Vistula University

The Faculty of Computer Engineering, Graphic Design and Architecture

Program of study Computer Science

Anatolii Shcherbak

Student number 63570

DESIGN AND IMPLEMENTATION OF AN ACCESSIBILITY-FOCUSED INTERACTIVE GAMING SYSTEM FOR SIMULATING DISABILITIES

The engineering thesis
written under the supervision of
M.Sc. Eng. Stanislav Mishchenko

Warsaw, 2026

Table of Contents

Abstract

I, Anatolii Shcherbak, declare that this thesis is my own work and has not been submitted elsewhere for academic credit.

In the preparation of this thesis, artificial intelligence tools were used only in a limited and supportive role. ChatGPT was used to generate placeholder images for the game project. The author developed conceptual design, technical implementation, analysis, and final conclusions independently.

Synopsis

This diploma project investigates the design and implementation of a digital game that simulates selected types of disabilities as integral gameplay mechanics. In the proposed game system, disabilities are conceptualized and implemented as difficulty-modifying parameters in order to create a challenging and engaging user experience while simultaneously increasing awareness of the barriers and limitations faced by people with disabilities.

The project combines game design practice with analytical and technical research. It includes a systematic analysis of the contemporary game market, with a focus on the representation of disability, accessibility standards, and inclusive design approaches. Furthermore, the study examines the impact of different disabilities on player interaction and performance, and reviews existing technological and design techniques that enable players with disabilities to effectively access and play digital games.

The outcome of the project is a prototype game and a documented design methodology that demonstrates how disability simulation and accessibility-oriented design can be integrated into interactive entertainment systems.

Keywords

Disability, Game design, challenges, empathy, market, Game development, Unity

Introduction

In this Thesis will be analyzed current existing games focused on disabilities, their audience market, technologies used there, how they could be used in other projects and significance of those projects. As well as Accessibility-focused features that are most used in games such as subtitles, sound navigation, color inversion etc. The project proposes a video game designed to simulate various disabilities, including color vision deficiency, as a means of delivering a novel and educational player experience. Through an interactive narrative, players will engage with stories inspired by real individuals with disabilities, gaining insight into their daily challenges. Gameplay contains mini games in which the simulated disabilities function as constraints or "*debuffs*," [1] providing challenges designed to keep players deeply involved and interested in long time while still providing a challenge.

This design aims to foster empathy and understanding while maintaining the motivational aspects of traditional game mechanics.

The thesis focuses on developing an interactive game incorporating features to simulate disabilities, fostering empathy, and understanding. The aim of the project is to show that games made with a focus on disabilities can also be interesting for a person who is healthy. The system aims to be inclusive and educational, adhering to guidelines such as those from the *Game Accessibility Guidelines*[2]. Potential users include people with disabilities, educators, and just usual gamers. The system can be used in entertainment and education.

According to the World Health Organization (WHO), 1.3 billion people, or about 16% of the world's population, have a "significant disability"[3]. This includes a broad range of impairments - physical, sensory, intellectual, or psychological - that, in combination with environmental or personal factors, reduce a person's ability to fully participate in social, economic, and daily activities. This emphasis on creating inclusive game design not only addresses current gaps in the industry but also sets a foundation for future.

Why It Matters for Game Development

[1] Debuff - is a negative effect that weakens your character during gameplay. It might lower your stats (like strength, speed, or accuracy), block some of your abilities, or apply a harmful status like poison, slow, or silence. https://www.g2a.com/news/glossary/debuff-in-gaming-meaning-explained/ (access date: 31 January 2026).

[2] Game Accessibility Guideline - https://gameaccessibilityguidelines.com/ (access date: 31 January 2026)

[3] World Health Organization, "Disability and health," fact sheet, December 7 2023, - https://www.who.int/news-room/fact-sheets/detail/disability-and-health (access date: 31 January 2026).

Ethical / Social & Business Reasons:

- Inclusion and equal opportunity. *In the review Accessibility in Video Games: authors assert that many games and the industry treat accessibility as a low priority - which from an ethical and social justice standpoint contributes to exclusion of disabled players from a cultural and social activity widely enjoyed by others*[4]. Designing with accessibility in mind supports the principle that digital entertainment should be inclusive and available to as many people as reasonably possible.
- Diversity of needs and fairness. Disabilities are diverse - ranging from small visual impairments, color vision deficiencies, to motor or cognitive limitations. Accessibility features (e.g., adjustable contrast, scalable UI, colorblind modes, remappable controls, alternative input modalities) help ensure that games are enjoyable for people with different abilities, reducing barriers to participation.
- Social responsibility and reputation. Game developers and publishers who prioritize accessibility show social responsibility, signal respect for diversity, and contribute to reducing exclusion. This can build goodwill, align with human rights norms, and reflect positively on their brand in a global, varied community of players. *Gemma Ellison at Making Games for Everyone: The Business Case for Accessibility: Prioritizing accessibility enhances your studio's reputation and brand image. It demonstrates a commitment to inclusivity, attracting diverse players and building loyalty*[5].

Business / Market & Financial Reasons:

- Large potential audience. Given the large amount of disabled people, accessible design expands the potential market. Many players who might be excluded by default game designs could become customers if games are made accessible. *On mainstream media side, the article The Independent titled "Video game makers aren't doing enough to cater for gamers with disabilities, study finds" reports on a poll of disabled gamers where 81% reported struggle due to inaccessible features (poor controls, unreadable text, fast gameplay, flashing lights); 39% said they were forced to stop playing or abandon a game*[6].

---

[4] Accessibility in video games: a systematic review, Universal Access in the Information Society, vol. 19, 2020, pp. 169–193, https://doi.org/10.1007/s10209-018-0628-2 (accessed: 31 January 2026).

[5] Wayline, Making Games for Everyone: The Business Case for Accessibility, 13 March 2025) - https://www.wayline.io/blog/making-games-for-everyone-the-business-case-for-accessibility(accessed: 31 January 2026).

[6] Robert Knight, "Video game makers aren't catering for gamers with disabilities, study finds," The Independent, October 4, 2023, https://www.independent.co.uk/tech/video-game-makers-disabilities-mental-health-b2423844.html? (accessed: 31 January 2026).

- High spending power & market growth. Inclusive design can tap into a demographic that may be neglected by standard game design. *The market is expected to reach 1.7 billion by 2033 and is growing at a CAGR of 8.00% between 2025 and 2033*[7]. As demographic trends (aging populations, chronic conditions) and global health burdens increase, the proportion of potential players with disabilities grow making accessibility a long-term commercial advantage.
- Regulatory and reputational risk mitigation. As awareness and legal frameworks around disability rights grow globally, companies that ignore accessibility risk public criticism, regulatory scrutiny, or exclusion from markets where disability rights are enforced. Proactively building inclusive games mitigates these risks. *Non-compliance can lead to enforcement by individual EU countries, with penalties varying: Germany imposes fines up to €100k, Spain fines up to €600k and bans the service for up to 2 years risking market exclusion through service ban*[8].
- Innovation and competitive differentiation. Accessible design often leads to innovations (e.g., UI flexibility, multiple control schemes, adaptive audio, assistive options) that can improve the experience for all players - not only those with disabilities. That can become a competitive strength and a feature distinguishing developers/ publishers committed to inclusion.

How the Implemented Project Is Used:

The game consists of five chapters, each centred on the personal story of a character living with a specific disability. The structure follows the conventions of a visual novel, featuring interactive dialogue, branching decisions, and narrative progression.

A core innovation is the automatic adaptation of the game environment and interface to simulate the selected disability. All modifications are applied seamlessly upon entering a chapter, ensuring tight integration between storytelling and experiential gameplay. This reates emotional engagement by placing players in authentic situations drawn from real-life accounts.

The project has practical applications in several areas:

- Entertainment. It offers inclusive gaming suitable for players of varying abilities,

[7] HTF Market Insights, Inclusive Gaming Accessibility Tech Market Industry Overview: https://www.htfmarketinsights.com/report/4399509-inclusive-gaming-accessibility-tech-market (accessed: 31 January 2026).

[8] IGDA Game Accessibility SIG, Demystifying EAA & GPSR: New EU Laws for Game Accessibility and Safety, 28 June 2025 - https://igda-gasig.org/what-and-why/demystifying-eaa-gpsr (accessed: 31 January 2026)

delivering compelling stories and novel mechanics that remain enjoyable for both disabled and non-disabled audiences.

- Education. By directly simulating the everyday challenges associated with different disabilities, the game serves as an effective tool for building empathy, raising awareness, and supporting training in schools, universities, or professional contexts focused on inclusivity and disability understanding.

Thesis Structure:

Introduction General introduction to this Thesis.

Chapter 1, "Analysis of Selected Design Solutions," reviews existing accessibility-focused games and disability simulation titles, identifies key design features, and evaluates their strengths and limitations.

Chapter 2, "Analysis and Selection of Technologies for Accessible Game Design," compares current technologies and methodologies, examines the characteristics of the selected tools, and justifies the primary technology chosen for implementation.

Chapter 3, "Project Description," presents the conceptual and technical design of the prototype, including the design document, detailed specifications of accessibility features and simulation mechanics, and a project user manual.

Chapter 4, "Implementation of the Thesis Project," This chapter documents the practical realisation of the game prototype developed as the core outcome of the thesis. It describes the development process, key coding and integration phases, and provides all necessary technical information to locate, access, download and run the project.

Summary, assesses the status of the implemented project, identifies necessary remaining works, proposes possible improvements and extensions, and discusses difficulties encountered.

The thesis concludes with the bibliography, list of figures and tables, and appendices.

1 Chapter 1

Analysis of Selected Design Solutions

This chapter reviews existing games and systems related to accessibility and disability simulation, providing a foundation for the proposed project.

To investigate further person should answer first about what disability is and what Accessible game is.

Definition of Disability - *Disability is defined as "a mismatch between the needs of the individual and the service, product or environment offered." This means that anyone can experience a disability, and that it can be a short term or situational condition. Envision what challenges gamers with these conditions might have when playing your game, and think about how your game can be better designed for them*[9].

Definition of What Makes a Game Accessible - *For a game to be considered accessible, it must account for various forms of interaction and perception. Interfaces should be clear and adaptable, while user experiences must balance challenge with accessibility*[10].

Existing solutions:

Visual Accessibility Solutions:

- Subtitles and Captions: Textual representation of spoken dialogue and key audio cues to assist players with hearing impairments.
- High Contrast Interfaces and Scalable Text: Improves visibility for users with low vision by ensuring sufficient contrast between UI elements and allowing font sizes to be increased.
- Colorblind Modes: Adjusts colors or applies filters to accommodate different types of color vision deficiency, ensuring important visual information remains distinguishable.
- Visual Audio Indicators: Displays on screen prompts or icons for critical sounds, helping users who cannot rely on auditory cues.

Motor Accessibility Solutions:

- Customizable Controls and Adaptive Inputs: Allows remapping of keys and compatibility with adaptive controllers for players with limited mobility.

[9] Disability definition by Microsoft - https://learn.microsoft.com/en-us/windows/uwp/gaming/accessibility-for-games?(accessed: 31 January 2026).

[10] Accessible game definition by Kokkugames - https://kokkugames.com/making-games-accessible-lessons-in-ui-ux/(accessed: 31 January 2026).

- Simplified Action Schemes: Features such as one-handed play or hold to confirm actions reduce accidental inputs and allow easier interaction.
- Software and Specialized Systems: Voice Recognition Software, Eye-Tracking Devices.

Audio Accessibility Solutions:

- Independent Volume Controls: Separate adjustment of music, effects, and dialogue ensures players can tailor audio levels according to their needs.
- Audio to Visual Redundancy: Critical audio cues are supplemented with visual indicators to aid players with hearing impairments.
- Text to Speech or Narration: Converts menus or in game text into audio output, providing additional accessibility for visually impaired users.

Cognitive and UX Accessibility Solutions:

- Consistent and Clear UI Layouts: Reduces complexity and helps users with cognitive or learning impairments navigate interfaces effectively.
- Tutorial and Assistance Modes: Provides step by step guidance to familiarize players with mechanics at a comfortable pace.
- Motion Reduction Options: Eliminates or reduces camera shake, rapid flashes, and other visual effects that could cause sensory overload.

Engine Level Accessibility Tools

Modern game engines provide tools to assist developers in implementing accessibility:

- Unity: *While Unity does not include a dedicated accessibility toolkit, its UI and input systems allow developers to create scalable interfaces, remappable controls, and visual/audio feedback systems. Developers often combine built in features with custom scripts or third party plugins(Unity Support)*[11].
- Unreal Engine: *Unreal supports screen readers for UI elements and provides APIs for integrating accessibility features such as subtitles, input remapping, and visual indicators. These tools allow developers to implement inclusive features that improve usability for a variety of impairments(Unreal Documentation)*[12].

1.1 Review of Existing Accessibility-Focused Games

[11] Unity Support - https://support.unity.com/hc/en-us/articles/31056643922068-Can-I-implement-accessibility-features-in-my-Unity-game(accessed: 31 January 2026).

[12] Unreal Documentation - https://dev.epicgames.com/documentation/en-us/unreal-engine/supporting-screen-readers-in-unreal-engine (accessed: 31 January 2026).

Accessibility in video games has developed over many decades, with some titles pioneering features that later became industry standards. This section highlights selected games.

Zork: Grand Inquisitor - Early Subtitle Implementation:

Zork: Grand Inquisitor(1997) was one of the first computer games to include true closed captioning so that the hearing impaired could play without missing any of the sound effects and spoken dialog in the game[13]. As video games transitioned from purely text-based narratives to fully voiced audio experiences, hearing-impaired players increasingly faced barriers to accessing important plot information. The introduction of subtitles in this game addressed these barriers by offering a text representation of spoken dialogue, providing better accessibility to the narrative. This implementation is an important early step in the development of accessibility practices and laid the foundation for subtitles to become a standard feature in narrative-driven video games.

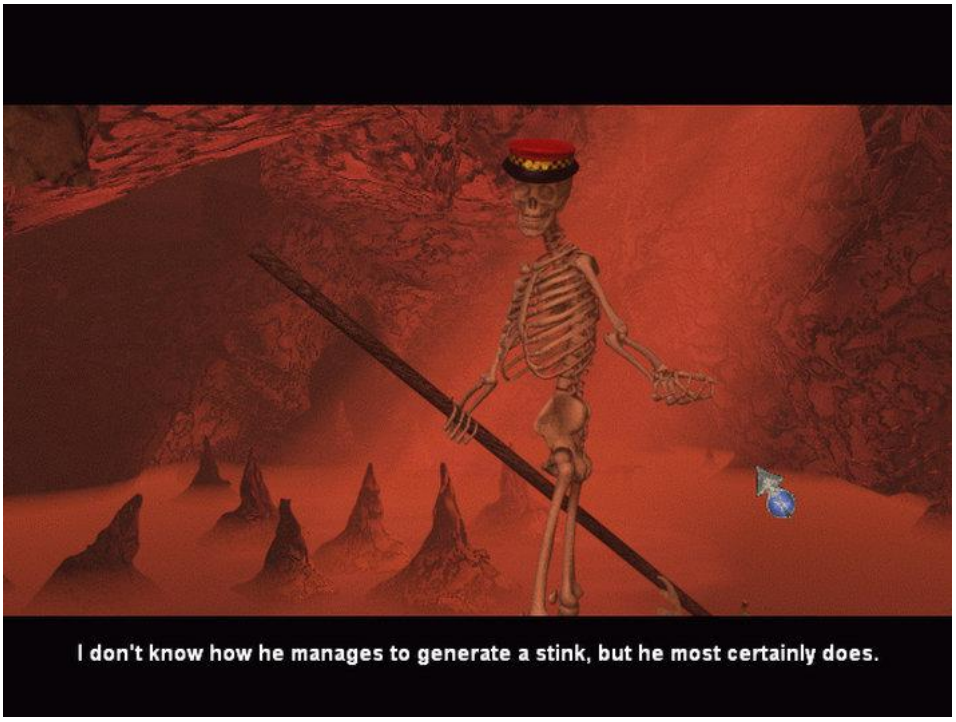

Figure 1.1 Example Game (Zork: Grand Inquisitor), date 01.01.2026

Source:< https://store.steampowered.com/app/570680/Zork_Grand_Inquisitor **>**

Audio Accessibility Solutions Example: Gears 5:

Gears 5 serves as a prominent example of audio accessibility in a mainstream game. Beyond enhanced subtitle support, the game integrates audio-based gameplay cues that communicate spatial and navigational information. A key example is the navigational

[13] Wikipedia Zork: Grand Inquisitor - https://en.wikipedia.org/wiki/Zork:_Grand_Inquisitor (accessed: 31 January 2026).

ping feature, where a beeping sound increases in volume and frequency as the player approaches significant objectives, helping players locate goals without relying solely on visual or textual indicators. *Enabling Fabricator Ping will cause the fabricator in Horde to emit a ping sound while in tac-com. The ping will increase in volume as you approach, making it a valuable navigational tool*[14].

Additionally, varied audio feedback for combat and interactive events supports comprehension of game states for players using assistive audio strategies or alternative sound processing. These design elements illustrate how complex game systems can incorporate auditory accessibility to benefit players with diverse sensory needs.

Figure 1.2 Example Game (Gears 5), date 01.01.2026

Source:< https://www.gearsofwar.com/games/gears-5/accessibility/ **>**

NeuroRacer (2013) - Cognitive Function Game:

According to a 2013 study published in the journal Nature by neuroscientists at the University of California, San Francisco[15]: NeuroRacer is a scientifically developed cognitive training game that represents an important milestone in the use of video games for cognitive accessibility and neuropsychological research. Developed by

[14] Family Gaming Database, Gears 5 Accessibility Report - https://www.familygamingdatabase.com/accessibility/Gears+5 (accessed: 31 January 2026).

[15] Anguera, J. A., et al., Video game training enhances cognitive control in older adults, Nature, vol. 501, 2013, pp. 97–101 - https://www.nature.com/articles/nature12486(accessed: 31 January 2026).

neuroscientists at the University of California, San Francisco, the game was designed primarily to study and improve cognitive control, attention, and multitasking abilities, particularly in older adults. Unlike traditional entertainment-focused games, NeuroRacer was created as a research-driven interactive system, positioning it as an early example of games intentionally designed around cognitive accessibility principles.

The gameplay of NeuroRacer combines two simultaneous tasks: a driving task that requires the player to keep a vehicle on a winding road, and a secondary visual signal task that demands quick responses to specific on-screen cues. This dual-task structure directly targets executive cognitive functions such as divided attention, task switching, and working memory. Importantly, the game dynamically adapts its difficulty based on the player's performance, ensuring that cognitive demands remain challenging without becoming overwhelming. This adaptive difficulty system is a key accessibility feature, as it allows players with varying cognitive abilities to engage meaningfully with the game.

From a cognitive accessibility and user experience (UX) perspective, NeuroRacer demonstrates how games can be designed to accommodate cognitive limitations rather than penalize them. The interface is intentionally minimalistic, avoiding unnecessary visual complexity or distracting elements that could increase cognitive load. Clear visual cues, simple controls, and consistent feedback mechanisms help players focus on the core cognitive tasks without confusion. These design choices align with accessibility principles aimed at supporting users with reduced attention span, slower reaction times, or age-related cognitive decline.

Although NeuroRacer does not address motor accessibility to the same extent as modern accessible games, its simplified control scheme reduces physical demands and prioritizes cognitive interaction over precise motor skills. This makes it accessible to a broader range of users, including those with limited gaming experience or mild motor impairments. The game's primary contribution lies in demonstrating that accessibility can extend beyond physical input and sensory accommodations to include cognitive inclusivity as a core design goal.

In the context of accessibility-focused game development, NeuroRacer is historically significant because it helped establish games as valid tools for cognitive training and assessment.. *Its technology influenced later developments, including commercial/prescribable games via Akili Interactive (e.g., EndeavorRx, FDA-approved*

in 2020 for ADHD treatment)[16].

Its success influenced subsequent research and commercial titles that incorporate adaptive difficulty, cognitive load management, and UX simplification as accessibility strategies. As such, NeuroRacer serves as an important early example of how game design can support cognitive accessibility through scientifically informed mechanics and user-centered design.

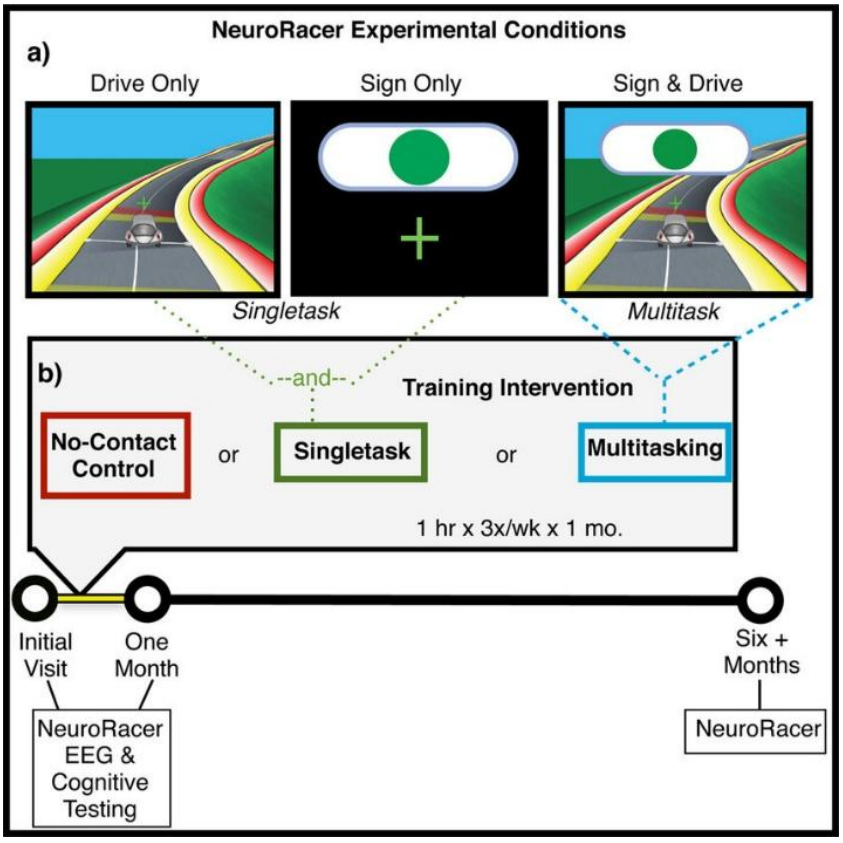

Figure 1.3 Example Game NeuroRacer (2013) >, date 01.01.2026

Source:< https://www.researchgate.net/figure/NeuroRacer-experimental-conditions-and-training-design-a-Screen-shot-captured-during_fig1_355126978**>**

1.2 Analysis of Disability Simulation Games

Disability simulation games are a subgenre of video games designed specifically to provide users with experiences that simulate the challenges faced by individuals with various disabilities[17].These games serve as both a form of education and empathy-

[16] Adam Gazzaley's biography: therapeutic game based on Neuroracer technology, was FDA-cleared as a prescription treatment for children with ADHD - https://gazzaley.com/adams-story(accessed: 01 February 2026).

[17] Disability Simulation games definition, Atypical advantage Simulation Games Workshop - https://atypicaladvantage.in/atypical-workshops/simulation-games-workshop (accessed: 01 February 2026).

building, offering players the opportunity to experience the world through the perspective of someone with a specific disability. Such games aim to foster understanding, raise awareness, and promote inclusivity within the gaming community. This chapter analyzes the design principles, objectives, and impact of disability simulation games.

Design Objectives of Disability Simulation Games - *Disability Simulation games aim at giving all the participants a better understanding of the real-life experiences of persons with disability and help to generate empathy and inclusive attitude towards persons with disability*[18].

Unlike traditional accessibility tools, which focus on providing players with disabilities the means to play conventional games, disability simulation games aim to engage a broader audience in understanding and experiencing disability firsthand.

The immersive nature of disability simulation games relies heavily on environmental design, sensory cues, and (UI) User interfacea djustments that reflect the player's limited ability. This unique design approach aims to evoke empathy by highlighting the complexities of day-to-day life that are often overlooked in mainstream game environments.

Empathy and Awareness through Gameplay - Primary objectives of disability simulation games are to foster empathy and awareness, by confronting players with challenges, encourage the player to experience frustration, limitations, and moments of triumph, which leads to better understanding of disability. *These games tend to be highly effective at helping participants empathize with people with disabilities and inspiring teams to prioritize digital accessibility*[19].

Case Study 1: *The Vale: Shadow of the Crown* (2020):

A noteworthy example of a disability simulation game is *The Vale: Shadow of the Crown* (Falling Squirrel, 2020) a narrative-driven action game designed specifically for visually impaired players. *The Vale is an interactive audio game for visually impaired and sighted gamers alike. The game is a story driven, action adventure with combat and exploration mechanics that utilize 3D spatial audio and haptic controller feedback*[20].

---

[18] Disability Simulation games purpose Atypical advantage Simulation Games Workshop - https://atypicaladvantage.in/atypical-workshops/simulation-games-workshop (accessed: 01 February 2026).

[19] Level Access, "Rethinking Disability Simulation Activities: Moving from Empathy to Informed Action, blog post,- https://www.levelaccess.com/blog/rethinking-disability-simulation-activities-moving-from-empathy-to-informed-action (accessed: 01 February 2026).

[20] The Vale: Shadow of the Crown (2020**)** Game Description Itch.io - https://falling-squirrel.itch.io/the-vale (accessed: 01 February 2026).

Figure 1.4 Example Game *The Vale: Shadow of the Crown* (2020), date 01.01.2026

Source:<https://store.steampowered.com/app/989790/The_Vale_Shadow_of_the_Crown/**>**

Core Design Features - In *The Vale: Shadow of the Crown*, players control a blind protagonist who embarks on a quest within a fantasy world. The gameplay primarily focuses on narrative exploration and combat through audio cues.Players navigate through soundscapes, interacting with the environment by listening for sounds such as footsteps, the rustle of leaves, or distant voices. Combat, often a central element in many action games, is translated into a series of audible cues and vibrations, requiring players to rely on auditory feedback rather than visual indicators.The decision to design the game in this manner addresses several significant barriers faced by visually impaired players, where reliance on visual cues often limits participation. By focusing on auditory input, *game* offers a compelling example of entertaining and educational inclusive game design.

Impact and Reception - Upon its release, The Vale: Shadow of the Crown received positive feedback for its innovative approach to game design Critics have highlighted the way in which the game challenges traditional perceptions of gameplay, where visual experiences are considered essential*(Based on Metacritic review*[21]*)*. The game's focus on sound design and immersive audio technologies has made it one of the few mainstream examples of a game specifically designed for players with visual disabilities. *The Vale offers a unique and touching experience in which sighted players will learn the unique abilities of blind people and how they 'see' the world*[22].

[21] Metacritic review "The Vale: Shadow of the Crown," - http://metacritic.com/game/the-vale-shadow-of-the-crown/ (accessed: 01 February 2026).

[22] Luis Mauricio, The Vale: Shadow of the Crown Review, RPGamer, August 19, 2021, https://rpgamer.com/review/the-vale-shadow-of-the-crown-review (accessed: 01 February)

By embracing audio as the primary mode of interaction, Game demonstrates that there are alternative ways to create an engaging narrative and game mechanics that can resonate with players who face sensory limitations.

Figure 1.5 Example Game *The Vale: Shadow of the Crown* (2020) >, date 01.01.2026
Source:<https://store.steampowered.com/app/989790/The_Vale_Shadow_of_the_Crown/**>**

Case Study 2: *BlindSide* (2012):

Another prominent example of a disability simulation game is BlindSide (2012), a text-based adventure game designed for blind players. Like The Vale: Shadow of the Crown, BlindSide uses audio cues and narrative-based exploration as the central gameplay mechanic. However, unlike the fantasy setting of The Vale, BlindSide focuses on a dystopian environment in which the player navigates a world in complete darkness.

Core Design Features - *BlindSide simulates the experience of being blind, generating 3D audio through standard stereo headphones. This allows BlindSide's players to navigate the world using only their ears*[23]. Players control a protagonist who has lost their sight and must navigate the world by interpreting auditory signals. The game challenges players to rely entirely on their hearing and memory, simulating the process of mental mapping that visually impaired individuals might employ in real life.

Standout feature of BlindSide is the real-time speech synthesis that describes objects and environments. This allows players to create a mental map of their surroundings without visual reference, simulating the experience of navigating through the world without sight.

Reception and Educational Value - BlindSide praised for its ability to offer both

[23] Michael T. Astolfi, BlindSide: An award-winning audio-only adventure game - https://michaeltastolfi.com/blindside (accessed: 01 February 2026).

entertainment and educational value by promoting understanding of the experience of blindness. Its success lies in its simplicity and ability to communicate the importance of auditory cues in navigating the world. *Michael T. Astolfi, the creative leader of BlindSide, stated in an interview that the game was inspired by a personal experience of his co-developer, Aaron Rasmussen, who was temporarily blinded in a high school chemistry accident. Astolfi explained that BlindSide simulates the experience of blindness by generating 3D audio through standard stereo headphones, enabling players to navigate the virtual world and providing an identical gameplay experience for both sighted and visually impaired individuals*[24].

Figure 1.6 Example Game *BlindSide* (2012), date01.01.2026

Source:< http://www.blindsidegame.com/ **>**

Challenges in Disability Simulation Games - While disability simulation games can be effective in promoting empathy, they also face certain challenges in their design and reception. One challenge lies in the accuracy of the experience. *The way that these exercises simulate disability is also overly simplistic, missing the complexity of peoples'*

[24] Michael T. Astolfi, "BlindSide: An award-winning audio-only adventure game," personal website - https://michaeltastolfi.com/blindside, (accessed 01 February 2026).

real, lived experience[25]. As disability simulation games attempt to convey complex sensory experiences, they risk oversimplifying or misrepresenting the difficulties faced by people with disabilities. For example, although The Vale: Shadow of the Crown offers an engaging auditory experience, it cannot fully replicate the breadth of challenges experienced by visually impaired people in real-life situations, such as the nuances of mobility or social interactions.

Additionally, game accessibility and sustainability remain concerns. While these games are designed to increase empathy, they may not always provide the level of accessibility required for long-term engagement. As these games target a specific group, the broader gaming community may have limited exposure to their educational potential.

1.3 Conclusion

This chapter delivers a systematic review of accessibility principles and disability simulation in video games, laying the conceptual and practical groundwork for subsequent analysis and proposed design solutions. Disability is defined as a mismatch between individual capabilities and the demands of products, services, or environments, underscoring its situational and potentially universal character. Game accessibility entails the creation of adaptable interfaces and balanced interactions to support diverse perceptual, motor, cognitive, and sensory needs.

The chapter examines established accessibility solutions organized into four categories: visual (subtitles, high-contrast UI, colorblind modes, visual audio cues), motor (remappable controls, simplified inputs, adaptive hardware), audio (independent volume sliders, audio-visual redundancy, text-to-speech), and cognitive/UX (consistent layouts, guided tutorials, motion-sickness reduction). Modern game engines such as Unity and Unreal provide built-in and extensible tools to enable these implementations.

A historical and contemporary overview of key titles illustrates progress: early innovations like Zork: Grand Inquisitor (1997) with closed captioning, mainstream advancements in Gears 5 via audio navigation, and research-oriented NeuroRacer (2013), which introduced adaptive cognitive training influencing therapeutic applications.

Disability simulation games emerge as a specialized subgenre designed to cultivate

[25] "Rethinking Disability Simulation Activities: Moving from Empathy to Informed Action," blog post, May 1, 2023 - https://www.levelaccess.com/blog/rethinking-disability-simulation-activities-moving-from-empathy-to-informed-action (accessed: 01 February 2026).

empathy and awareness through immersive, first-person mechanics. In-depth case studies feature The Vale: Shadow of the Crown (2020), an audio-only action-adventure enabling shared gameplay for sighted and visually impaired users via spatial audio and haptic feedback, and BlindSide (2012), a survival horror title simulating blindness with 3D soundscapes and narrated descriptions.

The analysis concludes by addressing limitations of disability simulation games, such as oversimplification of lived experiences, incomplete replication of real-world challenges, and limited long-term accessibility and audience exposure. Overall, the chapter highlights substantial advancements in mainstream accessibility and simulation design, while advocating further refinement for enhanced representational accuracy, sustainability, and educational reach. These insights form the theoretical and empirical foundation for the design project in ensuing chapters.

2 Chapter 2

Analysis and Selection of Technologies for Accessible Game Design

This chapter analyses existing technologies and methodologies relevant to the implementation of accessibility features in video games, with a particular focus on disability simulation mechanics. It examines the characteristics of basic and advanced tools, compares current solutions, identifies the primary technology best suited to the project's objectives, and provides a detailed justification for its selection. The review establishes the technical foundation necessary for the design and development of the proposed inclusive game prototype presented in subsequent chapters.

2.1 Comparison of Current Technologies and Methodologies

As the demand for accessible video games increases, developers are looking toward both game engines and accessibility frameworks to create inclusive experiences[26]. Game engines like Unity and Unreal Engine provide the foundation for game development, while specialized accessibility frameworks and guidelines offer structures to implement features that cater to players with disabilities. This chapter compares the current technologies and methodologies for accessibility, focusing on the capabilities of Unity and Unreal Engine, and examines the role of accessibility guidelines, particularly the *Game Accessibility Guidelines (GAG)*[27].

Unity Engine - Unity is known for its flexibility, ease of use, and large developer community. It is popular for 2D, mobile, and 3D games, and offers accessibility support primarily through plugins and some built-in features[28].

Visual Accessibility (Blind/Low Vision)

- Mobile screen reader support via iOS/Android VoiceOver and TalkBack.
- Dynamic text scaling and bold fonts through Accessibility Settings API.
- Colorblind filters can be applied using post-processing shaders.
- High contrast and reduced transparency available via Apple Accessibility Plugin.

Auditory Accessibility (Deaf/Hard of Hearing)

- Closed captioning with official prefab and tutorials.

[26] AccessForge, “Unity and Unreal Engine Accessibility: Tools, Plugins, and Developer Guide,” - https://accessforge.io/accessibility-game-engines, (accessed 01 February 2026).

[27] GAG Game Accessibility Guidelines - https://gameaccessibilityguidelines.com/, (accessed 01 February 2026).

[28] Unity Technologies, “Accessibility,” Unity Manual (2023.2) - https://docs.unity3d.com/2023.2/Documentation/Manual/com.unity.modules.accessibility.html, (accessed 01 February 2026).

- Audio visualization via custom indicators.
- Per-channel volume control with Audio Mixer.
- Haptic feedback supported on gamepads and mobile devices.

Motor Accessibility (Physical/Mobility)

- Input remapping at runtime using RebindActionUI.
- Supports multiple devices and adaptive controllers.
- Switch control integration via Apple Accessibility Plugin.
- Keyboard and gamepad UI navigation through UGUI Selectable system.
- Extensible to eye tracking, voice input, and other alternative methods.

Cognitive Accessibility (Neurodiversity, Learning)

- Localization with multi-language support.
- Assist modes adjusting gameplay pace.
- Simplified UI and tutorial systems.
- Motion sensitivity settings to reduce blur, camera shake, and other effects.

Accessible functions Data from (https://accessforge.io/accessibility-game-engines)

Unreal Engine - Unreal Engine excels in high-fidelity graphics and AAA game development. While it has a steeper learning curve, it provides extensive built-in accessibility features.

Visual Accessibility

- UMG screen reader support and Slate Screen Reader plugin for full UI narration.
- Built-in color vision deficiency correction (e.g., protanopia, deuteranopia).
- High contrast UI achievable through custom UMG styles.
- Text-to-speech plugins for additional narration and audio feedback.

Auditory Accessibility

- Built-in subtitle system and enhanced captioning (UE 5.6+).
- Visualized sound effects using UMG-based directional indicators.
- Haptic feedback including adaptive triggers.
- Voice chat transcription available via plugins.

Motor Accessibility

- Enhanced Input System with runtime key remapping.
- Alternate control schemes and Co-Pilot dual-controller support.
- Full vibration and adaptive trigger integration.
- Supports alternative inputs through plugins like eye tracking or voice commands.

Cognitive Accessibility

- Time dilation and global slowdown via built-in nodes.

- Persistent objective and quest tracking through UMG.
- Simplified, toggleable UI layouts.
- Interactive tutorial systems.

  IRIS plugin for photosensitivity and flash detection Accessible functions Data from (https://accessforge.io/accessibility-game-engines)

  Comparison of Key Accessibility Features in Unity and Unreal Engine

Feature	Unity	Unreal Engine
Screen Reader Support	Native support for Windows, macOS, Android, and iOS via UnityEngine.Accessibility.	Built-in **Screen Reader Plugin** supports NVDA/JAWS on Windows and Voiceover on iOS
Text-to-Speech (TTS)	Requires third-party plugins (e.g., Read Speaker) for real-time menu.	**Native TTS Plugin** allows developers to narrate UI text and element states directly.
UI Semantic Markup	UI Toolkit uses semantic properties (Hints, Descriptions) to label elements for assistive tech.	**UMG Accessibility Settings** in the Details panel allow overriding accessible defaults for widgets.
Color Vision Deficiency	Managed through custom shaders or camera post-processing volumes.	**Built-in CVD Settings** allow real-time simulation and adjustment of color space.
Subtitles / Captions	Manual implementation via UI or through localization-integrated text systems.	**Audio Subtitle Plugin** allows globally enabling/disabling subtitles and supports multiple entries per asset.
Input & Navigation	Supports focus rings and keyboard navigation by default in UI Toolkit.	Supports keyboard/controller navigation via UMG and "Common UI" for cross-platform input
Localization	Advanced text shaping for RTL (Arabic/Hebrew) and Unicode built into UI Toolkit.	Native localization dashboard for text formatting and pluralization support.

Table 2.1 (Comparison of Key Accessibility Features in Unity and Unreal Engine)

Accessibility Frameworks: Game Accessibility Guidelines (GAG)

The Game Accessibility Guidelines (GAG) provide a comprehensive set of recommendations for developers aiming to create inclusive and accessible gaming experiences. GAG covers a wide range of accessibility concerns, including visual, auditory, motor, and cognitive impairments, and provides actionable steps for integrating these features into game development processes.[29]

Key Areas of the Game Accessibility Guidelines:

1. **Visual Accessibility**: GAG recommends features like high-contrast modes, colorblind support, subtitle customization, and text-to-speech options. These features are crucial for ensuring that games are playable for those with visual impairments. Both Unity and Unreal Engine provide tools that align with these guidelines, but Unreal's Slate UI framework provides more comprehensive built-in support (Game Accessibility Guidelines, 2021).1
2. **Auditory Accessibility**: Guidelines suggest including closed captions, sound descriptions, and auditory cues to assist players who are deaf or hard of hearing. Unreal Engine offers spatialized audio and other auditory enhancements to ensure these features can be implemented effectively. Unity, while flexible, requires third-party solutions to achieve the same level of auditory accessibility (Seddon et al., 2021).
3. **Motor Accessibility**: The GAG outlines how games should allow for customizable controls and support for adaptive controllers. Unreal's Blueprint Visual Scripting system simplifies the implementation of rebindable controls and other motor accessibility features, while Unity developers must rely on external assets or manual coding to achieve similar functionality.
4. **Cognitive Accessibility**: GAG also emphasizes the importance of simplified UI, clear instructions, and adjustable difficulty levels for players with cognitive disabilities. Both engines allow for the implementation of these features, Unreal Engine's Blueprint system makes it easier for developers to integrate them without extensive programming knowledge.

Summary of Chapter 2.1

Both Unity and Unreal Engine offer robust support for creating accessible games, with each engine providing different strengths in terms of built-in features and customization options. **Unreal Engine** stands out for its comprehensive suite of accessibility tools, including spatialized audio, customizable UI, and visual scripting through Blueprints. It also aligns closely with Game Accessibility Guidelines (GAG), making it an attractive choice for developers focused on accessibility.

[29] (GAG) GameAccessibilityGuidelines - https://gameaccessibilityguidelines.com/, (accessed 01 February 2026).

Unity provides flexibility and ease of use, with a rich selection of third-party plugins available to enhance accessibility. However, it requires more effort to implement advanced accessibility features compared to Unreal, especially for visual and auditory feedback systems.

The Game Accessibility Guidelines (GAG) serve as a critical framework for both engines, offering actionable steps to ensure inclusivity. By following these guidelines, developers can create games that are accessible to a wider audience, enhancing the gaming experience for players with disabilities.

2.2 Characteristics of Used Technologies

This section describes the technologies selected for the development of the game prototype, with a focus on their technical capabilities and relevance to the implementation of accessibility features and disability simulation mechanics.

Game Engine: Unity - Unity was selected as the primary development platform due to its cross-platform capabilities, extensive documentation, and modular architecture. The engine supports the creation of interactive experiences across multiple devices and is particularly well-suited for projects requiring custom visual, auditory, and input modifications. Unity's scripting system, based on C#, enables precise control over game logic, user interface elements, and environmental rendering, which are essential for implementing dynamic disability simulations[30].

Programming Language: C# - C# serves as the core scripting language within Unity. It provides robust object-oriented programming features, strong type safety, and seamless integration with *Unity's API*[31]. This combination facilitates the development of complex behaviours, such as adaptive input processing, real-time visual effects, and conditional audio management, all of which are necessary for the simulation mechanics employed in the prototype.

Accessibility APIs and Adaptive Hardware Support - Unity incorporates support for accessibility through its built-in modules and third-party extensions. The Accessibility module (available in recent versions) enables basic integration with screen readers on mobile platforms (Android and iOS) and offers limited desktop compatibility. For

[30] Unity Technologies, “Scripting,” Unity Manual - https://docs.unity3d.com/Manual/ScriptingSection.html, (accessed 01 February 2026).

[31] Microsoft, “C# Documentation” - https://learn.microsoft.com/en-us/dotnet/csharp/, (accessed 01 February 2026)

advanced requirements, Unity supports custom input mapping, allowing compatibility with adaptive controllers such as the Xbox Adaptive Controller. These features enable developers to accommodate users with motor impairments by remapping controls and adjusting interaction thresholds.

Unity Accessibility Toolkit (UAT) - *The Unity Accessibility Toolkit provides a set of components designed to enhance usability for players with disabilities. Key functionalities include screen reader support, dynamic colour adjustments, font scaling, keyboard navigation, and focus indicators for UI elements*[32]. These tools were utilised to ensure baseline accessibility in menus and interface components, while custom scripts extended their capabilities to support the simulation-specific requirements of the prototype.

Adaptive Controllers *Support for adaptive hardware, such as the Xbox Adaptive Controller, is integrated through Unity's input system*[33]. This allows custom mapping of buttons, triggers, and analog inputs to accommodate users with limited mobility or alternative control preferences. In the project, this compatibility was leveraged to demonstrate how the prototype can be adapted for diverse input methods, aligning with inclusive design principles.

Simulation Features Developed in the Prototype The prototype implements five distinct simulation levels, each designed to replicate a specific disability through gameplay constraints. These constraints function as intentional modifications to the visual, auditory, or input environment, creating an experiential understanding of associated challenges.

- **Level 1 - Colour Vision Deficiency**: The entire scene is rendered in grayscale with contrast adjustments applied via a custom shader, simulating common forms of colour blindness. Players must rely on non-colour cues (shape, pattern, audio) to progress.
- **Level 2 - Auditory Impairment**: Audio output is filtered to simulate distant or muffled sound, with critical cues delayed or reduced in volume. Players must adapt to limited auditory information.
- **Level 3 - Partial Blindness**: The visible screen area is restricted to approximately

[32] Unity Technologies, "UI Toolkit," Unity Manual - https://docs.unity3d.com/Manual/UIElements.html, , (accessed 01 February 2026).

[33] Microsoft, "Xbox Adaptive Controller Developer Guide," - https://developer.microsoft.com/en-us/games/xbox/accessibility/xbox-adaptive-controller/, (accessed 01 February 2026).

half the display, replicating a reduced visual field. Navigation and interaction are limited to the unobstructed portion.

- **Level 4 - Motor Limitations**: Selected input keys and mouse controls are disabled or delayed, simulating reduced motor precision or limb restriction. Players must complete tasks with restricted control options.
- **Level 5 - Combined Blindness and Partial Deafness**: The screen is fully blacked out, and audio is heavily distorted or absent. Gameplay relies on haptic feedback, minimal spatial memory, and residual cues.

These features were implemented using Unity's shader system, audio mixer, input management API, and UI Toolkit, demonstrating the engine's capacity for precise, real-time modifications required for effective disability simulation.

2.3 Justification for Choice of Technology

Unity was selected as the primary development platform after evaluating its suitability against the project's specific requirements for disability simulation and educational intent.

Community Support Unity benefits from one of the largest and most active developer communities in the game industry. *Extensive forums*[34], official documentation, tutorials, and open-source accessibility assets provide reliable support for implementing non-standard features. This ecosystem proved essential during the development of custom simulation mechanics that extend beyond built-in accessibility tools.

Accessibility Features and Extensibility Unity supports a wide range of accessibility implementations through its *scripting API, UI Toolkit*[35], and integration with third-party plugins. Key advantages include:

- Custom shader support for real-time visual alterations (e.g., grayscale rendering, contrast manipulation).
- Flexible UI modification capabilities, enabling dynamic scaling, focus management, and semantic labelling.
- Native input remapping and compatibility with adaptive hardware, facilitating inclusive control schemes.

Suitability for Disability Simulation The project's core objective required the ability to

[34] Unity Forums, "Accessibility Forum,"- https://forum.unity.com/forums/accessibility.148/, (accessed 01 February 2026).

[35] Unity Technologies, "UI Toolkit," Unity Manual - https://docs.unity3d.com/Manual/UIElements.html, (accessed 01 February 2026).

create highly customised, level-specific simulations of visual, auditory, and motor impairments. Existing accessibility solutions and frameworks primarily address general usability (e.g., screen readers, subtitle support, input remapping) but do not offer ready-made tools for intentional, gameplay-integrated disability simulation. Unity's modular architecture and scripting flexibility allowed complete control over the user experience, including the application of targeted constraints (colour inversion, audio filtering, input restriction) for educational purposes.

Conclusion for Chapter 2

Unity was chosen for its combination of accessibility support, extensibility, community resources, and ability to implement bespoke simulation mechanics without reliance on pre-existing specialised tools. These characteristics enabled the development of a prototype that effectively balances technical feasibility with the educational goal of fostering empathy and awareness through experiential gameplay. The decision reflects a deliberate balance between practical development constraints and the project's innovative requirements.

3 Chapter 3

Project Description

This chapter presents the conceptual and technical design of the game prototype developed as the practical component of the thesis. It outlines the underlying assumptions, design decisions, and detailed specifications that guided the implementation process. The chapter is structured to provide a comprehensive overview of the project's architecture and functionality, beginning with the overall design document, followed by an in-depth examination of the accessibility features and simulation mechanics, and concluding with a user manual to facilitate understanding and potential future use of the prototype.

3.1 Design Document

1. Game Concept:

The game is designed as a visual novel in which each chapter presents a self-contained story focused on a different character living with a disability. The core concept is to combine narrative storytelling with interactive mechanics to both educate the player and create emotional empathy. Each chapter explores the personal experiences, challenges, and feelings of the character, allowing the player to see the world from their perspective. Beyond narrative elements, every chapter includes unique minigames that simulate aspects of the featured disability through gameplay mechanics. For example, in a chapter centered on a deaf character, sound may be distorted, distant, or completely absent, directly affecting player interaction. By integrating storytelling and gameplay, the project aims to raise awareness of disabilities while demonstrating how game mechanics can be used to communicate subjective human experiences.

2. Game Objectives

- Raise awareness of accessibility challenges faced by people with different disabilities by presenting their experiences through interactive storytelling.
- Encourage empathy and understanding by allowing players to experience the world from the perspective of characters with disabilities.
- Explore innovative game design approaches in which disabilities are represented not only narratively but also through gameplay mechanics.
- Use interactive challenges and minigames to simulate sensory and cognitive limitations, transforming them into meaningful gameplay experiences.

- Demonstrate how games can function as educational tools while remaining engaging and emotionally impactful for players.

3. Key Features

- Multiple chapters or levels, each representing a different disability and presented through a unique character and narrative scenario.
- Visual novel–style storytelling that emphasizes character development, emotional depth, and player choice.
- Disability-specific gameplay mechanics and minigames that simulate sensory or perceptual limitations, such as reduced or absent sound, altered visuals, or delayed input.
- Integration of narrative and gameplay, where mechanics directly support the story and emotional experience of each character.
- Educational focus combined with interactive design, ensuring that learning objectives are embedded naturally within the gameplay experience.

4. Accessibility Features:

- **Visual Simulation:** Grayscale or color-reversed palettes for colorblindness.
  Half, or nothing showed on screen to simulate blindness.
- **Auditory Simulation:** Reduced or delayed sound cues for auditory impairments.

5 Disability Simulation Modes

Level	Simulated Disability	Technical Implementation	Purpose
1	Visual (Colorblindness)	Grayscale shader, color inversion script	Demonstrates challenges with color perception
2	Auditory Impairment	Muted audio tracks, delayed sound cues	Demonstrates reduced access to auditory information.
3	Visual (partial blindness)	Only Half of the screen is visible	Demonstrates limitations of the visual field.
4	Motor limitations	Some keyboard key or mouse clicks are limited	Demonstrates restricted motor control.
5	Visual/Auditory (Blindness/ partial deafness)	Screen is black, sounds are distant, distorted)	Demonstrates navigation with severely limited sensory input.

Table 3.1 (Description of levels and disabilities presented in them)

6 User Interface (UI) Design

The user interface is designed to be minimalistic and intuitive to support narrative immersion and accessibility. UI elements such as dialogue boxes, menus, and interaction prompts are kept visually clear and unobtrusive, allowing players to focus on the story and gameplay. The interface adapts to each chapter's theme and mechanics, subtly changing to reflect the featured disability while maintaining overall usability. Accessibility considerations include readable typography, clear visual hierarchy, and consistent navigation across all chapters. The UI design aims to balance functional clarity with emotional storytelling, ensuring that interface elements enhance rather than distract from the player's experience.

Figure 3.1 Project Game (Bondless Colors)

Source:<Own research> date 11.01.2026

6 Flow and Gameplay

The game begins at Level 1, which presents the story of a colorblind character, Neil. In this level, the game automatically applies grayscale and inverted color settings throughout the entire interface, including the main menu (Look Photo 7). to simulate the character's visual experience. Players are required to complete each level before progressing to the next; subsequent story-levels remain locked until the previous one is finished.

Upon completing a level, the game automatically adjusts settings to simulate the next character's disability. For example, progression from Level 1 to Level 2 involves transitioning from Neil's colorblindness (grayscale, inverted colors) to a deaf character,

Soundiel, where colors are presented normally, but audio is distorted or absent. This design allows players to gradually experience abilities and perceptions that were inaccessible in earlier levels, such as distinguishing colors that were previously obscured. Completing all levels enables the player to gain a comprehensive understanding of each character's life experience. Each new perspective contributes to a fuller, more complete narrative, combining **educational value** with **emotional engagement**, as the game is based on real-life stories and prototypes of individuals with disabilities.

Figure 3.2 Project Game (Bondless Colors)

Source:<Own research> **date 11.01.2026**

7 System Requirements:

- **Operating System:** Windows 10 or higher
- **Processor**: Intel Core i3 (2nd generation) or equivalent
- or AMD Athlon II / FX-series or better
- **Memory:** 4 GB RAM
- **Graphics:** Integrated graphics (Intel HD 4000 or equivalent)
- DirectX 10 compatible GPU
- **Storage:** 2 GB free space
- **Input Devices:** Keyboard and mouse

8 Installation and Launch Guide:

Step-by-step guide on how to install and run the game.

Since the game auto-adjusts, no settings configuration instructions are required

Instructions:

1. Download the game folder to your computer.
2. Open the executable file BoundlessColors.exe.
3. The game will automatically apply Current level accessibility options, (Default Level 1 - monochromacy).
4. Click ‘Start’ Button to level selection menu. Click on character image or name on the image to start chosen characters story (By default unlocked only 1 Character Neil, player must finish whole story ‘Level’ to open next characters Story.
   After finishing story next characters accessibility configuration would automatically apply for game).
   The game has started, no further action is required.
   Additional Note: For evaluation, demonstration, and testing during development, the prototype includes a level-selection utility accessible via the Help menu.
   To access this feature:
   - Navigate to the Help menu in the main interface.
   - Two buttons are displayed: Level 1 and Level 2.
   - Selecting either button immediately changes the current level to the corresponding one.

   Please note that using this utility bypasses normal progression and does not trigger the automatic unlocking of achievements or completion states associated with finishing a level route through standard gameplay. This functionality is provided exclusively for testing and demonstration purposes and should not be considered part of the intended player experience.

3.2 Detailed Design of Accessibility Features and Simulations

The game incorporates accessibility features and sensory simulations in each level to allow players to experience the challenges faced by individuals with various disabilities. Each level is designed around a specific disability, with technical implementations that replicate the sensory or motor limitations associated with that condition. The goal is to provide both educational and emotional insights into the lived experiences of people with disabilities, while integrating these simulations seamlessly into gameplay.

Level 1: Visual (Colorblindness)

In the first level, the game simulates colorblindness using a combination of a grayscale shader and a color inversion script. These visual adjustments apply to all game elements,

including the main menu and in-game environment. This simulation allows players to experience the difficulties associated with distinguishing colors and recognizing visual cues that rely on color perception, demonstrating the challenges faced by individuals with color vision deficiencies. As additional challenge minigame to find right color is provided to player through sound - Each sound represents color and player must find colors according to sounds. This minigame based on real person experience *Neil Harbisson, a colorblind man who uses physical instrument which tells colors by its frequency to Neil with different sounds*[36].

Figure 3.3 Project Game (Bondless Colors)

Source:<Own research> **date 11.01.2026**

Level 2: Auditory Impairment

The second level focuses on auditory impairments. To simulate reduced hearing, the game employs muted audio tracks and delayed sound cues, which affect both environmental sounds and character interactions. This design demonstrates the difficulties of perceiving and interpreting auditory information, emphasizing how sound limitations impact gameplay and daily experiences.

Level 3: Visual (Partial Blindness)

In the third level, partial blindness is simulated by restricting the visible portion of the screen to only half of the normal display area. This mechanic highlights the limitations of

[36] Neil Harbisson, "The man who hears colour," interview by Katia Moskvitch, BBC News Magazine, February 27, 2012, https://www.bbc.com/news/magazine-16681630, (accessed: 31 January 2026).

the visual field, requiring players to navigate and interact with the game environment while experiencing constrained vision. This approach demonstrates the impact of restricted sight on spatial awareness and situational perception.

Level 4: Motor Limitations

Level four addresses motor limitations by limiting the player's ability to use certain keyboard keys or mouse clicks. This simulation replicates reduced motor control, slowing interaction speed and increasing the difficulty of performing precise in-game actions. It allows players to understand the challenges of engaging with standard input devices when motor abilities are impaired.

Level 5: Visual/Auditory (Blindness / Partial Deafness)

The fifth level combines visual and auditory impairments to simulate blindness and partial deafness. The screen is completely black, and audio is either distant, distorted, or absent. This design demonstrates the difficulty of navigating an environment with severely limited sensory input, emphasizing the reliance on remaining senses and alternative cues to complete tasks.

Conclusion:

Through these targeted simulations, the game provides players with progressive experiences of sensory and motor limitations, enabling a deeper understanding of the challenges faced by people with disabilities. Each level not only conveys the functional impact of a specific disability but also encourages empathy and reflection, making the gameplay both educational and emotionally engaging.

3.3 Project User Manual

Purpose:

- Explain what the game is and its objective.
- Clarify the target audience: healthy/able-bodied users.
- Highlight the educational goal: understanding the perspective of people with disabilities.

This game is an educational simulation designed to allow healthy users to experience the challenges faced by people with various disabilities. Each level of the game simulates a different impairment, such as visual, motor, or auditory limitations, automatically adjusting gameplay to reflect these experiences. The primary purpose of this project is to increase awareness and foster empathy toward accessibility challenges faced by individuals with disabilities by presenting these challenges through a realistic and experience-based

perspective. The project aims to demonstrate how limitations in accessibility affect interaction, navigation, and decision-making within digital environments.
In addition, the project seeks to illustrate that accessibility features should not be viewed solely as compensatory or auxiliary elements, but as integral components of thoughtful and engaging game design. By embedding accessibility mechanics directly into gameplay, the project explores how inclusive design can enhance player engagement, immersion, and narrative depth while remaining accessible to a wider audience.

Game Overview:

The game consists of five levels, each simulating a different disability. Players navigate through interactive visual novel story and minigames designed to demonstrate the challenges faced by people with impairments. All accessibility simulations are applied automatically, and players experience the limitations firsthand without having to adjust any settings.

Level Descriptions:

Level 1: The first level tells the story of 'Neil' an artist suffering from achromatopsia. The player is invited to see the world in black and white and to touch on the experiences and problems of people born with such disabilities. A person with this disability not only cannot recognize colors, but is also sensitive to light, making it difficult for them to look at bright colors. Despite this, there are famous artists in history who created works in color, even though they themselves could never see the "colors" of their paintings. The character is based on the artist *Neil Harbisson*[37], who attached a physical device to his head that reads the frequency of colors and transmits beeps to Neil to tell him what color is in front of him.

Level 2: The second level tells the story of 'Soundiel' a deaf singer. In today's world, we are surrounded by the sounds of cars, animals, people, electronic devices; sound is practically inescapable... Music has become a salvation for many. How painful it must be for those who have devoted their lives to music to lose their hearing. However, there are musicians who have created global hits, even though they themselves cannot hear because they have lost their hearing. For the love of music, they have resorted to various tricks, such as memorizing the rhythm of a song by watching other singers' lips. The story is based on the author's personal memories of a person with deafness and on the story of *Mandy Harvey*[38].

---

[37] Neil Harbisson, "The man who hears colour," interview by Katia Moskvitch, BBC News Magazine, February 27, 2012 - https://www.bbc.com/news/magazine-16681630, (accessed: 31 January 2026).

[38] Tanya Dalton, "Episode 102: How Mandy Harvey Turns Tragedy into Triumph," The Intentional Advantage (podcast), March 10, 2020 - https://tanyadalton.com/podcasts/episode-102-how-mandy-harvey-turns-tragedy-into-triumph/ (accessed: 01 February 2026).

Level 3: The third level tells the story of 'Elias' a police officer who lost his eye and replaced it with a synthetic cyber eye. Although this may sound like a plot from a book about the future, this story was inspired by a real-life story from our present day. *Rob Spence* [39] is a man who has a camera instead of an eye. Unfortunately, our technology has not yet advanced to the point where a "cyber" eye looks natural; it looks quite terrifying, like many other devices that exist to improve the lives of people who have suffered trauma. Although these devices improve lives, they are not yet capable of fully replacing a human organ, their price is very high, and their appearance can be frightening...

Level 4: The fourth level tells the story of 'Jessa' a female blacksmith. Our world is full of injustice and inequality. Despite all our laws on equality, women still face poor treatment and unequal working conditions, especially when it comes to physical labor, due to stereotypes. And for Jesse, it's even harder because she lost her arm, which complicates not only her work but also her everyday life... Many of the things we buy, such as flat-pack furniture, are designed in such a way that we must hold a part with one hand and screw it in with the other. When creating such instructions and designs, we don't even think about people who only have one hand, for example. This story is based on the author's memories of war veterans who lost limbs and the story of *Jessica Smith*[40].

Level 5: The fifth level is the culmination of all the previous stories from the perspective of a blind, deaf old woman. We don't need eyes to see the suffering of others, or ears to hear their cries. Even though everyday life is difficult for 'Zia Notte', she treats the previous characters as her children and watches over them. Sometimes bright desires can create miracles, and even the blind can see. The character is inspired by many people, as well as the story of *Daniel Kish*[41], who taught himself to make waves with his tongue like bats to use echo location.

Game Controls:

- **Interaction:** Spacebar / Left mouse button
- **Exit:** Esc key

---

[39] Rob Spence, "The man with a camera inside his eye," interview by Frank Swain, BBC Future, July 21, 2017 - https://www.bbc.com/future/article/20170721-the-man-with-a-camera-inside-his-eye, (accessed: 01 February 2026).

[40] Jessica Smith, interview by Libby-Jane Charleston, "Jessica Smith on life as a new mum with one hand," Mamamia, October 15, 2019 - https://www.mamamia.com.au/jessica-smith-interview, (accessed: 01 February 2026).

[41] Daniel Kish, "The eyeless man taught himself to see," interview by Youth Time Magazine, December 12, 2018 - https://youthtimemag.com/the-eyelessman-taught-himself-to-see/, (accessed: 01 February 2026).

- Minigame keyboards buttons which are assigned for minigames are shown during those minigames.
- **Additional** quick buttons for muting/unmuting game, quick save etc., could be seen as well as configured on the option menu.

  Note: Input and controls are affected by the simulation automatically. Players will notice differences in responsiveness depending on the level being played.

  Learning Outcomes
- Gain awareness of the challenges faced by people with visual, motor, or auditory impairments.
- Understand the importance of accessibility design in games and digital media.
- Develop empathy and a user-focused perspective for inclusive design.

  Troubleshooting:
- Ensure your system meets the minimum requirements.
- Verify that all game files are in the same folder.
- Restart the game if any input or audio issues occur.
- *Deleting any files in saving folders would lead to corruption of save file.

4 Chapter 4

Implementation of the Thesis Project

Project Source Code and Web Deployment

The complete source code for the game prototype developed as part of this diploma thesis is hosted in a separate public GitHub repository. The repository linked below contains the full Unity project, including all scripts, scenes, prefabs, assets, UI elements, shaders, audio files, and configuration settings required to open, build, and modify the prototype using the Unity Editor.

Primary Source Code Repository https://github.com/Anatolii-Shcherbak/Accesability-Novel

Delivery Archive Repository (Builds, Documentation, and Screenshots)

https://github.com/Anatolii-Shcherbak/DyplomWork_AnatoliiShcherbak_Id63570

This archive repository serves as the official submission package and includes:

- **BuildFinal.zip** — Final Windows executable build of the visual novel prototype.
- **Thesisi_Anatolii_Shcherbak_63570.pdf** — The full diploma thesis document.
- A series of screenshots (Photo 8 to Photo 23) documenting key gameplay features, accessibility simulations (e.g., Bondless Colors mode, level comparisons, Spray minigame), saved data handling, and UI elements.

WebGL Browser Build A WebGL version of the prototype is hosted for immediate evaluation and demonstration. For best performance and correct rendering of simulation features (particularly visual and audio constraints), it is recommended to play in **full-screen mode**.

Web build link: https://anatolii-shcherbak.github.io/DyplomProjectWork_AnatoliiShcherbak/

(Note: Mobile support is not implemented; desktop browsers are recommended. The Quit button is intended only for the desktop executable and should not be used in the web version.)

Non-Original Assets All core code, mechanics, narrative content, UI layouts, and simulation features were created specifically for this project. Placeholder text drafts, conceptual descriptions, and certain UI strings were generated with the assistance of large language models (GPT-based tools) during early ideation and documentation phases. No third-party commercial graphical, audio, or pre-existing asset packs were used in the final prototype.

This repository structure and deployment configuration ensure traceability, reproducibility, and accessibility for evaluation purposes. The separation between the source code repository (development files) and the delivery archive (builds and documentation).

4.1 Development Process

The development of the accessibility simulation game followed an iterative approach inspired by Agile methodologies. Each feature of the game was developed in small increments, allowing continuous testing and refinement. Version control was maintained using Git, which enabled tracking changes, managing different versions of the project, and ensuring code stability.
The development process included:

1. **Planning:** Defining game objectives, levels, and disability simulations, researching materials about people with disabilities to create a template for stories and characters
2. **Implementation:** Developing core mechanics in Unity, creating shaders, scripts, and UI components for accessibility features.
3. **Testing:** After each level or feature was implemented, functionality was tested to ensure it correctly simulated the intended disability.
4. **Iteration:** Adjustments and improvements were made based on testing results.

Git repositories were used to manage source code, allowing rollback to previous versions if necessary and ensuring that all changes were documented systematically.

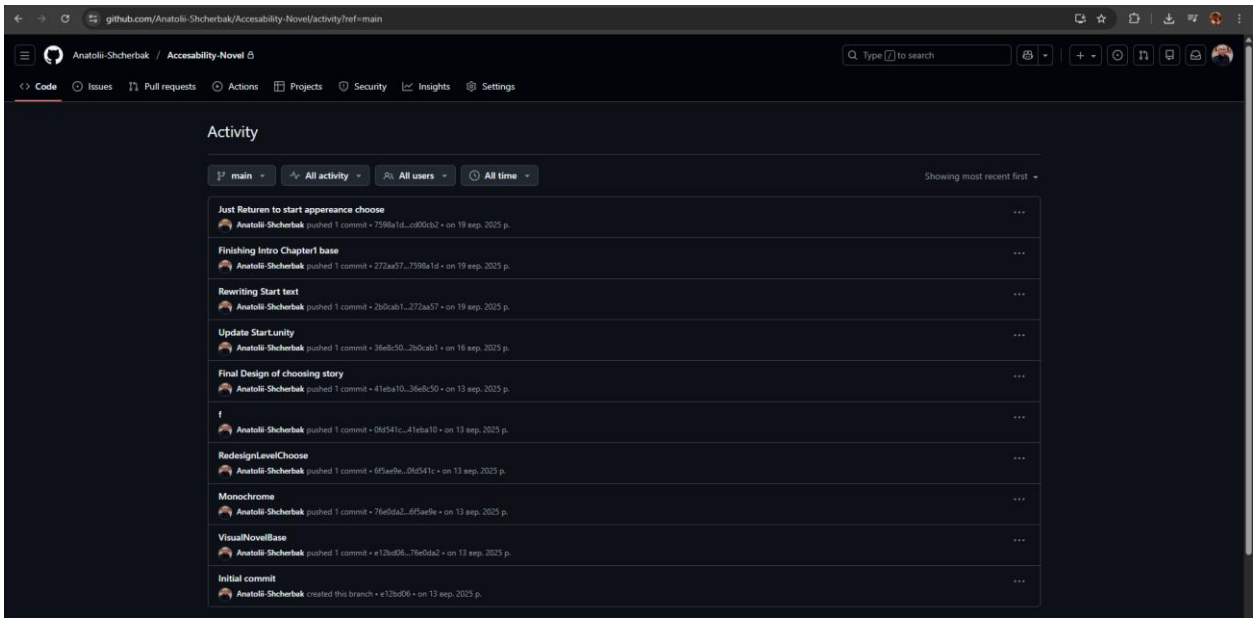

Figure 4.1 Git Control Project Game (Bondless Colors)

Source:<Own research> date 11.01.2026

4.2 Coding and Integration

The game is a visual novel, So, first, a complete foundation for a visual novel was developed, namely a system for parsing text from a TXT file to display lines and names on the screen, as well as commands such as change: scenes, character images, locations,

and music… In other words, the plot is written in a text file rather than in the code, game menu. Implemented save and load system, achievement system, and some other standard elements for visual novels that are still in development. Game was implemented entirely in Unity, with scripts and shaders used to create disability simulation modes. Each level was designed as a one scene, with using shaders and scripts specifically for level from settings script with specific features integrated to simulate a different disabilities.

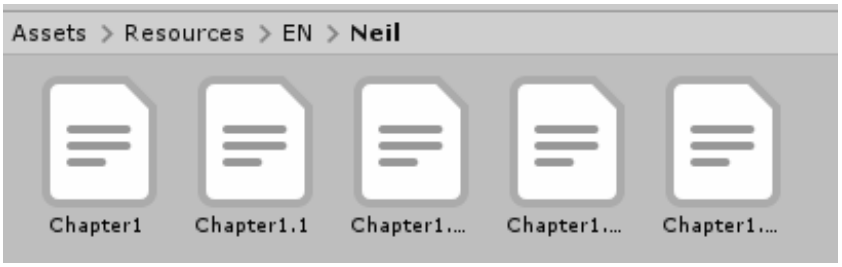

Figure 4.2 Game Chapters Source:<Own research>, date 20.01.2026

```
38
39          database.AddCommand("moveChar", new Func<string[], IEnumerator>(MoveCharacter));
40
41          database.AddCommand("Chscene", new Func<string, IEnumerator>(Chscen));
42
43          database.AddCommand("Chchapt", new Func<string, IEnumerator>(Chchapt));
44
45          database.AddCommand("Chchar", new Func<string[], IEnumerator>(Chchar));
46
47          database.AddCommand("Chmus", new Func<string, IEnumerator>(Chmus));
48          database.AddCommand("Chsou", new Func<string, IEnumerator>(Chsou));
49          database.AddCommand("UnmMus", new Func<IEnumerator>(UnmMus));
50          database.AddCommand("Chovervoice", new Func<string, IEnumerator>(Chovervoice));
51
52          database.AddCommand("Invischar", new Func<string, IEnumerator>(Invischar));
53
54          database.AddCommand("unact", new Func<string, IEnumerator>(UnActiveCharacter));
55
56          database.AddCommand("Hide", new Func<string, IEnumerator>(Hide));
57          database.AddCommand("show", new Func<string[], IEnumerator>(Show));
58
59          database.AddCommand("Chtext", new Func<string[], IEnumerator>(Chtext));
60          database.AddCommand("GetAchiv", new Func<string, IEnumerator>(GetAchiv));
61          database.AddCommand("GetAchiv2", new Func<string, IEnumerator>(GetAchiv2));
62          database.AddCommand("GetAchiv3", new Func<string, IEnumerator>(GetAchiv3));
63
64          database.AddCommand("ChEndBut", new Func<string, IEnumerator>(ChEndBut));
65
66          database.AddCommand("QuickSave", new Func<IEnumerator>(QuickSave));
67
68          database.AddCommand("DeadReason", new Func<string, IEnumerator>(DeadReason));
69
70          database.AddCommand("ResetCharPos", new Action<string>(ResetCharPos));
71          database.AddCommand("RecolorDialogue", new Action<string>(RecolorDialogue));
72
73          database.AddCommand("NextLevel", new Func<string, IEnumerator>(NextLevel));
```

Figure 4.3 Implemented Command database

Source:<Own research> date 20.01.2026

```
138         private static IEnumerator QuickSave()
139         {
140             // Find the GameObject by its name
141             GameObject saveGameObject = GameObject.Find("SaveLoad");
142
143             if (saveGameObject != null)
144             {
145                 // Get the SaveScript component from the GameObject
146                 SaveScript saveScript = saveGameObject.GetComponent<SaveScript>();
147
148                 if (saveScript != null)
149                 {
150                     // Call the QuickSave method
151                     saveScript.PlotSave();
152                 }
153                 else
154                 {
155                     Debug.LogError("SaveScript component not found on GameObject.");
156                 }
157             }
158             else
159             {
160                 Debug.LogError("GameObject 'GameManager' not found in the scene.");
161             }
162             yield return null;
163         }
                1 reference
164         private static IEnumerator DeadReason(string reason)
165         {
166            EndOption.DeathReason = reason;
167             yield return null;
168         }
169
                1 reference
170         private static IEnumerator Chchapt(string Character)
171         {
172             TestDialogueFiles instance = GameObject.FindObjectOfType<TestDialogueFiles>
173
```

Figure 4.4 Implemented Command database

Source:<Own research> **date 20.01.2026**

Chapter 1 (Text Asset)

```
Chmus("Lullabye - Density & Time"
Chovervoice("V1"
Narrator "I have seen this world in gray colors for as long as I can remember."
Chovervoice("V2"
"It's funny, but I didn't know that I didn't see the world like other people. As a child, I thought I
just had bad eyesight. I understood the concept of colors, but I could never see them."

Chscene("EliasScene1_Monochrome"

"The problem arose when I became interested in graffiti art. Others didn't understand me, they
thought my strange combination of colors and drawings was ridiculous."

Chsou("TeensMockingNeil"

"I hate thoose days..."

Chchapt("1"
```

Figure 4.5 Game Chapter Source:<Own research>, date 20.01.2026

The JSON save file, generated automatically upon saving the game, captures the current session state to enable reliable loading and continuation of gameplay. The complete structure of this file is presented in **Appendix A. Save File JSON Configuration(p53)** This file records the following key elements:

- The names and visual properties (colour overlays, size deltas, positions) of the main and supporting characters in the scene.
- The current script file name and line number for narrative progression.
- A timestamp (date and time) to display in the save/load menu.
- The loaded status flag, which manages loading conflicts and ensures proper command execution.

The loaded flag operates as follows: it is set to true immediately after a successful load operation. As the game processes commands executed via coroutines (to allow simultaneous execution), the line counter decreases accordingly. Once the counter reaches a non-text line or the beginning of the script, the loaded status is reset to false. This mechanism prevents unintended command skips, overlapping executions, or crashes during state restoration, thereby ensuring stable and predictable behaviour upon loading. The implementation of the save/load system, including serialisation, parsing, and conflict handling logic, is detailed in the project source code.This design choice supports the educational objective of the prototype by maintaining consistent and error-free gameplay across sessions. Also to make safe file weight as little as possible, all details of level are loaded from information of chapter and Main character name, for example, Main character name Neil, who is colorblind, grayscale automatically applied to game

Levels Code descriptions:

- **Level 1 - Visual Impairment:** A grayscale shader and color inversion script were applied to all visual elements to simulate colorblindness. This required integrating custom materials and adjusting the UI for readability.

  After competing Level 1, player view is changed for next Pearson, which gives possibility for player to view game colors after starting level 2, which is applied both for the game itself and Game menu, below is comparing pictures of game view for level 2(Upper picture) and level 1 (Picture below). Look at **Figure 4.6 (p43)** and **Figure 4.8(p44)*** For this level, a minigame was developed where the player must click on the ‘spray’ and, according to the frequency of the sounds heard and the number of connections, connect it to the correct color. To do this, you need to click on the spray while holding down the mouse button and, without releasing it, draw a line to the color.

At the end of the minigame, a window appears showing the number of correct connections and offers to finish the minigame or start over. If the minigame is completed without connecting 6 out of 6 correctly, the player gets the first bad ending. If the game is completed with the correct options, the player can continue and open the second level of the next story. Look at **Figure 4.7** for correct mapping to beat minigame, look for code implementation at **Figure 4.10(p45)** and visual representation at **Figure 4.9(p44)**

Figure 4.6 (Project Game) Comparison of level 2 and level 1 views

Source:<Own research>, date 20.01.2026

```
33
34      // >>> ADDED - Correct mapping
35      private int[] correctMapping = new int[] { 5, 0, 3, 4, 2, 1 };
36
```

Figure 4.7 Correct minigame mapping, Source:<Own research>, date 20.01.2026

Figure 4.8 (Project Game) Comparison of level 2 and level 1 UI views

Source:<Own research>, date 20.01.2026

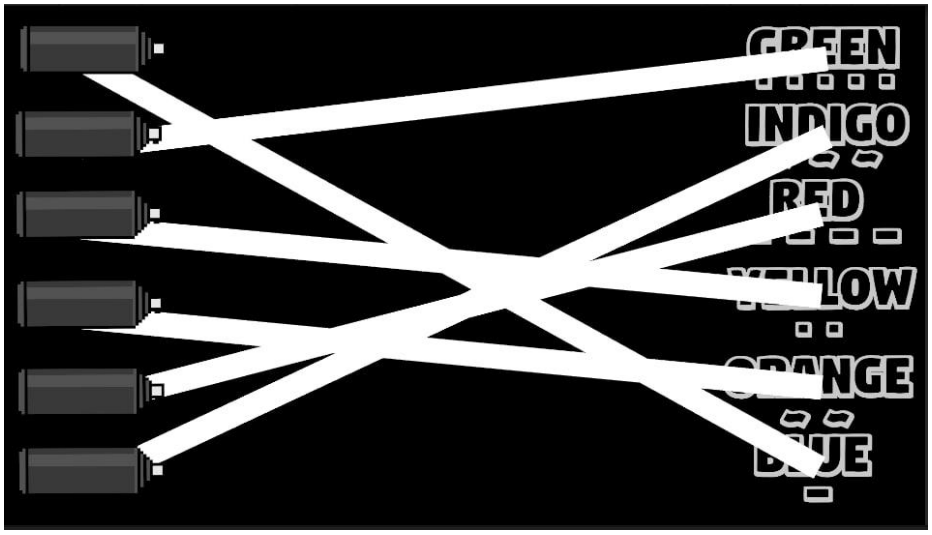

Figure 4.9 Chapter 1 Minigame, Source:<Own research>, date 20.01.2026

```
272     void EvaluateConnections()
273     {
274         // 1. Destroy all lines FIRST
275         for (int i = 0; i < spriteLines.Length; i++)
276         {
277             if (spriteLines[i] != null)
278             {
279                 Destroy(spriteLines[i].gameObject);
280                 spriteLines[i] = null;
281             }
282         }
283         Debug.Log("Lines destroyed. Checking connections:");
284         // 2. Now evaluate the connections
285         int correctCount = 0;
286
287         for (int left = 0; left < connectedRight.Length; left++)
288         {
289             int right = connectedRight[left];
290
291             string leftName = leftSprites[left].name;
292             string rightName = rightSprites[right].name;
293
294
295             if (right == correctMapping[left])
296                 correctCount++;
297         }
298         Debug.Log($"Correct: {correctCount}/{connectedRight.Length}");
299         if (correctCount == connectedRight.Length) Max = true;
300         ConfirmMenu.SetActive(true);
301         Gussed.text = correctCount.ToString();
302         OutOf.text = connectedRight.Length.ToString();
303
304         // 3. Clear connection data after checking
```

Figure 4.10 Source code Minigame, Source:<Own research>, date 20.01.2026

- **Level 2 - Auditory Impairment:** For this level, all colors were converted back from grayscale to RGB, but the music tracks and the song sung by the heroine were distorted to simulate deafness, allowing the player to hear approximately how deaf people hear.

Table of Sound effects

Effect Name	Recommended Setting / Value	Action in Audacity
High-Pass Filter	**Frequency:** 400 Hz **Rolloff:** 24 dB per octave	Removes the "muddiness" and bass to make it sound thin.
Low-Pass Filter	**Frequency:** 3500 Hz - 4000 Hz **Rolloff:** 12 dB per octave	Muffles the "sparkle" and high-end clarity.
Distortion	**Type:** Soft Clipping **Drive:** 15% - 25%	Adds that warm, slightly "broken" analog grit.
Vibrato	**Frequency:** 0.5 Hz - 1.5 Hz **Depth:** 5% - 10%	Creates the "pitch wobble" of a damaged cassette tape.

Effect Name	Recommended Setting / Value	Action in Audacity
Amplify	**New Peak Amplitude:** -3.0 dB	Use this after the filters to prevent the audio from clipping.
Compressor	**Threshold:** -20 dB **Ratio:** 3:1	"Squashes" the sound together to make it feel more compact.

Table 4.1 (Changes made to Audio Track in Audacity to simulate Deafness)

- **Levels 3 - 5** are currently under development; the code for these levels has not yet been implemented, and will not be implemented for the demo version for this thesis

Conclusion:

All scripts, shaders, and assets were integrated within Unity's scene hierarchy. Components such as player controller, environment objects, and interactive elements were linked to the scripts controlling accessibility features. Integration testing ensured that all features worked seamlessly: visual simulations did not interfere with controls, input modifications did not break navigation, and audio adjustments were correctly synchronized with gameplay events.

Version control through Git ensured that each change was documented, enabling systematic debugging and rollback when needed. This combination of Unity development and Git integration facilitated a stable, maintainable, and modular game architecture.

Summary

The design documentation and demo version of the project were presented to a selected group of gamers for evaluation. Overall feedback was positive. The majority of participants supported the concept of representing human disabilities as a meaningful and engaging gameplay challenge. In addition, the narrative approach of the project was well received, particularly its intention to raise awareness and encourage greater interest in understanding the experiences of people with disabilities.

However, a significant conceptual contradiction was identified during the evaluation process. While the game is designed to simulate various forms of disability and is intended to promote awareness and empathy, it currently lacks fundamental accessibility features that would allow people with disabilities to play it themselves. This creates a notable irony, as a project focused on representing disability does not yet implement inclusive or assistive design principles. For instance, the absence of features such as automatic narration in sound-oriented mini-games limits accessibility and highlights an important area for future improvement.

In conclusion, the project demonstrates strong potential both as an entertainment product and as an educational tool. Future development should focus on improving accessibility features in order to ensure that the game aligns more closely with its inclusive goals and can be experienced by a wider range of users.

Status of the Implemented Project

Current state: functional prototype.

Fully working Visual novel system game .txt file text converted and shown as text on the as well as character speaker name, and image. Commands database to change Images, characters, backgrounds, positions, levels in the .txt file.

Fully working Save/load system with date of saved game and screenshot.

Additional Menus such as: Achievements, Save/load menu, Level choose menu, Options.

Additional buttons: Skip.

Additional mechanic: Base for future language options, created language folders: EN, UA, all chapter files are saved in EN folder. Written separate script and check in important scripts like load for checking which language is used and accordingly to read chapters from folders, default is EN. Not working since UA folder is empty, left for be done after finishing whole game

Level 1 Chapter menu, Main minigame, Main achievements, and color inversion are implemented.
Lacks: Full written story, additional minigame, achievements
Level 2. Chapter menu, Color inversion, corrupted lead song of heroine simulating deafness are implemented
Lacks: Full written story, Main minigame, achievements
Levels 3 - 5. Design document with characters story, and disabilities mechanics that should be applied are created, no Code implementation
Additional buttons in progress: Auto, Config, shortcut buttons, Languages option.

Necessary Works to Complete the Project

Since the project was created to show that disability itself can be used as an interesting challenge for mechanics in game design, the game's plot text implementation is not important for this diploma. Ideally, it would be best to create a disability for each level, but since this affects every other level, and game as whole, it takes a lot of time, the presented working demo version and the description of the design document detailing which disability should be shown in the next levels are sufficient for this work.

Possible Improvements and Extensions

Writing full stories for all the characters, implementing levels 3-5, finishing shortcuts buttons, implementing Config and About menus, working auto button.

Difficulties Encountered

A significant amount of preliminary work was conducted to analyze existing tools aimed at improving accessibility and enhancing the gaming experience for people with disabilities. During this analysis, it was identified that three-dimensional input devices used for position tracking, effects, and similar interactions are often complex and difficult to operate for the target audience. Therefore, the decision was made to develop a two-dimensional project as a more accessible alternative.
One of the main challenges of the project was the development of the game design itself. This required an in-depth study of user feedback, including comments, reviews, and interviews with people with disabilities, in order to better understand their experiences and needs. Based on this research, the goal was to design engaging gameplay challenges that

both reflect real accessibility limitations and draw attention to the broader issues faced by this group.

Since the primary characters - players in the game are individuals with disabilities, the project introduces novel gameplay mechanics. As a result, there were no existing scripts, frameworks, or libraries that could be directly applied. This necessitated the development of custom solutions. Frequent modifications to the game logic and mechanics led to code conflicts and increased development time, particularly given that the project was implemented by a single developer.

Bibliography

1. AccessForge. “Unity and Unreal Engine Accessibility: Tools, Plugins, and Developer Guide.” Accessed 01 February 2026. https://accessforge.io/accessibility-game-engines.
2. Anguera, J. A., et al. “Video game training enhances cognitive control in older adults.” Nature 501 (2013): 97–101. https://www.nature.com/articles/nature12486, (accessed: 01 February 2026).
3. Atypical Advantage. “Simulation Games Workshop.” https://atypicaladvantage.in/atypical-workshops/simulation-games-workshop, (Accessed 01 February 2026).
4. Dalton, Tanya. “Episode 102: How Mandy Harvey Turns Tragedy into Triumph.” The Intentional Advantage (podcast). March 10, 2020. https://tanyadalton.com/podcasts/episode-102-how-mandy-harvey-turns-tragedy-into-triumph/, (accessed: 01 February 2026).
5. Family Gaming Database. “Gears 5 Accessibility Report.” https://www.familygamingdatabase.com/accessibility/Gears+5, (Accessed 31 January 2026).
6. Game Accessibility Guidelines. https://gameaccessibilityguidelines.com/, (accessed: 01 February 2026).
7. Gazzaley, Adam. “Adam’s Story.” https://gazzaley.com/adams-story, (accessed: 01 February 2026).
8. HTF Market Insights. “Inclusive Gaming Accessibility Tech Market Industry Overview.” https://www.htfmarketinsights.com/report/4399509-inclusive-gaming-accessibility-tech-market, (accessed: 31 January 2026).
9. IGDA Game Accessibility SIG. “Demystifying EAA & GPSR: New EU Laws for Game Accessibility and Safety.” June 28, 2025. https://igda-gasig.org/what-and-why/demystifying-eaa-gpsr, (accessed: 01 February 2026).
10. Kokkugames. “Making Games Accessible: Lessons in UI/UX.” https://kokkugames.com/making-games-accessible-lessons-in-ui-ux/, (accessed: 31 January 2026).
11. Level Access. “Rethinking Disability Simulation Activities: Moving from Empathy to Informed Action.” Blog post, May 1, 2023. https://www.levelaccess.com/blog/rethinking-disability-simulation-activities-moving-from-empathy-to-informed-action, (accessed: 01 February 2026).
12. Mauricio, Luis. “The Vale: Shadow of the Crown Review.” RPGamer, August 19, 2021. https://rpgamer.com/review/the-vale-shadow-of-the-crown-review, (accessed: 01 February 2026).

13. Microsoft. “Accessibility for Games.” https://learn.microsoft.com/en-us/windows/uwp/gaming/accessibility-for-games, (accessed: 31 January 2026).
14. Microsoft. “C# Documentation.” https://learn.microsoft.com/en-us/dotnet/csharp/, (accessed: 01 February 2026).
15. Shcherbak Anatolii, “DyplomWork.” GitHub repository - https://github.com/Anatolii-Shcherbak/DyplomWork_AnatoliiShcherbak_Id63570, (accessed: 01 February 2026).
16. Smith, Jessica. Interview by Libby-Jane Charleston. “Jessica Smith on life as a new mum with one hand.” Mamamia, October 15, 2019. https://www.mamamia.com.au/jessica-smith-interview, (accessed: 01 February 2026).
17. Swain, Frank. “The man with a camera inside his eye.” Interview with Rob Spence. BBC Future, July 21, 2017. https://www.bbc.com/future/article/20170721-the-man-with-a-camera-inside-his-eye, (accessed: 01 February 2026).
18. Unity Technologies. “Accessibility.” Unity Manual (2023.2) - https://docs.unity3d.com/2023.2/Documentation/Manual/com.unity.modules.accessibility.html, (accessed: 01 February 2026).
19. Unity Technologies. “Scripting.” Unity Manual. https://docs.unity3d.com/Manual/ScriptingSection.html, (accessed: 01 February 2026).
20. Unity Technologies. “UI Toolkit.” Unity Manual. https://docs.unity3d.com/Manual/UIElements.html, (accessed: 01 February 2026).
21. Wayline. “Making Games for Everyone: The Business Case for Accessibility.” March 13, 2025. https://www.wayline.io/blog/making-games-for-everyone-the-business-case-for-accessibility, (accessed: 31 January 2026)
22. World Health Organization. “Disability and health.” Fact sheet, December 7, 2023. https://www.who.int/news-room/fact-sheets/detail/disability-and-health, (accessed: 31 January 2026).
23. Youth Time Magazine. “The eyeless man taught himself to see.” Interview with Daniel Kish. December 12, 2018. https://youthtimemag.com/the-eyelessman-taught-himself-to-see/, (accessed: 01 February 2026).
24. Zork: Grand Inquisitor. Wikipedia. https://en.wikipedia.org/wiki/Zork:_Grand_Inquisitor, (accessed: 31 January 2026).

List of Figures and Tables

Appendix A. Save File JSON Configuration


```
{
    "mainCharacter": "Neil",
    "SupportCharacter": "Rey",
    "textureBase64": "XXX",
    "color": {
        "r": 1.0,
        "g": 1.0,
        "b": 1.0,
        "a": 1.0
    },
    "sizeDelta": {
        "x": 87.67900085449219,
        "y": 100.751953125
    },
    "sourceImageName": "Empty",
    "color1": {
        "r": 1.0,
        "g": 1.0,
        "b": 1.0,
        "a": 1.0
    },
    "size": {
        "x": 68.62975311279297,
        "y": 91.94135284423828
    },
    "position": {
        "x": -0.49142545461654665,
        "y": -0.3039150536060333,
        "z": 0.0
    },
    "sourceImageName2": "Empty",
    "color2": {
        "r": 1.0,
        "g": 1.0,
```

```
        "b": 1.0,
        "a": 0.501960813999176
    },
    "size2": {
        "x": 68.62975311279297,
        "y": 91.94135284423828
    },
    "position2": {
        "x": 0.24088595807552339,
        "y": -0.29001307487487795,
        "z": 0.0
    },
    "CurrLine1": 2,
    "CurentFile1": "Chapter1",
    "loadded": false,
    "date": "січ 20, 18:33:33"
}
```