UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

PLAYING BY PROXY

MESTRADO EM ENGENHARIA INFORMÁTICA

Especialização em Interação e Conhecimento

Salvador de Araújo Coder Meira

Dissertação orientada por:
Prof. Doutor Tiago João Vieira Guerreiro
Prof. Doutor André Filipe Pereira Rodrigues

2023

Acknowledgements

For the continuous support and guidance provided to me throughout the duration of the Playing by Proxy project, from its conception to the completion of this thesis, I would like to express my most sincere thanks to my advisors, Doctor Tiago Guerreiro and Doctor André Rodrigues. All the assistance, online meetings, and discussions have been invaluable.

For their contributions to the development of the Playing by Proxy platform, including their participation in numerous test sessions and provision of crucial feedback, significantly enriching the project, I would like to extend my heartfelt gratitude to David Gonçalves, Pedro Pais, and Ana Faustino.

For assisting me with the Parsec SDK get going and resolving many questions that emerged along the way, I would also like to acknowledge all the helpful users in the Parsec SDK Discord channel, in particular to users Skippy and Kodikuu. My most sincere thanks to them.

For integrating me in a very welcoming and professional environment, and providing the opportunity that was given to me to study and work in this project, I would like to extend my gratitude to the LASIGE team and to FCUL (Faculty of Sciences of the University of Lisbon), including all professors, staff, and fellow students who have all contributed to my academic journey.

For their availability, willingness to help and dispensed time for the final tests of the application, I would like to express my heartfelt thanks to all participants of the study. Their contribution and valuable feedback have been essential to uncover the limitations of our application and approach, as well as to show how various users deal with the application and rate its features, setting a solid ground for refining and improving the Playing by Proxy platform. I would like to write a special mention to my friend, Aliu Baio, for accepting the invitation to test the platform providing the insight that only a blind person could give. For his good will and availability, my heartfelt thanks to him.

Finally, for all the help that has been given in a variety of ways and constant support, I would like to acknowledge and send my most sincere thanks to my family and friends.

Resumo

Desde muito cedo que os jogos fazem parte da nossa sociedade, juntando pessoas em experiências sociais e interativas. Contudo, nem toda a gente reúne as capacidades motoras, cognitivas ou visuais exigidas pela grande maioria dos jogos. É certo que muitos jogos foram desenvolvidos tendo em conta diretrizes de acessibilidade, porém estes jogos constituem uma minoria naquele que é o universo de jogos em geral. Para jogadores com deficiências, os jogos vêm com barreiras adicionais que os impedem de desfrutar a experiência. Apesar dos recentes avanços no que toca a acessibilidade em jogos, muitas destas experiências continuam a não ser jogáveis para pessoas cegas.

Para além disso, os jogos digitais costumam estar divididos em duas categorias: jogos direcionados para pessoas com alguma deficiência específica e jogos direcionados para pessoas sem deficiências, ou seja, jogos comuns. Não restando espaço para a interseção entre jogadores com diferentes tipos de habilidades, pessoas com deficiências acabam por ser socialmente excluídas em meios onde se jogam jogos regulares, mas inacessíveis. A principal motivação desta tese surge da vontade de integrar pessoas com deficiências visuais em prismas mais comuns, fazendo com que pessoas cegas e não cegas possam experimentar os mesmos tipos de jogos e tentando mitigar esta exclusão existente no meio digital.

Sejam jogos físicos ou digitais, o desenho da experiência é tipicamente focado no jogador, mas os jogos são muitas vezes atividades partilhadas que podem incluir desde um a milhares de espetadores. Existem ainda vários exemplos onde o papel de jogador e o papel de espetador intersetam numa experiência conjunta. Questionámo-nos sobre a possibilidade de espetadores normovisuais envolverem-se na jogabilidade de um jogador cego, auxiliando e jogando colaborativamente. E, neste sentido, explorámos um modo de jogar alternativo, feito de forma assistida e que visa ultrapassar esta barreira de acessibilidade visual.

Nesta dissertação foi explorado um paradigma de jogo diferente do habitual: *Playing by Proxy*, isto é, experiências de jogo partilhado em que um jogador auxiliar normovisual facilita a interação do jogador cego com o jogo. O objetivo desta tese é estudar a possibilidade de pessoas cegas jogarem jogos originalmente inacessíveis. Deste modo, a ideia passa por acrescentar uma camada de acessibilidade através de jogadores normovisuais que têm diferentes ferramentas de ajuda ao seu dispor, desde enviar pistas de áudio ao jogador cego até partilhar os diferentes controlos de jogo.

Estendemos uma aplicação que conecta estes pares de jogadores, ao mesmo tempo que desenvolvemos um conjunto de funcionalidades que permitem um jogador normovisual auxiliar remotamente um jogador cego no progresso de um videojogo, jogando *por proxy*. Ao estabelecer-se a ligação, o jogador ajudante normovisual consegue ver o ecrã do jogador cego (*streaming*) e um menu com as funcionalidades disponíveis em *overlay*.

As funcionalidades desenvolvidas são selecionadas do lado do jogador normovisual e incluem:

- **Ping Points:** jogador normovisual envia as coordenadas de uma posição do ecrã ao jogador cego, através de um click com o seu rato na posição pretendida. O jogador cego ouvirá "*beeps*" mais frequentes o quão mais perto a posição do seu cursor estiver da posição enviada pelo jogador normovisual. Quando o jogador cego atingir uma determinada distância desta posição, os *"beeps"* param. Caso pretenda, o jogador cego pode mover instantaneamente o seu cursor para a posição recebida usando a tecla *enter* ou cancelar os *"beeps"* usando a tecla *escape*.
- **Snap Mouse:** o jogador normovisual clica numa posição do ecrã, fazendo o cursor do jogador cego mover-se instantaneamente para a posição escolhida. É ouvido um som quando o cursor é movido através deste modo.
- **Steer Mouse:** usando as setas do teclado, o jogador normovisual pode ajustar a posição do cursor do jogador cego.
- **Vocal Sounds:** sons simples e maioritariamente monossilábicos que podem ser enviados a qualquer momento para ajudar o jogador cego na sua situação de jogo (por exemplo, "yes", "no", "jump", entre outros).
- **Control Sharing:** através deste modo, o jogador normovisual pode controlar diretamente o *input* do rato, teclado ou gamepad do jogador cego. É possível controlar mais do que um tipo de input simultaneamente.
- **Press Key:** este modo serve para o jogador normovisual enviar uma indicação de uma tecla que deverá ser premida. Os quatro tipos diferentes de indicação disponíveis para premir uma tecla são: carregar, manter carregada, carregar duplamente e carregar repetidamente. Após escolhido o tipo de indicação, o jogador normovisual carrega na tecla pretendida. Esta indicação chegará ao jogador cego como uma breve mensagem sonora. Por exemplo: "Press A".
- **Voiced Message:** o jogador normovisual envia uma mensagem escrita que será vocalizada do lado do jogador cego através de *Text-to-Speech*. Esta mensagem pode ser escrita e lida em português ou em inglês.
- **Labelled Ping Points:** funciona do mesmo modo que os Ping Points regulares, mas através desta funcionalidade o jogador normovisual tem a oportunidade de escrever a mensagem que o jogador cego ouvirá ao chegar à posição enviada, de forma a que o jogador cego tenha uma breve descrição do ponto a que chegou com o seu cursor.

Com o objetivo de testar a aplicação final, realizámos um estudo com sete pares de jogadores, no qual em cada par uma pessoa vendada assumiu o papel de jogador cego e a outra pessoa o papel de jogador ajudante normovisual. No final, fizemos o mesmo estudo, mas desta vez com uma pessoa efetivamente cega a assumir o papel de jogador cego. Neste estudo, fez-se com que o jogador cego (ou vendado) jogasse, auxiliado pelo jogador normovisual, cinco jogos de complexidades e tipos diferentes: CS2D, Antenna, Rogue Glitch, Tadpole Tales e Super Raft Boat. Cada um dos jogos foi jogado em duas fases, entre 5 a 10 minutos cada, que diferiam no modo de comunicação entre os dois jogadores, sendo elas: 1) usando apenas as funcionalidades da aplicação para comunicar; 2) podendo comunicar livremente.

A aplicação e as suas funcionalidades foram avaliadas pelos utilizadores através de um questionário preenchido após o término do estudo. O feedback do estudo foi recolhido através deste questionário, de notas retiradas ao longo das sessões e de uma entrevista final com um utilizador cego. Através deste feedback, limitações relacionadas com as funcionalidades desenvolvidas e relacionadas com a nossa

abordagem foram descobertas e descritas. Tendo em conta os resultados do estudo, discutimos sobre as funcionalidades desenvolvidas (separando em assistência de rato, partilha de controlos e assistência por voz), sobre o papel do jogador cego enquanto jogador participativo, sobre as diferenças entre as experiências dos jogadores vendados com a experiência do jogador cego e, por fim, oportunidades para melhorar esta plataforma. Os resultados do estudo e subsequente discussão levaram-nos às seguintes conclusões: 1) esta forma de jogar torna algumas situações de jogo anteriormente inacessíveis em situações de jogo acessíveis quando jogados de forma partilhada com o uso desta ferramenta; 2) algumas funcionalidades mostraram serem úteis em determinadas circunstâncias, enquanto outras não foram práticas em qualquer situação de jogo ou existem melhores alternativas; 3) jogos bastante inacessíveis ou com um ritmo de jogo bastante rápido continuam a ser inacessíveis, mesmo com o uso desta plataforma. No final, descrevemos direções possíveis para as quais este projeto pode rumar no futuro. Este projeto procurou explorar os conceitos de jogabilidade partilhada e espetador participativo como forma de possibilitar que jogadores cegos experienciem jogos que não foram desenvolvidos tendo em consideração jogadores com deficiências visuais. Esperamos que este trabalho possa representar um passo importante para reagir à normatividade de habilidade visual existente nos videojogos, fazendo com que jogadores cegos e jogadores não cegos tenham possibilidade de jogar o mesmo tipo de jogos e combatendo a exclusão social existente neste meio digital.

Palavras-chave: Jogos, Acessibilidade, Deficiência Visual, Inclusão Social, Jogabilidade Assistida

Abstract

For players with disabilities, playing games comes with additional barriers that prevent them from enjoying the game. Despite recent advances in game accessibility, many of these experiences are still unplayable for blind people or targeted only towards them, leaving no room for intersection between blind and sighted players. We have extended an application that connects pairs of players, providing a set of features that allow a sighted player to remotely assist a blind player progress in video games. When the connection is established, this proxy player will be able to see the blind player's gaming screen and an overlay menu with a set of digital features.

We carried out a study with seven pairs, each including a player assuming the role of blind player, using a blindfold, and the other assuming the role of a proxy player. We made the blindfolded player play five games of different complexity and type aided by the proxy player. We did the same study at the end with a blind person. The final product was evaluated by users through a questionnaire and additional feedback was collected through notes taken during the sessions and an interview with the blind user. The results of the study led us to conclude that: 1) this shared play approach makes some previously inaccessible gaming situations accessible when playing collaboratively with the use of this platform; 2) certain features proves to be useful in specific circumstances, while others were not practical in gaming situations or had better alternatives; 3) highly inaccessible games or with a very fast pace remain inaccessible even with the use of this platform. This project aimed to explore the concepts of shared play and participative spectatorship as means to enable blind players to experience games that were not originally designed to include players with visual impairments.

Keywords: Gaming, Accessibility, Visual Impairments, Social Inclusion, Assisted Play

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Globally, it is estimated that around 285 million people are visually impaired, of whom 39 million are blind [1]. Great strides have been made towards enabling blind people to live independent, successful lives. Still, serious difficulties are faced when accessible physical and digital platforms are not provided. Most games are not developed to include total (or, sometimes, partial) blindness, so, when that is the case, people who are visually impaired end up being deprived of one of the most prevalent sources of entertainment today. Games have been part of our society for a long time, bringing people together in social and interactive experiences. However, not everyone has the motor, cognitive, or visual abilities required by most games.

With advances in technology, new ideas emerge in the accessibility field, such as the *Be My Eyes*[1] application, which allows blind users to connect with sighted users in order to get visual help doing everyday tasks. It is possible to follow analogous guidelines to shape the gaming world into becoming more accessible and inclusive. In this manner, *Playing by Proxy* emerges as an attempt to create a channel of understanding and a possible way of interaction between a blind player and a game where there was close to none.

1.1. Motivation

Games are experiences that bring people together, making them compete or cooperate to reach a mutual goal. The majority of games assumes that the player has already a predetermined set of skills, such as visual or hearing abilities that they will use to continuously react to the events of the game. As these abilities are usually taken for granted when developing games, an interaction barrier emerges for people with disabilities, discouraging them to reach or gain interest in trying new games at all.

Frequently, digital games are either targeted towards people without disabilities or targeted towards people with a certain disability. In one way or the other, people with disabilities will be socially excluded in communities that play regular but inaccessible games. The main motivation of this thesis comes from the will to integrate people with visual impairments in more common prisms, experiencing the same fun games their sighted peers do.

Whether physical or digital games, the experience's design is typically focused on the player, but games are often shared activities that can include from one to thousands of spectators. It is with this idea in mind that we explored an alternative form of play, made in an assistive way and that aims to overcome this accessibility barrier.

We explored an unusual game paradigm in this thesis: Playing by Proxy, i.e. shared game experiences in which the blind player relies on a sighted helper that will facilitate their interaction with

[1] Be my Eyes. `www.bemyeyes.com` (Last visited November 27th, 2022)

the game. The main goal of the thesis is to study the possibility of blind people playing games that are originally inaccessible. The focus of the study is to add an accessibility layer through sighted helpers which will have a set of digital features at their disposal to aid the blind player progress in the game, including audio cues, control sharing, and other helping modes.

We hope that Playing by Proxy can represent an important step towards reacting to the visual ability normativity existing in the video game world, allowing both blind and sighted players to enjoy the same types of games and addressing the social exclusion that exists in this digital medium.

1.2. Approach

We introduce an unusual game paradigm in this thesis: *Playing by Proxy*. This is the name we gave to a technique that allows two players to collaboratively play the same game – one being the original player, while the other is playing "*by proxy*" – turning an individual interaction with a game into a shared gaming experience. We explored how a sighted Proxy Player could assist the play of a Blind Player by using this technique.

For this end, we prototyped a desktop streaming application, making use of the Parsec SDK[2], the Window's API[3], and other C libraries. The goal of this application was to have a stream-like platform that allows the Proxy Player to watch Blind Player's screen at real time, containing an overlayed menu which includes a set of features that aim to aid the Blind Player overcome various gaming challenges. The different functionalities were implemented iteratively and were frequently tested with the team. The design and implementation of the application and all of its features is described in the upcoming chapters (Chapter 3 for design and Chapter 4 for implementation).

The provided features are selected in the sighted player's side and include:

- Ping Points: sighted player clicks the screen in a point of interest of the blind player's screen, sending the coordinates of this point to the blind player. The blind player will hear "beeps" that will be more frequent the closer their cursor is to the sent position.
- Snap Mouse: when the sighted player clicks a position on the screen, the blind player's cursor will transpose instantly to the chosen position.
- Steer Mouse: adjusts the position of the blind player's cursor, using the keyboard arrows.
- Vocal Sounds: simple sounds that may be sent at any time to aid the blind player in a game situation (e.g., "yes", "no", "jump", and more).
- Control Sharing: using this mode, the sighted player may directly control the blind player's mouse input, keyboard input or gamepad input.
- Press Key: the sighted player informs the blind player which keyboard key should be pressed, hold, double pressed or pressed repeatedly.
- Voiced Message: sends a message to the blind player that will be heard through Text-to-Speech.

[2] Parsec SDK. `https://parsecgaming.com/docs/sdk/` (Last visited November 30th, 2022)

[3] Windows API. `https://learn.microsoft.com/en-us/windows/win32/apiindex/windows-api-list` (Last visited November 30th, 2022)

- Labelled Ping Points: similar to Ping Points, but in this mode the sighted player will write the message that the blind player will hear when they reach the position.

When the application was ready to be tested first with sighted users, we made a study that included 14 participants to take the roles of Blind Players and Proxy Players in order to test the developed application. After experimenting the application with the seven sighted pairs, we conducted a session with one blind person. The goal of the study was to analyse whether the participants could progress in five games, with different levels of complexity, and which functionalities they choose to surpass the different adversities. The games we chose were relatively simple 2-dimensional games available on Steam[4]: CS2D[5], Antenna[6], Rogue Glitch[7], Tadpole Tales[8], and Super Raft Boat[9]. The application and its features were therefore evaluated by all the users and all feedback was collected through an online questionnaire, notes taken during the session, and an interview with the blind user after the final study session. We analysed all the data and highlighted important findings. We then discussed about various topics, which included: 1) the importance and usability of the features and how could they be improved; 2) the role of the Blind Player as a participative player; 3) the divergence between the experiences of blindfolded participants and the blind participant; 4) how could Playing by Proxy could be improved and some of its limitations. After discussing the aforementioned points, we proceeded to draw conclusions about the study and the general project, including noting future directions the Playing by Proxy project could take.

1.3. Goals

Our major goal is to introduce a new form of play that aims to create a bridge between Blind Players and games that are not accessible to them. With this goal, comes the need to evaluate whether gaming assistance via a Proxy Player with collaborative play is a possible solution to address this existing accessibility gap in video games. For this end, we developed a platform that allows Proxy Players to provide on-demand assistance to Blind Players through various features, as a means to make different visual challenges possible. We made a study to validate this approach and to understand the usability of the features developed in different gaming situations.

A more distant and future end-goal of the project is to have a virtual platform that connects sighted Proxy Players to Blind Players, making all the useful features available for the Proxy Players to assist Blind Players, ideally making it possible that games that used to be disregarded due to inaccessibility become playable and enjoyable shared experiences.

---

[4] Steam. `https://store.steampowered.com/` (Last visited on November 30th, 2022)

[5] CS2D. `https://store.steampowered.com/app/666220/CS2D/` (Last visited on November 30th, 2022)

[6] Antenna. `https://store.steampowered.com/app/443580/Antenna/` (Last visited on November 30th, 2022)

[7] Rogue Glitch. `https://store.steampowered.com/app/1092630/Rogue_Glitch/` (Last visited on November 30th, 2022)

[8] Tadpole Tales. `https://store.steampowered.com/app/1428900/Tadpole_Tales/` (Last visited on November 30th, 2022)

[9] Super Raft Boat. `https://store.steampowered.com/app/1541250/Super_Raft_Boat_Classic/` (Last visited on November 30th, 2022)

1.4. Contributions

Our main contributions with this dissertation are:

- A conceptual approach focused on creating collaborative gaming experiences designed for players with visual impairments and proxy-aiding players.
- A proof-of-concept instance of the approach. We designed and developed a prototype desktop application targeted towards pairs of mixed visual abilities.
- An exploration of the approach with 7 pairs of sighted players with one player of each pair playing blindfolded, plus one pair with a blind participant. The study results demonstrate that some accessibility barriers can potentially be overtaken with the assistance of a proxy helper. Some of the features developed for the application appear to be useful tools to address accessibility gaps in video games.

1.5. Document's Structure

This document is organised as follows:

- **Chapter 2 – Related Work.** We explore examples where accessibility is achieved via the social environment, in addition or alternatively to physical aidances. We also provide an overview of relevant approaches, issues and clever design techniques that aim to make games accessible.
- **Chapter 3 – Design.** All about the design behind Playing by Proxy. We start by explaining the design motivations and use case scenarios that motivated the project development. Then, we get into detail about the design of the application's features, how they were envisioned and what issues they aim to address.
- **Chapter 4 – Implementation.** We explain the implementation steps that led to the development of Playing by Proxy. We start by providing an overview about how the application functions and the tools we used during the implementation process. Then, we get into more detail about the different functionalities, how they work and the different built-in mechanics.
- **Chapter 5 – User Evaluation.** We conducted a study with 7 pairs composed of sighted and blindfolded players, plus a pair composed of a blind player with a member of the researching team. Each member of a pair had a role between Blind Player and Proxy Player, where the person portraying the role of Blind Player plays the game with the direct assistance of the Proxy Player. In this manner, each pair played collaboratively 5 games with different levels of complexity. After playing, each person responded to a questionnaire asking about their experience, background, which games were most enjoyable or difficult and which features were more relevant. Finally, we present and discuss the result achieved.
- **Chapter 6 – Conclusion.** Final thoughts and prospects for forthcoming work.

Chapter 2

Related Work

This chapter will cover ideas and research that set a starting point to this project. We have grouped the related work into five main topics: Human-powered Accessibility, Asymmetry, Control Sharing, Spectating Experience, and Earcons. Each relating to a significant degree and being relevant to the domain of this thesis. In the Human-powered Accessibility section, we will go through relevant examples in which accessibility was achieved by the means of other people and social awareness. In the Asymmetry section, we will discuss how asymmetric roles can lead to inclusion in video games. The Control Sharing section will cover some platforms that allow the sharing of gaming controls and then we will examine how this dynamic could work towards video game inclusion and accessibility. In the Spectating Experience section, we will mention various examples in which the player of a game is extended to a few or a multitude of players that collaborate to play a game and discuss how this concept can be related to video game accessibility. Finally, in the Earcons section we will describe what earcons are, how they can be implemented, and the benefits they could provide to digital media, such as video games. At the end of the Related Work Chapter, we will discuss the current state of video game accessibility and go through emerging ideas that aim to shape the gaming world into becoming more accessible and inclusive.

2.1. Human-powered Accessibility

Human-powered accessibility refers to making physical or digital environments accessible through the assistance of other people. We can find examples of this type of accessibility in mainstream media, such as sign language interpreters on public television[1]. *Video Relay Service*[2] is an example of human-powered accessibility, as it allows people with hearing disabilities communicate via the phone with hearing people through the assistance of intermediaries who use sign language to help the message get through between the two parties.

Another relevant example that significantly inspired this project is *Be my Eyes*[3]. *Be my Eyes* is a mobile application that allows a blind user to make phone calls to sighted volunteers for help on vision-demanding tasks, such as reading a label on a pills' bottle, confirming the expiration date of a product, or identifying cards in a wallet. Through this application, blind people can rely on the assistance of sighted people to perform actions that would otherwise be much more difficult. With *Playing by Proxy*, we attempted to create a similar dynamic in the gaming world: blind gamers connect with proxy assistants, that can either be friends or online players willing to collaborate, to help overcome existing barriers in video games.

---

[1] RTP Sign Language. `https://www.rtp.pt/wportal/acessibilidades/gestual/` (Last visited November 27th, 2022)

[2] Video Relay Service. `https://www.fcc.gov/vrs` (Last visited November 27th, 2022)

[3] Be my Eyes. `www.bemyeyes.com` (Last visited November 27th, 2022)

Studies have shown us the importance of co-creating accessibility in everyday settings, such as home [2] and work [3]. Braham and Kane believe that an accessible environment comes not only by its physical aidances and features, but also by the social awareness and collaborative help that people provide, co-constructing accessibility. This concept aligns with the approach taken in *Playing by Proxy*, which is fuelled by collaborative play and the involvement of sighted people in making ordinary games accessible for blind players.

2.2. Asymmetry

Usually, digital games are either targeted towards people without disabilities or targeted towards people with a certain disability. *Audio games*[4], for example, is a type of game that is targeted towards blind gamers. However, previous work has explored different design approaches in catering for mixed-ability play. Gerling and Buttrick presented *Last Tank Rolling* [4], a game on which players using wheelchairs apply their assistive device to control a tank, while able-bodied players control a soldier. The authors explore the value of shared video gameplay as a means of empowering people with disabilities and connecting players of all abilities to foster inclusion.

Gonçalves et al. highlighted the potential of asymmetric gameplay for groups with different abilities and preferences [5]. They found out that people with visual impairments play diverse games, but face limitations when playing with others who have different visual abilities. The authors designed and developed two collaboratively tested games that explored asymmetric and interdependence roles [6]. According to the authors, the results showed that experiences that were unequal by design can lead to an equitable joint experience.

In this project, the aim is not to develop new games with built-in asymmetric roles, but to develop a tool that enables blind and sighted people to play together already existing regular games. Depping and Mandryk found that cooperation and interdependence between players are two factors that can be used to improve the play experience and increase social bounds [7]. By using this tool, both players will collaborate to play together. Playing this way, interdependence will emerge naturally, as they will rely on each other to progress in the game, and asymmetric roles will be established. For instance, a sighted Proxy Player might take the role of a guide, while the blind gamer uses their problem-solving skills or strategic thinking. Or, in a scenario involving the sharing of controls, one can imagine a shooter game in which the Proxy Player is in charge of guiding the way and pointing the gun, while the Blind Player is in charge of the characters' movement and shooting. As a result, the gameplay becomes a cooperative and inclusive experience between two mixed-ability players in which both participate and contribute to the success and enjoyment of the experience.

[4] Audio games. `https://www.audiogames.net/` (Last visited November 30th, 2022)

2.3. Control Sharing

Control sharing is one of the central focuses of this study. Being able to divide the controls and linking each player's abilities to appropriate ways to interact with the game is something that we could not overlook when attempting mixed-ability gameplay within the same game. This "co-piloting" concept is not exactly new, as *Parsec*[5], *Xbox Copilot*[6], *Google Stadia*[7] (although the latter recently shut down) already provided shared controls within their services. Also, as far as in 2006, Loparev, Lasecki, Murray, and Bigham came up with *WeGame* [8], a system that allows the joint player control of a single in-game character, with the main goal being the improvement of the social aspect of existing single-player games.

Control sharing has already been tried by researchers on providing accessibility to a game where there was none before. Since virtual reality (VR) games are usually designed for a standing and mobile player, not everyone has the mobility requirements to play them. Thiel and Steed [9] explored the concept of co-piloting in VR, allowing a second player to remotely help the VR player by sharing the controls of the game. By doing this, the VR player's reach could be extended and their interaction with the game became easier. In *Playing by Proxy*, we explore control sharing in a similar way: with a sighted player complementing a blind player's gameplay by controlling parts of the game that are inaccessible, such as positioning the cursor on top of an enemy in a shooter game, for example.

It has also been proposed the use of artificial intelligence (AI) shared-controls in order to automatize some parts of the gameplay to make it more accessible [10]. Cimolino, Askari, and Graham explored the results of having AI agents controlling the movement of the in-game character, thus splitting the interaction with the game between players with motor impairments and the computer. Although the players could play the two proposed games by using this approach, one of the findings mentioned that this could make the players feel more disabled instead of empowered, as they are being required to play the games differently from others. This approach is tangent to one of our project topics, the sharing of controls in order to make games accessible, so it is essential to understand how our participants feel when playing games while having part of the interaction shared with, in our case, another person.

2.4. Spectating Experience

Spectating Experience in a gaming context refers to the act of watching others play instead of playing a game, although there are cases in which the spectators are welcomed to integrate the playing experience, such as in the "ask the audience" dynamic in *"Who Wants to Be a Millionaire?"*[8] or in the

[5] Parsec. `https://parsecgaming.com/` (Last visited November 30th, 2022)

[6] Xbox Copilot. `https://support.xbox.com/en-US/help/account-profile/accessibility/copilot` (Last visited November 30th, 2022)

[7] Google Stadia. `https://stadia.google.com/gg/` (Last visited Mars 10th, 2023)

[8] Ask the Audience, Who Wants to Be a Millionaire Wiki. `https://millionaire.fandom.com/wiki/Ask_the_Audience` (Last visited November 30th, 2022)

Netflix titles *"Minecraft: Story Mode"*[9] and *"Black Mirror: Bandersnatch"*[10], in which the spectators interactively participate in deciding the outcome of the show by choosing the way they want the story to go on certain point breaks. Recently, Gonçalves et al. [11] presented a systematic review of 263 publications on multiplayer gaming, with 18 works identified as exploring the spectator/audience role, yet none explored its potential for increasing the accessibility of games for visually impaired people.

As game streaming platforms such as *Twitch*[11] or *YouTube Gaming*[12] are growing in audiences, studies are being made to better understand the engagement of live streams [12] and the experiences of video game audiences [13], [14]. The original purpose of a video game's experience is the player's enjoyment. However, we will explore occasions where the enjoyment of someone who is watching the game equalises or even surpasses the original purpose. Seering et al. [15] investigated this area in which the line between the audience and the player is blurred, disclosing designing opportunities and challenges that are faced when creating engaging audience participatory games. Tekin and Reeves [16] also explored the act of spectating in video games and found that passive observation can become an interactive and participatory activity which can include actions such as guiding and assisting, critique player techniques, recognize and compliment players when they do well, and reflect on past gameplay to provide instructions to the player.

We can see that spectator experience can go as far as having the spectator acting as a player themselves and creative ideas of including the spectator as an active part of the game are emerging in a variety of ways. Maurer et al. innovated in this field by creating a gaze-based onlooker integration concept [17], where the observer's gaze acts as a "virtual flashlight" that highlights or uncovers important parts of the screen for the first player to see. They explored an in-between grey area on spectating experience, in which someone is neither an active player nor a passive spectator but something in between. They found out that integrating the observer's gaze in the game could positively influence the experience for both the player and the spectator, making it more immersive and engaging.

This participative role of spectators in video games is not necessarily limited to a single spectator. *What Lurks in the Dark* is a *Twitch*-based asymmetric horror game that allows participation by audience members while streamers play, in which the audience can decide if they want to help or hurt the streamer and plan their actions accordingly [18]. Also, Lessel et al. investigated how a participative audience on gaming live-streams can influence the game that is being played [19], having as cases of study *Rocket Beans TV*[13] and *Twitch plays Pokémon*[14]. *Rocket Beans TV* was a German *Twitch* channel broad-casting 24/7 that, in 2014, launched a pen & paper role-playing game format in which the audience was encouraged to participate through various means, such as voting via the *Twitch* chat in one of many options that would decide the unfold of the story or send, via *Twitter* posts, illustrations and descriptive texts of items the players found within the story, therefore having direct influence on the setting and the story. Also in 2014, *Twitch Plays Pokémon* streamed the *Game Boy's* game "Pokémon Red", having as

[9] Minecraft: Story Mode. https://www.imdb.com/title/tt10498322/(Last visited November 30th, 2022)

[10] Black Mirror: Bandersnatch. https://www.imdb.com/title/tt9495224/ (Last visited November 30th, 2022)

[11] Twitch. https://www.twitch.tv/ (Last visited November 30th, 2022)

[12] YouTube gaming. https://www.youtube.com/gaming (Last visited November 30th, 2022)

[13] Rocket Beans TV. https://www.rocketbeans.tv (Last visited November 30th, 2022)

[14] Twitch Plays Pokémon, Reddit post "The History of Twitch Plays Pokémon", 2014. https://www.reddit.com/r/twitchplayspokemon/wiki/historyoftpp_gen1/ (Last visited November 30th, 2022)

the player an entire audience of 1.1 million players that entered 122 commands[15], such as “down”, “up”, “left”, “right”, “A”, “B”, or “Start”, all affecting the same in-game character, and finishing the game in 16 days. This social experiment included different ways of participative audience engagement, from players working towards completing the game (the majority) to players hindering the progress (known as “trolls”) to even players worshipping fan-made deities. Two different game modes were implemented: “anarchy”, where every command by every user was carried out, and “democracy”, meaning only commands entered by the most players within a timeframe were carried out. Studies have also been made revealing the potential of interactive storytelling in educational contexts [20].

These examples illustrate how in *Playing by Proxy* the Proxy Player could embrace the role of a participative spectator regarding the Blind Player’s gameplay, having the option to take part in the game by playing with them (as in control sharing) or even guiding the Blind Player so they better understand their in-game situation.

2.5. Earcons

An earcon is a brief, distinctive sound that represents a specific event or conveys information. Similarly to semaphores auditorily informing us that it is safe to cross the road, earcons are used in the digital world to auditorily inform us about events that are happening on the device we are using. In video games, earcons are used in a variety of ways, such as informing that a character has picked up an item or that something new has appeared on the screen. Guidelines in the creation of earcons were previously drawn to make them easier to be created and organised [21]. McGookin and Brewster [22] also investigated how earcons could be more easily identified. They found that modifying the earcons so that each was presented with a unique timbre and altering their presentation so that there was a short time delay between each earcon were found to significantly increase the earcon’s identification.

Nair et al. presented NavStick [23], a game controller that allows visually impaired people to “look around” virtual environments and 3D games using audio cues. The researchers found that visually impaired people could form more accurate mental maps of a navigation tasks’ environment when compared to the classic menu-based surveying. In the context of a representative 3D game, the results showed that, using NavStick, the participants could play the presented study games, as it adds additional auditory information to the one already existing in the games.

Earcons seem to be a powerful tool to have integrated in a platform for visually impaired users as they are effective at conveying information to users. Also, they are preferred over other forms of audio feedback, such as speech or music, because they are simple and easily discerned from one another. In *Playing by Proxy*, we use earcons to inform the Blind Player about the activation and deactivation of different modes on the Proxy Player’s side, as well as when some features are being used.

[15] Twitch Plays Pokémon Final Stats. `http://www.engadget.com/2014/03/01/twitch-plays-pokemon-final-stats-1-1-million-players-36-millio.html` (Last visited November 30th, 2022)

2.6. Current State

Studies have shown us the importance of social playing, especially during the pandemic that struck the world in 2020 [24], [25]. Playing with others can lead to positive social interactions when the design is created to cater for positive, safe, and friendly interactions [25]. Although, not everyone can enjoy the benefits of social playing. When a game is inaccessible, an information gap emerges between the game and the player [26]. To raise awareness of this gap, Dimitris Grammenos presented "*Game Over!*", a game which is universally inaccessible (meaning it can be played by no one) and is meant to be used as an educational tool for disseminating and teaching game accessibility guidelines [27]. Guidelines are being drawn for both board [28] and digital games [29] so that designers and developers can create new games – or adapt already existing ones – so that they can be played by people with all sets of abilities. Andrade et al. [30] highlighted the need to balance complexity and accessibility in both games targeted towards players with visual impairments and mainstream games, and accessed opinions regarding concerns around new emerging technologies such as virtual reality and augmented reality.

The intersection between mainstream games and games that can be played by visually impaired people is still very thin. Although, over the last few years, accessibility has become an increasingly important part of games. Features like control remapping, colour-blind options, adjusting subtitles, and setting the difficulty are beginning to become more present in big trademark games[16][17][18][19]. The Last of Us Part 2[20], a Naughty Dog's 2020 title, may be one of the most accessible contemporary games with over 60 different accessibility options [31], ranging from lock-on aiming to enemy indicators, to motion sickness settings, to a system that allows a player to play without sight, including a library of audio cues, the ability to point the character towards their goal, ledge assistance that stops the character from falling, and an option that provides the player a read-out of the current relevant information (health points, which weapon is equipped, number of ammunition, etc.). However, a big budget is not mandatory to provide accessibility. In fact, there are *indie games* that do a great job in this regard[21][22][23][24][25]. Alongside new games, two new consoles were recently released: the *PlayStation 5*[26] and

[16] Marvel's Spider-Man: Miles Morales. https://www.playstation.com/pt-pt/games/marvels-spider-man-miles-morales/ (Last visited November 30th, 2022)

[17] Watch Dogs Legion. https://www.ubisoft.com/en-gb/game/watch-dogs/legion (Last visited November 30th, 2022)

[18] Assassin's Creed: Valhalla. https://www.ubisoft.com/en-gb/game/assassins-creed/valhalla (Last visited November 30th, 2022)

[19] Immortals Fenyx Rising. https://www.ubisoft.com/en-gb/game/immortals-fenyx-rising (Last visited November 30th, 2022)

[20] The Last of Us Part II. https://www.playstation.com/pt-pt/games/the-last-of-us-part-ii/ (Last visited November 30th, 2022)

[21] Lair of the Clockwork God. https://store.steampowered.com/app/1060600/Lair_of_the_Clockwork_God/ (Last visited November 30th, 2022)

[22] Inertial Drift. https://store.steampowered.com/app/1184480/Inertial_Drift/ (Last visited November 30th, 2022)

[23] Ikenfell. https://store.steampowered.com/app/854940/Ikenfell/ (Last visited November 30th, 2022)

[24] Wildfire. https://store.steampowered.com/app/431940/Wildfire/ (Last visited November 30th, 2022)

[25] HyperDot. https://store.steampowered.com/app/876500/HyperDot/ (Last visited November 30th, 2022)

[26] PlayStation 5. https://www.playstation.com/en-id/ps5/ (Last visited November 30th, 2022)

the *Xbox Series X*[27]. Both devices were made with accessibility taken into account, featuring useful options such as high contrast mode, closed captions, controller remapping, and starting up with a screen reader turned on, allowing those with visual disabilities to interact with the console from the get-go. In the *Xbox's Co-pilot* system, two controllers can be used for one input, being a meaningful feature for those who require someone else's assistance to get through games.

Studies have also been made to explore the experiences of streamers with visual impairments, as Jun et al. [32] identified the main challenges for blind streamers revolving around the themes of the absence of a community for streamers with visual impairments and accessibility issues regarding the streaming platform itself while learning how to live stream, use tools, and interacting with other people. Still, there are people with visual impairments who do not want to miss the gaming and streaming world and its opportunities. Anderson and Johnson's study [33] revealed four main themes around how streamers with visual impairments establish their identities regarding disability and gaming: establishing gaming skills, acknowledging disability, gaming to overcome challenges, and feeling empowered to inspire others.

Recently, Gonçalves et al. [34] investigated strategies and techniques that blind content-creator gamers use in order to play visual-centric digital games. They found that players often rely on persistent trial-and-error and ingenious strategies that leverage the game's own mechanics, such as purposefully bumping into walls or using the in-game character's sword against objects in order to understand the surroundings. But in other times, their strategy involves playing with other people, either in multiplayer settings, sharing control of the game through co-pilot features, or engaging with spectators. In *Playing by Proxy*, we will try to harmonise these topics – human-powered accessibility through playing with others, asymmetric roles employed by mixed-ability players, sharing controls with a sighted player, participative spectators – into a single platform so that, through shared play, blind people can be included in the gaming world.

[27] Xbox Series X. `https://www.xbox.com/en-US/consoles/xbox-series-x` (Last visited November 30th, 2022)

Chapter 3

Design

We are presenting Playing by Proxy: a desktop streaming application that creates an aiding channel between a Proxy Player and a Blind Player. In this chapter we will go through all the design process, the motivations, and the use case scenarios that inspired the creation of this application. At the end of the chapter, we will go into each developed feature of the application in detail.

3.1. Design Motivations

The research done initially disclosed sensibilities to have in mind when developing an application for visually impaired users. Since the beginning, the envisioned end-product was a stream-like application in which one or various Proxy Players could, at all times, see the gaming screen of a Blind Player. Being the end goal an application that provides assistance to multiple and diverse types of games, we began by thinking about different game mechanics and how they could be addressed by the features in the application.

3.2. Use Case Scenarios

Here we will present two fictional personas that set as the main examples of users that we had in mind when designing this desktop application: Pedro and Ana. Pedro is a blind person who enjoys games and has a friend who wants to play with him. Ana is a blind streamer who wants to play video games, looking for help from the viewers. Both study cases served as inspiration for the development of the platform.

3.2.1 Blind Player with a friend

Pedro always enjoyed playing accessible tabletop games with his friends and audio games when alone. Pedro has a lot of fun playing these games, but he wishes he could try some video games his friends play and talk about that are not accessible to visually impaired people. João, one of Pedro's friends, told him about Playing by Proxy, and how they could use this application to play together regular inaccessible games. Pedro decides to try this application with João and they choose a game that Pedro has heard about and got interested in the past. They connect via Discord[1] to be in continuous vocal contact with each other and then connect via Playing by Proxy. Now, João is on his computer watching Pedro's computer screen and has the application's tools at his hand to help Pedro. João uses *Control Sharing: Mouse Control* to open and start the game. They are playing a Zombies game and,

[1] Discord. `https://discord.com/` (Last visited November 28th, 2022)

using headphones, Pedro can recognize when a zombie is approaching, hearing its steps. They decide that João will be in charge of aiming the gun at the zombies and Pedro will do the shooting. Using *Mouse Control* and *Vocal Sounds* indications, they proceed to find a good synergy and collaborate to go through the game's challenges. João enjoys helping Pedro. Pedro feels accomplished to have finally played a game he always was keen to try, but couldn't otherwise.

3.2.2 Blind Player Streamer

Ana is a Twitch[2] streamer that usually streams about audio games she plays and audiobooks she listens to. She sometimes tries playing games which include accessibility options, but not many common games are made to be fully accessible to blind people. One day, she hears one of her viewers mention Playing by Proxy and decides to give it a try. Ana can now choose a game that was not accessible to hear before. After choosing the game, Ana opens a Playing by Proxy's host connection and tells her viewers they can connect to her client. Some of the viewers watching the stream connect via Playing by Proxy's client side to Ana's computer. These viewers will help Ana navigate through the game's menu and play the game. They use *Snap Mouse* and *Press Key* indications so Ana can open and start the game. Ana is playing a *platformer* game, so the viewers send *Vocal Sounds* indications such as "*Left!", "Right!", "Jump!",* or *"Great!"* and *Press Key* indications when the game asks for a specific key to be pressed at a time. When necessary, her viewers send *Voiced Messages* to inform about a situation with increased complexity. Using Playing by Proxy with her viewers, Ana can now play many more games, engaging with her viewers in collaborative and fulfilling experiences.

3.3. Conception

A fundamental aspect of the envisioned prototype was the requirement of a connection between two computers: the Proxy Player's computer and the Blind Player's computer. We know about existing applications that allow one user to stream what their screen is showing to others online, such as Discord, Twitch and Parsec[3]. So, we investigated further these applications' Software Development Kits (SDK) and Application Programming Interfaces (API) to know what we could use them for, the range and the limitations.

Because Parsec has a robust SDK[4], has almost no input latency, running at 60 frames per second, and allows remotely shared controls of one computer between users, this application was chosen to be the basis of the prototype that was to be developed. We then delve into Parsec SDK's documentation to better understand how it works, its methods and how we could try to implement our ideas on top of what was already built in.

A Parsec connection is divided by two parts: the Host side and the Client side. A Host sided machine starts running the application and waits for guests to join in. The guests are on the Client side of the

[2] Twitch. `https://www.twitch.tv/` (Last visited November 30th, 2022)
[3] Parsec. `https://parsecgaming.com/` (Last visited November 30th, 2022)
[4] Parsec SDK. `https://parsecgaming.com/docs/sdk/` (Last visited November 30th, 2022)

connection and will connect to the Host's machine. So, translating to this project's terms, the Blind Player will act as a Parsec Host and the Proxy Player as a Parsec Client.

Through the SDK's GitHub[5] repository, it was possible to access all the required files that were needed to get started. The folders include a Host example and a Client example that were the ones necessary to build the first, raw connection. By working on these files, using Parsec SDK's methods, as well as Simple DirectMedia Layer (SDL)[6], OpenGL[7], Windows API[8], and other C libraries, it was possible to forge the functionalities that were discussed to be Proxy Player's tools to assist the Blind Player in real time.

3.4. Navigation

The application's menu is always seen in the Proxy Player's side overlaying on top of the Blind Player's streaming screen, as shown in Figure 3.1. The user will use the F keys to navigate through the menus of the application. Activating and deactivating certain modes or features, will play different sounds that will help the Blind Player recognize that something has changed.

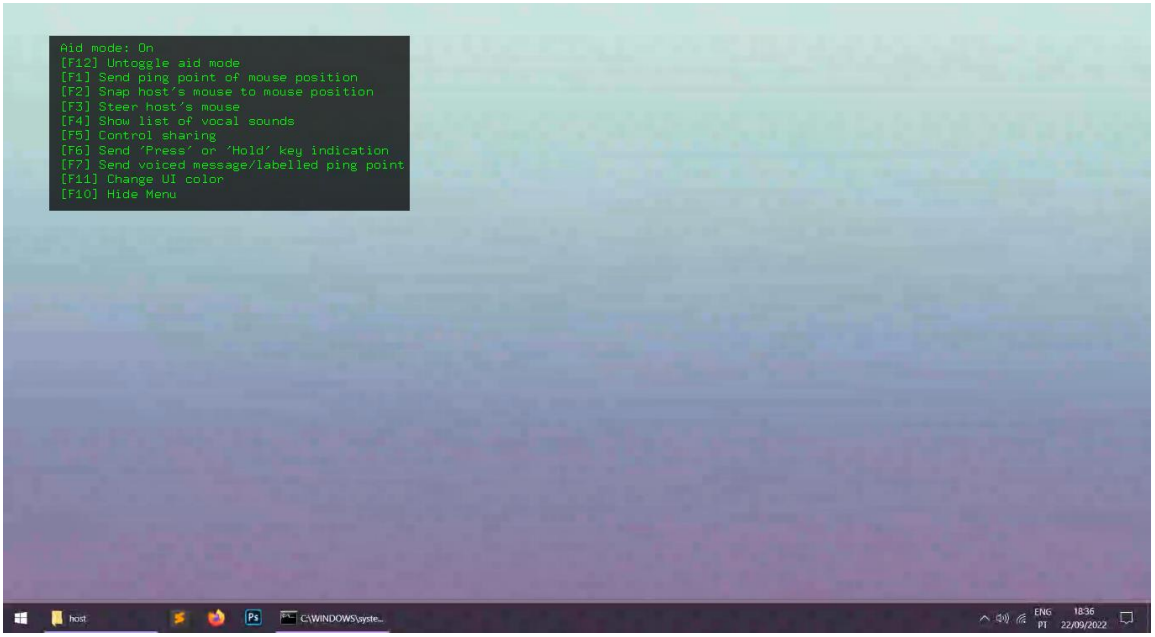

Figure 3.1: Menu overlayed in screen. The desktop of the image we see is from the Blind Player's computer. The menu on the top left corner, however, is only seen by the Proxy Player on their computer, overlaying on top of the Blind Player's stream.

When the application starts running, the aid mode is turned off. The user will have the options to activate aid mode, by pressing F12; change UI colour, by pressing F11; or to hide the application's menu, by pressing F10.

---

[5] Parsec SDK GitHub. `https://github.com/parsec-cloud/parsec-sdk` (Last visited November 30th, 2022)
[6] Simple DirectMedia Layer. `https://www.libsdl.org/` (Last visited November 30th, 2022)
[7] OpenGL. `https://www.opengl.org/` (Last visited November 30th, 2022)
[8] Windows API. `https://learn.microsoft.com/en-us/windows/win32/apiindex/windows-api-list` (Last visited November 30th, 2022)

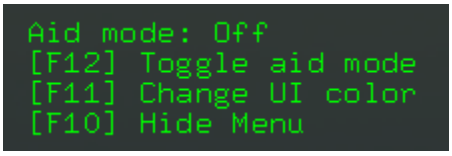

Figure 3.2: Aid mode off.

Activating aid mode will play a sound that will be heard by both Proxy and Blind players. Deactivating will play a different sound indicating that the aid mode is again turned off. While aid mode is turned on, a list with all the features and the necessary key to use them will be shown.

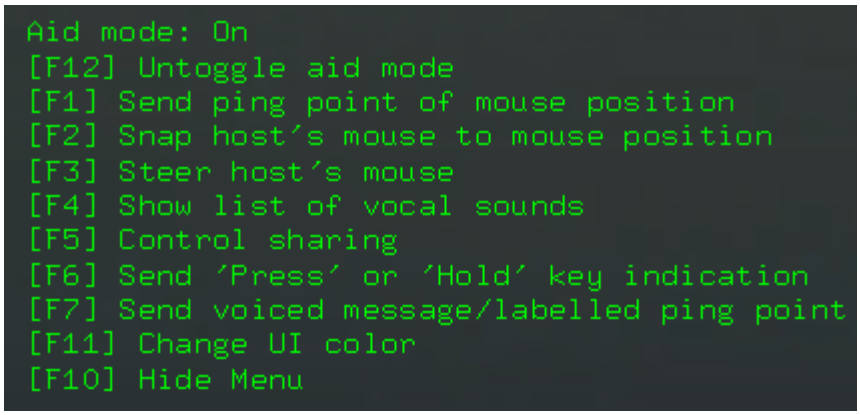

Figure 3.3: Aid mode on.

The user has the option to change the UI colour to light blue, white or yellow. This was made to increase the contrast between the menu and the background colours in certain situations to make it easier to the Proxy Player's eyes. The user can also change the menu's colour for aesthetics preferences.

The menu's window, despite not being very large and its background colour being semi-transparent (alpha value is 0.75), can in certain situations cover important details on the screen. Because of this reason, the user can always hide the application's menu, so it does not occupy much space in the streaming screen. The menu and features are still available and working while the menu is hidden, so in case the user has memorised the functionalities they will need, they can have the menu hidden and not be disturbed by its default size. The user may anytime press F10 again to make the menu visible again.

3.5. Features

The application's features provide the Proxy Player agency to assist the Blind Player in overcoming various existing obstacles in games. The set of features include audio cues, mouse motion aid, control sharing, among others.

All features are available on the Proxy Player's side and are received in different ways on the Blind Player's side.

3.5.1 Ping Points

Ping Points are used for the Blind Player to interactively find with their mouse cursor a position on the screen chosen by the Poxy Player. Usually, blind folks refrain from using mouse peripherals because the feedback of the cursor's movement is only visual. Using Ping Points, the goal is to give an auditive feedback so they can find a point of interest that was highlighted by the Proxy Player.

It works as follows:

- Proxy Player clicks on a screen position of their choice.
- Blind Player starts hearing pings that will be more frequent and with a higher pitch the closer their cursor is to the position chosen by the Proxy Player. Consequently, the pings will be less frequent and with a lower pitch the further away their cursor is to the chosen position.
- After finding the position, a completion sound is heard, and the pings stop.

The Blind Player does not have to find the exact pixel point chosen and sent by the Proxy Player. If the cursor is within a 75px ratio of the sent position, their cursor will instantly transpose to the exact position and the completion sound will be heard.

The Blind Player may cancel the pings anytime by pressing *escape* or go directly to the chosen position by pressing *enter*.

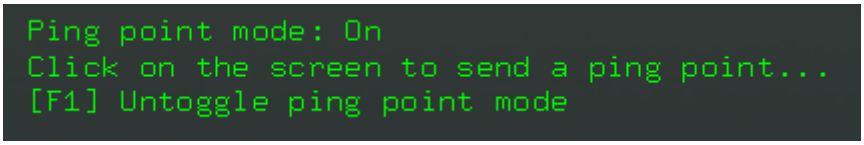

Figure 3.4: Ping point mode as seen on the Proxy Player's side.

3.5.2 Snap Mouse

The Snap Mouse mode is used to make the Blind Player instantly find a position on the screen, instead of having to search for it.

While Snap Mouse mode is active, the Proxy Player may choose to instantly snap the Blind Player's cursor to their chosen position. The Blind Player will hear a sound each time this happens, alerting that their cursor position has changed.

It works as follows:

- The Proxy Player clicks on a screen position of their choice.
- The Blind Player's cursor will be instantly transposed to the chosen position.

The Proxy Player may use Snap Mouse repeatedly to continuously change the cursor position of the Blind Player.

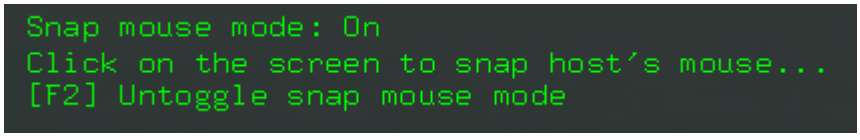

Figure 3.5: Snap mouse mode as seen on the Proxy Player's side.

3.5.3 Steer Mouse

Steer Mouse mode allows the Proxy Player to adjust the Blind Player's cursor position using the keyboard's arrow keys. This may be used when the Blind Player's cursor is near the position it should be on, and only needs an adjustment.

It works as follows:

- The Proxy Player presses a cardinal arrow direction on their keyboard, i.e. *up arrow*, *down arrow*, *left arrow*, or *right arrow*.
- The Blind Player's cursor moves accordingly.

Two cardinally adjacent arrow keys may be pressed simultaneously, making the cursor move diagonally.

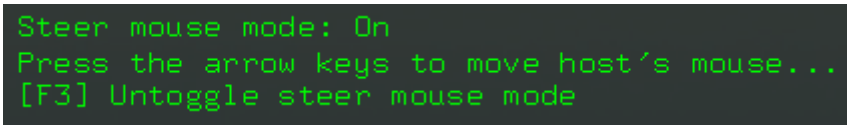

Figure 3.6: Steer mouse mode as seen on the Proxy Player's side.

3.5.4 Vocal Sounds

Vocal Sounds are sounds that are always available for the Proxy Player to send to the Blind Player. They include generic indications, confirmation sounds, and also cardinal directions.

Unlike other features, there is not a Vocal Sound mode that needs to be active in order for the Vocal Sounds to be sent. Because of this, the Proxy Player may choose to use Vocal Sounds alongside other features, complementing their assistance.

It works as follows:

- The Proxy Player chooses which Vocal Sound to send and presses the respective key (e.g., presses the key G to send the Vocal Sound "Great!").
- The Blind Player hears the respective Audio Sound.

Most Vocal Sounds were achieved using Amazon Polly[9] text-to-speech feature. "Jazzy Sound!", "Correct Sound!", and "Wrong Sound!" were available to use on an open audio library[10]. The full list of the Vocal Sounds can be seen on Figure 3.7.

[9] Amazon Polly. `https://aws.amazon.com/polly/` (Last visited November 30th, 2022)
[10] Freesound. `https://freesound.org/` (Last visited November 30th, 2022)

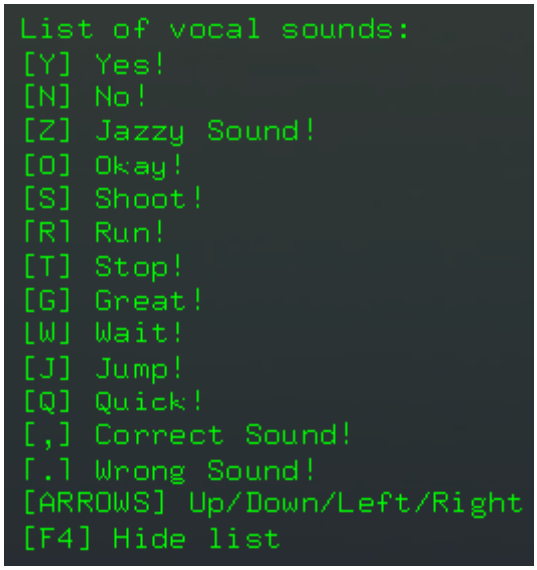

Figure 3.7: Vocal Sounds' list as seen on the Proxy Player's side.

3.5.5 Control Sharing

Control Sharing mode allows the Proxy Player to remotely control the Blind Player's mouse, keyboard, or gamepad in real time. This was made to allow collaborative play, in which, for example, one player controls the mouse and the other controls the keyboard. As an example, in a shooting game situation this can be translated to one of the players running and the other player aiming and shooting. Parsec natively supports the sharing of controls in its platform, so we integrated this functionality on our expanded application.

It works as follows:

- The Proxy Player chooses which input to control: *M* for Mouse input; *K* for Keyboard input; and *G* for Gamepad input.
- The Blind Player hears a sound informing the respective input is being controlled by the Proxy Player.
- The Proxy Player controls the chosen input.

More than one type of input can be controlled at the same time. At any time, the Blind Player may cancel the Control Sharing using the key *escape*. While controlling an input, the Proxy Player will have an indication stating it on the main menu.

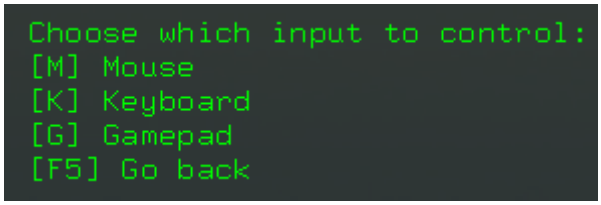

Figure 3.8: Control sharing mode as seen on the Proxy Player's side.

3.5.6 Press Key

Press Key mode is used for the Proxy Player to inform the Blind Player of a keyboard key or mouse button that they should press, hold, double press, or press repeatedly. This can be useful in game situations where a player must press or hold a key indicated by the game. Other situations may arise where Press Key mode comes to be useful, such as informing mouse buttons to be clicked during menu navigation or to directly indicate which key the Blind Player must press to overcome a determined game situation.

It works as follows:

- The Proxy Player chooses the mode the key should be pressed, between Press, Hold, Double Press, and Press Repeatedly. It is on "Press Key" by default, and it can be changed using the *alt* key.
- The Proxy Player presses the intended keyboard key or mouse button (including the mouse wheel) that should be pressed, held, double pressed, or pressed repeatedly.
- The Blind Player hears the respective indication, e.g., "Press Space" or "Double Press Left Click".

The vocal indication is achieved through eSpeak[11], a Text-to-Speech console program. Because they all use exclusive keyboard input, Vocal Sounds, Control Sharing: Keyboard Control and Press Key mode cannot be used simultaneously.

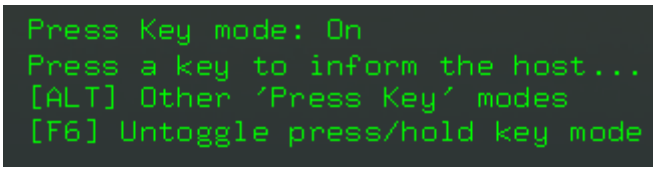

Figure 3.9: Press key mode as seen on the Proxy Player's side.

3.5.7 Voiced Message

In case the Proxy Player wants to write a word or a sentence to be read directly and heard by the Blind Player, they can use a Voiced Message.

It works as follows:

- The Proxy Player writes a word or a sentence that they want to be read out loud to the Blind Player.
- The Proxy Player presses *enter*, and the message will be read to the Blind Player, through Text-to-Speech program eSpeak.

The written input may be in English or Portuguese. The user may change the Text-to-Speech language by using the keyboard's *left arrow* or *right arrow*.

[11] eSpeak. `https://espeak.sourceforge.net/` (Last visited November 30th, 2022)

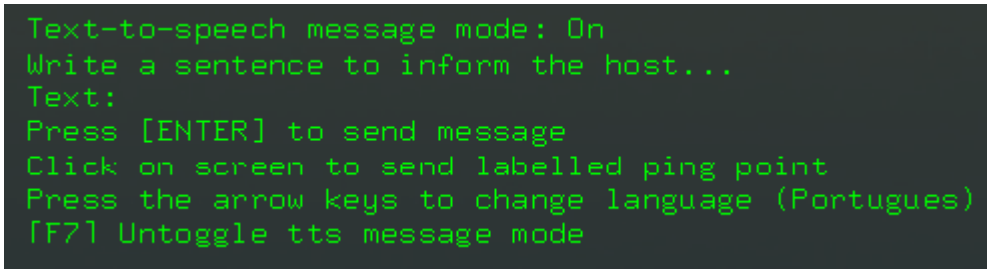

Figure 3.10: Text-to-speech mode as seen on the Proxy Player's side.

3.5.8 Labelled Ping Points

Labelled Ping Points are similar to regular Ping Points in all ways but one. The only difference being that the conclusion sound is instead a message written by the Proxy Player that will be heard when the Blind Player finds the chosen position. The goal is to indicate the location of a point of interest in the screen and at the same time label it in order to inform the Blind Player what lays in that position that they found, e.g., a door, an enemy, or even a menu option for starting a new game.

It works as follows:

- The Proxy Player writes the message that should be heard when the Blind Player finds the position.
- The Proxy Player clicks on the position that they want to inform the Blind Player about.
- The Blind Player will hear pings that will be more frequent and high pitch the closer their cursor is to the sent position and less frequent and lower pitch the further their cursor is to the sent position.
- Upon finding the position, the Blind Player will hear the message sent by the Proxy Player, informing about the object they just found.

The menu for the Labelled Ping Points is the same as the Voiced Message's menu, as they are both inside a Text-to-Speech message mode. After writing the message, if the Proxy Player presses *enter*, they will send a Voiced Message. Contrarily, if they instead use the mouse to click on a position in the screen, they will send a Labelled Ping Point to that point.

Just like regular Ping Points, Labelled Ping Points can be cancelled on the Blind Player's side using *escape* or the position can be instantly found by pressing *enter*.

Chapter 4

Implementation

In this chapter we will discuss the implementation process of Playing by Proxy. We will start by giving an overview of how the application functions. Secondly, we will go through all the tools used to make the application possible and how they were used. After this, we will talk about the application's interface on the Client side of the platform. At last, we will delve in more technical terms into each feature of the application.

4.1. Overview

There are two main sides that sustain the operation of Playing by Proxy: the Client side and the Host side. The Blind Player will be situated at the Host side and the Proxy Player at the Client side of the application. When starting the application, both users only have to run the respective bat files, submit the login credentials, and get started. Here we will delve into what happens behind the scenes so the connection between the users is made. Figure 4.1 shows the flow diagram of the relations between both players and the application itself.

To get started in Playing by Proxy, both users, i.e. the Proxy Player and the Blind Player, must acquire a Parsec Gaming account, given that the application is running using Parsec's Software Development Kit (SDK)[1]. Both logins are made through a python file that asks for the email and password as inputs and provides a Session ID. The Host login will also provide a Peer ID, this will be the code for the Client to connect to this Host.

The Blind Player will take the role of the Host in the Parsec connection. To get started, they will launch host.exe file with the only argument being their Session ID. After this task is succeeded, the Blind Player will hear a Text-to-Speech voice saying "Welcome to Playing by Proxy". After this, the Host will wait for Client guests. Or, in other words, the Blind Player will wait for the Proxy Player to connect.

The Proxy Player takes the role of the Client in the Parsec connection. After the login, the program will print all the available connections. The available connections refer to the hosting machines of this user's Parsec friends. Here, the Proxy Player chooses the Bind Player's machine to connect, and the connection is successfully done. Now, the Proxy Player will hear the same Text-to-Speech message: "Welcome to Playing by Proxy".

After the connection is made, the Host and the Client will be in constant contact with each other. The first communication they will make is informing each other about their screen's resolutions, so the features that deal with screen coordinates function with precision.

[1] Parsec SDK. `https://parsecgaming.com/docs/sdk/` (Last visited November 30th, 2022)

After this, each functionality activated in the Client side will inform the Host side about its activation. The Host side will then do the different operations according to the current functionality being used and the information sent. The technical building of each functionality and how they function can be read at point 4.4 of this chapter, Features.

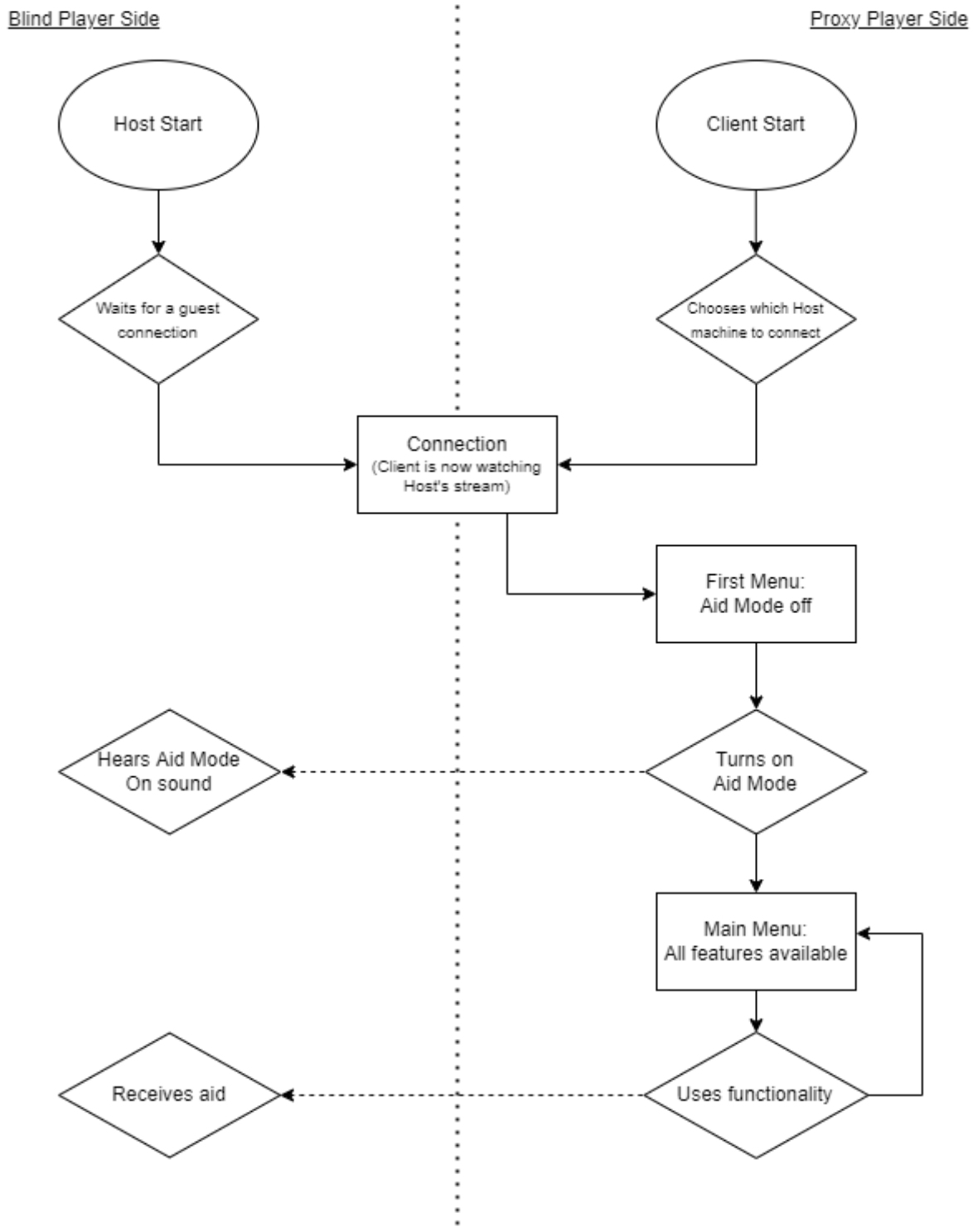

Figure 4.1: Flow Diagram.

4.2. Tools

The program of Playing by Proxy is mainly built in C language, using Parsec Gaming's Software Development Kit, but many were the libraries, API's and general tools used to create this desktop streaming application.

The Parsec SDK was the starting point of development of this application. Through compiling and running the Host and Client example files, the first raw connection was reached. The communication between the Host and the Client is made through Parsec Events. Therefore, the Client running cycle will register the Proxy Player's key inputs and send a message to the Host. This message always navigates attached with an ID, so the Host program will recognize its origin and finality and act accordingly.

The Windows API[2] was mainly used when dealing with the screens' resolutions and cursor methods, mainly *GetCursorPos()*[3] and *SetCursorPos()*[4], to get the current position of the cursor and to set the cursor to a different position, respectively. These methods were used to deal with cursor related features, namely Ping Points, Snap Mouse, and Steer Mouse. Apart from those, the methods *Beep()*[5], *Sleep()*[6], and *PlaySound()*[7] were also brought from Windows API. How these last three methods work and how they were used in our context will be explained further on subchapter 4.4 Features.

Simple DirectMedia Layer[8], or SDL, is the core media library used by Parsec. SDL covers the creation of the application window and deals with the video and audio rendering, as well as the handling of input events.

The graphism built on top of the SDL stream was made with OpenGL[9]. OpenGL is a graphics' library that provides tools and shapes to be drawn on the frames of the computer screen. The menu of the application was entirely drawn using OpenGL. The letters that composed the sentences in the menu were drawn using an OpenGL component called OpenGL Utility (or Glu)[10]. Glu was used to write characters from a text bitmap. Methods *writeLetter()* and *writeSentence()* were made to facilitate the writing of the words in the menu.

---

[2] Windows API. `https://learn.microsoft.com/en-us/windows/win32/apiindex/windows-api-list` (Last visited November 30th, 2022)

[3] GetCursorPos function. `https://learn.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-getcursorpos` (Last visited November 30th, 2022)

[4] SetCursorPos function. `https://learn.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-setcursorpos` (Last visited November 30th, 2022)

[5] Beep function. `https://learn.microsoft.com/en-us/windows/win32/api/utilapiset/nf-utilapiset-beep` (Last visited November 30th, 2022)

[6] Sleep function. `https://learn.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-sleep` (Last visited November 30th, 2022)

[7] PlaySound function. `https://learn.microsoft.com/en-us/previous-versions/dd743680(v=vs.85)` (Last visited November 30th, 2022)

[8] Simple DirectMedia Layer. `https://www.libsdl.org/` (Last visited November 30th, 2022)

[9] OpenGL. `https://www.opengl.org/` (Last visited November 30th, 2022)

[10] OpenGL Utility. `https://www.ibm.com/docs/en/aix/7.1?topic=manual-opengl-utility-glu-library` (Last visited November 30th, 2022)

Two text-to-speech programs were used in the building of this application. All the speech that was written beforehand, such as the Vocal Sounds, was achieved with Amazon Polly[11], a service that provides clean and lifelike speech. The Press Key mode, Voiced Messages and Labelled Ping Points used eSpeak[12] as the text-to-speech method. eSpeak is a speech synthesiser that functions with console commands, making it an appropriate and easy-to-use choice when dealing with text that will be generated during the use of the application.

4.3. Interface

The application's interface is visible in the Proxy Player's side and only audible in the Blind Player's side. We used OpenGL to draw graphism, on the Proxy Player's side, overlayed on top of the running stream coming from the Blind Player's computer. This graphism includes the application's starting menu, the main menu, and all feature menus.

The menus were built with light coloured letters on top of dark rectangle boxes. The rectangles that served as background were achieved using *GL_QUADS*. Navigating through the menu using the F keys will not only activate the features, but also activate the different menus of the application, showing the appropriate background box and text on top.

4.4. Features

All features of this application presented different development challenges, given their varied nature. Ping Points, Snap Mouse, and Steer Mouse dealt, among other issues, with recognizing or resetting Host's mouse position. Vocal Sounds commanded sound files to be played at a time. Control Sharing involved dealing with Parsecs' input permissions. Press Key mode made use of eSpeak text-to-speech console program to indicate which keys to be pressed, hold, double clicked, or pressed repeatedly. Voiced Message and Labelled Ping Points required multi-threading programming when asking for user text input, because this had to be synchronous with the main application running cycle.

The features are called and activated in the Client's side of the program. In order to start using the features, the user, in this case, the Proxy Player, will turn on the Aid Mode. Only then, the application's features will become available to use. The feature's modes work as switches, as in when a switch is turned on, all others that would use the same input are turned off. In other words, let's imagine the Proxy Player is using Snap Mouse to change the Blind Player's cursor position. Then, they decide to send a Ping Point instead. When they turn on Ping Points mode, preparing to send a Ping Point to the Blind Player, Snap Mouse mode will immediately turn off, so that the program distinguishes clicking on the screen to send a Ping Point to that position or clicking on the screen to snap the Host's cursor to that same position. Table 4.1 shows the features that require exclusive mouse input or keyboard input and thus cannot be activated at the same time.

---

[11] Amazon Polly. `https://aws.amazon.com/polly/` (Last visited November 30th, 2022)
[12] eSpeak. `https://espeak.sourceforge.net/` (Last visited November 30th, 2022)

Table 4.1: Exclusive Mouse and Keyboard Inputs

Exclusive Mouse Input	Ping Points, Snap Mouse, Control Sharing: Mouse Control, Press Key: Mouse buttons, Labelled Ping Points
Exclusive Keyboard Input	Vocal Sounds, Control Sharing: Keyboard Control, Press Key: Keyboard buttons, Voiced Messages, Labelled Ping Points

On the Host's side, the activation of the features is informed through Parsec Messages that arrive from the Client's side with an ID attached. The ID is asked when the message arrives, and the program will know which feature is being used. Messages may arrive with additional information, such as screen coordinates (x, y) or a text message to be read out loud.

4.4.1 Ping Points

Client Side

The Ping Point mode is activated in the Client Side using F1. While Ping Point is active, clicking anywhere on the screen will register the X and Y values of the position. These values are then sent to the Host side via a Parsec Message. When the connection between the machines of the players is made, the program registers both screen dimensions, so that when coordinates are sent, they are converted to the right resolution. The user may send a new Ping Point while the previous one was still active, replacing the latter.

When a Ping Point is made, there is graphical feedback in the Client's Side. A light blue point is drawn in the position clicked by the Proxy Player, allowing them to visualise the position where the point has been placed.

Pressing *escape* on the Client Side, as well as on the Host Side, will cancel the current Ping Point.

Host Side

In the Host Side, the Ping Point position is registered as soon as the message arrives. While there is an active Ping Point on the screen, the Host's cursor position is being registered, using *GetCursorPos(x, y)*. This method returns a point containing the actual x and y coordinates of the cursor in the Host's screen. The Euclidean distance between the Host's cursor position and the Ping Point position is continuously being calculated.

The two C methods used to create the pings are *Beep(frequency, duration)* and *Sleep(duration)*. *Beep* generates a beep sound with a frequency, measured in hertz, during a duration, measured in milliseconds. In our case, the durations of the beeps and interval times between the beeps are directly proportional to the distance between the two points, while the frequency of the beeps is inversely proportional.

When the Host's cursor gets very close to the Ping Point position, the cursor will snap to that exact position. This happens when the cursor is within a 75px radius from the position in which the Proxy Player placed the Ping Point on. The point is found, a "correct" sound effect is heard, and the Ping Point is done. The Proxy Player may send new points after or during the Host's search. In the latter case, the Ping Point is replaced by the new one.

In the Host side, the program also listens for user input while this mode is activated. If the user presses *escape*, the beeps stop, and the Ping Point is cancelled. If the user presses *enter*, their cursor will go directly to the pretended position, the "correct" sound effect will play, and the Ping Point will be done.

4.4.2 Snap Mouse

Client Side

This mode is turned on by pressing the F2 key. In the Client side, the Snap Mouse mode is akin to Ping Point mode, meaning while this mode is turned on, clicking on the screen with mouse's left click will register the position and send it to the Host via a Parsec Message, with its unique ID. This position is composed of a X value and a Y value, referring to the coordinates on the Client's screen.

Host Side

Because the screen resolution of the Client might not be the same as the Host's, when the Parsec Message arrives, the position is immediately converted to fit Host's screen resolution using the formulas:

$$X = (Received\ X * Host\ Width\ Resolution) / Client\ Width\ Resolution$$

$$Y = (Received\ Y * Host\ Height\ Resolution) / Client\ Height\ Resolution$$

Equation 4.1 X and Y converted positions.

The Host's cursor is transposed to the converted X and Y coordinates. The Client will see that, in this moment, the cursors occupy the same position on the screen. The Client may snap the mouse more times while the mode is active. When a snap is made, a beep-like sound is heard on the Host side.

4.4.3 Steer Mouse

Client Side

This mode is turned on by pressing the key F3. When it happens, a message is sent to the Host, informing that this mode has been activated. While Steer Mode is active, pressing down an arrow key on the keyboard will send a message to the Host, containing the according cardinal direction as attachment in the message. Two non-opposite arrow keys may be pressed at the same time, sending both messages to the Host, the intention being to make the Host's cursor move diagonally, according to the two directions sent.

Host Side

In the Host side, a message is received regarding the activation of Steer Mouse mode. While this mode is active, the actual Host cursor position is continuously being registered, using *GetCursorPos(x, y)*. When a message arrives, with an ID referring to a Steer Mouse direction, the Host cursor will move accordingly, using *SetCursorPos(x, y)* method in the following manner:

Table 4.2: How the Host cursor position changes, according to the cardinal direction received.

Direction Received	Consequent SetCursorPos() arguments
Right	x + 10, y
Left	x -10, y
Down	x, y + 10
Up	x, y - 10

SetCursorPos(x, y) sets the cursor to a new position. In our case, the cursor will move 10 pixels towards the direction received. In screen coordinates, we have the x horizontal axis, crescent from left to right, and the y vertical axis, crescent from top to bottom. This is why, when the goal is to move the cursor to the right, a value is added to the x, and to move the cursor up, a value is subtracted from the y, and so on.

When two messages regarding a Steer Mouse direction are received in the same cycle iteration, both values are added or subtracted, making the cursor move accordingly. When Steer Mouse mode is deactivated in the Client Side, a message is received, and the program will no longer register Steer Mouse indications.

4.4.4 Vocal Sounds

Client Side

Contrary to the other features, there is no Vocal Sounds mode that needs to be turned on in order to start using the feature. Clicking on F4 will show or hide the complete list of Vocal Sounds available for the Proxy Player to send to the Blind Player. All Vocal Sounds are available to use anytime, including in synchrony with other features that do not use keyboard input (Table 4.1).

The Vocal Sound list also shows which key should be pressed to send a certain sound. For example, the user presses the Y key to send the Vocal Sound “Yes!”, the key G to send “Great!”, or the left arrow key to send “Left!”. The full list of Vocal Sounds can be seen on Figure 3.7. Before sending a Vocal Sound, the program confirms that modes which use keyboard input are turned off, so that when a user

is controlling the Blind Player's keyboard or writing a text message to send a Voiced Message, does not accidentally and without intention send a Vocal Sound.

Host Side

All Vocal Sound audios are present inside a folder amongst the application files. The Host side receives a Parsec Message identified as a Vocal Sound and plays the respective audio file, using the Windows method *PlaySound()*.

4.4.5 Control Sharing

Client Side

In the Client Side, the Proxy Player opens their Control Sharing options using the key F5. Doing this, the menu will switch to show the input controlling options and the respective keys to activate them. The user chooses between Mouse Control, Keyboard Control, or Gamepad Control, by pressing the keys M, K, or G, respectively. Various input controls may be activated simultaneously. When turning each control on, a message is sent to the Host side of the application, informing about the input that the Client is requesting to control.

While an input control is being shared, the user may use other features at will (see Table 4.1 to check which features may not be used at the same time as Mouse Control or Keyboard Control). To cancel the control of a before-hand activated input, the user must follow the same steps they did to activate the input. As an example, the user turns Mouse Control on by pressing F5 and then M; the user will turn Mouse Control off by again pressing F5 and then M. This will inform the Host that the input control will no longer be at use by the Client. Alternatively, the Client may receive a Parsec Message by the Host informing that the Blind Player turned off the Control Sharing input themselves.

Host Side

Parsec natively supports the sharing of controls in its platform, so we integrated this functionality on our expanded application. In the Host Side, as soon as an input control initiates the sharing with a Client, an indicative sound is heard, informing the Blind Player about the happening. The Parsec Permission[13] referring to the input in question (i.e., mouse, keyboard, or gamepad) is set to true to the guest who activated it using the Parsec's SDK ParsecHostSetPermissions() method[14]. This method dictates which Host input controls are permitted for a Client guest to use. The Client can then control the authorised input until that Parsec Permission is set to false again.

If an input control is being shared and the Blind Player presses *escape*, that control stops being shared. A Parsec Message is sent to the Client in case this happens. Alternatively, the Host may receive a message from the Client informing that the sharing of the control has been deactivated. When the

[13] Parsec Permission. `https://parsec.app/docs/sdk/struct/ParsecPermissions` (Last visited November 30th, 2022)

[14] ParsecHostSetPermission. `https://parsec.app/docs/sdk/func/ParsecHostSetPermissions` (Last visited November 30th, 2022)

input control is no longer being shared, whether by Host or Client deactivation, a closure sound is played to the Host.

4.4.6 Press Key

Client Side

The Proxy Player activates Press Key mode by pressing the key F6. While Press Key mode is activated, pressing a keyboard key or clicking on a mouse button will send a Parsec Message to the Host Side regarding the button that is to be pressed and the manner in which it should be pressed. To prevent sending the same Press Key indication multiple times while a key is being held down, the message is sent only after the key has been released, therefore informing the Host only once.

Press Key mode has four different indications regarding how the Blind Player should press the chosen keyboard key or mouse button: press key, hold key, double press, and press repeatedly. By default, it starts with press key indication. Pressing the key *alt* allows the user to switch between the other Press Key variants, cycling in this order: hold key, double press, press repeatedly, and press key again.

The Parsec Message that will be sent to the Host Side will contain an ID referring to which Press Key variant is currently activated and a String text referring to the button or key that was pressed.

Host Side

In the Host Side of this feature, we make use of eSpeak console commands to vocalise the indication sent by the Client. A command will be sent to the system console containing in its text the Press Key variant that it will use, according to the message received.

4.4.7 Voiced Message

Client Side

Voiced Message and Labelled Ping Points come together, Client Side, in a single menu called Text Input Message mode. This mode is turned on by pressing F7. While this mode is active, the user can input a text message into the program to later send as a Voiced Message or as a Labelled Ping Point.

Multi-threading was used to capture the user's input synchronously with the program's main running loop. This way, the program can receive and register what the user is typing without interrupting the video and audio threads.

The text input is received using SDL methods and is then registered into a text variable. Upon pressing *enter,* the user sends the text to the Host via a Parsec Message.

Pressing *right* or *left* arrow keys will message the Host informing the Client's intention to change the Voiced Message's language. The two languages available are English and Portuguese.

Host Side

The Host side receives the Parsec Message informing that a Voiced Message has arrived. The attached text is removed from the message, passed as an argument for eSpeak, and is vocalised to the Blind Player.

4.4.8 Labelled Ping Points

Client Side

The Client Side of Labelled Ping Points is shared with the previous method, Voiced Messages. Being altogether a Text Message Input mode.

The difference being that instead of pressing *enter* to send the written text as a Voiced Message, the user will click anywhere on the screen and send that position as a Labelled Ping Point. The position is registered with a X and a Y value and is sent to the Host via a Parsec Message.

Host Side

In the Host Side, the program recognizes that a Labelled Ping Point has been received and deals with the message almost as if dealing with a regular Ping Point. However, instead of hearing a simple “correct” sound, the Blind Player will hear the text message received by the Proxy Player.

Chapter 5

User Evaluation

We conducted a user study with eight pairs, seven sighted with one of the members unable to see the screen, plus one last pair with a blind participant and the researcher. We had two main goals in this study. The first was to understand the potential of the approach in having a sighted player remotely aiding a blind player in a video game. The second was to evaluate the usability of the features we developed and to understand in which situations, if any, they would be particularly useful. We recruited 14 people to form 7 pairs of players. In each pair, one element was either blindfolded or had their computer screen turned off, and the other element took the role of the Proxy Player. The pairs went through a tutorial where they learned about each of the existing features and how to use them. Then, the study involved the collaborative play of five different games with varying complexity, using the developed platform. Although this user testing phase only included one-on-one interactions, as seen on the first user case (seen on the Design Chapter at the Use Case Scenarios section), users started each game with the condition that only the blindfolded player was able to speak, as happens in the streaming-like scenario seen in the second user case. At the end, all users filled a questionnaire about the experience, providing data to be further analysed. For the last experiment, we recruited a blind participant to try and test the platform. Instead of filling out a questionnaire, this last user was interviewed about his experience using the platform.

5.1 Participants

We recruited 15 participants, 7 pairs, aged 24-28 (M=26.43; SD=1.24) sighted, and one blind participant, aged 27, (B1). Additional information about the participants can be seen on Table 5.1. The recruitment was made through word of mouth by contacting friends and family available to participate and by approaching students at the Faculty of Sciences of the University of Lisbon. An effort was made to recruit people with different levels of gaming expertise, ranging from hardcore players to casual players, to people who do not play video games frequently at all, reaching users with diverse gaming habits. Participants applied to play in pairs, one playing blindfolded portraying the role of the Blind Player (BF1-BF7) and the other playing as a sighted helper, portraying the role of the Proxy Player (P1-P7). People playing blind would use a blindfold or turn off the computer screen. We suggested the screen turn off because they were not using their personal computers and recognizing the keys of a keyboard with a different layout while blindfolded would create an undesired hindrance when acting upon a Vocal Sound or Press Key indication.

Table 5.1: Demographic information and gaming frequency of participants. Each row represents a pair.

ID	Gender	Age	Plays	ID	Gender	Age	Plays
BF1	M	27	Daily	P1	F	25	Occasionally
BF2	M	25	Weekly	P2	F	26	Occasionally
BF3	F	27	Monthly	P3	M	28	Daily
BF4	F	26	Monthly	P4	M	27	Daily
BF5	M	28	Weekly	P5	M	25	Monthly
BF6	F	27	Daily	P6	M	27	Monthly
BF7	M	28	Occasionally	P7	F	24	Monthly
B1	M	27	Occasionally	Researcher	M	28	Daily

We are aware that the study-population is not strictly representative of the target-population. Although, at this stage, our focus was put on the technical evaluation of the solution, being the test with the blind participant the one which provides more information about the impact that this platform can have in the target-population.

5.2 Procedure

Participants were welcomed by learning the intuit of the study, how the developed platform is intended to be used, and how the study would proceed. In pairs, the participants voluntarily took the roles of Blind Player or Proxy Player and sat down in front of the respective computer. The participant portraying the role of the Blind Player put on a blindfold or turned off the computer screen. The video and audio of each study session was recorded using OBS Studio[1], with the respective consent of each participant.

The participants then went through an interactive tutorial for the Proxy Player to learn how to use each feature and for the blindfolded player to learn how to interpret it. We explained each feature to both players and suggested small tasks where they had to apply those features. The tasks were simple actions they did on an image opened in Microsoft Paint[2] application, such as to locate a specific object or to paint part of the image in some specific way. The document containing the tutorial script cand be seen at Appendix D.

[1] OBS Studio. `https://obsproject.com/` (Last visited November 30th, 2022)

[2] Microsoft Paint. `https://apps.microsoft.com/store/detail/paint/9PCFS5B6T72H?hl=pt-br&gl=br` (Last visited November 30th, 2022)

After completing the tutorial, participants played 5 games with diverse complexity and varying challenges. Before each game, a brief description of the game was read out loud so both players had a better understanding of the game they were playing. The participants were free to choose which feature would be appropriate in each gaming situation. Participants played each game for around 15 minutes. The Proxy Player was encouraged not to verbally speak to the blindfolded player in the first 5 minutes of each game and only communicate through the features they had at hand. The blindfolded player, though, was encouraged to speak or the Proxy Player for help whenever they felt necessary. We registered thoughts, suggestions and issues the participants had while playing the games.

At the end of the study, participants filled a questionnaire asking about their experience, general gaming background, which games were most enjoyable or difficult and which features were more relevant at each game or situation. With participant B1, the procedure of the study was the same, apart from the questionnaire. Instead, we conducted a voice interview asking about their experience, thoughts, and suggestions for improving this platform.

5.3 Games

After completing the tutorial and learning how to use the application, the participants played collaboratively five games that presented different challenges. We looked for games that were inaccessible for blind people to play alone, as we wanted to test how our platform would help mitigate the accessibility gap. However, one requisite for the games was the necessity to have auditory components, so that the blindfolded player could have some clues of the events of the game and not be fully dependent on the Proxy Player's assistance to a point where they felt they were not the ones playing the game. To understand the limitations of this platform, games had varying levels of complexity and difficulty. The five games we chose for the study are two-dimensional games since the features were not designed to cover depth perspective. The games also varied in style, ranging from platformer to top-down games. All five games are free to play and available on Steam[3]. Each game was played for about 10 to 15 minutes. For the first 5 minutes, Proxy Players were refrained from verbal communication. Blindfolded players could speak at will. For the remainder of each game, Proxy Players were able to communicate freely with the blindfolded player, as if they were on a voice call.

The five games we chose for the study were CS2D, Antenna, Rogue Glitch, Tadpole Tales, and Super Raft Boat. We tried to select games from a broad spectrum of genres and each game was chosen for different reasons. CS2D was chosen because it is a shooter game that required mouse movement to be played, so the mouse assistance features could be tested. Antenna was chosen because it is a slow-paced platform game that is rich in sounds and includes very unique and sound relying mechanics. We chose Rogue Glitch as one of the games because it is a classical platform game in which the player could take down enemies by getting close to them, as the character starts auto-shooting, or use alternative ways to take down enemies, such as bombs and special items. Tadpole Tales was chosen because it is a game in which the character is in constant movement and the players would need to react to the river's obstacles in a quick fashion. Finally, Super Raft Boat was chosen because, although resembling a top-view shooter game, it contains unique boat building mechanics. This game was also

[3] Steam. `https://store.steampowered.com/` (Last visited on November 30th, 2022)

chosen due to its high degree of difficulty, possibly allowing us to understand the limitations of our platform.

5.3.1 CS2D

CS2D[4], or Counter Strike in two dimensions, is a top-down shooter game. For our study, the game was played in a single-player Zombie mode. The players controlled a single soldier against a team of five zombies they could hear approaching. Players could also hear different game elements, such as their own shooting, reloading, and steps and knew when they reached a wall, because the steps' sound would stop.

Figure 5.1: Game CS2D with the application's menu overlayed on the top left corner, as seen by the Proxy Player (this menu can be hidden).

5.3.2 Antenna

Antenna[5] is a short platformer game where a spider-like robot wanders and investigates its loneliness. Players had to raise up signal towers using keyboard and mouse keys, and search for radio frequencies using the mouse wheel. The game has essentially four different missions, but the players were tasked to complete only the first two. The game is rich in audio elements, as players could easily identify when the signal towers were raising or when the machine was moving, sprinting, or even recognise that they were inside a tunnel.

[4] CS2D. `https://store.steampowered.com/app/666220/CS2D/` (Last visited on November 30th, 2022)

[5] Antenna. `https://store.steampowered.com/app/443580/Antenna/` (Last visited on November 30th, 2022)

Figure 5.2: Game Antenna with the application's menu overlayed on the top left corner (this menu can be hidden).

5.3.3 Rogue Glitch

Rogue Glitch[6] is a rogue-lite platformer in which the player must jump, dodge, shoot, and use items and bombs to surpass the many obstacles. All these elements had a respective sound and shooting was automatic when approaching an enemy, with aim not being necessary. For our study, the players were challenged to complete the interactive game tutorial and to experience the first level of the game. The tutorial had an aspect that made it easier: you could not die.

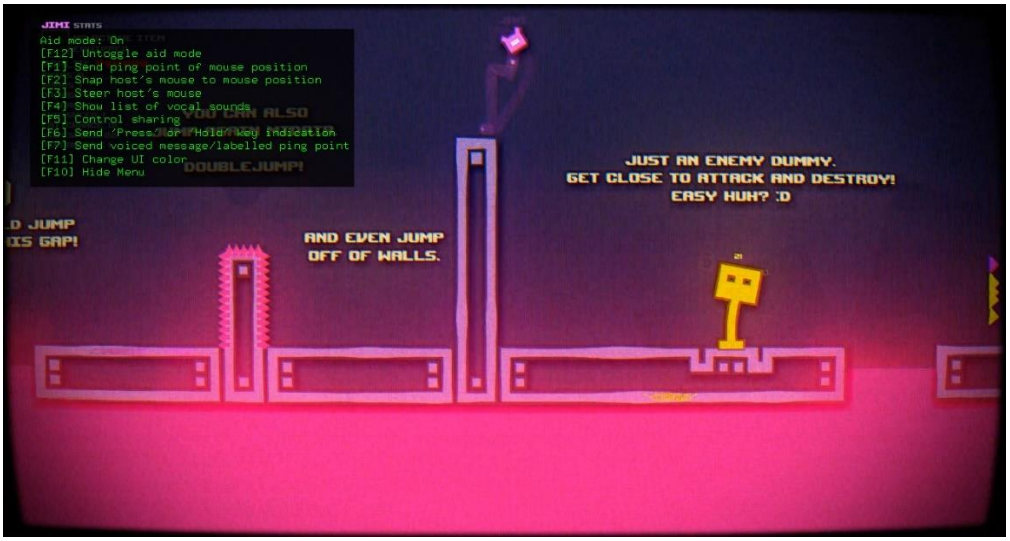

Figure 5.3: Game Rogue Glitch with the application's menu overlayed on the top left corner (this menu can be hidden).

[6] Rogue Glitch. https://store.steampowered.com/app/1092630/Rogue_Glitch/ (Last visited on November 30th, 2022)

5.3.4 Tadpole Tales

In Tadpole Tales[7], the player controls a tadpole that is in constant movement along a river, cleaning enemies with its water shots, and trying to avoid the polluted obstacles that come in the way. The player can also attempt to catch the hearths that spawn in the way, granting more health.

Figure 5.4: Game Tadpole Tales with the application's menu overlayed on the top left corner, as seen by the Proxy Player (this menu can be hidden).

5.3.5 Super Raft Boat

Super Raft Boat[8] is a game in which the player is in a raft boat surrounded by many dangers. Seagulls, rocks, and sharks can disturb the player and even destroy the raft boat the player is desperately hanging to. Due to its fast-paced nature, this game brought the biggest challenge of the study. Various enemies were approaching fast at all times, leaving the players little to no time to react to the many perils they faced. Players had to shoot against the approaching enemies and, at the same time, build more boat squares while dodging the surrounding dangers.

[7] Tadpole Tales. `https://store.steampowered.com/app/1428900/Tadpole_Tales/` (Last visited on November 30th, 2022)

[8] Super Raft Boat. `https://store.steampowered.com/app/1541250/Super_Raft_Boat_Classic/` (Last visited on November 30th, 2022)

Figure 5.5: Game Super Raft Boat with the application's menu overlayed on the top left corner, as seen by the Proxy Player (this menu can be hidden).

5.3.6 Menu Navigation

All games were installed on the blindfolded player's computer. The blindfolded player had to open each game, starting from the desktop, and navigate through the games' menus before starting to effectively play each game. They did this preliminary part using the aid of the Proxy Player as well.

5.4 Apparatus

The study took place in an available room at the Faculty of Sciences of the University of Lisbon. We used two laptop computers, with mice and headphones. Both computers had the platform application, Parsec, eSpeak, and OBS Studio installed. The tutorial file and the online questionnaire was available on both computers. The computer set for the blindfolded player had all the five games of the study installed. We also used a cell phone with hotspot to avoid any possible problems that might arise when logging into steam games in a university network. Apart from technical apparatus, we had two blindfolds, disinfectant, and face masks, due to the pandemic times at which the study took place.

5.5 Data Analysis

All data was collected via the questionnaires the participants filled at the end of the study, notes taken during the study of thoughts and observations the participants were making about the features and the games, video recording of the participants gameplay, and an audio recorded interview with Blind Player B1.

The questionnaire was filled after the study. Firstly, the participants filled in their personal data, including name, demographics, contact, and gaming background. Then, all application's features were

rated by the participants from “not useful at all” to “very useful” in a 5-point Likert scale. Figure 5.1 shows the results of this evaluation. All participants rated the games in terms of enjoyment (from “I did not enjoy playing this game at all” to “I really enjoyed playing this game”) and in terms of difficulty (from “I found this game very difficult to play” to “this game was not difficult at all”). These game ratings were both again in a 5-point Likert scale, where a higher score means the game was more enjoyable or easy, depending on the question. For the participants taking the role of Blind Player only, we adapted autonomy and competence scales from the Ubisoft Player Experience Questionnaire (UPEQ) [35]. The participants playing blind were presented with six autonomy and six competence related statements associated to their experience playing the games in this new manner and had to rate each sentence in in a 5-point Likert scale, value 1 meaning “I strongly disagree with this statement” and value 5 meaning “I strongly agree with this statement”. At the end, all participants responded to open-ended questions asking for thoughts about each feature and in which situations the named feature proved to be most and less useful. Participant B1 was interviewed at the end of the experience. The questions asked in this interview were similar to those asked in the questionnaire. Participant B1 was also asked to share his insight about the general experience and desirable future directions of this work.

5.6 Findings

While playing the games or navigating through the menu, the study participants dealt with challenges in their own ways. Through observation and written notes, we have gathered the player overview, including features and ways that the participants went through while playing each of the games and navigating the menu.

CS2D: Most players (P3, P4, P6, and P7) used Snap Mouse to aim the gun. One player (P2) used Control Sharing: Mouse Control to aim the gun and shoot. Vocal Sounds were used by all the pairs to indicate to shoot the gun, with Vocal Sound: “Shoot!”, and to indicate movement orientations, such as Vocal Sound: “Up!” or “Left!”. One player (P3) used Hold Key instead to indicate what direction the movement should take, addressing the key to be pressed directly, e.g., “Press S” or “Press D” instead of using Vocal Sounds “Down!” or “Right!”. Apart from one pair, all groups succeeded at killing more than one zombie.

Antenna: The first mission of the game was complicated for the players to understand without help from a researcher team member. All pairs used Press Key mode to indicate which keys should have been pressed on to raise the signal towers. To indicate movement direction, some pairs used Press Key, but most pairs used Vocal Sounds. All pairs were able to complete both missions, but the first mission was achieved always with our helping interference, due to the confusing nature of the task.

Rogue Glitch: Some obstacles in the tutorial proved to be a hindrance to some pairs. One of the first obstacles was a very high wall and to go past it the players had to move against the wall while jumping at the same time. Some pairs understood what had to be done, others surpassed that obstacle only with help from the research team. Proxy Players mostly used Press Key mode or Vocal Sounds to assist the blindfolded players actions. One pair (P2 and BF2) completed the tutorial only when free verbal communication between the Proxy Player and the blindfolded player was available. All other pairs completed the tutorial successfully, but none could complete the first level of the game. The pair which lasted the most time in the game first level (P3 and BF3) lasted for one minute.

Tadpole Tales: Vocal Sounds were mostly used by Proxy Players to aid their blindfolded player pairs, indicating the direction they should take to dodge an obstacle, or the time when they should be shooting. Being this a game with infinite ammunition, most Proxy Players indicated blindfolded players to keep shooting perpetually. Players competed with other pairs to get the best score out of the study experiences. The scores table, including with and without free verbal communication within the pair, is shown in Table 5.2, in the Findings Section.

Super Raft Boat: The mouse was used to aim, shoot, and build, while the keyboard was used for movement. Most pairs had trouble lasting during the tutorial and even more in the game itself. The features most used were Snap Mouse, Press Key, and Vocal Sounds.

Menu Navigation: To navigate through the menus, most players (P3, P4, P5, P6, and P7) used Snap Mouse, to snap the mouse to the place or button that was to be clicked on, and Press Key, usually with the indication to press or double press the mouse's left click. Few players used Ping Points (P1 and P3 at first) or Control Sharing: Mouse Control (P2) instead of Snap Mouse to assist in the cursor positioning when navigating through menus. One pair used Vocal Sounds instead of Press Key mode, when the Proxy Player (P5) gave the indication "Run!", the correspondent blindfolded player knew he was supposed to run the game, meaning, in this case, starting the game by double clicking the game icon.

When each study session was completed, participants filled the debriefing questionnaire. Participants, both Proxy and blindfolded players, individually rated each game in terms of enjoyment and difficulty.

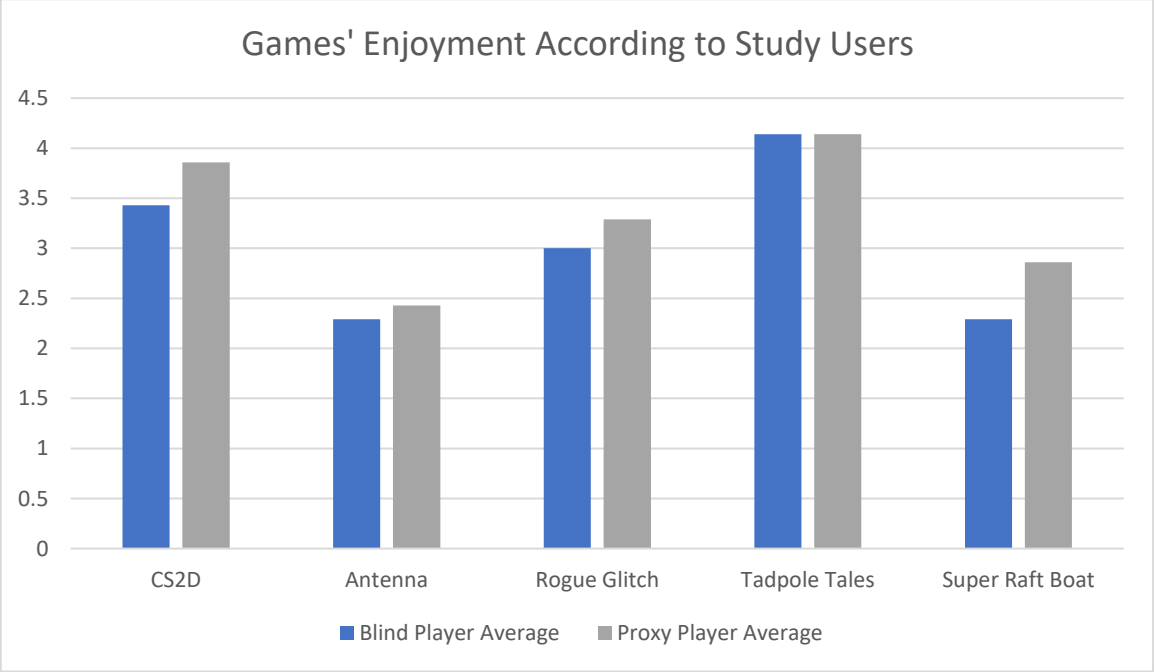

Figure 5.6: Average enjoyment of games according to the participants of the study, ranging from 1 to 5. The higher value meaning the player had the most fun playing this game.

As we can observe on Figure 5.6, the games were enjoyed in similar fashion between both the Proxy Player and the blindfolded players. In other words, there is no game in which one of the players had a

lot of fun playing, while the other did not enjoy playing the game at all. The game enjoyed the most by players, both Proxy and participants playing blind, was Tadpole Tales (M = 4.14, SD = 1.06), followed by CS2D (M = 3.64, SD = 0.81). Tadpole Tales scores achieved by the players are shown on Table 5.2.

Table 5.2: Tadpole Tales Scoreboard. The column "Without free verbal communication" shows the scores the players got while only using the applications' features. In the "With free verbal communication" column, players use the application's features alongside with free verbal communication.

Pair	Without free verbal communication	With free verbal communication
BF1 and P1	170	200
BF2 and P2	100	600
BF3 and P3	150	420
BF4 and P4	400	1190
BF5 and P5	1460	820
BF6 and P6	200	400
BF7 and P7	720	920
B1 and the researcher	430	830

Almost all pairs achieved a higher score when able to speak freely. Interestingly, one pair reached a higher score (and the all-around highest score) using only the application's features. Rogue Glitch (M = 3.14, SD = 0.91) was the third most enjoyed game. At last, we have Antenna (M = 2.36, SD = 1.23) and Super Raft Boat (M = 2.27, SD = 1.24). Participants found some missions in Antenna to be confusing, potentially leading to a lesser enjoyable experience. Most participants found Super Raft Boat to be very difficult and fast-paced even in the tutorial, leading to less enjoyment. The perceived difficulty for all players can be found in Figure 5.7.

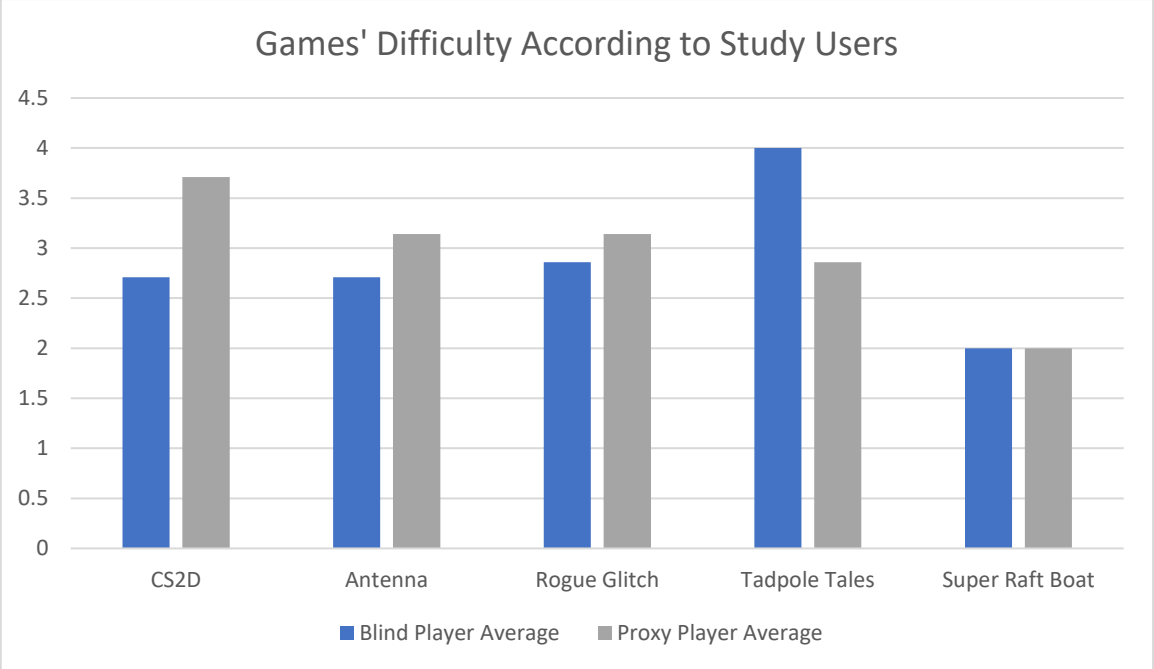

Figure 5.7: Average perceived difficulty of games according to the participants of the study, ranging from 1 to 5. The higher value meaning the player found this game easier to play.

Super Raft Boat (M = 2.00, SD = 1.13) was rated the most difficult game of all five games. Its fast-paced tempo with dangers arriving simultaneously from many different directions led almost all participants to have trouble even completing the tutorial. Antenna (M = 2.93, SD = 0.77), Rogue Glitch (M = 3.00, SD = 1.13), and CS2D (M = 3.21, SD = 0.77) followed next with close results. Antenna rated more difficult among those three probably due to including some enigmatic missions that were only surpassed with help from the researching team. Tadpole Tales (M = 3.43, SD = 1.29) was rated the least difficult of all games played.

The order in which the participants rated, on average, the difficulty was inversely proportional to the order that the participants rated the enjoyability of the games: Tadpole Tales, CS2D, Rogue Glitch, Antenna, and Super Raft Boat, in descending order of enjoyability (ascending order of difficulty). This result demonstrates the players had generally the most fun while playing games they found easier to play.

5.6.1 Features

During the study's tutorial, the participants learned about all the application's features and how to use them. While playing the games, participants were free to decide which feature to use in which situation. After playing the games, all players, Blind and Proxy, rated the features in terms of usefulness. The results are shown on Figure 5.8.

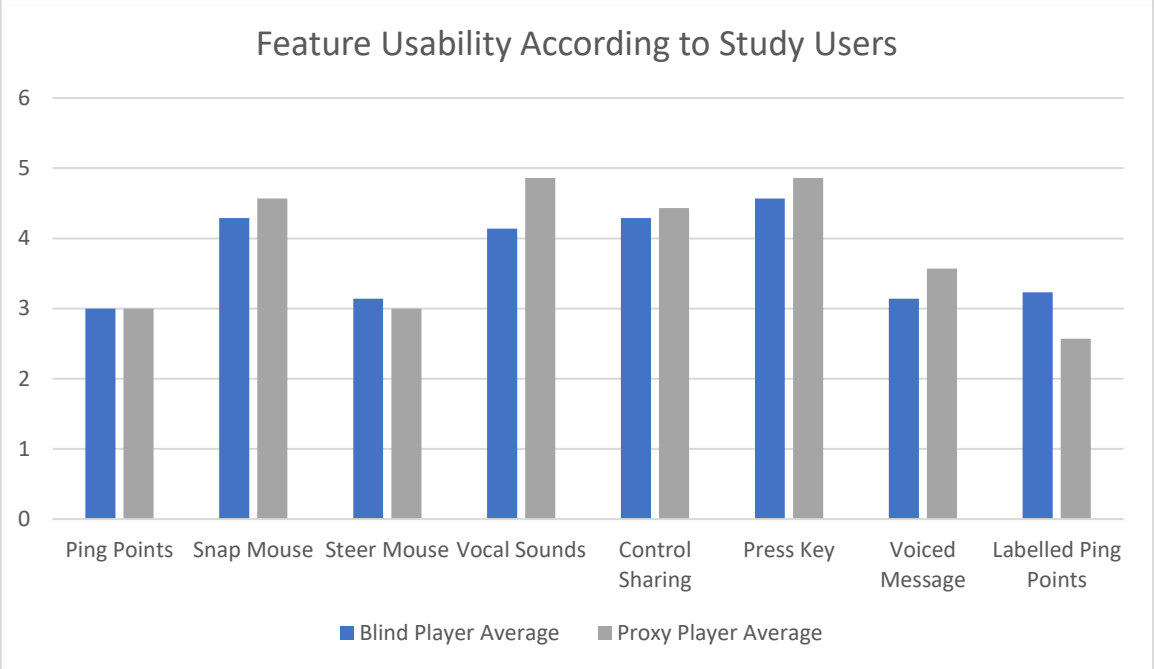

Figure 5.8: Graph of the average feature usability according to the participants of the study, ranging from 1 to 5. A higher value meaning the feature is useful.

There were four features that were ranked with an average usefulness above 4: Press Key (M = 4.71, SD = 0.59), Vocal Sounds (M = 4.50, SD = 0.63), Snap Mouse (M = 4.43, SD = 0.82), and Control Sharing (M = 4.36, 0.81). The other four features ranked lower than 4, but no less than three: Voiced Messages (M = 3.36, SD = 1.54), Steer Mouse (M = 3.07, SD = 1.58), Ping Points (M = 3.00, SD = 1.41) and Labelled Ping Points (M = 3.00, SD = 1.41), in decreasing order.

Both labelled and unlabelled Ping Points were the features that ranked the lowest in terms of usefulness regarding to the players. Participants mostly accused the feature of being slow when compared to other features that achieve the same end in a much quicker way.

"I didn't find it very useful in any situation. Games that use a mouse require a fast input and it is not feasible to wait for the blind user to find the point." – P3

"I think if Snap Mouse is not available, then it is a good tool. Otherwise, Snap Mouse is much faster and straightforward." – P5

In general, participants think that it is a feature better suited for menu navigation and not for dynamic game situations. But even so, other features are able to do the same instantly. Some participants also noted that the noise of the pings could become annoying in the long run or *"could interfere with the game sound"*, as stated by participant BF4.

"It might come handy in a choice menu, pre-game or on pause. In a dynamic game it doesn't seem to work well." – BF7

"Ping Points turned out to be somewhat slow and perhaps (on the long run) annoying due to the noise. Maybe a less noisy option would be good to explore. Its usefulness depends on the game, for example on CS2D it would be difficult to use because it is a fast game, but it could be useful to start the game if the Snap Mouse is not available." – BF6

Regarding particularly the labelled version of Ping Points, some participants stated that it is preferable to regular Ping Points in the sense that the Proxy Player writes what the Blind Player hears upon reaching the position, saving the time of sending another indication in an attempt to identify the point.

"The guide probably saves time and effort by using one of these Ping Points instead of using an "unlabelled" one, just because you don't have to change mode to give an order. As a Blind Player, I found the instruction slightly faster as well." – BF6

Still, participants say it is very time consuming and taking time to write the input may not be feasible in a dynamic game situation.

"I didn't find it very useful in any situation. I think in-game the Proxy Player doesn't have time to write text." – P3

A few pairs used regular Ping Points to open the games on the desktop and no pair used it while playing a game. Labelled Ping Points were not used by any pair after the tutorial, as pairs did not find a gaming situation in which it would be fit to use.

Participant B1 faced difficulties when trying to reach Ping Points' positions. As a blind user, he does not use mouse peripherals while accessing his computer in his day-to-day life. Despite recognizing the direction which the pings were coming from and knowing the relative position of its cursor to the position, participant B1 was not familiar with mouse movement, so the position could not be reached, rendering the method useless.

After the labelled and unlabelled Ping Points, the feature which ranked the least useful was Steer Mouse. Participants compared it to Snap Mouse, but claimed that the latter would be preferable, since it is more strictly to the point.

"Snap Mouse is preferred as you can click exactly where you want." – P5

Participants did not use Steer Mouse mode after the tutorial, as the situations where it might come useful were dealt with in a more direct manner, using Snap Mouse or Control Sharing: Mouse Control.

Still ranked below 4 on average usefulness, but above the aforementioned features, there is the Voiced Messages feature. Some participants said it is a useful feature in situations in which voice chat or an audio call is not available. Participant P7 mentioned it would be *"useful when you want to say something that goes outside the limits of Vocal Sounds"*, as the number of different Vocal Sounds is fixed, while a Voiced Message allows the user to write exactly what they intend to send.

"Voiced Messages are excellent to give quick feedback or to ask for something, in cases where it is not possible to have an audio call." – P5

"It facilitates communication during games and the perception of what to do." – BF5

Unfortunately, many participants found it hard to comprehend the Text-to-Speech voice when the language was set to Portuguese, as, unlike the English voice, it was not very perceptible. Some participants mentioned that, as in Labelled Ping Points, the Proxy Player takes a few seconds to write

and send the text message and it could become an issue in a gaming situation in which the time is scarce or the player needs to communicate various instructions in a short period of time.

The four remaining features scored above 4 in terms of usefulness according to the participants of the study (see Figure 5.8) and received mainly positive feedback from the users. Participants commented that Control Sharing is an all-around useful feature, as it allows the Proxy Player to directly control the Blind Player's input and deal with more complex situations. Some participants noted, though, that the feature might compromise the Blind Player's autonomy or agency.

"It comes handy in a menu or in emergency situations, but it seems to take away the [blind] player's agency." – BF7

One of the intended ways to use Control Sharing was to address situations in which the players took different roles, for example movement and attacking, splitting the controls for each role. Only the pair BF2 and P2 used the feature in this manner while playing CS2D: the blindfolded player (BF2) controlled the movement while the Proxy Player (P2) controlled the gun. Apart from this, pairs used Control Sharing to assist in menu and desktop navigation.

Despite Control Sharing being used by most pairs for navigation, the feature most often used for this end was actually Snap Mouse. Participants used Snap Mouse in most situations regarding mouse actions, including menu navigation and game dynamics that required an active use of the player's cursor, such as the weapon usage of CS2D and Super Raft Boat.

"Very useful in situations where there is more than one dimension of movement, e.g., having to move the mouse and the player with arrow keys. It's easier to Snap Mouse and indicate left/right/up/down on the keys." – P5

"It is useful to start a game, for example, or to help target enemies or in point-and-click games." – BF6

Participant BF6 referred a few situations in which the Snap Mouse feature is useful. Participant P5 mentioned a particular gaming situation in CS2D where they made use of this feature. Most Proxy Players, including participant P5, assisted their blindfolded players in weapon movement using Snap Mouse and in character movement using the Vocal Sound's cardinal indications.

The Vocal Sounds feature was used by all pairs to, among other actions, assist in character locomotion by using the sounds "Up!", "Down!", "Left!", and "Right!" in all five games.

"During the games, it is good to guide the Blind Player in terms of movement." – P7

The sounds "Okay!" and "Great!" were often used to validate some previous action or to bring a state of conclusion to an action. "Wait!" and "Stop!" were used to control the movement flow of the player. Some sounds were specific to some in-game actions that did not exist in all games: participants often used the sound "Shoot!" while playing CS2D and Super Raft Boat and the sound "Jump!" was only used in the game Rogue Glitch, as it was the only game which had jumping as a possible action.

Some participants playing blind noted that sometimes it was not very clear how much time should be taken when following some Vocal Sounds' instructions, as the instruction was not specific in that manner or to that point. For example, participant BF6 mentioned a game situation where the Proxy

Player used the Vocal Sounds “Run!” and “Left!” but needed to use the Vocal Sound “Stop!” afterwards so the blindfolded player knew they should not run left indefinitely.

“Sometimes it wasn’t obvious how long to hold down the key when the voice instructed it. It was necessary to use the “Stop!” command, which often came too late for what the guide intended.” – BF6

Other participants with the Blind Player role mentioned that the times of sending the Vocal Sound instruction, receiving the instruction, and proceeding to perform the action of the instruction given could sum up to an increased amount of time which would make some quick dynamic situations not feasible.

“It is useful in space exploration situations, but in dynamic situations it can be a tool that is too slow.” – BF7

The feature ranked the most useful on average among the participants of the study was the Press Key feature. Using this feature, participants indicated which keyboard keys and mouse buttons should be pressed at specific times during the games and how they should be pressed. Almost every participant commented on the usefulness of the feature and the situations in which it was used, including both menu navigation and games that required mainly discrete keyboard input, such as Antenna.

Similarly to Vocal Sounds, some participants noted that it was difficult to get the timing right when instructing the blindfolded player to make an action in a quick and dynamic gaming situation.

“It is useful, but for very fast games it’s hard to get the timing right for the blind user to press the key.” – P3

“This mode works for slower games. For high dexterity-demanding games, it becomes complicated if the guide is not comfortable with games of this genre.” – BF6

Participant BF7 observed that the Hold Key sub-mode of the Press Key feature could lack some precision when addressing the duration that the key should be held. The same could be said about the Press Repeatedly sub-mode as, again, there is no indication of the intended duration the key should be pressed. In these cases, the Proxy Player would have to articulate with the Vocal Sound “Stop!” or “Okay!” to let the blindfolded player know the key could be released.

“It works in dynamic games. There is some inaccuracy regarding short and long holds.” – BF7

Some participants provided input on how to improve or redesign some of the platform’s features. We discuss the received feedback and recommendations in the Suggestions and Improvements Section of this subchapter.

5.6.2 Autonomy and Competence

We used UPEQ to quantitatively measure participants playing blind engagement in the games and analysed its subscales of autonomy and competence. Each item was ranked by the participants playing blind on a 5-point Linkert scale where 1 means “I highly disagree with this statement” and 5 means “I highly agree with this statement”. The results can be found on Tables 5.3 and 5.4.

Table 5.3: Autonomy related items and correspondent Mean (M) and Standard Deviation (SD) of the blindfolded participants' results.

Autonomy	**M**	**SD**
"I was free to decide how I wanted to play."	2.71	1.28
"I was able to approach games in my own way."	2.43	1.05
"The games allowed me to play the way I wanted to."	2.57	1.05
"I had important decisions to make while I was playing."	2.29	0.88
"The decisions I made while playing had influence on what was happening."	3.43	1.18
"My actions had impact on the games."	4.00	0.93

Blindfolded participants reported an average autonomy of 2.90 (SD = 1.23). Standing just below the mid value of 3, this value might indicate that these participants perceived to have low autonomy while playing the games and felt they were dependent on the Proxy Player indications to progress through the games. The mean scores for the first four items of Table 5.3, which focus on the autonomy of the blindfolded player to make their own decisions in the games, are all below 3, which suggests that these players did not feel they had a sense of agency to take meaningful decisions while playing the games. On the other side, the mean score for the items "The decisions I made while playing had influence on what was happening" and "My actions had impact on the games" are both above 3, indicating that the participants might have felt that the decisions they had to make and the actions they did during the games had influence on the games they've played.

On the other hand, the Blind Player B1 evaluated all of the autonomy related items with a value of 5, meaning he highly agrees with each of the statements. Contrarily to the blindfolded players, participant B1 actually felt like he was the person playing the games and making the decisions, relying on the Proxy Player to provide complementary help only.

Table 5.4: Competence related items and correspondent Mean (M) and Standard Deviation (SD) of the blindfolded participants' results.

Competence	**M**	**SD**
"Over time, I started to play better each of the games."	3.43	1.18
"My playing skills have improved since the beginning of the experience."	3.29	0.88
"My mastery of each game improved with practice."	3.29	1.28
"I felt like a good player."	2.86	1.12
"I felt competent while playing."	2.86	0.83
"I felt capable and effective while playing."	3.00	1.20

The blindfolded players reported an average competence of 3.12 (SD = 1.18) during the study. As seen in Table 5.4, the items "I felt like a good player" and "I felt competent while playing" both failed to reach a value of 3, suggesting that the players did not feel particularly confident in their playing abilities during the experience. The items regarding an improvement over time of the players' skills and mastery of the games during the experience all have mean scores greater than 3, indicating that the blindfolded players felt that their playing capabilities improved over time, to a certain extent. The item "I felt capable and effective while playing" has a mean score of exactly 3, as the players had mixed opinions on the capability and effectiveness felt while playing the games. The relatively high SD values also indicate that the participants had divergent opinions regarding their sense of competence during the experience.

Regarding the precepted competence during the study, the Blind Player B1 felt like a better and more capable player overall when comparing to the blindfolded players. He evaluated all competence related statements with a value of 5, except for the items “I felt like a good player” and “I felt competent while playing” to which he assigned a value of 4. This participant highly recognizes that, over time, his skills got better at the games and, although not attributing the highest possible values to the two items mentioned, he still felt like a competent player.

5.6.3 Suggestions and Improvements

While and after testing the platform, the study participants offered some feedback and suggestions on how to improve the application on further iterations. Regarding the applications’ menu, two Proxy Players suggested some interface design adaptations. Participant P2 wrote *“the menu for those who are guiding the Blind Player needs to be decorated so that the games flow more spontaneously, which might involve a learning period at the beginning”*. Participant P4 also added that icons would make navigating through the application easier. Regarding the features, the participants mentioned that the Portuguese TTS voice was not very perceptible and suggested a change of the software to one which sounds more humane. Participant BF6 recommended the use of less strident and noisy sounds, referring to the Ping Point’s beeps. This participant playing blind also advocated a system to clarify the intended durations to which a certain Vocal Sound instruction should be followed or a key indicated by Hold Key should be pressed. Regarding the Press/Hold Key method specifically, some Proxy Player participants (P3, P4, and P5) referred some changes that would benefit this feature and make it easier to use. Participant P4 suggested *“it should recognize if I’ve pressed or if I’m holding the key and send the correspondent indication”*. Participants P3 and P5 commented that this feature should allow the chaining of multiple keys, for example to press *shift* and *W* at the same time for the character to run.

5.7 Limitations

We conducted a presential study using the researcher’s personal computers, in which the Playing by Proxy platform and the five games were installed. Due to this, the participants played on computers they were unfamiliar with, which may have affected their performance and perception of the platform and the games, especially in participants with the Blind Player role. After the second pair tested the platform, the procedure changed from blindfolding the person playing blind to turning off their screen, helping to set them in a more comfortable position for the experiment.

Some participants of the study did not have a very rich gaming background, sometimes leading to in-game situations in which these users with less experience did not know how to act or progress in the game. Participant P1 faced particular difficulties in understanding how to advance in Antenna and how to build more boat space during Super Raft Boat, receiving shortly after aid from the researching team. Similar situations happened with participants P2 and P6. When the Proxy Player is less knowledgeable in games, the ability to guide the Blind Player in this universe becomes limited, since this platform involves a collaborative play of the games.

Another limitation we had was the fact that we could only count with one blind participant in our study, referred as participant B1, along with seven blindfolded participants. This blind participant was

very important to the study because he provided fundamental insight and feedback that blindfolded participants could not give, regarding the experience of a player that does not see, instead of not seeing only during the experiment. With only one blind participant, it might be difficult to generalize the results to a larger number of blind players.

5.8 Discussion

The participants commented that they found both the experience and the project itself very interesting. For our blind user, participant B1, this experience was a chance to play games that are marketed towards people with a different set of abilities. These games usually come with an accessibility gap that prevents blind players like him from accessing.

"I really enjoyed the experience. I like games a lot. Because of that, it is good to know that there is a possibility of being able to access those types of games that we normally don't get to access. People who don't see don't get access. Of course, there are adapted games, but these specifically, games that sighted people play, we normally don't have access to. For me it is a joy." – B1

The games presented were not accessible for a blind person to play independently, without the assistance of someone who can see or an adequate software. The designed platform did provide an accessibility layer to the games that allowed people playing blind to experiment the games.

"I don't think the games were accessible to blind people. I had the help to start the session, to open the game and there are things that we can't understand right away. But if it was on my computer, maybe I could by chance." – B1

"I think all help was useful. In that zombie game [CS2D] I think you can tell when they're approaching. However, the help is good because we can understand where they come from, if they are more to the right or more to the left." – B1

However, a result we found was that very difficult games still present hard challenges for blind players, even with the use of this platform. The final game, Super Raft Boat, stayed inaccessible for someone playing blind, due to its elevated difficulty level. This helped us identify the limitations of our platform in providing accessibility to games with increased complexity.

"The raft game [Super Raft Boat], did not become that much accessible. The zombie game [CS2D] became quite a lot [accessible]. The one with the spider [Antenna] I think maybe I could play alone if there were more sounds, directing the way the spider should go." – B1

5.8.1 Mouse Assistance

One of the goals of our project was to evaluate the usefulness of the features we developed for the application. The feedback received from our users was crucial for us to understand which features are valuable in different situations and which features did not meet the intended usability we had in mind when in the designing phase of the project.

We learned that the Ping Points, both labelled and unlabelled, were too slow to be a practical tool to use during video games. Even when the goal is to navigate through menus, other features, such as Snap Mouse or Control Sharing, are quicker to achieve this end. Additionally, the feature was not viable when used by a player that is actually blind: participant B1 could perceive the direction the beeps were coming from but could not reach the intended position, even after an elongated period of time. The functionality as it stays is not useful for blind players and requires further improvement or a redesign for it to become viable to use for a gaming end.

The Steer Mouse functionality intent was also to navigate through menus or to adjust the Blind Player's cursor position in situations that required mouse precision. This feature was practically not used during the games because participants had better options for its intended goals: both Snap Mouse and Control Sharing: Mouse Control allowed the users to control the Blind Player's cursor in a quicker and more precise way.

Snap Mouse was seen by our participants as a very viable tool. It was highly rated mainly because it is a lot quicker to use than Ping Points or Steer Mouse. It was widely used both in navigation and in games that relied heavily on cursor movement, such as CS2D and Super Raft Boat. In these games, the character used cursor-controlled guns to defeat enemies. When the Proxy Player snapped the mouse to a particular position, the Blind Player heard a sound that indicated that the mouse had been snapped and knew they had to shoot. This feature, intercalated with other features (particularly ones that assisted in player movement, such as Vocal Sounds or Press Key) and the games' sound effects, made games playable by blind and blindfolded players, bridging a gap in accessibility and allowing for an inclusive gaming experience.

Apart from the aforementioned mouse assistance features, Control Sharing: Mouse Control was also rated as a useful tool by the participants, because it allows the Proxy Player to directly control the Blind Player's cursor for them. As many games include mouse action and blind people don't usually use mouses, features such as Snap Mouse or Control Sharing: Mouse Control could help fill this gap. Being the Proxy Player in charge of mouse movement, it is up for them to choose which Mouse Assistance feature they find the most comfortable using or which fits the best for a given situation, trying to provide an enjoyable experience for both players.

5.8.2 Control Sharing

Control Sharing was seen during this project as more than a feature of the application, but as a different way of play. A way that comes to light, in the context of this work, when a Blind Player and a Proxy Player take asymmetric roles in a game, thus dividing the interaction according to their abilities. An example of this happens in shooter games, when one of the players is in charge of shooting and the other player is in charge of moving the character. This example happened in our experience in the games CS2D and Super Raft Boat. The pair consisting of the participants P2 and BF2 used Control Sharing: Mouse Control with this described intuit, where the Proxy Player P2 controlled the gun with their mouse and the blindfolded player BF2 took care of the movement, while also receiving vocal aid to help guiding this movement. Apart from this, some participants used this remote form of mouse control to assist with menu navigation, while other participants used Snap Mouse or, at times, Ping Points.

The Control Sharing feature also allowed users to share the input of the keyboard or the gamepad. In our study, there was no situation that specifically required the Proxy Player to input text in place of the person playing blind. The situations that required the participants playing blind to use their keyboard in the games were all dealt with independently or with the help of a different feature, typically Press Key or Vocal Sounds. As for gamepad control, although it was implemented, this kind of device was not used during the study.

After the study, participants commented on the usefulness of Control Sharing to deal with complex tasks on behalf of the Blind Player. The possibility to take control of the Blind Player's input to help with tasks like opening and starting the games made participants acknowledge Control Sharing as a helpful tool. Participants also noted that taking full control of the Blind Player's inputs during a game would compromise the player's agency. For this reason, Proxy Players did not use this feature to play the games on behalf of the blindfolded player and, instead, used it for collaborative play during the games. As it is, Control Sharing is a valuable feature that allows for help on complex tasks and, if properly used, assisted and collaborative play.

5.8.3 Voice Assistance

There were four features that assisted the play by generated voice: Labelled Ping Points, Voiced Message, Vocal Sounds, and Press Key mode. In the first two, Labelled Ping Points and Voiced Message, the Proxy Player could write the message that would be read via Text-to-Speech to the Blind Player. As discussed in 5.8.1 Mouse Assistance, Labelled Ping Points, as well as regular Ping Points, proved to be an impractical tool for game assistance and would need further improvement or a redesign to become viable. As for Voiced Messages, participants commented that the fact that it can go beyond the predefined messages of Vocal Sounds or Press Key made it a useful tool. In some very fast-paced games, the users may not be able to write the intended messages on time, unless the message is short and is sent quickly. In more slow-paced or even turn-based games, it can be a powerful tool to describe the context and nature of a gaming situation or to suggest the Blind Player to follow a given instruction.

Vocal Sounds and Press Key were the two features best rated by the users in terms of general usefulness. Those methods allowed the Proxy Players to assist their blindfolded partners in a quick, direct, and understandable way. The Vocal Sounds were used in all games to help guide actions (such as moving, shooting, jumping) or to communicate with simple messages with the "Yes!", "No!", "Okay!", "Great!", and the "correct" and "wrong" sound effects. The Press Key mode was also used by all participants in most games. It allowed the Proxy Player to guide in an even more direct way, telling the blindfolded player specifically which key they ought to press or hold at a given situation. It has four sub-modes: Press, Hold, Double Press, and Press Repeatedly. The Double Press indication was used mostly to help with the starting of games and to navigate through menus, often with "Double Press: Left Click!". Although pressing repeatedly a key is a mechanic commonly found in video games, in our study it was used only in one situation when, in Antenna, the player had to scroll up the mouse's wheel to make rise a signal tower, so most users indicated with "Press Repeatedly: Mouse Wheel Up!". The indications Press and Hold Key were widely used in the games, to indicate keyboard keys or mouse buttons that had to be pressed or held at times, leaving no room for doubt or misunderstanding. Vocal Sounds and Press Key features proved to be important assets that helped make the games playable by blindfolded and blind players.

While voice assistance features were highly rated by the participants, some have noted that they were useful in circumstances where direct audio communication wasn't available. A direct audio call between the Proxy Player and the Blind Player would eliminate the need for features such as Vocal Sounds, Press Key, and Voiced Message, as the Proxy Player could directly convey the information they intend to pass through. For the first five minutes of each game, the participants were proposed to play without free verbal communication. During this time, only the participant playing blind could speak, while the Proxy Player had to use voice assistance features to communicate with their blinded counterpart. This was made to test the voice assistance features we have developed and also to simulate scenarios in which a proxy helper would connect to a player without a voice chat being available. An envisioned scenario is in a Twitch-like streaming context in which a blind streamer has viewers assisting the play. These viewers would assist by not communicating directly with the streamer but, in this case, using the available features of the platform.

Participants found the games easier to play when communicating openly with each other and, in most cases, achieved better scores, as can be seen on Table 5.2. The ability to speak freely with each other allowed the players to overcome the obstacles presented more easily, being able to provide indications, instructions, or to describe the gaming situation or context in a quicker and more natural way than with the features available. Anyhow, the features related to voice assistance were still among the ones which received highest ratings on average by the participants. Receiving mainly positive feedback, especially Vocal Sounds and Press Key, these features had an important part in making the games playable for people playing blind in situations where reciprocal communication was not in effect. This positive performance makes us optimistic to believe that these features would be effective in scenarios where the players are not participating in a voice call, such as using the Playing by Proxy platform on its own or integrated in a service that does not include a reciprocal voice communication channel.

5.8.4 Blind Player Role as a Participative Player

One issue that arose was the concern that the Blind Player could feel like an intermediary between the Proxy Player and the game. We felt that, in some cases, the Proxy Player was playing the game through the Blind Player, instead of with, sending indications of in-game actions they wanted to see fulfilled or, more rarely, controlling the input in order to play the game. In these cases, one can say that the Blind Player is not actually playing the game, but responding to the Proxy Player's commands, being the latter's decision making the factor that determines the interaction with the game.

In the Findings Section, we described how the participants playing blind evaluated their engagement with the games in terms of autonomy and competence. The results suggest that the blindfolded participants felt they had relatively low autonomy during the study, although they felt that their actions had impact on games. In other words, these participants may have felt they were not in charge of making many important decisions while playing the games, but the ones they had were impactful in the course of the games. The competence scores achieved higher values when compared to the autonomy scores. Although the blindfolded participants did not feel highly confident in their abilities, they felt that their skills improved over time. On the other hand, the Blind Player B1 felt significantly more autonomous and competent while playing the games. As his approach was to try and play the games instead of waiting for the Proxy Player's indications, he actually felt like he was the person playing the games and in charge of the in-game decisions. This approach made him feel like an autonomous and competent

player, as the Proxy Player assisted with complementary help instead of playing the game on his behalf and through him.

Regarding the application, there are some features that may grant more agency to the Blind Player than others. For example, using both Mouse Control and Keyboard Control functions from Control Sharing one can play the games on behalf of the Blind Player, removing all of their agency. Of course, this was not the purpose of this project. The goal was to develop features so that the Proxy Player could aid the Blind Player, filling accessibility gaps and coming together in a shared experience. Choosing which feature to use at which time, while staying aware of the degree of autonomy one is taking can be delicate. There are features that aim to achieve the same goal as others, but in a manner that lets the Blind Player keep some agency or even understand better the present situation or the trigger that called for a proxy indication.

Ping Points and Snap Mouse are two of these features, both having the goal to help the Blind Player reach a screen position with their cursor. The Ping Points feature was developed as an attempt to enable this in a more interactive way, having the Blind Player following audio cues to get to the desired location, fostering a sense of agency. Meanwhile, Snap Mouse achieves this end instantly, depriving the Blind Player of agency regarding cursor usage. Ping Points were used by blindfolded users outside the games, to click on icons and navigate through menus, but still this feature proved to be impractical for two main reasons: being too slow in games that required quick reactions; and being unsuitable for blind people who are not accustomed to using mouses. In theory, Ping Points allow the Blind Player to have more autonomy when comparing to Snap Mouse or Control Sharing: Mouse Control, since with Ping Points the Blind Player still have to reach the screen position by themselves, but since the feature was impractical by the reasons stated, the other methods seem to be better suited to deal with mouse assistance.

Regarding voice assistance, Press Key and Vocal Sounds are two features that use audio indications to help the Blind Player. Press Key methods inform the Blind Player exactly what key or mouse button they should press or hold in order to overcome a challenge or complete a task, while Vocal Sounds is used to indicate an in-game action the Blind Player should take, presupposing that the Blind Player knows the necessary keyboard key or mouse button to do it, or to give an auditive feedback. Both features can be used to indicate an action for the Blind Player to take, such as jumping or moving left, although Vocal Sounds can provide more context of the action's nature to the Blind Player. For example, it is different to hear "Jump!" from hearing "Press Space", or hearing "Run Left!" instead of "Hold A". Each first option allows the Blind Player to understand what is happening in the game or what action they are taking by pressing a determined key, as opposed to just hearing the name of the key they need to press. Of course, the number of available Vocal Sounds is limited and, as such, there are Press Key indications that Vocal Sounds cannot cover, for example holding "X" to raise a signal tower in the game Antenna. Therefore, these two features may be used alternately to cover different issues. Vocal Sounds seems preferable to address game actions that are included in the Vocal Sounds' list, as this feature provides the Blind Player with enlightening information about the action they are taking. Press Key complements Vocal Sounds and attempts to cover all the other cases, including game situations that are not enclosed in the Vocal Sounds' list and game mechanics that require more peculiar actions to be done, such as holding particular keyboard keys, pressing repeatedly a button, or double clicking a mouse button for menu navigation.

During the study, many participants hesitated to use the Control Sharing features because, as some have stated, it can remove the Blind Player's agency entirely, as the Proxy Player is controlling the set

input for them. While it is true that this feature hijacks the Blind Player's autonomy if used to control all the Blind Player's action, its main intent, as the name suggests, is to allow for a shared control of the inputs: one player controlling the mouse and the other controlling the keyboard, or both controlling the keyboard but in charge of a different set of keys. The Control Sharing feature is better used for collaborative play, dividing the interaction with the game according to the players' abilities and allowing the Blind Player to keep their agency in asymmetric roles. In particular cases, this feature may remove the Blind Player's autonomy but with the intention to assist in more complex or mundane tasks, such as starting up the games.

In sum, the intent of the platform is to serve as complementary to the Blind Player's playability, as mainly happened with Blind Player B1, and not be a vehicle for the Proxy Player to play the game through the Blind Player. It is also important that the Proxy Player takes into account the degree of autonomy that is being given to the Blind Player, choosing the features that are more adequate for each gaming event while also providing the Blind Player with a fair amount of understanding of the current situation and agency in the game's actions and decision making.

5.8.5 Blindfolded vs Blind Experiences

In the first seven experiments of our study, we counted with 14 sighted participants (Proxy Players P1 – P7 and blindfolded players BF1 – BF7), seven of which playing blind by using a blindfold or turning off the computer screen. Much positive and constructive feedback was received by the blindfolded participants, helping us understand the utility of the features, uncovering several different issues, and assessing our approach. This feedback received by the blindfolded participants, although crucial for the project, had the perspective of a sighted person temporarily without vision during the experience and could lack the kind of insight that only a blind person could give.

In the final experiment, we welcomed a blind participant for the ultimate test to the platform (Blind Player B1). One immediate difference that was noted was the way that this participant approached the games. The common approach taken by the blindfolded participants was to wait for the Proxy Player's instructions and then play the games according to the indications received. As opposed to this, participant B1 proceeded to try to play the games by himself and the instructions received were only complementary. He was more comfortable than the blindfolded participants to experiment the games' mechanics by himself, making him more of a participative player and less of an intermediary between the Proxy Player and the games. In the second game, Antenna, participant B1 almost discarded proxy assistance, as the game was rich in auditory feedback. Because this participant was eager to play the games by himself, the help he received in this game was minor. Vocal Sounds and Press Key features were still utilized to inform which direction should be taken and which key should be held at times, as there were no auditory indications in this game informing about that. This elevated sense of autonomy that we observed on participant B1 in the final test was later corroborated by his debriefing questionnaire in which he evaluated all autonomy related items with a score of 5 points (significantly higher than the average score of the blindfolded players which was 2.90) and felt competent while playing the games, rating the competence related items with an average score of 4.67 (again higher that the blindfolded average score, 3.12).

Another significant divergence noted when experimenting the application with Blind Player B1 was how he responded to the Ping Points feature. As opposed to his blindfolded peers, this participant was

not very familiar with using a computer mouse as it is uncommon for him to use it in his day-to-day life. As such, reaching the Ping Points' position was a task that came with additional difficulty for him. He could tell the direction in which the sound of the pings was coming from and move the mouse towards it, but he could not reach the intended position. After this trial with the Ping Points feature, participant B1 only used the mouse to click on its buttons when games called for it: left click for shooting in CS2D and Super Raft Boat, and mouse wheel to do a variety of tasks in Antenna, such as looking for radio signal or freeing the robot spider at the beginning of the game. As participant B1 did not deal with cursor movement, the mouse assistance features used to assist his play were the ones that were preferable for the Proxy Player at that time.

5.8.6 Improvement Opportunities

The feedback received by the participants, along with the observation of the experiments, helped us discover some technical issues and imperfections existing in the developed features and general application, creating room for improvement. Through the Vocal Sounds feature, Proxy Players can inform Blind Players of a direction towards where they should move. However, participants playing blind often were not able to distinguish whether they should move a short or a long distance, as the indication was not specific in that regard. One way to address this is to create different sounds that specify the intended indication or, in a more subtle way, have the sounds vocalize the indication in a different manner. Analogously, a similar system may be used to indicate whether a key should be held for a short or long period in a Press Key indication.

Regarding the Press Key feature specifically, participants mentioned that, although useful in a variety of situations, it presents a limitation when attempting to indicate two or more keys to be pressed or held simultaneously. Or, as another example, holding *shift* while pressing repeatedly the mouse's left button for the duration. To deal with this issue, strategic key words may be added to the press key instructions, such as "while" or "and", this way allowing the concatenation of indications. For instance, in the example given, the Blind Player would hear "Hold Shift *and* Press Repeatedly Mouse Left Click". Another insight regarding this feature was given by participant P4. This participant, with the role of Proxy Player, mentioned that it would be more intuitive for him if the application recognized if he was pressing or holding a key and sent the correspondent indication, instead of having to select the type of indication (Press, Hold, Double Press, or Press Repeatedly) before pressing the respective key. Implementing the suggestions received by the users would make the Press Key feature easier to use and more viable to cover a variety of in-game situations.

The application would also benefit from having a TTS engine with a more comprehensible Portuguese generated voice, as many users playing blind commented they had trouble understanding the indications that the Proxy Player was sending through the Voiced Message feature. Participant B6 also provided feedback on the beeps' sound when using the Ping Point feature, mentioning they are strident and noisy. Using a softer sound for the pings would make the experience better and more comfortable for both players.

Regarding the menu of the application, seen only by the Proxy Player, two participants provided suggestions on how to make it look better and easier to use. The current design of the menu's interface is very basic, as can be seen on the Navigation Section of the Design chapter, and just suited enough for the intent of the study. As such, it would benefit greatly from having the changes suggested by the

users which included the use of icons. Using icons instead of most of the text would make the navigation through the application more fluid and easier for the Proxy Player.

One issue that hindered the gameplay was an existing short delay between a Proxy Player's vocal assistance indication (Vocal Sound or Press Key) and the actual Blind Player's resulting action. As participant B1 put it, *"one problem was the delay of the help; between the help and the action there is a small delay that sometimes makes it not work so well"*. In some fast-paced games, like Tadpole Tales or Super Raft Boat, agile movement is crucial for success, whether it is for quickly dodging an enemy's projectile or trying to shoot a moving target. If there is a delay in these dexterity demanding tasks, even if it is short, the whole action is hindered. This latency that is felt lasts less than half a second and includes the runtime of the functions in action and the time that the Parsec Message takes to go from the Client to the Host side, that is from the Proxy Player's machine to the Blind Player's machine. If we add this delay to the Blind Player's reaction time it all sums up to an amount of time that makes in-game actions that should be quickly done hindered or not feasible at all. One way to mitigate this is to optimize our code to reduce the runtime and ensure that the users have access to a fast and stable internet connection when using this platform. Still, the human reaction time may be enough to hamper the gameplay when trying to complete very fast-paced gaming tasks, even if the Proxy Player and the Blind Player are side by side talking. This might be an overall limitation as the Blind Player is not reacting instantly to the events of the game, as some in-game situations require.

5.9 Study Conclusions

At an earlier stage of the project, we have set the following research goals for our study: 1. Evaluate whether gaming assistance via a Proxy Player with collaborative play is a possible solution to address the existing accessibility gap for blind people in video games; 2. Validate or invalidate the developed features and understand their usability. Having reached the end of the study, we may now conclude that it is in fact possible to play collaboratively in such a way that some accessibility barriers are overthrown. Still, we have come across some limitations. In the proposed study procedure, the participants chose which feature to use at each moment. Due to this, it is possible that, in some cases, participants stuck to features in which they found immediate comfort, instead of trying other features that might have been better for certain situations. An alternative proposal could have led to different results regarding the evaluation of the features. A major limitation that we could trace after the conclusion of the study was that games that are very inaccessible remain inaccessible even with the use of this platform. The most evident case in our study illustrating this emerged when the participants attempted to play the game Super Raft Boat. This is a fast-paced game in which many perils approach the player quickly from various directions, setting a challenge even for sighted players. For this reason, most participants could not complete the game's tutorial or last more than one minute on the game itself. Adding to the inherent difficulty of fast-paced games, we have found that the human reaction time, together with a short delay on the voice assistance features (Vocal Sounds and Press Key), creates a fraction of delayed time that may hinder agile gameplay, making the platform not effective to the degree that was intended in these types of games.

Chapter 6

Conclusion

For many years, games have been a popular source of entertainment for many people. More than that, games are also a unique way for people to socialize and get together doing a fun and interactive activity. Although video game accessibility is a theme that is becoming more present in the developers and designers' minds, still the gaming world is very divided between games for blind people, such as audio games, and games for sighted people, commonly taken as regular games. This work is created as an effort to make these two worlds of gaming collide, attempting to enable everyone to experience games that, by norm, tend to exclude players with vision disabilities.

We introduced Playing by Proxy, a platform that allows Proxy Players to provide on-demand assistance to Blind Players, as a means to make some gaming visual challenges accessible. Using the Parsec SDK, we developed a stream-like application that allows a sighted person to see the gaming screen of a blind gamer in real time. This application adds an overlapped menu to the screen that grants access to the features of the application. These eight features aim to address different gaming challenges and obstacles and can be divided into three categories: Mouse Assistance, Voice Assistance, and Control Sharing. In Chapter 3 we explain the design process and motivations that drove us to create this application. In Chapter 4 we delve into the implementation of Playing by Proxy, including all of its features. In Chapter 5 we go through a detailed review of the user study, in which 7 pairs of sighted and blindfolded participants, as well as a final blind person, tried and tested the application. We gathered all feedback from the participants related to the features and the experience itself by recording the sessions, writing down commentaries on a notebook, a final online questionnaire, and a recorded interview with a blind participant (B1). After the study was completed, we were glad to confirm that this unusual gaming paradigm allows the playability of some games and the overcome of some accessibility obstacles for blind players. Limitations to this approach have been found and are further described on the Study Conclusion Section of Chapter 5. We have learned that some of the features developed are good tools to address accessibility challenges, while others are not very effective on their intent and need further implementation of reconceptualization in order to become useful in these gaming contexts. With Playing by Proxy, we aimed to explore a shared gameplay approach to gaming as a way to bridge the existing gap in ability-categorized video games by connecting players of all abilities within the same video games. We believe that this approach was successful to some extent, and the feedback received by the study participants was crucial to make us better understand the potential and the flaws of the application. We hope that this project sets an advance on accessibility gaming or inspires others to conceptualize unique techniques or different approaches to help foster inclusion in the gaming world.

This project was a huge opportunity for me to learn more about gaming accessibility, collaborative play, asymmetric roles in video games, and social inclusion in the gaming world. Going through this project and dealing with the many issues that arise, I feel that I have become more aware of accessibility issues in the digital and physical world. I have also come to understand that the accessibility of an environment is not only composed of its physical features, such as wheelchair ramps or a tactile surface with Braille. It also includes the social awareness of the needs of people with disabilities. The communal acknowledgement that there are invisible barriers with higher challenges for people with impairments or conditions represents an important primary step towards addressing these issues. In the digital world, this acknowledgement may elicit developers and designers to have accessibility into consideration when

developing and designing games or other applications, as well as other users to be aware of the needs of people with disabilities and become an active part in trying to close this accessibility gap, instigating for a more welcoming and inclusive environment. In short, and akin to the real world, the digital world should have a culture of inclusion for all people in order to become accessible.

6.1. Future Directions

We hope that this project can represent an important stride towards gaming accessibility and inclusion. Still, there is a long path to be trodden in order to achieve the desired outcome. The first few steps to refine the Playing by Proxy application in the upcoming work are related to trying to implement some of the suggestions received by the study participants and aftermath thoughts the researching team had on how the platform could be improved. This might include, but should not be limited to: 1. Redesigning the application's menu, using icons for example, to make it more intuitive to use; 2. Changing the Portuguese text-to-speech voice to a cleaner and more comprehensible one; 3. Reviewing the Vocal Sounds and Press Key indications to make it possible for the Blind Player to have an idea of how long they should follow the received indication, discerning from a long or short action to be taken; 4. Adding ways to the Press Key feature that allow the user to indicate actions that are currently outside the scope of the feature (pressing, holding, double pressing, and pressing repeatedly) such as, for example, holding two keys simultaneously, holding a key while pressing other repeatedly, or press alternately and repeatedly two different keys. This might be achievable with the integration of strategic key words such as "and" or "while" between indications. And, finally, to rethink, reconceptualize, or remove entirely the features that, for different reasons, were not useful or efficient to the purpose of the project.

We believe that a shared-play approach to video game accessibility is a theme that is still not very explored, while holding a lot of potential. Our tests were made on a very specific scenario, with one-on-one pairings playing relatively simple 2D games, but this same approach could be transposed into a broader range of gaming contexts. In order to incorporate more complex games, including 3D games, the features of Playing by Proxy should be reviewed and adapted to the new scenarios, if needed. Another possible direction Playing by Proxy could take would be to allow for multiple Proxy Players assisting one Blind Player at the same time, as a crowd-source service. In this case, the application would have to be re-tweaked, as there would be many people sending instructions through the features instead of one. One idea to achieve this would be to implement a kind of voting system in which each use of a feature, a Vocal Sound, for example, would correspond to a vote and the most voted instruction would go through. In other words, the mode of a list of instructions would be the chosen one by this example. We used Parsec SDK for our project, but other API's and services, such as Twitch, could be used instead to explore a way for the viewers to connect directly and assist a blind streamer, as participative spectators. The Blind Player could, at times, enable or disable the connection of new Proxy Players. Or it could work as a Be My Eyes style service, in which the Blind Players could call for Proxy Players in times of necessity. These are just some ideas that could serve as inspiration for the basis of a discussion for the future and continuation of the Playing by Proxy project.

"That would be amazing. To become able to play FIFA or Call of Duty in that way. It would be incredible!" – Participant B1 after hearing about the intended direction for the future of Playing by Proxy and the eventual possibility for a blind streamer to play games with the assistance of participative viewers.

Chapter 7

References

[1] World Health Organization, “Global Data on Visual Impairment 2010,” *Global Data on Visual Impairment 2010*, p. 17, 2010, [Online]. Available: http://www.who.int/blindness/GLOBALDATAFINALforweb.pdf

[2] S. M. Branham and S. K. Kane, “Collaborative Accessibility: How Blind and Sighted Companions Co-Create Accessible Home Spaces,” pp. 2373–2382, 2015, doi: 10.1145/2702123.2702511.

[3] S. M. Branham and S. K. Kane, “The Invisible Work of Accessibility: How Blind Employees Manage Accessibility in Mixed-Ability Workplaces,” in *ASSETS 2015 - Proceedings of the 17th International ACM SIGACCESS Conference on Computers and Accessibility*, Association for Computing Machinery, Inc, Oct. 2015, pp. 163–171. doi: 10.1145/2700648.2809864.

[4] K. M. Gerling and L. Buttrick, “Last tank rolling: Exploring shared motion-based play to empower persons using wheelchairs,” *CHI PLAY 2014 - Proceedings of the 2014 Annual Symposium on Computer-Human Interaction in Play*, no. Figure 1, pp. 415–416, 2014, doi: 10.1145/2658537.2661303.

[5] D. Gonçalves, A. Rodrigues, and T. Guerreiro, “Playing With Others: Depicting Multiplayer Gaming Experiences of People With Visual Impairments,” pp. 1–20, 2020.

[6] D. Gonçalves, T. Guerreiro, and A. Rodrigues, “Exploring Asymmetric Roles in Mixed-Ability Gaming,” 2020.

[7] A. E. Depping and R. L. Mandryk, “Cooperation and interdependence: How multiplayer games increase social closeness,” *CHI PLAY 2017 - Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pp. 449–461, 2017, doi: 10.1145/3116595.3116639.

[8] R. O. C. Hci and M. I. T. Csail, “Introducing Shared Character Control to Existing Video Games University of Rochester Technical Report # 986,” 2007.

[9] F. J. Thiel and A. Steed, “‘Lend Me a Hand’ - Extending the reach of seated VR players in unmodified games through remote co-piloting,” in *Proceedings - 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, VRW 2021*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021, pp. 214–219. doi: 10.1109/VRW52623.2021.00047.

[10] G. Cimolino, S. Askari, and T. C. N. Graham, “The Role of Partial Automation in Increasing the Accessibility of Digital Games,” in *Proceedings of the ACM on Human-Computer Interaction*, Association for Computing Machinery, Sep. 2021. doi: 10.1145/3474693.

[11] D. Gonçalves, P. Pais, K. Gerling, T. Guerreiro, and A. Rodrigues, “Social gaming: A systematic review,” *Comput Human Behav*, vol. 147, p. 107851, Oct. 2023, doi: 10.1016/j.chb.2023.107851.

[12] L. M. G. Fonseca and S. D. J. Barbosa, “Knowledge Sharing Live Streams: Real-time and On-demand Engagement,” in *International Conference on Enterprise Information Systems, ICEIS - Proceedings*, Science and Technology Publications, Lda, 2021, pp. 441–450. doi: 10.5220/0010401104410450.

[13] M. Horton, J. C. Read, and C. Willitts, “Incude: Heuristics for enhancing spectator experience in streamed games,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12211 LNCS, pp. 97–116, 2020, doi: 10.1007/978-3-030-50164-8_7.

[14] A. Striner, A. M. Webb, J. Hammer, and A. S. Cook, “Mapping design spaces for audience participation in game live streaming,” in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, May 2021. doi: 10.1145/3411764.3445511.

[15] J. Seering *et al.*, “Audience participation games: Blurring the line between player and spectator,” in *DIS 2017 - Proceedings of the 2017 ACM Conference on Designing Interactive Systems*, Association for Computing Machinery, Inc, Jun. 2017, pp. 429–440. doi: 10.1145/3064663.3064732.

[16] B. S. Tekin and S. Reeves, “Ways of spectating: Unravelling spectator participation in Kinect play,” *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2017-May, pp. 1558–1570, 2017, doi: 10.1145/3025453.3025813.

[17] B. Maurer, I. Aslan, M. Wuchse, K. Neureiter, and M. Tscheligi, “Gaze-Based Onlooker Integration,” pp. 163–173, 2015, doi: 10.1145/2793107.2793126.

[18] J. Fanzo *et al.*, “What Lurks in the Dark: An audience participation horror game,” *CHI PLAY 2017 Extended Abstracts - Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*, pp. 621–624, 2017, doi: 10.1145/3130859.3130865.

[19] P. Lessel, M. Mauderer, Christianwolff, and A. Krüger, “Let’s play my way: Investigating audience influence in user-generated gaming live-streams,” *TVX 2017 - Proceedings of the 2017 ACM International Conference on Interactive Experiences for TV and Online Video*, pp. 51–63, 2017, doi: 10.1145/3077548.3077556.

[20] G. Kalmpourtzis, “Examining the Impact of an Interactive Storytelling Platform on Educational Contexts Through Contemporary Crowdsourcing Methods of Audiovisual Content Publishing,” pp. 0–4, 2020.

[21] L. Engeln, M. Hanke, T. Auerswald, F. Kallenbach, and R. Groh, “The Earconizer-A Tool for constructing hierarchical Earcons VisualAudio-Design View project Dresden Design Hub View project The Earconizer-A Tool for constructing hierarchical Earcons,” 2019, doi: 10.18420/muc2019-ws-436.

[22] D. K. Mcgookin and S. A. Brewster, “Understanding Concurrent Earcons: Applying Auditory Scene Analysis Principles to Concurrent Earcon Recognition,” *ACM Trans Appl Percept*, vol. 1, no. 2, pp. 130–155, 2004, doi: 10.1145/1024083.1024087.

[23] V. Nair *et al.*, “NavStick: Making Video Games Blind-Accessible via the Ability to Look around,” in *UIST 2021 - Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology*, Association for Computing Machinery, Inc, Oct. 2021, pp. 538–551. doi: 10.1145/3472749.3474768.

[24] S. Cmentowski and J. Krüger, “Playing With Friends - The Importance of Social Play During the COVID-19 Pandemic,” pp. 209–212, 2020, doi: 10.1145/3383668.3419911.

[25] X. Tong, Di. Gromala, C. Neustaedter, F. D. Fracchia, Y. Dai, and Z. Lu, “Players’ Stories and Secrets in Animal Crossing: New Horizons Exploring Design Factors for Positive Emotions and Social Interactions in a Multiplayer Online Game,” in *Proceedings of the ACM on Human-Computer Interaction*, Association for Computing Machinery, Sep. 2021. doi: 10.1145/3474711.

[26] T. Olsson, “Improving Accessibility for Shooter Games An explorative study of the possibility to systematically improve the accessibility for shooter games,” 2020.

[27] D. Grammenos, “Game over: Learning by dying,” *Conference on Human Factors in Computing Systems - Proceedings*, no. August, pp. 1443–1452, 2008, doi: 10.1145/1357054.1357281.

[28] F. Da Rocha Tomé Filho, P. Mirza-Babaei, B. Kapralos, and G. M. Mendonça, “Let’s Play Together: Adaptation guidelines of board games for players with visual impairment,” *Conference on Human Factors in Computing Systems - Proceedings*, no. April, 2019, doi: 10.1145/3290605.3300861.

[29] “Game Accessibility Guidelines.” http://gameaccessibilityguidelines.com/ (accessed Nov. 30, 2022).

[30] R. Andrade, M. J. Rogerson, J. Waycott, S. Baker, and F. Vetere, “Playing blind: Revealing the world of gamers with visual impairment,” *Conference on Human Factors in Computing Systems - Proceedings*, no. April, 2019, doi: 10.1145/3290605.3300346.

[31] “Blind People Will Be Able To Play The Last of Us Part 2.” https://www.digitalphablet.com/blind-people-play-the-last-of-us-part-2/ (accessed Nov. 30, 2022).

[32] J. Jun, W. Seo, J. Park, S. Park, and H. Jung, “Exploring the Experiences of Streamers with Visual Impairments,” *Proc ACM Hum Comput Interact*, vol. 5, no. CSCW2, Oct. 2021, doi: 10.1145/3476038.

[33] S. L. R. Anderson and M. R. Johnson, “Gamer identities of video game live streamers with disabilities,” *Inf Commun Soc*, vol. 25, no. 13, pp. 1900–1915, 2022, doi: 10.1080/1369118X.2021.1907433.

[34] D. Gonçalves, M. Piçarra, P. Pais, J. Guerreiro, and A. Rodrigues, “‘My Zelda Cane’: Strategies Used by Blind Players to Play Visual-Centric Digital Games,” in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, Apr. 2023. doi: 10.1145/3544548.3580702.

[35] A. Azadvar and A. Canossa, “UPEQ: Ubisoft perceived experience questionnaire: A self-determination evaluation tool for video games,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Aug. 2018. doi: 10.1145/3235765.3235780.

Appendix A

Call for Users

Olá a todos! @everyone

Estou a fazer uma dissertação chamada “Playing by Proxy”, cujo objetivo é tornar possível que pessoas cegas joguem jogos originalmente inacessíveis.

A ideia do estudo é acrescentar uma camada de acessibilidade através de ajudantes normo-visuais que terão diferentes ajudas ao seu dispor, desde enviar pistas de áudio ao jogador cego até partilhar os diferentes controlos de jogo.

Como o pede esta fase do projeto, necessito de testar a aplicação criada com utilizadores voluntários para recolha de dados e feedback.

Este estudo será feito em pares Blind Player (jogador vendado) / Proxy Player (ajudante não vendado) e terá lugar na FCUL. A duração de cada sessão de estudo é à volta de uma hora e inclui:

- Tutorial (~10 minutos)
- 5 jogos (entre 5 a 15 minutos cada)
- Questionário final (~ 10 minutos)

Caso estejas interessado em participar no estudo, por favor preenche este doodle e marca as horas em que tenhas disponibilidade:
https://doodle.com/meeting/participate/id/bkRq8EXa

Obrigado!
Salvador Meira

Appendix B

Guião de Estudo

Introdução

Muito obrigado por terem aceitado participar neste estudo. O meu nome é Salvador Meira e sou estudante de Mestrado em Engenharia Informática aqui na FCUL.

O objetivo da tese é tornar possível que pessoas cegas joguem jogos originalmente inacessíveis. Assim, a ideia do estudo é acrescentar uma camada de acessibilidade através de ajudantes normo-visuais que terão diferentes ajudas ao seu dispor, desde enviar pistas de áudio ao jogador cego até partilhar os diferentes controlos de jogo.

Estudo

Temos como objetivo para este estudo perceber se as dinâmicas que criámos funcionam e que tipos de jogos se tornam jogáveis através de que funcionalidades. Assim, vamos pedir que formem pares "Proxy Player" e "Blind Player" para jogar os vários jogos e experimentar as diferentes funcionalidades/features. O "Proxy Player" será o jogador ajudante normo-visual, que utilizará as diferentes features para auxiliar o "Blind Player". O "Blind Player" será o jogador cego, como o nome indica, e deverá tentar progredir nos diferentes jogos, sendo auxiliado pelo "Proxy Player".

Os dois jogadores estarão ligados pela aplicação e a todo o momento o Proxy Player consegue ver o ecrã do Blind Player. O Proxy Player terá à sua disposição um menu em overlay para ajudar na navegação da aplicação e na escolha das ajudas a utilizar.

No final da sessão, iremos pedir que preencham um questionário sobre a experiência.

Até agora, alguma questão?

Sessão

Em primeiro lugar, vou perguntar quem quer fazer o papel de Proxy Player (ajudante não vendado) e quem quer fazer o papel de Blind Player (jogador vendado).

Peço que se sentem à frente dos respectivos computadores. A partir daqui, peço para o Blind Player se vendar.

Vamos começar por fazer o tutorial, onde usarão cada uma das features **[Tutorial]**. Antes de cada ponto, vou mostrar-vos as funcionalidades/features (i.e. as ajudas que serão usadas do lado do Proxy Player) para que possam experimentar e concluir cada tarefa do tutorial **[Guião de**

Features]. Por favor informem-me sobre qualquer questão que surja. (Estará sempre um ecrã aberto com o guião de features).

A qualquer momento, o Blind Player pode, se quiser, tirar a venda. Vou também pedir o vosso consentimento para gravar a sessão, através de captura de ecrã e áudio, usando o programa OBS Studio.

Vamos então passar para os jogos. Antes de cada jogo, vou dar-vos uma breve descrição para terem uma ideia do que irão jogar **[Descrição dos Jogos]**.

Em cada situação de jogo, o Proxy Player pode escolher as features que achar mais adequadas. Peço que joguem cada jogo pelo menos 5 minutos e no máximo 15 minutos. Para já, o Proxy Player não deverá falar, apenas auxiliar através da aplicação.

Depois dos primeiros (3/4/5) minutos, os dois jogadores podem falar à vontade, juntando a ajuda verbal direta às ajudas dadas pela aplicação. O Proxy Player pode tirar os fones, para facilitar a comunicação.

O objetivo em cada jogo será progredir o máximo que se conseguir. No CS2D começamos com um modo de sobrevivência Zombies e depois passamos para um modo de desativar a bomba.

Debriefing

Jogados os jogos e feita a experiência, peço apenas que preencham este questionário final **[Questionário]**.

Mais uma vez, muito obrigado pela participação!

Appendix C

Guião de Features

As features estão disponíveis do lado do **Proxy Player** e são recebidas de diferentes formas do lado do **Blind Player**.

Ping Points (F1)

Os **Ping Points** servem para o **Blind Player** encontrar, interativamente, uma posição no ecrã escolhida pelo **Proxy Player**. Funciona da seguinte maneira:

- O **Proxy Player** carrega numa posição do ecrã à escolha.
- O **Blind Player** ouve "pings" que serão cada vez mais frequentes ao aproximar o seu rato da posição escolhida pelo **Proxy Player**.
- Após encontrada a posição, ouve-se um som de conclusão e os "pings" param.

O **Blind Player** pode cancelar os **Ping Points (ESCAPE)** ou ir diretamente até à posição escolhida **(ENTER)**.

Snap Mouse (F2)

Com o modo **Snap Mouse**, o **Proxy Player** transpõe instantaneamente o rato do **Blind Player** para a posição escolhida. O **Blind Player** ouve um som cada vez que isto acontece para alertar que a posição do seu rato foi alterada.

Steer Mouse (F3)

O modo **Steer Mouse** permite ao **Proxy Player** ajustar a posição do rato do **Blind Player**, usando as setas do teclado.

Vocal Sounds (F4)

Os **Vocal Sounds** são sons que estão sempre disponíveis para o **Proxy Player** enviar ao **Blind Player** e dão diferentes tipos de informação. A lista de **Vocal Sounds** é:
[Y] - Yes!
[N] - No!
[J] - Jazzy Sound!
[O] - Okay!
[S] - Shoot!
[R] - Run!
[T] - Stop!
[G] - Great!
[W] - Wait!
[,] - Correct Sound!
[.] - Wrong Sound!
[ARROWS] - Up/Down/Left/Right

Control Sharing (F5)

O modo **Control Sharing** permite ao **Proxy Player** controlar diretamente diferentes tipos de input do **Blind Player**, podendo inclusive controlar vários ao mesmo tempo:
[M] - Controla o rato.
[K] - Controla o teclado.
[G] - Controla o gamepad.
A qualquer altura, o **Blind Player** pode cancelar o **Control Sharing (ESCAPE)**.

Press/Hold Key (F6)

Neste modo, o **Proxy Player** indica ao **Blind Player** uma tecla que deve ser premida ou mantida. O **Proxy Player** carrega na tecla pretendida (podendo incluir botões do rato) e o **Blind Player** ouvirá a indicação, e.g. “Press SPACE”, através dum programa Text-to-Speech (eSpeak).

Voice Message / Labeled Ping Point (F7)

Por último, o **Proxy Player** pode escrever uma palavra ou frase para enviar ao **Blind Player**, que será lida através do eSpeak. Carregando **(ENTER)**, a frase é lida diretamente; carregando com o botão do rato, é enviado um Labeled Ping Point com a palavra ou frase escrita: o **Blind Player** ouvirá os “pings” como no modo Ping Point, mas aqui, ao chegar à posição pretendida, ouvirá a palavra ou frase escolinha pelo **Proxy Player**.

Appendix D

Tutorial

Esta imagem deverá estar aberta com o Paint do lado do **Blind Player** e o **Proxy Player** deve se ligar através da aplicação.

Figure D.1: *The Binding of Isaac*[1] gameplay, where Isaac, the main character, is at a shop.

0. Antes de começarmos a experimentar as funcionalidades, experimenta mudar a cor do menu **[F11]** e esconder o menu **[F10]**.

1. Ping Points [F1]

Proxy Player: Mete um Ping Point em cima da carta.

Blind Player: Encontra a posição escolhida. Quanto mais perto estiveres, mais frequentes serão os pings. Podes também carregar **[ESCAPE]** para parar os pings ou **[ENTER]** para ir diretamente para a posição.

2. Snap Mouse [F2]

Proxy Player: Faz com que o cursor do Blind Player vá para o fogo da direita, depois o fogo da esquerda e finalmente termine em cima do mapa

3. Steer Mouse [F3]

Proxy Player: Utiliza agora este modo para levar o cursor do jogador até ao comprimido.

---

[1] The Binding of Isaac: Rebirth. `https://store.steampowered.com/app/250900/The_Binding_of_Isaac_Rebirth/` (Last visited on November 30th, 2022)

4. Vocal Sounds [F4]

Proxy Player: Tenta levar o cursor do Blind Player até ao escadote (em cima, à direita) só através de Vocal Sounds. Carrega **[F4]** para ver a lista de Vocal Sounds.

Blind Player: Encontra com o cursor a posição escolhida, através dos Vocal Sounds recebidos.

5. Control Sharing [F5]

Proxy Player: Usa o modo Control Sharing -> Mouse Control (**[F5] -> [M]**) para controlar o rato do Blind Player. Usa este modo para desenhar sobrancelhas ao Isaac (personagem que está a chorar). No final, desactiva o controlo do rato da mesma maneira (**[F5] -> [M]**).

6. Labeled Ping Points [F7]

Proxy Player: Envia um Labeled Ping Point para o sítio onde está a porta (em baixo) informando que é a porta.

Blind Player: Encontra a posição escolhida.

7. Snap Mouse [F2] e Press Key [F6]

Proxy Player: Utiliza estes dois modos para gravar a imagem.

8. Voiced Messages [F8]

Proxy Player: Utiliza este método para enviar uma mensagem final ao Blind Player.

Appendix E

Descrição dos Jogos

CS2D

O famoso jogo Shooter, Counter Strike, mas em duas dimensões e visto de cima (top-down). Será jogado em single-player e no modo Zombies. O Player Character movimenta-se usando as teclas "WASD", faz pontaria e dispara usando o rato e recarrega a arma com a tecla "R". Não deixes que os Zombies te apanhem!
Depois no modo Standard, esperar que os terroristas plantem a bomba e tentar desarmar a bomba (E).

Antenna

Antenna é um curto platformer onde uma máquina investiga a sua solidão. Faz erguer torres de sinal usando teclas do teclado e busca frequências de rádio usando o rato. O jogo tem essencialmente quatro missões diferentes.

Rogue Glitch

Rogue Glitch é um platformer roguelite onde o jogador deve saltar, esgueirar-se, disparar e usar itens e bombas para ultrapassar os obstáculos. O jogo permite duplo salto, saltar em paredes, falar com NPCs e para disparar basta aproximar-se do adversário.

Tadpole Tales

A Terra dos Sapos foi atacada por monstros que poluíram as águas. O jogador controla um girinho que está em constante movimento ao longo dum rio e tem como objetivo limpar os adversários com os seus disparos de água, ao mesmo tempo que deve desviar-se dos obstáculos. Usa o espaço para disparar e o teclado para te desviares (WASD ou setas).

Super Raft Boat

O mundo ficou inundado e tudo o que nos resta é uma jangada. Este é um jogo onde o jogador está num barco com perigos à sua volta. Usa-se o teclado para movimentar/desviar dos inimigos (WASD), o botão esquerdo do rato para disparar e o botão direito do rato para construir mais blocos de jangada, onde te podes movimentar. A dada altura, chega ao barco uma garrafa que permite ao jogador escolher um de três trinkets que ajudarão o jogador na sua demanda. Cuidado que há inimigos que destroem a jangada e se caíres na água tens pouco tempo até começares a perder vida! **Pode-se ligar o auto-disparo nas opções do jogo (auto-fire: on).**

Appendix F

Questionário

1. Nome:
2. Email:
3. Idade:
4. Quantas vezes jogas jogos digitais?
    a. Menos de uma vez por mês.
    b. Uma ou duas vezes por mês.
    c. Uma ou duas vezes por semana.
    d. A maior parte dos dias.
5. Em média, quão longas são as tuas sessões de jogo?
    a. Menos de uma hora.
    b. De uma a duas horas.
    c. De duas a quatro horas.
    d. Mais de quatro horas.
6. Eu considero-me:
    a. Um jogador casual.
    b. Um jogador hardcore.
    c. Algo entre um jogador casual e um jogador hardcore.
    d. Não costumo jogar jogos digitais.
    e. Não sei.
7. Dá alguns exemplos de jogos (podem ser de tabuleiro) que representem aquilo de que gostas em jogos (escreve os títulos dos jogos, separando-os com vírgulas):
8. Papel que desempenhei neste estudo:
    a. Proxy Player
    b. Blind Player

Lembrete: Descrições resumidas das funcionalidades
Ping Points: Proxy Player seleciona uma posição do ecrã, Blind Player ouve "pings" que serão mais frequentes quanto mais perto esta de encontrar a posição.
Snap Mouse: Proxy Player transporta instantaneamente o rato do Blind Player para a posição do seu rato.
Steer Mouse: Proxy Player ajusta a posição do rato do Blind Player usando as setas do teclado.
Vocal Sounds: Sons padrão que o Proxy Player envia ao Blind Player (eg: "yes", "no", "left", "wrong").
Control Sharing: Partilha de controlos, como o rato, teclado ou joystick.
Press/Hold Key: Proxy Player informa ao Blind Player que tecla deve premir ou manter premida.
Voiced Message: Proxy Player escreve uma frase que será lida em text-to-speech para o Blind Player.
Labeled Ping Point: Igual ao Ping Point, mas no final é dita uma frase à escolha do Proxy Player.

For both the Proxy and the Blind Player, but the results are separated:

1. Classifica o grau de utilidade de cada feature, sendo 1 "nada útil" e 5 "muito útil".
    a. Ping Points
    b. Snap Mouse
    c. Steer Mouse
    d. Vocal Sounds
    e. Control Sharing
    f. Press/Hold key
    g. Voiced Message
    h. Labeled Ping Point
2. Classifica cada jogo em relação ao divertimento, sendo 1 "eu não me diverti a jogar este jogo" e 5 "eu diverti-me muito a jogar este jogo".
    a. Antenna
    b. CS2D
    c. Rogue Glitch
    d. Super Raft Boat
    e. Tadpole Tales
3. Classifica cada jogo em relação à dificuldade, sendo 1 "eu achei este jogo muito difícil de jogar" e 5 "eu achei este jogo muito fácil de jogar".
    a. Antenna
    b. CS2D
    c. Rogue Glitch
    d. Super Raft Boat
    e. Tadpole Tales

For the Blind Player only:

1. Para cada item, seleciona a opção que melhor caracteriza a tua experiência (Colunas: Discordo totalmente, Discordo, Não tenho a certeza, Concordo, Concordo totalmente).
    a. Eu fui livre de decidir como queria jogar.
    b. Eu pude abordar os jogos à minha própria maneira.
    c. Os jogos permitiram-me jogar da maneira que eu quis.
    d. Tive decisões importantes para tomar enquanto jogava.
    e. As decisões que tomei enquanto jogava influenciaram o que acontecia.
    f. As minhas ações tiveram impacto nos jogos.
    g. Para cada um dos jogos, com o tempo, comecei a jogar melhor.
    h. As capacidades a jogar melhoraram desde o início da experiência.
    i. O meu domínio do jogo melhorou com a prática.
    j. Eu senti-me bom jogador.
    k. Eu senti-me competente a jogar.
    l. Eu senti-me capaz e eficaz quando estava a jogar.

Again, for both the Proxy and the Blind Player:

1. O que achaste dos Ping Points? Em que situações achaste mais e menos úteis?
2. O que achaste do modo Snap Mouse? Em que situações achaste mais e menos úteis?
3. O que achaste do modo Steer Mouse? Em que situações achaste mais e menos úteis?

4. O que achaste dos Vocal Sounds? Em que situações achaste mais e menos úteis?
5. O que achaste do Control Sharing? Em que situações achaste mais e menos úteis?
6. O que achaste do modo Press/Hold Key? Em que situações achaste mais e menos úteis?
7. O que achaste das Voiced Messages? Em que situações achaste mais e menos úteis?
8. O que achaste dos Labeled Ping Points? Em que situações achaste mais e menos úteis?
9. Comentários, feedback ou sugestões de novas funcionalidades: