules governing those quantifiers are still only the standard ones for predicate calculus. If we have “For all x , such-and-such” as one sentence, then the next sentence is not able to refer to the same x , because the x is bound to the limited scope of the “For all” that we have left behind. That makes it difficult to accommodate text like “Every man has a weakness. It need only be found out.” It may be that the best solution is to relax the rules on the binding of variables by quantifiers, which is essentially Kamp’s solution to the problem of pronouns, but I wonder if it isn’t better either to (a) disallow the use of pronouns except in limited circumstances (the current solution), or (b) use an ad-hoc system for pronouns that does not try to accommodate them into the predicate calculus. Years of kicking the Inform parser this way and that, in its implementation of pronouns, has made me more skeptical than most philosophers of language seem to be about the “principle of compositionality,” that the meaning of a whole discourse is a function of the meanings of its constituent parts. Maybe the mind also handles pronouns with a poorly implemented lookup table, quite separate from its parsing of the rest of sentences: who knows.

To read today’s philosophers of language is to become aware that there are very clever people, with enormously lucid gifts for expression, who after millennia of work have almost no idea of how the mind of a writer works. But that does not mean they have nothing to teach the

designers of computer software, and in particular software that helps an artist to create a new work. In a very modest way, in a highly simplified “toy” environment, an IF design system is also an attempt to see what are the natural ways in which a writer imagines and expresses a situation: the closer it gets to this goal, the easier it will be to use, and the more powerful its results. If there is one lesson learned that I wish to record about the Inform project, it is that the professionals—philosophers and linguists—do know what they are doing, and are worth listening to. I spent the middle year of the project reading; it is how I should have spent the first.

Conclusion

Whether Inform 7 will be found useful, or whether it will join the zoo of improbable and neglected design systems through the ages that forms the most melancholy part of the IF-archive, others will tell. At time of writing, its user base can be counted on the number of fingers I type with, which I may say is fewer than ten: the four or five most substantial works of IF written in Inform 7 run to about 60,000 words each, so the total quantity of prose passed through it is still no more than might be found in a long novel. But regardless of how Inform 7 is ultimately received, I hope to have demonstrated in this paper that the practical experience of recasting IF into natural language has been highly suggestive of what might emerge as a theory for IF design. Work on semantics can help us by identifying what is important in conceptual pictures of the world, and therefore in stories, and therefore in interactive fiction.

References

- Allen, James, *Natural Language Understanding* (Benjamin/Cummings, 1995).
- Barwise, Jon, and Cooper, Robin, “Generalized quantifiers and natural language,” *Linguistics and Philosophy*, 4 (1981), 159-219.
- Davis, Steven, and Gillon, Brendan S. (eds.), *Semantics: A Reader* (Oxford University Press, 2004).
- Culicover, Peter W., *Principles and Parameters: An Introduction to Syntactic*

- Theory* (Oxford University Press, 1997).
- and Jackendoff, Ray, *Simpler Syntax* (Oxford University Press, 2005).
- Emmott, Catherine, *Narrative Comprehension: A Discourse Perspective* (Oxford University Press, 1997).
- Goddard, Cliff, *Semantic Analysis: A Practical Introduction* (Oxford University Press, 1998).
- Hertzfeld, Andy, *Revolution in the Valley* (O'Reilly, 2004).
- Jackendoff, Ray, *The Fundamentals of Language* (Oxford University Press, 2002).
- Jespersen, Oscar, *A Modern English Grammar on Historical Principles* (Allen and Unwin, 1909).
- Johnstone, P. T., *Notes on Logic and Set Theory* (Cambridge University Press, 1987).
- Kamp, Hans, “A theory of truth and semantic representation” (1981), reprinted in Davis and Gillon (2004).
- Kernighan, Brian W., and Plauger, P. J., *The Elements of Programming Style* (McGraw-Hill, 1974).
- —, *Software Tools* (Addison-Wesley, 1976).
- Knuth, Donald, *Literate Programming* (Addison-Wesley, 1992).
- , and Levy, Sylvio, *The CWEB System of Structured Documentation* (Addison-Wesley, 1994).
- Kübler-Ross, Elizabeth, *On Death and Dying* (1969)
- Montfort, Nick, *Twisty Little Passages* (MIT Press, 2004).
- Nelson, Graham, *The Inform Designer's Manual*, 4th edn. (The Interactive Fiction Library, 2001).
- Raymond, Eric S., *The Cathedral and the Bazaar*, 2nd edn. (O'Reilly, 2001).
- Reichenbach, Hans, “The tenses of verbs” (1947), reprinted in Davis and Gillon (2004).
- Spolsky, Joel (ed.), *The Best Software Writing I* (Apress, 2005).
- Stallman, Richard, *GNU Coding Standards* (Free Software Foundation, ongoing: October 2004 edition consulted).

Afterword: Five Years Later

Graham Nelson

23 January 2011

I

Inform 7 made its public debut, as a public beta, on 30 April 2006, so that it is now nearly five years old—an appreciable fraction of the age of IF itself, now perhaps 35. Can we say that natural language did seem natural to authors? IFDB records around 300 published works having used Inform 7. Averaged over the period, that means about 45% of all the IF written was written in natural language, with the proportion rising from 35% in 2007 to 65% in 2010. (Inform 6 usage fell from 30% to 13%, with works in non-English languages partly responsible for its survival.) This apparent popularity may not imply that our views were widely shared. The ease of installation, the existence of a peer group of other authors, brand loyalty to the “Inform” name, a feeling of newness, and so on, may all have been as, or more, important. Still, we can at least say that the community did not reject natural language.

In retrospect, this was a good period to be advancing a new design system. In my history of twentieth-century IF (DM4, §46), I wrote that “Changing conditions of computer networking have, throughout this story, had greater effect than the changing technology of the computers themselves.” In the period 2004–6, broadband internet connections—always on, and much faster than dialup—became ubiquitous among enthusiasts in the US, western Europe, Japan, and parts of China, with access in some places passing 10% of the population. Wikipedia established its dominance among reference sites, and its rate of growth stabilized to a level roughly constant since. Blogging went mainstream, with bloggers accredited to the US Presidential campaigns of 2004. Kodak, Nikon, and others abandoned production of non-digital cameras. We were entering the period of a mass user-generated web, a period when anyone with spare time and a creative impulse could fill both. This made it a good time to be reconsidering the use-case for any creativity software. We stopped thinking of our users as developers accustomed to installing software and creating file-system trees for

projects, and started thinking of them as bloggers. Inform's greatest strength today is that almost no time is wasted in getting started, from a cold download of the application to a first running story file.

If anything, the 2006 white paper underestimated the importance of this. Most compilers produce programs that are incomplete parts of what a consumer will experience, but Inform produces whole cultural products. Expressivity in terms of the look and feel of this product is perhaps as important as expressivity in terms of the underlying mechanics of play. At any rate, Inform pushes the idea of providing cover art and bibliographic text quite aggressively, and its Release options have become steadily more powerful. (In a controversial piece of social engineering, Inform now supplies cover art of its own if you provide none yourself.) In June 2010, it became possible to release an Inform project as a playable stand-alone website, needing no player-side downloads or installations at all. This brings its own challenges: run-time performance has become critical for the first time in years, with Javascript interpretation inserting an extra layer of overhead between the game's logic and what the user's computer actually does. Portable devices such as the Kindle, iPhone, and iPod Touch (all 2007) and the iPad (2010) have processor speeds much lower than their seductive modernity might lead users to think. The rise of "curated computing" poses further challenges for IF, with Apple's App Store regarding an IF story file as a forbidden download, which is why Frotz is banned from integration with IFDB. Between April and September 2010, Apple's draft regulations prohibited story file execution altogether, and though the policy was relaxed—Frotz remained available—it showed the vulnerability of our traditional packaging of IF into story files. Inform would very much like to have a "Release to iPhone" button, but Apple would forbid the result. Possibly the more liberal values of the Android App Store will triumph, but more likely not. We must move IF out from interpreter software and into the sandbox of a modern web browser, because that is now the only sandbox that anybody trusts. To sum up, in-browser Javascript looks to be the platform for future IF, whatever its nuisances.

Inform was far from alone in realizing that creative networking had arrived, of course. These five years have seen something of a diaspora, in which the community has slowly migrated to the web. The bare-bones catalogue files of the IF-archive FTP site have been replaced by IFDB, the Interactive Fiction Database, and for historical data we browse ifwiki.org rather than flat downloaded text files. Blog aggregators such as

planet-if.com, and forums such as that at intfiction.org, provide better filtering of news and practical support than the old Usenet groups ever did.

Is the Inform application an IDE, an integrated development environment, like Visual Studio, or Xcode, or Eclipse? Many people refer to it that way, though I don't care for the term myself. In my own mind, Inform is more like a word processor or graphic design program than a traditional coding tool. I am fairly sure that many of its users, especially casual ones, treat it that way and are comfortable doing so. But we have still not really reconciled this with the needs of harder-core coders, who would like integration with (say) the emacs text editor and the Subversion source code manager. I have only anecdotal evidence, but my impression is that a substantial block of users never use the Skein and Transcript. There remains some persuading to be done in convincing coders, in particular, that the Examples in the built-in documentation are intended to make up a standard library of code; whereas C++'s Standard Template Library or Python's modules are brought to the programmer as table service, so to speak, Inform's examples are a buffet. The idea is to help yourself to whatever looks appealing, taking as much or as little as wanted. But some users never take anything. Extensions, on the other hand, have been universally welcomed as a packaging mechanism, and 236 have been published on the Inform website. Expectations of quality are evidently higher than in Inform 6 days, and many extensions have a high degree of "finish," supplying documentation and test cases. Some, such as Erik Temple's GLIMMR suite, are remarkably ambitious.

Has Inform 7 succeeded in bringing new IF authors in? This is hard to say. IFDB statistics suggest only modest growth, if any, in the number of published IF titles per year. These do, I think, have better production quality, but that was a different goal. Also, we don't know what would have happened without it. Undoubtedly Inform 6 users, tiring of increasingly dated features, would have moved on, probably to the much more sophisticated TADS 3 (2006). On the other hand, some people coming in to IF seem to feel most comfortable with constructor-kit software to simplify the business rather than ramify it. (Adrift 4 (2002) remains in use as such today.) Perhaps Inform's relative ease of use has provided a path into IF writing for a part of this constituency.

What can be said, at least, is that Inform 7 has broken through into the classroom in a way that its precursors did not. This was a prime motivation for the expansion of the Inform website in July 2008 (in

making which the electronic media expert Liza Daly helped us considerably): once simply an access point for downloading the primary software, the Inform website was beginning to gather educational usage resources. The news blog's first posting was on a teaching program in Medieval Studies, which encouraged children to use Inform to simulate Viking village life. This is the IF equivalent of writing an essay rather than writing a novel, and it surely has value, even if the writer never tries again. The first third-party book about the system, *Creating Interactive Fiction with Inform 7* (2010) by Aaron Reed, was brought out by an educational publisher, Cengage Learning.

II

Once software is published, even with “public beta” disclaimers, it enters a new phase of its life-cycle: maintenance and incremental improvement. There are users with genuine needs, and sometimes there are even users who are right about what their genuine needs are. Speed and memory usage become relevant, and above all, the software needs to scale. After 2006, Inform projects became steadily larger and more ambitious—Aaron Reed's 365,000-word *Blue Lacuna* (2008) is the current record-holder. Inform's public face has slowly, tentatively perhaps, become more confident: the “public beta” status was dropped in 2008, and 2010 saw the arrival of an open bug-tracker, set up by Jesse McGrew, and a public suggestions forum, by Emily Short. This marked a change in the relationship between the writers and users of Inform. Similarly, we usually now give public notice in advance if features are likely to be withdrawn.

The information density of the Index has steadily increased, and so has its interconnectedness with source text and documentation. Fashion comes and goes as between dense and sparse displays, but for myself I generally buy Edward Tufte's view that status readouts should pack in all of the contextual data that clean design will allow. Problem messages also now key in with documentation and are augmented, in some cases, with a sort of backtrace of Inform's thinking. The number of distinct problem messages has risen from under 500 to, at time of writing, 842. A steadily narrowing focus on the specific cause of problems leads each maintenance release to add a further 20 to 50 problem messages.

Platform usage has changed surprisingly little in five years. Inform

doesn't send telemetry data home, so we have no data on casual users, but from bug report submissions it appears that Mac OS X and Windows are favored about equally by the hard-core IF community. Linux users, many of whom owned Apple laptops, initially either used Inform for OS X or else Adam Thornton's command-line Inform 7. 5G67 (November 2007) was a milestone build: Philip Chimento joined the Inform team as author of the Gnome user interface on Linux. We continue to provide builds for more rarefied platforms, such as Solaris, and for different Linux distributions. Adam's experiments with Sugar, running on the XO laptop, proved viable, but the XO never really lived up to the dream of changing education in the Third World; still, we wanted to be ready in case it did. The conspicuous omissions are iOS for iPad, where Apple's restrictions are a complicating factor, and Android. The latter has yet to make a significant inroad into non-handheld devices, but 2011 may be a turning point. In any case, it's not clear that many people want to write IF on a keyboardless tablet.

The hidden wiring of Inform has evolved more than its users might realize. Andrew Plotkin's Glulx virtual machine was created independently of, and earlier than, Inform 7, but Inform 7 users are now overwhelmingly its main customers. In 2006, Glulx was a reliable and fully worked-out design, exercised by a few hard-core enthusiasts, but it had not broken through into wider acceptance. Inform 6 had been patched to compile to it, but in 2001–6 there were a number of patches to I6 floating around, and the status of "Glulx Inform" was unclear. I7's public debut in 2006 necessarily chose a version of I6 to be used internally as a code-generator, and although this was a conservative choice it did make Glulx support official. David Kinder became custodian of this now canonical code-base and formalized a choice of bug fixes and patches proposed since 2001; in November 2010 this was formally designated Inform 6.32.

Inform 7 compiles natural language source text to I6 code, which in turn runs alongside a sort of kernel of run-time support code written natively in I6: for example, this code supervises the typing and parsing of commands and the running of rulebooks. In early builds, this support code was version 6/11 of the same "library" of standard I6 code that any I6 programmer would use, but in 3V01 (August 2006) the library was forked, with Inform 7 using the specialized version 6/11N. This roughly halved in size in 4U65 (May 2007) when the standard actions were reimplemented in natural language. Library 6/11N vanished entirely in

5T18 (April 2008), to be replaced by the “template,” repackaging the code in a better-documented and more modifiable way. Most users have no idea what goes on so far down the software stack, but translators to non-English languages made good use of this new freedom to reconfigure.

The Glulx story file format was first made available as an option for I7 projects from build 3V01 (August 2006). Migration was as easy as clicking a radio-button setting, but any sense of novelty was played down, with Glulx presenting itself by default in a cleanly minimalist way, with a text-only display in a single window. Graphics were enabled in 3Z95 (September 2006) and sound effects in 4W37 (July 2007), while external file input/output did not appear until 4X60 (August 2007). Because Inform was now acting as a sort of cut-out between Glulx and its users, it was also possible for Glulx to evolve without inconveniencing them. Thus, users could benefit from dynamic memory allocation, veneer acceleration (effectively making frequently needed Inform run-time operations into opcodes), and full Unicode support, without needing to do anything to take advantage. Moreover, Inform made a clean separation of the basics needed to run Glulx, which are built in, and more elaborate possibilities, which are left to extension-writers to experiment with: this freed Glulx enthusiasts to experiment. However, floating-point number support, added in to Glulx in 2010, is likely to find its way into the core Inform language in 2011: an interesting moment, since it will be the first language feature that can’t be compiled to the traditional Z-machine story file format. While Inform will not drop support for Z, we shouldn’t be held back by the limitations of a 1979 format on sentimental grounds. From 2011, Glulx is likely to become the default format for Inform, becoming opt-out rather than opt-in.

As noted above, Inform benefits from active development work on in-browser story file interpreters. It became able to release Z-machine projects to websites incorporating the Parchment interpreter with 6E59 (June 2010), and Quixe for Glulx with 6F95 (October 2010).

III

Behind the scenes, the biggest motive force for change was and is an end-to-end rewrite of the core compiler, a process that sometimes seems

as endless as the proverbial painting of the Forth Bridge. Even excluding user interfaces, the I6 compiler, the interpreters, and so on, the core of Inform is a 155,000-line program. Much of this code was initially improvised, then tinkered with, then optimized to run faster, then thrown away and replaced, then . . . and so on. Most subsystems are now on their second or third implementation. Much of this is invisible to users, as it should be. The maintenance release 6F95 hardly changed the language at all, but in fact over the three-month period since 6E72, the internal data structures “world_object,” “kind_of_value,” and “named_constant” had all been abolished and replaced by “instance” and “kind.” For the most delicate state of the operation, a week of sitting out on a terrace overlooking the Mediterranean but with my head entirely inside my laptop, Inform wouldn’t even compile for days at a time. I gazed vacantly at the distant white wakes of speedboats, like chalk on a baize snooker-table, while suppressing an inner panic. On the day when Inform finally passed the 1500-source-text test suite again, I stepped on a viper in the garden, a weird, springy, faintly supernatural and happily non-harmful experience. By the time 6F95 was released, my only ambition was for nobody to notice, and I’m glad to say that nobody much did.

This kind of reform typically goes in cycles. In the first stage, two different concepts appear in the language, existing for different purposes. As time passes, the semantics of Inform gradually change and these concepts gain more and more of the same behaviors. Users begin to see them as the same, and file bug reports when they differ. Once these are fixed, sections of the program that once had different aims become baroque infestations of code delivering the same functionality in two differently bad ways. At last, after a period of must-I-undergo-this-operation dread, a new unified implementation replaces the two old ones. In the case of 6F95, there were few visible benefits—the devil of it was that the old implementations had been exquisitely debugged—but in other cases, users did reap benefits. For example, in summer 2008 an abstraction of property ownership to “inference subjects” unified the compiler’s internal handling of object and value properties. The happy consequences in terms of lifted restrictions and removed exceptions consumed about half of the change log for build 5U92.

Rewriting is also mingled with actual writing, when the compiler’s code-base simply grows in size. Major extensions are rare, but have included: the implementation of descriptions as values; a new call

mechanism for functions allowing call by reference and automatic memory allocation for flexible-sized data structures; a more elaborate type-checking system, expressive enough for functional programming; and the relayering of the software to support the template and to modularize the software to isolate IF-specific code from the core language. Significant challenges remain, which I don't want to promise too much about, in terms of making Inform more flexible about its source language. While Inform 7 is successfully used to write IF in French, German, Italian, and Spanish, this involves nothing like the level of integration of source text and final output that we see for English.

We are also careful not to promise too much about our responsiveness to bug reports, but the historical record is quite good. There are usually one or two days every year on which Inform has no known defects. Bugs are increasingly arcane, or come down to subtle differences of interpretation of the documentation, or debates about what form of a problem message is most helpful.

The painting of the Forth railway bridge is, in fact, currently expected to finish in 2012. Similarly, the end-to-end rewrite of Inform's core is now about three-quarters complete and will culminate in its publication under the Artistic License 2.0, when it will, I believe, be one of the largest literate programs in the world. This isn't the place to comment further on that, but as well as an IF tool, Inform has been a sort of practical research project into the question of whether, and how, a computer program is a literary text. So it's a small ambition of mine to publish it as one.

IV

A brief sketch of the language's development over the last five years shows, I think, that while the 2006 language was focused on creation of interactive fiction—its core task, which it did extremely well right from the outset—it was far from adequate as a general-purpose programming language.

Punctuation has hardly altered, except that Inform now uses Pythonesque colons and indentation to denote blocks of code. We half-wanted this from the first, but had dithered, in part, because of concerns about accessibility to blind users. So, when build 5T18 (April 2008) finally adopted Python spacing, old-style begin/end syntax remained

legal as an alternative, and this is not merely an interim measure.

Inform's hierarchy of kinds of objects has stood the test of time remarkably well, and the process of whittling-down described in the 2006 white paper did seem to lead to a good compromise. The one kind we didn't really believe in, "player-character," was indeed removed as unnecessary in 4W37 (July 2007), just as the white paper speculated. The built-in set of kinds of value, on the other hand, has steadily expanded. Whereas in early 2006, Inform was optimized and only really suitable for the sort of computation needed to sustain a traditional interactive fiction, its core task, users soon wanted access to more and more of the features of a general-purpose programming language.

The first builds had no Boolean type, and indeed, Inform deliberately avoided the C convention that conditions and values were freely convertible; this was an example of strong typing being employed to protect the innocent. Also, the past tenses were supposed to reduce the need for true/false flag variables. But they didn't remove the need altogether, and Inform acquired a "truth state" kind with 5G67 (November 2007). It still requires explicit syntaxes to convert it to or from conditions, and this I think is the right outcome: it's not helpful in a language aimed at clarity for non-experts to have a silent casting between values and conditions. Other new kinds included "figure name" (3Z95, September 2006), "sound name" (4W37, July 2007) and "external file" (4X60, August 2007), provided to support the Glulx virtual machine's multimedia capabilities.

More central in importance were "indexed text" and "stored action" (also 5G67), which made it possible to dissect and alter parts of textual strings and actions for the first time. A suite of regular-expression phrases gave Inform text-handling abilities comparable with those of scripting languages such as Perl and Python. This led to an interesting syntactic decision: should Inform try to find a more novice-friendly notation for regular expressions? In the event, we decided not, on the grounds that when printed books include material in a non-alphabetic notation, they simply reproduce that notation verbatim: we did the same. The result is legible only to the cognoscenti, but that is better than being legible to nobody. Regular expressions can be so unreadable as to make one flinch, but the same can be said of chemical formulae, harpsichord scores filled with mordants, and the International Phonetic Alphabet, and specialist printed books reproduce all of those notations verbatim rather than trying to paraphrase.

“List of K,” added in 5J39 (December 2007), expanded the system of kinds in two respects. Firstly it was, like “indexed text,” a new dynamic data structure (whereas Tables were static in size); but, secondly, it was Inform’s first kind constructor. For any kind K, there was now a kind “list of K,” and so the set of possible kinds became infinite for the first time. Jesse McGrew contributed run-time code to 6E59 (June 2010) that made it possible for relations to be dynamic data structures, too; and one possible future addition might be a tree structure, since it is currently fiddly to build trees using lists or tables. Inform’s internals do support a primordial record or structure kind, “combination (K1, K2, . . . , Kn),” but at present this isn’t exposed to users. I am not yet convinced that this is needed, nor of how best to deal with it in natural language.

These additions and others, such as “equation name,” inevitably complicated Inform’s once-simple system of kinds. If 2006’s set was a farmyard, 2009’s was a zoo. Two rounds of simplification were made. The first round, in 4W37 (July 2007), systematized the built-in kinds and transferred their specifications into text files (which later migrated into the template). The second round, in 6E59, consolidated some obscure internal-use-only kinds, dropped others, and made many of the existing ones more expressive. For example, “relation” became the binary kind constructor “relation of K to L.” Phrases were allowed to have more complex kinds still, allowing for generic programming, thus:

To allocate (V - value of kind L) to (Q - list of L): . . .

The status of previously existing but anomalous pseudo-kinds such as “value” was clarified: these were generic kinds, allowing definitions like so:

To discuss (V - value): . . .

but not like so:

The thingummy is a value that varies.

At time of writing, there are 18 base kinds of value and 9 kind constructors built in. Experiment suggests that Inform’s users can sketch the hierarchy of objects (15 sub-kinds) pretty well from memory, but otherwise can only remember the handful of kinds they use every day: number, time, text, scene, list. This makes the Kinds index page, which tabulates the full range, all the more important, and intensive work has gone into its design.

The most novel feature of Inform's system of kinds is its dimensional checking: its ability, for example, to know that a length is written differently from a number, and that if multiplied by another length it will become an area. This was developed further in 5Z71 (April 2009), which added an extension for "Metric Units," declaring almost the entire range of SI units for physical calculations: length, mass, elapsed time, electric current, temperature, luminosity, angle, frequency, force, energy, pressure, power, electric charge, voltage, luminance, area, volume, velocity, acceleration, momentum, density, heat capacity, specific heat capacity. This was only partly done for the benefit of crazily simulationist IF writers; we also wanted to provide Inform with another educational use case. Once floating-point numbers are incorporated, Inform will be rather a seductive program for working out physics calculations. In the mean time the semantics of unit scaling—when it's natural to talk in kilometers, when to use millimeters, how to handle alternatives such as miles—turned out to be interesting to investigate. Just as the white paper speculated, we eventually added displayed equations to the language, imitating the look of scientific papers and textbooks:

Equation - Newton's Second Law

$$F=ma$$

where F is a force, m is a mass, a is an acceleration.

And such equations are allowed only if dimensionally correct.

The semantics of rules and rulebooks have steadily become closer to those of functions. In Inform's earliest design, they were simply procedures whose applicability depended on the current action. But by the time of Inform's first public beta, this simple arrangement was already inadequate. A few rulebooks were allowed to depend on an object instead (notably the "reaching inside" and "reaching outside" rulebooks used to simulate barriers in IF); and it was also possible for a rulebook to have a sort of return value, though the handling of this wasn't typesafe, and it was one of those distasteful corners of language design that manuals prefer not to cover. Rulebook variables—temporary variables whose scope was the currently executing rulebook—were introduced with 4U65 (May 2007). By this point, then, a rulebook had most of the semantics of a function in a typical programming language. In 6E59 (June 2010) it acquired the rest, with an ability to depend on values of any kind, and with a new and typesafe mechanism to return

them; and this became expressible as an idea with the advent of the kind “K based rule producing L.”

This may some day be seen as another step in a slow process by which rulebooks and phrases come to be seen as the same linguistic construct. Significantly, phrases became more like rules in 4S08 (March 2007) when it became legal to define them with constant values instead of kinds for their domains: for example,

To judge (M - the Duchess of Devonshire): . . .

To judge (M - a woman): . . .

were now both legal definitions, the former taking precedence over the latter when it applied. Compare the rules:

After kissing the Duchess of Devonshire: . . .

After kissing a woman: . . .

which have almost the same semantics. Similarly, phrases became optionally nameable in 6E59, and could be passed as values. This enabled a variety of functional programming tricks, notably map, reduce, and filter. Inform does not yet have a lambda operator allowing phrases to be created dynamically, but it may well get one.

Descriptions have become steadily richer over time. To itemize these gradual steps would be tedious, but the decisive change was in 5U92 (September 2008), when it became legal to define adjectives that applied to any kind, and not only to objects. Inform at last recognized descriptions such as “an even number” or “a recurring scene” on a par with “an open door” or “a portable container.” It now seems obvious that this should have been in the design from the outset, but that’s hindsight: in 2006, values and objects were seen as conceptually different. Moreover, after 5U92, adjectives could have different definitions in different concepts: for instance, “empty” currently has seven meanings, for texts, tables, rulebooks, and so on. Relations, perhaps the fundamental linguistic concept of Inform, were flexible right from the start, but verb definitions have yet to gain the flexibility of adjectives. Thus we cannot yet define “X follows Y” in a way giving it different meanings for scenes and for people. We hope eventually to enable this.

Inform’s Standard Rules define a conceptual world model for interactive fiction that, as its name suggests, is intended to be “standard.” In particular, the set of simulated actions—taking, dropping, going, and

so on—was initially carried directly over from Inform 6's 1993 implementation, which in turn had carried them from the genre-establishing IF of the 1980s. (They were, however, rewritten to be symmetrical, that is, to work when carried out by any actor, and not only the player; this led to one of the most stressful test cases in our suite, the 567-turn "Obedience," in which Clark Gable is instructed to perform every action in every way.) The traditional actions did, however, have long-standing deficiencies, and later in 2006 we took the opportunity to put some of these right: notably to clarify "removing" (3R85 in June) and to reform access—touchability versus visibility (3Z95 in September). Some of the more picaresque actions were dropped from the Standard Rules altogether: blowing, praying, digging, jumping over, filling, and swimming. A rewriting of the built-in actions in Inform 7 source text rather than Inform 6 primitive code (4U65, the following May) reorganized the rules, but didn't much alter their semantics. There matters rested for two years, but in 6E59 (June 2010) we finally adopted changes that recognized that player expectations had moved on from 1980s' conventions.

V

It would be too much to say that the Inform language is discovered rather than invented. Still, the design process sometimes felt like trying to find computational ways to describe what a set of already-existing source texts had in common. Clearly this sort of sentence should be legal:

Mr Darcy wears a top hat.

And so should many others. But what does that mean the language "is"? It's as if the programs came first, and the programming language for which they would be valid was a result of them, not vice versa. In functional terms, Inform is that language in which all of the Examples do what they look as if they should, but this is not an easy definition to work with. It upsets the conventional paradigm of a compiler as a device for translating expressions in one formally defined language into expressions in another; and it is correspondingly hard to give proofs of correctness. At any rate Inform's language is a subset of English that gradually changes in shape as we add new possibilities, or tidy up by

pruning away old curiosities. We want to simplify, and a number of changes pending for 2011 will remove redundant syntaxes. But it's not likely that the language will fundamentally alter, at this point.

I hope that the period of experimentation has not entirely ended, just the same. Looking back at the white paper, it's hard not to feel a slight pang of loss for that giddy period of anything-is-possible invention. There are things we never did achieve, quite: a satisfactory handling of liquids through a natural-language syntax based on mass nouns, for example. But, well, we got something made. And I still believe that natural language is the natural language for writing interactive fiction.

Challenges of a Broad Geography

Emily Short

A first version of this article was written in 2001 or 2002 and was founded solidly in the assumption that, for games of any significant size, the map was the ultimate structural principle. Puzzles, plot, pacing, and atmosphere all depended on how, and how quickly, the player was able to move through the simulated world.

Years of IF development have shown that that assumption was incorrect, or at least insufficient. There are now more games that are structured around the scene as a fundamental unit: the player plays through a conversation or an event, then the action cuts to a new time and place. Some of these pieces treat IF's room model as a stage for action, moving characters and props in and out of a single space rather than inviting the player to go wandering around a large consistent area. *Make It Good* (Jon Ingold, 2009) draws most of its complexity from character interaction and events rather than from map size; *The Shadow in the Cathedral* (Ian Finley and Jon Ingold, Textfyre, 2009) moves the player with speed and aplomb through a series of distinctive set pieces.

Nonetheless, there are still games written on a large scale and on a continuous space. *Blue Lacuna* (Aaron Reed, 2009), *The King of Shreds and Patches* (Jimmy Maher, 2009), and *One Eye Open* (Colin Sandel and Carolyn VanEseltine, 2010) are big, space-centric games that rely heavily on the map to control player experience.

And there is no reason why not. A large, well-designed world is a pleasure to explore. Setting is a strong point for games as a medium. As exciting as it is to see new formats of IF becoming popular, it seems unlikely that games set in big, roomy maps will fall entirely out of favor.

Making a playable game with a large geography requires some discipline about design and about user interface. If the player spends too much time wandering around without a clear goal, he may get exasperated and give up.

Mapping is a problem too: modern players are frequently less patient with the need to make a map than the players of old classics like *Adventure*, perhaps because geography as a puzzle in itself is no longer new and interesting.

And finally, if one has a specific story to tell, an expansive layout can make it hard to show the game to the player in the right order.

The challenges, then, are to build a layout that is easy for the player to understand; to support learning and discovery with a helpful user interface; to control access in a way that sets the desired pace for the game, through puzzles and other design techniques; and to build a strong sense of setting.

Principles of Playability

This article makes certain assumptions throughout about what constitutes good design for a large-map game:

1. The player's task is to explore the game world and see each step of the story. The author's task is to make that experience easy, fun—and unavoidable. Engineering a good map is one way to make sure the player will see all the sights and experience the game structure as it was intended.
2. The player should not have to spend much mental energy on memorization or note taking. It is fair to ask him to think about puzzles, but taking notes is generally boring and feels like homework. Besides, notes are hard on a player who is playing on a PDA, cell phone, or temporary browser window, and they can get lost between play sessions. Nothing should be difficult without also being rewarding.
3. The mechanisms that support these goals should fit the fiction of the game.

A Playable Layout

Structuring for Comprehension

Some map layouts are easier to remember than others. A maze is simply an arrangement of locations that defies easy memorization and defeats the player's ability to navigate; conversely, some structures are easier to understand and organize mentally than others.

Hub

A hub map is one in which a single location links several otherwise unconnected sections, like a roundabout with several streets leading away. *Zork II* (Marc Blank, Dave Lebling, Bruce Daniels, and Tim

Anderson, Infocom, 1981) has a “Carousel Room” that controls access to a number of arms of the map. (Diabolically, the carousel room spins to disorient the player until a certain puzzle is solved.) *One Eye Open* uses an elevator as a hub for movement between different floors.

The hub does not always consist of a geographically central location. It may be a special kind of space that exists in some other dimension of reality from the places to which it leads, or even a mode of transport. *Jigsaw* (Graham Nelson, 1995) lets the player travel to different times from the central room containing the time machine.

Formal Symmetry

Some maps avoid using a single nexus point but do structure themselves on a formal symmetry. Adam Cadre’s *Varicella* (1999) uses a highly symmetrical design for the palace, with staircases, hallways, and towers all arranged in a square. This design makes the palace seem familiar to the player almost immediately, since he can predict on general principles where certain rooms must be. Buildings in which each floor follows the same basic floor plan provide a handy vertical symmetry as well: see for instance *The Mulldoon Legacy* (Jon Ingold, 1999).

Similarities between matched rooms, whether thematic (all corner rooms are associated with one of the four elements) or functional (all corner rooms are bedrooms) can also be useful, as long as they don’t lead to overly mechanical design.

Street Map

A street map design allows the player to think of the structure of the world in terms of roads with a number of subordinate locations, such as shops. Hallways giving onto bedrooms and closets work the same way.

The player automatically creates a hierarchy between two types of location: arterials, which can be remembered with respect to each other, and cul-de-sacs, whose location only needs to be remembered with respect to the arterial.

Deadline (Mark Blank, Infocom, 1982) does a good job of balancing hallway space with subordinate rooms; subsequent games have done more to make the hallways themselves interesting in some way. *Varicella* allows the player to see characters moving through and into adjacent rooms, which does a good deal to make the hallways seem like genuinely connected space rather than discrete segments.

This kind of design is most helpful to the exploring player if

descriptions make it clear which exits lead to arterials. On a street map, the indoors/outdoors distinction makes this clear: on entering a new area, the player can decide whether to prioritize depth-first exploration (by heading inward to shops and houses) or breadth-first exploration (by checking out all the streets first).

Any pacing-controlling puzzles can then be placed along the arterials or at the entries to especially important cul-de-sacs.

It helps to augment this design with a rigorous implementation of OUT, so that if the player is in an indoors room, OUT will always move him towards the arterial. This eases some of the burden of remembering all the auxiliary directions.

Disjoined Segments

A few games—usually those with a primarily narrative goal—consist of a series of areas without any geographical connection between them. In effect, the game takes place in a series of small maps rather than in a single large one.

In most such games, plot triggers determine when the player will move to a new segment of the map. *Photopia* (Adam Cadre, 1998) relies heavily on this technique. New areas are associated with new scenes of the story, and it is not immediately obvious how the areas are related to each other.

The chief danger of taking the map apart in this manner is that the author, not the player, controls the player character's movements. On the other hand, the map subsections are usually fairly small and manageable, so mapping becomes unnecessary. Moreover, there's no need to control map progress with puzzles per se.

There are also segmented maps that are not plot-driven—that is, movements between segments don't rely on a plot trigger occurring but simply require the player to move between locations with a vehicle or portal. *Reality's End* (Harry Hol, 2003) does roughly this with a bicycle the player can ride to a specific list of places.

Arguably this idea is merely the logical extension of the principle that a game should elide the uninteresting bits—where “interesting” is determined by the goals of the work. Whereas *Queen of Swords* (Jessica Knoch, 2003) spends dozens of turns on the task of getting the player character suited up in fencing gear, another game might make this into a cut scene or allow it to occur in a single command. Similarly, taking out the uninteresting intervening bits in a world allows for a sense of

considerable space, at the expense of the freedom and control the player might otherwise have experienced in wandering across the map in any direction.

Hybridization

It is not necessary to adopt one or another of these techniques and apply it rigorously to your entire game map. Many large games rely on a combination of these. *Jigsaw* uses the hub system as its overall organizing principle for the game as a whole but relies on room-and-hallway organization for many of its subordinate zones. Indeed, it's moderately rare for a game not to make use of more than one.

Sources of Confusion

Many confusing elements of IF maps come about because the author was trying to model too realistically and introduced features that are hard to visualize or remember. Streamlining the map to essentials keeps gameplay focused and reduces mental overhead for the player.

Asymmetrical Exits

Asymmetrical exits are a classic feature of maze geography, designed to confuse. The player goes east to get from "Twisty Room One" to "Twisty Room Two"; in order to get back, he has to go up. This makes very little sense and is hard to draw on maps.

Mazes have become unpopular in recent IF, but some authors still use asymmetrical exits to represent curving paths: one goes north from the front of the house and winds up north of the house; to make the return journey, one has to go east. Some players like the sense of variety this gives, but many find it hard to visualize. Instead, one could add corner locations (so that the player could go east or south from the "Corner of the House"), or else make travel between the rooms a simple NW/SE type of connection. The second solution is often preferable.

Too Many Rooms

Even a small building in real life is likely to have more rooms than would be interesting to simulate in a game. If a room doesn't have any significant functional content (people to talk to, things to interact with, plot-significant scenery), then it's probably a candidate to be cut. So are rooms that have no personality other than as an antechamber to another

room: “In Front of the Courthouse,” say, if there’s nothing for the player to do there other than walk through to the courthouse proper. And service rooms, like bathrooms and closets, are often left out of IF locations, just as novels rarely mention every time the protagonist uses the toilet or changes clothes.

Identical Rooms

Too many identical rooms, even if their connections are symmetrical and straightforward, become a kind of maze. Because players moving back over a map are likely to be traveling more quickly and want mental shortcuts to orient themselves with, it helps to give distinctive names to each room.

Help from the Interface

In a large game, even a well-designed map is going to challenge a player’s memory, especially if the player leaves the game and comes back to it after days or weeks. Clues in the game’s interface or packaging can help.

Indicating Exits

It probably ought to go without saying that a game involving navigation at all needs some way to tell the player which directions are viable exits. The most common ways to do this are to call out the exits on their own line (in many old-format IF games such as those by Scott Adams) or to weave that information into the room descriptions. The art of writing a good room description is a complex bit of craft in itself so is beyond the scope of this article (but see “Writing Descriptive Prose in IF” by Stephen Granade and “Mapping the Tale: Scene Description in IF” by J. Robinson Wheeler, both in this book). But at the very minimum, any game that is larger than one room must somehow tell the player where he can go next.

“Can’t Go” Parser Messages

Now and then the player may get a little disoriented and try heading in a direction that has no exit. Many games by default tend to reply unhelpfully (“You can’t go that way”) or even snarkily (“Bumping into the walls again, are we?”).

A more friendly approach is to offer the player a list of which ways he *can* go. “You can’t go that way. The only ways out are north to the

chancel garden or south through the mysterious time vortex.”

Maps

Often it is worth simply telling the player how the world is laid out—at least in a rudimentary way—rather than leaving him to learn this information on his own.

From a gameplay design perspective, there are two possible methods: allowing the game to take notes for the player and build up the map as elements are discovered, or showing a complete map to the player in the form of an in-game image or bundled document.

Beyond Zork (Brian Moriarty, Infocom, 1987) pioneered the former technique for IF. At any given moment, there is a display of the surrounding locations that have been explored. Many ADRIFT games come with similar functionality when played in the standard ADRIFT runner. These systems add rooms to the map as the player discovers them and may allow the player to return to a room visited earlier by clicking on the map.

The advantage of automatic mapping is that it keeps up with the player’s experience and never gives away information he hasn’t found yet. It can be programmed systematically with a consistent box-and-line drawing procedure, and it doesn’t require elegant design or drawing skills from the author.

Automatic mapping may not be a good fit for the fiction when the player character already knows the area being explored. *The King of Shreds and Patches* gets around this difficulty by adding rooms as the player character discovers the need to visit them, whether he’s been there yet or not. Because the game is set in the protagonist’s hometown (London), this display mimics the cognitive experience of the character. The character always knows, say, where Tower Hill is, but the location doesn’t appear on his conceptual map of the area until he has been reminded of the place by a task that will take him in that direction.

Wishbringer (Brian Moriarty, Infocom, 1985), *A Mind Forever Voyaging* (Steve Meretzky, Infocom, 1985), and *Suspended* (Michael Berlyn, Infocom, 1983) came with physical maps of the game world. These maps don’t show every location to be explored, but they do orient the player. *Treasures of a Slaver’s Kingdom* (S. John Ross, Cumberland Games, 2007) takes the feelie map concept in a more abstract direction, with a “Cell-Map” that accompanies the game in PDF form. The simple grid of iconic drawings gives little away until the player has already found the

room pictured.

A stylish map can convey a lot of flavor and hint at the setting before the player even starts up the game: see the torn museum brochure for *Byzantine Perspective* (Lea Albaugh, 2009) or the ship's blueprint for *Piracy 2.0* (Sean Huxter, 2008). And because these are not box-and-line diagrams but the sorts of maps one might find in the real world, they help the player translate the game's discrete locations into a seamless and coherent space.

Implementation note: For games written in Glulx, Erik Temple has recently released a substantial package of extensions (collectively called GLIMMR) that facilitate building complex maps, compass roses, and other similar features. The suite includes an automatic mapping tool.

Compass Roses and Exit Lists

Some authors include a compass rose that displays which directions are currently available in the game. Kathleen Fischer's *Masquerade* (2000) demonstrates a simple ASCII compass in the status line; games with graphics sometimes display an actual image, like *1893: A World's Fair Mystery* (Peter Nepstad, 2002). Some games also distinguish rooms that haven't yet been visited by marking unexplored directions with a color or boldface.

While the compass rose isn't as informative as a full map, it does help players build a sense of continuous space.

A second option, appropriate when there are few exits per room or the author strongly prefers an all-textual presentation, is to have a text list of available exits displayed constantly in the status line.

Whatever the method, allowing the game to track which directions have been explored and what has been found there takes a lot of cognitive load off the player and reduces the overwhelming feeling that sometimes comes from entering a room with many side passages. If the game is doing the work of "remembering" what the player has seen so far, the player himself doesn't have to make a mental note to come back to this location and is free to relax and enjoy the scenery instead.

Pathfinding

Sometimes it is simplest not to require the player to understand the map in order to navigate. The original *Adventure* allowed the player to go to an adjacent room just by typing part of that room's name ("**>ORANGE**", e.g.) as well as by compass direction. Some Magnetic Scrolls games

followed up on this by allowing players to type >GO TO another location in the game: for instance, *Jinxter* (Georgina Sinclair and Michael Bywater, Magnetic Scrolls, 1987) will automatically find a path to the nearest appropriate room.

The GO TO (location) command didn't appear in Infocom games and was not terribly common in hobbyist games until the mid-2000s.

The primary coding challenge consists of figuring out the shortest path for the player to take from his starting location to his destination, and moving him along it, stopping if he meets a locked door or a spot where a puzzle prevents progress. If there are no obstacles anywhere on the map, we can safely dispense with pathfinding entirely and simply teleport him to his endpoint. But this rarely applies, and creating one's own pathfinding code can be a burden for new authors. The recent return of GO TO (location) in hobbyist IF is thanks partly to support in TADS 3 and Inform 7 that makes this feature easier to implement.

Using GO TO eases a couple of burdens for the player. It removes the need to remember a specific route, and it makes execution faster, especially late in a game where the player may be traveling long distances across the game map. Being able to type GO TO THE EASTERN MOUNTAINS may be a shortcut for 20 individual commands. At the same time, if the player moves too freely across the map, he may not get a sense of coherent setting, may miss elements of room descriptions that change over time, and so on. It is a method that needs to be used with some care.

Occasionally this kind of command is just a little too powerful and needs to be limited a bit. Eric Eve's *Nightfall* (2008) models a time-pressured journey through a city the player character knows well. In order to maintain the sense of distance and impose penalties for traveling a long way, Eve replaces the single GO TO (location) command with one that sets a destination, instead. The player then moves one step towards the location and receives another prompt. If he types C (for CONTINUE) at this point, he will move another step along his journey, and so on until he reaches his goal. If he ever chooses to make another command, the journey is interrupted.

Output is another challenge for GO TO (location) commands. One approach is to describe every room the player moves through as he passes through it—which can lead to very long, daunting output dumps. A more succinct description is often easier for the player to follow and more attractive, but this requires some extra code to describe movement

through various areas and to handle the case where the player meets an obstacle on the way: “You get as far as the Workroom when the locked green baize door stops your progress.”

Finally, pathfinding over a very large map may be process-intensive and may make the game pause longer than desirable. *Blue Lacuna* uses a modified pathfinding approach in which the player is allowed to travel to specific landmarks, rather than to any room on the map. This allows the game to simplify route information. It also establishes in the player’s mind that certain spots are important and memorable locations.

Compass Directions and Compass-free Navigation

The use of compass directions (north, south, etc.) goes back to *Adventure* and is one of IF’s most tenacious conventions. New players and authors have frequently challenged the convention, on the grounds that real-life navigation rarely involves exact compass orientation and that compass directions may be difficult to remember.

Several experiments suggest that getting rid of compass directions doesn’t make life easier for all players. Michael J. Roberts’s *Rat In Control* (2003) was designed to test this question by providing navigation by compass direction and relative navigation (LEFT, AHEAD, etc.). The subsequent discussion (<http://groups.google.com/group/rec.games.int-fiction/msg/206427d8cf5c6d3d> and the following thread) indicates that some players found the relative navigation deeply confusing and much harder to use than compass directions.

Nonetheless, a few IF games have substituted other innovative techniques for compass navigation. *Blue Lacuna* can be played either with or without compass; if the compass directions are omitted, the player travels simply by typing the name of the next room he wants to visit or the door or path he wants to take. *Gun Mute* (C. E. J. Pacian, 2008) is designed as a completely linear path that the player can travel by moving FORWARD or BACKWARD.

These are specialized options that wouldn’t work for every story, but they do have their place.

Pacing Access to Geography

Even a well-designed map with a strong interface can confuse a player who has access to too much of it at a time.

This is especially true at the beginning of the game, when the player hasn't yet learned what to expect from the game as a whole and may also be wrestling with new verbs, inventory, or conceptual content. Limiting the player to a few rooms or a very linear path at the outset helps limit the mental overhead.

The simplest and most popular way to regulate movement through the game map is to divide the geography into sections, with puzzle-controlled barricades.

Where to Block the Map

Bottlenecks

A bottleneck is a spot in the map where the only way to reach a whole area is via a single door or exit. It's easy to control with a puzzle.

A hub design comes with a lot of natural bottlenecks and is easy to design into an early game sequence. The author can set the early game in one of the arms of the map. Gaining access to the hub is then a major reward that opens up access to much more of the game world.

Rings and One-Way Paths

A map with a ring construction provides shorter paths between parts of the game world. A game built on the street map model is especially likely to have circuits and areas of the map that can be entered from several points.

This is good for realism and variety, but it makes it harder to block map areas. Putting a block-point puzzle at each entry to the area means that the player may solve the first puzzle and then go back and try to solve the second as well, only to be disappointed when it leads to the same area he's already visited.

One way to resolve this is to place two blocks but give them an identical solution—a key or passcard that opens both doors, for instance.

Zork I (Marc Blank, Dave Lebling, Bruce Daniels, and Tim Anderson, Infocom, 1980) solves the ring-construction problem with passages that can be traveled only in one direction: the trap door that closes behind the player, the fireplace that can only be ascended. Slides, chutes, and spaces too narrow to re-enter from the other direction ensure that the player will move through the ring in the direction chosen by the author.

A Stop For the Night (Joe Mason, 2003) has an especially clever

variation on the one-way path. A certain area is dark; it has multiple entrances, but the only way to bring in light is to enter from a well-lit direction, opening the door in the process. Entering first through the wrong entrance leaves the player fumbling in shadow. The design guarantees that the player will first reach the room from the correct direction, but once the puzzle has been solved, the room becomes part of a two-way path and is easy to navigate for the rest of the game.

Forbidden Regions

A whole area, with multiple entrances, can be made off-limits until a puzzle is solved. This is often useful in broad outdoor spaces where too many gates and fences would be hard to account for fictionally. It also has the advantage that once the area is opened, the map becomes easier to move across.

The trick then is to present the movement block to the player not as a single obstacle (“The heavy iron door is locked!”) but as a reason why every room in the region is not enterable (“You take a few steps into the swamp, but the snakes in the water scare you back to shore.”)

How to Block the Map

Physical Blocks

There is a barrier that prevents a player from moving a certain direction from a certain location, usually until he has fetched some useful puzzle-solving device from another area of the map (and thus proven that he has explored adequately and is prepared to move on). The lock-and-key puzzle is the classic example, though it has a number of equally obvious variations: the monster who must be killed with a specific weapon, the NPC who must be placated with a specific gift, a bridge that is lowered by a lever in another area of the map.

Elegant variations involve requiring more than one item to defeat the block-point puzzle, but from a map-design point of view, the effects are the same. The author must put a bottleneck into his map, a single passageway that controls access to the next area.

It’s often satisfying to put a block-point where the player will see it before he has any chance of finding the solution: this gives him a sense of where things are going and what it is he is ultimately supposed to solve.

Conceptual Blocks

There is a barrier that prevents a player from moving from one location to the next, but it does not require any objects from any other areas of the map.

Conceptual-block puzzles in early games often came in styles that required patience or a flash of pure insight to solve: mazes, doors locked with riddles and decipherment puzzles, secret doors only discovered by methodical searching of the whole map. Such puzzles do not guarantee that the player will experience the map in a different order than if the conceptual block were not there at all. They have largely fallen out of favor in recent years.

Recent games more often use conceptual blocks as a way to teach core gameplay techniques. In *Flight of the Hummingbird* (Michael Martin, 2010), the player has to use his flight powers in order to leave the prologue and get access to the midgame—not because that puzzle requires a complete exploration of the prologue area, but because if the player does not yet understand how flight works, he won't be able to succeed with later puzzles based around the same ideas. Crucially, this puzzle also has more feedback than the typical riddle-style puzzle: there are several ways to attempt the flight and fail informatively.

A player who has already played once can easily move through a conceptual block puzzle; knowledge puzzles, unlike those involving portable objects, tend to be simple to repeat. For that reason, games meant to be replayed several times can make good use of a conceptual puzzle with a simple one- or two-step solution (like speaking a password) rather than a many-step solution (passing through a maze or configuring a complicated machine).

On replay, there will be no entertainment in finding the multi-step solution (because it is already known), so it may be better to avoid the tedium and go with a simple solution.

Continuous Need

The player can enter an area freely, but in order to survive and explore it, he has to have some kind of equipment. The archetypal form of this is the light-source puzzle, but it can take a number of other functionally equivalent forms: the underwater area that can only be visited with scuba gear, the mine that requires a gas mask, the sensitized floor that can only be passed over with levitation boots.

Continuous-need puzzles are an excellent way to shut off forbidden

regions or ring structures—any part of the map that isn’t controlled by a single bottleneck. The player can enter the water anywhere he likes, but he’ll always still need the scuba gear.

Change Over Time

Another approach is to work with a fairly small map that the player can learn quickly and to then add complexity to that map as the game’s plot unfolds.

Degeneracy (Leonard Richardson, 2001) shows the world under the influence of a strange spell that causes it to become progressively less well-implemented. *A Change in the Weather* (Andrew Plotkin, 1995) and *Blue Lacuna* both make significant landscape changes due to weather and flooding. *Zork III* (Dave Lebling and Marc Blank, Infocom, 1982) includes a major, geography-changing earthquake, while *A Mind Forever Voyaging* (Steve Meretzky, Infocom, 1985) has the player visit the same location at several different historical periods.

It is even possible, with care, to make the player directly responsible for environmental changes. *First Things First* (J. Robinson Wheeler, 2001) allows the player to explore the same areas over several different time periods. The pattern of the map is consistent from one time period to the next, but each era has its own quirks—which change as the player’s activities in one time period ripple forward to affect the future ones. Likewise, items in early periods take on new significance as the plot progresses.

Gating by Plot

It is sometimes workable to have the player character simply refuse to take a certain exit until he has a plot-based reason to do so (“You have no reason to head out into the jungle right now.”).

This technique is generally frustrating in games that otherwise promote free exploration: the player may ask himself why it’s fine for him to spend hours mapping out the deserted forest, but his character refuses to enter the woodcutter’s cottage. It works best in stories with focused narrative development or a strongly characterized player character.

Rewarding Effort

Gaining access to a new part of the map should be a reward in itself. If the author makes the player work hard to get through a door, what lies

on the far side of the door should repay the effort.

As a game progresses, it is less and less feasible for the solved puzzle to open major new areas—after all, things are drawing to a close, and the amount of new geography left to explore is diminishing. In that case, it's appropriate to reward the player with bigger and better plot secrets, more exotic loot, or tools that have a transformative effect on the game world. *Enchanter* (Marc Blank and Dave Lebling, Infocom, 1983) amuses the late-game player by handing out spells that have funny effects when cast on random objects.

When the Barriers Are Down

Planning a game's geography doesn't just mean planning for how the player will experience it the first time he moves through the rooms. Especially in a large game, the player will continue moving across areas of a map that have long since been solved.

Making a large map playable in the mid-to-late stages of a game is no longer a question of learnability but of access and variety. At the very least, the map should not become too much of an impediment to late-game play—as it can be if too much space stands between the sole remaining puzzles. At best, the late-game transformation of the map can connect things in new ways that shed new light on the player's entire experience.

Limiting Frontiers

Suppose we say the frontier of a game map is any area where there are still puzzles to be solved and new exits to be used. The problem with having frontiers on all sides of a game map is that it expands concentrically, forcing the player to traverse ever more territory between puzzles.

There are a variety of design strategies to get around this. A tight hub design is the ultimate in limited frontiers: one is always pushing outward from the same location. Some games encourage the player to push in one direction and discourage exploration along other sides of the map, by strategic placements of walls (*For A Change*, Dan Schmidt, 1999), coastlines (*Anchorhead*, Michael Gentry, 1998), and trackless desert (*Infidel*, Michael Berlyn, Infocom, 1983). Some use a street map design where most or all of the streets are available early on, but puzzles block off the buildings and interior spaces (*Wishbringer*).

Closing Used Areas

Some games also make some areas of the map accessible only once. This effectively embeds a miniature game—a small segment with its own goals and requirements—into the larger structure. The player can't come back later, so that area doesn't cause the game world to sprawl unnecessarily.

The chief danger of using this method is that it's easy for the game to become unwinnable if the player returns from this subsection without achieving all of its goals or if he accidentally drops a critical object there.

Shortcuts and Vehicles

Applied injudiciously, the one-way ring structure can be annoying, in that it forces unnecessary trekking even at those stages of the game where all the puzzles involved have been solved.

Likewise annoying is any puzzle—maze, complicated machine, irascible creature—that requires a complex series of steps to traverse each time the player wants to pass through that area.

A canonical solution is to provide the player with shortcuts, ways to pass between rooms that only open up when the puzzles have been solved. A new shortcut should not be the sole reward for solving a puzzle, but it does make a nice bonus. *Curses* (Graham Nelson, 1994) deals with this by giving the player magical teleportation powers, *Zork I* by having the Cyclops crash through the wall and make a new opening; there are all sorts of reasons one might provide for removing the barrier between two rooms unexpectedly, without the player having previously thought of that barrier as a puzzle to be solved.

Vehicles also make good shortcut devices: they let the player cross large territories in a single turn, but only after he has found his car keys or the bus pass.

The Gameplay Content of the Map

Navigation as Challenge

IF has a long unhappy history with mazes, sections of a map that require careful mapping simply to navigate. Mazes in new works of IF are generally received with horror and disbelief: hasn't the author gotten the memo that mazes are out?

Many players hate mazes because they are boring and fiddly to solve. Frequently, the player realizes *how* to solve the maze but then faces a long sequence of steps in order to carry out the solution. Since the fun part of the puzzle is discovering the solution algorithm, not the task of applying that algorithm, classic mazes often mean anticlimax and dullness.

Nonetheless, there are still authors who find interesting navigational challenges for their games. *Flight of the Hummingbird* allows vertical movement but puts some restrictions on how the player can fly, which means that traversing an area can become a challenge in timing and coordinating movement turns with the other turns required to sustain the hummingbird hero's flight. *Delightful Wallpaper* (Andrew Plotkin, 2006) offers a space that the player has to navigate in a specific order to unlock its secrets.

Good navigational challenges avoid the pitfalls of the classic maze. They don't leave the player confused in the end (the map as a whole is still comprehensible even if part of it is a challenge to get through the first time), they only need to be traversed once to be "solved" (so the player doesn't have to repeat a long and tedious set of steps), and they minimize the busywork of implementing the solution.

Designing in Levels

Space in commercial, graphical games is often described in terms of level design. A level is (ideally) an area with a coherent aesthetic concept and gameplay style and is self-contained enough to be distinct from other levels on a map. The sequence of levels teaches the player the core game mechanics and offers increasing difficulty.

IF puzzles are often too varied to lend themselves to the same kind of systematic progression. Nonetheless, sometimes it's appropriate to connect specific areas of the map with specific tasks or abilities for the player.

One Goal per Level

A simple style of level design is to use a strongly segmented map (often with a hub structure) and then give each section of the map a unique sub-goal.

In *Jigsaw*, the player must travel between time zones and prevent the antagonist Black's meddling in each time zone. In *Tookie's Song* (Jessica

Knoch, 2002), each seasonally themed area has a single treasure for the player to collect. *Gigantomania* (Michelle Tirto and Mike Ciul, 2010) has four sections of gameplay, each with its own distinct protagonist, linked by theme rather than by storyline.

Having consistent relationship between space and intended goals makes it easier for the player to tell when she has exhausted the possibilities in one section of the game. Some games go a step further and make it impossible to return to the hub until the section puzzles are completely solved, as in *MythTale* (Temari Seikaiha, 2002).

Limited-Access Puzzles

Another way to restrict the gameplay in a certain area is to let the player bring a limited subset of the inventory with him into some portion of the map. Perhaps he has to pass through a narrow crack, or climb up a rope, and has to leave most of his possessions behind.

These sections can provide useful focus to the player's experience, especially during a late phase of the gameplay. If the player has discovered dozens of inventory items, restricting his options to a few tools may make it easier for him to figure out which applies to the puzzle in the limited area. Similarly, limiting the amount of time spent with a complex machine can convey to the player that only a few turns of manipulation are required and that he needn't seek a 37-step solution.

Toy as Well as Game

Not every interactive thing in a game has to be a puzzle. Interactive toys can provide distraction when the player is stuck on the main puzzle arc, flesh out the narrative with hints of unexplored story, or deepen the atmosphere.

A toy can be as simple as an amusing bit of randomized content. *Curses* includes a radio that plays silly songs. *Tales of the Kissing Bandit* (J. Robinson Wheeler, 2001) includes shelves of books with funny titles. Both give the player something to look at and establish tone. A late sequence in *Zork II* gives the player a wand useful for several puzzles but that also opens a huge number of silly, useless spells that can be cast throughout the game.

The key to designing toy content is to make it highly responsive (the player can always get an interesting interaction out of it) but unnecessary (the player does not *have* to keep playing with the object). *Adventurer's*

Consumer Guide (Øyvind Thorsby, 2007) contains a “cow-o-meter” the player can point at any character to find out how many cows that character has seen in their lifetime. This information is often funny, never useful, and works on every character.

Toys are great for filling in bare spots in the map. They also do well placed near major block-points because they give the player something entertaining and stress-relieving to do when stuck—as long as it doesn’t look as though the toy is somehow part of the solution to the main puzzle.

The Fictive Content of the Map

A Coherent World

The topological organizations just discussed exist at the level of the world model and consist of the actual links between rooms. But the text of the descriptions is important as well. A memorable topology helps the player survive without notes. A well-developed landscape deepens his sense of immersion in the story.

Regions

Spellbreaker (Dave Lebling, Infocom, 1985) is divided into a series of themed regions, each associated with a different element or symbol. The overall map design is disjointed, and the player travels from one region to another by magic. Each region has its own flavor and its own set of possibilities, helping the player remember where he was relative to the larger scope of the game. The different time zones in *Jigsaw* and the varied planets in *The Legend Lives* (Dave Baggett, 1994) perform similar functions.

Dividing a large space like a city into regions can also help sell the concept that it’s a coherent place, part of a working environment with a defined economy and culture. *Blue Lacuna* gives its main island a varied geography and different landscape features and then reinforces them with atmospheric effects. Some animals and sounds appear randomly only in certain areas. Even when these creatures aren’t interactive, they reinforce the player’s sense of a cohesive map with distinct subsections.

Rivers, Coastlines, Planet Faces

Any geographical feature that runs through or beside several locations

reinforces a sense of continuous space. A coastline, a wall that runs past several locations, a cliff—all such linear features give the player a way to string rooms together and remember where he is with respect to other locations.

Graham Nelson's *The Meteor, the Stone and a Long Glass of Sherbet* (1996) places locations around the perimeter of a central cave, and then along its floor, so that the spaces in the game could be imagined as being positioned around the surface of a sphere. This shape is both novel and visually evocative, especially since the connecting route between levels is an enormous inverted tree with roots in the ceiling. *Small World* (Andrew Pontious, 1996) and *Earth and Sky 2: Another Earth, Another Sky* (Paul O'Brian, 2002) invert this effect, with maps set on the surfaces of tiny planets. The locations are more tightly connected than they could be on a planar map, and they are further organized by time zones (in the former case) or polar and tropical regions (in the latter).

Landmark Objects

In Kathleen Fischer's *Inevitable* (2003), the player's position is described with respect to an enormous ziggurat and several towers that can be seen from a long way away. The vista opens up as the player approaches these landmarks. Different areas of the map can be understood not merely as north and south of each other but—more simply—as north or south of a central point.

The player builds a hub-like conception of the game world even though he's free to move around the hub as well as towards and away from it.

Vistas

A similar trick is to let the player see from one place into others, even if he can't travel that direction. A window that shows who is in the garden below, for instance, or a telescope that shows objects in distant areas (*The Meteor, the Stone and a Long Glass of Sherbet*) help convince the player that the space he inhabits is solid and three-dimensional.

Multi-location Rooms

Many games feature locations that are all part of the same large space (such as the several-part "Hall of Mirrors" in *Enchanter*). The mnemonic effect here is to allow the player to effectively cluster several places under a single mental heading: four or five locations can all be mentally tagged

“the plaza,” with the internal relationships of “North Plaza,” “West Plaza,” etc., being obvious enough not to need special memorization.

A multi-location room can anchor a game map in much the same way that a landmark does. *Wishbringer* puts a town square (or, actually, a “rotary”) at the center of Festeron. It consists of north, south, east, west, and central locations, with one indoor location and one arterial location accessible from each side of the rotary. This area lies approximately in the middle of the game map and is the activity center of the town; the edge of the town is bordered by water, providing a continuous coastline and a tidy boundary to exploration. It is difficult to be more than a move or two away from either the rotary or the waterfront.

The People in the Map

The inhabitants of a world give it its meaning. That includes characters the player meets scattered through a landscape, sidekicks for the player, and even the protagonist himself. *Blighted Isle* (Eric Eve, 2007) populates its eponymous land with a large cast of characters all able to discuss what they’re doing here, and why; it even lets the player acquire one of three women as a companion and love interest.

When incidental characters aren’t enough, some games give the player a sidekick who comments on locations visited or acts as a tour guide. *Fine-Tuned* (Dennis Jerz, 2001) gives the player a companion who offers witty commentary and advice. *An Act of Murder* (Chris Huang, 2007) opens with a character that gives a tour of the crime scene, introducing key figures and ensuring that the player has found the key points on the map.

Viewpoint shifts are another deepening technique. *Common Ground* (Stephen Granade, 1999) and *Being Andrew Plotkin* (J. Robinson Wheeler, 2000) show the same places from the perspectives of several different characters, adding new layers of significance to the environment and defining the character personalities at the same time.

Additional viewpoint material can even be saved for second or third playthroughs, if the aim is to reward a persistent player. *Broken Legs* (Sarah Morayati, 2009) lets the player replay with extra thoughts from the protagonist. *Child’s Play* (Stephen Granade, 2006) offers a director’s commentary presented as footnotes to the main action.

At the Edge of the World

Sooner or later, however small or large your game world may be, the player will come to the edge of it. This is not a problem if you've chosen to set your game in an area with a natural border: a spaceship, an island surrounded by hundreds of miles of ocean, a cave system. A building will do, if you can give the player good enough reasons not to climb out the windows or walk out the doors.

If the setting is somewhere in the middle of an open field, however, the player may wonder why he can't wander off any old direction. We may want to give him some explanation that focuses his attention on the playable part of the map while allowing him to maintain the fiction that the game world makes sense and is not artificially structured.

Long Roads

Enchanter handles its map border by offering the player a long road in the wrong direction. Each location of the road is identical to the previous one, except that a series of taunting signs will eventually suggest that the player is headed the wrong way.

This is sufficiently frustrating that subsequent games have used the technique as a practical joke. *Annoyotron* (Ben Parrish, 1999) derives its eponymous feature from forcing the player to trudge up and down a long hallway of entirely identical rooms.

False Doors

Any barrier that can be used to block part of the map temporarily can also be used to mark a permanent edge: a locked door, a room full of poisonous gas, an impenetrable thicket. Used very carefully, a false door makes it seem as though the game environment is larger than the area the player is actually allowed to walk through.

The trick here is to communicate that this barrier is not one the player should try to open or solve. Vague descriptions are often effective: "Doors line the hallway on both sides; your own is E5, to the north." Now the author has created an environment that suggests adjoining locations, as complex as the real world, but the non-descript doors do not even have compass directions explicitly associated with them. The description signals to the player that the north door is the interactive one.

So Far (Andrew Plotkin, 1996) uses this technique well. The locked

door in the game's prologue, which obviously cannot be opened, but behind which there are speaking characters, adds a sense of breadth and habitation to the world. At the same time, the unresponsiveness of that door to interaction discourages the player from trying to reach the characters beyond—keeping the gameplay within bounds and developing the game's theme of unbridgeable separation.

Besieging Threats

One Eye Open barricades some possible corridors by making them too dangerous to move through for more than a turn or two. There are plenty of warnings (killing a player suddenly and without preparation just for entering a new area is not a very popular technique in modern IF). The player that chooses to ignore them gets what he deserves.

Now the boundaries of the world aren't signs of isolation or limited implementation but of the constant and serious threat that the player is under.

Tonally, this is the riskiest of all options for defining the edge of the world. It makes the environment a hostile place and encourages the player to think of himself as besieged in a small zone of safety. But for some games this is a desirable and compelling effect.

Trompe l'Oeil Vistas

The author can describe the view of what lies in some direction but in such a way that it's clear to the player that there's nothing interesting over there: a broad expanse of desert or wasteland, a thick forest, a trackless wilderness, or a tangle of suburban streets will all discourage the intrepid adventurer. Or tell the player how the landscape continues on the far side of a chasm or rushing river.

The point of all this is to avoid having directions where there's no described barrier, no reason to assume the presence of a building, a cave wall, or an impenetrable forest, but where the game nonetheless tells the player only, "You can't go that way." This situation leaves a strange blank in the imagined world, as though the player had looked around and seen only a grey fog to the west, the wall at the end of the world.

The Map and the Plot

If plot events are tied to locations on the map, geographical design can double as a way to structure the story and make sure the player has

learned everything he needs to know before going on to the next event.

Enforcing a Sequence

The author guards the king's audience chamber with a servant who must be bribed; he places the bag of gold in the guest bedroom so that the player is guaranteed to explore that area before proceeding; in order to reach it, the player must pass through the hallway, where the palace vizier is wandering around. The author can now plot accurately (enough) the path the player will take through the space and guarantee that he will meet the vizier before he meets the king.

Many games use a sequence like this to guarantee that the player learns one piece of critical information before going on to the next plot element.

Moving Actors

Christminster (Gareth Rees, 1995) traps the player character the first time she enters a specific area: some hostile characters enter the room. While the protagonist hides, she has the chance to overhear a conversation and gain valuable plot information.

This method is a little more sophisticated than a simple puzzle sequence. It assigns motives and intention to the non-player characters. Because the player encounters the characters and their related scene on the way *out* of a space rather than on the way *in*, Rees disrupts the sense that the game is a static mechanism for doling out plot. Nonetheless, the trigger for this plot-advancing scene is still the movement of the player.

Magician's Choice

Sometimes an author wants to give the player the experience of freedom—wandering in any direction he likes, exploring the map in any order—but there's a significant risk that doing so will damage the pacing of the plot. In this case, it's possible to cheat.

Photopia gives the player freedom to explore the surface of Mars, but no matter which directions the player tries, he'll always arrive at the same locations: the map is not actually fixed until the player tries to traverse it. Other games supply content when the player has explored a certain portion of the map or collected some percentage of key objects.

These strategies tend to emphasize *pacing* as much as or more than information; what matters to the story is the experience of looking for something, rather than what and how much the player happens to find.

Beyond Layout

Many considerations that go into a good setting are beyond the scope of this article. Strong world design and room description writing helps build a persuasive fictional environment. Well-paced puzzle design makes the block points of a map interesting rather than annoying. Nuanced viewpoint writing makes a story world feel lived in and narratively meaningful.

Good map structure is important, but it is only a foundation for the features that make IF fun to play.

Thinking Into the Box

On the Use and Deployment of Puzzles

Jon Ingold

Introduction

Since I started writing interactive fiction I've spent a fair amount of time not just creating it but trying to introduce and explain it to other people. The synopsis I give always begins the same way: "You remember those gamebooks where you choose which paragraph to read?"

To most IF devotees (including myself) such a description seems inadequate, not to mention toe-curling. It is like comparing hopscotch with football or "find the lady" with poker. A gamebook is a diversion, we might say, but a good IF game is alive.

But why? A mathematician (again, such as myself) might say that a parser-based IF game is just a gamebook with a very wide parameter space, and a lot of very similar-looking paragraphs that say things like "You smell nothing unexpected." Except, of course, that the options in an IF work are not placed on display. Is there something about this user-interface quirk that gives IF its curious sense of immersion?

A Postulate

I would argue that IF is made what it is by the prompt—that blank chevron that waits, blinking and mocking, for the player to have an idea. You can interact with it without thinking, of course: you can type LOOK or INVENTORY, and indeed do so repeatedly. But to make any progress in the game-world, and in its story, the player will be required at some point to think of something to do; something constrained both by the set-up of the game-world and the vocabulary of the parser. I would postulate that it is by this process—in which a player is forced to think in the context of the world, and translate his ideas into a form that can be understood—that the sense of immersion or "mimesis" (that in any other form of fiction would be called "suspension of disbelief") is generated.

What separates IF from a fixed-choice gamebook then is not the presence of choice and the ability to decide, but the *requirement* to consider that decision first. A player must read the text, closely, to pick up the relevant nouns; she must also picture that world and perhaps imagine herself as the protagonist in order to determine a good next move. In so doing she places herself into the story, losing a little of the chair beneath her and the screen in front. In short, I would suggest that when players consider their options they are also quite inadvertently imbuing them with meaning.

The Role of Puzzles

So how do we, the writers and designers of an IF game, encourage the players to make this effort? Graham Nelson wrote in his seminal essay “The Craft of Adventure” (1995): “The author of a text adventure has to be schizophrenic in a way that the author of a novel does not.” That is, when designing a text game it pays to think both as an all-knowing writer/designer and as an unknowing player. As designers, we try to ensure our games are well-balanced, well-developed, and filled with interesting and intriguing interactions. As players, we want a game to listen to our ideas and respect our attempts at progression, without blocking us with unreasonable or impenetrable obstacles.

Taking this view, a puzzle—a sticking point—becomes the point at which both sides meet head-on. The designer is offering the player the chance to have an impact on the game-world and to open up the story through their ingenuity; the player is speaking up with suggestions and risking wasted effort and defeat. Both hope a solution will be found so the game can progress, but neither wants a solution that is too easily won.

What Makes a Puzzle?

Here we ought to define a “puzzle.” A Tower of Hanoi is a puzzle; so is a maze. So is a code, or a complicated system of pulleys. But a puzzle does not have to be a black-box entity whose sole reason for existence is the bafflement of others. A puzzle can arise from any situation or scenario in which a productive next step is not immediately apparent. Interaction with another game character might qualify, from the standard (feeding the animal with food, so that it doesn’t notice you leaving) to the more cunning (upsetting the grandmother with a photograph of her

late husband, so that she doesn't notice you leaving). A closed, unlocked, well-hinged door, in a game that understands the verb OPEN, is not a puzzle. The same door, if found in Carl Muckenhoupt's *The Gostak* (2001), is a puzzle—because that game is entirely in an invented language, and working out the word for OPEN (and indeed, deducing the presence of a door) is extremely difficult indeed.

A puzzle, then, is anything outside of the basic interaction set of the game that generates progress within the game. This leaves “puzzle” as an unhelpfully wide category to design within. We now proceed to look more closely.

Thinking Like a Player

Magicians have long known that the way to cover a trick is with careful patter, distracting and involving the audience while simultaneously frantically rooting around in a pocket for the right playing card. Similarly, they've known that the easiest way to invent a new trick isn't to invent a new mathematical device, or a new black-velvet-box-with-string, but simply to dress up an old trick in different patter. An audience with no knowledge of magic may well never realize that five consecutive cards tricks are all the same trick wearing different clothes.

IF writers are not magicians: our players are. Our players are confronted by the impossible, obstacles with no apparent resolution, and then overcome them through cunning, wit, and logic. If we are thinking like players, then, we need to think about our puzzles in terms of the underlying mechanics required in finding their solutions.

So, for example, the problem of finding a gold key for a gold door is then the same puzzle as finding cheese for a hungry mouse—both rely on an obvious connection and on the finding of an object that the player will guess is present even before she sees it. On the other hand, finding a full-length mirror behind which to hide when slipping past a vampire is a very different puzzle indeed.

Hunter-Gathering

The most primitive mode of puzzle-solving is that of finding. Every puzzle game ever written has had a fork under a table, a diary under a bed, or a handkerchief in a handbag. Generally speaking, authors of games don't like to leave every important object in plain view, just lying

around; they prefer instead to stack them tidily away. Before this was easily possible (in the early Cambridge games published by Acornsoft and Topologika, for instance, which lacked the standard model of hierarchical containment now included in every IF authoring language), they would still trouble to hide things in convoluted maps (the woods at the start of *Colossal Cave* (Will Crowther and Don Woods, 1976)) or full-blown mazes (*Crobo* (Jonathan Partington, 1986), *Fyleet* (Jonathan Partington, 1985), etc.).

Does this wandering around and looking underneath things really constitute a puzzle? Certainly it does, if finding the item will further the player's progress. This may be presented to the player as a lack ("The old door has a key, somewhere") or as an implied lack (on a scale from "Maybe I can jimmy the lock with something" through to "I seem to be stuck; maybe I've missed something.") And it's a useful kind of puzzle, too, as it allows us as designers to pace out the player's discoveries and knowledge of the world they're in through the balance of what's easily visible and what's hidden.

A player, trapped into a small enough set of locations, will read every description twice and take his inventory four times. By hiding things (say, the key to leave the first room of the game), you force the player to familiarize himself with whatever information is available (say, his identity and his initial inventory items). You can be reasonably sure that a player who has hunted every nook and cranny for his sister's diary will have found the hairpin with which to pick open its lock and, indeed, may even have already put on the reading glasses he's carrying before the tome is open. Further, if later on he finds an ice cube containing something he needs (a particle of guilty conscience, perhaps, in the style of Meretzky and Adams's *The Hitchhiker's Guide to the Galaxy* (Infocom, 1984)), then you can reliably expect him to think very quickly of the blow-dryer on the dresser, just by the bottles of spot-cream.

And just as one room or one collection of objects can be made familiar, so can the game map itself; if you force the poor player to traipse backwards and forwards for a while, then in later scenes of the game it will be quite alright to require him to trace a route with step-perfect accuracy. If your game is going to have a car-chase later on, it seems only fair to allow him to learn the street map, and since your player may not realize this will be important, it seems only fair to force him to do so. On an immersion level, a good use of searching rewards

the player with a wider vista, a deeper visual model, and the certainty of where to find a telephone. A really good use will do so without the player noticing.

Old-school players of old-school IF probably find themselves sidestepping all this, by habitually ransacking whatever location they're in before stopping to consider the situation. It's a brute-force solution, much as trying every combination on a combination lock would be. In this mode, players make efficient progress but lose the simple interactive joy of needing something and finding it somewhere appropriate (a diary under a pillow or the spare key in the flowerpot).

Brute-force solutions are the bane of a designer's life, and in the case of a searching puzzle they can be difficult to guard against. You could implement non-fixed solutions, for example: Plotkin's *Shade* (2000) opens with a search for plane tickets, which will always be in the last place of three you look—enough to achieve its effect (a bit of irritation, a bit of familiarization, and a lot of contextualization ready to be attacked in the game's later phases)—but not enough to outstay its welcome. The benefit of such an approach is you can control how long the player must hunt; the cost is that the finding will be less satisfying (the plane tickets showing up in the paper recycling, for instance).

It's worth noting that many computer games use hunter-gathering as a primary mechanic: in RPGs they're called "fetch quests," and in platform games they involve collecting keys. While useful and satisfying enough, they are hardly IF-specific, and any IF game that relied solely on them would probably be underusing the potential of a parser-based game.

Research

Probably the next most common puzzle after a hidden key is a combination lock: a hidden key where the key is non-physical and stored in the player's head rather than his hands. The canonical example of a research puzzle is a magic potion, where the right books and cross-references yield a list of random objects that can be boiled up to create infinite youth (*Christminster*, Rees, 1995), deep sleep (*The Mulldoon Legacy*, Ingold, 1999), or violent explosions (*Trinity*, Moriarty, Infocom, 1986). This is a much cleaner and more efficient (and, perhaps, more prone to error) type of hunting puzzle, but for a player it feels quite different: they are being asked to do something actively (remember), and they may need to judge and sift information to determine relevant facts. They are

working, outside of the command interface, to put together a plan. They will predict the next steps of the game (“I bet I’ll have to get that jewel from the dragon’s cave!”). More likely than not, the research puzzle will lead into the player setting herself a series of hunter-gathering puzzles, allowing the designer to sit back and watch as, for once, the player pushes herself onward.

For a designer, the research puzzle also offers a perfect opportunity to sit the player down and tell her a story, and furthermore, to force her to listen—or better yet, force her to ask. The barred palace door will act as an infallible ice-breaker with the palace guard (who may prove to be a fascinating, deep, and humorous fellow). The need for a password to begin that conversation will have the player happily reading the history of the Empire in the British Library. Along the way she may need to understand the crazy indexing system—but no doubt there’s a book that will take her through it. And if that book drops information on the weird old man who built the library, his alchemical secrets, and the tragic fate of his sister, then so much the better. Indeed, it could be argued that this kind of retrospective, detective-like story-telling is what interactive fiction does best. *Curses* (Nelson, 1994) constructs most of its story in footnotes, in the form of biographies and diary entries. (And does so with consummate skill: for example, the “Vol. II” label on the book “History of the Meldrews” cannot help but raise a smile in anyone who has played the game.)

Research-based puzzles tend to chain together nicely, so a player will begin reading one book in a library, only to be led to look up another and another, using keywords located along the way as “combinations” to unlock new information. Sometimes the keywords are clearly stated, and sometimes they will involve some intuitive leaps—the finest examples of which are all in *Curses*, such as the romance novel puzzle or the prayer book. Sometimes they will form a line of stepping stones, leading linearly from start to finish, and sometimes they will involve choices and tangents and sidelines of information—a maze of information for the player to navigate through until the key result is unearthed.

A good research puzzle can be hugely rewarding for a player, both in terms of plot-progression and immersion, and sheer crossword-puzzle entertainment; for a designer it gives you license to push your fiction further into the player’s mind. Interestingly, the CONSULT verb appears to be going out of fashion, in favor of the more direct but more difficult to code ASK ABOUT, but the effect is still much the same.

Simulation

Simulation is an inherent feature of the IF medium, and every game features basic simulations of things such as space, containment, and light/dark, even if they don't choose to employ them. Simulation is a feature of building things with a computer: by simulating, an author saves the trouble of having to set up a large set of repetitive rules when just a few general ones will do.

A simulation puzzle is one where the player is asked to take a similar approach—to learn a rule in one situation (either in the game or in the real world) and apply it to a different but comparable situation. To give a simple example, if the player learns that the in-game microwave-like object makes water hot, they might expect to be able to use it to make a cup of tea, or to melt the ice cube mentioned earlier. A player might even expect to be able to blow the microwave up by putting a fork inside and turning it on. If this was used as a way to distract the guards, by taking the microwave to the museum, this would be a simulation puzzle: the explosion made in the kitchen also works in the museum foyer.

Simulation puzzles tend to have multiple steps, be about changing the state of things, and take the form of strategy puzzles—in which the result and process are clear, but the ordering of steps becomes important. The classic simulated problem is the Tower of Hanoi (the irritating children's toy involving stacked discs on pegs)—where the goal is stated (move stack from peg A to peg B), the governing mechanics are clear (one disc moved at a time, larger discs always below smaller); but the problem is to apply the correct move at each stage. Chess problems are a similar idea, though much more complex in nature.

IF takes this concept and codifies it into the manipulations of containment, visibility, and gravity; also magic and strange machines. *Metamorphoses* (Short, 2000) simulates a variety of substances and the ability to morph objects between them, and then has puzzles that require fragility or density or flammability in the object produced. A lot of games also like to simulate time—*A Change in the Weather* (Plotkin, 1995) notably simulates a thunderstorm into an extremely evil timing puzzle.

Cadre's *Varicella* (1999) takes this idea to an extreme, in a game in which the simulated actions of several people in a palace are tightly interwoven, with the player attempting to thread a manipulation of his own through the mesh. This makes for a difficult game to play, partly because people are complicated and there are lots of ways to engage

with them, partly because people do things when you're not looking, making them hard to track, and partly because people are nothing like the people simulated by a game. This highlights a problem inherent in simulation puzzles: finding the boundaries.

The player entering an IF game will start with some hazy expectations of what is simulated and what is not. She will expect, for example, to be able to carry small things around with her, though she may not be sure whether she can carry an unlimited number, or an unlimited weight. She will most likely not expect to be told which hand she's carrying things in, or which pocket, and will probably become annoyed if told her hands are too full to push door handles if she's carrying two items (even if this may be true in real life). To say the simulation should be as accurate as possible is untrue; players automatically expect a level of simplification on entering the text game world. No novice will expect to have to type BLINK every four moves. However, they may be (and, I find, often are) surprised to find WALK OVER TO DESK is not understood. In conversation with game characters, players will not bat an eyelid (i.e., blink) if asking the same question elicits a word-for-word repeat of the response. Players will not complain when a poison gas puzzle cannot be solved by typing HOLD MY BREATH (there being no "breath" object in the game-world); indeed, if this is the only solution, I would expect most players would be irate and demand to know how they were supposed to think of something so outlandish.

Simulation puzzles require careful tutelage. If a puzzle relies on the player knowing that waving a gun around will make characters react, then you'd better show him early on that characters can see what he's waving. Further, they'd better react to everything else as well—no good them screaming at the sight of a bloodied murder weapon if they don't notice their wife who's been killed by it.

The two important things then are consistency and pacing. Consistency consolidates the player's world-model: if it breaks in a few places most players will believe they've encountered a bug; if it breaks in almost every place then most players won't believe it's simulated at all and will not expect it work elsewhere. If you want the player to know how your game-world operates (what liquids can do, what fire can do, and so forth) then you must be prepared to make sure it does everything one would expect. This doesn't mean every fire has to potentially devour the entire game-map, but it does mean every sheet of paper stuck into it

won't be described as "Unlikely to do anything." "Everything one would expect" means "Everything one would expect on the basis of what has already occurred," not "on the basis of the real world." The real world is neither here nor there.

Pacing is the more difficult point. To take an example from outside the IF world, the original *Tomb Raider* games, published by Eidos, introduced the various jumps, rolls, and runs the heroine could perform through a series of obvious challenges in a "exercise course" level. This served to allow the player to practice, not just the act of performing a maneuver (which is not relevant in a text game, unless it is a typing tutor), but also the process of gauging distances and timings. Similarly, an IF game that extends the simulation should give the player a series of simple manipulations to perform so that they learn how the simulation works in various conditions: how liquids respond to heat, whether some objects float, if flame spreads to nearby objects.

The principle is that of Chekov's gun: if the player must produce a noisy distraction in the endgame, then make sure, earlier on, he inadvertently knocked over a vase and brought the housemaid running. If a character will reveal more information on repeated interrogation (maybe if the player now knows more about the topic himself and so "phrases the question differently"), then make sure the player is forced to ask the same question repeatedly at some point, even if you employ a simple, pointed "Perhaps he's not letting on all that he knows," at the end of his first surly reply.

A well-designed set of simulation puzzles starts with those that shift the player's expectations of what the game will understand, then consolidates it, and then relies on it for more and more complex puzzles exploiting the ways the various features interact. The characters in *Christminster* work on this principle, with the player finding more and different ways throughout to cajole Edward into helping her and to use his various skills (or lack thereof) to further the game.

An interesting branch of simulation is when the player is considering the game itself and the rule is that "there must be something to do." An excellent example of this is the award-winning puzzle from *Spider and Web* (Plotkin, 1998) concerned with escaping from the chair. The player will not solve this by thought—through cause and effect, simply because he does not have enough information—but rather by looking at the only options available to him, and the knowledge that one must be right. Similarly, a player may look at a location or item and deduce that it must

be useful for something (for example, the hidden room in the maze in *Curses*).

It should be noted that requiring a solution by this method can have a negative impact on immersion—a player looking at the design of the game, and at available information, is not a player emoting. In a game such as *Spider and Web*, which places a distance between player and player character, this isn't too much of a problem, and in wider, puzzle-oriented games, it will not matter either. Games in between should be careful, though, as otherwise they risk giving their protagonists something verging on precognition.

If the simulated aspect of a game is well-executed it will both extend the design possibilities in unique ways and allow for interesting new puzzles (the *Enchanter* trilogy of Infocom introduced magic spells to great effect), but it will also increase the player's confidence in the author, allowing for more and more ambitious leaps of logic to be expected as the game continues. If the game remains fair, then players will expect it to remain so and be willing to give you the benefit of the doubt on later, harder problems.

This leads finally to a branch of problems that can only be described as “beta-tester puzzles,” being those that the player solves less from a desire to overcome an obstacle and more from a desire to break the game and come out with a permutation that has not been accounted for. A good example is the robot cleaning mouse in *Starcross* (Lebling, Infocom, 1982), which disappears into a service hatch at various intervals, leaving the curious player to wonder if the room beyond is actually implemented or not and to devise a nefarious method of entry. These puzzles can be some of the most satisfying to solve, as the player will feel quite certain before trying his solution that it should work, but they can also be some of the hardest, because there is little or no direct indication in the game that there is a challenge to be attempted at all.

Intuition

“Beta-tester puzzles,” like the “leap of research” puzzles mentioned previously, are both examples of a more general class: intuitive puzzles. The definition of an intuitive puzzle is essentially “an unfair one,” as these are the problems in which the player is expected to make a mental leap on the basis of little logic and very few, usually obscure clues. They are the black magic of puzzle design, game-quitting failures when they miss and immersion-creating moments of true connection when they

work. There are a hundred examples, and whether they work or not often depends on the individual player and how receptive she is to the subtler hints of the author's general mode of thinking.

Intuitive puzzles are not always extensions of other puzzle types. A concrete example of a puzzle uniquely in this category is the chipped square in Plotkin's *So Far* (1996). Here the player is given an alien object in its natural setting with a few clues to its function that are only really comprehensible in retrospect, and is not even asked to deduce its operation. It's a pretty simple object, and a one-word solution, but the important point is that, once solved, the square's minimal decorations and its notch are transmuted from opacity to transparency in, literally, a flash.

The beauty of this kind of solution is that the player *does not need to be playing the game to check the answer*. A good intuitive puzzle can be solved while shopping or at work; once seen, the answer illuminates all the clues that have preceded it. The player's only job then is to check that the game matches their expectations. Intuitive puzzles are great hooks for keeping players coming back to your game. (It was while waiting for a number 42W bus that I worked out what the soft grey brick in *Zork II* (Lebling and Blank, Infocom, 1982) was for, after a three-year break from the game, and went on to complete it).

Hints are crucial to intuitive puzzles; the best are those a player only realizes are hints once the puzzle is solved and yet somehow plant the seed of solution in the player's mind. They can be placed in a thousand ways—and perhaps should be in as many as possible—though elegance is an obvious plus. *So Far* is excellent in its hinting, simply by constructing a logic behind every puzzle but refusing to present anything but the elements at face value, leaving the player to construct every implication. Done once, this would be infuriating, but the game plays this hand consistently: the puzzles in *So Far* share a thematic feel (beyond simply being really rather hard)—as do the puzzles in the author's earlier *A Change in the Weather*. Fans of Infocom's David Lebling admire his tight logic and scientific use of cause and effect in games like *Starcross*, *Spellbreaker* (1985), and *Zork III* (1982).

The leaps of intuition made by a successful player have a heavy payoff, as suddenly, instead of the player having to read the mind of the author, it seems the author has read the mind of the player. (This is of course not the case; in fact, exactly the former has occurred, but it's a lovely effect to generate). By tuning oneself into the author's mindset the

player becomes more deeply immersed in the patterns of game-world and story, and by allowing himself to believe in (the existence of) an underlying logic, the world develops a continuity outside of the player's interactions and feels all the more real. These are the puzzles that remove the game as far from a game-book as is possible—when the action to be taken must be constructed entirely inside the player's mind before it can be applied to the game itself.

Often, the very best intuitive puzzles are easy ones—where a player walks into an area and immediately types the correct input, without really knowing how. Many players found the puzzle with the hatchway in *Spider and Web* to work this way (though I spent about half an hour with permutations of boxes and wrenches, far overcomplicating the problem). Similarly, I think the best intuitive puzzle I have ever written myself is at the start of *All Roads* (2001), in which the player's hands are tied with rope and the only available object is a bottle of wine. The solution is no doubt quite clear—but to produce it the player must take the phrase “bottle of wine” and apply his knowledge of what bottles of wine are really like. To do this she must fully visualize the object, and this situation is elevated from being an inventory listing:

You are carrying:
a bottle of wine

to a character, holding a bottle between loose and shaking fingers. (And note, this isn't really a simulation puzzle, because the same mechanics are never used again.)

Intuitive puzzles are, in the end, the bar by which all puzzles are judged, and the target all puzzles should aim for. They are the moments in IF when the player and designer feel as one: when the game feels as it should, and the plot progresses in leaps and bounds. They are also the single most likely thing to make players stop playing. No-one said it would be easy.

Thinking Like an Author

Just as there is an art to designing puzzles that will linger in a player's mind, there is also an art to codifying them into ropes and rabbits and rodeo-machines and placing them in the right order to achieve your effect. This is where the authorial mindset comes in—in planning the

structure and plotting the order by which you unfold your story and your world. Here the game's status as narrative and crossword becomes quite pointed, as puzzles can be deployed to aid either or both. The following sections detail the most important areas of influence.

Plot

The most obvious use of puzzles is to slow down plot development and allow interaction to break up the text from a single chunk into a series of responses. In an old AGT game *Murder at the Folkestone Inn* (Anne Laughlin, 1993), the plot occurs entirely in room descriptions, and the player simply moves from one to the next to experience the story. Most games are more sophisticated than this, but the basic idea is perhaps the same. Puzzles can be used as a device to bullet-point the plot and also have the story react to the player's presence in some way (fixed or flexible, it doesn't matter).

Gareth Rees remarked in an article on the design of *Christminster* that the motive for one puzzle—in which the player is trapped in a small room with Professor Wilderspin—is to force the player to talk to him, even though this is not important to the solution of the problem. This is an example of another trick—using a puzzle to constrain the player's ability to wander away from something they should be paying attention to. This can be just an extension of the research puzzle idea; just as you can bury a password in a family history book, you can bury a key in a family graveyard, and either way the player must scavenge through the past if he wants to continue, collecting information along the way. It can also be more active, having the player confined to an area in which events are occurring or, indeed, having the player's solution to a problem affect the plot in some way (taking care to ensure the outcome does not feel too arbitrary, otherwise the player may start taking his decisions without any thought, just as he does in a gamebook). There will come a point in the game, of course, when the plot is interesting enough for its own sake that players will be happily rooted to the spot—however, if they are not, providing them with a point of interest, a point of interaction on which to center, is a clear way of drawing attention to anything important around. In a way, this is the hunting idea again, only now the search the player does not realize he is on is for the story-line itself.

Interactive fiction needs this sort of device to pace it, due to the inherent lack of structure arising from the flexible pace of the player's

progress. More cunning ways are available, not using puzzles—a game could keep track of how well a player was doing and advance the plot if he is failing to do so, for instance. However, puzzles are probably the easiest and most widely understood mechanism to date.

Finally, puzzles can form the very stuff of your story—making them attractive to the plot-hungry player and decisive in your game design. The chair problem of *Spider and Web* is a classic example, being a moment where the plot moves round and become something quite unexpected. Placing this on the end of a puzzle allows the player to feel that it's her achievement, that it's her work that has produced the outcome—and makes the player properly receptive for the moment, when it finally comes. She will be reading more closely after solving a problem than after simply exploring around.

Geography

A player mapping a game uses the same convention that authors do; no open hallways or expanses of grass and trees sketched in delicate pastel shades, but rather boxes on a grid with lines for doors, and the occasional frown of irritation when presented with an up/down connection. Each location is marked as a separate object and linked by walls with teleporters in them, that could be vertical, horizontal, widdershins, or inside-out for all a player may care. There are ways to try to alleviate this—designing maps that fit neatly together, or implementing larger rooms in two or more sections to give the sensation of space. But the real problem is the lack of connectivity between discrete locations, and this is best overcome by implementing something that causes the rooms to bond. A game may include, for instance, a roaming character who goes from room to room (such as the thief from *Zork*), but this sort of approach is not tailored towards the player's own actions and so feels less relevant. (Indeed, the thief never made *Zork* feel like a real world, being more just a source of irritation for mappers, treasurer-seekers, and anyone trying to avoid the grue).

The solution once again comes down to thought—to placing the idea of connectivity directly into your player's mind. If he is told—by the entrance of a thief, or the presence of a river—that rooms are connected, he will draw a line on his box-grid map as required. If, however, he is required to realize that the solution of some puzzle relies on a continuity between locations (floating something down the river, or setting a trail of gold coins for the thief), then the act of thinking out

this connection will realize it in the player's mind. It's the same idea again—that forcing a player to visualize rather than catalog the text game world will increase his immersion within it. So, by the construction of puzzles that exploit geographical links, chutes from floor to ceiling, windows looking out over other rooms, pulleys down well-shafts or attached to mine-buckets, we glue a map together. Indeed, Nelson wrote in “The Craft of Adventure” that a good game map has a river running through it, and this is true—but note your river could equally well be a conveyor belt, a dumb-waiter, or anything else that allows what happens in one location to affect another.

This also allows for pleasingly devious puzzles. One could have a search for a secret door motivated by a “hole” in the game map, which then requires the player to deduce which room is above the missing space and work his way down. A time-travel game could allow the player's knowledge of the future map to influence his explorations in the past. A game based on C. S. Lewis's *The Silver Chair* could have players exploring a region that, when viewed from above, spells out a message. And so forth.

Hooks

Lemmings (DMA Design, 1991) is one of the best-designed games of all time and is also one of the most addictive. Why? Because the basic problem is so easily remembered, and so frustratingly tricky, that it sticks in the head and refuses to leave. So too the best IF puzzles, which keep players wanting to play. The *Lemmings* addiction stems from the desire to make “just one more attempt”—in the text game set-up this corresponds to problems involving complex machinery and with rich feedback and a lot to fiddle with (such as the machines in *Metamorphoses*).

You can hook the player whilst no longer at the keyboard, too, with a different kind of problem. If the goal is clear and the set-up quickly committed to memory, players can be inadvertently solving—and failing to solve—your problem as they go about their lives. To construct a good hook, you need to ensure your puzzle doesn't require too much fiddling about, or experimentation; the solution will almost certainly be intuitive. Mazes do not fit this category. Perhaps the classic fox-goose-grain-and-boat problem does—but everyone learns how to solve that at primary school.

A good type of puzzle for the player to carry around is the “unfamiliar object” problem—such as the chipped square in *So Far* or

the grey brick in *Zork II*—which requires the player to make a deduction from a small amount of very relevant information, and possibly through connections with other things seen before. However, for a puzzle of this type to truly hook the player, your object needs to be intriguing, or the world it is in must be. The mundane is no good for a hook.

Sheer Fun

The Hitchhiker's Guide to the Galaxy, written by Steve Meretzky and Douglas Adams in 1984, contains numerous puzzles whose sole purpose in the game is that they're fun. Not fun-to-solve in the brain-bashing sense, but silly, wacky, and downright weird. The experience of solving the best is akin to producing the punchline of a joke. The puzzle of Marvin's door, involving as it does no tea and tea, is a triumph.

Infocom's games were good for this sort of pay-off. Meretzky's masterpiece *Leather Goddesses of Phobos* (1986) contains the infamous T-remover, capable of producing rabbis and unangling cream, as well as the black paint with which to turn a white spot into a fully function teleportational black spot. More recently, Neil deMause's *The Frenetic Five vs. Mr. Redundancy Man* (1999) had a side-splitting climactic puzzle.

Writers have long known that punctuating a piece with humor is an excellent way to avoid the tedium of a single atmosphere or situation and to lessen the sensation of pretentiousness that monotonic works can quickly attain. Comedy generally makes the reader more amenable to the writer, and the occasional well-placed witty puzzle will refresh your player a little and prevent them from becoming too bored or restless. Which bring us to the subject of:

Rewards

We note, to begin with, that puzzles are often hard to solve.

Playing *Curses*, I found one of its weaker moments to be just after solving the problem of the garden's hedge maze. This is one of the cleverest puzzles the game offers, requiring a moment of inspiration followed by a long sequence of moves in which to implement it. Upon returning to the site of the maze, once successful, a new location has opened up—but this is not enough reward in itself, because the location is not very interesting and there is very little in it, just one object, of no use until very near the end of the game (the stone rose), and another locked door. This anticlimax was partly down to the game's width—the

author had no way of knowing at what point in solving all the other, unrelated, puzzles I would be—but also partly to the assumption that the player is grateful for any new object. This is not true. Rather, the player—like the reader of a novel—is rewarded by those things that spark his imagination. It could be a beautifully rendered location (though this may be unlikely in IF, as players are not often in it for the prose), an empathetic moment, a twist of plot—but equally it could be a fresh idea on a thorny problem, a ray of hope, a new object with which to toy in interesting ways and experiment. A player of a game with puzzles follows both the narrative and the progress of the puzzle structure—progress in either is rewarding, but progress in neither is not. A new location in itself progresses neither, an object with no properties of intrinsic interest (such as the stone rose) similarly, and the promise of a new puzzle that the player has no idea how to solve or even start is perhaps actively demoralizing.

The problem could be alleviated perhaps by a more vigilant game that cheats to overcome the width problem, providing different objects upon the solution of problems depending on where in the game the player is. In *The Mulldoon Legacy* there are four “key” objects to be collected and then used in sequence, but since the methods of acquiring each are somewhat unrelated, the game is careful to provide the next useful key on solution of any of the four problems: thereby each time direct progress is made. A similar device could be employed more subtly—consider a magic-based game in which the spell received on a spell scroll is tailored to where you are stuck, or a game with money in which the level of pay received is measured by what you next need to buy. Rewarding the player is one aspect of design that it is vital to get right, and there is absolutely no reason not to cheat, if it furthers your goal as an entertainer.

Conclusion

The principles and mechanisms outlined above do not apply solely to puzzle-oriented games, such as the much-quoted *Curses* and *Christminster*. Rather they are general principles relating to the way players experience the interaction element of a game. As soon as a player is expected to think about the effect of their actions and manipulations of the game-world, then the game is asking for above basic input, and there needs to

be a reward accordingly. Interactive fiction works are not game-books; they cannot be plotted out as simple finite trees that a computer could step through in optimum time. But neither are they unaware of their status as simulations and approximations of both worlds and stories. The balance between the functional and artistic qualities of an IF work is managed by the player's thought process within the game-world, and so it is the cultivation and guidance of that thought process that is the mainstay of good design. By asking the player to think on what he has learned, you cause the player to paint between fragments of information and create a continuity and a picture of his own.

By being aware of the design decisions taken, the path a player will (most probably) take, and the way their expectations of the world change as they take it, an author can make the game more accessible, more rewarding, and more complete than by simply placing ideas into an arena and letting the player walk around them, poking.

Finally

To follow T. H. White:

“What makes a good puzzle?” asked the King.

Merlyn chewed his beard a moment and frowned, before finally announcing:

“To make a good puzzle you need some sort of simulation.

Around the simulation you need to build feedback-rich experimentation.

Alongside this experimentation, you need complementary information buried in the game world.

Riddled throughout the information should be gaps, patterns, and inferences for the player to speculate upon.

Amidst the speculation you should give a suggestion of the author's mindset.

Whilst communicating that mindset, you must be mindful of the rewards of the eventual solution.

For the solution you should have a cunning idea.”

Arthur waited a moment to be sure the wizard had finished, then asked:

“Do any but the last have any relevance at all?”

Merlyn shook his head.

“None whatsoever.”

PC Personality and Motivations

Duncan Stevens

There's a fundamental decision that should be made early in the design of any IF: does the player character (PC) have a personality of his or her own, or not? By that, I mean not simply that the PC has a name and a background, though that should be a start; the *Zork* model, where the PC has no discernible persona, is widely viewed as inadequate. The question is really whether the PC has ideas and goals that are distinct from those that the player brought into the game. Since, ordinarily, the goals that the player brings in are quite simple, and look something like "see all the text and surmount all the obstacles," it might seem that even the most bare-bones PC will have his or her own objectives.

In practice, though, that's not the case. More often, in classic IF, the PC's motivations supply an initial premise, or perhaps give the player an overall goal (one that, perhaps, explains why the Magic MacGuffin acquired at the end has such fascination for the PC)—and at that point the author shoves the player into the game and commands him to start solving puzzles. What follows, in most cases, can be viewed as driven by the PC's motivations only if the PC is a cross between a magpie and a *Games* magazine subscriber; most puzzle-driven games depend so much on the player picking up everything that isn't nailed down, and viewing every locked door as a personal affront, that the only thought that could conceivably animate the PC throughout is "you never know." In a small but growing minority of games, however, the entire course of the story, and everything the player does, can be tied directly to the PC's motivations. This is, I argue, an encouraging trend: many of the most interesting experiments in IF have come about because the author took the trouble to convey, and encourage the player to understand and embrace (for purposes of the game), the motivations of the PC. Further development of that trend, one might logically conjecture, could produce even more interesting experiments.

A Brief History of PC Development

Infocom

(This section contains major spoilers for Beyond Zork, Planetfall, and Plundered Hearts and minor spoilers for The Lurking Horror.)

What I have described above as the old paradigm of PC characterization will be readily recognized by most IF veterans, but some examples from the days of Infocom may illustrate the point. In *Zork I* (Marc Blank and Dave Lebling, Infocom, 1980), the PC has no name, no distinguishing characteristics, and, seemingly, no interest in anything but treasure. (Even that is debatable; nothing in the game indicates that your character is genuinely excited by all this treasure-hunting. It could plausibly be argued that your character secretly longs only for a nice snack, or a good nap, and that the heartless player is dragging him/her around the map against his/her will. The point is that there isn't much evidence either way; the game never stops to say, "You tremble with avarice at the sight of the jeweled trident," or any such thing.) In *Beyond Zork* (Brian Moriarty, Infocom, 1987), the PC has an initial goal—finding the Coconut of Quendor. (For the uninitiated: yes, really.) The player is told initially that the Coconut of Quendor resides with the Implementors (the Infocom authors themselves, who regularly made cameo appearances in their games), so the PC's initial goal is presumably to find the Implementors, but in practice the puzzles that the player solves to reach that point have no foreseeable connection to that goal. (They include such things as solving a riddle written on the side of a cliff and searching a cellar for a wine bottle in order to obtain a giant onion.) When the player finds the Implementors, the coconut is snatched away by a nefarious presence, and the goal becomes finding that nefarious presence. (In fact, the Implementors explicitly state as much, lest the player think the game is over.) The same process follows: the player solves a long string of puzzles, and for only a few of them (the last few) could it be said that solving the puzzles has a foreseeable connection to tracking down the enemy. The PC's motivation, therefore, serves only as a framing device—or, less charitably, an excuse for the puzzles.

In fact, that was true for most of Infocom's games, though to varying degrees. The fantasy and science-fiction games were more recognizably a set of puzzles with little direct relation to the overall

objective, except perhaps a thematic relation (most notably in the cases of *Trinity* (Brian Moriarty, Infocom, 1986) and *Zork III* (Dave Lebling and Marc Blank, Infocom, 1982)). Gathering a set of objects—whether treasures or some sort of recipe for a desired concoction—was particularly popular, since it allowed the PC to indulge his/her magpie impulse, picking up everything in sight, and to regard every barrier as a challenge on the theory that you never know where those objects might hide.

Planetfall (Steve Meretzky, Infocom, 1983) is a particularly striking example of Infocom's general inattention to motivation, as the game's premise (you escape a doomed space station and crash-land on a seemingly deserted planet) seems to suggest a variety of actions (calling for help, say, or finding a way off the planet)—and the game suggests, through various red herrings, that those obvious actions are possible when in fact they are not. The real goal, suffice it to say, is wholly unguessable, and the protagonist arguably seeks out the climactic location only because some monsters chase him/her there. Several of Infocom's other games—particularly the mysteries—had puzzles that were more closely related to the plot, but even in those the PC had little personality of note; if it could be assumed that the PC was just as interested in solving the mystery as the player was, there was no need at any point for the player to take stock of the PC's motivations. Representative in this respect is *The Lurking Horror* (Dave Lebling, Infocom, 1987), where the PC, a university student, is given a clear initial motivation, namely finishing a paper; when he discovers that something has gone wrong with his paper and that he might learn more in a certain department, the motivation becomes getting to that department. So far, so good, but various sinister events happen on the way (and more happen when he gets there), and at some undefined point the PC apparently says, "Forget the paper—I'm going to figure out what's going on around here!" That's not entirely impossible, but it does essentially merge the PC with the player, since both apparently have as a goal Understanding Everything In Sight—which conveniently allows the PC to do all sorts of stuff that merely finishing a paper wouldn't justify.

One notable exception to the general rule, however, was *Plundered Hearts* (Amy Briggs, Infocom, 1987), where the PC is a 17th-century girl charged with the tasks of (a) surviving various dangers, (b) rescuing her father from the clutches of an unsavory fellow, (c) defending her own chastity from the wiles of the same fellow, and (d) catching the eye of,

rescuing from the same villain, and ultimately marrying a dashing young swain. All of the goals can ultimately be accomplished, naturally, though it's sometimes unclear which one the game wants the player to focus on, and it's possible to bog down if the player focuses, say, too exclusively on saving the PC's skin and insufficiently on bailing the father or the swain out of jams. (Interestingly, multiple endings are possible, and the suboptimal ones correspond roughly to achieving only some of the motivations: in one, the heroine saves herself but abandons her lover and father; in another, she sacrifices herself to save her lover; and in another, she saves her lover but not her father.) Still, having too many motivations represents a refreshing change from having no particular motivation (or having a general goal that has no obvious impact on any particular decision), and the puzzle-solving in *Plundered Hearts* is usually sufficiently plot-driven that it cannot fairly be reduced to "go poke around." For example, at one point the plot depends on your gaining admission to a ball, which requires (a) producing your invitation and (b) getting appropriately attired. It's not obvious that any of the overall motivations for the game require attendance at the ball, but the game helpfully supplies you with some—"You haven't been to a ball in months!" A few stretches of the game involve simple exploration—but it's exploration of quite limited environments, and for clearly defined objectives (getting off a drifting ship in one case, finding your father in another).

That moment, though, is an exception; the game is relatively restrained about pointing you in the direction of the story it wants to tell (and hence spelling out the PC's motivations). At one key point, for example, you take a moment to signal your friends that you need help (even though your need is not, at that point, all that pressing), and your friends conveniently show up just in time later; it's possible, as I recall, to simply not think to signal, with the result that the friends don't show up later and things go awry. There, for purposes of keeping the story on track, the game might simply have said something like, "Before you do anything else, you should signal your friends," a bit intrusive but helpful for purposes of making sure the PC and the player are on the same general page. It helps that *Plundered Hearts* isn't a particularly long game and that the story isn't all that wildly complex; the player might conceivably do the signaling simply because the opportunity presents itself. A more complicated storyline might demand more such signals about what the PC really wants.

It's also worth noting that the *Plundered Hearts* PC's personality is considerably more vivid than that of any other Infocom PC; there are things, for example, that you won't do because it's not in your nature or because you've been advised against those things. Whether that strong personality is a cause or effect of the overabundance of motivations—or is wholly unrelated—doesn't ultimately matter.

Latter-Day IF

(This section contains major spoilers for Delusions and minor spoilers for Jigsaw, Sunset Over Savannah, Little Blue Men, and Varicella.)

Plundered Hearts stood as an exception to the general pattern for quite a while, even after commercial IF died out, and it was not until the latter half of the '90s, when IF authors began to move away from puzzle-solving as IF's *raison d'être*, that PC motivation once more became the subject of experiments. An early precursor to that trend came in 1995 with Graham Nelson's *Jigsaw*, a time-travel game that moves the player through a variety of 20th-century scenes, each time chasing a nemesis determined to change history and each time attempting to set history back on course. The larger question of motivation—why is the PC so convinced that history must unfold just as it did?—is not explored as fully as it might be, but everything that the PC needs to do in each self-contained scene proceeds reasonably logically from the objective—which is itself obvious (if the player knows the history that needs to be set right).

Similarly cognizant of the problem of PC motivation was C. E. Forman's *Delusions* in 1996, where the PC's goals change repeatedly over the course of the game: the initial goal is to test a VR simulation, a subsequent goal is to escape a different simulation, and a final goal involves defeating a certain villain. The transitions between goals are reasonably logical, enough so that there's never a question about what objective the player is pursuing at any given time. For example, the player emerges from one simulation into a suddenly deserted lab (which turns out to be another simulation), and immediately both player and PC have a new goal: to figure out where everyone went and, when the nature of the problem emerges, to find a way out. When the way out is found, the PC stumbles directly into a conversation involving the villain, and the new motivation is obvious. Perhaps more importantly, however, a fourth goal emerges along the way, namely that the PC wants an explanation of

some nagging inconsistencies tugging at the edge of his consciousness. The subtlety with which this particular goal emerges—through various discoveries that initially appear unconnected—makes *Delusions* noteworthy, as the game manages to encourage the player to solve the problem without spelling things out directly.

Other recent games have likewise tackled the problem of motivation in innovative ways. In Michael Gentry's *Little Blue Men* (1998), for instance, the PC's primary goal is to eliminate things that annoy him—in the game's parlance, to become “frosty” rather than “steamed”—and pursuing that goal single-mindedly (how to do that is usually, though not invariably, clear) sends the story in all sorts of unexpected places. Achieving the PC's motivations means, in other words, that the player has to take on a distinctly antisocial mindset—meaning, if not an outright conflict between the player and the PC, at least a bit of tension. The execution isn't, alas, quite perfect; there are moments where the player's sheer curiosity has to take over from the PC's quest to maintain his emotional balance, as the author can't quite connect the PC's goals with certain bits of puzzle-solving. Ivan Cockrum's *Sunset Over Savannah* (1997) is in some ways the obverse of *Little Blue Men*: the PC has decided that she hates her job and is trying to decide whether to quit, and she wanders a beach, encountering various experiences that remind her of her forgotten joie de vivre. The PC's sole motivation is to make the decision; the player's motivation becomes seeking out the experiences that will help the PC decide. Again, the puzzle-solving strains the model; the player repeatedly compels the PC to do things that simple restlessness would be unlikely to motivate (since a few of them are a bit on the suicidal side). Still, the underlying motivations are unusual, which is worth noting in itself, and they keep the player thinking.

Adam Cadre's *Varicella* (1999), like *Little Blue Men*, poses the problem of a PC with unattractive motivations; unlike *Little Blue Men*, though, the story is sufficiently well engineered that the PC's motivations consistently drive the plot. The PC is a palace minister aspiring to a recently vacated throne and scheming to bump off those with precedence in the succession order. The game's puzzles all tie directly into one of the various murders—there are no occasions where the PC has to get through a locked door or solve a Towers of Hanoi puzzle in order to obtain some unforeseeable object that will prove useful. (In other words, the author didn't just happen to pick a good plot; the game is designed in such a way that the player's magpie/Mensa member personality rarely, if

ever, shows through.) Other games of note in this regard include Stephen Granade's *Losing Your Grip* (1998), in which the PC goes through a lengthy internal odyssey, confronting various inner demons—and while it would be difficult to say that the PC ever has an overall objective, at various times he is given smaller goals that advance the plot in a variety of logical ways. (There are a few exceptions—scenes that seem more akin to a set of puzzles than a stage in the PC's odyssey—and, not surprisingly, those scenes were heavily criticized.)

PC Motivations and Game Design

(This section contains major spoilers for Spider and Web, Fail-Safe, The Weapon, LASH, and 9:05 and minor spoilers for Enchanter, Sorcerer, My Angel, and Photopia.)

What are the implications of all this for the would-be designer? First, shorter is usually better; most of the games that have done a creditable job of accounting for the PC's motivations have been relatively short. Most IF puzzles are set-pieces—it isn't easy to develop a lot of puzzles that fit seamlessly into a game, unless your story is "solve the puzzles in your latest *Games* magazine." The longer the game, and the more puzzles, the more likely it is that the PC will need to solve puzzles that he/she has no obvious reason to solve.

Second, many of the games that have developed strong PC motivations have been segmented, so that a particular motivation, once accomplished through solving a puzzle or two, gives rise to another motivation that is accompanied by another puzzle. (Of course, if your game's so short that there's just one puzzle—and everything the PC does has an obvious connection to that puzzle—segmenting is less of a big deal. *Enlightenment* (Taro Ogawa, 1998) is one example.) Each goal should be relatively immediate, sufficiently so that puzzle-solving has an obvious connection—if there's a door to unlock, there should be some good reason for thinking that something interesting to the PC is on the other side of the door. Many games begin this way, typically with some humdrum task—in *The Lurking Horror* it was writing a paper, in *Wishbringer* (Brian Moriarty, Infocom, 1985) it was delivering a letter—but lose motivational focus when the wide middle section of the game comes along. The solution may lie in escaping the clutches of the short, narrow introduction, long, wide middle game, and short, narrow

endgame structure, as it's the middle game that tends to be closest to an unabashed collection of puzzles (whereas the humdrum-task introduction and the my-objective-is-before-me endgame tend to have goals that are directly related to the puzzle-solving). *Enchanter* (Marc Blank and Dave Lebling, Infocom, 1983) and *Sorcerer* (Steve Meretzky, Infocom, 1984), where the beginning tasks are, respectively, getting into the bad guy's castle and getting out of your own Guild Hall (and the endgame objectives are essentially kill-the-bad-guy), are examples. Obviously, the goals also need to be less cosmic than save-the-world (which Infocom was rather fond of but the latter-day IF community has moved away from).

Third, imputing emotions and thoughts may be necessary to tell the story you want to tell, if that story is at all complex. *Delusions* did this to great effect, in part because it was unexpected; a PC without strong reactions to most stimuli had a progressively stronger reaction as it confronted the inconsistency, and it helped convince the player that this was an avenue worth exploring. To be sure, it's possible to "discover" your own feelings—you could, say, have the PC read a letter that he or she has written—but if you're already inside the PC's head, it seems easiest simply to tell the player what's going on in that head, absent some compelling reason not to share that information with the player (as discussed below). Again, the puzzle-oriented roots of IF tend to encourage an emphasis on the objective—but the current state of the art calls for more subjectivity, a less Hemingwayesque persona at the center of it all. The occasionally confusing nature of *Plundered Hearts*, which, as noted, generally avoided informing the player of the PC's thoughts, suggests that less, in this regard, is not necessarily more.

Fourth, emphasizing story and de-emphasizing puzzles may make it easier to tie the PC's motivations to the course of the game (assuming the story is even vaguely plausible). One of the more striking examples of this was Jon Ingold's *My Angel* (2000), where there are very few puzzles (and what puzzles there are tend to be small-scale and short-term) but the game tells a reasonably complex story. The PC's immediate motivations are clear throughout the story, even though he doesn't really have a discernible large-scale objective (and there are good reasons for that); the story begins with him following his companion, and each stage of the journey brings up a new objective. There, overcoming whatever obstacles impede the journey tells the story in itself, and there's little ambiguity about why the PC continues the journey—and hence the few

puzzles feel adequately motivated. (In theory, they could have fit into the story even if they'd been more numerous and more conventional puzzles, but then you'd get questions about why exactly this journey seems to lead through so many barriers and require so much riddle-solving. Again, the idea is a *plausible* story.) Conversely, it's hard to tell a convincing story if the character at the center of it remains a cipher.

Illustrative in this respect is Adam Cadre's *Photopia* (1998), which alternates between fantasy and real life—and most of the fantasy segments have puzzles associated with them, while the real-life scenes have interaction but no puzzles. Each of the scenes from real life features a different PC, each of which has a reasonably obvious motivation in that scene, and each scene advances the plot in measurable ways. The fantasy bits, on the other hand, have thematic connections to the plot but don't (by the very nature of the story) advance the ball, and the puzzle-solving in each of those segments is a mite aimless—even when you know what you're trying to do, there's no obvious way to go about doing it. There are good reasons in the context of this particular game why the path of the fantasy story would wander, but the contrast is still striking: the progress of the story hinges on clearly motivated non-puzzle interactions on the one side and on less clearly motivated puzzle-solving on the other.

A wholly different approach to the story/puzzles balance, and to bringing the PC to life, was seen in Stephen Bond's *Rameses* (2000), which had no puzzles at all and a PC who was so paralyzed by his own insecurities and neuroses that virtually every attempt to interact meaningfully with the environment was rebuffed by some variant of "You're afraid to do that." In *Rameses*, the *failure* to overcome the obstacles confronting the PC tells the story.

More attention to the PC's motivations not only tends to go along with better storytelling, moreover, but also tends to make for better puzzles, for those who still prefer to write puzzle-oriented games. Set pieces, or "soup can" puzzles (so named when Russ Bryan commented in a newsgroup discussion of the graphic adventure *Seventh Guest*, "What the hell kind of villain thwarts the hero's progress with soup cans in the kitchen pantry?"), have become increasingly disfavored for that very reason: they don't fit into the plot, and the PC (usually) has no particular reason to solve them. Likewise, players often complain about puzzles that require suicidal tendencies (as in *Sunset Over Savannah*) on the theory that the PC should be assumed to be a relatively rational actor. It's worth

remembering that several of the most innovative puzzles of recent years have turned on well-developed PC motivations. In Andrew Plotkin's *Spider and Web* (1998), for instance, the PC is a captured spy who lies to his interrogator about his movements inside the lab where he is caught. The player, however, does not know that the PC is lying, and hence has no reason to believe that the PC has any reason to conceal anything from the interpreter—and the puzzle that calls for the player to draw that conclusion is a masterstroke. This sort of split between player and PC is difficult to accomplish unless the PC has his/her own goals and motivations, and centering the story on the PC motivations, in this case, makes both the story and the puzzle work. PC motivations, and their concealment from the player, play similar roles in Jon Ingold's *Fail-Safe* (2000) and Sean Barrett's *The Weapon* (2001), and in each case the game's primary puzzle involved picking up on those motivations.

Furthermore, some of the most interesting IF experiments of recent years—experiments that have pushed the bounds of IF theory—depended on developing the PC's persona and motivations. *Spider and Web*, *Fail-Safe*, and *The Weapon* are all examples; Paul O'Brian's *LASH* (2000), where the PC is a robot that eventually rebels against the humans controlling it (and who, naturally, control it in much the way a player controls a PC), is another. (One can read into this device a bit of commentary about how the conventional IF PC resembles an automaton mindlessly picking up objects.) One of the cleverest such experiments is Adam Cadre's *9:05* (2000), where the PC is a burglar who has broken into a house and killed the occupant—but the player, waking up the next morning in the house, assumes it to be his and takes the PC off to the victim's workplace in the victim's car (and promptly gets caught). The joke there, obviously, is that the PC has motivations that he doesn't get around to sharing with the player and that the player's assumption that the PC has certain motivations (getting to work) turns out to be unwarranted. Many interesting things have yet to be done with the player-PC relationship, and giving the PC a persona—with personality, worldview, and highly specific motivations—open up the possibilities.

Conclusion

Character development in IF—whether of the PC or of the non-player characters (NPCs)—long lagged behind puzzle design and other aspects

of world-modeling, in part, no doubt, because of the difficulty of coding even vaguely realistic human interactions. While development of a PC doesn't depend on coding as such, it's arguably true that a game with minimal human interaction (as IF has historically tended to be) tends to need less in the way of PC development; who cares about crafting an intricate PC psychology when the PC has nothing to match wits with? (Technically, of course, the human interactions could be offscreen, or appear in cut scenes, but the tendency, I think, has been simply to write barren, austere games.) NPC interaction has matured over the past few years, however; the hybrid conversation structure (accommodating both menus and general topic selection) that Emily Short has pioneered and that some others have attempted allows for conversation with considerably more depth and range. That, in turn, calls for a more developed PC to hold up an end of those conversations. The newfound emphasis on the storytelling aspects of IF—partly driving and partly resulting from advancements in NPC development—likewise calls for a PC whose complexity matches that of the narrative. Letting the PC's motivations, rather than the player's motivations, determine the course of the story is among the challenges that IF developers should look to meet.

Landscape and Character in IF

Paul O'Brian

If we reduce interactive fiction to its essence, we can view it as a triangular relationship between three basic elements: landscape, character, and action. It's possible to write IF without objects, plot, NPCs, or myriad other ingredients, but as soon as that first room description appears, it introduces a landscape, just as the first prompt ushers in the concept of action.¹ I would further argue that the interaction between these two elements inevitably creates some concept of character. The character that emerges is the being that would perform the actions selected when presented with the landscape (and situation) at hand. Even if that character is not human, not organic, or not even embodied (an omniscient narrator, for instance, though that voice is almost never used in IF because of the form's powerful insistence on connecting action with viewpoint), action must have an agent, and that agent is what we call the player character.

In this formulation, the only one of the triumvirate completely under the game's control is landscape. Action is entirely in the hands of the player, and character lies halfway between the two. That last statement may require a bit more unpacking. If character is determined by action, why isn't it entirely in the hands of the player as well? The answer is that while action *does* determine character, it isn't the sole determining factor. The game itself can shape character by statements as blatant as "You're Tracy Valencia," or by something as subtle as a particular word choice in a parser response. However, I would contend that while blatant character-shaping statements and even subtle nudges from default responses are far from inevitable in IF, some sense of landscape must be

1 A note about terms here: First, I should note that by "interactive fiction," I refer to text IF. Some of the points here are certainly applicable to graphical or mixed-media IF as well, but some may not be. Secondly, the general concepts of landscape, character, and action aren't meant to be taken too literally. IF could be (and in many cases has been) created with a map of entirely abstract locations, or one location, or location descriptions that consist entirely of describing what's absent. Similarly, actions might involve no actual action (WAIT, for example, or THINK), and a character can be anything from an intrepid adventurer to an ear of corn. However, I would contend that these elements are present in some form in all IF—indeed, the absence of these elements (such as the absence of landscape in Eliza) removes the work from what might reasonably be called interactive fiction.

included in any IF game, and that both the design and the description of this landscape are extremely powerful factors in determining character. It is my aim in this essay to examine the ways in which landscape influences and creates character and to raise what I hope will be some interesting questions about the nature of their interrelationship.

Map Design

[This section contains minor spoilers for Colossal Cave, Planet of the Infinite Minds, Strangers in the Night, and Suspect. It contains medium-level spoilers for Lost New York and Stone Cell and major spoilers for Shade and 1981.]

Space is continuous. The landscape of interactive fiction, however, consists of discrete units, connected to each other in various ordinary and sometimes extraordinary ways.² By convention we call these units “rooms,” but in fact they can be anything from a tiny subsection of a room to an entire town, country, planet, or universe. How does a game’s subdivision of continuous space affect our perception of the character in that game? Let’s look at some examples.

An illuminating comparison exists between two pieces of IF with urban settings: Neil deMause’s *Lost New York* (1996) and Rich Pizor’s *Strangers in the Night* (1999). In the former, the character travels through Manhattan and other areas of New York City during various points in history. The game frequently compresses neighborhoods, boroughs, and other such swaths of territory into single rooms, albeit lovingly described ones:

Lower East Side

The scene around you is one unmatched in any other time and place in human history: Acres of identical four- and five-story tenements packed cheek-to-jowl with people, people who spill out onto the sidewalks and fire escapes in search of a little space, a little air. The el tracks continue down the street to the north and south; to the

2 This trait isn’t entirely restricted to text games, but while no text game offers continuous space, some graphical games, such as *Half-Life* and *Zork: Grand Inquisitor*, do in fact offer a continuous, unbroken environment through which the PC travels. In that case, map design becomes a much less powerful factor in fixing character, and in fact it might be argued that in those cases, the term “map design” has more or less lost its meaning and might be better called “level design” or something similar.

east, the tenements seem to stretch on forever, though you're pretty sure they eventually end at the East River shore.

Within each time period, these areas connect to each other directly, even though they may have been separated by miles in reality. Occasionally, a "traveling message" such as "You trudge north for close to a mile, finally arriving at . . ." will interpose itself between locations, but more often the traveling interval passes instantaneously and without comment.

Strangers in the Night, on the other hand, painstakingly sets out its generic city map as a street grid and provides almost no description for the lion's share of its locations:

Broadway and 11th

You are at the intersection of Broadway and 11th Avenue. To the southwest is the security door for your apartment building; the Broadway Sineplex (which a few downtown residents still consider an amusing name for a movie theatre) lies to the northeast.

>n

Broadway (10th & 11th)

>n

Broadway and 10th

One of the streetlights is dim here; the shadows that are cast against the sidewalk are oddly deformed, giving the corner an otherworldly feel.

>n

Broadway (9th & 10th)

Somewhere in the distance, a car alarm starts blaring it's [sic] Call of the Wild to the concrete jungle. This is followed a few moments later by the sound of figerglass [sic] crunching and safety windshields shattering until the alarm ceases.

>n

Broadway and 9th

>n

Broadway (8th & 9th)

>n

8th and Broadway

Carl Tuck's Coffehouse [sic] is to the southeast.

At first glance, it might appear that about half the locations contain room descriptions. In fact, however, only the first and the last do; the dim streetlight and the car alarm are random atmospheric messages that can pop up in any street location. In fact, the only time a non-random message, or a description of any kind, appears is when the location adjoins the entrance to a puzzle-solving area or to the PC's home. The game's city grid is comprised of about 80 locations, all of which may well have been compressed into one room in *Lost New York*.

It's a natural impulse to discuss these choices as they relate to game design, or to talk about their successes and failures in creating immersion or facilitating strategy. What may not be so natural is to think about how these choices influence the way we think about the player character; I would contend that consciously or not, we perceive these two characters differently based on the way the games construct their surroundings. What we know about the PC of *Lost New York* is that she³ knows New York City well enough to identify its various areas instantly, even as they appeared over a hundred years ago. In fact, the game's easy recognition of areas such as "The Goats" and "Ladies Mile," not to mention the copious historical detail infused into many room descriptions, creates a tension between the game's identification of the PC as "a tourist" and what we know about her from her subsequent experiences. Someone who could wander through New York's past with so much information at hand must be intimately familiar with the city, either through experience or study. Her interest and perception is mostly broad strokes—she's more interested in generalities of an area than in its specific details, and her sense of history is sweeping rather than finely grained—but her knowledge is quite comprehensive. Even if the game had insisted that this was the PC's first trip to the city, we would have to conclude that she is someone who for whatever reason has immersed herself in New York City history; how else to explain such detailed knowledge in the midst of the extraordinary experience of time travel? If the game proved unable or unwilling to address and resolve this

3 The subject of how description influences our perception of PC gender could occupy another entire essay and is out of scope for this one. Consequently, in the case of games that don't explicitly specify the gender of the PC, I'll rather arbitrarily select one, trying to hit a more-or-less even ratio between the two.

question, that lacuna could hardly be anything but a flaw in the work, just as it would be in a novel where the main character knows things she shouldn't.

Unlike the PC of *Lost New York*, the character in *Strangers in the Night* has almost no interest in ordinary detail, let alone history. He never finds himself musing about ironies or architecture as he treads the streets and, in fact, usually notices nothing but the bare identity of the location. Together with the game's specification of the PC as a vampire, these facts can lead us to a few conclusions about this character. He sees the city not as human tapestry or even interesting backdrop but rather as a sort of maze he must navigate in order to locate prey. The lack of room descriptions impels us to move quickly from one location to the next, replicating the urgency of the character's thirst for blood. His disinterest in local color might even be seen as an undead disdain for the fleeting effluvia of mortal life. The game's overall presentation isn't quite strong enough to give this effect full potency, but all the same we know quite well that there is a significant difference between these two characters. The *Lost New York* PC, even if she *were* a vampire searching the streets for prey, couldn't help but notice the landscape and be aware of its heritage, while the *Strangers* PC could be thrust into any time in the city's history and would evince a similar disregard for anything but the most minimal details of place.

On the other end of the detail spectrum from *Strangers* is Andrew Plotkin's *Shade* (2000), where the entirety of the action appears to be taking place in the player character's studio apartment. In this one-room environment, however, movement is possible, and the game responds to this movement not by placing the character in a new room (as is the case with most IF) but rather by making a series of alterations to room description and scope for the current room. If the PC is in the center of the apartment, for example, the game first mentions objects close at hand, such as the computer desk and stereo, while reserving mention of the kitchen and bathroom areas for the later parts of the room description. When that character moves to the kitchen, however, text about the counter, the refrigerator, and such occupies the beginning of the room description. The desk and stereo are still visible from that location, and still mentioned, but are only visible, not accessible for touching or other manipulation until the character returns to the center of the apartment.

When *Shade* was released, this approach to map design was hailed as

an innovative subversion of the conventional IF map, which it is. It is also a fitting choice for characterization purposes. The overlapping, connected nature of the apartment landscape makes clear to us that this is an environment with which the character is intimately familiar and that even while he inhabits one area of it, his awareness of the other areas does not abate. The map design makes the apartment *belong* to the character in a way that it would not were it separated into discrete rooms. This sense of familiarity, of safety, and of enclosure makes the game's later revelations all the more powerful, as the familiar dissolves into the strange, and safe enclosure into fatal exposure.

A similar effect, the subdivision of one room into many separate locations, appears in a variety of games, including Infocom's *Suspect* (Dave Lebling, 1984) and Steve Kodat's *Stone Cell* (1999). In the former, it's a grand ballroom that the game presents as nine separate locations, and the effect is to make the room feel enormous. The character in *Suspect* is a reporter at a party being given in a mansion, and the game's map design underscores her sense of awe at the opulent surroundings—where the house's owner's perception of the ballroom might be closer to that of the character in *Shade*, the guest's mind demands more concrete conceptual boundaries in order to take in the scope of the area. *Stone Cell* achieves a different effect by performing the same gridlike subdivision on a much smaller room, the eponymous stone cell. Room descriptions and common sense tell us that this room is much smaller than the ballroom in *Suspect*, so the game's partitioning of that space, rather than conveying immensity, instead reflects the PC's awareness of the room's tiniest details as a result of his imprisonment. What makes this design particularly effective is that the game initially presents the cell as one location, then expands it into a grid after the character sleeps, thus reflecting not only the character's growing familiarity with his surroundings but also his growing desire to scrutinize each detail of the premises in hopes of escape.

The opposite effect is available, too, when games compress the extremely large, even the inconceivably large, into a single room description. One of the more extreme examples of this technique occurs in Alfredo Garcia's *Planet of the Infinite Minds* (2000), where the character might find himself here:

The Beginning of Space

All around you, distant suns flicker and twinkle. Painfully bright points of light seem to appear suddenly from out of the ether, as

another retracts into obscurity. En masse, the effect renders a carnival of vibrant colours and astonishing beauty.

The simple fact of the character's existence in this location tells us something about that character: that she has transcended humanity, attaining a sort of bodiless, godlike status. Since the game starts with the PC as a simple librarian, its transportation of her to such an abstract vista carries with it the implication of personal disembodiment and removal from reality as well. What's more, her ability to know that the location is "The Beginning of Space" rather than, say, a Christmas tree festival viewed through a hangover, suggests a metahuman omnipotence that we must assume has been granted to the character, at least temporarily.

The connection between map design and character stretches to the deepest roots of IF, for the majority of *Colossal Cave's* (Will Crowther, 1976) map is named and divided in ways that would make sense to a spelunker. From the way that the game comfortably names areas of the cave as "rooms," and indeed even the names of those rooms, which draw on caving vocabulary such as "Bedquilt" and "Y2," we can clearly identify that the character in that game is an experienced cave explorer. Thus, even in the earliest days of IF, when games made virtually no overt effort to characterize the PC, character was already emerging as a function of landscape. The character in *Colossal Cave*, while unraced, gender-neutral, ageless, nameless, and faceless, was nonetheless made distinct from the player herself by the way he perceived the landscape of the cave, seeing rooms and twisty little passages where a different character might have experienced the area quite differently.

In the hands of a skilled author, the effect of landscape on character can make for a portrayal that is very striking indeed. Take, for instance, Adam Cadre's *1981* (2001).⁴ The first room description of the game is as follows:

New Haven, Connecticut

New Haven. The worst place on earth. The town is dirty and

4 *1981* is credited to the pseudonymous A.D. Mcmlxxxi, and Cadre has never claimed credit for it. In private correspondence, he explained that this is because the game was a bit of a rush job, not polished enough for something he would put his own name on. He agreed to be credited for the game in this essay on the condition that I put in a note explaining that he "wasn't actually *trying* or anything with that one." That *1981* is the game Cadre produces when he isn't even breaking a sweat is a testament to his skill as an author.

industrial, the students are sloppy, everything is horribly expensive. And you had to cash in \$3600 of your stock to get here. But it was necessary. Four years at this place is enough to ravage anyone. You have to rescue her, your first true love.

Her dormitory lies to the north.

Already, we can see a dramatic narrowing of scope occurring. The character is so unconcerned with the details of his location that he compresses an entire town into one unit, dismissing all of it as “the worst place on earth.” Then the broad outlines of location gain sudden, sharp focus: “Her dormitory lies to the north.” The contrast between the vague, reviled whole of New Haven and the focus on the dormitory, set apart in its own line, suggests to us that the character’s concentration on his goal is unhealthy, perhaps even obsessive, and moving north confirms this suspicion:

New Haven, in her dormitory

You’re standing in front of her door. It’s closed. It’s always closed.

You’ve shoved approximately one hundred poems and letters under that door. You figure she’s probably read about half of them.

Scope narrows even further here, from one building to the tiny area in front of one of that building’s doors. The room descriptions certainly confirm our impression of the main character’s unbalanced and obsessive nature, but even without them, the basic funneling performed by the map design would get the point across admirably. When we discover that the PC is John Hinckley, Jr., and that the door in question is to Jodie Foster’s dorm room at Yale, the revelation is terrifically powerful, because via its map design, the game has already taken us directly into the viewpoint of its would-be assassin.

Room Descriptions

[This section contains minor spoilers for A Change in the Weather, Heroes, Varicella, and Zork I. It contains medium-level spoilers for Once and Future and Wearing the Claw and major spoilers for Nothing More, Nothing Less.]

Of course, in 1981 it’s more than just map design that clues us into the character—the room descriptions themselves make it clear that we are

seeing the game's landscape as filtered through one individual's highly idiosyncratic viewpoint. Short, choppy sentences give the text a jittery feel, contributing to the general tone of uneasiness. We know the character has some access to wealth because of the "\$3600 of your stock" line. We also know the character is either a heterosexual male or a homosexual female from the reference to the true love as "her."⁵ And we certainly know how he feels about New Haven.

Cadre is particularly skilled at bringing character across through room description, as in this example, the first room in *Varicella* (1999):

Salon

You've funneled the lion's share of the palace improvements budget -- and most likely the tiger's share as well -- into renovating the salon... not that the Philistines you live among are equipped to appreciate it. From the plush Quattordici chairs to the handsome volumes in the library to the imported Ming tea service to the steward you hired to attend to your grooming needs, this is an oasis of taste and comfort in what is otherwise a fairly uncomfortable and tasteless building. Ah, well. When you become Regent you'll have greater latitude to redecorate. The arched windows overlook the western gardens, while the exit leads east.

This description follows several paragraphs of introduction, which announce the player character as one Primo Varicella, Palace Minister to a recently deceased king, and Machiavellian schemer for the throne. Even without that introduction, though, this room description would frame the character aptly. From the "lion's share" clause we know that the character is in charge of improvements to a palace, and from the room name we know that he is in the Salon mentioned in the first sentence; therefore we can conclude that he is employed by the palace in which the game begins—a succinct way to bring across Varicella's position and occupation. Moreover, the phrase "live among" tells us that he resides at the palace as well. The "tiger's share" clause gives us an example of his sardonic humor, and the "Philistines" reference an example of his snobbery. His identification of the chairs and tea service, and the contrast to the "uncomfortable and tasteless" remainder of the building, communicate clearly that this is a man of very strong preferences, a persnickety aristocrat whose refined tastes run to the extremely expensive. Finally, the character's ambitions, and the drive behind them,

5 Or a bisexual of either sex, it probably should be said.

are summarized neatly: “When you become Regent you’ll have greater latitude to redecorate.” Just by seeing one room through this character’s eyes, we learn all the essential facts about him that will carry throughout the game.

If there’s a continuum that measures the degree to which a game’s room descriptions blatantly shape character, it’s fair to say that *Varicella* is probably on the extreme end of it. Does that mean that the room descriptions of games on the other end don’t shape character at all? Predictably, my answer is no—the effect is just a bit more subtle. To illustrate, let’s compare descriptions from two different games, neither of which has character as its focus. First, from Andrew Plotkin’s *A Change in the Weather* (1996):

Rocky Outlook

A wide angular tongue juts out from the hillside. The park stretches off to the north and west, a vast expanse of luminous meadowland, patched with the dark emerald of forest. The streams are already shadowed in their beds. In the distance, a lake reflects red fire, beneath the greater fire that leaps silently on the horizon.

A trail leads southwest down the hill, towards the bridge. From where you stand, it turns southeast and continues upward, deeply cut into the hillside. A narrower trail leads more steeply up to the east.

Zork I (Marc Blank and Dave Lebling, Infocom, 1981) offers a location that is very nearly analogous:

Rocky Ledge

You are on a ledge about halfway up the wall of the river canyon. You can see from here that the main flow from Aragain Falls twists along a passage which it is impossible for you to enter. Below you is the canyon bottom. Above you is more cliff, which appears climbable.

Though their locations may be similar, these two characters are very different indeed. *Weather*’s wanderer takes the entire first paragraph to describe the area with intense, poetic language. The words don’t directly narrate the emotions felt by the character, nor impute opinions like the descriptions in *Varicella*, but they deploy vivid adjectives like “luminous” and “dark emerald” and powerful metaphors—the tongue of rock, the

red fire of sunset reflected in a lake, the setting sun as a “greater fire that leaps silently on the horizon.” This is a character whose soul is moved by the grandeur of a natural landscape. Only after this reverent depiction does the character notice practical details: the trails and where they lead.

Zork’s PC, on the other hand, goes directly for the practical. She mentions the river’s passage only in terms of whether she can enter it. What she notices about the cliff is that it is climbable. Though the natural scene—a canyon, a river, a waterfall—is probably quite impressive, the description is almost entirely mechanical. There are no rapturous sentences about the stark rock of the cliff or the sparkling river. Adjectives are almost entirely absent, and where they do exist their purpose is highly prosaic: “river” further identifies “canyon,” as “Aragain” does “Falls” and “canyon” does “bottom.” Other descriptors exist solely to describe travel options: “impossible” and “climbable.” Indeed, she sees every element of the scene only in terms of how it can be manipulated or traversed, and this viewpoint is consistent throughout the game, just as the intense description of natural phenomena is a constant in *Weather*. Both games’ main focus is puzzle-solving, but when we compare how their characters each view a similar scene, it becomes clear how different the characters are from each other.

Comparing the PCs of two different games illuminates important differences between the characters, and the effect is even more potent when several points of view are available within the same game—instead of seeing how two different characters view analogous locations, we get to see how they view the exact *same* location. Several recent games have made use of this technique: J. Robinson Wheeler’s *Being Andrew Plotkin* (2000), Stephen Granade’s *Common Ground* (2000), and my own *LASH* (2000) among them. The current apex of POV-diversity, though, probably belongs to *Heroes* (2001) by Sean Barrett. This game offers a minimal landscape of something like a dozen locations but gives five different viewpoint options through which to view it. For instance, the opening location of the game as viewed by a *Zork*-like adventurer:

Temple Way

The grimy, ramshackle buildings of Oldtown dutifully try to reform themselves as you progress east down Temple Way, but nothing besides the temple itself makes any real pretense of belonging anywhere other than Oldtown. Or rather, nothing besides the temple and Baron Sedmon’s nearby mansion.

a king:

Avenue

This broad avenue leads right into Temple Square, the heart of fabulous New Oldtown. Towering over the square to the east you do perceive your stark white Temple of Justice, beautiful and well-appointed, offering a statement to the neighborhood: this, this is what progress is about. Sadly, the buildings around you are scarcely up to this new standard; Baron Ventillado's house north of the square is much more satisfactory. How you hate having to come here. This would all be so much simpler if Blackhelm were found dead one morning, but it's never happened yet, despite your best efforts.

a thief:

Shadowy Road

Sturdy, functional buildings lie in and out of shadow on the road to the temple square. Simple architecture, devoid of handholds; closely spaced buildings, devoid of alleyways; uncut walls, devoid of windows: the builders in this area knew how to encourage amateurs to go elsewhere.

a mage:

East-West Road

Randomly arranged paving stones form this street, proceeding east towards a more attractive arrangement. The darkened buildings lean sloppily over the edge of the street, reducing the energetic potential of the strict east-west layout. West the road leads back into the seething mess that is Oldtown.

and finally, a dragon:

Open Tunnel

We were surrounded by the man-things' structures, structures of dead trees and rock and distortions of iron. Beneath us we felt the arrangements of stone into a path for man-things' mobile receptacles. We could smell hints of the Crystal along the path to the east.

Where the adventurer just sees a temple, the king sees the temple as his own possession, a symbol of his attempts to renovate and improve the

city. Where the mage sees leaning buildings distorting the street's pristine geometry, the thief sees those same leaning buildings as a source of precious shadow. Through the use of a past-tense, first-person plural voice, *Heroes* renders the dragon's viewpoint quite alien and emphasizes that dissonance by showing us how the dragon sees the street: an "open tunnel," contrasted with the more irregular shapes of nature and constructed by contemptible "man-things." *Heroes* takes excellent advantage of landscape's ability to reveal character, and through its use of multiple viewpoints, it leverages the power of the landscape-character axis to accomplish something more: the revelation of landscape via accumulated details from a variety of characters. The descriptions coalesce in the player's mind to create a picture of the location that is much more complete than any one viewpoint could provide, while at the same time establishing distinct portraits of each viewpoint character.

Other games have made use of changing room descriptions in order to demonstrate change or progression in a single PC, or to give us that character's revised perspective as a situation changes. *Nothing More, Nothing Less* (2001) by Gilles Duchesne is a case in point. The first puzzle of this slice-of-life game takes place in a bathroom, initially described like so:

Bathroom

I've seen bigger bathrooms, but must admit this smaller one suits our needs well. There's a small sink with a cabinet under it, a mirror, a bathtub (equiped [sic] with a shower head and curtain) and a toilet.

However, after the character urinates,⁶ the toilet clogs and begins to overflow. Unprompted, the game reprints the room description, which now reads as follows:

Bathroom

I've seen bigger bathrooms, but must admit this smaller one suits our needs well. There's a small sink with a cabinet under it, a mirror, a bathtub (equiped with a shower head and curtain) and a toilet. Right now, my attention is also grabbed by: the toilet tank. Water keeps flowing from the tank, nearing the bowl's edge.

The room description stays the same, but the game adds a sentence to

6 This is one of the very few times that excretion has appeared in an IF game without being a function of rather dodgy toilet humor. Instead, the game plays it completely straight—just another element in its realistic scenario.

demonstrate that the character's attention has become focused on one particular aspect of the room: the toilet tank. This sentence serves gameplay purposes, indicating that the toilet tank is in fact implemented and thereby hinting toward the solution of the "overflowing toilet" puzzle. In addition to this, the attention sentence demonstrates a shift in the character, showing us his revised perspective as well as the fact that he's quick-witted enough to think immediately of the toilet tank in this crisis. The other sentence is typical of IF room descriptions, indicating an action currently taking place in the room and lending urgency to the character's desire to solve the puzzle. After the character opens the tank, lifts the toilet float, and fixes the stuck valve to stop the toilet running (alas, too late to prevent water flowing onto the floor), the game once again reprints the room description, this time altered considerably:

Bathroom

I'm now standing barefoot in some icy water. I've seen bigger bathrooms, but must admit this smaller one suits our needs well. In fact, at this very moment I'm terribly glad the floor isn't bigger, as it would only mean more water to remove. There's a small sink with a cabinet under it, a mirror, a bathtub (equipped with a shower head and curtain) and a toilet.

Right now, my attention is also grabbed by: my towel.

There are several changes, doing several different sorts of work within the description. The first, the "icy water" sentence, indicates a change in the room itself, one that is reflective of situation rather than character, though of course the way the character chooses to relate this situation—emphasizing discomfort by noting his bare feet and describing the water as "icy," conveying a mood of urgency without panic—does accomplish some characterization. A later sentence takes a fact of the bathroom addressed by earlier descriptions (its small size) and relates it to the new situation, revealing a practical and rather optimistic side to the viewpoint character. This sentence also demonstrates that the character's perspective, while pragmatic, is not particularly scientific, since a larger floor wouldn't actually mean more water to remove, only a greater surface area from which to remove it. Later, we get a new "attention" sentence; the toilet tank is no longer in focus, and instead the character is thinking of his towel. Note that this towel was not mentioned in any of the previous room descriptions, because the character had no particular

need of it. *Nothing More, Nothing Less* makes extensive use of this technique, heightening realism by filtering not only the general experience of landscape through the PC, but also specific points of focus as well. Finally, once the toilet is plunged and the water towed and mopped, the PC has showered, and his feline nemesis has entered the room, the bathroom's description changes to this:

Bathroom

This a bathroom, of which I've seen more than enough in the last minutes. Come to think of it, I've seen enough of it for the whole day. And the presence of that hairy pest doesn't improve my morale. Azrael licks one of his paws, while keeping an eye on me.

The character's perspective on the bathroom has changed once more, marking the end of his progression from bland interest, through urgent focus, and resting finally at mild exasperation. The emotional registers aren't extreme, but the room descriptions convey very clearly the changes taking place within the character as a response to the changes that occur around him. In games like this, landscape does even more shaping of character than usual by virtue of its changing prose.

A final aspect of how landscape reveals character lies in the concept of elision: what rooms does the game *avoid* describing, and how do those gaps influence our understanding of the character? Many games take the character, via non-interactive cutscenes, or even simple transitions, through landscape that we never get to see from the PC's perspective. My experiences as an author have taught me about this phenomenon; in my first game, *Wearing the Claw* (1996), I elided an entire sea voyage. In practical terms, I made this choice because I didn't have the time, energy, or skill to implement the journey as an interactive experience, but its absence from the game couldn't help but affect the PC's characterization. His reluctance to relate the details had to be explained somehow, so I made him someone who is deeply intimidated by the ocean, someone who would want to block out the experience of being at sea as much as possible:

Soon you find yourself at sea for the first time in your life, and you learn that the rocking and swaying of a small boat on a choppy sea does little to relax you. Nausea swells and recedes like the waves beneath you, and though the journey to the isle of the Goergs takes little more than an hour, it ends none too soon for you.

I'm not willing to make the claim that elision *always* contributes to characterization—sometimes cuts are in place just to serve a story's structure, leaving things unimplemented even though the character certainly would notice them. However, there are times that what isn't described is just as important as what is. These sorts of gaps are particularly noticeable when they contrast with the player's expectations, as happens from time to time in Kevin Wilson's *Once and Future* (1998). One particularly memorable absence in that game is the matter of the cat: late in the game, Frank Leandro (the PC) is required to obtain a bit of cat hair for a magical recipe and conveniently enough happens across a stray cat who sheds a bit into his hand and rides his shoulder for a while. A while later, that cat jumps into the chimney of a boarded-up house (chasing a bird) and disappears. Frank has a sword that cuts through anything, but the game forbids him from cutting through the boards to find the cat, saying "You could, but there's not much point to it." So however much the player may want to make sure that the kitty is okay, she is constrained by Frank's disinterest; the inside of the house isn't part of the map, because Frank doesn't see the point of exploring it.

A PC-centric View of Interactive Fiction

[This section contains minor spoilers for The Beetmonger's Journal, medium-level spoilers for Hamlet, and major spoilers for Photopia.]

It's possible that objections may arise to some of the points I make above, on the grounds that what I ascribe to character could just as easily be seen as a particular author's writing style, a game's depth of implementation, or even the formal constraints of IF itself. It's quite true that I'm taking a PC-centric view—this is how I experience interactive fiction, and it's easy to feel that it's simply how the form *works*, but I certainly acknowledge that there are other, equally valid approaches. It's also true that the PC is not the only possible point of view within a work of interactive fiction. In *The Beetmonger's Journal* (2001) by Scott Starkey, for instance, some very nifty POV-jumping occurs in sections where the PC is the hero of some stories being read by the frame characters—from time to time those characters are interrupted in their reading, and we get a small cutscene from their point of view.

However, what I would argue for is the extreme difficulty of disconnecting the point-of-view from the player character at the point of

action. The IF prompt implies a certain kind of remote control: the player is to type in an action that will then be executed within the game. Invariably, this action is performed by the PC. Indeed, this is the very definition of player character. Similarly, landscape descriptions, especially when that landscape is available for traversal and manipulation from the game prompt, almost cannot help but be filtered through the PC, because all the knowledge conveyed in them is available for use at the point of action. If room description were to convey something that the PC couldn't possibly know, such as the color of an object when the character is blind, the result would be severe cognitive dissonance for the player. If we type "OPEN BLUE DOOR" and the blind PC is able to do so, we must conclude that the PC is not blind after all—that's how powerful the connection is between character and action. Because landscape, character, and action are so intimately connected, it's quite difficult to avoid making landscape a function of character, especially as the two get nearer and nearer to action.

Given this PC-centric take on IF, it's worth asking what possibilities reveal themselves as open or closed in its light. We've already seen some of what's opened, from *Heroes'* cumulative place-building to *Shade's* resonant evocation of the familiar, and no doubt future games will continue to explore the power of the landscape-character axis. Conversely, one element that seems rather alarmingly curtailed is the possibility of dramatic irony. For instance, imagine Shakespeare's *Hamlet* as an IF game, in which the player controls Hamlet but is allowed (as a reader) to see Polonius stepping behind the arras in the queen's bedroom. In order to retain the dramatic irony of the scene, Hamlet must stab the arras and inadvertently kill Polonius, but why would the player order him to do so, knowing what Hamlet doesn't? In other words, how can the player be allowed to know things that the character doesn't if that knowledge is expected to facilitate dramatic irony? The only answer I can think of is to force the PC's actions, to make Hamlet stab Polonius no matter what the player orders, but as soon as that happens, the interactivity drops out of the IF game, and thus action is removed from the equation. I'm not prepared to contend that this sort of dramatic irony is impossible, but the game that solves this problem will be a major breakthrough.

The work that's probably come the closest to this grail is Adam Cadre's justly revered *Photopia* (1998). Thanks to its fragmentation of the narrative line and its array of POV characters, when the climactic scene

arrives, we know all the awful freight of what's about to happen. We also can't stop it—in order to achieve its dramatic irony, *Photopia* must remove our power to act. There's an argument to be made that this sort of moment becomes even more powerful in interactive fiction, the useless prompt underscoring the inevitability of the character's tragic fate. That's as may be, but it doesn't change the fact that PC and action are still inextricably connected, and the only way the PC can be made to do something inevitable is to remove control from the player. *Photopia* cleverly makes the inevitable moment a car accident, thus giving the PC only a split second to react (and thus providing a plausible context for lack of choice) and making his default desires identical to the player's desires (STOP THE CAR!), but in the final analysis, the moment is still achieved by removing control from the player, and indeed the great majority of the criticism directed towards *Photopia* has been of its non-interactivity.

If action is to retain its place in the IF triangle, landscape and character must remain inextricably connected. Their powerful bond to each other creates many exciting possibilities for the development of both, possibilities that have begun to be exploited in the last several years and that no doubt will continue to yield opportunities for development. What's also true is that noticing this connection and its potential still only scratches the surface of character development and landscape exposition in IF. Character can be revealed not just through landscape but through objects, plot, direct narrative, and many other devices. In turn, while character is the primary lens for landscape, that landscape can alter greatly from the passage of time, from plot events, from NPC actions, or hundreds of other vectors, and each change to character and landscape deepens both. We've only just started finding the techniques, and it's a heady feeling. We're at the beginning of an art form—there's much more undiscovered territory to explore.

Hint Development for IF

Lucian Smith

Hints

People get stuck in games with puzzles. People even get stuck in games without puzzles. Even the best game will find one player who, for whatever reason, doesn't get it. Maybe they are stuck thinking along a different logical track. Maybe they misread a line, or missed an exit. Maybe they don't know English well enough to correctly phrase their input. If the player is to continue and see the complete work, they will need a hint. And that's where you, the hint-writer, come in. The player, getting stuck, has been pulled out of the game. It is your job to get them back into it as quickly and as smoothly as possible.

This is an important and difficult task. If you are the author, two forces work against you. For one, you are often so close to the work that it can be hard to adopt the outside perspective needed to see where your prospective player is coming from. You know the streets, alleys, and footpaths of your neighborhood, but haven't a clue what freeway they should take. For another, the hint-writing process must, by necessity, come at the end stages of game development, but it requires the same amount of energy and creativity as the initial design. It can be hard to maintain that level of output for so long—fixing bugs is about as creative a task as one can manage.

The outside hint writer also faces a challenge opposite of the author's: you don't have an intimate knowledge of the game's ins and outs. In fact, you often start with no map at all, but merely the directions you followed to get to where you ended up. In some cases, you might not even be sure if where you ended up was where the author intended you to finish. To someone who did things differently or wants a different ending, your hints might not be able to steer them in the right direction.

The solution to this initial hurdle, whichever side you started on, is the same: seek outside advice. A single set of hints must apply to a wide variety of people; it must have a wide variety of inputs if it is to serve. The author has a natural group in their beta-testers—pay close attention to their reports to find out where they were stuck, and what path they

found to get them moving again. Painstakingly go over their transcripts to find even moments of temporary confusion—all could be places where another could get stuck.

Once you have completed your research and know the potential trouble spots, you must work to make your hints complete, adaptable, and inspiring. Complete in that they must both provide answers to each potential trouble spot as well as fully answer that problem. Adaptable in that they must meet the player at the level at which they are stuck—on the one hand they must not simply rehash what is already known, but on the other, they must not jump straight to the solution and “spoil” the puzzle. Inspiring in that they must prompt the player to return to playing the game and attempt the new solution you did not tell them, but which you made them think of. This last is by far your hardest task, and we shall explore artificial ways to accomplish it to supplement the inspiration you can muster on your own.

The most common (and, to my mind, most effective) method of accomplishing these goals is to follow the InvisiClues format. InvisiClues were hint booklets published by Infocom containing readable lists of hint topics, each followed by a list of progressively more explicit hints, all printed in an invisible ink that would appear only when you drew on it with an enclosed pen. Players would look up the problem on which they were stuck, then reveal the hints one at a time until they solved the puzzle, ran out of clues, or lost the pen. Though the invisible ink technology is sadly not used much today, the InvisiClues format of readable topics followed by hidden clues is widely used in a variety of implementations, from web pages to encoded files to a hint menu within the game itself. A survey of the available options and variations on this format is included at the end of this article, following an in-depth study of how to go about actually writing the hints themselves.

Complete

Being complete in your hints is probably the most crucial task you face. Nothing can be more frustrating to a game player than reading hints and still being unable to progress in the game. As the author, this might be harder than you imagine—bits you put in that you did not intend as a puzzle may be unclear or simply missed. Whether you are the author or not, a good method might be to compile a walkthrough of the game first

and note any commands that are not directions, “EXAMINE” or “GET” as potential places where people might need a hint. Even these are not always safe—in many games it seems I get stuck at least once when I have failed to notice an exit. Depending on the format of your hints, you might consider including maps of each section, noting the “obvious” places one can get stuck.

Based on your walkthrough and list of puzzles and trouble spots, compile the list of hint topics you want the player to see. Make this list as tangible as possible, and try to formulate it as a question the player might pose having seen the puzzle but not knowing anything about the solution. That means that if a diamond is hidden in a locked box, “Opening the locked box” or even just “The locked box” would be a suitable topic, while “Finding the diamond” would not. Even if the player knows they need a diamond for some other puzzle, they might not know that it rests inside. Likewise, you don’t want to give away anything in the topic. “Finding the key for the locked box” tells the player more than they know when they encounter the box for the first time.

Sometimes, there’s just no way to word a topic without that itself being some kind of a spoiler. (“Aha! One of the questions is ‘How do I get into the house?’ so there must be some way of doing that!”) The accepted method of dealing with this is to put bogus questions on the list as well as real ones. (“How do I get on top of the house?” or “How do I revive the skeleton?”) Infocom did this a lot in their *InvisiClues*. Don’t go overboard, though—too many false hints will leave the player wading through them, not realizing when they’ve hit a real one.

Once compiled, you will often find that the topics can naturally be grouped together somehow, often by geography (“The Forest”), sometimes by progress within the game (“The Prologue”). Grouping clues this way provides both a logical framework for looking up where one is stuck and a way to avoid “looking ahead” to see what sorts of challenges await. Ideally, a player reading hints will only see the topics for the particular section of the game for which they need help. This can be accomplished fairly easily for electronic hints, but if providing hints in some hard-copy format, be sure to include enough white space or page breaks between sections so that a player can, if they choose, only view the hint topics for where they are.

The other side of being complete is to be sure to provide explicit instructions for how to get past each puzzle. The simplest way to do this is to simply relate the commands the player is to type as the last hint in

each topic. So, after skating around the fact that the key is in the mouse hole, and after finally saying explicitly, “The key is in the mouse hole,” say, “>REACH IN MOUSE HOLE. You will get a silver key. Then, >UNLOCK BOX WITH SILVER KEY.” The reason for being explicit is that invariably some player will have the right idea but not be able to phrase it in a way that the game understands. Maybe they tried, “>PUT HAND IN MOUSE HOLE” or “>FEEL MOUSE HOLE” or even, “>GROPE ABOUT INSIDE THE MOUSE HOLE”, none of which worked. They can send a bug report to the author later—for now, they need to know what to do to solve the puzzle.

Adaptable

Before the player needs explicit instructions, however, they need gentle nudges in the right direction. And to do that, the hints need to be adaptable. The very format of InvisiClue-style hints caters to this need by gradually revealing more and more information as the player reveals each new hint. The following checklist should help you give exactly the right amount of information to the player.

1. Is the Puzzle Currently Solvable?

The player may feel that they are stuck, say, on “How do I unlock the grate?” but they will actually be stuck on “How do I enter the house?” inside which will be a skeleton key. The first clue given to a player should convey to the player whether or not the puzzle is solvable yet. As an example, let’s say that there’s a grate in the game that must be unlocked with a skeleton key. The first hint might be:

How Do I Open the Grate?

You’ll have to get in the house first. (see How Do I Enter the House?)

Sometimes it might be difficult to reveal this information without giving away too much. If a credit card can be obtained from a broken ATM and subsequently used to jimmy open a shed door, then telling the player looking for a shed door clue “You must first solve the ATM puzzle” reveals too much. Try to be circumspect in this situation, perhaps telling the player they must solve some puzzles in the bank area first. If there’s no way to tell the player this information without some spoilage, you can bump this information down the list and give the player other clues first.

If the hints are included in the game itself, they can bypass this step

by only providing hints for those puzzles that are currently solvable, or by stating “This puzzle cannot yet be solved.” Excruciating care must be taken to make sure this is actually the case! In my own game *The Edifice*, the hint system acknowledged only one way to solve a particular puzzle and got stuck there if the player used a slightly different method. Carefully check this code to not make the same mistake I did, and provide “outs” for the hints to continue in case it happens anyway (checking to see if a door is open, for example, rather than noting when the player types “open door”).

2. What Are the Clues within the Game?

A good puzzle should have indicators to its solution within the game itself. The next few clues should point the player in that direction. In our example, say the author has put a skeleton inside the grate as a clue. The next clue might therefore be:

How Do I Open the Grate?

You’ll have to get in the house first. (see How Do I Enter the House?)

Have you looked inside the grate?

Another type of in-game hint, often overlooked, is the non-working solution to the puzzle. This is anything the player might do to solve the puzzle that doesn’t resolve the situation but that the game acknowledges and responds to with some kind of pointer to the real solution. *So Far* (Andrew Plotkin, 1996) is an excellent example of a game with many puzzles like this, the animal puzzle in particular. In my hints for that game, I attempted to steer the player toward as many dead-ends as I could find, since each dead-end provided information one could use toward understanding the situation and eventually hitting upon the right solution. Indeed, I myself solved the animal puzzle by first failing to solve it in several different ways.

3. What Is the Logic Involved?

A good puzzle should be logical. The next clue or clues should try to get the player thinking along the lines of logic needed to solve the puzzle. This might be a good place to say explicitly what you hoped would be inferred from your pointers to the in-game clues. So, for example:

How Do I Open the Grate?

You'll have to get in the house first. (see How Do I Enter the House?)

Have you looked inside the grate?

The original key is probably lost.

But another key might be useful...

What kind of key is known to open many different locks?

Didn't the skeleton remind you of anything?

4. What Must Be Done to Solve the Puzzle?

If the hint writer has conceded that the author's thinking might be alien to the player, they should probably also concede that the hint writer's thinking might be alien to the player as well (especially if the hint writer is the author!). As mentioned before, explicit instructions should be included after the hints have otherwise been exhausted.

How Do I Open the Grate?

You'll have to get in the house first. (see How Do I Enter the House?)

Have you looked inside the grate?

The original key is probably lost.

But another key might be useful,...

What kind of key is known to open many different locks?

Didn't the skeleton remind you of anything?

A skeleton key!

UNLOCK GRATE WITH SKELETON KEY, then OPEN GRATE

It is traditional to give commands that should be entered directly as ALL CAPS, as above.

Inspiring

Hints are not an end in and of themselves. They are there to provide inspiration to the player to go back to the game and try something new. Every hint you give your player should be written to make them stop reading hints and go back to playing the game. Sadly, there are no easy tricks to accomplishing this in your writing, except to say take your time

and try to think about the problem from a variety of angles. Even the most inspiring clue may hit the reader at the wrong time and make them want to hurry on to the next clue. So, we cheat.

To cheat, we provide some artificial or external means of preventing the player from reading one hint right after another. Different techniques may be used, depending on the format of the hints (paper, electronic, or in-game) and on how many hoops you want the player to jump through before getting the next one. The motivation behind all of these techniques is the same: to slow down the player so they have time to think about the first hint before going on to the next, and to prevent the player from accidentally viewing hints they did not want.

Encryption

This option can take a wide variety of forms and be used in all three formats (paper, electronic, or in-game). Put simply, the hints are provided in a format that is not easily readable, and effort must be expended by the player to decipher each line. The amount of effort can vary greatly and should be kept in mind when writing the hints themselves. For a great amount of effort, players expect a greater degree of hintage, while for less effort, you can get away with including “filler” hints. A few examples:

Letter Substitution

Each letter in the clue is replaced by a different, unique letter. The simplest method for this is to use ROT-13, or substituting each letter by the letter 13 away from it in the alphabet. ROT-13 has the advantage of being reversible—A is substituted by N, and N by A.

Word Substitution

Each word in the clue is replaced by a number, and a booklet or file is provided with a list of words and numbers. Some games may provide in-game hints with just the numbers, with the booklet available along with registration, or for a price.

Sentence Substitution

Each sentence is replaced by a number, and a booklet or file is provided with a list of sentences and numbers. This method has the disadvantage of allowing a wandering eye to see answers to puzzles the player didn't

want to see; this can be allayed by mixing up the clues and including fake ones, too.

Text Mangling

The letters are rearranged or spaced oddly, perhaps by writing the clues backwards (“.pmuj .elbat no dnats”) or by inserting extra characters (“xsxtxaxnxdxoxnxtxaxbxxex.xjxuxmxpx”). This prevents accidental reading of the hint, is easy to produce, but only requires mental effort on the part of the player to understand.

Whichever method of encryption is used, it is probably best to write the clues in plain English first, then use or write a simple computer program to encrypt them, to ensure consistency and accuracy.

Physical Impediments

Game companies are most likely to use this method, since they usually involve a significant increase in cost than the simpler encryption methods above. Infocom’s InvisiClues are a famous example, though unused in recent years. A simpler method, popular with Sierra among others, is to print the clues in light blue against a red pixellated background, which could be read if covered with a piece of red cellophane. One problem here is that once you have the booklet and the cellophane, it can be too easy to slide down to the next hint, even by accident. One of the advantages of the Invisiclue system was the finality—once you exposed a hint, it was forever exposed on the pages of the book, perhaps giving the player enough pause to contemplate that last clue again and discover the answer for themselves.

Electronic “Encryption”

Once you move the hints to the computer, a wide variety of new options become available for use, in addition to those above (well, maybe not literal invisible ink). A typical implementation will allow the player to read the hint topics, then select one to go to the first hint, and continue selecting to go to successive hints. One advantage of this format is that it provides a way to easily hide from view graphical or even auditory clues. While it is possible to create your own format, here are two that will probably suffice:

HTML.

This is most often used for plain text clues with individual links to each clue, as Peter Scheyen's HTMLized InvisiClues at <http://www.csd.uwo.ca/Infocom/Invisiclues/>. My own hints for *So Far* used this same basic format, with the slight modification that links were only provided to the first hint for any topic, with subsequent hints available only through manually editing the location bar. The idea was to require more effort from the player, giving them more time to think about the previous hint. Another option is to simply print all the clues in white text on a white background, which the player can view by highlighting the appropriate section. The wandering eye problem becomes a wandering mouse problem for players with jerky wrists or sticky mouse pads, but it's a simple method that can be created with minimal effort.

UHS, or Universal Hint System

Currently online at <http://www.uhs-hints.com/>, this is a fairly popular multi-platform system that provides these types of hints in a simple electronic format, either from a standalone reader or online. The standalone reader is not free, and the online format is ad-supported, which is a mixed blessing—on one hand, not everyone will be able to access your hints; on the other, if you write hints for a popular game, you can actually get paid. Sending in your hints nets you a free version of the reader, at least, so if nothing else it's good for that.

In-game Options

Providing hints within the game itself allows a new level of options, probably the greatest of which is the ability to provide context-sensitive hints. This means a lot more work for the hint writer but a better experience for the player. While the most popular in-game hint systems simply provide a way to get to a menu of hint topics, there are other systems that might be appropriate for your game.

One Hint per Room

In each room, the player can ask for a hint and get a clue about something to do in that room or about something related to that room. Probably not the best option for complex games, but it has the advantage of being automatically context-sensitive

One Hint per Object

The player can ask for a hint about an object they possess or have encountered. Again perhaps not appropriate for some games, but it might work well for others

Hintus ex Machina

Some object, character, or device within the game itself can give clues to the PC. *Curses* (Graham Nelson, 1994) and *The Magic Toyshop* (Gareth Rees, 1995) both had characters the PC could show objects to and get clues from, my own *The Edifice* had special rooms where hints would appear etched into the wall, and Infocom's *Arthur* (Bob Bates, Infocom, 1989) had a toque you could peer into to get hints. This method can have the advantage of furthering the atmosphere of the game, and players tend to be more forgiving of hint restrictions placed on the PC than restrictions placed upon themselves.

Time-delay

Theoretically available for external electronic formats too, users of this option simply force the player to wait to receive the next hint, either by waiting a certain number of turns or by tying into the system clock and forcing a delay of an actual period of time (though I personally have never seen the latter and imagine it might get annoying rather quickly). This option is most often used with the 'hintus ex machina' (as in *Arthur* and *The Edifice*) but would theoretically also be an option for menu-based hints.

The Penalty

The player is penalized in some way (usually a decrease in score) for getting hints. This technique has been around since an early version of *Adventure*, which allowed the player to get a hint at the cost of a point, and has been used over the years with mixed success. (Occasionally the reverse is seen—the player is awarded an extra point if they make it through the game without using any hints.) The problems with this method are that it is easy to get around through saving and restoring and, perhaps more importantly, that it furthers the breakdown of trust between the player and the author. The author has already betrayed the player by creating a puzzle too hard for the player, and deducting points cements this division, rather than gently cajoling the player back to the game. I include the option here for completeness but strongly

recommend against it.

No Explicit Hints

The one sure-fire method of keeping the next hint from the player is to never include it in the first place. *Theatre* (Brendon Wyber, 1995), for example, while providing a number of hints for each topic, never explicitly told you the command to type, instead ending on indirect clues like “You need to have a girlfriend called Jane” when Tarzan-like activity was required. The NPCs in *Curses* and *The Magic Toyshop* likewise never told the PC explicitly what to do. Be aware that by using this option, you will probably force some players to use some external resource (probably a walkthrough) to solve their problem, and once there, they may never return.

Letting the Player Decide

Rather than having to choose among the above options, in-game hints (and, theoretically, electronic hints), can leave the format up to the player. At the most basic level, many Infocom games that came with hints allowed you to type “hints off,” which would disable the hints for the rest of that game session. This method is somewhat ham-fisted, however, and easily gotten around by simply restarting the game for the express purpose of reading the hints. A middle-of-the-road solution was utilized in *The Legend Lives!* (David Baggett, 1994), where the user could request encrypted hints, which would pass all the hints (and, perhaps unfortunately, hint topics) through a ROT-13 filter. Any number of the above options could be used in this way, and it might be a good way to include the “no explicit hints” option without alienating the player.

>

When written and utilized well, the result of a good hint system is that the player finds out how to get past were they were stuck with exactly the right amount of a “push” and is able to go on and enjoy the game. They still feel challenged and can complete the game without the sense of loss that goes with either giving up on the game altogether or from being “spoiled” for that puzzle. (See “Cheat, Beg, Wheedle, Cajole” from *XYZZYnews* #12, at <http://www.xyzzynews.com/xyzzyn12f.html>.) I received the greatest validation of my own hint writing efforts when I read a post from someone who said they had just finished *So Far*

“without cheating” and with only a “gentle nudge” from my clues. Hint writing can take a lot of effort, but the players will thank you for it.

Descriptions Constructed

Stephen Granade

A work of interactive fiction is a web of descriptions. Nearly every object and room in a work has a description, a block of text associated with that object that is printed after a LOOK or EXAMINE command.

The word “description” is a lie, though, or at the very least misleading. The main purpose of a description may not be to describe an object. At the most basic level, descriptions are the primary method by which an author conveys information about a game world.

Authors don’t have to reveal much about an object or room at all in a description. They can overtly or subtly mislead players. Elements of an object can go undescribed in order to emphasize other elements. An author can write a description that lies, if the narrator telling the story is unreliable. Descriptions can also carry more information about an object than the physical: what mood the room or object evokes, the state of the game world, and even meta-game information.

Letting go of the preconception that descriptions must describe frees authors to achieve any number of effects. This hasn’t been discussed in depth before, despite the sizable amount of extant advice on writing descriptions, especially for rooms. Graham Nelson, in “The Craft of Adventure” (later subsumed into *The Inform Designer’s Manual*), explores the art of writing room descriptions. He focuses on physical descriptions, a relaying of what the room looks like, as well as how you provide the player with directional information. Similarly, in *Writing Basic Adventure Programs for the TRS-80*, Frank DaCosta gives five guidelines for writing room descriptions, four of which are generally applicable: include pathway hints, use non-oriented language (i.e., avoid language that depends on how you entered the room, such as “You fall into a dark, slimy pit”), avoid describing unimplemented objects, and use creative descriptions. In both, the assumption is that descriptions describe. This assumption is so self-evident that neither author discusses how you describe objects, since you’ll just be talking about what the object looks like.

I’m interested in the things you can do with a description besides describe. If you’re looking for a primer on how to describe objects, look elsewhere, preferably at how games you like handle this task. Throughout I’ll be speaking to authors, though players may find this

article interesting. I won't be talking much about the *art* of description-writing. I'm taking a nuts-and-bolts approach to the task: if I have three goals for a description, how do I go about assembling sentences to achieve those goals? Finally, I'll continue to call the text associated with objects and rooms "descriptions," but keep in mind that a description doesn't necessarily have to describe.

Rather than pull examples from other games, I'm going to invent descriptions for a completely new game, *Quod Erat*. In it, the player is a tourist visiting the ruins of Ostia, once a thriving harbor city some 20 miles from Rome. While there, the player will be thrown back in time to when Ostia was a living city.

Let's start with mood and atmosphere. How you describe is as important as what you describe. Choice of words and phrasing, the use of literary techniques such as metaphor, assonance, and juxtaposition of images, and what you emphasize in your descriptions have as much of an effect on the atmosphere you invoke as Monet's choice of colors did on his series of paintings of the Rouen Cathedral.

In our game, the first location the player will reach is the tombs just outside of Ostia. I want to emphasize the emptiness of the ruins as a contrast to later in the game, when the player sees Ostia in its prime. I'm not above using cheap dramatic tricks to get my point across—good thing I'm discussing craft and not art.

Tombs Outside Ostia

Walls, or the remains of them, box you in on all sides. Wind whistles through the head-height niches in some of the walls. Underneath your feet loose stones crunch. The southwest corner where the walls meet is low enough and the dirt in the corner piled high enough that you could scramble out.

I've pulled out every clichéd trick in the book. I have wind whistling through empty openings, I have crunching stones underfoot, and I have a total lack of life. Players will undoubtedly look at the niches, so I'll re-emphasize the dead nature of Ostia. I'll also remind them that they're tourists.

>EXAMINE THE NICHES

Burial niches, according to the poorly photocopied sheet you read back in the Welcome Center. They are just deeper than you can reach, and all are empty.

One of the main locations in Ostia is the Capitolium, an ancient temple dedicated to the Capitoline triad of Jupiter, Juno, and Minerva. The Capitolium is an imposing edifice, despite its lack of a roof or front wall. As above, I want to mention the ruins of Ostia and the general disrepair of the Capitolium.

In Front of a Brick Shell

To the north rises a shell of reddish brick. It's twice as wide as it is deep or tall, and it's rather tall. The wall in front of you is missing, giving you room to enter the shell. The whole thing is on top of a tall podium that has stairs on your side. Ruins are on all other sides, though most of them look to be to the south.

I didn't call it the Capitolium, because the player doesn't know that's what it is yet. This segues nicely into my second point: descriptions don't have to be static. They can reflect the increasing knowledge of the player or the changing state of the game world. In the case of *Quod Erat*, the player will eventually find a tourbook and map, the better to learn about Ostia. Once they find the tourbook and return to the Capitolium, they'll see the following:

In Front of the Capitolium

According to your tourbook, the rectangular shell of reddish-brown brick rising in front of you is the Capitolium, dating back to Hadrian's reign. The shell is over fifty feet high and deep, and twice as wide. It stands atop a tall podium; luckily there are stairs leading up to the Capitolium. The wall in front of you to the north is missing, giving you room to enter the former temple. The forum stretches out on all other sides, the bulk of it lying to the south.

I would limit how much game-critical information is in a changed room description. Players tend to skim room descriptions after they've read them one or two times.

An object's description can change as the player learns more about it as well. A common trick is to describe an object in sketchy details when listed in a room description, only to have the object's description tell the player what it really is. The EXAMINE command is commonly assumed to mean, "I want to take a close look at the object." It's thus reasonable for what is described as a piece of paper in a room description to turn out to be a grimy dollar bill when examined.

The coin that will take the player back into the past begins as a dirt-

covered nub on the floor of a room.

Base of the Western Spiral Staircase

This narrow niche off the *cella* ends to the west in a spiral staircase winding around a thick column. The floor of the niche has been swept smooth by countless feet, except for raised nubs of dirt near the north wall. The *cella* itself is to the east.

>EXAMINE THE NUBS OF DIRT

For the most part, rocks covered in dirt. There is a glint from one of them.

>EXAMINE THE GLINT

Something disc-shaped and covered in dirt. Metal shows through the dirt in places.

The glint serves to tell the player that there's something special about one of the nubs of dirt. The hint of metal suggests that the disc should be cleaned. This is important, since the coin is activated by rubbing it. When the dirt is cleaned off, it is revealed to be a coin.

>EXAMINE THE DISC

Now that you've cleaned off the dirt, you can see that the disc is a coin. One side is badly damaged and nearly smooth, but the other side has a picture of a man on a chariot holding a spear aloft.

And to old Ostia the player goes! Mentioning the results of an action in an object description is a good way to acknowledge the player's actions but can become tiresome if they carry the object throughout the remainder of the game. After the first time they examine the coin, I'll strike the first sentence and add the phrase "of the coin" to the remaining sentence, so that the description will begin, "One side of the coin is badly damaged . . ."

I'm glossing over a number of game details, of course, like what synonyms for "nub(s) of dirt" or "glint" the game should recognize. Since I'm worried about descriptions and not general game design, I'll merely wave my hands about furiously and assume we'll get those details right.

So much for evolving player knowledge; what about an evolving game world? In Infocom's *Enchanter* (Marc Blank and Dave Lebling, Infocom, 1983), the room descriptions change as the game progresses,

describing a castle that decays further with each passing day. The change I have in mind for *Quod Erat* is even more dramatic. The player can, by mucking about in the past, end up destroying the Capitolium. When the player does so and returns to the present, the description of the Capitolium has changed.

In Front of the Capitolium Podium

The podium where the Capitolium used to be rises in front of you. Marble stairs lead northward to its top, where a few of the Capitolium walls, none more than two feet high, still stand. The forum stretches out to all sides of you.

After the very first description in this article, I mentioned that players will undoubtedly look at the niches. When you mention things in room descriptions, players will often examine them. The more prominently you mention objects, or the odder you make their description within the room description, the more notice they'll draw.

You can use this fact to direct players' attention and lead them in the direction you want them to go. For instance, once in old Ostia, the player will come across a Roman terra-cotta lamp. This lamp will serve a two-fold purpose. One, it will provide light. Two, it will provide a source of heat. (It will also provide fire, always a dangerous substance in interactive fiction. I'm sure the game will deal with it properly, hand wave, hand wave, hand wave.) The heat from the lamp will be necessary to soften the wax stopper of a bottle later on.

Since players won't necessarily connect heat with light, I'll mention it explicitly so as to direct players and help them follow my lead. When reading a paragraph, people remember best what's in the first sentence, followed by what's in the last sentence. For this reason I'm going to mention the heat in the first sentence. Don't overuse this effect! Players may figure out that you always put important details in the first or last sentence if you don't vary your technique. Save the first-or-last-sentence trick for situations where you want to give a stronger-than-normal clue.

>EXAMINE THE LAMP

Heat rises from the lit wick of the terra-cotta lamp. The lamp is teapot-shaped, with a circular handle on one end and the wick emerging from the spout on the other end. Where the convex lid of a teapot would be is a concave top inscribed with a battle scene.

Similarly, if there's something important in a room that a player needs to

examine, you can make sure to mention it in such a way as to set it apart from the rest of the room description. For instance, in the builder's guild courtyard, there are some marble columns that contain clues to a puzzle. I want to make sure players notice them.

Courtyard of the Builder's Guild

The courtyard, surrounded by the builder's guild, is a rectangle with its narrow ends to the north and south. The entire courtyard is surrounded by a porticus with brick piers. The north and south sides have squat marble columns that differ from all the others you've seen. Leaning in one corner of the courtyard is a statue.

The phrase "differ from all the others" is meant to catch the player's attention: hey, here's something out of the ordinary. This effect can also be used as a stage magician uses patter and fast hand movements: to distract the player for a while. There is a bathhouse in Ostia, and one of the rooms with a basin has a trap door that leads to a tunnel below. The room's floor is covered with mosaics, except where the trap door is. If the player examines the gap in the mosaic, or even the mosaic or floor itself, the trap door will be described.

To delay that event, I'm going to make one of the architectural details of the room more prominent. I'll also make sure to mention the basin in every sentence and only mention the mosaic as a lead-in to describing the basin.

Northern Cold Basin

Geometric designs picked out in black-and-white mosaics run nearly all the way around the basin in the center of the room. All three walls of the basin have niches; from them, water jets from a pipe. The west niche seems deeper than the other three. Tall columns separate the basin from the frigidarium.

I said I wouldn't discuss how you deal with room exits in room descriptions, but I lied just the tiniest bit. There is a south exit from the room that I didn't describe because the only initial way into the room is through that exit.

Object descriptions can carry meta-game information. There are times when using an object requires special syntax that isn't easily guessed. The stone-placing game in *Quod Erat* is one such object. Players must place marbles on a rectangular grid.

>EXAMINE THE BOARD

The wooden playing board has hemispherical depressions arranged in a six-by-five grid. Each depression is just big enough to hold one of the marbles from the trough beside the board.

[To place a marble in a specific depression, type >PUT MARBLE IN X, Y where X, Y are the x - and y -coordinates on the board.]

It's customary to set such meta-game information off from the regular description by surrounding it with square braces.

While descriptions customarily give physical information, they don't have to. You can describe without really describing by depending on what the player knows from real life.

>EXAMINE THE PEBBLE

Pebbles in the past are just like pebbles in the present, only newer.

Sometimes what you want to describe isn't really visible. At one point in *Quod Erat*, a mouse runs across the floor of the room and into a mousehole. An attempt to examine the mouse will result in

>EXAMINE THE MOUSE

You can see nothing of it, as fast as it ran across the floor and into its mousehole. There is the hint of light glinting off eyes deep within the hole.

Sometimes you'll want to use descriptions to indicate the player's state rather than that of the world. In old Ostia the player can find a bottle of drugged liquor. The player will, of course, drink from it, despite the inadvisability of such an act. Shortly after that occurs, the player's grip on reality will slip rather badly. The resulting room description when it first occurs will bear little resemblance to the actual room description when the player isn't drugged.

In a Gray Fog

You are surrounded by a gray fog that hides most everything. Indistinct shapes surround you, some near, some far. Thinner patches of fog lead south and southwest.

The directions in the description will correspond to actual room exits for whatever room the player is in, though subsequent movement will take the player into uncharted territory.

I've given a number of examples of what you can do with descriptions besides describe. They can set the tone and atmosphere of parts of the game. They can reflect the changing nature of player knowledge and the state of the game world. They can both lead and mislead players, letting you choose in part where the players will focus their attention. They can carry meta-game information, and they can even carry no information about the true physical nature of the object they purport to describe.

The above list is not exhaustive. It's meant to serve as a starting point, not an ending one. Nor is it meant to suggest that all descriptions must do more than describe. Descriptions are meant to convey information in one form or another, and most of the time that information will be the physical description of the object in question. Nevertheless, knowing some of what descriptions can do will hopefully give you ideas of things to do in your own games.

Mapping the Tale

Scene Description in IF

J. Robinson Wheeler

The more accurate the map, the more it resembles the territory. The most accurate map possible would be the territory.

The tale is the map that is the territory.

You must remember this.

—Neil Gaiman, *American Gods*

Intrinsic to interactive fiction at its inception was the simulation of location, of giving the player the ability to move from place to place. This illusion is achieved, simply and efficiently, by strings of text, location names and scene descriptions, and by a map of connections between the objects that print these strings. When the player surrenders his sense of place to the game's map, he responds to a new bit of text by imagining he has walked (or crawled, or slid, or climbed) somewhere he hasn't been before and is taking in all of the details. If he recognizes a bit of text, he knows he has re-entered familiar territory.

The basic requirements of scene descriptions go no further than this. If you let the player know where he is, and where the exits are, your job is done. This article, however, will provide many examples of scene descriptions that first meet and then, in some way, exceed these requirements. If your IF game has a story, your scene descriptions should serve the story. If you want to give your game atmosphere, broaden your use of sense details within your scene descriptions to include sounds, smells, and even textures, instead of just sights. If you want your game to have a strong narrative voice, scene descriptions are a good place to establish it. If you are using a well-defined PC, the scene descriptions can be used to reinforce your protagonist's point of view of each location. If your game has a complicated backstory, scene descriptions can provide expositional as well as locational detail. If the pace of the game quickens, scene descriptions should keep pace, becoming briefer, more active, even changing from turn to turn to sustain the player's feeling of urgency.

Caves

In the beginning, Will Crowther's original *Advent* (1972), the basic map-and-scene-description system we still use today was invented and put to use in a simulated exploration of Bedquilt Cave. The scene descriptions were evocations of actual locations, and the location names were encapsulations of these evocations. Some of them were fictional, serving to create the ambience of adventure rather than the pure simulation of caving, as in the following example:

In Hall of Mists

You are at one end of a vast hall stretching forward out of sight to the west. There are openings to either side. Nearby, a wide stone staircase leads downward. The hall is filled with wisps of white mist swaying to and fro almost as if alive. A cold wind blows up the staircase. There is a passage at the top of a dome behind you.

Rough stone steps lead up the dome.

From a sense of nostalgia, or an interest in re-exploring well-trod IF territory for one's own creative satisfaction, many modern authors continue to use caves as a setting, even knowing full well that any cave descriptions are going to be compared to what's already been done (and done many times, as IF in its first decade was dedicated almost exclusively to *Advent* derivatives). The evocative potential of cave descriptions seems not to have run dry, but new authors should proceed with care; an IF trope, even if intended as homage, may be perceived as an IF cliché instead.

Base of Canyon

You are in an alcove, a side chamber at the base of a vaulting canyon. The walls spread upward around you. And the distant roof is hung with glimmering stars -- droplet-tipped stalactites in some hidden suffusion of light. A crevice runs along the canyon at your feet. You can cross it and continue on ahead, or re-enter the crawl behind you.

A pool of thick, dark, reeking blood is spilled across the ground. Bats crawl all around it -- rustling, fluttering, feasting.

The above example is from Andrew Plotkin's *Hunter, in Darkness* (1999), in which he revisits not just *Advent* but an equally early game, *Hunt the Wumpus* (Gregory Yob, 1972), which is usually considered an IF precedent but not IF. *Wumpus* does feature a small map of locations and the ability to walk or shoot crooked arrows through the connecting passages. Here, in a scene late in the game, Plotkin follows the conventions of scene descriptions but reaches for a bit more atmosphere. After the matter-of-fact location name, the description itself starts with "You are in . . .," a tradition that is still perfectly acceptable. Note, though, that it does not read, "You are in the base of a canyon," nor is the location name "Alcove." The disjoint adds to the sense of size and space, of the player taking in the proportions of the canyon as he steps into it. Before this location, the game has squeezed us through a succession of claustrophobic rooms and tunnels; here, there is space, vaulting up around us as we peer out of the alcove.

The next sentence is a blunt declaration, "The walls spread upward around you." This simple sentence serves two functions: repetition for impact ("vaulting canyon" carries the same information) and the directing of the player's eyes upward, following the canyon walls up. The next sentence starts with "And," which is a stylistic choice that will bother grammar pedants, but it works in this context. The player's eyes then switch abruptly from the lofty beauty of the ceiling to the crevice in the floor, the mention of which segues into the convention of listing the available exits from the room. Here, Plotkin avoids having to mention compass directions but still points the player at appropriate exit actions: either going back or going ahead, across the crevice. There is a certain momentum built into the scene description, backed up by the momentum of the plot that has accrued by the time the player reaches this location, that points across the crevice and to whatever lies beyond. The player has reached the climax of the game, and the scene description is crafted not merely to describe where he's standing but to keep the tension building.

Shore of An Underground Lake

A narrow ledge of solid rock at the southern end of a great cavern. Beyond it lies a body of water so flat, so black and tranquil, that it might be a surface of polished obsidian. Embedded in the wall, a mirror reflects your movements: an odd smoothness in the unshaped stone.

A heavy bronze bell hangs from a stand.

This example, from the beginning of Emily Short's *Metamorphoses* (2000), is highly economical and strongly styled. The location name itself conjures the broad picture, and then the description text ticks off details to flesh it out and build the atmosphere. Missing is "You are on . . .," leaving a chiseled sentence fragment as the starting point. This is a stylistic trait of this particular author, but it works especially well here. Scenes before this one have already established an identity for the player character and a tightly focused narrative point of view, hers. Omitting "You are on . . ." preserves the game's voice and introduces a note of tension that is never resolved.

Know the territory, so that you can go somewhere new. To paraphrase Andrew Plotkin, if you want to write a cave, go visit a cave. In addition, play some IF with cave descriptions and study them before you try writing your own.

Here is an example that appears in *Being Andrew Plotkin* (Wheeler, 2000), and it is an agglomeration of shorter descriptions written by Andrew Plotkin himself, taken from various pages of Plotkin's personal website (which uses cave exploration as an elaborate metaphor for site navigation). These descriptions predate *Hunter, in Darkness*, but the stylistic similarities are recognizable:

Tunnels (as *Peter and Valerie*)

A huge cavern rises above you. The far reaches are lost in shadow mist, and the vaults above fade into a darkness pierced with long columns of stone. Chill water drips and pools in broken declivities. You can hear little else. A river of smoke-grey travertine flows down one wall of the chamber. Pale fungi gleam in phosphorescence somewhere above. Gloomy side chambers stretch off in many directions.

What comes out of this scene description is a sense of contour and shape: the declivities that pool with water, the flowing of the travertine down a wall, chambers and pockets, flat darkness pierced by rounded stone. Of equal importance is the sense of location provided by aural descriptors: chill drips and "little else," implying perhaps a cavernous echo you can hear in your mind.

The pieced-together nature of the description is evident, unfortunately; the ceiling is described as being lost in mist, fading into darkness, and hiding phosphorescent fungi all at once. The attention to

where the player's eyes are looking, evident in the example from *Hunter*, is missing here; instead of a progression from eye-level to ceiling to floor, we look up, then we see pools of water, then we listen, then we look at the wall, then we notice a light source whose location we aren't even sure of, then a collection of exits. A certain precision is perhaps missing; authors should endeavor to better this example.

>

It is worth noting that scene descriptions often are followed by a sentence or a short paragraph of supplementary scenery, leaving the player with a final detail in the form of an obvious object to interact with.

In the example from *Metamorphoses*, the player's eyes fall last on a heavy bronze bell. Note that the mirror embedded in the wall is described in the main scene description, leaving the bell to stand starkly alone. If we have studied our Classics, we might know already that ringing the bell will help us to cross the lake, but the author is giving all players a giant push in the right direction.

In *Hunter*, above, the gruesome supplement is at first encounter merely atmospheric, but it later reveals itself to be as important to the player's progress and, in the same way, as worth directing the player's attention to as is the heavy bronze bell in the *Metamorphoses* example.

In the example from *Advent*, the supplement is another room exit. The player will eventually discover that this is because the steps that lead up are an obstacle to carrying out a particularly heavy item. The steps are thus part of a puzzle, justifying the emphasis that is placed on them by listing them separately.

In some IF languages, authors will write these supplementary descriptions in a separate object from the room object that prints the main description, but the author should conceive them to be part of the one scene description for the room. Separate the most important ones, and leave out what does not need to be described, even if creating scenery objects is your idea of fun.

>

White Houses

Let us leave Bedquilt Cave behind and visit an equivalently nostalgic setting from IF history. This, one of the most well-remembered and often-quoted scene descriptions in IF, the starting location of *Zork* (Blank, Lebling, and Anderson, 1979), is bafflingly minimal and inert:

West of House

You are standing in an open field west of a white house, with a boarded front door.

There is a small mailbox here.

What is memorable about this? Perhaps that it is iconic, starkly so; “a white house with a boarded front door” is easily and quickly drawn in the mind, drawn and then stored. Trying further to explain its appeal would be pointless and grasping. There simply isn’t much there, which becomes clear as we walk all the way around the house and view equally slim scene descriptions:

West of House

You are standing in an open field west of a white house, with a boarded front door.

There is a small mailbox here.

>SOUTH

South of House

You are facing the south side of a white house. There is no door here, and all the windows are boarded.

>EAST

Behind House

You are behind the white house. A path leads into the forest to the east. In one corner of the house there is a small window which is slightly ajar.

With dark wit and a bit of daring, Adam Cadre’s *Shrapnel* (2000) riffs on our expectations of this familiar location:

West of the house

You are standing in an open field west of a white house with a boarded front door. Towering Carolina pines loom all around this clearing, silhouetted in the dusk; soon it will be night.

Three snarling attack dogs fight with one another over the remains of your corpse.

>S

You walk around to the south side of the house...

South of the house

You are standing south of a white house. There is no door here, and all the windows are boarded. However, one window upstairs is unevenly boarded, and you can see light shining through a crack between two boards.

>E

You walk around to the back of the house...

Behind the house

You are behind the white house, where the husks of the other buildings on the estate stand: the stables, carriage house, outhouse, niggers' quarters, all are just collapsed heaps of weathered wood now. The house itself stands to the west, the back door slightly ajar.

Starting from the scene descriptions found in *Zork*, Cadre has embroidered them with just enough extra detail to evoke an actual setting, a sense of time, place, and character that are missing from the original. Choose your details carefully, because they can tell a whole story.

Minimalist scene descriptions like those above are no longer common to any games but those of the speed-IF variety and the occasional large-scale IF, where they are the fatigued work of an author trying to bully through to the end of a long project. From *First Things First* (Wheeler, 2001):

Laura's office

Laura's office is a small, windowless room with a file cabinet and a plain white desk. The only exit is to the east.

If you find yourself writing a scene description like this one, drink some strong coffee and try again.

Secondary Sense Descriptions

Sight is the natural sense that we cater to in a scene description, answering the player's question "What can I see here?", but secondary sense details can be useful tools for IF authors.

Darkness

It is too dark to see.

A rattling, rustling hiss courses back and forth here. It reminds you of... what? Perhaps the sound of the wind in the dry grassy plains, back wherever they are. If so, the wind itself is a distant high wail to the northwest; a deep rumble is barely detectable to the west; and a cacophony of shrieks and brassy calls pours from the northeast.

A singing, thrumming chord hangs above you as well.

A cheery bubbling sound surrounds you.

The chord is getting louder.

This scene description from *So Far* (Plotkin, 1996) completely dispenses with sight, replacing it with sound. Note how many different kinds of sounds are evoked, and that the final sentence, "The chord is getting louder," is used to create suspense.

Full sense description doesn't always matter to a scene, and the appropriate time to use it is when it does. In the earlier example from *Being Andrew Plotkin*, the player is told what it sounds like to be standing in the cave, which adds an appropriate extra dimension and some atmosphere, but it is not deployed with the aim of conjuring a specific effect. Here is a scene description from the same game that makes use of tactile descriptions instead of aural:

Weird Tunnel

You are on your hands and knees in a claustrophobic tunnel. It leads down, as nearly as you can tell, but your equilibrium is distorted in here. The curving walls gleam with the semblance of wet rock, but the palms of your hands tell a different story. It feels like organic tissue, a thick layer of hide, with the elastic strength of muscle. Like sharkskin, it is silken and slick in one direction but resists any

backwards movement with a roughened grain.

Here, sight is given less attention than the sense of touch; it's not what the place looks like that makes it weird, it's how it feels on your bare palms. The direction of the room's exit, down, appears at the top of the scene description instead of at the tail, but the fact that there is only one way to go is underscored by what does come at the end, the sharkskin feel of the floor. The description gives the player the unsettling impression that he is sliding down a gullet rather than doing some casual spelunking. "This is no cave," as Han Solo once put it.

Smells, too, can be the right detail to add in certain locations, as in this example from *The Tale of the Kissing Bandit* (Wheeler, 2001):

Garden park

The luscious gardens, abloom in the early spring, send forth their heady aromas on the nuzzling breezes. Aloft, too, are the erotic perfumes of the finely dressed ladies in white, walking two by two in the early eve, innocent as does, poised as summer swans.

Active Scene Descriptions

Scene descriptions can move even further away from being passive scenery, instead serving as active instruments of narration. They describe this present moment of the story, what is happening instead of what is visible. Here is an example from *First Things First*:

Hanging for dear life

You are hanging for dear life, with your body slipping around the trunk of the tree. Soon you'll be upside down, trying to support your weight with atrophied muscles, with fingers that can't find a grip on waterlogged bark.

You feel your body shift again.

Aiding the author in the transition to this more active style of scene description is the use of linking text. In the typical case, IF games move the player from room to room directly, eliding the short journey in between. Transitional text is used for special cases, where how you got there matters, and it makes sense to think of them as part of the scene

descriptions that follow them. Two examples, once again from *Hunter, in Darkness* and *Being Andrew Plotkin*, respectively, illustrate this technique:

>FORWARD

You continue. Scrabble with fingers, brace arms, push with toes. You're definitely tilted head-down now. Is the stone pressing more tightly to either side of you?

Tight Crawl

You lie on your side, gasping, trying to recover some strength. Your back is sore; your neck is worse. Your head aches from cracking into unexpected stone. A cold stony knot presses into your left side, your right knee. You stretch forward once more, feeling for a few more inches before you continue.

Splash.

Your fingers are submerged in water.

--

>OUT

You experience blind panic for the first time in your life, despite the absurdity of the setting, and the recognizable fiction of the threat. Your life; how small and petty it all seems, and how short. There is little light in these twisting tunnels, and the faster you run, the more time moves in slow motion, and your thoughts turn inward. Your life has been blind and short, you think; appropriate, because down this last dark turn of the cave tunnel you have met with an unexpected, dead end.

Dead End

It is the classic dead end. More than that, it is the archetypal dead end. Nowhere forward, nowhere back. The ceiling shocks with its height, the walls oppress with their closeness. There are no exits.

A helix of light hangs in the air here, twirling and changing colors.

"We're trapped!" whispers Valerie, as much color drained from her face as there is pulsating in the helix above.

Authors should be aware that certain effects work only once and that a scene description written to take advantage of a strong but one-time lead-in may look awkward all other times the player reads it. The above example from *Hunter* is somewhat guilty of this, but Plotkin does replace the supplementary text; the one-time-only moment of your fingers splashing into water changes, on repeated looks, to:

A trickle of water runs past you, and merges into a shallow pool at your fingertips.

In the following example from *Christminster* (Gareth Rees, 1995), the player is ascribed an overly generous respectfulness, in that she bows her head in memory of the honored dead every time she passes through:

Archway between the Courts

A long, low stone archway with First Court to the west, and a door to Second Court to the east. On the north wall is a war memorial, a series of simple slate tablets listing the names of the members of Biblioll College, undergraduate and graduate, who died in the First World War of 1914-1918. You bow your head for a moment in memory of these young men and what they might have become.

This is passable, and perhaps forgivable if one takes into account that "Brief" was still the default mode for IF games at the time, and the author may have assumed that most players would only see it the one time.

More notorious is an example from *First Things First*, in which a location (to which the player repeatedly returns) sports a scene description that is only appropriate once:

Inside house

You finally step into your house. Not much of a victory, given the circumstances. Just a lot of planks and rafters. A temporary work ladder is nailed into the structure, and leads up to the second floor.

Ignoring this scene description's other faults, we see that the same basic problem is on view in *The Mulldoon Legacy* (Jon Ingold, 1999):

>DOWN

(down a short flight of stairs)

Outside the Botanical Room

Success! After descending a few steps northwards on the catwalk you have slipped into a darker archway in the middle of the cavern and are now standing face to face with the Botanical Room door to the north. Typically, all you have to do now is work out how it opens, as there is no window you could look through to check if the plants are still alive.

The Botanical Room door is closed and locked.

Note also the parenthetical transition text, which appears on every trip to this location even though it is not important. The main problem with this scene description is the author's intrusiveness, a problem often expressed as telling instead of showing.

The same author continues to write scene descriptions in a similar vein but has learned better how to get away with it, by quickening the pace of his stories, so that a one-use scene description is likely not to be accessed more than once. Here, in *All Roads* (Ingold, 2001), his scene descriptions are clearly functioning as narrative:

Scaffold in the City Square

Your head is pulled back, held by the rope pressing on your throat. Your toes pivot on a rickety stool, which shakes as your legs shake. The crowd filling the square are chattering like monkeys, but they are just flits of colour in the corners of your eyes. You cannot look round - instead you gaze straight across the Square, to the great dial of the Clock.

Slowly, the hand of that great dial swings lower. You are going to hang.

--

Venetian Streets

You walk through streets and winding alleys, avoiding the canals where the crowds will be. The Captain has one hand on the manacles around your hands, and the sharp tip of metal is ever present on your back. Tall buildings rise impassive on either side, cutting out

the sun, and you wonder if you shall ever see it again.

But though it is dark, it is never dark enough for you to leave.

In the first example, Ingold never even provides a description of the city square itself, or the scaffold, but the PC's predicament is explanation enough for the omission. It is hardly the time and place to be taking in the sights. In the second, the moment is effectively caught of walking against your will through unfamiliar backstreets. In both cases, the player will be engaged in trying to escape from the situations at hand, and typing "Look" repeatedly is probably not going to be his first priority.

Mutable Scene Descriptions

Sometimes in IF, room descriptions provide game scenery but do not remain static; when we return to a location later on, there will be differences. In *A Change in the Weather* (Plotkin, 1995), the mutability of the scene descriptions makes them memorable. This is one of the key experiments of the game, and its theme: change. The sky grows dark, the sun sets, and the rains come; the map is technically the same, but the descriptions morph dynamically.

Rocky Outlook

A wide angular tongue juts out from the hillside. The park stretches off to the north and west, a vast expanse of bright meadowland, patched with dark woods and stitched with streams that glitter in the sunlight. In the distance, a lake reflects white fire from the setting sun.

The sun is lazily approaching the horizon.

Rocky Outlook

A wide angular tongue juts out from the hillside. The park stretches off to the north and west, a vast expanse of bright meadowland, patched with dark woods and stitched with streams that glitter ruby in the sunlight. In the distance, a lake reflects red fire from the setting sun.

The western horizon has become a surging sea of gold and scarlet waves. The light is magical -- a cool bronze radiance that somehow makes the grass and foliage more intensely green than ever.

Rocky Outlook

A wide angular tongue juts out from the hillside. The park stretches off to the north and west, a vast expanse of dim meadowland, patched with dark woods and stitched with dark streams. A layer of mist is rolling across the landscape.

The mist is turning to a drizzle of rain.

Rocky Outlook

A wide angular tongue juts out from the hillside. A dark expanse stretches to the north and west, impenetrable with rain.

A stream of runoff water is flowing down from the southeast, and pouring down the trench to the unseen stream below.

It's dark and it's raining. Hard.

The question, "How would my PC view this location?" is a useful one for authors to keep in mind when writing scene descriptions. In *Being Andrew Plotkin*, which featured multiple player characters, a particular room description repeatedly changed to reflect the point of view of each PC the player inhabited during the course of the game.

File Room (as Peter)

This bleak room with its short, slumping ceiling does nothing to brighten your morale. Short file cabinets, marked in reverse alphabetical order, crawl in a line along the walls like an army of stupid robots. One measly window lets in a tiny square of sunlight.

You see a copier machine here.

File Room (as Valerie)

The file room is one of your favorite rooms in the company building. One, it's always orderly and clean; and two, it doesn't sport brightly colored IKEA

furniture. It's businesslike and efficient. There's even a window to give the area a sense of openness. Early moonlight peeks in, drawing a long amber trapezoid on the carpet.

Peter is here.

File Room (*as Zarf*)

The file room is an unimpressive rectangular room full of squat cabinets. The file cabinets are a pale yellow, like raw milk, and each stands about 38 inches high. There is a maladjusted ceiling tile, and scruffy stains on the padded carpeting. In the north wall is a window, about two and a half feet square, with a crank latch. A copier machine sits near the wall, bearing no make or marking that you recognize, even though no company produces generic photocopiers that you know of. Even more curiously, the wall socket behind it is empty, meaning that the copier is not plugged in, and yet it definitely seems to be turned on.

File Room (*as Peter*)

The file room looks considerably cleaner, perhaps better than before. The reassuring familiarity of its short, slumping ceiling seems now to brighten your morale. Short file cabinets, marked in reverse alphabetical order, stand in a clean line along the walls like a perfect set of teeth. A window affords a beautiful view and a kind square of sunlight.

A different kind of mutable description is used in *Varicella* (Adam Cadre, 1999). Here, in a spin on the effects of the meta-command "Brief," scene descriptions are elaborate on the first visit to the room and succinct thereafter (though the game is still considered to be in "Verbose" mode by default). Cadre stocks these initial scene descriptions carefully, choosing details that allow him to fill out the game's backstory and to flesh out his large cast of characters. Although the narration is in the traditional second-person, Cadre filters everything through the PC's particular view of the world, always exploring his main character's attitudes and memories.

Dining Hall

Though this dining hall was able to hold the entire Venetian delegation with ease when they were here for

the (failed) peace negotiations, you've grown far more accustomed to seeing it at one-thirty in the morning as King Charles and Miss Sierra indulge in a postcoital late-night snack at a table built for sixty. It's been ages since this chamber even remotely resembled the raucous banquet depicted on the tapestry decorating the southern wall. An old suit of armor stands guard over the entrance to the kitchen, off to the north; other exits lead east and west.

Note how much information about the game's world is contained in the description of just one room, and also that the size of the room is implied by these side details rather than spelled out. This is a game world with a full history, and the sense of the kingdom's grander past as opposed to its seedy present is also brought out.

The shorter description of this same location is succinct and functional but still carries a memory of the original:

Dining Hall

The tapestry decorating the southern wall depicts a raucous banquet, but the dining hall is quiet at the moment. An old suit of armor stands guard over the entrance to the kitchen, off to the north; other exits lead east and west.

Conclusion: Getting the Job Done

It is not often said, in praising works of IF, "Great scene descriptions!" Even at their best, scene descriptions provide scenery and atmosphere and are not interactive in and of themselves. They are there to be read but are usually not the parts of the game that players find most remarkable or memorable.

In practice, authors mainly have to concentrate on getting the job done. A scene description, once written, tends to remain static, because an IF author's to-do list is always full, from the beginning of a project to its release and even after, and never leaves room for rewriting a scene description that gets the job done. Keeping the game from exploding with error messages is much more important. So, to close, here are some examples of authors just getting the job done, with the reminder that it did not stop any of them from winning acclaim for their work.

*Christminster:***Library**

A vast room, filled with bookshelves from floor to ceiling: rows and rows of narrow dark stacks stretching away into the distance. There is a card index next to the door, which leads out to the north.

*So Far:***Chill Tunnel**

The tunnel is very straight. You can tell that it's not quite east-west, though; the distant bright spot of outside snow is more east-northeast, and a strange watery glow is visible west-southwest.

*Spider and Web:***Angle Branch**

The corridor runs north and south, like every other hallway in this place. A short hallway branches off to the northeast. To the west is a blank metal door, with attendant black plate beside it.

*Varicella:***Northeast Tower Antechamber**

The northeast tower lies, stunningly, to the northeast, while other exits lead west to the dining hall and south to the eastern ballroom.

Pytho's Mask (Short, 2001):**Archway**

Inside a sort of archway through the body of the palace itself. To the east lies the square garden; west, more gardens but more wild.

The small door to the south stands open.

*All Roads:***Office of Guiseppe Florantine**

The aide's office is simple enough - cabinets line one wall, and a large desk fills the centre of the room, its piles of paper extremely organised. The aide himself is seated in a deep chair behind, looking up at you expectantly.

Everyone writes their fair share of workmanlike scene descriptions, and the above examples should remind authors to not be deterred from releasing work because their scene descriptions are functional but not spectacular. As I said at the beginning: Let the player know where he is, and where the exits are, and your job is done.

References

- *Advent*, Will Crowther, 1972.
- *Hunt the Wumpus*, Gregory Yob, 1972.
- *Zork*, Marc Blank, David Lebling, and Thomas Anderson, 1979.
- *Christminster*, Gareth Rees, 1995.
- *A Change in the Weather*, Andrew Plotkin, 1995.
- *So Far*, Andrew Plotkin, 1996.
- *Spider and Web*, Andrew Plotkin, 1998.
- *The Mulldoon Legacy*, Jon Ingold, 1999.
- *Varicella*, Adam Cadre, 1999.
- *Hunter, in Darkness*, Andrew Plotkin, 1999.
- *Shrapnel*, Adam Cadre, 2000.
- *Metamorphoses*, Emily Short, 2000.
- *Being Andrew Plotkin*, J. Robinson Wheeler, 2000.
- *The Tale of the Kissing Bandit*, J. Robinson Wheeler, 2001.
- *Pytho's Mask*, Emily Short, 2001.
- *First Things First*, J. Robinson Wheeler, 2001.
- *All Roads*, Jon Ingold, 2001.

Repetition of Text in Interactive Fiction

Jason Dyer

The next day... Poot approached Cat. “hello,” Cat said to Poot. “hello,” Poot said to Cat. “hi,” Cat said to Poot. “howdy,” Poot said to Cat. “hello,” Cat said to Poot. “hi,” Poot said to Cat. “hi,” Cat said to Poot. “hey there,” Poot said to Cat. “hi there,” Cat said to Poot. “howdy,” Poot said to Cat. “hi,” Cat said to Poot. “hello,” Poot said to Cat.

—Laura Mixon, *Shattertown Sky*

Repetition of text is a problem unique to interactive fiction. In a conventional novel, the author chooses the flow of text, but in interactive fiction the same text is often seen many times due to a save/restore cycle, dialogue repetition, default parsing, repeated viewings of room descriptions, or simulationist generation of text. Here I briefly cover all five, but first a common solution must be addressed:

The Addition of Aleatory Elements

WhEEEEEEEE!!!!

Are you enjoying yourself?

Very good. Now you can go to the second grade.

Do you expect me to applaud?

—Marc Blank and David Lebling, *Zork I* (1980)

“Aleatory text” simply means setting some random factor so that a piece of text may be printed differently at different times. The quotes above demonstrate one of the first uses of this: are all responses to the verb JUMP. It shows up more commonly in parts of works that are considered frivolous rather than vital: the JUMP example above, or the seven responses to XYZZY in Jacob Weinstein’s *Toonesia* (1995).

“Aleatory structure” refers to adding randomness to other aspects of the game: for example, the appearance of the dwarf in Crowther and Woods’s *Adventure* (1977), the thief in *Zork I*, or the spinning room in

Zork II (leaving it led to a random direction until the spinning effect was shut down). Since a different situation is presented, quite naturally the text has changed, but also the actions necessary to deal with the work itself.

The Save/Restore Cycle

A CHANGE IN THE WEATHER is a *cruel* game, by the Zarfian Interactive Fiction Rating System. Any choice you make may be a wrong choice, and you may not discover it was wrong until later; and not choosing is always a choice. Timing may be critical. Save often and keep your old saved games.

—Andrew Plotkin, ABOUT text from *A Change in the Weather* (1995)

Timing-critical games such as mysteries and the work quoted above give the reader the same text many times not through the regular story flow but in continuous saving and restoring. When a section has been run through many times, the actual text becomes mentally omitted, and the story becomes reduced to structure: how do I get past this obstacle, or exactly what time do I hide to see this suspect?

This is a case where adding aleatory text can do more harm than good. When a player has entered “structural playing mode,” a change of text with identical actions would be disconcerting—did I do something different this time?—when there was no real change at all.

Aleatory structure here is a much more ambiguous matter, dependent on the situation. If a save/restore cycle is forced because, say, one is having difficulty fighting the thief in *Zork I* and needs to try a couple times, but meets the thief in different locations, the underlying logic behind the player’s actions still remains. However, something that represents a fundamental change in timing—say, moving an important object such that the player has five turns to complete a puzzle rather than the six on last play—would be unfair.

Dialogue Repetition

>ask bill about thom

Bill says, “I, uh, I’m really not sure how much longer I can take his

attitude. He's been really, ah, annoying lately."

>ask bill about thom

Bill says, "I don't, um, really know anything else about it."

>ask bill about thom

Bill says, "I don't, um, really know anything else about it."

—Stephen Granade, *Undertow* (1995)

One of the classic worries of authors of interactive fiction is the exposure of their NPCs as automatons rather than mimetic beings by simply repeating the same question over and over. The above quote demonstrates one of the solutions, adding a flag to account for a repeated question and giving a different response. But a repetition of that response still gives an automated feel—most appropriately the person being questioned would start getting annoyed and stop responding, or slap the PC, or wonder out loud if the PC has been having too much caffeine.

Adam Cadre's *Varicella* (1999) sometimes has a different approach when the flag is set: replying, "You sense little profit in repeating that question." But it also has natural responses like:

"What can you tell me about the circumstances of your husband's demise?" you ask.

"No, no, I don't want to think about it anymore, please," the Queen says.

or attempts to rephrase the question:

"Is there anything else you care to say about the King?" you ask.

How deep a concern should this be in reality? Most players realize the writer has limits and will only commit such actions intentionally when bored or stuck with the regular storyline. Integrating important information into further responses would come across as clever to the reader who notices but also suffers the same problem as puzzles that require examining an object multiple times—the medium is expected to respond with an identical answer so the player doesn't know it is accounted for.

Also, restricting the question to one display can cause problems in a

long work if the player is not taking notes. They may remember their NPC companion had some vital bit of knowledge but become frustrated when they can't bring it up again because of a "little profit" or "I don't know anything more" response. In this case a "decay" flag might solve both problems—that is, you can't ask the same question many times in a short period, but if you ask again later the player gets the information (and is possibly berated by adding a "you forgot already?" comment).

Aleatory additions to dialogue can end up causing more repetition rather than less. Consider the books in Nate Cull's *Nevermore* (not characters, but they demonstrate the same problem). When a book is read a random segment is given, and to get all the pieces of information, repeated readings are required. But due to the nature of random numbers (in the roll of a die a one may not come up in 30 tries by sheer luck), far more readings than necessary are needed just to get all the information, which must then be recompiled together. So in the end the same text will be in practice seen many times through the attempt to add variety. In the same way, multiple responses to an NPC query (especially if important information is conveyed) will force the player to type `ASK CHARACTER ABOUT THING` far more times than necessary.

Default Message Parsing

Violence isn't the answer to this one.

—Graham Nelson, Response to `ATTACK` in the Inform parser library

It is literally impossible for a writer to anticipate every response a player might put in, so the player will invariably receive many default messages in the course of a story, many of them the same. This is clearly not a problem with messages that are clearly non-diegetic like, "That's not a verb I recognize." Problems surface when the messages become far less neutral, like the (in)famous Inform example quoted above. What if the player isn't solving a puzzle? What if the player is solving a puzzle and a violent approach would be a perfectly logical solution but the author never anticipated it?

The most easily apparent workaround would be to make every parser message neutral, such as "That's impossible." However, a great deal of

the character of a work can be defined by the default messages, such as the Zork series response when eating a rock: “I don’t think that the rock would agree with you.”

The problems surface when the messages are situational; that is, the message works well in some situations but not in others. (This is unfortunately defined by both the player mindset and the actions in game, so customization can’t simply take this into account. For example, players may try hitting a door because they are frustrated in general, or they may try hitting the door to genuinely try to solve a puzzle.) So, customizations should take into account the overall tone of the work—or the section of the work currently being read—but not attempt to guess the player’s motivations.

Aleatory additions to default messages are for the most part acceptable since if the player attempts to see them all it is out of a sense of play rather than importance to the story. The only danger here is if neutral messages are mixed with flippant messages. Players who see a “you can’t do that” message in attempting BURN on one object and a long joke about flammability attempting the same on another may assume the long joke indicated they were on the right track and act accordingly.

Room Descriptions

A winding passage disappears to the south.

A gently sloping trail winds south.

A doorway to the south winds into a chamber.

—Spectral Associates, *Madness and the Minotaur* (1981)

The lines above are all responses to successive LOOK commands in the same room. That is, the writers almost completely randomized the display of exits in room descriptions. The result is a confusing, surreal mess where the player ends up ignoring the exit descriptions entirely and losing a sense of environment.

Most authors are not nearly as worried about repetition of room description text. Pieces may change depending on changes to the environment, but an element of a room will not be described in three different ways. This is partly due to the use of the BRIEF command, which allows the player annoyed by such things to revert to room titles

after seeing a full room description, and partly due to the same structural reduction that happens in a SAVE/RESTORE cycle—the player will only reread text when necessary as a reminder.

Repetition becomes a problem in the same manner as parser messages: when given inappropriate to a situation. In *The Inform Designer's Manual*, Graham Nelson quotes Johnathan Mestel's *Xenophobia* (1989):

On the wall by the bed is a slightly curved, full-length mirror. You reflect upon this for a while.

and mentions the text works on repeated viewings, while “Astonished to see a mirror, you leap back” would not. Both are cases of describing action in the description: reacting for the player, rather than letting the player react. The revised version is obviously incorrect, since one would not be surprised the fifth time, but why is the first correct? Nelson writes that the subtle humor works here, but to this author's taste even a good joke repeated five times works too much—best would be the technique of using an extended description for the first (and possibly second) viewing and then cut back to a simpler form.

An example of this technique is used in room descriptions that give background detail, as in Emily Short's *Savoir-Faire* (2002):

The top of a broad curving stair: east is the long salon that goes the length of the house, downstairs are the foyer and the grand receiving rooms. (You and Marie used to sit up here when the Count had guests, watching them arrive downstairs in their magnificent clothing, until you got old enough to be introduced yourselves.)

Quite clearly the parenthetical statement need not be repeated, so in later iterations of the same text it becomes merely:

The top of a broad curving stair: east is the long salon that goes the length of the house, downstairs are the foyer and the grand receiving rooms.

The danger in cutting text from description is that important clues that were missed the first time through may be removed. Consider this from Stephen Granade's *Losing Your Grip* (1998):

The building looms to the west, shadowing the path which leads north and south.

Something on the building catches your eye.

The last line (which mentions something vital to the main story) is removed on successive visits to the room, so if the player missed it the first time they would have a difficult time making progress. While background material and jokes can be forgotten, it should always be possible for the player to see important objects.

Simulationist Text

The new element, called a “bitsy,” promises to revolutionize the software industry by making it possible to put vast amounts of information onto floppy disks. In the present case, Cognetics has put 3,545 intersections, 94 subway stations, and 200 landmarks onto two disks.

—Mock-up newspaper inside box cover for Thomas Disch’s *Amnesia* (1986)

Simulationist repetition often occurs in works where the author wants to simulate a realistic number of locations without writing the text for each. As the blurb above demonstrates, *Amnesia* attempts to simulate the entire city of Manhattan. As one might expect, there isn’t a great deal of variety from intersection to intersection.

This is handled—at the uninteresting intersections, at least—by dispensing with a room description altogether. All that changes is the intersection name on the status line. This doesn’t dispense with repetition entirely, considering for example the 94 subway stations. Also, absence of text is a repetition of sorts, with the city’s lack of description making every location feel the same. These considerations can be boiled down to an interface problem; if the city were presented as an above view grid, like in some computer RPGs, the empty intersections would become connective tissue between important locations rather than empty waste.

NPC Dialogue Writing

Robb Sherwin

beaver@zombieworld.com

The very first time I recall being completely smitten by NPC dialogue, I was a kid playing *Spellcasting 101: Sorcerers Get All the Girls* (Steve Meretzky, Legend Entertainment, 1990). The game depicted a group of role-playing college students engaged in a round of *Malls n' Muggers*. I had plenty of things I could do in the game at that point—classes to attend, spells to find, co-eds to maybe seduce once my parents had gone to bed and it wouldn't be quite so weird—but I had my player character stay put in the dorm and just listen to this group of NPCs play a game with each other.

It was enthralling. I had no tools to make text games of my own at the time, but I was intoxicated by this world that was simulated so well that there were a group of people doing their own thing in it, talking to each other, having fun together, oblivious to anything else in the game world. There isn't anything that excites me more in interactive fiction than engaging non-player characters in conversation, and I hope that I can communicate what I've learned over the last decade in making these games, so that you might learn from my mistakes and bring forth your own world, one filled with text-based chatterboxes of your own.

Modern-day text game development systems offer many tools for authors to create rich and memorable non-player characters. The aspects of the genre that are considered “weaknesses” when compared to commercial games—the lack of graphics and sound—are really strengths when it comes to developing a fully realized, talkative NPC. As an author, you will never find yourself forced to code a scene where your player is trying to get information out of a creepy-eyed, plot-sensitive fellow, graphically depicted within the lowest dip of the uncanny valley, speaking with unconvincing voice acting. The limitations of graphical games that made text adventures so appealing in the 1980s exist again, in a way, now that so many commercial games record hours of sometimes-unimpressive audio for their possible conversations and desperately attempt to show players realistic faces and expressions.

As a player, when I find myself playing a game with characters that have charmed me, I can't be given enough content. There is no insight or take that they may have that I don't want to read. It is a delight, in a well-

written game, to be around NPCs that talk to each other. There are many conversation systems available, with their own pros and cons.

The three systems that are most readily seen are “ASK/TELL,” “TALK TO,” and menu-based conversations. While an author will be best served by using the system he or she is most interested in, the types of NPCs that inhabit the game world can make a strong argument for a particular system.

“ASK/TELL” works really well when the player is in an open-ended world and is expected to find clues and solve puzzles that reveal and further the plot. Perhaps the player isn’t really friends with any of the NPCs, so the non-player characters aren’t going to particularly offer up mountains of exposition, due to the fact that the player is an outsider, or investigator. A modern example of this is Kent Tessman’s *Guilty Bastards* (1998). The player is stonewalled by many characters until investigation reveals the truth, which can later be asked about, evidence in hand. “ASK/TELL” is perfect for mysteries and detective-themed stories. One pitfall in this system is that of player versus player-character knowledge. It can be frustrating for a player restoring from an earlier save to have to repeat steps just to get to the point where we can re-ask an NPC something that will advance the plot—and it just comes off as weird if the author allows his player to ask NPCs about things whether or not they have been introduced. In Level 9’s *Knight Orc* (Pete Austin, 1987), one can ask the Rainbird about any object or character in the game, regardless of whether it’s actually been encountered in that particular session. The delight of gaining omnipotence quickly gives way to a sneaking suspicion that a bug has been found, and mimesis is completely broken.

“TALK TO” is a fine option when the crux of the game is not necessarily about conversations. Oftentimes, “TALK TO” is a shorthand way for the author to handle players gaining knowledge, without requiring the player to phrase questions and statements involving a particular noun or happenstance. I used “TALK TO” with Mike Sousa in our 2001 release for the Interactive Fiction Competition, *No Time To Squeal*, not only because we had a two-hour maximum for our players to experience our game, but because we were pressed for time in the development cycle as well. “TALK TO” was the quickest method we could pull off as authors.

Menu-based conversations shine when one of the chief draws of the work is the interplay between the player and NPCs. A menu allows you

to select the exact phrasing the player character will use, giving the player maximum power over tone, word choice, and level of sarcasm. More, the NPC is able to jibjab right back in the same amount of detail. Menus take away the feeling of immersion from some players, as a desire to “lawnmow” through the choices may arise, but it is also an excellent way to construct quick back-and-forth dialogues.

(It is worth noting that the 2009 collaboration *Alabaster* (Emily Short et al.) features a system all its own—you simply type what you might want to say to an NPC in plain English. A triumph in conversation development, this system has been shown to be wonderful when the bulk of a game is speaking, at length, to a single NPC. Plans to release the code for the system used in *Alabaster* are in the works.)

Talking to NPCs can also reveal more about the player character than it reveals about the NPCs. A game written in the first person, or one in the third person with an extremely sympathetic player character, can easily make the PC look like the coolest kid on the block, but if everyone he talks to has a fair amount of contempt for him, it can provide a contrast for the player to chew over. I tried to do that in my 2006 release *Pantomime*—many NPCs barely have any time for the player character Raif and are rather short and disrespectful when they are talking to him. The narrator otherwise tries to paint Raif as the most beloved person on the colony.

NPC dialogue can also more effectively reveal nuances about the characters than straight text. Following are two scenes relating to the characters of George Alec Effinger, whose traditional novel *When Gravity Fails* was the basis for the 1990 Infocom release *Circuit's Edge*. First, in the book, we can see the protagonist, Marid Audran, interact with seedy characters that he owes a sum of money to:

“I have the whole sum, Abdoulaye,” I said, “but you’re going to write me out a receipt. I don’t want you claiming that I never paid you.”

He looked angry. “You dare imagine I’d do such a thing?”

I glared back at him. “The receipt. Then you get your money.”

He called me a couple of foul names, then ducked back into the room. He scrawled out the receipt and showed it to me. “Give me

the fifteen hundred kiam,” he said, growling.

“Give me the receipt first.”

“Give me the accursed money, you pimp!”

For a second I thought about hitting him hard with the edge of my hand across the flat of his nose, breaking his face for him. It was a delicious image. “Christ, Abdoulaye! Get Karim back here. Karim!” I called. When the gray-bearded old man returned, I said to him, “I’m going to give you some money, Karim, and Abdoulaye is going to give you that piece of paper in his hand. You give him the money, and give me the paper.”

Effinger’s depiction of Marid’s terse speech patterns when talking to those he is disgusted with allows us to dislike them, too. Compare this to a different scene in the text adventure adaptation, where this sense of intimacy is lost:

Saied snatches it from your hands, muttering his thanks and asking you what took so long. Hastily he unwraps it. What was so important that he sent you chasing around half the Budayeen for it? “Orgasmic!” he exclaims, holding the small glass bottle aloft like a trophy. “The most expensive cologne this side of the Sahara. You know you have to get this stuff specially made? Here, take a whiff.” He squirts some Orgasmic your way. Instantly you are overwhelmed by a cloying flowery scent. “Incredible, isn’t it?”

Granted, *Circuit’s Edge* is an old game that has severe screen real estate issues—the game can’t display much more than six lines at 80 columns each before requesting that the player deal with a “MORE” pause. But not setting up the gameplay to allow us to choose how Marid verbally interacts with his friend makes everyone involved weaker and less-developed than the ones we see in the rapid-fire exchange from the novel proper. Interactive fiction developed today has no screen size issues or disk space limitations at all. Thus the “feature” that comes for “free” in the medium of the book is ours to leverage in our games as well.

Although we only really have words at our command to present our characters and make them seem whole, we do have advantages over commercial games that make ours extremely relevant—our players have

already agreed to read. Nobody is buying into, and getting halfway through, *Varicella* (Adam Cadre, 1999) when they suddenly flip a keyboard because of all the text. But many players will spam their gamepad's buttons to race through dialogue choices in something like *Mass Effect* (BioWare, 2007) or *Baldur's Gate* (BioWare, 1998) so they can get back to the core gameplay elements. Many role-playing games from Bethesda contain volumes of text to read about the game world's characters, but they bring the experience of pilfering, looting, and casting magic missile at the darkness to a jarring standstill. There is no genre of computer game that better allows characters to speak for themselves, at length, than the text adventure.

I think that I ultimately find that NPC dialogue is the one aspect of interactive fiction that has to so clearly come from the heart. Dialogue that sounds natural is challenging. It's putting yourself on display as an artist. It's the most fragile thing to share between yourself and the world. I mean, we come together with our players in a shared illusion. We build a virtual reality for them to enter, and we will hold their hands tight as we direct them toward the things we constructed for them that we want them to see, and maybe misdirect them away from things like fire, rope, and water. But a world without people to talk to is a lonely, empty one, so we include people who can talk, and it's up to us to breathe into them that tiny bit of soul. Through decades of advances, a text-game author can download and install an extension that provides a pre-built bathroom, kitchen, or variable-length fire hose. But nobody but you can flesh out the characters in your game and make them compelling. Nobody but you can turn them into distinct entities that we will love and loathe and journey with and remember. What will they say?

NPC Conversation Systems

Emily Short

Conversation is one of the most challenging things to code in interactive fiction and also one of the most widely discussed. There are a number of issues: how will the player communicate with the game what he wants to say to the NPC? How will information be represented internally? How will mood and context be represented within the work? To what extent will the NPC control the flow of discussion?

There is no single right answer to these questions. Conversation design depends very much on the author's intentions for a work.

Questions to Start With

The Purpose of the Non-Player Character

Before you code anything, you should consider what kind of game you are writing and what purpose you have for the non-player characters (hereafter NPC) who will appear in it. A game with a strong emphasis on puzzles, where NPCs are present only to provide another kind of challenge, will have a very different treatment from a linear, story-oriented game where NPC interaction is the chief purpose of the game. A mystery with a lot of knowledge puzzles will again have a different set of requirements from a romance, where emotional interaction is emphasized.

Even within a game, different NPCs can have different roles from a gameplay perspective. Do you want your player to talk extensively with NPCs and have a great deal of flexibility in the outcome? Or would you prefer to have control over how each conversation goes? Are the NPCs there mainly for local color, or do they provide vital exposition? Do they have to accomplish anything in the story?

The answers to these questions will affect the rest of your NPC design. Here are a few ways conversation tends to be used in interactive fiction already. It's hardly an exclusive list:

Conversation as a Framing Device

In some games, conversations are there mostly to frame the action of

the game, rather than to *be* the action of the game. The king takes the player aside, describes the important quest that lies ahead, and sends him on his way. The dialogue may be brief and minimally interactive, because it is intended to send the player on a quest.

Or, again, the player comes into a new location and needs to get a sense of the place. Other characters greet him with conversation that establishes their purpose and the sorts of activities that are possible there.

Conversation as Tutorial and Hint System

The player has a sidekick, a teacher, or an aide who tells him what to do. Conversation with this person is rarely actively required for the story to progress; it's just there to guide the player. Matt Wigdahl's *Aotearoa* (2010) makes strong use of that idea, with a mentor who can be reached via radio to answer questions. Interaction is highly focused on what the player is doing at the moment, not on the interlocutor's background, moods, etc.

Conversation as an Exploration System or Mystery

Many conversations in IF are designed as another kind of exploration. Just as the player explores the map and discovers objects, he also explores conversation with NPCs and recognizes the topics that are most worth following up on. What have we discovered? What do we still need to discover?

At its best, this kind of system can play a compelling role, especially in mystery games, because it requires the player to develop a mental model of what information is important and pursue new elements of that information.

At its worst, however, exploratory conversation can become a morass. If the player doesn't have clear leads about what topics to follow up, or if the majority of conversation attempts get a default response from the NPC, the player may start to feel that he is engaged in a kind of guessing game. This is not entertaining.

Conversation as a Tactical Challenge System

Victor Gijbers's *Mid the Sagebrush and the Cactus* (2010) treats conversation as an extension of combat code. The game includes a sophisticated combat system, and the player's choice to EXPLAIN rather than attacking is another way to pacify the opponent or gain time.

This takes attention away from choosing *what* to say and focuses it instead on *when* to speak (in contrast with other choices). Conversation becomes part of a challenge of pace and timing.

A few other games have developed conversation into a kind of tactical game rather than an exploration task. The iPhone game *Think Like A Shrink* (Mind Gamz, 2009) teaches therapeutic methods—challenging one’s patient on distancing remarks, for instance.

Conversation as a Simulation

Blue Lacuna (Aaron Reed, 2009) presents Progue, a deeply complicated character whom the player encounters many times over the course of the game. Depending on how those interactions go, Progue may end the game as a friend, a father-figure, a lover, or an enemy. Though Progue has a few story-critical scenes that happen no matter what, to a large extent his presence and his implementation is intended to prove something else: that the player’s actions matter, and specifically that the player’s interactions with other characters are meaningful.

The specific interaction that a given player has with Progue is thus less important than the realization that that interaction *has* shaped their relationship.

Conversation as Constraint

At the opposite end of the scale, *Rameses* (Stephen Bond, 2000) and *East Grove Hills* (Anonymous, 2010) are games about what the protagonist cannot or will not say. They give the player conversation choices just to prove how powerless he is. Choices that seem like they might lead to a happy resolution prove not to, or the protagonist just can’t bring himself even to say them.

This kind of game presents the unhelpful dialogue possibilities and acknowledges them as part of the protagonist’s mental universe but makes the player experience the same limitations that the protagonist does. The things the protagonist thinks about saying are as important to the story as the things he actually does say.

Conversation as Juice

Lost Pig (Admiral Jota, 2007) includes a gnome character who has an astonishing amount of dialogue. A few of his lines assign the player quests (framing tasks to do) or suggest solutions (working as a hint system). But a lot of them are essentially jokes or bits of local color,

things that the player can interact with to make the game feel more juicy and responsive.

>

Each of these types of intended interaction suggests its own approach to dialogue design. Plot structure is also an important consideration, because some of these interaction styles work poorly with very linear plots and some poorly with very unstructured ones. In a game with a highly predetermined plot—that is, one in which you could sit down and write a list of all the scenes that must occur—a simulation approach to NPC design would probably be the wrong approach. You will probably have less use for variables to track emotion and behavior in general, and you will accomplish more of your effects by scripting them specifically.

By contrast, in a game where the play is very broad, and the player can spend as much or little time with the NPCs as he would like, you may find yourself writing a fair amount of generic code to cover issues like NPC attitude, behavior, and goal-seeking.

Input and the Parser

Conversation is probably the most difficult thing to code for an NPC. There are several problems, but one of the most pressing is that we can't yet express conversation in interactive fiction the way we express it in real life. The most natural thing to do, from the player's point of view, would be to type exactly what he would like to say, using plain English (or plain Italian, or whatever) to convey both content and tone. The Ideal NPC would understand perfectly and would react to the player's attitude as well as the factual content of what was being said.

At the moment, however, this kind of parsing is beyond us. Moreover, while it might be an ideal solution from the player's perspective, it could only be a nightmare for the author. In other aspects of IF design, it's possible to limit the number of things a player could reasonably do: there are only so many verbs, and only a specific set of objects available, and the scope of action is fairly well understood. An NPC who understood all topics of conversation, and all kinds of tonality and mood, could never be exhaustively programmed; there would always be another quirk unaccounted for. And such a character would almost certainly exceed the boundaries of the intended plot, as

well. It would be quite hard to write a story that had any sense of structure or continuity if the intended plot could be set aside while the player taught the NPC the basic rules of cricket.

Here are some of the forms of interaction that authors have explored so far:

Wordless Communication

A reactive NPC is one who primarily responds to what the player does or to stimulus provided by the player. One way to build a character who feels solid but doesn't require as much special coding is to keep the player from speaking to the character directly but have that character respond to things the player does. Burning a valuable object in front of the PC, or discovering a treasure, or vanquishing an enemy, all will earn some kind of response. The more emotionally fraught the environment and situation of the game, the more powerful the NPC's reactions are likely to be.

Sidekick NPCs provide running commentary in *Adventurer's Consumer Guide* (Øyvind Thorsby, 2007), *Treasures of a Slaver's Kingdom* (S. John Ross, 2007), and Neil deMause's Frenetic Five series, among others.

Many commercial graphical games use this approach: *Halo 3: ODST* (Bungie, 2009) does not allow the player to select dialogue, for instance, but has an extensive range of ways the other characters can react to the things the protagonist does in battle. Wordless communication has the advantage, both for graphical games and for interactive fiction, that it allows the player to continue thinking in terms of the same actions and the same verbs that he's already using for the rest of the game.

The disadvantage is that it requires everything the player can communicate to another player to be focused somehow through a physical interaction—and there are times when that may be extremely contrived.

Yes/No Conversation

This is a one-trick pony: the NPC asks questions and the player is allowed to answer "yes" or "no." Andrew Plotkin did a brilliant job of handling this in *Spider and Web* (1998), where it works because there are good fictional reasons why the player is only allowed to say one of those two things.

Talk To

When your player wants to communicate, he types >TALK TO JONES and the conversation takes place without further interference from him. The advantages are, first, that you can put realistic, situation-appropriate dialogue in the mouths of both PC and NPC and, second, that you don't have to worry about parsing anything funny at all, just about disabling (if they're implemented in the language of your choice) ASK and TELL. The disadvantage is that it leaves relatively little power in the hands of the player (whose only choice is whether to have the conversation or not to have it). TALK TO also locks the player into whatever characterization you have chosen for the PC. Stephen Granade's *Common Ground* (1999) exemplifies both the advantages and the drawbacks very well. Ian Finley's *Kaged* (2000), Kathleen Fischer's *Masquerade* (2000), and assorted others have also made use of TALK TO.

This is an arena in which the writing skill of the author will make or break the game. If you are a skilled author capable of conveying a fair amount of character and emotion even in these set pieces (as Ian Finley does, in my opinion), then you may be able to maintain a sense of immersion.

Menu Conversations

When your player wants to communicate, he types >TALK TO JONES and there appears on his screen a menu of three to six sentences he can say at the moment. Perhaps there is also an option to say nothing. Jones then replies, and the player may be given another menu, and so on, until the conversation ends. *Photopia* (Adam Cadre, 1998) worked on a system like this, and some of the existing libraries for handling menus in Inform are at least partly derivative of the *Photopia* ones.

The advantage here is that again the author has control over the form of communication. You can hand the player a bunch of clever quips to say, which characterizes the player character as well as the NPC. (For player characterization that takes place to a large extent in the library menus and the PC's deployment of them, see *Rameses*.)

The problem is that menu systems are fairly restrictive; sometimes the menu doesn't contain anything that the player wants to say, and there's no way to change what's on the menu, or even the illusory feeling of freedom that comes from typing >ASK JONES ABOUT THEOLOGY, even if no response has been implemented.

Another problem is what Duncan Stevens has referred to as the lawnmower effect. If you give me a series of menus, I don't have to do any work to get through the conversation, and I can methodically (using undo, for instance) go back and replay different variations, taking now the first and now the second path, until I am sure that I've seen the whole thing. The NPC is then finished, with no more thought on my part than I give to methodically mowing a lawn.

From my point of view, this lessens involvement. If you are writing a highly directed game like *Photopia* or *Rameses* or *Being Andrew Plotkin*—preferably something so vividly written that the story or the humor of the narrative will make me want to move forward as rapidly as possible—then this may be right for you. If you're writing a game based on investigation, allowing the player to shape his own character, or leaving large stretches of the plot in the player's hands, then you may be better off with something more open-ended.

ASK/TELL

ASK/TELL is the most common form of NPC interaction in the Infocom games and some other old-school works. It allows the player to ASK or TELL the NPC about any keyword he chooses and get a response. The approach is more flexible for the player than a menu conversation and works better with knowledge-based puzzles where the player may be discovering and ASKing about new information as his understanding improves. On the other hand, it is typically more difficult to code a conversation that appears to have a natural flow with ASK/TELL than with a branching menu, where you can ensure that each remark rationally follows from the conversation path that has led up to it. It can also be a challenge to avoid guess-the-noun problems, where the player is required to think of the specific keywords the author had in mind in order to advance the game.

ASK/TELL conversation can also mean minimal characterization of the PC. With a menu system, the player sees the PC's dialogue; with ASK/TELL, he may get only responses from the other character, like so:

>ASK FRED ABOUT THE PORSCHE

"It belonged to my uncle," Fred replies. "Don't tell me you want to borrow it too."

There's no indication here of what the player's character might have said.

We can code around this and fill in what the player says too, if we like:

>ASK FRED ABOUT THE PORSCHE

“Say, that’s a beautiful machine,” you say. “Where did you get it?”

“It belonged to my uncle,” Fred replies. “Don’t tell me you want to borrow it too.”

Nonetheless, the risky part of this is that the player may not have intended ASK ABOUT THE PORSCHE to mean quite what we’ve decided it means. Maybe he wanted to ask about something else—the mileage, how well it drives, whether it’s for sale, why there are blood stains in the back seat. Providing dialogue for the player in this context adds player-character attitude and characterization but at the expense of some of the player’s sense of control.

Conversely, with ASK/TELL it is hard to allow the player to express those more complex ideas if he wants to. Usually the game won’t accept more than one or two words there, so >ASK JONES ABOUT THE TIME OF THE MURDER is likely to fail flamingly. The 2000 Comp game *1-2-3...* (Chris Mudd) tries to get around this, but it does so by prompting the player, and it is remarkably inflexible about *which* long string of words it will accept.

Topic Words

Used in games such as J. D. Berry’s SmoochieComp game *Sparrow’s Song* (2001) and the older comp game *She’s Got a Thing for a Spring* (Brent VanFossen, 1997), this functions very much like ASK/TELL except that the verbs themselves are omitted. The player simply types a word he wants to bring up, and the conversation proceeds accordingly. This isn’t so much a new interface as it is a slight streamlining of an existing one: the author does not need to code separate answers for ASK and TELL, and the player does not need to try both verbs.

ASK/TELL with Context and Special Topics

TADS 3 comes with a conversation implementation that adds to the standard ASK/TELL system in some important ways. First of all, it adds an idea of conversational context, so that the game keeps track of whether the player is currently conversing with anyone. If the player tries to speak with someone without saying hi first, the game may generate a

greeting; similarly, the NPC may say goodbye when the player walks out on a conversation in progress.

Secondly, TADS 3 introduces the idea of special topics. A special topic usually involves an entire phrase (like “ASK JONES ABOUT THE TIME OF THE MURDER:”), and the player is given a hint when such a special topic becomes relevant in the conversation, with a message like “You could ask Jones about the time of the murder or tell him about the smoking gun.” For example, from Eric Eve’s *All Hope Abandon* (2005):

The blonde woman turned round just as you joined the queue and asked, “So, how are you enjoying the conference?”

(You could say it’s great, or be unenthusiastic.)

> be unenthusiastic

“Well, to be honest, I’m not a great enthusiast for conferences,” you confessed, “and I’m not sure this one has changed my mind so far.”

“That’s a shame!” she laughed, “But maybe our star speaker this afternoon will enthuse you more - are you looking forward to him?”

(You could say yes or no.)

> ask woman about conference

Answering the question she asked struck you as being an elementary courtesy, especially since you wanted to create a good impression.

> yes

“Professor Wortschlahter, you mean?” you replied, “Yes, his topic looks very interesting.” That was a lie, of course, but you didn’t want to appear too negative.

“You think so?” she replied, “I’m not so sure - to be honest, I haven’t been all that impressed by his books.”

Special topics do not preclude the use of single-keyword ASK/TELL, but they introduce some of the specificity of menus to those parts of a conversation that most need them. At the same time, they avoid dropping into an alternative mode of user interaction. Some players find it jarring, when most of their input is in the form of textual commands,

to be asked to click on a menu or select a number from a list: it can be an unwelcome reminder that the game is just a game. The special topics system avoids this kind of uncomfortable transition.

Modified Menu/Topic Hybrid

This system combines the freedom of ASK/TELL with a menu system. When you begin a conversation with someone, you see a menu of the possible things to say listed in the status line, and you may say one of them simply by typing the corresponding letter. If, however, you would like to change the subject, you may also type >TOPIC DIAMOND NECKLACE, and a new menu appears. For instance, >TOPIC JONES might bring up a menu

1. Have you seen Jones anywhere?
2. What does Jones do here?
3. How long has Jones been working for the company?
4. What is your opinion of Jones?

This gets rid of the lawnmower problem and forces the player to take some initiative in choosing how the conversation will go. It also means that you can allow the player to ask questions much more complex than are available in an ASK/TELL system but without completely giving the game away by including questions like >ASK THE QUEEN WHETHER IT IS TRUE THAT SHE STOLE THE PRINCESS'S DIAMONDS into a single main menu with >QUEEN, HELLO and >DO YOU KNOW WHERE I COULD GET MORE OF THIS SCRUMPTIOUS CAVIAR?

One major drawback of this system is that it requires more writing to implement usefully than any other. Simple ASK/TELL usually means that the author has to write two responses per character for each major topic of conversation; if the plot is very complicated, or if it's possible to get the NPC into more than one state of mind, then the author might have to write some variations on these responses, as well. With the menu system, in order to give the impression of a full implementation, the author winds up writing several questions and answers per topic per character. This can rapidly slide into the realm of the ridiculous.

Chatbot-like “Natural Language” Input

Chatbot programming often involves the recognition of phrase

structures, like “Who is *?” or “What is *?”

Jon Ingold’s game *Insight* (2003) allows the player to type complex natural questions such as BOB, WHO IS YOUR WIFE? or FRED, WHY ARE YOU ANGRY? Without having seen the source code, I would guess that it involves picking out the keywords (WIFE, ANGRY), identifying a type of question (WHO, WHY, etc), and figuring out an appropriate response from there. To take a relatively spoiler-free selection from early in the game:

>man, who are you?

“My name’s Mackenzie. But I, er, guess you already knew that. What do you want to know? You know it all already, right? I’ve been working - living - in Olympia. I’m a genetic designer.”

>mackenzie, what is your name?

“You already know my name, of course you do,” he replies.

>mackenzie, where is olympia?

“Nice enough place, I guess,” he says. “We have a lot of problems with the windstorms because of the nearby mountains. I’ve been working on solutions for that, using plants.”

>mackenzie, do you come from olympia?

-- Please be more specific about what you want to say.

>mackenzie, who else lives in olympia?

“I’m sorry,” Mackenzie replies. “I didn’t quite follow that.”

When the parser gets the exchange right, the effect is magical. Where the parser realizes it can’t interpret, it can give a relatively satisfying excuse. It’s the middle case that’s the most disconcerting, where the game mostly understands but misses some critical nuance. The response for “where is Olympia?” doesn’t sound exactly right; it seems to have caught the keyword “Olympia” but not to have interpreted “where?” properly. And the “Please be more specific” line isn’t helpful at all.

It’s unfair to be too critical, because this game is attempting something extremely difficult. The problem is that faking a natural language understanding always leaves some notable gaps. One might expect that it would be easier to write a chatbot for an IF game (where there is a small modeled world whose state is described within the

program) than one whose domain of conversation is real life. Within the IF world, the chatbot has access to the same facts about the state of the world that the player does. On the other hand, in IF the performance demands are, in a sense, higher: what can be forgiven in a chatbot becomes a bit more serious in an IF game, where the success or failure of an interaction determines whether or not the player will be able to see the rest of the plot and finish the game in a satisfying way.

At the extreme of this kind of design, it's worth looking at the graphical interactive story *Façade* (Andrew Stern and Michael Mateas, 2005). *Façade* uses sophisticated techniques to understand natural language input as conversational actions, such as asking for more information, insulting a character, hitting on a character, and so on. *Façade* has impressive depth but is still imperfectly tuned.

Meta-conversation Verbs.

Adam Cadre's *Varicella* (1999) uses a form of modified ASK/TELL that allows for a little more player control of the PC's behavior. The ASK/TELL system works the same way as ever, but you are allowed to adopt one of three tones of voice: hostile, cordial, and servile. To take an example from the very beginning of the game:

>tone cordial

You adopt a cordial manner.

>ask steward about nails

"How's the manicure proceeding?" you ask.

"Shouldn't be much longer, sir," the steward says.

The steward expertly attends to your fingernails with an emery board.

>tone hostile

You adopt a hostile tone.

>ask steward about nails

"How much longer is this going to take, you mediocre manservant?" you bellow.

“Shouldn’t be much longer, sir,” the steward says.

The steward lightly blows on your fingertips.

>tone servile

You adopt a servile posture.

>ask steward about nails

You’re scarcely about to address a common servant in an obsequious tone. For heaven’s sake, where is your self-respect?

Reactions to this system have been mixed. I found it entertaining to go around seeing what interesting variations on the various statements I could get by changing my tone of voice, but I also frequently forgot to set the tone correctly and found myself acting inappropriately. And the more engrossed I was in the game, the more likely I was to forget about the tone system, which meant that I used it more as a toy than to get at the actually interesting variations that I understand are buried there as a result.

Another experiment along similar lines is *Elizabeth Hawke’s Forever Always* (Iain Merrick, 2002), which permits the player to use adverbs to control the tone of conversation. The player can, for instance, WHISPER HUSKILY, SHOUT ANGRILY, SPEAK POLITELY, etc. A menu of options appears, and its contents depend on what manner of speech you chose. This system, unlike the one in *Varicella*, lets the player see what he is going to say before he says it, so the effect of the different tones is a bit more obvious. The game is not flawless, but the problems in the later scenes seem to stem more from bugginess and lack of testing than from problems with the system as such, which is fairly interesting.

Both *Varicella* and the *Forever Always* have systems designed for a game where emotional states and relationships between characters are of primary interest; one’s a palace intrigue, the other a romance-novel parody. The *Forever Always* system might not be at all successful for a game that centered on information gathering, since the player isn’t allowed to specify keywords, and there’s no potential for following up on, say, the clues of a mystery. On the other hand, I think it actually works better than the *Varicella* approach for the specific and limited purpose of doing character-emotion-based IF. (Since *Varicella* is partly about discovering information, the adverbs-only approach wouldn’t have

worked there.)

A word of caution, however. It's important, when expanding a conversation system to include new verbs, not to leave the player with an unmanageable number of options. In *Varicella* it was possible to keep track of the three tones of voice, but other suggestions I've heard (such as a >BE SYMPATHETIC command) or tried to implement myself (a system including COMFORT, INSULT, APOLOGIZE, FLIRT, SEDUCE, SMILE, LAUGH...) suggest perhaps-unmanageable systems.

The Model: Representing Conversational Information Internally

So far, we have looked primarily at how the player will communicate his instructions to the game and not very much at how the conversation will be modeled internally. To go further, I also want to define a few pieces of jargon for the purposes of clear discussion:

topic: a subject of conversation, such as “the weather,” “religion,” “employment,” “the Red Sox,” etc.

fact: a proposition about a topic, such as that the Red Sox have lost the ball game, rain is expected tomorrow, etc.

quip: the actual verbatim dialogue used, such as “The Red Sox were let down by their bullpen again this afternoon” or “Seattle can look forward to its thirty-ninth day of rain tomorrow.”

effect: the result of saying a given piece of verbatim dialogue beyond merely expressing information, such as causing the non-player character to become sad, committing the player to do something for the non-player character, etc.

conversational goal: something the player or NPC is trying to achieve through the conversation—to find out a specific piece of information, to get one of the characters into a given mood, etc.

scene: a particular section of the plot; the responsibility for deciding

which scene is in progress typically lies with plot-modeling algorithms that consider the whole state of the world and are not part of the conversation system per se.

Even a very simple conversation model usually represents at least one of these elements in code. Complex models may treat several elements at once or may apply more rules to determine what the player is allowed to say when.

Three Traditional Models

Topic Quips

Traditionally underpinning the ASK/TELL system is a model in which any given topic of conversation is tied to a single response. This might be implemented with a table or switch statement that exactly matches content from the player's command, or it might treat the topics internally as modeled objects, but there is usually a one-to-one match of quip to topic for any given NPC. In this sense, the conversation is like looking words up in the dictionary: the replies will always be the same, and there is no sense of continuity, of conversational context, or of a rapport established between the player character and the NPC. A slight variation on this is to have a few keywords that the NPC will not talk about until first adequately bribed: the NPC is still a dictionary, but a few entries are written in invisible ink.

Quip Tree

Traditionally underpinning the menu system is a model in which dialogue is a branching tree. At the first node, you may pick A, B, or C to say; if you pick A, you're then confronted with a choice of D, E, or F. Dialogue flows, since the player is never confronted with the option to say anything in any order other than the one specifically anticipated by the author. On the other hand, the player's freedom is constrained significantly.

Scene Quips

Traditionally underpinning the TALK TO system, this model shifts the burden of conversation context entirely to the plot model: each scene offers the player a single prewritten exchange with a given NPC. This leaves the player with almost no freedom, except inasmuch as he can

affect the rest of the world to bring about new scenes in which new dialogue is appropriate.

What the Traditional Models Miss

These models of conversation are legitimately popular, especially in work where NPC interaction is not the most important aspect of gameplay: they are simple to understand and relatively easy to write and extend. Adding new keywords or dialogue branches does not require much work on any other pieces of the system.

The problem becomes much harder when we want to devise a model that combines player freedom with a sense of developing context. Now we have to be able to keep track of what has already been said in the conversation, model the effects of the exchange on the NPC (and perhaps on the player, for that matter), and determine what can legitimately be said next. We might also want to take into account some external information about the world state: what stage of the plot we are in, what the NPC has seen the player do, and so on. Here are some of the design concerns that arise with one or more of the traditional models:

Avoiding Repetitious Dialogue

One of the least person-like habits of the typical IF NPC is that he always answers the same questions in the exact same words, regardless of how many times the player has asked. This issue need not come up in quite the same way in a menu-based conversation, since you could disable questions that have already been asked, whereas there's no good way to prevent the player from typing `>ASK JONES ABOUT HAT` ten times in a row.

At its most basic level, this is just about preventing the NPC from saying the same thing over and over and over again. Real people don't repeat the same words in the same language a hundred times in a row, and it detracts from the feeling of realism if your NPC does. There are several options for dealing with this: having the parser cut in and say, "You remember that Jones told you . . ."; having Jones tell you again but in a slightly modified form using some kind of randomization of text (so that over time you would get similar text over and over, but it wouldn't be identical each time); or describing the conversation without telling you the exact words ("Jones tells you again that . . ."). Alternatively, if Jones

is a feisty sort of person, he can complain if the player asks him the same question multiple times. This is dangerous, though, since if Jones has important information to impart, the player may find himself stuck because he didn't take notes the first time through the conversation.

Contextually Based Reactions

In real life if you're talking to someone and that person starts to read a book, you may take a message from the fact. Likewise, there are spots in conversations where it may be more or less appropriate to react to the other person with advances (>KISS JONES) or violence (>KILL JONES WITH ROCK). If you have a system of conversation that tracks what the current topic of conversation is, and whether anything is actively going on, you can use it to tailor appropriate reactions for KISS, GIVE, SHOW, HIT, et al.

Somewhat more subtly, context in conversation can also be used to interpret the meaning of the player's keywords.

For instance:

[The PC and Inspector Lynley have been discussing murder victims.]

>ask lynley about veronica

"Do you think it could possibly have been Veronica?" you suggest.

"I overheard her arguing with the victim last night."

As opposed to:

[The PC and Inspector Lynley have been chatting about their love lives.]

>ask lynley about veronica

"How well do you know Veronica?" you ask. "I'd like to ask her out, but I'm not sure whether things are really over between her and Marcus."

This kind of refinement is irrelevant in a menu-based conversation, but for ASK/TELL it can lend a sense of depth. It takes some work, though, to make sure that really important questions never become entirely impossible to ask just because the conversation context is set wrong. If the player desperately wants to accuse Veronica of murder, he'll be frustrated if the game only permits questions about her love life.

Abstract Knowledge

One of the artificial abilities we might like to give our NPCs, aside from the ability to wander around a map intelligently and carry out complex goals, is the ability to understand what they are told: to keep track of what items of knowledge they have so far, use them to change their plans and goals, and even draw logical inferences from what they've learned.

Purpose

NPCs give the impression of being much more active and thoughtful if they show signs of having a private agenda of their own—which may include raising new conversational topics, deciding to cut a conversation short, and so on. There's a trade-off here again: the NPC who takes actions and doesn't wait for the player may seem more dynamic and alive than an NPC who sits around being questioned at the player's whim, interspersed with turns of the PC taking inventory, looking under sofa cushions, and unlocking safes. And if you have a specific set of information you need to convey to the player, sometimes it's useful to have an NPC who will just keep coming back to that topic until it's been adequately covered.

A Few Alternate Models, and Thoughts on Model Design in General

What follows is a discussion of some alternative conversation models I have tried, my reasons for trying them, and how well I thought they worked. I focus on my own work here because discussing these requires some understanding of the code base; it is not always possible to tell with any certainty how someone else's game is modeled internally.

Topic Quips with Mood Tracking and Quip-tagging

Galatea (2000) has a number of topics. Each of these will produce only one quip of response at any given time: the interface is ASK/TELL, so there are no quip options presented to the player. However, quips have a variety of effects, especially on *Galatea*'s mood and position, so that the state of the conversation is in constant flux; and the state of conversation in turn affects which of the available quips is used when topics are mentioned. Quips that have been used are tagged as used so that they will never be repeated; some quips can be used only after other

quips. There is no systematic tracking of facts, even though certain facts do come to light and have a profound effect on the state of the game (but this is programmed by checking whether any of the relevant quips have been used).

This is a somewhat shaggy system and is challenging to extend and maintain, and it does not entirely protect against contextual breaks where the flow of conversation is lost. Moreover, Galatea herself is mostly reactive rather than active. There are a few points where she is specially programmed to make a follow-up comment if the player does not speak about something on the next turn, but for the most part she tends to be silent until spoken to.

Topics with Multiple Quips

In *Pytho's Mask* (2001), an assortment of different quips are associated with different topics. Interaction is handled through a topic-menu system, so when the player asks about a topic he is given a selection of all the currently relevant quips associated with that topic. Quips are marked when used so that they won't be repeated unless it is particularly desirable for the player to be able to re-ask a question; sometimes in that case there is one quip used for the first time the question is asked and an alternate form for subsequent askings. Using a quip can also have the effect of changing the topic, as well as producing emotional responses.

This model arose from experimentation with the topic-menu interface, used here for the first time. However, it is possible to use a different interface with this kind of model, as demonstrated by Kathleen Fischer's *Redemption* (2003) and the enhanced ASK/TELL conversation system built into TADS 3; these input systems are probably best for cases where the number of quips per topic is sparse, even if it is not one-to-one.

Topics with Multiple Quips and Abstract Facts

The model in *Best of Three* (2001) is designed to support an NPC whose conversational goal was to discover information from the player by asking questions and drawing logical inferences. Like *Pytho's Mask*, the model associates a number of quips with each topic and uses a topic-menu system to present this to the player. However, it also implements separately a tree of facts; a quip can indicate one or more of these facts.

The tree structure represents the way in which the NPC will draw inferences. He is curious about certain facts and has the ability to ask

questions about them or direct the conversation, since he is allowed to choose and speak a quip of his own after answering any quip offered by the player. When the NPC has learned all the facts underlying one node, he then infers that that node is correct; he may ask the PC a question to verify his conclusion, but essentially the reasoning process is complete.

In practice, the game turns out to be not very much fun to play. The system of inference is cumbersome, and it is not always obvious to the player that a reasoning process was going on behind the scenes, rather than a prewritten script. Moreover, conversation always tends up the factual tree to arrive at the same goals in the end, so despite the dynamic internals of the game, the difference between play-throughs is usually a matter of reaching the same quips in a different order, rather than entirely different lines of discussion. The NPC's behavior might have been more interesting if he had not driven the conversation so relentlessly (too much NPC autonomy makes the player feel helpless) and if the inference system had caused more complex behavior, making it more obvious how the NPC was responding to changes by the player.

Topics with Cross-indexed Quips

City of Secrets (2003) uses an elaboration of the multiple-quip implementation described above, except that quips can be associated with multiple topics, as well: asking about any of the topics covered by a quip will make that quip available. Moreover, topics are nested, so that topics about specific items (like a particular character or place) are treated as sub-items of general topics (such as an entire group or region). When the player runs out of quips to say about the current topic, the game explores whether any quips are available for the more general subject, and so on. The result is only partially successful at providing a sense of continuity and keeping the player constantly prompted with possible things to say: I wanted to avoid having the topic menu become empty of quips any more often than necessary.

To make matters more complicated, there are a few meta topics representing abstract actions such as >INSULT, >COMPLIMENT, and the like: quips are associated with these topics because of their effects rather than because of their content, but otherwise the command >INSULT functions much like >ASK JONES ABOUT INSULT: every insult-related quip that is currently available becomes accessible for the player's use.

Contextual determinations are messy and not handled very

systematically. There is no representation of facts as such. Quips can be tagged to indicate that they could follow only immediately after other quips (emulating the effect of a dialogue tree in small) or only after other quips had been used (but not necessarily immediately before). Moreover, arbitrary information about the game state is sometimes used to determine whether a quip is available for use. Finally, some quips are associated with specific NPCs, while others can be used with any member of a class of NPCs (e.g., any shopkeeper, or any member of a political faction).

NPC conversation goals were coded in a similarly ad-hoc way: the plot was divided into scenes, and during a scene the NPC might have a script of quips to present to the player. The player had some flexibility in that he could delay the script by asking his own questions or (sometimes) changing the subject, but the NPC would revert to the main script when the player did not take action or ran out of available quips.

This is by far the most complicated system I have ever constructed, and it was, frankly, out of hand: hard to program and even harder to maintain. It did provide a certain richness of interaction, since the player had a lot of freedom to change the subject and to give abstract commands as well as concrete ones; nonetheless I believe that similar effects could be achieved better in other ways. One particular failing not only of this model but of the entire game is a lack of focus: because I was insufficiently clear in my mind about how I wanted the player to be able to interact and affect the plot, I tried to implement “everything reasonable.” As a result, unsurprisingly, play is not always very well directed. Throughout the project I struggled to produce enough material, to handle the ramifications of massive combinatorial effects, and to keep pacing problems at bay.

Database Queries (Multi-fact Topics)

This model works with a chatbot-like interface, and the player’s input is scanned for keywords and standard sentence forms: input such as SAM, WHERE IS MR GREEN will trigger on “where” and “Mr. Green” and dynamically generate a response from Sam on the whereabouts of Green, the topic. Some five or six categories of fact are provided for each topic; because there is so much to ask about, quips are not all prewritten but are made up on the fly.

This system was used for parts of *Mystery House Possessed* (2005) (a game that actually implemented several conversation models for NPCs

of differing intelligence—see below as well). The game presented the player with a dynamically generated mystery in which a randomly selected NPC murdered other NPCs in turn, leaving clues behind; the ability to ask specific questions was intended to assist in investigation. A drawback, of course, was that response quips to the database-style queries tended to be fairly lacking in personality, though I made some attempt to add local color when the speaker might have strong feelings about the topic.

Two-topic Quips

In this model, also used in *Mystery House Possessed*, the player's input is scanned for topic keywords, and the last two distinct topics are used to select a quip from a table. The idea is that the interesting things to say are about relationships between topics (how does Daisy feel about Tom?) rather than in the topics themselves; this seemed appropriate in a context where I wanted to supply considerable amounts of information about how people reacted to one another.

This system provides a very limited kind of context, as well. The player might say >ASK SAM ABOUT WORK in one sentence and in the next >ASK SAM ABOUT JOE, in which case the second answer will trigger on WORK and JOE: Sam will give a reply about Joe's employment, rather than some other aspect such as Joe's love-life, attitude to another character, health, etc. A selection of concrete topics were provided (mostly the characters in the game, plus a few who were missing) as well as a few abstract topics (work, love, etc) that might be thought to have some bearing on people's motivations. It was also possible to say >ASK SAM ABOUT JOE AND TOM explicitly in order to query a relationship in a single move.

The results were interesting, but, as always, it is dangerous to permit the player to use a chatbot-style interface in IF; it's all too easy for him to get completely nonsensical reactions out of the system. Moreover, while the system offered good continuity over a couple of moves, it was not so good at producing the sense of an evolving conversation over a long time. This was less a liability in *Mystery House Possessed* than it might have been in some other games, because *MHP* involved a number of mentally unstable people who might be expected not to converse rationally, and also because the pacing of the game precluded very extended conversations with anyone: other NPCs kept being killed, discoveries made, characters wandering from room to room, and so on, so that the

number of turns that could be spent drawing out a single conversation was limited. The flaws of the system would have shown more severely had that not been the case.

Topics with Interstitial Quips

In this model, used for *Glass* (2006), quips are associated not with specific topics but with the transition from one topic to another. Order of topics is important here, where it was not important in the two-topic quips model above. When the player or an NPC mentions a new topic, the model looks up the current topic and the last topic, finds a quip associated with that specific conversational transition, and prints the result. The quip used is then erased from the table in order to avoid repetition. The conversation is divided into a series of scenes, and in each scene the NPCs pursue a conversational goal, namely, to move the conversation to a given topic. The player can interfere by changing the topic himself. This model allows for a simple AI implementation to handle the other characters' conversational initiative: conversation topics are defined to be related to one another (much as one room on a map leads to another), and the NPCs use a pathfinding algorithm to discover the current best path from the current topic to their goal topic. Facts, on the other hand, are not explicitly modeled. Major state changes in the conversation, where the characters can be assumed to be using a new set of information and moving towards a new endpoint, are modeled as scene breaks; so each scene might be understood as a contextual domain in which certain facts are known and certain attitudes are at work.

The result is a game that moves quickly and fluidly and does a good job of preserving contextual flow: because quips are tied to a before and after, it is hard to reach one that does not follow from what went before. NPC AI is easy to write for this, as well, because the conversational goal of the NPC can be easily expressed in terms of pathfinding through relations inherent in the model.

On the other hand, player freedom is somewhat diminished (as in *Best of Three*) because the NPCs direct so much of the conversation. Play-throughs are markedly different only if the player succeeds in intervening at an important point, rather than being as free-form as in *Galatea*.

Topic Quips with Tight Scene Correlation

Used in *When in Rome 1* (2006), this model uses a traditional ASK/TELL

interface and provides one quip per topic of conversation per scene but changes scenes very frequently. Thus it is difficult for the conversational context to change very much within the space of a scene. This essentially shifts the burden of context-tracking so that it becomes part of the plot model rather than part of the conversation model. This works but is probably put to best effect in games with a focused, fast-moving narrative.

Quip Menus with Missions, Abstract Facts, and Trust Effects

This model, for an unreleased work in progress, is designed to present a dialogue menu interface for a CRPG. Since there is no ASK/TELL input, no topics are modeled as such. Instead, we model missions that the player is currently working on or being requested to undertake, and this largely determines the context in which different quips become available: some things can be said only when the relevant mission is in progress.

NPCs pursue conversational goals by offering missions to the PC, inquiring after missions already in progress, or asking questions about the PC's intentions and loyalties. The player pursues his conversational goals by asking for information that will help him solve missions; his ability to find information is partly determined by how much he is trusted by his interlocutor. Quips that have been used once are occasionally discarded if they can't be repeated, but more often they are instead moved to the base of the dialogue tree, so that the player can ask again any question he has already reached and asked once. This means that after he has gone to the trouble of finding something out through careful investigation, he is not required to repeat the entire process.

Quips may also be associated with facts. Not every quip has a related fact; facts are modeled, sparsely, where they are of particular importance. Some facts are said to exclude other facts—that is, if one is true, the other must be false. When an NPC asks a question of the player, the player may respond with quips indicating facts, or with evasive quips that put the NPC off (“none of your business,” “I don’t remember,” etc.). The NPC next considers what he has heard from the player so far and determines whether it is internally consistent. If he has heard a consistent fact from the player, his trust rises. If he has heard an evasion, his trust remains the same. If he has heard a fact that excludes or contradicts an earlier fact, his trust drops, he indicates that

he has noticed an inconsistency, and he asks the player to clarify his position.

This model (though in an unfinished work) seems to be satisfactory so far in producing dynamic menu dialogue and providing situations in which the player must acquire and share information strategically. Revealing information can help the player in the short term by providing a trust boost, at the expense of diminishing his later options to lie successfully. Dialogue repetition in this genre is considered a less serious flaw than it generally is in textual interactive fiction, and therefore we are not concerned about rephrasing things that have already been said once.

Some Conclusions

The point of this overview has not been to recommend any particular system of conversational modeling but rather to suggest a few ideas.

First, while some models best support some interfaces, the correlation is not simple, and it is worth thinking about the model explicitly rather than throwing it together ad hoc.

Second, a “conversational model” can treat facts, topics, or quips as the basic units of conversation; or it can model combinations of these; or it can work with other base units. I have considered, but never attempted, a conversation model whose primary elements and actions would be emotive rather than verbal, so that, for instance, a quip might represent movement from an angry state to a happy state, rather than from one topic to another topic; NPC goal seeking might also involve seeking a series of quips that would lead to the desired emotional outcome. In short, there are numerous possibilities. Keep an open mind; if you find yourself designing a system that requires a great deal of special-casing (as I did with *City of Secrets*), stop and ask yourself what these special cases are usually accounting for and whether you can model that aspect of the world systematically instead.

Third, it is important to focus on the kind of interaction you expect the player to do and (if you intend some degree of AI for your NPCs) the kinds of conversational goal-seeking you expect of your NPCs, as well. It is easy to be seduced into adding all sorts of things to the conversation model that are not strictly necessary and that produce horrible complexity. For one thing, adding new features to the model usually means writing much more content later on (and possibly spending more time debugging, too).

Output: Sharing Information with the Player

When we build a conversation model, we need to think not just about levels of implementation and our choice of conversational input; we also need to consider how we are going to represent the model's output to the player, to make the most out of our simulated world and give the player enough information to make meaningful choices.

Offering Recaps

If past conversation affects the current state of the game, or if there are substantial amounts of information that can only be gained from conversation, it is often wise to provide the player with some way to review what has already been said without taking extensive notes. To this end, some games offer commands like RECAP, REMEMBER, or THINK ABOUT, allowing the player to recall what was said about specific topics or to review what topics have been discussed in general.

Exposing the Mechanism (Partly)

If we're modeling a character's mood as something separate from the conversation exchanged, we want to let the player know in subtle ways when the mood has changed; we may also want to let the player get a sense of the character's attitude by examining him. Gestures, facial expressions, and tone of voice can all be described as part of the flow of dialogue but (if necessary) implemented separately. *Galatea*, for instance, has conversational replies with blank spots that essentially mean "insert an appropriate gesture here": she might use the same set of words but a different movement or tone of voice depending on the overall atmosphere of the conversation.

Some games with a partially graphical component have tried using an image of the NPC to convey current mood or expression: Chris Crawford's *Erasmatron* dynamically created facial expressions, and *Façade* uses body language and faces in addition to dialogue to convey the moods of the characters. Multimedia IF has not done too much with this possibility so far, but on the other hand, moods and emotional reactions can be expressed textually as well.

Similarly, if the NPC is using some kind of logical model to draw conclusions or pursue goals, it may be worth making that fact explicit as well: when the character realizes something, tell the player what he

realized and why.

This may seem terribly unsubtle; indeed, it goes against an accepted wisdom that one is trying to build a mechanism that doesn't appear mechanical and that the ideal end result will be a conversation that feels real but in which the player is never conscious of how it works. That goal is an interesting one to pursue, but in my opinion (and there are other opinions out there), most interactive fiction is better served by a model that gives the player some clues about what is being modeled and how he can interact productively. If you show the player that the characters are drawing logical deductions from his statements (say), then he will realize that choosing what facts to reveal is an important part of the game and pursue that angle rather than others.

If you can focus your players on the kinds of interaction that you've anticipated and written for, they're more likely to enjoy the work and less likely to run into the boundaries and weaknesses of your system.

Afterword

Except for a few paragraphs of updating, this article is as I wrote it some years ago. Since then, a few things have changed within the IF community, as I've tried to acknowledge. My own perspective has shifted as well. Partly that comes out of a few years' more work in IF; partly out of the commercial game writing that has been my career since 2009.

It's not that what I wrote before is wrong, just that it's not enough. This is largely an article about implementation, because there is very little discussion of implementing conversation out there in the world. But many of the best-loved IF NPCs do not use any especially wild code tricks to accomplish their aims. Michael in *Anchorhead* (Michael Gentry, 998), Dr Sliss in *Rogue of the Multiverse* (C.E.J. Pacian, 2010), almost any of Robb Sherwin's characters—these characters are effective because of the way they're written, not just the way they're implemented. To that end, I recommend people look into the many resources on writing good dialogue for other media. But here are a handful of particular points that arise over and over.

- Write less. Your dialogue could almost always gain from being briefer.
- Wherever possible, write dialogue someone could speak without feeling like an idiot.

- Observe people. Write what could be true, or at least true within the tone of this particular work and the rules of this particular universe. It is very easy to be led away from plausibility by the needs of gameplay.
- Avoid writing on the nose. People usually do not blurt out exactly what they think at the moment they think it. Typically they have an attitude, an angle, or a personal style.
- Know the character, and know the universe. Knowing more about a character's background or environment makes it easier to pull in specific details. Generic content is the devil.
- Where dialogue frames gameplay or provides instructions, lead with the information that the player *must* know to play the game. Allow him to dig deeper into flavorful extras as an option.
- Focus on the two or three main things you want your work to accomplish. Streamline everything else away. Otherwise, you will go mad, and you will also confuse your players.

10 Years of IF: 1994–2004

Duncan Stevens

The following is a slightly revised version of an article published in 2004 in the webzine SPAG, the Society for the Promotion of the Adventure Games, on the occasion of SPAG's 10th anniversary, to note the developments on the IF scene during that period.

By 2004, SPAG had been around for ten years—and what had happened in those ten years? Oh, not much. Consider:

The Rise of Freeware IF

In 1994, much of what was produced in the IF community was some variety of commercial effort—often shareware or crippleware. Legend was on its last legs (*Gateway 2: Homeworld* was released in 1993), the Adventions games saw their last commercial installment with 1993's *Unnkulia Zero*, and amateur efforts like *MacWesleyan* (Neil deMause, 1995), *Save Princeton* (Jacob Weinstein and Karine Schaefer, 1991), *Perdition's Flames* (Michael J. Roberts, 1993), and *Enhanced* (Hans Persson and Dominik Zemmler, 1994) were all shareware. Much of what was made available for free was the leavings of the annual Softworks AGT competition, about which, honorable exceptions like *Cosmoserve* (Judith Pinter, 1991) and *Shades of Gray* (Mark Baker et al., 1992) aside, the less said the better. The 1994 freeware release of *Curses* (Graham Nelson, 1994) heralded a trend of high-quality freeware games (*The Legend Lives!* (David Baggett) in 1994, *Christminster* (Gareth Rees) and *Jigsaw* (Graham Nelson) in 1995, etc.) that left shareware largely a memory in a few years.

The move away from commercial IF was arguably less than salutary for authors, who lost a chance at even the meager compensation available from shareware registrations—but the freeware revolution likely broadened the IF audience (in that new players are arguably more willing to try a free game with no pressure to register) and diversified the IF available (since the decline of commercial avenues for IF left fewer constraints on authors' creativity).

Brevity, the Soul of IF

Relatively short IF games were all but unknown in 1994; *Unnkulia One-Half* (D. A. Leary) was written in 1993 as a teaser for *Unnkulia Zero* (D. A. Leary, 1993), and medium-length fare like *Busted!* (Jon Drukman and Derek Pizzuto, 1993) was rare (and only half-serious). The advent of the annual competition in 1995, with its “One Rule” that entries must be finishable in under two hours, heralded a movement toward shorter games, and popular games like *John’s Fire Witch* (John Baker, 1995) both followed and pushed the trend—fast enough that longer IF, by 2004, was largely unknown, with maybe a release or two each year. (Three of the four Best Game winners between 2000 and 2003 were competition entries.)

As above, this was both good and bad. New writers could break into the IF-writing biz more easily if the median length of an IF release was relatively compact; if all new games were expected to be *Curses*-length, there would likely have been much fewer IF authors. At the same time, however, the trend toward shorter games lessened players’ patience for highly difficult, epic-length IF on the scale of *Jigsaw*; it’s hard to reaccustom oneself to devoting months to a single game when the more common IF experience takes only an evening (and when there are hundreds more worthy efforts in the archive). The diminished appetite for long games in turn made authors reluctant to devote the considerable energies required, and the spiral proceeded from there. How significant a loss this was for the IF world was a matter of taste and opinion, of course, but it was certainly a striking trend.

A somewhat unfortunate example of this was G. Kevin Wilson’s *Once and Future*, a sprawling Arthurian epic written over the course of five years and released commercially in 1998. While reviews were generally positive, interest (at least, as measured by sales) was tepid—in part, it seemed, because epic-length IF had fallen out of style to some extent over the course of the game’s creation, and the ample supply of short, high-quality freeware games narrowed the game’s appeal somewhat.

The Narrative Caught Up with the Crossword

Through most of its history, IF consisted largely of puzzles wrapped in

an ostensible plot premise—sometimes with obvious set-piece puzzles (see *Zork Zero* (Steve Meretzky, Infocom, 1988), and usually with the seams only slightly better hidden. Aberrations like *Trinity* (Brian Moriarty, Infocom, 1986) and *A Mind Forever Voyaging* (Steve Meretzky, Infocom, 1985) aside, most IF had so little plot that it was all but inevitable that puzzles would predominate and largely displace what passed for a story. Many of the games produced in the freeware IF revolution featured stronger stories, however (perhaps because, in shorter IF, it's easier to sustain a narrative arc), spawning a trend toward smoother integration of plot and puzzle (in other words, fewer locked doors that have to be unlocked because no IF player can leave a door unlocked in good conscience, etc., and more puzzles that the PC might actually want to solve). Notable harbingers of this trend were *Christminster*, *The One That Got Away* (Leon Lin, 1995), *Delusions* (C. E. Forman, 1996), and *Kissing the Buddha's Feet* (Leon Lin, 1996); by the 1997 competition, the plots of the top-ranked games (*The Edifice* (Lucian P. Smith), *Babel* (Ian Finley), and *Glowgrass* (Nate Cull)) had become highly focused, and the puzzles to be solved would have made little to no sense in any other context. Increasingly, by 2004, a plot that both made sense and drove the bulk of a game's action had become an expected IF feature, and well-regarded games integrated their plot and puzzles seamlessly. (It would have been difficult to imagine the puzzles in the 2001–2003 competition winners—*Slouching Towards Bedlam* (Star Foster and Daniel Ravipinto, 2003), *Another Earth, Another Sky* (Paul O'Brian, 2002), and *All Roads* (Jon Ingold, 2001)—transposed into any other game.)

This was a welcome development in a lot of respects, but in one in particular: it suggested that the IF medium can be valuable for expressing ideas and telling a story in new ways, not simply for the crafting of puzzles (which was, largely, its original purpose). Some have argued that the ability to not only put the player in the shoes of the protagonist (as static fiction does with first-person narration) but actually direct the protagonist's actions heightens a sense of complicity in the plot as it unfolds and offers an opportunity to make a statement that simply could not be made as effectively in static fiction. Similarly, the experience of interacting with characters and experiencing a setting gives IF a potential emotional impact not available in static fiction—and the best writers found ways to make such an impact, notably Adam Cadre in *Photopia* (1998) and Andrew Plotkin in *Shade* (2000).

Much ink was spilled in the 1994–2004 period on the subject of the

decline of puzzlefests, of course, a large portion of it by yours truly, but the nature of the shift was sometimes oversimplified. Even by 2004, relatively few IF games eschewed puzzles altogether (and even those that did had to contend with players' puzzle-solving expectations; thus, the branching plots of *Galatea* (Emily Short, 2000) were taken by many as an invitation to find each and every narrative possibility), and even after the freeware revolution, there were well-regarded games that could reasonably be viewed as puzzles with some ostensible story (e.g., *Lock & Key* (Adam Cadre, 2002)). The few genuinely puzzleless games that were produced in this period (many in the annual Art Show, a few—*Exhibition* (Ian Finley, 1999), *Best of Three* (Emily Short, 2001)—in the annual competition) garnered respect but hardly set off a stampede. More common were games that used a few puzzles for pacing (*Photopia*, *All Roads*, *My Angel* (Jon Ingold, 2000)) but focus their efforts on plot and character development rather than complex puzzles.

Moreover, it was hard to view puzzle development as a lost art in 2004, as the previous few years of IF development had seen some of the best puzzles ever devised. The language puzzle in *The Edifice*, the entirety of *Rematch* (Andrew D. Pontious, 2000), and a certain puzzle in *Spider and Web* (Andrew Plotkin, 1998) all required persistence, ingenuity, and a bit of lateral thinking. Nor were full-blown puzzlefests entirely things of the past: the previous few years had produced the gargantuan *The Mulldoon Legacy* (Jon Ingold, 1999), *Not Just an Ordinary Ballerina* (Jim Aikin, 1999), and *First Things First* (J. Robinson Wheeler, 2001), each crammed with creative puzzles. The demise of the puzzle, in short, had been greatly exaggerated.

NPC Interaction: More, Better

Part of the rise in the storytelling element of IF was increased attention to NPC development: while many NPCs in the early days of IF simply served as puzzle props (recognize what the NPC wants, give it to him/her, obtain knowledge/object to solve another puzzle—or disable/distract NPC guarding exit/treasure), the latter-day IF revolution increasingly offered more complex characters with whom the PC could interact more extensively. Some of this arose simply from authors giving NPCs more personality—both *Small World* (Andrew D. Pontious) and *Kissing the Buddha's Feet* in the 1996 competition gave key NPCs a wide

variety of one-liners and amusing reactions to the game's events, making them feel like well-developed characters even though the PC didn't need to interact much with them. But the complexity of NPC interactions, in the form of more elaborate conversation systems, also played a key role.

At the forefront of this particular development was Emily Short, whose *Galatea* consisted almost entirely of interactions with a single exhaustively realized NPC, interactions that, both through an ASK/TELL conversation system and a variety of other means, helped develop a highly complex NPC personality; different questions or approaches would elicit different reactions depending on the conversational context and send the relationship between the PC and NPC down a variety of different paths. Subsequent Short efforts, including 2001's *Pytho's Mask* and *Best of Three* and 2003's *City of Secrets*, featured a novel conversation system that blends the freedom of an ASK/TELL interface with the specific phrasings (and associated tone choices) of menu-based systems, allowing for considerably more complex interactions—and more complex characters, like Grant from *Best of Three* and Evaine from *City of Secrets*, have emerged as a result. Other notable NPC-centric games were Adam Cadre's *Varicella* (1999), many of whose NPCs were vividly rendered loathsome characters, and Stephen Granade's *Common Ground* (1999), a shifting-perspective look at complex family relationships. Both the tools and the precedents were there for multilayered NPCs, in short, characters that drive the story rather than merely being cogs in the wheel.

Experimentation

As noted, commercial IF was largely gone by 2004—but with the decline of commercial IF came a great deal of narrative experimentation with the IF medium. Some of it worked better than others, but most of it offered something worthwhile.

Those experiments took a variety of forms, many of which cannot be revealed without spoilers. Authors began testing the waters with puzzleless and puzzle-light IF in 1996 and 1997 (*In the End* (Joe Mason, 1996), *Tapestry* (Daniel Ravipinto, 1996), *The Space Under the Window* (Andrew Plotkin, 1997)) and from there moved on to more radical experiments. Some of the experiments have included novel PC points of view, from dogs (*Ralph* (Miron Schmidt, 1996)) to cats (*A Day for Soft*

Food (Tod Levi, 1999)) to teddy bears (*A Bear's Night Out* (David Dyte, 1997)) to robots (*Bad Machine* (Dan Shiovitz), 1998)) to genies (*The Djinni Chronicles* (J. D. Berry, 2000)), and others took the form of PCs that proved unreliable in a variety of ways. An even more striking experiment was *The Gostak* (Carl Muckenhoupt, 2001), written in a language whose syntax was akin to English but whose vocabulary was entirely unfamiliar, and the challenge was to decipher it sufficiently and interact sensibly enough to solve some simple puzzles. Other notable experiments included *Aisle* (Sam Barlow, 1999) and *Rematch*, both of which offer just one turn (repeated over and over) but manage to provide surprisingly varied and deep exploration of the game's world, and *Heroes* (Sean Barrett, 2001), where the player has a task to achieve and can assume any of five separate roles to achieve it. There were also attempts at literary adaptation (*The Tempest* (Graham Nelson, 1997) and *Nevermore* (Nate Cull, 2000)), surreal/symbolic settings (*So Far* (Andrew Plotkin, 1996) and *For a Change* (Dan Schmidt, 1999)), IF games in reverse (*Zero Sum Game* (Cody Sandifer, 1997) and *Janitor* (Peter Seebach and Kevin Lynn, 2002)), and games where most or all of the challenge is to figure out what is going on (*Shade* and *All Roads*). These and other successful experiments helped pushed the boundaries of what IF authors can do with the craft.

The common thread here is that many, if not all, of these experiments would have been hard to market (at least, the history of commercial IF includes little boundary-pushing as ambitious as the above efforts, which says something); it's reasonable to conclude that the freeware revolution gave rise to an environment that made innovation of this sort possible. It's undeniable, however, that things were done with IF tools in the 1994–2004 period that expanded the frontiers of the possible—at least in this setting. (No, not Z-abuses.)

Multimedia for the Common Man

While most IF in 2004 was still text-only, development tools facilitating the use of multimedia had, if not flourished, at least achieved a modicum of popularity, notably HTML-TADS, Glulx, and Hugo, such that basic graphics and sound files were relatively commonplace. Particularly notable in this regard were *Carma* (Marnie Parker, 2001), which had fairly polished animations and well-produced music, and *Kaged* (Ian Finley, 2000), which had photographs and a nicely mood-

enhancing soundtrack.

This probably did not represent a significant advance in relative terms—the progress of multimedia IF was slow enough, and the enhancement of multimedia in commercial games fast enough, that potential players of multimedia IF who were accustomed to highly vivid graphics and professionally produced sound were likely in for a rude shock. Still, by 2004, there were viable multimedia tools available, which was certainly an improvement on 1994.

Parody/Commentary

By 2004, IF had developed the ability to comment on itself and on its own limitations. Arguably the first example was *Undo* (Neil deMause1995), a peculiar little effort with no puzzles in the traditional sense, one problem that is solved with linguistic trickery, a possible score of 86 points but no actual opportunity to score said points, and a variety of bizarre red herrings. *Zero Sum Game*, where the goal was to undo the entirety of a hack-and-slash fantasy quest and thereby bring the score down to 0 (and the protagonist caused considerably more mayhem in trying to set things right than he/she had caused during the “original” game), was another game that explicitly poked fun at IF conventions, and the list grew from there: *9:05* (Adam Cadre, 2000), *Shrapnel* (Adam Cadre, 2000), *Being Andrew Plotkin* (J. Robinson Wheeler, 2000), *LASH* (Paul O’Brian, 2000), *Guess the Verb!* (Leonard Richardson, 2000), *Voices* (Aris Katsaris, 2001), and *Janitor* all employed self-reference in one way or another to amuse or inform. If nothing else, these efforts suggest that the freeware-era IF medium was stable and defined enough that critique and mockery made some level of sense, which was progress of a sort.

Genre with the Wind

Finally, one noteworthy aspect of 1994–2004 IF development was that most of the better games transcended genre limitations in ways that very little old-school commercial IF managed to do. Whereas much of the most successful IF in the ’80s fell firmly into well-trodden genre categories, much of the most acclaimed IF between 1994 and 2004 avoided such categories.

The competition winners were illustrative over the first nine years of

the competition: the first-place games included an elliptical little nightmare about being stuck in the rain, an allegory of sorts for evolution and civilization, a fragmented tale of untimely death that defies categorization, a medieval-Venice metaphysical-fantasy story involving political scheming and a narrator who moves into and out of various bodies, and a steampunk story set in an insane asylum with a dash of unreliable narrator. Other notables of the period included: an 18th-century France drama with incursions of fantasy-style magic; a palace-intrigue game in an anachronism-heavy alternate-history 19th-century Italy where none of the contenders for the throne, including the PC, are even vaguely sympathetic; a delve-into-your-own-head saga arising from the PC's attempts to quit smoking; a parody of an offbeat indie movie, shot through with IF reference; and a neo-Platonist story freighted with symbolism and object transformation. By 2004, well-regarded IF games that fell within genre boundaries at all were more the exception than the rule: *Anchorhead* (Michael Gentry, 1998), *Spider and Web*, and *Worlds Apart* (Suzanne Britton, 1999) could fairly be called Lovecraftian horror, espionage, and sci-fi games without stretching the definitions too much, but not many other top-flight IF games of the period could be so classified.

That latter-day IF increasingly ignored genre boundaries was noteworthy in a few respects. First, it tended to elevate the significance of story over puzzles, a dynamic discussed above; if the setting and plot are generic fantasy or sci-fi, the game often becomes an excuse for set-piece puzzles (since the appeal of the game tends to lie more in the puzzles grafted into the setting than in a tired plot). If the concepts and settings are fresh, however, the author is less tempted to play them down, or ignore them for long stretches, in favor of puzzles. To be sure, genre does not necessarily mean trite—but without striking innovations that stamp a game as somehow transcending a category (notable examples are *Enlightenment* (Taro Ogawa, 1998), which both inhabits and satirizes fantasy, and *LASH*, science fiction that encourages the player to apply the story's themes on multiple levels), it can be difficult to make genre fiction feel fresh. (The increasing complaints over the years about genre IF, particularly fantasy—without regard for the cleverness of the puzzles in the genre setting in question—underscored the IF audience's increasing dissatisfaction with puzzles as a game's *raison d'être*.)

Second, the decline of genre IF—in particular, the decline of fantasy and sci-fi—suggested a growing acknowledgment that realistic and

complex characters and compelling human conflicts matter, as those genres traditionally have been better on outlandish visions of alternative worlds, or on bizarre occurrences in traditional settings, than on bringing characters to life. (At the very least, the rise in “realistic” high-quality IF that eschews both fantastic settings and strange/mystical events—*Varicella*, *Common Ground*, *Exhibition*, *Gourmet* (Aaron A. Reed and Chad Barb, 2003)—indicates more interest in bringing to life the interactions within those settings than in portraying the fantastic and otherworldly.) Nothing about those hallmarks of fantasy and science fiction precluded character development, of course, but it says something that authors, more and more, no longer needed to rely on the outlandish and unusual as a hook. In addition, whatever one might think of IF’s aspirations toward serious literature, going beyond genre—either by subverting it or disregarding it outright—certainly reflected those aspirations.

>

In short, the 1994–2004 period was quite a time for the development of IF—and while the long-sought foothold in the commercial gaming world, a few heroic attempts to the contrary notwithstanding, had not yet materialized, the medium had become a good deal more mature than it was when the market petered out in the early ’90s.

The Evolution of Short Works

From Sprawling Cave Crawls to Tiny Experiments

Stephen Granade

Short works of interactive fiction offer many pleasures. If the short work is a more traditional game, when well-designed it is a fine Swiss watch, every piece meshing together and turning in harmony, with no unneeded parts. If it is more experimental, it can surprise and delight without becoming tiresome, a brief flurry of fireworks. Short works are constrained, the short story to the novel of longer works, and have become the form of choice for many interactive fiction authors.

Short games have not always been popular. From interactive fiction's beginning in 1976 through the mid-1990s, interactive fiction authors generally ignored the possibility of a short game that was entirely self-contained, rather than a demonstration of or an ad for a long game. While commercial companies had economic pressures that encouraged longer works, even the fledgling IF hobbyist community of the early 1990s focused on writing long works, despite not having the same commercial concerns. But from 1995 to 2001, the number of short games increased dramatically, driven principally by the annual Interactive Fiction Competition. Short games became so prevalent that, in 1998 and 1999, multiple threads in the Usenet newsgroup rec.arts.int-fiction were dedicated to arguing whether or not the annual IF Competition was hindering the release of long games. Given the dramatic shift in game size from 1995 to 2001, it is worth seeing how short games came to be the near-standard.

What is a short game? How do we measure the length of a piece of interactive fiction? Works are conventionally considered long or short based on how long it takes to play them to completion, but no reliable metric exists for measuring the length of play time across even a majority of players. File size alone cannot serve, as some games have large file sizes due to complex code yet can be readily completed in the space of an hour. One of the longest games on record, *Journey to Alpha Centauri (In Real Time)* (Julian Fleetwood, 1998), which would take 3000 years to play to completion, has a file size of a mere 50 kb. Similar numeric measurements, such as the number of rooms, amount of descriptive text, or number of puzzles in a work, correlate weakly with

play time. The length of time to complete interactive fiction is very player-specific and depends strongly on how much time, if any, a given player spends solving a work's puzzles, the player's reading and typing speed, the use of a walkthrough or hints to bypass puzzles, the player's familiarity with IF conventions, and how motivated a player is to finish a work. Since all but a few works of interactive fiction do not occur in real time, players are free to pause for any length of time between moves and consider their next course of action, time that is by custom counted as "play time." Then there are works that can be completed in a few moves yet encourage repeated play. The most notable examples of this are *Aisle* (Sam Barlow, 1999), a puzzleless game that lasts for one turn yet has many possible endings, and *Rematch* (Andrew Pontious, 2000), which also lasts for one turn but has a complex puzzle that must be solved in that one turn.

The annual Interactive Fiction Competition, which was created as a showcase for short games, had to answer the question of "how short is short?" G. Kevin Wilson, competition creator and organizer, settled on a fixed length of play. Judges were to play games for no more than two hours before voting on a game. Though play time varies among players, the average time is unlikely to vary that widely on such a short time scale. In practice this has produced competition games that are commonly regarded as short, and consensus about which games are too long for the competition is readily reached. For the purposes of this article I will use a similarly nebulous definition and say that any work widely regarded as taking 10 or fewer hours to play is short.

Every element of early works of interactive fiction was a reflection of those in *ADVENT* (Crowther and Woods, 1976), and game size was no exception. *ADVENT* was a sprawling game that took place in a large underground land filled with puzzles and treasures. While some players were able to complete *ADVENT* in a handful of hours, many took tens of hours to finish it. Early works mimicked *ADVENT* in size and length of play time. Games such as *DUNGEON* (Tim Anderson, Marc Blank, Bruce Daniels, and Dave Lebling, 1978) and *Acheton* (Jonathan Partington, David Seal, and Jon Thackray, 1978) were large affairs, taking many hours to complete. For a while, each successive game was larger than the last in an ongoing game of one-upsmanship, driven by programmers' natural inclinations to top each other.

The trend towards increasing size was temporarily reversed when interactive fiction moved from mainframe computers to personal ones.

Personal computers had small amounts of memory—the standard maximum memory of 16 kb on the TRS-80 Model I was typical—and programs were stored on cassette tapes. Scott Adams led the way in this field, becoming the first to sell interactive fiction commercially for play on personal computers. The adventures he wrote for his company Adventure International were tiny and used every imaginable trick to squeeze more game into the finite space available. His tricks were of sufficient interest to warrant an article in the December 1980 issue of *Byte* magazine. Adams's adventures had far less prose, both in length and complexity, than *ADVENT*. Consider the following room description from *Pirate Adventure* (Alexis and Scott Adams, Adventure International, 1978):

I'm in a sandy beach on a tropical isle
Obvious exit: EAST
Visible items: Small ship's keel and mast, Sand,
Lagoon, Sign in the sand says: 'Welcome to Pirates
Island, watch out for the tide!'

When Infocom was formed, its first product was *DUNGEON* ported to personal computers. To make this possible, *DUNGEON* was cut into three parts, and each part rounded out to make a complete game. The results were *Zork I*, *Zork II*, and *Zork III* (Tim Anderson, Marc Blank, Bruce Daniels, and Dave Lebling, Infocom, 1979). Yet despite their small file sizes, works such as the three *Zork* games and *Pirate Adventure* took a while to play. And works did not become any shorter as time went on.

Economic pressures, more than mere homage to *ADVENT*, drove the size of commercial works of interactive fiction. Once companies began charging for interactive fiction, players expected a certain amount of play time for their money, and those companies responded by boasting about how long their games took to complete. While most companies bragged about their games' putative hours of gameplay, Level 9 went a step further and trumpeted the 7,000 rooms of their adventure *Snowball* (Mike Austin, Nick Austin, Pete Austin, and Ian Buxton, Level 9, 1983). (While a true boast, some 6,800 of those rooms formed a maze consisting of near-identical rooms.) In addition, many companies used hint books as a revenue stream. The best way a company could insure the sale of hint books was to include fiendishly difficult puzzles in the company's works, which in turn led to longer play times for those who refused to buy hint books.

At the same time, economics, not just available memory and storage, kept works from becoming extremely long. Longer works require more time to design, develop, and debug than shorter ones. Combinatorial explosion becomes a problem, as writers must consider how every new room and item will interact with other items. As more memory and storage space became available, works became deeper, with the addition of more prose, items, and non-player characters (NPCs), rather than more game play *per se*.

This situation remained more or less unchanged throughout the 1980s and on into the 1990s. Even as the commercial market for text-based interactive fiction declined, game length changed little. Given this history, it is unsurprising that early works from the nascent interactive fiction community were similar to commercial ones in size and scope, limited mainly by the capabilities of interactive fiction languages of the time, such as GAGS and AGT. A goal of many authors who took part in the Cosmoserve online community for interactive fiction was to make games just like Infocom's. AGT games like *Cliff Diver*, *Investigator for Hire: Crime to the Ninth Power* (Patrick Farley, 1990) and *The Multi-Dimensional Thief* (Joel Finch, 1991), TADS games like *Save Princeton* (Karine Schaefer and Jacob Weinstein, 1991) and *Unnkulian Underworld: The Unknown Unventure* (D. A. Leary, 1990), and independently programmed games such as *Jacaranda Jim* (Graham Cluley, 1987) were of moderate-to-large size, comparable to the sizes of many commercial games of the 1980s.

This was further reinforced by Activision rereleasing all of Infocom's games in two volumes, *The Lost Treasures of Infocom* (Activision, 1992) and *The Lost Treasures of Infocom II* (Activision, 1992). The collection made all of Infocom's catalog available for around \$100, a significant saving over buying the games individually, and resulted in an influx of new people into the interactive fiction community. Many of the newcomers were interested in writing their own games and understandably took Infocom's games as their standard.

Short games were rare, and the ones that did appear were mostly sections from a longer work made available for demonstration purposes or, like *Unnkulia One-Half: The Salesman Triumphant* (David Baggett, 1993), self-contained games that served as ads for other, longer games.

1995 marked a turning point due to two major events. The first was the release of *John's Fire Witch* (John Baker, 1995) on January 30, 1995. Described by author John Baker as a "snack-sized game," the work was very favorably received. There had been short games before, such as *The*

Sound of One Hand Clapping (Erica Sadun, 1993), but *John's Fire Witch* had three advantages. One, it offered traditional Infocom-style gameplay. Two, it was highly polished, with logical puzzles and solid implementation. Three, it was developed using TADS, the most popular IF design language of the time, and so many IF players already had the necessary interpreter to play the game. Because of these advantages, *John's Fire Witch* was widely played and discussed, and it introduced the IF community to the potential of short games.

The second event was the creation of the annual Interactive Fiction Competition. The competition was organized by G. Kevin Wilson and the denizens of the two Usenet newsgroups dedicated to interactive fiction, *rec.arts.int-fiction* and *rec.games.int-fiction*. The competition's initial purpose was to increase the number of works written using the interactive fiction language Inform; only after much discussion was the competition opened to games written in TADS as well. The decision to limit the competition to short games was a pragmatic one, as judges were required to play all entries, and who would have the time to play five or six longer works? Thus the first competition had but one rule: games had to be designed to be completed within two hours of play.

Other than the lengths of entries, what was most notable about the competition was the plethora of game reviews and discussion that came on the heels of the competition's conclusion. Kevin Wilson asked the community not to discuss the games during the adjudication period, resulting in a pent-up flood of opinions bursting forth after the competition. Entries were heavily discussed and reviewed, garnering feedback far in excess of that enjoyed by non-competition games.

These two decisions about the competition kick-started short games. The feedback available through the competition encouraged authors to enter the competition, and the two-hour rule forced them to make at least a passing nod at writing a short game. Even the longest competition games are short by Infocom standards.

The early short games had a new length but stuck with traditional gameplay. Certainly *Uncle Zebulon's Will* (Magnus Olssen, 1995), which won the TADS division of the competition, was much like *John's Fire Witch* in feel and style. Even *A Change in the Weather* (Andrew Plotkin, 1995), winner of the Inform division, was traditional in style despite its cruel scheduling of events and rigid in-game timeline. The most unusual game of the competition was *Undo* (Neil deMause, 1995), which made meta-contextual jokes about adventure games and Heidegger-esque

being and not-being. It finished last in the TADS division.

Most short games entered in the next several annual competitions remained traditional in style and design. *The Meteor, the Stone and a Long Glass of Sherbet* (Graham Nelson, 1996), which won the 1996 Interactive Fiction Competition, could easily have been an Infocom-era work that just happened to take around two hours to complete instead of twenty. Even then, though, authors were experimenting with what was possible in short works. Andrew Plotkin wrote two early experimental short games: *Lists and Lists* (Andrew Plotkin, 1996), a compiler for the computer language Scheme coupled with a Scheme tutorial, and *The Space Under the Window* (Andrew Plotkin, 1997), which was more an interactive presentation of text than a traditional game. Other authors were also experimenting with the form. *Pick Up the Phone Booth and Die* (Rob Noyes, 1996) was a Dadaist game whose title is entirely descriptive: the work contains one room and one item, a phone booth. If you pick up the phone booth, you die.

Short works' evolution was sped up in 1998 by the introduction of mini-comps and SpeedIF. In May of 1998, Lucian P. Smith and Adam Cadre both began mini-comps, small competitions in which entrants were to write short works that fit a given premise. Smith's mini-comp involved a rich relative leaving behind a wacky will and a scavenger hunt, while Cadre's merely specified that the game should involve chickens. The two mini-comps resulted in 25 works between them, all short.

In October of 1998, David Cornelson held the first SpeedIF event on ifMUD, an online meeting place for members of the interactive fiction community. People who were online and logged into ifMUD at the time generated a list of suggested requirements in the style of a mad-lib: Cornelson would call out a request for things such as "what you might find in a glass" and collate his favorite responses. Entrants then had two hours (or, in many cases, up to 12 hours) in which to write a short work that fit the chosen requirements. Both the mini-comps and SpeedIF sparked the creation of short, often silly works whose content was as likely to be a convoluted community in-joke as a carefully designed traditional game. SpeedIF also provided a new approach to writing interactive fiction quickly, as Rob Wheeler discussed in his article "Make IF Fast!" (2001, http://raddial.com/if/theory/make_if_fast.html)

If the annual IF Competition had increased the acceptance of short works in the community, the mini-comps and SpeedIF helped make short works the preferred form for experimentation and one-off jokes

and made competitions the standard way of encouraging new short works on a given theme. For instance, Marnie Parker created the IF Art Show in 1999 for works that expounded upon a single theme rather than being complete games in their own right and that created a single object, scene, or character and explored it fully. The competitions served as a petri dish in which unusual experiments could be bred. The result was a slew of forgettable games but with the occasional standout, such as *Galatea* (Emily Short, 1999). *Galatea*, the eponymous non-player character of that game, was at the time the best NPC and is still a benchmark against which other NPCs are measured. *Galatea* appears in a work devoid of the usual trappings of interactive fiction such as objects, puzzles, or even multiple rooms. Similarly, *The Fire Tower* (Jacqueline Lott, 2004), entered in the 2004 IF Art Show, let players hike around a puzzle-free environment, one so detailed that the work won the XYZZY Award for its setting. Even SpeedIF produced notable games, such as *You Are a Chef!* (Dan Shiovitz, 2000), a humorous game involving collecting random food that falls from the sky with memorably gonzo writing, and *ASCII and the Argonauts* (J. Robinson Wheeler, 2003), a send-up of Scott Adams games with surprisingly solid game design underneath the terse descriptions.

One 1998 work had arguably the greatest influence on later short works: *Photopia* (Adam Cadre, 1998), winner of the 1998 Interactive Fiction Competition. *Photopia* combined multiple viewpoints and a nonlinear narrative in a way that had not been seen in interactive fiction before. It eschewed traditional puzzles for narrative drive and a tightly constrained plotline. Many authors later imitated elements of *Photopia* in their own short works. By winning the annual Interactive Fiction Competition, *Photopia* marked the acceptance of “experimental” interactive fiction by the mainstream community, though not without controversy.

During this period, literary and design theory had risen in prominence in IF discussion. Authors used short works, with their rapid development cycle, to experiment with new techniques arising from the theory discussions. In 1998 Andrew Plotkin posted a humorous message about what he wanted to see in IF in the next year, including “Time limits instead of space limits—games with lots of rooms, but which last exactly one turn” and “Games written in languages that nobody understands, possibly not even the author.” *Aisle* provided part of the first by allowing players one and only one turn. It had only one room,

but packed into the game's one turn were some four distinct stories that players could discover through extensive replay. *The Gostak* (Carl Muckenhoupt, 2001) provided the second by dispensing with known language altogether, using instead a language whose grammar was similar to English but whose invented vocabulary was not defined. *For a Change* (Dan Schmidt, 2000) also played with language, using altered English words and phrases that nevertheless conveyed meaning. Games like *Heroes* (Sean Barrett, 2001) and *Common Ground* (Stephen Granade, 1999) employed multiple viewpoints in the same time and place. *shrapnel* (Adam Cadre, 2000) is a deconstruction of interactive fiction that, in the end, dissolves into a jumble of random characters—random, except for one encoded message. By the end of 2001, writing short games to illustrate design theory points was so common that, in 2003, Mike Roberts tried to answer whether IF's standard compass directions were really to be preferred over relative directions by writing *Rat in Control* (Mike Roberts, 2003), in which players could choose whether to use compass directions or left-right-ahead-behind directions while navigating a maze.

One measure of short works' ascendancy is the XZZZY Awards. In late 1996 Eileen Mullin, editor of the e-zine XZZZYnews, created the XZZZY Awards. The awards honor works in categories such as Best NPCs, Best Puzzles, and Best Game and are chosen by any members of the interactive fiction community who wish to vote, thus favoring games that have been more widely played. Several games are nominated in each category, after which a single winner is chosen. The XZZZY Awards serve as a measure of what games are well-regarded and widely played by the IF community.

The 1996 awards were dominated by the long game "So Far" (Andrew Plotkin, 1996), but every XZZZY award for 1997 went to a short game. All but *I-0* (Adam Cadre, 1997), winner of Best Game, were entries in the 1997 Interactive Fiction Competition. Despite the inroads made by short works of interactive fiction, most of the 1998 XZZZY Awards went to longer games such as *Spider and Web* (Andrew Plotkin, 1998), *Once and Future* (G. Kevin Wilson, 1998), and *Anchorhead* (Michael Gentry, 1998). Only two short games, *Photopia* and *Little Blue Men* (Michael Gentry, 1998), received awards, with each garnering one award apiece. In 1999, the XZZZY Awards were split between long works such as *Varicella* (Adam Cadre, 1999) and *Worlds Apart* (Suzanne Briton, 1999) and short ones such as *For a Change* and *Hunter, in Darkness* (Andrew Plotkin, 1999).

1999 was the last year that the XYZZY Awards saw multiple long games win awards. In 2000, despite the release of such lauded longer works as *LASH* (Paul O'Brian, 2000), *Augmented Fourth* (Brian Urii, 2000), and *Dangerous Curves* (Irene Callaci, 2000), every one of the 2000 XYZZY Awards went to short works. In addition, 2000 marked the first time that a SpeedIF game, *You Are a Chef!*, was nominated for an XYZZY Award. In 2001 only one long work, *First Things First* (J. Robinson Wheeler, 2001), was released. During the same year, there were 52 games entered in the annual Interactive Fiction Competition, nine in SmoochieComp, six in TelegramComp, four in the IF Art Show, 17 IF Arcade Games, and many SpeedIF games, not to mention *Textfire Golf* (Adam Cadre, 2001) and *The Weapon* (Sean Barrett, 2001). In just seven years the balance between short works and long ones had completely reversed.

Since the period of 1995 to 2001, these trends have only continued. The IntroComp, begun in 2002, is for games' beginnings, to encourage authors to polish their games' opening. Few IntroComp games are ever completed, making the competition's true legacy a collection of openings to non-existent games. In 2010, the casual games website Jay Is Games held a competition for interactive fiction. The competition's theme was escape, and the organizers encouraged authors to constrain their games to one room. This was partially to capitalize on the popularity of escape-the-room games, short Flash games that present the player with a series of context-less puzzles that must be solved in order to escape the room.

Long works still exist, and there have been efforts to increase their number. In 2002, Adam Cadre created the Spring Thing, a competition designed to be the antithesis of the annual IF Competition. The Spring Thing is offset by half a year from the annual IF Competition, is open to long works, and requires authors to pay into a prize pool in order to discourage unfinished and joke games. The Spring Thing has had limited success, though, both in length of games and in number of entries. The length of most Spring Thing entries has been comparable to those of the annual IF Competition. In terms of number of entries, Cadre set a maximum number of entries of 20, but the 2002 Spring Thing only had one entry, *Tinseltown Blues* (Chip Hayes, 2002), while the 2003 Spring Thing had four entries. Some authors continue to release mainly long works, such as Robb Sherwin, whose *Necrotic Drift* (Robb Sherwin, 2004) won an XYZZY Award in 2004. Other long works have won XYZZY Awards, including *City of Secrets* (Emily Short, 2003), *Savoir-Faire* (Emily

Short, 2002), and *1893: A World's Fair Mystery* (Peter Nepstad, 2002). More recently, *Blue Lacuna* (Aaron Reed, 2008) won four XZZZY Awards in 2009. Despite these examples, short works continue to dominate the XZZZY Awards, and many of those short works were entered in the annual IF Competition.

This focus on short works over longer ones has increased experimentation and the rate at which new ideas about IF spread into the community. The trade-off has been a decrease in narrative content. The narrative pace of IF is typically slower than that of static media such as books and requires more work on the part of the author to create. Long works of IF have narrative content comparable to a novella, while short works are closer to a short story. Short works are thus very constrained in terms of their narrative complexity and character development. The dearth of long works limits the amount and complexity of narrative players will experience, but short works are more easily completed by players who have many other forms of entertainment and art from which to choose. They are also viewed as the best starting point for novice authors. In his book *Creating Interactive Fiction with Inform 7* (2010), Aaron Reed writes, "I'd highly recommend that your first few IF projects be short story length or even shorter. . . . You'll get a bigger sense of accomplishment for actually finishing something, more feedback, and can get started on your next, better story right away."

Shorter works will remain the most popular length for the foreseeable future. They give authors the chance to experiment with ideas or gimmicks that would be stretched thin if used in a longer work. They are more quickly written and more easily finished, as longer works may take multiple years to write, and provide authors with player feedback much sooner. In the case of entries in the annual IF Competition, the amount of feedback and number of reviews will likely be an order of magnitude greater than that associated with non-competition games. The trend towards short games has been reinforced by the rise of casual video games, ones that are easily played in a lunch hour and do not require a significant investment of time or effort. Given this dynamic, it is doubtful that long games will once again become the default form.

History of Italian IF

Francesco Cordella
www.avventuretestuali.com

“The best thing would be a parser with just one word:

>BARMAN

Barman: What?

>BEER

Barman: A beer for you.

>MONEY?

Barman: Five bucks.

>PAY

Barman: Thanks.

So, this series of sentences seems perfect to me, maybe I'll use it to write a game.”

The man who said this is Enrico Colombini, a programmer from Brescia (northern Italy) and the pioneer of adventure games in Italy—the Will Crowther of the Mediterranean, just to be clear. He has a curious aversion to the sophisticated parser invented by Infocom, and that's why he proposed such a bizarre idea: the one-word parser. Fun, fantasy, and easy, straightforward work: no complicated sentences and no NPCs to ask “What's wrong?” It's certainly provocative, but it's a way to focus mainly on the story and to create an “interactive fiction” for the modern era in Italy, where this genre seems dead: less puzzles, more story.

The First Game

An outburst from the man who, in 1982, wrote the first text adventure in Italian history, for Apple II computers: *Avventura nel castello* (*Castle Adventure*).

“I didn't know Infocom games yet and, certainly, that was a good thing”, he says. “Had I played them, I would have tried to emulate their parser: *put the red sheet and the green one in the big drawer, then close it with the strange key*. Things would have gotten really complicated. I would have lost a lot of time, and maybe I would have given up.

“At the time, I hadn't yet discovered that easy things are, often, the best way to gain victory. I needed a lot of years to understand this, but

that's another story."

Twenty years later, *Avventura nel castello*, the story of a blank hero (a man with neither a story nor a past) trapped in a magic castle, can still compete with most Italian adventure games. At the time, without exaggeration, the game—for its sheer quality—would have turned Scott Adams pale.

Avventura nel castello was at first sold only in a shop owned by a friend of Colombini, and later it was distributed by Gruppo Editoriale Jackson. In 1985 the MS-DOS version of the game was released (the last edition is dated 1996). "The sales were never high, but the satisfaction was great because a lot of players wrote to know more about the game. . . . I earned, if I remember right, three or four million. The same as Scott Adams—but my millions were in lire, his in dollars." (Three or four million lire were equivalent to six or eight thousand dollars at the time.)

The game didn't have a large distribution because it ran on the Apple and then on MS-DOS, when the most common home computers were the Commodore 64 and the Spectrum.

On the shelves of computer shops, it was easier to find "games from abroad": Infocom, Level 9, Adventure International by Scott Adams. The problem was that text adventures were still games for a small audience, young people able to understand English (at the time, in Italy, only young people played videogames). For this, the most appreciated games were those of Scott Adams: not much text to translate, an easy parser, speed, and, in some cases, graphics (and graphics, whatever you might think, means audience).

That is, until someone realized that adventure games could have a larger market in Italy, as well.

The Newspaper Stands

Things changed when games arrived at the newspaper stands. Yes: the newsstands became gold mines for adventurers. In April 1985, the first issue of a monthly publication called *Next Strategy* was released: on the tape were three graphic adventures for the Commodore 64, written by Roberto Tabacco and Hans Piu. The games were *Le avventure di Jack Byteson: Il tesoro di Yumak* (*Adventures of Jack Byteson: The Treasure of Yumak*), *Il segreto della Fenice: Tempio delle Illusioni* (*The Secret of the Phoenix: Temple of Illusions*), and *Dust Hanter*.

The names of the characters were Anglophone instead of Italian because they sounded more fascinating to the audience: Jack Byteson, Rex Wright, and Dust Hanter, a detective that became the hero of text adventures at the time, the star of a series.

If you play them today, a great pleasure comes from satisfying your nostalgia. The games are fascinating because they were the first interactive text games in Italy, but most everything else about them is frustrating. It's impossible to say that these games have a parser. The game understands only the precise sequence of commands that lead to victory—nothing outside the solution scheme, no provision made for responses (even silly ones) for wrong actions. We are far from *Avventura nel castello* and also from Scott Adams. Regression? Maybe. To win, it was not enough to solve the puzzles; you also had to guess the exact words in the mind of the author. Eventually, this reaches an extreme, like a game that doesn't understand the word "follow" but only the word "track."

Certainly there were reasons for this. At the time, this was more or less the standard (as it was in other countries, too). Games like that were easier to write and easier to play. For this historical reason, Italian players, especially those who don't know English (and there are many), are not used to a complex parser. It sounds bad to their ears. At the time, nobody dreamed of emulating Infocom. The magazine was sold every month, and you could not expect that the editor would hire a team of programmers and beta-testers. And everyone was happy with the results because the players were enthusiastic—and that was enough to continue without increasing the budget (which was already very high for such games).

For the same "market reasons," the games were very difficult. In fact, the magazine had a high price: eight or nine thousand lire (roughly 16 or 18 dollars at the time). Had the games been very easy, the line of people requesting a refund would have been longer than the line of people ready to buy the next issue with new games and the solutions of the old ones. The strategy, instead, was to insert a very difficult puzzle in one of the games so that the players had to buy the next issue.

More or less at the same level were the games of the *Dream* series in Editions Fermont. The first issue was in November 1985, and every issue had three games for the Commodore 64. They had black-and-white graphics, a retro style, and good oneiric plots. In every issue there was a game without graphics, to make the most of the memory of the Commodore 64 and offer more text and more puzzles.

The publisher, Arscom, launched a new series, *Epic 3000*, with three games every month. The first issue was sold in April 1986. Roberto Tabacco was the author of the ones for Commodore 64, and Bonaventura Di Bello wrote the ones for Spectrum (with the programs The Quill and The Illustrator). Here, the influence of the Level 9 games was clear.

In December 1986, the last issue of the magazine was published; it then became *Viking* (first issue: January 1987), with a new publisher, Edizioni Hobby. Tabacco was no longer working there, and the games (three, the same for Commodore 64 and Spectrum) were all written by Di Bello. Two months earlier, in November 1986, Edizioni Hobby had launched the series *Explorer* (three games, the same for CBM and MSX); here, Di Bello ported to the Commodore the Spectrum games from the *Epic 3000* series. Sometimes, a game continued in two or more episodes.

The Quill and Higher Quality

The quality of the games got better with Di Bello. The games did not only understand the actions that lead to victory, but many others as well. The stories were well written and had a good variety of genres: horror, thriller, science fiction. At the time, it was a pleasure to play a game written by Di Bello. He wrote good games because he had a good background as a player. It all began when, in his 20s, he bought by chance *Adventure A: Planet of Death* by Artic Computing (while the other Italian pioneer, Colombini, was hypnotized by Crowther and Woods, authors of *Adventure*). A little time later, Di Bello discovered Infocom games. Then he found a tool to write games by himself, the “adventure games generator” The Quill (an illegal copy, of course, but then he later bought an original one from England), and wrote a game on his Spectrum. He sent it to a magazine called *Load’n’Run* and, in that moment, began his fortune. “I was paid 200 thousand lire (100 dollars at the time),” he said. “Another two and I would have been able to buy a new Spectrum!”

Di Bello wrote a total of 74 adventure games. He was able to write a game every week and was paid 500 thousand lire (250 dollars) per game. “It would have been wonderful,” he said, “to live writing those games; it would have been the same as living as a writer, or maybe better, because the audience of text games was superb.”

Not Only Verb/Noun

The parser was still predominately verb/noun, but there were some exceptions, above all in single games or in short series such as the mystery stories published by Systems (*Gialli Commodore*, just one game per issue). These games were well written and, among the games of the time, are the most similar to interactive stories. Consider *Il mistero di Zambesi Waters* (*The Mystery of Zambesi Waters*) by Sandro Certi and Franco Todi. The location is Africa, and in the opening screen there is a promise: “This game accepts sentences with one or two words, with a maximum of NINE.” It is possible to type something like “jump down from the bus.” However, those are predefined sentences that are good only for one event, one puzzle. If, in another and not predefined situation, you type “jump down,” inevitably the game says “I don’t understand” (and not “Jump down from what?”). A trick, sure, an illusion . . . but also an experiment.

The System Inside and Out

The popularity of text adventures prompted Italian software houses to create writing tools. To write a good adventure game, with an elementary parser, is undeniably easier than writing a good shoot-’em-up. So the market was open to this kind of experiment.

To do so, you could find in newsstands (of course) an illegal Italian version of *The Quill*. But there were also legal ways to write games.

The Systems company produced a series of lessons to write text adventures published by the magazines *Personal Computer* and *Commodore Computer Club*. The lessons were very good, and in them appeared, for the first time to Italian players, the concept of a library and a world model, with chapters covering the theory of locations and inventory and suggestions for an “intelligent” parser.

In 1986, Enrico Colombini, the pioneer author of *Avventura nel castello*, wrote the book *Scrivere un gioco di avventura sul personal computer* (*How to Write an Adventure Game*), which had great success. With the book came a tape with some adventure games and the famous Modulo Base, a system, a standard library of verbs to enlarge to create your own game using BASIC.

Many less-than-brilliant games were written using Modulo Base; the

memorable exceptions are by Colombini, among them *L'Anello di Lucrezia Borgia* (*The Ring of Lucrezia Borgia*), *L'Apprendista Stregone* (*The Sorcerer's Apprentice*), and *L'astronave condannata* (*The Damned Starship*). Di Bello, too, used Modulo Base, and here his history as an author of text adventures meets that of Colombini.

"I used Modulo Base to realize a sort of 'The Quill for MSX,'" he said. "I used it to create a dark/light routine (with timers for lamps or other light objects), and other things offered by The Quill. Then I tried to do the same with Amiga's Amos Basic, but it was too late to launch text adventures again and the project never took life. But I'm sure Enrico remembers that night when I woke him to ask a suggestion—I hope he forgave me!"

The "adventure toolkits" opened the market to new authors, but the truth is that only one game was destined to be remembered: *Avventura nel castello* by Colombini, the first one, the best one. None of the others excelled. In Italy, at the time, none wrote a new masterpiece after *Avventura nel castello*.

The Dark Age

The lack of masterpieces and a market made of games that were all similar contributed to reduce the Italian audience's interest in text adventures. So, as in the rest of the world, at the end of 80s Italian IF entered the dark age. The age of games sold in newspapers stands came to a close with the series *Epyx 3001* (published by Arscom, first issue January 1988). Every issue had five or six games only for the Commodore 64, but the sales were poor, and the series closed after six months. There were few games written at the end of the 80s and at the beginning of the 90s, and almost none that sold. There were, however, some bad attempts to realize games using well-known comic characters like Diabolik and Dylan Dog, but the results were generic action adventures with many lows and nearly no highs.

One of the most interesting turns of events happens in 1990, when the magazine *Amiga Byte* published Demo Adventure, an adventure toolkit (written by Maurizio Giunti) that enabled you to create games with a good parser ("attack orc with the sword") but that, curiously, didn't allow synonyms for the nouns (while it did for verbs).

With this system, Marco Vallarino, a man from Imperia

(northwestern Italy), wrote nearly 20 text adventures, among them *Dracula*—notable because it offers a lot of locations (84). At the same time, there was an innovative experiment by Alessandro Uber and Fabrizio Venerandi. They created a text adventure for Multi-User Dungeon (MUD), *Necronomicom*, played via Videotel—a system of connected computers used by pubs and restaurants to attract guests—until 1994 and via telnet since 2001.

In 1995, there were a few other “flames” in the dark age. Vallarino wrote *Sfida all'ignoto* (*Battle against the Unknown*), an MS-DOS game with an Infocom-like status line, whose first scene is similar to the opening scene of *Avventura nel castello*: an airplane crash. Robero Barabino wrote a very nice but underrated game, *Alieni per sempre* (*Aliens Forever*), a text adventure with graphics that is a parody of the TV series *Star Trek* and that bears structural similarities to the post-Magnetic Scrolls game *Wonderland*.

In any case, these games didn't have much audience: the Internet was just beginning in Italy, and they were mostly distributed by word of mouth.

The Revival

Everything changed with the arrival of Inform in the late 90s. (Today, there still don't exist games written with TADS or Hugo, because of the lack of Italian libraries.) Ilario Nardinocchi wrote the first Italian Inform libraries, while Luca Melchionna and Barabino wrote the first Italian Inform games: *Zazie*, inspired by the book of French writer Raymond Queneau, and *Non sarà un'avventura* (*This Is Not an Adventure*), a funny game that takes place on a beach full of people and girls to catch. The year is 1999, and thus begins a new era for text adventures in Italy, thanks also to word of mouth via it.comp.giochi.avventure.testuali, the Italian IF newsgroup.

There were still not so many fans of these kind of games, but the Italian IF community was strong, and it launched an award for the Best Adventure of the Year: the first winner was Barabino with *Non sarà un'avventura*.

In the meantime, Giovanni Riccardi rewrote the Italian libraries, opened an Italian website for Inform, and translated, for didactical purposes, *Adventure* and *Toysshop*. On the scene then arrived Tommaso

Caldarola, who wrote the first game using the Riccardi libraries: *Uno zombie a Deadville* (*A Zombie in Deadville*), horror splatter winner of the 2001 Italian Adventure of the Year award. Caldarola also wrote another interesting game, the western *Pecos Bill*.

But the most prolific year was 2002. Vallarino gained much more success with the game *Enigma*, very popular among Italian players, winner of the Italian Adventure of the Year, while my mystery game *Flamel*, runner-up, was also much appreciated by the Italian scene. At the same time, for the first time an Italian author entered international IF Comp. Daniele Gewurz entered his game *Ramon and Jonathan* (36th place out of 38).

Since then, the Italian scene has experienced ups and downs. One month, on the newsgroup it.comp.giochi.avventure-testuali, there would be a lot of people discussing games, theories, and puzzles, and the next month, nothing. Most of discussions revolved around a central question: is it better to write interactive fiction (more story than puzzles) or text adventures (more puzzles than story)?

In the end, the best results were obtained with “interactive fiction” games, like the ones of Roberto Grassi. He, with his team Mondì Confinanti, founded also by Paolo Lucchesi and Alessandro Peretti, developed solid and interesting graphic adventure games like *Little Falls* and *Beyond*, which won second place in IF Comp 2005.

Another good example of an interactive fiction game, mostly story-oriented, was *Natalie*, a horror-thriller by Fabrizio Venerandi.

But it is hard to maintain the interest of the audience, which is very poor in Italy. I tried to revitalize the scene, organizing, with the help of Giovanni Riccardi, the One Room Game Competition (<http://www.avventuretestuali.com/orgc>), which revealed a “rising star” like Massimo Stella, author of well-written games such as *Sotto la pioggia* (*In the Rain*). The problem is that, in the last two or three years, not many games were produced, and the audience vanished.

>

So, the question is: is interactive fiction dead in Italy? Maybe. Anyone can find most Italian games on <http://www.ifitalia.info>, but the question is how to resuscitate the interest in these games and to gain a young audience. One solution is offered by Fabrizio Venerandi, who now produces interactive stories without a parser and involved the legend

Colombini in his project (<http://www.quintadicovertina.com/>).

Another solution is offered by Marco Vallarino, who just recently released a new game, *Darkiss*, an “old school” text adventure about a vampire trapped in a dungeon, which has been quite successful and has revived some interest in these kind of games.

But maybe the big solution is to remember the lesson of the pioneer Enrico Colombini: to restart, writing easy games with one-word parsers, which might also be good for new technology such as smartphones.

Racontons une histoire ensemble

History and Characteristics of French IF

Hugo Labrande

From the beginning to the present day, it seems that the language of interactive fiction is for the most part English. The first interactive fiction, *Adventure*, developed by Will Crowther, was written in English, modeled after a cave in Kentucky, and spread via the ARPANET, which was a strictly American network. Later, Infocom wrote games that are considered the canon of interactive fiction, again in English. In the early 90s, TADS and Inform were developed by English speakers, and the majority of the games that were subsequently developed with those two authoring systems were in English. As a matter of fact, at the date of the writing of this article, there are 3732 games in the IFDB, of which 388 are not written in English¹: 90% of all interactive fiction is written in English. The majority of authors, reviewers, and IF critics are thus English speakers, and interactive fiction is mainly an English-speaking genre.

But interactive fiction in other languages exists, though in smaller numbers. As a matter of fact, Inform and TADS have been translated for other languages; translations of Inform libraries are publicly available for eight other languages,² which means one can create a game in each of these languages. The biggest non-English speaking community is the Spanish IF community, mainly centered around the CAAD (Club de Aventuras AD); as for the second biggest, it is unclear: the German community appears to have a lot of games but was thought dead a few years ago³ (although it was apparently reborn last year); the Italian community was very active at the beginning of the 2000s, but their activity has declined since then; the French community enjoys steady activity but possibly fewer games.

As pointed out by a few critics, among them Jeremy Douglass,⁴ the

1 As of January 4, 2011, the count is as follows: 142 games in Spanish, 139 in German, 55 in French, 25 in Italian, 17 in Swedish, 4 in Dutch, 2 in Russian, and 1 (often a demonstration game) in Esperanto, Norwegian, Portuguese, and Slovenian.

2 Found at <http://www.inform-fiction.org/translations/complete.html>

3 rec.arts.int-fiction topic "German IF is dead", June 21, 2007.

4 *Command Lines: Aesthetics and Technique in Interactive Fiction and New Media*, page 17, note 3.

dominance of English in interactive fiction means that most of the histories of the genre are centered around works written in English and thus mention Infocom games as the canon of interactive fiction, from which everything else was derived, and acknowledge them as a major influence, if not the biggest. But as noted by some, while Infocom games were a huge success in North America, their success in other countries where they were also available varies greatly. Therefore, how can we talk of *a* history of interactive fiction that mentions Infocom as a major influence when every other community grew up without Infocom games? As a matter of fact, each other language community has its own history—one could say *parallel* histories—of the development of interactive fiction. Those histories are certainly interesting, as they might, for example, provide other perspectives about the market of interactive fiction (did IF die with the fall of Infocom, or was that a more general trend of the video game market?) and also give the opportunity to those communities to establish themselves as independent communities with their own interactive fiction culture.

Unfortunately, as of today, such histories are sometimes incomplete, and few are available, if very detailed. A few of them have been written by various members of the communities and published in SPAG: a timeline of French IF was written by Grégoire Schneller (“Eriorg”) and is featured in SPAG#47, Russian IF was covered by Sergey Minin in SPAG#48, the history of Spanish IF written by Pablo Martinez Merino (“Depresiv”) was published in SPAG#49, and a special feature about Italian IF (and its history, written by “torredifuoco”) was published in SPAG#51. Those histories are in general fairly long and well-crafted and provide a timeline of interactive fiction in those different communities and languages; however, they remain the only ones of their kind and thus are incomplete: there is much more to say, to study, to look for. Those histories need to be examined more, for they can teach us much more; I hope that in the future, we will see more studies of the genre in other languages, tackling some aspects of interactive fiction with another, different perspective.

As a member of the French-speaking interactive fiction community, I can only speak about this community and its history, for I don’t know any other community—the language barrier, as always, makes it hard to communicate with other communities. In the following, I will try to push further the study of the history of interactive fiction written in French, by talking in more detail about the 80s, a period that hasn’t really been

covered yet. This history of the 80s (as well as, briefly, the modern era) will also allow me to talk in a more in-depth fashion about various technical aspects of French interactive fiction and compare them to those of English interactive fiction (and, more precisely, Infocom games).

The Video Game Market in France in the 80s

Let's talk briefly about the video game market in France in the 80s—and of particular interest to us, the microcomputer market. First of all, we have to say a couple of things about France in the 80s: the country had about 55 million inhabitants, and the currency was the “franc français” (French franc, abbreviated FF). Due to inflation, it is quite hard to give an equivalent in euros (which was worth 6.55957 francs at the time it was introduced in 2002) or any other currency. The rule of thumb is that computers generally cost a few thousand francs, while games cost a few hundred francs.

First of all, it seems that the video game market was not really developed before 1980; a few microcomputers were available, but they were quite expensive. Then came the Sinclair ZX series of microcomputers: the ZX-80 came out in France in February 1980 and was the first microcomputer to be sold at less than 1000FF. The next year, the ZX-81 came out and was the most successful microcomputer at the time in France. It is hard to give exact sales figures for it, but a 1981 ad for this computer states that “tens of thousands of people in France already bought this computer,”⁵ offers a retail price of 790FF (only 500FF for the unmounted version), and states that games on tape are sold at a price “between 50FF and 150FF” (which was really cheap). The computer was sold from 1981 to 1987, at which date its production was stopped in favor of the ZX Spectrum (released in 1984 in France), which didn't sell as well as the ZX-81 (even though its retail price was seemingly as low as the ZX-81's).

The Commodore C64 was released the next year, in 1982, and was extremely successful. Despite of its price (4800FF) and its weight, it was a great success in France: about 1.5 million were sold throughout the 80s, not only for gaming purposes but also as a desktop computer.⁶ The

5 http://www.obsolete-tears.com/Sinclair/pub_zx81.jpg

6 <http://www.obsolete-tears.com/commodore-c64-machine-26.html>

C64 was very successful around the world, which means that a lot of games were available at the time (including classics such as *Arkanoid* and *Pong*). Commodore tried to improve the console with the release of the Commodore Plus/4, which was cheaper (1990FF) but which wasn't as successful as the C64 (partly because it wasn't compatible with C64 applications). In 1985, as machines with better capabilities were sold on the market, the price of the C64 dropped significantly to make it a more affordable machine. Games for the C64 were sold until 1994.

The following years saw the start of a boom in microcomputer sales in France, around the end of 1983 and the beginning of 1984.⁷ Several very successful microcomputers were released during this period, as well as a lot of games: this was truly the start of video gaming in France.

The first microcomputer to initiate this boom is the Oric-1. Oric was a British company (later bought by Eurêka, a French company, in 1985) that sold its microcomputers mainly in Europe; even though the computers had some bugs and issues (for instance with their HyperBasic language), they were affordable and extremely successful in the UK and France. The Oric-1 had 48kb of memory and a processor running at 1MHz. At first it cost 2000FF but was later sold at 1000FF, and it could be connected to the television, making it attractive for every family. In France, in 1983 alone, 50,000 Oric-1 computers were sold;⁸ it was later chosen as "Computer of the Year 1983."⁹ The following year, the Oric Atmos was released (first at 2490FF, then the next year at 990FF¹⁰) and was equally successful, if not more: 27,000 sold in the three months following its release in February 1984, and 120,000 in its first two years.¹¹ The success of those two Oric computers led to the development of a good number of French games, particularly interactive fiction games. The first Oric conceived by Eurêka, the Telestrat, came out in 1986 but sold badly (around 2,000 units).

The next computer that was highly successful in France was the Amstrad CPC. The Amstrad CPC464 came out in September 1984 and was an instant success; the business model was to build a computer that would be cheap, ready-to-use, and sold in supermarkets to attract

7 *Tilt* magazine, July 1984 issue, page 16.

8 We have to keep in mind here that this was before the boom of microcomputers, of course.

9 http://fr.wikipedia.org/wiki/Oric_1

10 <http://www.obsolete-tears.com/oric-atmos-machine-23.html>

11 http://fr.wikipedia.org/wiki/Oric_Atmos

families. It came out at the price of 2990FF (4990FF with a color screen¹²) and sold extremely well: 2 million units were sold in France in the 80s!¹³ Following this huge success, a great number of magazines about CPC464 gaming were started: the boom of the video game industry was definitely there. Its successor, the CPC6128, came out in 1985 and sold very well too. Those two computers reigned over the video game market in France for years, before the Japanese consoles took over at the end of the 80s, with the Nintendo NES and the Sega Master System.

Apple computers were starting to be successful too. The first Apple microcomputer that was sold in France was the Apple // Europlus (which is basically the same as the Apple //+ but for the European market) in 1980; it did not sell very well, because of some conception mistakes as well as a very high price (12000FF with the disk drive,¹⁴ which was expensive even for a color-displaying computer), but this computer, as we will see, had a very important role in the creation of interactive fiction. The next version of the Apple //, the Apple //e, sold pretty well, but it is the Apple //c that was the most successful: released in 1984, it had a mouse and a color screen and sold really well, though not as well as the previously mentioned computers.

Finally, the Atari ST microcomputers were pretty successful too. The first of them was released in 1985, and a couple of others were released a bit later; the sales of this microcomputer were 6 million worldwide. This microcomputer was attractive for its capabilities as well as its relatively low price (3000FF, a third the price of other comparable microcomputers with color capabilities at the time). Most of the success of the Atari ST happened in Europe rather than in the U.S.; in Germany, 2 million of units were sold, and in France it was 600,000 units.¹⁵ As a matter of fact, in France, a few magazines were exclusively dedicated to this computer, and we will see that there were quite a few interactive fiction games released for the Atari ST.¹⁶

>

12 <http://www.obsolete-tears.com/amstrad-cpc-464-machine-5.html>

13 Marion Vannier, head of Amstrad France, in *Les Chroniques de Player One*, p.38 (Éd. Pika, 2010)

14 <http://www.obsolete-tears.com/apple-europlus-machine-37.html>

15 <http://www.albatos.free.fr/ordinateurs.php>

16 It was even the machine used by some members of the community in their youth to play interactive fiction or adventure games.

To sum up, the Amstrad CPC, Oric, and Atari ST computers were the most successful of this period and initiated a boom in the gaming industry in France starting in 1984. Other microcomputers that did well were the ZX, Commodore, and Apple microcomputers. We will see that most of the interactive fiction games developed in the 80s were either for the Atari ST, CPC, Oric, Apple, or even Commodore computers—it seems that the ZX was already too old when the first interactive fiction games were developed.

How about the sales of video games in that period? The July 1984 issue of the magazine *Tilt* featured an article titled “La puce aux œufs d’or” (“The chip that lays golden eggs”) about the rising market of video gaming in France, that it could bring wealth and fame to any good game programmer, as was happening in the U.S. at the time. The article states that since “the number of computers in France is thirty times less than the number of computers in the U.S.”¹⁷ then “while a game can sell between 100,000 and 1 million units there, a French game can only hope for figures 50 to 100 times less.”¹⁸ Later in the article, it is stated that “3,000 units is considered as a good figure for a game.”¹⁹ The creators of a game could ask at the time for royalties between 10% and 25%—that seems high, but considering the sales figures, this doesn’t make the game developers very rich. Still, this article talks about a huge ambition from French publishers to make the most of the boom that was starting at the time; it features comments from Laurent Weill, one of the creators of Loricels, which we will talk about in a bit.

>

Now that we have laid out the landscape for video gaming in France in the 80s, let’s focus on interactive fiction *per se*. If we consider the history of interactive fiction as it is in general written, interactive fiction was spawned by the mainframe games *Adventure* and *Zork*, and a bit later by the Infocom games for microcomputers. Does this version of the story still hold for the development of interactive fiction in France?

Where are *Adventure* and Infocom?: English

¹⁷ *Tilt*, July 1984 issue, page 16.

¹⁸ *Tilt*, July 1984 issue, page 82.

¹⁹ *Tilt*, July 1984 issue, page 82.

Interactive Fiction in the 80s

It is widely acknowledged that the first interactive fiction ever was *Adventure*, written in 1975 by Will Crowther in Fortran on a PDP-10 mainframe. This game was widely spread on the American network ARPANET and was a huge success. Following this, a few other games were developed—on mainframes as well; there were clones of *Adventure* but also more and more original games, quite often in the cave-crawling genre as well: *Zork*, developed inside MIT in 1979, as well as Phoenix adventures across the pond in Cambridge, U.K., and a game in Swedish, *Stuga*, released around 1978. The creation of adventures in Cambridge wasn't spontaneous; in fact, it was one of the very few places in the world to have a connection to ARPANET, which means a copy of *Adventure*, and later *Zork*, transited there, spawning interest in interactive fiction.²⁰ *Adventure* was thus incredibly influential, creating a new genre of video game and generating a great interest for the genre—an interest so great that it prompted some players of the games to create their own *Adventure*-like games.

Is the story the same in France? We don't know for sure, but the answer is most probably no. It is very difficult to find information about mainframes in France at the end of the 70s and the beginning of the 80s, and unlike the English-speaking community, no ex-mainframe user is part of the French community to provide useful information about them. However, a few things makes us think that *Adventure* could very well have never reached France. First of all, no text adventure in French playable on a mainframe has been found or mentioned anywhere (and if *Adventure* was so interesting and even a bit widespread in France, it is likely that someone would have attempted to create an *Adventure*-like game). Also, it is unlikely that at some point a computer in France ever had a connection to ARPANET: first of all, ARPANET was an American network, and (for all I know) the only international connections of ARPANET were to Norway and the United Kingdom; also, after a French delegation was sent to BBN in 1970, France became increasingly interested in computer networks, and from 1972 was

20 As for *Stuga*, the authors had played *Adventure* and wanted to create a Swedish version: groups.google.com/group/rec.games.int-fiction/msg/70df36d635f1ad19. However, I don't know how the authors managed to play *Adventure* in the first place—maybe the game first transited through the Norwegian connection to ARPANET and then somehow ended up on a mainframe in Stockholm?

involved in the creation of its own network, CYCLADES. Another network, Transpac, was also developed by the French Ministry of Communications, and in the end CYCLADES was dropped in 1978 in favor of Transpac (which was used a few years later for the French network Minitel, which became very commonly used in the 1980s). Thus, it is unlikely that France ever requested a connection to the ARPANET, since it was involved in the creation of its own network; thus it would seem that we can safely assume that neither *Adventure* nor *Zork* ever crossed the English Channel. But even if those games were not known in France, one could imagine that someone would have had the idea of such a game and independently created the French equivalent of *Adventure*; however, as far as I know, this is not the case.

>

How about Infocom games—and, more generally, English interactive fiction games? This is a bit difficult to answer, as well: information about it is quite sparse. However, we can say with quasi-certainty that Infocom games were not (or very little) influential; the average gamer of the 80s will most likely quote French games rather than Infocom games. There are several points to consider here: were they influential for developers and game designers, were they well-received by critics, and were they successful in terms of sales figures?

The thing is, the information about the release of Infocom (or English) games in France is very sparse and is certainly a field that would be worth exploring. I thought for a while—maybe as a preconceived opinion, but more likely because nostalgic gamers weren't talking about Infocom games when they were talking about games they played in the 80s, or didn't know those games—that Infocom games simply had never been released in France, or if they had, it was as “import games” that were thus sold at a pretty expensive price,²¹ explaining a small audience. It turns out that this is incorrect: Infocom games, along with some other English or American interactive fiction games, were indeed sold in France, sometimes at an affordable price, and were even reviewed in video game magazines. It is worth noting that those games weren't

21 Ciel Bleu, the publisher of the first French interactive fiction game, was previously importing software from Canada to sell at a pretty expensive price in France; Jean-Louis Le Breton, founder of Froggy Software, remembers that his goal was to create French games at an affordable price (200FF or 250FF) compared to the games in English that were then sold at a very high price (350FF or 400FF).

translated and were sold in English, which could have contributed to their lack of influence or success. However, according to a nostalgic CPC gamer, English games became less popular with the boom of the French video game industry around the end of 1984.

>

Let's start with Infocom games: by reading old issues of video game magazines of the 80s, I could gather²² quite a lot of information about the release of Infocom games. The earliest reference to any Infocom game is in the December 1983 issue of *Micro 7*; the test of *Le Manoir du Docteur Génius*²³ mentions that “of course, the parser doesn't have the capabilities of Infocom games”;²⁴ this would mean that some Infocom games were available in France around this time.

The most significant trace of Infocom games that can be found is in the April 1984 issue of *SVM: Infidel* is the “Game of the Month”! The magazine reviews it on two pages:²⁵ it talks about the story of the game and the feelies, praises the parser that can recognize a lot of words and sentences, and mentions the presence of verbose and superbrief modes. Several things are worth noting in this article that give clues about Infocom games in France at the time. First of all, Infocom is introduced as “the creators of the famous *Zork*,” which (associated with the previous reference) makes us think that *Zork* had been published in France at the time.²⁶ Then, there is information about the release of the game: the publisher of the game in France is SIDEG,²⁷ the game was first released for Apple //e (the article says that versions for IBM-PC and Commodore 64 “should be released soon”), and it cost 695FF.²⁸

22 With the invaluable help of Grégoire Schneller, who provided most of the references here.

23 More on this game a bit later in this article.

24 *Micro 7*, December 1983 issue, page 137.

25 *SVM*, April 1984 issue, pages 88–89.

26 A name-dropping of an American game that nobody can find in France is an interpretation that makes less sense.

27 A very mysterious company—I could not find any information about it anywhere and thus was unable to determine if they published any other Infocom games in France around this time.

28 Which is a very expensive retail price for a game (about \$120) and is probably the kind of things Le Breton was alluding to (cf reference above). One could at the time find games—and even English games, as we will see—on this platform for about 200FF.

However, even if some magazines occasionally wrote about Infocom games, sometimes even reviewing those games, they were not famous enough to be considered by French gamers as classics games to which you compare other adventure games. This can be correlated with the remark mentioned above: after the boom of microcomputers and video games in France in 1984, a lot of French games were released, and games in English became less common. There are a few references to Infocom games in magazines, though. For instance, in its January 1985 issue, *Tilt* mentions the “good detective game” *Witness*.²⁹ In April 1985, *SVM*, in an article about text adventures,³⁰ mentions Infocom and its “good stories” as well as its “incredible parser [that] has an answer for every sentence in natural language”;³¹ the article mentions the games available in France at the time:³² *Zork*, *Infidel*, *Deadline*, and *Sorcerer*, available on Apple //, IBM-PC, and Commodore 64.³³

The release of the Atari ST and its relative success apparently gave a new platform for Infocom to release its games on. The first issue (September 1985) of *ST Magazine* mentions the software available on this computer; the category “text adventure” (implied: without graphics) is almost entirely made of Infocom games: *Zork* (all three episodes), *Wishbringer*, and *The Hitchhiker’s Guide to Galaxy* are listed as available, while every other Infocom game³⁴ is listed as “available in October.”³⁵ Another reference in the December 1985 issue of *ST Magazine* mentions the exact same thing: “Every Infocom game is available for the ST. Very high in the U.S. sales charts for years, they require a good knowledge of English to be played.”³⁶ A few years later, in December 1987, the magazine *Atari*

29 *Tilt*, January 1985 issue, page 76.

30 Both text-only and with graphics, but all with a parser, though the article focuses more on the text adventures with graphics since the genre was the most common in France at the time.

31 *SVM*, April 1985 issue, page 115.

32 Though maybe not all the games available, since *Witness* is not in this list even though *Tilt* mentions it four months earlier.

33 Those three computers thus seem the platforms for which Infocom games were released at the time.

34 Every other Infocom game released before September 1985—thus, every Infocom game released before 1986 except *Spellbreaker*.

35 *ST Magazine*, September 1985 issue, page 11.

36 Let’s just note another (hilarious) list made by the same magazine a few years later, in its August 1987 edition (page 55), where the listed Infocom games include *Brimstone*, *Essex*, “*Mind Forever*,” *Mindwheel*, *Sorcery*, and *Skul/West*.

1st lists again some Infocom games³⁷ that are available on the Atari; we note that in this list, a few games are missing, for instance *Hollywood Hijinx*, *Trinity*, and *Bureaucracy*³⁸: it is likely that those games were not released in France, for an unknown reason.³⁹ Unfortunately, no information about the retail price was included in those references. In other magazines, there were occasionally reviews of Infocom games; however, we have to note that the majority of those reviews were written after 1987, and as we will see later, adventure games with parsers were declining at that time.

In a nutshell, that is all we could gather in old magazines about Infocom games. It appears that the games were first released on Apple //, IBM-PC, and Commodore 64; then the games were released in a more systematic way for the Atari ST. It seems that Infocom games didn't enjoy a great success in France;⁴⁰ we could cite as reasons the high retail price (for the first years, at least—Infocom games for the Atari weren't as expensive, probably around 200FF) and the suboptimal choice of platforms,⁴¹ but the main reason is certainly that those games were text-only and in English.

>

A word about other games in English. It seems that other games in English were released in France in the 80s, and some of them were more

37 The games listed are: *Ballyhoo*, *Cutthroats*, *Deadline*, *Enchanter*, *Hitch Hikers Guide to the galaxie* [sic], *Infidel*, *Lurking Horror*, *Moonmist*, *Planet Fall* [sic], *Seastalker*, *Wishbringer*, *Witness*, and *Zork I,II,III*.

38 This game was indeed never released in France; a review of the game in the September 1987 issue of *ST Magazine* (page 54) states: “We found this game. Not in France, but in Belgium.” and “For those who want to play this game, they will have to solve a tough puzzle first: finding the game!”

39 Was it a lack of success of Infocom games, or the beginning of the fall of Infocom, that maybe prompted them to choose their releases carefully in an attempt to avoid losing money?

40 In the September 1987 issue of *ST Magazine* (page 54), the journalist writes, “Every single publisher that tried to sell Infocom games in France ended up with stocks that no one would buy.”

41 Infocom games were apparently never released in France for the Amstrad CPC or for the Oric, which were the two more popular microcomputers at the time; the reason might be that those computers weren't that popular in the U.S., and it would have required more work to adapt the Z-Machine on those computers first to get the European market; in any case, it was perhaps a mistake by Infocom, but it seems that it wouldn't have changed anything either.

successful than Infocom games.⁴²

It appears that some Scott Adams games were released in France and were pretty successful. The versions that were released were the games with graphics published by Adventure International. We can find a quick review of *Saga 1: Adventureland* in the April 1984 issue of *Micro 7*, stating “Scott Adams games are reference games,” and “the next three are now available”; the game was released for ZX Spectrum and Commodore 64, and the retail price was somewhere around 200FF (which is pretty cheap).⁴³ The January 1985 issue of *Tilt* (which featured a lot of tests of adventure games) mentions *The Hulk* (for Commodore 64 and Atari 800; an ad on the next page advertised a retail price of 185FF⁴⁴) and *Voodoo Castle* (for Apple //).⁴⁵ The last reference we could locate is in *ST Magazine* in December 1987, where the games *Questprobe 2: Spiderman* and *Questprobe 3: The Fantastic Four part I* were listed.⁴⁶

A couple of games in English got very good reviews in French magazines and were seemingly quite successful. First, *The Hobbit*, for Spectrum 48K and Commodore 64 (and later, interestingly, for Oric Atmos), had a very good review in the March 1984 issue of *Micro 7*,⁴⁷ as well as in *Tilt* in January 1985.⁴⁸ Then, *The Pawn*, which got very good reviews in Atari ST magazines^{49,50}; as a matter of fact, Magnetic Scrolls games enjoyed quite a good critical success, winning two *Tilts d’Or*⁵¹ in the 80s.

Mystery House was also available in France before 1982 for Apple // and may be the first text adventure in English released in France—we know it was released in France before 1982 for Apple // because it inspired the first French interactive fiction game, as we will see in the next part. We can also quote *Masquerade* on Apple // (a couple of French adventures for Apple // were compared to it by critics), games by Legend Entertainment such as *Frederik Pohl’s Gateway* and *Eric the*

42 Unfortunately, the research about those games is less comprehensive than for Infocom games, leaving room for a more precise and complete analysis.

43 *Micro 7*, April 1984 issue, page 118.

44 *Tilt*, January 1985 issue, pages 71 and 73.

45 *Tilt*, January 1985 issue, page 76.

46 *ST Magazine*, December 1987 issue, page 55.

47 *Micro 7*, March 1984 issue, page 107.

48 *Tilt*, January 1985, page 70.

49 *ST Magazine*, December 1985 issue, page 31.

50 *ST Magazine*, August 1987 issue, page 55.

51 *Tilt d’Or* for Best Graphics in 1986 for *The Pawn*, and *Tilt d’Or* for Best Adventure Game in English in 1988 for *Corruption*.

Unready, as well as games by Interceptor Micro such as *Forest at World's End* and *Jewels of Babylon*⁵²—the latter was actually translated into French a bit later, which is rare enough to be noticed.⁵³

>

So why exactly were some extremely successful and influential games in English not successful at all in France in the 80s? The answer is very simple, and it is as always the language barrier. Those influential and originally English-speaking games were never translated into French, and since France is not traditionally a country where English is widespread,⁵⁴ games in English had a more limited audience and thus were not very successful or influential. We note that games in English were imported mostly before 1984: English games were played because there were not a lot of games in French at the time, but as soon as there were more games created by French video games companies (after the boom of the video game industry in France in 1984), it seems that not a lot of people played games in English anymore.

But can we say that, had Infocom games been translated into French, they would have been more successful? It is very hard to say. We could imagine that with such a translation, and considering the quality of Infocom games, more people would have played them in the early years,⁵⁵ leading potentially to a cementation of the genre that could compete with text adventures with graphics. But as a matter of fact, French gamers at the time seemed very attached to adventures with graphics (to put it mildly) and very patronizing toward text-only adventures. For instance, in the January 1985 issue of *Tilt*, a short paragraph titled “One dimension less” talks about some text-only adventures and notes that adventures with graphics are better because they spare the player the

52 The reviewer in *Hebdomagiciel* couldn't get past the first screen, even after trying every verb he could think of, deeming the game “unplayable unless you have a Master's degree in English studies.”

53 Other examples are a translation of *Eureka*, by Ian Livingstone, and four translations of text adventure games by the French company Les Aventures in 1985 for Spectrum.

54 As compared to other countries, such as the Netherlands or in Scandinavia, where learning to speak English is more emphasized (by for instance having TV programs with subtitles instead of dubbing); the traditional cliché is that the French are bad at speaking English, which is probably not too far from reality.

55 Leading potentially to an earlier boom, because there would then have been excellent games on microcomputers?

trouble of imagining things, and the memory excuse doesn't hold for the most recent computers.⁵⁶ The April 1985 edition of *SVM* is harsher, writing in an article about adventure games,⁵⁷ "We have to mention the sinister incarnation, fortunately on its way towards extinction, of games that ignore graphics. For instance the Infocom series [sic] (such as *Infidel*), who think they are even by compensating with a (very good) parser and a broad vocabulary. *C'est un peu court, jeune homme!*"⁵⁸ In Generation 4's first issue at the end of 1987, we can read, "The scenarios and the quality of those adventures are why they are among best-sellers in the U.S. In France, it is very different, since we seem to consider that a game in English without any graphics is not a proper game." But were French gamers attached to graphics just because they had the habit of them, or is it deeper than that?⁵⁹ An interesting question, for sure.

And in fact, we may be able to argue that Infocom was in fact quite influential, in a way. Because Infocom games were text-only, you had to be able to read English very well to be able to enjoy the game; whereas with an adventure with graphics, you understood the setting with much less effort and could still advance in the adventure with only rudiments of English (the classic verbs in adventure games, for instance) or a dictionary next to you. Thus, French people were more able to understand graphic adventures; before French adventures were developed, all that a French gamer could play, and thus would play, were graphic adventures. It is then quite logical that the first adventure games in French that were developed featured graphics, since the genre was pretty much the reference for a majority of gamers (and perhaps authors as well); the trend carried on throughout the 80s, and soon enough French gamers couldn't see why they would play adventure games without graphics, seeing text-only adventures as lacking something in comparison. Thus, this (unverified) theory would explain the market of adventure games in France in the 80s: the fact that Infocom games were never translated to French can very well have been somewhat influential in the 80s by bringing gamers to graphic adventures, shaping the landscape of French adventure games for the whole decade.

56 *Tilt*, January 1985 issue, page 76.

57 *SVM*, April 1985 issue, page 115.

58 Quote of a famous tirade of Cyrano in the play *Cyrano de Bergerac*.

59 After all, the boom of microcomputers happened in 1984, with computers with very good graphics capabilities; could it be that the French were just waiting for graphics to buy computers?

>

Now that we have talked about games in English in France, and have seen how even the most influential games (that are part of the canonical, English-biased history of interactive fiction) were unsuccessful in France in the 80s, we can focus on text adventures in French during this period. It all began in the summer of 1982, with a summer holiday in the south of France.⁶⁰

The Birth of Interactive Fiction: Froggy Software

In 1982, Jean-Louis Le Breton was 30 and was living in Paris; he had a band, called “Los Gonococcus”, with Yves Frémion⁶¹ and Jean Bonnefoy. The band split up, and he sold his keyboards in a Parisian store; in the store next to it, someone was selling back his Apple //+, so Le Breton bought it because, in his own words, “it was the first good microcomputer in France and I wanted to know how all that worked.”⁶² Along with the computer, he only bought one game: *Mystery House*, by Ken and Roberta Williams.

60 Actually, shortly after the publication of this book, Eriorg (as always!) found some information that would challenge this version of the history of French IF! What he found is that in the very first issue of *Tilt* (September-October 1982), the game *La Caverne des Lutins* was reviewed; this game was classified as “RPG” but was actually a dungeon-crawling text-only game in French! Interestingly, the review mentions some of the parser’s limitations but considers them as interesting concepts and charming novelties rather than spoling the fun. The price of the game is not mentioned, but it is mentioned that the game is available on the Victor Lambda microcomputer, a microcomputer that came out in the early 80s and that wasn’t very successful. We don’t know anything else about this game, but it certainly challenges the view of *Le Vampire Fou* as the first interactive fiction in French. What we do know is that Le Breton created *Le Vampire Fou* just before the release of this game and didn’t own a Victor Lambda microcomputer, so both games were developed independently and have no influence on each other whatsoever. (Furthermore, Le Breton went on to create the first publisher of interactive fiction in French, thus possibly cementing him as the creator of the genre in France; we should note, however, that we don’t know anything about the author of *La Caverne des Lutins*.)

61 Who ended up being a successful cartoonist in several magazines such as *Fluide Glacial*.

62 http://www.jeanlouislebreton.com/fiches/01.php?id_news=45&SECTION=17

As summer was approaching, he went on holiday in the Gers,⁶³ with the Apple // in the trunk of his car. He played *Mystery House* and liked it—he states that he didn’t like video games before: the fact that you could move your character around was interesting, but there were too many fights for his liking.⁶⁴ However, he thought that judging by the quality of the graphics and the scenario, he could easily do as well in a French game.⁶⁵ He thus learned BASIC in a month and programmed what would be the first text adventure game in French: *Le Vampire Fou* (*Mad Vampire*). It was a pretty simple adventure, where the goal was to enter the Vampire’s castle to kill him before he killed you. Le Breton rewrote the game shortly after he completed it and published it in 1983. But its publisher, Ciel Bleu (whose previous activities were mainly importing software from Canada to France), went bankrupt shortly after the release of the game. Moreover, this happened before the success of Apple microcomputers in France (the Apple //e and Apple //c); even if it drew the attention of some magazines as the first interactive fiction game in French,⁶⁶ it is unlikely that the game had a much greater success than other games at the time, and thus the release of the game wasn’t exactly a “defining moment.”

This story of the first French interactive fiction game highlights some very interesting things. First of all, the date of conception is the summer of 1982, and the date of publication is 1983; at this time, Infocom was already a few years old, had already published the famous and influential *Zork* trilogy (as well as *Deadline* and *Starcross*) and published no less than five games in 1983, while Scott Adams had already created quite a number of games for microcomputers: interactive fiction was already a booming genre in the U.S. when a handful of French gamers discovered *Le Vampire Fou* (which wasn’t a long or complex or literary game). Then, we note that the influences of Le Breton for this first game is *Mystery House*: this isn’t exactly an interactive fiction game but more an adventure with graphics and a parser. *Le*

63 Rural and sunny area in the south of France, not too far away from Toulouse.

64 *Wired.com*, excerpts from *Replay: The History of Video Games* by Tristan Donovan, <http://www.wired.com/gamelife/2010/06/french-touch-games/>

65 *Idem*.

66 A hilarious photo of Le Breton can be found in the December 1983 issue of *Micro 7* on page 31; but beware of the clichés about French people. (http://download.abandonware.org/magazines/Micro%207/micro7_numero11/Micro%20N11%20%28Decembre%201983%29%20-%20Page%20031.jpg)

Vampire Fou is not exactly an interactive fiction game, but it is the first adventure game with a parser that was published in France. This point is of great interest, and we will return to it a bit later.

Le Breton then met Fabrice Gille, the son of a friend of his, who was 18 and had gotten his Apple // a short time before. Le Breton gave him a copy of his game, which was supposed to be copy-protected. Gille cracked it in no time, which impressed Le Breton and prompted him to want to work with Gille.⁶⁷ Both then founded Froggy Software to publish Le Breton's next game, *Paranoïak*; Gille programmed both the software and the copy-protection. *Paranoïak* was Froggy Software's first game and won the Golden Apple 1984.⁶⁸

Froggy Software went on to publish a dozen games on the Apple // before closing in 1987; the main reason for this was that the games were becoming obsolete because of the rise of the Macintosh. Their games, mostly written or coded by Le Breton, had a particular flavor: they were not serious games at all ("adventure, humour, leftfield and a willingness to make fun of anything"⁶⁹), and they dealt with very different themes than the usual fantasy/sci-fi production of English games: the themes tackled were often political, for example. The games encountered a good critical reception in magazines, prompting the games magazine *Tilt* to dub Le Breton as "the Alfred Hitchcock of gaming."⁷⁰

Paranoïak was the first success of the company; in the game, the player has to battle against mental illnesses, all with a humorous tone. Then came *Le Crime du Parking*, published in 1984 as well, which had an even greater success; the player has to solve the murder of Odile Conchoux, found strangled in a parking lot, and the game deals (much more seriously, but with silver linings of humor) with themes such as rape, homosexuality, and drug addiction.⁷¹ *Même les Pommes de Terre ont des Yeux!* (*Even the Potatoes Have Eyes!*), published in 1985, was also a big success: it was set in a South American dictatorship where spies were everywhere (hence the reference to potatoes potentially watching you),

67 http://www.jeanlouislebreton.com/fiches/01.php?id_news=45&SECTION=17

68 A prestigious award given by Apple every year for games on the Apple // computers.

69 *Wired.com*, excerpts from *Replay: The History of Video Games* by Tristan Donovan, <http://www.wired.com/gamelif/2010/06/french-touch-games>.

70 *Idem*.

71 Apparently, because of that, none of the American editors that Froggy Software contacted to try to get this game published in the U.S. was really keen on doing so.

and the tone is very humorous.⁷²

So what were those games like anyway? First of all their parser was quite primitive—it was just a two-word parser, but it could recognize quite a number of words; moreover, the quality of the parser didn't fluctuate from one game to another, which is a less trivial concern than it may appear at first: Infocom (and others) had the good idea of building the parser in an interpreter that could be used for all of their games, but for other French game companies the quality of the parser would often fluctuate. Second, Froggy Software was the first company to include funny answers to some inputs; the tone of the games was very humorous for sure, but they were the first to include funny default responses in their games, as well as a recognition of curses and insults. Third, all of their games included graphics; actually, most of the screen (about the top three-quarters of the screen) consisted of a picture of the room and the objects. As a result, the descriptions were really sparse, and the graphics were necessary to advance in the games (though you could turn them off at any time). The only exception to this is the game *La femme qui ne supportait pas les ordinateurs* (*The Woman Who Couldn't Stand Computers*), written by Chine Lanzmann and coded by Jean-Louis Le Breton, published in 1986; this story of a flirtatious, then murderous computer and your adventures on the Calvados network (a French network of Apple // computers that people—among them the creators of the game—used as chatrooms) had seven different endings that you had to complete to win the game. The game was text-only, and the parser was quite primitive since it only consisted in Yes/No answers.⁷³

>

So as a matter of fact, every Froggy Software game featured graphics and two-line descriptions.⁷⁴ But in fact, as we mentioned briefly in the previous part, most interactive fiction games in French that were released in the 80s featured graphics as a prominent part of the game. As a matter of fact, we can say that *interactive fiction didn't exist in France in the 80s*. Interactive fiction as we know it—one can say, Infocom-style IF, or

72 A port of this game in Inform (without graphics) has been done by the author of this article and can be found here: <http://ifiction.free.fr/index.php?id=jeu&j=013>.

73 An Inform port of this game was done by the author of the article and is available on the IFDB, as well as the original ROM: see <http://ifdb.tads.org/viewgame?id=brxdd0j3xu8mmgmc>.

74 Very much in the style of *Masquerade*, for example.

“literary” IF—didn’t exist: only a handful of games were text-only, and almost none of the rest were aiming at any kind of literary quality whatsoever. The term “interactive fiction” was never used in the 80s by any game company or any reviewer (at least, none that I know of⁷⁵): people were talking about “adventure games,” “role-playing games,” and then when point-and-click games started, “text adventure games.” The descriptions were sparse at best, and the graphics took up most of the screen (we will detail this a bit more later). And, even if a handful of English-speaking interactive fiction games had been translated into French at the time (such as some from Interceptor Micro), no Infocom game was ever translated into French; game creators in France thus didn’t really have a model of literary text-only interactive fiction to be inspired by.

In what follows, we will still continue to refer as those games as “interactive fiction games” but as well as “text adventure games” or simply “adventure games”; the reason is that they are what was the closest to interactive fiction that was available then in France, they had a parser and a keyboard-based input, and some of them didn’t have any graphics.

Other Successful Publishers: Loriciels, ERE Informatique

Froggy Software wasn’t the only successful company that produced text adventure games at the time; around 1984, a few other French game companies achieved a very similar (and in some cases, greater) success by publishing adventures written in French. There are quite a lot of them, for example Titus, Lankhor, CobraSoft, Excalibur, and even Infogrames; in the following part we will talk mainly about two companies: Loriciels and ERE Informatique.

>

Loriciels was a company created in 1983 by Marc Bayle and Laurant Weill; the name is a pun between “logiciel” (software) and “Oric”

⁷⁵ Note that *La Femme Qui Ne Supportait Pas Les Ordinateurs* was decried by Froggy Software as an “interactive novel,” a term that wasn’t re-used by anyone else, though.

because they first focused on publishing games for the Oric microcomputers. They didn't just stay on the Oric, and they expanded their area of publishing to the ZX Spectrum (often for adaptations of their Oric hits), then the Amstrad CPC, and later the Atari ST and the Amiga. The company first achieved great success with a non-textual adventure game, *L'Aigle d'Or*,⁷⁶ in 1985, for which they won the first-ever "Tilt d'Or";⁷⁷ they won a second one for the text adventure with graphics *Le Mystère de Kikekankoi* (*The Whomatwen Mystery*). They were among the biggest game publishers in France in the 1980s, publishing about 150 games in 10 years;⁷⁸ but a stream of financial problems forced the company to shut down in 1993.

The first text adventure game Loricels published was *Le Manoir du Docteur Génius* (*Dr Génius's Manor*). The game wasn't exactly a success; it was a short adventure where you had to escape a manor filled with traps, with bits of humor but described as lacking in atmosphere and surprises.⁷⁹ The graphics were just lines drawn *à la Mystery House*; the parser recognized quite a lot of actions but was limited to the first four letters. A sequel of this game, *Le retour du Docteur Génius* (*The Return of Dr Génius*) was released in 1985 for Oric as well.

One of their biggest successes in the genre of text adventure is *Le Mystère de Kikekankoi*, released in 1983 for Oric, and then in 1985 for CPC with greatly improved graphics. Once again, this is a text adventure with graphics very much in the spirit of *Mystery House* (for the original version). You find a message in a bottle from a woman imprisoned by a mad scientist; you must rescue her by exploring the city of Kikekankoi and the nearby cave. The game is timed—you have 500 turns to rescue her—and instant deaths are numerous; the parser recognizes about fifty verbs and a hundred nouns (a list of what is recognized by the parser is given at the beginning of the game). The game had a big success upon its release and was praised in *Tilt* as "still enjoyable even if a bit old" in

76 That can be translated as *The Golden Hawk*; the game was an adventure game where your on-screen character had to progress through a series of rooms in a castle to find an artifact; the game was praised for its atmosphere and its isometric-2D graphics.

77 That can be translated as "Golden Tilt"; it was the Game of the Year award given by the influential video game magazine *Tilt*.

78 Their strategy was to be very open to new games: anyone could send them their game, and they would publish it if they liked it; that's how they published *L'Aigle d'Or*.

79 *Tilt*, January 1985 issue, page 74.

1985;⁸⁰ it's probably one of the most famous French adventure games of the 1980s.

Le diamante de l'île maudite (*The Diamond in the Cursed Island*) was also a great success in its time; released in 1984 for Oric and 1985 for Amstrad CPC, you had to explore an island to discover a diamond in underground caves. The graphics show a notable improvement compared to those of *Le mystère de Kikekankoi*; the game screen shows a picture, below it the parser, and consistently asks the question “What do you do?” The game is a really long one as well: for the first time, it is likely that not one, but two developers worked on this game. The improvements of the parser system earned rave reviews from the press: the vocabulary was a good size, but what compelled the critics was that the words were completed from the third or fourth letter, and bad words were deleted with a beep; it was also possible to enter several commands at the same time using a slash. This game had great success and is also one of the most famous of the era.

Other quite successful text adventures released by Loriciels are *Citadelle* (a role-playing text-only adventure with a parser recognizing 260 words), *Tony Truand* (a game with a complex story and 120 locations), *Le pacte* (a horror game created by Eric Chahi, who went on to create *Another World*—known as *Out of This World* in the U.S.—several years later), *Han d'Islande* (an adaptation of a novel by Victor Hugo; featured graphics but also a noticeably longer prose than usual), and *Orphée* (with graphics, a sidebar indicating the characters in the area and the inventory of the player, and the parser above the graphics; the game was beautiful but very hard).

>

ERE Informatique was created in 1981 (which makes it one of the oldest French video game companies) by Philippe Ulrich and Emmanuel Viau. They released a variety of games in diverse genres, but their biggest hits were text adventure games; all of their games were released only for the Amstrad CPC. They were bought by Infogrames in 1986, allowing them to focus more on the game crafting aspect and less on the commercial aspects. However, financial and royalties problems with Infogrames led to the closure of the studio several years later, in 1989, with most of the designers leaving to found another video game company.

80 *Idem*.

One of the most famous games by ERE Informatique was the *SRAM* series (*SRAM* and *SRAM II*, both released in 1986). In those widely acclaimed games, the player is on a strange planet,⁸¹ and a huge political change occurs; a hermit and a witch call for your help to get Egres IV on the throne. In the second game, Egres IV has become a bloodthirsty sovereign, and you have to dethrone him. The tone of the game is humorous, though the humor in the game is more subtle than for other games at this time; the world is a sort of medieval world with anachronisms—there are fire extinguishers—and influences from other genres, which makes for an incoherent but funny setting. The graphics of the game were praised and were top-notch at the time of its release. The game itself required a lot of work from the three authors, Serge Hauduc, Ludovic Hauduc, and Jacques Hemonic: the first game underwent nine months of development (and the second required three months); still, both games are pretty short games. The game sold very well and was a massive hit for ERE Informatique.

Another famous game was *Le passager du temps* (*The Time Passenger*), released in 1986. This game was another great success for ERE Informatique: using a simple but well-crafted story (your uncle disappeared, and you are traveling through time as you're looking for him), the game manages to stay long and difficult; the graphics were praised, but the most beloved feature of this game is the cat that appears in the sidebar, commenting on the action in a humorous tone.

Perhaps the most acclaimed game by ERE Informatique is *L'arche du Capitaine Blood* (*Captain Blood*), released in 1987; it sold well in France, as well as in many other countries.⁸² While it's not interactive fiction *per se*, it's still worth mentioning for its conversation system: the game has 120 icons you can combine to form sentences to communicate with the aliens you encounter in the game; this system required a lot of work by Philippe Ulrich, who reportedly wrote tens of pages of dialogue with the aliens. This novel conversation system (that wasn't really emulated later) was praised, as well as the graphics and the sound; the story is a fairly complex sci-fi story of clones and aliens. We note that this is one of the first cases of a successful point-and-click game, paving the way for the golden age of the genre a few years later, and taking the adventure game further from interactive fiction.

Other interactive fiction titles published by ERE Informatique are

81 The title of the game read backwards surely gives a clue about the planet.

82 It charted first in France, Germany, Italy, and the U.K. and sixth in the U.S.

Oxphar (an adaptation of a play, set in a medieval-fantasy world; the reviews praised the graphics as well as the wit and poetry displayed by the game; the parser was a simple two-word parser, with the feature that the game could learn new synonyms for verbs), *Harry & Harry*, *Crash Garrett* (a humorous and almost parodic story about undercover Nazis in the U.S. in the 1930s), and *1001 BC*.

>

As we saw with the example of these two publishers—surely among the most successful publishers of adventure games in France in the 80s—interactive fiction wasn't really common; the games were more text adventures with graphics. However, they share common paradigms with “literary,” text-only interactive fiction; the games underwent a period of evolution throughout the 80s, and different systems and game design concepts were tried. In the following part, we're going to attempt a transversal survey of IF in French in the 80s: rather than enumerating games chronologically, we're going to enumerate topics in interactive fiction design theory and see how they were addressed in various games of this period.

The “French Touch”: Interactive Fiction in France in the 80s

The production of adventure games in French in the 80s was very diverse, as well as numerous: hundreds of games were released, with different themes, different interfaces, different tones, and the genre was extremely popular at the time. Enumerating all the games published during the period would be tedious, and to be fair quite useless; instead, we are going to attempt a review of the genre throughout the 80s in a transversal way, looking at some characteristics of adventure games rather than the games individually. This methodology will allow us to see better the evolution of the genre, as well as its specifics.

>

The first point we are going to discuss is the form of the games. As we said before, most adventure games with parsers in France featured graphics, and that is what the gamers and reviewers were really used to

and liked. Text-only adventure games were not really successful, and as a matter of fact a bit patronized by reviewers, who saw text-only adventures as a thing from the past (before adventures with parsers as a whole became a thing from the past at the end of the decade). Is that to say that no text-only adventure game in French was ever released? As a matter of fact, almost. Browsing through countless adventure games with parsers released at the time, I was only able to find a handful of them that were text-only, which makes me say that 99.9% of French text adventures in the 80s featured graphics;⁸³ let's have a look at this list.

There is Froggy Software's *La femme qui ne supportait pas les ordinateurs* (*The Woman Who Couldn't Stand Computers*),⁸⁴ written by Chine Lanzmann and coded by Jean-Louis Le Breton, released in 1986. The story is as follows: your computer went crazy and tries to seduce you; adventures on the Calvados network⁸⁵ ensue, with seven different endings you must explore to win the game. The game is text-only, but it is still a very particular type of interactive fiction: the parser only recognizes Yes or No, and the story unfolds by choosing your answer to questions the characters sometimes ask you. However, the setting is quite clever and noteworthy. The game looks like an online chat with different characters, along with system messages (such as "Connecting network... Done"), and there is absolutely no description of an external or imaginary element. you could almost think that what happens in the game is in fact happening to you. The only thing that reminds you that it is a game are the title screen and the ending screen (the only two illustrations of the game).

Another text-only game was *Citadelle*, published in 1984 by Loriciels. This is seemingly the first text-only adventure⁸⁶ and was presumably not

83 Even of bad quality: one can find a remark in the January 1985 issue of *Tilt* (page 76) about how some (quite dishonest) developers drew beautiful graphics for the title menu and the first few rooms, while the rest of the graphics looked quite bad, the goal being to lure the player with a promise of quality graphics.

84 The ROM for this game is actually hard to find, but an Inform port of this game was done by the author of this article and is available here: <http://ifiction.free.fr/index.php?id=jeu&j=029>.

85 A French network for Apple // computers, using the Transpac communication network, that featured services such as chatrooms, email, message boards, and online news (via the Agence France Presse). The network was used by Jean-Louis Le Breton (under the alias "Pépé Louis"), Chine Lanzmann ("Chine"), and other online friends, who all make a cameo in the game.

86 Though the review of the game in *Tilt* describes it as "the first RPG entirely in French."

a huge success.⁸⁷ Still, the game was praised by *Tilt*,⁸⁸ noting that the game was long and offered quite a lot of challenges for the adventurous gamer; the only drawback they note was the absence of graphics, which made the combats boring (they could consist in a succession of “You miss. The gnoll misses. You miss. (etc.)”).

The other text-only games I could find are CobraSoft’s *Dossier G: l’Affaire du Rainbow Warrior*⁸⁹ (*File G: The Rainbow Warrior Scandal*), published in 1985, which seems more a CYOA than a regular text-adventure and doesn’t seem to include graphics; *Mission secrète à Colditz* released by Soracom in 1986.

This completes the list of text-only adventure games in French released in the 80s. Interestingly, it seems that *Citadelle* is the only text-only adventure game with parser released in France in the 80s, making it possibly the only interactive fiction game in French of the decade.

>

Since virtually almost all French text adventure games featured graphics, we will talk briefly about the different type of graphics in those adventure games.

The first text adventure game in French, as we said before, is *Le Vampire Fou* by Jean-Louis Le Breton. Le Breton had the idea of writing such a game after playing *Mystery House*, by Ken and Roberta Williams, on his newly acquired Apple //+; as a result, the graphics in *Le Vampire Fou* are simple lines drawn on the screen.⁹⁰ This style of graphics can actually be found in a couple of other games. For example, the early game *Le Manoir du Docteur Génius*, published in 1983 for Oric and the first game published by Loricels, has similar graphics;⁹¹ its sequel, *Le Retour du Docteur Génius*, was published in 1985 with similar graphics, though slightly better (some areas are colored on the screen, and the drawing is simply better done). This style of graphics, a consequence of limited graphics capabilities of the microcomputers on which they were

87 Numerous articles about Loricels state that *L’aigle d’or* was their first successful game.

88 *Tilt*, July 1984 issue, pages 54–55.

89 Inspired by the Rainbow Warrior scandal of the summer of 1985, in which French intelligence sank a Greenpeace ship while it was stationed in New Zealand but got caught. The whole affair was revealed to the public, creating a scandal in France.

90 As seen here: <http://grospixels.com/site/images/vampirefou/vampirefou02.gif>.

91 One can wonder if the graphics here are inspired by *Mystery House* or by *Le Vampire Fou*.

released,⁹² quickly disappeared with new computers with improved graphic capabilities. The only noteworthy attempt was the 1985 CPC game *Bad Max*;⁹³ the story is heavily inspired by *Mad Max*, the music of the game was composed by the Alan Parsons Project, and the game also features line-drawn graphics, with an interesting (and in retrospect, quite funny) twist: the game is in “Stereo-3D,” that is to say everything is drawn twice, once with red lines and once with blue lines a few pixels away, presumably creating a 3D effect when playing the game while wearing the 3D glasses that went along with the game.

Later, games had better graphics, but the design remained unchanged. This was the design chosen by Froggy Software:⁹⁴ the graphics are on the upper part of the screen, and the descriptions and parser are below. This design can be found in almost every Froggy Software game, as well as in games such as *Le diamant de l'île maudite* (Loriciels, 1985), *Atlantis* (CobraSoft, 1985), *Attentat* (Rainbow Productions, 1986), *Le Pacte* (Loriciels, 1986), *La cité perdue* (Excalibur, 1987), and many others.

Another, different interface was also very popular: the graphics were embedded in a frame at the center of the screen, and there was a sidebar to the right, with the parser either above or below the picture frame. While not totally revolutionary, this interface could be quite beautiful and allowed some improvements—the sidebar could be used to list the exits, the inventory, the objects or characters present in the room, or even, as in *Le passage du temps*, a cat that was commenting on the action. This kind of interface was used in games such as *Le mystère de Kikekankoi* (Loriciels, 1985, CPC version), *Orphée: Voyage aux Enfers* (Loriciels, 1985), *Oxphar* (ERE Informatique, 1987), *Le passager du temps* (ERE Informatique, 1987), *La Chose de Grotenburg* (Ubi Soft, 1988), *Excalibur Quest* (Excalibur, 1988) and *L'île oubliée* (Bruno Fonters, 1993).

>

As always in this type of game, the quality of the parser is always vital; there were numerous systems and improvements that were designed over the years.

92 The Apple //+ and Oric-I could only display a handful of colors, and the display was quite imprecise as well. Thus, every adventure game that was published for the Oric-I had similar graphics, such as *Le Mystère de Kikekankoi*. The next generations of those computers, the Apple //c and Oric Atmos, could display more colors.

93 http://cpcrulez.fr/GamesTest/view.php?game=bad_max

94 Such an interface design is comparable to the one in *Masquerade*.

As for *Le Vampire Fou*, the parser was extremely primitive: it seems like a two-word parser (that didn't recognize a lot of words), but in fact opening the ROM file with a simple text editor shows that the actions that you had to do to progress in the story were hard-coded! This was obviously not a very good parser, and it could only get better.

The vast majority of the parsers of this era were simple two-word parsers, that quite often only recognized the first few letters of a word. As a matter of fact, the reviews of some games (such as *Conspiration de l'an III* by Ubisoft in 1988) state that those games were correcting typos, but it is possible that the game in fact only recognized the first few letters of each word. The fact that only the first few letters were recognized was cleverly hidden by *Le diamant de l'île maudite* (Loriciels, 1985): the game had an auto-completion feature that recognized the word after the third or fourth letter was input by the player and deleted words it didn't recognize with a bleep. This clever feature (praised by critics⁹⁵) accelerated the input for the player: everyone wins.

Interestingly, there is one major difference with interactive fiction companies in the English-speaking video game scene: the fluctuation in quality of the parsers. Typically, the most famous companies that produced interactive fiction in English had a parser that was designed and improved inside the company and used for all the games the company produced; in fact, in general it was not just the parser that was identical: the games were created in a programming language that was internal to the company and then played embedded in an interpreter.⁹⁶ Surprisingly, this system wasn't really used in France; the only company that reused the same parser several times was Froggy Software—and it's probably because Jean-Louis Le Breton was a programmer on the majority of Froggy Software's games (he probably reused some code he wrote for other games). But as for other companies, the parser was seemingly rewritten every time, leading to parsers of uneven quality (for example, in 1985 Loriciels published *Orphée: Descente aux enfers* with a parser that could recognize complex sentences—such as “X, take key to Y”—as well as *Le diamant de l'île maudite* that had a two-word parser that recognized 90 words⁹⁷). As a consequence, there wasn't really an increase in the quality of parsers over time, which means that even at the end of

95 *Amstrad Magazine*, February 1986 issue.

96 This is the case for Infocom, Magnetic Scrolls, and Level 9 Computing games.

97 *Amstrad Magazine*, February 1986 issue.

the decade, some games commercially released by relatively successful companies could have a parser of very bad quality.

A particularity of the parsers in text adventures in French of this era was that they reacted to insults. The first games to do so were games by Froggy Software (which had in general a humorous tone): upon input of an insult, the game reacted in various ways: in *Même les pommes de terre ont des yeux*, a picture of a big and ferocious man was displayed, and the only way to continue playing was to type what he asked, that is to say “pardon à genoux” (“bowing my head and sorry”), whereas in *La femme qui ne supportait pas les ordinateurs* it was just a “Oh, that’s cheap!” A lot of subsequent games—especially the ones with a humorous tone—also recognized the input of insults and reacted in various ways: sometimes by insulting the player back,⁹⁸ or more creatively.⁹⁹

>

As for the tone of the games, we have to notice that a lot of them were humorous. The first one, *Le Vampire Fou*, had funny descriptions—and more generally, most Froggy Software games featured quite a lot of humor in the descriptions and the answers of the parser.¹⁰⁰ Numerous other games, even if they featured a long and complex adventure, had an overall funny tone or occasional funny descriptions; for example, *Le passager du temps* memorably featured a cat to the right side of the graphics, which could occasionally provide hints and commented on the action, leading to a lot of puns and pop-culture references. But in fairness, a lot of the humor displayed in the games wasn’t exactly subtle; when badly done, this kind of humor would lead to games that aged quite badly and were not exactly noticeable for their literary qualities. As a side note, the fact that a lot of text adventure games featured a lot of humor (in their descriptions and answers to the player) in an otherwise normal setting and adventure is an artifact that we don’t really find in

98 *JRAM* (ERE Informatique, 1986) displayed a picture of a pig, saying “Here’s a photo of you last year.”

99 *Oxphar* (ERE Informatique, 1987) displayed “You have to clear this infamy!” and the game suddenly changed to an Arkanoid-style mini game where you had to break the bricks forming the word INFAMY—which, according to reviewers, could take some time!

100 In *Même les Pommes de Terre ont des Yeux!*, a game set in some South American military dictatorship, the default response “I don’t understand” was the parser saying with a strong parodic Spanish-inspired accent that he didn’t understand what you were saying.

later French adventure games of any kind. Most point-and-clicks that were released by French companies didn't feature this sort of compulsive humor; apparently, this was a phenomenon that was limited specially to text adventures in the 80s.

>

Finally, let's talk a bit about the themes, settings, and stories told by these text adventures. It is interesting to note that the themes were very different from adventure games in English, which very often featured fantasy and sci-fi themes. Instead, French interactive fiction seems to have had a lot of games with an historical setting: Ancient Greece (*1001 BC*, ERE Informatique, 1986), the French Revolution (*Conspiration de l'an III*, Ubi Soft, 1988), World War II (*Mission Secrète à Colditz*, Soracom, 1986). A great variety of periods were explored¹⁰¹—with more than a few times the excuse of a time-travel machine (*Le passage du temps*). The more common among historical games were certainly games set in the Middle Ages (*La geste d'Artillac*, SRAM, *Montségur*, *Les Templiers d'Orven*, etc.); this may seem pretty logical, as fantasy settings are classic adventure game settings and offer a lot of challenges to the player as well as a very particular atmosphere. However, it is worth pointing out that there actually seem to be more games with a Middle Ages setting than games with a fantasy setting.¹⁰² Fantasy is certainly an English genre, and while the genre has become more and more popular in France, the “real” Middle Ages is a period that is part of French culture and that surely is familiar to more people. Speaking of typically English-speaking genres, there are very few French text adventures with a science-fiction setting.¹⁰³ There are also quite a variety of games that were set in a contemporary world, with, as Tristan Donovan points it out,¹⁰⁴ some adventures deeply grounded in reality and sometimes news: the publisher CobraSoft published games like *Meurtre à grande vitesse* in 1985, set in the French high-speed train TGV in which you have the two hours between Paris

101 It seems that French people have a particular taste for history; in video gaming, it is shown by, for example, the success of *Les voyageurs du temps (Future Wars)* by Delphine Software, or by the French studio Cryo Interactive (in collaboration with the French National Museum Reunion) authoring a very successful series of more than a dozen 3D point-and-click adventures, each in a different historical setting—Pompeii, Versailles, Greece, China, Aztec Mexico, etc.

102 That quite often blend in role-playing elements, in fact—among others, *Citadelle*.

103 Compared to the six science-fiction games written by Infocom.

104 <http://www.wired.com/gamelifelife/2010/06/french-touch-games/>

and Lyon to solve a murder, or *Dossier G: l'affaire du Rainbow Warrior* in 1985 as well, echoing the affair of the Rainbow Warrior that everyone in France talked about in the summer of 1985;¹⁰⁵ there was also *Mokowe* (Lankhor, 1991), which was about poachers in Kenya. Finally, the horror genre was quite popular as well.¹⁰⁶

>

Here are, in a nutshell, some of the aspects of French text adventures of the 1980s; as we can see, there are quite a few particularities that are worth noting, both as interesting for the history of the craft of French text adventures at this time as well as in comparison to other text adventure scenes. We will now move forward a bit in history to talk about the end of text adventures in France, at the end of the 1980s.

When Adventure Games Take Over: The Downfall of Interactive Fiction

As the 80s came to a close, it seemed that interactive fiction and adventure games with parsers were less and less common and more and more considered as a thing from the past.

This is easily seen in reviews of games in various magazines. Actually, it seems that starting in 1988, the critics considered text-based games as a prehistoric genre—even though successful text adventures came out as late as than two years before! For example, the game *Mike & Moko*, published by MBC¹⁰⁷ in 1988, got a fairly positive review in *Micro News* that still expressed not understanding why MBC was wasting good ideas (here, an adventure playable by two players simultaneously) by using an overused, worn-out gaming form; the review starts with, “The kings of adventure games with keyboard-input commands (though this genre

¹⁰⁵ As an interesting side-note, CobraSoft was the only French publisher I know that included some kind of feelies with some of their games: included with *Meurtres à grande vitesse* were some clues that the player was to discover in the train, such as a tape or some nails.

¹⁰⁶ With games such as *Le Vampire Fou*, *Le Manoir du Docteur Génius*, or *Le Pacte*.

¹⁰⁷ A French publishing company that specialized in text adventures with graphics; created pretty late (1985), it published half a dozen games before going bankrupt, and it designed its own authoring language, Jade/Jadis.

disappeared years ago) strike again!”¹⁰⁸ Another game, *Le Maraudeur*, released by Ubi Soft in 1989, gets a review in *Amstrad Cent Pour Cent* that rates the game fairly poorly, starting with, “Here is one of the last adventure games in the direct style of old games,” and stating (with a bit of humor), “The style of the game is not surprising at all, since it’s exactly the same as the old games (those released last year).” With this comment, it is as if in 1988, the critics suddenly felt (or maybe just decided) that the genre of the text adventure with graphics was old and outdated.

What happened in 1988 (or the year before) that triggered this sudden qualification of text-based adventure games as an outdated genre? We’ll have to look at the history of adventure games and the games released around this date to find some clues about what happened; and in fact, we find that we can consider 1987 as the year in which mouse-controlled adventure games became massively successful. In this year, ERE Informatique released *L’arche du Capitaine Blood*, which didn’t use command-line prompts but rather point-and-click systems and an icon-based conversation system; the game was extremely successful and an evident artistic success, with critics praising every aspect of the game,¹⁰⁹ and it enjoyed very good sales in Europe and around the world. 1987 was also the year of the release of *Maniac Mansion* by LucasArts Studios: this game was also very successful, established LucasArts as one of the best developers around, and popularized the system of point-and-click with a few action verbs. It was also the year of the release of *Le Manoir de Mortevielle* (*Mortville Manor*), developed by the French studio Lankhor; it was a point-and-click game set in a manor, where you had to solve a murder mystery. The game received very good reviews (noting its stunning voice synthesis feature) and is still considered a classic French adventure game. This combination of no less than three classic point-and-click adventures in the same year surely generated a lot of attention to the point-and-click system as a very welcome change (easier to manipulate, better graphics, and coincidentally better games); we can easily think that when text adventures with somewhat weak parsers, still pictures, and stories that weren’t as good as the aforementioned games were released after them, the comparison wasn’t particularly flattering and quite possibly made them look outdated.

In the following years, quite a few successful point-and-click

108 http://cpcrulez.fr/GamesTest/view.php?game=mike_et_moko

109 The game won the “Tilt d’Or” the same year.

adventures were released as well: as for classic French adventure games, we can list for instance *Les Voyageurs du Temps* (*Future Wars*), published in 1989, *Maupiti Island* (the sequel of *Le Manoir de Mortevielle*) in 1990, and *Croisière pour un Cadavre* (*Cruise for a Corpse*) in 1991. The LucasArts games also enjoyed some success in France around this time. This means that from 1987 to the beginning of the 1990s, numerous good point-and-click adventure games were released in France; the text-based adventure games, already considered as an outdated genre, couldn't rival this new genre, and soon enough the genre was becoming extinct.

Soon enough, the only publisher that released new French text adventures was Lankhor—and paradoxically, Lankhor was the publisher of *Le Manoir de Mortevielle* and *Maupiti Island*: the company made both text adventure games as well as point-and-clicks that supposedly ended up killing the text adventure genre. In 1990, *La secte noire* got some nice reviews—it is described in the September 1990 issue of *Joystick* as a “very classic, but still enjoyable, adventure”;¹¹⁰ its sequel, *La crypte des maudits*, was published in 1991 and had as a feature an improved parser: it was equally well received.¹¹¹ *Mokowé* was one of Lankhor's last games, an adventure about poachers in Kenya, with features such as activity in the village and in the jungle depending on the time. It was a very hard game but also very interesting.¹¹² Lankhor published a couple of other text adventures, as well as some point-and-clicks. Unfortunately, for reasons that remain unclear, they stopped making text adventures. They actually lost a lot of money with their 1993 point-and-click game *Black Sect* (only 3000 units sold, because of a mediocre interface and too-easy puzzles), which apparently prompted them to review their strategies and stop the development of some games. They ended up not making any more adventure games of any kind: from 1992 until its closing (in December 2001, because of some financial difficulties), the studio only made racing games.

Upon examination of the history of different studios of the time, we can note a very interesting pattern. In 1985, Eliott Grassiano, who worked at the time for Loriciels, founded Microïds with the help of the founders of Loriciels; Microïds went on to be very successful, creating the famous adventure game series *Syberia*. Eric Chahi, who wrote *Le pacte* for Loriciels, later worked for Delphine Software and created *Les*

110 http://cpcrulez.fr/GamesTest/view.php?game=la_secte_noire

111 http://cpcrulez.fr/GamesTest/view.php?game=la_crypte_des_maudits

112 <http://cpcrulez.fr/GamesTest/view.php?game=mokowe>

voyageurs du temps (*Future Wars*), *Another World* (otherwise known as *Out of this World*), and *Heart of Darkness*. When ERE Informatique went bankrupt, a lot of the people who were working on its games founded Cryo Interactive, which became a very successful company, creating for example *Under a Killing Moon*, *DragonLore*, *Chroniques de la Lune Noire*, *Faust*, and a series of historical adventure games in 3D that were successful in France (with titles such as *Versailles* and *Egypt:1156 B.C.: Tomb of the Pharaoh*). Also, we can note that Infogrames and Ubisoft, founded in the 80s, published quite a number of text adventure games during this period (but not only text adventures). The pattern here is that a lot of the people involved in the creation of text adventures in France in the 80s went on to work on a variety of other adventure games that were very successful worldwide, prompting some critics to talk about a “French touch” in adventure games: the people that created the French touch had the opportunity to create text adventures first, and we can thus think of those text adventures as precursors of this French touch.

After the last Lankhor text adventure (in 1991), this seems to be the end of text adventures in French. I couldn’t find any other text adventure on any computer with a date of release later than 1992. Thus, in a very similar fashion to what happened in other communities—the English and Spanish ones, for instance—the genre seems to be definitely dead. In fact, it is just hibernating, as we will see in the next part: for the French community, springtime came at the beginning of the millennium.

The 2000s: The Genre Rises from Its Ashes¹¹³

The year 2000 saw the resurrection of interactive fiction—at least, an organized attempt to centralize the interest in interactive fiction, both playing and creating. The Yahoo! mailing list “Inform_fr” was created this year and featured discussions about “French adaptations of Inform text adventures [as well as] discussions about translations and creations of ‘interactive fictions’ [sic] in French.” Several members of this mailing list stuck around and are still active members of the French community.

The real kickstart for French creation of modern interactive fiction in Inform is certainly the translation of the Inform 6 libraries. The

113 The interested reader can refer to the timeline of modern French interactive fiction written by Eriorg and published in SPAG#47; this timeline is fairly complete up until 2006. It is available from here: http://www.ifwiki.org/index.php/History_of_Interactive_Fiction_in_French.

translation was done by Jean-Luc Pontico, who released them in January 2001, along with *Aventure*, the (first-ever) translation of the classic game *Adventure*. The next year, Eric Forgeot released a demo of *Le pouvoir délaissé*, an upcoming game; this was the first attempt at the creation of a novel French interactive fiction game, but unfortunately it is still unfinished to this date (the author moved on to other games instead).

The first completed original interactive fiction game is *Filaments*, by JB Ferrant. The story of the game is about a young girl, Margot, living in Paris and uncovering strange and surreal events; the game is mainly an adventure game, with quite a bit of humor as well, but mostly serious (and even dramatic) events. The game is fairly long and unfortunately has a few annoying bugs, but it remains a very good game; it is most certainly a modern game as well and features no graphics. It was translated to Italian later the same year and won the Best Game in Italian of the Year award.

JB is actually a very important author in the French interactive fiction community, if not the most important. After authoring the first original French interactive fiction game, he went on to release a couple more games in 2004 and 2005. He then undertook a huge project, a very ambitious game named *Ekephrasis*; it is actually the first French game with graphics, sound, and music and is a long game featuring a fine arts teacher traveling around Europe (complete with actual photos of the monuments he visits) to uncover a mystery involving forgers and Renaissance art. Recently, he released *Works of Fiction*, his first game in English; unfortunately, no French version of the game is available. He also participated in a handful of Speed-IFs.

The next step in the development of the French interactive fiction community was the creation of a message board in August 2004 in an attempt to centralize the people interested in reading interactive fiction (both in French and English) and creating it too. The forum has been moderated by Eric Forgeot and remains quite active today, as more and more members (and potential authors) have joined the forum since its creation.

Eric Forgeot is a central figure in the French interactive fiction community; under his pseudonym “Otto Grimwald,” he has been the moderator of the forum for years and often gives technical advice to young authors who ask for help on the forum. He authored quite a number of games, winning the French IF Comp in 2007 with *Les Heures du Vent* (*Hours of the Wind*) and participating in every Speed-IF event that

was organized. He also provided a few technical advances to the community, as he created the Inform 7 extension that allows the creation French games with it (he has been using Inform 7 for his games since it was out), as well as translating the libraries for JACL and Hugo into French and creating a Linux Live-CD complete with IDEs, interpreters, and games to get started in interactive fiction. Recently he wrote a tutorial for Inform 7 for *lesiteduzero.com*, a famous French website that compiles a variety of tutorials for programming languages, which brought new people (and potential authors) to the community.

The French interactive fiction community was becoming more and more organized; it was only a matter of time until an equivalent of the IF Comp was created. The idea was prompted by “Stab” in April 2005, and thus the first French IF Comp (or Minicomp, because of its small number of games that are entered every year) was organized by Eric Forgeot shortly after. The French IF Comp has been organized every year since then and features in general no more than four or five games; it always provides an opportunity for people to try to complete a project of theirs, and the community, though small, tries to get involved as much as possible.

The first French IF Comp saw five participants entering: the winner was Adrien Saurat, with a humorous one-room game called *Le cercle des gros geeks disparus* (*Dead Geeks Society*); he went on to win the 2006 edition with a post-apocalyptic game called *La Cité des Eaux* (*City of the Waters*) and the 2009 edition with a story of chimney-sweeping men in an underground city, *Catapole* (this game was played in an “international edition” of Club Floyd in 2010¹¹⁴); as a matter of fact, he won every edition of the French IF Comp he entered. He was also a participant of the first two French Speed-IFs in 2007; he recently entered IntroComp 2010 with a game called *Plan 6 From Inner Earth*.

Another event among the French community was its participation in the Commonplace Book Project: as part of a museum exhibition about the Commonplace Book by H. P. Lovecraft, several interactive fiction games were created using themes from this book. About half a dozen games were written in English, but the French community (as well as the Spanish one, as a matter of fact) participated in this project; various members of the community wrote a chapter using a sentence from the book, and the various chapters were tied in a Glulx game (with pictures

114 A transcript of the session is available at the ClubFloyd website: http://www.allthingsjacq.com/intfic_clubfloyd_20100704.html.

and music) that was ultimately shown at the exhibition. This was the first (and to this date, the only) game created in a collaborative effort, and was quite a success.

Interestingly, the French community also rediscovered the concept of Speed-IF and organized four of them, the first one being in the summer of 2007. A few of these games were actually expanded by their authors to lead to reasonable-sized (and reasonably bug-free) games. Moreover, the organization of such Speed-IFs prompted the organization of a Speed-IF in English (organized by Jacqueline A. Lott, who is also a regular visitor to the forum) on the theme “The Francophones stole the spirit of Speed-IF!”

The Contemporary French-speaking IF Scene

Judging by the very different history of French interactive fiction, one can ask how, and to what extent, the contemporary scene is shaped by this history.

The answer is brutally simple: no direct legacy of this history remains among the contemporary scene. In fact, no author of the contemporary French-speaking IF scene declares to be influenced by any 80s games whatsoever, and a lot of them didn’t discover interactive fiction because they played it back in the 80s; moreover, French games that were published in the 80s are seen as outdated, with very little to learn from them. The contrast with the English-speaking scene is striking: a lot of people writing and playing interactive fiction in English played Infocom games, or Scott Adams or Magnetic Scrolls or even Phoenix games; *Curses*, the first game written in Inform, has been described (even by its author) as an interactive fiction exactly in the style of Infocom games; Infocom games are still praised as being the canon of interactive fiction and for their literary qualities and inventiveness. In comparison, the contemporary French-speaking interactive fiction community barely makes any reference to 80s adventure games in their discussion or in their creations, and the history of 80s text adventures in France is not very well known to the members of the French interactive fiction community (as shown by the present article, which is an attempt to write this history for the first time ever).

In fact, we could say that there is no common interactive fiction culture that ties the members of the French interactive fiction

community together. This is a major difference from the English-speaking interactive fiction community. Is that bad? In a way, yes, but it is actually a double-edged sword. Surely it is a drawback: the fact that no company creating interactive fiction in the 80s was as extremely successful as Infocom means that not a lot of people were playing interactive fiction in the 80s, and if they did, they might not remember such games as extraordinary, breath-taking, epic adventures. In fact, the great success of Infocom probably relies on two factors: the availability of their games on every microcomputer, and the quality of their games, which were long, epic, hard, and very pleasant adventures. Neither of those factors are present in the 80s French interactive fiction scene. The games were for the most part exclusive to one platform, and some were available for only a couple of computers; moreover, they aged pretty badly and were quickly considered as outdated. As a result, while a lot of English-speaking people played and enjoyed Infocom adventures and can nowadays find games that are very similar to them, a significantly smaller number of people played interactive fiction in the 80s, and even though they probably have their favorites among those games, they appear outdated, a thing from the past. Moreover, their form was very different from what interactive fiction is nowadays: there is no automatic identification between modern interactive fiction and 80s interactive fiction. Conjugating all those factors, this leads to a very, very small audience for interactive fiction in French, and this is an enormous drawback; surely the success of an interactive fiction company such as Infocom would have given the French community a bigger base of players and potential authors and might even have shortened the “hibernation period” we mentioned before. As another proof for this, we can take a look at the Spanish community: the company Aventuras AD created interactive fiction games from 1988 to 1992 and was massively successful, spawning a great interest in interactive fiction, creation of fanzines, and so forth. The community then entered a hibernation period and woke up in 1997, with the creation of a newsgroup (and then a mailing list, and then a website with forum) about interactive fiction, and the same year the first competition was organized:¹¹⁵ the success of Aventuras AD (as well as the interest generated by the success of this company) gave the Spanish community a wealth of potential players and

115 SPAG #49, “A History of Spanish IF”

authors and makes this community bigger¹¹⁶ and older than the French community.

But in a way, not having canonical references for what good interactive fiction is means that canonical interactive fiction in French is still yet to be written: the community is only a few years old and has the opportunity to attempt to create influential games and explore new game design and storytelling paradigms. Moreover, since the community, as well as the number of games, is small, a lot of the members of this community have played the majority of French games, and this hopefully creates an exchange, a reciprocal influence that can make the whole community aware of what is done in itself and give authors new ideas, which in return will influence other authors.

But this affirmation is to be contrasted: as a matter of fact, a lot of people in the community can read or speak English and thus can play English interactive fiction games. As a consequence we cannot really talk about French IF as a “closed world” where everything is yet to be (re-) discovered. In fact, the French-speaking IF community is very much aware of what happens in the English-speaking community and sometimes talks about various events happening in it; a lot of (and perhaps even the majority of) French-speaking authors played, and continue to play, interactive fiction in English. Thus, a lot of the IF theory that is discussed in various newsgroups, forums, or webzines is known to the French-speaking authors; they know about game design, storytelling, conversation systems, and other important questions, as much as an English-speaking author knows. Thus, as any other author, their creations and designs are built upon these theories: French-speaking games are every bit as modern as English-speaking games. But unfortunately, it is unlikely that a debate about an aspect of the theory of interactive fiction, or a novelty in a game, will have any influence on

116 Another possibility to explain the size differences would be to take into account the number of people speaking Spanish/Castellano worldwide (half a billion) and the number of people speaking French (a quarter of a billion). I don't know if, as it seems to be the case in the French-speaking community, a lot of people playing interactive fiction in Spanish are Spanish; if it is the case, then considering that France has 65 million inhabitants when Spain has 45 million, this would definitely prove that the ratio of people playing IF in Spanish is greater than the ratio of people playing IF in French. But even if we consider the worldwide numbers, the Wiki of CAAD shows that the average number of games released in a year is about 25, when in the French community 10 released games means a good year; the ratio still seems higher.

English interactive fiction: the language barrier, as well as the fact that the English-speaking community is a busy one, means that very few people of the English-speaking community will look at what the French community (or as a matter of fact, any other community) produces and talks about.¹¹⁷

Still, if authors of interactive fiction in French know about what the English community is discussing, how and to what extent are they influenced by interactive fiction in English? First of all, it turns out that a few of the members of the French-speaking community played some Infocom games before joining the community, either at the end of the 80s on their microcomputer¹¹⁸ or by rediscovering those games on abandonware websites;¹¹⁹ once again, Infocom games created an interest for interactive fiction for some people. But even though some people played those classic games, they are not quoted as a major influence among the community. They are not considered as “classics” in the community,¹²⁰ but those who played them agreed that they are indeed very good games. Instead, the French community plays a lot more games from the modern era—as we mentioned before, the French community keeps up with what the English community is doing. Sometimes, it’s a modern English game that prompted someone’s interest in interactive fiction—for example, JB Ferrant’s first interactive fiction game he played was *Aisle*.¹²¹ More generally, a lot of members of the French IF community play interactive fiction in English, either Infocom’s

117 Very few people from the English-speaking community had a look at what was produced by the other communities; the only examples I know of are reviews of *Ekphrasis* by Emily Short (personal website) and Felix Plesoianu (SPAG #47), a translation of *Olvido Mortal* by Nick Monfort, and playthrough of *Catapole* at ClubFloyd last year (organized by Jacqueline A. Lott, who is also an occasional contributor to the forum of the French community).

118 Cf. SPAG #47, “Interview of Adrien Saurat,” and Eriorg’s presentation post on the community forum.

119 Such as Samuel Verschelde (“Stormi”), who found *The Hitchhiker’s Guide to Galaxy* and *A Mind Forever Voyaging* on such a website and then discovered the French-speaking interactive fiction community.

120 But again, there don’t seem to be a lot of games considered universally as “classics.” There are several reasons behind this: no common interactive fiction culture and little heritage left by the 80s (thus no potential “old classics”), perhaps even a reluctance to call games written in English “classics” (because some people potentially can’t read English), and a very young community that hasn’t produce a lot of games yet. The only candidate could be the first French game, *Filaments*—we’ll talk a bit about this later.

121 SPAG #47, “Interview of JB”.

games or more recent ones,¹²² but very few cite them as major influences.

But then what exactly are the influences of the French-speaking IF community? What prompted the interest in interactive fiction of the members of the community? There are multiple answers¹²³. It appears that in almost all cases, people stumbled on a game that they liked and that made them continue their search for interactive fiction, then landed on the forum of the community;¹²⁴ some of them had already played this kind of games before,¹²⁵ whereas for some¹²⁶ it was a totally new discovery. The games that people stumbled upon were sometimes Infocom games,¹²⁷ sometimes modern interactive fiction games in English,¹²⁸ but quite often modern interactive fiction games in French. In fact, *Filaments* can be considered as a cornerstone in this regard: a lot of people that joined the community after its release said that they found this game and loved it, prompting them to look for more games of the same kind.¹²⁹ *Filaments* may in fact very well be considered as the first classic of the modern era of French-speaking interactive fiction. But as a matter of fact, other games in French are sometimes quoted as being the game that generated interest in interactive fiction.¹³⁰ But interestingly, it

122 Cf. various sources, such as SPAG #47 (“Interview of Adrien Saurat”) and the list of played games on the IFDB profile of Eric Forgeot (“Otto Grimwald”), Grégoire Schneller (“Eriorg”), Samuel Verschelde (“Stormi”), and myself.

123 For a more detailed account of those factors, the interested reader may have a look at the presentation topic in the community forum (<http://ifiction.free.fr/taverne/viewtopic.php?f=1&t=7>), as well as the interviews of JB Ferrant and Adrien Saurat in SPAG #47.

124 This is actually not true for some of the most recent members, who found the tutorial for Inform 7 posted by Eric Forgeot on a famous website of tutorials of programming languages; information is lacking concerning the origin of their interest in interactive fiction.

125 Such as Adrien Saurat, Eric Forgeot, and Grégoire Schneller.

126 Such as JB Ferrant, Benjamin Roux, Samuel Verschelde (who describes having been dumbfounded when he found (in 2000) that this kind of games existed), and myself.

127 Such as for Samuel Verschelde and Adrien Saurat (*The Hitchhiker's Guide to Galaxy*) or Benjamin Roux (*Zork*).

128 Such as for JB Ferrant (*Aisle*) and Grégoire Schneller (*Savoir-Faire*).

129 Such as for Grégoire Schneller (who played *Savoir-Faire* first, but didn't really like it, and then played and loved *Filaments*) and myself.

130 Such as the aforementioned winner of the French IF Comp 2006, *La cité des eaux*, or even the humorous one-room “few-moves” *Les espions ne meurent jamais* (*Spies Never Die*).

seems that the members of this community share very similar interests that could somehow explain (or be put in relation to) their interest in interactive fiction—other than an interest in computers and programming. Those influences are, among others, CYOA books,¹³¹¹³² role-playing¹³³ (with, interestingly, a few authors of interactive fiction being or having been game masters in various role-playing games), role-playing computer games,¹³⁴ and quite logically adventure games.¹³⁵ Those influences may possibly be quite common in other communities such as the English one, of course, but they are worth mentioning here for the reason that few people in the French-speaking community came to it (and became part of the “modern era” interactive fiction community) because they knew interactive fiction from games they played in their youth, for instance: we’re looking here at possible influences that could, by their similarity to interactive fiction,¹³⁶ explain why most people became interested in interactive fiction when they first discovered it only a few years ago. Trying to figure out what are the previous influences of the newcomers in interactive fiction is useful to determine which fields are closely related to interactive fiction (thus possibly giving some clues about what interactive fiction is similar to and what characteristics are similar), as well as understanding what can bring people to interactive fiction (and what potential audiences can be interested in interactive fiction). Furthermore, it is easy to determine those influences for the French community, because it

131 CYOA books were very popular in France in the 80s and the 90s. They were edited in France by Folio Junior under the collection *Un livre dont vous êtes le héros* (*A book in which you are the hero*), which incidentally is the name most people now use to talk about gamebooks. Several series were translated and edited, such as the *Fighting Fantasy*, *Lone Wolf*, and *Sorcery!* series, as well as numerous books by Herbie Brennan. The *Choose Your Own Adventure* series of gamebooks is, however, unknown in France.

132 In fact, JB Ferrant collects the Steve Jackson gamebooks (he says he has over 300 books now), wrote three gamebooks (that can be bought on his website), and his second interactive fiction, *La Mort Pour Seul Destin* (*Death as Your Only Fate*) is an homage to the *Sorcery!* series of gamebooks. I myself was an avid gamebook player (around 50 books) in my youth.

133 Such as for JB Ferrant, Jean-Luc Pontico, and Adrien Saurat.

134 With the *Ultima* series being quoted as a reference, as well as *Baldur's Gate*.

135 The *Monkey Island* series seems to be a reference for most people, as well as the *Myst* series, and countless others.

136 What those influences had obviously in common with interactive fiction were things like interactivity, puzzle-solving, and branching narratives; we could probably find more.

is a small community where, so to speak, everyone knows each other.

Finally, let's have a look at the games produced by the French community. It might not be very relevant to try to find any pattern in the games created by this community, because the number of games, as well as (and perhaps more importantly) the number of authors, is very small: there are about 60 original games belonging to this "modern IF" era,¹³⁷ and the number of authors is about a dozen.¹³⁸ Thus, the patterns we may end up finding depend too heavily on individual preference. We'll then just note that a lot of games have a contemporary setting; also, fantasy and medieval games are very well represented (as well as a couple of "historical" games set in some ancient period).¹³⁹ We can also note that quite a few games are actually very short,¹⁴⁰ and games in general are of short length;¹⁴¹ however, this doesn't seem to be too peculiar when compared to modern interactive fiction in English, for example.

In a nutshell, contrary to bigger communities, the contemporary French-speaking interactive fiction scene is not influenced by any previous history of interactive fiction; this actually harms the community, because interactive fiction is not an established genre in the eyes of a certain gaming audience, and it lacks any reference point in the past that players could associate with interactive fiction. This is, in a way, a totally new genre, which can deter players from trying it, as well as the fact that there will be no nostalgic players that discovered and/or participate in the modern scene to relive similar experiences from games

137 We have to note that some of them are actually games created by a young author to get more familiar with an authoring language.

138 Actually, there are exactly 16 authors, only 9 of whom authored more than one game.

139 The *Wired* article that we quoted before wrote that French games of the 80s had very different themes from the games written in English; far away from fantasy and sci-fi, French adventures were more rooted in reality, said the article. This is actually still true, as most recent IF games in French are set in our modern world, and very few belong to the sci-fi genre. As for the fantasy genre, it is indeed a bit represented, but medieval settings are very common in French adventure games (thus making a fantasy setting maybe less far-fetched): it was represented in the 80s by games such as *Citadelle*, *Montségur*, and *La geste d'Artillac*, and as for recent IF, a lot of games by Eric Forgeot have a medieval setting, out of personal interest it seems.

140 Because of the games created to get familiar with the programming language, as well as Speed-IF games, but not only.

141 The only long games are actually JB's *Filaments* and *Ekephrasis*, as well as Loïc B.'s *Largo Winch* and *Enquête à hauts risques*.

they played in their youth. Thus, the community is still very small, and it seems that its audience is equally small. Interestingly, the members of the community thus have different and composite influences, which surely leads to different approaches, tastes, and takes on interactive fiction—but in fairness, the community is probably too small and too young to make this mean something.

Conclusion and Perspectives

The goal of this article was to present a history of, as well as some more general perspectives about, the French-speaking interactive fiction community. As we saw, this community is very different from the English-speaking community on many levels.

Writing the history of interactive fiction in French in the 80s for the first time, we saw that this history was a very different one from the one (centered on English-speaking countries) that is usually told. The influence of Infocom games is negligible to non-existent; as a consequence, the form of interactive fiction in French in the 80s was closer to adventures with graphics and a parser than to purely text-based games. We also saw that while a few companies were fairly successful, none of them had the success or the influence and the market dominance of a company like Infocom;¹⁴² the reasons are numerous, from the late blooming of the market, thus giving them less time to get established as giants before the rise of point-and-click adventures, to the possible concurrence between skilled studios, or even a lower literary quality that made that games sometimes quickly outdated.

The consequences on the contemporary French-speaking IF scene are very important: because interactive fiction didn't have as much success in the 80s—and in fact one could argue that they simply didn't exist before the modern era—the community lacks a large base of players (and potential authors) that could probably have been brought by a greater popularity of the genre some decades ago. The community has found some other influences, and various people from diverse backgrounds are now part of the community. Still, even if the

142 On the other hand, the French interactive fiction scene did enter a state of hibernation very similar to what happen to the English and Spanish communities, probably because of the rise of the point-and-click genre; the fact that the whole genre didn't rely on one enormously successful company didn't prevent this hibernation.

community seems to have reached a maturity and a stability that ensures that it will continue to create and stay active for some years, things are not looking wonderful: the community is still very small and doesn't seem to grow (or to increase its potential audience significantly by reaching out to more players) very fast. This created a paradoxical situation in the community, where the few authors that keep the community alive are sometimes tempted to write their own games in English so that they could be played and reviewed by a greater number of people. The fact that very few English-speaking interactive fiction players play and review any game that's not written in English contributes to a sort of one-way relationship that could be harmful to every other, non-English-speaking community.¹⁴³ Of course, the English-speaking community is hardly responsible for that; the "culprit" is the language barrier, and the fact that English is nowadays widely acknowledged as the global language. This is a challenge that the French community has to face: to manage to keep a healthy number of games published in French while looking for ways of reaching new audiences—for example by making their work more well-known among the English-speaking community. This is definitely a crucial time for this community, and there is certainly a lot to do for its members.

>

Acknowledgments: The author wishes to thanks Grégoire Schneller for his invaluable help digging in old magazines to find references to games in English in France, as well as the French IF community for "keeping French text adventures alive"!

¹⁴³ This is seemingly what happened to the German community for a time, where every German author switched to English as the language of their games because it would attract more attention to their games.