A Writer's Guide to Interactive Fiction

(or, why am I still interested in a videogame fad from when I was ten years old?)

This is a talk I gave at Readercon 21, July 2010. I wanted to describe IF for writers (and readers) of traditional fiction.

This is nearly the talk I gave, anyway. These are the notes I wrote up beforehand. On the one hand, this makes them much more coherent and less "um"-ish. On the other hand, of course, notes never quite survive contact with reality. The introductory bit about IF in popular culture? I didn't have to explain that -- because Charlie Stross practically did it for me, in his guest-of-honor interview two hours previously. He *was talking about his novel* Halting State, *which is written in second-person, "the narrative voice of computer games". Thank you, Mr. Stross, for anonymously introducing me. :)*

Interactive fiction -- text adventures -- was the most advanced computer game technology of the early PC era. These games were the best-selling and most highly-rated games of the early 80s. Then they were overtaken and crushed by games with graphics, and that was the last anyone heard of them.

Most people.

Only that's wrong, in a couple of ways. IF has lingered in popular culture. If I mention "...a twisty little maze of passages", you know they're "all alike". (*I* know they're all different, but that's the other side of the chasm.) The call-and-response form of IF -- "TAKE LAMP", "I don't know how to take the lamp" -- has been a familiar gag for thirty years now.

And somewhere, in the basement of the computer game world, a few people are still writing text adventures. A few people... a few games... Actually, lots of games. Over a hundred last year, with rapidly-advancing development tools. This is a more active fandom than you might think.

So why am I interested? Books are more popular than IF; even short stories pay better than IF. What can I do with text adventures that I can't do with traditional fiction?

Interactive works have two features going for them:

The first is *interactivity* -- obviously. The game reacts to what the player does. He or she discovers the situation, the rules, the world, by picking it up and *playing* with it. Don't tell, don't show, but rather *emulate*.

The second factor, less obvious, is *complicity*. I don't just mean the first-hand experience of discovery, although that's powerful. More powerful is the first-hand experience of choice. Whatever direction your story goes, the player has chosen it. Even if it's a guided choice, or a forced choice -- and all games have an element of forced choice -- the player has to make the decision to go forwards (or ctrl-alt-delete). Not just "it happened to me", but "I did it."

Those are general features of computer games; they apply to any program that offers a simulated world and an avatar. Let me talk about what IF in particular can do.

IF is characterized by text output and text input -- it prints normal narrative prose, in response to an open-ended command prompt. (I can go off on graphical adventures, and ways they are analogous to text adventures, but in this essay I'll stick to text games.)

The text interface has one clear advantage for me: I can write, but I can't draw. Producing the art and music assets for a standard commercial videogame requires a team of artists, whereas a text game can be created by one person and a word processor. This is no small matter. Text IF has a tradition of rapid-fire experimentation and iteration. I can write a short game in a month (admittedly, I'm fast). That's much more like a short story than like even a short computer game.

The command-line interface... is a strength and a weakness. Most everybody remembers IF's failures rather than its successes: it's the computer game that rejects everything you type. This isn't actually true, for experienced players; but that leaves a big learning cliff to scale. I'm not going to get into that here, because it's about the most heavily-discussed topic in IF today. I'll just insist that there *is* value in the parser, and I'll move on.

IF has more people than you're used to. I mean in the sense of first-person, second-person... There's this nice formal structure which was invented, almost accidentally, by Will Crowther in the very first Adventure game. The prose is second-person: "You are standing at the end of a road, before a small brick building." And the parser, the game-master, is first-person: "I'm afraid I don't understand."

This structure is borrowed from paper-and-pencil RPGs. Just as with RPGs, the *player* is distinct from the *protagonist*, the role being played. But the balance is a little different from traditional D&D. The protagonist is something of a collaboration: the game provides background, the player provides choice, and... everything else is up for grabs.

I mentioned failure messages. Consider these different ways that game command can fail:

- "I don't understand the word 'fly'."
- "You can't fly."
- "You haven't learned how to fly."
- "You try to fly, but the winds here are too strong."
- "You approach the edge, but you are afraid..."

Each of these expresses something different about the game. This is where IF starts getting fun: you can convey information about the command structure, the environment, the protagonist's physical abilities, the protagonist's character, or even the possible course of the story.

You can play all sorts of tricks with these layers. Here's a transcript of a game from a couple years ago, by Admiral Jota, called Lost Pig:

> **Outside**
> Grunk think that pig probably go this way. It hard to tell at night time, because moon not bright as sun. There forest to east and north. It even darker there, and Grunk hear lots of strange animal. West of Grunk, there big field with little stone wall. Farm back to south.
>
> *> NORTH*
> Forest look dark. Pig probably some place in there, but Grunk not know which way to

> go. Not want end up lost in forest with no pig.

Here the protagonist is this cute little orc character. This is broad characterization through narration. But notice that the roles of the protagonist and narrator have gotten mixed together, and the game is quite consciously avoiding *either* first-person or second-person prose: they're both "Grunk". You are Grunk, and Grunk is telling the story. And of course the naive POV becomes an acceptable excuse for parser problems.

This is from Rameses, by Stephen Bond:

> Gordon struts into the room, fresh and dripping from the shower. He is wearing nothing apart from a towel which is wrapped loosely around his waist. Very loosely around his waist. As he walks in, the towel falls off. He tries to make out it was an accident, but it obviously wasn't. He just did it so we could all admire the enormous size of his cock.
>
> It happens too often for it to have been an accident.
>
> \> *GET UP*
> I'd rather stay here for the moment. My lie-in is the best part of my day. Gordon walks over to the sink. He pretends to root in his medicine bag for something, but I can see he's really admiring himself in the mirror. God, that bastard spends so much time looking at his own reflection, admiring his face, and his chest, and his muscles, and his dick.
>
> \> *TALK TO GORDON*
> Please select one:
>
> (1) "Gordon, put on your fucking towel. Nothing we haven't seen before a zillion times."
> (2) "Will you look at that weener! I've seen bigger dicks at a Little Richard concert."
> (3) "Oh hello sailor!"
> (4) "Wow! Look at that monster! We're so impressed!"
> (5) "Hi Gordon!"
>
> Select a choice or 0 to keep quiet. >> *2*
>
> No, I couldn't say that! Little Richard concert? Everyone here would think I was a complete weirdo.
>
> ...Not that they don't already.

Here the protagonist is this sullen asshole, who is characterized entirely by his view of others and his *refusal to do anything*. You can't get up, you can't tell off these bozos. The player's intentions are completely short-circuited, and that's how the game is built. It's an extreme example of that collaboration I mentioned; most games don't go this far, but nearly every IF game has an element of it.

A final example: Violet by Jeremy Freese:

> You told me you were making progress. Then I find out you haven't finished a

paragraph in five months. What did you expect?

So now here you are, up in your office. I am sorry that today is a Saturday. I am especially sorry that today is your thirtieth birthday. We had all kinds of adventures planned. I refuse to feel guilty.

Of course I am still upset. I will try to set that aside and be encouraging. I really do want you to succeed today.

Please just write.

> *EXAMINE ME*
You look tired, which is understandable given the length of our "discussion" last night.

You're carrying nothing.

You've a tattoo on the back of your hand.

> *EXAMINE HAND*
"Tattoo" isn't right. You scrawled "TYVTWD" on the back of your hand before leaving this morning. It stands for Take Your Violet To Work Day. This is your plan: you are going to get through today by pretending I'm here with you.

Just so we're clear: you are actually alone, and presumably I am back at our apartment, packing and crying.

In this game the role of the parser is played by the protagonist's awesome Australian girlfriend. Except not his *actual* girlfriend; rather, it's his imagined version of his girlfriend, hovering over him, pleading with him to finish his thesis. So we have characterization by presenting a character's description of the protagonist, except we really only have *his* not-very-reliable description of *her*, because she's off-stage but narrating the whole story...

Maybe just have an overly baroque mind, but I think that's awesome.

The second leg of my interactivity stilts was "complicity". That's where you get the player to do something clever and feel clever... or do something dangerous and feel brave... or do something *wrong* and feel like it was *inevitable*.

The archetypical moment of a horror movie is the protagonist reaching for a door, and everybody in the audience screaming "Don't open that door!" But the archetypical moment of a horror *game* is when you face that door, and you think "Opening this door is the worst thing I could do right now," and then you type "OPEN DOOR". As Bujold once said: "It will be simple and elegant, and *you will do it to yourself.*"

The naive response is, "The player will never do that, because players have free will." Short answer: no. We have lots of ways to put the player in that position. The simplest one is, you lock all the other doors. But it gets subtler than that.

The thing is, the game isn't just offering a simple choice between "open the door" and "don't open the door". Choose-your-own-adventure books have those simple choices, and they never wind up being very compelling. IF does more.

Maybe the game has a complex puzzle built around the door, and the player has been working hard to solve it. Opening the door is the reward, and you want your reward, *even though* that door is the worst thing you could do right now.

Maybe the game has been building up your character as a fatally curious person, and you've *bought into* that character, so you *want* to open the door.

Maybe the game has convinced you that the protagonist is a loathesome creep, and *deserves* to open the door.

Every game -- and I'm talking about all videogames again -- has many layers of choice going you. You have the physical interaction with the controller or mouse or keyboard. You have the moment-by-moment choice of which way to face or what to touch. There are immediate goals, and story goals, and long-term character arcs. There may be choices about how to solve this puzzle, or how to end this chapter, or how the overarchive game plot will work out.

At any moment, some of these layers are shallow and transparent to the player -- he or she isn't thinking about them. Other layers are rigid, completely controlled by the game. Other layers are complex and interesting; those are where the player's attention is focussed.

Every games-versus-art discussion ever gets bogged down in this canard of "the player and the author fighting for control of the story." It is 100% French duck, because the player and the author *simultaneously* have control -- at these different levels.

Which level is which depends on the particular game. One game might give the player a wide range of choice in arranging objects to solve a puzzle, but string puzzle-chapters together in a fixed order. Another game might offer only a few choices at a time, but give each one wide-ranging consequences. A game might have one ending, or several endings that the player chooses from at a critical moment, or several endings that derive invisibly from everything the player does. These are all valid designs.

IF has some conventions of interactivity: the controller is a keyboard. The command prompt accepts a certain range of well-known verbs and a wider range of possible, hinted-at verbs. (Although even that can vary, as _Rameses_ demonstrated.) But beyond those levels, there are lots of possibilities. All of the "door" examples I noted above can be split apart into these different scopes of player and author choice.

Given all of that awesomeness, why isn't everybody writing IF?

I mentioned, or rather glossed over, the learning curve. An adventure game works when the player has a clear sense of what will *probably* work -- the familiar core commands of IF -- and a broader, peripheral view of what *might* work *if it makes sense in game terms*. An experienced IF player has that sense. A beginning player, plopped down in front of an IF game, is lost in the fog.

If people won't play the games, you have to be a serious fanatic to keep writing them.

We've spent years debating possible solutions to the problem: tutorial games, guiding interfaces, optional alternatives to the command line. None of them have precipitated a tidal wave of IF interested. (The problem may not be any of the ideas -- it may just be a question of getting people to *try* them, which is why I'm giving talks at conventions.)

I am not personally interested in dropping the command line from IF. IF input is the root of all the tricks I've been describing; if you simplify the interface, you can't build games of the same complexity.

At the same time, I'm not particularly interested in making the IF parser *more* complex. We're often asked why the IF games of 2010 recognize roughly the same command language as the games of 1983.

It's not *exactly* the same. We have more flexible parsing of noun and verb phrases, we have more options for handling totally off-the-wall newbie mistakes, and Emily Short has done inestimable work on context-sensitive keyword (and keyphrase) parsing in dialogue scenes. But yes, the underlying command *model* is the same as in Zork: a verb and maybe a couple of object phrases.

This is deliberate. A basic principle of all gaming is that the range of choices should match the range of interesting results. We know how to *build* games that react to a verb-plus-zero-to-two-objects in useful, comprehensive, and comprehensible ways. (That is, games where every response to such a command is *either* an interesting success or an informative failure.) Our traditional strawman for extension is adverbs. It would be easy to adjust the parser to accept "OPEN DOOR SLOWLY" or "OPEN DOOR QUIETLY", but it would *not* be easy to create a game in which these variations were a consistent, integral part of gameplay. So we don't try.

I freely admit that this may be convention, habit, and rheumatoid arthritis binding us (or, if you like, binding *me*) to an outdated interface model. If so, so be it. Somebody else will have to set me straight.

Aside from the interface, I find that not all kinds of stories are tellable as IF. In fact, many kinds of stories *aren't.*

Scenes of exploration are of course standard IF fare. The same goes for physical manipulation: the swords, books, locks, and discoverable machines of IF tradition. Beyond that -- in principle, any kind of action is possible. We have that open-ended command line, after all. If the game demonstrates a set of story-specific actions at the beginning, the player will happily pick them up and apply them.

But this leads to a second-level trap. Having taught those actions, it's hard to go beyond them and get the player into *different* kinds of actions.

Let me put it this way... A novel may start with the protagonist running for her life. That's a fine way to start a novel. But the reader won't expect chapter two to *also* be the same person running for her life. The hot entree is a teaser. The story may continue with a flashback (explaining why she's running), or it may expand the focus to solving the broader problem, removing the underlying threat. There will be scenes of recovery, library research, conferring with allies. Maybe even time to catch up on the day job, walk the dog, buy groceries.

Whereas in a game, if the first scene is running for your life, the player expects that to be the *tutorial level.* Players will expect that the entire game will use these running mechanics. Jumping to some kind of library research interaction can throw the player off-stride.

It's not an insoluble problem. The usual fallback is to familiar IF actions -- walking, taking, reading --

something so common that it doesn't need to be introduced. You send the player home, and give him easy things to do.

Or, of course, you can switch to dialogue. There isn't *one* conversation model in IF, but if you introduce one of the standard ones in chapter two, nobody's going to kick. And if you *can* introduce a second set of game mechanics, naturally, that's great.

Given all those difficulties, let me reassure you that IF is easy.

My current IF-creation tool is a programming language called Inform 7. It's a heck of a tool. You can start building an interactive world with a few lines of code:

```
The Dining Room is a room. The Kitchen is a room, west of the Dining Room.
The apple is in the Kitchen. The apple is edible.
```

This looks like somebody's notes on a game, but it is working code, and it lets you play this much of a game:

```
Dining Room

> W
Kitchen
You can see an apple here.

> GET APPLE
Taken.

> EAT IT
You eat the apple. Not bad.
```

Very simple and lacking in description, but that's where you start.

Don't get me wrong: Inform 7 is a programming language, and to use it is to program. It has rules of syntax, and if you screw them up you'll get error messages. (Very much like the learning experience of an IF player, which is not exactly a coincidence.) But there are several good manuals, and you can work in tiny steps if you need to.

Try it and see what you can do.

Updated July 11, 2010.

More Rambles

Zarfhome (map)