



AUTHOR(S):

TITLE:

YEAR:

Publisher citation:

OpenAIR citation:

Publisher copyright statement:

This is the _____ version of proceedings originally published by _____
and presented at _____
(ISBN _____; eISBN _____; ISSN _____).

OpenAIR takedown statement:

Section 6 of the "Repository policy for OpenAIR @ RGU" (available from <http://www.rgu.ac.uk/staff-and-current-students/library/library-policies/repository-policies>) provides guidance on the criteria under which RGU will consider withdrawing material from OpenAIR. If you believe that this item is subject to any of these criteria, or for any other reason should not be held on OpenAIR, then please contact openair-help@rgu.ac.uk with the details of the item and the nature of your complaint.

This publication is distributed under a CC _____ license.

A Case Study into the Accessibility of Text-Parser Based Interaction

Michael James Heron
Robert Gordon University
Aberdeen, Scotland
m.j.heron1@rgu.ac.uk

ABSTRACT

The academic issues surrounding the accessibility of video games are reasonably well understood although compensations and inclusive design have not yet been comprehensively adopted by professional game developers. Several sets of guidelines have been produced to support developers wishing to ensure a greater degree of accessibility in their titles, and while the recommendations are broadly harmonious they only address the issues in isolation without being mindful of context or the subtle relationships between interaction choices and verisimilitude within game interfaces. That is not to denigrate the value of these resources, which is considerable – instead it is to highlight a deficiency in the literature which can be addressed with reflective case studies.

This paper represents one such case study, aimed at addressing accessibility concerns within interactive text interfaces. While the specifics of this paper are aimed at multiplayer text game accessibility improvements, it is anticipated that many of the lessons learned would be appropriate for any environment, such as command line interfaces, where the accessibility of written and read text is currently suboptimal.

Author Keywords

Case study; video games; accessibility; text games; command line interfaces; cli; mud; multiuser dungeons

ACM Classification Keywords

H.1.1.2: User/Machine Systems.

INTRODUCTION

The relatively unhappy status of accessibility in video games is now academically understood [1] [17] but not well addressed in mainstream game titles [6] [17]. We are

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

EICS'15, June 23 - 26, 2015, Duisburg, Germany

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3646-8/15/06...\$15.00

DOI: <http://dx.doi.org/10.1145/2774225.2774833>

however now at the stage where disabled gamers and their advocates make representations at the Penny Arcade Expo (PAX) and the Game Developer Conference (GDC) as well as other gaming events. Accessibility in video games is considered of sufficient importance to merit its own bullet-points in conference calls for papers in both video games and accessibility. Despite the increased profile of the topic, the titles produced by developers continue to frustrate with their lack of accessibility, both physical and sociological [9]. The sets of ‘academic researcher’ and ‘professional game developer’ rarely intersect, and there is difficulty communicating to time-pressed developers the importance of the topic. Video games represent both a significant business domain, worth an estimated \$55bn in 2015¹, and an important source of cultural and social capital. The AbleGamers foundation estimates around 33 million disabled gamers in the US alone, representing a very significant demographic currently ill-served by many mainstream titles.

There is much well-meaning advice available, and a number of useful resources that help highlight issues and identify possible compensations. However when considered in isolation the unthinking implementation of guidelines can end up creating a situation where accessibility is lessened [11] as a result of being unaware of the subtle implications of design decisions. Accessibility is a complex issue, complicated further still by the often subtle inter-relationships of afflictions [13]. Minor issues may exist in combination – each acting together to create problems while the individual suffering may not see themselves as having any significant accessibility requirements [13]. Certain conditions too have symptoms which vary in intensity on a day to day basis [12]. The tension between being fully cognizant of these issues and yet still tractable to a non-specialist audience is considerable.

That is not to say these endeavors are futile, for they manifestly are not. However, it is necessary for these guidelines to be understood within the relative complexity of an implementation context. Every game is going to bring with it a different set of accessibility considerations.

¹ <http://www.gartner.com/newsroom/id/2614915>

As such there is benefit to be had from the provision of exemplar case studies that range from the mundane to the miraculous.

One such unusual context is that of text-based games such as those represented by classical interactive fiction and multiuser dungeons. Such games have long been considered dead as a commercial genre [7]. Modern development tools such as Inkle, deployment contexts such as mobile devices and a die-hard group of hobbyists and electronic literature devotees have converged to create an environment where such games can flourish provided they are developed on a hobbyist and indie basis. However, while the games themselves may struggle to find significant levels of popularity, the interaction style symbolized by parser based text games has deeper applicability – many of the interaction difficulties one might encounter within such games are also encountered in standard command line interfaces.

In this paper, we discuss Epitaph Online, a multiplayer, hobbyist text game set in the grim darkness of the zombie apocalypse. The author of this paper is owner, administrator and lead developer of Epitaph, and works with a small team of developers to expand and enhance the game on an ongoing basis. One of the key projects is that of increasing accessibility, and to that end this author has been engaged in an ongoing, participatory research project aimed at reaching that goal. This paper then is a case study in how that project has been conducted. This has been an iterative process, involving numerous small and large changes over various stages within the project's lifetime. It has not been a formal, discrete research project but a long-term activity which has generated numerous insights into the complexity of implementing effective accessibility within a large and varied text game. The base of players of Epitaph is small in real terms, averaging five players online at peak times. It has at current count seventy-nine 'active' player accounts, where 'active' is defined as a player having logged on for an hour or more within the past two weeks. Usual player patterns in Epitaph are marked by short periods of intensive play, separated by longer periods of inactivity. Some active players however have extremely long accumulated playtime, with our current oldest player having accumulated 4805 hours of playtime since 9/4/2012. The average retained player accumulates roughly 50 hours of play-time, although within the context of huge variation ($N=1660$, $Mean=50.21972$, $SD=452.2$).

The lessons of implementing accessibility in text games are unlikely to be truly generalizable, and encompass a wide variety of issues that can only truly be appreciated as they are encountered in context. It is hoped however that this case study will prove illuminating as to the iterative process of identifying issues, soliciting player feedback, and making adjustments based on that feedback. It is also anticipated that the lessons learned within the development

of these accessibility tools have general applicability to other text-driven domains such as command line interfaces.

CASE STUDIES IN VIDEO GAME ACCESSIBILITY

Many of the accessibility issues that are encountered within video games occur as a result of oversight rather than as deliberately exclusionary design decisions [6]. For many game developers, it's simply not a topic that is considered due to the intense focus on the development of the ludic and narrative elements of complex projects. Accessibility issues are sometimes considered a job for 'someone else' or something that is only important for 'different kinds of software, not games'.

The idea of making a game like 'Rock Band' accessible may seem a task which is impossible by the simple nature of the game. The combination of auditory feedback, visual cueing and physical interaction with a bespoke controller seems to create a situation where the only two possible outcomes are to fail, or to create a game which lacks all of the elements that make Rock Band innovative. And yet, accessible versions of Rock Band exist[16] and while those who are blind may not be playing the same game as those who are not they are still playing a game that they enjoy with sufficient overlap to ensure that they can meaningfully engage in conversation with friends about the title. It is not an appropriate solution to consign disabled gamers to the comparative ghetto of 'accessible video games', although many compelling titles do exist within that subgenre. Part of the appeal of playing video games however is in the social capital it helps generate [13] and disabled gamers are often excluded from this important source of popular culture.

Some accessibility issues are as a result of flaws in coding. As an example of this, consider the PC version of the game Far Cry 3. During its story-based quick-time events, the player is often asked to mash certain key combinations to accomplish the on-screen task. Usually these keys are those that are being used regularly – jump, crouch and use. Mashing keys at all can be difficult for those with physical impairments [6] but an additional problem comes from a bug within the game itself. If those keys are bound to a non-keyboard element, such as a dedicated gaming mouse, no matter how hard they are mashed they won't be recognized within the game. More frustrating still, this is only true of some remapped combinations and so it may take some time for the player to realise that the problem is with the reconfigured controls, not with their speed at mashing. Being able to remap controls and offload them onto dedicated input devices is a key compensation for disabled gamers [6] and this particular flaw requires the player to rapidly remap keys back to their original settings for some portions of the game.

These story-based elements often require a combination of in-game action sequences, key mashing quick-time events, and rapid navigation against a time limit. Checkpoints are provided as progress is made, and failure returns you to the

last point your progress was saved. This creates a tense and frustrating situation where progress must be made towards the narrative goal, but the need to play the game with one control scheme (the in-character first person perspective parts) clashes against the need to reset the game controls so that progress through a quick-time event can be made through inelegant button mashing. It is both an accessibility design deficiency, represented by the need to mash the key, and a bug that impacts on only a small portion of gamers and so has remained unfixed at the time of writing. Far Cry 3 is not a unique title in this respect, but neatly encapsulates many of the issues that so define the lack of accessibility within modern game development.

Each game brings with it its own accessibility issues, due to the unusual need for games to incorporate elements of verisimilitude into their interface designs. Swinging a Wiimote to cast 'Alohamora' at a lock in a Harry Potter game is a qualitatively different activity to pressing the 'cast Alohamora' button on a gamepad. Physically bobbing and ducking with Kinect is qualitatively different from hitting the 'down' key on a keyboard. The first approach helps build immersion through verisimilitude and a kind of physical skeuomorphism. The other creates an abstraction which distances the player from the physicality of the action their character takes. Neither is necessarily better than the other, but they are not meaningless distinctions in the same way that pressing the 'F1' versus the 'F2' key in a word processor usually are. Games must draw us into the narrative, and the interfaces which they use to accomplish this are powerful tools for ensuring our suspension of disbelief and our identification with our avatar [8].

It is this subtle relationship between the interface, the ludic and the narrative that makes games so challenging as vehicles for implementing accessibility. It is this relationship that is least served in guidelines for accessibility – to carelessly implement a suggested compensation may not only result in less accessibility for others as an unexpected consequence, but may also throw off the balance of interaction within the title.

The use of case studies to document accessibility in its ethnographic context is not unknown. Within the broader topic of usability, case studies are more common with much academic literature addressed at performing tear-downs of existing resources. Accessibility case studies of video games are vanishingly rare, with only a few examples that turn up during a literature survey of the area (c.f. [3] and [15]). Given the peculiarities of video game interaction, this is a significant gap in the record. Accessibility case studies are not a replacement for formal quantitative and qualitative evaluation of titles, but neatly encapsulate the key lessons along with the surrounding implications when assessing a given piece of software.

Video game production too adds its own complexity to the task of evaluation– building a modern video game is complex, involving many skills usually over a long period

of time. Indie titles offer more flexibility and scope for experimentation but are still expensive to produce. Developing proof of concept software and then evaluating it is a common practice in computing research which allows for a brisk pace from conception, to implementation, to evaluation of a new technique. This allows for the development process itself to be a topic of evaluation.

The scale of the task when directed towards video games means that this may not be feasible. As a consequence, accessibility analyses tend to be *post-hoc* – useful in determining how well a target may have been met, and valuable in providing a core of 'lessons learned' of a finished project. However, they do not provide an ongoing overview of attaining accessibility as a 'living process'. They do not offer a chance to look at the decisions during development, or the influences that may have argued for or against the implementation of particular features. These issues are important in gaining a fully holistic view of the process, ensuring an opportunity for developers to see exemplars and perhaps adjust their development plans accordingly. Extended, reflective case studies such as this are intended to help bridge this gap to some extent.

INTERACTION IN EPITAPH

Given the relationship between interface and immersion as discussed above, it's important to discuss the way in which Epitaph interfaces with players. This in turn requires a short discussion of the way in which the game handles spatiality. Everything within Epitaph is written in a language called LPC [5] and every program within the game is represented as an LPC object. Some of these objects have no physical presence within the game, as is the case with inherits or general handlers². Others have an environment, which sets their physical relationship to other objects in the game world. Many of these objects are room objects, which represent particular locations. Rooms have relationships to other rooms, which is expressed in terms of which exit can be taken within a room to move to another. Rooms can have many exits, and there is no technical need for fidelity of spatial relationship – it's perfectly possible to go east from one room into another, and then east again to return to the first.

Players within the game are also objects, and the environment of a player is usually a room. Rooms can also contain items, and in this case the environment of the item is the room in which it is contained. A player picking up the item changes its environment to be the player object. This creates three key relationships between objects:

² A handler is an object designed to act as an interface to many kinds of external objects, through the use of an internal API and state handling.

1. In the same environment, where objects are co-located with another common object as their environment.
2. Within an inventory, where an object is located within another object
3. Outside an environment, where objects are not directly co-located.

Players issue written instructions to the game, and the game reveals the outcome through the use of text. Some of this is automatically generated based on context, and some is pre-written as narrative. Input is handled via the expression of a command followed by numerous parameters and separator words. The game's parser is responsible for taking this written instruction and generating the appropriate object references for manipulation within code. Commands in turn fall into three categories:

- Contextual commands, available when the appropriate object is in the player's environment, or within the player's inventory. The ability to 'blow' a whistle for example is contextual, dependent on the player having a whistle available.
- Innate commands, available at all times regardless of environmental considerations. The ability to 'say' something is available regardless of object co-location.
- Special commands, available at all times provided they have been externally added as available to the player. The ability to 'hunt', 'cast' or 'ambush' for example may be commands that a player must purchase before they can be used.

Commands come with patterns which outline the way in which they must be expressed to the parser. For example, a command may have the following pattern:

```
blow    {furiously|angrily|calmly}
[intro] <indirect:object>
```

This provides the structure of a command called 'blow', which can be executed with one of three parameters (furiously, angrily and calmly) with 'into' as an optional bridging word and indirect:object referencing an item in the player's inventory or their environment. The combination of structures permitted through Epitaph's system is complex – the task of writing code on Epitaph is discussed in [5]), with the pattern system covered fully in [5], pages 295-304.

The text presented back to the player is sometimes dependent on the command issued – the parser system also handles output, and special tokens are used to provide what are known as success messages and failure message. The former are used when a command passes all internal checks and triggers some action in the game. The latter are used to deal with issues where commands cannot actually be executed – for example, when applied to an invalid target,

or when necessary equipment is missing. When this occurs, the parser system looks for the next best match for the command it had been provided – some commands have multiple syntaxes, and many commands may share the same name. Simple tokenization rules are used to make sure that text is seen appropriately by all players, with words like 'blow\$s' becoming 'blow' or 'blows' depending on who is looking. Tokens such as \$N and \$I are expanded to the player that triggers the message, and the indirect object with which the player interacted, respectively. The player sees:

```
You blow the whistle, which emits
a whistling sound, as whistles do.
```

Bystanders see:

```
Michael blows the whistle, which
emits a whistling sound, as
whistles do.
```

Other pieces of text are handled entirely by the developer, such as descriptions of objects or flavor text. These take the form of things like standard descriptions of rooms or items. It's possible for these to be dynamically rendered, but more traditionally it is a set piece of text displayed to the player at the appropriate time – such as moving from one room to another.

It is clear that this kind of structure represents the farthest extreme of the relationship between interface and player. There is little verisimilitude here, and the nature of the command system ensures that there is a steep learning curve to those interesting in trying out the game. Syntax issues of command structuring are notorious in text games with open parsers, and the fact that some commands are contextual means that one situation may use the same command name with a completely different pattern. Uncommon verbs may be employed when more common ones are omitted. Help-files may be limited, or consciously omitted for those areas where finding the syntax is part of a gameplay puzzle. The notorious 'guess the syntax' quest is a staple, albeit usually unintentionally, of text game puzzle systems.

However, this represents perhaps one of the best possible examples for an accessibility case study. It is an interaction regime that is distinct from what might be considered mainstream and thus with complexities and interrelationship of elements that other developers may not have spent time considering. It is also an environment that is especially important for the blind, as those with visual impairments find text games accessible in a way to which more graphical titles cannot aspire [7].

ACCESSIBILITY ISSUES IN TEXT GAMES

This interaction regime offers an environment which is close to ideal for those with visual and auditory accessibility issues [7] – with the addition of a screen-reader, the entirety of a game's output can be rendered in a

format that is tractable to those with visual impairments. Sound is delivered as text, and so those who are deaf are not disadvantaged as a result of audio being used for conveying information without appropriate close captioning.

Visual Impairments

The nature of gameplay for most visually impaired players is to experience the game through a dedicated game client. Epitaph is delivered primarily as a telnet stream, and players generally use specialized clients to make use of bespoke MUD protocols such as MXP, MSSP or MSP. Clients for the blind exist, the most significant of which is VIPMud. However, the nature of a telnet stream and the linear recitation of a screen-reader creates several gameplay issues:

1. Screen-readers cannot deal with ornamentation, such as ASCII art.
2. Screen-readers cannot prioritize important information if it occurs after less important information.
3. Too much written content makes it difficult to pick out important information, or is simply cumbersome to experience.

ASCII art is a common feature on MUDs and within interactive fiction. The use of the standard ASCII characters to create graphical representations was very popular in the days of Bulletin Board Systems (BBSes) when modem speeds and computer storage capacities meant detailed graphics were a luxury that most could not afford. The result was a flowering interest in ASCII and ANSI art to produce some element of visual flourish, with some artists in these mediums accomplishing unusually impressive results. However, ASCII art is impossible for a screen-reader to parse, and results in the output of gibberish.

Within games such as Epitaph, the default system is that when a player enters the room they are given several pieces of information, in order:

1. The short description of the room, such as ‘a dark cellar’
2. The long description of the room, usually 50-60 words in length
3. Weather details, if outside
4. The inventory of the room, living or otherwise

However, the linear nature of a screen reader means that if the inventory of the room contains a hostile non-player character (NPC), it may be some time before the player is alerted to the fact they have been drawn into combat because the descriptive elements of the game are encountered first. This is an important playability issue because some NPCs require special strategies or rapid evaluation of risk versus reward. Epitaph also encourages

caution by ensuring that running away from combat is a complex task that can easily go wrong when faced with large numbers of foes – avoiding combat rapidly is important here. By the time the combat begins the player may have lost their chance to escape. Most interactive fiction does not suffer from this issue, as it is single player and largely waits for player input before advancing game-state – the player need not respond to gameplay situations before they are ready to do so. Epitaph however is multi-player and all objects within the game can operate independently of player input.

The ASCII art issue is common to all games offering text as an important channel of output, while the issue of ordering in output elements is one that is largely restricted to games that are not turn-based.

Cognitive Impairments

The issues relating to cognitive impairments can be difficult to generalize, but within the game design of Epitaph there are two key elements of concern:

1. The heavy emphasis on recall rather than recognition, which disadvantages those with working memory impairments.
2. The cerebral nature of many of the game puzzles, which are an integral part of gaining advancement within the game.

Epitaph can in many ways be thought of as a recreational command line interface system and the common grievances regarding ease of use and learnability [2] that go around such systems are true for Epitaph also. Lacking many of the contextual clues as to player attention, such as mouse location, avatar orientation or spatial proximity to objects, it is difficult to offer relevant interaction cues. Within a graphical game, if we look at a particular object from nearby, we may be given a prompt that certain interaction choices are available. Within Far Cry 3, the title discussed earlier in this paper, contextual command cues are displayed whenever we can perform particular actions such as enemy takedowns, or interact with a vehicle. Such cues turn what would be recall into recognition with the attendant decrease in interaction difficulty. Such techniques are not available to the same degree in a game where spatial proximity is limited to either ‘in this room’, ‘in this player, or ‘somewhere else’.

Within Epitaph too are many puzzles, or what are known in the parlance of the game, quests [10]. Quests may involve performing a simple action in a particular location, or involve complex deduction and mental gymnastics. The load some of these place on fluid intelligence is substantial. Those with impairments in their cognitive capacities may find that the mental agility required to successfully complete them is out-with reasonable expectation. Quests are in many cases optional, but confer considerable in-game advantage upon completion such as achievements, points that can be spent on buying special commands, preferment

within the in-game factions, or narrative advancement. While it is tempting to argue that the nature of text-games generally is likely to preclude many with cognitive impairments from becoming too deeply involved, it is the same argument as is used against the blind playing Rock Band. An easy answer, but extremely unsatisfying.

Physical Impairments

Those with physical impairments find their difficulties largely revolve around the need to precisely enter what may be complex commands, often within timed constraints. The nature of the command system means that upon occasion precise instructions may be needed, such as:

- stab zombie 2 with garden fork 3
- get all except clothes&jewellery from corpses

Many of the non-diegetic commands too, those relating to out of character facilities such as scoring, achievement progress or in-game blog readers, have syntax complications commensurate with the complexity of the system with which they are interfacing. To read the game blogs within the game world for example requires the identification of blog and post number, which in turn comes from listing all posts from a blog, which in turn comes from listing all the available blogs. All of these represent a potentially substantial effort for those who may find keying in instructions to be time-consuming. This is also a system which is not tolerant of failure. A typo can render an entire instruction incomprehensible to the parser. While it will try to make suggestions as to what may have gone wrong, it still requires the command to be revised in line with the game error messages. Many clients allow the player to arrow-up and get access to the previously typed command, but this is a facility that comes from external tools and is not supported within the game itself.

EPITAPH COMPENSATIONS

All of these issues introduce complexity of interaction within Epitaph, and a responsible accessibility programme will look to find new areas where improvements can be made. Epitaph has a small player base, as is common for new games of this nature, and this player base contains a disproportionate number of individuals with visual impairments. This is in part because of the fact that all promotional material about the game makes mention of the importance of accessibility to the development team and this in turn attracts attention from members of the demographic most affected by accessibility improvements.

As such the development team has logged a constant stream of suggestions and requests for improvements in the game experience. These have been given a high priority by the developers both in the years leading up to release and the months since the game formally opened. Epitaph development is done on a closed server, and then pushed to the live game in the form of patches. A full list of patches is available, along with the accompanying patch notes for anyone who wishes to see them (available at

<http://epitaphonline.co.uk/patches.c>). Many of these patches contain accessibility fixes, which in turn increases the perceived prominence that the issue gets in the development agenda. This in turn becomes a reason for those with impairments to log on to Epitaph rather than another, less accessible game. Several players have indicated both their respect for the agenda, and revealed that they have used it as a selling point in conversation with their friends as a reason to play the game.

Visual

Many of the accessibility improvements for the visually impaired are aimed at increasing the control players have over what game information is presented to them. Epitaph makes use of a system known as the 'earmuff' to allow players to switch off certain categories of game information if they are not interested in it. During development, this system was expanded to encompass many more game systems so as to allow increased brevity of representation. Some elements of the game are intentionally held off of this system due to the fact that the information provided is too important to allow players to opt-out of receiving it. Many elements of the game world were easily ignored by sighted players, but very awkward for the visually impaired. One significant example of this is the 'linker', which is used to make rooms visible from other rooms – a wide town square for example might be made up of nine rooms arranged in a 3x3 grid. A player moving from one part of the square to another would trigger an event to tell all others in the grid that they have moved. This involved a lot of extraneous text, and so the system was internally re-engineered to allow it to be earmuffed since it offered no gameplay critical information that could not be obtained through other channels.

Other improvements focused around filling in the gaps left by client capabilities. Different game clients offer different facilities, and special clients like VIPMud do not necessarily have the same protocol support as general clients like Mushclient. NPCs in Epitaph can be conversed with through a keyword system, and to aid in navigating what may be several dozen possible communication options keywords are highlighted using either clickable hyperlinks or colour coding. However, some clients have no support for clickable hyperlinks and those with visual impairments have little ability to discern the colour coding. An additional option for highlighting keywords was added, where they were presented encased in braces which allowed for a screen-reader to identify them.

Similarly with the normal presentation of room text – a briefer option was added that players could enable. This provided the player with a room description only if it was different from the last one they had encountered. This allowed for a more elegant presentation of rooms with shared descriptions, such as many of the rooms representing city streets. Wandering around a street of twenty rooms normally results in the room description being repeated

twenty times. An already existing option, called ‘glance’, showed only the short description of rooms without the long description when the player moved. The ‘change’ option allowed for a middle path to be taken.

The issue of linearity in ordering of output was addressed through another option which allowed for the contents of a room to be displayed before the descriptive elements. This allowed for the inventory, and any enemies contained within it, to be presented first to the player so immediate action could be taken as and when needed.

An alt text system was introduced, which allowed for players to set themselves as receiving plain, rather than ornamented text. This removed any attempt at using columns of data, or ASCII characters to improve the aesthetic appeal of text headings – all such text was provided raw to those with the setting enabled.

Reading graphical maps is usually difficult or impossible for those with visual impairments, and the game-world of Epitaph is large. To aid players in navigating the world, a landmark system was added which allowed the player to find themselves in relation to the orientation of prominent locations within the game and be told in what direction they lay.

Epitaph offers many game options, accessed through the use of the ‘options’ command. The range of these is significant, and for new players it’s not immediately apparent which are important from an accessibility perspective. To ease the learning curve here, a ‘screenreader’ command was added to enable the most helpful accessibility options. To ensure that players did not have to find this out within the game themselves, an option to identify as visually impaired was added to the login process too. The screen-reader command earmuffs most extraneous output, switches off the in-game ASCII maps, and otherwise optimizes the game for the blind. Additional coded support was given to deal with the loss of certain channels of information. The in-game ASCII maps, which were switched off through the screenreader command, also gave information about NPCs in surrounding rooms. When the map was switched off blind players were at a gameplay disadvantage. A new command was added that gave a written summary of NPCs that could be seen in adjacent locations, and in what direction they lay.

Support for accessibility has not been restricted simply to ‘causing no harm’. Many of the quests in Epitaph use some form of grid-based graphical representation. For example, Epitaph contains a textual implementation of the games Minesweeper, Bejewelled, and Light’s Out amongst others. These systems, while visually simple for a sighted player to parse, are almost completely inaccessible for those using a screen-reader, as indicated in figure 1. Identifying the spatial relationships between text elements is not a domain in which such compensatory technology thrives.

```

A B C D E F G H
0 # * @ # & # @ +
1 # & @ @ @ & +
2 @ * + * $ # * $
3 $ & * # + & $ @
4 # @ * $ * & + +
5 + & & + * $ # &
6 $ @ * & & + + &
7 $ $ + * * + + #
Code segments decrypted: 0
Hack percentage completed: 0%

```

Figure 1 - Hacking within Epitaph Online

For core game systems, games which used a grid were given an ‘alternate’ system for those with visual impairments. This however represents a tension between implementing new game systems and ensuring accessibility – each secondary method for accomplishing a task was a game system that could have been used in a different context for everyone’s benefit. However, simply removing the grid-based quests would be a considerable loss to the richness and depth of the game. Blind players indicated that they accepted it would not be possible for Epitaph to be 100% accessible to them, but this was an attitude that was not shared by the development team.

Instead, a new experimental system called the ‘soundgrid’ was introduced – this allowed for a special command to be used on game objects to create a sound enabled grid representation of the game state. Two HTML pages were created – one that contained a sonic key, and another that contained a grid which players could use to navigate the soundscape of the puzzle. Each grid element contained a hyperlink that caused the player, in game, to execute the appropriate command for manipulating that grid element. This system has not yet been rolled out to all grid based systems, and evaluation of its effectiveness has been difficult due to geographical distribution, timing constraints and the small number of potential participants. However, it represents an example of innovation within constraints – where text by itself is not fully appropriate, we look to use what tools we can to offer appropriate compensations.

The second experimental system introduced was called narrative combat. Combat on Epitaph consists of linear text for all participants in a battle, indicating what each did each combat round. Battles may involve allies, enemies and innocent bystanders. It can be set to ‘brief’ mode which means that only those attacks that cause damage are reported, but combat between multiple participants soon becomes extremely long-winded and very difficult to follow even for those who are sighted.

Narrative combat worked to summarize combat, indicating the cumulative impact of what happened within the period of time the data was gathered. Instead of one hundred lines

of very repetitive text, players get a short summary paragraph that puts the situation forward as a simple story. Initial reactions were positive on the part of the visually impaired players, although sighted players reported that they didn't really see much point in it. Narrative combat was introduced, as with all the accessibility compensations discussed in this section, as an option – those who did not see the value could continue to use the original system.

Physical

Physical impairments within Epitaph are addressed through the liberal use of the MXP protocol within the game, which allows for hyperlinks to be added to game text. All objects that can be referenced within the game are given context sensitive menus that can be brought up with a right click on the hyperlink. While not all functionality in the game is yet represented through this system, it's possible to play a reasonably intricate session of Epitaph using nothing more than mouse-clicks. To support ease of use in this regard, an option to enable an MXP toolbar upon moving was added – this made available persistent references to the player, the room in which they are standing, and various external game tools. Many of the non-diegetic commands make use of extensive MXP, allowing players to navigate the wiki, blog and forums without typing a single command.

Here however the nature of MUD clients becomes a problem – since Epitaph is delivered primarily over standard telnet protocols, systems such as MXP rely on client implementations before they work. The set of clients that support the protocol is limited, and without the protocol it's necessary to default to writing full commands or making use of the in-game alias system. There is a limit then to what can be accomplished directly within the game.

A secondary solution is through creating aliases. The basic use of aliases is to allow for long, complex commands to be given a shorter synonym defined by the player. In its simplest form it allows one to type, for example, 'a' instead of 'attack all with chainsaw'. However, commands within Epitaph often require parametric configuration based on context – we may not wish to attack all in our room, and we may not wish to use a chainsaw. The alias system is remarkably versatile, allowing for conditions, branches and sequences to be implemented via creating the right patterns. It also allows for parameters to be included and integrated into the alias call itself. It also allows for dealing with uncertainty through the 'it' and 'them' pronouns – these change what they refer to based on the last thing with which the player interacted, such as:

```
alias bc look at corpse; bury it;
```

The system allows for chaining together instructions, separating them via a semi colon:

```
alias loot get all from  
corpses;put them in loot sack
```

We can use `$$` as a token to refer to 'all input that followed the alias', or `$1` to refer to the 'first word to follow the alias'. Wildcards can be used too, allowing for constructions such as `*$2`, meaning 'all words up to and including the second one'. This allows for constructions such as:

```
alias getthing open backpack;get  
*$ from backpack;close backpack
```

This is then used with followup text to provide the specific way it should be used at the time of execution:

- getthing wallet
- getthing cleaver

Aliases in turn can be used in other aliases:

```
Alias stuff getthing loot sack;  
loot; bc
```

The system is more powerful still, allowing for all manner of complex conditions and arguments and parameters to be incorporated. While it takes time to learn, it offers an impressive range of flexibility for those with physical impairments – a good set of aliases can support even very complex activities with a minimum of issued instruction. By trading off some setup time, a player can navigate and explore the game with only a few letter combinations of their own choice. However, here we see the impact of interaction of ailments – those with both cognitive and physical impairments may find the alias system too demanding to master,

Neither MXP or the alias system were originally envisioned as accessibility tools – rather they were general improvements to usability. However, they represent compelling evidence for the argument made by accessibility adherents – accessible interfaces are not just for people with disabilities, but instead make interfaces better for all of us [4]. MXP and aliases in particular are used just as widely by those without impairments as those with – this in turn creates a situation where issues of social stigmatization do not emerge. Many with impairments or special requirements do not wish to mark themselves out as requiring special support [14], and so may shy away from discussing issues of screen-reader configuration on public communication channels. If these are difficult to use, or behaving in a way that is not expected, it is easier to simply play a different game rather than risk people knowing your personal situation. With MXP and aliases, they are core game features and as such can be freely discussed without risking making a revelation with which one may not be comfortable.

Cognitive

Cognitive impairments are the most difficult category to deal with given the nature of text games like Epitaph. Where possible, recall is replaced with recognition, through the use of categorized contextual help that can be viewed at

any time and reset for when the player wishes to experience it afresh.

Unlike many text games Epitaph uses a standard tutorial system upon first login rather than the old fashioned, passive ‘newbie school’ more common to the genre. Players are introduced early to the basics of the game – how to move, how to interact with objects, how to talk to NPCs, and how to fight enemies. This is handled as a partially freeform, but mostly scripted, scenario in which players escape from their over-run flat to the safety of the game’s main starting area. The tutorial is lightweight, limits the amount of information to which the player is initially exposed, and creates a sense of (mostly false) urgency to ensure engagement. Once the tutorial is over, the same basic approach continues with the sandbox game environment – new commands are introduced only when they are appropriate through the use of small contextual pieces of conversation from the game’s ‘Maestro’ - a term used to refer to any non-diegetic communication done through the MUD. When encountering a training object for example, Maestro will tell the player that the object they just found is used for training, and outline the commands used to interact with it. All objects too come with various help files outlining what roles they fulfill and the commands used to accomplish various kinds of interaction.

The contextual help provided is available for players to review at any time, breaking up what is a very large and complex game into manageable chunks. Deeper discussion of concepts and commands is available through the help system, which is also accessible at any time. Upon login, players are also presented with login tips which remind them of game features that may have been forgotten.

As far as is feasible, players are also presented with a structured view of what they have accomplished to date and what they are currently attempting. Progress towards achievements is recorded within the player’s data files and freely visible, although this is not extended to quests because of the difficulty of balancing full revelation of progress against the desire to protect against spoilers [10]. However, all dialog used to introduce a quest is available in the player’s quest log, and the design of quests is such that they tend to be self-contained rather than large and rambling. Generally they have one or two tasks to perform, with the expository text explaining what is to be done and where the player should be investigating. As quests and achievements are completed, these too are recorded in the player’s file. Skill advancements, accomplishments, conversations and such are all available for recall through the use of various ‘history’ commands... the ‘hachievements’ command for example lists the last twenty achievements a player earned. The ‘hskills’ does the same for skill advancement, and also outlines where those advancements came from.

All systems have in-game browsers that allow for players to remind themselves of what options may be available, and

many of these are available in web format also – Epitaph has its own built-in web server and LPC objects can be accessed through a standard browser. Players can browse through factions, professions, achievements, commands, crafting patterns and more. Where possible, these also link to the appropriate help-files and use MXP to limit the amount of typing needed to explore the systems.

In game help uses a pattern matching system to try and recommend appropriate synonyms when a player cannot remember the correct term to use. The help system too outputs web based versions of all help-files for easy browsing. Synonyms are permitted where appropriate, and multiple syntaxes are provided to attempt to capture sensible forms of interaction. There is less emphasis on the traditional CLI philosophy of ‘one and only one’ way of issuing an instruction but the difficulties in working with such interfaces remain. MXP too becomes a useful tool for addressing this issue, as the need for recall is reduced if the protocol is supported. Right clicking on an object will reveal a wide list of interaction opportunities for the player, and these opportunities are usually contextually appropriate – if a player has no access to a particular command, the command will not be presented as an option.

However, one element that remains significant is that many of the game puzzles require an agile mind and good working memory. They may involve complex logic puzzles, careful deduction, or creative leaps of logic. None of these quests are particularly accessible to those with cognitive impairments in fluid or crystallized intelligence. However, these quests are only a subset of a much broader system which includes action oriented quests as well as simple ‘go there and do this’ quests. Here, we acknowledge that only a sampling of the quests may be appropriate for these players, but this remains an area where a keen eye will be kept open for opportunities for improvement.

CONCLUSION

Work on improving the accessibility of Epitaph is ongoing – with a large, varied and ever expanding game it is likely that we will never reach a point where we believe that we have done all we can to create a truly inclusive game environment.

The nature of the text games is such that numerous issues that are unlikely to affect most games are important, and the compensations we have put in place may have little relevance to more mainstream titles. The techniques discussed within this paper to make a more accessible game have only been evaluated in terms of anecdotal player satisfaction. Game development is a fluid process. Epitaph is a multiplayer game, available within a small niche with many games and few players. This in turn creates a tension – issues must be addressed rapidly to retain player loyalty, and this in turn means that it is difficult to properly quantify or qualify the impact that changes may have on the player-base. To keep players, it is necessary to be responsive and this precludes setting up trials and controlling for variables

within an experimental setup. Within a hobbyist game like Epitaph, the time available for development is only that which is otherwise unallocated, and this means development may occur in fitful spurts and starts.

However, within these constraints we have identified a number of valuable accessibility compensations that could also be useful within purely text driven environments such as those of a standard operating system shell. The accessibility of such interfaces receives very little academic attention, and yet within text gaming there is a long, albeit inconsistent, history of identifying issues of poor interface design and addressing them. Alias systems, hyperlinked command sets, and ongoing dynamic hints would go a long way to reducing the physical and cognitive burden that go along with command line interfaces. Allowing ‘silent’ mode options to function at the user rather than command level would be hugely valuable in restricting the flow of information to that which is critical to the user. These are not overwhelmingly complex to implement, and can largely be ‘dropped in’ to an existing text based interface without extensive restructuring. Indeed, the developer interface to the underlying code within Epitaph is Unix-based, and benefits from many of the techniques discussed in this paper.

What this case study is intended to do is not argue for the correctness or effectiveness of the compensations and expansions made, but instead to outline a set of possible solutions to the interaction difficulties that the nature of the environment introduces. Only by viewing these issues in the context of the game itself can the subtle interrelationships that a compensation for one kind of impairment may have on another be effectively understood within a specialized and not well discussed interaction paradigm.

REFERENCES

1. Bierre, K., Chetwynd, J., Ellis, B., Hinn, D. M., Ludi, S., & Westin, T. (2005). Game not over: Accessibility issues in video games. In *Proc. of the 3rd International Conference on Universal Access in Human-Computer Interaction*. 22-27.
2. Durham, A. G., & Emurian, H. H. (1998). Learning and retention with a menu and a command line interface. *Computers in human behavior*, 14(4), 597-620.
3. Gerling, K. M., Schild, J., & Masuch, M. (2010, November). Exergame design for elderly users: the case study of SilverBalance. In *Proceedings of the 7th International Conference on Advances in Computer Entertainment Technology*(pp. 66-69). ACM.
4. Gregor, P., Newell, A. F., & Zajicek, M. (2002). Designing for dynamic diversity: Interfaces for older people. *Assets '02: Proceedings of the Fifth International ACM Conference on Assistive technologies*, 151-156.
5. Heron, M. J. (2010). The Epitaph Survival Guide. Retrieved June, 10, 2013.
6. Heron, M. (2012). Inaccessible through oversight: the need for inclusive game design. *Computer Games Journal*, 1(1), 29-38.
7. Heron, M., (2013). Likely to be Eaten by a Grue. *Computer Games Journal*, 2(1). 55-67.
8. Heron, M.J., Belford, P. (2014). It's Only A Game: Ethics, Empathy and Identification in Game Morality Systems. *The Computer Games Journal*. 3(1). Scotland.
9. Heron, M. and Belford, P. (2014). Sexism in the Circuitry: Female Participation in Male Dominated Popular Culture. *Computers and Society*. 44 (4). ACM, New York.
10. Karlsen, F. (2008). Quests in context: a comparative analysis of discworld and world of warcraft. *Game Studies*, 8(1), 1-18.
11. Milne, S., Dickinson, A., Carmichael, A., Sloan, D., Eisma, R., & Gregor, P. (2005). Are guidelines enough? An introduction to designing Web sites accessible to older people. *IBM systems journal*, 44(3), 557-571.
12. Newell, A. F., & Gregor, P. (2000, November). “User sensitive inclusive design”—in search of a new paradigm. In *Proceedings on the 2000 conference on Universal Usability* (pp. 39-44). ACM.
13. Shaw, A. (2010). What is video game culture? Cultural studies and game studies. *Games and Culture*, 5(4), 403-424.
14. Sokoler, T., & Svensson, M. S. (2007). Embracing ambiguity in the design of non-stigmatizing digital technology for social interaction among senior citizens. *Behaviour & Information Technology*, 26(4), 297-307.
15. Torrente, J., Del Blanco, Á., Moreno-Ger, P., Martínez-Ortiz, I., & Fernández-Manjón, B. (2009). Implementing accessibility in educational videogames with.<eAdventure>. In *Proceedings of the first ACM international workshop on Multimedia technologies for distance learning* (pp. 57-66). ACM.
16. Yuan, B., & Folmer, E. (2008). Blind hero: enabling guitar hero for the visually impaired. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility* (pp. 169-176). ACM.
17. Yuan, B., Folmer, E., & Harris Jr, F. C. (2011). Game accessibility: a survey. *Universal Access in the Information Society*, 10(1), 81-100.