

All Hope Abandon: Biblical Text and Interactive Fiction

Eric Eve <eric_dot_eve_at_harris-manchester_dot_oxford_dot_ac_dot_uk>, Harris Manchester College, University of Oxford

Abstract

Among the alternative kinds of narrative opened up by computer technology, one of the earliest is interactive fiction (and specifically the “text adventure” or “adventure game”), which first came into being in the 1970s. Text-based interactive fiction enjoyed a brief period of commercial success in the 1980s until it was overtaken by advances in computer graphics, but it continues to be read and written by enthusiastic amateurs. Although interactive fiction clearly has roots in computer gaming, it also has potential as a new form of literature.

Since interactive fiction may be a medium unfamiliar to some readers, this article will start by defining it, explaining some of its conventions, and outlining its origins. It will then describe how one recent piece of interactive fiction, “All Hope Abandon”, explores aspects of Biblical Studies through the medium of IF, and will end by suggesting a number of fruitful links between interactive fiction and biblical text. In the course of the article several examples from “All Hope Abandon” will be given, and instructions on how to install and play the game will be provided at the end.

What is Interactive Fiction?

The term “interactive fiction” (or IF) can have more than one meaning in the context of computer-based storytelling. In this article, however, the term will be used in the narrow sense of a turn-based program driven by textual input from the player, responding with output that is principally or wholly textual, and involving a parser and a world model (two terms which will be explained shortly). For the sake of brevity, the term “game” will be used to denote a work of interactive fiction, though not all such works are equally ludic, and the reader should not rush to equate IF “games” with the video-games more commonly associated with contemporary recreational computing. 1

Such a work is *interactive* fiction since it is up to the reader (or player) to drive the story forward, usually by directing the actions of the main protagonist (known in IF-parlance as the player character, or PC). A work of IF typically opens with a brief textual description of the PC’s situation, perhaps also giving some indication of the tasks the PC has to perform, or the challenges he or she has to overcome, though these may not become apparent until the game gets under way. In order to progress through the game the PC must explore his or her environment, discover how things work, interact with any other characters in the game (known in IF-parlance as non-player characters or NPCs), and often solve a number of puzzles. The role of the player is to direct the PC in these tasks by entering commands at the keyboard.^[1] 2

Part of the appeal of IF is the illusion that one has complete freedom to type anything one likes at the game’s command prompt. It is an illusion because, although one can indeed type anything one likes, the game will in fact understand only a very restricted subset of English (or whatever language the game was written in). The commands typically understood by a modern IF parser are largely restricted to the following three formats: either a simple imperative verb (such as LOOK, which generates a description of the PC’s current location), or an imperative verb plus a direct object (such as TAKE THE BLUE BALL, which would cause the PC to pick up the blue ball, assuming the action is possible), or an imperative verb plus a direct object plus a preposition plus a second or indirect object (such as PUT THE BLUE BALL IN THE BROWN BOX). This basic syntax can be slightly extended to allow conversing with or giving orders to NPCs; so, for example, FRED, PUT THE BLUE BALL IN THE BROWN BOX would be construed as a command to the PC to tell 3

the NPC called Fred to put the blue ball in the brown box.

The role of the *parser* in interactive fiction is to interpret such commands so that the game can either respond to them appropriately, or tell the player why it can't. A modern IF parser is reasonably sophisticated at this task. If there is both a red ball and a blue ball in the game, a decent parser will nevertheless know what to do with a seemingly ambiguous command like TAKE BALL. If only one of the balls is in sight, it will assume that this is the one the player meant. If both balls are in sight but the PC is already holding one of them, the parser will assume the player meant to refer to the other. Only if the parser is unable to resolve the ambiguity in any such way will it ask the player to do so (typically by asking a question like "Which ball do you mean: the blue ball or the red ball?").

4

A more sophisticated parser will also deal automatically with certain implicit actions needed to carry the player's explicit command. For example if the player types N (the usual abbreviation for GO NORTH) when going north would take the PC through a closed door, a good parser will assume that the player wants the PC to open the door first, and will first attempt opening the door as an implicit action, only carrying out the main action if the required implicit action succeeds (it may not, since the door may be locked or jammed or inaccessible for some reason). Or if the PC needs to be holding the blue ball before he can put it in the brown box, then a courteous parser will carrying out an implicit TAKE BLUE BALL command when the player types PUT BLUE BALL IN BROWN BOX, rather than complaining that the PC needs to be holding the blue ball first. Unfortunately, not all parsers achieve this level of user-friendliness.

5

A modern IF parser works first by identifying what *action* the player intends, and then by resolving the nouns involved in the action. For example, the TADS 3 parser (written in the TADS 3 language) first matches the player's input against the grammatical patterns (known as "grammar productions") contained in the TADS 3 library and the game author's own code: thus if the player types PUT BALL IN BOX the parser first determines that this matches the pattern `PUT <noun phrase> IN <noun phrase>` and so decides that the current action is of type *PutInAction*.^[2] The parser then constructs lists of possible objects that could match the two noun-phrase slots. Possible matches are restricted to those objects in the game that (a) possess vocabulary corresponding to what the player typed and (b) are currently in scope (roughly speaking, this means that they must be currently visible to the player character). These lists are then sorted in order of logicity (i.e. fitness for their role in the proposed action). For example, for a PUT IN action to succeed its direct object (occupied by the first noun slot) must be something that the player character can move, so that a small rubber ball will be a more "logical" target for such a command than a large stone ball fixed to the top of an ornamental balustrade. Likewise, for a PUT IN action to succeed, its indirect object (occupying the second noun slot) must be a container of some sort, so that, for example, a large cardboard box is a more "logical" target for a PUT IN command than a box girder. If this object resolution phase results in one and only one most logical possibility for each of the noun slots, then the parser will settle on these objects; otherwise it will either request further clarification from the player (to select from a number of possibilities) or complain that there is nothing in scope that matches the command entered.^[3]

6

The second defining characteristic of interactive fiction is the *world model*. Most IF is set in a simulated environment representing physical space and the objects it contains (though there may be the occasional exception). Broadly speaking, this corresponds to the *setting* of a conventional narrative. Different IF systems and different games written with those systems will implement this simulated environment to different depths, but there are certain features that most if not all works of IF share. Firstly, the physical world is generally modelled as a series of discrete locations known as *rooms*.^[4] The totality of rooms in a given work of IF is often referred to as the *map*.^[5] Such rooms could correspond to rooms in a building, but they need not and frequently do not: a "room" in the IF sense is just as likely to be a section of path through a forest, or one corner of a large town square, or a section of meadow-bank overlooking a lake. Conceptually, a room is that segment of physical space that is immediately accessible to the player character. The notional size of a room can vary greatly, but is assumed that, with certain qualifications, the PC can see and hear everything in the room in which he or she is located unless it is purposely hidden; and can taste, touch or smell everything in the room unless it is deliberately placed out of reach. The standard IF convention is that, with one exception to be dealt with below, the movement of the characters and objects within a room is not normally modelled. It may be that I would in fact have to cross a large study to take a book off the shelf, but this detail is generally ignored in IF; it is assumed that the PC (and any NPCs) will move around within a room as necessary without being explicitly

7

commanded to, and such internal movements will not normally be reported, unless the game author explicitly wishes to mention them for artistic effect. Thus few games bother to implement commands such as WALK OVER TO THE TABLE or APPROACH THE BOOKCASE, since they would generally be superfluous.^[6]

It is perfectly possible for a complete work of IF to take place within a single room, and some do, but it is more normal for a game to contain multiple rooms and for the PC (and possibly the NPCs as well) to move around between them. By far the most normal convention is for movement commands to be given in terms of the eight principal compass directions together with UP, DOWN, IN and OUT. This should be understood as a convention of the medium, not as a pretense that the PC in fact carries an infallible compass in his or her head; experience has shown that in practice this is generally the most convenient way for a player to communicate movement intentions to the parser, not least because movement commands can be abbreviated to N, SW and the like. But other styles of movement command such as GO THROUGH RED DOOR and ENTER THE BIG HOUSE are also common. Less common, but also possible are commands to go to a specific location, such as GO TO THE KITCHEN.

In the normal IF world model, rooms are populated with *objects*. These may be portable objects that the PC can pick up and carry around, or non-portable objects such as trees and houses and heavy furniture that remain fixed in place. Some objects may play an integral part in the game, while others may only be scenery. Standard IF world models also implement a *containment hierarchy* for objects. Everything currently in the game must be contained in a room, but within rooms some objects may contain other objects. The most commonly implemented containment relations are *inside* (for containers like boxes, desk drawers, pockets and sacks) and *on top of* (for supporters like desks, tables and trays). Some objects, such as chairs, beds and platforms, may also contain the PC and other characters; these constitute the exception to the rule that movement within rooms is not normally modelled, since movement in and out of containers like beds and chairs normally is dealt with explicitly.

Although most objects in a game are normally inanimate (as are most objects we commonly interact with in the real world), some may be designed to represent animate objects such as cats and cows or even people (or quasi-animate objects such as robots and talking computers). Clearly such objects – especially people – are a good deal more complicated to implement well.

The earliest game of this type was principally a simulation of the Bedquilt/Flint Mammoth cave network created by Will Crowther, originally as an amusement for his daughters ([Montfort 2003, 85–93]; [Nelson 2001, 342–45]; also described elsewhere in this issue). It was subsequently expanded by Don Woods with a number of additional puzzles and treasures, and has continued to be expanded in many versions since, being readily available, generally under the name “Adventure” but sometimes called “Colossal Cave”, “Colossal Adventure” or a number of other names.^[7] This may be one reason why works of IF are also known as “text adventures” (even when the plot may involve little that resembles adventuring in any conventional sense).

Adventure’s parser was originally written in FORTRAN, and was primitive by modern standards: it was a “two-word” parser since the most complex commands it could understand were in the form of an imperative verb and a single noun. The game had little narrative plot, being mainly a treasure hunt, but the fact that, unlike many of its successors and imitators, it was based on a real cave system by an author who knew it well gives it a certain enduring charm.

The most significant successor and imitator of “Adventure” was a game first called “Dungeon” and then “Zork”, originally written on a university mainframe (at MIT, beginning in May 1977) [Montfort 2003, 97]. Although “Zork” was arguably even more of a puzzle-driven treasure-hunt set in a less coherent and authentic world than “Adventure”, it did achieve a more sophisticated parser capable of two-object commands (such as PUT BALL IN BUCKET or HIT TROLL WITH SWORD). Of even more significance is that in June 1979 the authors of the original Zork went on to found a company called Infocom, which proved highly successful in writing and marketing text adventures, or what it came to call interactive fiction [Montfort 2003, 125].^[8] Not only did Infocom establish the conventions that IF still largely follows today, it also showed that the medium was capable of far more than cave-crawling treasure-hunts.

Infocom (and other companies like it) made a great success of interactive fiction in the 1980s, but as home computers

became more powerful and ever more capable of displaying good graphics, text-based games fell out of favour with computer gamers who could be offered much more visually appealing graphical video-games instead. But while interactive fiction largely fell out of sight, it did not die, but continued to develop in the hands of enthusiastic amateurs.^[9]

It is possible to write a work of interactive fiction from scratch in a general-purpose programming language such as Basic, C#, or Pascal, but it is extremely inadvisable to do so unless your primary aim is to practice programming in one of those languages. Implementing a robust parser and world model is a decidedly non-trivial task, and any case IF is best served by a domain-specific language. Systems for authoring IF had been available for some time, but amateur IF received a considerable boost in the early 1990s from the release first of Mike Roberts's TADS (Text Adventure Design System), and then Inform, created by Oxford mathematician Graham Nelson.

15

These systems comprise special-purpose programming languages (TADS, for example, is quite similar to C), together with a library (usually written in the same language) providing a standard parser and world model (which can then be extended or modified to a greater or lesser extent in the game author's own code). They may also contain additional tools such as an integrated development environment or a debugger. The compilers for these languages typically output, not a native executable, but a byte-code file designed for running in a virtual machine on a program called an "interpreter" (similar to the way a Java compiler produces byte-code to run on a Java Virtual Machine).^[10] This allows games written in TADS, Inform and similar IF languages to run on any system for which the appropriate interpreter has been written.

16

Inform had two great advantages when it was first released: first, it was free, at a time when Mike Roberts still charged for TADS (though it has now been available free for many years),^[11] and second, it compiled to the same format (known as the Z-machine) as the much-loved Infocom games.^[12] With these tools it became possible for amateur IF enthusiasts to write games as good as or even better than those produced by Infocom and its competitors (which is not to say that the systems have not also been used to produce a glut of games that are far worse). 2006 saw the release of the latest version of both of these authoring systems: Inform 7 (still in beta) and TADS 3 (finally out of beta after several years).^[13] Inform 7 is an experiment in a natural-language style of programming that is proving attractive to many potential authors who would not feel so comfortable with conventional coding.^[14] TADS 3 is more like a conventional programming language (much of its syntax is identical to C),^[15] but incorporates the most sophisticated parser and world model yet seen in IF, as well as providing the best toolset to date for the implementation of NPCs. Between them, Inform 7 and TADS 3 provide the IF author with a choice of state-of-the-art development tools that take radically different approaches.^[16]

17

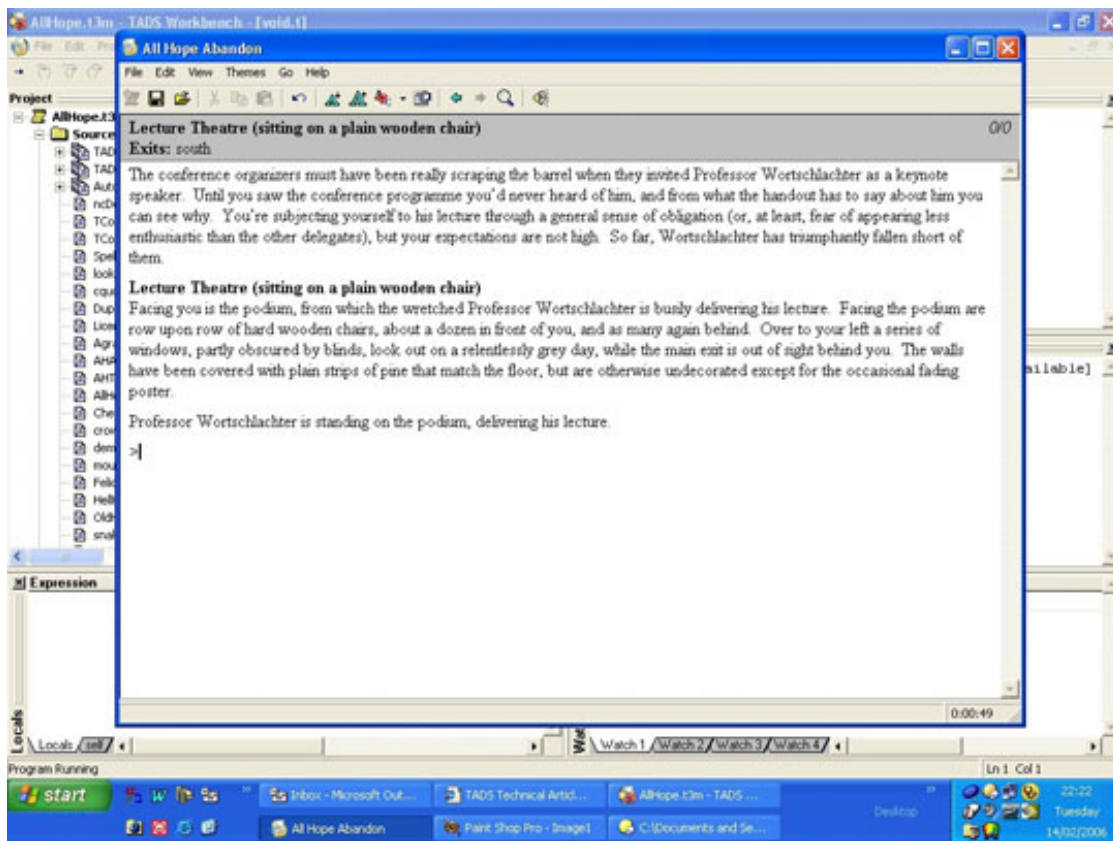


Figure 1. The opening scene of “All Hope Abandon” running on the HTML TADS interpreter for Windows (with TADS 3 Workbench for Windows, the development environment, open behind it). Note how the status line (the grey area towards the top of the screen) indicates the Player Character’s current location and the currently available exits. The 0/0 at the right hand side of the status line shows the score and turn count (both zero since nothing has happened yet). The main window (the white area) shows the introductory text and the description of the opening location.

Amateurs writing for other amateurs have perhaps felt freer to innovate and experiment, and one trend has been an increasing appreciation and ambition for story-driven as opposed to puzzle-driven IF. That is not to say that well-made puzzle games do not continue to be appreciated – they clearly are^[17] – but there has also been increasing interest in the narrative possibilities of IF. There have also been developments in the generally accepted conventions of the medium. Back in the 1980s, and even the early 1990s, IF games often contained huge illogical mazes and puzzles that were frequently arbitrary and unfair, as well as all too often allowing a player to continue unknowingly with a game long after it had been put into an un-winnable state. Such things are now usually frowned upon. The modern ideals are that puzzles should be properly and fairly clued, and fully integrated into the narrative structure of the game (although this can often be hard to achieve), that the player should be given fair warning if the game gets into an un-winnable state, and that no game should contain a maze without a very good reason for it indeed (and even then it had better be a very original kind of maze). Again, modern standards are much more demanding on depth of implementation. If a room description mentions striped wallpaper then the parser had better recognize striped wallpaper if the player tries to refer to it. It may be acceptable for the parser to respond with “The striped wallpaper is not important” if the wallpaper is indeed pure scenery, but it looks very clumsy if, having told the player that the room is decorated with striped wallpaper, the game responds with “You see no such thing” when the player tries to examine it (as would all too often be the case in much “old-school” IF and all too much poorly-implemented modern IF). It is better still if a description of the wallpaper is provided in response to “X WALLPAPER” (i.e. EXAMINE WALLPAPER), even if the “not important” message is used in response to any other command involving the wallpaper; this kind of detail can make the difference between immersion in the story world and the feeling that, after all, you are merely interacting with a computer program.^[18]

These modern expectations raise a further pair of issues, namely the roles of room descriptions and puzzles. In conventional (or “static”) fiction a description of a location may be used, *inter alia*, to paint a picture in the eye of the

reader, or to evoke a mood, or to provide information needed to understand the plot. All these functions apply to room descriptions in interactive fiction, but with additional constraints. First, whereas the author of a novel or short story has considerable latitude in choosing the length of any description (provided it is not so long as to kill the narrative momentum), an IF author has to be economical; the text of a room description (or any other kind of output) will appear on a scrolling screen between one command prompt and the next, and for that reason will appear too long if it amounts to more than half a dozen lines of text or so. Second, there are certain pieces of information that a room description must convey to the player, not least the direction of the available exits from the current location. Third, the description ideally needs to be so arranged as to draw the player's attention to items that merit further investigation (say, the large desk in the corner, in which resides an important letter from the PC's husband's lover) rather than to items that are purely decorative (such as an ornamental vase on the mantelpiece). To achieve all these goals while writing good prose is not always easy; one technique is to minimize the kind of extraneous detail that might typically be used for scene-setting in static fiction.^[19]

It is possible to write IF that is virtually puzzleless, but puzzles can play an important part even in IF that aims to be more literary than ludic.^[20] What exactly constitutes a puzzle is a little hard to define; in essence it is a task that the player has to perform or an obstacle he or she has to overcome where the necessary sequence of actions is not immediately obvious but requires some thought or experimentation to arrive at; but what may seem immediately obvious to one player may seem more obscure to another. However defined, puzzles have a number of important uses. First, they may simply be integral to the plot, since the game may describe a situation in which the protagonist would naturally have to overcome obstacles or achieve difficult tasks of this type (for example, if the protagonist is a detective investigating a murder). Second, puzzles are one of the main devices in IF for creating a sense of tension and opposition, without which any narrative is liable to feel dull and slack. Third, they can help pace and direct the plot. In static fiction, authors have complete control over the course of events and can accordingly structure their narrative to suit their plot. Authors of IF surrender much of this control to the player, who controls the actions of the protagonist and thus the course of events. Putting strategically placed obstacles in the path of the protagonist is a means of regaining a measure of authorial control; for example certain parts of the map can be closed off until certain tasks have been performed (paradigmatically, the PC cannot pass through the locked door until he or she has discovered the key), or the accomplishment of certain tasks can trigger certain key events. Third, placing obstacles in the path of the PC can deepen the players' involvement in the narrative by forcing them to think about the PC's situation and come up with a solution. Players are made to identify more closely with the actions of the protagonist if they have directed those actions themselves, and can feel a considerable sense of satisfaction at being able to overcome a difficult obstacle. But puzzles can also misfire in a number of ways. If they are too hard they can bring the narrative to a standstill as the player thrashes about in increasing frustration trying to find a solution. If they seem arbitrary, insufficiently well clued, or otherwise unfair they can simply provoke annoyance. Conversely, if they seem too easy there is little satisfaction at solving them, and a sense of tedium may set in if the player is forced to go through a series of routine actions (find next key, unlock next door).^[21]

Puzzles are not the only way to achieve some of these effects; for example, it would be possible to present the player with a series of choices that were not so much puzzles as dilemmas, or to write a puzzleless simulation,^[22] or a game that works in some other way, but this can be difficult to bring off. If a game presents players with more or less the same linear narrative without offering any significant challenges, it can all too easily create the feeling that one may as well be pressing the space bar between reading chunks of text, and that the story might better have been presented as conventional static fiction.

What drives the narrative in most static fiction is the interaction between the (normally human) characters, but this is the hardest thing to do well in interactive fiction, particularly when it comes to representing the subtleties of human relationships. Many NPCs in IF turn out to be little more than elaborate puzzles or talking robots. Examples of the first category might be someone I have to get out of the way so I can see what's in his desk or the bouncer who won't let me past the door. The second category is part of the wider reason it is so difficult to produce lifelike NPCs in IF, namely the difficulty of producing realistic dialogue in the medium.

Authors of static fiction have complete control over what their characters say, with all the nuances that may imply for the relationships between them. Authors of IF surrender control of speech as well as action to their players, but whereas a good modern IF parser can cope very well with interpreting commands directed at physical action, the same is not so true of speech commands. The reason is quite straightforward: whereas there is only a fairly limited range of physical actions a PC might sensibly attempt at any one time, there is no such limit to what he or she might sensibly say, but the technology to make sense of any textual input and provide a good response simply does not exist, and would in any case be a move away from interactive fiction to artificial intelligence (at least as things are now). Attempting fake AI in NPC responses (for example, by getting the parser to look out for certain keywords) is nearly always a bad idea, since it is almost bound to be given away by incongruous responses sooner or later. The only practical solution with existing IF parser technology is to put a drastic limit on the range and syntax of conversational commands, and to program a finite range of suitable NPC responses, together with a range of totally non-committal “default” responses for topics not otherwise catered for.

23

A number of approaches are possible within these limitations.^[23] The simplest is to restrict conversational commands to TALK TO X, and then respond with canned conversation; this can work well enough in certain contexts, and does retain maximum authorial control over the course of conversations, but greatly restricts players’ freedom of choice with concomitant loss of interactivity. Another option is to have TALK TO X respond with a menu of things the PC can say (possibly linked with a second command to switch topic); again this can work reasonably well, but is not without its disadvantages, the most prominent of which are first that switching to a restricted-choice menu interface may not sit well with the open-ended command line interface used for every other kind of player input, and second that when players are presented with a menu interface for conversations the temptation is to try to navigate them like a kind of conversational maze, which risks breaking immersion in the narrative.

24

Perhaps the most commonly used conversation system, and that most frequently provided as standard in many IF authoring system, is the ASK/TELL system. This allows players to type commands of the form ASK BOB ABOUT THE LETTER or TELL JILL ABOUT JANE, where what follows “ABOUT” is a topic to which the NPC may have a programmed response (the ASK/TELL system usually also allows SHOW SOMETHING TO SOMEONE and GIVE SOMETHING TO SOMEONE and often ASK SOMEONE FOR SOMETHING). The advantages of this system is that it allows a form of user input congruent with the user interface used for all other commands, preserves the illusion of player freedom, but makes it reasonably possible for the author to provide suitable responses, at least to the most significant topics (the others can then be fielded with a non-committal response such as “Let’s talk about that some other time” or “Bob pretends not to hear you”). The disadvantages are first that the NPC can all too easily end up resembling a talking robot more than a believable human being (for example if the command ASK BOB ABOUT LETTER repeatedly provokes the response, “ ‘It’s from Jill’; he tells you, ‘she’s left me to run off with Jack!’ ”), and second that the system allows little or no control over what precisely is asked or told: if I issue the command ASK BOB ABOUT LETTER do I want to ask Bob who the letter is from, or what it says, or when he received it? If I type the command TELL JILL ABOUT JANE do I want to tell Jill that Jane is pregnant, or that I’m in love with her, or that she’s emigrated to Australia? In particular, the only dimension of human conversation it copes with is the exchange of information; it is thus quite incapable of modelling, say, a developing relationship (whether of affection or enmity) in any realistic way.

25

One of the great achievements of TADS 3 is that it extends the ASK/TELL system in a number of ways that greatly mitigate these limitations. For example, it provides tools for varying NPCs’ responses in accordance with what has been said before, and for threading conversations in a more realistic manner, allowing a greater range of possible responses that just ASK or TELL at critical points in the dialogue. It also encourages authors to write both halves of a conversational exchange, which generally makes the dialogue read much better.^[24] It is not that it is impossible to do these things in other IF languages, but rather that TADS 3 provides by far the best set of tools as standard (and also makes it reasonably easy for more ambitious authors to extend them as desired). This is one reason why TADS 3 was chosen for “All Hope Abandon” (as well as a number of other games I have written).

26

The difficulty of implementing life-like NPCs is perhaps one of the major differences between IF and conventional static fiction; it can certainly restrict the kind of story that can be told well in the medium, since it is far harder to produce a

27

game involving the emotional interaction of characters (such as a love story) than one focusing mainly on the gathering and exchange of information (a detective story or spy story, for example). That is not to say that IF necessarily lacks emotional impact, but this must often be achieved by other means, and it would be fair to say that the implementation of convincing NPCs is one area in which there is currently vast scope for experimentation. But the more fundamental difference between static and interactive fiction is that the latter is interactive, and requires a particular kind of input from the reader to advance the story. This is not to say that the reader of static fiction is wholly passive, as reader-response criticism in particular emphasizes, but IF does raise reader participation to a greater level; it also both operates under different constraints and offers different opportunities from static fiction.^[25]

Implementing the Player Character also presents its own challenges. In early works such as “Adventure” and “Zork” this was not an issue, since the PC was usually anonymous with no characteristics beyond the ability to act as the player’s eyes and hands in the game. Once authors became more interested in the narrative possibilities of IF this changed, however; PCs needed to become more specific, of a determinate gender, with particular characteristics and a history of their own, more like the protagonists of conventional figure than the convenient puppets of the early games. This then raises the possibility that the PC’s motivations might clash with those of the player who is supposed to be controlling his or her actions. In such a case the game author has to choose between allowing the action at the cost of the integrity of the character, or disallowing the action at the cost of restricting player freedom. This potential conflict is normally resolved by means of some kind of compromise: typically a PC might be motivationally restricted from committing murder or suicide while being far less scrupulous about stealing (since players of IF quickly develop the kleptomaniac strategy of taking anything that isn’t nailed down on the assumption that it will probably come in useful later in the game). It is in any case impossible to satisfy all players, since some will always complain if the PC won’t do their bidding (“but Bob is such an irritating character that I *want* to punch him on the jaw”, while others will complain if they can make their PC act too obviously out of character (“my PC is meant to be a genteel elderly spinster who would *never* punch anyone on the jaw, however irritating she found him!”); nevertheless, some kind of reasonable compromise seems to satisfy most players most of the time.

28

Less easy to negotiate is the discrepancy that can arise between what the player knows and what the player character supposedly knows. This can work both ways. Someone playing a game for the second or subsequent time almost certainly starts out knowing more than the PC knows at the start of the game, and this can create problems when the player wants to put that additional knowledge to use. Before the PC has been introduced to Mrs. Prancealot he logically cannot know that she *is* Mrs. Prancealot, but the player may try to refer to her as Mrs. Prancealot before the introduction takes place and become irritated either because the parser allows the use of the name that the PC cannot yet know or because it refuses to recognize the name that the player knows perfectly well. Or, to take another example, if a key has been dropped under a sofa, should second time players be forced to issue a LOOK UNDER SOFA command in order to be allowed to refer to a key they already know to be there? Some players may find such a restriction irksome, while others may feel it unreasonable that the PC can pick up a key from under the sofa before he has any good reason to suspect its presence there.

29

Conversely, the PC may well know things the player (especially the first time player) does not. Such knowledge may be relatively trivial, such as the PC’s knowledge of his or her own locality, which the player has to discover by exploration (this can usually be handled by writing room descriptions that emphasize the PC’s familiarity with the locations visited). But it may be something rather more complex, such as the specialist knowledge the PC ought to possess in his or her persona as safe-cracker, starship captain, or (as in the example about to be discussed) scholar. This can be particularly tricky if it results in the player fumbling to perform tasks the PC ought to be able to manage in his or her sleep, or if the PC’s specialist knowledge is particularly relevant to understanding the puzzles and other plot elements, as turned out to be the case in “All Hope Abandon”.^[26]

30

All Hope Abandon

The particular example of IF to be discussed here is one of my own works, “All Hope Abandon”. The reason this game may be of interest to readers of *Digital Humanities Quarterly* is that it combines relatively a new digital medium (interactive fiction) with a very traditional academic discipline (Biblical Studies). I shall start by explaining what the game

31

is about, and then go on to discuss some of the design goals and implementation challenges.

The game opens with the protagonist, Dr. William Fisher, listening to a particularly dire lecture at a New Testament conference. Since our discussion of interactive fiction has been fairly abstract up to now, it may be helpful to show a sample transcript of the first few turns by way of illustration (lines beginning > show player commands):

The conference organizers must have been really scraping the barrel when they invited Professor Wortschlachter as a keynote speaker. Until you saw the conference programme you'd never heard of him, and from what the handout has to say about him you can see why. You're subjecting yourself to his lecture through a general sense of obligation (or, at least, fear of appearing less enthusiastic than the other delegates), but your expectations are not high. So far, Wortschlachter has triumphantly fallen short of them.

Lecture Theatre (sitting on a plain wooden chair)

Facing you is the podium, from which the wretched Professor Wortschlachter is busily delivering his lecture. Facing the podium are row upon row of hard wooden chairs, about a dozen in front of you, and as many again behind. Over to your left a series of windows, partly obscured by blinds, look out on a relentlessly grey day, while the main exit is out of sight behind you. The walls have been covered with plain strips of pine that match the floor, but are otherwise undecorated except for the occasional fading poster.

Professor Wortschlachter is standing on the podium, delivering his lecture.

>X ME

You're in your early thirties, but you firmly believe that you look younger. In your more honest moments, however, you have to admit that you're not exactly as fit as you might be. You are dressed quite formally for the occasion, in a dark jacket and tie.

Professor Wortschlachter peers down at his notes, grasps the top of the lectern, and announces, "As you all know, our pericope, Mark 16.1-8 is one that *The Acts of Jesus* prints in heavy black type, indicating that the Jesus Seminar are quite clear that the account is utterly unhistorical. We can assume that a first-century Jesus Seminar would have voted the same way, at once recognizing the midrashic intertexture of this text and not for a moment is taking it for a putative historical report."

You wonder briefly at this astonishing leap of logic, but Wortschlachter ploughs relentlessly on, leaving minimal time for critical assessment.

>I

You are carrying a conference handout, and you're wearing a tie, a name badge, and a dark jacket.

"Since we can assume that any literalist reading of the empty tomb story can be instantly deconstructed", Professor Wortschlachter drones on, "we can progress to the correct identification of its *Sitz im Leben* in the proper *religionsgeschichtlichen* context."

Bullshit! you think, convinced even more than ever now that Wortschlachter is just throwing in a random mish-mash of technical language in order to disguise his total lack of methodological rigour.

>READ HANDOUT

It reads:

Our Special Guest Speaker, Herr Doktor Doktor Professor Karlheinz Wortschlachter is delivering today's keynote address on the exciting topic of "Rolling Away the Misconceived Stone: Structural Semiotic

Signification in Mark 16:1-8”

The organizers particularly request all Conference delegates to attend this afternoon’s plenary session at which this internationally renowned scholar will reveal his latest thinking on this exciting and vital topic.

Dr. Wortschlachter is Professor of New Testament History and Exegesis at the Protestant Faculty of the University of Wirtstadt. He is the author of several important books, including the renowned *Deconstructing the Deconstruction of Structurally Symbolic Social-Scientific Semiotic Signification in Second Temple Judaeo-Christianity*. This is the first time we have been fortunate enough to have him address a meeting of the Pan-Atlantic New Testament Society.

“... but this must include, *inter alia* its thorough demythologization in line with the Bultmannian perspective,” the professor grinds on.

At this point your attention starts to wander as you find your eyes attracted to the blonde woman sitting three rows ahead.

The PC’s attention now wanders as he recalls a brief encounter he had with an attractive female delegate, Felicity Hope, at breakfast that morning. His attention soon returns to the lecture, but not long thereafter he suffers a heart attack, and the bulk of the game takes place in his subsequent near-death experience. 1

How exactly this experience is to be interpreted (for example, as an objective depiction of an after-life, or as a purely subjective dream-like experience) is left up to the player to decide. In creating this environment I thought of it as mythological, and as containing both subjective and objective elements. It is closely connected to the theme of the preceding lecture, in that the Empty Tomb story recurs in various guises as a leitmotiv, as does the PC’s attraction to Felicity Hope. On the other hand what the PC does in this segment of the game can have real world consequences, in particular whether he returns to life or ends up in ultimate frustration or oblivion. 2

This mythological section is structured in two main halves surrounded by a prologue and epilogue. The prologue effectively allows players to start to get their bearings in this strange environment. There they will discover that a great chasm separates the PC from a distant golden glow that looks like it might be a city of the blessed. On this side of the chasm is the gate of hell, bearing a sign saying that hell has been closed for demythologization. A workdemon is busily chipping away at the inscription over the gate, but has so far only manage to obliterate the final letter of “Abandon Hope All Ye Who Enter Here” (thereby creating a nice double entendre for those who spot it). It turns out that the only way to progress is to go through hell, via another entrance (the debts to Dante’s *Divine Comedy* will be obvious). 3

The first main section begins once the player has found out how to get into hell. The PC soon arrives in a Victorian gallery with a choice of routes labelled “Via Antiqua” and “Via Moderna”. I originally intended to force the player to take the Via Moderna, which pokes gentle (or not-so-gentle) fun at various aspects of twentieth-century New Testament scholarship (appropriate to the PC’s profession). I soon felt, however, that this left the Via Antiqua as a rather pointless digression, so I decided to add an alternative route through that as well, in which the PC has to overcome the temptations of an alluring female demones (a sort of demonic counterpart to Felicity Hope). Both routes eventually converge at the same barrier (it would be too much of a spoiler to be more specific) that the PC has to overcome to proceed to the second main section of his near-death experience. 4

This second section is intended to be a recapitulation of the main lines of New Testament eschatology. Symbolically speaking the PC must past through the crucifixion and the Empty Tomb and then play an apocalyptic game of chess against an opponent who starts out as Pontius Pilate but is soon replaced with a composite Roman emperor (representing an amalgam of all the emperors from Augustus to Domitian). The chess game is played out with somewhat unconventional pieces and is meant to reflect, first the events leading up to the Jewish war of 66-70 CE, and then aspects of the apocalyptic scenarios of Revelation and 4 Ezra. If the PC loses the game he loses his soul; if he wins he can pass on to the next stage. In neither case is the game played much like an ordinary game of chess. 5

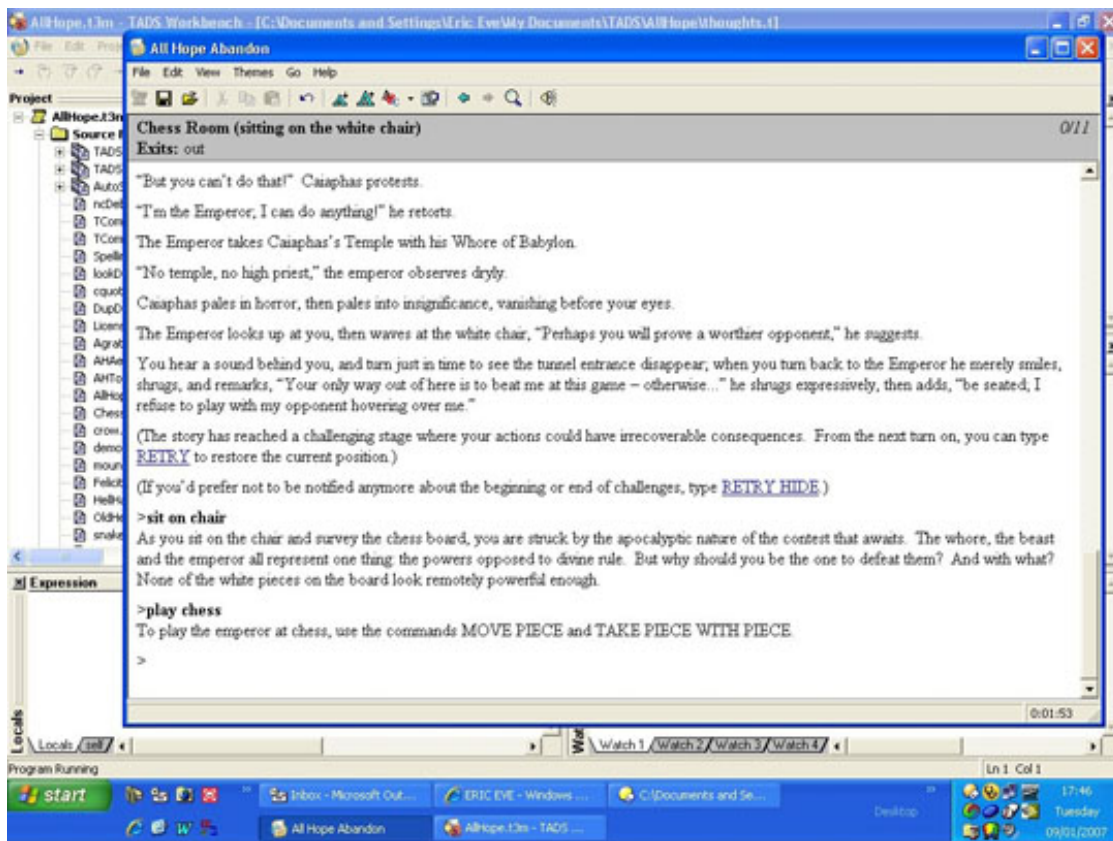


Figure 2. The chess game, at the point where the Player Characters takes over from Caiaphas as the Emperor's opponent. The text in bold in the main (white) window represents commands entered by the player (sit on chair, play chess); the text that follows constitutes the game's response. Note how the command "play chess" does not actually achieve anything beyond causing the player to be prompted with the specific commands to be used; this is a useful aid to a player who may be unsure what kind of command to enter at this point.

Thereafter the player character finds a sort of reverse Eden, in which the serpent is helpful and in fact prevents the PC from taking the forbidden fruit (the idea is to represent an *Urzeit – Endzeit* typology^[1] and the reversal of the fall through the acts of redemption), but realizes that this is not where he is meant to stay. Instead he must rescue Felicity Hope from the swamp in which she is mired (and which is clearly meant to represent Professor Wortschlachter's lecture). Escape from this region leads to the epilogue of the near-death experience. Between them Felicity Hope and the PC discover a way to cross the chasm to reach the distant golden glow, but only one of them can cross. The player must decide whether it is the PC or Felicity who crosses the chasm, and on that decision rests the ultimate fate of the PC. If the right choice is made, the PC wakes up in a hospital bed with the real Felicity Hope at his side; otherwise he is eternally lost.

6

In the most general terms, my design goals were to combine my interest in Biblical Studies with my interest in interactive fiction in a game that contained a substantial surreal (or at least, non-real-world) segment that would give some sense of what it might be like to inhabit one interpretation of the Christian symbolic/mythological universe. I also wanted to poke fun at some aspects of New Testament scholarship of the last hundred years, while unobtrusively informing players about it. The game also had to be entertaining and engaging to play, given that my target audience would be fellow IF-enthusiasts rather than fellow New Testament scholars. It therefore had to conform to the conventions and expectations of the medium, and it seemed advisable to include elements such as puzzles that would hopefully prove amusing and accessible to all players, regardless of their background knowledge, on the basis of information provided in the game. It was quite emphatically not meant to preach to the player.

7

The game seems to have achieved mixed success in relation to those goals. On the positive side "All Hope Abandon" received nominations in six out of the ten categories for Xyzy awards (the Oscars of IF, named after a magic word featured in "Adventure") for games released in 2005, although it failed to win the prize in any of them (see

8

<http://wurb.com/if/award/3#518>). In general it also seems to have been well received; the fact that it does not try to preach to the player was generally appreciated, and it received a number of broadly favourable reviews.^[2] On the other hand the overarching scholarly and mythological structures that were intended to lend coherence to the two main afterlife sections were clearly missed by some players, to whom these sections then appeared to be a somewhat rag-bag collection of puzzles [Shiovitz 2004a].

This underlines the main design challenge I faced: the huge gap between PC and player knowledge. Although this can be an issue in any work of IF, in this case the problem was acute: my PC was a New Testament scholar who could be expected to have a thorough knowledge not only of the biblical text and related ancient texts, but also of the contemporary scholarly debate surrounding them. My target audience might have none of this knowledge, and it became apparent in the course of beta-testing and subsequent player feedback that many players did not possess even the most elementary knowledge of the biblical stories presupposed and alluded to in the game. 9

My first attempt at meeting this problem was to add footnotes to responses I thought might be a little obscure.^[3] But I quickly came to the conclusion that this solution was both cumbersome and obtrusive. Players who wished to refresh their memories of what a footnote said on a particular topic would have to remember the number of the footnote in which the topic was discussed (or else scroll back to find it), and there were so many responses that potentially needed further explanation that the game was beginning to resemble an article in an academic journal; the too frequent appearance of footnote markers would make the game look too forbidding. 10

I therefore decided to scrap the footnotes and employ a different approach: implementing a THINK ABOUT command. If the game mentioned something that might be obscure to the player, such as the Synoptic Problem or Q, or even Moses or the Empty Tomb, the player could simply type, say, THINK ABOUT Q, to tap into the player character's specialist knowledge (and opinions). The advantage of this approach is that it never thrusts itself on players' attention; there is seldom any need to think about anything in the game,^[4] and players can use the THINK ABOUT command when and only when their curiosity is roused. This approach also opens up further educational potential for the game; those players can learn quite a lot about the biblical text and scholarly approaches to it if they are so minded, but none is compelled to do so, though hopefully the context of the game makes it an enjoyable way to learn. 11

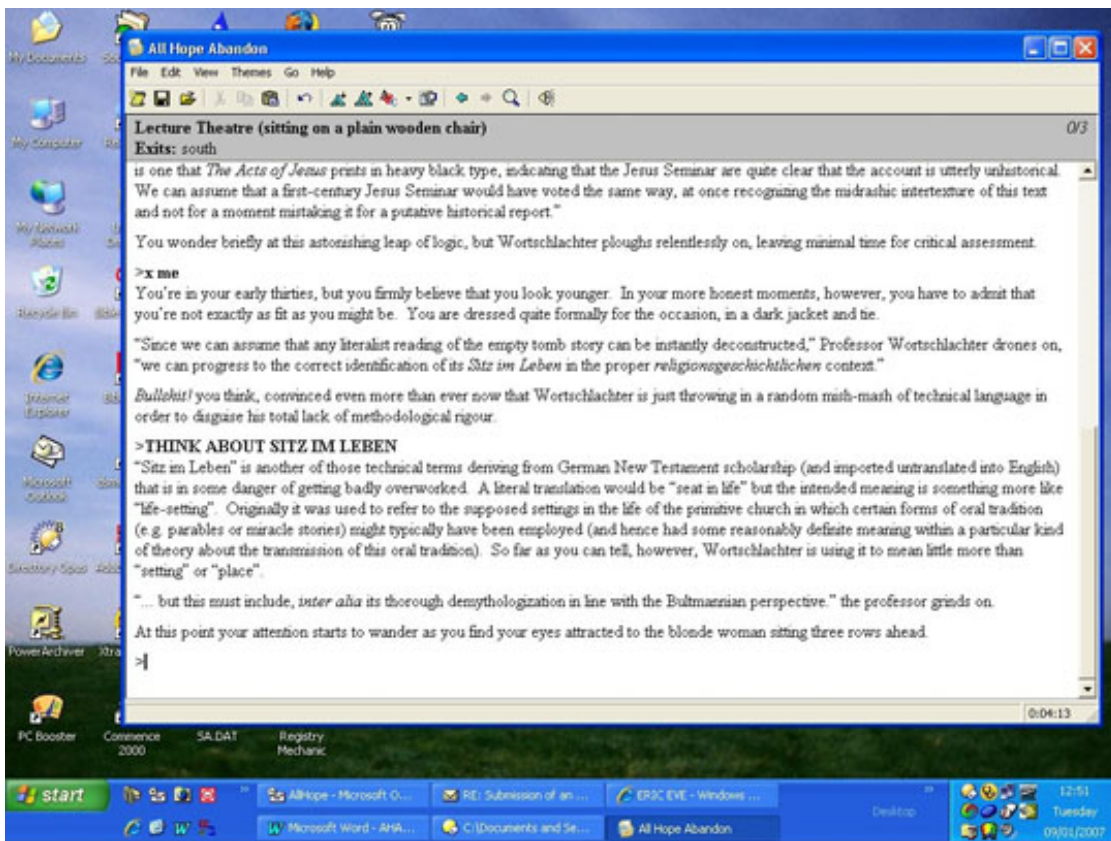


Figure 3. The THINK ABOUT command in action. On the previous turn Professor Wortschlachter used the expression “Sitz im Leben” in the course of his lecture; since this term is likely to be unfamiliar to most players, the inquisitive player can use the THINK ABOUT command to find out what it means. Note that the command on the previous turn was X ME, X being a standard IF abbreviation for EXAMINE; providing a suitable response to X ME is one way an author can quickly convey something about the Player Character to the Player. In this instance both THINK ABOUT and X ME serve to close the gap between what the Player Character knows and what the Player knows.

Many of the other challenges and design decisions affecting “All Hope Abandon” were of a kind common to many works of IF, such a pacing and puzzle design. The key decision made in the design of “All Hope Abandon” was to link narrative progress with geographical progress; the narrative advances through the near-death experience as the player character advances through the map. This in turns means that most of all the puzzles are “gating” puzzles, obstacles that need to be overcome in order to advance to the next set of map locations. All this was part of the design from the beginning. A subsidiary aspect of this design included making the gating puzzles irreversible at crucial points, limiting the ability of the PC to backtrack and so driving the narrative forwards. Linked to this were puzzles that forced the PC to part with all of his inventory (the items carried) at various points, thus preventing an accumulation of an ever-growing stack of objects that were no longer useful.^[5] 12

Apart from a number of purely technical issues, the other main design challenge was the implementation of the NPCs. Here, the setting of the main part of the game in a mythological landscape greatly helped; since the setting was clearly not a real-world one, and many of the NPCs were so strange (they included a demon, a demoness, a talking crow and a talking serpent) that it would not be particularly jarring if some of their responses were a little odd; on the contrary, oddity is only to be expected. “All Hope Abandon” does make full use of the conversational tools provided by TADS 3, but its use of them is less taxing than it would be in a game set in more realistic world. Moreover, the two real-world segments right at the beginning and end of the game are so brief that the PC only has the opportunity to converse with one NPC (Felicity), and then only for such a short span of time that any unnaturalness in the conversation hardly has a chance to become apparent. 13

“All Hope Abandon” breaks no new ground in the matter of room descriptions, the constraints of supplying adequate information in limited space remain, but it does add one related feature, the ability of the PC not simply to LOOK at the 14

current location, but to LOOK in particular directions (such as LOOK NORTH or LOOK SOUTHEAST). The game then responds with a brief description of what can be seen in that direction from where the PC is currently located. This probably turned out to be little more than a gimmick, however, since it would have been too unwieldy to have supplied any new information to the player by this means. If new information had been conveyed to the player through such commands, then players would have been forced to look in every direction in every location to be sure of not missing anything, and this would rapidly have become intolerably tedious. It would also have gone against the convention that a normal room description should supply players with all the information they need (which may include calling attention to objects they need to examine more closely). The ability to LOOK in particular directions may have created a slightly deeper sense of immersion for some players, since the game could at least give a reasonably appropriate response to such commands, but it is not an experiment I have so far felt like repeating in subsequent works.

Interactive Fiction and Biblical Text

At first sight interactive fiction, which began life as a species of computer game, would seem to be a world apart from Biblical Studies (or any other humanities discipline). It may nevertheless be interesting to reflect on how a work such as “All Hope Abandon” may be suggestive for relating these seemingly disparate worlds.

15

The first point to make is that “All Hope Abandon” makes no attempt to render the biblical narratives it employs in any straightforward, realistic manner. It alludes to a number of biblical stories, and full appreciation of the game probably requires quite a good grasp of the sweep of biblical narrative, but nowhere does “All Hope Abandon” make a serious attempt to retell a biblical story. This is partly a recognition of the real differences between the media of static text and IF; attempts to render existing stories as IF rarely come off well.^[6] Taking all or part of a static text like Mark’s Gospel and trying to turn it into IF almost certainly would not have worked (though a piece of IF loosely inspired by the biblical text could be a very different matter). For one thing there is an essential mismatch between the linearity of the narrative in a conventional static text, and the player’s apparent freedom in a work of IF. This freedom may only be apparent; a skilled IF author can effectively force players to a predetermined conclusion while creating the illusion of almost total freedom, but the route to that conclusion cannot be so tightly controlled as in static fiction (or, if it is, the resulting game will scarcely work as a piece of IF; players generally object when the game too obviously puts them on rails). There is no way that an IF author can successfully force a player to follow the precise sequence of events that appears in a static text, and a well-written piece of IF has to reckon with the possibility that a player may choose to do something different from the protagonist of a pre-existing story.^[7]

16

That said, there are a number of ways in which “All Hope Abandon” functions similarly to the biblical texts it plays off. The first is in the employment of intertextuality. For the most part “All Hope Abandon” uses the biblical text (and other texts) allusively, occasionally quoting them, sometimes representing aspects of them more or less closely. This is similar to the way in which many New Testament texts make use of the Old Testament. Thus, for example, the opening of Luke’s gospel is written in such a way as to resemble the opening of 1 Samuel, or the opening of Matthew’s gospel recalls the birth of Moses and a dreaming patriarch named Joseph. Thus, too, is the way that several of the more spectacular miracle stories operate; the Feeding of the Five Thousand (Mark 6.30-44 and parallels) resembles a similar feeding conducted by the prophet Elisha (2 Kings 4.42-44) and is probably meant to recall the feeding in the wilderness (Exodus 16) as well as the promise of a Davidic shepherd who would heal and feed his flock (Ezekiel 34). Likewise the Stilling of the Storm (Mark 4.35) recalls both the story of the storm at sea suffered by Jonah (Jonah 1) and the song of praise from those who go down to the sea in ships (Psalm 107.23-32). Such examples could be multiplied almost endlessly; they are integral to the way much of the New Testament (and also quite a bit of the Old Testament) functions.

17

^[8] To be sure the New Testament is not only a static text but a sacred one; its employment of intertextuality is inevitably both different in nature and more serious in intent from that of any work of IF. Again, the intertextual allusiveness of “All Hope Abandon” is hardly characteristic of IF as a whole. There are nevertheless similarities in intent insofar as both the New Testament and “All Hope Abandon” seek to create new narratives that rely for much of their meaning on earlier sacred texts, and neither is simply giving a new rendering of an old story.^[9]

A less incidental link between IF and biblical text lies in their common relation to the riddle. Nick Montfort argues that the

18

riddle is one of the main literary precursors to interactive fiction in that it is both textual (or at least verbal) and challenges the reader/hearer to find a solution [Montfort 2003, 37–63]. There is also a riddling quality to some biblical texts, not least the Gospel of Mark, which provides some of the main biblical allusions in “All Hope Abandon”. Mark 4.1-34 to some extent presents Jesus’ parables as riddles;^[10] to those outside, everything comes “in parables” so that “seeing they may see and not perceive, and hearing they may hear and not understand” (Mark 4.10-12, alluding to Isa. 6.9-10). The parable of the sower is thus made a parable about the reception of parables, but just as seed is sown in the ground for the purpose of growth, so the concealment of teaching in the guise of parables is apparently intended for the purpose of revelation: “for nothing is hidden except in order to be made known, nor does it become hidden but that it might come into the open” (Mark 4.22).^[11] The Markan Jesus continually challenges the disciples to understand, and they continually fail (Mark 4.10-13, 41; 6.52; 7.17-18; 8.14-21). Mark’s readers are challenged to do better: “if anyone has ears to hear, let them hear” (Mark 4.23). The careful reader of Mark’s Gospel may easily come to the conclusion that this applies, not just to the parables in Mark, but to the entire narrative: there is clearly much more going on in this gospel than appears on the surface.^[12]

Hiding something in order that it may come to light is (both literally and figuratively) a common strategy in writing IF; it is the essence of puzzles that the player/reader is meant to solve them in order to advance the plot, and perhaps also the player’s understanding of the work as a whole.^[13] Used well, this is intended to deepen the player’s involvement in the narrative; to the extent players are forced to provide their own solutions to the puzzles the story provides they tend to find themselves taking ownership of the outcomes. The puzzling nature of the parables (and other material) in Mark’s Gospel seems directed at a similar (though deeper) end; in particular it seems to be part of Mark’s strategy to bring his target audience round to the authorial viewpoint (or, at least, the point of view of the implied author of the Gospel).^[14] Gospel and IF games remain examples of different media, but there is here a convergence of reader responses, even though in the case of the Gospel the reader cannot direct the actions of the protagonist. Insofar as a work of IF is designed so that part of the challenge of the work is to understand what is going on, that convergence becomes greater (without ever approximating to identity).^[15]

Although a modern reader cannot affect either the course of events or even the style of narration in a static text like a gospel, the situation may have been a little different when the gospels were first written. Like most texts in antiquity (when literacy was restricted to a small fraction of the population), they would have been written as scripts for oral performance, not books for private perusal, and some proportion of the material now incorporated in the gospels probably enjoyed oral circulation for some time before being committed to writing. Mark’s Gospel is particularly close to oral storytelling, both in its folksy narrative style (such as the frequent use of the historic present, parataxis, the word “immediately”, and otherwise non-literary Greek), and in its overall structure.^[16] Although it is now in fixed textual form, it was very likely shaped in interaction with its first hearers.^[17] What we have now is a static text, and so a modern reading does not reproduce this experience of shaping the text (as opposed to its interpretation), but there is a possible analogy between the way many biblical texts came into being and the way the textual output of a work of interactive fiction comes into being, by the co-operation of performer/author and active listener/reader.^[18]

Finally, IF provides an interesting analogy in the age-old problem of divine sovereignty and human free will. One obvious place where this surfaces in the biblical text being used for comparison here is where the Markan Jesus declares that his betrayer is performing a preordained role in the divine plan, but must nevertheless bear full responsibility for his actions (Mark 14.21), but the problem is clearly much wider than that one instance. If the universe has been made by an omnipotent and omniscient creator, what space is left for human free will? One useful analogy is that between the author of a novel and the characters within the novel. Clearly the novelist has full control over the story world he or she creates and the actions of everyone with it, yet if the characters are not to appear mere puppets, they must have a certain authenticity and vitality of their own; from the inner-narrative perspective of the story world they act as free agents. It would not be appropriate, for example, for a criminal collared by Sherlock Holmes to plead, “It wasn’t my fault, it was that scoundrel Conan Doyle who made me do it; he’s writing this story after all.” Such an excuse would involve a confusion of levels, since at the story-level the villain is free and responsible for his actions.

19

20

21

But useful though that analogy is, it breaks down if pressed too far, for, after all, novelists do control all the thoughts and actions of all their characters, so that their characters enjoy no real independence. Interactive fiction provides a more satisfactory analogy, since here the protagonist is genuinely guided by an intelligence distinct from that of his or her creator, namely that of the player. The story-world is still the creation of the author, but the player is genuinely free to direct the actions of the PC. It is up to the author what difference that makes; a work of IF can be so structured that the outcome is the same regardless, or so that the PC's choices make a real difference. Both design options are intriguing in this context, the one modelling freedom of action leading to a predetermined conclusion, and the other a created world in which the author allows choice to influence outcome.

22

It is true that the actions open to a player character in a work of interactive fiction are generally circumscribed, but this is often if not mostly true of real life as well; in practice we all operate under a number of constraints that limit our practical options to a finite and frequently quite small number. Reflecting on the relation of author, player and player character will not solve any age-old theological problems, but it may provide some new insights for thinking about human free will in relation to the powers of a creator.

23

Conclusions

Interactive fiction is a distinct medium. In many important respects it is unlike static fiction, film, drama, opera, poetry or any such more conventional medium; in certain respects it perhaps most closely resembles the give-and-take of oral narrative performed in the presence of a participatory audience. Its distinctiveness means that it is unlikely to shed much light on, say, the kind of traditional historical-critical questions that still occupy much of contemporary biblical scholarship. But it may help to generate fresh insights in other areas, such as how readers and listeners interact with texts, and how the puzzling nature of a text like Mark's gospel helps draw readers into the narrative and take ownership of their interpretations. That virtually all interactive fiction has an element of play does not of itself disqualify it from serious consideration; there can be a playfulness about static texts as well, including the way some biblical texts play with other biblical texts.

24

"All Hope Abandon" has so far had a small audience of IF enthusiasts. It is unlikely that those will have included many who are also biblical scholars or theology students. Yet it may have the potential to entertain both (not least the former who will be in the best position to understand its jokes and appreciate its allusions) and perhaps even contribute to the education of the latter. It would be interesting to see the reactions if it ever did find such an audience.

25

If you would like to try out "All Hope Abandon" for yourself, you can play through a demonstration of its opening. To experience the complete game you will need to obtain both the game file and a TADS interpreter for your system (see <http://www.tads.org/tads3.htm> for a list of those available);^[19] for Windows the best option is probably the TADS 3 Player Kit (<http://www.tads.org/t3dl/pksetup.exe>); for Linux/Unix or BeOS you can use FrobTADS (<http://www.tads.org/frobtads.htm>); for Mac OS X use either FrobTADS or Spatterlight (<http://ccxvii.net/spatterlight/>). If you need more help, refer to the "Beginner Resources" page at <http://brasslantern.org/beginners/> or the FAQ at <http://www.plover.net/~textfire/raiffaq/ifaq/>. In addition, you can always post questions (or comments) to the internet newsgroup rec.games.int-fiction, either through a newsreader (news://rec.games.int-fiction) or a web browser (at <http://groups.google.com/group/rec.games.int-fiction>). Readers may also be interested in the source code for "All Hope Abandon".

26

Glossary

Apocalyptic: A term that is used almost as loosely within biblical scholarship as outside it; etymologically the word is derived from the Greek for "revelation" or "unveiling" and some scholars would wish to restrict its use to a type of literature in which heavenly secrets are revealed to the seer by a vision, a heavenly journey, or the words of an angel (or some combination of all three). All too often, however, the term is used as if it meant eschatological, and for want of a better term it is often used to describe a particular type of eschatology in which ordinary human history is expected to be interrupted by a catastrophic divine intervention in the near future (the reason being that such eschatology is frequently expressed through the

medium of apocalyptic in the first sense).

Byte Code: The name often given to the intermediate product of compilation; a byte code file can typically be run on an interpreter or virtual machine rather than directly by a computer's operating system.

Code: Without further qualification this normally refers to "Source Code", the set of instructions the author/programmer writes in whichever programming language he or she is writing to tell his or her program (in this case, work of IF) how to behave.

Compiler: A piece of software to translate human readable source code into machine readable form. Compilation may be to a native executable (a file that can be executed by a particular operating system without further ado) or, more usually with Interactive Fiction, to some intermediate (byte code) form that can be run by different interpreters on different operating systems.

Containment Hierarchy: The system of containment that determines which objects are within which other objects. In this context "within" includes inside, on top of, underneath or behind. At the top of the containment hierarchy stand the rooms; rooms are regarded as not being contained by anything but as directly or indirectly containing everything else (except for objects temporarily out of play and so outside the map altogether).

Debugger: A tool which helps a programmer (or in this case, game author) track down programming errors or "bugs". For example TADS 3 Workbench for Windows incorporates a highly sophisticated debugger that allows the programmer to step through his code line by line to see what it is doing each step of the way (and so determine where something is going wrong if it is going wrong).

Domain-Specific Language: A computer programming language (such as Inform or TADS) specifically tailored to a particular type of task (such as writing Interactive Fiction).

Eschatology: Anything that deals with the last things, the end of the age or the consummation of history (from the Greek eschatos, meaning "last"). In Christian theology eschatology traditionally deals with matters such as Heaven, Hell, the Second Coming and the Last Judgement; in a first-century context it is better thought of in terms of the coming of the kingdom of God, the age to come in which the evils of the present age would be overcome.

IDE (Integrated Development Environment): A single piece of software containing all (or at least) most of the tools a game author needs in order to write a game, including an editor, a compiler, a debugger and an interpreter on which the game may be tested.

Inform: The most popular language/system for authoring Interactive Fiction, written and maintained by Graham Nelson. Two versions of Inform are currently in use. Inform 6 (the older version) resembles a conventional programming language. Inform 7 adopts a more "natural language" style of programming with the aim of making the process of writing Interactive Fiction more like writing prose and less like conventional programming.

Implicit Action: An action automatically carried out by the parser in order to facilitate the command the player actually typed; for example if the player types GO THROUGH RED DOOR when the red door is locked but the player has the key, a well-behaved parser would carry out implicit UNLOCK RED DOOR and OPEN RED DOOR actions rather than forcing the player to type these commands explicitly.

Inventory: The items currently carried by the Player Character.

Interpreter: A piece of software that (typically) implements a Virtual Machine and so allows a game compiled to byte code to be run on the player's computer; one generally needs an interpreter to run a work of Interactive Fiction in the same way that would need a copy of MS-Word to read a Word file, or a program such as Windows Media Player to watch a DVD on a computer. The advantage is that an interpreter needs to be written only once for each operating system (Windows, MAC-OS, Linux, etc.) and can then play all the games written for that interpreter.

Library: A set of supporting routines supplied with an IF authoring system such as TADS or Inform that handles most of the tasks common to most works of Interactive Fiction; a library will typically implement a parser and at least a basic world model, and will generally be written in the same language as that supplied with the authoring system to allow ready customization by game authors.

Map: The totality of rooms within a particular work of Interactive Fiction, together with the interconnections between them.

Menu: A set of options presented to a computer user from which he or she is meant to select one.

Non Player Character (or NPC): Any animate character (human, animal, alien or robotic) that appears in a work of Interactive Fiction apart from the Player Character; characters whose actions are not controlled by the Player Character. Usually, the term is restricted to characters who are actually implemented and can be interacted with rather than characters who are only mentioned or who never remain on stage long enough for the Player Character to interact with.

Object: Something physical that the Player Character can interact with (or at least examine) in the course of the game, such as a table, chair, pen, or book (or, indeed, another character). Some objects may be quasi-physical (such as smoke or sunlight). In many IF authoring systems based on OOP (object-oriented programming) design, "object" may also refer to a programming object; in such systems a game object (in the first sense) will usually be represented by a programming object (the second sense), but programming objects may also be used for other purposes (if you don't know anything about OOP, this second sense of "object" is probably irrelevant to you).

Parser: That part of the software in a work in the Interactive Fiction that interprets the player's command and translates it into an action that the program can perform (or else complain if translation is impossible or ambiguous).

Player: The flesh-and-blood human being typing the commands at a keyboard and reading the game's responses on screen.

Player Character (or PC): The protagonist of a work of Interactive Fiction, whose actions are controlled (or at least guided) by the player, and through whose eyes and ears the fictional world is described to the player.

Room: A (normally discrete) unit of space within the game's map. The usual convention is that the Player Character can interact with anything that is in the same room as himself or herself (unless it is deliberately concealed), and so a room is effectively that volume of space that is immediately present to the Player Character. A room may be a room in the ordinary sense of a room within a building, but it may also be a section of space outdoors, such as a meadow, the top of a hill, or a section of street.

Scope: Roughly speaking, the set of objects with which the Player Character can currently interact, given

the nature of the action he or she is meant to be carrying out. Normally this will be restricted to the objects the PC can see or touch, but this is not always the case. If the action is conversational, e.g. ASK ACTOR ABOUT TOPIC, then any object (or topic) is potentially in scope. If the author has implemented a GO TO ROOM command to take the Player Character to anywhere on the map, then any (known) room will be in scope.

Spoiler: A piece of information (typically in a review) which reveals too much to someone who has yet to play the game, and so spoils the experience of playing it for the first time.

TADS: An acronym for Text Adventure Development System, one of the two major languages/systems in which amateur Interactive Fiction is currently written. TADS 2 is still in use, but the latest incarnation (in which “All Hope Abandon” was written) is TADS 3. TADS was written by and is maintained by Mike Roberts.

Virtual Machine (or VM): technically, a computing device implemented in software rather than as a piece of hardware (hence “virtual”); a piece of software that can execute byte code written for it, usually implemented as an interpreter for Interactive Fiction.

World Model: That part of a work of an IF that defines the behaviour of the story-world in which the action takes place (such as movement around the map, the containment hierarchy, and some basic physics); typically, much of a game’s world model is supplied by the library of the authoring system used.

Z-Machine: The Virtual Machine for which the Infocom games were written, and to which games written in Inform compile.

Notes

[1] For other definitions of Interactive Fiction, see e.g. Roberts (2006); Maher (2006); RAIF (2007); Jerz (2000a); Montfort (2003a), vii-xi, 1-35. For a more thorough analysis of the medium see Montfort (2003b).

[2] For more details of how TADS 3 uses such grammar productions for pattern matching, see <http://www.tads.org/t3doc/doc/sysman/gramprod.htm>.

[3] Although the description here is specific to TADS 3, the principles would apply to most modern IF authoring system. For an account of the Inform 6 Parser see Nelson (2001), 194-269.

[4] The term is actually derived from its use in caving, since the first ever work of IF was a caving simulation [Montfort 2003, 85–93]; [Nelson 2001, 342–5].

[5] Probably because someone designing a work of IF containing more than a handful of rooms almost certainly needs to draw a map indicating their spatial relations before attempting to write the game, and players often find it useful to draw schematic maps as they play.

[6] It is sometimes desirable to model sense passing between rooms, for example looking out through a window, or modelling the contiguity of neighbouring locations in a large open space, and this can be done. It is also possible to experiment with moving around within a room. The description given here nevertheless broadly applies to most IF.

[7] Many versions of the game are available at <http://www.wurb.com/if/game/1>

[8] Infocom was acquired by Activision in 1985 and there is no longer a current website for it. A website paying homage to the company and supplying some information about it can be found at <http://www.csd.uwo.ca/Infocom/>.

[9] Constraints of space prevent a fuller history of interactive fiction here. For rather more comprehensive accounts see Montfort (2003a) and Nelson (2001), 342-63 (available for download from http://www.inform-fiction.org/manual/download_dm4.html); see also Maher (2006); and the on-line articles at <http://brasslantern.org/community/history/>.

[10] The TADS 3 compiler can also produce a Windows native executable, which effectively bundles the interpreter with the game code. The TADS 3 compiler and tools are themselves written in C++.

[11] See <http://www.tads.org/>.

[12] For Inform 6, see <http://www.inform-fiction.org/inform6.html>. For the Z-machine see note 14 below.

[13] For Inform 7 see <http://www.inform-fiction.org/I7/Welcome.html>; for TADS 3 see <http://www.tads.org/tads3.htm>.

[14] Games written in Inform 7 undergo two stages of compilation. The Inform 7 compiler first translates the “natural language” code into the Inform 6 programming language, and then compiles the Inform 6 code into byte-code that will run on the Z-machine (a virtual machine originally designed by Infocom for running games they wrote in ZIL, which was short for “Zork Implementation Language”). For details of the Z-machine see Nelson (2001), 304-41.

[15] There are also significant departures from C, however; for example variable types are not defined at compile-time, provision for string-handling is much more programmer-friendly, and although the object/class model bears some resemblance to that of C++, there are important differences in detail, not least in the fact that TADS 3 minimizes the distinction between objects and classes.

[16] Historically, Inform 6 has proved the most popular IF development system, and it seems likely that the majority of IF will continue to be written in Inform 6 and 7 for some time to come. TADS has also always had a significant user base, and there is considerable interest in TADS 3, which should become more widely used once its power is fully appreciated. Other authoring systems include Hugo, Alan and Adrift, the latter two being aimed at non-programmers. It is fair to say that rather less work of high quality has been produced with these systems, although the few people who use Hugo do produce good work with it, and Adrift can be used to good effect in the hands of the adept. For a convenient brief comparison of IF authoring systems see Firth (2007).

[17] See, for example, the generally favourable reviews of Rune Berg’s 2004 game “Isle of the Cult” at <http://www.sparkynet.com/spagi.html#isle>, <http://www.drizzle.com/~dans/if/reviews.html#cult> and <http://www.ifreviews.org/index.php?analyse=2>,

[18] That said, different works of IF can legitimately employ different depths of implementation; the generally accepted standard is that the implementation depth should be consistent throughout any given work.

[19] For further discussion of writing room descriptions, see Nelson (2001), 396-401; Granade (2007); and Coyne (2003).

[20] Adam Cadre’s “Photopia” is perhaps the most celebrated example of a virtually-puzzleless story-based game; see Cadre (1998).

[21] On puzzle design see further [Nelson 2001, 382–95]; [Bates 1997]; [Jerz 2000b].

[22] Jacqueline A. Lott’s game “The Fire Tower”, a simulation of a hike to the Mt. Cammerer Fire Tower in the Great Smoky Mountains National Park, is a good example of this done well. See Lott (2004).

[23] For a useful discussion, see Short (2007a).

[24] A full description of the NPC toolset TADS 3 provides and the rationale behind it cannot be presented here; for a fuller account see the series of articles by Mike Roberts on “Creating Dynamic Characters”, starting at <http://www.tads.org/t3doc/doc/techman/t3actor.htm>.

[25] For further reading on the creation of NPCs in Interactive Fiction see Short (2007b). For an on-line bibliography on Interactive Fiction in general, see Jerz (2001).

[26] For a fuller discussion of Player Character design, see Stevens (2001).

[1] I.e. the notion that the endtime (*Endzeit*) will resemble the beginning (*Urzeit*), particularly in the restoration of paradisaal conditions.

[2] For those by Paul Lee and Valentine Kopteltsev, see Lee (2006) and Kopteltsev (2006). For the mini-review by Damian Dollahite and the subsequent discussion on rec.games.int-fiction, see Dollahite (2005).

[3] This can be done quite readily, and is a standard feature of TADS 3. The player sees a footnote number and can either click on it or enter a suitable command to read the text of the footnote.

[4] With one exception, where doing so constitutes the solution to a puzzle.

[5] This had proved to be something of a problem in my previous game, “Square Circle” [Eve 2004], in which the PC could very easily end up carrying a huge stack of mostly useless items around; regular players of IF quickly learn to pick up and retain anything that is not firmly nailed down, just in case it proves useful or vital at some later stage in the game.

[6] One notable example would be Graham Nelson’s attempt to produce an IF version of the “The Tempest” [Nelson 1997]; more recent examples might include Bill Powell’s IF version of G.K. Chesterton’s novel *Man Alive* [Powell 2006], and, closer to our theme (and to Mark’s Gospel), “The Bible Retold: Loaves and Fishes”, (<http://www.wurb.com/if/game/2962>) though the latter was clearly tongue-in-cheek. Some better received recent attempts to produce IF versions of Interactive Fiction include Torbjørn Anderson, “The Tower of the Elephant” [Anderson 2006], based on the short story by Robert E. Howard, and Peter Nepstad’s games “The Journey of the King” [Nepstad 2006a] and “The Ebb and Flow of the Tide” [Nepstad 2006b] based on short stories by Lord Dunsany (<http://www.illuminatedlantern.com/if/>).

[7] This problem can be eased by making the PC someone who was a relatively minor character in the original story and giving the original protagonist’s role to an NPC. Thus, for example, one could probably write a perfectly good IF rendering of the Empty Tomb story by making Mary Magdalene the PC; but then most of the content of such a piece of IF would have to come from the author’s imagination rather than the biblical text, and although the resulting game *would* in some sense be a retelling of the Empty Tomb story, it would nevertheless be a new telling of it, and not simply a reproduction of the story as found in Mark or any of the other gospels.

[8] For a general introduction, see Moyise (2001). For examples of discussions of Mark’s use of the Old Testament see Marcus (1993) and Watts (2000). For the literary background to the miracle stories mentioned see Meier (2004), 874-970, and Eve (2002), 259-63, 381-84.

[9] Although new renderings of old biblical stories were common enough at around the time the New Testament was being written, the most notable being the first half of Josephus’s *Jewish Antiquities*, which retells the entire Old Testament in the form of Hellenistic historiography. Other examples from around this time would include Philo’s *Life of Moses*, Pseudo-Philo’s *Book of Biblical Antiquities*, and the book of *Jubilees*.

[10] The Greek word *parabolé* (literally “thrown alongside”) means some form of comparison; the corresponding Hebrew and Aramaic terms *mashal* and *methel* have a far wider semantic range, including “riddle” among their possible meanings.

[11] The translation is my own, intended to bring out the force of the purpose clauses. The literature on this section of Mark’s Gospel is vast; for two contrasting approaches see van Iersel (1998), 176-92 and Hooker (1991), 119-38.

[12] It’s possible that this is due in part to a modern audience’s distance from the intended audience of Mark’s Gospel; we could well have lost a great many shared assumptions that would otherwise have helped determine the meaning of this text. The original author quite likely did not intend his gospel to appear quite so puzzling as its modern scholarly interpreters tend to find it.

[13] Some puzzles may literally involve concealing something, such as the key needed to open a particular door, which the player can find by diligently searching the PC’s environment; in more creative puzzles it is the solution that is hidden and needs to be revealed through diligent reading of the text and thinking through the clues the text (i.e. the various responses to player commands) provides.

[14] There is obviously a vast body of scholarly literature on the interpretation of Mark’s Gospel. For this issues raised here interested readers may like to start with Rhoads and Michie (1982).

[15] This should not be misunderstood as claiming any kind of direct genetic relationship between biblical narrative and interactive fiction, beyond a common debt to the riddle.

[16] On the narrative structure of Mark, see van Iersel (1998), 68-86, and Donahue and Harrington (2002), 16-19. For a detailed discussion of Mark’s Greek style, see Elliott (1993).

[17] See Downing (2000), 75-94; also Vansina (1985), 34-5, and Ong (2002), 143; but note the contrasts Ong draws between literary and primary oral narration (pp. 131-52). Neither Mark’s Gospel nor any work of Interactive fiction was composed in a situation of primary orality, and the nature of the audience’s participation is clearly different in each case.

[18] This analogy is perhaps not very apparent in comparing, say, “All Hope Abandon” with Mark’s Gospel; the analogy between the audience’s role in oral storytelling and the player’s role in IF becomes far more explicit in Aaron Reed’s game “Whom the Telling Changed”; for which see Reed (2005).

[19] An interpreter is a program that reads an IF game file, much as MS-Word or MS-Excel is necessary to read a Word or Excel file.

Works Cited

- Anderson 2006** Anderson, Craig. *The Tower of the Elephant*. Video Game, 2006. <http://www.wurb.com/if/game/2955>.
- Bates 1997** Bates, Bob. *Designing the Puzzle*. 1997. <http://www.scottkim.com/thinkinggames/GDC00/bates.html>.
- Berg 2004** Berg, Rune. *The Isle of the Cult*. Video Game, 2004.
- Brass Lantern** Brass Lantern. *Brass Lantern*. 2007. <http://brasslantern.org/>.
- Cadre 1998** Cadre, Adam. *Photopia*. Video Game, 1998.
- Coyne 2003** Coyne, Michael. *Laconic Locations*. 2003. <http://www.mts.net/~coyne/foibles.html#locats>.
- Crowther 1976** Crowther, Will, and D. Woods. *Adventure*. Video Game, 1976.
- Dollahite 2005** Dollahite, Damian. *[GOTW]: All Hope Abandon*. Google, 2005. <https://groups.google.com/forum/#!topic/rec.games.int-fiction/0usWaSB9SIE>.
- Donahue and Harrington 2002** Donahue, John, and Daniel Harrington. *The Gospel of Mark*. Collegeville: Michael Glazier, 2002.
- Downing 2000** Downing, F. Gerald. *Doing Things with Words in the First Christian Century*. Sheffield: Sheffield Academic Press, 2000.
- Elliott 1993** Elliott, J. K. *The Language & Style of the Gospel of Mark: An Edition of C.H. Turner's "Notes on Marcan Usage" Together with Other Comparable Studies*. Leiden: E.J. Brill, 1993.
- Eve 2002** Eve, Eric. *The Jewish Context of Jesus' Miracles*. Sheffield: Sheffield Academic Press, 2002.
- Eve 2004** Eve, Eric. *Square Circle*. Video Game, 2004.
- Eve 2005** Eve, Eric. *All Hope Abandon*. Video Game, 2005.
- Firth 2007** Firth, Roger. *Cloak of Darkness*. 2007. <http://www.firthworks.com/roger/cloak/index.html>.
- Granade 2007** Granade, Stephen. *Descriptions Constructed. 1997-2007*. Brass Lantern, 2007. <http://brasslantern.org/writers/iftheory/descriptions.html>.
- Hooker 1991** Hooker, Morna. *The Gospel According to St Mark*. London: A & C Black, 1991.
- Jasper 2005** Jasper, Pete. *Review of Rune Berg's The Isle of The Cult*. 2005. <http://www.ifreviews.org/index.php?analyse=2>.
- Jerz 2000b** Jerz, Dennis. *What is Interactive Fiction?* 2000. <http://jerz.setonhill.edu/if/intro.htm>.
- Jerz 2000c** Jerz, Dennis. *Puzzles in Interactive Fiction*. 2000. <http://jerz.setonhill.edu/if/Puzzles.htm>.
- Jerz 2001** Jerz, Dennis. *An Annotated Bibliography of Interactive Fiction Scholarship (Including Fan-Produced Criticism and Theory)*. 2001. <http://jerz.setonhill.edu/if/bibliography/index.html>.
- Kopteltsev 2006** Kopteltsev, Valentine. *Review of All Hope Abandon*. 2006. <http://sparkynet.com/spag/a.html#abandon>.
- Lee 2006b** Lee, Paul. *Review of All Hope Abandon*. 2006. <http://sparkynet.com/spag/a.html#abandon>.
- Lott 2004** Lott, J. *The Fire Tower*. Video Game, 2004. http://www.allthingsjacq.com/intfic_firetower.html.
- Maher 2005** Maher, Jimmy. *Review of Rune Berg's The Isle of the Cult*. 2005. <http://www.sparkynet.com/spag/i.html#isle>.
- Maher 2006** Maher, Jimmy. *Let's Tell a Story Together*. 2006. <http://home.grandecom.net/~maher/if-book/>.
- Marcus 1993a** Marcus, Joel. *The Way of the Lord: Christological Exegesis of the Old Testament in the Gospel of Mark*. Edingburgh: T & T Clark, 1993.
- Marshall 1989** Marshall, Christopher. *Faith as a Theme in Mark's Narrative*. Cambridge: Cambridge University Press, 1989.
- Meier 1994** Meier, John. "Mentor, Message, and Miracles". In John Meier, *A Marginal Jew: Rethinking the Historical Jesus*. New York: Doubleday, 1994.
- Montfort 2003** Montfort, Nick. *Twisty Little Passages: An Approach to Interactive Fiction*. Cambridge: MIT Press, 2003.

- Montfort 2003b** Montfort, Nick. *Towards a Theory of Interactive Fiction*. 2003. <http://www.nickm.com/if/toward.html>.
- Morgan 2006** Morgan, J., and Celestianpower. *The Bible Retold: Loaves and Fishes*. Game based loosely on some of the miracle stories in Mark 4-8. Video Game, 2006.
- Moyise 2001** Moyise, Steve. *The Old Testament in the New: An Introduction*. London and New York: Continuum, 2001.
- Nelson 1997** Nelson, G. *The Tempest*. Game based on the play by William Shakespeare. Video Game, 1997. <http://www.wurb.com/if/game/316>.
- Nelson 2001** Nelson, Graham. *Inform Designer's Manual*. St. Charles, Illinois: The Interactive Fiction Library, 2001.
- Nelson 2007a** Nelson, G., and E. Short. *Inform 7 website*. 2007. <http://www.inform-fiction.org/I7/Welcome.html>.
- Nelson and Knight 2007b** Nelson, G., and C. Knight. *Inform 6 website*. <http://www.inform-fiction.org/inform6.html>.
- Nepstad 2006a** Nepstad, P. *The Journey of the King*. Video Game, 2006. <http://www.wurb.com/if/game/3000>.
- Nepstad 2006b** Nepstad, P. *The Ebb and Flow of the Tide*. Video Game, 2006. <http://www.wurb.com/if/game/3002>.
- Ong 2002** Ong, Walter. *Orality and Literacy: The Technologizing of the Word*. London and New York: Routledge, 2002.
- Powell 2006** Powell, B. *Man Alive: A Mystery of Madness*. Pair of games based on the novel by G.K. Chesterton. Video Game, 2006.
- RAIF 2007** rec.arts.int-fiction. *FAQ 3.1. What is Interactive Fiction?* rec.arts.int-fiction. 2007. <http://www.plover.net/~textfire/raiffaq/online/whatisif.htm>.
- Reed 2005** Reed, A. *Whom the Telling Changed*. Video Game, 2005.
- Rhoads and Michie 1982** Rhoads, David, and Donald Michie. *Mark As Story: An Introduction to the Narrative of a Gospel*. Philadelphia: Fortress Press, 1982.
- Roberts 2006** Roberts, Michael J. *A Brief Introduction to Interactive Fiction*. 2006. <http://www.tads.org/if.htm>.
- Roberts 2007a** Roberts, Michael J. *GrammarProd*. TADS 3 System Manual. 2007. <http://www.tads.org/t3doc/doc/sysman/gramprod.htm>.
- Roberts 2007b** Roberts, Michael J. *Text Adventure Development System*. 2007. <http://www.tads.org/>.
- Roberts 2007c** Roberts, Michael J. *Creating Dynamic Characters*. TADS 3 System Manual. 2007. <http://www.tads.org/t3doc/doc/techman/t3actor.htm>.
- Shiovitz 2004a** Shiovitz, Dan. *Review of Rune Berg's The Isle of the Cult*. 2004. <http://www.drizzle.com/~dans/if/reviews.html#cult>.
- Shiovitz 2004b** Shiovitz, Dan. *Review of All Hope Abandon*. 2004. <http://www.drizzle.com/~dans/if/reviews.html#allhope>.
- Short 2007a** Short, Emily. *Conversation*. 2007. <http://emshort.wordpress.com/writing-if/my-articles/conversation/>.
- Short 2007b** Short, Emily. *Characters*. 2007. <http://emshort.wordpress.com/reading-if/characters/>.
- Stevens 2001** Stevens, Duncan. *The Player Character's Role in Game Design*. Brass Lantern: The Adventure Game Website, 2001. <http://brasslantern.org/writers/iftheory/pcrole.html>.
- Vansina 1985** Vansina, Jan. *Oral Tradition as History*. London: James Currey, 1985.
- Watts 2000** Watts, Rikki. *Isaiah's New Exodus in Mark*. Grand Rapids, MI: Baker Books, 2000.
- Woods 2005** Woods, Eric. *Review of Rune Berg's The Isle of the Cult*. 2005. <http://www.sparkynet.com/spag/i.html#isle>.
- van Iersel 1998** van Iersel, Bas. *Mark: A Reader-Response Commentary*. Sheffield: Sheffield Academic Press, 1998.

