CO-AUTHORSHIP AND COMMUNITY
An essay on innovating Interactive Fiction

Victor Gijsbers

April 30, 2007

Contents

Chapter 1

Theory and Rationale

1.1 Two necessary innovations

Interactive fiction – which I will understand as parser-based, world-modeling pieces of interactive text – has come a long way since *Adventure* and *Zork*. We have witnessed the rise of a completely new approach to puzzle design, where 'fairness' is the key word. We have seen many and diverse literary experiments, including unreliable narrators (*Spider and Web*), multiple points of view (*Common Ground*), multiple endings (*Galatea*) and multilinear plots (*Losing your Grip*). Most radically, there have been games that no longer challenge the player to reach the ideal ending: sometimes, choosing which ending is the best one is itself the most important act of the game (*Slouching towards Bedlam, Floatpoint, Fate*). At first sight, we can be pretty pleased with the amount of innovation that has gone on.

But such a sense of satisfaction would be premature. In order to reach its full potential as a viable and vital form of literature, interactive fiction will have to be released from the grip of the old text adventures in several major ways. To change the figure, we are still much more captive in the *Zork*-paradigm than we are generally aware of. As I see it, the two – not unrelated – changes that are most necessary are the following:

1. **Doing away with puzzles without sacrificing interactivity.**

   There have been few attempts to write interactive fiction that does not involve puzzles, either as a pacing device or as the central concern of the piece. The very fact that it is usual to speak of interactive fiction works as 'games' testifies to the power that is still held by the old paradigm of a player, armed only with his wits, confronting the devious tricks thought up by an author. This is damaging to the literary potential of interactive fiction in several ways, two of which I wish to describe.

   First, basing a work around puzzles impregnates the entire reading/playing experience with the logic of ends and means. The first thing an experienced reader of interactive fiction thinks whenever she starts up a new game is: what am I supposed to *achieve*? Whenever she encounters an object, a location or a non-player character, she will think: how can I *use* this thing, this location, this person? Each action that the player takes is interpreted first of all as an attempt to get something done with the means given.

But thereby other attitudes and other meanings always come second, or at least have to share the stage with the logic of ends and means. It is very hard in a puzzle-based work, for instance, to make the *moral* meaning of an action involving a non-player character trump the pragmatic meaning. ("Why did you console the little girl?" "In order to get the key to the library, of course. Oh yeah, and I liked helping her as well.") It is very hard for a player who is attempting to solve puzzles to be involved at the same time with telling the most beautiful story, or with exploring the bounds of loyalty, or with any of the other marvelous things that interactive fiction could, in principle, offer. Apart from that, puzzles take up space and time that might have been better spent on something which actually reinforces the central goal of the game.

Second, basing a work around puzzles necessitates a strict conception of the relationship between the writer and the player. In this way, what I call the *Zork*-paradigm has stopped us from considering other ways to conceive of this relationship. I will return to this in the second point below.

Now, people *have* written interactive fiction without puzzles, but often this has been done at the expense of interactivity – *Photopia* is a well-known example. What we need to do is experiment with non-puzzle-based interactivity, and become as good at it as we are at constructing puzzles. *The Baron* was such an experiment, and I hope we can learn something from where it succeeds and where it fails.

2. **Giving the player power over the work.**

All the interactive fiction which has been written, *all* of it, conforms to a conception of the relationship between the author and the reader that was natural within the *Zork*-paradigm, but fails to be natural outside of it. That conception is as follows: the author creates a world, creates rules for manipulating that world, and controls the generation of text by the work based on what happens in the world. The player explores this world, manipulates it in the ways which were allowed by the author, and reads the text the author has created. To use a metaphor, the author is the God of the fictional world, and the player is allowed to live out his life within it. The author has all the power, except as much as he deigns to give away.

This means that if I, the player, dislike how the world works, or have a better idea, I am out of luck. I can only interact with the world in the ways that the author has provided for me. Now this is very natural if the world is a kind of puzzle which I have to solve: allowing the player to change the rules of the puzzle is going to take all of the challenge away. But if the world is not a puzzle, why not let the player tinker with the world itself? Or to say it a different way, why do we allow the player to *play* the piece, but not to *play with* the piece?

Changing this would open up a vast new territory to explore. It would allow completely new ways of interacting with a piece of interactive fiction, new ways which would allow the player to freely use his creativity for the

> first time, and which would allow the player to be a real *co-author* for the first time.[1] The player could change the work, add to it, rewrite it, improve it, or accept the voice of the author out of free choice.

The first of these changes, though difficult, should essentially be possible in our current authoring systems; and if we do not know exactly how to do it, we at least know the directions in which we might look for the solution. The second change, on the other hand, is much more radical; it requires a rethinking of our authoring systems, and in fact a rethinking of the whole process of playing an interactive story. It is this second change which I will discuss in the rest of this essay, focusing on two questions: that of co-authorship and that of community.

1.2 Co-authorship

Literature is complex, and interactive literature no less so. This means that works are active on many different levels: a commentary on social inequality may at the same time be an engrossing love story as well as a murder mystery. But let us say that all fictions, whether interactive or not, have one or more main points of focus. If the work is a good one, the author addresses these points of focus during the work. In the case of a commentary on social inequality, we will probably be shown scenes with privileged people and scenes with non-privileged people and scenes with interaction between them, and the story will suggest that this kind of inequality is morally wrong / pragmatically necessary / inescapable / only reducible by very small increments, or whatever else the writer wants to tell us. In the case of the love story, we might be shown an Aristotelian progress from the initial (psychological, social) situation by necessary steps to the point where they get each other – or not. In the case of the murder mystery, we are given the clues, and the detective convinces us with his superior powers of reasoning that his deductions from the clues must be the right ones.

When we write a piece of interactive fiction which has any of these points of focus, we are faced with a choice. The piece will be interactive, and therefore the player must be able to influence some aspects of the game. These aspects can either be tangential to the points of focus, or they can be directly related to the point of focus. That is, either we write the piece so that the point of focus is addressed *our* way and let the player fool around with harmless stuff that cannot interfere with our purpose; or we let the player address the point of focus herself.

I suggest that the first choice is that which has been made almost exclusively in the current history of interactive fiction. Its dominant implementation has been the "I tell you my story, you solve puzzles to get access to it"-design. There is nothing inherently wrong with this choice, and if you like it, that's fine. But it seem obvious to me that the second choice, the choice where we allow players to actually address the points of focus of the piece, is the one that makes interactive fiction a medium with huge potential. If we choose this second option, we opt for *co-authorship*.

[1] Those with some knowledge of the current indie roleplaying scene will see a clear parallel between what I am am advocating here and the kind of player empowerment that has made games like *The Mountain Witch* and *Polaris* so excitingly different from the traditional GameMaster-centred games.

Let me give you two concrete examples. In a review[2] of *Pytho's Mask* I suggested that the piece might have been improved by allowing the player to choose the fate of the kingdom, instead of asking her to find the solution that was considered optimal by the author. This remark was based on the idea that the points of focus of *Pytho's Mask* are the clash between law and chaos, and (perhaps) the clash between love and responsibility. If these are the points of focus, and the player is allowed co-authorship, the player should be allowed to make the final pronouncements about the right balance between law and chaos, or whether one is allowed to take big risks for the sake of love.

As a second example, consider *So Far*. If the point of focus of this piece is the evocation of a profound meaning through the sustained use of symbolism, then, to achieve co-authorship, the player should be allowed to fool around with the symbols. Of course, that means departing in a rather radical way from traditional interactive fiction models: players would probably have to be able to change the descriptions and behaviour of objects, or even to create new objects with the right symbolic value on the spot. Such actions are, within the current systems, the prerogative of the author.

But if we return for a moment to *Pytho's Mask* – or other piece that are about a protagonist who must make a moral choice (*Floatpoint*, *Slouching towards Bedlam*, *Fate*) – we see that effective co-authorship there might also be unachievable within the limits of classical interactive fiction. Certainly, just being able to make the final choice without the piece saying that you are right or wrong is not good enough. For suppose that the author has made in clear in all the scenes that law (or the way of the Earth, or the Logos, or sacrificing everyone for your child) is the best possible choice, then choosing the opposite at the end of the piece is not real co-authorship but merely defiance. The meaning that emanates from the points of focus pervades all aspects of the piece, and therefore, all aspects may be relevant for addressing it.

Quite in general, if the player is to effectively comment on the vision of the author, she needs to have the same powers as the author. The player must be able to change the objects in the world, add and remove locations, change the rules that govern how the world works, and change the texts generated by the work based on the state of the world. At the same time, the player must be a player, not just a programmer changing someone else's source code. Some thoughts about how this could be implemented are presented in the next chapter.

1.3 Community

First, however, we must rethink the entire approach. For if the player is allowed unlimited power over the work, something seems to be lost that is central to interactive fiction: it's dialogical character. Like a face-to-face roleplaying game, an interactive fiction is fundamentally an exchange between two (or more) people who are constantly offering, accepting and rejecting ideas. In traditional interactive fiction (as in traditional roleplaying games) this relationship is highly asymmetric: the player always offers, the program always accepts or rejects; but

[2] *Puzzle of Masks, Masked*, IF-Review, `http://www.ministryofpeace.com/if-review/reviews/20050916.html`

the relationship is nevertheless present, and central to our experience of interactive fiction.

If the player is given unlimited control over the work, as was suggested in the previous section, this relationship is broken. For now the player both offers ideas, and accepts or rejects them. (The program is only allowed to accept and reject as long as the player does not decide to override it.) Presumably, this will turn playing such an interactive fiction into a rather roundabout and unfulfilling way of writing a story. It might turn out to be as empty an experience as playing a face-to-face roleplaying game alone – and that is quite empty indeed.

Ideally, what we would like is that the player of an interactive fiction is engaged in a game of offering, accepting and rejecting in which she is allowed both to offer, and to accept and reject offers made to her. The radical way to achieve this is to turn the playing of interactive fiction from a private to a public activity. By using the combined creative and judging powers of the community, we can transform the playing of interactive fiction into a game of offering, accepting and rejecting that we play not with the program, but with the other players of the interactive fiction.

What does that mean? Well, the kind of manipulations that I talked about in the last section were manipulations that would change the game world itself. A player might create new objects; change rules; add new dialogue options; and so forth. All these actions change the game itself. This is where the cunning plan comes in: after playing the game, the player presses a 'Release' button, and thereby contributes the changes she made back to the original story file, which is kept on a central sever.

Obviously, that is not cunning enough: one fool could destroy a game. Three ingredients must be added to the mix. First, every released change to the game must be kept. If you change the behaviour of a door and contribute it back to the central server, the server will now contain both the original behaviour of the door and the new behaviour of the door. A player who downloads the game will be presented with the original version as the 'canonical' one, but your alternative rule will be visible to the interested. Instead of thinking up their own changes to the game, they can adopt yours; or they can leave the door alone if they think the original author had the best idea.

Second, changes must be rateable. That is, while playing, people should be able to rate changes proposed by others, and these rating should be contributed back to the central server. Other people who download the game can now see whether the proposed changes are thought of well or thought of poorly; and they can base their decision to adopt them on these ratings. This is all very 'Web 2.0', so at least my proposal comes in time to use the hype!

Third, players must be able to merge sets of proposed changes into consistent alternative version of the game. Players would be able to switch between these sets, instead of having to adopt dozens of changes if they wished to play an alternative vision on the game. So for instance, you could start up a game and be presented with the following choice:

> Which version do you wish to play:
>
> 1. Original. [Maintainer: Emily Short. Rating: +3.]
> 2. Puzzleless. [Maintainer: Victor Gijsbers. Rating: +1.]
> 3. Existential. [Maintainer: Andrew Plotkin. Rating: +4.]

4. Revealed as the failure she is! [Maintainer: Jucok Padle. Rating: -3.]

The player then chooses one of these, based on the description and the rating; and all the objects, rules, text, and so forth which have been lumped together by the maintainer will be turned on. All the other things people have contributed will also be available to the curious player, but they will no longer be the default. (It might also be a good idea to have a more powerful filter system: the ability to hide all the changes proposed by a certain individual, or rated beneath a certain rating.)

The effect of this scheme will be that when you are playing a game, you are confronted with the various visions of previous players. You are asked to comment on their proposals, by rating them. You are invited to think up your own changes, and contribute them back, where they in turn will be judged. Maintainers of the several versions may incorporate your changes into their game. You may decide to create a version yourself and become its maintainer. You and the rest of the community will engage in a dialogical game of proposal, rejection and acceptation.

1.4 Visions

If this could be implemented, reading interactive fiction would become a very different experience. You could still play a game in exactly the same way that you used to; but even then, different versions are offered, and you can explore them and think about the different visions they incorporate. Which one do you prefer? And then, a slight change here, a good idea there: they are made so easily, why not make them? Of course Margareth should have alluded to having a boyfriend in that first conversation; only that gives Charles' moral choice real poignancy. Just change this, here, and – well, why not press that 'Release' button? It turns out that people like your change: it has already received a +3 rating!

Suppose you are the author of a work, returning to it after a year. Many people have played it; a good number has been inspired to release some change. Two of them have even gone as far as to start a new version. You love one of them, but hate the other – it completely fails to take into account the thematic content you so carefully placed! Let's rate it down. Oh, but this change is cute! Let's incorporate it into the official version...

In short, the admittedly radical changes I propose would completely alter the nature, not only of reading interactive fiction, but in fact of the interactive work itself. It used to be a model world frozen into binary code, playable again and again in the exact same way by ever new readers. But now – now it is something organic, something alive: something that changes as it is played, and that can take off in directions never envisaged by its author.

The voyage has only begun. The seed have only been planted.

Chapter 2

Implementation

2.1 Introduction

Everything is this chapter is offered as a tentative suggestion. I have a fairly good idea of the direction you'd have to take interactive fiction in if you wish to significantly increase the amount of creative involvement of the reader. But I have only some vague hunches about the best way to implement this. The sections below describe one form that such an implementation might take; but it is just one form, and it may not be the most practical one.

2.2 Co-authorship

I envision a graphical user interface that looks a lot like the current Inform 7 IDE. On the left side, there is a game. It looks just like the games we are used to, and it is interacted with in much the same way. The only difference is that a powerful undo/redo feature (possibly allowing us to fork our game into different possible play-throughs, much like the Inform 7 skein) is implemented. After all, we are going to be changing the very way the world works, and we'll often want to see the effect that this has on the actions we were trying.

On the right side, in a smaller window, we see the relevant rules. In its most basic state, it will only show the most interesting rule that was triggered by the last action the player took. So if the player typed "enter swimming pool" and got the response "While there are sharks in the pool? Are you mad?", the right side window would show:

> **Rule for entering the pool:** if the sharks are in the pool, say 'While there are sharks in the pool? Are you mad?' instead.

Sometimes, we need to show states of affairs as well as rules. If the player typed "open door", and got the message "The red door is locked.", the right side window might show:

> **Rule for opening a door:** If the door is locked, say 'The door is locked' instead.
>
> **State of the red door:** locked.

From a technical point of view, this may not be easy, but it is certainly not impossible. Presumably, if we use a rule-based game architecture, the game 'knows' which rule was used to generate the output. It should show this rule. Sometimes, a general rule is used to generate the output; in that case, the game should also show the state of affairs that made this rule applicable here. This is what happens in the example of the red door: the 'rule for opening a door' is used, because the red door is locked. All the game needs to do is keep track of its internal logic and make it visible to the player.

As long as the player is happy with what is happening, she can simply ignore the right side window. (In the GUI, she can probably hide it with a single key combination.) But when she wants to start changing the world, all she needs to do is look at the right side window and locate the rule or state of affairs that she wishes to change.

How does such a change work? In the case of the red door, the player might click on the word 'locked' in the state of the red door, and be presented with a small pull-down menu containing 'locked' and 'unlocked'. She could choose 'unlocked', and now. . . what? There are two possibilities: either the door becomes unlocked now, or the door now has been unlocked from the start of the game. In the first case, the new functionality would be little more than a very powerful cheat command, a bit like 'purloin' and other debugging verbs. This is uninteresting, for implementing it would destroy the idea of playing out a story in a consistent, rule-based world – but that is the very core of interactive fiction. So we have to take the second possibility: changing the world means changing how the world worked from the beginning onward.

This is where the real technical difficulties of the proposal are located. For depending on the history of our play session, the correct response of the program to changing the state of the red door might be any of the following examples:

- Done.
- This would have triggered the "All doors are unlocked" rule on turn 132. Would you like to (a) edit that rule, (b) continue playing from turn 132, or (c) attempt to continue from the current turn (214)?
- This would have allowed you to open the red door on turn 83 (in the Zen Room). Would you like to (a) remove that action from the session history, (b) leave the action in and continue playing from turn 83, or (c) leave the action in and attempt to continue from the current turn (214)?
- (Free editing mode: ignoring session inconsistencies.)

I do not know whether such real-time evaluation of changes in the session history is technically feasible. I certainly do not know – and I don't think we can know it without experimenting with it – whether one could build an intuitive user interface incorporating this kind of on-the-spot manipulation of the world model. All I am saying is that it might be worth investigating.

Given the speculative nature of this design, going into too much detail is not a good idea.[1] But I would like to mention two additional things, one having to do with the implementation of the right side window, and one having to do with the underlying language.

[1] How do we create new objects, for instance?

Let us look at the example with the red door again. As a player, it might be the case that I do not want to have the door unlocked from the beginning (it is entirely appropriate that Dr. Frank keeps his monster-creating lab locked up), but that I want to change the rules that govern unlocking it. So there must be some icon I can click on (perhaps next to the 'locked' attribute) in order to see the rules and states that govern unlocking the red door. Perhaps:

> **Rule for unlocking a door:** If the player has the relevant key, success.
>
> **State of the bronze key:** relevant key for the red door.

And these must in turn be editable. Making this mechanism of increasing the depth of our code survey useful will presumably also prove to be quite a challenge.

Then, the underlying programming language. My examples have been influenced by Inform 7. It seems to me that a natural language, rule-based approach is perfect for the project as described here, and Inform 7 or a variant would be a good choice. Obviously, though, the compilation process currently used is absolutely unsuitable. The virtual machine would have to accept the Inform 7 code itself, not a compiled binary or semi-compiled procedure-based code. It must be able to keep track of all the rules, and thus must be able to see these intact, in their full, pre-compiled glory.[2]

Again, I do not know whether this project is feasible. But if it turns out to be, a new kind of player-world interaction will have become possible.

2.3 Community

Much of the way a community structure should be implemented will be clear from what I said in the theory chapter and from the kind of GUI I have been describing above. On the network-level, we need a central server which accepts lists of proposed changes and adds them (as proposed changes) to the story file.

Then, the player must be able to see all the proposed changes to the rules. The right hand window should therefore not only show the active rules, but also all the proposed alternatives that are compatible with the selected filters. (Filters might be used to show only rules in certain version, only rules by certain people, or only rules with a high enough rating.) Rules combined in a single version might be give the same colour throughout the game, for ease of use.

The GUI should also allow us to combine all the active rules into a new version, and release it to the archive with ourself as the maintainer. The archive should not allow us to change version which someone else maintains.

But here I stop, remaining vague about practical matters. That is the prerogative of the revolutionary.

2.4 The moderate way

Being radical is always more fun than being moderate, at least as long as the possibilities of actual revolution are remote and flights of fancy are relatively

[2] This is compatible with *some* kinds of compilation, but not – I think – the kind that goes on in either the Inform 7 - Inform 6 compilation or the Inform 6 - Z-code compilation.

safe. But being moderate is sometimes more productive, so I will here present a way of implementing, in the authoring systems we have today, a weaker version of the ideas developed in this chapter. The easiest way to do this might be through a transcript.[3]

> > PEEK
>
> You peek through the keyhole of the large chest. The room outside is dimly lit, and it is hard to make out the features of the two people kissing in the corner. You can easily recognise the woman as Susanna (who couldn't recognise his own wife?); but the man. . .
>
> > X MAN
>
> As the clouds part outside, a ray of moonlight illuminates the face of the adulterous swine. It is –
>
> [What about the effeminate Cherubino?]
>
> > NO
>
> [Ok; maybe the count Almaviva himself?]
>
> > YES
>
> As the clouds part outside, a ray of moonlight illuminates the face of the adulterous swine. It is Almaviva, the lecherous count! The lust evident on his face makes you gasp audibly.

What is distinctive about this transcript is that the piece offers the player several potential ways the game world might be. These are presumably taken from a finite list, so the player has only limited control, and only where the game author has decided to give her that control. This would still constitute a significant break with current interactive fiction, where the result of an action is always determined by the author, and where the game-world is always conceived of as already being determined at the start of the game. (Playing interactive fiction is currently conceived of as exploration, not as creation.) It would be a step in the direction of co-authorship, because it allows the player to make some meta-game decisions. But, of course, it would be nowhere as revolutionary as the true co-authorship/community approach which I have been dreaming about in the rest of this document.

[3] See also the game *Figaro* released with this document.