Clara Fernández-Vara

Shaping Player Experience in Adventure Games: History of the Adventure Game Interface

THE SEARCH FOR DIRECT MANIPULATION

The adventure game genre is almost as old as videogames themselves. It takes its name from *Adventure*, programmed by Will Crowther in 1975 and expanded by Don Woods the following year. During the 1980s, adventure games were the leading genre in home computers, thanks to companies such as Inform, Sierra Online or LucasArts. In 1993, Cyan's *Myst* became the best-selling computer game (until the arrival of *The Sims*), and reinvented the genre. In the following ten years, the models set by the companies just mentioned were replicated by other developers, without there really being any significant overhauls to the conventions of the genre. This lack of innovation has contributed to the decrease in popularity of the genre in the mainstream. The development of adventure games, however, has not stopped, thanks to diverse independent developers and publishers, as well as a growing community of amateur developers using outdated engines to make their games. In the last couple of years, changes in input devices, such as the popularization of game controllers, touch screens, and the use of gyroscopes have brought fresh air to the genre, creating new ways to immerse into the gameworld.

Current adventure games are very different from *Adventure* – the genre has been in a constant flux, which is most obvious in its interface design. From the prompt for keyboard input of Infocom games, to verb menus, to point-and-click contextual icons, adventure games have gone a long way in shaping player interaction. This constant re-design is not a question of mere aesthetics – it is key to help the player learn what can or cannot be done in the game. The evolution of the adventure game interface has looked for improving direct manipulation of the entities in the game, where the instructions to be followed by the computer imitate situations and objects of the outside world, using visuals instead of a command line (Shneiderman, 2003). Direct manipulation does not only refer to interface metaphors alone, but to any visual representation that is analogous to some concept the user/player is already familiar with. Visuals give immediate feedback about the success or failure of the command, and makes syntax error messages unnecessary. Enhancing direct manipulation also contributes to the illusion of immersion in the game, because the player does not have to learn the command syntax on top of the individual actions, as well as helping sustain the fiction of the world. The player can intervene in the gameworld directly, without typed commands or error messages to remind her she is using a computer program.

This chapter is a history of the transformations of the interface adventure games have gone through in Europe and the US, and how the increase in direct manipulation interfaces has affected the playing experience, from text adventures to the current generation of consoles. The terms and concepts are based on Shneiderman's seminal article 'Direct Manipulation: A Step Beyond Programming Languages' (Shneiderman, 2003), as well as the definitions used by Nick Montfort (2003) to talk about Interactive Fiction, and the game design elements that are being worked on in the Game Ontology Project (see Zagal *et al.*, 2005).

DEFINITION OF ADVENTURE GAMES

The phrase "adventure game" has been used to refer to many different types of games, several of which will not be considered for this chapter. Generic definitions are always complex, and hardly account for all the possible texts belonging to a genre; it is the case in videogames as it is in film, literature or music. Given the limited scope of the chapter, we will list the features that are common to the games I will be dealing with.

The task of defining adventure games is a slippery one. Traditionally adventure games are associated with stories, since it allows players to progress through a narrative event structure. In this sense, one might reckon they fit Espen Aarseth's concept of *ergodic literature* (Aarseth, 1997), because they require "nontrivial effort […] to allow the reader to traverse the text." However, if advancing in a story requires a nontrivial effort, then any game that rewards the player with a segment of a story after she has completed a particular challenge could also be considered ergodic literature. This type of reward is becoming a commonplace in videogames, so that we have puzzle games such as *Magical Drop II* (1996) or fighting games *Soul Calibur* (1999), which will present short cut-scenes after every stage is completed. This way of fragmenting the narrative may not be considered ergodic literature (or storytelling, for that matter). Adventure games, by contrast, use the same framework and entities to tell the story as well as play the game – there is an overlap in the creation of the fictional world, where the story and the gameplay are using the same devices and signifying systems. In other words, the world is the same for the game and the story, the actions of the player overlap with the events in the story, and the entities of the game overlap with the characters and objects of the story. Challenges in adventure games appear in the form of puzzles, i.e. challenges where there is no active agent against which the player is competing (Crawford, 2003). They usually have at most a few correct solutions that must be figured out (Rollings & Adams, 2003). Puzzles slow down the pace of gameplay, since hand-eye coordination and quick reflexes are usually not required.[1]

[1] See the Game Ontology entry for Puzzle, for further discussion and examples: http://www.gameontology.org/index.php/Puzzle

The textual interface and programming language of early adventure games also served as the basis for the literary genre of Interactive Fiction (IF), a genre that, though related, extends beyond the domain of videogames. IF is an exclusively text-based genre of electronic literature; according to Montfort (2003), it is a type of narrative whose content is revealed usually by problem-solving. The pleasure does not come from reading alone, but also from figuring out the puzzles and riddles. Text adventures (i.e. adventure games with a text-only interface) are a sub-genre of IF, defined as "an interactive fiction work in which the interactor controls a player character who sets out on out of the ordinary undertakings involving risk or danger" (Monfort, 2003). This distinction between IF and text adventures is merely thematic – text adventures usually involve accomplishing an extraordinary task, emulating the heroes of folk tradition. For instance, Infocom's *Deadline* (1982), as Montfort (2003) remarks, is a detective mystery, which may qualify as an adventure even if its setting is not "out of the ordinary". On the other hand, a work like *Photopia* (1998) by Adam Cadre is interactive fiction, but would not be considered a text adventure, since there are no complicated puzzles to solve; however, in some sections of the game the player controls a character in a fantasy world who sets out on a quest. The definition between these two textual genres cannot be clear-cut, but rather they are the two extremes. Story-driven pieces with less puzzles tend towards IF, while more challenge-oriented pieces following the traditional quest structure are closer to text adventures.

IF has evolved to give even more emphasis to storytelling, becoming more literary by restricting the player's choice of actions (by using menus, for example), and leaving out game elements such as combat or treasure hunting. Conversely, puzzle-solving, treasure-hoarding and spatial navigation prevail as forms of interaction in text adventures. The differences between both genres are now more evident, given that adventure games have left behind the text-only screen and developed a graphical user interface.

Compared to IF, adventure games are story-driven puzzle games, where the story and the puzzles use the same signifying systems. This is clear in the case of text adventures, since the story appears written on the screen, and the player must also type to affect the development of the game. Graphic adventure games (i.e. adventure games with a graphical interface) told stories with images and text; the characters that appeared in the story would also be available for manipulation during gameplay. It is common in most game genres to present cut-scenes with the stories of the manipulable entities; however, the actions presented in these cut-scenes are not available as types of manipulation in the game. For example, in *Metal Gear Solid 2* (2001), the player character only speaks face-to-face in cut-scenes, but dialogue is not an option during gameplay.[2] Other game genres are incorporating strong backstory components as a way to engage the player, also as a commercial hook-up to relate the game to some pre-existing franchise that extends to films, comics or novels, such as

[2] Metal Gear Solid 2 incorporates a radio-talking system, which takes the player outside the gameworld, to save the game and get various pieces of information about the place/situation the player character is in. It is more of an information system in dialogue form than an actual dialogue system.

the games inspired by the James Bond franchise *Goldeneye* (1997) and *From Russia with Love* (2005).

As a consequence of the overlap between the manipulable entities and characters in the story, there is always going to be a player character, which will be the main entity carrying out the player's commands in the gameworld and providing the point of view for spatial navigation. The relationship between the character and the player will waver from being a surrogate to complete detachment; the interface will help define the relationship between both. The player character does not have to be unique – *Maniac Mansion* (1987) (and its sequel *Day of the Tentacle* (1993)) start with three player characters. A more intricate and self-referential example can be found in *Suspended* (1983), where the player character is in cryogenic suspension and cannot move, he can only send orders to the robots in the complex where he is kept, so that they communicate with different computers to save the world. That is, the player manipulates a character who manipulates robots that move around and interact with computers.

The features listed so far (story-driven games, with a player character) overlap with those of many other videogames that would not be considered objects of study here. For instance, the commercial site Mobygames lists *The Secret of Monkey Island* (1990) along with *Final Fantasy VII* (1997), *Planescape: Torment* (1999) and *The Legend of Zelda* (1986). Only the first title in the list is relevant for this study, because of the main mode of interaction is based on a verb + object structure. The other examples are different types of role-playing games (RPGs), a game genre that certainly relates to adventure games because it is also story-driven. However, the predominant form of challenges in RPGs is combat, the success of the player depends on managing resources in the form of character stats.

Adventure games usually offer a variety of actions to perform in the form of verbs; the player advances in the game by choosing the correct action and object in the gameworld. For example, in *Zork II* (1981) the player can "pick up brass lantern" to add the item to her inventory; she can then "light lantern" or "turn off lantern". All those actions are supported by the game parser and have an effect; however, trying "eat the lamp" will be responded with "I don't think that the lamp would agree with you." It is in the larger range of actions (verbs) that adventure games have tried to differentiate themselves from other genres.[3]

Unfortunately, this range is also the Achilles' heel of the interaction – how does the player know what she has to do? What is the right verb? At their worst, a session of playing an adventure game turns into a festival of combinatorics of verb + object (+ object). The crux that adventure games have been addressing over the years relates to the struggle between giving the maximum number of actions possible to the player – to create the illusion of freedom of interaction – and letting the player know exactly what has to be done to prevent frustration.

[3] Verbs are also the key for what Chris Crawford calls "interactive storytelling" – the more interactive the designer wants to make a story, the more verbs are needed (Crawford, 2003).

The variety of actions usually goes beyond mere movement commands, attacks and trading, as in other genres such as first-person shooters or real-time strategy games. In order to advance in the game, the player must navigate the space to solve a series of puzzles by resorting to the verb + object type of interaction. These puzzles are usually interwoven, so that by solving a puzzle the player obtains an object to solve a previous one, or a new puzzle or puzzles appear,

The resolution of concatenated puzzles is not the only reward – adventure games encourage exploring the environment to its last possible detail, through navigation, examination of the objects and settings, and interaction with the non-player characters (NPCs). The exploration element has been handed down from the original *Adventure*, which was supposed to be a computer recreation of a real cave, imbued with Tolkienesque puzzles (Nelson, 2001). When the balance between exploration, NPC interaction and puzzle-solving is achieved, the game offers a world that is alive, enticing, and populated by characters that respond believably to the player's input.

Of the three elements just mentioned (exploration, NPC interaction and puzzles), convincing NPCs can be key to the appeal and consistency of the world. Most times an elaborate AI system does not guarantee a believable character (though it can certainly help). Adventure games try to achieve the verisimilitude of their characters usually through their dialogue, rather than their behaviour, which usually implies the presence of a dialogue system that is part of the wide range of actions available. The command "talk to + living object" can be enough most times to start the interaction of the player character with NPCs, though some systems may offer the possibility to choose what to talk about. Dialogue is always a problem, especially in the case of this genre, where it is supposed to be a means of enhancing the exploration of the gameworld and providing information about it. If finding the correct combination of verb + object can be tedious and frustrating, this grows exponentially when it comes to find a whole sentence structure to obtain the information needed. Adventure games have come up with different solutions to this problem, from typing the dialogue to menus and pre-canned conversations displayed when the "speak" command is chosen (this last being equivalent to a cut-scene at times). Significantly, the mechanics of these systems are highly dependant on the design of the overall interface, as the examples below will demonstrate.

The existence of different diegetic levels of interaction is another defining feature of adventure games. The gameworld is the main level of interaction where the events and actions of the game take place, being roughly equivalent to the concept of *diegesis* in literature and film. Montfort (2003), taking literature as his reference, defines the player actions that take place at this level as *commands*. Extending the literary equivalent (and following Genette (1980)), Monfort also defines possible hypodiegetic levels, as sub-areas of the diegesis; for instance, dream-worlds or simulations of other worlds within the main world. I would extend the concept of hypodiegesis to books and documents that can be read within the game – from the novel entitled *Deadline* found in the text adventure of the same name (which mysteriously has a plot that the player seems to know already), to the books in the library of Phatt Island in *Monkey Island 2: Lechuck's Revenge* (1991), to the notes and books that reveal the story behind the landscape in the *Myst* series. Other examples of hypodiegesis can be mini-games, which can be played and replayed within the gameworld – a darts game in *The Lost Files of Sherlock Holmes: The Case of the Serrated Scalpel* (1992); or a whole other adventure game, as in the case of *Day of the Tentacle*, which included his prequel *Maniac Mansion* as playable game in one of the computers within the game. These mini-games may involve entities that are not manipulable outside of the mini-game (such as the darts) and may be replayed, without the new outcome having an effect on the development of the game. Adventure games are particularly accommodating to mini-games, since part of their nature is to involve concatenated puzzles and riddles, as mentioned above.

The third and last level of interaction, common to most videogames, is the extradiegetic. The actions that the player performs at this level are directives (Montfort 2003); they do not have an effect on the gameworld, but rather refer to the state of the game as a computer program: saving, loading, quitting, turning verbose descriptions on/off, music, sound effects on or off or skipping lines of dialogue or cut-scenes.

The development of strategies to improve direct manipulation in games has affected the features defining adventure games described above. To analyze this development, I have chosen three game engines from some of the most influential game companies in the genre to provide the main examples: Infocom, Sierra Online, and Lucasfilm Games/LucasArts. These engines provided a consistent underlying structure to their interfaces and set a standard for other developers. I have included *Myst* because it brought about significant innovations to the point-and-click interface, as well as *Farenheit* a.k.a. *Indigo Prophecy* (2005), because it explicitly proposed to establish a new standard interface for adventure games.

Text Adventures: Infocom and the Z-Machine

Infocom started off with the release in 1979 of *Zork* ('Infocom: The Next Dimension', 2005), the game they had programmed in a fortnight while they were studying at MIT inspired by Crowther and Woods' *Adventure*. Until its layoff ten years later, the company published 35 adventure games (Montfort, 2003). In order to facilitate the development of the same game for many different platforms, Infocom developed a multi-format emulator called Z-machine Interpretive Program. The emulator would be different for every platform; its purpose was to run the Z-machine virtual processor, to run any game written in the *Zork Implementation Language* (ZIP) ('Infocom: The Next Dimension', 2005). This meant that a game could be programmed once and then run in a variety of platforms.

The interface of text adventures, as their name indicates, is exclusively typed and syntactical, i.e. the interaction requires the user to learn the language and syntax of the commands. The syntax must be correct before even trying to process the command; if it is not, a syntax error message will be displayed (Shneiderman, 2003). In text adventure games, it means that if the user makes a mistake in typing, even if the action is valid, it will not produce a result and will probably waste a turn. Text adventures base their interaction on indirect manipulation, which is not very intuitive, rather than direct manipulation, enacting directly where she wants to move by pressing up or down, as it was the case with arcade games (Shneiderman, 2003).[4]

The prompt of the blinking cursor to start typing can also be daunting; according to Donald Norman (2002), it's the antithesis of user-centred design. The player faces "the tyranny of the blank screen" and she is given "no hint of what is expected.". She has to type the action to tell her character what to do, and wait for a positive response to that input.

Starting to play an adventure game can be intimidating, given the detachment and lack of guidelines of the interface. Take the initial output of *Deadline:*

> You are on a wide lawn just north of the entrance to the Robner estate. Directly north at the end of a pebbled path is the Robner house, flanked to the northeast and northwest by a vast expanse of well-kept lawn. Beyond the house can be seen the lakefront.

The top of the screen displays a location name ("South Lawn") and the time of the fictional world the player has just set in motion (8:00 am). After this, a ">" appears and then the blinking cursor that indicates the player that it is her turn. Though the interface itself does not give any cues or pointers, as Norman complains, the screen is by no means blank. This opening is already giving clues to the player as to what to do: the player character has arrived to a house, she is at the entrance, so the logical course of action to take is to walk up to the door and knock. The interface may be bare, but the game comes with an instruction package that helps the player build the context to the adventure, and starts with a situation that the player is familiar with in real life (arriving to a house). A basic pleasure derived from playing adventure games is precisely being thrown into an unknown environment, forcing the player to explore it to "find her feet" in the gameworld, which includes drawing a map of the different rooms / locations. Text adventure games compensate for the lack of direct manipulation with carefully written descriptions, which is what gives the cues of possible actions.

It is false that anything can be typed, only certain commands will receive a positive response from the parser. There are no cues inside the game for the player to know what kind of syntactical structures will be valid. In one Infocom manual, they explain that in the game "you type your sentence in plain English each time you see the prompt (>). [The game] usually acts as if your sentence begins "I want to…" (*Wishbringer* manual, 1986). In earlier videogames like *Adventure*, or non-commercial works of interactive fiction, however, there are no instruction booklets, so the player would keep on using

[4] See also Game Ontology, http://www.gameontology.org/index.php/Indirect_Manipulation

trial-and-error until something worked or she ran out of patience. Infocom's manuals usually included a list of only some of the verbs that are recognized by the game, inviting the player to experiment to find more, and giving a sensation of false liberty when typing the commands. Trial-and-error was therefore encouraged, which could also foster frustration if the player could only think of one way to express what she wanted the player character to do.[5]

The inclusion of dialogue was an early issue of adventure games. Zork avoided it completely, preferring to implement an RPG-like combat system to interact with the few NPCs. On the other hand, Infocom's *Deadline* was the first game that included a game parser to converse with NPCs. This mechanic was necessary because the player character is a detective to solve a case, and therefore must be able to cross-question witnesses and suspects. This is an example of an exchange in the game:

```
> ask mrs robner about mr robner.
"I loved my husband, no matter what you think. I am very sorry to have lost
him." The speech is almost a set piece, and not too convincing.
> ask mrs robner about the will
Which will do you mean, the new will or the present will?
> ask mrs robner about the new will
"If there is a new will, I certainly don't know where it is."
> ask mrs robner about the present will
"Marshall takes care of that sort of thing. The will, I suppose, leaves the estate
to George and me, but I can't be sure."
```

Even though the parser is very limited, the player can ask rather specific questions, and the game will even give hints about what can those questions can be. However, the player cannot input questions as if she was talking to a Weizenbaum's *Eliza* – the dialogue topics are always limited to those that are relevant to the game, or else the NPC will ignore your question, or the system will inform you that there is no object that corresponds to the topic you're talking about.[6]

The dialogue system adds yet another voice to the many that the player reads and types. Text adventures build a polyphonic environment – since it all appears in text, all the different levels of interaction share the same channel to communicate with the player. Nelson (2001) identifies those voices as that of the player (as the human who types and reads), the protagonist (the main character, who is usually the player character) and the narrator (the voice informing the player what the player character sees, feels and, occasionally, thinks). As Montfort (2003) remarks, the narrator can split into a fourth voice, the program itself usually communicating between square brackets,

[5] Of course, many text adventures also have an obscure command, such as the magic word "xyzzy" in Adventure, or "Odysseus" / "Ulysses" in Zork to chase away the Cyclops. These cases are famous, but otherwise their oddity may very well lead the player to give up.

[6] For a good tutorial to create NPCs, and a review of different dialogue systems in Interactive Fiction see Short (2003).

as a result of the syntactical mode of interaction. This voice tells the player whether it cannot parse a sentence, asks for clarification as for what item the sentence refers to ("Which will do you mean, (…)?"), or whether a certain action has made the player score any points. This polyphony makes the textual interface even more complex to use, and requires the player to be able to learn to identify the different levels of interaction.

The polyphony also makes it difficult to represent the player character. As Wood (1996) remarks, there are several possible types of player identity in IF (extensible to text adventures), of which the most common is the indeterminate character, usually male. This character becomes a surrogate where the player can project "whatever motives and emotions [she] like[s] onto [her] character" (Wood, 1996). This may seem the easiest character to write, since the designer is ignoring who the player character is. It does not mean that text adventures cannot provide the player with determinate identities, being by gender, by the player's own choosing, or determined by the player's actions (Wood, 1996), but these cases are the least common.

The textual interface makes it more difficult to allow the player to control more than one character during the game, or choose who she wants to control.

> Text adventures depend heavily on description and interaction for their atmosphere. There would be little benefit in having multiple PCs unless those characters were well defined, and their interactions with NPCs suitably varied. This requires enormous effort on the part of a game's authors if the different characters are interacting with the same people and places, since NPCs need different responses for each, and ideally location and object description would vary slightly too.

Wood (1996) contrasts the lack of multiple player characters in text adventures with the triad of player characters in LucasArts' *Day of the Tentacle*. Graphic adventures can communicate visually who the player is controlling and where the character is; their interface design fosters direct manipulation. Nevertheless, text adventures occasionally offer multiple player characters: at a certain point of *The Hitchhiker's Guide to the Galaxy* (1984), the player has to control the other main character of the game, Ford Prefect; in *Suspended* (1983), since your player character is in suspended animation, you are controlling six different robots in six different locations (Wood, 1996).

GRAPHICAL TEXT ADVENTURES

Graphical text adventures coexisted with text adventures until the mid-80s; in them, there would be an illustration for each location. The first graphical text adventure was *Mystery House*, released in 1980 by Ken and Roberta Williams. It had very basic line drawings to display the settings in most of the screen, and the text and prompt in the lower part of the screen. The illustrations helped the player see what the environment was like, though she could not interact with them – it was just a first step towards direct manipulation.

Specific player characters started to be more common with the advent of graphics, be it Bilbo (Melbourne House's *The Hobbit* (1983)); Arthur before he became king in Infocom's last adventure *Arthur: The Quest for Excalibur* (1989); Jim Hawkins in *Treasure Island* (1985), or *Dorothy in The Wizard of Oz* (1985) in Telarium's games. Somehow, providing a visual point of view helped construct a specific identity to the player character, which the player had to re-enact. Apart from the visuals, graphical text adventures offered the same type of parser as text adventures, the manipulation was still indirect so that they were plagued with the same problems, such as finding the right combination of words to solve puzzles, or find one's bearings in the gameworld.

Early Graphic Adventures: Sierra Onlines's AGI Engine

After setting up of the company Online Systems (later called Sierra Online), Ken and Roberta Williams went on to release more graphical text adventures. The breakthrough for Sierra came in 1984 with *King's Quest*. IBM commissioned a game that would make the best of the technical features of the new IBM PCjr.[7] The result was a new game engine called AGI (Adventure Game Interpreter), which brought about a new interface for adventure games. Most of the screen was filled with a graphical representation of the space the player character was in, as in textual graphic adventures, but the graphics were dynamic, and the player character could be directly manipulated with the arrow keys. The navigation was not limited from moving from room to room (or location to location), the player could now navigate within the screen. Maps would still be necessary, but not as indispensable as in text adventures. The manipulation of objects was still syntactical, and dependant on the command line at the bottom. Three of the commands (swim, jump and duck) could either be typed or selected through the game menu; these were the three intransitive verbs in the game (i.e., the commands that did not need an object to be complete). The AGI engine, as a transitional model, presents direct manipulation to navigate, helping the player situate her character in the space, but keeps indirect manipulation (by typing and using menus) to allow the player a greater range of possible actions.

Text was not eschewed from the interface – apart from the command line, the player would read the responses of the parser in a pop-up window in the middle of the screen, which would make time freeze in the diegesis while being read. This window gave more information about objects that could be examined, or messages from the parser telling the player that she could not do that, "at least not now!" indicating that the command was recognized but not successful. Descriptions would make up for the low resolution of the graphics (160x200 pixels, 16 colours). The text descriptions have remained in adventure games, handed down from IF (*Myst* and its acolytes excepted), even after graphics improved – they still include hints and pointers to advance in the adventure.

[7] The ultimate AGI & SCI website http://www.classicgaming.com/agisci/agiinfo.shtml

The graphical user interface also meant that the different levels of interaction were now distinct on the display. The gameworld took most of the screen frame; the command line was outside the gameworld but visible along with it. In order to access the non-diegetic directives, players had to press a button (ESC), that would call up a menu for saving / restoring / restarting / quitting the game. The inventory was also accessed through this menu, situating it at a hypodiegetic level; in both cases, when the menu appeared, time stopped in the gameworld.

The distinction between diegetic / extradiegetic levels was also reinforced by the messages that appeared when the player character died: "We, at Sierra, wish to thank you for playing *King's Quest*. We are very sorry that you did not succeed and hope you will fare better next time. Good luck." This message calls attention to the fictive nature of the game, and to the people who created it.

Sierra's AGI engine resolved the ambiguous polyphony of IF by visualizing the different levels of interaction and making them distinct. The existing overlap between commands that could be typed or selected in the menu was a way to ease the transition from the conventions of text to graphic adventures.

The indeterminate player character was not used in graphic adventures. In spite of the low-res graphics, it is possible to identify who the player character is. The character was now distinct, so the player could establish a closer relationship with it through direct manipulation. The protagonists of graphic adventures were also usually associated with a narrative genre, and thus with a set of actions within a generic environment – *King's Quest's* Sir Graham lives in a fairy-tale world, Roger Wilco in *Space Quest* (1986) is a janitor in a spaceship.

The only feature that AGI or subsequent Sierra engines did not address was a dialogue system. The player could only choose which character to talk to, which would display a canned conversation, with no chance for the player to choose the topic of the conversation, so that "talk to" was therefore another form of examining an object (in this case a living entity) for information.

POINT-AND-CLICK: LUCASFILM/LUCASARTS SCUMM ENGINE AND SIERRA'S SCI ENGINE

Lucasfilm Games was the videogame subsidiary of Lucasfilm Ltd., created in 1982; it changed its name to LucasArts in 1990. It started making adventure games with the movie adaptation of Lucasfilms' movie *Labyrinth* in 1986. The company released *Maniac Mansion* the following year, which used the SCUMM game engine (Script Creation Utility for *Maniac Mansion*). This engine, and its subsequent versions, was used in all their adventure games until *The Curse of Monkey Island* in 1997.

The SCUMM engine presented a similar layout as Sierra's AGI, with the gameworld depicted on the top half of the screen. The difference was that at the bottom were listed the verbs that could be used in the game (no more, no less), so that the player did not have to guess which exact word would have to be used, nor look at the word list outside the game. The list of inventory items the player character was carrying appeared next to the commands. The layout thus reinforced the mechanics of the game – on the one hand exploring the space, on the other, the verb + object mechanic. The extradiegetic level was kept out of the screen, as in the AGI engine – the directives could only be called with a function key (F5), which is also separated from the keyboard shortcuts of the game.

The main innovation of the SCUMM engine was that the only input method of the game was moving a pointer, either with a mouse or the arrow keys. The widespread use of the mouse in home computers soon made the point-and-click interface the most popular input device for adventure games.[8] The player moves the pointer and clicks over both the gameworld and the menus to make the desired command. Instead of typing, she can click on the verb list first, and then on the inventory or gameworld object she wants to use. The interface combined direct and indirect manipulation; the pointer moves analogously to the mouse, which constitutes direct manipulation, but the instructions to the player character are still syntactic. The way that the syntax is restrained is what makes the interaction easier: the player can only use the verbs and objects on the screen, and some actions are interpreted automatically for the player. For example, the act of clicking on the place of the gameworld where the player wants the character to go is interpreted as "Walk to."

The player character remained distinct, it could even refuse or complain about the commands instead of giving an error message. The parser of text adventures could be personified and be sarcastic (remember "I do not think the lantern would agree with that"); now the character on the screen can turn to the player and talk to her directly. Thus, the error messages from the parser became personified, sustaining the fiction of the gameworld.

The interaction is considerably simplified, diminishing the hair-pulling caused by trying to find the right word for what the player wanted her character to do, even if she knew what had to be done. On the other hand, giving the player the verbs that could be used, and reducing them, also favours the combinatorics of the verbs on the screen with the inventory or the gameworld as a valid mechanic to solve puzzles, instead of trying to use logic and/or come up with creative solutions.

LucasArts' graphic adventures had an effective – though not very sophisticated – menu-driven dialogue system. Most times, the user just had to go through every single option in the menu to have all the possible conversations. Though this is still canned dialogue, at least it gave the chance to choose what to say, as well as providing the basis for one of the most ingenious mechanics to make your character "learn." In *The Secret of Monkey Island*, the player character Guybrush Threepwood has to learn how to sword-fight; the secret to win is not being skilful, but being good at insulting your opponent to undermine their confidence. Guybrush must learn all the insults and their respective comebacks; he has to fight pirates, and let him-

[8] This also facilitated porting these games to consoles, since the keyboard input had made adventure games an exclusively home computer genre.

self be insulted and beaten, to then have those insults appear in his dialogue menu, then he uses the insults with other pirates to learn the replies. Thus the player has to earn each possible line of dialogue by fighting in order to become a sword master.

Later versions of SCUMM reduced the number of verbs (Give, Open/Close, Pick Up, Look at, Talk to, Use, Push/Pull), and included a context-sensitive cue, so that the verbs that could be used with a certain object would be highlighted when the mouse hovered over it. On the other hand, these cues also curtail the range of actions possible in the game. Usability finished off one of the basic premises of text adventures, which was giving apparent freedom of agency to the player.

The reduction of usable verbs came to a minimum both in the Sierra and LucasArts engines in the early 1990s. Sierra's SCI (Sierra Creative Interpreter), which substituted AGI, introduced the point-and-click system with *Leisure Suit Larry III* (1989), and eliminated the command line. The verbs were reduced to Walk, Look/Examine, Use / Pick up, Talk, selected inventory object and access inventory. The SCUMM engine also changed the interface in *Sam 'n Max Hit the Road* in 1993, leaving out menus, letting the gameworld fill up the screen, and reducing the commands to basically the same list. The syntactic commands disappeared, and though the manipulation of the pointer was direct, the actions in the gameworld were still indirect. This again enhanced the usability of the game, but it also reduced most of the verbs to "use." Verbs were represented by icons instead of words, so that they referred to as many related actions as possible. Some games would still have actions that were character-specific, for instance, Sierra's Leisure Suit Larry would have a kiss and zipper icon (whose meaning would change depending on context); Ben, the protagonist of LucasArts' *Full Throttle* (1995), could kick (which is essential if you are the leader of a motorcycle gang). These special actions helped define the character, though they were also rare.

Myst's minimalistic interface

Cyan Worlds' *Myst* was released in 1993 – just as both Sierra and LucasArts were refining their point-and-click interfaces – and revolutionized adventure games. It had beautifully rendered landscapes filling the screen, as well as a context-sensitive point-and-click interface. The game invited the player to explore a fantasy environment at leisure, pacing down adventure games even more. The game also slowed down considerably due to the absence of NPCs to interact with,[9] avoiding awkward behaviours that would shatter the illusion of being immersed in the fictional world. The lack of NPCs, as well as the first person point of view, took the identity of the player back to the indeterminate protagonist of most text adventures. There was no inventory, though the player could take some objects from one location to another. On the other hand, the point-and-click interface allowed direct manipulation, pushing but-

[9] There would appear occasional NPCs in later instalments of the saga, though the player still could not interact with them.

tons, or dragging objects by moving the mouse. *Myst* incorporated in its interface most of the features of direct manipulation, as proposed by Shneiderman – ease of use, visual representation of the objects to be manipulated, "rapid, reversible, incremental actions," whose form of input was in some form analogous to how those actions were performed in the real world; the whole system made error messages such as "you can't do that" superfluous (Shneiderman, 2003). All these elements favoured the feeling that the player was actually in the gameworld, as well as attracted new players who preferred not to act as the default player character in graphic adventures. There were no descriptions of objects, which may have broken the illusion of physical immersion, though there was still a good amount of reading. The player had to reconstruct the backstory of the gameworld he had been thrown into by reading books and notes scattered over the different locations.

Myst offered an enticing environment, which solved many of the frustrating issues of adventure games simply by avoiding them. The *Myst* series certainly revolutionized adventure games in many ways, especially by following the tenets of direct manipulation. It was also so minimalistic that it became very difficult to change the interface model without expanding it. By leaving out all verb mechanics, it eliminated the cornerstone of earlier adventure games, which was verb-oriented agency through a wide range of actions. For almost a decade after the release of *Myst*, adventure games wavered between re-hashing the Sierra/LucasArts model of interaction (*Broken Sword: The Shadow of the Templars* (1996)) and *Myst*-like environments (*Syberia* (2002); *Aura: Fate of the Ages* (2004)). There are honourable exceptions, such as *Bad Mojo* (1996) where the player character is a cockroach and navigation is the only control available, and *The Last Express* (1997), another point-and-click adventure which made a brave attempt to incorporate real time into adventure games.

ADVENTURE GAMES FOR THE NEXT GENERATION CONSOLES: GESTURAL INTERFACES

The adventure game genre is still a predominantly home computer genre. Only some of the LucasArts titles were released for consoles; the rise of videogame consoles during the 1990s could also account for the steady decrease of adventure games sales. Also, though it has taken time for console-type controllers to latch on with home computers, games are increasingly optimizing their controls depending on which peripheral the player is going to use. Current interface innovation in adventure games is now propitiated by the incorporation of game controllers to PCs, as well as new input devices in consoles, such as touch screens.

The game *Farenheit* (Europe) / *Indigo Prophecy* (US) (2005) attempted to design a new interface for adventure games based on the game controller. Presented by its developers as "interactive drama," they admit that it belongs to the adventure game genre (McDonald, 2005). The interface evokes a movie being shown on television, with black bands along the top and bottom of the screen. The player needs a game controller with two analog sticks to play –

with one she will move the selected player character, with the other she selects the action she wants the character to perform, in the form of a movement/gesture. These actions are always determined by context, and the way the player performs them is by moving the analog stick in a way that imitates what the character will do. For instance, in a kitchen she has to move the analog down to open a fridge, or up to push a cupboard. If the action is an effort the character had to make, the player also had to make an "effort" with the controls; for example, pushing the left and right shoulder buttons alternatively, rhythmically and fast to run or swim. In all these cases, this gestural interface follows the tenets of direct manipulation by trying to make the movements in the controller imitate the actions in the game, as well as by giving visual cues to the player to how to perform gestures in the form of animated icons.

Farenheit / Indigo Prophecy does not particularly encourage exploration of the space – the puzzles are time-sensitive, so that if the player takes too much time in one location, for instance, the player character may be caught by the police and the game will be over. On the other hand, it encourages replay to solve a problem. For instance, in the opening scene, the player character has just killed a man for no apparent reason. The player can opt to go out covered in blood and be immediately identified as the prime suspect, or wash himself, hide the body in a cubicle and wipe the floor, to buy some time before they find the corpse. This brings back the exploration of possible actions that characterized text adventures, and is also the basis for the overall structure of the game, by which there are multiple ways to traverse the game by choosing different actions.

The gestural interface of *Farenheit / Indigo Prophecy* brings back many of the essentials of adventure games. Though the gestures themselves turn repetitive after playing for a while, it brings back the illusion of agency in the world by offering at least two or three possible actions per active area.

Interactive drama seems to be the next offshoot of adventure games with games such as *Farenheit / Indigo Prophecy*, as well as experimental videogames such as Mateas and Stern's *Façade* (2005). The move into direct manipulation is even more evident in this last case, because it has been adapted to use Augmented Reality interface (Dow *et al.*, 2007). In Augmented Reality the interactor sees the characters superimposed on an actual room through a head-worn display, and interacts with them by walking around, making hand gestures and talking.

Interfaces encouraging direct manipulation of the gameworld keep appearing in adventure games, as was the case of *Another Code* (Europe) / *Trace Memory* (US), for Nintendo DS. In order to solve some of the puzzles of the game, the player has to blow off dust by blowing on the console's microphone, "scratch" surfaces with the stylus, and even close down and open the DS again to use an ink stamp. As Marek Bronstring (2006) remarks, these are novelty puzzles, whose novelty will wear out as more games use these gestural mechanics – what is interesting is how they turn everyday actions into a fun activity, and how they encourage game designers to come up with new ways to use the physicality of the controller.

CONCLUSION

Throughout this chapter, we have observed the evolutional search of direct manipulation in adventure games. Giving the player cues about what to do and facilitating the input are first steps to facilitating the illusion of immersion in the gameworld. Error messages, which remind the player of the program behind the game, disrupt this illusion. On the other hand, making error messages dramatic, as in having the player character refuse to do what the player commanded, is a way to ameliorate this disruption.

In the process of transformation of the adventure games interface, we see how new models do not automatically cancel out previous ones. Each interface –textual, menu-driven, point-and-click, gestural – has its strengths and weaknesses in relation with the different features that characterize adventure games. The choice of what interface model will be used may depend on the type of experience the designer wants to create for the player. A good example of this is Nick Montfort's *Ad Verbum* (2000), an interactive fiction piece where the challenges are based on riddles and puns. Some of the puzzles are entirely based on words, as in some rooms where all the descriptions start with the same letter (e.g. "s"), and the commands must start with the same letter in order to be effective.

Adventure games are still being released using old engines in non-commercial or independent games. There are thriving communities of amateur adventure game developers, using for instance Inform, a programming language to make adventures for the Z-Machine, developed by Graham Nelson (Nelson, 2001) or Adventure Game Studio, a development environment for point-and-click adventure games. Even though adventure games are usually not released as AAA commercial games any more, they are still very popular in Europe, where new games are still released every year by companies such as Péndulo Studios in Spain, or MC2-Microïds in France. In Japan, the genre is known as "visual novels," and has always been alive and kicking, with series such as *Phoenix Wright* (2005), originally released for GBA and now remade for the rest of the world for the Nintendo DS. In the US, where most of the engines listed above were developed, adventure games are now seeing a revival with companies such as Telltale Games, which not only develop their own adventure games, releasing them as instalments, but also have started distributing European adventure games.

Adventure games also hold strong ties with less commercial, innovative forms of digital storytelling and electronic literature. The versatility of the adventure games genre is remarkable – from commercial to fan development, from games to experimental storytelling. Contrary to popular belief, the genre of adventure games is still alive and seeking innovation in game design, as well as in constructing fictional worlds.

REFERENCES

Aarseth, E.J. (1997). Cybertext : Perspectives on Ergodic Literature. Baltimore, Md.: Johns Hopkins University Press.

Bronstring, M. (2006). A vision for adventure games on Nintendo Wii [Electronic Version]. Adventure Gamers. Retrieved 31st August, 2006 from http://www.adventuregamers.com/article/id,666

Crawford, C. (2003). Chris Crawford on Game Design. Indianapolis, Indiana: New Riders Publishing.

Crawford, C. (2005). Chris Crawford On Interactive Storytelling (1st ed.). Berkeley, Ca: New Riders.

Dow, S., Mehta, M., Harmon, E. MacIntyre, B., Mateas, M. (2007). Presence and Engagements in an Interactive Drama. In Proceedings of ACM CHI 2007 Conference on Human Factors in Computing Systems 2007, 1475-1484.

Game Ontology Project (2005). [Electronic Version] Retrieved August 31, 2006 from http://www.gameontology.org

Genette, G. (1980). Narrative Discourse : An Essay in Method. Ithaca, N.Y.: Cornell University Press.

Infocom: The Next Dimension. (2005). Retro Gamer, 1(10), 30-41.

MacDonald, L. (2005). Quantic Dream - David Cage and Guillaume de Fondaumiere interview [Electronic Version]. Adventure Gamers. Retrieved August 31, 2006 from http://www.adventuregamers.com/article/id,554.

Montfort, N. (2003). Twisty Little Passages : an approach to interactive fiction. Cambridge, Mass.: MIT Press.

Murray, J. H. (2001). Hamlet on the Holodeck : the Future of Narrative in Cyberspace. Cambridge, MA: The MIT Press.

Nelson, G. (2001). The Inform Designer's Manual: Interactive Fiction Library.

Norman, D.A. (2002). The Design of Everyday Things (Reprint ed.). New York: Basic Books.

Provinciano, B. (2003). The Ultimate AGI & SCI Web Site [Electronic Version]. Retrieved August 31, 2006 from http://www.classicgaming.com/agisci/

Rollings, A. & Adams, E. (2003). Andrew Rollings and Ernest Adams on Game Design. Indianapolis, IN: New Riders Publishing.

Short, E. (2003). NPC Characterization [Electronic version] Retrieved 31 August, 2006 from http://www.mindspring.com/~emshort/NPC4.htm

Shneiderman, B. (2003). Direct Manipulation: A Step Beyond Programming Languages. In N. Wardrip-Fruin & N. Montfort (Eds.): The New Media Reader, 486-499. Cambridge, Mass.: MIT Press.

Wood, J. (1996). Player Character Identity in IF. XYZZY News, (9), 7-11.

Zagal, J., Mateas, M., Fernández Vara, C., Hochhalter, B. & Lichti, N. (2005). Towards an Ontological Language for Game Analysis. Paper presented at the Digital Interactive Games Research Association Conference (DiGRA), Vancouver, B.C., Canada. Retrieved October 17 2007, from, http://www.digra.org/dl/db/06276.09313.pdf

GAMES

Adam Cadre. (1998). Photopia. Self-published. (Z-Machine)
Beam Software. (1983). The Hobbit. Melbourne House. (Various platforms.)
Black Isle Studios. (1999). Planescape: Torment. Interplay Entertainment. (PC).
Capcom Co. Ltd. (2005). Phoenix Wright: Ace Attorney. (NDS).
Challenge Inc. (1989). Arthur: The Quest for Excalibur. Infocom (Z-Machine).
CING, Inc. (2005). Another Code (a.k.a. Trace Memory). Nintendo. (NDS).
Crowther, W. & Woods, D. (1975-76). Adventure (a.k.a. Colossal Cave) (PDP-1).
Cyan Worlds Inc. (1993). Myst. Brøderbund Software. (Amiga / PS / Saturn / PC).
Data East Corporation. (1996). Magical Drop II. (Neo Geo / SNES).
Electronic Arts. (2005). From Russia With Love. (GC, PS, XBox).
Infocom. (1980). Zork I: The Great Underground Empire. (Z-Machine).

Infocom. (1981). Zork II: The Wizard of Frobozz. (Z-Machine).

Infocom. (1982). Deadline. (Z-Machine).

Infocom. (1983). Suspended. (Z-Machine).

Infocom. (1984). The Hitchhiker's Guide to the Galaxy. (Z-Machine).

Infocom. (1985). Wishbringer. Infocom. (Z-Machine).

Konami. (2001). Metal Gear Solid 2: Sons of Liberty. Konami Computer Entertainment Japan. (PS2).

LucasArts. (1993). Day of the Tentacle (a.k.a. Maniac Mansion II). (PC / MAC).

LucasArts. (1993). Sam 'n Max Hit the Road. (PC / MAC).

LucasArts. (1995). Full Throttle. (PC / MAC).

Lucasfilm Games. (1986). Labyrinth. Activision Publishing. (Apple II / C64 / MSX).

Lucasfilm Games. (1987). Maniac Mansion. (PC / Amiga / Apple II / Atari ST / C64 / NES).

Lucasfilm Games. (1990). The Secret of Monkey Island. (PC / Amiga).

Lucasfilm Games. (1991). Monkey Island 2: LeChuck's Revenge. (PC / Amiga / Mac).

Maxis Software Inc. (2000). The Sims. Electronic Arts. (PC).

MC2-Microïds. (2002). Syberia. The Adventure Company. (PC / PS2 / Xbox).

Massachusetts Institute of Technology (1978-79) Zork. (PDP-10).

Montfort, N. (2000). Ad Verbum. (Z-Machine).

Mythos Software Inc. (1992). The Lost Files of Sherlock Holmes: The Case of the Serrated Scalpel. Electronic Arts Inc. (PC).

Namco. (1979). Pac-Man. Midway. (Arcade).

Namco Ltd. (1999). Soul Calibur. (DC).

Nintendo Co. Ltd. (1986). The Legend of Zelda. (NES / GBA).

On-Line Systems. (1980). Mystery House. (Apple II).

Procedural Arts. (2005). Façade. (PC).

Pulse Entertainment. (1996). Bad Mojo. Acclaim Entertainment Inc. (PC).

Quantic Dream. (2005). Fahrenheit. (a.k.a Indigo Prophecy) Atari Inc. (PC / XBOX / PS2).

Rare Ltd. (1997). Goldeneye. Nintendo of America. (N64).

Sierra Online. (1984). King's Quest. (Amiga / Apple II / Atari ST / PC / Sega Master System).

Sierra Online. (1986). Space Quest: The Sarien Encounter. (Amiga / Apple II / Atari ST / PC).

Sierra Online. (1989). Leisure Suit Larry III: Passionate Patty in Pursuit of the Pulsating Pectorals. (PC / Amiga / Atari ST).

Sierra Online. (1991). Leisure Suit Larry 5: Passionate Patti Does a Little Undercover Work. (PC / Amiga / Mac).

Streko-Graphics Inc. (2004). Aura: Fate of Ages. The Adventure Company. (PC).

Square Co. (1997). Final Fantasy VII. Square / Eidos (PS / PC).

Smoking Car Productions, Inc. (1997). The Last Express. Broderbund Software, Inc. (PC / MAC).

Telarium. (1985). The Wizard of Oz. (PC / C64).

Telarium. (1985). Treasure Island. (PC / MSX2).