UNIVERSITY OF TURKU

Enhancing HTML5 Games Accessibility: An Exploration of Assistive Technology for People with Visual Impairments

Master of Science Thesis
University of Turku
Department of Computing
Software Engineering
2024
Xuexian Chen

UNIVERSITY OF TURKU
Department of Computing

Xuexian Chen: Enhancing HTML5 Games Accessibility: An Exploration of Assistive Technology for People with Visual Impairments

Master of Science Thesis, 70 p., 5 app. p.
Software Engineering
May 2024

Digital games have offered enjoyment to countless people, however, people with disabilities are often excluded from these experiences. Visually impaired individuals, in particular, encounter challenges in gaming as the majority of digital games heavily rely on graphics and visual content. While there are accessible games, such as audio games, tailored for them, there remains a lack of an assistive tool that can support a wide range of digital games.

To fill this gap, this thesis aims to explore the feasibility of a universal assistive technology to support multiple HTML5 games. HTML5 games are a subset of web games that can be played via web browsers without the need for installation. An assistive technology prototype in the form of a Chrome extension was developed in this thesis, with a requirements design drawing insights from existing accessibility guidelines and related accessible games. The prototype covers common accessibility features such as font, cursor customization and magnifier, along with advanced features including narration, highlight and sonification.

Exploratory testing and task-based usability testing were conducted to evaluate the prototype. Testing results indicate that most features of the prototype can effectively enhance game accessibility, yet some require further optimization. Despite limitations, this work sets a solid foundation for further research and development in this area.

Keywords: HTML5 game, Assistive technology, Accessibility, Sonification, Chrome extension, Audio game, Accessibility guideline

Contents

1 Introduction

Over the past few decades, digital games have become a prominent source of entertainment for countless individuals, with their market share steadily on the rise [1]. Despite ubiquity, many people are still excluded from this activity due to various factors, among which, the lack of accessibility stands as a primary concern, especially for individuals with disabilities [2][3].

Even with challenges, individuals with disabilities are inclined to engage in digital gaming [4]. A plethora of studies indicates that digital gaming can not only promote physical rehabilitation and improve cognitive abilities but also foster a sense of competence and facilitate social connections, ultimately contributing to their long-term wellness (e.g., [5]–[8]). Furthermore, considerable commercial potential exists in people with disabilities as game players. Research points out that a growing population of gamers with disabilities actively engage in mainstream commercial games [8]. Thus, enhancing game accessibility is a mutually beneficial opportunity.

This opportunity has not gone unnoticed by the gaming industry, which is why, accessibility support within the commercial gaming sector has substantially developed recently in terms of the range and variety of options available [9]. However, accessibility options are often tailored for specific games, making them replicable but non-transferable. The majority of games still lack adequate accessibility support, as an assistive technology that can accommodate a variety of digital games is notably rare.

To address the absence of an assistive technology that effectively supports a wide range of digital games to enhance the general gaming experience of people with disabilities, this thesis focuses on the possibility of developing an assistive tool that supports multiple digital games.

1.1 Scope and Objective

Existing assistive technology solutions are often customized for a specific user group, reflecting the common awareness that "perceptions and needs on assistive technologies vary according to the type of disability" [10]. Present accessibility guidelines also tend to categorize based on specific disability types though the breakdown may not be clean as different groups can benefit from common solutions [4]. The disability types that have been extensively researched concerning gaming experiences typically include blindness and visual impairments, hearing impairments, physical impairments and cognitive disabilities.

To minimize the scope and effectively manage the workload, this thesis narrows its scope to focus on the needs of people with visual impairments. Nevertheless, assistive technology solutions for other disability groups are also studied to ensure a comprehensive understanding and identify any insights that can be utilized.

Given the diverse range of genres and platforms of digital games [11], the focus of assistive technology features can vary based on the specific genre of the game [12]. In addition, the technological platform on which the game is executed plays a crucial role in determining the technical solution adopted during development. Factors such as engine selection and code are implicitly impacted, thereby influencing the coverage and portability of the assistive tool.

This thesis primarily concentrates on HTML5 games within the point-and-click adventure genre. A point-and-click adventure game is a story-driven game where "players progress through an overarching narrative by solving puzzles" [13], the main interaction is done through a point-and-click interface like a computer mouse. The mechanics and content of point-and-click adventure games offer adequate material for assessing assistive technology.

HTML5 games are a subset of web games which can be played through a web browser without downloading or installing [14]. The decision to focus on HTML5 games is based on several factors:

1. Significant developments in web accessibility support have been witnessed in the past few decades, with many screen readers being invented to support non-sighted people to surf online as effortlessly as possible. This provides a solid foundation for researching HTML5 game accessibility.

2. Compared to other platforms, it is relatively easier to modify HTML5 games and make further adjustments to align with accessibility standards. This minimizes challenges for actual implementation.

3. Most HTML5 games are free-to-play and easily accessible, contributing to a large number of active gamers. Besides, HTML5 games have a huge commercial potential yet to be exploited [15]. This ensures the research findings can be valuable for both the gamers and the market.

In summary, based on the above-mentioned scope, the primary objectives of this thesis are:

1. To examine existing assistive technology solutions designed for individuals with visual impairments,

2. To explore the possibility of developing an assistive tool for visually impaired people to support a range of narrative-driven HTML5 games with point-and-click interaction as the major mechanics.

1.2 Research Questions

By examining the existing solutions for visually impaired people in gaming and exploring the possibility of developing an assistive tool for visually impaired people to enhance their experience with a range of HTML5 games, this thesis aims to answer the following three research questions:

- RQ1: What assistive technology features can be integrated into HTML5 games while preserving the integrity and originality of the games?
- RQ2: How to implement such a tool?
- RQ3: How effective does the implementation of the assistive tool enhance the game experience for visually impaired people?

1.3 Thesis Structure

This thesis is organized into six chapters. Following this chapter, Chapter 2 provides an overview of various types of visual impairment, accessibility, and assistive technology, along with existing accessibility guidelines. Chapter 3 explains key concepts relevant to the thesis scope and reviews previous research and work on audio games and accessible games. Drawing insights from the prior chapters, Chapter 4 proposes a way to implement an assistive tool for HTML5 games, adopting a classic software development process from requirements analysis to feature implementation. To evaluate the implemented prototype, Chapter 5 presents the adopted testing methodology and analysis of the testing results. Finally, Chapter 6 concludes the thesis and discusses the future work.

1.4 Acknowledgment

I would like to thank Professor Jouni Smed at the University of Turku for his supervision and support of this thesis. I would also like to thank every participant in the testing.

Additionally, ChatGPT [16] is used for proofreading in this thesis.

2 Background

In this chapter, essential background information regarding game accessibility will be provided to set the stage for the subsequent chapters. Section 2.1 offers a concise overview of different types of visual impairments and the challenges they pose in gaming. Following this, Section 2.2 introduces key concepts related to accessibility, including a discussion of existing accessibility guidelines.

2.1 Visual Impairement

Visual impairment, also known as vision impairment, refers to the partial or complete loss of vision that cannot be corrected through the use of glasses or surgical procedures. At least 2.2 billion individuals worldwide are affected by visual impairments [17]. Common types of visual impairments include:

- *Blindness* is the complete loss of vision and, in most severe scenarios, the inability to perceive any light.
- *Low vision* encompasses any partial loss of eyesight that negatively affects one's ability to see and therefore perform daily activities [18]. Typical examples include blurred vision, tunnel vision and blind spots, as illustrated by simulated perspectives in Figure 2.1:
  - Blurred vision, such as near-sightedness, far-sightedness, presbyopia or astigmatism, is often caused by refractive errors [19].

  - Tunnel vision, the loss of peripheral vision while retaining the central vision [20].
  - Blind spots, which partially obstruct or obscure the visual field [21].

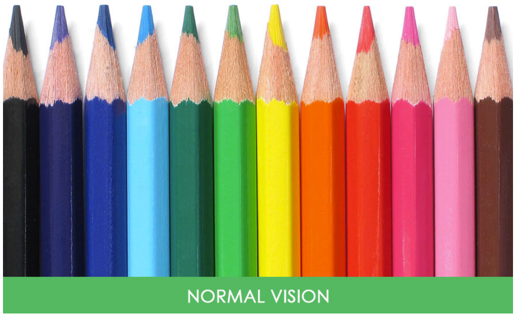
- *Colour Blindness*, or colour vision deficiency, refers to a diminished capacity to distinguish between colours and in severe cases, an inability to see differences [22]. It includes deuteranopia, protanopia, tritanopia, and monochromacy, which is complete colour blindness. Figure 2.2a shows the perception of colours for people with normal vision, while figures 2.2b - 2.2e simulate the presentation of those colours to people with each type of colour blindness.

(a) Normal vision (b) Blurred vision (c) Tunnel vision (d) Blind spots

Figure 2.1: The perspective of normal vision and simulated low vision. Source: adapted from [18]

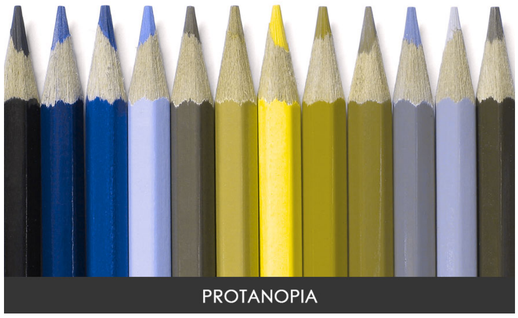
2.1.1 Challenges for Visually Impaired People in Gaming

Barriers for individuals with visual impairments can vary, but a significant challenge arises from the heavy reliance on visual content, including graphics, text, and animations, to convey information in games [18], [23]. While auditory and haptic cues often serve as a complement, useful in catching attention and enhancing immersion, they are usually insufficient for those completely blind, making progression in the



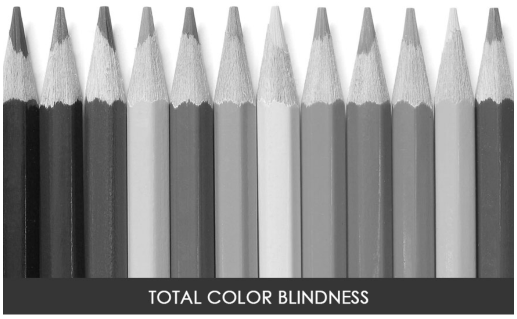
(a) Normal Vision

(b) Deuteranomalia



(c) Protanopia

(d) Tritanopia



(e) Monochromacy

Figure 2.2: How colours are perceived by people with normal vision and with colour blindness. Source: `https://www.boredpanda.com/different-types-color-blindness-photos/`

game world and navigation in user interfaces extremely difficult even impossible. This constraint also negatively impacts low-vision and colour-blind people.

Additionally, visually impaired individuals may require more time to process information conveyed through visual cues, thereby challenges increase when the game quest is time-constrained or irreversible. According to [23], compared to individuals with normal vision, visually impaired players may spend more time on a game as they pay careful attention to dialogues and game content, thus, excess information can become a distracting challenge, slowing down their game progress.

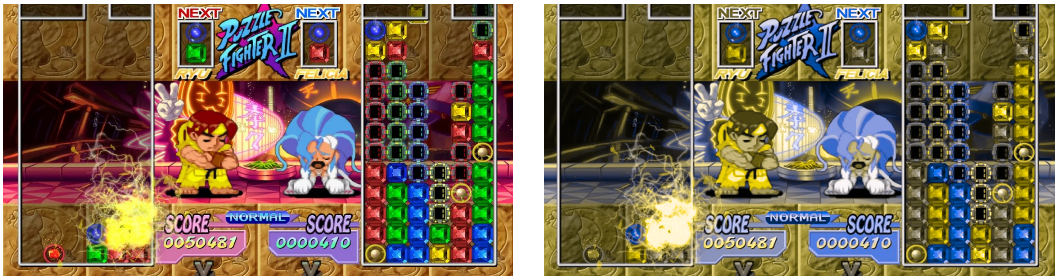
Specifically speaking, for low-vision and colour-blind people, the lack of adequate assistive tools and accessibility options remains a common challenge. A 2020 survey shows that more than two-thirds of game developers do not consider accessibility when designing their games [24].

Moreover, games that rely solely on colour to convey critical information can present challenges for individuals with colour vision deficiencies. For instance, in *Super Puzzle Fighter II Turbo*, a puzzle game where players must match gems of the same colour, those with protanopia may struggle to distinguish between yellow and green gems because they appear almost identical, as demonstrated by Figure 2.3.

2.2 Accessibility

The International Organization for Standardization (ISO) has issued a series of standards and guidelines for accessibility across different sectors. According to [25], accessibility refers to the degree to which individuals from the widest range of diverse contexts of use can access a product or service to achieve their specified goals. As a subcategory of usability, which focuses on the overall use experience encompassing effectiveness, efficiency and satisfaction [26], accessibility is designed to expand the target user group [25].

A common yet inaccurate belief is accessibility is intended for people with disabil-



(a) Normal vision (b) Protanopia

Figure 2.3: How the game *Super Puzzle Fighter II Turbo* appears to people with normal vision and with protanopia. Note: 2.3b is generated through a simulator at `https://www.color-blindness.com/coblis-color-blindness-simulator/`. Source: `https://static1.thegamerimages.com/wordpress/wp-content/uploads/2021/02/Super-Puzzle-Fighter-II-Turbo.jpg`

ities, in fact, accessibility benefits everyone [27]. A typical example is the wheelchair ramp, which was initially designed for physically disabled people in wheelchairs but was found to be beneficial for many others like those with items of luggage or strollers when put into real use. Additionally, everyone may experience temporary disabilities or injuries at some point, making everyone a potential beneficiary of accessible design.

However, despite the well-established instructions and guidelines for the accessibility of interactive systems in the Human-Computer Interaction (HCI) field, accessibility and broader usability remain common factors for the failure to meet user needs [28]. The root cause of such an issue is efforts focusing on functional requirements at the expense of non-functional accessibility requirements during software development, ultimately resulting in higher costs and more development time [29].

2.2.1 Assistive Technology

In the late 1980s, assistive technology was initially defined with the goal "to enable independence by disabled and older people" [30]. However, as societal perceptions evolve and new technologies emerge over the years, it is now defined as any technology-based service or product that assists individuals in completing their daily activities and therefore promotes their well-being [31], [32]. The World Health Organization (WHO) identified assistive technology as a subset of health technology [33].

Assistive products, as the critical element of assistive technology, encompass any device designed to improve a person's functioning and reduce disability [34]. Common examples of assistive products include wheelchairs, hearing aids, and smart homes.

As for assistive products in computer use, sip-and-puff switches can be used by individuals with mobility impairments to perform actions through inhaling and exhaling, while screen readers, with output to Text-to-Speech (TTS) voice or refreshable braille displays, are widely used by non-sighted and low-vision people.

2.2.2 Accessible Game

Accessible games are those designed to ensure that players with various disabilities, such as visual, hearing, physical, and cognitive impairments, can fully enjoy the gaming experience [18]. These games provide equal opportunities to all target players, regardless of their limitations or challenges.

One common way to make existing games accessible is by adding accessibility options. For example, the first-person shooting game *Overwatch* by Blizzard Entertainment provides several colour-blind modes that will change the colour display of the whole screen tailored for people with deuteranopia, protanopia, and tritanopia [35]. Similarly, accessibility options can be seen in *Call of Duty: Black Ops 4*, *Forza Horizon*, and *Battlefield 4*.

Additionally, there are games specifically designed and developed to meet the needs of people with particular types of disabilities, such as audio games tailored for non-sighted players. For more details of audio games, see Section 3.2.

2.2.3 Accessibility Guidelines

Accessibility guidelines comprise a collection of principles and recommendations, often drawn from studies, experiences and literature, to ensure product accessibility [4]. They usually serve as a checklist for developers to evaluate their products, often during the product acceptance testing phase.

As this thesis explores the feasibility of creating an assistive tool for HTML5 games, it is necessary to form a comprehensive understanding of accessibility requirements and user expectations, which can be derived from these guidelines. This guarantees that essential features are considered during the implementation phase.

The following paragraphs will discuss the three most extensive and well-known accessibility guidelines relevant to this thesis, before looking at a guideline from previous research. Due to the extensive content of each guideline, which often consists of dozens of items or more, only necessary items will be mentioned in this thesis.

Web Content Accessibility Guidelines (WCAG)

The World Wide Web Consortium (W3C) initially released a set of internationally recognized standards known as the Web Content Accessibility Guidelines (WCAG) in the late 1990s, to ensure that web content is accessible to all users. The guidelines were updated periodically in the past years and the newest version is available in [36]. While the specific content may evolve slightly between each version, the four primary principles remain consistent:

1. *Perceivable*, which emphasizes that web content and user interface should be presented in a way that can be perceived by all users, regardless of their usable modality. It prioritizes alternatives for text, images and other media to be inclusive for users with different sensory abilities.

2. *Operable*, which ensures that all functionality within web content can be operated using various inputs, such as keyboards, and that the content is easily navigable.

3. *Understandable*, making sure that web content and user interfaces are clear and easy to understand, including readable text, consistent presentation, and effective error corrections.

4. *Robust*, which means that content should be compatible with a wide range of user agents, including assistive technologies.

Although this guideline was originally designed for assessing web applications and may not be perfectly suited for evaluating digital games, it still offers insights. Consequently, research [37] proposes a serious game accessibility guideline for low-vision people, with knowledge drawn from WCAG.

Additionally, both HTML5 games and the assistive tool to be implemented can be classified as web applications, which requires a thorough study of WACG.

In the subsequent subsections, two well-known game accessibility guidelines will be introduced: the Accessible Player Experience (APX) and the Game Accessibility Guidelines (GAG). Given their shared focus on game accessibility and their foundations in the knowledge and experience of game developers and players, it is unsurprising to find common principles between them [4]. Furthermore, both sets of guidelines categorize principles according to types of disabilities, including visual, auditory, motor, and cognitive impairments.

Accessible Player Experiences (APX)

The Accessible Player Experience (APX) [38] [39], created by the AbleGamers Charity, organizes its principles, which in this case are referred to as design patterns, into three layers. These layers form the core of APX, represented by the APX Triangle, where Access occupies the bottom layer, Challenge sits in the middle layer, and APX resides in the top layer, as demonstrated in Figure 2.4a. Each layer serves as a necessary foundation for its upper layer.

The Access layer ensures the feedback loop between the game world and the player is unimpeded, which means, the user input can be successfully transmitted into the game world, and the output from the game world is perceivable by the player. The Access Design Pattern, as shown in Figure 2.4b, consisting of twelve principles, provides various instructions on how to facilitate this feedback loop. Following are several patterns related to visual impairments:

- Second channel: low-vision and blind players should be able to select alternative channels, auditory or haptic, for visual information.
- Personal interface: players can rearrange user interface components for more efficient usage.
- Clear Text: players can customize text presentation for better readability, such as changing the background colour of a text.
- Do More with Less: players can progress the game with fewer actions.
- Distinguish This from That: players should be able to change content presentation for clear differentiation, for instance, by customizing colour contrast.
- Clear Channel: players have the option to change attributes of information in visual and auditory channels for more efficient perceiving.

Once players have full access to games, APX addresses the importance of providing diverse levels of challenges, which can be achieved by following the ten design patterns in the challenge layer, see Figure 2.4c. Some worthy noticing patterns include:

- Redo, Undo: players can reverse or repeat their previous actions.
- Slow it Down: players can reduce the level of challenge in games by decreasing the speed, volume and frequency of events.
- Bypass: players have the option to skip some parts of a game to continue the game.
- Training Ground: games should offer various ways for players to practice their skills at their own pace.
- Total recall: players can review game instructions.

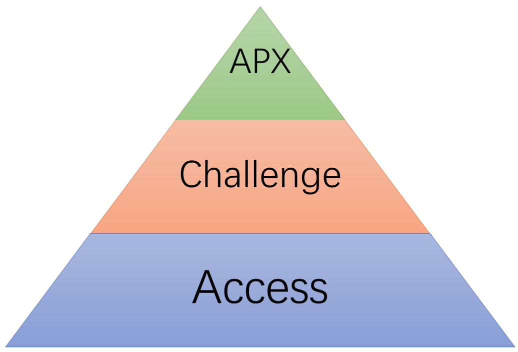
The top layer, APX, is automatically achieved once the requirements of the lower two layers are fulfilled. Therefore, there are no specific design patterns for the APX layer.

Game Accessibility Guidelines (GAG)

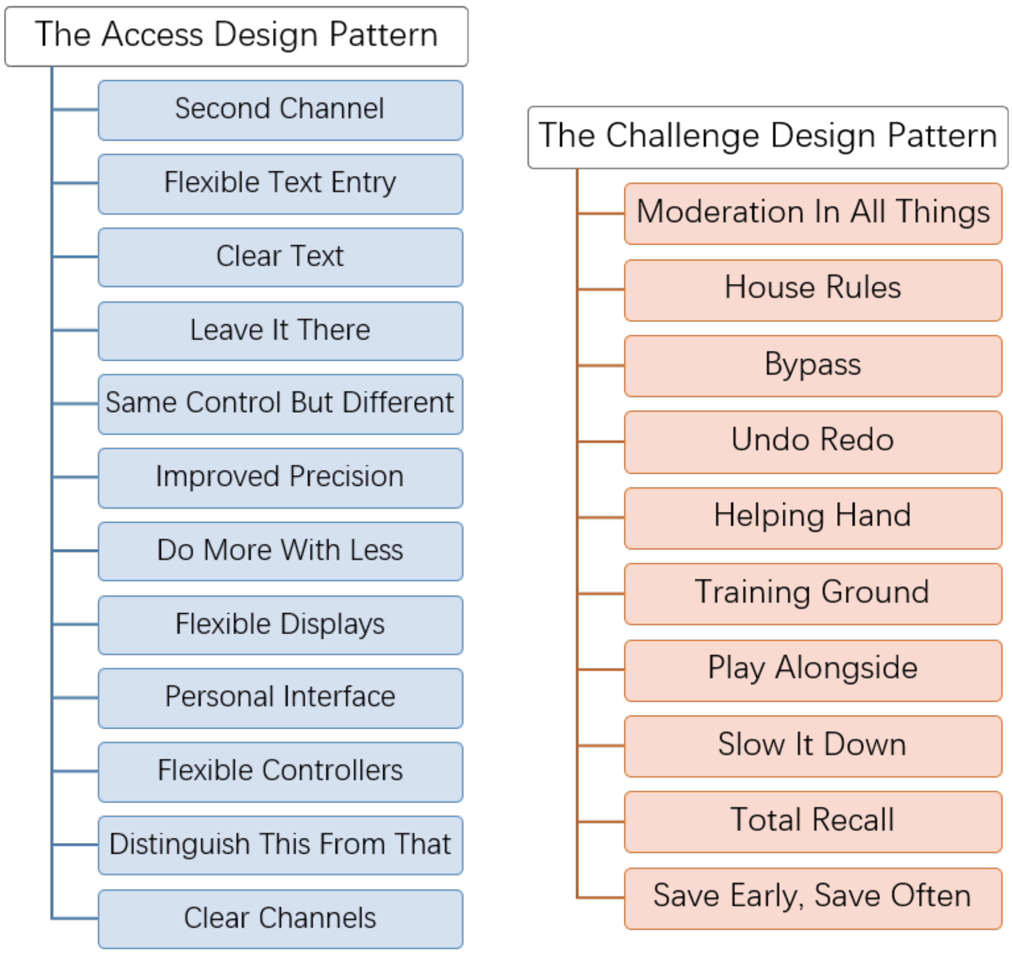
Game Accessibility Guidelines (GAG) [40] is a distilled and well-maintained living document, that provides thorough guidance for game developers to incorporate accessibility design into their projects. Like APX, GAG organizes its principles into three layers, but based on the difficulty level of implementing related features. However, GAG is an extensive document and may continue growing, only some principles will be explained in this subsection, for a complete set of vision-related principles see Appendix A.

Some worth-noticing items include:

- "Ensure no essential information is conveyed by a colour alone", which emphasizes the importance of using alternative identifiers beyond colour, for example, shape.



(a) The APX Triangle



(b) The Access Design Pattern

(c) The Challenge Design Pattern

Figure 2.4: Accessible Player Experiences (APX). Adapted from: [38]

- “Ensure interactive elements and virtual controls are large and well-spaced”, it can effectively reduce the occurrence of unintended and unwanted actions.
- “Give a clear indication that interactive elements are interactive”, this can be achieved by offering additional hints.
- “Provide a choice of cursor/crosshair colours/designs”, which helps distinguish the cursor from the background.
- “Ensure screen reader support, including menus and installers”.
- “Ensure screen reader support”, including menus and installers.
- “Simulate binaural recording” to help navigation and so on. For further details, see Section 3.1.2.

In addition to the above principles, also other important principles are mentioned in both GAG and APX. For example, “Use an easily readable default font size” and “Allow the font size to be adjusted” in GAG align with the concept of “Clear Text” from APX. Moreover, “Provide high contrast between text/UI and background” and “Provide an option to adjust the contrast” in GAG can be seen as a part of “Clear Channel” from APX.

Similar repetitions can also be observed in guidelines summarized in research papers. For instance, a study [41] focusing on low-vision people, particularly those with permanent central vision loss due to damaged macula, lists 25 principles that fit this demographic.

These repeated principles primarily focus on the text, cursor, and colour contrast customization, navigation, placement of interactive elements, and audio design. Notably, principles such as “Identify and describe the nearest objects using audio” can facilitate a navigation system aligning with the Clear Channel design pattern from APX.

The guidelines outlined above will serve as valuable inputs for designing features and functionalities for our assistive tool. Additionally, insights from prior research and existing solutions will be introduced in the next chapter, providing supplementary references for our design decision.

3 Relevant Research and Work

In this chapter, we examine concretely accessible games for additional insights into game accessibility. Relevant key terms are defined and explained in Section 3.1, before looking at audio games in Section 3.2 and games from prior research in Section 3.3.

3.1 Key Terms and Concepts

Key terms and concepts central to relevant research and this thesis are introduced to provide a foundational understanding of the subsequent sections. These concepts include sensory substitution, binaural sound, and sonification, each playing an important role in improving the accessibility of digital applications for visually impaired people.

3.1.1 Sensory Substitution

Sensory substitution is the capacity of the human brain to utilize one sensory modality as a compensatory mechanism for another. According to [42], in the case of people who are blind, while the peripheral sensory system may be compromised, the central visual mechanisms often remain intact. This means that although the optical image may not directly reach the retina as it does in a fully functioning visual system, alternative sensory inputs can effectively substitute and later be used to generate subjective images for blind and low-vision people.

Two sensory systems have been intensively researched in substituting vision loss, visual-to-auditory and visual-to-tactile substitution, which convey visual information through audition and tactile sensation respectively. Similarly, in the case of hearing loss, touch and vision can serve as compensations in previous research [43][44]. Notably, research [45] studies how colour can be used in conveying sentiments for people with hearing impairment, which leverages auditory-to-visual substitution.

It has been verified that sensory substitution can be effectively used in identifying objects, including the recognition of their visual forms [46] and spatial positions [47]. Thus, this theory has been widely used in application development for people with visual impairments. For instance, research [48] develops an online shopping system with a hapticaudio-enabled browser for visually impaired people to identify the product's form and dimension and feel the material through the tactile device. One of the most successful applications of sensory substitution is Braille, which non-sighted people use to acquire information through touch.

Sensory substitution also plays an important role in gaming. Two typical forms of visual-to-auditory substitution are semantically significant narration and non-verbal audio effects. Narration commonly aids in storytelling, scene navigation, and user interface navigation, while audio effects mainly serve as game cues and feedback for player actions. Audio games are a prime example of utilizing visual-to-auditory substitution to its limits; for more details, see Section 3.2.

Moreover, visual-to-tactile substitution was also widely adopted, particularly in console games which can be used with specific controllers or gamepads capable of sending and receiving tactile signals. For instance, Nintendo has published a series of motion-sensing games including *Nintendo Switch Sports* [49], *1-2-Switch* [50] and *Mario Tennis Aces* [51]. It is worth noticing that *1-2-Switch* [50] is fully accessible to non-sighted individuals.

3.1.2 Binaural Sound

Binaural sound is created through binaural recording to mimic the natural auditory experience of humans with two ears and it usually works with stereo headphones or loudspeakers [52]. By placing two microphones respectively alongside each ear canal of an artificial head, the binaural recording captures sounds from multidimensional sound fields [53]. Both the direction and distance of the sound source from the ears are recorded to precisely reproduce the audio.

The concept of binaural sound is massively used in games to create an immersive virtual environment, especially in audio games where spatial navigation and object orientation are conveyed through auditory cues. Research [54] develops an auditory version of an existing arcade game *Pong*, where the traditional graphical display was replaced with an audio-only loudspeaker display. Similarly, research [55] showcases an Android musical game leveraging an improved binaural sound technology for visually impaired people. Both studies mention the positive impact of binaural sound on visually impaired people's gaming experiences.

3.1.3 Sonification

Sonification refers to the process of converting non-audio data into non-speech audio. For example, some research sonify images to get an auditory representation of visual information, while some sonify objects to convey their visual and spatial information. Sonification can be used to improve accessibility for visually impaired people, helping them raise their spatial awareness and recognize surrounding items.

Most research in this domain focuses on sonifying the visual properties of images and objects. A notable case is the *vOICe* system, detailed in [56], which converts images into sound patterns by a specific methodology. Specifically speaking, it converts the vertical position and brightness of each pixel as the amplitude and pitch of an acoustic signal at the time represented by the horizontal position of the pixel. The conversion process is invertible, which ensures data stability.

A similar methodology was adopted in [57], where a visual sonification system named *MVSS* is designed. The system is designed to assist visually impaired people in distinguishing different items in their surroundings. It first extracts depth information from the scene to divide and localize conspicuous objects and subsequently convert the visual properties in the form of histograms to audio signals.

Although image sonification lies out of the focus of this thesis, the above research reveals the significance and limitations of sonification. On one hand, people can effectively build a mental map from acoustic signals, making sonification a potent asset in spatial navigation [58]. On the other hand, sonification intensively relies on the human brain to recognize and categorize audio, potentially causing cognitive overload [57].

In digital game fields, sonification is widely used in the design and assignment of distinct audio effects for different items, characters and events, especially in audio games. Research [59] presents the design process of sonifying. Additionally, two sonification techniques, tempo and loudness of sound, representing the distance between the player and the object are evaluated during the research. Testing results show that “players were more active when playing under the loudness condition compared to the tempo condition”.

In this thesis, sonification serves as a way to assign objects proper audio effects to represent the distance between the object and the observer.

3.2 Audio Games

Audio games are a genre of digital games, which give priority to auditory cues and feedback over images and vision, in contrast to standard video games [60]. As such, they are typical cases of sensory substitution.

Although audio games are primarily tailored for players with visual impairments, they have also gained popularity among players with normal vision. Visual fiction, adventure, and racing games are common genres of audio games, which will be covered in this section with concrete examples.

The *Red Riding Hood* [61] is an iOS interactive visual novel based on the classic fairy tale of the same name, where players can take on the role of Little Red Riding Hood or the wolf and make choices that lead to different endings. Designed to be inclusive, the game offers both graphical and auditory user interfaces, enabling accessibility for both sighted and non-sighted players. With professional voice acting, sound effects, environmental music and full support with *VoiceOver* – a screen reader for iOS and other Apple devices, non-sighted people can fully experience the whole storytelling.

Figure 3.1 demonstrates how this game is played with *VoiceOver*. A green highlight border is generated by iOS accessibility features to indicate the interactive element under focus, accompanied by voice prompts, providing details about the element's identity and content. Additionally, for some elements such as sliders, the voice prompts may include their status and available operations.

A Blind Legend [62] is an action-based adventure game where players play the role of a blind knight embarking on his journey to confront enemies with help from his daughter Louise. Unlike *Red Riding Hood*, *A Blind Legend* is audio-only and the game screen is shrouded in mist throughout the journey, though the colour of the mist may change at some points. Players can control their character to move in a certain direction, use weapons, and interact with other characters and objects, including enemies, horses and obstacles. All actions are executed through multipoint tactile gestures on touchscreens, or via joysticks and keyboards on PCs. The diverse actions provide a solid foundation for a well-balanced combat system, which challenges the stereotype that games for disabled people lack intrigue.

Additionally, Louise takes the role of a guide to aid navigation in the 3D game

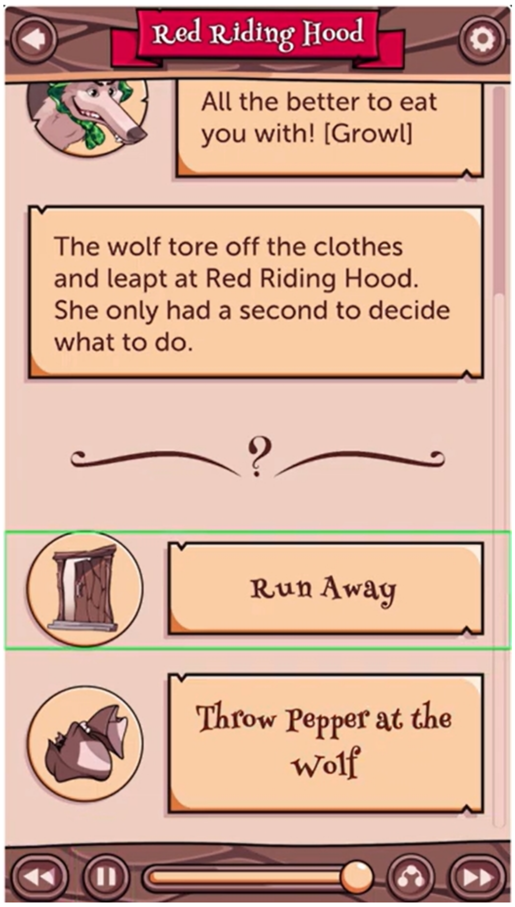


Figure 3.1: A screenshot of playing the game *Red Riding Hood* with VoiceOver. The green highlight border is generated by iOS accessibility features. Source: `https://www.youtube.com/watch?v=6K2hRPc92iI`

world. In each scene, the storyline progresses in a segmented manner in different spots, with Louise providing direction or hints upon reaching each spot, and then waiting for players to catch up. Voice prompts from a non-player character provide a unique immersion, meanwhile, voice acting made from binaural recording enhances enhances directional accuracy. However, an issue arises when players struggle to find their way, as Louise may repetitively deliver her lines, causing frustration and annoyance among some players.

Similar designs can be seen in another audio-based adventure game, *ENGK-WENTRO* [63], a mobile game for the Android system. Like *A Blind Legend*, it offers auditory navigation, a combat system, spatial audio and various character controls. In this game, a fairy serves as the guide, which also leads to the issue of constantly repeating dialogues. However, well-designed audio cues stand out in this game. For instance, the player's health is represented by the heartbeat sound effect, which accelerates as the player's health diminishes.

The issue of repetitive dialogue from navigation guides, as observed in both *A Blind Legend* and *ENGKWENTRO*, is effectively avoided in game *Sonar Island* [64], which adopts a unique method for navigation. *Sonar Island* is a multiplayer adventure game where players' main objective is to collect treasure chests hidden in various islands. In this game, players explore the environment using echolocation, specifically, players manually emit a sonar pulse and listen for echoes to get the direction information towards the target object. Thereby, the only potential distraction is ambient sounds, greatly reducing the overall difficulty level.

Despite being replaced by echolocation in navigation, voice prompts continue to serve as notifications in *Sonar Island.* For example, players' performance is evaluated based on the number of treasure chests they collect within a specified time limit for each difficulty level. Voice prompts are used to notify players about the remaining time for each level.

Sonar Island is not the only audio game that incorporates a time-constraint feature. *Drive* [65] is an audio-only racing game where players need to navigate a car through various courses and achieve high scores. It enables two distinct voices for different information cues. A female voice provides guidance on controls and operations, relaying information about time constraints and checkpoints, while a male voice serves as a stimulus, triggering whenever players can accelerate. Despite its compact size, this game offers a relatively complete audio gaming experience.

A similar yet more enriched game is *Feer*, an endless runner game where players move forward continuously along a road, as they do in *Drive.* Figure 3.2 shows the game scene, which is a 3D space comprising three lanes with various obstacles, enemies, and fairies approaching from the opposite direction. players need to switch between these lanes or take actions, such as sliding and jumping over, to avoid collisions.

Besides, each type of character and game object in *Feer* is sonified with unique sound effects, and binaural sounds are used to indicate which lane the character or object is on, enabling players to understand their encounters accurately and promptly. As the game progresses, the challenge escalates with the protagonist accelerates, providing a sustainable experience for players.

Figure 3.2: A screenshot of the audio game *Feer*, where a zombie is coming from the left lane.

Given the aforementioned examples, it is clear that the main focus of audio games is audio design, including providing unique audio effects for each unique game object. Furthermore, it is evident that a robust navigation system is critical for making audio games accessible [6], which can be also exemplified by research-based experimental games in the next section.

3.3 Experimental Games in Prior Research

This section contains accessible games from prior research. Unlike finalized audio games, these research-based games offer detailed explanations for their accessibility design decisions, providing information for our design process. The primary focus of this section is to examine the accessibility features adopted and important findings within.

Our results indicate that research on accessible games primarily focuses on two types of games: serious games and multiplayer. The former, such as educational games designed to improve math skills, is frequently used to increase specific abilities of the visually impaired and to raise social awareness. The latter is intended to be inclusive, enabling both visually impaired and sighted people to have fun together.

Study [66] introduces a roller coaster simulation game to help children learn Mechanical Energy, based on the hypothesis that blind people can understand the rail's shape and can perceive the simulator's haptic and auditory feedback. As seen in Figure 3.3, the primary screen is divided into a gameplay area and a control panel. While all menu items and game objects have narration provided through text-to-Speech (TTS) technology, some participants in usability testing still struggle to locate target interactive elements on the screen. This might occur because there are too many small interactive elements displayed simultaneously on the screen, which violates the accessibility principle that requires interactive controls to be large and well-spaced.

In another study targeting children as the primary users [67], two educational games were developed. One focuses on geography knowledge while the other enhances calculation skills. Both games aim to improve visually impaired users' memory capability and executive function. As both games heavily rely on voice prompts, an interesting finding from testing is that participants tend to hear different voices, which gives them the impression that they are speaking with new people.

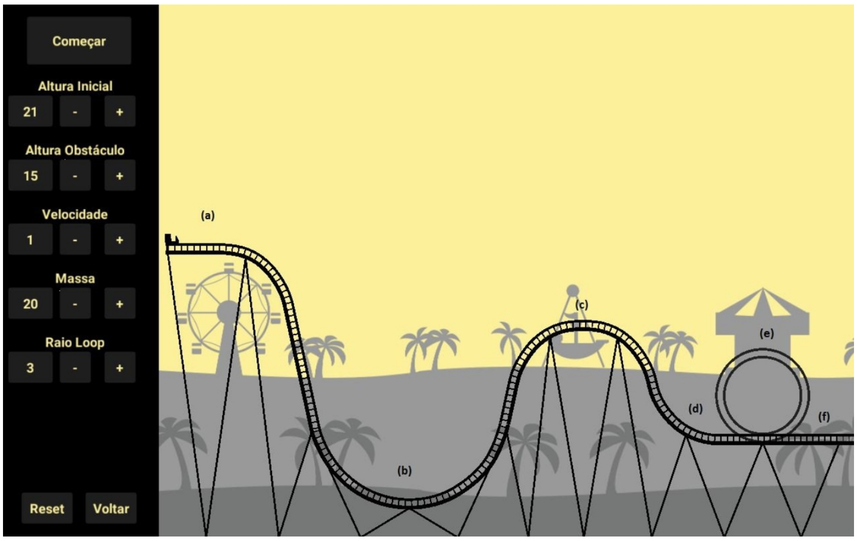


Figure 3.3: The main screen of a roller coaster simulator, with tightly spaced UI elements. Source: [66].

Beyond the educational domain, a great deal of research focuses on helping visually impaired people develop essential skills for their daily lives, such as navigation and spatial perception.

The audio-based computer game *Legend of Iris* [68] is designed to assist blind children in training their audio-based navigation skills. It utilizes the virtual reality device Oculus Rift to track players' orientation while immersing them in a well-crafted fantasy storyline. One of its standout features is the creation of an accurate and realistic spatial soundscape.

Research [69] has a similar goal where a game named *In.line* is developed, based on a navigation system that features indoor path recognition with augmented reality. This game utilizes haptic feedback to guide players along the correct route, with vibrations corresponding to their position and direction. One interesting finding about this game is that all testing participants would like to gain more information about their surrounding environment.

Navigation stands as one of the paramount functionalities in both the real world and the game world, thus, it is not surprising that countless accessible games use various strategies to facilitate navigation features.

Research [70], focuses on developing user interfaces for serious games that suit non-sighted people, and proposes three different solutions. Of particular relevance to this thesis are:

1. A two-level cyclic navigation system using arrow keys, as displayed in Figure 3.4, players can browse all interactive elements in order in the first level, and interact with them in the second level upon pressing the enter key. This is similar to how users with screen readers navigate and interact with web page elements.

2. A sonar system using a mouse, with each interactive element featuring a unique sound effect. The frequency and pitch represent the distance between the mouse cursor and the target element, with pitch specifically indicating information about the vertical distance.

Figure 3.4: A two-level cyclic navigation system using arrow keys in an educational game. Source: [70].

Testing results indicate that the navigation system outperforms the sonar system in terms of usability, but lags in terms of engagement, which reflects personal fulfillment and enjoyment.

A similar sonar system with slight variations is observed in research [71], where a 2D action role-playing game is developed. Within this system, lateral differences and changes in sound pressure are utilized to indicate the horizontal and vertical coordinates of the cursor.

Multiplayer games are also a focus of research on inclusive and accessible games. Research [72] introduces a project comprising multiple mini-games for both sighted and non-sighted people. These touchscreen-based mini-games use a maximum of six gestures for action inputs, all while incorporating auditory and haptic feedback for accessibility design.

In another study focusing on accessible multiplayer games, researchers present a mobile card game based on the classic *Chase the Ace* rule [73]. They emphasize the importance of prioritizing enjoyable experiences over mere task completion, particularly for visually impaired players who face inherent challenges for the latter.

Building upon insights drawn from existing work and prior research, we will shift our focus to the design and implementation of an assistive tool in the next chapter.

4 Design and Development

This chapter offers answers to RQ1 and RQ2, introduced in Section 1.2, by constructing a prototype of the assistive tool with HTML5 games. We start with a requirement analysis through writing user stories in Section 4.1, before looking at the project architecture in Section 4.2. Section 4.3 explains the adopted UI/UX design, while Section 4.4 elaborates on implementations of key features.

4.1 Requirement Analysis

In software development, whether following a traditional process such as the Waterfall model, or modern Agile methodology, requirement analysis serves as the foundation of a project. The use case, use scenario and user story are common approaches to reflect user needs during requirement analysis, guiding system design and practical development [74]. User stories, in particular, offer a simplified and straightforward approach to capturing requirements. Therefore, in this section, the user story is chosen to describe the requirements for the assistive tool.

Before extracting user stories, a range of accessibility guidelines from Section 2.2.3 and insights gained from prior work in Chapter 3 are selected based on the following two criteria:

1. Relevance: the underlying accessibility feature must accommodate the needs of visually impaired people.

2. Applicability: it applies to click-and-point adventure games for a single player.

Eighteen user stories are created to describe users' expectations for a compact assistive tool for games, as listed in Table 4.1.

Table 4.1: 18 user stories for the assistive tool.

ID	**User Story**	**Reference**
US1	As a low-vision player, I want to be able to adjust the text font size in games so that it is comfortable for me to read.	APX: Clear Text [38], GAG [40]
US2	As a low-vision player, I want to customize the game text style so that I can choose a font style that is easier for me to read.	APX: Clear Text [38], GAG [40]
US3	As a low-vision player, I want to be able to change the size of the cursor in games so that it is easier for me to track.	GAG [40]
US4	As a visually impaired player, I want to be able to change the colour of the cursor so that I can make it stand out against the game background.	APX: Distinguish This from That [38]
US5	As a game player, I want to be able to activate a magnifier tool so that I can zoom in on specific areas in games for better examination.	GAG [40]
US6	As a visually impaired user, I want to have voice prompts for all game content so that I can understand what I am interacting with without relying on vision.	APX: Second Channel [38], GAG [40]

ID (Contd.)	**User Story (Contd.)**	**Reference (Contd.)**
US7	As a visually impaired player, I want to use my screen reader seamlessly with games so that I can listen to text-based content in the game.	GAG [40]
US8	As a blind user, I want to have keyboard shortcuts for navigation in both games and the assistive tool so that I can quickly find interactive elements.	GAG [40]
US9	As a visually impaired player, I want to highlight interactive elements in games so that I can clearly distinguish important game elements.	GAG [40]
US10	As a visually impaired player, I want important game elements to emit unique sounds so that I can use my hearing to locate them.	GAG [40], Sonar Island [64]
US11	As a visually impaired player, I want to be able to freely change the layout of the user interface of the assistive tool, so that I can rearrange the user interface elements in a way that suits my needs and preferences.	APX: Personal Interface [38], GAG [40]
US12	As a colour-blind player, I want to have customizable colour schemes for the user interface and game content, so that I can adjust the colours to ones that are distinguishable for me.	APX: Distinguish This from That [38], GAG [40]
US13	As a visually impaired player, I want to be able to adjust the volume separately for audio effects, speech and background music so that I can customize them according to my preferences.	GAG [40]

ID (Contd.)	**User Story (Contd.)**	**Reference (Contd.)**
US14	As a visually impaired player, I want to be able to control the animation speed and even remove them so that I would not be distracted by them.	GAG [40]
US15	As a game player, I want to be able to redo and undo my actions in games so that I can explore games without fear of irreversible errors.	APX: Redo Undo [38]
US16	As a game player, I want to be able to select the difficulty level of games so that I can choose the challenge according to my skill level.	GAG [40]
US17	As a game player, I want to skip some operations or levels so that I can progress through the game without getting stuck on some sections.	APX: Bypass [38]
US18	As a game player, I want to have autosave before important plots and to be able to save the game often, so that I can resume from some points without losing progress.	APX: Save Early, Save Often [38]

4.1.1 Feasibility Assessment

The Feasibility Assessment is conducted to determine whether the identified requirements are technically and operationally feasible, thereby addressing "RQ1: What assistive technology features can be integrated into HTML5 games while preserving the integrity and originality of the games?"

To answer this question, an extensive examination of HTML5 games sourced primarily from two free online game websites, itch.io [75] and html5games.com [76], was carried out, along with an evaluation of browser add-ons having similar functions. The preliminary assessment indicates that:

- Accessibility features described by US15 - US18 are not technically possible for a universal assistive tool. The therein features such as redo and undo, and save game progress, require knowledge of game-specific statistics and parameters, which violates the tool's objective to accommodate multiple games instead of tailoring for one.

- On the other hand, features described by US1 - US14 appear to be technically possible. These features either find parallels in existing browser add-ons or have been validated through rapid prototyping.

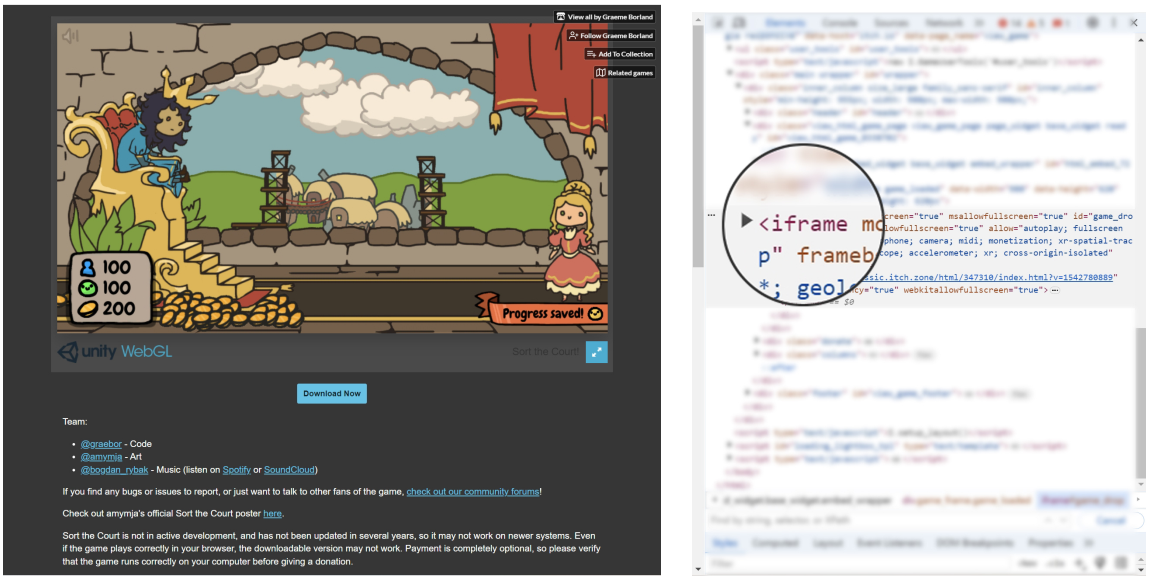
The assessment is supported by subsequent practical implementation. However, due to time constraints, only features described by US1 - US11 are incorporated into this implementation, while, whether US11 - US14 can be achieved in the assistive tool requires further validation.

To summarize the answer to RQ1, accessibility features described by US1 - US11 can be integrated into HTML5 games through external assistive tools without affecting the original games. These features include font and cursor customization, magnifier, keyboard shortcuts, narration, sonification, highlighting for interactive elements and compatibility with screen readers.

4.1.2 Challenges and Limitations

Despite the widely acknowledged web accessibility guidelines and the availability of various assistive technologies for browser activities, a universal assistive tool for HTML5 games is still noticeably absent. The underlying cause of such a phenomenon is closely linked to the technical challenges presented by HTML5 games.

Before discussing the technical challenges, a basic explanation of HTML5 games is indispensable. HTML5 games are digital games developed utilizing modern web technology, including HTML, Javascript, CSS and sometimes webGL [77]. HTML5 games can be standalone applications that run directly through a web browser. Nevertheless, it is more common that they are packaged and embedded into other webpages using the `<iframe>` element, as can be seen in Figure 4.1.



(a) Webpage hosting the game

(b) The source code with the `<iframe>` element

Figure 4.1: A HTML5 game embeded in a game hosting website through the `<iframe>` element. Source: [78]

However, the Same-Origin Policy (SOP) [79] presents a barrier. It prevents scripts in the webpage from directly accessing the content embedded within `<iframe>` elements. This restriction makes it more difficult to provide accessibility support for games embedded in `<iframe>` elements. One workaround is to open the embedded HTML5 game in a new browser tab, which can be done through scripting.

Furthermore, the diverse development approaches for HTML5 games remain a challenge for building a universal tool. Beyond pure Javascript, there is a wide range of Javascript frameworks and libraries used in creating HTML5 games, for example, PixiJS is used in building the indie game *Witch Hat and Ears of Cat* [80]. While these frameworks reduce the scripting workload, they can add difficulty for external scripting to understand how the game works. Besides, game engines like Unity, Twine, RenPy, and RPG Maker are widely used for making HTML5 games. They encrypt the source code and assets in various ways, which not only enhances security but also adds layers of complexity.

Additionally, modern graphics rendering technologies such as the Canvas API and WebGL are prevalent in games, offering advanced graphics and visuals [81]. Modifying graphics rendered by these technologies requires proficiency in computer graphics, which may pose challenges. Due to time constraints, this thesis bypasses this challenge in the initial prototyping, by focusing on pure HTML5 games without advanced graphics rendering.

4.2 Project Architecture

The following three sections address "RQ2: How to implement such a tool?" by explaining the construction of the assistive tool from various aspects. This section primarily deals with the project architecture.

To develop an assistive tool that can add accessibility features to multiple HTML5 games, the most intuitive solution is a browser add-on. The following prototyping is based on the Chrome extension, chosen for its comprehensive documentation and active community support.

The most essential component for a Chrome extension is the `manifest.json` file, which defines metadata for the extension, including name, description, the default popup and many other critical settings [82]. This thesis adopts the manifest v3 framework, in which a service worker script runs the background code, while content scripts run in the context of target web pages. Besides, The default popup URL defined in the manifest file directs to an HTML file, complemented by referenced

CSS and Javascript files, which together define a complete user interface.

Figure 4.2 illustrates the relationships between the main parts of a Chrome extension under the manifest v3 framework. The manifest file references the default popup, the service worker script, and the content scripts. The service worker script injects content scripts into the target web page, and their communications are done through the messaging system.

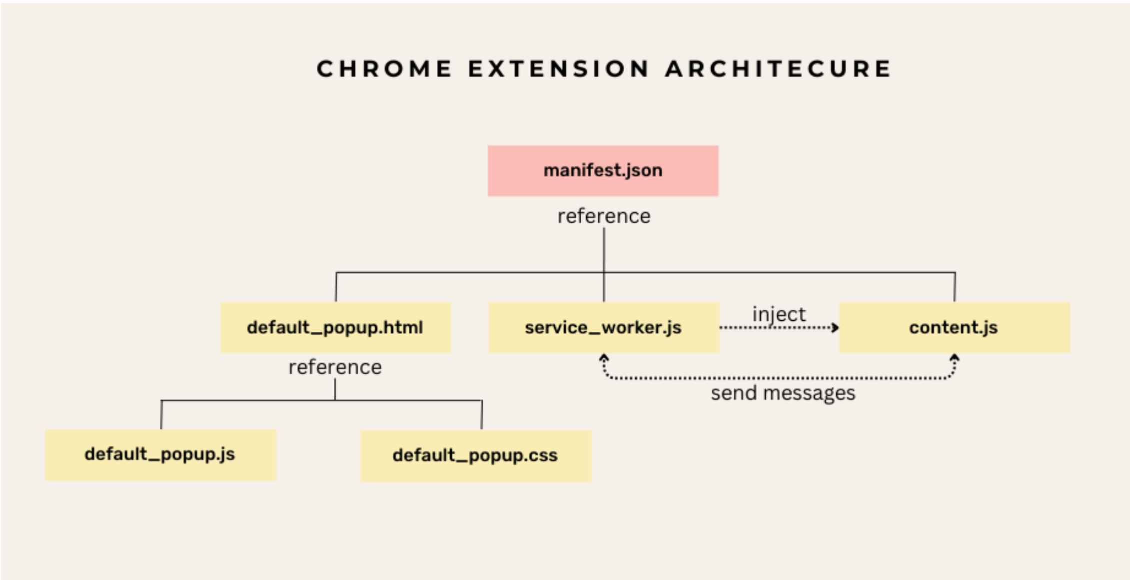


Figure 4.2: The primary components of a chrome extension and their relationships.

However, the default popup that appears next to the extension icon in the toolbar is fixed in position. To achieve a more flexible user interface, permitting dragging and dropping anywhere within a webpage, the default popup is out of use in this thesis. Instead, a popup will be generated through the content script once users click on the default extension icon.

After adopting this new mechanism, content scripts now act as the intermediary between the background script and the popup, which is the user interface. To explain the interaction between users and the three main components of the extension, Figure 4.3 depicts a sequence diagram detailing how a user opens the extension to

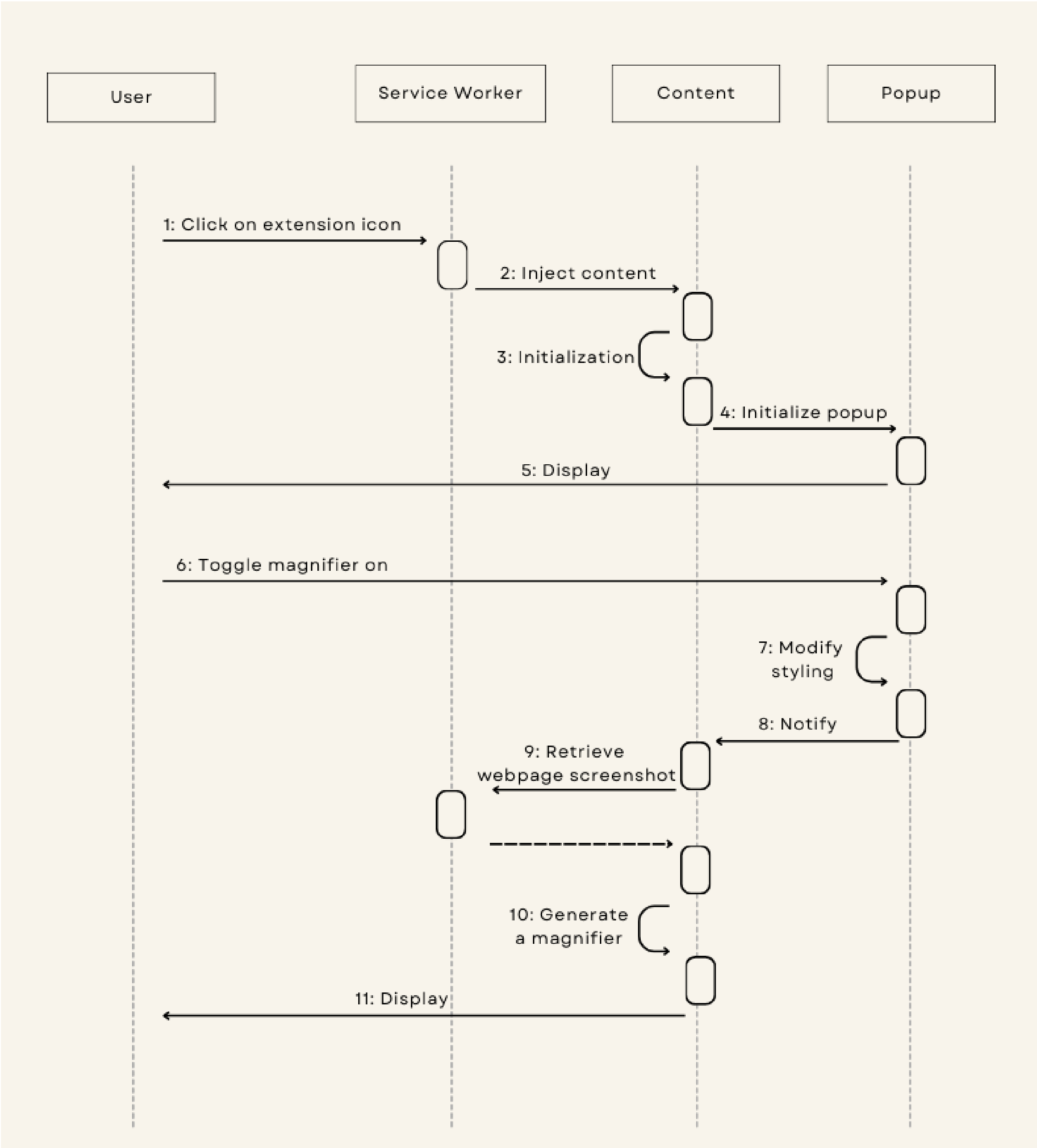


Figure 4.3: A sequence diagram illustrating the interactions between users and the Chrome extension components.

activate the magnifying glass feature.

The process begins when the user clicks on the extension icon, triggering the service worker to inject content scripts into the target webpage. Upon initialization, the content script generates and initializes the popup, displaying it to the user. To activate the magnifier, the user toggles it on via the popup, which then notifies the content scripts after changing its styling to align with the current status. Upon receiving the message, the content script generates a magnifier placeholder, retrieves a screenshot of the current webpage, and displays them together to the user.

4.3 UI/UX Design

In this section, we focus on the user interface (UI) and user experience (UX) aspects of implementing the assistive tool as a Chrome extension. UI/UX design is crucial in digital products, especially for individuals with special needs, as it directly impacts usability and accessibility [83].

As previously mentioned, the popup serves as the primary user interface in this assistive tool, comprising HTML, CSS, and Javascript files. HTML and CSS are utilized for the construction and styling of the UI, while Javascript handles user interactions. However, details of the UI/UX implementation are beyond the primary focus of this thesis, only the design decisions will be covered in this section. A full version of the project can be found in Appendix B.

The UI is designed to resemble a sticky note, allowing users to attach it to any part of a webpage, as can be seen in Figure 4.4. This design choice caters to people with tunnel vision or blind spots, enabling them to position the UI in their optimal viewing spot. Besides, to ensure clarity and ease of use, only the header of the UI panel is enabled for drag action. When users hover their cursor over it, the cursor style will change to the classic Move style.

As for the main body of the popup, as shown in Figure 4.4, a series of widgets are vertically arranged within the main body. Each widget corresponds to a specific accessibility feature, and their arrangement makes it easy to expand more features. Notably, all widgets are designed to be focusable and navigable via key-

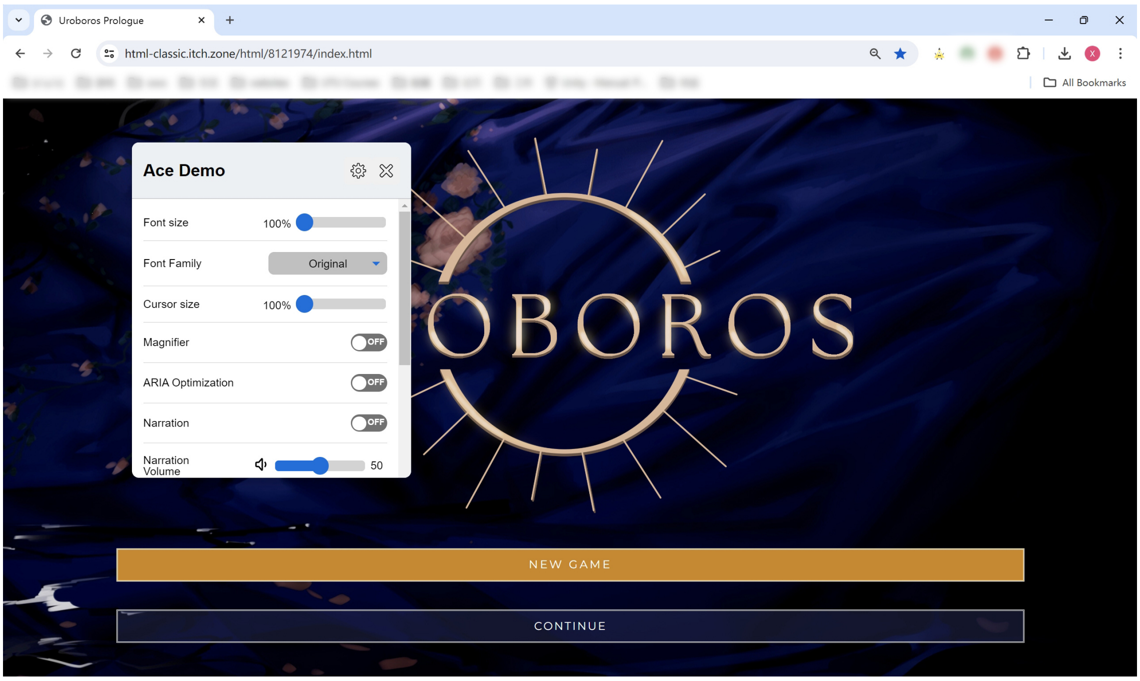


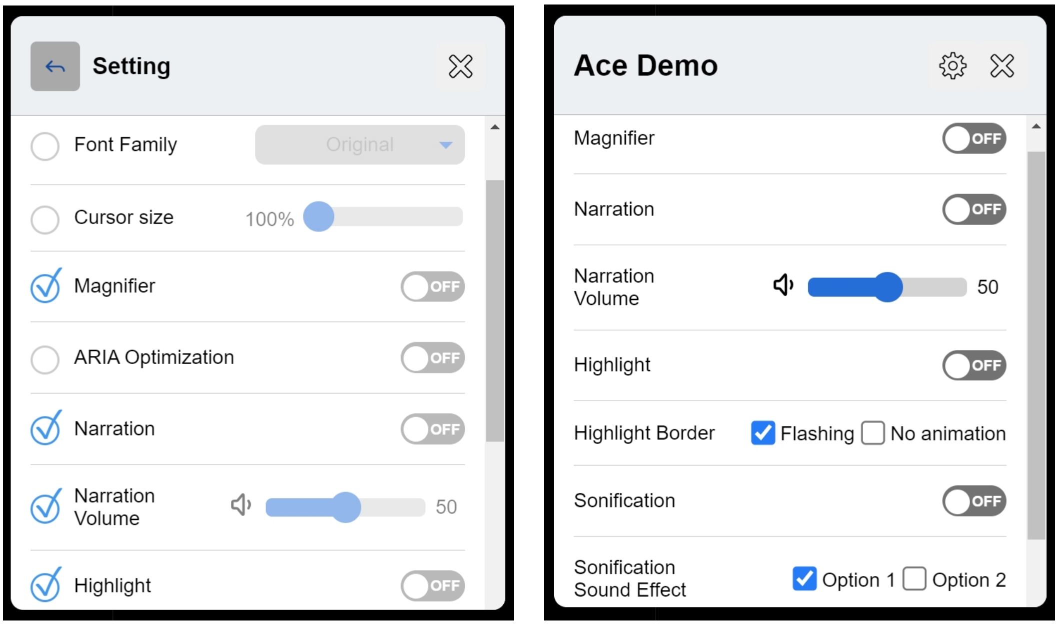
Figure 4.4: Using the assistive tool with an HTML5 game, the panel can be dragged and dropped at anywhere of the screen.

board shortcuts, aligning with the browsing preferences of individuals with visual impairments.

Additionally, all these widgets can be hidden from the panel in the setting mode. Figure 4.5a demonstrates the setting mode where each widget is equipped with a toggle to its left, through which the feature can be toggled off in the main panel as shown in Figure 4.5b. This functionality effectively prevents a situation where there are too many widgets inside the panel, causing frustration for users to locate the one they need.

4.4 Feature Implementation

This section provides an overview of key features implemented in the assistive tool. Relevant source code snippets are included to facilitate understanding.



(a) The setting mode (b) Hiding some widgets from the UI panel

Figure 4.5: The UI of the assistive tool.

4.4.1 Font Customization

In HTML5 games, text is usually enclosed within specific HTML elements like `<p>` and `<span>`. By adjusting the values of CSS properties assigned to these elements, the appearance of the text can be easily modified. The 'font-size', 'font-family', and 'font' are three key properties that influence the font size and font family of text, as shown in Listing 1. Notably, the 'font' property serves as a shorthand for multiple font-related properties, which is why it cannot be placed in the same CSS rule with other font-related properties.

CSS properties can be assigned to HTML elements through three ways, as demonstrated in Listing 2:

1. Inline CSS, which is assigned directly to an HTML element using the 'style' attribute.

2. Internal CSS, which is defined within the `<style>` element of an HTML document and associated with HTML elements using CSS selectors.

Listing 1 Three font-related CSS properties: 'font-size', 'font-family' and 'font'.

```
{css}
body{
    font-size: 18px;
    font-family: "Arial", sans-serif;
}
p{
   font: bold 18px "Arial", sans-serif;
}
```

3. External CSS, which is defined in another file and referenced through the `<link>` element.

Thus, to implement the accessibility feature that alters all the fonts inside an HTML5 game, one needs to make sure all font-related CSS rules are handled.

In our solution, the initial step is parsing the game source code to gather all font-related CSS rules, which is done during the content script initialization phase. Next, regular expressions are used to remove font-irrelevant part from internal and external CSS rules, before they are duplicated and injected back to the game. Notably, these rules are placed within a generated `<style>` block at the end of the HTML `<head>` section, to ensure their styling rules takes precedence over the original ones. Besides, inline CSS rules, which possess the highest precedence, are not duplicated, instead, references to HTML elements associated with them are stored.

Once preparations are done, users can interact with the popup to adjust font appearance. A message containing the desired outcome is sent from the popup to the content script. Eventually, the content script overrides the duplicated CSS rules and the referenced inline CSS rules to achieve the intended changes.

Listing 2 Three types of CSS: Inline CSS, internal CSS and external CSS.

```
{html}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" type="text/css" href="externalStyleSheet.css">
    <style>
        #internalCSS{
            font-size: 12px;
        }
    </style>
</head>
<body>
    <p style="font-size: 18px;">Inline CSS.</p>
    <p id="internalCSS">Internal CSS.</p>
</body>
</html>
```

4.4.2 Cursor Customization

Typically, the appearance and size of cursors are controlled by the user's operating system, which can not be adjusted directly through web technology. However, a compromise workaround is to override the default cursor with custom cursor images, consisting of various predefined appearances and sizes. Listing 3 exemplifies how to customize the cursor style inside the `<body>` element, in which the custom cursor image is located within the same directory as the code file.

Listing 3 A custom cursor.

```
{css}

body {

    cursor: url('custom_cursor.png'), auto;

}
```

When it comes to modifying cursor style in HTML5 games through a Chrome extension, it is apparent that the custom cursor images should be provided by the extension. Therefore, the URL of these images needs to be retrieved through the 'chrome.runtime.getURL' API.

Our approach to modifying cursor size follows a similar pattern to adjusting font styles. During initialization, cursor-related CSS rules are duplicated or marked, so that they are modified when users request. It is worth noting that games sometimes customize cursor styles to convey specific information, such as indicating interactable characters. However, due to SOP [79], which restricts access to resources from other origins. Thereby, game-customized cursors are not addressed by this solution. Listing 4 shows how only default and pointer cursor style are recognized using regular expression given a set of cursor-related CSS rules.

Listing 4 Using regular expression to extract default and pointer cursor styles.

```
{jsvascript}

const regexCursor = /(?<=cursor\s*:\s*)pointer|default(?=\s*[;|}])/g;

const rule = "p{cursor:default;}" +

             "h1{cursor:url(images/custom.png), auto;}";

const match = rule.match(regexCursor);

if(match !== null){

    console.log(match); // output:['default']

}
```

4.4.3 Magnifier

A typical approach of creating a magnifier effect in HTML5 games consists of 2 key steps. Firstly, capturing a screenshot of the current screen. Secondly, displaying a magnified view of a specific area of the screenshot within a magnifying glass element based on the cursor's position on the screen.

There are several libraries available for capturing screenshots of HTML documents, such as *html2canvas* [84] and *dom-to-image* [85]. Both libraries can convert HTML elements into common image formats or HTML `<canvas>` elements, which can directly display an image. However, since we are developing a Chrome extension, we can utilize the Chrome API 'chrome.tabs.captureVisibleTab' to capture a screenshot, as shown in Listing 5.

Listing 5 Use Chrome APIs to capture a screenshot and get screen zoom factor.

```
{jsvascript}
chrome.tabs.captureVisibleTab({ format: "png" },
    async function (image_url)
    {
        chrome.tabs.getZoom(sender.tab.id,  function (zoomFactor) {
            chrome.tabs.sendMessage(sender.tab.id, {
                code: "GET_CAPTURED_TAB_RESPONSE",
                data_url: image_url, zoom: zoomFactor,
            });
        });
    });
```

As for displaying the screenshot within a magnifying glass element, one should first set the screenshot as the background image with magnified dimensions, then register a 'mousemove' event listener to dynamically adjust the portion of the background image being displayed, as shown in Listing 6.

Listing 6 Set and display the background image of a magnifying glass element.

```
{jsvascript}
const power = 2;
const glass = document.createElement("div");
glass.style.backgroundSize =
    `${window.innerWidth * power}px ${window.innerHeight * power}px`;
document.addEventListener("mousemove", function (event){
    let x = -1 * (event.clientX) * power + glass.offsetWidth / 2;
    let y = -1 * (event.clientY) * power + glass.offsetHeight / 2;
    glass.style.backgroundPosition = x + "px " + y + "px";
});
```

4.4.4 Narration

Voice narration is indispensable for visually impaired people to perceive and understand the game content. Screen readers such as NVDA [86] and Jaws [87], a common assistive technology for digital devices, are widely used by visually impaired people to narrate on-screen content. However, many HTML5 games lack support for screen readers, making games inaccessible. This subsection proposes an approach to optimize HTML5 games for screen reader users.

Additionally, in consideration of players without screen readers, Text-to-Speech (TTS) functionality is also implemented in this assistive tool to offer similar features.

Use with Screen Reader

To ensure screen readers can recognize and differentiate game elements, such as text, images, and interactive elements, it is essential to assign Accessible Rich Internet Applications (ARIA) [88] attributes and values to necessary elements. ARIA attributes provide additional information for assistive technologies to understand the layout of the webpage and the role of its elements. Assigning proper ARIA attributes is the key to this accessibility feature.

Generally, HTML5 games without proper ARIA support pose two main issues:

1. The screen reader fails to recognize content changes, for example, loading another game scene.

2. The screen reader cannot identify the purpose and role of interactive elements.

The first issue can be solved by setting the ‘role’ attribute or ‘aria-live’ attribute of the root element of an HTML game as ‘status’. This marks the whole game content as a live region, which requires announcements when updates occur.

The second issue requires proper assignments of values to the ‘aria-label’ attributes. During practical implementation, a few worth-noticing problems emerge:

- Firstly, some buttons considered interactable by the screen reader may not be practically interactable, due to factors like being outside the viewport or hidden behind other HTML elements. Thus, a comprehensive filtering mechanism is needed to identify and exclude such elements from consideration. Besides, the ‘aria-hidden’ attribute is necessary to conceal these elements from the screen reader.

- Secondly, game content changes may introduce new interactive elements or cause previously visible elements to become invisible. To tackle this, the ‘MutationObserver’ class can be used to monitor changes and take action accordingly.

- Thirdly, generating semantically meaningful descriptions for the ‘aria-label’ attribute of an HTML element remains a challenge. The adopted solution by this thesis is to extract content from game developer annotations, such as explanatory names for elements. However, this approach presents limitations especially when annotations are ambiguous or even absent.

Despite the limitations, this solution can positively improve the gaming experience of screen reader users.

Text to Speech (TTS)

Similar to the previous section, this approach filters interactive elements and extracts semantic descriptions. However, instead of labelling HTML elements with extracted descriptions, TTS technology is directly used for narration.

The Web Speech API [89] provides a 'SpeechSynthesis' interface, which can be utilized to narrate text-based content. Listing 7 shows a 'narrate' function, which accepts a string type parameter as input and uses the interface to narrate it accordingly.

Listing 7 A 'narrate' function utilizing the Web Speech API.

```
{jsvascript}

function narrate(text){

    const utterance = new SpeechSynthesisUtterance();

    utterance.text = text;

    utterance.lang = 'en-US';

    utterance.volume = 1;

    speechSynthesis.speak(utterance);

}
```

4.4.5 Interactive Elements Highlighting

The 'border' property in CSS can be used to define the size and style of the border of an HTML element. To prevent modifying the game elements directly, a transparent HTML element with an identical size, featuring only the border, is generated for each interactive element. The highlight border element is positioned over each interactive element in the same location.

Listing 8 shows the CSS rules for the highlight border, with the 'point-events' attribute set to 'none' to ensure interactions are not blocked. Listing 9 shows how to create a highlight border given an interactive element.

Listing 8 CSS rules for highlight border element.

```
{css}
.ace_demo_highlight {
    border: 4px dashed greenyellow;!important;
    position: absolute;!important;
    pointer-events: none;!important;
}
```

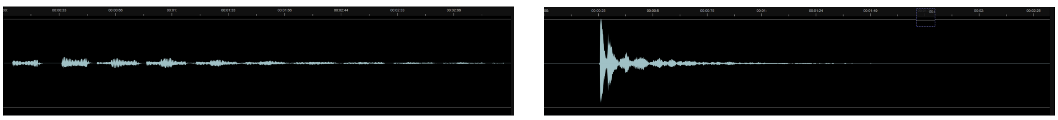
4.4.6 Sonification

As defined in Section 3.1.3, sonification involves assigning unique sound effects to game objects to help players locate them. In this assistive tool, the sonification feature is used for players to locate interactive elements. The sonification mode is separated from the regular gaming mode, requiring players to toggle the sonification feature to enter this mode. To ensure a clear auditory channel, only one element will emit sound effects at a time.

Two audio clips can be selected as the sound effect emitted by interactive elements, one with fluctuating decay and the other with sharp decay, as demonstrated

Listing 9 Create highlight border for an interactive element.

```
{javascript}

function generateHighlight(interactiveElement){

    const highlight = document.createElement('div');

    highlight.classList.add('ace_demo_highlight');

    // set the height and width of the highlight border

    setHighlightRect(interactiveElement, highlight);

    // getNumericalZIndex: get the z-index of the interactive element

    highlight.style.zIndex =

        getNumericalZIndex(interactiveElement).toString();

    interactiveElement.parentNode

        .insertBefore(highlight, interactiveElement);

}
```



(a) Audio clip with fluctuating decay (b) Audio clip with sharp decay

Figure 4.6: Two audio options for the sonification feature.

in Figure 4.6.

Binaural effects are implemented for both audio clips using the stereo panner functionality provided by the Web Audio API [90]. Listing 10 shows the creation of a panner and how to play the connected audio only in the left channel. The binaural sound effect is used to indicate the relative location of the cursor with the target interactive elements. For instance, if the target element is positioned to the left of the cursor, players will only hear the audio from the left channel.

Furthermore, to indicate the distance between the cursor and the interactive elements, both tempo-based and volume-based strategies are implemented. As a

Listing 10 Create a stereo panner for playing audio

```
{javascript}

const radarAudio = new Audio(chrome.runtime.getURL('audios/radar.mp3'));

const audioContext = new AudioContext();

const panner = audioContext.createStereoPanner();

const source = audioContext.createMediaElementSource(radarAudio);

source.connect(panner);

panner.connect(audioContext.destination);

panner.pan.value = -1; // left ear only
```

result, the volume or tempo of the audio increases as the cursor moves closer to the target. Players can choose one strategy or both.

These options for sound effects and sonification strategies provide a possibility for exploring users' preferences in the testing phase, which will be further discussed in the next chapter.

5 Testing and Evaluation

In the previous chapter, a detailed implementation of an assistive tool for HTML5 games was presented, with features derived from existing game accessibility guidelines and accessible games. A prototype of the assistive tool serves as the test subject in the testing phase described in this chapter.

The following sections begin with an introduction to the testing methodology in Section 5.1, including the usability testing design, a questionnaire design, and the data collection method. Then, an examination and analysis of the testing outcomes are given in Section 5.2. The answer to RQ3 is discussed in Section 5.2.5. Finally, in Section 5.3, we review all research questions and summarize the responses to them.

5.1 Methodology

Usability testing is a widely used method for assessing the ease of use of digital products. During the testing, participants are usually asked to interact with the product and offer feedback accordingly. Meanwhile, researchers can observe participants' behaviour and record any noteworthy observations or encountered issues. Usability testing provides valuable insights with real-world use experience and can even uncover issues that may not be apparent during development.

In this testing, both exploratory testing and task-based usability testing are adopted. During exploratory testing, participants are granted the freedom to explore the assistive tool with the provided HTML5 game in a natural and instinctive way

[91]. This approach allows us to observe if users would use the designed accessibility feature in our intended way. It also gives participants some time to become familiar with the assistive tool, before they are asked to carry out specific tasks in the subsequent task-based testing.

While exploratory testing is designed to evaluate most accessibility features, task-based testing was tailored to assess the sonification feature. Participants are asked to complete two tasks of increasing difficulty levels, allowing us to measure the effectiveness and users' satisfaction with the feature. Notably, participants with normal vision are blindfolded, to simulate the experience of visually impaired users who can only rely on auditory cues.

In addition to usability testing, participants are asked to complete a questionnaire to provide their feedback. The question design draws insights from the Questionnaire Template for Game Testing by ENI CBC Med [92] and Nelson's 10 Usability Heuristics [93]. The questionnaire is distributed via Google Forms, and answers are collected anonymously. For a full version, see Appendix C. It is noteworthy that the term 'HTML5 games' might be unfamiliar to those without a technical background, so it has been substituted with 'web games' in the questionnaire. However, it is necessary to clarify that both terms typically refer to the same concept in this thesis.

The questionnaire is divided into five parts. The initial two parts gather information about participants' demographics, browser usage habits, digital gaming experience, and familiarity with assistive technologies. This information can provide us with context for interpreting participants' behaviours during usability testing. Both open-ended and closed-ended questions are included in these two parts to ensure the depth of information collected. Besides, some questions are designed as conditional, allowing participants to skip them if certain criteria are not met, thereby simplifying the process.

The third and fourth parts of the questionnaire ask about participants' feelings and reflections on exploratory testing and task-based usability testing, respectively. The Likert scale method is used for designing questions in these two parts, to collect quantitative statistics of users' satisfaction and preferences.

The fifth and final part of the questionnaire consists of open questions that collect users' thoughts and future expectations towards this assistive tool. This section allows researchers to gather qualitative feedback and suggestions from participants, which are invaluable for the future development of such an assistive tool.

Testing participants are recruited through my local contacts at the University of Turku, which enables in-person support and on-site observation during the testing process.

5.2 Result Analysis

This section presents the testing outcome and a brief analysis thereof, including participants' background and prior experience, two usability testings and participants' feedback.

5.2.1 Demographic and Background Information

A total of nine participants were involved in this testing. All participants are from 22 to 33 years old as shown in Figure 5.1, of which 56% are male and 44 % are female.

Two-thirds of the participants have slight visual impairments or conditions affecting their vision, which include nearsightedness, farsightedness, and astigmatism. The rest of the participants are fully sighted.

We can see that the testing result lacks the involvement of non-sighted people, which can be attributed to the limited recruitment process. Although all respondents

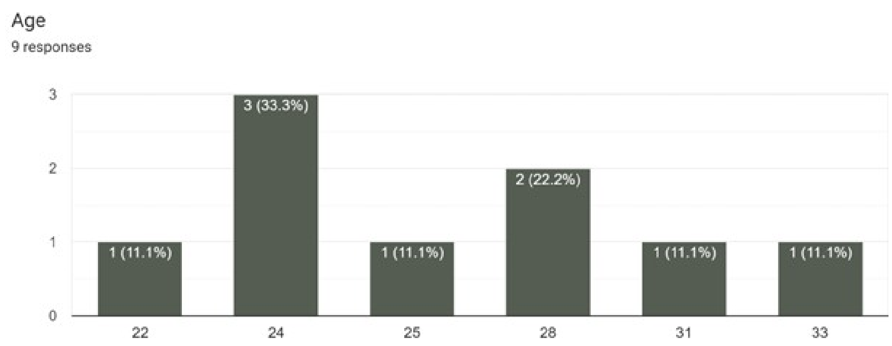


Figure 5.1: Age distribution of participants.

are blindfolded in the subsequent task-based testing, the absence of people with permanent visual impairments may still lead to incomplete insights into the usability of the assistive tool.

In terms of their use habits with web browsers, almost all the respondents use web browsers daily, with Google Chrome and Safari ranking as the top two choices, as indicated in Figure 5.2. This aligns with our decision to choose the Chrome extension as the platform for developing the assistive tool prototype.

When it comes to digital games, 44% of participants play them at least once per week and 12% daily. The rest of the respondents admit they only play digital games from time to time. Portable devices, such as Smartphone and tablets, ranks as the favourite devices for playing video games, while PCs and gaming consoles hold a tie for second place and web browsers rank last.

As for the game genre participants typically play, the results in Figure 5.3 indicate that most of them have experience in various types of games. Additionally, nearly half of them marked adventure games as a familiar game genre, which can be beneficial for usability testing as the test subject includes a point-and-click adventure game.

However, only 33% of participants have experience with games utilizing auditory

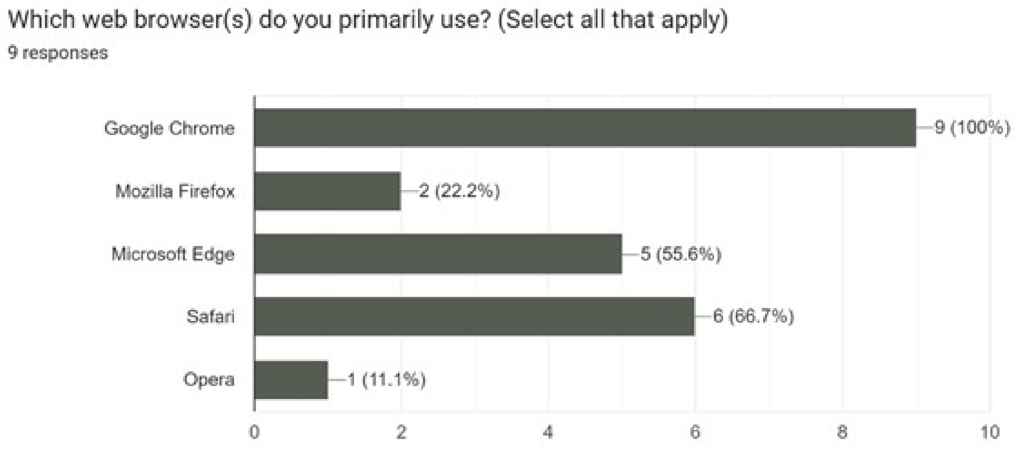


Figure 5.2: Web browser preference, in which Google Chrome dominates the usage among participants.

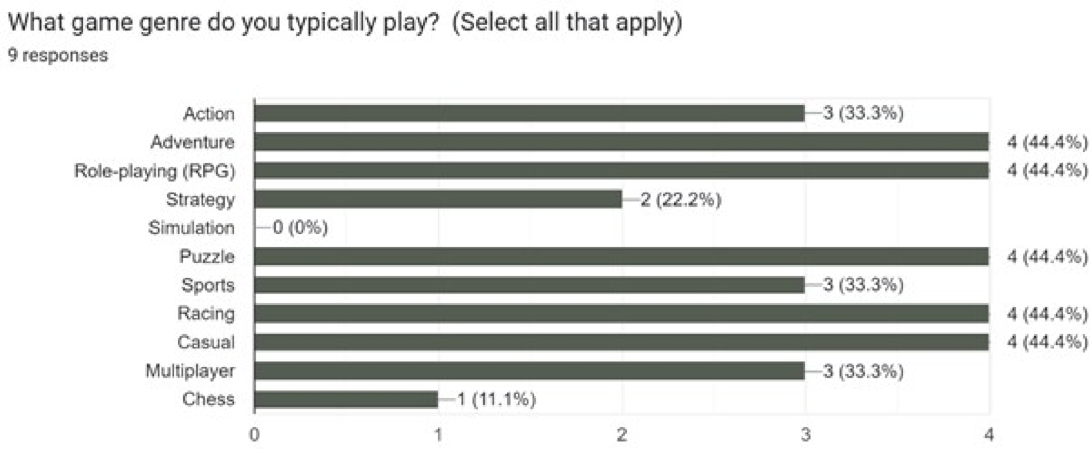


Figure 5.3: Game genres that participants typically play, in which half of them play adventure games.

cues for conveying information. This can have a negative impact on the testing outcome, especially for evaluating the effectiveness of auditory-based features such as the sonification feature.

In Figure 5.4, we can also observe that around 44% of respondents have used assistive technologies, including a screen magnifier, virtual assistant, and teams subtitle, to aid their digital device usage. Nevertheless, none of them has experience with screen readers, which can lead to insufficient feedback regarding the effectiveness of such features within the assistive tool.

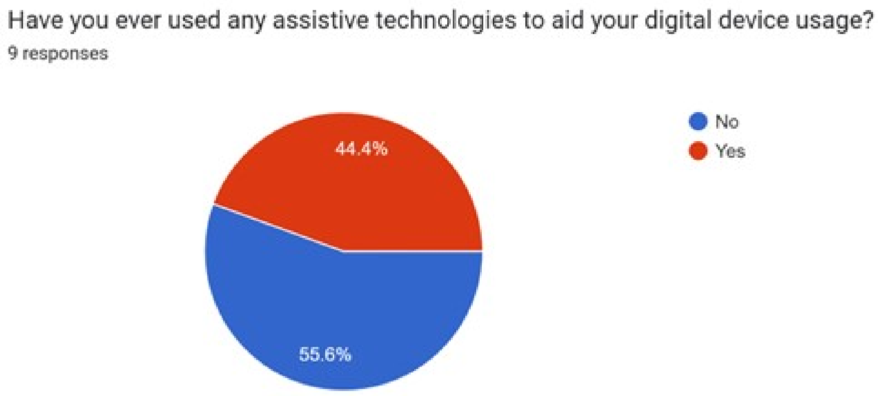


Figure 5.4: Participants' prior usage of assistive technologies.

5.2.2 General Use Experience

In the third part of the questionnaire, the overall use experience feedback of the assistive tool has been collected through five questions, as shown in Figure 5.5.

Specifically speaking, most participants agree that the assistive tool was easy to use. Meanwhile, more than half of them agree that it is unnecessarily complex. Further interviews indicate that the unnecessary complexity partially comes from the ambiguity of feature names. For instance, the feature that injects appropriate

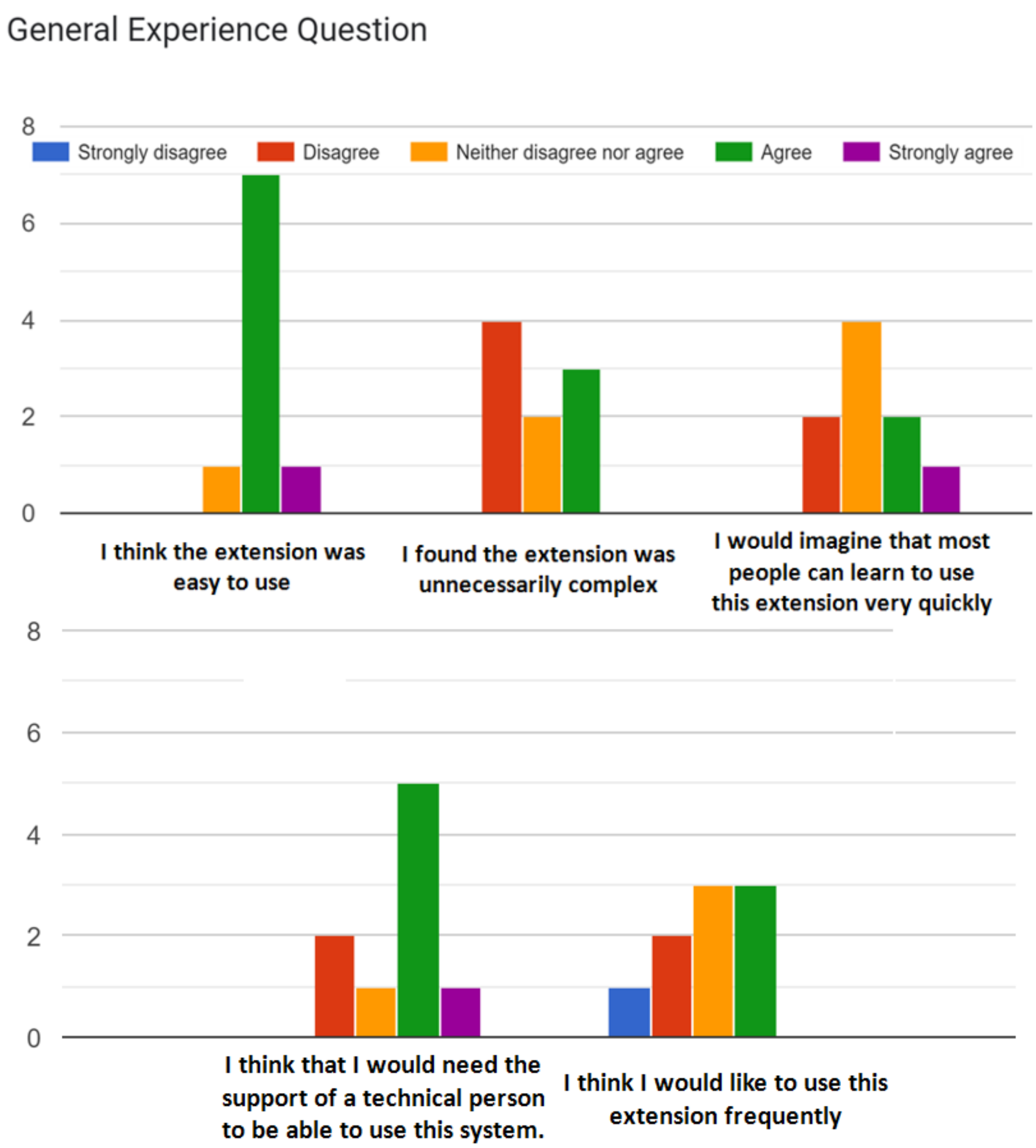


Figure 5.5: General use experience survey result.

ARIA attributes to enhance the compatibility of HTML5 games with screen readers is named 'ARIA Optimization' in the assistive tool, which raises confusion among participants without relevant background knowledge.

Moreover, most participants agree that individuals can quickly learn to use the assistive tool. However, there is a strong consensus that support from a technical expert is necessary. This observation correlates with the findings of the second question, indicating that the tool's unnecessary complexity presents challenges for users to use it independently.

Despite the majority of participants' negative answers about using the assistive tool frequently, this may be due to the absence of non-sighted participants in this survey, as indicated in the last section.

5.2.3 Feature-Specific Use Experience

As can be seen in Figure 5.6, the majority of the users affirm that the ability to customize game fonts and adjust cursor sizes enhances their gaming experience, aligning with the emphasis placed on these basic features in most accessibility guidelines.

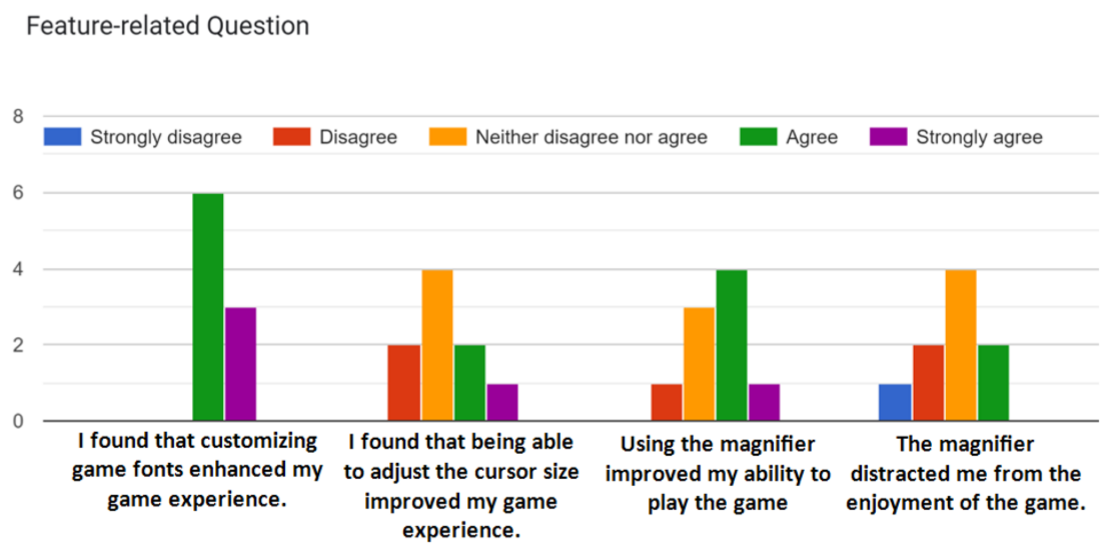


Figure 5.6: Feature-specific use experience feedback, including font customization, cursor size customization and the magnifier.

As for the magnifier, most participants think it improved their ability to play the game, with some acknowledging it would distract them from enjoying the game.

Further interview indicates that the distraction is likely due to other content being obscured when the magnifier is activated.

When questioned about the ARIA Optimization feature, the majority of participants indicated that there is no significant improvement in their gaming experience when using this feature, see Figure 5.7. This outcome is understandable, as ARIA optimization is specifically designed for screen reader users, a demographic that was lacking in this testing. Only two participants agreed that ARIA Optimization can be beneficial after receiving a comprehensive explanation about screen readers and this feature from technical support.

Additionally, nearly half of the participants mentioned that the narration feature enhances their understanding of the game without causing disruption or detracting from immersion.

As for the highlight feature, participants highly agree that it facilitates the identification of important elements, which effectively improves their gaming experience. Besides, more than half of the participants show a preference for flashing highlight borders rather than the static border.

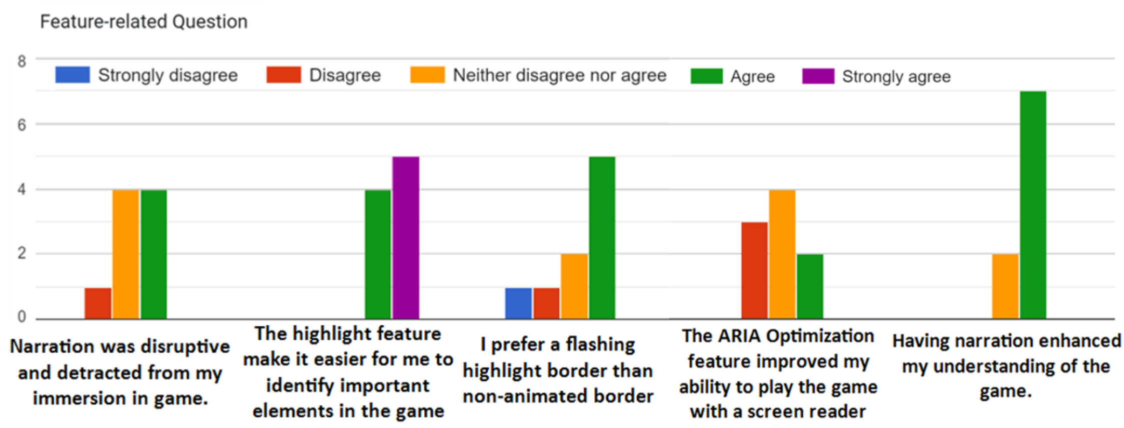


Figure 5.7: Feature-specific use experience feedback, including ARIA optimization, narration and highlight.

Questions and responses regarding the sonification feature are shown in Figure 5.8. It is evident from the data that most participants do not find this feature useful in locating important elements within the game. Moreover, only one participant finds it helpful, while two others hold opposing opinions.

Besides, participants do not show significant preference towards sound options 1 and 2, nor towards volume-based and tempo-based strategies. This contrasts with findings from a related study [59], where players showed a preference for the loudness condition over the tempo condition. However, given the limited number of testers, further research is needed before reaching any definitive conclusions.

Even so, a dominant portion of participants agree that combining both volume and tempo strategies in sonifying objects is more effective than utilizing a single strategy. This finding is potentially valuable for future development and similar studies.

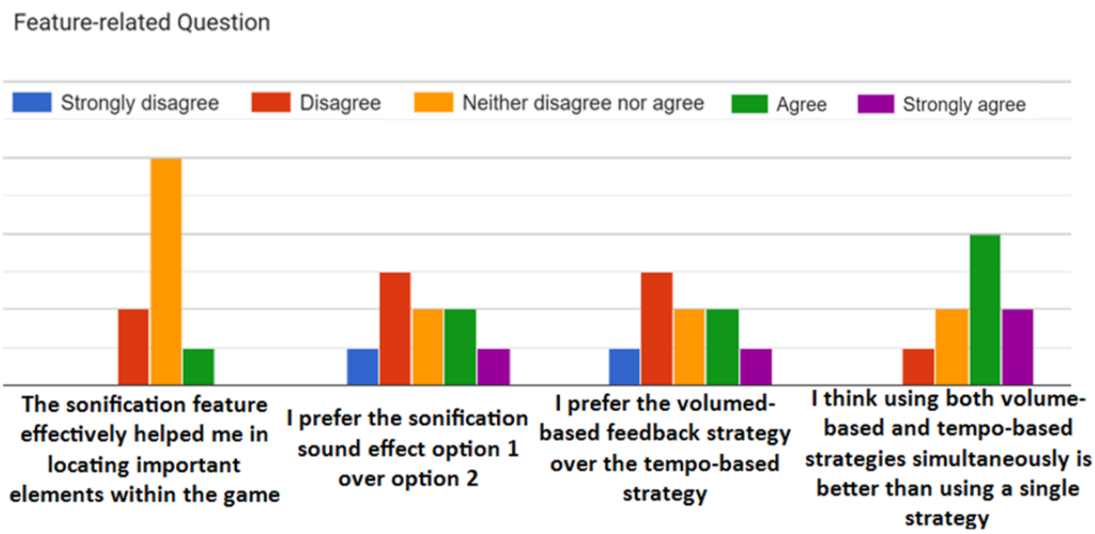
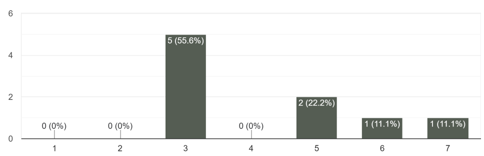


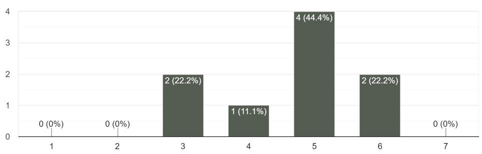
Figure 5.8: Use experience feedback towards the sonification feature.

5.2.4 Task-Based Use Experience

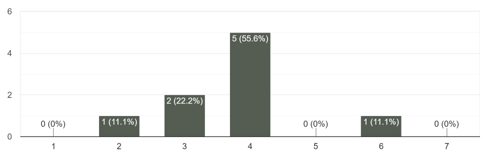
Two tasks were designed for this part of the survey to assess the effectiveness and users' satisfaction with the sonification feature. In Task 1, participants need to locate the contact information in the game's main menu. In Task 2, participants were required to solve a puzzle by finding a flashlight in the glove box inside a car game scene. Both tasks were conducted within the same point-and-click adventure game, and participants were blindfolded in both scenarios. The results of participants' feedback for both tasks are presented in Figure 5.9 and Figure 5.10, respectively.



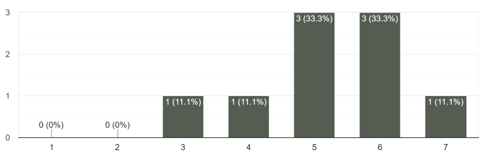
(a) How easy was it to perform the task?



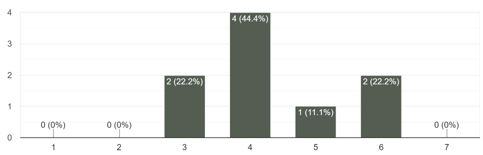
(b) How mentally demanding was the task?



(c) How physically demanding was the task?



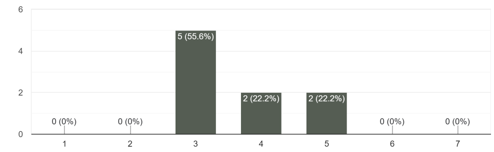
(d) How successful were you in accomplishing the task?



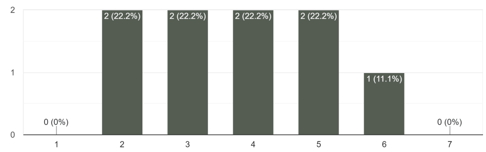
(e) How hard did you have to work to accomplish your level of performance?

Figure 5.9: Results of usability testing Task 1.

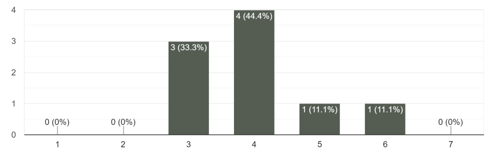
Approximately 56% of participants found both Task 1 and Task 2 slightly difficult to complete. Furthermore, more participants think that Task 1 is easier than Task 2, as It is more straightforward. This result aligns with our prejudgement and further validates our decision to set Task 1 before Task 2.



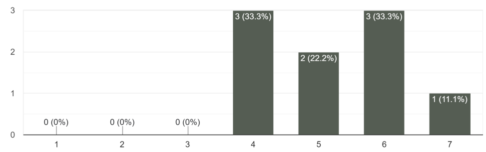
(a) How easy was it to perform the task?



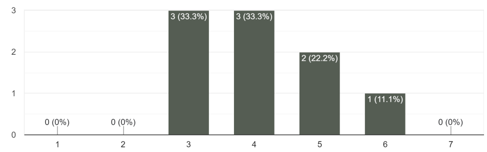
(b) How mentally demanding was the task?



(c) How physically demanding was the task?



(d) How successful were you in accomplishing the task?



(e) How hard did you have to work to accomplish your level of performance?

Figure 5.10: Results of usability testing Task 2.

When questioned about how mentally demanding it was to accomplish these tasks, around 44% of participants perceived Task 2 as mentally demanding, whereas only 22% expressed the same opinion for Task 1. Further interviews indicated that this difference was due to the background rain sound in Task 2 interfering with the sonification function.

A similar distribution is observed in the third question of both tasks, which asks how physically demanding the task was. Only 33% of participants considered the two tasks to be physically demanding, yet more participants believed Task 1 was more physically demanding than Task 2. On-site observation shows that this can be caused by the tight placement of interactive elements in the main menu involved in Task 1. Hence, some participants have to pay extra attention to stop mouse movement promptly when hearing the item-found sound effect, otherwise, they will end up on nearby elements eventually.

General positive answers are observed in the fourth question asking how successful participants think they were in accomplishing tasks 1 and 2. More than 66% of participants believe they were successful in their missions, showing a positive potential of the sonification feature in guiding players through games.

As for the fifth and last question, approximately 33% of participants think they need to work hard to accomplish task 2 while the percentage for task 1 is 22%. Despite this, the majority of participants did not find the tasks overly challenging.

Additionally, direct observation shows differences in participants' behaviour using the sonification feature. For instance, some participants only move their mouse when they hear the navigation sound and pause between movements, while others continuously move their mouse regardless of auditory cues, which occasionally results in inaccurate navigation. Moreover, we observed that the mouse cursor speed can influence the effectiveness of sonification, if the cursor moves too quickly, participants are more likely to miss the target element.

Furthermore, two participants reported finding it easier to determine orientation when the binaural sound is only played from one ear, in contrast to hearing sounds from both ears. However, this statement was not common among other participants.

5.2.5 Feedback and Expectation

The last section of the questionnaire consists of three open questions asking participants about their likes and dislikes towards such an assistive tool, as well as their expectations for future versions of this prototype. The subsequent content first evaluates the effectiveness of each feature in enhancing the gaming experience for testing participants. This evaluation is based on the first two questions, which ask about which part of the assistive tool is their favourite and least used, respectively. Then, a discussion of the answer to RQ3 is given, before a summary of users' expected features and improvements provided in the end.

The font customization and cursor customization features are considered useful yet most basic features in enhancing the general game experience. Only one participant singled them out as their favourite feature, while three participants listed them as the least used, as the default format in the provided game was adequate for them.

Next, with three votes in the favourite feature question and only one vote in the least used question, the magnifier feature is one of the participants' favourite features in this testing. Participants claimed that it is really helpful in dimly lit game environments, particularly in aiding item recognition. However, It is worth noting that this outcome may be influenced by the provided testing subject, which is a suspense adventure game comprising mainly dark and low-light scenes.

Conversely, the ARIA optimization feature is considered the least popular among participants. It gained two votes in the least frequently used question and none in the favorite question. This is understandable given the lack of screen reader users in this testing.

However, the built-in narration feature is considered one of the favourite features in this testing. Participants appreciated its utility as it allows them to receive game information without relying on vision.

The highlight feature received one vote for the favourite feature and one for the least frequently used. The participant who selected it as the favourite claimed that it was useful, especially when used in dark game scenes. Conversely, another participant claimed that such a feature would spoil the fun of the game.

As for the sonification feature, while it gained two votes as a favorite for its innovation, it also received a solid opposing vote. Some participants found it to be challenging, for it requires patience and practice. Moreover, some participants expressed confusion and even stress when they failed to locate the target element.

In summary, participants in this testing generally agreed that most features, except for the ARIA optimization and highlight feature, effectively enhance their gaming experience.

Although the feedback from participants in this testing is invaluable, a definitive answer to RQ3 cannot be formed with the data collected so far. RQ3 is about how effectively the implementation of the assistive tool enhances the game experience for visually impaired people. However, insufficient diversity of demographics in this testing, especially the absence of people with different eye conditions and prior experience with screen readers, contributes to the limitation of this testing.

The last question of this questionnaire asks participants what features they would like to see in such assistive tools. The top three features that were nominated most are voice commands, colour contrast adjustment, and tactile feedback. Participants hope to incorporate voice recognition so that the assistive tool can understand vocal commands and operate accordingly. Besides, participants wish for tactile feedback to be added so that they do not need to rely solely on hearing in navigation. Colour contrast adjustment is considered a fundamental accessibility feature, similar to font and cursor customization, according to some participants.

As for the improvement of existing features in the current prototype. More than one participant noted insufficient instructions for features in the popup. Besides, participants mentioned that separate volume adjustment is necessary, especially in using sonification, as it can be greatly interfered with by the original game's sounds. Some suggested implementing a drag-and-drop feature in popup widgets, which allows them to adjust the order of features according to their preferences.

Some of the suggestions mentioned above align with the initial plans for the assistive tool but were not implemented due to time constraints. The incorporation of these features will depend on the future work plan, which will be discussed in the next chapter.

5.3 Discussion

In this section, we revisit the research questions of this thesis and provide summarized responses to each, based on findings from the development process and testing results. We also discuss the limitations of the proposed answers.

Firstly, RQ1 discusses what assistive technology features can be integrated into HTML5 games while preserving the integrity and originality of the games. The prototype developed in this thesis validates the feasibility of incorporating a range of basic accessibility features into HTML5 games, including font and cursor customization, magnifier, and keyboard shortcuts. Moreover, narration through TTS and screen readers, highlights for indicating interactive elements, and sonification are also successfully added. However, the preliminary prototype was designed for HTML5 games without advanced graphic renderings. Further research is needed to determine if these features are feasible in more complex games.

The second research question (RQ2) explores how to implement such an assistive tool for HTML5 games. This thesis implements the tool as a browser add-on, more specifically a Chrome extension, allowing it to run simultaneously with HTML5 games. The user interface was built using HTML, CSS, and JavaScript, while Javascript was also used to inject accessibility features. The implementation for each feature can be found in Section 4.4.

RQ3 focuses on the effectiveness of the assistive tool in enhancing the game experience for visually impaired people. Usability testing was conducted, and the results indicate that most implemented accessibility features positively impact the gaming experience. However, due to the limited number of participants with various eye conditions, it is impossible to draw a conclusion for this research question. Further observation and testing with a wider range of participants are necessary.

The next chapter will conclude this thesis, with an initial plan for future work.

6 Conclusion

Visually impaired people encounter challenges when accessing digital games, as most games massively rely on visual cues for conveying information. Accessible games such as audio games have raised attention among game studios and researchers. Audio games often utilize sensory substitution methods to substitute visual content with auditory and haptic cues. Navigation in the UI and game world is a key point in designing accessible games for visually impaired people, with binaural sounds being an effective technique to facilitate it. Besides, sonification has emerged as an innovative approach in this field.

Despite the popularity of HTML5 games and well-established web accessibility guidelines, an assistive tool supporting multiple HTML5 games is absent. This can be attributed to technical constraints such as the Same-Origin Policy (SOP) and complex graphical rendering, as well as a diverse range of development frameworks and engines. This thesis aims to fill this gap by exploring the possibility of developing such an assistive tool.

Based on requirements extracted from accessibility guidelines and related work, a prototype in the form of the Chrome extension was developed. Due to time constraints, only select features were implemented. The prototype supports HTML5 games without advanced graphic rendering techniques.

A usability testing was conducted to evaluate the effectiveness of the implementation in enhancing the game experience. Basic features like font, cursor customization and the magnifier, were well-received. The built-in narration also gained positive feedback. However, opinions on the highlight feature varied, with some criticizing it spoils the game's fun. Sonification also garnered mixed reviews, for its innovation and complexity.

Due to time constraints, the current prototype includes only a limited number of features, and the testing demographic lacks diversity. Although participants are blindfolded in the testing, the lack of blind testers limits our findings. Nevertheless, the design and development process of this assistive tool offers some experiences for further research in this area.

Although participants in this testing expressed interest in additional accessibility features, whether these features can be incorporated into this assistive tool requires further research. Except for this, future work of this thesis will focus on enhancing the prototype and further testing. Optimizing existing features based on participants' feedback is essential. Besides, expanding the prototype to support HTML5 games with advanced graphic rendering techniques and mainstream game engines is an interesting research direction. Moreover, a broader range of testing participants is necessary to ensure a more solid analysis of the effectiveness of the assistive tool.

References

[1] Newzoo. “Newzoo’s global games market report 2023”. Retrieved 2024-04-18. (2024), [Online]. Available: https://newzoo.com/resources/trend-reports/newzoo-global-games-market-report-2023-free-version.

[2] S. L. Anderson and K. Schrier, “Disability and video games journalism: A discourse analysis of accessibility and gaming culture”, *Games and Culture*, vol. 17, no. 2, pp. 179–197, 2022. DOI: 10.1177/15554120211021005.

[3] M. Brown and S. L. Anderson, “Designing for disability: Evaluating the state of accessibility design in video games”, *Games and Culture*, vol. 16, no. 6, pp. 702–718, 2021. DOI: 10.1177/1555412020971500.

[4] P. Cairns, C. Power, M. Barlet, and G. Haynes, “Future design of accessibility in games: A design vocabulary”, *International Journal of Human-Computer Studies*, vol. 131, pp. 64–71, 2019, ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2019.06.010.

[5] C. M. Jones, L. Scholes, D. Johnson, M. Katsikitis, and M. C. Carras, “Gaming well: Links between videogames and nourishing mental health”, *Frontiers in Psychology*, vol. 5, p. 260, 2014. DOI: 10.3389/fpsyg.2014.00260.

[6] F. Pallavicini, A. Ferrari, and F. Mantovani, “Video games for well-being: A systematic review on the application of computer games for cognitive and emotional training in the adult population”, *Frontiers in Psychology*, vol. 9, p. 2127, 2018. DOI: 10.3389/fpsyg.2018.02127.

[7] H. Kuo, M. Yeomans, D. Ruiz, and C. Lin, "Video games and disability—a risk and benefit analysis", *Frontiers in Rehabilitation Sciences*, vol. 5, 2024, ISSN: 2673-6861. DOI: 10.3389/fresc.2024.1343057.

[8] J. Beeston, C. Power, P. Cairns, and M. Barlet, "Accessible player experiences (apx): The players", K. Miesenberger and G. Kouroupetroglou, Eds., 2018. DOI: 10.1007/978-3-319-94277-3_40.

[9] C. Power, P. Cairns, M. Barlet, and S. Weitz, "Whose Responsibility is Accessibility in Games Anyway? Everyone", vol. 14145, 2023. DOI: 10.1007/978-3-031-42293-5_21.

[10] E. Parliament, D.-G. for Parliamentary Research Services, L. Nierling, *et al.*, *Assistive technologies for people with disabilities. Part III, Perspectives, needs and opportunities*. European Parliament, 2018. DOI: 10.2861/11162.

[11] T. H. Apperley, "Genre and game studies: Toward a critical approach to video game genres", *Simulation & Gaming*, vol. 37, no. 1, pp. 6–23, 2006. DOI: 10.1177/1046878105282278.

[12] T. H. Apperley. "My experience using assistive technology as an indie game developer". Retrieved 2024-04-18. (2024), [Online]. Available: https://allyant.com/my-experiences-of-using-assistive-technology-as-an-indie-game-developer/.

[13] A. Salter, *What Is Your Quest?: From Adventure Games to Interactive Books*. University of Iowa Press, 2014, Retrieved 2024-04-18. [Online]. Available: http://www.jstor.org/stable/j.ctt20q1t0b.

[14] J. Seidelin, *HTML5 Games : Creating Fun with HTML5, CSS3, and WebGL*. John Wiley and Sons, Incorporated, 2011, Retrieved 2024-04-18. [Online]. Available: https://ebookcentral.proquest.com/lib/kutu/detail.action?docID=698120.

[15] QYResearch. "Global H5 Games Market Size, Status and Forecast 2022-2028". Retrieved 2024-04-18. (2022), [Online]. Available: https://www.giiresearch.com/report/qyr1170014-global-h5-games-market-size-status-forecast.html.

[16] OpenAI. "Chatgpt". Retrieved 2024-04-30. (2022), [Online]. Available: https://openai.com/chatgpt.

[17] G. L. Giudice and A. Catalá, *Visual Impairment and Blindness*. Rijeka: IntechOpen, 2020, ISBN: 978-1-83880-258-5. DOI: 10.5772/intechopen.73976. [Online]. Available: https://doi.org/10.5772/intechopen.73976.

[18] I. Dela Torre and I. Khaliq, "A study on accessibility in games for the visually impaired", in *2019 IEEE Games, Entertainment, Media Conference (GEM)*, 2019, pp. 1–7. DOI: 10.1109/GEM.2019.8811534.

[19] M. Lowth. "Refraction and refractive errors". Retrieved 2024-04-18. (2023), [Online]. Available: https://patient.info/doctor/refraction-and-refractive-errors.

[20] N. H. Mackworth, "Visual noise causes tunnel vision", *Psychonomic Science*, vol. 3, pp. 67–83, 1965. DOI: 10.3758/BF03343023.

[21] R. Gregory and P. Cavanagh, "The Blind Spot", *Scholarpedia*, vol. 6, no. 10, p. 9618, 2011, revision #150975. DOI: 10.4249/scholarpedia.9618.

[22] N. E. Institute. "Color blindness". Retrieved 2024-04-18. (2023), [Online]. Available: https://www.nei.nih.gov/learnabout-eye-health/eye-conditions-and-diseases/color-blindness.

[23] U. Wilhelmsson, H. Engstrom, J. Brusk, and P.-A. Ostblad, "Accessible game culture using inclusive game design - participating in a visual culture that you cannot see", in *2015 7th International Conference on Games and Virtual*

Worlds for Serious Applications (VS-Games), 2015, pp. 1–8. DOI: 10.1109/VS-GAMES.2015.7295764.

[24] G. D. Conference. "State of the game industry report 2020". Retrieved 2024-04-18. (2020), [Online]. Available: https://reg.gdconf.com/gdc-state-ofgame-industry-2020.

[25] International Organization for Standardization, *ISO 9241-11:2018: Ergonomics of human-system interaction — part 11: Usability: Definitions and concepts*, ISO, Geneva, Switzerland, 2018.

[26] International Organization for Standardization/International Electrotechnical Commission, *ISO/IEC 25010:2011: Systems and software engineering – systems and software quality requirements and evaluation (square) – system and software quality models*, ISO/IEC, Geneva, Switzerland, 2011.

[27] K. Cunningham, *Accessibility Handbook*. O'Reilly Media, Inc., 2012, ISBN: 9781449322816.

[28] M. Ferati, B. Raufi, A. Kurti, and B. Vogel, "Accessibility requirements for blind and visually impaired in a regional context: An exploratory study", in *2014 IEEE 2nd International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE)*, 2014, pp. 13–16. DOI: 10.1109/UsARE.2014.6890995.

[29] T. Catarci, A. Perini, N. Seyff, S. Humayoun, and N. Qureshi, "First international workshop on usability and accessibility focused requirements engineering (usare 2012): Summary report", *ACM SIGSOFT Software Engineering Notes*, vol. 38, pp. 43–46, Jan. 2013. DOI: 10.1145/2382756.2382767.

[30] C. Abbott, "Defining assistive technologies-a discussion", *Journal of Assistive Technologies*, vol. 1, no. 1, pp. 6–9, 2007. DOI: 10.1108/17549450200700002.

[31] V. Austin and C. Holloway, "Assistive technology (at), for what?", *Societies*, vol. 12, no. 6, 2022, ISSN: 2075-4698. DOI: 10.3390/soc12060169. [Online]. Available: https://www.mdpi.com/2075-4698/12/6/169.

[32] C. Sik-Lányi, E.-J. Hoogerwerf, and K. Miesenberger, Eds., *Assistive Technology: Building Bridges.* Amsterdam: IOS Press, Incorporated, 2015.

[33] W. H. Organization and U. N. C. Fund, *Global report on assistive technology.* World Health Organization, 2022.

[34] International Organization for Standardization, *Assistive Products—Classification and Terminology.* Geneva, Switzerland: ISO, 2022.

[35] "Overwatch wiki - game options". Retrieved 2024-04-18. (2017), [Online]. Available: https://overwatch-archive.fandom.com/wiki/Game_Options.

[36] W. W. W. C. (W3C). "Web content accessibility guidelines (wcag) 2.1". Retrieved 2024-04-23. (2023), [Online]. Available: https://www.w3.org/TR/WCAG21/.

[37] H.-J. Park and S.-B. Kim, "Guidelines of serious game accessibility for the disabled", in *2013 International Conference on Information Science and Applications (ICISA)*, 2013, pp. 1–3. DOI: 10.1109/ICISA.2013.6579380.

[38] AbleGamers. "Accessible player experiences". Retrieved 2024-04-23. (2022), [Online]. Available: https://accessible.games/accessible-player-experiences/.

[39] C. Power, P. Cairns, M. Barlet, G. Haynes, J. Beeston, and T. DeHaven, "Validation and Prioritization of Design Options for Accessible Player Experiences", *Interacting with Computers*, vol. 33, no. 6, pp. 641–656, Jun. 2022, ISSN: 1873-7951. DOI: 10.1093/iwc/iwac017. [Online]. Available: https://doi.org/10.1093/iwc/iwac017.

[40] Game Accessibility Guidelines. "Game accessibility guidelines". Retrieved 2024-04-23. (2019), [Online]. Available: https://gameaccessibilityguidelines.com/.

[41] A. F. Pereira and F. R. S. Coutinho, "Game accessibility guidelines for people with sequelae from macular chorioretinitis", in *2017 16th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 2017, pp. 96–105. DOI: 10.1109/SBGames.2017.00019.

[42] P. Bach-y-Rita and S. W. Kercel, "Sensory substitution and the human–machine interface", *Trends in Cognitive Sciences*, vol. 7, no. 12, pp. 541–546, 2003, ISSN: 1364-6613. DOI: https://doi.org/10.1016/j.tics.2003.10.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364661303002900.

[43] J. Wang and Q. Shao, "Basic experimental research on electrotactile physiology for deaf auditory substitution", *Acta Acad. Med. Shandong*, vol. 35, no. 1, pp. 1–5, 1997.

[44] D. Eagleman, "Plenary talks: A vibrotactile sensory substitution device for the deaf and profoundly hearing impaired", in *2014 IEEE Haptics Symposium (HAPTICS)*, 2014, pp. xvii–xvii. DOI: 10.1109/HAPTICS.2014.6775419.

[45] J. M. E. da Silva, A. d. C. Callado, and P. M. Jucá, "Representing sentiment using colors and particles to provide accessibility for deaf and hard of hearing players", in *2018 17th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 2018, pp. 221–22 109. DOI: 10.1109/SBGAMES.2018.00034.

[46] A. T. Woods and F. N. Newell, "Visual, haptic and cross-modal recognition of objects and scenes", *Journal of Physiology, Paris*, vol. 98, no. 1-3, pp. 147–159, 2004. DOI: 10.1016/j.jphysparis.2004.03.006.

[47] A. Pasqualotto and T. Esenkaya, “Sensory substitution: The spatial updating of auditory scenes “mimics” the spatial updating of visual scenes”, *Frontiers in Behavioral Neuroscience*, vol. 10, 2016, ISSN: 1662-5153. DOI: 10.3389/fnbeh.2016.00079. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnbeh.2016.00079.

[48] E. J. Wong, K. M. Yap, J. Alexander, and A. Karnik, “Habos: An exploratory study of haptic-audio based online shopping for the visually impaired”, in *2015 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, 2015, pp. 1–6. DOI: 10.1109/HAVE.2015.7359444.

[49] Nintendo. “Nintendo Switch Sports”. Retrieved 2024-04-22. (2022), [Online]. Available: https://www.nintendo.fi/nintendo-switch-perhe/pelit/nintendo-switch-sports.

[50] Nintendo. “1-2-Switch”. Retrieved 2024-04-22. (2017), [Online]. Available: https://www.nintendo.com/us/store/products/1-2-switch-switch/.

[51] Nintendo. “Mario Tennis Aces”. Retrieved 2024-04-22. (2018), [Online]. Available: https://www.nintendo.fi/nintendo-switch-perhe/pelit/mario-tennis-aces.

[52] W. S. Gan and J.-W. Choi, *Spatial Audio*. 2017, ISBN: 9783038425861. DOI: 10.3390/books978-3-03842-586-1.

[53] H. Møller, “Fundamentals of binaural technology”, *Applied Acoustics*, vol. 36, no. 3, pp. 171–218, 1992.

[54] A. Neidhardt and A. Rüppel, “Multiplayer audio-only game: Pong on a massive multichannel loudspeaker system”, in *Proceedings of the 7th Audio Mostly Conference: A Conference on Interaction with Sound*, Sep. 2012, pp. 130–134. DOI: 10.1145/2371456.2371477.

[55] P. A. de Oliveira, E. P. Lotto, A. G. D. Correa, L. G. G. Taboada, L. C. P. Costa, and R. D. Lopes, "Virtual stage: An immersive musical game for people with visual impairment", in *2015 14th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 2015, pp. 135–141. DOI: 10.1109/SBGames.2015.26.

[56] P. Meijer, "An experimental system for auditory image representations", *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 2, pp. 112–121, 1992. DOI: 10.1109/10.121642.

[57] J. Clemons, S. Y. Bao, S. Savarese, T. Austin, and V. Sharma, "MVSS: Michigan visual sonification system", in *2012 IEEE International Conference on Emerging Signal Processing Applications*, 2012, pp. 143–146. DOI: 10.1109/ESPA.2012.6152466.

[58] F. Ribeiro, D. Florêncio, P. A. Chou, and Z. Zhang, "Auditory augmented reality: Object sonification for the visually impaired", in *2012 IEEE 14th International Workshop on Multimedia Signal Processing (MMSP)*, 2012, pp. 319–324. DOI: 10.1109/MMSP.2012.6343462.

[59] Y. Sekhavat, M. Azadehfar, and H. e. a. Zarei, "Sonification and interaction design in computer games for visually impaired individuals", *Multimed Tools Appl*, vol. 81, pp. 7847–7871, 2022. DOI: 10.1007/s11042-022-11984-3.

[60] M. Urbanek, P. Fikar, and F. Güldenpfennig, "About the sound of bananas — anti rules for audio game design", in *2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)*, 2018, pp. 1–7. DOI: 10.1109/SeGAH.2018.8401361.

[61] Sonnar. "Red riding hood". Retrieved 2024-04-23. (2017), [Online]. Available: https://apps.apple.com/us/app/red-riding-hood/id1281845225.

[62] DOWiNO. "A blind legend". Retrieved 2024-04-23. (2015), [Online]. Available: http://www.ablindlegend.com/.

[63] Unknown. "Engkwentro: An interesting journey on android". Retrieved 2024-04-23. (2020), [Online]. Available: https://blindhelp.net/software/engkwentro.

[64] M. Home. "Sonar island". Retrieved 2024-04-23. (2022), [Online]. Available: https://www.mentalhome.eu/sonar-islands/.

[65] Unknown. "Drive". Retrieved 2024-04-23. (2017), [Online]. Available: https://blindhelp.net/software/drive.

[66] L. C. De Biase, A. G. Correa, L. Dias, E. P. Lotto e, and R. D. Lopes, "An accessible roller coaster simulator for touchscreen devices: An educational game for the visually impaired", in *2018 IEEE Games, Entertainment, Media Conference (GEM)*, 2018, pp. 101–105. DOI: 10.1109/GEM.2018.8516457.

[67] D. Song, A. Karimi, and P. Kim, "Toward designing mobile games for visually challenged children", in *Proceeding of the International Conference on e-Education, Entertainment and e-Management*, 2011, pp. 234–238. DOI: 10.1109/ICeEEM.2011.6137794.

[68] K. Allain, B. Dado, M. Van Gelderen, *et al.*, "An audio game for training navigation skills of blind children", in *2015 IEEE 2nd VR Workshop on Sonic Interactions for Virtual Environments (SIVE)*, 2015, pp. 1–4. DOI: 10.1109/SIVE.2015.7361292.

[69] L. Giarré, I. Tinnirello, and L. Jaccheri, "In.line: A navigation game for visually impaired people", in *Entertainment Computing – ICEC 2017*, N. Munekata, I. Kunita, and J. Hoshino, Eds., Cham: Springer International Publishing, 2017, pp. 147–153, ISBN: 978-3-319-66715-7.

[70] J. Torrente, E. J. Marchiori, J. Á. Vallejo-Pinto, M. Ortega-Moral, P. Moreno-Ger, and B. Fernández-Manjón, "Eyes-free interfaces for educational games", in *2012 International Symposium on Computers in Education (SIIE)*, 2012, pp. 1–6.

[71] M. Matsuo, T. Miura, M. Sakajiri, J. Onishi, and T. Ono, "Shadowrine: Accessible game for blind users, and accessible action rpg for visually impaired gamers", in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 002 826–002 827. DOI: 10.1109/SMC.2016.7844667.

[72] P. Escudeiro, N. Escudeiro, P. Oliveira, and M. Barbosa, "Blind's inclusion in mobile games", in *2017 27th EAEEIE Annual Conference (EAEEIE)*, 2017, pp. 1–5. DOI: 10.1109/EAEEIE.2017.8768494.

[73] H. Cho, K. Park, and S. Choi, "Equal-level interaction: A case study for improving user experiences of visually-impaired and sighted people in group activities", in *2018 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, 2018, pp. 1–6. DOI: 10.1109/HAVE.2018.8547502.

[74] International Organization for Standardization, "Iso/iec/ieee draft international standard - systems and software engineering – life cycle processes – requirements engineering", *ISO/IEC/IEEE P29148*$_F$*DIS*, *September*2018, pp. 1–104, 2018.

[75] I. Corp. "Itch.io". Retrieved 2024-04-30. (2024), [Online]. Available: https://itch.io/.

[76] Html5games. "Html5games.com". Retrieved 2024-04-30. (2024), [Online]. Available: https://html5games.com/.

[77] Makzan, *HTML5 Games Development by Example Beginner's Guide: Beginner's Guide*. Packt Publishing, Limited, 2011. [Online]. Available: https://ebookcentral.proquest.com/lib/kutu/detail.action?docID=948541.

[78] G. Borland. "Sort the court". Retrieved 2024-04-30. (2016), [Online]. Available: https://graebor.itch.io/sort-the-court.

[79] Mozilla. "Same-origin policy". Retrieved 2024-04-30. (2024), [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy.

[80] Tiagox. "Witch hat and ears of cat". Retrieved 2024-04-30. (2021), [Online]. Available: https://tiagoxos.itch.io/witch-hat-n-ears-of-cat.

[81] G. Raturi. "Why html5 is the future of online gaming: An overview". Retrieved 2024-04-30. (2022), [Online]. Available: https://medium.com/codex/why-html5-is-the-future-of-online-gaming-an-overview-f9c129ca5518.

[82] Mozilla. "Manifest file format". Retrieved 2024-04-30. (2012), [Online]. Available: https://developer.chrome.com/docs/extensions/reference/manifest.

[83] T. Todorov and J. Dochkova-Todorova, "Accessible ux/ui design", in *2023 International Conference Automatics and Informatics (ICAI)*, 2023, pp. 362–366. DOI: 10.1109/ICAI58806.2023.10339066.

[84] N. von Hertzen. "Html2canvas". Retrieved 2024-04-30. (2011), [Online]. Available: https://html2canvas.hertzen.com/.

[85] A. Saienko. "Dom to image". Retrieved 2024-04-30. (2016), [Online]. Available: https://github.com/tsayen/dom-to-image.

[86] NV Access. "Nvda screen reader download". Retrieved 2024-04-30. (2024), [Online]. Available: https://www.nvaccess.org/download/.

[87] Freedom Scientific. “Jaws screen reader downloads”. Retrieved 2024-04-30. (2024), [Online]. Available: https://support.freedomscientific.com/Downloads/JAWS.

[88] Mozilla. “Aria”. Retrieved 2024-04-30. (2024), [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA.

[89] Mozilla. “Web speech api”. (2024), [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API.

[90] Mozilla. “Web audio api”. (2024), [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API.

[91] J. Yu, J. Zhang, L. Pan, Y. Chen, N. Wu, and W. Sun, “Software exploratory testing: Present, problem and prospect”, in *2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST)*, 2021, pp. 44–47. DOI: 10.1109/IAECST54258.2021.9695695.

[92] ENI CBC Med. “Game testing questionnaire”. Retrieved 2024-04-30. (2024), [Online]. Available: https://www.enicbcmed.eu/sites/default/files/2024-03/Game%20Testing%20Questionnaire.pdf.

[93] J. Nielsen. “Ten usability heuristics”. Retrieved 2024-04-30. (1994), [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/.

Appendix A GAG - Vision

Category	**Principle**
Basic	Ensure no essential information is conveyed by a colour alone. If the game uses field of view (3D engine only), set an appropriate default for expected viewing environment. Avoid VR simulation sickness triggers. Use an easily readable default font size. Use simple clear text formatting. Provide high contrast between text/UI and background. Ensure interactive elements/virtual controls are large and well spaced, particularly on small or touch screens.
Intermediate	If the game uses field of view (3D engine only), allow a means for it to be adjusted. Avoid (or provide option to disable) any difference between controller movement and camera movement, such as weapon/walk bobbing or mouse smoothing. Use surround sound. Provide an option to turn off/hide background animation. Ensure screenreader support for mobile devices. Provide an option to adjust contrast.

Category (Contd.)	**Principle (Contd.)**
	Ensure sound / music choices for key objects/events are distinct from each other. Provide a choice of cursor / crosshair colours/designs. Give a clear indication that interactive elements are interactive. Ensure manual/website are provided in a screenreader friendly format. Provide separate volume controls or mutes for effects, speech and background/music. Avoid placing essential temporary information outside the player's eye-line. Allow interfaces to be resized.
Advanced	Allow the font size to be adjusted. Provide a pingable sonar-style audio map. Provide pre-recorded voiceovers for all text, including menus and installers. Provide a voiced GPS. Allow easy orientation to/movement along compass points. Ensure that all key actions can be carried out by digital controls (pads/ keys/presses), with more complex input (eg. analogue, gesture) not required, and included only as supplementary/alternative input methods. Ensure screenreader support, including menus and installers. Use distinct sound/music design for all objects and events.

Category (Contd.)	**Principle (Contd.)**
	Simulate binaural recording. Provide an audio description track.

Appendix B Project Source Code

Github repository: https://github.com/cherr66/ACE-demo.

Appendix C Web Games Assistive Technology Survey

Google Form PDF: https://seafile.utu.fi/smart-link/4fcf2e31-e786-4d85-9ec2-e84d5dbc012c/