University of Nevada
Reno

Towards Generalized Accessibility of Video Games for the Visually Impaired

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in
Computer Science and Engineering

by

Bei Yuan

Dr. Frederick C. Harris, Jr./Dissertation Advisor
Dr. Eelke Folmer/Dissertation Co-Advisor

May, 2009

University of Nevada, Reno
Statewide • Worldwide

THE GRADUATE SCHOOL

We recommend that the dissertation
prepared under our supervision by

BEI YUAN

entitled

Towards Generalized Accessibility Of Video Games For The Visually Impaired

be accepted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

Frederick C. Harris, Jr., Advisor

Eelke Folmer, Co-advisor

Sergiu Dascalu, Committee Member

Monica Nicolescu, Committee Member

Mehdi Etezadi-Amoli, Committee Member

W. Larry Williams, Graduate School Representative

Marsha H. Read, Ph. D., Associate Dean, Graduate School

May, 2009

To my grandmother.

Acknowledgments

Many individuals have contributed to the creation of this dissertation. This would not have been possible without their help, support, and encouragement.

First, I would like to thank my advisor, Dr. Frederick C. Harris, Jr., for his continuously help and guidance, and providing valuable comments and constructive criticism over the last several years.

Secondly, I would like to thank my co-advisor, Dr. Eelke Folmer, for his inspiring and encouraging guidance, and his countless discussions with me over the years. Without his expertise in game accessibility, I doubt this dissertation topic would have presented itself. I cherish the friendship we have from our never-ending "arguments".

I am very grateful to my other committee members: Dr. Sergiu Dascalu, Dr. Monica Nicolescu, Dr. Mehdi Etezadi-Amoli, and Dr. W. Larry Williams. Each contributed to this experience in numerous ways.

I also want to express my gratitude to two individuals who helped with the implementation of some of this projects, Donald J. Bartley and Manjari Sapre. Their help and discussions were invaluable.

I must thank my dear friends Judy Fredrickson and Molli Luke, who have always been there to help when I need it the most.

I have been lucky to work and study at the Department of Computer Science and Engineering. Their constant effort in providing a good working environment and supporting me with many assistant positions within the department made this process enjoyable and unforgettable.

I am truly and deeply indebted to my husband, Brian Westphal, for his persistent love, support, and help throughout the entire process. Without him, I would never have come this far.

Finally, I dedicate my work to my grandmother, Shue Wang, my parents, Maozi Yuan and Yuqin Hou, and my twin sister, Lei Yuan. Their unconditional love has lightened my way over this long journey. Wo Ai Ni Men!

This work was partially supported by NSF Grant IIS-0738921. April 29, 2009

Abstract

Over the last three decades, video games have evolved from an obscure pastime to a force of change that is transforming the way people perceive, learn about, and interact with the world around them. Video games are not only a popular form of entertainment, but are increasingly being used for other purposes, such as education and health, as well. Despite this increased interest, a significant number of people encounter barriers when playing games, due to a disability.

This dissertation, and our identification of a generalized game interaction model, helps provide an understanding of how video games can be designed and modified to improve their accessibility features. An estimated 11% of U.S. population are discovered to play video games with sub-optimal gaming experience because of a disability. A large number of existing, accessible games have been studied and analyzed to provide insights and understanding as to the importance of encouraging universal access in this field. Though our survey work covered several types of disabilities, the bulk of this dissertation focuses on improving accessibility for the visually impaired. Specific design strategies are illustrated and proven by the development and evaluation of actual blind-accessible games.

Case studies are presented for each of the three games we developed during the research period. We developed the first mainstream game using haptic feedback. The first screen-reader-accessible virtual world interface is built to explore more strategies for developing blind-accessible games. A third game, developed for sighted users, demonstrates that data collected during gameplay can be used for other purposes including improving accessibility in another game (Second Life). Furthermore, user studies were conducted that focus on the enjoyment, educational, and social interaction aspects of these games while evaluating their ease of access.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Video games have become, over the past few decades, a mainstream form of entertainment, currently eclipsing Hollywood's box office sales [148]. According to the NPD Group, currently more than 100 million home video game consoles are present in U.S. households [85] and an estimated 63% of the U.S. population plays video games. About 51% of these gamers play on at least a weekly basis [32]. An explanation for this popularity could be that games provide something that other forms of entertainment, such as books, music, and movies, cannot provide: interaction. Games are also increasingly used for purposes other than entertainment. Educators are discovering the cognitive potential of games in the classroom [42, 63, 66, 118], and many other learning environments, as a viable and efficient platform for education [43, 95, 124]. In addition, games have also been used in medicine [121], rehabilitation [74], religion [125], and politics [89].

Despite the high level of interest in games, a large number of people are unable to play games because of a disability. Based on a variety of studies compiled by the United Nations, it appears that between 10% and 20% of the people in a given country can be considered disabled [28]. As games have grown significantly in complexity over the past decades [13], not only has development become more costly and risky but the difficulty of making games usable and accessible has also increased. The earliest games such as Pong and Space Invaders were relatively easy to control because the complexity of the gameplay was limited by hardware constraints. Limited amounts of processing power and memory simply did not allow for very complex game simulations. Consequently, the simplicity of most games required relatively small amounts of input. Most of the controllers at the time only had one analog input (such as a joystick or rotary dial) and one or two input buttons, making games straightforward and intuitive to play. Usability problems were uncommon, though accessibility problems existed. Deaf players were not limited in any way because the games didn't rely upon one's ability to hear sounds as most games did not have sound at all. The simplicity of joysticks allowed motor impaired individuals to hook up specialized input devices, such as switches, to play games.

However, most games were inaccessible to the blind players, even in the earliest days of video gaming. Globally, it is estimated that more than 161 million people were visually impaired (2002), of whom 124 million people had low vision and 37 million were blind [146]. Because the number of potential gamers with vision impairments is high and because vision impairments generally present critical obsticles to participating in video gaming, we chose to focus this research on the generalized accessibility of games for the visually impaired.

In video games, visual representation are the most common means of conveying information and modern, advanced graphics allow for greater and more detailed articulation of such information. Finding alternative information conveyance mechanisms to replace visualizations, with the goal of being able to share information as well as one can using visualizations, can be incredibly difficult. Though various research projects and assistive technologies to promote game accessibility for the visually impaired have been developed, most existing solutions are similar in some respects, using audio to replace or enhance visual representation. The ultimate goal in this field of research is to develop games for visually impaired people that are entertaining and beneficial in the same ways that sighted individuals enjoy. This dissertation seeks to enable this ambitious goal through the following contributions: i) a general interaction model for video games that allows better understanding of gameplay and interaction requirements for people with disabilities; ii) developing two new blind-accessible games; and iii) introducing new strategies for making games for the visually impaired.

The remainder of this dissertation is structured as follows: Chapter 2 presents basic definitions and background information on games and disability, including a survey of the related literature. Description of strategies of making games accessible are also addressed in details. A case study of a blind-accessible game, BlindHero, is created and discussed in Chapter 3. The process from the initial game design to user study is explained in detail, along with the brand new strategy introduced for developing blind-accessible game. Chapter 4 presents a text-based Second Life client that we developed, TextSL, which allows blind people to access virtual world purely relying on a screen reader. Existing problems with this development are discussed including lacking of meta information. This leads to the final study of creating a game within Second Life to improve virtual world meta information, presented in Chapter 5. We follow the idea of "game with a purpose" and successfully develop a fun game to improve object labeling and taxonomy in virtual world. Conclusions and a look at open problems that have presented themselves appear in Chapter 6.

Chapter 2

Game Accessibility: Current State-of-the-art

Over the past decades, technological advances in console technology, including advances in processor, storage media, memory and graphic cards, have allowed significantly complex gameplay and high-resolution, realistic graphics and sound. As games evolved over the years, the barriers were raised for people with disabilities to play games. For example, the introduction of the CD-ROM allowed game developers to provide audio dialogue with their game, which often replaced traditional text output, making the game inaccessible to those with hearing impairments. Instead of relying purely on visuals, games started to rely upon the ability to hear (explosions or gunfire) and to feel (haptic feedback). Current highly realistic games require more elaborate and precise control schemes and more sophisticated controllers allowing concurrent multi-button inputs. Figure 2.1 shows a brief overview of the evolution of different

Figure 2.1: Controllers: Atari Controller (1980), Super Nintendo (1992), Playstation 3 (2005)

controllers. The Atari controller used in the eighties only had one analog input and one button. The controller that came with the Super Nintendo, one of the most popular consoles in the nineties, had one digital/directional input and 8 buttons. The PlayStation 3 controller, a console that came out in 2007, has a stunning 5 analog inputs and 17 buttons. A consequence with more control options in games is that the player often has to interact with different inputs simultaneously. For instance, a simple tennis simulator such as Pong only allows the player to move the paddle up and down. A modern day tennis simulator such as Virtua Tennis 3 [110] offers significantly more control options to the player such as the ability to move the player on the court, using different tennis strokes (forehand/ backhand), and using spin effects on the ball. All these advances in technologies introduced more and more accessibility problems to games.

We started this dissertation by surveying a large number of accessible games to reflect the current state-of-the-art on game accessibility. In this chapter, Section 2.1 introduces a generic interaction model for games, which we use to identify how a disability affects one's ability to play games. Estimates are provided for the number of people in the U.S. whose disability affects their ability to play games. Section 2.2 surveys accessible games for a number of different types of impairments and game genres and discusses the different strategies used in making them accessible. Section 2.3 summarizes the different strategies that have been identified and discusses our intention and goal for this research.

2.1 Games and Impairments

Before introducing the game interaction model, we provide background information on disabilities and video games, and define consistent terminologies with which we refer to throughout this dissertation.

2.1.1 Video Games

Video games come in different genres, more than 20 of which have been identified [33, 59, 143]. Due to a general lack of commonly agreed-upon genres or criteria for the definition of game genres, the classification of games is not always consistent, and some games may fit into more than one category. Therefore, we limit ourselves to the following eight seminal game genres:

1. First-person Shooter (FPS) games are action games played from a first-person perspective that involves shooting enemies with a range of different weapons. Popular examples include *Doom*, *Quake*, *Halo*, and *Half-Life.*

2. Strategy games are combat games involving resource management and strategy. Players must build and manage units that can be used to harvest resources or attack enemies. Some of these games are played in real-time (RTS) while others are turn-based (TBS). Typically these games are played from a top-down perspective. Popular examples include *Command and Conquer*, *Warcraft*, and *Civilization.*

3. Sports games simulate the playing of traditional sports such as tennis, football or soccer. Examples include *Madden* and *FIFA Soccer.*

4. Role-playing Games(RPG) involves playing a character that goes on quests and grows more powerful along the way. Examples include *Neverwinter Nights* and *World of Warcraft (WoW).*

5. Puzzle games test problem solving skills such as logic, strategy, pattern recognition, and sequence/word completion. Examples include *Tetris* and *Bejewelled.*

6. Racing games emulate driving a car on a race track such as *Burnout Revenge*, *Project Gotham Racing*, and *Super Mario Karts.*

7. Dance/Rhythm games emulate the playing of or dancing to music through custom controllers such as *Guitar Hero* and *Dance Dance Revolution.*

8. Adventure games are characterized by investigation, exploration, puzzle-solving, interaction with game characters, and a focus on narrative rather than sensi-

motor challenges. Popular adventure games include *Myst* and the *Sam and Max* series.

The first games, such as Pong, were limited to be played on arcade machines but nowadays, a plethora of platforms exist that can be used for game playing:

- Consoles: a computer specifically designed for the purpose of playing games. Players typically use a controller to interact with their console. Examples are the Xbox 360, PlayStation 3, and Nintendo Wii.
- Handheld systems: A portable gaming device, similar to a console, that has the controls built into the device. Examples include the PlayStation Portable (PSP) and Nintendo DS.
- Mobile phones typically allow for playing arcade-style games such as Tetris, but because mobile phones have significantly advanced in hardware over the past years, their distinction from handheld systems becomes less obvious. Games on cell phones are typically referred to as casual games.
- Personal computers allow for a variety of usages including playing games. Users typically play games using a mouse, keyboard, or controller.
- Arcade machines can still be found in arcades and they typically offer more sophisticated ways to interact with the game than other gaming systems. For example a racing game may be embedded in a race car with sophisticated haptic feedback.

2.1.2 Classification of Impairments

There are many ways to define a disability. The definition should be broad enough to cover all persons of interest, yet detailed enough to reflect the scope of our research. In the World Health Organization's (WHO) manual - the *International Classification of Impairments, Disabilities and Handicaps* (ICIDH) (1, 2) [145], "an impairment is a disturbance affecting [:] functions that are essentially mental (memory, consciousness) or sensory, internal organs (heart, kidney), the head, the trunk or the limbs; disability is a restriction or inability to perform an activity in the manner or within the range considered normal for a human being, mostly resulting from impairment."In this disserataion, the following disability classification is used based on the ICIDH standards and also in accordance with the writing guidelines for technology and people with disabilities in [20]:

- **Visual impairment** - is the consequence of a certain degree of vision loss such as low vision/partially sightness, legal blindness, complete blindness. Color blindness is considered a visual impairment.

- **Hearing impairment** - refers to complete or partial loss of the ability to hear from one or both ears. The level of impairment can be mild to profound. Deafness refers to the complete loss of ability to hear from one or both ears.

- **Motor impairment** - is a loss or limitation of function in muscle control or movement or a limitation in mobility. Common causes include arthritis, paralysis, cerebral palsy, or repetitive strain injury. Motor impairment may also include difficulties in speech control and the need to use input devices other than a mouse or keyboard.

- **Cognitive impairment** - is a mental and psychological disorder ranging from mental retardation developed during childhood to Alzheimers or senility as a result of aging. People with autism, Down syndrome or other mental retardation are also included. Their main characteristics are impairments in social interaction, impairments in communication, restricted interests and repetitive behavior. Dyslexia and ADD are also common cognitive impairments.

This is not a comprehensive list of impairments, as different categories of disabilities can be defined based on the needs and the targets of a study. For instance, American community survey [2] categorized visual and hearing impairments as sensory disabilities. In this survey, however, we separate visual and hearing impairments as the barriers these players face may vary significantly.

2.1.3 Game Interaction Model

Gameplay and interaction design are two different, but related, concepts. Designing gameplay is all about finding a way to interact with a computer that is fun and engaging. For example, to simulate the game of tennis, the game Pong allows the player to reflect a ball using a paddle, which is considered a fun way of interacting with a computer. Interaction design, however, deals with providing "fun" in the way that the player:

- *Expects* it (usability): If Pong is played with a joystick, the player expects to control the paddle in the most intuitive or simple way, by moving the joystick up and down and not by some exotic combination of pushing buttons. Usability problems in games [34, 37], such as not being able to skip cutscenes or awkward control schemes, may severely impact the commercial success of a game [105].

- *Requires* it (accessibility): Not every player is able to provide input or receive feedback to the extent that most game developers assume they are. Someone suffering from a disability might not be able to see or hear, which is a problem because all games use these mechanisms to provide feedback to players. Someone who is suffering from severe motor impairment, such as paralyzation, might not be able to use an existing controller and requires specific controllers that can be controlled with the mouth, head or feet . These controllers may limit the player's ability to respond as quickly and precisely as most games require them to. Interaction design must incorporate accessibility requirements so players with disabilities can participate.

Because an impairment may affect a player's ability to play games to different extents, a generic interaction model for games is defined, allowing us to identify exactly what kind of barrier a player with an impairment faces. This model was derived by analyzing how a player interacts with a game for a number of different game genres and finding commonalities between the steps that are performed when playing a game. As an example of our analysis and to illustrate how we derived the model, Figure 2.2 shows the interaction of player with three different game genres (FPS, puzzle and racing) broken down into different steps.

	FPS (Quake)	Puzzle (Tetris)	Racing (NFS)
Receive Stimuli	An enemy appears on the screen and sounds of gunfire and explosions are made. Haptic feedback is provided through the controller.	A block falls from the top and the background music's tempo increases to indicate that blocks are stacked too high are falling faster.	The engine noise of an opponent's car is heard to the right and behind and haptic feedback is issued to the control to indicate a slight collision.
Determine Response	The player receives feedback and decides to fire his/her gun.	The player decides to change the block's orientation and position then lets the block continue to fall naturally.	The play may decide to move right to block the competitor while speeding up.
Provide Input	The player physically issues the fire command through the game controller.	The player uses the controller to issue the commands using the digital input pad.	The analog controller is used to steer the car and an analog trigger is used to accelerate.
Repeat	The enemy is killed and the game state changes to reflect that.	A line is cleared and points are increased.	The speedometer and time elapsed update.
	Repeat these steps until the player dies, finishes, or quits the game.	These steps are repeated until the player dies, finishes or quits the game.	These steps are repeated until the player finishes or quits the game.

Figure 2.2: Interaction with a Game

We followed the same standard for the rest of the game genres, and generalized these steps into a generic interaction model for playing games, which serves us later on in our analysis of how players with disabilities are limited in playing games. This model consists of the following three steps:

1. **Receive stimuli.** Games provide stimuli in three different forms: visual, auditory, and haptic. Depending on the type of game, stimuli can be further divided into two categories:

   (a) *Primary* stimuli must be perceivable by the player in order to play the game. Almost all games use visuals as primary stimuli. For example in an FPS game, visuals are used as a primary stimuli and without visual feedback the game cannot be played. Though sound and haptic feedback may be provided, this typically does not provide sufficient information to be able to play the game. For example, the player may hear or feel the presence of an enemy but may not be able to determine the enemy's location. Though a game typically has only one primary stimuli, some games such as dance/music games rely upon music to such an extent that visuals and audio are both primary stimuli.

(b) *Secondary* stimuli is provided as a supplement to a primary stimulus. Being able to play a game does not depend upon being able to perceive a secondary stimuli. In an FPS without sound or haptic feedback, the player can still play the game to a large extent, but the player may suffer from a reduced gaming experience.

2. **Determine response.** Based on the specific set of stimuli that the game provides, the player must cognitively determine which in-game response(s) to provide from the set of available game actions. These actions are specific to the game and are typically defined by the game's genre. For example an FPS may allow the player to navigate his character, whereas an RTS game may allow the player to group units together. The player either chooses a combination of one or more actions or may decide to not provide an action at all.

3. **Provide input.** After deciding which in-game response(s) to provide, the player must physically issue the chosen action(s) through the input device used to interact with the game. Typically games require the player to actuate a physical device such as keyboard, mouse, controller or a steering wheel. Some games may allow voice commands. Input devices can be categorized into two different groups:

   - **Discrete input:** A device that measures discrete input, such as an on-off switch. A keyboard and the buttons on controllers are examples of this type of input.
   - **Analog input:** A device that measures continuous input, such as a mouse or a thumbstick on a controller.

Because the amount of interaction required to control an analog input device is significantly higher than that of a discrete device, precise motor skills are required. Discrete input devices are relatively easy to control if the number is inputs is small. However, as the number of inputs increases (such as on a keyboard with dozens of keys) or when multiple inputs need to be provided simultaneously, it becomes increasingly difficult to provide input efficiently. Controllers typically provide a combination of analog and discrete inputs. Most game controllers have two analog inputs (thumbsticks) and a number of discrete inputs (buttons or triggers).

After performing these three steps, the internal state of the game may change and new feedback may be provided. The steps are repeated until the player wins, loses or quits the game. These steps are illustrated by a message sequence diagram and a state machine in Figures 2.3 and 2.4.

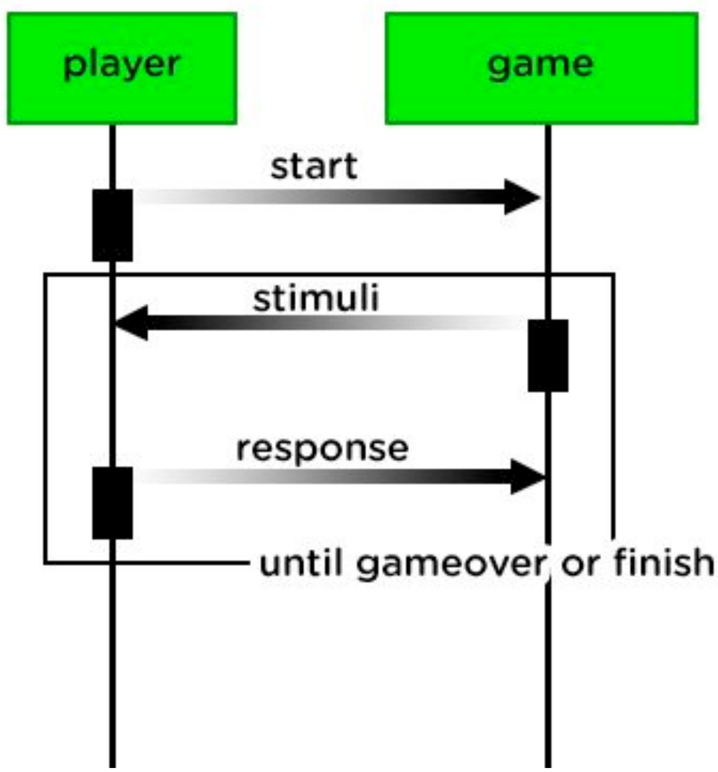


Figure 2.3: Interaction Sequence Diagram

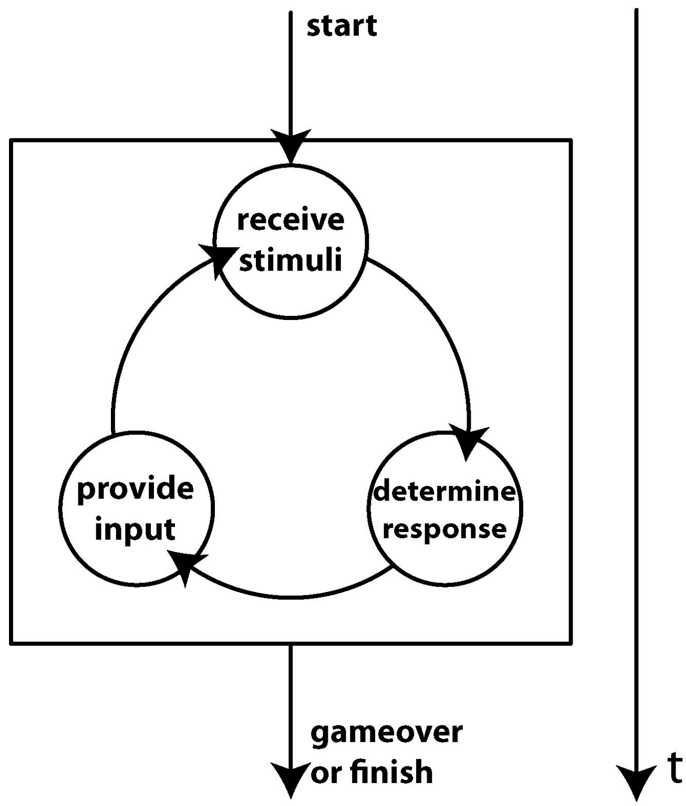


Figure 2.4: Finite State Machine

2.1.4 Playing a Game with an Impairment

Using our interaction model, we can analyze precisely how an impairment affects each one of the steps defined in the model:

- **Visually impaired** players may be unable to perceive primary stimuli. Without this feedback it is impossible to determine what in-game response and what physical input to provide, though these players are cognitively and physically able to perform such tasks.

- **Hearing impaired** players may be unable to perceive secondary stimuli. Because audio is typically used as a secondary stimuli, the auditory impaired player may still be able to play games, but players may experience a reduced gaming experience and may miss out on spoken dialog in cutscenes and ambient sounds like approaching footsteps. There are some games that rely on both audio and visuals as primary stimuli, such as dance/rhythm games. In such cases, the auditory impaired player may be unable to perform steps 2 and 3 of our model.

- **Cognitive impaired** players are primarily affected in their ability to cognitively determine an in-game response. A learning disability, for example, may make it difficult for the player to learn how to play a game and understand what in-game response options the game provides. Limited cognitive reasoning skills may prevent a cognitively impaired player from determining what in-game response to provide based on feedback provided.

- **Motor impaired** players are limited in their ability to provide input physically. A quadriplegic player may be unable to use a regular controller or a mouse and keyboard and often has to rely upon specific input devices with limited input capabilities. Consequently, these players may have a hard time providing specific types of input because they may find it difficult to position a game object precisely, or activate multiple input devices simultaneously, especially when these inputs need to be provided within a certain amount of time.

This analysis reveals that each type of impairment corresponds with exactly one step in the game interaction model. Sensory impairments such as vision and auditory impairments correspond to the first step, cognitive impairments to the second step and motor impairments to the third step of the game interaction model. This strict separation may indicate that, for these types of impairments, fundamentally different solution strategies are needed to make games accessible.

2.1.5 Game Accessibility Statistics

"How many people cannot play video games because of a disability?" is a key question to investigate because, to our knowledge, such data has not been determined in any study and could lead to more awareness. International disability statistics can be found at the United Nations Statistics on Human Functioning and Disability [28]. However, it is not recommended to combine statistics from different countries or regions, given the many differences in survey design, definitions, concepts and methods. As a result, we limit ourselves to compiling statistics for the U.S. because it is the largest gaming market in the world and also has detailed data available on people with disabilities.

According to the census of 2002 [130], about 51.2 million people have some level of disability, and 32.5 million of them are severely disabled. Although not all of them are unable to play games or have a desire to play games. According to the NPD Group, 63% of the U.S. population plays video games [85]. This report is based on online survey responses from 5,039 members of NPDs consumer panel. However, this estimate does not indicate any numbers of players by different age categories. It is not a surprise that elder generations play fewer video games than younger ones, since video games were invented in the early nineteen-seventies and only became commonplace in the eighties and nineties. Another NPD Group study revealed that there are 100 million video game consoles in U.S. households [85]. The largest audience segment is 18- to 34-year-olds, who own 40% of the market, followed by 35+ and 13- to 17-year-olds, each at 18%. This study does not take into account that players may own multiple consoles. A recent survey conducted by Information Solutions Group on behalf of PopCap Games (one of the largest casual game producers) showed that more than one in five (20.5%) players of casual video games have a physical, mental or developmental disability [93]. A total of 13,296 casual game players responded to this survey, with 2,728 respondents identifying themselves as “mildly” (22%), “moderately” (54%) or “severely” (24%) disabled. This study is restricted to the casual game genre and may not be representative for other genres.

All of the aforementioned statistics further illustrate that video games are popular and that the participation of a significant percentage of the population is affected by disabilities. It is difficult to provide an exact number of players that are unable to play games, because those statistics lack the total number of people who play games differentiated across age categories. Detailed analysis of how a disability affects one’s ability to play games is needed. To our best knowledge such statistics have not been determined in any study.

Our analysis in the previous section reveals that a disability limits one's ability to play games to different extents: players with severe vision impairments typically find themselves excluded from playing games, whereas severe auditory impaired people can still play games but typically suffer from a reduced gaming experience. The barriers to playing games may also differ within a particular type of impairment. A severe vision impairment such as blindness restricts the ability to perceive visual feedback completely excluding someone from playing games whereas a minor vision impairment like colorblindness often only leads to minor problems while playing games. It is important to distinguish the different types of barriers disabled individuals face when trying to play computer games. Consequently we define two different types of barriers:

- **Critical:** A disabled player is unable to play the game.
- **Non-critical:** The game is playable, but the disabled player may suffer from a reduced or sub-optimal gaming experience.

Table 2.1 has been constructed by analyzing the data collected for the 2002 census [130]. The number under each age group on the first row indicates the total surveyed population for that group. The 2002 census provides a detailed breakdown of each type of impairment into more specific impairments. For example, motor impairments are broken down into specific physical tasks that cannot be performed (difficulty walking, difficulty grasping objects, *etc.*). Sensory disabilities are subdivided into severe and non-severe categories, allowing us to distinguish players who are blind or deaf from those with less severe vision or auditory impairments. For each impairment subtype in the census data, we first analyze whether it affects the ability to play games and then, if that is the case, we categorize the type of barrier (critical/non-critical) based on our analysis of how a disability affects the ability to play games. Because audio is typically used as secondary stimuli in games, we identified a severe hearing impairment as a non-critical barrier, and a non-severe hearing impairment is not identified as a barrier. To select motor impaired players who are unable to provide primary input, we selected the task “difficulty grasping objects” from the 2002 census data. Unfortunately this task is missing for the data for children age 5 to 14 and instead we selected the related physical tasks such as difficulty eating. These two numbers are listed in parentheses.

In Table 2.1, the total number of people in the U.S. estimated to have their ability to play games affected by a disability is 32,213,000 (∼11% of U.S. population). This can be further detailed into 6,267,000 individuals (∼2%) who are unable to play a game at all and 25,946,000 individuals (∼9%) who are able to play games but suffer from a reduced gaming experience.

Table 2.1: Selected Disability Measures (Number in thousands)

Disability Level from Census	Barrier Type	Age 0-14 60,605	Age 15-24 39,453	Age 25-64 149,031	Age > 64 33,742	All Age 282,831
Motor Impaired - difficulty grasping objects						
severe	critical	(147)	130	310	331	918
not severe	non-critical	(433)	382	283	2,647	3,745
Visually Impaired - difficulty seeing words/ letters						
severe	critical	42	210	762	951	1,965
not severe	non-critical	147	383	2,987	2,852	6,369
Hearing Impaired - difficulty hearing conversation						
severe	non-critical	39	155	418	506	1,118
Hearing or Cognitive Impaired - difficulty with speech						
severe	non-critical	135	123	313	156	727
Cognitive Impaired - cognitive limitations						
learning disability	non-critical	1,082	1,217	2,036	154	4,489
mental retardation	critical	226	339	827	55	1,447
alzheimer, senility, dementia	critical	n/a	30	648	1,259	1,937
other mental or emotional	non-critical	256	483	2,140	352	3,231
Totals						
all disabilities	critical	415	709	2,547	2,596	6,267
all disabilities	non-critical	2,092	2,743	8,177	6,667	25,946
all disabilities	all barrier	2,507	3,452	10,724	9,263	32,213

The number of people who want to play games but find themselves excluded from playing games is likely to be smaller than the entire population of disabled individuals, because not everybody plays games. It is also worth noting that age is strongly correlated to the likelihood of disability. As shown in Table 2.1, less than 4% of people under age 15 are impaired, and over 27% of senior citizens (persons 65 years old and older) have a disability. While the percentage of senior game players is relatively low, we should keep in mind that current gamers are getting older and accessibility problems might present themselves in the very near future as they age.

2.2 Overview of Accessible Games and Strategies

This section surveys existing accessible games for each of the disability categories defined in Section 2.1.2 for a number of different game genres. Section 2.1.3 revealed that each type of impairment corresponds with one step in our interaction model, which gave rise to the conjecture that fundamentally different solution strategies may be required to make these games accessible. By analyzing the strategies used in each game, we seek to find evidence to support this conjecture. The accessible games have been found by searching information technology websites, online and paper-based journals, academic databases, academic institution websites and related conference websites.

2.2.1 Games for Motor Impaired Players

Motor impaired players often find it difficult or impossible to use conventional game input devices such as controllers or a mouse and keyboard, and they have to revert to alternative input devices specifically designed to accommodate their abilities.

Alternative Input Devices

A number of input devices have been developed that allow motor impaired players to interact with games, such as switch inputs (Figure 5.2.1) [90], head trackers [81], eye controllers [115], mouth controllers (Figure 5.2.2) [71], one-handed controllers

(Figure 2.5.3) or brain wave controllers (Figure 2.5.4) [82]. Software solutions exist such as a vocal joystick [52, 57] that enables the player to control a cursor using voice commands. This technique has been successfully used to control games as well, such as Tetris [116], either using speech or non-speech techniques.

2.5.1: Switch Controller

2.5.2: Mouth Controller [71]

2.5.3: One-Handed Controller

2.5.4: Brain Controller [82]

Figure 2.5: Controllers for Players with Motor Impairments

Alternative input devices are typically constrained with regard to the amount and types of input that can be provided when compared with conventional game input devices. A game that allows voice control will accept only one voice command at the same time, whereas a regular controller allows one to provide a combination of inputs. A one-handed controller typically provides only one analog input, whereas a regular controller has two analog thumb stick inputs. Playing a game with a one-handed controller that requires two analog inputs often leads to problems. For example, a first-person shooter may use one analog input for moving and another one for controlling the camera. Typically these controls are used simultaneously, which is not possible with a one-handed controller. An eye controller or brain wave controller may not allow for precisely controlling a cursor on the screen with the same precision as a non-disabled player with a mouse or a thumbstick. A switch is an assistive technology device that replaces the use of a mouse, keyboard, controller or joystick which severely motor impaired players may find difficult to use. Switches can be operated by any body part that is able to produce consistent and voluntary movement, and different types of switches can be identified based upon the type of action required to use them (sip and puff, pull, push, or squeeze). Severely motor impaired individuals may sometimes be able to use only one switch, whereas less severely motor impaired individuals may be able to use multiple switches. A mouth controller, like the quad controller [71] used by players with quadriplegia, offers a number of different switch inputs such as a sip and puff. Binary input is the smallest amount of interaction that can be provided with a switch, because holding down the switch for a certain amount of time may be impossible for a sip and puff device or painful for someone with arthritis.

Switch players who have good cognitive skills, including the ability to time their input, distinguish between choices, or understand the results of a particular choice, can provide more complex input (at a slower rate) through a mechanism called scanning [113]. Scanning folds a set of interaction options into a chain and iterates over each link, allowing the player to select that interaction option. For example, Microsoft Windows XP has a built-in, on-screen keyboard with support for scanning, allowing the player to input 40 different keys through a switch device. Input through a scanning mechanism inherently slows down the rate of input because it may take time before the right input is selected, depending on how many input options are attached to each chain. A number of different scanning mechanisms [120] have been developed that aggregate several different input options and offer multi-step selection as a mechanism to access each aggregated option. For example, to select a point in a 2-D grid, one can use a two step row-column scanning mechanism.

Selected Games

Alternative input devices are constrained with regard to the amount and type of input that can be provided. As a result, motor impaired players who use such devices are constrained to playing games that take into account the limited amount of input the player can provide. Though some games exist with limited input options, we are interested in finding more strategies for making games accessible to players with motor impairments. We therefore focus on games that can be played with the smallest amount of input, *e.g.* one-switch games. The one-switch website (http://www.oneswitch.org.uk) is a non-profit website dedicated to such games, and it lists about 70 arcade-style one-switch games, all of which can be downloaded for free. A number of games have been selected that cover all game genres as well as all the different strategies used to make the game switch accessible.

- **Strange Attractors** [138] is a one-switch puzzle game, where the goal is to reach the exit of a level with a spaceship (see Figure 2.6). Holding the switch activates the spaceships gravity drive allowing the player to sling and bounce himself through a field of planets towards the exit. The gameplay is built around switch input and is completely original.

Figure 2.6: Strange Attractors [138]

- **Mini Golf** [119] is a remake of a two-dimensional golf game. It uses a two-step scanning mechanism that allows the player to first select a direction where the ball needs to go and then select the amount of impulse to give the ball. After these two variables are set, the ball is hit. If the ball does not end up in the hole, the two-step scanning process is repeated.

- **Jet Boarder** [21] is a racing game see Figure 2.7. During the race, the player will always accelerate and will continuously steer left. Holding the switch will steer the player to the right.

Figure 2.7: Jet Boarder [21]

- **Frogger** [117] is remake of the classic 1981 Sega game *Frogger*. To allow for one-switch access, the amount of interaction in this game has been reduced. In the original version of the game it is possible to move the frog left and right as well as jump forward or backward. In this version, however, the frog can move only forward and backward using a scan mechanism.

- **Branston and the Lost Machine** [36] is a one-switch adventure game. The game uses a context-sensitive scan mechanism that changes the action bound to the switch input to allow for a larger amount of actions to be provided. For instance, when moving through the world, holding down the key will move the character in its facing direction, and tapping the button will change the direction he is facing. If the character is standing in front of a door however, tapping the key will make the character move into the building.

◦ **Sudoku Access** [84] uses a three step column/row scanning mechanism (shown in Figure 2.8) to allow a player to play the game of *Sudoku* using one-switch input. The game also allows for control using voice input.

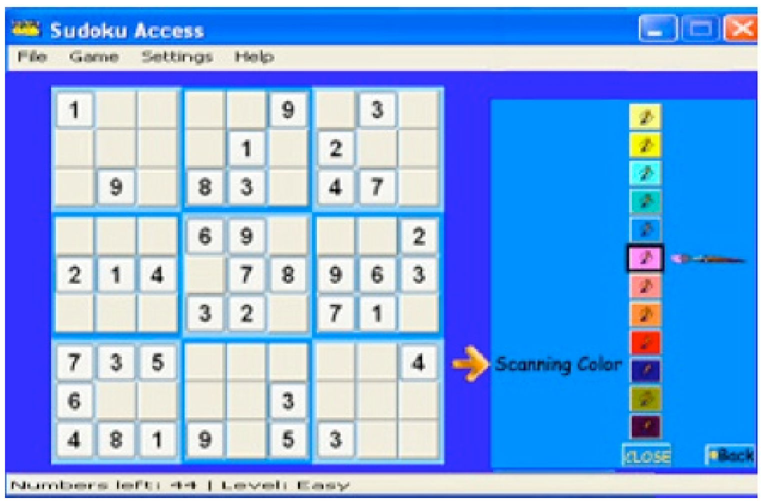


Figure 2.8: Sudoku Access [84]

◦ **Gordon's Trigger Finger** [55] is a modification of the multi-player version of the popular FPS *Half-Life 2*. This modification puts the player on top of a bot which automates navigation as well as aiming, allowing the player to control the gun using one-switch input. A player can play against other bots or against other players.

Strategies Identified

All the games use scan-accessible menus that allow the player to configure or start the game. In addition, a combination of strategies are used to make the game accessible:

- **Reduction:** Part of the original interaction is completely removed. For example, in one-switch Frogger the option to make the frog go left or right has been removed, and the player can move only up or down.

- **Automation:** Part of the original interaction is automated. For example, firing the gun in Gordon's Trigger Finger is done automatically.

- **Scanning:** Several games use scanning and we identified two different forms. A context-sensitive (CS) scanning mechanism changes depending on the state of the game. In Branston and the Lost Machine, the switch action changes depending on whether the character stands in front of an object such as a door. A context-agnostic (CA) scanning mechanism is the same throughout the game. Multistep scanning mechanisms are used in Sudoku Access, which uses a three-step row-column-number selection, and in Mini Golf with a two-step scanning mechanism to determine the direction and magnitude. Most of the games that use scanning are not time sensitive with the exception of Frogger, which only has two different actions to be scanned over.

Table 2.2 shows the strategies used in each game. With the exception of Strange Attractors, most games use one or a combination of strategies. Games can further be distinguished by whether or not they are binary, one-switch compatible. Applying

Table 2.2: Strategies Used to Make Games One-switch Accessible

Name of Game	Genre	Modification	Binary One Switch	Reduction	Automation	Scanning
Strange Attractors	Puzzle	no	no	-	-	-
Sudoku Access	Puzzle	yes	yes	no	no	CA
Jet Boarder	Racing	yes	no	yes	yes	no
Mini Golf	Sports	yes	yes	no	no	CA
Frogger	Arcade	yes	yes	yes	no	CA
Branston	Adventure	yes	no	no	yes	CS
Gordon's Trigger Finger	FPS	yes	yes	yes	yes	no

particular strategies to an existing game to make it one-switch accessible may significantly alter the gameplay. For example, in Frogger, by removing the option to go left or right, the gameplay changes from being able to positioning the frog in two dimensions to more of a timing-based challenge game that requires carefully timing when to move the frog back or forward. By automating the navigation and aiming in Gordon's Trigger Finger, the gameplay has fundamentally changed from an FPS to a "rail shooter" where the player feels he is tied on a rail like a roller coaster.

When making games accessible to the motor impaired, it is important to understand how the gameplay must change to avoid creating a game that is not fun or challenging to play. To avoid any changes in the gameplay, it is important to preserve the original input options as much as possible. Sudoku Access and Mini Golf are examples of one-switch games that offer the same number of input options as the non-accessible versions. They have been implemented using a scanning mechanism. One of the disadvantages of using a scanning mechanism is that it slows down the input rate, but this is not a major problem for these games because they are not time sensitive. For time-sensitive games such as Frogger, having the player interact through a scan mechanism with many inputs may be too slow and frustrating, and as a solution some of the input options have been removed to keep the game fun to play.

2.2.2 Games for Visually Impaired Players

Visually impaired players may find it difficult or impossible to receive feedback from games because visuals are typically used as a primary stimulus. Features, such as high-contrast color schemes accommodating color blindness, scalable fonts or the ability to zoom in, are recommended by accessibility guidelines for software and the web [137] and can be found in a number of commercial games. For example, many games allow for increasing the font size. A number of puzzle games allow for using different color schemes when a player is colorblind [108], and many FPS and RTS games allow for customizing the colors of the units of the enemy. Severely visually impaired individuals, however, do not benefit from such features and specific strategies are used to make games accessible to them.

Selected Games

Many games have been developed for the blind [39, 102]. An extensive list can be found on the website (http://audiogames.net). A number of games have been selected that cover a broad range of game genres as well as all the different strategies that are used to make the game accessible to blind gamers.

- **Mach 1 Car Racing** [68] is a remake of the classic racing game *Pole Position*. Audio cues, such as the echo of your engine, are used to indicate which direction to turn. The game uses self-voicing with an adjustable rate of speech. Players control their car using the arrow keys.

- **Shades of Doom** [46], **AudioQuake** [6], **Terraformers** [139] are FPS games. These three games use different strategies to make them accessible. Shades of Doom uses audio cues such as footsteps to help find the way through the levels. A navigation tool provides synthetic speech about the player's surrounding. AudioQuake uses "earcons" which are structured sounds that obey musical conventions [16]. These alert the player to an object or event. Terraformers uses a sound compass where different tones are used to indicate the direction the player is pointing. It also offers a sound radar that can be used to identify what is in front of the player. By using a "ping", it is possible to tell how far objects are in front of the character. Using a key on the keyboard, it is also possible to tell what type of object is in front of the character, using a voice playback system. All games use stereo sound to convey spatial information. These games have the same controls as regular FPS games.

- **The Last Crusade** [30] is a role-playing RPG game that uses self voicing to read out the events of the game. A player uses a keyboard for basic control including spacebar for stats reporting and instructions and arrow keys for navigation.

- **AudioBattleShip** [106] is a turn-based strategy game that was initially designed to be used for cognitive development purposes with blind children. A wacom tablet is used as an input device. A grid system is built over the tablet in order to represent the matrix of the battlefield and some additional help buttons for triggering actions. Audio cues are provided to inform about a specific spatial location on the board or the occurrence of certain actions such as the sound of a bomb dropping over certain cells in the battle grid. Though the wacom tablet does not provide haptic feedback, because it is a constrained input device it allows for better mental mapping of the battlefield.

- **GMA Tank Commander** [47] is an audio-only version of the classic game *Tank Commander*. The type of enemy is indicated using audio cues and its location through surround sound. The player controls their tank using his keyboard.

- **Speed Sonic Across the Span** [87] is a platform game. Audio cues are used to indicate objects and obstacles. The sound is consecutive with the platform panning to the right-left throughout the game. Players control their characters using a controller.

- **Metris** [60] is a musical version of *Tetris* that works with a screen reader. Audio cues such as tone and beats determine what input the player must provide. There is not much information provided by this game except the menu systems, which is why it could be handled by a screen reader.

- **Finger Dance** [79] and **AudiOdyssey** [44] are dance/rhythm games. Finger Dance is modifications of *Dance Dance Revolution* (DDR). Music games provide visual feedback based on the music that is played, indicating which inputs to provide on specific input devices such as a dance mat or a guitar controller. Finger Dance changes the original game play of the DDR game significantly because it replaces the original music with audio cues that indicate which keys the player must provide. AudiOdyssey caters to sighted as well as visually impaired players and uses audio cues to indicate to the player when to provide input with a motion sensitive Wii-mote controller.

Strategies Identified

The menus in all games are accessible through speech except for Metris, which uses a screenreader. The overall strategy used to make games accessible to severely visually impaired players is to replace visual feedback with a form of feedback that the player is still able to perceive.A number of different strategies for turning visuals into audio have been identified:

- **Replace** visuals with audio.
  - **Speech** can be provided either by a screen reader when text is provided or using self voicing.
  - **Audio cues** use real world sounds such as the sound of wind or footsteps to provide information or hints to the player.
  - **Sonification** uses non-speech audio to convey information using changes in pitch, amplitude or tempo [8]. A number of specific sonification techniques can be used:
    - **Auditory Icons:** sound effects indicate different objects or actions.
    - **Ear Cons:** one or more tones are used as a language to indicate different objects or actions.
    - **Sonar:** a sonar-like mechanism conveys spatial information on the locations of objects.
- **Enhance** visuals. Players with low vision can be supported by modifying visual stimuli using high-contrast color schemes, color schemes for color blind players, scalable fonts, zoom options or the ability to customize the color of enemy units.

Table 2.3 provides an overview of the games we surveyed, the accessibility mechanisms used as well as the input devices the player uses to play the game. Most games we surveyed use regular input devices such as a keyboard or controller. Some input devices are able to provide haptic feedback [103].

Table 2.3: Blind-accessible Games and Strategies used

Name of Game	Genre	Modification	Feedback	Input
Mach 1 Car Racing	Racing	yes	audio cues	keyboard
Shades of Doom	FPS	yes	audio cues, speech	keyboard
Audio Quake		yes	sonification, speech	
Terraformers		no	sonification, speech	
The Last Crusade	RPG	no	speech	keyboard
AudioBattleShip	RTS	yes	audio cues	tablet
GMA Tank Commander	Arcade	yes	audio cues	keyboard
Speed Sonic Across the Span	Platform	no	audio cues	controller
Metris	Puzzle	yes	sonification	keyboard
AudiOdyssey	Music	no	audio cues	Wiimote
Finger Dance		yes	audio cues	keyboard

2.2.3 Games for Hearing Impaired Players

Audio plays an important role in most games. Sound and music can add an extra dimension to the gaming experience. Few games use audio [133] as a primary stimulus, but the gaming experience for hearing impaired players may be significantly reduced when games fail to provide alternative forms of output. Though many games support subtitles [25], audio cues such as the sound of footsteps may indicate important events to a player, especially in an FPS game.

Selected Games

A number of games offer additional features that aid hearing impaired players:

- **XIII** [128] is an FPS based on a comic book series that transcribes sounds in a comic-like style (see Figure 2.9 for a screenshot).

Figure 2.9: XIII [128]

- **Zork: Grand Inquisitor** [72], **Half-Life 2** [132] and **Sin Episodes** [98] are the only three commercial games that feature closed captioning.

- **Smile** [3] and **Copycat** [15] are educational games targeted to deaf children. Smile has in-game characters that are able to perform hand gestures allowing for sign language communication shown in Figure 2.10.

Figure 2.10: Smile [3]

- **The Sims** [76] is a strategic life-simulation computer game that offers visual clues to indicate when audio is being heard. For example, a radio will display little music notes coming out of it when playing.

- **Doom3 [CC]** [67] and **Torque CC**[144] (see Figure 2.11) are third-party modifications to an existing game and game engine that describe sound effects and music as captions. For example, gunfire can be indicated by the text (Gunfire). Spatial sound is indicated using a sound radar that displays a dot on a radar indicating the location of the source of the sound. Different sources can be distinguished through color coding. Torque CC can be used with any game using the 3D Torque engine.

Figure 2.11: Torque

Strategies Identified

The overall strategy used in making games accessible to hearing impaired players is converting audio into visuals. The following types of visualizations have been identified to indicate the type, source and location of the audio:

- **Replace Audio with Text:**

  - **Subtitles** are captions for in-game speech, such as dialog between different characters. Portraits of the character speaking can be used to indicate the source or subtitles can be prefixed with the name of the character speaking.

- **Closed Captioning** captions for speech, sound effects, and music. Color coding or codes (soldier:) can be used to indicate the source of the sound. The location of the audio can be indicated either through a code (south:), or the position of the caption on the screen. Non-speech sounds can be captioned (Explosion) or transcribed (Bang).

- **Replace Audio with Non-text:**

  - **Visual cues**, such as displaying notes, can indicate that a device is making sound. The way sound is being transcribed in XIII is a blend of using text and visual cues.

  - **Sound Radar**, is used to indicate the source of the audio on a radar and is often used in conjunction with closed captioning.

  - **Signing** provides in-game characters that communicate with the player using sign language. To effectively use sign language, game characters need to be able to express different hand gestures.

Table 2.4 lists an overview of all games we discussed, the genre to which they belong and which accessibility strategies are used.

Table 2.4: Strategies Used to Make Games Accessible to Hearing Impaired Players

Name of Game	Genre	Modification	Text	Non-text
XIII	FPS	no	-	Transcribing
The Sims	Simulation	no	-	Visual Cues
Doom 3 Torque CC	FPS Any 3D game	yes	Closed Captioning	Sound Radar
Zork:GI	Adventure	no	Closed Captioning	-
Halflife 2 Sin Episodes	FPS	no	Closed Captioning	-
Smile CopyCat	Virtual World Adventure	no	Closed Captioning	-

2.2.4 Games for Cognitive Impaired Players

In our game interaction model, we discovered that players with a cognitive impairment may find it difficult to determine which in-game response to provide. Because cognitive impairments are complex and variable[73], the barriers cognitively impaired individuals may face when playing games vary significantly. Studies [24] of cognitively impaired individuals and children with Down Syndrome [35] reveal that they do play video games. No specific details are provided in these studies on what games are played, only that simplicity of the game is a key factor in their specific use.

Selected Games

In our survey of games, only two games were found that specifically accommodate cognitive disabilities.

- **Ilbo** [70] is a game specifically developed for players with learning impairments. The game was designed around a new form of providing input. Players interact with this game while sitting on a chair. This chair has limited input options as players navigate their character through a 3D maze using their weight. To prevent the player from perceiving too much feedback, the interaction with the game is paused periodically.

- **PowerUp** [127] is a multi-player virtual world education game that supports various disabilities, including cognitive disabilities such as dyslexia and low reading proficiency, by providing synthetic speech. It also offers a journal that keeps track of places for the player to go next. This feature may reduce the amount of information the player is required to memorize. The game can also be played using either a mouse-only input or keyboard-only input. Auto aim is available to reduce the amount of input that the player needs to provide.

A recent development in games research is the notion of serious games, which are not developed for entertainment purposes but for other purposes such as education, rehabilitation, or health. A number of serious games for individuals with cognitive impairments are included in our analysis:

- **Coucou Cachè** [111] is a puzzle game specifically developed for autistic children. The game (see Figure 2.12) is characterized by simplicity in order to not perturb autistic children who are often sensitive to complex environments. Players interact with the game using a touch screen.

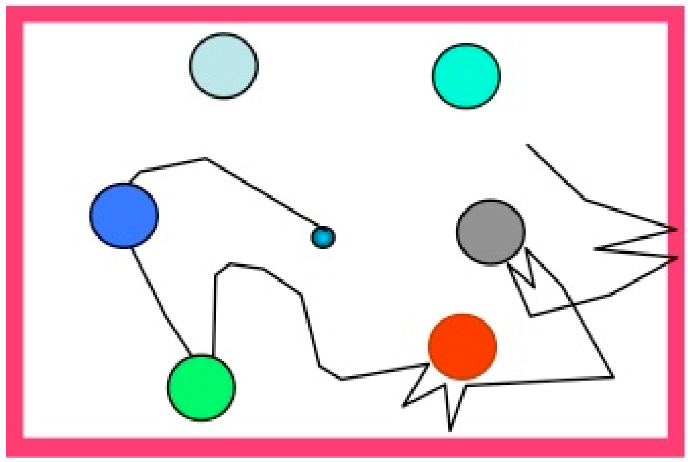


Figure 2.12: Coucou Cachè [112]

- **Flexibility Learning on the Web (FLOW)** [86] is a web-based multi-player puzzle game, as shown in Figure 2.13, which is designed for children with autism. The goal of this game is to enhance social interaction and cognitive flexibility by having children cooperate with each other to solve problems and thereby successfully complete games and activities. Players interact using a keyboard and simple controls such as the arrow key and spacebar are used to move an avatar or interact with other players.

- **Shared Interfaces to Develop Effective Social Skills (SIDES)** [92] is a four-player puzzle game designed to help adolescents with Aspergers Syndrome practice effective group work skills. Players interact with the game using a tabletop touch screen in Figure 2.14.

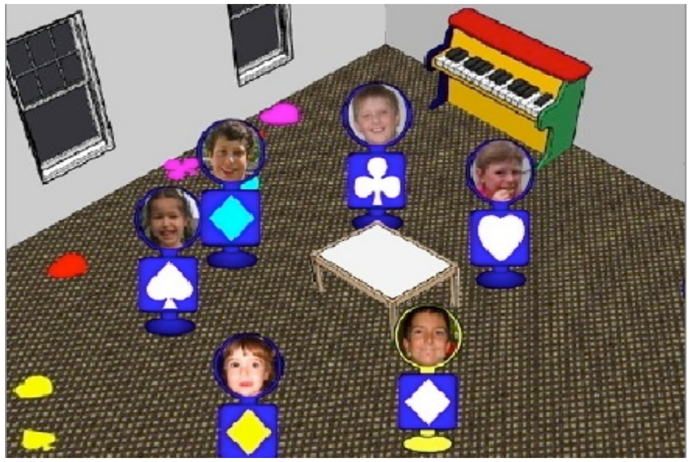


Figure 2.13: FLOW [86]

Figure 2.14: SIDES [92]

Strategies Identified

Though only a small number of games accommodate the needs of cognitive impaired individuals, some strategies are provided based on the similarities we identified between these games:

- **Reduce time complexity** - none of the games are time sensitive, so there is no pressure for the player to make a decision within a certain amount of time. The Ilbo game pauses the game frequently so as not to overwhelm the player.

- **Reducing the amount of stimuli** - all of the rehabilitation games provide a small amount of visual stimuli. The number of game objects on screen is limited, and all the information required to play the game is available on screen. The PowerUp game offers a journal that keeps track of events and places. All these features aim to minimize the need for memorizing information.

- **Reduce input** - similar to the strategies used for to make games accessible to motor impaired players, almost all games for cognitive impaired players offer limited but intuitive controls, such as using a touch screen. Reducing the amount of input can be achieved with techniques such as removing input or automating input.

Table 2.5 lists an overview of each game, the genre to which it belongs and which accessibility strategies are used.

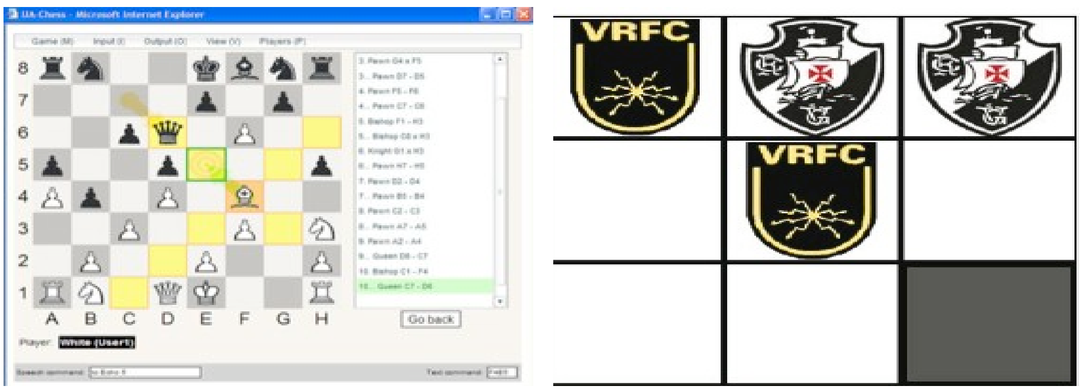
Table 2.5: Strategies used to make games accessible to cognitive impaired.

Name of Game	Genre	Modification	Reduce Input	Reduce Stimuli	Reduce Time
PowerUp	RPG	no	yes	no	no
Ilbo	FPS	no	yes	yes	yes
Coucou Cachè	-	no	yes	yes	yes
FLOW	-	no	yes	yes	yes
SIDES	Puzzle	no	yes	yes	yes

2.2.5 Universally Accessible Games

In this chapter, we identified a small number of games that incorporate the "design for all" paradigm. Most of the accessible games we surveyed accommodate the needs of players with one particular impairment. A few games implement a multi-modal approach where multiple interfaces are designed for different impairments. Universal design does not indicate that all impairments are supported but rather that multiple types of impairments are supported. Only one game (PowerUp) supports all four impairments though this game is not one-switch accessible and has limited support for cognitive impairments. We list them here for informative purposes because the strategies used in these universally accessible games are the ones we identified in the previous subsections.

- **UA Chess** [49] and **Universal Tic-Tac-Toe** [88], as shown in Figure 2.15, are board games supporting access by vision- and motor- impaired players using scanning, voice input and synthetic speech. Scanning technique and voice command are used for providing computer access to people with motor impairments. A built-in screen reader that offers full auditory access to visually impaired players. Additionally, the game can be sized according to player preference and zoomed in and out at different levels.



2.15.1: UA Chess [49] 2.15.2: Universal Tic-Tac-Toe [88]

Figure 2.15: Universal Access Games

- **Access Invaders** [50] is a remake of the classic game of *Space Invaders*. It offers one-switch control using scanning as well as larger amounts of input, allowing access to motor impaired players. Synthetic speech, audio cues and the ability to increase the size of game objects allow access to visually impaired players.

- **PowerUp** [127], as mentioned in previous section in Figure 2.16, offers a broad set of accessibility features, such as synthetic speech, audio cues, scalable fonts and high-contrast color schemes that allow for access by vision impaired people. Single mouse or keyboard-only options support motor impaired players. Closed captioning and visual effects for sounds support hearing impaired player.

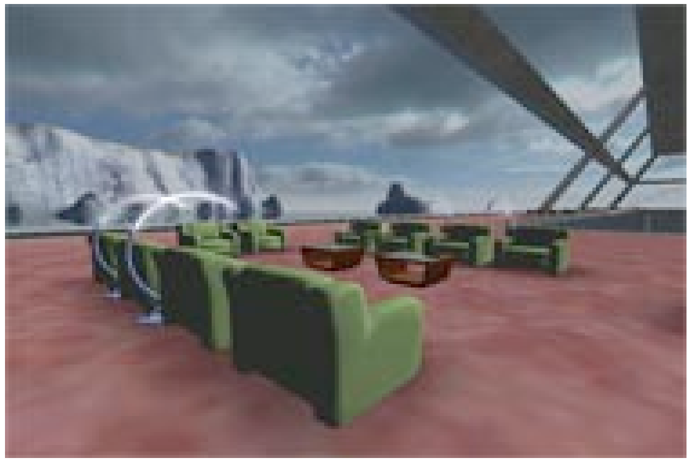


Figure 2.16: PowerUp [127]

2.3 Discussion & Research Issues

The games surveyed in Section 2.2 allowed us to identify a number of issues that we discuss in this section along with our research target.

2.3.1 Accessibility Strategies

In the definition of our game interaction model, we identified that each type of impairment corresponds with one step in the game interaction model (see Section 2.1.3). Table 2.6 summarizes the strategies that have been harvested from each accessible game for each type of impairment based on our interaction model. This information led to the assumption that fundamentally different solutions strategies may need to be explored to make games accessible to different impairments. Based on the strategies that have been harvested from the different games we can, to some extent, confirm this conjecture.

Sensory impairments correspond to the first step of our interaction model and the strategies used to make games accessible to individuals with these impairments are substantially different from those used for motor and cognitive impairments. Although motor and cognitive impairments affect different steps in the game interaction model there is some overlap in their identified strategies. For example, reducing the

Table 2.6: Accessibility Strategies Summary

Interaction Model	High-level Strategies		Low-level Strategies
Receive Stimuli (Visual & Hearing)	-Enhance Stimuli	*visual*	-High contrast color schemes -Increase font size -Color blind color schemes -Zoom options
	-Replace Stimuli	*audio* → *visual*	-Text (subtitles, closed captioning) -Non-text (visual cues, sound radar, signing)
		visual → *audio*	-Speech (screenreader, self voicing) -Audio cues -Sonification (earcons, sonar, auditory icons)
Determine response (Cognitive)	-Reduce Stimuli	*visual*	-Limit number of game objects -Simplify storyline
	-Reduce Time Complexity		-Increase response time -Slow down game
	-Reduce Input		-Remove input -Automate input
Provide Input (Motor)	-Reduce Input		-Scanning -Remove input -Automate input
	-Replace input		-Voice/brain control

amount of input is a strategy used in games for both motor impaired and cognitive impaired players. Their difference exists in the low-level strategies. Scanning is not identified in the games for the cognitive impaired, but it is used in one-switch games to allow for providing larger amounts of input for the motor impaired. A possible explanation may be that a scanning mechanism typically increases the cognitive load whereas accessibility strategies for cognitive impaired aim to reduce the cognitive load.

The strategies we identified in this survey express design knowledge in a more usable format as well as at a higher level of abstraction than existing game accessibility guidelines. The game interaction model points out exactly what types of barriers a player with an impairment faces, whereas game accessibility guidelines [58, 126] do not identify the accessibility problem the guideline solves. There are many similarities between our low-level strategies and the two sets of game accessibility guidelines. Some accessibility guidelines, such as “use sound radar” [126] or “provide subtitles” [58], are part of the low-level strategy “replace audio with text based visualization”. Though each one of the low-level strategies has been identified a number of times in the games we surveyed, they are neither complete nor exhaustive. The games we surveyed cover only a limited number of genres and the low-level strategies for players with cognitive impairments have been harvested from a very small number of games.

More confidence can be attributed to the high-level strategies because they are derived from the game interaction model as well as from the numerous low-level strategies at the implementation level. The high-level strategies may also be used to explore and develop new solutions for which guidelines fail to provide advice. For example, someone with the rare condition of anaphia is unable to feel touch. Existing guidelines do not provide any advice on how to make a game accessible that purely relies upon haptic feedback. Using our game interaction model, a developer could analyze that someone with anaphia would not be able to perform the first step - receive stimuli. The high-level strategy “replace stimuli” points out that visual or audio feedback could be used to make a game accessible. In the future, when more accessible games may have been developed for new game genres, new low-level strategies may have been identified that complete our model.

2.3.2 Tradeoffs

We identified two different approaches toward developing accessible games: (1) An existing game is made accessible using one of the strategies in Table 2.6. The majority of the games surveyed in this paper are modified games. (2) A game is specifically developed to accommodate a particular disability. In our survey we found two examples of such games; Strange Attractors and AudiOdyssey are two accessible games not based on an existing game. Both approaches have their advantages and disadvantages. It may be more resource-intensive to create a game from scratch than it is to modify an existing game. By modifying an existing game, a developer does not have to worry about parts of the game that are expensive to create such as the graphics and sounds or worry about whether the game will be fun, because the gameplay is taken from an existing game. However, by applying certain accessibility strategies, the original gameplay may be significantly altered. For example, a one-switch racing game does not allow the player to brake or speed up because certain input options may have been removed to allow for one-switch input. Converting feedback from one modality into another modality often leads to loss of information. For example, the resolution and detail of stimuli received from a visual modality is much larger than what can be perceived through audio or haptic modalities. Audio games often reduce the amount of feedback provided when compared with the original game so as not to overwhelm the player. Players must determine what in-game response to provide based on a smaller amount of information.

These tradeoffs may significantly alter the original gameplay, and a developer may end up creating a game that is not fun to play. An additional concern with regard to multiplayer games is the notion that games must be fair [22]. A player with an impairment who is playing against a player with a different impairment or without an impairment should not have an advantage nor a disadvantage. A one-switch FPS [55] that automates moving and aiming may give a motor impaired player a significant advantage over players that have to perform these tasks themselves. When modifying an existing game, the original gameplay must be preserved as much as possible, but because gameplay and interaction are typically closely intertwined, it is often challenging to do so, especially for games for players with motor impairments. Few games, such as UA Chess, Universal Tic-Tac-Toe and Sudoku Access, manage to succeed in making the game accessible while preserving the amount of input options the player has available. Because their original counterparts, *Chess*, *Tic-Tac-Toe* and *Sudoku* are not time-sensitive games, a scanning mechanism can be applied, whereas most other games involve combat and require the player to respond quickly. The scanning mechanism may be too slow for the player to be able to respond fast, and tradeoffs are required when parts of the input are removed or automated. When developing a new accessible game, such tradeoffs do not have to be made because a game is developed from within the constraints imposed by an impairment (bottom-up), whereas when a game is modified, gameplay often has to be compromised to meet the constraints imposed by the impairment (top-down).

2.3.3 Directions for Research

We conducted a survey on gaming and accessibility, as described in previous sections, which provides insights for understanding the current state-of-the-art of research and real-world applications in the field. This initial research also helped us to further refine our research goals and key areas of interest. In this section, we describe our motivation for the subset of the disabled population we chose to focus on, the types of games we are interested in, and the tools we will use to validate the results of our research.

Population

Focusing on the most severe form of each disability type and analyzing the census data presented in Table 2.1, one can see that individuals with severe visual impairments make up the largest single group ($\approx$ 2 million in the U.S. alone) having a critical barrier to playing most types of games. Because this is the largest group and there remain great challenges to developing blind-accessible games, we have chosen to focus our research mainly towards blind users and their ability to play games of various genres.

Area of Interests

Games are meant to be fun and are mainly designed for entertainment. The more fun or entertaining a game is, the more likely it is to be financially successful in the current market. In recent years however, games have been designed for a diverse purposes, encompassing a broad spectrum of varied content for education[118, 42, 63, 66], health and rehabilitation [74, 121], religion [125], politics [89], and corporate training purpose [77]. In recent years, now integrated with the World Wide Web, games are even bringing people together from all over the world. Games such as *WoW* and *EverQuest* may have millions of players in their online worlds, who meet, converse, and form alliances and guilds without having to meet their partners face-to-face outside the games. The required social elements enrich the gameplay and attract players greatly. Accordingly, for this research, we select our case studies and focus ourselves based on i) the entertainment value, ii) educational elements, and iii) social activities of games with regard to exploring new high- and low- level strategies in conjunction with those identified in Section 2.3.1.

Proposed Case Studies

In the previous section, we identified two approaches for developing accessible games and explained their pros and cons including inevitable tradeoffs. From a research point-of-view, adapting existing games for blind users seems to be more effective and efficient, because we have a lot of popular and successful mainstream games to start with, and the amount of initial development can be reduced greatly. Based on our interests, we selected Guitar Hero and Second Life as our platforms for studying the possibilities and effectiveness of creating their blind-accessible counterparts.

All of the existing blind-accessible games surveyed in Section 2.2.2 used "replacing visual with audio" strategy. This really brought to our attention and made us ask, "What about haptic feedback?" Guitar Hero is a fun game as proven by its sales figures (ranking top 5 selling video game of 2007 [31]). In addition, the audio channel of the game is fully occupied as a necessary element to achieve success. This leaves few options other than exploring haptic solutions for its blind-accessible counterpart. Second Life, a free online virtual world, was chosen because of its rich social interactions and successful use in education and business.

In the next chapter, we discuss our first case study: developing and validating Blind Hero.

Chapter 3

Case Study I: Blind Hero

Research on technology for visually impaired people has primarily focused on web accessibility [10] and assistive technology [54, 69]. There are approximately 10 million blind and visually impaired people in the United States, including an estimated 1.3 million people who are legally blind and 1.5 million who use regular computers [4]. Research has focused on helping people with visual impairments to improve their reading, orientation and mobility skills. They are also trained to lead more independent and productive lives by including them in our information society. Studies conducted in assistive technologies with regard to vision are in great demand as our population ages and more people suffer from age related vision loss. Over the last several decades, video games have become a very popular form of entertainment, though very few of them are actually accessible to the visually impaired. Being able to play a game is a quality of life issue that, especially for younger people, may affect their psychological well being.

Games are mainly designed for entertainment, which puts one major constraint on the way people interact with them: it must be fun. This is different from the requirements of interacting, for example, with a web browser or a word processor, which are designed to help the user perform productivity-related tasks. In this chapter, we are introducing Blind Hero, a modification of an existing game, which we developed based on the popular video game Guitar Hero. Blind Hero transforms visual stimuli into haptic stimuli to allow visually impaired individuals to play a Guitar Hero-like game. One of the high-level strategies that we used for this case study is replacing stimuli as identified in Section 2.3.1. We want to find out if this haptic version of the game can bring the same level of entertainment to blind people as it does to sighted players and whether replacing visual with haptic is a viable low-level strategy to making games accessible in general.

The rest of the chapter is organized as follows: Section 3.1 gives an overview of the game we modified and briefly discusses some audio games. Section 3.2 presents our implementation of the game. This is followed by the results of our usability study with 12 players in Section 3.3. This chapter concludes with discussions on some unsolved problems with the implementation in Section 3.4.

3.1 Audio Games and Guitar Hero

Many audio games recognized in [7, 40] are designed to provide auditory instructions and non-speech sound commands for blind people to control the game. Most of these games are simple arcade, puzzle, or racing games entirely controlled by audio commands as listed in Section 2.2.2. We want to briefly mention the following two mainstream games that were made blind accessible focus on implementation aspects.

AudioQuake [6] and Terraformers [139] are first person shooter, mainstream games that have been made blind-accessible. AudioQuake replaces visual feedback based on the idea of "earcons", which are types of structured sounds, often obeying musical conventions. They are designed to alert the player to an object or event, though they do not "sound like" their referents in the real-world [6]. However, learning the mappings between the sound effects and real-world objects is not that difficult. AudioQuake uses stereo sound to help the character navigate. For example, a sound to the left or right indicates if a wall, door, or other obstacle is present near the player and in which direction. Sounds gets louder as a player moves closer to an object. Arrow keys are used to move and turn with some customizable options to help players control their characters. Terraformers uses a 3D sound response system. The main character can either be navigated using the visual display (which also has a high-contrast setting for low-vision players) or sound. There is a tone that represents

North and different variations of the tone to represent 8 compass points. The player can tell what is in front of the player by using a sound radar. By using a “ping” it is possible to tell how far objects are in front of the character. Using a key on the keyboard, it is also possible to tell what type of object is in front of the character, using a voice playback system.

Dance/rhythm games are suitable for the blind as they already rely on audio as major feedback mechanism. Dance Dance Revolution (DDR) is a popular rhythm game where the player provides input by dancing on a large mat trying to match step instructions shown on the screen while the song plays. An accessible version of DDR has been developed for the blind, called Finger Dance [79]. There are high and low pitched drum-roll sounds coming from either left or right speakers that consist of four output elements. Instead of four arrow step instructions in DDR, Finger Dance maps the four distinguishable sound to four keys on the keyboard for players to control. This is of course a significant departure from the original gameplay of DDR. AudioOdyssey [45] is another dance/rhythm game that uses a Wii-mote controller and that can be played by sighted as well as non-sighted players. It provides instructions and feedback entirely through audio. Our approach is related to the two aforementioned games as it is also a dance/rhythm game.

Guitar Hero is a popular rhythm game developed by Red Octane, that lets the player use a guitar-shaped controller with various colored buttons to simulate the playing of rock music. More than 14 million copies have been sold of this game to date [141]. The goal of this game is that the player must press the right button combination matching the colored dots on the screen, which are moving along with the music. Some dots have lines attached to them, implying that one should hold down that corresponding button for a longer period of time: until the end of the line hits the bottom of the screen (See Figure 3.1). When a player hits the right combination of buttons matching the dots on the screen, the score increases.

Despite unsuccessfull predecessors, Guitar Hero became a million dollar franchise and has dominated the sales charts for the past 2 years. If we break down the

Figure 3.1: Guitar Hero & Frets on Fire

interaction between the player and the game, we can make the following observations. The game continuously provides visual stimuli (dots and lines on the screen), which the player uses to know when to press certain buttons on the controller. If the player presses the correct combination it will hear the correct guitar riff for that part of the song.

Because Guitar Hero is closed source we cannot make any changes to the game. Frets on Fire[129] is an open source clone of Guitar Hero that is written in Python and runs on Linux, Mac, and Windows. Since we are mainly interested in exploring whether we could make this style of game accessible to the vidually impaired, starting with Frets on Fire reduced the amount of development overhead required to run our experiments.

The amount of feedback provided through visuals in Guitar Hero is relatively small. This makes the game a good candidate to apply the "stimuli replacement" technique. However, we cannot turn Guitar Hero into an audio game because of the predominance of the focus on musical sound already in the game. Hence, haptic feedback seems most suitable.

Haptic interfaces generate mechanical signals that provide information via sense of touch, which is much less commonly lost, compared to the sense of vision [101]. This technique is mostly adopted in medical fields and with assistive technologies. In some cases, non-visual (e.g. auditory or haptic) and multimodal (bi- and trimodal) feedback forms demonstrate significant performance gains over visual feedback forms [61] for both Age-Related Macular Degeneration (AMD) and normally-sighted users. Some scholars even argue that haptic-only feedback is more beneficial than auditory-only feedback [5, 17]. Haptics are also used to convey spatial information such as size and location [29].

3.2 Blind Hero

The first thing we created was a device that was able to provide haptic feedback.

3.2.1 Hardware

The design of the glove was motivated by being able to preserve as much of the original interaction as possible. Most of the critical input and output need to be interpreted fully and correctly. Three major pieces of hardware are listed as follows.

Pager Motors

The idea was to use small "pager" motors, most commonly found in pagers and cell phones, to provide haptic feedback. Ideally, one motor would represent each of the five colored buttons on the guitar controller. Whenever a pager motor buzzes the corresponding button must be pressed. Initially, we explored placing pager motors underneath each button on the guitar controller. However, the feedback of the pager motors turned out to be so strong that the whole guitar vibrated and players could not distinguish which button needed to be pressed. Therefore, we decided to mount the pager motors on the fingers of a special glove that the player would wear while playing the game. The original Guitar Hero controller has 5 buttons that the player operates using only four fingers. When it comes to the dots on the fifth "fret", the hand position must shift down to cover the last button. The design of the guitar makes it very hard to press the fifth button with the thumb. We cannot attach more than one motor to each finger. This leaves no option other than using only the four buttons and having to ignore the dots on the fifth fret. In our user studies, we used four 7mm in diameter, 12.5mm long pager motors attached on each of the four fingers of a left hand glove. These motors are rated 1.3 volts DC at 70 milliamps.

Circuit Board

With the help of Multilabs [80], we chose FT232R device, released by Future Technology Devices International (FTDI) [41]. It is a USB to serial UART (Universal Asynchronous Receiver/Transmitter) interface with optional clock generator output, and asynchronous and synchronous bit bang interface modes. This chip allows us to control each motor through the USB port. The port shows up as a virtual com port handling communication between the application and the device. The source code is programed into PIC16F688 microcontroller made by MICROCHIP [78] to control the motors and its running time. Figure 3.2 is the blue print of the board design. Four vibrating motors were mounted on the board and controlled by 3 bytes of data

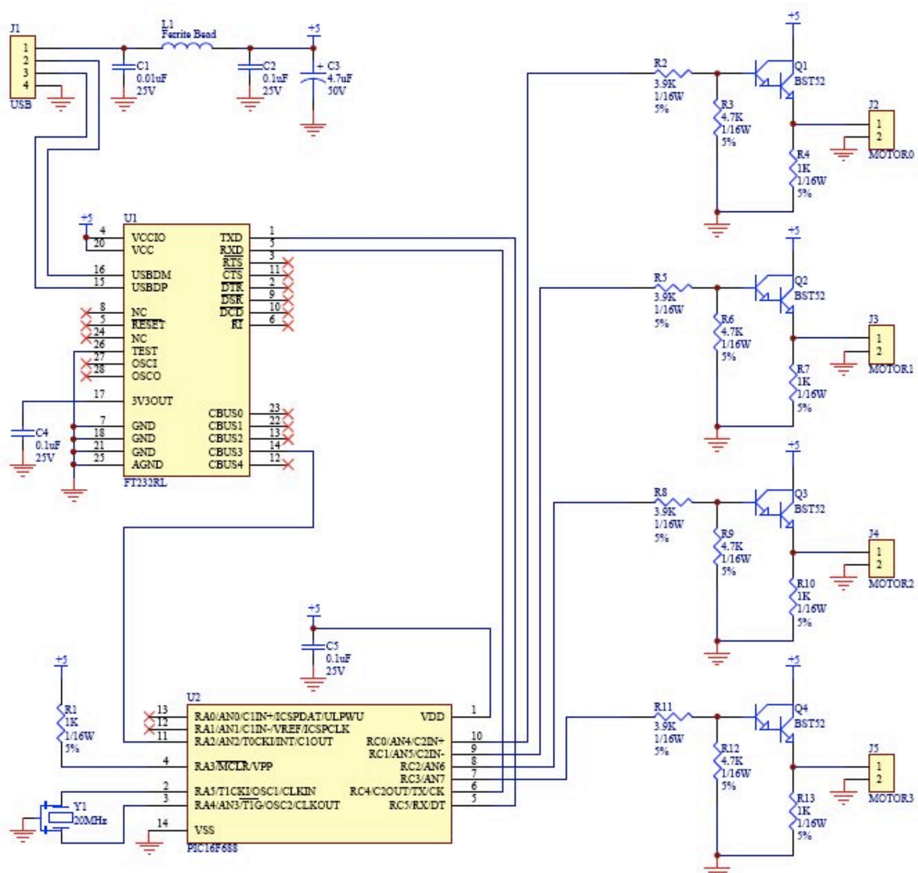


Figure 3.2: Circuit Board Design [80]

sent from the game via the USB port. The communication protocol defines that the first byte controls the number of motors that need to be on, and the rest of the bytes indicate how long those motor will be on ranging from 0 to 65535 milliseconds. This time schema can represent all required note length precisely. Figure 3.3 is a snapshot of our circuit board after built.

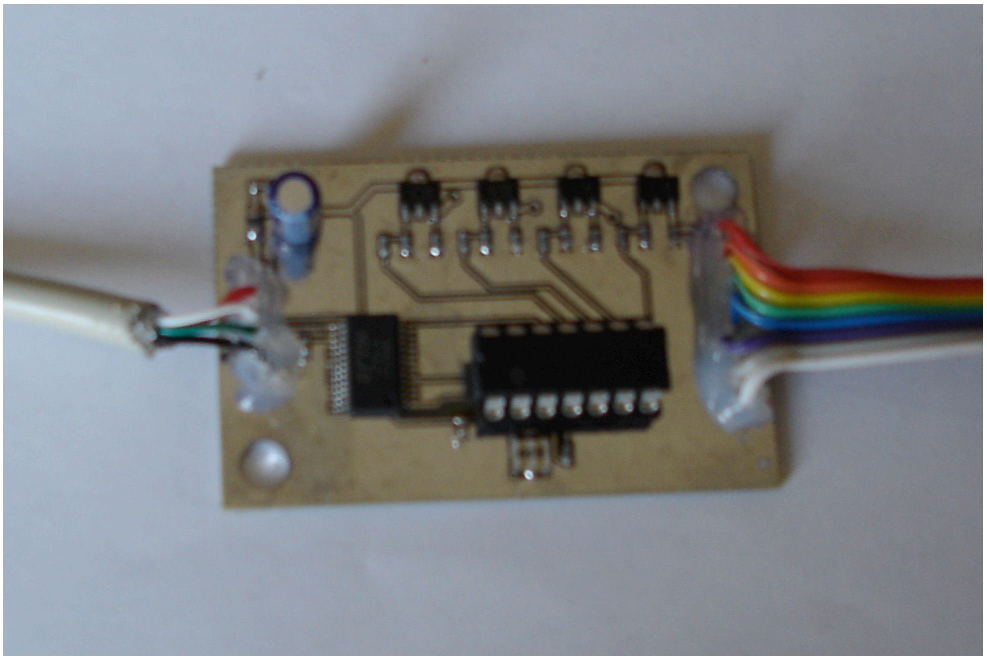


Figure 3.3: Circuit Board

The Glove

Figure 3.4 shows what our glove looks like. We chose regular workout glove because it is common, durable, and cost-effective. There is one tube attached at the back of each finger right below the finger nail. Our pager motor is well protected from the tubes, yet still give the right amount of vibration power. The circuit box will be attached to the player's wrist at the time of the play.

3.2.2 Software

We modified Frets on Fire to send a message to the virtual com port indicating which pager motor must vibrate and for how long. The problem here was that we needed

Figure 3.4: Haptic Glove

to precisely time when to start a pager motor. Players can respond faster to visual stimuli than to haptic stimuli. It will also take a while for the pager engine to operate at full speed. It will take the player some time to notice when a pager engine has started vibrating and so we needed to build in a delay e.g. the pager needs to start buzzing slightly earlier than the player needs to press the note. This delay needed to be determined accurately because if the delay was too short the player would not be able to respond fast enough, resulting in missing notes. If the delay was too long the player may press the button too early, also resulting in missing notes and potentially missing dots that are close by afterwards. A quarter of a second delay time was chosen based on the shortest time distance between every two dots of music notes and corresponds to average human reaction time. We came up with the following game tasks (See Figure 3.5) and corresponding haptic feedback to send to the virtual com port (See Table 3.1).

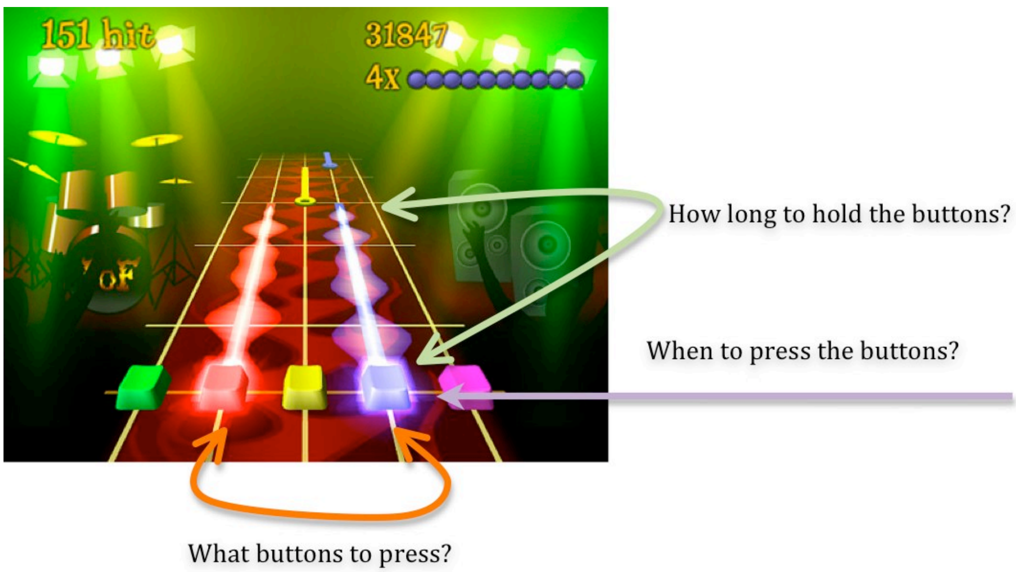


Figure 3.5: Game Tasks

Table 3.1: List of Tasks & Solutions

Tasks	Haptic Feedback
When to press the buttons?	Start Vibration 1/4 seconds before the screen moves onto the dots.
Which buttons to press?	Vibrating the corresponding pagers attached to the right finger to press.
How long to hold those buttons?	Vibration lasts the length of each note.

3.3 Usability Study

The "Beyond Accessibility to Efficiency" (BATE) principle states that the goal of assistive technology should be more ambitious than simply to provide access[54]. A user with a disability should be able to perform a task with the same efficiency as anyone else. Our application must bring the same level of entertainment to people with disabilities as without disabilities. For the Blind Hero user study we split the evaluation into two parts:

- **Quantitative** oriented analysis: we collected raw data including players' game score, accuracy percentages, and hardware effectiveness during the game play. Objectively measuring these components allows us to analyze how well a disabled player is able to play the game and observe his/her learning curve and improvement ratios.

- **Qualitative** oriented analysis: is the game fun to play? If the game fails to provide a fun and engaging experience no one will want to play the game and the game loses its purpose, despite players being able to use it efficiently for a task. Through a questionnaire (Figure 3.6) we asked players whether or not the game is fun.

Does the glove fit? ______________

What would you change on the glove part of the controller? ________________

____

Is the pager's vibration ☐ Strong ☐ Weak ☐ Others _________________________

Can you differentiate each pager's vibration? _____________

What would you change on the position of the pagers attached to the fingers to make the vibration more significant/effective? _________________________

____

Will you play this game if the glove and game are available in the future? ________

If you played Guitar Hero,

Is the amount graphics/interaction on screen ☐ Moderate ☐Overwhelming ☐ Others ______________________________

For all groups what do you think of the game in general? Do you enjoy this game? Any suggestions?

Figure 3.6: Questionnaire on Blind Hero

Many sighted and visually impaired players were involved in this study to create a broad range of experiences and to be able to compare performances.

3.3.1 Participants

All participants were new to the Blind Hero game. Some participants, however, are expert Guitar Hero players, whereas some have never played Guitar Hero before (See Figure 3.7). All the sighted players are blindfolded when playing Blind Hero. The players were stratified into four, three person groups based on visual acuity and the

Figure 3.7: A blind player playing Blind Hero

game they had to play (See Table 3.2). The questions that needed to be answered for these groups are:

- Will visually impaired individuals have the same gameplay experience with Blind Hero as sighted users have with Guitar Hero?
- Which group will do better on Blind Hero among groups 1, 2, and 3?

Table 3.2: Group Description

Group #	Vision Acuity	Game to Play in Experiment	Guitar Hero Experience
1	Sighted (Blindfolded)	Blind Hero	New
2	Blind	Blind Hero	Inaccessible
3	Sighted (Blindfolded)	Blind Hero	Expert
4	Sighted	Guitar Hero	New

3.3.2 Collected Data Analysis

The accuracy percentile over short and long dots, for each song played, is calculated for each player during the game. In general, players start on the easy level when they play a game for the first time. We picked relatively easy songs for our participants to test. On the easiest level, no more than one button has to be pressed simultaneously throughout the whole game. Each song comes with 3 or 4 different levels of difficulty, with the harder levels requiring more buttons to be pressed simultaneously. Different songs vary significantly with regard to rhythm, speed, and the occurrence of patterns in the music. Because of this, when playing different songs, one is likely to end up with very different accuracy rates, even on the same difficulty level. By only playing two different songs repeatedly (see Table 3.3) we were able to measure players' performance increases for each song. Each song takes approximately three to four minutes to complete and our participants were able to finish playing the whole set in less than 40 minutes. After finishing the set we asked them to fill out a brief questionnaire collecting qualitative feedback on varying aspects of playing Blind Hero.

Table 3.3: Song Playing Order & Labels

Music #	Song	Repetitive Count	Label
1	Song 1	1	1-1
2	Song 2	1	2-1
3	Song 1	2	1-2
4	Song 1	3	1-3
5	Song 1	4	1-4
6	Song 2	2	2-2
7	Song 2	3	2-3
8	Song 2	4	2-4

Figure 3.8 a) - d) shows the average accuracy level (e.g. how many dots they got right) over these two songs as defined in Table 3.3. All groups show a similar performance difference between song 1 and 2. Song 2's lower accuracy for all groups indicates that this song is more difficult to play. Looking at each song individually we

see, for all groups, that the accuracy increases after each time playing the song. Comparing the accuracy percentage for all four groups, we have the following observations: While playing Blind Hero, sighted individuals with no experience with the original Guitar Hero game had the worst level of performance in our study. Visually impaired players had much better performance than the previous group. Group 3, which includes sighted players that have already played Guitar Hero before, performed slightly better than the group of blind players. We also compared sighted players playing the original Guitar Hero game for the first time. They had the highest levels of accuracy among all groups.

Another interesting observation that can be drawn from Figure 3.8 is that all groups' performance levels follow similar curve patterns. For both songs, it takes the player a while to learn to play the game and increase their performance. In the very beginning, the accuracy percentages are relative similar and low. After that, performance starts to increase significantly each time a song is played.

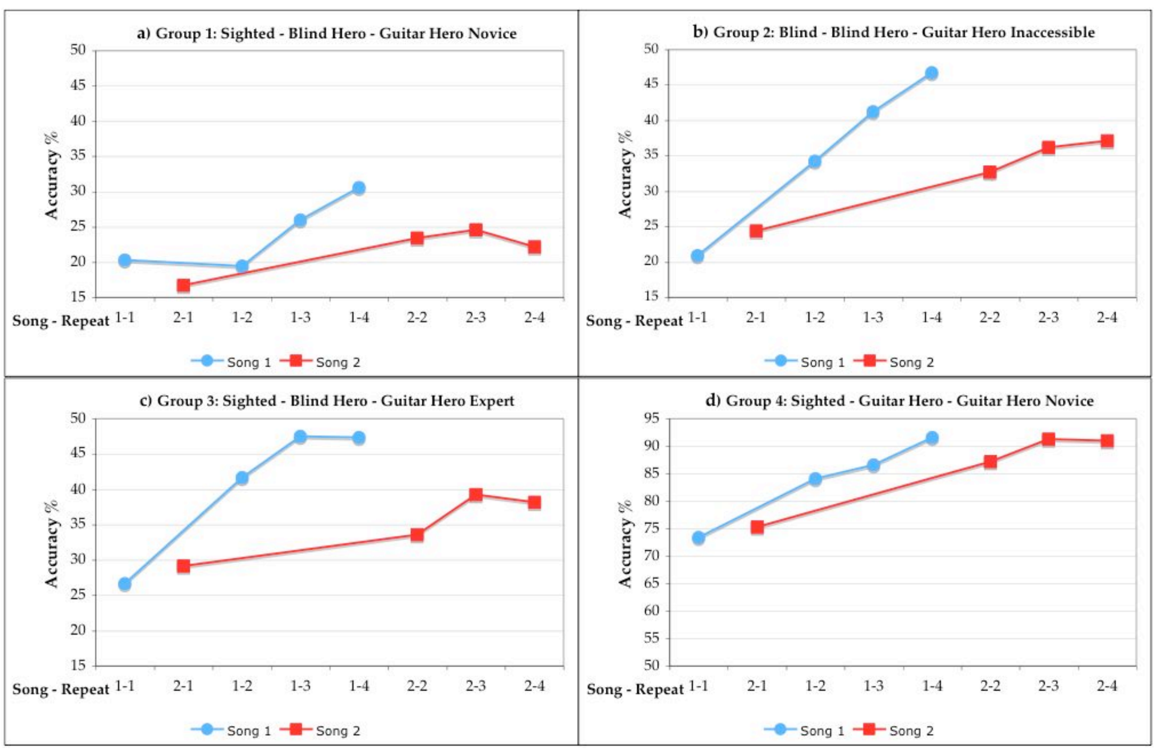


Figure 3.8: Average Performance for each group playing songs 1 & 2.

Based on the results we found, we have generated another chart that shows performance in Blind Hero by combining groups 1-3 and comparing against the results with Guitar Hero by group 4. The comparison results are split into different zones (see Figure 3.9) explained as follows:

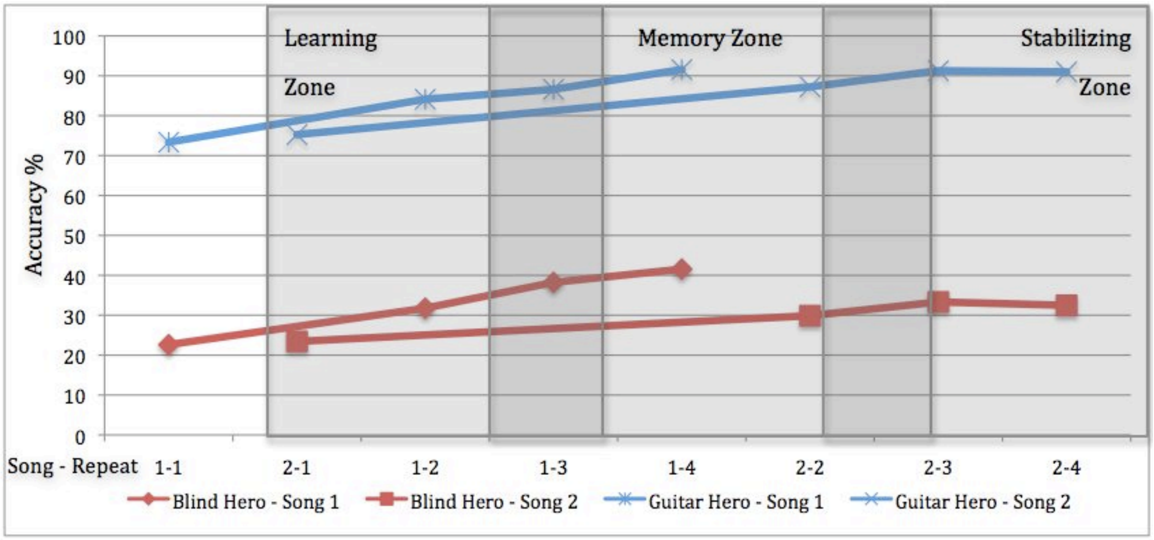


Figure 3.9: Guitar Hero vs. Blind Hero.

1. **Learning Zone:** hardware including the guitar controller and the glove are new to the participants that play Blind Hero and it will take a while for them to get accustomed to feeling which pager vibrates and for how long. The first minutes of the first song are to get used to the haptic glove. For Guitar hero this is similar e.g. they have to learn what the dots mean on the screen and how long they have to press the button by watching the screen carefully. All of the players will then learn when and what button to press by either following the glove's vibration or visual observation. The feedback for both games is the same; when the player presses the correct button; a guitar riff will play corresponding to that part of the song. This predominance of music is the reason why replacing visual stimuli by audio is not an option.

2. **Memory Zone:** at some point, players start recognize patterns that correspond to specific parts of the music such as the chorus or solos. From recognizing

patterns in the music, the player is then able to play parts from memory rather than by haptic input. We see a big performance increase in the transition between learning and playing from memory. Sometimes however, when relying on memory and ignoring the haptic inputs, players actually hit the correct notes too early, decreasing their performance rating. We realized that the glove is able to translate visual stimuli into haptic stimuli as it contributes to the memorization process and allows players of Blind Hero to feel-then-act, rather than visualize-then-act as sighted players of Guitar Hero do.

3. **Stabilizing Zone:** after 30 minutes of playing the game, the performance of all groups starts to stabilize. Some performance ratings may have even slightly decreased from one song to the next due to fatigue. The limits of players short-term memories were reached and further improvements in skill level might take longer-term practice with the game.

After playing the 8 songs, we gave each player a questionnaire focused on identifying whether this was a fun game to play. We received all positive feedback, despite the setup of our test requiring players to play the same songs repeatedly. Over the duration of the songs, we noticed players were enthusiastically trying to increase their performance, so the elements of challenge and progress is definitely present in our game. Players were convinced they could improve their performance given enough practice. Overall all, the players in the study were very impressed with the game.

3.4 Discussion

The development of Blind Hero and the results of the study have given us valuable insights into whether replacing stimuli is a viable technique to make games accessible. We made the following observations:

3.4.1 Compromising Gameplay

There is not a one-to-one mapping when transferring feedback from the visual domain to the haptic domain. The level of detail that the eye can separate including colors, shape, motion is a much higher level than what the sense-of-touch can differentiate. For Blind Hero we tried as much as possible to preserve the gameplay, as it is with Guitar Hero. Nevertheless, because there is not a one-to-one mapping, we had to make the following compromises:

- **Lookahead:** in Guitar Hero, the dots on the frets move from top to bottom leaving future dots on top of the current playing ones. Sighted people can see a whole screen of dots during the play, which gives them time to prepare which buttons to play next. For our haptic glove we only provide direct feedback on the button that needs to be played and we don't provide this type of lookahead information. We explored the use of additional motors for providing lookahead information, but this made the gameplay overly complicated, and would likely take players a longer time to learn. We compromised gameplay by not providing any lookahead information. This is the main reason that people score higher playing Guitar Hero than Blind Hero.

- **One Button Less:** as stated in Section 3.2, we carefully chose to only use four buttons out of five on the guitar. This will not affect any songs played at easy or medium levels, since the fifth fret is not used. The hard level uses the fifth fret and our glove currently cannot accommodate that. We are convinced that this compromise is acceptable, especially since all of our participants indicated that the current level of interaction is appropriate, which as players get more experienced may change.

In implementing these compromises and departing from the original Guitar Hero gameplay, we run the risk of creating a game experience that is not fun. Hence, compromising decisions must always try to optimize gameplay. We did not know how to provide lookahead information. Providing lookahead information through the use of extra pager motors was a solution we considered but that would most likely not be an enjoyable experience because of the steep learning curve. In this case we decided to get rid of the lookahead information all together, to make sure the game is easy to learn. Despite the compromises, we have the impression that Blind Hero is a very close representation of Guitar Hero.

3.4.2 Hardware Cost

For this game, we had to create a customized glove that players need to use to be able to play the game. The glove we developed cost about $1500 including custom design, hardware, and manual labor. Even with possible cost reduction when mass produced, cost may still be a barrier to the success of this game. In the future we might explore more cost-effective solutions, such as the use of headphones where the music could be playing in one ear while the other ear gets audio cues indicating which note to play. However, currently our haptic device can provide about 200 different stimuli; e.g. four frets, and dots varying in length. Using audio cues will require more research as players may have a hard time distinguishing different audio cues. Another viable option could be using a Force Feedback mouse and adapt the game accordingly.

3.4.3 Glove Design Pitfalls

Each motor is currently mounted just below the last joint on the finger. However, since players may have different sized hands, we need to find a solution that allows for customization of the glove to fit most peoples' hands. Initially we attached the pagers below the second last joint on each finger, which made it hard for players to identify which finger was buzzing. A glove where pagers could be attached anywhere on the finger would be optimal as it would allow the player to place the motors such that they could more accurately identify which finger is buzzing. This would be especially useful for songs that require one to play two notes at the same time, when it becomes harder to identify which finger is actually buzzing.

3.4.4 Short Dots vs Long Dots

The dots of the song come with different lengths. As part of our data collection, we did an analysis on the accuracy of long dots versus short dots using data from all groups. For all groups that played Blind Hero, players made almost twice as many long dots as short ones, while in the Guitar Hero game, the accuracy over long dots is just slightly higher than the short ones. We noticed that for shorter dots, the motor tends to generate weaker vibrations i.e. less effective to the player to make a move. This is because the duration of a pager's vibration affects the motor power level. We should be able to find another type of motor or alter the program to treat dots lengths the same cross the whole game in the future, or find a way to weaken the signal the pager sends for the longer notes so the shorter dots will be more easily distinguishable from longer dots.

3.4.5 The "Misconception"

A common notion is that blind people might have better senses of hearing and touch than sighted people because the loss of vision improves their other senses. Some think it is a misconception, others don't. What does this study with Blind Hero contribute to this statement? The performance comparison reveals that, when looking at the blind group's accuracy level and comparing it against average of group 1&3, blind people do have some level of advanced performance as shown in Figure 3.10. One interesting comment from one of our blind participants was that "Your brain does get too trained to react on things you touch on a day-to-day basis". Without being able to visualize, a blind person learns his or her surroundings by touching objects actively, and performs reading by touching the braille. Their brain has to constantly build bridges from retrieving feeling signals to sending a correct response signal to the right body part. From sensing motor vibrations to taking immediate action on the correct button, blind people practice this sort of behavior in their everyday lives. Sighted people are trained to visualize-then-act in daily life, while feel-then-act is not as common. Therefore, maybe it is the practice of feel-then-act that makes visually

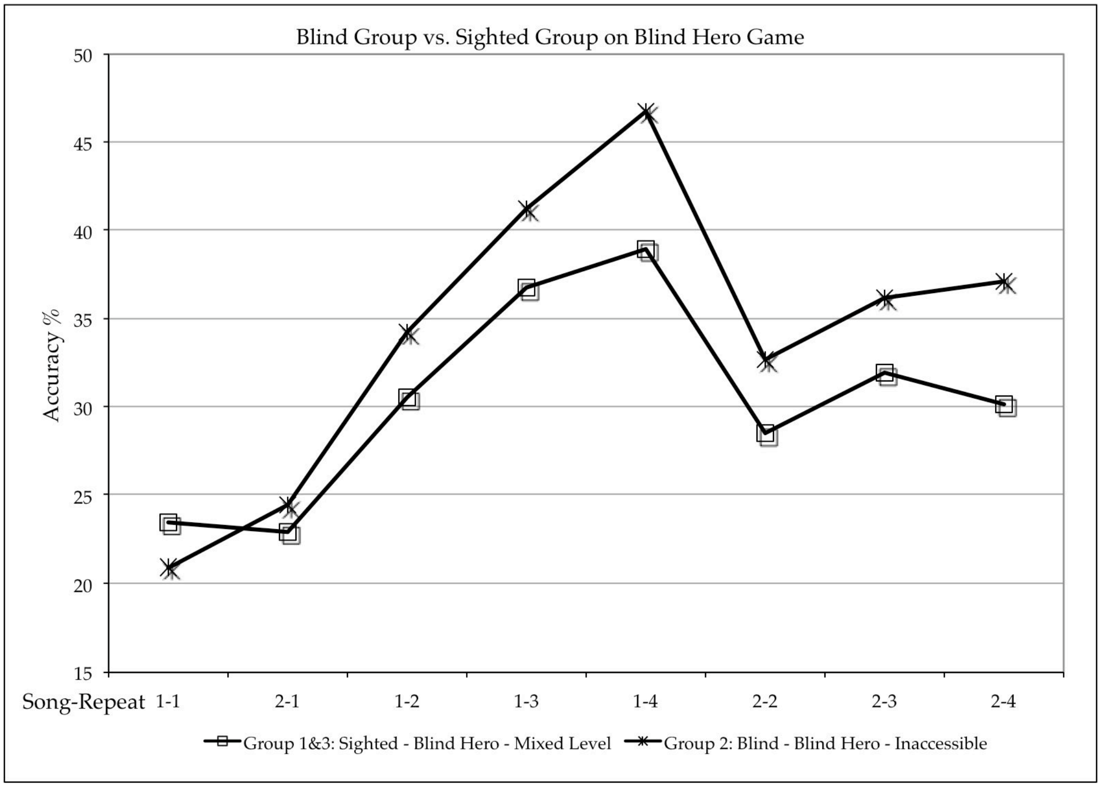


Figure 3.10: Blind people have better sense of touch?

impaired players perform better than sighted groups. Nevertheless our study was relatively small with only 3 non-sighted players. To verify that blind players have a greater sense of touch, the experiment needs to be repeated on a larger scale to make these claims generalizable.

Our haptic glove can successfully translate visual stimuli into haptic stimuli which allowed us to add “replace audio with haptic feedback” into the high-level strategies. Except for some compromises, all participants considered playing Blind Hero as a fun and enjoyable experience. With the completion of our first case study, the discussion turns in Chapter 4 for the second case study that focus on social and education purpose of a game remedied for blind users.

Chapter 4

Case Study II: TextSL

Virtual worlds, also known as Massive Multiplayer Online games (MMO's), offer rich, three-dimensional environments for great entertainment and vast social interaction. Following its popular MMO predecessors EverQuest and EverQuest 2, Second Life and World of Warcraft (WoW) are two virtual worlds that have enjoyed significant commercial success. WoW has more than 11 million subscribers who pay $15 a month and log in weekly [12], while Second Life has over 16 million residents [109] spending around $1.4 million per day [96]. These virtual worlds typically allow the user to control a digital puppet, called an avatar, with human capabilities such as walking and gesturing, through a game-like third-person interaction mechanism. Users can explore virtual worlds and interact with other avatars and objects. WoW is modeled after a role-playing game with all the typical elements found in most games, such as enemies to beat, levels to attain, story lines to follow, and the possibility for the avatar to die. Second Life is different from WoW because of its absence of these elements. It is more like a social community, similar to Myspace or Facebook, except it contains certain game like elements such as goals that can be achieved. A fundamental difference with WoW is that Second Life is entirely created and owned by its "residents". In Second Life individuals can create content and sell it to others using a virtual currency that can be exchanged for real currency. As a result, a thriving economy has emerged, where a number of people are able to make a living out of selling virtual content.

Despite the popularity of these virtual worlds, getting started and learning to interact inside of them and with the controls may take some practice. Controlling an avatar and managing multiple streams of visual and audio information requires some experience, aspects of which are often impractical for people with certain types of disabilities. 3D virtual worlds require a higher degree of input than traditional 2D interfaces, which raises concerns about its accessibility [1, 97]. For example, as most of the virtual worlds lack any textual representation, visually impaired people will find it impossible to access them using a screen reader [1]. This is unlike traditional software applications (such as web browsers), where meta information is more commonly available to help such individuals access non-textual content.

In this chapter, we start out with an overview of Second Life in Section 4.1. Section 4.2 explains our motivation of choosing SL. The Second Life interface that we created for the visually impaired are presented in Section 4.3. The conducted pilot study with 4 blind users and 8 sighted individuals are discussed in Section 4.4. Section 4.5 concludes this chapter with insights of improving the text client for SL from the observations.

4.1 Second Life

Second Life, developed by Linden Lab, surpasses the boundaries of the traditional two dimensional internet by offering a much higher degree of interaction. Interaction with other avatars mimics real life interaction as the avatar can perform a variety of activities such as dancing or playing games with other avatars. Since its opening to the public in 2003, Second Life has grown explosively; currently more than 13 million accounts have been registered. However, many of these accounts are inactive and some residents own multiple accounts [142]. Nevertheless, at any given time an estimated 38,000 people are online and an estimated 475,000 residents log in on a weekly basis.

A number of virtual worlds are in use, including There, Active Worlds, and Lively. Second Life, being the most successful and popular platform, has drawn great academic interest as a viable environment for learning due to its high degree of customizability. Because access to Second Life is free and its client is multi-platform and offers features such as being able to show videos and create content, it has successfully been used as a collaborative workspace in industrial [97] and academic environments [27, 100]. With these in mind, we chose Second Life as our medium to seek possible solutions allowing visually impaired players to explore virtual worlds.

Second Life claims to offer social communities agnostic of race, age, gender, and disabilities. Places like the Heron Sanctuary [123], currently housing 125 residents, are meeting spots for people with disabilities such as multiple sclerosis and muscular dystrophy. Naughty Auties is a virtual resource center within Second Life for those with autism [104]. Still, not all disability groups can access Second Life with equal ease and support. Visually impaired people are currently excluded from participating in such communities, due to a lack of supportive technologies. The social interaction offered by Second Life could benefit those individuals greatly as the severity of their disability may make them feel socially and physically isolated within their geographic communities [123].

In addition, Second Life offers a range of new and exciting possibilities for educators and their students. There are prominent educational institutions and organizations who are using virtual worlds for the delivery of a wide range of courses and educational events. With the increasing popularity of virtual classrooms hosted by universities in Second Life, it is important to explore how to make it blind-accessible to further comply with Section 508 of the U.S. Rehabilitation Act [131].

Most game designers and developers may not consider visually impaired individuals during their user interface design processes. Rich 3D graphics, incomplete meta-information, and software compatibility issues with screen readers all present great obstacles for this type research. This paper describes an investigation of two major area of interests provided by Second Life and proposes a text-based interface allowing access by visually impaired individuals.

4.2 Motivation

People enjoy the excitement, challenges, escape, and so many other elements that games offer. Of all the features within Second Life, we focus on the following two areas to test the contemplation and functionality of our blind-accessible interface.

- **Educational Use:** Rankin *et al.* believe that learning that occurs in the virtual world can be transferred to learning in the real world [94]. Games provides an entertaining experience and approach to informal science education [107]. In recent years, games have been applied to many knowledge learning areas as a viable and efficient platform for education [43, 95, 124]. Students could potentially perform better in virtual classrooms versus in traditional classrooms where the physical appearance and attention could distract them. Educational uses of Second Life can be found in [65]. Current research [14, 62, 99] have demonstrated success in distance and flexible education in Second Life, which is why we believe that it is important to involve visually impaired people.

- **Social Benefits:** In a study of virtual worlds [51], social interaction has been found to be players' most favorite aspect of MMO's. The social interaction possibilities offered by Second Life significantly exceed what is possible with traditional internet-based social interaction tools such as web forums, chat boxes, and messenger clients. From streaming video to built-in VoIP and endless user-created content, Second Life residents are provided with a wide variety of social activities to meet with new people. From conversations with local American Foundation of the Blind (AFB) chapter members, we saw an enthusiasm for meeting new friends and being exposed to similar opportunities as sighted individuals.

The Second Life viewer was made open source in 2007. This opened up the opportunity for developers to enhance the interface with new features and also for us to investigate whether and how Second Life could be made accessible. A number of accessibility problems were identified pertaining to different disabilities. In this paper we focus on developing an accessible solution for visually impaired people. Interaction within virtual worlds resembles that found in games. A number of games have been developed for this group of people [39, 102]. In the detailed analysis of blind-accessible games in Section 2.2.2, Powerup [127] is the closest virtual world in terms of allowing for visually impaired users. It is a new multiplayer virtual world educational game developed by IBM that supports various accessibility features for visual, motor and cognitive impairments. Visually impaired users access this game using audio through built-in self-voicing and sound objects that guide the user. They use the same controls as sighted and can issue a number of commands activated by key presses to get information on their environment. However, Powerup is created completely from scratch with accessibility issues in mind.

The massive amount of user-generated content in Second Life severely limits the ability to use any of the aforementioned low-level strategies in Section 2.3.1. Only the owner of the land can make changes to the objects on it. Augmenting the Second Life environment could change the experience for sighted users which may be undesirable. Though Second Life has been made open source, the underlying infrastructure cannot be changed and a post hoc approach must be adopted. One post hoc approach by White *et al.* [140] identified the barriers blind players face when accessing Second Life and proposed their on-going haptic solution, which allows visually impaired players to navigate their avatar using force feedback. Different objects can be distinguished through different vibration frequencies.

We verified that it is impossible to access Second Life using a screen reader as it lacks any textual representation [1] that can be read by a screen readers. Second Life also lacks features like self-voicing. Blind people use screen reader for various applications including word processors, email programs, and spreadsheets, and they usually tailor the settings of their screen readers to their personal preferences such as rate and pitch, something that self-voicing games may not support. Though Second Life recently started offering voice chat, the majority of users use text chat to interact with other avatars. Carter and Corona [18] proposed applying 2D Internet accessibility to a 3D environment, since blind people can use 2D Internet highly successfully via screen readers[19]. Rather than trying to shoehorn an interaction mechanism designed for sighted users to a solution that is accessible to visually impaired users, we opted for a radically different solution, an alternative text interface that works with a screen reader and may potentially be a better way of making virtual worlds accessible.

One of the limitations of a screen reader is that it can only provide speech from one source. This imposed a problem because virtual environments and games typically produce feedback from different modalities (visuals/audio) and sources (objects/avatars) simultaneously. An interface was required that could extract a meaningful textual representation from the Second Life in a linear form while at the same time accepting input iteratively.

Nevertheless, the types of interaction we were looking to implement was surprisingly common to text-based adventure games. Virtual worlds have their origins in Multi-User Dungeon (MUD) games [23] which are server-based games that accept network connections from multiple users. Feedback and input were provided in a text only form because these games were played in a terminal window over a network connection decades ago. Yet, players could still interact and chat with each other using command lines. MUDs have their roots in single player text adventure games with Zork [72] being one of the first such games that focused don exploring dungeons, interacting with characters and fighting enemies. Figure 4.1 shows Zork interface and some sample game play. Zork offers a number of interaction mechanisms that we deem particularly useful for the proposed text client. The interaction mechanisms used in text adventure games and MUDs have proven to be fun, efficient, and engaging and cater to the constraints imposed by the use of a screen reader. As a result, we decided to model our text client for Second Life after Zork.

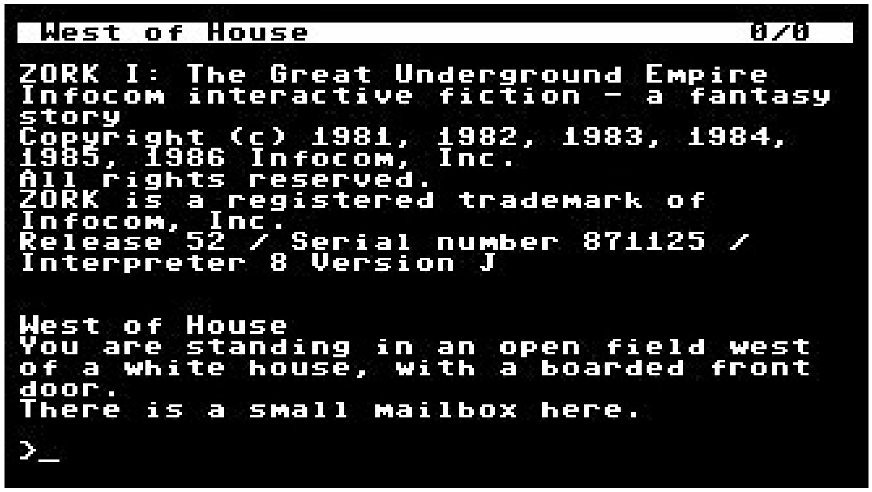


Figure 4.1: Zork Text Adventure Game

4.3 TextSL: A Second Life Interface for the Visually Impaired

Supported by a small NSF grant for exploratory research, an application was developed called TextSL that could mediate between a screen reader and Second Life. To develop accessible solutions, it is important to involve users in the initial design and requirements collection, because blind users did not formerly have any access to Second Life. We decided that performing pilot studies with a prototype would yield more valuable feedback. Our goal was to investigate whether it was technically possible to extract a textual description from Second Life that could be read by a screen reader. We initially explored whether it was possible to build the required interface on top of the existing Second Life viewer. Digging through the source code, we learned that the Second Life world is partitioned into chunks called sims. To avoid sending all of Second Life's content to each client, the client only requests the content of the sim the user's avatar is on. To render the Second Life world, the viewer asks the server to send all the identifiers of all the prims (which are the building blocks used to create objects) on the users' sim to the viewer. After receiving the identifiers, the viewer requests all the information (location, textures, etc) from the server and starts rendering all the objects on the sim. Since most of the viewer's functionality is related to rendering resource intensive graphical interfaces designed for sighted users, it seemed inefficient to build the accessible interface on top of the viewer. Furthermore, to build a textual extraction we only need to know the identifiers of all the objects on the user's sim so we can query the server for more information on each object such as its name and description. A separate text-based interface now seems more viable and promising.

During our analysis, a C# library was made available called libsecondlife [75], which allows direct communication with the Second Life servers and offers a simple API for the most common functions in Second Life. Libsecondlife provides most of the functionality required for our text client. The first feature we implemented was a simple console window that allowed the user to issue a *"login"* command to login their avatar. After issuing this command, textual feedback (success/fail) is provided through the same window that could be read by the JAWS screen reader. We then continued our technical analysis by adding features that would allow for some basic form of navigation and interaction in Second Life to further explore the educational and social benefit offered to the visually impaired and blind people.

4.3.1 Navigation

A basic form of navigation is offered using the *"move direction distance"* command. The user can set a direction (forward, back, left, right, up, down) and a distance either specified as a number or a word. Our interpreter also allows for using prepositions e.g. *"move to the left with ten"*. Zork only allows discrete movements with a distance of one, whereas we allow the user to specify a distance. If no distance has been specified the user moves with one game meter (a unit of measure in Second Life). Moving up puts one in flying mode. When the avatar's feet touch the ground flying is automatically disabled. Navigation is relative to the orientation of the avatar, which is different from Zork as it moves the user around on a discrete grid aligned with compass directions and ignores the direction the user is facing. In Second Life we cannot ignore which direction the user is facing. Other avatars may perceive it as rude if an avatar talks to them while not facing them. Users can also move towards an object or avatar by issuing *"move name"* where "name" is the name of an object or avatar. The user will automatically face the object or avatar it is moving to. As an avatar will be facing different directions, a positioning system relative to the avatar is used. In addition to moving, users can teleport to another sim using the *"teleport"* and specifying the name of the sim. Users can also follow other avatars by typing *"follow"* and the name of an avatar. Basic pathfinding is implemented to avoid any obstacles. Occasionally the avatar may get stuck such as when entering a building. We continuously monitor the user's movements and if we detect it gets stuck we teleport the user to the desired location.

4.3.2 Interaction

Any public conversation between avatars within a certain range of the user's avatar is displayed in the console window, similar to the Second Life viewer, and is read out by a screen reader. What another avatar says is preceded by their name e.g. *"Coool Magic says...."*. To avoid overwhelming the user in crowded areas, users can mute all conversations or particular avatars using the *"mute"* command that takes as a parameter *"all"* or an avatar's name. Users can talk in public using the *"say"* command or send private messages to other avatars using the *"IM"* command. Like going to a classroom in real world, the user is able to use *"sit"* command to sit in the classroom on defined objects like chairs. The *"touch"* command is also defined in TextSL though no output is provided yet.

4.3.3 Explore

A *"describe"* command is implemented that provides detailed information to users, further helping them explore Second Life socially. In order to build a textual representation, we needed to know which objects and avatars are located around the user. Objects and avatars have properties such as a name and description that could help us build a narrative. Objects in Second Life are constructed of smaller entities called prims. Each sim supports up to 15,000 prims, which in theory could define as many as 15,000 objects. In practice, an object is usually comprised of a number of prims. Prims without a pointer to a parent identifier are objects. Information in virtual worlds is constantly changing: objects may move, new objects may be created and avatars may come and go. That said, we must periodically request the server to send us all the object and avatar identifiers of the sim the user is logged in on. Now a textual representation can be built by requesting more information on each object such as its location, its name and description, pointers to parent identifiers and any other relevant properties such as whether the object allows interaction or has embedded media. Because we don't request the server to send any media, such as textures, sound or video, a significant amount of bandwidth and extra rendering and waiting time is saved. To build a textual representation we identified two problems:

1. Many objects lack a (descriptive) name or a description. As many content creators figure that you can see what an object is many objects simply have their default name - *"object"*, which essentially makes this object invisible in TextSL.

2. Virtual worlds contain large amounts of content. Depending on where you are the user's avatar may be surrounded by ten to hundreds of objects. Many objects just serve as decoration like trees, bushes, lampposts. Though these objects may be interesting to the user, there are simply too many to list them all. Similar to how Zork keeps the amount of feedback provided digestible, some mechanism is needed to abstract, prune and prioritize this information to keep interaction with our text client fast.

Pruning Method

A basic form of pruning is implemented by computing a value for each prim on the sim using the following function:

$$F = root \times distance^{-1} \times namelength$$

To avoid users interacting with parts of an object (such as the wheels of a car) any prim with a pointer to a parent identifier is assigned the value 0 by a root function. Though our function essentially works on prims, we refer to them as objects in further discussion to keep things simple. We cull all objects that are located up to 30 meters away from the user's avatar, which is about the same range a user could see on a screen in Second Life viewer. We then compute the inverse of the distance between the user's avatar and the object to make closer objects more interesting than objects far away. To get rid of objects lacking an accurate name description we multiply by a length function which assigns a value based on the length of the name of the object, assuming that a longer name implies a more accurate textual descriptor. The length of the name of objects called called *"object"* is assigned the value 0. Short names such as *"cat"* are not penalized and the value of long names (more than 15 characters) is capped. This value function filters out most objects that are not of interest and is easily extendable to include other objects of interest. Similar function is used for avatar except it only depends on the distance between avatars.

Aggregation Method

In addition to pruning some form of aggregation is used by grouping the objects and avatars found. When a user initially issues *"describe"* we return: *"you are at name_of_sim, you see X objects and you see Y people."*. The number of objects and avatars returned through TextSL are capped by customizable variables. Currently both numbers are default to 5 to keep our information digestable. After the number of objects and avatars are returned, users can query for more information by issuing *"describe people"* or *"describe objects"*, which reveal the names of the objects and avatars. Users can get even more information on specific objects and avatars by issuing a *"describe name"* where name is the name of an object or avatar. Specific descriptors have been implemented to return detailed information. For an object, it returns the owner, its description (if available), whether the object makes sound and whether it is for sale. For avatars, it returns a description (if available), the avatar's pose, when the avatar was born and which languages the owner of the avatar speaks. Partial name matching is implemented to avoid players having to type in the name exactly as it may be long or exotic.

A *"help"* function is available that lists all available commands. Help on a particular command and examples on what parameters a command may take are provided when typing in *"help name"* with the name of the command. For example, "help describe" will return instructions related to the command "describe"; *"help mute"* will return: *"To Mute conversations, type "Mute avatar name", to Mute all, type "mute all", to unmute an avatar issue the same command to toggle muting.* Figure 4.2 shows a scene in Second Life. Figure 4.3 shows the exact same scene interpreted in TextSL.

Figure 4.2: Scene in Second Life

Figure 4.3: Same scene as represented in TextSL

4.4 Pilot Study

To validate our approach a pilot study was conducted that involved 4 blind as well as 8 sighted participants. Our study has two specific goals:

- **Accessibility:** is TextSL accessible? Virtual worlds lend themselves to doing remote evaluations, however, one of the risks of remote evaluations is that user experience specialists often lack detailed understanding of how people with disabilities use their assistive technologies [91]. To mitigate this risk we perform local user studies with blind participants to verify our assumptions on how blind users interact with their computer using a screen reader. Questions that we seek to answer are: Is TextSL accessible with a screen reader? Can they control their avatar through a text interface? Is the amount of feedback provided overwhelming.

- **Usability:** Is TextSL usable? A user with a disability should be able to perform a task with the same efficiency as anyone else. TextSL offers a number of basic

> commands allowing the user to navigate their avatar, do some basic exploration of Second Life and interact with objects and avatars. TextSL aims to offer the same functionality that is available in the Second life viewer. Blind users, however, are unable to access Second Life using the viewer. Therefore in order to do a qualitative comparison of the functionality implemented in TextSL, we decided to include sighted participants. Sighted users will not use a screen reader to access TextSL but will see the text output. Though sighted may be able to process visual information at a higher speeds than blind users are able to process audio, we think a sighted user's inexperience with the use of screen readers poses a bigger threat to the accuracy of this user study. Some of the questions we seek to answer are: Will sighted users be able to perform the same tasks in both clients? Will they be able to achieve certain goals within similar time frame for efficiency? How much difference do they experience and is this acceptable? We also collect suggestions to improve our client from as well as blind as well as sighted.

The study was performed on our private island in Second Life, where we had similar setup as some in the rest of the world. Access to the island was restricted to avoid any interference from other avatars during study. Another benefit of a controlled environment is that we can make sure all objects on our island had accurate names and that we can keep the number of objects to a minimum. The user tests require both groups of subjects to perform a set of tasks. The objects and avatars on our island were arranged in such way to accommodate for these tasks. To provide sighted users with a context to evaluate the functionality implemented in TextSL, sighted users perform the same set of tasks with the Second Life viewer before doing the same tasks in TextSL. Both tests are designed to be identical with regard to number of objects to be recognized and spatial layout of objects, but with different objects to be recognized, to allow for an optimal comparison between both clients.

As emphasized throughout this chapter, our interests fall into the following two categories and so do our test setups.

- **Educational Use:** Second Life has becoming increasingly popular as an education forum. We demonstrated this by teaching TextSL commands as an educational topic. Instead of giving each participant face-to-face training in a real classroom, we scripted a tutorial session, where the first task each participant was asked to do is to learn TextSL via the tutorial hosted in TextSL itself. Users would not be able to perform the rest of the tasks successfully without learning the commands for TextSL.

- **Social Benefits:** In order to socialize within Second Life, one needs basic forms of communication (chatting), navigation (moving around and locate avatars/ objects), and interaction and exploring (describing physical surroundings). The rest of the tasks is designed in a way to fully test these features.

Design/Methods: Within the standard Second Life viewer, the participant first receives a tutorial on how to move their avatar, interact with other avatars and objects after logging in Second Life. Similar tutorial was scripted in TextSL for its participants (sighted and blind users) to follow. During the tutorial the user has to perform a number of simple tasks such as navigating their avatar to a particular location and interacting with an object and an avatar. Before starting the tutorial blind participants are allowed to customize the screen reader (JAWS) to their personal preferences. The following four tasks are given to all participants in both viewers after the tutorial is done.

1. Move to a location and identify a number of objects.

2. Move to a location and interact with an object.

3. Repeat of task 1 for a different location and objects.

4. Move to a location and chat with another avatar.

The answer to each one of these tasks is previously known and written into the application as a pass/fail condition by the designer. The user has a limited amount of time to complete each task. During this process the user is able to issue any command but to pass a task he or she must enter the right pass condition. This pass condition can be the name of an object that must be recognized or the result of a particular command. If the user fails to complete the task it will timeout and move onto the next task. The tasks were designed independently in a way that failing one task would not affect the next task. After the tutorial and the tasks were complete the participants were allowed to play with TextSL and explore the island. A questionnaire was filled in at the end. The evaluation was split up in two parts:

- **Quantitative** analysis: the following data is collected: total time it takes to perform each task and tutorial, number of tasks successfully completed, total number of commands used and number of unrecognized commands. Specific screen reader (such as rate and pitch) settings are also recorded.

- **Qualitative** analysis: subjective feedback on the usability and accessibility of our approach is collected through a questionnaire. Different questionnaires are used for sighted and blind subjects. Questionnaires in Figure 4.4 for the blind focus on identifying how the blind use screen readers in general. We also ask them how they evaluate their experience with using a screen reader with TextSL and how this experience compares with other applications such as a browser. Questionnaires, shown in Figure 4.5, for sighted participants focus on determining a qualitative usability comparison between the functionality implemented in TextSL and Second Life.

Results: Our participants included 4 females and 8 males with ages in between 21 and 51 years. All our subjects had experience with computers and with playing games. All blind subjects used screen readers for various applications but none of them had ever played a game with a screen reader. Two of them who used screen readers with browsers on a regular basis used them at rates between 50% and 55%. The other two subjects who did not surf the web used them at rates around 30%. During the test, all the subjects were able to complete the four tasks successfully. A

Usability Text SL.

I found it easy to move my avatar around in text SL. ☐ Yes ☐ No
because____________________

It was easy to find out what objects and avatars were around me ☐ Yes ☐ No
because________________________

Text SL allowed me to chat with other avatars easily ☐ Yes ☐ No
because_______________________

Overall

The amount of feedback provided by TextSL is: ☐ too little ☐ just right ☐ overwhelming

Having to type in commands to control your avatar is ☐ easy ☐ difficult

If so list why____

Is accessing a virtual world using Text SL similar to accessing the web?

Figure 4.4: Part of Questionnaire for Visually Impaired Participants

	Agree				Not Agree
Text SL allows me to move my avatar around to the same extent As I am able to move around in the regular viewer	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
Text SL allows me to find out what objects and avatars are around my avatar to the same extent as in the regular viewer	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
Text SL allows me to interact with other avatars to the same extent as in the regular viewer	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

Figure 4.5: Part of Questionnaire for Sighted Participants

few erroneous commands were typed in by 1 sighted participants and by all four of our blind participants but these are no cause for concern as they were merely typos. In one case typos were caused by JAWS where the input cursor would randomly move to the beginning of the input box. All participants played around with TextSL after test. From the transcripts we noticed that when in conversation users sometimes forget to put *"say"* in front of that they want to say which leads to an unrecognized command error. The *"help"* command can be used throughout the entire test. We receive an average of one *"help"* command typed during the test per sighted participant. We observed that some of the sighted participants would consult previous command outputs rather than using help as they were able to see the previous output. Blind subjects did not use any *"help"* during the test which could imply that blind were able to memorize the commands from the tutorial.

Figure 4.6 shows the average performance times for each one of the test tasks for both clients and for sighted (who used both clients) as well as blind subjects. We understand that in virtual world it might not be as critical as in games in terms of how quickly a player finish each tasks. However, we believe that demonstrating the speed helps evaluating the efficiency of each task completion. The average execution

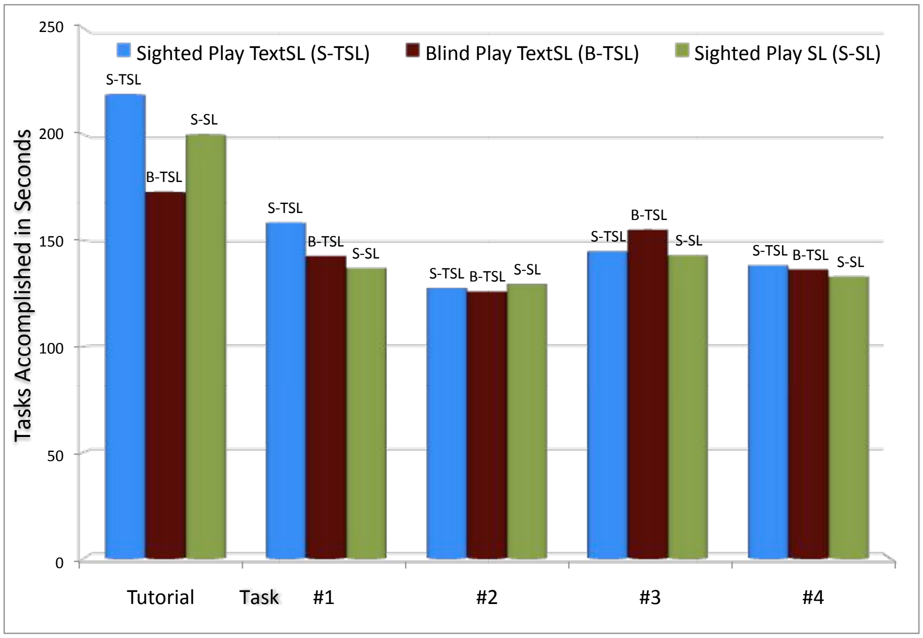


Figure 4.6: Average Task performance times

time of tutorial and tasks 2 and 4 are very similar for both clients and both types of subjects. Only the tasks related to navigation and exploration shows a slight worse performance for TextSL. The avatar in both clients moves with the same speed and the amount of distance that needed to be travelled was the same for both clients, so the difference in performance may be explained by that it may be easier to recognize an object visually than having to issue a command to find out what the object is.

Figure 4.7 shows a relative comparison of the usability of the features implemented in TextSL compared with the Second Life viewer by the sighted users on a scale of 1 (not the same) to 5 (the same). Some of the suggestions we received to improve the usability *"I can only do one thing at once"* and *"why do I have to type in the same command multiple times to find the name of an object?"* were related to the way we designed the interaction of TextSL to accommodate the use of a screen reader. Other suggestions such as *"I can't move diagonally"* or *"I can't find out where the object is"* were simply not implemented or activated for this user test, which may explain why the usability of TextSL for many features is ranked lower than the Second Life viewer. Two features that were ranked as usable as the viewer were communication and tutorial, which proves TextSL is a good education tool and support the most important aspect of MMO's: social interaction as well[51].

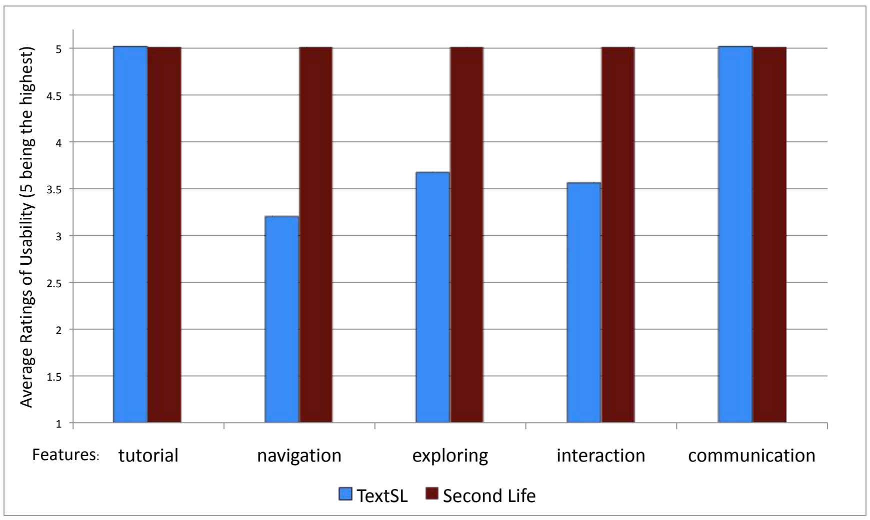


Figure 4.7: Usability Comparison TextSL vs SL

The result from the rest of the questionnaire was also highly promising. All subjects (with the exception of one) found TextSL easy to use. All subjects found the amount of feedback provided adequate (with the exception of one sighted subject who found the amount of feedback too little and one blind subject who found it overwhelming). Overall the results of the user studies were positive and the many suggestions we received will be taken into consideration in the future developments of TextSL.

4.5 Observations

The development of TextSL and the results of the study have provided us valuable insights into how to make virtual worlds blind-accessible. Screen reader support has proven to be a promising option, especially when the user can skip the shock from a self-voice engine and reduce initial learning curb using screen reader with their customized settings. Though compromises have to be made transforming a 3D virtual world to a 2D text-based interface.

A number of problems presented themselves during our study including navigation efficiency and effectiveness. We made the following observations and point out some areas for future research:

4.5.1 Meta Information

Second Life has been very successful in attracting users that contribute to its virtual world by creating content. However, one of the biggest problems that we identified is the apparent lack of meta data for objects. Objects in Second Life can be given a name and a description but as some content creators figure that you can see what the object is, they frequently leave these properties to their default values. To build a meaningful textual representation objects named *"object"* are just as useless as a web image lacking an alt tag. This problem is actually worse for virtual worlds as the meaning of a web image can sometimes be derived from the text surrounding the image or even from the filename [9] which objects in Second Life both lack. We also noticed that certain objects have long names. Stores who sell virtual objects sometimes put ads for their store in the name and description fields. The lack of meta information is an inherent problem of the post hoc approach that we had to take towards making Second Life accessible, which distinguishes our approach from other efforts to make games accessible [6, 46, 127, 139]. Had accessibility requirements for the blind been taken into account when Second Life was designed, a mechanism could have been implemented that would enforce the user to provide names for objects, similar to how web images could be made accessible, if html standards would agree on not to render images without an alt tag. From a post hoc perspective the lack of meta information is very challenging to change. The Second Life world already contains millions of objects and assigning proper names to all existing objects may require a significant human effort. An external name database has been implemented that can store meta information as only the owners of objects are able to change the name of their objects. TextSL communicates with this database transparently through the instant messaging protocol. An avatar acting like an agent is continuously logged in and provides various database related services to all the text clients by parsing messages that are sent to the avatar. If an object with a bad name is encountered in TextSL its identifier is queried against the name database and if an entry is found the name and description are sent to TextSL.To add meta data we propose the following features:

- **Automatic labeling:** unlike two dimensional web images that may contain multiple objects of interest, objects in virtual worlds are defined by themselves and have more properties available that could potentially allow for an approach based upon recognition. For every object we have properties such as dimensions, topology and composition into prims or objects. Based on a simple set of heuristics we were already able to recognize houses and billboards with acceptable accuracy. Unidentified objects whose identifier does not have an entry in the name database could be automatically recognized using a set of heuristics and if a match is found their entry is added to the database. TextSL users then act as spiders automatically adding meta data to Second Life.

- **Manual labeling:** this approach allows sighted users to add meta information to objects. As virtual worlds contain massive amounts of content it is important to provide users with an incentive. Along the lines of "games with a purpose" that have helped label the web [134], we propose a scavenger hunt game that helps users label objects in Second Life in the next Chapter.

4.5.2 Narrative

When a user issues a *"describe"*, TextSL provides the number of objects and avatars around you. When compared with the feedback that Zork provides *"...you are standing in an open field west of a white house... "* one cannot ignore the fact that the feedback we provide is rather limited. One reason for this is that in virtual worlds we have to deal with a much larger number of objects than in Zork. For a virtual world where all objects have meta data available, the number of objects found will be even larger and the need for a mechanism that can reduce some of the feedback becomes evident. Though all of our blind subjects found the amount of feedback provided adequate, we noticed that two of them used their screen readers at significantly higher rates which may indicate that the amount of feedback that is acceptable varies per user. One user may be comfortable with up to 30 words of feedback whereas someone else may allow up to 80 words. Both subjects found the amount of feedback provided adequate and we did not test for larger amounts of feedback. We propose to allow the user to modify the amount of acceptable feedback measured in words. A mechanism is proposed that will automatically provide the right amount of feedback depending on the amount of objects around the user and the amount of acceptable feedback. Concise feedback facilitates interaction though we must avoid trivial interaction as much as possible. If there is only one avatar in front of the user there is no need for the user to have to issue multiple commands to find out what that avatar is doing. The result of a *"describe"* for that case could be *"you see Coool Magic and she is sitting down in front of you"* as long as the amount of feedback is under the acceptable world limit. When more objects or avatars are around some mechanism is required to reduce the amount of feedback. For now a very simple mechanism has been implemented that groups objects and avatars but we seek to extend it as follows to provide a more usable narrative. (1) Grouping objects, instead of *"you see a tree, a cat and a tree"*, we list *"you see two trees and a cat"*. (2) Abstraction of object names, instead of *"you see twenty trees"*, we list *"you see a forest"*. Instead of *"you see an* *oak and a spruce"*, we list *"you see two trees"*. Such abstractions are not possible without some sort of taxonomy which may not be easy to create. (3) Detail: after the feedback has been sufficiently reduced and the amount of feedback is under the acceptable word limit it could be expanded by adding spatial information or specific descriptors, such as the location of objects or what the avatar is doing. These are all transformations that work on an initial set of object names and that expand or abstract it into a more descriptive narrative. Transformations are applied only if the resulting feedback keeps within the user specified word limit.

However, none of the narrative mechanisms could be implemented without the accurate meta information of objects in the virtual world. This bares further contemplation that, a mechanism that will fill out the missing meta information needs to be developed, in order to push the idea of relying on screen reader or improving accessibility feature of Second Life. A manual labeling game is presented in Chapter 5 aiming to solve this dilemma.

Chapter 5

Case Study III: Seek-n-Tag

Virtual worlds have not been built for the blind. Unlike the Web, where blind users can access using screen readers, virtual worlds do not provide screen reader support. One of the major problems is the lack of metadata for virtual objects. On the Web, metadata are generally interspersed into original documents to improve web accessibility. The use of metadata to improve web accessibility is very widespread. A good example of one of the simplest and most useful forms of metadata that benefits visually impaired people browsing websites is the alternative (alt) text added to image tags, useful since screen readers are not able to interpret images. The same tactic could be used to solve accessibility issue in virtual worlds if this metadata were widely available. In general, there are two types of metadata: internal metadata that is embedded in original documents, and external metadata that is stored separately, but associated with the original documents [64].

For this chapter, we picked Second Life to represent virtual worlds in general, due to its popularity and open source nature. As a long-term goal, it might be possible to make our game portable to more than one virtual world. In Second Life, internal (embedded) metadata can only be amended with the appropriate permissions, in most cases authorized by the land owners. External metadata, on the other hand, can be created by anybody. The power of external metadata is that people with different interests or requirements can act on their own without affecting the original content or requiring the content owners to be involved. There are two potential approaches for adding external metadata to objects in Second Life as proposed in

Section 4.5.1: automatic labeling and manual labeling. However, with the huge quantity of information and amount of missing information on objects in Second Life, we are far from creating any heuristics for computers to identify objects successfully. For this reason, we favor the manual labeling approach, which can provide accurate information for objects.

Second Life residents are provided with a wide variety of tools to communicate, collaborate, and create. People participate in social events, exchange merchandise, attend classes for wide array of topics, play games, and more. Simply giving users tasks like manually labeling objects does not sound very interesting, and is unlikely to draw attention. Instead, we disguised this task as a game, very similar to a scavenger hunt game. It is not a new concept to use games as a medium for people solving critical problems. It is well-known as designing "games with a purpose" (GWAP) [135].

The rest of the chapter is organized as follows: Section 5.1 discusses related work and provides more background on the game we created. Section 5.2 presents our implementation of the game. This is followed by the results of our pilot study with 6 players in Section 5.3. Section 5.4 discusses some unsolved problems with proposed solutions for future work.

5.1 Related Work

Second Life has been commonly used for education [65, 62, 99] and business purposes. Previous work has been done towards making SL accessible to people with various disabilities including cognitive [104] and visual impairments [38, 140].

Numerous research on making virtual world accessible to the visually impaired users [18, 19, 97, 127] has already advanced this field. However, a common problem was identified within this: virtual worlds lack consistent and necessary meta information. Our current assistive technologies rely heavily on accessing object information to provide information to visually impaired user via audio. Without the metadata for objects, the current technologies can provide little use.

Two approaches could be implemented to have a virtual world with useful metadata: **original**: Rules could be enforced in the development of the game that for every piece of created content, meta information needs to be descriptive and precise. Some guidelines or instructions could be distributed to content owners, aiming at providing strong internal metadata. Obviously this puts a lot of demand on owners and is less likely to be done correctly or consistently. **assistive**: external metadata could be added to aid existing, internal metadata. This is a more flexible approach and likely to be achieved by either an automatic or manual labeling process.

5.1.1 Web Solutions

The assistive approach is widely adopted in web accessibility. There are many research projects in this area. Several significant manual or automatic schemas have been developed including *Social Accessibility* [122], *WebInSight for Images* [11], and *SADIe* [53]. With the idea of GWAP, the following games are the most influential solutions to improving image labels on the web, where people play and perform basic tasks that cannot be solved or automated by computers.

- **The ESP Game** [134] and **the Google Image Labeler** [48]: are almost identical web image labeling games except for the graphics and minor game rules. In both games, two players pair up to try to come up with matching names for the same pictures randomly extracted from websites, within a certain amount of time. Points are given based on the matching words.

- **Phetch** [136], is another game produced by Luis von Ahn. It annotates images with descriptive paragraphs. It is designed to be played by 3 to 5 players, with one of the (randomly selected) players as the Describer, while the others are the Seekers. The data generated from this game are proven to be more accurate for web images to blind web users. Unfortunately, it was discontinued in mid to late 2008.

The gameplay utilizes competing and rewarding system, which made these game-like applications fun and intriguing.

5.1.2 IBM Solution

In an effort to make virtual worlds for the blind [56], IBM started a mechanism that allows sighted users to add descriptive information to virtual world (Second Life). A special annotation device is “equipped on the avatars of sighted users. When clicking on the objects in SL, a Web browser is triggered, allowing sighted users to enter accessible attributes, such as custom name and short and long descriptions of items or locations. The information is then saved in an external database. Figure 5.1 shows the webpage that is brought up every time the user tries to label an object. This technique could greatly improve metadata in virtual worlds. However, it would take a long time to fill in the gap with such great detail. Therefor we do not consider this solution to be scalable enough to meet the needs for Second Life or other virtual worlds.

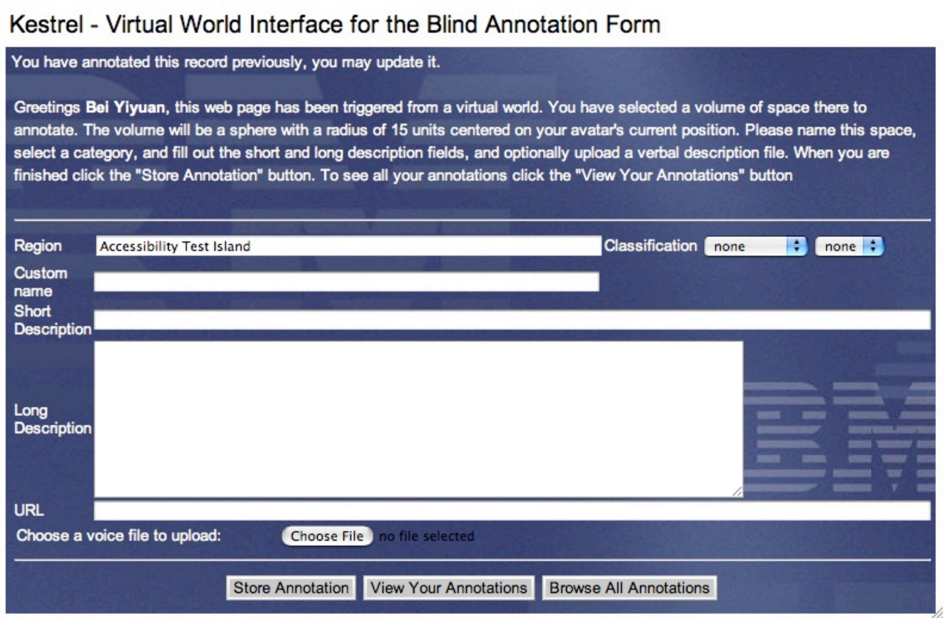


Figure 5.1: Webpage for SL Annotation

5.2 Method

Previous research has already recognized that SL lacks metadata. This directs our investigation to find out what type of objects are most popular in SL and what is the percentage of objects with missing or bad labeling. We are hoping that detailed statistics for this information will make SL residents realize the importance of providing meaningful data and further encourage them to help improve the current status on metadata in SL.

5.2.1 Data Collection of Objects in Second Life

We built a virtual bot who has a basic capability for wandering around and finding locations and objects. The implementation is as follows: the bot selects a random direction, starts walking and collects information about the objects it encounters along the way. Object attributes such as name, location, shape, and UUID (unique identifier) are then sent back to our database. Other information, including the number of prims and parcels on the sim/region, are also gathered by the bot.

The Bot

According to SLBuzz.com [114], there are 13543 sim/regions in the history of SL. Although a lot of those regions are not available any more, the list did contain almost 150 regions that are active almost all the time. We then sent our bot on a mission to collect data from over 700 different sims (selected arbitrarily) from that list. It is not possible to follow all SL regions, because: i) SL is growing fast, compared with 700 regions from 2006, and ii) a lot of sims in SL are unreachable from time to time [147].

Compiled Data

Over 4 weeks period of time, our bot collected over 423,000 prims, which constructed 120,000 root objects that we analyze. Out of the 700 sims we fed into the program, we were able to reach closed to 500 of them, since some of the sims might not be available anymore. There were about 5% to 65% of root objects collected per each of those sims. Depending on the travel range of our bot, the percentage of the objects collection on each sim varies. Some area of the sim could have limited access, which results in a lower number of collection on those sims. In general, we gathered a mean of over 25% per each sim. We continued the study with the most used object labels and conduct the following findings using data in Table 5.1:

1. Objects are named *Object*. Within the 120,000 collected root objects, 36% of them are labeled *Object*, ranking the first of the most used names. We even saw regions with 70% of objects labeled that way. This number is more than we expected and further proves the fact of poor labeling in SL.

Object Label	Potential Usage in Second Life
Object	Anything
ShapeBlock	Part of a flight control center
BuiltusingSkidzPrimz	Part of a roof for a building
20x20x.5	Part of a wall separator
King Prim	Unknown (objects deleted from the sim before analysis)
VirtuPrim	Part of a building with various shapes of objects
PANELLO	Part of raised wood floor

Table 5.1: Most Common Used Labels

2. Objects are grouped poorly in Second Life. The content creators group their prims to form the root object. However, to the user point of view, those objects are still too fine to be used for navigation and exploration in the virtual world. As shown in Figure 5.2, a piece of architecture could consist of a great number of adjacent objects with the same name. Ideally, those objects should be grouped together and labeled as one object.

   In the real world, people treat a building as one object, even if it is composed of many walls, windows, and a roof. In virtual worlds, however, each of those component is retrievable as individual objects, which is also the reason that SL consists of a huge number of objects. Table 5.1 explains why those objects are so frequently used and where they have been referenced. We also find out that the same label could be used for different shapes of objects to form a higher

5.2.1: ShapeBlock 5.2.2: Panello

Figure 5.2: Same Label Objects

level abstraction. For intance, “VirtuPrim” is labeled on all objects that serve as beams for a building, regardless of whether a beam is shaped like a stick or a block.

Research Goal

The purpose of our game is to improve the semantics of the objects in virtual world by i) labeling the missing name objects, ii) building a taxonomy, and iii) possibly grouping adjacent objects with higher granularity. It is possible that not all objects need to be correctly labeled, especially if they are just part of a higher class, as long as they are grouped correctly. An object could also be namely quite differently depending on the situation. For example, *motorcycle* and *car* are two different labels, but they could also be labeled *vehicle*. The collected object labels help to build a taxonomy that group objects by their categories. Consequently, it will ultimately help for better searching and navigation around Second Life, especially if the information needs to be given in a concise manner.

5.2.2 Seek-n-Tag: A Virtual World Labeling Game

We started with the goal of creating a compelling tool for improving the metadata stored in SL. With the existing systems in SL, relabeling components is not compelling because it is not “fun”. As such, SL residents rarely correctly label or relabel components.

Conceptualization

Generally speaking, a prerequisite to having "fun" in a game is establishing a rewards system and often a sense of competition. The players of Seek-n-Tag are initially shown a countdown clock, starting with 5 minutes remaining. Individual tasks must be completed within 30 seconds and the user is rewarded when a task is completed earlier, by being given more time on the countdown clock. During the game, the player needs to "tag" objects that match names chosen by the software. For example, one of the tasks is: "Go find a house". The faster the user tags an object, the more time he will gain per round. Also, if the object is tagged correctly, 50 points are given. The game starts with easier tasks and gets progressively more difficult. An example of an easier task might be very general, for instance, finding a category such as building. A more difficult task tends to be more specific, with less common components, for example, a gazebo.

Prototyping

Programming is achieved with the Linden Scripting Language (LSL), which provides a wide range of capabilities. It allowed us to quickly implement an alpha version that gives players an opportunity to play the game and collect feedback on their experiences. The Seek-n-Tag game is stored in an object that is transferable between residents in Second Life. The players, avatars in SL, simply "attach" such an object to themselves to start playing the game.

5.3 Evaluation

Evaluation was performed through a study that was conducted in a controlled environment in Second Life (Accessibility Test Island). A controlled environment was required for these tests because the more general SL environment changes too rapidly and is unpredicatable. However, to maintain much of the authenticity of a real SL experience, the objects we used in our experiments were directly copied from existing components in SL.

5.3.1 Study Design

There are many ways to label objects in SL. The most common method is to click on an object and then manually type in a name. We call this process, in general, "object naming". In Seek-n-Tag, the user is provided with a list of object names/categories, so there is no need to type. A user must simply "tag" (or touch) an object that represents whatever the current name/category states.

We created two environments on our island: A and B. Each location contains 25 objects, pre-labeled *Object*. In each, A and B, the same mixture of object categories were used: *Plant*, *Animal*, *Furniture*, *Vehicle*, and *Structure*. Specific object types within those categories, and locations of objects differed however. A total of six participants, between 24 to 39 years old, who have never played our game, were recruited to perform the following tasks:

1. **Object Naming**: The user is asked to come up with a name using their own words and choose one the five categories provided, for each of the 25 objects.

2. **Seek-n-Tag**: The user is given names and asked to find an object to match each. Same names might be presented multiple times since multiple objects could match it. Figure 5.3 is a screenshot taken during the tagging process of the game.

Participants, separated into two, three-person groups (as illustrated in Table 5.2), started in either environment A or environment B. They first performed Object Naming then switched environments and played Seek-n-Tag. The goal of this study is to

Table 5.2: Task, Group, and Location

Task	Group #1	Group #2
Object Naming	A	B
Object Tagging (the game)	B	A

Figure 5.3: Seek-n-Tag

find out: i) if users behave differently between the Seek-n-Tag and the traditional Object Naming process. ii) if users prefer Seek-n-Tag to label objects.

5.3.2 Results

Each of the 50 objects is named up to three times by three different users, depending on how many times a category is given to a user. We consider the application of a label to be correct as long as the name and category match between Seek-n-Tag and Object Naming. The following is our quantitative data collection analysis:

- All of the objects got labeled the same set of names in Seek-n-Tag as in Object Naming. For example,“ Object 1” is named as couch and sofa by the first group, the label it got in the game would also be either couch or sofa. This proves that people behave the same in naming as well as the game. If we assume players name objects correctly, they would also tag accurately.

- Objects are labeled the same categories in both methods. For example if we have 5 objects that a named under category ‘vehicle’, we also collected 5 objects for the same category in the game labeling process.

5.3.3 Usability

Current heuristics for designing and evaluating games are mostly focused on three aspects of usability in games: interface (controls and display); mechanics (interacting with the game world); and gameplay (problems and challenges) [34]. In addition, a game also has to be fun to play. However, enjoyment is difficult to quantify, depending on the exact implementation and design of the game. Thus, we included "Enjoyment" in the questionnaire and used the consistent rating system to measure all four of the aspects for this game. Figure 5.4 shows questions we ask our participants to fill at the end of both processes and the ratings we received. We are at the early development

0 = N/A, 1 = Strongly Disagree, 2 = Disagree, 3 = Neither Agree nor Disagree, 4 = Agree, 5 = Strongly Agree

Aspects	Elements	Rate (1-5)
Interface	The graphical design of the game is esthetically pleasing.	4.4
	The visual information displayed is not overwhelming.	4.2
Mechanics	The interaction with objects is simple and efficient.	3.2
	The game control is effective.	3.6
Gameplay	The game provides good amount of challenges.	3.2
	The rewarding system is fair and motivates the player.	3.6
Enjoyment	The game is fun to play.	3.6
	I will play the game in the future when it's available.	3.6
	Labeling with seek-n-tag is better than naming in SL.	4
Overall		3.7

Figure 5.4: Questionnaire on Usability

stage of this game. There are several game elements, like competing with others, we could add to the game to improve our gameplay ratings.

5.4 Discussion

As identified in the web accessibility field, various metadata collection services are disconnected from one another, isolated in separate tools, stored in disparate repositories, and represented in incompatible formats[64]. These limitations also apply to our approach. We need to create an open infrastructure through which it is easier to share resources. For now however, we could build an object, similar to our Seek-n-Tag game object within Second Life, that residents can wear to retrieve metadata from our system, directly from the standard SL viewer.

We were unable to achieve and validate the improvement of object grouping in this study. Our game does not currently collect enough information for each object. With better object labels and a taxonomy of names and categories, we can automatically label adjacent objects with the same original labels to the new tags based on their location. In the future, we can just group those objects automatically in our database, if they share similar attributes such as names and coordinates.

This chapter describes a game that encourage users in Second Life to manually label and categorize objects with reasonable terminological control (normalization of word inflections). The data generated as a side effect of the gameplay also solves the bad labeling and missing taxonomy problems and could potentially train AI algorithms to learn automatic labeling in Second Life in the future. This game prototype can also be extended to other virtual worlds, where accurate metadata are needed. Widespead use of a product like Seek-n-Tag could greatly improve metadata in virtual worlds, and eventually deliver a world with information needed for visually impaired users to navigate and explore around.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This dissertation starts with a survey study that reports the current state-of-the-art of game accessibility. The definitions for each game genre and disability category are given to reflect the scope of this study. Through analysis of a number of player interactions with games of various genres, a general interaction model was defined. This interaction model consists of three distinct steps that players perform: (1) receiving stimuli; (2) determining response; and (3) providing input. The model allows for more precise identification of how a disability affects a player's abilities, specifically during each phase. In addition, detailed analysis of statistical data revealed that an estimated 11% of the U.S. population are unable to play games and 9% suffer from a reduced gaming experience because of a disability. While age is strongly correlated to the likelihood of disability and these estimates may include a large number of people who have no interest in playing games, the demographics of game players are shifting to gradually include an aging population.

This study extensively examined and presented 35 accessible games from 8 major genres, with respect to different disabilities. Many other related resources are mentioned in the dissertation as well. The surveyed games were picked based on genre, the various techniques applied, and the availability for the most severe forms of disability. Two major approaches are identified for making games accessible: dedication and adaptation. Their pros and cons are discussed in terms of development time and cost, effectiveness and efficiency, and effect on gameplay. The analysis of each game allows us to extract and collect accessibility strategies into two categories: high level and low level. Low-level strategies can be used to evaluate the accessibility of a game heuristically. High-level strategies may be used to point out solutions to accessibility problems that low-level strategies fail to address. Based on the findings via survey study, this dissertation continued by developing three distinct games to further advance the state of game accessibility for visually impaired individuals.

This body of work contributes three complete games along with their design and validation processes. The first two case studies aim at creating blind-accessible games using new strategies. Based on the findings from developing the first two games, a third game was created and used to improve potential accessibility of existing games.

- We developed Blind Hero [149] so blind people could enjoy a popular dance/rhythm game by making use of our customized haptic glove. This game introduces a high-level strategy: replacing visual feedback with haptic feedback.
- During our research on virtual worlds, we built a user interface from scratch, TextSL, to allow visually impaired users to access Second Life via a screen reader. Providing that access to an existing virtual world with only screen reader support is a new low-level tactic we present.
- A game is created for the purpose of providing critical information to improve TextSL or any future blind-accessible version of Second Life. This piece is a crucial component to this research, because it helps to prove games can be used to solve problems where a human component is needed, even for an otherwise monotonous task. This demonstrates that games can have purposes outside of entertainment and can have multiple facets to their experience. In this case: entertainment and improving accessibility.

All of the above created games focus on three major features that game provides: fun, education, and social interaction. User studies were conducted for all of these games. Feedback (which was largely positive) and suggestions were collected during this process. Working with, and observing, visually impaired individuals closely is a rewarding process that has so far yielded valuable insights into how to improve accessibility features to maximize their leisure, interest, and benefit.

6.2 Future Work

This dissertation points out the following areas for future research. Popular game genres, such as strategy, sports, and role playing games, lack accessible games for motor impaired and visually impaired players. The data on games for cognitive impairments are very limited. Those games are more difficult to expand to a generic level, because they are mostly targeting on very specific type of cognitive impairments. It will be valuable to collect more information to find out links between particular cognitive impairments and available game genres in the future. A common problem presented itself after the development of all three games: we do not have a unique set of standards to measure how effective our approaches are.

"You can't control what you can't measure" [26] is a well known statement in software engineering, which is also applicable to game accessibility. What are the metrics for measuring if a game is accessible? No set of accessibility guidelines for games, similar to the W3C Web Content Accessibility Guidelines, exists. Games are different from regular software as they require an additional constraint on what the interaction should provide: games should be fun. Measuring "fun" is not a simple endeavor since that is a subjective measure depending on player preference. Nevertheless, similar to usability evaluation, a heuristics-based approach [83] can be used, in which a number of experts identify which accessibility guidelines have been implemented.

The low-level strategies proposed in Chapter 2 could be used for this. However, we identified a number of problems: (1) The different low-level strategies accommodate impairments to different extents. For example, closed captioning is a better solution than subtitles to accommodate hearing impaired players. For one-switch games, a scanning mechanism that preserves all input is preferred over one that automates input or reduces it because such modification may significantly change gameplay. (2) The application of low-level strategies is context-sensitive. For instance, the game of chess may not have dialog or sound, and a strategy such as closed captions cannot be applied. However, this strategy can be applied to other types of games such as an FPS, but it does not imply that a chess game lacking closed captions is less accessible than an FPS with closed captions. (3) A game that supports multiple impairments is likely to be more accessible than a game that supports only one impairment. However, the level of accessibility depends on the degree to which a particular impairment is supported. A game that offers all types of interaction, ranging from regular input to one-switch control can be considered to be more accessible than a game that offers only subtitles and a high contrast color scheme.

Measuring the accessibility of a game is complicated because of these factors. In this context, our future work comprises two parallel paths: i) continuously develop accessible games for popular game genres; and ii) some sort of accessibility evaluation metrics should be developed to further validate our games and better serve visually impaired people.

Bibliography

[1] Peter Abrahams. Second life class action, http://www.it-analysis.com/blogs/abrahams_accessibility/2006/11/second_life_class_action.html. 2006.

[2] ACS. American community survey subject definitions, http://www.census.gov/hhes/www/ disability/acs.html, 2008. Last accessed August 2008.

[3] Nicoletta Adamo-Villani and Kelly Wright. Smile: an immersive learning game for deaf and hearing children. In *SIGGRAPH '07: ACM SIGGRAPH 2007 educators program*, page 17, New York, NY, USA, 2007. ACM.

[4] AFB. American foundation for the blind - facts and figures on americans with vision loss, http://www.afb.org/section.asp?sectionid=15&documentid =4398#legal. 2008. Last accessed September 2008.

[5] Motoyuki Akamatsu and I. Scott MacKenzie. Movement characteristics using a mouse with tactile and force feedback. *Int. J. Hum.-Comput. Stud.*, 45(4):483–493, 1996.

[6] Matthew T. Atkinson, Sabahattin Gucukoglu, Colin H. C. Machin, and Adrian E. Lawrence. Making the mainstream accessible: redefining the game. In *sandbox '06: Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames*, pages 21–28, New York, NY, USA, 2006. ACM.

[7] AudioGames. http://audiogames.net/listgames.php, 2008.

[8] Stephen Barrass and Gregory Kramer. Using sonification. *Multimedia Systems*, 7(1):23–31, January 1999.

[9] Jeffrey P. Bigham. Increasing web accessibility by automatically judging alternative text quality. In *IUI '07: Proceedings of the 12th international conference on Intelligent user interfaces*, pages 349–352, New York, NY, USA, 2007. ACM.

[10] Jeffrey P. Bigham, Anna C. Cavender, Jeremy T. Brudvik, Jacob O. Wobbrock, and Richard E. Lander. Webinsitu: a comparative analysis of blind and sighted browsing behavior. In *Assets '07: Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 51–58, New York, NY, USA, 2007. ACM.

[11] Jeffrey P. Bigham, Ryan S. Kaminsky, Richard E. Ladner, Oscar M. Danielsson, and Gordon L. Hempton. Webinsight:: making web images accessible. In *Assets '06: Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, pages 181–188, New York, NY, USA, 2006. ACM.

[12] Blizzard Entertainment. World of warcraft surpasses 11 million subscribers worldwide, http://www.blizzard.com/us/press/081028.html, 2008. Last accessed November 2008.

[13] Jonathan Blow. Game development: Harder than you think. *Queue*, 1(10):28–37, 2004.

[14] Boris Brandherm, Sebastian Ullrich, and Helmut Prendinger. Simulation of sensor-based tracking in second life. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 1689–1690, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[15] Helene Brashear, Valerie Henderson, Kwang-Hyun Park, Harley Hamilton, Seungyon Lee, and Thad Starner. American sign language recognition in game development for deaf children. In *Assets '06: Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, pages 79–86, New York, NY, USA, 2006. ACM.

[16] Stephen A. Brewster, Peter C. Wright, and Alistair D. N. Edwards. An evaluation of earcons for use in auditory human-computer interfaces. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 222–227, New York, NY, USA, 1993. ACM.

[17] Jennifer L. Burke, Matthew S. Prewett, Ashley A. Gray, Liuquin Yang, Frederick R. B. Stilson, Michael D. Coovert, Linda R. Elliot, and Elizabeth Redden. Comparing the effects of visual-auditory and visual-tactile feedback on user performance: a meta-analysis. In *ICMI '06: Proceedings of the 8th international conference on Multimodal interfaces*, pages 108–117, New York, NY, USA, 2006. ACM.

[18] William S. Carter and Guido D. Corona. Exploring methods of accessing virtual worlds, http://www.afb.org/afbpress/pub.asp ?docid=aw090207. Last Accessed October 2008.

[19] William S. Carter and Guido D. Corona. Accessibility for the blind in virtual worlds. 2008.

[20] Anna Cavender, Shari Trewin, and Vicki Hanson. General writing guidelines for technology and people with disabilities. *SIGACCESS Access. Comput.*, (92):17–22, 2008.

[21] Leandro Correia. Jet boarder, http://www.leandrocorreia.com/jetboardere.htm, 2008. Last accessed May 2008.

[22] Chris Crawford. *Chris Crawford on Game Design*. New Riders Games, June 2003.

[23] Pavel Curtis and David A. Nichols. MUDs grow up: Social virtual reality in the real world. In *COMPCON*, pages 193–200, 1994.

[24] Melissa Dawe. Desperately seeking simplicity: how young adults with cognitive disabilities and their families adopt assistive technologies. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1143–1152, New York, NY, USA, 2006. ACM.

[25] Deaf Gamers. Deaf gamers, game reviews for auditory impaired, http://www.deafgamers.com, 2000. Last accessed March 2009.

[26] T. DeMarco. *Controlling Software Projects: Management, Measurement, and Estimates*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1986.

[27] M.D. Dickey. Three-dimensional virtual worlds and distance learning: two case studies of active worlds as a medium for distance education. volume 36, pages 439–451.

[28] DISTAT. Disability statistics compendium, http://unstats.un.org/unsd/demographic/sconcerns/disability/. 1990. Last accessed December 2008.

[29] Sarah A. Douglas and Shasta Willson. Haptic comparison of size (relative magnitude) in blind and sighted people. In *Assets '07: Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 83–90, New York, NY, USA, 2007. ACM.

[30] Patrick Dwyer and Peter VanLund. The last crusade, http://www.cs.unc.edu/research/assist/ et/projects/rpg/index.html, 2008. Last Accessed August 2008.

[31] Entertainment Software Association. Essential facts about the computer and video game industry, http://www.theesa.com/facts/pdfs/esa_ef_2008.pdf, 2008. Last accessed March 2009.

[32] ESA. Entertainment software association, http://www.theesa.com/facts/gameplayer.asp, 2008. Last accessed September 2008.

[33] ESA. Essential facts about the computer and video game industry 2008, http://www.theesa.com/facts/pdfs/esa_ef_2008.pdf, 2008. Last accessed March 2009.

[34] M. Federoff. Heuristics and usability guidelines for the creation and evaluation of fun in video games. Master's thesis, Department of Telecommunications of Indiana University, 2002.

[35] J. Feng, J. Lazar, L. Kumin, and A. Ozok. Computer usage by young individuals with down syndrome: an exploratory study. In *Assets '08: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pages 35–42, New York, NY, USA, 2008. ACM.

[36] Hanan Finnerty. Branston and the lost machine, http://www.donationcoder.com/contests/agame/entries/branston/branston.swf, 2008. Last accessed April 2008.

[37] Eelke Folmer. Usability patterns in games. In *Future Play '06: Proceedings of the 2006 conference*, 2006.

[38] Eelke Folmer, David Carr, Manjari Sapre, and Bei Yuan. Making second life accessible to the blind. In *Technical Report: http://www.textsl.org/tr01.pdf*, 2008.

[39] Johnny Friberg and Dan Gärdenfors. Audio games: new perspectives on game audio. In *ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 148–154, New York, NY, USA, 2004. ACM.

[40] Johnny Friberg and Dan Gärdenfors. Audio games: new perspectives on game audio. In *ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 148–154, New York, NY, USA, 2004. ACM.

[41] Future Technology Devices International (FTDI). http://www.ftdichip.com/products/ft232r.htm, 2007. Last Accessed March 2009.

[42] James P. Gee. *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave Macmillan, May 2003.

[43] James Paul Gee. *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave Macmillan, 2004.

[44] Eitan Glinert and Lonce Wyse. Audiodyssey: an accessible video game for both sighted and non-sighted gamers. In *Future Play '07: Proceedings of the 2007 conference*, pages 251–252, New York, NY, USA, 2007. ACM.

[45] Eitan Glinert and Lonce Wyse. Audiodyssey: an accessible video game for both sighted and non-sighted gamers. In *Future Play '07: Proceedings of the 2007 conference on Future Play*, pages 251–252, New York, NY, USA, 2007. ACM.

[46] GMA. Shades of doom, http://www.gmagames.com/sod.html, 2008. Last accessed September 2008.

[47] GMA. Tank commander, http://www.gmagames.com/gtc1.shtml, 2008. Last accessed June 2008.

[48] Google. Google image labeler, http://images.google.com/imagelabeler/, 2009. Last accessed March 2009.

[49] Dimitris Grammenos and Yiannis Georgalis. Ua-chess: A universally accessible board game. In *3rd International Conference on Universal Access in Human Computer Interaction*. Lawrence Erlbaum, 2005.

[50] Dimitris Grammenos, Anthony Savidis, Yannis Georgalis, and Constantine Stephanidis. Access invaders: Developing a universally accessible action game. In Klaus Miesenberger, Joachim Klaus, Wolfgang L. Zagler, and Arthur I. Karshmer, editors, *ICCHP: International Conference on Computers Helping People with Special Needs*, volume 4061 of *Lecture Notes in Computer Science*, pages 388–395. Springer, 2006.

[51] M.D. Griffiths, M.N.O. Davies, and D. Chappell. Breaking the stereotype: The case of online gaming. *Cyber Psychology and Behavior*, 6(1):81–91, 2003.

[52] Susumu Harada, James A. Landay, Jonathan Malkin, Xiao Li, and Jeff A. Bilmes. The vocal joystick: evaluation of voice-based cursor control techniques. In *Assets '06: Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, pages 197–204, New York, NY, USA, 2006. ACM.

[53] Simon Harper and Sean Bechhofer. Sadie: Structural semantics for accessibility and device independence. *ACM Trans. Comput.-Hum. Interact.*, 14(2):10, 2007.

[54] Terri Hedgpeth, Jr. John A. Black, and Sethuraman Panchanathan. A demonstration of the icare portable reader. In *Assets '06: Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, pages 279–280, New York, NY, USA, 2006. ACM.

[55] Roger Hoang, Chris Franklin, and Eelke Folmer. Gordon's trigger finger, one switch first person shooter, http://gtf.eelke.com, 2008.

[56] IBM. Virtual worlds user interface for the blind, http://services.alphaworks.ibm.com/virtualworlds/, 2009. Last accessed March 2009.

[57] Takeo Igarashi and John F. Hughes. Voice as sound: using non-verbal voice input for interactive control. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 155–156, New York, NY, USA, 2001. ACM.

[58] IGDA-SIG. Igda game accessibility white paper. Technical report, 2004. Last accessed December 2008.

[59] Steve Ince. *Writing for Video Games.* A&C Black, 2006.

[60] Inspired Code. Music tetris, http://www.inspiredcode.net/metris.htm, 2008. Last accessed May 2008.

[61] Julie A. Jacko, Ingrid U. Scott, Francois Sainfort, Leon Barnard, Paula J. Edwards, V. Kathlene Emery, Thitima Kongnakorn, Kevin P. Moloney, and Brynley S. Zorich. Older adults and visual impairment: what do exposure times and accuracy tell us about performance gains associated with multimodal

feedback? In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 33–40, New York, NY, USA, 2003. ACM.

[62] Debbie Jeffers. Is there a second life in your future? In *SIGUCCS '08: Proceedings of the 36th annual ACM SIGUCCS conference on User services conference*, pages 187–190, New York, NY, USA, 2008. ACM.

[63] H. Jenkins. Game theory: How should we teach kids newtonian physics? simple. play computer games. available at http://www.technologyreview.com/articles/wo_jenkins032902.asp. *Technology Review*, 3, 2002.

[64] Shinya Kawanaka, Yevgen Borodin, Jeffrey P. Bigham, Darren Lunn, Hironobu Takagi, and Chieko Asakawa. Accessibility commons: a metadata infrastructure for web accessibility. In *Assets '08: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pages 153–160, New York, NY, USA, 2008. ACM.

[65] Jo Kay and Sean FitzGerald. Educational uses of second life, http://sleducation.wiki spaces.com/educationaluses. Last accessed May 2008.

[66] Mansureh Kebritchi and Atsusi "2c" Hirumi. Examining the pedagogical foundations of modern educational computer games. *Comput. Educ.*, 51(4):1729–1743, 2008.

[67] Reid Kimball. Doom3 cc, http://gamescc.rbkdesign.com/, 2008. Last accessed August 2008.

[68] Jim Kitchen. Mach 1 car racing, http://www.kitchensinc.net/, 2008. Last accessed August 2008.

[69] S. H. Kurniawan, A. Sporka, V. Nemec, and P. Slavik. Design and user evaluation of a spatial audio system for blind users. In *Proceedings of the 5th International Conference on Disability, Virtual Reality & Assoc. Tech*, pages 175–182, Oxford, UK, 2004.

[70] Boy Kwekkeboom and Ilse van Well. Ilbo, http://www.game-accessibility.com/index.php?pagefile=ilbo, 2002. Last accessed March 2009.

[71] KYEnterprises. Mouth-operated joysticks for quadriphlegics, http://www.quadcontrol. com, 2008. Last accessed September 2008.

[72] Dave Lebling and Marc Blank. Infocom games, zork I, http://infocom-if.org/games/zork1/zork1.html, 1980.

[73] Clayton Lewis. Hci and cognitive disabilities. *interactions*, 13(3):14–15, 2006.

[74] William Li. http://www.engineering.utoronto.ca/news/012420073.htm, 2008. Last Accessed May 2008.

[75] libsecondlife. libsecondlife, http://www.libsecondlife.org/wiki/ main_page. Last accessed December 2008.

[76] Maxis. The sims, http://thesims2.ea.com/, 2009. Last accessed March 2009.

[77] David R. Michael and Sandra L. Chen. *Serious Games: Games That Educate, Train, and Inform*. Muska & Lipman/Premier-Trade, 2005.

[78] MICROCHIP. http://www.microchip.com/wwwproducts/devices.aspx? ddocname=en010215, 2009. Last Accessed April 2009.

[79] Daniel Miller, Aaron Parecki, and Sarah A. Douglas. Finger dance: a sound game for blind people. In *Assets '07: Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 253–254, New York, NY, USA, 2007. ACM.

[80] Multilabs. http://www.multilabs.net, 2009. Last Accessed April 2009.

[81] Natural Point. Trackir 4: The premium optical head tracker, http://www.naturalpoint. com/trackir/, 1997. Last accessed March 2009.

[82] BBC News. Brain waves control videogames, http://news.bbc.co.uk/2/hi/technology/3485918.stm, 2004. Last accessed May 2008.

[83] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, New York, NY, USA, 1990. ACM.

[84] Stéphane Norte and Fernando G. Lobo. Sudoku access: a sudoku game for people with motor disabilities. In *Assets '08: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pages 161–168, New York, NY, USA, 2008. ACM.

[85] NPD Group. Playing video games viewed as family/group activity and stress reducer, http://www.npd.com/press/releases/press_071212.html, December 2007. Last accessed May 2008.

[86] Peter Ohring. Web-based multi-player games to encourage flexibility and social interaction in high-functioning children with autism spectrum disorder. In *IDC '08: Proceedings of the 7th international conference on Interaction design and children*, pages 171–172, New York, NY, USA, 2008. ACM.

[87] Michael A. Oren. Speed sonic across the span: building a platform audio game. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 2231–2236, New York, NY, USA, 2007. ACM.

[88] Roland Ossmann, Klaus Miesenberger, and Dominique Archambault. A computer game designed for all. In *ICCHP '08: Proceedings of the 11th international conference on Computers Helping People with Special Needs*, pages 585–592, Berlin, Heidelberg, 2008. Springer-Verlag.

[89] Jeremy Page. Tiny island nation opens the first real embassy in virtual world, http://technology.timesonline.co.uk/tol/news/tech_and_web/article1832158.ece, 2007. Last accessed June 2007.

[90] W.J. Perkins and B.F. Stenning. Control units for operation of computers by severely physically handicapped persons. *J. Med. Eng. Technol.*, 10(1):21–23, 1986.

[91] Helen Petrie, Fraser Hamilton, Neil King, and Pete Pavan. Remote usability evaluations with disabled people. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1133–1141, New York, NY, USA, 2006. ACM.

[92] Anne Marie Piper, Eileen O'Brien, Meredith Ringel Morris, and Terry Winograd. Sides: a cooperative tabletop computer game for social skills development. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 1–10, New York, NY, USA, 2006. ACM.

[93] PopCapGames. Popcap games. survey: "'disabled gamers"' comprise 20% of casual video games audience,http://www.popcap.com/press/release.php?pid=410, 2008. Last Accessed July 2008.

[94] Yolanda A. Rankin, Rachel Gold, and Bruce Gooch. Playing for keeps: gaming as a language learning tool. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Educators program*, page 44, New York, NY, USA, 2006. ACM.

[95] Yolanda A. Rankin, McKenzie McNeal, Marcus W. Shute, and Bruce Gooch. User centered game design: evaluating massive multiplayer online role playing games for second language acquisition. In *Sandbox '08: Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, pages 43–49, New York, NY, USA, 2008. ACM.

[96] Reuters. Us dollars spent in second life over last 24 hours. http://secondlife.reuters.com/stories/2008/01/31/data-shows-growing-divide-between-casual -and-heavy-sl-users/. Last accessed May 2008.

[97] Reuters. Ibm accelerates push into 3d virtual worlds, http://secondlife.reuters.com/stories/2006/11/09/ibm-accelerates-push-into-3d-virtual-worlds/, 2006. Last accessed December 2006.

[98] Ritualistic. The sims,http://www.ritualistic.com/games.php/sineps/faq/ index.inc.php, 2006. Last accessed March 2009.

[99] Tim Ritzema and Billy Harris. The use of second life for distance education. *J. Comput. Small Coll.*, 23(6):110–116, 2008.

[100] Sarah Smith Robbins. Immersion and engagement in a virtual classroom: Using second life for higher education. In *EDUCAUSE Learning Initiative Spring 2007 Focus Session*, 2007.

[101] Gabriel Robles-De-La-Torre. The importance of the sense of touch in virtual and real environments. *IEEE MultiMedia*, 13(3):24–30, 2006.

[102] Timothy Roden and Ian Parberry. Designing a narrative-based audio only 3d game engine. In *ACE '05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 274–277, New York, NY, USA, 2005. ACM.

[103] Louis Rosenberg and Scott Brave. Using force feedback to enhance human performance in graphical user interfaces. In *CHI '96: Conference companion on Human factors in computing systems*, pages 291–292, New York, NY, USA, 1996. ACM.

[104] Nicole Saidi. 'naughty auties' battle autism with virtual interaction, http://www.cnn.com/2008/health/conditions/03/28/sl.autism.irpt/index.html, 2008. Last accessed April 2008.

[105] Marc Saltzman. Review: 'lair' looks gorgeous, but game play a bungled mess. Technical report, CNN Tech review, 2007.

[106] Jaime Sánchez, Nelson Baloian, Tiago Hassler, and Ulrich Hoppe. Audiobattleship: blind learners collaboration through sound. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 798–799, New York, NY, USA, 2003. ACM.

[107] W. Scacchi, R. Nideffer, and J. Adams. A collaborative science learning game environment for informal science education: Dinoquest online. 279:71–82, 2008.

[108] Seth Schiesel. From the creator of bejeweled, another digital diversion from the days work, http://www.nytimes.com/2008/09/18/arts/television/18pegg.html, 2008. Last accessed Feb 2009.

[109] SecondLife. Economic statistics. http://secondlife.com/whatis/economy_stats.php. Last accessed Dec. 2008.

[110] SEGA. Virtua tennis 3, http://www.virtuatennis.net/, 2007. Last accessed March 2008.

[111] K. Sehaba, P. Estraillier, and D. Lambert. *Interactive educational games for autistic children with agent-based system.* Springer Berlin / Heidelberg, 2005.

[112] Karim Sehaba and Pascal Estraillier. Game execution control by analysis of player's behavior. In *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, page 19, New York, NY, USA, 2006. ACM.

[113] R.C. Simpson and H.H. Koester. Adaptive one-switch row-column scanning. *Rehabilitation Engineering, IEEE Transactions on*, 7(4):464–473, Dec 1999.

[114] SLBuzz.com. Sl stats http://stats.slbuzz.com/sims/browse/, 2009. Last accessed March 2009.

[115] J. David Smith and T. C. Nicholas Graham. Use of eye movements for video game control. In *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, page 20, New York, NY, USA, 2006. ACM.

[116] Adam J. Sporka, Sri H. Kurniawan, Murni Mahmud, and Pavel Slavík. Non-speech input and speech recognition for real-time control of computer games. In *Assets '06: Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, pages 213–220, New York, NY, USA, 2006. ACM.

[117] H. Spring. One switch frogger, http://havsoft.co.uk/one%20switch.htm, 2005. Last accessed May 2008.

[118] K. Squire. Changing the game: What happens when video games enter the classroom? *Innovate: Journal of Online Education*, 1(6), 2005.

[119] Generation Stars and Danny Boyd. One switch mini golf, http://www.oneswitch.org.uk/2/sd-sport.htm, 2005. Last accessed April 2008.

[120] Constantine E. Steriadis and Philip Constantinou. Designing human-computer interfaces for quadriplegic people. *ACM Trans. Comput.-Hum. Interact.*, 10(2):87–118, 2003.

[121] R.L. Street, W.R. Gold, and T Manning, editors. *Interactive video games for health promotion: Effects on knowledge, self-efficacy, social support, and health.* Lawrence Erlbaum Associates, 1997.

[122] Hironobu Takagi, Shinya Kawanaka, Masatomo Kobayashi, Takashi Itoh, and Chieko Asakawa. Social accessibility: achieving accessibility through collaborative metadata authoring. In *Assets '08: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pages 193–200, New York, NY, USA, 2008. ACM.

[123] Akela Talamasca. The heron sanctuary helps the disabled find a second life, http://www.massively.com/2008/ 01/29/the-heron-sanctuary-helps-the-disabled-find-a- second-life/. Last accessed April 2008.

[124] Jay Shiro Tashiro and David Dunlap. The impact of realism on learning engagement in educational games. In *Future Play '07: Proceedings of the 2007 conference on Future Play*, pages 113–120, New York, NY, USA, 2007. ACM.

[125] Don Teague. Give me that online religion, http://www.msnbc.msn.com/id/18789168, 2007. Last accessed May 2007.

[126] Media Lunde Tollefsen. Guidelines for developing accessible games. based on guidelines defined by medialt and igda., 2006.

[127] Shari Trewin, Vicki L. Hanson, Mark R. Laff, and Anna Cavender. Powerup: an accessible virtual world. In *Assets '08: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pages 177–184, New York, NY, USA, 2008. ACM.

[128] Ubisoft. Xiii, http://www.ubi.com/us/games/info.aspx?pid=39, 2003. Last accessed May 2008.

[129] Unreal. Unreal voodoo http://fretsonfire.sourceforge.net/, 2008.

[130] U.S. Census Bureau. Statistics on disability, http://www.census.gov/hhes/www/disability/disability.html, 2002. Last accessed March 2008.

[131] U.S.Government. 1998 amendment to section 508 of the rehabilitation act. *SEC. 508. Electronic and information Technology*, 1998.

[132] Valve. Half-life 2, http://orange.half-life2.com/hl2.html, 2008. Last accessed August 2008.

[133] Richard van Tol. Gaming with an auditory disability, http://www.game-accessibility.com/index.php?pagefile=auditory, 2006. Last accessed May 2008.

[134] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, New York, NY, USA, 2004. ACM.

[135] Luis von Ahn and Laura Dabbish. Designing games with a purpose. *Commun. ACM*, 51(8):58–67, 2008.

[136] Luis von Ahn, Shiry Ginosar, Mihir Kedia, Ruoran Liu, and Manuel Blum. Improving accessibility of the web with a computer game. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 79–82, New York, NY, USA, 2006. ACM.

[137] W3C. Web content accessibility guidelines 2.0, http://www.w3.org/tr/wcag20/, 2008. Last accessed March 2009.

[138] Eric Walker, Scott Stanfield, Bret Alfieri, and Christopher McGarry. Strange attractors 1, http://www.ominousdev.com/games.htm, 2008. Last accessed February 2009.

[139] T Westin. Game accessibility case study: Terraformers - a real-time 3d graphic game. In *In Proc. of the The Fifth International Conference on Disability, Virtual Reality and Associated Technologies*, 2004.

[140] Gareth R. White, Geraldine Fitzpatrick, and Graham McAllister. Toward accessible 3d virtual environments for the blind and visually impaired. In *DIMEA '08: Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts*, pages 134–141, New York, NY, USA, 2008. ACM.

[141] Wikipedia. http://en.wikipedia.org/wiki/guitar_hero.

[142] Wikipedia. Secondlife, http://en.wikipedia.org/wiki/second_life.

[143] Wikipedia. Video game genres, http://en.wikipedia.org/wiki/video_game_genres, 2008. Last accessed August 2008.

[144] Tyler Winters and Eelke Folmer. Torque cc - closed captioning component for the torque engine, http://www.garagegames.com/community/resources/view/13437, 2008. Last accessed March 2009.

[145] World Health Organization. Prevalence of impairments, disabilities, handicaps and quality of life in the general population: a review of recent literature. http://whqlibdoc.who.int/bulletin/2001/issue11/bul-11-2001/79(11)1047-1055.pdf. 2001. Last accessed December 2008.

[146] World Health Organization. Magnitude and causes of visual impairment, http://www.who.int/mediacentre/factsheets/fs282/en/, 2004. Last accessed March 2009.

[147] Nick Yee, Jeremy N. Bailenson, Mark Urbanek, Francis Chang, and Dan Merget. The unbearable likeness of being digital: The persistence of nonverbal social norms in online virtual environments. *CyberPsychology and Behavior*, 2006.

[148] Matthew Yi. They got game stacks of new releases for hungry video game enthusiasts mean it's boom time for an industry now even bigger than hollywood, http://www.sfgate.com/cgi-bin/article.cgi?f=/chronicle/archive/2004/12/18/mnguoae36i1.dtl, 2006.

[149] Bei Yuan and Eelke Folmer. Blind hero: enabling guitar hero for the visually impaired. In *Assets '08: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pages 169–176, New York, NY, USA, 2008. ACM.