Iurii Fedorenko

ANALYSIS OF THE TECHNOLOGIES AND TECHNIQUES IN ACCESSIBILITY FEATURES OF VIDEO GAMES

Thesis
CENTRIA UNIVERSITY OF APPLIED SCIENCES
Bachelor of Engineering, Information Technology
May 2024

ABSTRACT

Centria University of Applied Sciences	**Date** May 2024	**Author** Iurii Fedorenko

Degree programme
Bachelor of Engineering, Information Technology

Name of thesis
ANALYSIS OF THE TECHNOLOGIES AND TECHNIQUES IN ACCESSIBILITY FEATURES OF VIDEO GAMES

Centria supervisor Henry Paananen	**Pages** 52

Since the popularity of video games has been increasing, more players with disabilities have been joining the community, making the developers think about designing and developing necessary accessibility features for those players. However, two significant issues stand out: a lack of easily available information regarding what techniques and technologies are employed in various accessibility features and differences that must be considered when developing accessibility features across multiple gaming platforms.

This study aims to identify the techniques and technologies that are used for features that enable players at least to be able to interact with the game and the game interfaces, as well as how these features work across various gaming platforms. Additionally, the study aims to highlight the limitations and challenges impacting the implementation of the accessibility features. Thus, a qualitative approach to the research was used along with thematic analysis as the primary method for data analysis, which consists of available accessibility guidelines, literature, and publicly available videos and articles of developers about accessibility features in games.

Results from the thematic analysis reveal what techniques and technologies are used in accessibility features, for example, eye tracking, spatial audio, post-processing color filters, and letterboxing. The main nuances that developers should consider when developing accessibility features for various platforms are also revealed, such as differences in controls, screen resolution, input devices, and availability of technologies on the platforms. Moreover, results indicate that developers face common challenges, such as time and budget constraints, as well as inconsistencies in the availability of features across games that are released by the same company. Additionally, the considered laws that oblige developers to make accessibility features for the communication part of the game do not foster the implementation of more accessible games, especially for companies that have the budget for their implementation, because of the fragmented accessibility standards that are required for implementation since most of the games do not have any communication features in the game like voice chat.

This study contributes to the discourse and advancement of accessibility in video games by providing developers, especially indie developers, with a starting point for creating accessibility features in their games.

Key words
Accessibility features, accessibility guidelines, disabilities, gaming platforms, inclusion, techniques, technologies, video games

CONCEPT DEFINITIONS

AAA (triple–A) – “a major video game publisher or video game studio, in terms of budget (for both development and marketing” (IGN 2020)

Prefab – pre-made object or group of objects that can be shared between the scenes (Unity Documentation 2023)

Asset – “any item that can be used in the game, such as 3D models, sprites, etc.” (Unity Documentation 2023)

ABBREVIATIONS

API – Application Programming Interface

VR – Virtual Reality

AR – Augmented Reality

ABSTRACT
CONCEPT DEFINITIONS
CONTENTS

FIGURES

TABLES

1 INTRODUCTION

Video games have become a significant part of modern culture and entertainment on par with books and cinematography by providing players with a more immersive experience using their interactivity. However, these advantages over other forms of entertainment most lead to more problems with accessibility for disabled people because of video games' complexity over other pieces of entertainment, to enjoy video games as much as people with no disabilities. As a result, accessibility features have become increasingly important to ensure that all players can enjoy games without any barriers. To overcome these barriers, some game developers are implementing accessibility features, which rely on various technologies and techniques to cater to the needs of players with different kinds of disabilities and their extent.

Although accessibility features can be found more frequently in video games in the last several years, most of the AAA game studios and indie developers either implement most of the features poorly or do not see a reason why they should spend their time and money on such feature. The reason is that it could delay the release of their product and, therefore, disappoint their target customers and lead to lower revenue in the end. Despite that, research proves otherwise; according to Accenture's research (2018) on the effect of prioritizing digital inclusion demonstrates that companies that have a high score on the Disability Equality Index (DEI) are more likely to have higher revenue, net income, and better performance on economic profit margin than the companies that do not prioritize digital inclusion. In addition, people with disabilities, as revealed by a survey conducted by Scope UK, tend to purchase in-game add-ons or subscriptions 13% more regularly than non-disabled people (Scope UK, 2022). Therefore, people with disabilities could be seen as valuable customers for a company and could help it outperform the competitors in the market. A lack of awareness of the topic can explain why studios and indie developers have such stereotypes. This lack of awareness is prominent, especially when it comes to understanding how many kinds of disabilities there are and the various extents of their development, leading to poorly implemented accessibility features or a lack of such features at all.

This thesis aims to provide developers with a starting point on this complex topic by analysing techniques and technologies that are used in the best implementations of accessibility features, which are aligned with Cairns’s access level of vocabulary, with examples from the games. Only access features that belong to the Cairns’ vocabulary will be discussed due to the differences between the approaches to access and challenge levels in the vocabulary (Cairns, Power, Barlet & Haynes 2019b). Additionally, the thesis will not consider features that are used only for specific game, for example, predefined messages in the chat, which is a feature that only employed in multiplayer games. The focus will be on games that prioritize entertainment, excluding educational games due to their distinct purpose since they prioritize learning purposes over fun, which is the primary reason why most gamers play games. VR/AR games are also excluded due to their current limitations, particularly for prolonged game sessions by individuals with and without disabilities. Also, the popularity of such games is low among people with and without disabilities. (Thompson, Ehrhardt, Lippincott, Anderson, & Morris 2021, 164; AnalyticsIQ 2023.)

In accordance with the thesis' aims, the following research questions were developed:

1. What kind of techniques and technologies are used across different accessibility features in video games?
2. What developers should consider when implementing accessibility features in games for different platforms?

Since the thesis aims to contribute to the ongoing discourse on fostering inclusivity in the gaming industry, the according sub-question must be considered:

1. What challenges and limitations hinder a proper development of accessibility features for game developers?

This thesis is divided into seven chapters. The first part is the introduction to the topic. The second chapter provides a comprehensive overview of the diverse landscape of disabilities and their potential impact on the gaming experience of people with disabilities. Also, the appropriate vocabulary and framework will be introduced for categorizing accessibility features. Moreover, the game interaction model will explain the barriers that stand against "fun" in games for people with disabilities. In addition, the two different approaches to game accessibility – "all-inclusive" and "multitargeting"- will be discussed. The third chapter delves into the research methodology that is used for the research in the thesis. The fourth chapter includes specific considerations that game developers should consider when developing their games for different game platforms. In the fifth chapter, the discussion of best practices of accessibility features will be based on the barriers and facilitators' approach. Also, the techniques and technologies used for these practices will be considered that can be employed in Unity game engine, providing sources for further research on the specific accessibility feature for developers. The sixth part is centered around challenges and limitations that stand in the way of implementing more accessibility features in more games. The last chapter discusses the analysis results of the previous chapters.

2 THEORETICAL FRAMEWORK

The following section will be focused on a definition of game accessibility and highlight two approaches to designing accessible games: all-inclusive and multitargeting. Also, a game accessibility vocabulary, which is based on the APX framework, will be introduced to categorise accessibility features in a more understandable way for game designers and not by impairment to underscore an importance of all-inclusive approach. After that, a game interaction model will be discussed to identify barriers that people with disabilities may encounter while playing a video game. Additionally, five main impairment groups will be discussed so that these groups can be listed for an appropriate accessibility feature, and developers will know who to hire to test these features.

2.1 Game accessibility

According to the Game Accessibility Special Interest Group of the IGDA (2004, 5), game accessibility is “the ability to play a game even when functioning under limiting conditions. Limiting conditions can be functional limitations or disabilities — such as blindness, deafness, or mobility limitations.” Following this definition, Matamala and Orero (2016, 75-77) identify two main approaches to game accessibility. The first approach is to design games accessible to all people or, as referred to, the “all-inclusive” approach by authors. Westin, Ku, Dupire and Hamilton’s game accessibility definition follows this approach: game accessibility is “about adapting a game’s hardware and software (such as game controllers, difficulty level, or feed-back modality) to individual needs, regardless of having a disability or not” (Westin, Ku, Dupire & Hamilton 2018 [Hassan & Baltzar 2022]). As noted in Xbox Guidelines, the main goal is to broaden the audience since not only people with medical disabilities but also people with short-term or situational conditions benefit from accessibility in games (Xbox 2021, 5).

The alternative approach is named the multitargeting approach here, where targeting different player groups and their needs is the main goal since the broad spectrum of disabilities provides various kinds of barriers to players, requiring other accessibility solutions, or as they are called in XBOX Guidelines, facilitators. The previously mentioned definition by GASIG can be classified as this approach since the focus is on people with disabilities. (Matamala & Orero 2016, 75-77.) Nevertheless, since the thesis is focused on accessibility features, this thesis adopts a definition of game accessibility influenced by Westin et al. (2018) and Matamala & Orero (2016), where offering different accessibility features in the game is considered game accessibility and is not seen only through the lens of people with disabilities (Baltzar, Hassan & Turunen 2023).

2.2 The APX framework and accessibility vocabulary

The guidelines were created to increase awareness of game accessibility among game developers and bring game accessibility to the mainstream. Despite that, only three sets of guidelines can be considered comprehensive at this moment: Includification from the AbleGamers Charity, the Game Accessibility Guidelines (GaG), and the recently updated Xbox Accessibility Guidelines. The last two sets update on a regular basis, making them more valuable for developing new features. Crafted in collaboration with industry experts and organizations, these guidelines share commonalities. This result is not unpredictable, given the alignment of common features in the guidelines, which are recommended by these experts. (Cairns et al. 2019.)

As noted by Cairns, although these guidelines are well-structured and can be used as a tool for identifying the parts of the game that make it inaccessible by providing different levels of accessibility features' implementation and examples from various games, they do not help developers to generalize these features. Certain features must be implemented differently depending on the design choice for different in-game situations to make games accessible. At the same time, developers have to consider how they plan to deliver the game experience they intended. Therefore, it is understandable why some developers design their games relying more on the experience of individual developers and traditional methods rather than adopting a player-centric approach for accessible games. Cairns, while admitting the cons of such guidelines, tries to solve this issue with them by creating the generative design with a broad range of accessibility vocabulary, which builds upon the Cairn's APX triangle framework (FIGURE 1.), to reformulate the guidelines in a form that would be more suitable for game developers so that they would be able to create the accessibility features specific to their games. (Cairns et al. 2019.)

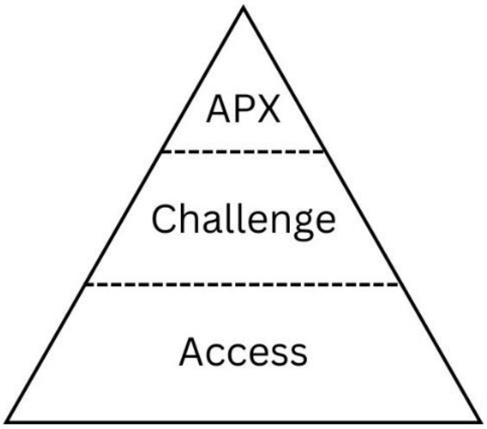


FIGURE 1. APX triangle (adapted from Cairns et al. 2019)

The APX triangle consists of three levels: APX, challenge and access. According to Cairns et al. (2019), the first two levels from the bottom represent options that work in tandem to deliver the accessible player experience (APX). Nevertheless, the main aim of the options, which are aligned with the access level, is to provide a player a way to engage with the in-game mechanics and interfaces. Options that are responsible for controls, input options, are about alternative controls (gamepads, eye trackers, accessibility controllers) to enable people with medical or temporary disabilities. Players can modify in-game controls by remapping controls, adding new ones, while also adjusting how the game reacts to the players' actions and adding macros or other interfaces to minimize number of buttons presses that are required to proceed in the game by using control options. These options and the input options have their differences with since control options handle controller customization and not alternation of controls. Presentation options allow to customize information that the user is able to see and hear to their own preferences, including giving a player control over how much information should be presented on a channel (i.e., visual or hearing), such as moving backgrounds or abrupt flashed, or variable audio channels. This may also include changing some aspects of the in-game UI. Possibility of switching between output devices, including their modifications are called output options. Although these options are different from presentation options (e.g. screen resolution that could make text too small), these options can still interact with them. In particular, these options are responsible for replacing or altering the device, for instance, by switching to a large screen from a mobile one, as a choice that Google made with the Google Stadia platform. Moreover, the options are about making more drastic adjustments to create components like those seen in headsets and displays related to virtual and augmented reality. These options relate to games that support streaming services, for example, Xbox Cloud Gaming, which are not the primary platforms of the analysis. Also, since the time Cairns et al.

(2019) released this vocabulary, Google Stadia was deprecated by Google in 2023, highlighting the niche of such streaming platforms.

On the other hand, challenge options are a next step to the APX and they are about an ability to tune the game to the player's needs, for example, an ability to make the game slower. Although these options will not be considered in this study, they will be listed to differentiate the difference between the access and challenge options.

Performance options are concentrated around adjusting the intensities or types of reactions that players would need to have in the game for better experience. Examples of modifying reactions include slowing down game events, allowing choices to pause or queue operations, and adjusting/removing timers from content. These options are essential for the majority of the games and accommodate all medical and temporary impairments. Training options are responsible for training a player different skill in games by giving player a possibility to play either a tutorial level or some sandbox levels as well as helping during game session in the form of visual cues or overlays. When a player finds an obstacle that they may find too challenging, progress options is a handy way to facilitate barriers for such players and to continue processing the game through various choices for the players. This includes an ability to return to previous points in the game, and bypassing areas where the players may have stuck for a variety of reasons. Therefore, any options that provide a possibility to skip a difficult section in the game, such as puzzle autopass detection, objective tracking, and puzzle hinting are all considered as progress options. To negate negative aspects that players can get from social interactions in the game to the minimum, customization of the way players interact, team up, and compete with others should be implemented, and they are considered social options. Players can manage various parts of the emotional presentation in a game by using moderation options, for example, turning off some explicit material. Additionally, this may include trigger warnings, possibility to turn off gore, sexual content, or some other settings that are needed for adjusting the disturbing material the player may encounter. (Cairns et al. 2019)

2.3 Game interaction model

Despite being similar in some aspects to traditional forms of media, video games are the only dynamic media that provides interactivity to the players, which is limited either due to hardware or choice limitations. This interactivity can manifest in reacting to the Player's actions in the game to provide a unique game experience for each Player, compared to other media, which can be defined as consumptive, where consumers, for instance, get the same ending in the story. (U.S. Department of State, American English n.d.) From a technical point of view, video games provide continuous output and feedback (i.e., audio and visual feedback), adapting to players' interactions through input devices, such as keyboards and controllers. Continuous output and feedback are built around the assumption that the player perceives them with their sight and hearing. (Rtc AI 2020.) Therefore, making the output and feedback the main barriers for people with medical or temporary disabilities affects what information they can perceive. Yuan, Folmer and Harris (2011) correspond to that problem with their generalized game interaction model, which helps identify the barriers a player with an impairment could face. This game interaction model comprises three steps (FIGURE 2.). The first step includes the player who receiving stimuli in the following forms: visual, auditory, and haptic, while a game provides stimuli. In the second step, based on the provided set of stimuli, players can have the ability to determine their response. In the third step, players provide the input through their input devices. These steps work in tandem, so if the Player cannot determine a response because they could not receive stimuli, they subsequently will not be able to provide an appropriate input.

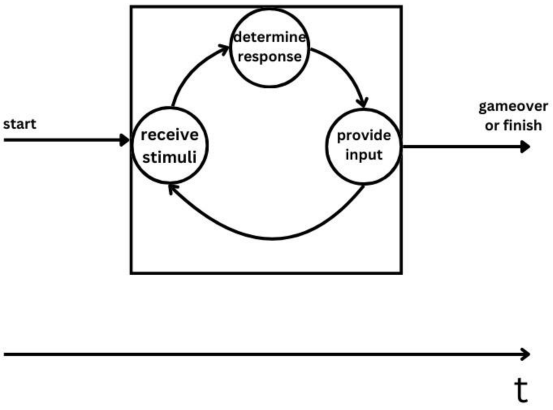


FIGURE 2. Interaction Finite State Machine (adapted from Yuan et al. 2011)

2.4 Motor disabilities

Motor disabilities relate to "a loss or limitation of function in muscle control or movement or a limitation in mobility" (Yuan et al. 2011, 3). Common illnesses that cause that include arthritis, paralysis, cerebral palsy, repetitive strain injury. Additionally, the natural aging can cause a loss of motor functions, such as reduction in muscle strength and controls. According to the interaction model, motor impaired players can have challenges with providing the input leading to receiving unintended stimuli or no stimuli at all. Therefore, people with motor impairments often rely on using adaptive input options, such as Xbox Accessibility Controller and eye trackers. (Yuan et al 2011; Cavender, Trewin & Hanson

2008.) Although these devices can heavily facilitate interactions with the game, these players can encounter other in-game barriers, such as complex control schemes, time-limited inputs (QTE segments) and mashing buttons (i.e. in action moments of the game) (Xbox 2023). Yuan et al. highlights the importance of providing the adoption of devices that support discrete inputs since they require less precise movement compared to devices with analog inputs, such as, a mouse or a controller analog sticks input (Yuan et al. 2011).

2.5 Vision disabilities

Visual disability “is the consequence of a certain degree of vision loss such as low vision/partially sightness, legal blindness, complete blindness” (Yuan et al. 2011, 3). Amblyopia (Lazy Eye), refractive errors, cataract, and color blindness are some of the diseases that cause this disability. Natural aging is also can lead to macular degradation and may result in blurred or no vision. In a game interaction model, people with visual impairments are not able to perceive visual stimuli and, therefore, are not able to determine their response to it and the physical input, despite the physical and cognitive possibility to provide that input. Therefore, it is important to provide these players with stimuli through other forms: auditory and/or haptic. This can include game narration, audio descriptions, haptic rumble patterns, and distinct audio cues. (Yuan et al. 2011; Cavender et al. 2008; Xbox 2023.)

2.6 Cognitive Disabilities

Cognitive disability is a mental and psychological disorder that manifests when a person has issues with remembering, memorizing, concentrating, and learning new things. Alzheimer, autism, dyslexia, down syndrome, and mental retardation developed during childhood are some of the most common cognitive impairments. Alzheimer and senility are cognitive disabilities that usually develop with the natural aging of the brain. The second step in the game interaction model is a barrier that people with cognitive disability encounter due to difficulty to cognitively determining an in/game response. This can be overcome by providing additional time for thinking (i.e. getting rid of time-limited QTEs), in-game tutorials, and presenting the information in a clear and concise manner. (Yuan et al. 2011; Cavender et al. 2008; Xbox 2023.)

2.7 Hearing disabilities

“Hearing disability is a complete or partial loss of the ability to hear from one or both ears” (Yuan et al. 2011, 3). This disability can have different levels of development from not severe to more serious, such as deafness. Players with this disability may not fully experience what the game has to offer due to the use of audio as a secondary stimulus in games. Also, if the game uses audio and visuals as primary stimuli, such as dance/rhythm games (i.e. Just Dance Ubisoft), hearing impaired players will not be able to perform steps 2 and 3 of the game interaction model. Depending on the level of hearing impairment development, the game can support spatial audio output, represent audio cues through visual and haptic channels, provide directional indication of where an audio cue or dialogue is coming from, etc. (Yuan et al. 2011; Cavender et al. 2008; Xbox 2023.)

2.8 Speech disabilities

Speech disability is an impairment that makes it hard for people to communicate properly. This impairment does not have a single cause, since it can be related to heredity, or some development is the brain. Additionally, the speech disability can indicate other conditions that are related to other disabilities, such as dyslexia, hearing loss, or Autism Spectrum Disorder (ASD). (Prelock, Hutchins & Glascoe 2008.) The barrier with this disability mostly appears in game that have online communication elements, such as voice-based communication channels. Also, games that completely rely on using player’s voice as a main feature of the game can have the same barriers for people with speech disabilities as online games. Therefore, the third step in the interaction model will not be performed due to inability to provide the input to the game. Thus, it is important to have an alternative input for such games, and text-based communication channels for online games. (Yuan et al. 2011; Xbox 2023.)

3 RESEARCH METHODOLOGY

This chapter explores the research methodology and data that will be used to answer the previously established research questions. Moreover, the thematic analysis as a method for the data analysis will be considered along with the thematic analysis of the six main phases used in this study's creation. Additionally, the limitations of this research will be discussed to establish possible gaps and outdated information that can be found in the study. Also, the table with the final classification of the accessibility features is presented, aligning with Cairns et al. (2019b) access level vocabulary and with the broader approach to features so that they would be helpful in most games and not particular genres of games.

3.1 Research method

Since this study is focused on the understanding techniques and technologies employed in accessibility features as well as differences in their implementations across various gaming platforms, a qualitative approach is adopted to provide a detailed view on the accessibility and in-depth analysis on the previously stated research questions using existing data on the topic as a basis for the study.

3.2 Data analysis

Thematic analysis method will be used as a tool for the data analysis. This method is useful for identifying common patterns in the data that aligns with the main objective of this study, which is to identify and analyse techniques and technologies that are used for best practices in accessibility features based on the existing guidelines. According to Clarke and Braun (2006, 86-91), the thematic analysis comprises six phases: familiarization, coding, generating themes, reviewing themes, defining and naming themes, and writing up. The first phases of the analysis will be centered around familiarizing with existing guidelines and resources related to accessibility features, including the Cairns accessibility vocabulary. After familiarizing yourself with the guidelines, the accessibility features will be categorized according to the vocabulary, providing a structured framework for analysis. After that, other sources focusing on the implementation of these practices will be examined, which will facilitate systematic coding of the data to identify corresponding techniques and technologies. These identified techniques and technologies will be reviewed to ensure accuracy and relevance to the guidelines’ best practices. In the last phases, the identified techniques and technologies will be named according to how they are called in most of the sources so that developers and other researchers would be able to identify each one of them easier for further research, and then these techniques and technologies will be analysed.

3.3 Research limitations

This study aims to provide a comprehensive analysis of techniques and technologies in the accessibility features of video games. Nevertheless, the use of qualitative analysis may lead to some limitations. The reason is that this thesis relies on existing literature, where some implementations of accessibility features can be outdated, including employed techniques and technologies, compared to the implementation of the same features in the recently released games. Additionally, proprietary techniques and technologies employed in accessibility features will not be considered because of the limited access to the proprietary information of game companies. However, this study aims to provide valuable information about techniques and technologies used in video game accessibility features for game developers, especially indie developers, despite these limitations.

3.4 Research data

Data for this research, which will be used for this study, will consist of Game Accessibility Guidelines (GaG) and the Xbox Accessibility Guidelines, which are more focused on the technical aspects of accessibility features implemented in video games compared to other guidelines. Moreover, other relevant resources, scholarly and company articles, and disability experts' presentations that discuss various techniques and technologies used in accessibility features across different games on various gaming platforms will be considered. Also, the combination of Xbox Accessibility Guidelines and GaG will serve as a primary source for classifying the features through the vocabulary developed by Cairns et al. (2019b). Nevertheless, some of the features, such as vibration customization, screen reader, and audio description support, will be discussed in the section related to the platform-specific considerations. Since these features have distinct platform considerations compared to other features, developers should consider that. The challenges and limitations of a broader implementation of accessibility features into modern games will be discussed using mostly publicly available articles and not scholarly ones since this theme is less technical than the techniques and technologies used in such accessibility features. The final classification of accessibility features is depicted in Table 1.

TABLE 1. Access level accessibility features

Access level accessibility features
Input options
Alternative input device
Control options
Controls customization
Presentation options
Screen reader
Audio description
Subtitles
Closed captions
Independent volume control
Customization of visuals in motion
Customization of UI elements
Colour customization
Spatial audio support
Haptic feedback

4 PLATFORM-SPECIFIC CONSIDERATIONS IN ACCESSIBILITY IMPLEMENTATION

When developing accessibility features for a game, developers should consider on which platforms they plan to release their game. For indie developers, the strategy may involve an initial release on PC or mobile platforms, followed by porting to other systems, such as Nintendo Switch, PlayStation 5, due to the lack of budget. On the other side, AA and AAA companies can afford to target a wide array of gaming platforms to maximize audience reach. Nevertheless, it is important to know what differences in implementation of the features there are on different platforms to successfully implement them.

4.1 Controls

Three of the most popular gaming platforms – PC, station/handheld consoles, and mobile – employ three different input methods. PC's highly precise controls can be achieved through the numerous keys on the keyboard and the mouse that is usually responsible for moving the in-game camera; consoles have less accurate controls due to a defined controller layout, or a touchscreen which provides more customizable controls, while heavily affecting the UI, and therefore, gaming experience. Although it is possible to connect any of these controls to other devices, most users tend to play the games with built-in controls into the hardware, except for some game genres where, for example, a game controller is a preferable choice in fighting or racing games. (Pingle 2021; Main Leaf 2023.) Designing controls for diverse input devices involves combining Fitt's Law, predicting human movement, and understanding hand limitations. The fundamental principle is placing frequently used actions in easily accessible locations aligned with the player's primary control group (FIGURE 3.). (Dotsenko 2017.)

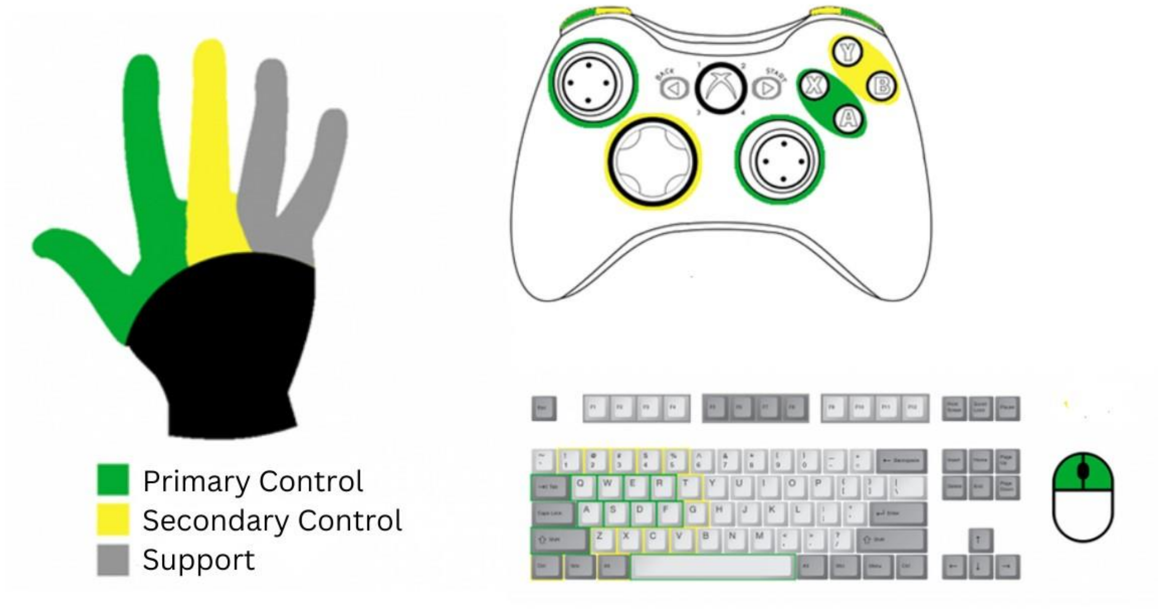


FIGURE 3. The controls layout of gamepads, keyboards, and mice, which is designed around three primary finger groups, following Fitts's law. (adapted from Dotsenko 2017).

For touchscreen controls, the primary control group of the player's hand is one or two thumbs, as adoption for a mobile device can be implemented in two ways, depending on the game's complexity and the genre: one-handed and two-handed input controls. The principle remains applicable but with a different tracking area for actions. On touchscreen devices, a gesture-based interaction (GBI) is employed and comprises two interactions: one involving a surface and the other including a motion gesture. According to Wobbrock, Morris & Wilson (2009), mobile users tend to prefer one-handed input controls over two-handed because of the effectiveness of the one-handed UI. Therefore, it is recommended to design or adapt, if the game is ported from the other platform, a one-handed input controls for games where it is applicable. Also, it is essential to provide tactile feedback for the user to minimize the user's mistakes and errors in touchscreen-based controls. Since they have such technology built-in, the vibration technology can provide ease of use, usability, and cognitive concentration of gameplay to game controllers and touchscreens, meaning reduced barriers for people with cognitive, motor disabilities and vision disabilities. Therefore, the process of porting a mobile game to handheld consoles, such as the Nintendo Switch, can be time-saving due to shared touchscreen functionality, similar hardware efficiency, and technologies like vibration. (Seo & Kang 2019.)

Therefore, optimizing the control scheme and making it intuitive for the players for the specific input devices is vital (Seo & Kang 2019). Furthermore, certain actions requiring multiple inputs need to be accounted for. It is easy to jump and press an action button on PC or consoles. But on mobile, multiple inputs are often not supported. Moreover, game controller support should be available for the game since people with motor impairment can utilize specific controllers, such as the XBOX accessibility controller. Developers should consider diverse control types even if they are not planning to port their game to other platforms. (Xbox 2021.)

4.2 Screen resolution and aspect ratio

Game UI design is a crucial aspect of any game since UI design is responsible for creating interfaces that facilitate the player's interactions with the game. By developing a solid UI design, the player will be able to navigate between various in-game systems and mechanics. Subsequently, this will increase players' engagement with the game. Usually, an appropriate UI design is created for each platform due to the variations in screen sizes and aspect ratios across different devices. Therefore, the number of various devices complicates porting the game from the initially intended playable device to completely different platforms, such as porting a PC game to mobile platforms. Hence, developers take extra considerations when developing UI designs for the same game but for other platforms. To address the issue with various devices, developers often employ the Content Scaling technique, designing their games initially for the lowest resolution they intend to support and subsequently scaling up for larger resolutions. (Felgo n.d; Gupta 2023.)

Nevertheless, it is important to consider the default aspect ratio, which can be described as a relationship between the width and the height of a display, as it defines the shape of the image on the screen and fosters the compatibility across devices. A standard ratio in the game industry is 16:9 (or 9:16 for the smartphones) since it is a common ratio across TVs, monitors, and phones. The resolution is interconnected with the aspect ratio since it defines the level of the image’s detail based on the number of pixels. Additionally, any changes in resolution can impact the overall visual appearance of the image and affect the easiness to read of menus, tutorials, by making the image blurrier, especially if the aspect ratio does not change. Therefore, considerations to mobile devices and monitors with various existing resolutions is essential to achieve better accessibility of the UI design. (Felgo N.D; Gupta 2023.)

Responsive UI and adaptive UI approaches can be utilised to address and solve the possible issues that could come up during the porting of the game to other platforms. Moreover, if the controls are designed as a virtual controller, they could take much space on the screen affecting the overall UI design, so it is necessary to optimize the game’s UI elements to fit these screen sizes properly. Also, if the game supports multiple input devices, this could impact on the tutorial's information, for instance, the tutorial may say "swipe up" or "tap the screen once" for touch screen controls, but for controllers the tips should be edited accordingly, so that they say, "press X button", when a user changes the input method to controller. (Pingle 2021; Main Leaf 2023; GaG 2024.)

4.3 Screen reader and audio description

A screen-reader is an external piece of software that enables blind or visually impaired users by rendering all text as speech output. There are different screen-readers available for people with visual disabilities, such as NVDA, Microsoft's Windows built-in Narrator on Windows. Nevertheless, screen-readers can output the UI text on the screen, like UI elements or the in-game main menu, to speech only if it has access to the UI tree information of in-game interface. This means that a screen reader needs to know, for example, where labels or buttons are positioned on the screen, their content (text), and associated possible interactions with these UI elements (i.e. buttons). Unfortunately, most popular game engines lack comprehensive screen reader support, as they do not render their UI, outputting a single block of pixels rather than the system UI elements, which screen readers can comprehend. (GaG 2024; Xbox 2023, 106; Unity 2024.) Nevertheless, in a recent engine release, Unity introduced beta support for mobile screen readers and plans to extend full support across all platforms (Perdigueiro & Oke 2024; Skerry 2023). Despite advancements, some game platforms, such as Nintendo Switch, lack support for screen readers, leaving developers unable to integrate such support. iOS and Android have built-in screen readers integrated into operating systems – Voiceover and TalkBack. Moreover, they work differently from other platforms due to different controls. Achieving screen reader support on mobile platforms is easier to achieve and applicable to a broader range of games, given the simpler interfaces and game mechanics prevalent in most mobile games compared to other platforms. On the other hand, screenreaders have a significant drawback since screenreaders are able to translate only UI text elements to speech and not visuals. One of the ways to fix that is by adding closed captions so that a screen reader can describe in-game actions using these captions. However, since screen readers have a "robotic" voice, this can affect the immersion of the player into the game world heavily, especially if the game has voiced dialogues, which would lead to situations where the dialogues are voices of the actual actors, while in-game actions with a robotic voice. Another approach is an audio description that requires hiring a professional who would voice all in-game actions, for example, this was done in Mortal Kombat 1 that also worked on Nintendo Switch. Although this approach offers a better cross-platform integration and places audio narration within the game rather than relying on a third-party program, it lacks certain player customization options, such as narration speed, voice, and language, since it would require hiring a lot of actors that are proficient in different languages. Moreover, this approach includes a more significant investment of time and resources than supporting third-party software. Hence, the choice between these approaches should consider platform-specific nuances, player preferences, and development resources. (GaG 2024; Xbox 2023, 106&111; Matamala & Orero 2016, 87-90.)

4.4 Haptic feedback

In video games, haptic feedback can provide a player with simulated information of in-game visuals or sounds through the relationship between touch and mechanical changes in the device that works based on the built-in technology in that device to initiate the changes. Devices that support haptic feedback include controllers, except for the Xbox Accessibility Controller and smartphones; the mouse and keyboard do not have built-in hardware that would support haptic feedback. Therefore, even though haptic feedback can play the role of another channel of information if a game supports devices such as a mouse, keyboard, and Xbox/Playstation accessibility controllers, it is necessary to concentrate more on other channels like visuals and sounds. (Yuan et al. 2011; Xbox 2023, 110; O'malley & Gupta 2008, 25-27.) At the same time, game platforms either have their own proprietary controllers that use their own technology to implement haptics, this also applies to different generations of consoles that can have new features in haptics feedback, or like in the case with smartphones, have built technology into smartphone that has some intricacies. For instance, comparing the PlayStation 4's DualShock controller to the PlayStation 5's DualSense controller reveals significant advancements in haptic feedback. The DualShock controller utilizes two Eccentric Rotating Mass (ERM) motors, commonly found in smartphones, to provide a basic rumble effect. In contrast, the DualSense controller uses Linear Resonant Actuators, which are more sophisticated as they utilize a voice coil instead of a traditional DC motor. The main advantage of linear resonant actuators is that they are programmed using soundwaves that allow vibration to start and stop immediately, therefore having a minimal output delay, which leads to greater precise feedback, compared to ERM motors that require time to spin up. (Dixon 2019; Somatic labs 2016; Gallant 2019.) The accessibility team of The Last of Us Part 1 was able to utilize new DualSense functionality and provide an option to play spoken dialogue through the haptics alone (Gallant 2019). Therefore, it is important to adapt the vibrations of controllers and smartphones with new technologies to provide a better experience for players with different disabilities.

In Unity, controlling the rumble of different devices is possible, for instance, through the new Input System, which offers classes such as Gamepad for controllers and Handheld for smartphones, each with their respective methods for managing rumble. However, Unity's implementation of rumble is basic to ensure compatibility across devices. Therefore, utilizing specific assets from the Unity store or developing a custom rumble system is recommended to implement the capabilities of devices like Apple's with their advanced haptic engines to their full potential. (Unity n.d.)

5 ANALYSIS OF THE TECHNOLOGIES AND TECHNIQUES IN ACCESSIBILITY FEATURES: BARRIERS AND FACILITATORS

To implement and design accessibility features into their games, developers often rely on established accessibility guidelines and the opinions of game testers with disabilities, if a company or developers can afford that. This chapter delves into the accessibility features that are recommended in these guidelines for implementation and examines the associated techniques or technologies that can be used for successful implementation by game developers. Additionally, the tools, techniques and technologies used for according features that are used in Unity engine will be discussed since it is the most popular game engine among indie developers, which is a target group of this study.

5.1 Input options

Cairns et al. (2019b) distinguish input and control options since they have different purposes. Despite that, only one of the guidelines belongs to these options. The gist of this guideline is a recommendation is to support more than one input devices in the game so that people with disabilities can use alternative controls.

5.1.1 Alternative input device

Across different game platforms there can be various supported alternative input devices, which depends on where the game is released. For instance, for a 3D shooter game, which is released on PC, it is possible to add not only mouse and keyboard support but also a variety of game controllers and eye-tracking technology. However, leading consoles on the market, such as Nintendo Switch, PlayStation 5, and Xbox utilize proprietary controllers, limiting support for alternative inputs. Notably, PlayStation and Microsoft have released their own accessibility controllers without requiring specific control implementations from developers (Sony Entertainment n.d.; Xbox n.d.). Therefore, these controllers are not considered in the chapter, considering their inherent compatibility with original controllers in terms of controls. Nevertheless, it is worth mentioning that Microsoft's accessibility controller is easily recognized on game platforms like Steam as an Xbox controller since Microsoft owns both Windows and Xbox, making it easier to add support on both game platforms. This highlights the importance for developers supporting controllers in their games in general, which would be beneficial for people with disabilities regardless.

The other alternative input device is an eye tracker, where Tobii solution stands out as the most popular one in gaming since it is the only one eye tracking solution that is available for PC gamers. Despite the variety of alternative input devices, game consoles and mobile devices lacking comparable alternative input solutions for game integration due to the lack of demand since most people play games on consoles far from the TV, therefore, will not use mouse or eye-tracker. Additionally, the adoption of alternative input devices can significantly enhance the gaming experience for individuals with motor impairments or fatigue-related conditions, allowing them to change between input devices as needed. Additionally, the support of technologies like eye tracking can improve overall immersion, offering new and innovative gaming experiences for all players, not just those with disabilities. (GAG 2024; Xbox 2023; Tobii 2024.)

5.1.2 Game controllers

In Unity, developers utilize one of the two techniques to handle player input: direct attachment of device controls to script functions and a modular input system. The first technique relies on directly linking input triggers to specific script functions. For example, a jump button can be linked to according jump functions that executes, when the button is pressed (FIGURE 4.). Although this technique can be useful for small games, where input can be represented with one button, in other cases this method can lead to accessibility issues. For instance, it can be difficult to add support for multiple input devices simultaneously so that players would be able to switch them on the fly or provide players with options to customize controls. (French 2023; Ramirez 2023.)

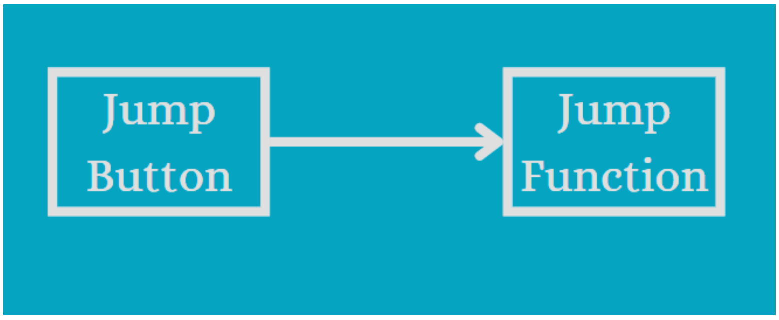


FIGURE 4. Device directly attached to the script functions they trigger (adapted from French 2023)

In comparison, the modular input system is a more flexible technique. This technique separates input triggers from the corresponding script functions by adding a special middlestep (actions), allowing the script to listen for actions triggered by various buttons or input devices without needing to know the specific source of the input (FIGURE 5.). This flexibility addresses the limitations of the direct attachment technique, which makes it easier to manage and adopt controls the player needs. For instance, by using the modular input system in the example with a jump, there is an intermediate step between the jump button and the jump function, called Jump Action, allowing us to add the remapping functionality of these buttons more easily. (French 2023.)

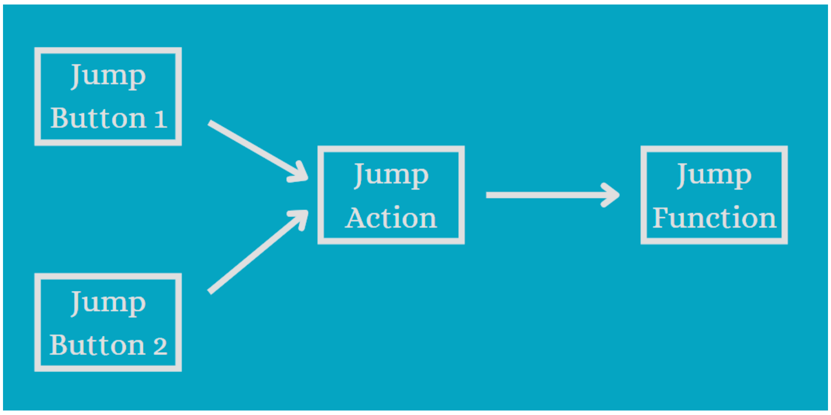


FIGURE 5. Middle step responsible for dividing the actual input device that causes an action and the function that will cause it to occur in-game (adapted from French 2023)

To implement the modular input system in Unity, developers can choose between two input systems: Input Manager and the newer system called Input System. While the Input Manager is a legacy system with limitations, such as limited cross-platform support and controls customization, it can still be used for prototyping purposes. However, the newer Input System is preferable for improved support for different input devices and better cross-platform compatibility, making it easier to address accessibility challenges across multiple platforms. (French 2023; Lee 2020; Ramirez 2023.) By using Figure 6 the basic operation of the Input System can be explained. First, the Unity Engine collects data from connected devices and notify the Input System about significant events, for example, button presses. After that, the Input System evaluates these events and assigns them to specific actions based on the information stored in the Input Action Asset. In the end, the Input System communicates these actions to the PlayerInput script, which causes the execution of corresponding functions (events) within the game code. (Lee 2020.)

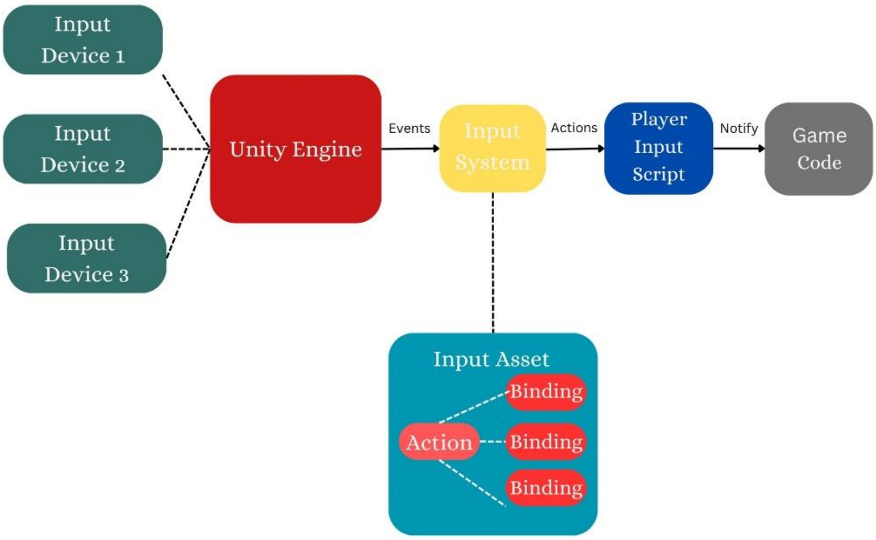


FIGURE 6. Visual perspective on the new input system in Unity (adapted from Lee 2020)

5.1.3 Eye tracker

The eye-tracking devices are an assistive technology developed for people with motor disabilities. The concept of eye-tracking can be defined as a process of measuring the movement of the eye relative to the head and the point where, in this case, the player is looking at - gaze point. (Cullipher, Hasen & VandenPlas 2018; Tobii 2024.) When it comes to eye-trackers, Tobii corp. is the leading company in the game industry since the developers can use their API – Tobii SDK – to integrate the tracker support to the game. Tobii Eye Tracker 5 is the latest model of eye-tracker devices, which consists of many illuminators and cameras. The reason is that it relies on pupil center corneal reflection (PCCR) technique to detect and track the eye movement relative to the head. Basically, it uses near-infrared illumination to create reflection patterns on the cornea and pupils on the eyes of the player. After that, the cameras in the eye-tracker take a high-resolution picture of the eye's reflection patterns so that image processing algorithms based on these patterns would be able to estimate the eyes' position and reflections patterns. Using the Tobii SDK, it is possible for developers to translate these calculations into digital input commands and adopt eye-trackers as another input device. (Cullipher et al. 2018; Tobii 2024.)

5.2 Control options

Cairns et al. (2019b) highlights that control options are firmly tightened to controls customization options, such as remapping of controls, adjusting the sensitivity of input devices (i.e. mouse, controller) and a possibility to disable or simplify repeated inputs (i.e. button-mashing/quick time events). According to Accenture report (2018), these options are one of the most used options not only among the people with motor disabilities but also people with other medical or temporary disabilities and no disabilities at all.

5.2.1 Controls customization

Controls remapping is a feature that allows a player to customize the input device's controls according to their preferences. There are several variations of this feature presented in modern games, such as remappable controls, available presets of different controls, and remappable controls with a possibility to change the input type. The choice behind the use of one or another technique can be explained by several factors: input device of the game, complexity of controls, and production difficulties. (Xbox 2023, 107; Accessible Video Game Design n.d.) For instance, when dealing with the extensive layout of a keyboard, developers have the flexibility to assign one action to each key. However, controllers like those of Xbox and Playstation have a more limited number of buttons than a keyboard does. Therefore, constraining the number of actions the gamepad can perform. To address this limitation, developers utilize multiple input types for the same button to represent different actions. For instance, in platformer games the character's jump height can be determined by the duration of the button/key press (Figure 7.) as opposed to the number of available buttons. (Xbox 2023, 107; Ash, Bungie & Brown 2020.)

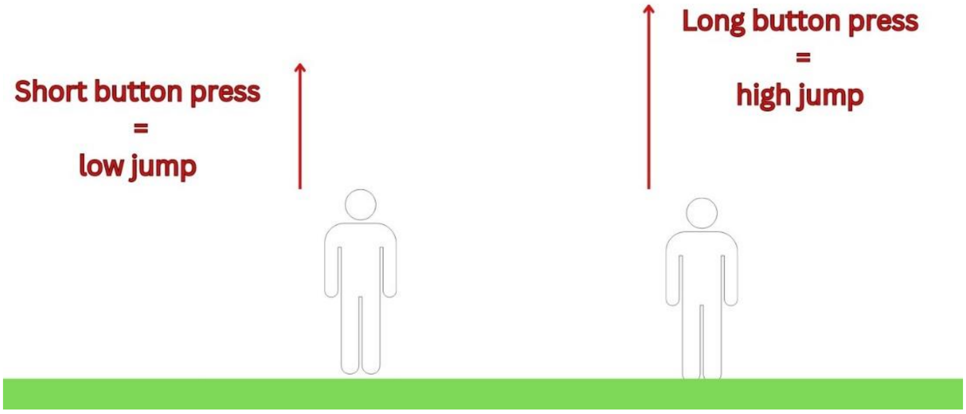


FIGURE 7. Representation of different actions through the same button (adapted from Pav Creations 2021)

Additionally, developers can add a double press of the same button to trigger a double jump action, allowing players to execute similar or different actions using a single button on the controller. This approach helps developers to create more mechanics and provide more intuitive controls for players. In some cases, developers implement simultaneous presses of two buttons (chords) for specific actions. For example, in Final Fantasy 16, executing a Limit Break action requires the player to press both the L3 and R3 buttons simultaneously (FIGURE 8.).

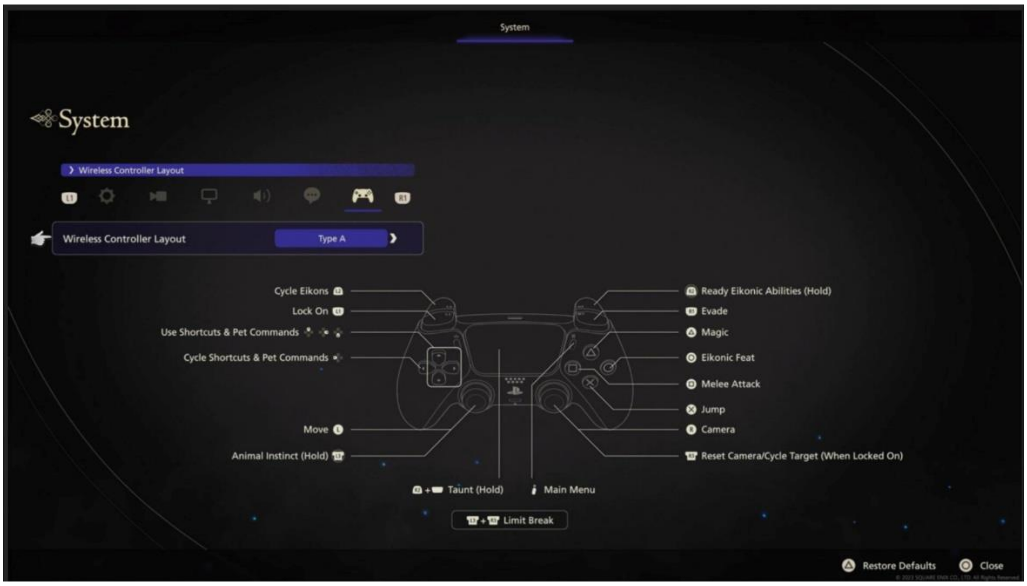
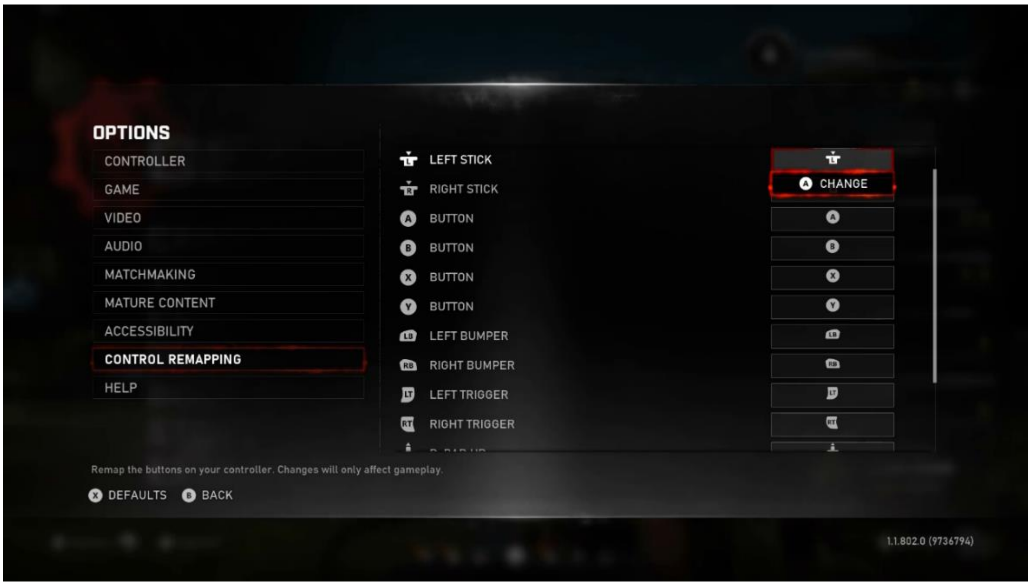

Wireless Controller Layout: Type A

FIGURE 8. Controller layouts presets in Final Fantasy 16 (Square Enix 2023)

While these techniques extend the possible range of actions with a limited controller layout, they also introduce difficulties, especially in action games like Final Fantasy 16, where the timing of button presses is crucial, which may hinder experience for players with motor and cognitive disabilities.

Nevertheless, Final Fantasy 16 has special accessibility rings, which, when equipped, can simplify, for example, the execution of combo attacks from several buttons to one, but this is more related to the design choice of developers (Purslow 2023).

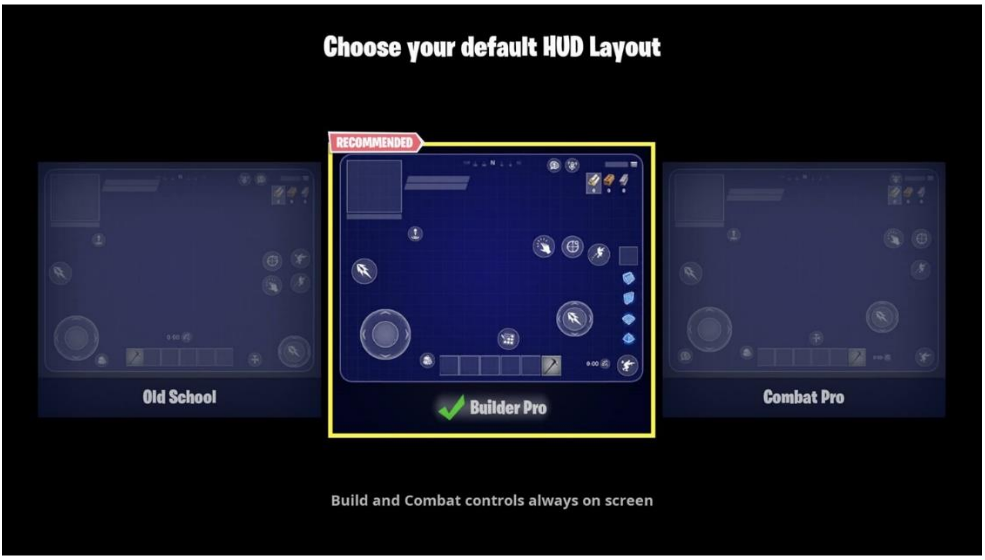
Moreover, immense usage of chords or multiple input types for most buttons can create other challenges for players, particularly those with cognitive disabilities, who may struggle to memorize these button combinations and variations during tutorial sections, leading to player frustration and possible disengagement from the game. Therefore, it is essential to provide players with the ability to remap buttons, including chords, and adjust input types if a game relies heavily on these techniques (FIGURE 9.). Despite the benefits of such customization, some games may not include full control remapping due to production difficulties. (Xbox 2023, 107; GaG 2024.) For instance, a possible late implementation of the feature during development, which can require significant resources to retrospectively integrate the features, as it was with Bungie's Tiger Engine in Destiny 2 (Ash 2020). Additionally, limitations inherent in the game engine's input system may also impact the implementation of control remapping features, especially if developers used the game engine for the first time for game development, as was the case with Square Enix's proprietary engine used for Final Fantasy 16 (Gamerant 2023).



Remap the buttons on your controller. Changes will only affect gameplay.

FIGURE 9. Changeable input types in Gears 5 (Microsoft 2019)

For mobile games, the multiple presets technique is the most usable one because of the several reasons. Firstly, mobile games are usually more simplistic in terms of game mechanics than games on other platforms and often rely on motion-gesture approaches like swipes or double taps for controls. As a result, using presets is a logical choice since the game may only require a few gestures to play. However, if the mobile game is more complex than a standard 3-row game, developers tend to implement a heads-up display (HUD) layout for visual control, which can and should be more customizable in a game. This visual control is especially useful for games that are ported from PC or consoles (FIGURE 10.).

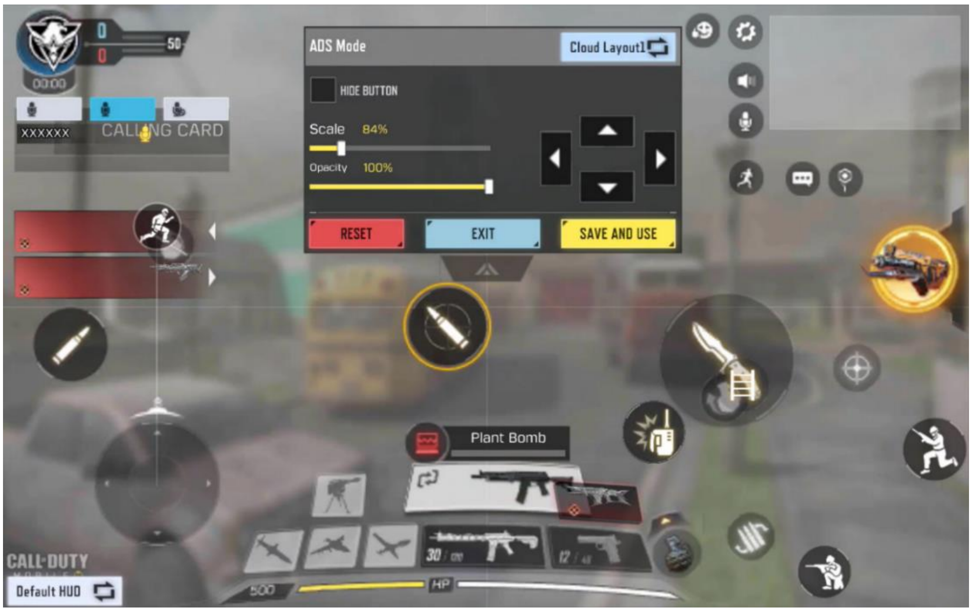


Build and Combat controls always on screen

.

FIGURE 10. Presets of HUD layouts in Fortnite Mobile (Epic Games 2017)

Although some games like Call of Duty Mobile allow players to customize touch controls in terms of size, spacing (FIGURE 11.), these customization options come with a high time cost for developers. Additionally, mobile games that support other input devices like controllers provide either presets for the controller or one predefined preset. This leads developers to choose presets as a default for remapping controls feature that is present to the player through UI, even in complex mobile games, considering the diverse range of devices and input methods available to players.



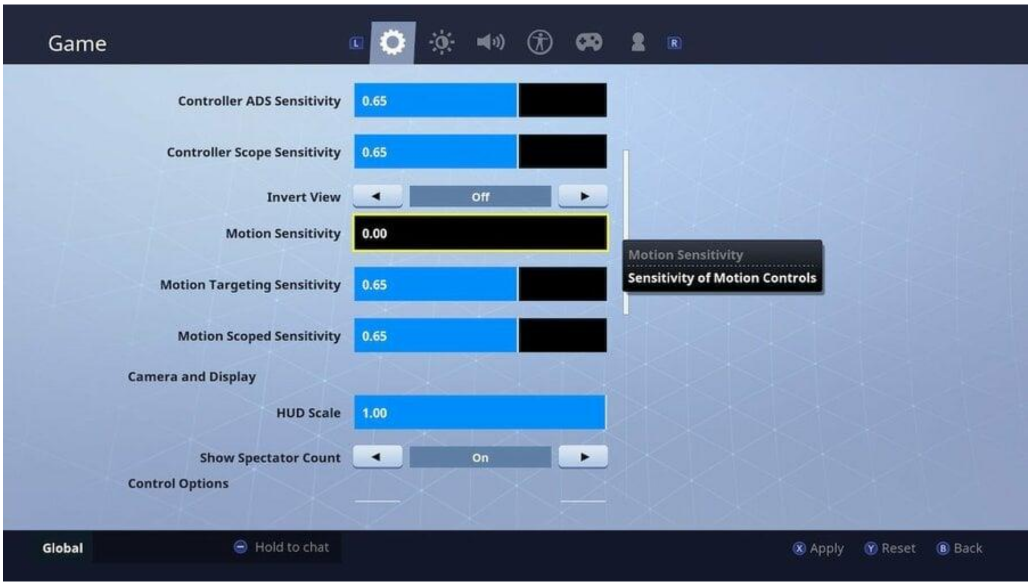
Scale 84%

Opacity 100%

FIGURE 11. Customizable Touch Controls in Call of Duty: Mobile (Microsoft 2019)

Also, controls customization should include the adjustment of analog sensitivity. Analog sensitivity determines how in-game actions respond to analog inputs, such as mouse or analog stick movements, from the player. This sensitivity is particularly important for ensuring smooth control of in-game elements like the camera, minimizing the impact of motion sickness. (Xbox 2023, 107; GaG 2024.) For gamepads featuring dual analog sticks, analog sensitivity plays a key role since each of them is mostly responsible for two different aspects of the game – the camera and the character's movement respectively. For instance, in Mario 64, a single analog stick could be used for the character's movement. By moving the analog stick along the X or Y axis, players could control Mario's speed, allowing him to either walk or run within the game. Although Nintendo 64 controller analog stick worked quite differently compared to modern controllers, this concept still applies to them. However, in some cases, players may prefer to use the controller's directional arrows for character movement instead of the analog stick due to their better precision, which should be considered by developers. (Special Effect DevKit 2023.)

Depending on the disability and the variation, players with disabilities may require different levels of sensitivity, from very low to very high. Additionally, if the game supports multiple input devices or released on a platform that have two built-in controls and use them in gameplay, for example, gyroscope/motion and analog sticks in Nintendo Switch (FIGURE 12.), it is important to ensure that there are sliders to adjust the sensitivity of all supported controls. (GaG 2024.)



Game

Controller ADS Sensitivity 0.65

Controller Scope Sensitivity 0.65

Invert View Off

Motion Sensitivity 0.00

Motion Sensitivity
Sensitivity of Motion Controls

Motion Targeting Sensitivity 0.65

Motion Scoped Sensitivity 0.65

Camera and Display

HUD Scale 1.00

Show Spectator Count On

Control Options

FIGURE 12. Adjustable gyro and controller sensitivity in Fortnite on Nintendo Switch (Epic Games 2018)

In order to implement these features, developers tend to use an input system technology. Depending on the game engine, this system can help handle input from various input devices. For in-stance, Unity has two input systems: Input Manager and Input System. Input Manager is a legacy in-put system that is built in into the engine’s editor. This system has limitations, such as limited cross-platform support or an absence of a debugger that would help to detect issues in the code more easily. To address these issues, Unity released newer Input System, which can be downloaded through Unity’s package manager. The newer system is more versatile and extensive than the legacy one. This system also has samples like Rebinding UI that provide an easier approach for developers with a set of scripts and prefabs to create a UI for players where they will be able to customize their controls. (Lee 2020; French 2023.) Some Unity developers do not use either a legacy input system or the newer one due to limitations of the previous one and complex documentation of the newer one or if the developers do not have enough knowledge for coding using the newer input system. The alternative comes from the asset store, where members of the Unity community can build and sell their tools to the developers. (French 2023; Sacco 2023.) One such tool, which is recommended by the Game Accessibility Guidelines regarding the remapping features, is an InControl API (GaG 2024). Unfortunately, this API does not have extensive documentation like a newer input system, which makes it more difficult for developers to use it properly. Moreover, it is not a free tool like Unity’s system. The best alternative to the newer Input System is a Rewired API, which has been supported for 10 years, has extensive documentation, is a debugger, and even supports more devices than Uni-ty’s new input system. However, since Rewired is built on top of a legacy input manager, the future deprecation of the input manager in new versions of

Unity may affect this alternative usefulness in the future unless it is updated to a newer version of the input system. (French 2023.)

5.3 Presentation options

Presentation options focus on allowing players to customize the amount of information presented through different output channels, leading to an experience that is more approachable and suitable for a specific player (Cairns et al. 2019b). Additionally, providing alternative channels for conveying information that may not be accessible to all players through the central channel for various reasons, such as medical or language barriers, facilitates the barriers these people may face during the game (Yuan et al. 2011).

5.3.1 Subtitles and closed captions

Among these options, subtitles may be considered an essential feature in games with any in-game speech or dialogue since they are considered a text equivalent of and used by most gamers with and without disabilities (Xbox 2023, 104). According to Ubisoft, 95 percent of Assassin's Creed: Odyssey players and 97 percent of Far Cry New Dawn players did not turn off the subtitles, which were on by default, highlighting the need for such feature in games among players (Sinclair 2019). One of the reasons why the subtitles feature is essential is a lack of games that have full audio support in other languages, which can be explained by the budget and time limitations that companies have, so they provide only subtitles that are translated into other languages and not multilingual audio dialogue support. Therefore, people that do not understand, for example, the English language, tend to toggle the subtitles on in-game to their language just to find out what the characters discuss. Also, even if the game supports only one language, it is still possible for people who are highly interested in the game to use some AI-translation software or online dictionaries that can be used to understand the gist of in-game dialogues. Additionally, even if people understand the language the characters in-game speak in, most of them still toggle the subtitles on due to the clarity and comprehension they provide to the players, especially in complex dialogues and fast-paced action scenes. (Xbox 2023, 104; Zieliński 2023.)

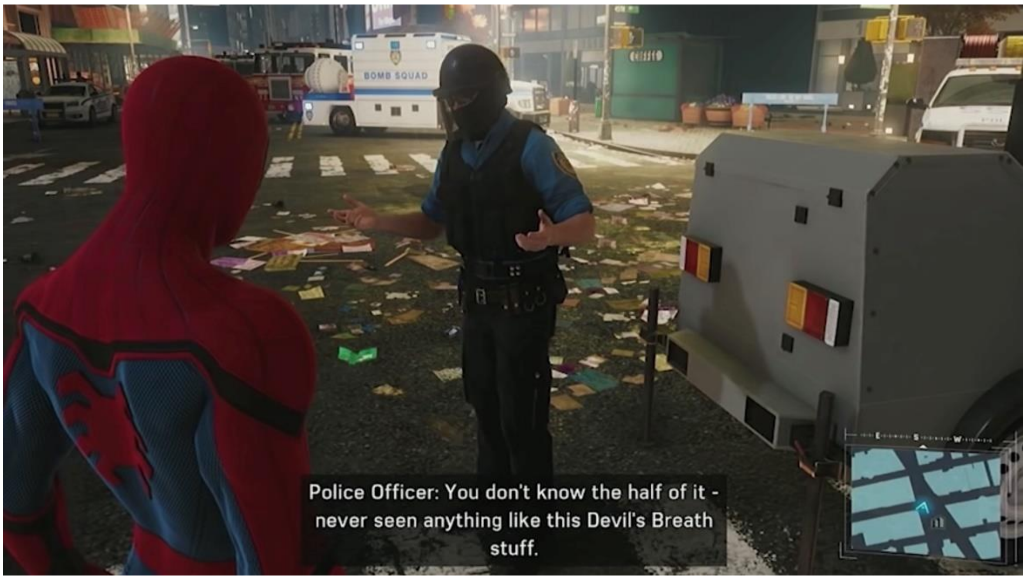
On the other hand, closed captions differ from the subtitles since they are text equivalents of all in-game audio and not only the speech as the subtitles. The in-game audio usually refers to full motion videos (FMVs), cutscenes, non-player chatter, music, and audio cues of actions that happen in the game (i.e. footsteps, wind). Unlike subtitles, closed captions cater specifically to players with hearing disabilities, ensuring they can fully comprehend the game's sensory information through visual text representations. (Xbox 2023, 104; GaG 2024.) Therefore, it is not surprising to see the subtitles more ubiquitous as a feature in games than the closed captions.

To effectively implement subtitles and closed captions in video games, developers should consider three crucial aspects: the user interface, event triggers, and timing. There are no standards on how the subtitles must look, compared to other media, but there exist recommendations from various authors or companies regarding how they should look, and most of them rely on the standards that are established by BBC and Netflix. (GaG 2024; Zucconi 2021.) One of the techniques from these recommendations, which is used for all UI elements and not only subtitles and captions, is the existence of a strong contrast ratio between the text/visual cues (i.e. buttons, icons, etc.) and background behind this text that enables people with low vision or cognitive disabilities to comprehend the text easier (Hamilton 2015). Contrast ratio in UI design is the measurement of the ratio of the luminance between two colours brightest and darkest shades. The contrast ratio represents how much brighter the text/visual elements (foreground) are compared to the background, and typically, the higher this contrast ratio, the better the visibility of the UI element. (Xbox 2023, 102; WCAG Techniques 2023.) Nevertheless, according to Microsoft Accessibility Guidelines (2023, 102), the minimum requirements for UI elements contrast ratio differ since the elements in the game can vary in terms of sizes (TABLE 2.).

TABLE 2. Recommended minimal contrast ratio for specific text sizes (adapted from Xbox 2023, 102)

Text size	Contrast ratio
Standard-size text or visual elements	4:5:1
Large-scale text and visual elements	3:1
Inactive-element text	3:1
High contrast mode elements	7:1
Placeholder or input field text	4:5:1 (standard size) 3:1 (large scale)

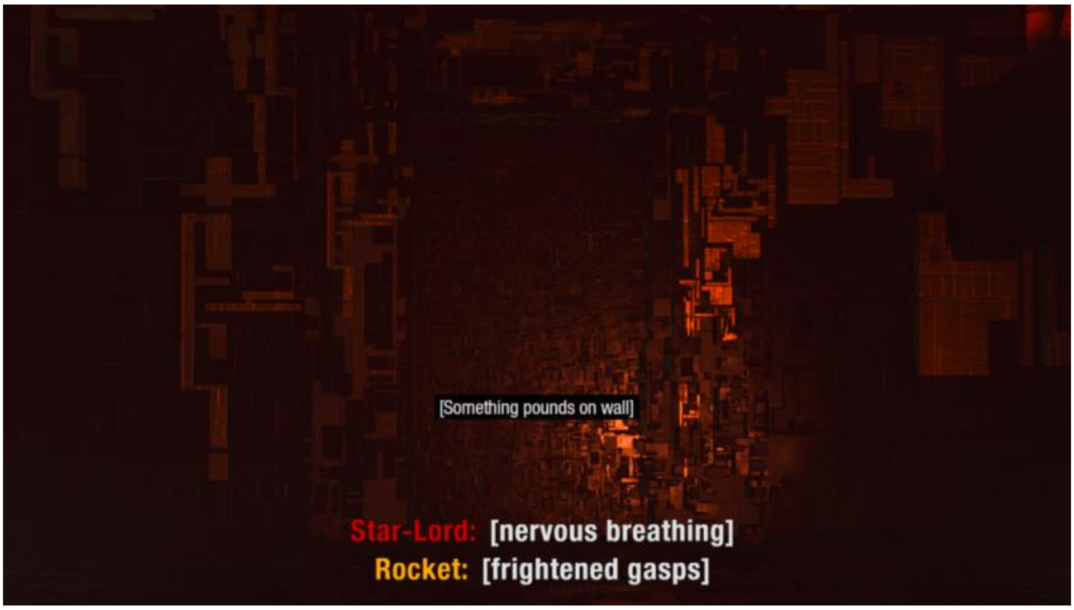
In order to calculate the contrast ratio between the UI elements, developers tend to use specific tools or their APIs such as, Color Contrast Analyzer, Accessibility Insights for Windows, etc. The advantage of these tools is that they show developers if the UI elements passed the requirements of minimum contrast ratio according to WCAG. (Xbox 2023, 102.) Another technique, which is used for subtitles and closed captions to increase the contrast, is called letterboxing. Usually, this technique associates with story driven games that uses it, so that the game would look more cinematic. When it comes to contrast, the gist of the letterboxing is to put the text against a solid or semi-opaque background to increase the contrast between the text and background (FIGURE 13.) (Hamilton 2015).



Police Officer: You don't know the half of it - never seen anything like this Devil's Breath stuff.

FIGURE 13. Use of letterboxing for subtitles in Marvel's Spider-Man (Sony Entertainment 2019)

Furthermore, by implementing other recommendations, such as large text sizes, ensuring adequate spacing, avoiding excessively long text lines, positioning captions towards the bottom of the screen, and distinguishing between speakers by displaying their names and also presenting them in a specific colour, it is easier to follow the in-game dialogues or monologues. Additionally, it is crucial to differentiate the appearance of subtitles and closed captions to avoid confusion for players with hearing disabilities, for instance, by using brackets for sounds or playing music (FIGURE 14.). (Xbox 2023, 104; Hamilton 2015.)



[Something pounds on wall]
Star-Lord: [nervous breathing]
Rocket: [frightened gasps]

FIGURE 14. Differently coloured names of the characters in subtitles and differentiation between subtitles and captions in Marvel's Guardians of the Galaxy (Square Enix 2021)

Notably, the font of the text should be a clean sand serif, and it is the best font for all players, even for people with dyslexia. Although the GAG and Xbox guidelines recommend giving the option of using OpenDyslexia-dyslexia font for people with dyslexia, recent studies that compared these two fonts, show that for people with dyslexia it is easier to comprehend the text with a standard font than the dyslexia friendly fonts. Therefore, it is instead recommended not to include these kinds of fonts in the game and to stick to the standard ones. (Kuster, van Weerdenburg, Gompel & Bosman 2018; Wery & Diliberto 2017.)

The other aspect of subtitles and closed captions is how long and when they should be displayed, referring to event triggers and timecodes. Event triggers are a way of handling gameplay subtitles and captions, which are triggered by in-game interactions, whereas cinematic scenes, which are similar to movie scenes, are fully scripted and, depending on how the audio file of this cinematic scene is handled, it is possible to use timecode metadata to play corresponding voice line, instead of event triggers. Unity has the UnityEvents system that can be utilized to trigger gameplay subtitles and captions. In this system, Events are components with which it is possible to subscribe the specific event to different functions (events) and call them subsequently when this event is triggered. For example, when a player dies in a game, it can be represented as a death event and triggered when the health bar reaches zero so that other functions to which this event has been subscribed, which can be responsible for updating UI, game states, are being called (FIGURE 15.). (French 2022.)

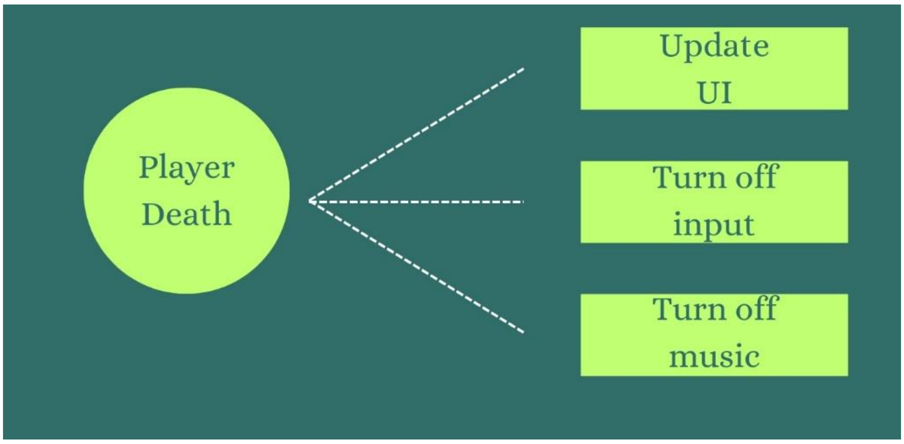
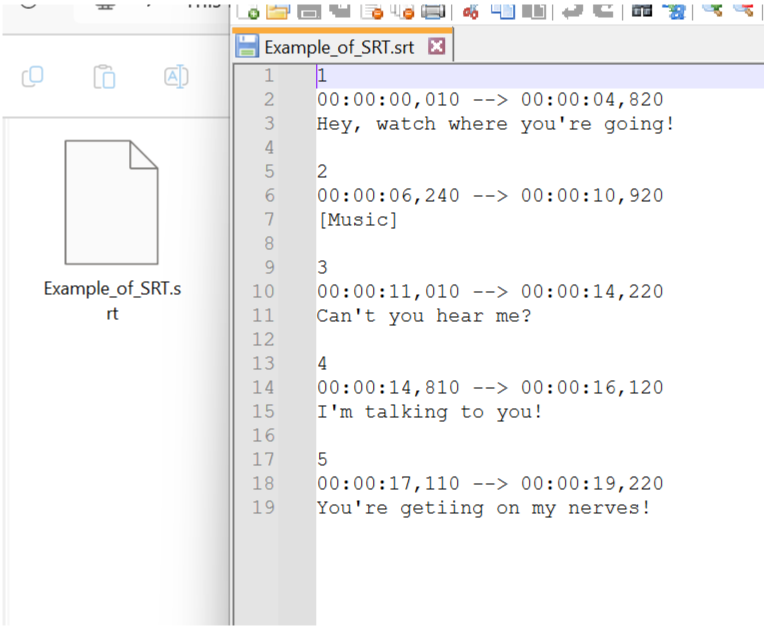


FIGURE 15. Example of in-game situation usage of Unity's Event System (adapted from French 2022)

Also, these events are helpful since they allow the check of whether the response to the action is needed only when the action is triggered, instead of continuous checking (French 2022). On the other hand, according to a game producer, Kari Hattner (n.d.), for FMVs or cinematic cutscenes, the situation may differ since the audio can be handled differently. From his experience, this cinematics can be represented as a one squeezed audio file with all the voice tracks mixed in. In these cases, developers tend to use timecode metadata to play each line at the correct time. One of the ways to create this metadata is by parsing a SubRip Subtitle file (SRT). SRT file is a plain-text file that includes the start and end timecodes of the subtitles/captions as well as a sequential number of subtitles (FIGURE 16.). (Forsey 2021.)



```
1
00:00:00,010 --> 00:00:04,820
Hey, watch where you're going!

2
00:00:06,240 --> 00:00:10,920
[Music]

3
00:00:11,010 --> 00:00:14,220
Can't you hear me?

4
00:00:14,810 --> 00:00:16,120
I'm talking to you!

5
00:00:17,110 --> 00:00:19,220
You're getiing on my nerves!
```

FIGURE 16. Example of a SRT file

This approach offers explicit advantages as it simplifies the process of providing localization for multiple languages through a file name prefix. By separating the code from the content, it is easier to localize the text for translators. Additionally, by using SRT files, it is possible to “provide localization for any number of languages or dialects based on e.g. a file name prefix”. (Vive Developers n.d.)
Unreal Engine has the built-in dialogue functionality, but it is not as customizable and flexible (i.e. not possible to change the fonts size and position of subtitles) as a system that can be created specifically for the dialogue and captions, for example, by using data tables and canvas panel for customizing the text (Aspland 2024).

5.3.2 Independent volume control

A standard video game comprises different types of sound, such as music, sound effects (SFX), ambient sounds. By giving players the option to customize each of these sounds according to their preferences and needs, both people with and without disabilities can benefit from it. This customization reduces the need for constant adjustments through their device's volume controls. For instance, if a game features only one slider controlling all audio elements, players may encounter difficulties with the volume of a specific sound during gameplay or cutscenes. Nevertheless, it is still important to have a master volume slider that would control all sounds in case a player does not want to set each slider at a time or wants to turn off all sounds instantly. (Xbox 2023, 105; GaG 2024.)

To categorise the sounds and create a slider for each sound, developers tend to work with audio mixing tool provided by the engine. In Unity, the Audio Mixer asset (FIGURE 17.) is responsible for that since it allows mixing various audio sources and applying various effects to them. By creating new audio mixer groups and adding them as children to the master group, it is possible to create C# scripts for sliders and connect each audio group mixer channel to its own slider. (Unity Manual 2023.)

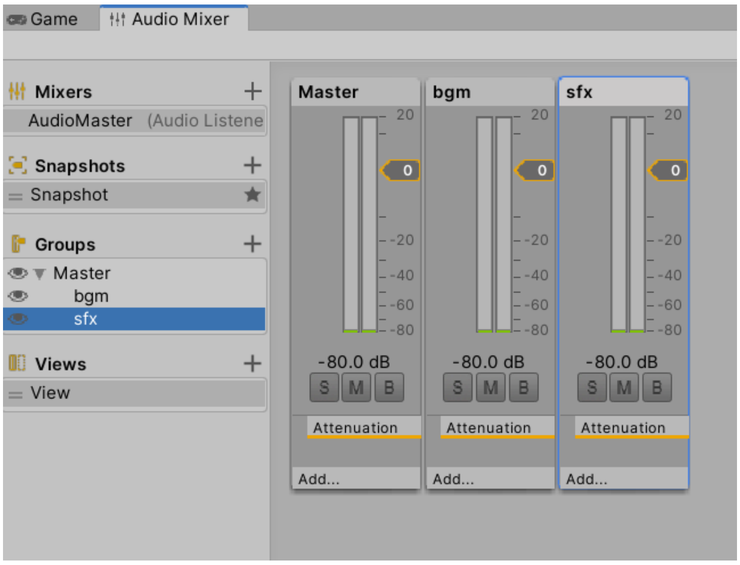


FIGURE 17. Audio Mixer in Unity (Unity)

Notably, a volume slider value is linear, where each increment is even, whereas the audio mixer attenuation, which is measured in decibels (db), is logarithmic, where each increment represents an exponential increase in value, which is better for humans since humans' ears perceive sounds logarithmically. Therefore, without converting the slider value to logarithmic, the slider would become excessively sensitive, making precise adjustments of the volume challenging. To ensure accurate volume adjustment in the option, it is necessary to convert the volume slider value to logarithmic by using a math formula for decibels:

$$dB = 20 * \log\left(\frac{Value\ 2}{Value\ 1}\right)$$

where dB stands for decibels, Value 2 and Value 1 stand for intensity ratio of two sounds. Additionally, it is important to set the minimum slider value to 0.001 to not break the slider at zero value. (French 2018.)

5.3.3 Spatial audio support

In most games, the objects in the game environment are implemented through the visuals, such as the next objective, characters, and warnings (i.e. reload, low health). According to Gröhn and Lokki (2006), audio cues are about twice as effective as visual ones when it comes to locating these objects. Although these audio cues are not so effective, people with no vision are not able to perceive these visuals and have to rely on these cues. Additionally, Gröhn et al. (2006) measurements highlight that players use audio cues (if they exist) first to approximately locate the in-game object, while visual cues (if they exist) are used for the final approach. Nevertheless, with two ears, humans are able to locate sounds in three dimensions: direction, elevation, and distance – this helps to find, for example, mobile phone more easily when someone calls. This process involves several auditory cues, including interaural level differences (ILD), interaural time differences (ITD), and spectral effects. Interaural time difference (ITD) is the time difference between the arrival of sound waves at humans' left and right ears, which is crucial for determining the horizontal location of low-frequency sounds. ITD varies based on a sound source's horizontal position relative to the listener, with greater distances resulting in larger time differences. On the other hand, interaural level differences (ILD) are utilized for higher-frequency sounds, as humans cannot rely on ITD for their detection. Furthermore, as sounds from different directions interact with the inside of humans' outer ears, they produce changes in frequency, which are called spectral effects. These frequency alterations aid in determining the vertical position of a sound source (FIGURE 18.). (Google Resonance Audio N.D; VRTonung n.d.)

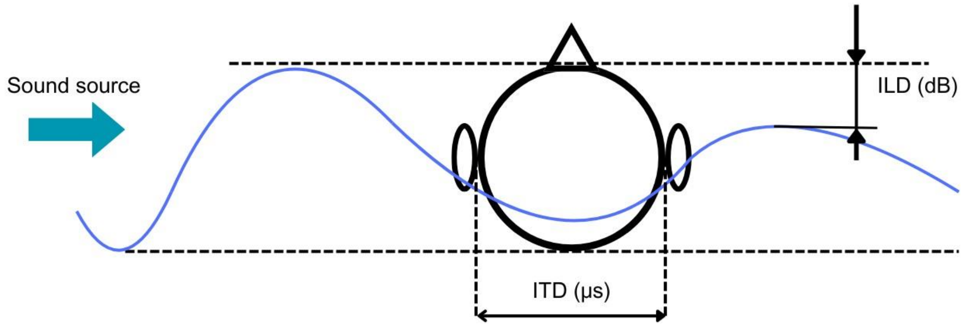


FIGURE 18. Illustration of ILD and ITD for a sound incident from the left side of the head (adapted from Francart 2008, 8)

This means that even if the game has audio cues, we still need to simulate 3D sound to locate where the sounds are coming from. Spatial audio technology, which heavily relies on the Head-Related Transfer Function (HRTF) technique that measures changes in the ITD and ILD, as well as other techniques for encoding, allows developers to simulate similar sound waves on headphones by creating an illusion that the in-game sounds are coming from specific locations in the game world. (Google Resonance Audio n.d.; Broderick, Duggan & Redfern 2018.) Mostly, developers tend to use special middleware solutions, which are used as plugins in the game engines, to simulate spatial sound because the creation of spatial sound system would take too much time due to the complex algorithms, physics, optimization, etc. There are two most popular plugins that are used by game developers and game sound engineers – Wwise and fmod. These tools provide various features, including spatial audio, to affect the audio and are supported by both Unreal Engine and Unity, including other game engines like Godot, which would take a bit more time to configure compared to UE and Unity. Additionally, Google released its own open-sourced SDK in 2018, which is focused on spatial audio alone and supported by both game engines and is called Resonance Audio. Google's solution is also free to use for everyone, whereas Wwise and fmod have their own licensing models, which makes Resonance Audio a viable choice for indie developers who have a tighter budget than game studios. Notably, Wwise offers support for the Project Acoustics plugin, created by Microsoft, which developers can use to simulate acoustics in the game. Project Acoustics allows developers to simulate more realistic movement of the sound between in-game spaces and the sound within these spaces (Xbox Project Acoustics 2023). Therefore, creating a better audio experience for players, particularly those with no vision, by providing a more immersive and spatially realistic auditory environment.

5.3.4 Customization of visuals in motion

There are various visuals in motion techniques that are used in games for different purposes. Camera shaking is highly used in cinematographic moments of the game or a motion blur, which is one of the post-processing techniques, and is added for different reasons, such as achieving a greater sense of realism or reducing artifacts in games that occur because of some GPUs miscalculations during post-processing (Wilde 2023). Nevertheless, these techniques can negatively affect people with heightened sensitivity to them, leading to dizziness or nausea. But since these techniques are added to the game, it is not a problem to turn them off by providing such an option for players. (Xbox 2023, 117.) Developers that make 3D games, especially first-person ones, also take into consideration Field of View (FoV). This parameter, simulated by a 3D engine like Unity, determines the viewing angle through the in-game camera. FoV, typically measured in degrees, mirrors the 360-degree scope of the game world. For example, a game with a default FOV of 90 degrees allows the camera lens to capture the in-game environment within this specter, which is then rendered and displayed on the screen.

While the human eye naturally has a field of view of approximately 60 degrees, providing a focused vision, binocular vision enhances this with a central focus point. Furthermore, humans possess a peripheral vision surrounding humans' focal area, which detects movement but lacks detailed focus. In contrast, the in-game camera offers flexibility, with its FOV ranging from 0 to 360 degrees. However, maintaining the same FOV across different platforms, such as console and PC versions of a game, can lead to motion sickness among some players. This issue appears due to two primary factors influencing how the screen occupies a player's visual field: screen resolution and viewing distance. For instance, console players do not sit too close to a TV and, therefore, a field of view of about 60 degrees is enough to prevent motion sickness, whereas PC players sit near the monitor, and if the game has the same FoV as the console version it leads to a mismatch between the incoming visual information and incoming vestibular information. Additionally, there are monitors that differ from each other in terms of screen resolution, highlighting the importance of FoV customization. Therefore, the in-game FoV customization option can decrease the number of players who would feel dizziness or nausea. (FZD School 2010, Xbox 2023.)

In Unity, there is an API and a script example of a FoV slider implementation using this API which is called Camera.fieldOfView that facilitates the implementation of this feature for developers (Unity Documentation 2023).

5.3.5 Customization of UI elements

The user interface customization accessibility feature caters to the needs of people with low vision or people with situational factors impacting their vision, such as playing on devices that make UI elements appear small. To implement not only this feature but a more compelling and flexible UI, developers rely on implementing adaptive and responsive UI designs, which are the advanced layout techniques in UI design. The main philosophy of the responsive UI is to create an interface that would adapt to different screen sizes and resolutions. (Gupta 2023.) In Unity, several layout techniques are used to create such a design: anchor points, content size fitters, and layout groups. Achor points allows to specify a point within the UI element (i.e. HUD elements like health bar) that determines its position-ing on the screen, making it easier to adjust UI elements size based on the device screen size. On the other hand, content size fitters allow the automatic control of the size of the UI element depending on the amount of text inside it. (Hardik 2023.) This tool is also preferrable to use when developing UI if developers plan to translate their game into other languages. Each of the languages has its own differences, and they have an effect on the content within UI elements. For instance, according to Mohammed Al-Batineh (2021, 60-62), the absence of acronyms in Arabic and the length of the equivalent Arabic terms differ from English. This variation can lead to situations where the size or positioning of UI elements needs to be adjusted to accommodate longer text strings. Otherwise, the translation team would need to decrease the font size of the content within the UI element, leading to worse readability. (Al-Batineh 2021, 60-62.) Therefore, setting anchor points appropriately becomes crucial to ensure that UI elements automatically adapt to text variations across different languages. As for the layout groups, they allow us to adjust the layout of several UI elements at once, for example, to align several UI elements for better adjustments of these elements for various screen sizes and resolutions (Hardik 2023).

Adaptive UI, which is sometimes called multi-platform design, is an adaptable design that caters to the needs of the end user (Hardik, 2023). This is where UI customization plays a major role in the implementation of such UI techniques. Unity developers tend to use Unity’s UI ToolKit to provide players with the ability to customize UI elements in terms of size, colour, etc.

5.3.6 Color customisation

Colour customization relates to a possibility of changing various in-game objects, including textures’ colour for 3D games, sprites’ colour for 2D games, UI objects (i.e. Health Points bar, destination marker). Mostly, this feature is beneficial for people with any of the visual blindness sub-types from deuteranopia (inability to see red/green colours) to achromatopsia (inability to perceive most or all colours) because this disability affects the perception of colours. The significance of implementing this customization feature can vary depending on the role of colours in specific in-game scenarios, such as solving puzzles, fast orientation using UI elements like map icons. (GaG 2024; Xbox 2023, 102.) For instance, in Paper Mario the Origami King, the final boss has multiple stages that a player must beat in order to defeat the boss, one of them includes a colour-matching puzzle with a short time limit. The added time limit affects the ability to solve this puzzle for people, for example, with deuteranopia time since it would take more attempts for them to solve it because of the low contrast between the colours of the tiles (FIGURE 19.). (Perry 2022.)

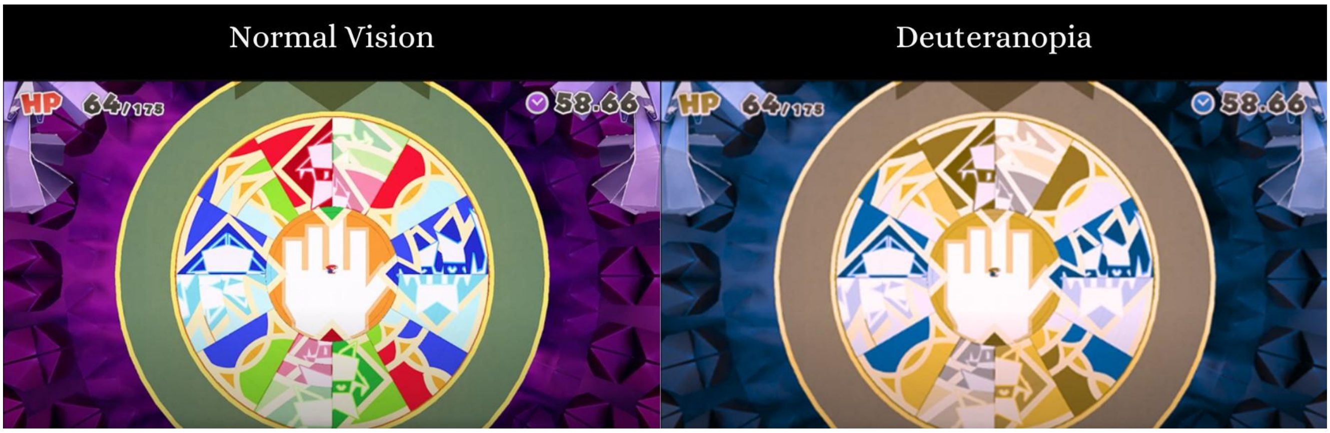
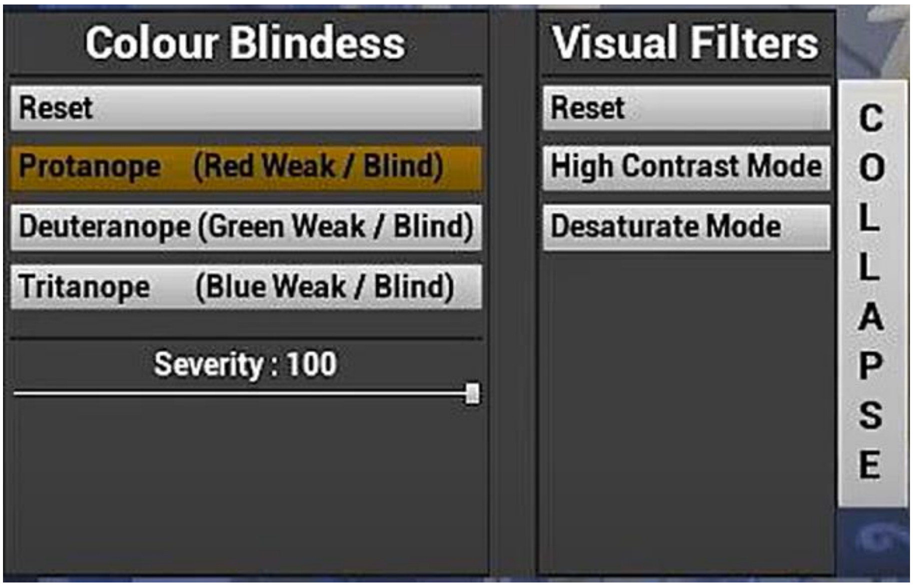


FIGURE 19. Colour puzzle in Paper Mario the Origami King from the perspective of the person with normal vision and deuteranopia (Nintendo 2020)

In order to avoid such situations, it is important to know if the game needs to have a colour customisation or a specific in-game situation like the puzzle mentioned before can be redesigned into one that does not rely on colours. Nevertheless, to determine if a specific in-game situation would be less approachable for people with colour-blindness, developers tend to create their own colour-blindness tool (FIGURE 20.) that would let them to simulate the experience of people with such disability. Usually, this tool comprises of different colour blindness filters that post-process the rendered final image by GPU. This also be referred to as one of the post-processing techniques. (Zucconi 2015.)


Colour Blindess
Reset
Protanope (Red Weak / Blind)
Deuteranope (Green Weak / Blind)
Tritanope (Blue Weak / Blind)
Visual Filters
Reset
High Contrast Mode
Desaturate Mode
COLLAPSE

Severity : 100

FIGURE 20. Example of colour blindness tool (Watterson 2022)

To create such filters, developers utilize image effects. These effects, implemented as scripts and attached to cameras, can manipulate the rendering output, or image, in various ways. In Unity, these effects are typically computed using shaders (FIGURE 21.), which are concise C# scripts responsible for rendering graphic data. On the other hand, rendering refers to the process of generating an image on the GPU. (Zucconi 2015; Kodeco 2020.)

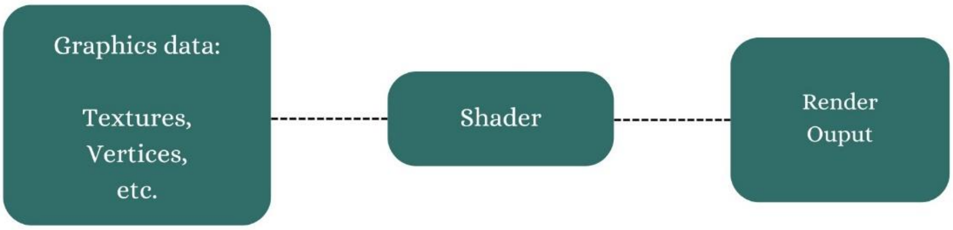


FIGURE 21. The process of rendering graphic data with shaders (adapted from Kodeco 2020)

It is important to note that while shaders play a crucial role in rendering, individual shaders are not responsible for the entire rendering process. Instead, different types of shaders collaborate to render the complete image. Among these shaders, the Fragment Shader stands out as it is specifically responsible for determining the color of each individual pixel in the rendered image, therefore, the one that can help to alter a texture (3D) or sprite’s (2D) colours. Whether the game operates in three-dimensional (XYZ - 3D) or two-dimensional (XY - 2D) coordinates, the Fragment Shader remains one of the best ways to manipulate the colors of visual elements to create specific filters. (Zucconi 2015; Kodeco 2020.)

6 CHALLENGES AND LIMITATIONS

The video game industry has become more aware of the importance of accessibility in video games. Microsoft, Sony, and Nintendo, as the main representatives of the console market, spend more and more resources on accessibility in their video games. For example, two games developed by the companies owned by Sony, God of War: Ragnarok and The Last of Us 2, won the award for "Innovation in Accessibility" at The Game Awards, which is an annual awards ceremony where video games are rewarded for their achievements. Additionally, new accessibility hardware has been developed by Sony and Microsoft. For instance, Sony has recently released its Access Controller in response to Microsoft's Adaptive Controller. Furthermore, Microsoft has been even more dedicated to accessibility by developing its guidelines for game developers in collaboration with industry experts and the disability community. Other companies are also putting effort into making their games more accessible by hiring accessibility experts to develop new accessibility features effectively. (Farokhmanesh 2021.) Moreover, Electronic Arts has launched its accessibility patent pledge so that other developers can use EA's "most innovative technologies designed to break down barriers for players with disabilities" for free (Electronic Arts 2021). Indie developers and smaller companies are contacting charities like the AbleGamers Foundation to ask for help with accessibility in their games (Farokhmanesh 2021). Although the video game industry has become aware of why accessibility is needed and significantly improved in terms of accessibility, the challenges are still present, especially for indie developers, which make it difficult to foster the implementation of features and changes in game design to more accessible so that games become accessible for everyone.

6.1 Lack of resources and inconsistency

Implementation of accessibility features comes at a cost either in terms of time, budget spent, or both. For instance, to implement features that are not depicted in the GAG basic section, it is recommended to have a disability test team to make corrections and improvements to the features, like audio cues or audio descriptions. (GaG 2024, Cairns et al. 2019b.) Nevertheless, indie developers cannot afford to hire so many people for testing purposes. Although it is possible to get feedback from people with disabilities who played the game, it is well-known that adding new features post-release costs more than designing them starting from the pre-production (Powers, Nguyen, & Frieden 2015). Additionally, features like audio description require developers to hire a professional and record their voice lines for all in-game actions, especially if the game is not a visual novel, where most actions are told through text and not visuals. Most indie developers and small studios cannot afford that. Therefore, it is necessary to have a smaller list of requirements for small game studios and indie developers when it comes to game accessibility. On the other hand, AAA studios have enough time and budget to hire professionals and design accessibility features during the whole game development phase. Therefore, it is not surprising that these studios are winning awards for innovation in accessibility and getting praise from the disabled community. However, there can be an inconsistency in the number of accessibility features in different games published by the same company despite the company owning the rights to these games. Comparing the remapping controls feature of the recently released The Legend of Zelda Tears of the Kingdom to Kirby and Forgotten Land, it is surprising that Zelda, which was released two years after Kirby, does not have either fully remappable controls or different presets of controls, whereas Kirby has provided players with various presets. (Bunting 2023; Family Gaming Database 2022.) Notably, Kirby was developed by HAL Laboratory, an affiliated studio of Nintendo, while Zelda was created by Nintendo EDP, the division responsible for developing games at Nintendo. This highlights the importance of AAA companies establishing in-house accessibility standards for their intellectual properties (IPs), even when games based on these IPs are developed by affiliated studios.

6.2 Lack of Accessibility Standards in the Video Game Industry

The problem with inconsistencies between studios could be solved with the accessibility standards for games, which other media have. And there are requirements, which were released in 2018, that are outlined in the 21st Century Communications and Video Accessibility Act (CVAA) legislation. Games that are released after December 31, 2018, in the USA, either through game stores like GameStop or digital storefronts like Steam, should comply with this legislation. The issue with this legislation and other legislations with equivalent provisions for local game markets, like the European Accessibility Law in the European Union, is that they are required to provide accessibility “as reasonably as possible to people with disabilities” in-game communication functionality, such as text/voice/video chat. This means that if a game lacks these communication features, which are common in many video games, developers are not obliged to make their games accessible. Moreover, even if these features are present, they only have to meet these accessibility standards, leaving other sides of the game that may impact the experience of people with disabilities. This approach can lead to inconsistencies within the same game, where communication functionalities may achieve a certain level of accessibility while other aspects remain inaccessible. Even though it is appreciated that the level of accessibility of communication features depends on the developers’ reasonable effort and budget, these fragmented standards fail to address the accessibility needs of all players, particularly those with disabilities. (Igda-gasig n.d., Johnson 2019.)

7 CONCLUSION

In summary, this thesis aimed to examine the technologies and techniques that are used for the implementation of the accessibility features that belong to the access level of the Cairns vocabulary so that developers would have a better understanding as to what features are needed for people with and without disabilities and where to start the implementation of such features. Additionally, the differences that various game platforms have, especially mobile and PC with consoles, in implementing and designing the same accessibility features that developers should consider delivering the same result for all platforms were discussed. By analysing accessibility guidelines and existing resources, including literature, articles, and presentations by experts in the field, the best practices in the implementation of accessibility features were identified, and techniques and technologies that are employed for these practices were discussed in addition to differences between game platforms when it comes to accessibility features implementation. Furthermore, the study aimed to understand the limitations and challenges impacting the implementation of game accessibility features. This included exploring resource constraints during game development and the absence of standardized accessibility guidelines in the gaming industry. In this thesis, a qualitative research approach was applied with a thematic analysis method for data analysis of the existing literature on the accessibility features in video games.

It is important to note that this research focused exclusively on accessibility features aligned with the access level of the Cairns et al. (2019) vocabulary, so other features that fall within the challenge level of the Cairns' vocabulary were neglected. Moreover, due to restricted access to certain proprietary literature and technologies employed by game studios, some aspects of accessibility features and their implementation were not fully explored in this thesis. Future research could address these issues, providing a more detailed view of other accessibility features.

REFERENCES

Accenture. 2018. *Getting to equal: The disability inclusion advantage.* Available at: https://www.accenture.com/content/dam/accenture/final/a-com-migration/pdf/pdf-89/accenture-disability-inclusion-research-report.pdf, 6-7. Accessed 10 January 2024.

Accessible Video Game Design. n.d. *Best practices for controls.* Available at: https://accessiblegamedesign.com/guidelines/controls.html. Accessed 5 March 2024.

Adam Telfer. N. D. *Touch Control Design: Ways of Playing on Mobile.* Available at: https://mobilefreetoplay.com/control-mechanics/. Accessed 25 March 2024.
Al-Batineh, M., 2021. *Issues in Arabic video game localization: A descriptive study. Translation & Interpreting*, 13(2), pp.45-64. Available at: https://www.trans-int.org/index.php/transint/article/view/1380/388. Accessed 9 March 2024.

AnalyticsIQ. 2023. *The Data-Driven Reality of Virtual Reality.* Available at: https://analytics-iq.com/wp-content/uploads/AnalyticsIQ-Research-Report-The-Data-Driven-Reality-Behind-Virtual-Reality.pdf. Accessed 27 March 2024.

Andrew Dotsenko. 2017. *Designing Game Controls.* Available at: https://www.gamedeveloper.com/design/designing-game-controls

Ash J., Bungie & Brown L. 2020. *Adding Controller Remapping to Destiny 2.* Video on YouTube. Available at: https://www.youtube.com/watch?v=u1PDqhD22BA&list=PLVEo4bPIUOski1CTT-TAbpPW9w1v10UuZe&index=6. Accessed 5 March 2024.

Aspland M. 2024. *How To Add Subtitles In Unreal Engine 5 The Right Way | Unreal Engine 5 Tutorial.* Video on YouTube. Available at: https://www.youtube.com/watch?v=0uYzq7JqPCI. Accessed 5 March 2024.

Baltzar, P., Hassan, L. and Turunen, M., 2023. *"It's Easier to Play Alone": A Survey Study of Gaming With Disabilities. Journal of Electronic Gaming and Esports*, 1(1). Available at: https://journals.humankinetics.com/view/journals/jege/1/1/article-jege.2022-0029.xml#r46. Accessed 10 February 2024.

Bleeding Edge Studio. 2023. *Efficient Game Development: Mastering Unity's Event System.* Available at: https://www.linkedin.com/pulse/mastering-unitys-event-system-more-efficient-game-development/. Accessed 5 March 2023.

Braun, V. & Clarke, V. 2006. *Using thematic analysis in psychology. Qualitative research in psychology*, 3(2), pp. 77-101. Available at: https://web-p-ebscohost-com.ezproxy.centria.fi/ehost/pdfviewer/pdfviewer?vid=0&sid=716ecda1-1cb6-4d6f-9c18-64be92d10100%40redis. Accessed 9 March 2024.

Broderick, J., Duggan, J. and Redfern, S., 2018, August. *The importance of spatial audio in modern games and virtual environments.* In *2018 IEEE Games, Entertainment, Media Conference (GEM)* (pp. 1-9). IEEE. Available at: https://ieeexplore.ieee.org/abstract/document/8516445. Accessed 5 March 2024.

Bunting G. 2023. *Tears of the Kingdom shows that without change, accessibility in Nintendo games will remain accidental.* Available at: https://www.eurogamer.net/tears-of-the-kingdom-shows-that-without-change-accessibility-in-nintendo-games-will-remain-accidental. Accessed 5 March 2024.

Cairns, P., Power, C., Barlet, M. & Haynes, G. 2019. *Future design of accessibility in games: A design vocabulary. International journal of human-computer studies,* 131, pp. 64-71. Available at: https://www-sciencedirect-com.ezproxy.centria.fi/science/article/pii/S1071581919300801?via%3Dihub#bib0029. Accessed 10 February 2024

Cavender, A., Trewin, S. and Hanson, V., 2008. *General writing guidelines for technology and people with disabilities. ACM SIGACCESS Accessibility and Computing*, (92), pp.17-22. Available at: https://www.sigaccess.org/wp-content/uploads/formidable/sep08_all.pdf

Dixon T. 2019. *Good Vibrations: How Apple Dominates the Touch Feedback Game*. Available at: https://www.ifixit.com/News/16768/apple-taptic-engine-haptic-feedback. Accessed 4 March 2024.

Electronic Arts. 2021. *Using our patents to drive positive play*. Available at: https://www.ea.com/able/news/using-our-patents-to-drive-positive-play. Accessed 5 March 2024.

Family Gaming Database. 2022. *Kirby and the Forgotten Land Accessibility Report*. Available at: https://www.familygamingdatabase.com/accessibility/Kirby+and+the+Forgotten+Land. Accessed 5 March 2024.

Farokhmanesh M. 2021. *More developers than ever are making accessible games*. Available at: https://www.axios.com/2021/05/27/video-game-accessibility-xbox-able-gamers. Accessed 5 March 2024.

Felgo. n.d. *How to create mobile games for different screen sizes and resolutions*. Available at: https://felgo.com/doc/felgo-different-screen-sizes/. Accessed 4 March 2024.

Forsey C. 2021. *What Is an SRT File, & Why Is It So Important for Video?* Available at: https://blog.hubspot.com/marketing/srt-file. Accessed 5 March 2024.

Francart, Tom. 2008. *Perception of Binaural Localization Cues with Combined Electric and Acoustic Hearing.* Available at: https://www.researchgate.net/publication/28360722_Perception_of_Binaural_Localization_Cues_with_Combined_Electric_and_Acoustic_Hearing. Accessed 26 March 2024.

French J. 2022. *Events & Delegates in Unity*. Available at: https://gamedevbeginner.com/events-and-delegates-in-unity/. Accessed 9 March 2024.

French J. L. 2018. *The right way to make a volume slider in Unity (using logarithmic conversion)*. Available at: https://johnleonardfrench.com/the-right-way-to-make-a-volume-slider-in-unity-using-logarithmic-conversion/. Accessed 5 March 2024.

FZD School. 2010. *Design Cinema – EP 29 - FOV in Games Part 01*. Video on YouTube. Available at: https://www.youtube.com/watch?v=blZUao2jTGA. Accessed 5 March 2024.

GAG. 2024. Game Accessibility Guidelines. Available at: https://gameaccessibilityguidelines.com/full-list/

Gallant M. 2022. *Expanding Accessibility in The Last of Us Part 1*. Video on YouTube. Available at: https://www.youtube.com/watch?v=__KsDXSeOQo&list=PLVEo4bPIUOski2wDMqh9L3YNTJAy-GHTqa&index=22. Accessed 4 March 2024.

Gamerant. 2023. *Final Fantasy 16 Made an Unexpected Engine Choice*. Available at: https://gamerant.com/final-fantasy-16-made-an-unexpected-engine-choice/. Accessed 5 March 2024.

Google Resonance Audio. n.d. *Resonance Audio – Fundamental Concepts*. Available at: https://resonance-audio.github.io/resonance-audio/discover/concepts.html. Accessed 5 March 2024.

Gröhn, M., Lokki, T. and Takala, T., 2005. *Comparison of auditory, visual, and audiovisual navigation in a 3D space. ACM Transactions on Applied Perception* (TAP), 2(4), pp.564-570. Available at: https://www.researchgate.net/publication/220245087_Comparison_of_Auditory_Visual_and_Audio-visual_Navigation_in_a_3D_Space. Accessed 5 March 2024.

Gupta, A. 2023. *Game UI Design: Improving Video Game Experience*. Available at: https://www.searchmyexpert.com/resources/game-development/game-ui-design. Accessed 4 March 2024.

Hamilton I. 2015. *How to do subtitles well – basics and good practices*. Available at: https://www.gamedeveloper.com/audio/how-to-do-subtitles-well-basics-and-good-practices. Accessed 5 March 2024.

Hardik P. 2023. *Responsive Design and Adaptive UI in Unity*. Available at: https://medium.com/@hardikparmarexpert/responsive-design-and-adaptive-ui-in-unity-bab765055c4d. Accessed 5 March 2024.

Hassan, L. and Baltzar, P., 2022. *Social aspects in game accessibility research: a literature review*. In *DIGRA*. Available at: http://www.digra.org/wp-content/uploads/digital-library/DiGRA_2022_paper_111.pdf. Accessed 10 February 2024.

Hattner K. n.d. *The Ins and Outs of Video Game Subtitling: An Interview*. Available at: https://www.md-subs.com/ins-and-outs-of-game-subtitling. Accessed 5 March 2024.

IGDA. 2004. *IGDA accessibility in games: Motivations and approaches*. *G3ict: The Global Initiative for Inclusive ICTs*. Available at: https://g3ict.org/publication/igda-accessibility-in-games-motivations-and-approaches. Accessed 10 February 2024.

Igda-gasig. n.d. *Demystifying CVAA*. Available at: https://igda-gasig.org/what-and-why/demystifying-cvaa/. Accessed 5 March 2024.

IGN. 2020. Video Game Dictionary Guide. Available at: https://www.ign.com/wikis/video-game-dictionary/AAA_(triple-A). Accessed 27 March 2024.

John French. 2023. *Input in Unity made easy (complete guide to the new system)*. Available at: https://gamedevbeginner.com/input-in-unity-made-easy-complete-guide-to-the-new-system/

Johnson P. 2019. *GDC 2019: INDUSTRY DISCUSSIONS ON GAME ACCESSIBILITY*. Available at: https://igda-gasig.org/2019/05/15/gdc-2019-industry-discussions-on-game-accessibility/. Accessed 9 March 2024.

Ken Lee. 2020. *New Unity Input System: Getting Started*. Available at: https://www.kodeco.com/9671886-new-unity-input-system-getting-started.

Kodeco. 2020. *Introduction to Shaders in Unity*. Available at: https://www.kodeco.com/5671826-introduction-to-shaders-in-unity. Accessed 5 March 2024.

Kuster, S.M., van Weerdenburg, M., Gompel, M. and Bosman, A.M., 2018. *Dyslexie font does not benefit reading in children with or without dyslexia. Annals of dyslexia*, *68*, pp.25-42. Available at: https://link.springer.com/article/10.1007/s11881-017-0154-6. Accessed 5 March 2024.

Luis Ramirez Jr. 2023. *Unity's New Input System (+ How To Use It!)*. Available at: https://zerotomastery.io/blog/unity-new-input-system/

Main Leaf. 2023. *How to port a game: an initial 6-step guide*. Available at: https://mainleaf.com/how-to-port-a-game/

Matamala, A. & Orero, P. 2016. *Game Accessibility for the Blind: Current Overview and the Potential Application of Audio Description as the Way Forward*. Available at: https://ebookcentral-proquest-com.ezproxy.centria.fi/lib/cop-ebooks/reader.action?docID=4719968&ppg=91. Accessed 4 March 2024.

Xbox. 2021. *Gaming and Disability Player Experience Guide*. Available at: https://learn.microsoft.com/en-us/gaming/accessibility/gadpeg. Accessed 10 February 2024

Xbox. 2023. *Xbox Accessibility Guidelines V3.2*. Available at: https://learn.microsoft.com/en-us/gaming/accessibility/guidelines. Accessed 4 March 2024.

Xbox. n.d. *Xbox Adaptive Controller*. Available at: https://www.xbox.com/fi-FI/accessories/controllers/xbox-adaptive-controller. Accessed 4 March 2024.

O'malley, M.K. and Gupta, A., 2008. *Haptic interfaces. HCI beyond the GUI: Design for Haptic, Speech, Olfactory, and other nontraditional Interfaces*, pp.25-64. Available at: https://www.researchgate.net/publication/279616558_Haptic_Interfaces. Accessed 4 March 2024.

Pav Creations. 2021. *Jumping controls in 2D pixel-perfect platformers*. Available at: https://pavcreations.com/jumping-controls-in-2d-pixel-perfect-platformers/. Accessed 4 March 2023.

Perry A. 2022. *Video games taught me I was colorblind but it's not always a friendly lesson*. Available at: https://mashable.com/article/colorblindness-video-games-accessibility. Accessed 5 March 2024.

Pingle, Yuriy Denisyuk. 2021. *How to Port a PC game to Android: Pingle's Experience*. Available at: https://pinglestudio.com/blog/co-development/how-to-port-a-pc-game-to-android-pingles-experience.

Powers, G., Nguyen, V. and Frieden, L., 2015. *Video game accessibility: A legal approach. Disability Studies Quarterly*, 35(1). Available at: https://dsq-sds.org/index.php/dsq/article/view/4513/3833. Accessed 9 March 2024.

Prelock, P.A., Hutchins, T. and Glascoe, F.P., 2008. *Speech-language impairment: how to identify the most common and least diagnosed disability of childhood. The Medscape Journal of Medicine*, 10(6), p.136. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2491683/#R4. Accessed 27 March 2024.

Purslow M. 2023. *Final Fantasy 16 has Accessibility Items Instead of Difficulty Levels*. Available at: https://nordic.ign.com/final-fantasy-xvi/64633/news/final-fantasy-16-has-accessibility-items-instead-of-difficulty-levels. Accessed 8 March 2024.

Rct AI. *Research on Video Game Interactivity*. 2020. Available at: https://rctai.medium.com/research-on-video-game-interactivity-460df8f42791. Accessed 10 February 2024

S. Cullipher, S. J. R. Hansen, J. R. VandenPlas. 2018. *Eye Tracking as a Research Tool: An Introduction*. Available at: https://pubs.acs.org/doi/pdf/10.1021/bk-2018-1292.ch001

Sacco M. 2023. *Unity Asset Store*. Available at: https://www.occasoftware.com/blog/unity-asset-store. Accessed 5 March 2024.

Scope UK. 2022. *Accessibility in gaming report*. Available at: https://www.scope.org.uk/campaigns/research-policy/accessibility-in-gaming/. Accessed 10 January 2024.

Seo, S.D. and Kang, S., 2019. *A Comparison Study of the Smartphone Gaming Control. Journal of Usability Studies*, 14(4). Available at: https://uxpajournal.org/comparison-study-smartphone-gaming-control/

Sinclair B. 2019. *Ubisoft sees high acceptance for opt-out subtitles*. Available at: https://www.gamesindustry.biz/ubisoft-sees-high-acceptance-for-opt-out-subtitles. Accessed 5 March 2024.

Skerry L., Unity. GAconf USA 2023. *Unity's Screen Reader Support*. Video on YouTube. Available at: https://youtu.be/TGO_SkDLoJA?si=_eWmbse0FpPDlDpf. Accessed 4 March 2024.

Somatic Labs. 2016. *How do devices provide haptic feedback*? Available at: https://medium.com/@SomaticLabs/how-do-devices-provide-haptic-feedback-1b20529238a. Accessed 4 March 2024.

Sony Entertainment. n.d. *Access controller*. Available at: https://www.playstation.com/en-fi/accessories/access-controller/. Accessed 4 March 2024.

Special Effect DevKit. 2023. *Introduction to Analog Sensitivity*. Available at: https://specialeffectdevkit.info/input/4_analog_sensitivity/4_1_introduction_to_analog_sensitivity/. Accessed 5 March 2024.

Thompson, N.A., Ehrhardt, N., Lippincott, B.R., Anderson, R.K. and Morris, J.T., 2021. *Survey of User Needs: eGaming and People with Disabilities. The Journal on Technology and Persons with Disabilities*, *157*. Available at: https://scholarworks.calstate.edu/downloads/02871366t. Accessed 27 March 2024.

Tobii. 2024. *How do Tobii eye trackers work?* Available at: https://connect.tobii.com/s/article/How-do-Tobii-eye-trackers-work?language=en_US

U.S. Department of State, American English. No Date. *Video games: The New Media Literacy*. Available at: https://americanenglish.state.gov/files/ae/resource_files/week_2_-_video_games.pdf. Accessed 10 February 2024

Unity Documentation. 2023. *Camera*. Available at: https://docs.unity3d.com/2023.3/Documentation/ScriptReference/Camera-fieldOfView.html. Accessed 5 March 2024.

Unity Documentation. 2023. *Prefabs*. Available at: https://docs.unity3d.com/2023.2/Documentation/Manual/Prefabs.html. Accessed 27 March 2024.

Unity Documentation. 2023. Asset workflow. Available at: https://docs.unity3d.com/2023.2/Documentation/Manual/AssetWorkflow.html. Accessed 27 March 2024.

Unity Manual. 2023. *Audio Mixer*. Available at: https://docs.unity3d.com/2023.3/Documentation/Manual/AudioMixer.html. Accessed 5 March 2024.

Unity, Perdigueiro J., Oke S. 2024. *A look at mobile screen reader support in the Unity Engine*. Available at: https://blog.unity.com/engine-platform/mobile-screen-reader-support-in-unity. Accessed 4 March 2024.

Unity. n.d. *Gamepad Support*. Available at: https://docs.unity3d.com/Packages/com.unity.inputsystem@1.7/manual/Gamepad.html.

Vive Developers. n.d. *Getting Started With Accessibility: Adding Subtitles*. Available at: https://developer.vive.com/resources/accessibilty/tutorials/getting-started-with-accessibility-adding-subtitles/. Accessed 5 March 2024.

Vrtonung. n.d. *Personalized Spatial Audio – The holy grail called HRTF*. Available at: https://www.vrtonung.de/en/personalized-spatial-audio-hrtf/. Accessed 5 March 2024.
Watterson C. 2022. *An Indie Approach to Accessibility*. Video on YouTube. Available at: https://www.youtube.com/watch?v=6RAuDpy994A&list=PLVEo4bPIUOsmaH73s5-9N0cKefJJvSe9M&index=15. Accessed 26 March 2024.

WCAG Techniques. 2023. *Technique G18*. Available at: https://www.w3.org/WAI/WCAG21/Techniques/general/G18. Accessed 5 March 2024.

Wery, J.J. and Diliberto, J.A., 2017. *The effect of a specialized dyslexia font, OpenDyslexic, on reading rate and accuracy. Annals of dyslexia*, 67, pp.114-127. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5629233/. Accessed 5 March 2024.

Westin, T., J. J. Ku, J. Dupire, and I. and Hamilton. 2018. "*Game Accessibility Guidelines and WCAG 2.0 – A Gap Analysis*." *International Conference on Computers Helping People with Special Needs*. Linz Austria. 270-279.

Wilde T. 2023. *Why you're right to hate motion blur in games (but devs aren't wrong to include it)*. Available at: https://www.pcgamer.com/why-people-hate-motion-blur-in-videogames/. Accessed 5 March 2024.

Wobbrock, J.O., Morris, M.R. and Wilson, A.D., 2009, April. *User-defined gestures for surface computing. In Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1083-1092).

Xbox Project Acoustics. 2023. *What is Project Acoustics?* Available at: https://learn.microsoft.com/en-us/gaming/acoustics/what-is-acoustics#setup. Accessed 5 March 2024.

Yuan, B., Folmer, E., & Harris, F. C. 2011. *Game accessibility: a survey. Universal Access in the information Society*, 10, 81-100. Available at: https://www.cse.unr.edu/~fredh/papers/journal/29-gaas/paper.pdf. Accessed 10 January 2024.

Zieliński A. 2023. *The Importance of Subtitles in Games: Why 95% of Players Use Them?* Available at: https://vocal.media/gamers/the-importance-of-subtitles-in-games-why-95-of-players-use-them. Accessed 5 March 2024.

Zucconi A. 2015. *Accessibility Design: Color Blindness.* Available at: https://www.alanzucconi.com/2015/12/16/color-blindness/. Accessed 5 March 2024.

Zucconi A. 2021. *Accessibility in Videogames*. Available at: https://www.alanzucconi.com/2021/05/24/accessibility-in-videogames/. Accessed 5 March 2024.