# Augmented Reality Social Storybook: AR Social Stories for Individuals with ASD

Clodagh Murphy - 120378826

Final-Year Project - Digital Humanities and Information Technology

Supervisor Mr David Murphy

College of Arts, Celtic Civilisation and Social Sciences
University College Cork

April 2024

# Abstract

This project aims to improve social skills training for individuals with Autism Spectrum Disorder (ASD) by developing an Augmented Reality (AR) application that overcomes the limitations of traditional, less interactive methods. The application utilises AR.js technology and brings social storybooks to life with interactive 3D models that enhance learning through a multi-sensory experience. The project focused on User-Centred Design and accessibility, resulting in two application variants, one with text boxes and the other with voiceovers. After a comprehensive evaluation, the voiceover version was found to be preferred by users, with high System Usability Scale (SUS) and User Satisfaction Evaluation (USE) scores that validate its effectiveness and user satisfaction. The evaluation also highlighted the significance of auditory feedback in boosting engagement and comprehension. The project included stability testing, performance optimisation, and user feedback to ensure the application's quality and accessibility.

The project findings demonstrate the potential of integrating AR with social stories to enhance users' educational experience significantly. The project highlights AR's ability to engage users more deeply and suggests a promising direction for the future of special education tools, emphasising the need for further exploration in this area.

# Acknowledgements

I am immensely grateful to my supervisor, David Murphy, for his exceptional guidance, patience, and support throughout this project. His expertise and insights have been fundamental in its development and success.

A special thank you to SkillsBase for sponsoring this project. I extend my gratitude to Mark Horgan, whose expert advice on report writing and insightful feedback on the prototype has been invaluable. Similarly, I sincerely appreciate Steven Scully for his thoughtful feedback and guidance on the prototype and the design of the storybook. Their contributions have been instrumental in the development and refinement of this project.

I am also grateful to my coursemates, friends, and family, who participated in the user testing phase and offered constructive feedback that was vital for improving the application. Their willingness to contribute their time and thoughts has been greatly appreciated.

# Declaration of Originality

In signing this declaration, you are conforming, in writing, that the submitted work is entirely your own original work, except where clearly attributed otherwise, and that it has not been submitted partly or wholly for any other educational award. I hereby declare that:

- this is all my own work, unless clearly indicated otherwise, with full and proper accreditation;
- with respect to my own work: none of it has been submitted at any educational institution contributing in any way to an educational award;
- with respect to another's work: all text, diagrams, code, or ideas, whether verbatim, paraphrased or otherwise modified or adapted, have been duly attributed to the source in a scholarly manner, whether from books, papers, lecture notes or any other student's work, whether published or unpublished, electronically or in print.

Signed ........C. M........ Date ....26/04/2024....

Clodagh Murphy

# Contents

# List of Figures

# Listings

# Chapter 1

# Introduction

## 1.1 Problem Statement

Individuals with Autism Spectrum Disorder (ASD) often face unique challenges in social and educational contexts that require specialised approaches to learning and interaction. Traditional social skills learning methods, such as social stories, utilise stories with accompanying visuals to teach behavioural norms and social cues. While beneficial, these approaches have limitations. They lack interactivity and engagement, which are crucial factors for capturing and retaining the attention of individuals with ASD. This project seeks to address these challenges by leveraging Augmented Reality (AR) technology to enhance the learning experience, making it more interactive and accessible for individuals with ASD.

## 1.2 Objectives

The primary objective of this project is to design and develop an AR application utilising AR.js to enhance the social skills education experience for individuals with ASD. Specifically, the application seeks to:

- Enhance social skills learning for individuals with ASD through AR-enhanced interactive and engaging social stories.
- Provide a user-friendly interface that ensures accessibility and ease of use.
- Leverage AR.js and AR technology to create immersive and interactive learning experiences that captivate and educate.

## 1.3 Terminology Used

1. **Autism Spectrum Disorder (ASD):** A neurological and developmental disorder impacting interpersonal interactions, communication, learning, and behaviour. Recognised as a "spectrum" disorder, autism exhibits a diverse range of symptoms in terms of type and severity across individuals [1].

2. **Social Skills Training (SST):** An evidence-based approach to assist individuals with ASD in gaining a better understanding of social interactions and improving their ability to engage socially with others [2].

3. **Social Stories:** A tool for social learning, facilitating the secure and meaningful exchange of information among parents, professionals, and individuals of all ages on the autism spectrum [3].

4. **Augmented Reality (AR):** Augmented reality is an enhanced and interactive version of real-world surroundings, achieved by integrating digital visual elements, sounds, and other sensory stimuli through holographic technology. Three key elements characterise AR: combining digital and physical worlds, real-time interactive experiences, and precise 3D identification of virtual and natural objects [4].

5. **Virtual Reality (VR):** Virtual reality represents a computer-generated 3D space that allows individuals to navigate and engage with a digital world that mimics real-life experiences the user senses. This virtual setting is constructed using computer technology, and interaction may require specialised equipment like headsets or glasses [5].

6. **Pattern Marker:** Custom markers generated from user-uploaded images. These images are typically straightforward with high contrast and serve as the basis for creating custom markers [6].

7. **User Interface (UI):** The user interface serves as the bridge for interaction and communication between a person and a computer within a device. It encompasses elements such as display screens, keyboards, a mouse, and the visual design of a desktop. UI also describes how users engage with software applications or websites [7].

8. **User Experience (UX):** User experience includes all the aspects of the end-user's interaction with a company, its services, and its products [8].

9. **GL Transmission Format (glTF):** glTF™ (GL Transmission Format) is a standard format for efficiently transmitting and loading 3D models across the web and native applications [9].

10. **QR Code:** A QR code, standing for "Quick Response code," consists of a pattern of black and white pixels in a square grid that stores data for a machine to read. Smartphones and cameras can rapidly decode the data encoded in the unique pixel pattern of a QR code. This makes QR codes a practical tool for quickly storing and retrieving information [10].

## 1.4 Outline

The remainder of this report is organised as follows:

- **Chapter 2** Analysis. This section reviews existing research and projects related to the use of AR in educational and skill training settings and the challenges faced by individuals with ASD.
- **Chapter 3** Design. Discuss the design approach for the AR application, including principles, development tools, and user interface considerations.
- **Chapter 4** Implementation. This section details the AR application's development process, including coding, modelling, animating, integrating interactive elements, and testing.
- **Chapter 5** Evaluation & Testing. This section discusses the methods used to assess the AR application's usability and design, analyses the findings, and suggests areas for further research.
- **Chapter 6** Conclusion. Summarises the findings and contributions of the project.

# Chapter 2

# Analysis

## 2.1 Research Objectives and Goals

The purpose of the analysis section is to:

1. Examine existing literature regarding the various applications of AR in addressing the needs of individuals with ASD, with a specific emphasis on social skills training.
2. Analyse the User Interface (UI) and design aspects of AR applications discussed in the literature, identifying best practices and challenges in designing interfaces suitable for individuals with ASD.
3. Explore various technologies, including AR libraries and frameworks, to comprehensively understand the tools available for creating and designing AR applications.
4. Gather all the information to design a complete AR social story application.

The research findings should dictate the design of the final AR app, influencing its functionalities and features.

## 2.2 Literature Review

### 2.2.1 Evaluating user interface of a mobile augmented reality colouring application for children with autism: An eye-tracking investigation

This study [11] examines the accessibility of AR applications for children with ASD through a detailed analysis of the UI design. The study uses eye-tracking technology to explore the usability of the AR mobile app wARna, which was created to help children with ASD express their emotions and learn social skills. The study's main objective is to understand how children with ASD interact with the app in contrast to typically developing children, providing essential insights for designing AR applications that are tailored to the needs of this particular user group.

The research involved 23 children with ASD and 26 typically developing children tasked with interacting with 13 AR app interfaces. The results of the eye-tracking study showed that children with ASD engage with mobile AR interfaces differently from their typically developing peers. Notably, text comprehension tasks presented difficulties for participants with ASD. It was also noted that participants with ASD tended to be more distracted by unrelated visual elements and that they preferred icons to text.

The study emphasises the importance of tailored UI design for individuals with ASD, considering their varied information-processing methods, sensory preferences, unique visual preferences, and information-seeking techniques. The studies recommended accessibility design considerations include:

1. **Make Key Elements Big:** Enlarge key elements like buttons, strategically placing them for intuitive navigation.
2. **Minimise Visual Distractions:** Prioritise a clean design, minimising visual distractions like excess buttons, icons, or text.
3. **Keep text Brief and Concise:** Use simple, concise sentences; emphasise visuals like images and icons to convey instructions.
4. **Use Clear and Simple Elements:** Select universally recognised icons, ensuring clarity through usability tests with ASD children.
5. **Provide Customization Options:** Provide users with ASD customisation options, adapting sound, visuals, and features to individual preferences.

6. **Minimise Auditory Distractions:** Reduce background noise, use gentle sound cues, and ensure soothing sound overstimulation.

7. **Include Interactive Features:** Incorporate AR technology, games, or interactive elements to engage and stimulate learning.

**Remarks**

- The study provides a foundation for thoughtful and inclusive design practices when designing accessible mobile AR applications for individuals with ASD.

- A design strategy can be guided by recognising the benefits mentioned in the study's discussion of related projects, especially the possibility of improving social, behavioural, learning, and attention skills among individuals with ASD. The study emphasises the importance of appropriately designing multimedia elements like interface graphics, animations, and text. The emphasis on keeping screen elements in an AR app simple and avoiding extraneous background distractions will influence my app in the design process.

### 2.2.2 How to Use the Advantages of AR and VR Techniques to Integrate Special Visual Training Strategies in Non-Verbal Communication Skills Training for Children with Autism

This study [12] discusses using AR and VR technologies to enhance non-verbal communication skills training for children with ASD. It highlights these children's challenges in social interactions and how A/VR technologies can create immersive learning experiences to improve social reciprocity skills. Lee suggests that A/VR offers a novel training framework beyond traditional methods, providing interactive and engaging ways to teach social cues through visual simulations.

The study examines various ways and methods that AR technologies can enhance non-verbal communication skills. The paper emphasises how AR technologies can enhance autistic children's attention and mastery of social cues, provide visual interpretations of abstract social concepts, and support understanding of metaphorical social relationships. It can deconstruct social concepts using visualisation and images, extend sensory levels of static images, and give new life to role-playing games by superimposing 3D animation and situational sound, aiding in overcoming weak imagination in autism. AR can be effective in teaching through additional visual information. They can be used in pretend and symbolic play, helping autistic children practice social reciprocity and understand social status from different perspectives.

**Remarks**

- The study described how AR can enhance social stories for autistic children by providing visual primers and a structured visual framework. It can extend sensory experiences beyond static images, offering animated responses and dynamic video overlays on social storybooks. This can improve children's understanding of social interactions, such as handshakes or eye contact, and increase their attention and motivation.
- AR technology aids in deconstructing social films and storybooks into visual elements, increasing attention and motivation and offering visual interpretations of abstract concepts, which can be challenging for autistic children to grasp through traditional methods.
- It is essential to use a fixed visual framework, such as a social storybook, to guide autistic children through the main structure of social stories and then overlay dynamic animations to elaborate on complex social cues and contexts. This approach helps to reduce cognitive load and enhance understanding and attention, making the learning experience more engaging.

### 2.2.3 Augmented reality-based video-modeling storybook of nonverbal facial cues for children with autism spectrum disorder to improve their perceptions and judgments of facial expressions and emotions

The study [13] addresses the effectiveness of AR in teaching social skills to children with ASD by using an AR-based video-modelling storybook (ARVMS). The ARVMS system includes a static image layer (storybook) and an augmented dynamic video layer, using the Vuforia platform and Unity API to create an AR environment. It features 20 video fragments corresponding to freeze-frames from a story, focusing specifically on social cues. The system overlays video clips on a storybook to enhance focus on social interactions using markerless tracking.

The ARVMS system enhances the learning experience for children with ASD by actively engaging them with dynamic visual videos and emotional expressions of characters, which helps them better judge facial expressions and social cues than static storybooks.

It attracts their attention, encourages observation of nonverbal signals, and improves social skills, making the learning process more exciting and compelling. This study showed significant improvements in the children's ability to pay attention to relevant parts of the videos and understand the characters' emotional states.

**Remarks**

- ARVMS displays repeated video fragments by displaying videos that last 30-45 seconds and repeat at least six times before timing out. This approach is designed to help direct focus on specific social cues that may be hard to detect. This repeated exposure reinforces the user's attention, allowing them to better understand and recognise these social cues through repetition.
- The study found that AR technology effectively attracted children's attention and maintained their focus on essential parts of the story, emphasising the importance of focusing on specific visual representations and facial cues.

### 2.2.4 Augmented Reality for Learning of Children and Adolescents With Autism Spectrum Disorder (ASD): A Systematic Review

The paper [14] is a systematic literature review (SLR) focusing on AR applications for children with ASD. The review provides a comprehensive evaluation of primary studies that investigate the effectiveness of AR in various learning and developmental contexts for children with ASD. The review covers various aspects such as the skills targeted, participant characteristics, technologies used, research designs, data collection methods, settings, evaluation parameters, and the overall evaluation process. Key findings highlight the growing trend of using AR for ASD interventions since 2014 and the need for future research to address the limitations found in the primary studies.

The review discusses the importance of technology integration for AR interventions, emphasising that due to the heterogeneity of impairments and symptoms among participants with ASD, there is no one-size-fits-all technology. Technologies like smartphones and tablets are discussed for their portability and built-in cameras, which facilitate interactions with physical objects while attracting the attention of children with ASD. AR-based applications also provide multimodal interaction, which can be beneficial for teaching various skills during intervention or therapy sessions. The paper also outlines the use of AR in teaching specific skills such as navigation, pretend play, handling plants, literacy, cooking, recognising facial expressions and emotions, and social communication skills.

**Remarks**

- The market for AR is expected to grow, and researchers are exploring different technologies to find the best fit for individuals with ASD.
- The review acknowledges that technology is not one-size-fits-all, especially for individuals with diverse needs. It discusses various technologies available, such as glass- and projection-based technologies. However, it emphasises that due to cost and availability, smartphones and tablets have become an "obvious choice" for use.

### 2.2.5 Features of implementation of modern AR technologies in the process of psychological and pedagogical support of children with autism spectrum disorders

The paper [15] analyses the implementation and use of AR technologies in the psychological and pedagogical support of children with ASD by reviewing existing scientific literature, clinical protocols, and educational methods to identify suitable AR technologies for a comprehensive support program. The paper outlines the criteria used for selecting AR technologies for psychological and pedagogical support of children with ASD, emphasising the importance of accessibility, instructiveness, understandability, visualisation, complexity, availability of correction-developmental and psychological construct, logic, systematic and structural properties, a straightforward interface, the ability to implement an individual approach, and multi-functionality. These criteria ensure that the selected AR technologies are adequate and suitable for supporting children with ASD in their development and communication needs.

The paper's findings indicate that AR technologies can significantly improve communication, cognitive, emotional-volitional, and mnemonic abilities in children with ASD. Additionally, these technologies enhance adaptive potential and promote socially accepted behaviours. They also suggest that combining AR technologies with traditional psychological and pedagogical methods can individualise and maximise the effectiveness of corrective actions. The paper also suggests that further research should focus on developing a comprehensive model for implementing AR technologies tailored to the individual needs of children with ASD.

The paper examines a variety of existing AR applications, of which all have different purposes aimed at supporting children with ASD. Some of the applications discussed in the paper include SceneSpeak for creating interactive displays and stories, Milo language for developing communication skills, and "Animated Visual Amplifiers for Social Skills" (AViSSS) for practising social skills in different environments. The paper also examines work by Chung et al. [pg 266] discussing the use of AR for visualising social stories to adjust social interaction deficit and develop soft skills in children with ASD. It describes interactive social stories played with tangible markers and AR technologies corresponding to virtual images. Other research discussed in the paper also suggests the use of three-dimensional (3-D) animation to simulate emotional expressions on faces, aiming to develop the emotional spectrum and social skills of autistic adolescents.

#### Remarks

- The paper highlights the effectiveness of AR technologies in supporting children with ASD, particularly in enhancing communication and social skills.
- The paper emphasises the importance of design features in AR technologies, such as ease of use, accessibility, and compelling content visualisation.
- Additionally, it highlights the need for AR developers to consider and apply the principles of web content accessibility as outlined in the ISO/IEC 40500:2012 standards by the W3C. It recommends considering the principles of Universal Design for Learning (UDL) to visualise content further.

## 2.3 Background Research

The development of the AR application required several technologies, including HTML [16], CSS [17], and JavaScript [18]. The application's construction of AR scenes and interactions utilised the A-Frame [19] and AR.js [20] libraries, which provided a comprehensive framework for creating AR content directly in the web environment. The project's code and digital assets were hosted and managed using Glitch [21]. Blender [22] created, animated and exported 3D models in the glTF [23] format. These tools and technologies are described in more detail below.

### 2.3.1 Web Technologies

In web development, HTML, CSS, and JavaScript are three fundamental technologies that play a vital role in creating and enhancing user experiences online. HTML is the standard markup language for structuring and laying out web pages. It operates as the standard framework for designing web pages, with each HTML element providing specific instructions to the browser regarding content display. This element-based framework is crucial for developers to design consistent and organised web pages, facilitating easy online creation and dissemination of content online [24].

CSS complements HTML by managing the presentation aspects of web pages. It separates content from design, allowing developers to control layouts, colours, and fonts. This separation enhances website flexibility, making it responsive to various device screens and user preferences. CSS is instrumental in developing visually appealing interfaces adaptable to different viewing environments, offering extensive properties and capabilities to create rich, interactive user experiences [17].

JavaScript, a dynamic programming language, is integral to implementing interactive features within web pages. Known for its lightweight and versatile nature, it supports various programming paradigms, including object-oriented, imperative, and declarative approaches. JavaScript's functionality extends beyond browsers and is executable in non-browser environments like Node.js. It enhances user interactions through animations, form validations, and dynamic content modification, adding depth to the web browsing experience [18].

### 2.3.2 A-Frame

A-Frame is an open-source web framework for constructing VR experiences. Built on top of HTML, A-Frame offers a straightforward entry point into VR development. However, it is more than a mere 3D scene graph or markup language; A-Frame is a robust entity-component framework at its core. This framework delivers declarative, extensible, and composable structure to three.js [25].

A-Frame is a versatile and accessible framework that leverages the simplicity of declarative HTML to ensure readability and ease of use by employing a robust entity-component architecture rooted in three.js, offering a declarative, composable, and reusable structure that extends beyond HTML. While HTML provides a user-friendly starting point, developers have extensive access to JavaScript, DOM APIs, three.js, WebVR, and WebGL, facilitating the creation of intricate and customisable VR content.

A-Frame's emphasis on performance optimisation for WebVR is evident in its design, ensuring smooth operation even in the most interactive and large-scale applications. By minimising interaction with the browser layout engine and handling 3D object updates efficiently in memory, A-Frame consistently delivers high-quality VR experiences, with applications running seamlessly at 90fps [25].

### 2.3.3 AR.js - Primary Project Technology

AR.js is a powerful and open-source JavaScript library designed to create AR experiences on the web. It is built on top of A-Frame, three.js, and ARToolKit. By harnessing the capabilities of WebGL and WebVR, AR.js can run AR experiences smoothly directly in a web browser [20].

The key to AR.js lies in its use of the three.js library, which forms the backbone of its 3D modelling capabilities. By integrating three.js, AR.js gains access to a rich suite of complex object interactions. This integration allows developers to implement detailed, interactive 3D objects within AR scenes, enhancing the realism and engagement of AR applications. The source code for AR.js includes specific adaptations for three.js, such as custom components for marker-based and image-tracking AR scenarios, which are crucial for the library's function and versatility. Additionally, AR.js's structure atop A-Frame, an entity-component framework with three.js at its core, further simplifies AR development. This arrangement enhances flexibility and ease of use and empowers developers to write AR content using HTML-like syntax, which is familiar to many web developers [20].

AR.js adopts a marker-based approach, where entities are affixed to physical markers within the user's environment rather than linked to a generated surface of points [26]. While the marker-based design is acknowledged as one of the more limiting factors of AR.js, the creative potential exists to leverage this design for more creative AR interactions. This marker-based AR development system is advantageous in educational settings like museums and classrooms. Participants in these environments can actively engage with the AR display by physically relocating markers, creating an interactive and immersive learning experience. When the camera detects a marker, AR.js enables content presentation, similar to Image Tracking. While markers offer stability, they have limitations in shape, colour, and size. This design is recommended for scenarios requiring numerous markers with diverse content, making it suitable for applications like augmented books, flyers, and advertising [20].

The library supports other AR experiences besides marker tracking, including image and location-based AR. With Image Tracking, the camera identifies 2D images, enabling the overlay of diverse content like 2D images, GIFs, animated 3D models, and 2D videos. Location-based AR takes advantage of real-world places, allowing users to experience AR content tied to their locations. Users moving outdoors can view AR content changing dynamically according to their positions and rotations, offering applications like interactive tourist guides, city exploration assistance, educational treasure hunts, and situated art experiences linked to specific real-world coordinates [20].

### 2.3.4 Glitch

Glitch is a user-friendly, collaborative platform for coding, hosting, and sharing web applications. It supports real-time collaboration, allowing multiple users to edit code simultaneously, akin to Google Docs for coding. Glitch emphasises community and learning, offering an environment where users can remix existing projects to create something new. This makes it an excellent tool for both education and prototype development. It simplifies developing, deploying, and sharing web applications, making coding more accessible to a broader audience [27].

### 2.3.5 Blender

Blender is a freely available open-source 3D content creation tool that supports the entire 3D creation process, encompassing modelling, rigging, animation, simulation, rendering, compositing, motion tracking, video editing, and game creation [28].

Blender has the tools to effectively design and export 3D models for AR mobile apps. Blender can be used to work on both static and animated 3D models, ensuring the correct configuration of parameters essential for mobile compatibility. This involves considerations like optimising polygon count, texture optimisation, and adhering to specific file formats [29].

### 2.3.6 glTF

"glTF™ is a royalty-free specification for the efficient transmission and loading of 3D scenes and models by engines and applications." [23] glTF minimises the size of 3D assets and the runtime processing needed to unpack and use them. glTF defines an extensible publishing format that streamlines authoring workflows and interactive services by enabling the interoperable use of 3D content across the industry. glTF 2.0 has been released as the ISO/IEC 12113:2022 International Standard." [23]

glTF aims to achieve widespread deployment across various devices and platforms, including the web and mobile devices, with limited processing and memory resources. glTF's evolving flexibility ensures compatibility with advancing computing capabilities, fostering broad industry consensus on universally applicable 3D functionality, including physically based rendering [30].

As illustrated in Figure 2.1, glTF has been developed with the following objectives in mind [30]:

- **Compact file sizes:** The plain text glTF JSON file description is compact and rapid to parse. Substantial data like geometry, textures, and animations are stored in binary files that are more space-efficient than their equivalent text representations to achieve smaller file sizes.
- **Runtime independence:** As an asset format, glTF imposes no requirements on runtime behaviour. This flexibility allows it to be used by any application for various purposes.
- **Complete 3D scene representation:** Not restricted to individual objects, glTF can depict entire scenes, including nodes, transformations, transform hierarchy, meshes, materials, cameras, and animations.
- **Extensibility:** glTF is fully extensible, allowing for incorporating both general-purpose and vendor-specific extensions, such as those for geometry and texture compression.

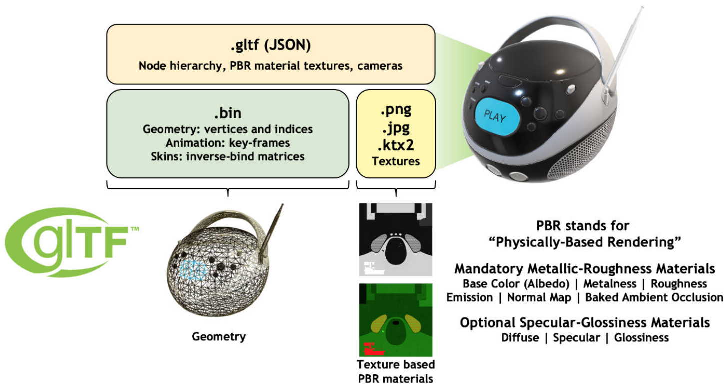



Figure 2.1: glTF Scene Description Structure [23]

# Chapter 3

# Design

This chapter outlines the rationale behind each design decision. The primary objective was to craft an interface and user experience that meets and truly understands the unique requirements of children and young adults with ASD. To achieve this, the design had to be sensitive to their needs while appealing and easy for children. Emphasis was given to creating a straightforward but thoughtfully engineered design to avoid unintentionally triggering distress or episodes.

## 3.1 Exploring Potential Development Pathways

Exploring potential development pathways was crucial when considering how to develop the application. The goal was to identify a platform aligning with the design principles essential for the target audience, primarily children and young adults with ASD. These users required a platform that was not only child-friendly and ASD-friendly but also simple yet carefully designed to navigate and mitigate potential triggers effectively.

When considering developing the AR application, a comprehensive evaluation of several options was conducted to ensure the best possible choice for the final application:

- **Game Engines:** Initially, creating the application using a game engine like Unity was appealing due to its powerful features and immense documentation. However, game engines often present stability issues and require significant time to overcome a steep learning curve. The resulting application might not provide the universally smooth experience necessary for this application's target audience [31].

- **Native Applications:** Developing native applications for each operating system, such as iOS and Android, can optimise users' experiences. However, this approach requires creating and managing separate code bases and interfaces for each platform. The learning curve for native languages, such as Swift for iOS, can add complexity to the development process and extend the project timeline beyond its scope [32].

- **Web Applications:** Although web technology may not provide all the features that native applications offer, it has its unique advantages. Web applications are compatible with different devices and operating systems, which improves interoperability. Additionally, web applications benefit from continuous deployment, removing the need for users to update them manually. With the introduction of web standards like WebXR, web applications can now be accessed by a wide range of users with minimal barriers to entry [33].

After considering all the options, it was decided that the web was the most suitable platform for this project. The advantages of web technologies, such as scalability, ease of deployment, adherence to standards, accessibility, and interoperability, far outweigh those offered by game engines and native applications. A web-based approach aims to create an inclusive and universally accessible application that can reach users regardless of their device or operating system [34]. The design process, hence, was oriented around leveraging web technologies to create an application that is as accessible and user-friendly as possible while still delivering an engaging, interactive AR experience.

## 3.2 AR Architecture

The architecture of AR is a multifaceted system that seamlessly integrates various components, each playing a crucial role in delivering immersive experiences. This architecture, as illustrated in Figure 3.1, is comprised of six components [35]:

1. **User:** The user is at the core of augmented reality. The user is also responsible for creating AR models.
2. **Interaction:** This involves the communication between the device and the user. The term itself implies actions performed by one entity that results in the creation or response from another entity.
3. **Device:** The device is responsible for creating, displaying, and interacting with 3D models. It can take various forms, such as mobile phones, computers, or AR headsets.
4. **Virtual Content:** This refers to the digital information, like 3D models, textures, text, and images, created or generated by the system or AR application. This virtual

content can be integrated into the user's real-world environment.

5. **Tracking:** This component involves processes that enable the creation of AR models. Tracking uses algorithms to determine where to place or integrate the 3D models into the real-world environment. Various tracking algorithms can be utilised in the development of AR applications.

6. **Real-life Entity:** These real-world entities can be anything like trees, books, fruits, computers or anything visible on the screen. AR applications don't change the position of real-life entities; instead, they integrate digital information with these entities.

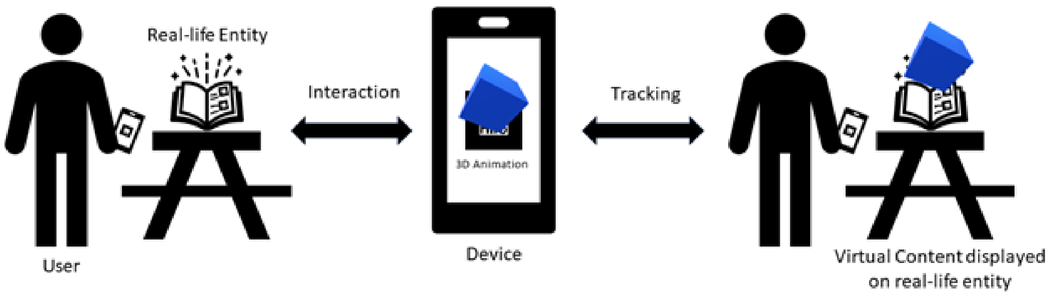



Figure 3.1: AR Architecture

### 3.2.1 Application-Specific AR Architecture

This subsection details the architecture of an AR system tailored for users with ASD. This system intends to deliver an engaging and supportive AR experience by integrating web technologies, user interface design, 3D rendering, and tracking mechanisms.

The architecture diagram illustrated in Figure 3.2 showcases an intricate AR system designed for individuals with ASD. The diagram depicts a clear sequential flow, starting from the user's action of scanning the QR code, progressing through the web browser's interpretation, and finally rendering the AR content. This system is meticulously designed to be a cohesive and uninterrupted process.

The structure of this architecture reflects the sequential flow presented in the diagram, offering an in-depth look at the critical elements tailored for users with ASD:

- **User Interaction:** The user's engagement with the AR system begins upon scanning a QR code. This action opens a web browser, which acts as the portal to the AR

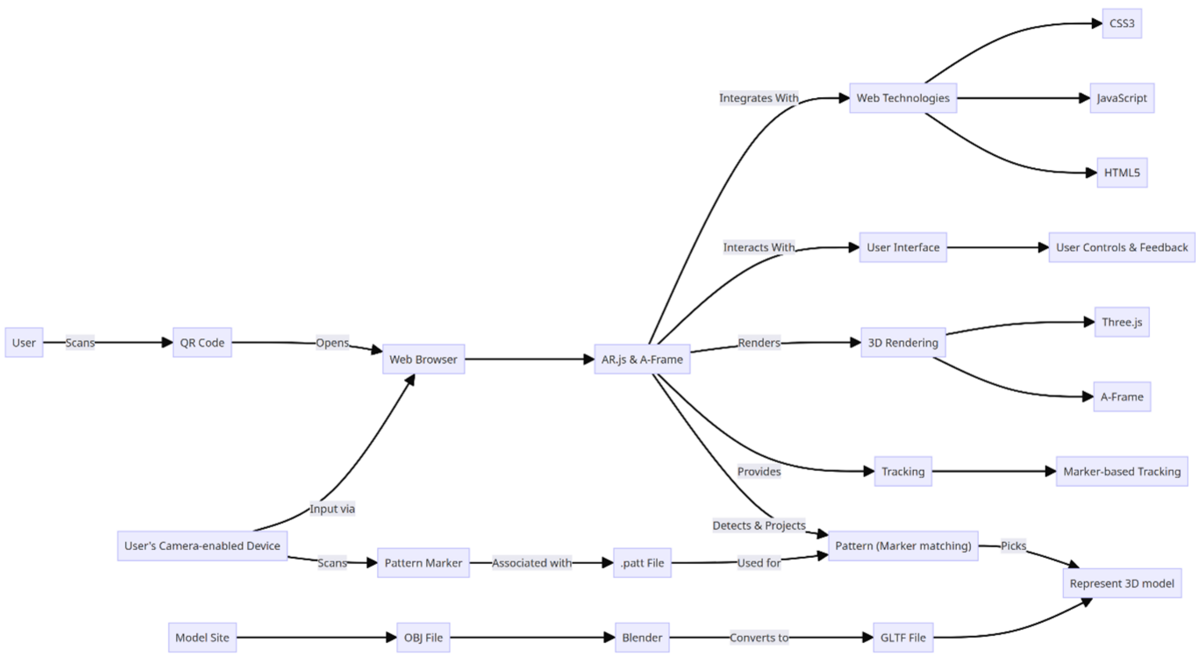



Figure 3.2: Application-Specific Architecture

experience. This interaction smoothly transitions users into the AR environment.

- **Web Technologies:** The integration of CSS, JavaScript, and HTML forms the structural and interactive backbone of the AR system. These technologies ensure the user interface is accessible and engaging, which is crucial for users with ASD and supports diverse learning styles.
- **User Interface:** The user interface design emphasises minimalism and ease of use, with user controls specifically tailored to individuals with ASD. By focusing on reducing cognitive overload, the interface facilitates a more comfortable and less overwhelming experience for the user.
- **3D Rendering:** The system provides rich visualisations within the web environment using libraries such as AR.js, Three.js, and A-Frame. These libraries were selected for their compatibility across various devices and ability to accurately superimpose AR content onto the user's physical surroundings, enhancing the sense of immersion for users with ASD.
- **Tracking:** The system employs marker-based tracking to ensure AR content's precise and stable display. Using pattern markers and corresponding .patt files, the AR visuals are seamlessly projected onto the user's environment. This consistency is vital in helping users with ASD maintain focus and interact meaningfully with the content.
- **Model Integration:** 3D models are initially sourced in OBJ format and converted

to glTF format through Blender to optimise device performance. This step is crucial for maintaining visual fidelity and ensuring a smooth user experience. The system's responsive design and lightweight models guarantee accessibility and usability on most modern devices.

In summary, the design of this AR system is carefully put together to support the unique needs of users with ASD by offering a user-friendly interface, robust web technologies, and high-quality 3D content. The thoughtful integration of these components ensures that the AR system is not only engaging but also highly accessible and responsive.

## 3.3 Design Principles

### 3.3.1 User-Centred Design (UCD)

UCD is an iterative design approach in which designers prioritise users and their needs at every stage of the design process. UCD involves users extensively throughout the design process by employing various research and design methods to develop usable and accessible applications [36].

As illustrated in Figure 3.3, the general phases in the UCD process include [37]:

- **Specify the context of use:** Identify the people who will use the product, what they will use it for, and under what conditions they will use it.
- **Specify requirements:** Identify any business requirements or user goals that must be met for the product to be successful.
- **Create design solutions:** This process may be done in stages, building from a rough concept to a complete design.
- **Evaluate designs:** Evaluation – ideally through usability testing with actual users.

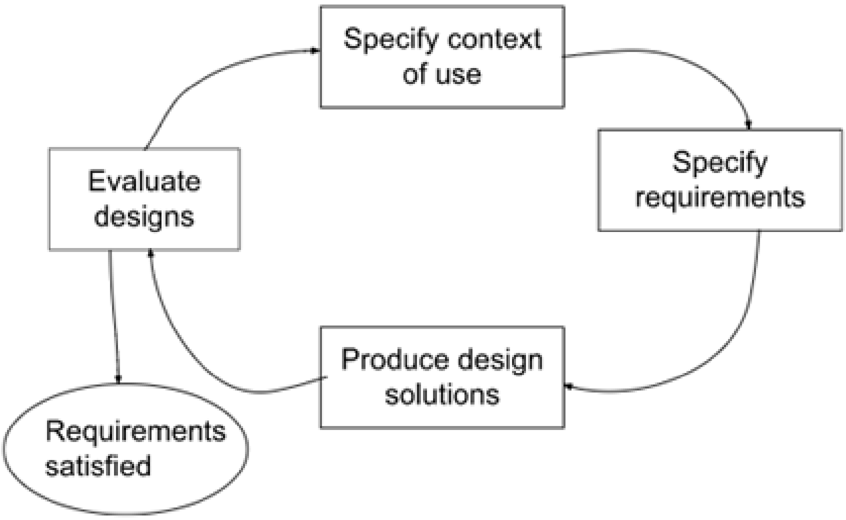



Figure 3.3: UCD Process [35]

### 3.3.2 Universal Design for Learning (UDL) Guidelines

The UDL Guidelines provided by CAST are principles and strategies to improve teaching and learning environments to accommodate all learners. These guidelines are structured around three primary principles, each focusing on a different aspect of learning [38]:

1. **Multiple Means of Engagement (The "WHY" of Learning):** This principle addresses how educators can stimulate interest and motivation for learning. It includes strategies for recruiting interest, sustaining effort and persistence, and fostering self-regulation in learners.

2. **Multiple Means of Representation (The "WHAT" of Learning):** This principle focuses on presenting information and content differently to suit diverse learning styles. It covers guidelines for perception, language and symbols, and comprehension, providing various methods for learners to access and understand information.

3. **Multiple Means of Action and Expression (The "HOW" of Learning):** This principle allows learners to demonstrate what they know in different ways. It includes strategies for physical action, expression and communication, and executive functions, offering multiple options for learners to express and apply their knowledge.

### 3.3.3 Web Content Accessibility Guidelines (WCAG)

WCAG encompass a diverse set of recommendations aimed at enhancing the accessibility of web content. Following these guidelines broadens the accessibility of content for individuals with various disabilities, such as those related to vision, hearing, movement, speech, photosensitivity, and combinations of these. While some considerations are made for learning disabilities and cognitive limitations, it's important to note that these guidelines may not fulfil every user need for individuals with these disabilities [39].

The Web Content Accessibility Guidelines are founded on four distinct principles, and it is essential to consider each of these principles when addressing accessibility in the mobile app development process [40]:

- **Perceivable:** Ensure that all information and components within your mobile app are presented in the most easily perceivable manner.
- **Operable:** Ensure that all components of the app interface, including navigation, are operable for users of varying abilities.
- **Understandable:** Ensure all information and app functionality are comprehensible to users of diverse backgrounds and capabilities.
- **Robust:** Ensure that all content in the mobile app is robust enough to be interpreted seamlessly by various user agents, including assistive technologies.

## 3.4 Enhancing Accessibility

### 3.4.1 HTML Accessibility

HTML plays a vital role in creating accessible web content. Accessibility in the context of HTML refers to designing and coding web pages to make them usable and navigable by a diverse range of users, including those with disabilities. More accessible web pages can be created by implementing:

- **Using good semantics:** According to w3Schools, semantic HTML means "using [the] correct HTML elements for their correct purpose as much as possible." [41] Semantic HTML is particularly significant for screen readers, which read the content of a page to users. Using the correct HTML elements provides context, enhancing the experience for screen readers. Additionally, semantic buttons ensure accessibility for individuals who rely solely on keyboard navigation. Such buttons are seamlessly clickable using both mouse and keys, and users can navigate between them using the tab key on the keyboard. This approach contributes to a more inclusive and user-friendly web environment [41].

- **Providing alternative text:** The `<alt>` attribute gives an alternative text explanation for an image in situations where the user cannot view it [41]. Consider whether an image holds meaningful content within the web page or serves as a visual embellishment without conveying any specific information. If the images are purely decorative, it is advisable to use an empty text as the value for the alt attribute or to incorporate them as CSS background images. If the aim is to provide additional context, embed it within the surrounding text or use a title attribute [42].

- **Declaring the language:** Incorporating the "lang" attribute within the `<html>` tag to specify the web page's language aids search engines and browsers [41].

- **Use clear language:** Use precise language and avoid characters that a screen reader cannot easily read. Use short sentences, refrain from using dashes, and avoid abbreviations and slang words [41].

- **Good link text:** The links, represented by the `<a>` element with an href attribute, can either enhance or impede accessibility based on their use. By default, links possess an accessible appearance and contribute positively to accessibility by facilitating easy navigation to various sections within a document. However, they may harm accessibility if their accessible styling is altered or JavaScript alters their behaviour in unforeseen ways [42].

### 3.4.2 Accessible Rich Internet Applications (ARIA)

Accessible Rich Internet Applications (ARIA) "provides features to define accessible user interface elements and improve the accessibility and interoperability of web content and applications. It is primarily for developers of Web browsers, assistive technologies, and other user agents, developers of Web technologies (technical specifications), and developers of accessibility evaluation tools."[43]

ARIA provides Web authors with the following [43]:

- Roles to describe the type of widget presented, such as "menu", "tree item", "slider", and "progress bar".

- Roles to describe the Web page's structure, such as headings and regions.

- Properties to describe the state widgets are in, such as "checked" for a check box or “read-only” for most form controls.

- Properties to define live regions of a page.
- A way to provide keyboard navigation for the Web objects and events.

While ARIA can significantly improve accessibility by providing additional context to assistive technologies, incorrect use can lead to a poorer experience for users of such technologies. Therefore, its application should be guided by understanding the technology and the users’ needs [43] [44].

## 3.5 Social Story Structure

Social stories present information in a straightforward, ‘literal’ manner, which may improve the comprehension of previously challenging or unclear scenarios. The format and the content are flexible, allowing for customisation to suit diverse individual needs.

They can help organise sequences of activities (sequencing) and enhance executive functioning (planning and organisation). By providing information about the potential outcomes in specific situations and suggesting behavioural guidelines, these stories contribute to establishing structure in a person’s life, subsequently reducing anxiety [45].

### 3.5.1 Composing the Social Story

Incorporating the following key elements is essential to creating a compelling social story [45]:

1. **Picture the Goal:** Identify the purpose of the social story and consider what the individual needs to understand to achieve the goal, emphasising the importance of the behaviour.
2. **Gather Information:** Collect information about the situation addressed in the social story, including location, people involved, initiation, duration, and potential outcomes. Tailor the content to the individual’s interests and age, using age-appropriate visuals and language while also considering the person’s attention span and level of ability and understanding.
3. **Tailor the Text:** Structure the social story with a title, introduction, body, and conclusion. Use gentle and supportive language and descriptive sentences to

accurately depict the context and coaching sentences to guide behaviour. Answer essential questions: where, when, who, what, how and why.

When deciding what social stories to compose, the focus was placed on everyday situations that could be particularly useful for individuals with ASD. The final stories, "Being Nice to Pets" and "Going to the Hairdressers", represent everyday real-life experiences. These scenarios were chosen for their relevance and potential to assist individuals with ASD in understanding and navigating similar situations in their own lives.

### 3.5.2 Physical Social Story Book

Social stories have long served as a fundamental component in social skills training for individuals with ASD by providing a structured format that combines text with visual aids, enhancing comprehension and engagement. These stories traditionally feature images directly above the corresponding text, creating a clear and immediate visual association with the story.

In the design of the educational booklet for AR-enhanced social stories, principles derived from Hick's Law [46] and Fitts' Law [47] have been meticulously applied to ensure an intuitive and efficient user experience for individuals with ASD. Hick's Law suggests that the more choices a user has, the longer it takes to make a decision [46]. By streamlining the layout of the booklet, particularly in the presentation of pattern markers and QR codes, the aim is to minimise cognitive load. Each social story features a simple, straightforward choice: scan the QR code to access the AR content or use the pattern markers to trigger animations. This design choice reduces decision-making time and simplifies the user's interaction with the booklet, making the experience more straightforward and less overwhelming.

Fitts' Law is a principle that predicts the time it takes to quickly move to a target area, such as a button or QR code [47]. This principle was used to determine the placement of elements within a booklet. To optimise the user interface for easy and rapid access, the QR code was positioned beside or below the story's title, and pattern markers were made prominent and scannable. This consideration is essential for users with motor difficulties, as it minimises the physical effort required to interact with the AR features.

Adhering to these psychological principles during the booklet's design phase not only enhances the educational resource's usability and accessibility but also ensures that the incorporation of AR technology into social stories remains a beneficial augmentation rather than a potential source of confusion or frustration. The careful layout of text, images, QR codes, and pattern markers aims to provide a seamless and engaging learning experience. This design approach is illustrated in the mockup figure (see Figure 3.4), which visually reinforces the connection between visual aids and the accompanying narrative while introducing an interactive dimension that leverages modern technology to enhance storytelling.

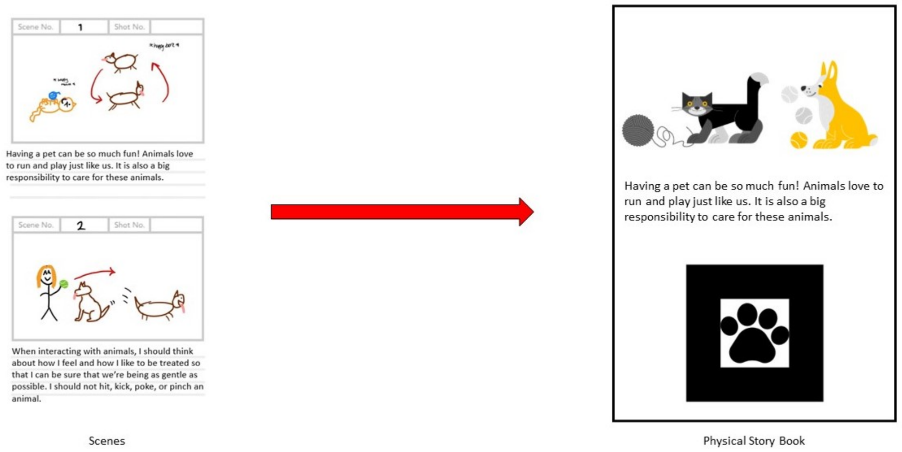



Figure 3.4: Physical Story Book Mock-up Design

## 3.6 Marker

Markers can be thought of as a basic version of QR codes. When a user's device camera detects a marker, the web application displays a 3D model on top of it [48]. Markers can be of three different types [49]:

- **Hiro:** the default marker.
- **Barcode:** auto-generated from matrix computations.
- **Pattern:** custom makers created from a straightforward, high-contrast image.

To make the AR application more accessible and engaging, especially for individuals with ASD, custom pattern markers with icons related directly to the story were used. This adds a layer of interpretability and fun to the experience. Unlike generic QR or bar codes, which often appear as arbitrary patterns to users, these story-specific markers carry meaningful visuals that are easily recognisable and directly linked to the narrative's content. This design choice not only enhances the user's ability to connect with and understand the purpose of the marker but also enriches the storytelling by integrating visually interpretative elements.

### 3.6.1 Marker Design

To improve the user experience, with a particular focus on accessibility, the markers have been designed to include icons that visually represent key elements of the story. This approach is different from traditional QR or bar codes that do not provide any visual clues, and it adds a layer of immediate recognition and engagement for the user. This choice has a significant impact on accessibility and engagement by:

- **Interpretability:** Incorporating icons that reflect story elements can make the markers an integral part of the storytelling process. This enables users, especially those with ASD, to anticipate and connect with the story's content before the AR elements are activated.
- **Increased Engagement:** Visually thematic markers enhance engagement and transform scanning into integrated story exploration.
- **Accessibility from a Design Perspective:** Designing markers to be easily understandable to people with ASD is crucial for creating a more inclusive and user-friendly environment. This design choice showcases the overall dedication to accessibility by ensuring that every element, including markers, contributes to a seamless and accessible experience.

### 3.6.2 Implementing Markers

To design an effective custom pattern marker, it's crucial to adhere to several essential guidelines to ensure the creation of a marker that is both functional and suitable for the intended application:

- The maximum resolution of a marker is 16x16 pixels [50].
- They must be square [50].
- They cannot have white/transparent areas, only light grey [50].
- They cannot contain colours, only black and grey [50].
- They must contain simple text, such as one letter, a number, or a symbol [48].

- They must have a light area surrounding the marker of half the marker border width [50].
- The marker image must be rotationally asymmetrical [50].

Creating the pattern markers involves using the AR.js Marker Training [51] online tool, which produces a scannable image and a corresponding .patt file. The .patt file is crucial as it enables the AR.js web application to recognise and correctly interpret the marker when the application is running. This tool ensures uniformity in the creation process, creating each marker with consistent quality and specifications.

The next step in the marker creation process was to source appropriate icons representing different parts of the story. For this purpose, Flaticon [52] was chosen. Relevant icons that best aligned with specific segments of the story were chosen under a premium license, which permits usage for personal projects. To enhance the recognition capabilities of the AR tool, it was necessary to consider the requirement that the marker background should not be white. To achieve this, each icon was imported into Photoshop and placed against a grey backdrop aligned with the AR.js Marker Training tool's specifications for optimal recognition.

After setting the icons against the grey background, the AR.js Marker Training tool created the marker as shown in Figure 3.5. This powerful tool facilitated the transformation of the visually adapted icons into scannable markers, each accompanied by a corresponding .patt file for each marker, which is essential for the AR application to recognise the markers.

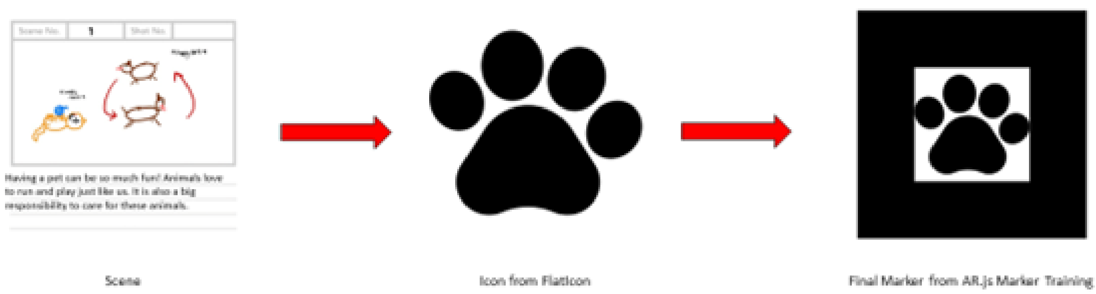



Figure 3.5: Marker Design Process

After generating the markers, the marker files were re-imported into Photoshop to refine and enhance their effectiveness. A black border was added around each marker for aesthetic purposes and to improve their contrast and distinctiveness, as shown in Figure 3.6. This step aimed to enhance their detectability and functionality significantly within the AR application.

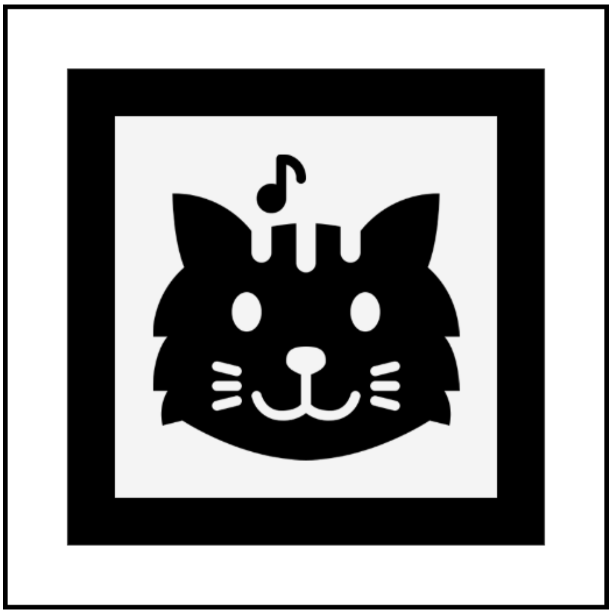


Figure 3.6: Example of Final Marker Design

## 3.7 Prototyping

### 3.7.1 Paper Prototype

A paper prototype is a low-fidelity prototype that is a basic representation of an initial design concept used for quick idea testing, recognising potential issues, and ruling out designs that don't work [53]. To develop this paper prototype, storyboards were created for each story, as shown in Figure 3.7 and Figure 3.8, illustrating the animations that closely mirror what is intended for the final user experience.

The storyboard creation involved a crucial decision about the animation style: choosing between realistic models and stylised characters. This choice was significant, as it would influence how users engage and relate to the story. While realistic models directly connect to social scenarios, character-based animations offer creative appeal. Ultimately, models resembling real people were selected to help users better see themselves in the stories, enhancing their connection and understanding of social behaviours and cues.

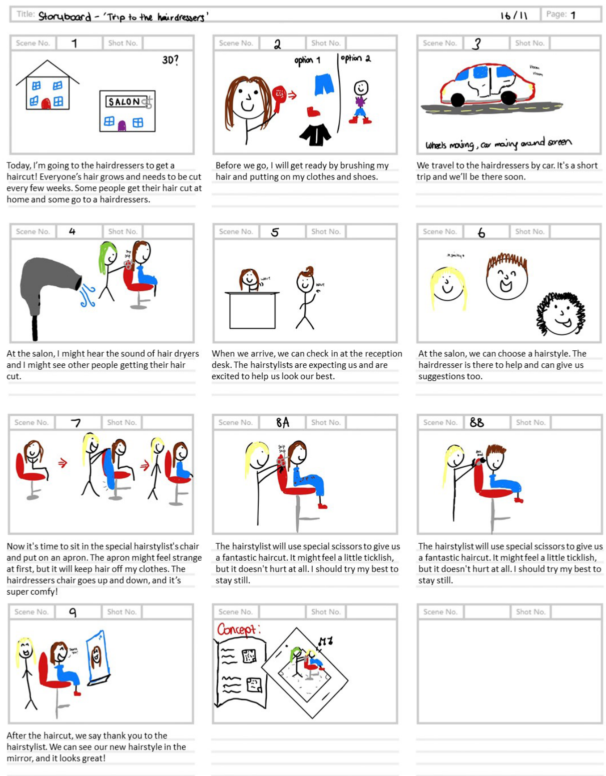



Figure 3.7: “Trip to the Hairdressers” Storyboard

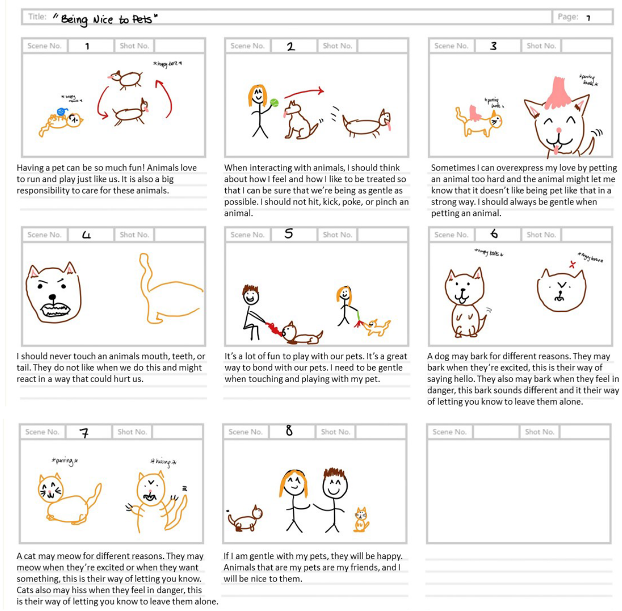



Figure 3.8: “Being Nice to Pets” Storyboard

### 3.7.2 Digital Prototype

Digital prototypes are high-fidelity prototypes of a completed app. Digital prototypes allow for refining a design's effectiveness and conducting thorough user testing for optimal performance [54]. For the digital prototyping phase, high-fidelity wireframes were created to showcase the user experience of the AR.js application, as illustrated in Figure 3.9.

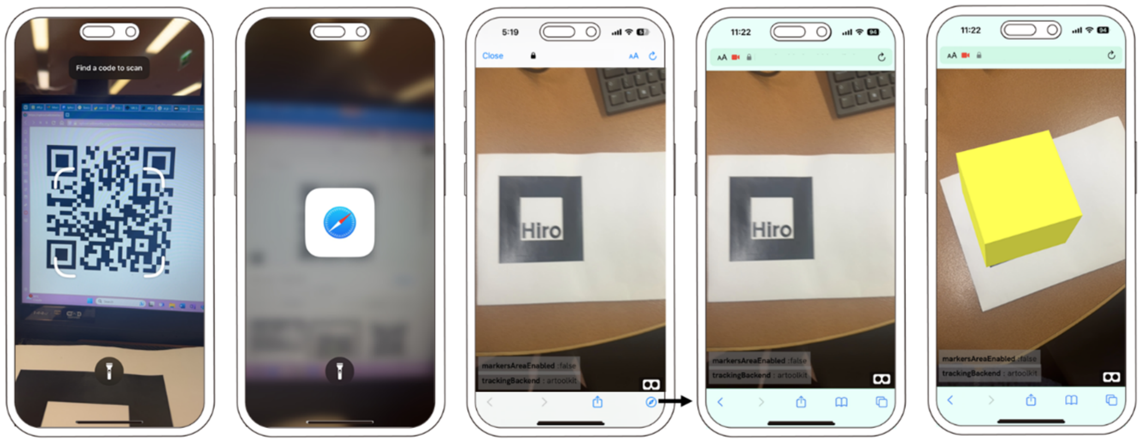



Figure 3.9: Digital Prototype

These wireframes (Figure 3.9) were constructed by creating a simple AR.js app using Glitch [21] that shows a cube when the marker, in this case, a Hiro marker, is scanned. The prototyping process involved capturing this interaction step-by-step through smartphone screenshots, which were then incorporated into the wireframes.

These wireframes (Figure 3.9) effectively map out the user's journey, from scanning the story's QR code, which leads them to a designated web page, to the point where the app recognises a marker and displays the related animation—in this case, a cube. Plans for future improvements include adding a mute button for sound and refining the interface to enhance the overall user experience in the final app.

## 3.8 Model Integration

### 3.8.1 Source and Selection

For sourcing and selecting 3D models and animations for the AR application, assets are sourced from renowned platforms such as Sketchfab [55], Turbosquid [56], CGTrader [57], and Mixamo [58]. These platforms are chosen for their comprehensive range of free and premium assets suitable for personal and educational purposes and their support for multiple file types, ensuring compatibility with different development tools.

When selecting models and animations, it's important to consider:

- **Relevance to the story:** Ensure that the selected assets align with the theme of the corresponding story.
- **Quality and optimisation:** Look for high-quality assets optimised for real-time rendering to ensure they perform well on various devices.
- **Licensing:** Check each platform's licensing policies to verify that the assets are available for their intended use.
- **File compatibility:** Choose assets compatible with the development environment and easily integrated and modified.

### 3.8.2 Optimisation

Integrating and optimising 3D models and animations involves several essential practices and considerations to ensure AR content runs smoothly and efficiently on the web. Here's a comprehensive approach based on various resources:

- **Exporting Models:** Exporting models in glTF or GLB format is recommended for web-based AR applications. These formats are designed to efficiently transmit and load 3D models and animations in web environments, making them ideal for AR.js projects [59].
- **Optimize 3D Models for Mobile Devices:** Given the constraints of mobile devices, such as limited storage, processing power, memory, and battery life, it's crucial to optimise 3D models heavily. This involves reducing file size, part count, and polygon count significantly. A general guideline suggests limiting your model's complexity to 100,000 triangles for the entire scene and adhering to specific vertices, UVs, shaders, and texture resolution counts [60].

# Chapter 4

# Implementation

This section of the report provides detailed information on the practical aspects involved in the development and creation process of the AR application. The development process was iterative, with each step building upon the previous one to enhance the application's usability and effectiveness. The implementation process required flexibility, adaptability, and exploration of various technical avenues to meet the project's objectives.

## 4.1 Creating the AR World

Glitch, a cloud-based platform known for its real-time collaborative coding capabilities, was the development environment selected for creating the AR world. Its support for HTTPS (HyperText Transfer Protocol Secure) is essential for AR applications as it encrypts data between the user's browser and the server. This ensures the security of user data and compliance with web standards for modern browsers [61]. This security feature is especially crucial in AR applications, which often require access to camera feeds and personal data. Furthermore, its live code access feature facilitated testing on mobile devices, providing instant feedback and accommodating an iterative development process necessary for refining the AR experience.

### 4.1.1 Initial Setup

The initial stage in developing the AR application involved creating a basic HTML document named `index.html`.This document provided the foundation for the application, enabling the gradual addition of AR functionalities. The development process was strategically broken down into manageable steps to prevent the application from becoming overly complex. After creating the HTML document with standard HTML elements, the AR.js library was imported to facilitate the creation of the AR environment.

The decision to use AR.js was motivated by its strong support for web-based augmented reality. AR.js stood out for its compatibility with a wide range of devices and ability to deliver high-performance AR directly in web browsers without additional software. Importing AR.js into the project was straightforward by including a script tag in the <head> element linked to the AR.js library and A-Frame (see Listing 4.5).

```
<head>
    <meta charset="utf-8" />
    <title>Being Nice to Pets</title>
    <link rel="stylesheet" href="/style.css" />
    <script src="https://aframe.io/releases/1.3.0/aframe.min.js">
    </script>
    <script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe-ar.js"></script>
</head>
```

Listing 4.1: Importing AR.js Libraries

With AR.js successfully imported, the next step involved creating the AR scene by integrating an <a-scene> element into the document's body, which served as a container for all AR content. A Hiro marker was initially utilised for its simplicity and effectiveness to anchor digital content to the physical world. It was added to the scene with an <a-marker> tag. A simple 3D object—a box—was introduced into the scene using an <a-box> element, demonstrating the seamless integration of 3D objects and marking the first instance of tangible AR content creation. Finally, the setup was completed by activating the AR camera with an <a-entity> tag, enabling viewing the scene from the device's perspective and establishing a basic but functional AR scene (see Listing 4.2).

```
<a-scene embedded arjs>
    <a-marker preset="hiro">
        <a-box position="0,1,1" material="opacity: 0.5"></a-box>
    </a-marker>
    <a-entity camera></a-entity>
</a-scene>
```

Listing 4.2: Creating the AR Scene

After viewing the scene on a device, the box anchored by the Hiro marker confirmed the successful integration of AR elements (see Figure 4.1), creating an interactive AR world.

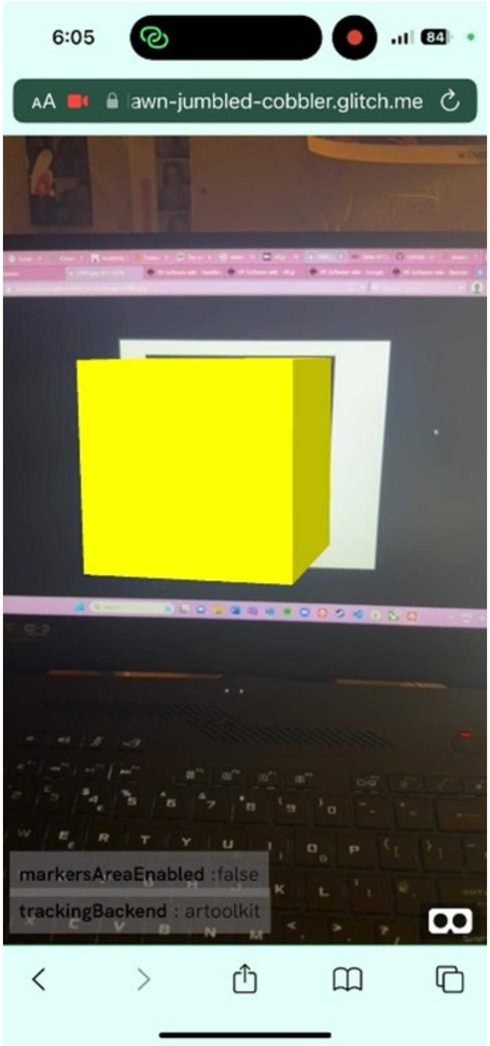



Figure 4.1: First AR.js Attempt

### 4.1.2 Iterative Enhancements

The initial stages of building the AR world focused on embedding key functionalities to enable a basic operational framework. To make this digital world more visually appealing, a low-poly animated glTF model was added. This design approach focuses on achieving a good balance between visual quality and performance efficiency. Low-poly models, characterised by their minimal polygon count, are pivotal in AR applications for their ability to deliver smooth performance across diverse devices, ensuring the AR experience remains accessible and engaging for all users. After increasing the scene's complexity without impacting device performance, limitations were encountered with A-Frame's default animation properties when managing the intricate animations of glTF models.

The pursuit of animating these 3D glTF models revealed a critical compatibility issue within A-Frame's animation system, failing to execute the embedded animations accurately. This challenge necessitated an exploration of alternative solutions to enable precise animation playback. As part of the iterative enhancements to the AR world's development, the `patternRatio` attribute was added within the `<a-scene>` tag to refine marker detection sensitivity (see Listing 4.3). This fine-tuning aimed to improve the reliability and responsiveness of the AR content's interaction with physical markers, address potential detection inconsistencies and ensure a more seamless user experience.

```
<a-scene embedded arjs="patternRatio: 0.57" >
```

Listing 4.3: Pattern Ratio Enhancement

This adjustment addressed the animation challenge using Don McCurdy's animation mixer component. The component is an extension within the A-Frame community designed specifically for managing built-in animations of 3D models in the glTF format. This choice, advocated by the A-Frame documentation for its compatibility with glTF models, offered extensive animation controls, allowing for responsive and precise animation playback [62]. Integrating this component into the project was straightforward, simply adding a script tag linking to the A-Frame extras library in the HTML document's `<head>` element (see Listing 4.4).

```
<script src="https://cdn.jsdelivr.net/gh/c-frame/aframe-extras@7.2.0/
    dist/aframe-extras.min.js"></script> >
```

Listing 4.4: Importing A-Frame Extras

Once integrated, the animation-mixer component was applied directly within the `<a-entity>` tags associated with each model, enhancing the level of control over animations (see Listing 4.5). In this instance, enabling looped animations facilitated a continuous animation of the model within the AR scene (see Figure 4.2).

```
</head>
  <body style="margin: 0px; overflow: hidden">
    <a-scene embedded arjs>
        <a-marker preset=   hiro   >
            <a-entity gltf-model="#cat" scale="1 1 1" position="0 0 0"
                rotation="0 0 0" animation-mixer="loop: repeat">
            </a-entity>
        </a-marker>
        <a-entity camera></a-entity>
    </a-scene>
  </body>
```

Listing 4.5: Importing AR.js Libraries

Figure 4.2: Low-Poly Animated Model

After successfully implementing the animation mixer component, further testing was carried out with a more detailed model to evaluate the environment's ability to support high visual fidelity while maintaining optimal performance. Despite importing the detailed model without prior optimisation, the initial tests showed promising results (see Figure 4.3). However, it was clear that proper optimisation would be crucial to ensure a seamless and enjoyable AR experience for users, as the final application would consist of multiple detailed models.

Figure 4.3: High-Quality Animated Model

## 4.2 Modelling and Animating

The progression from static models to animated figures in AR showcases the intricate digital creation and optimisation process. This section will explore the detailed steps involved in preparing, animating, exporting, and importing 3D models to ensure their smooth integration into the AR environment.

### 4.2.1 Preparing the Models

The correct preparation of the 3D models is crucial for ensuring optimal performance and appearance within the AR environment. After importing the models into Blender, various steps were taken to reduce the final file size of the exported model.

The first step involved reducing the texture files associated with the model to reduce the overall file size of a model. This included both the normal and diffused images. The cat textures, initially 4096px x 4096px and took up 11.1 MB of space, were reduced to a more manageable 1024px x 1024px. This decreased the file size to only 212 KB, significantly enhancing the web app's loading time. Although this reduction in texture quality might seem concerning, it doesn't create a noticeable difference. After reducing the texture file sizes, the next step was to apply the textures to the model.

To enhance the realism of models within the AR environment, textures were applied using Blender's Principled BSDF shader, selected for its extensive support for physically based rendering (PBR) principles. Unlike simpler shaders, such as the Diffuse BSDF, the Principled BSDF shader offers a versatile framework that accurately simulates a broad spectrum of material properties, crucial for creating realistic textures that comply with the glTF format's requirements for precise light interaction and material definition [63].

Node Wrangler, an addon in Blender, significantly simplified the material adjustment and texturing process. It automated the setup of Principled BSDF nodes, allowing for rapid and straightforward mapping of textures to the model by using the "Add Principled Setup" option from the Node Wrangler menu [64] (see Figure 4.4).

Initially, texturing models was straightforward, as most were imported without textures or had Principled BSDF shaders automatically selected. However, an issue arose with an imported model preset with the Diffuse BSDF shader. When exported to the glTF format and integrated into the AR scene, the model failed to display its intended textures,

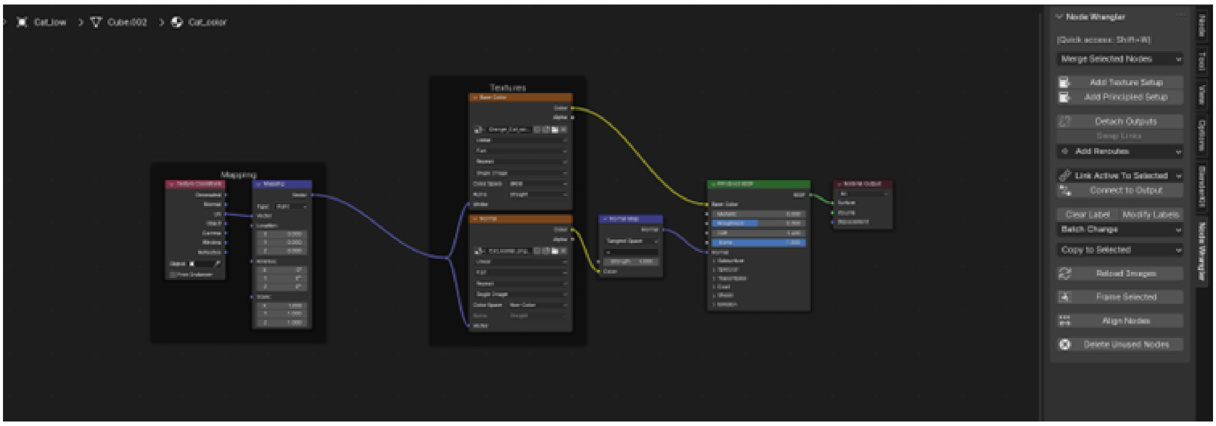


Figure 4.4: Texturing using Nodes in Blender

rendering instead in a solid colour (see Figure 4.5). This issue led to a review of the model's material settings in Blender, which revealed that the base colour was mistakenly set to use a diffuse BSDF shader.

After consulting the Blender documentation, discrepancies were discovered in how these shaders depict material properties. The Diffuse BSDF shader mainly deals with simple colouration and lacks the advanced PBR attributes necessary for realistic rendering in the glTF format. As a result, when using the Diffuse BSDF shader, the model's textures were not exported correctly, resulting in the model appearing as a solid colour (see Figure 4.5).

Switching the shader to Principled BSDF and re-exporting the model resolved the issue of the solid colour appearance. However, the model then appeared overly dark. Further investigation revealed that the material's metallic setting was mistakenly set at 0.8, implying a highly metallic surface inappropriate for the model's actual materials. The Principled BSDF shader, adhering to PBR principles, uses the metallic value to determine a material's reflectivity, where a high value suggests a fully metallic surface—adjusting this value to 0 to denote a non-metallic material corrected the issue, rendering the model with the correct textures and appearance.

This experience emphasises the importance of precise shader settings and material properties in PBR-based rendering to achieve the desired visual outcomes in AR environments. After optimising the textural and material aspects of the models, the final step in model preparation involved geometric optimisation.

A model's complexity level can significantly impact the performance and visual quality


Figure 4.5: Exported Model with Metallic Value.

of augmented reality experiences. While highly detailed models may look visually appealing, they can strain the rendering process, leading to a slower and less responsive AR environment. To overcome this, decimation techniques were used to reduce the complexity of the models.

Decimation is a process that reduces the vertex and face count of a model without compromising its visual appearance [65]. To achieve this, the decimation modifier in Blender was used to reduce the number of faces in a 3D model. The collapse option is then applied to collapse the edges, reducing the number of faces by half. For instance, the original cat model with 1056 faces was reduced to 712 faces by applying a decimation modifier with a ratio of 0.5 (see Figure 4.6b). However, the incorrect application of decimation can lead to model distortion, where the geometry becomes visibly altered or deformed. Therefore, careful adjustment and testing are essential to determine the optimal decimation ratio for a model that balances detail retention with performance improvement.

(a) Undecimated Cat with Original Textures

(b) Decimated Cat with Reduced Textures

Figure 4.6: Comparison of Cat Model

With the models optimised through careful texture reduction, shader adjustment, and geometric simplification, they were ready for the next phase of development - animation.

### 4.2.2 Animating the Models

The animation process was a crucial element in enhancing the immersive experience of the AR environment, involving a combination of pre-animated models and manual animation efforts to bring each character to life. Initially, the plan was to create two stories - “Being Nice to Pets” and “A Trip to the Hairdressers.” However, during the model selection phase of development, it became apparent that creating “A Trip to the Hairdressers” as envisioned initially was difficult due to the unavailability of suitable pre-animated models and animations.

Initially, the plan was to source rigged models and motion capture (mocap) files or external animations to animate the models effectively. However, finding the appropriate animations that met each story’s specific requirements proved challenging. After being unable to locate suitable animations, manually animating the models was considered but quickly deemed infeasible due to the significant time investment and steep learning curve required.

Confronted with these challenges, the strategy shifted towards utilising primarily pre-animated models and shifting focus to the “Being Nice to Pets” story. Fortunately, various pre-rigged and pre-animated animal models were readily available online, offering a practical solution. These pre-rigged and pre-animated models saved considerable time and resources, allowing for the direct integration of complex animations without extensive animation expertise.

However, not all models came pre-animated, particularly the human characters. To solve this problem, Mixamo, an Adobe service offering various 3D character animation options, became a vital resource. Mixamo's user-friendly platform provides auto-rigging features, where models uploaded by users are automatically rigged with skeletal bones, making them ready for animation. This was very convenient as it eliminated manual rigging, which required significant technical skill and time (see Figure 4.7).

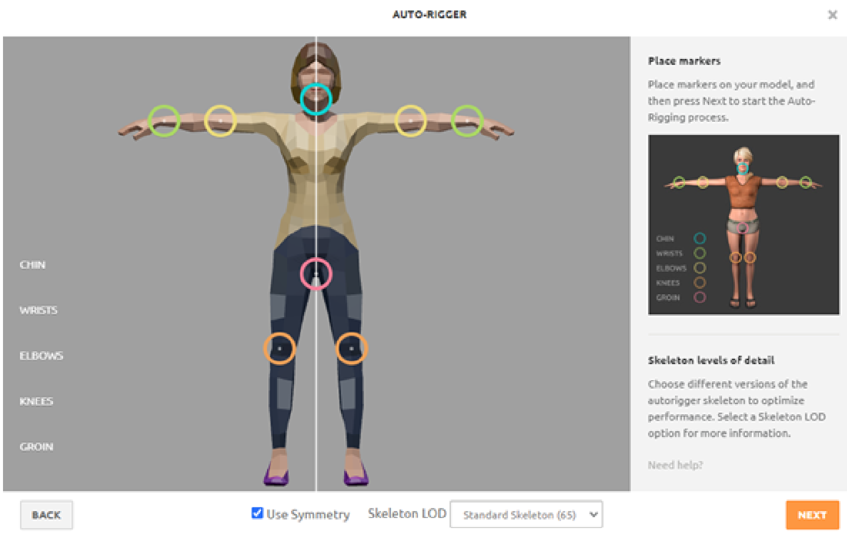


Figure 4.7: Mixamo Auto-rigging

After uploading the models to Mixamo, various animations could be directly applied to the characters. This included everything from simple gestures to complex sequences of movement, all customisable to fit the narrative needs of the AR experience. Mixamo's extensive library of animations and the ability to fine-tune these animations to match the story's context significantly streamlined the animation process (see Figure 4.8).

While working with pre-existing models, selecting the appropriate animations was crucial for effectively conveying the story within the AR environment. Each animation was carefully selected to ensure it aligned with the narrative and added visual appeal to the immersive experience. The available pre-animated models were reviewed, and animations that matched the actions and emotions depicted in the reworked storyboards were chosen. Each model was animated separately before being added to the main scene,

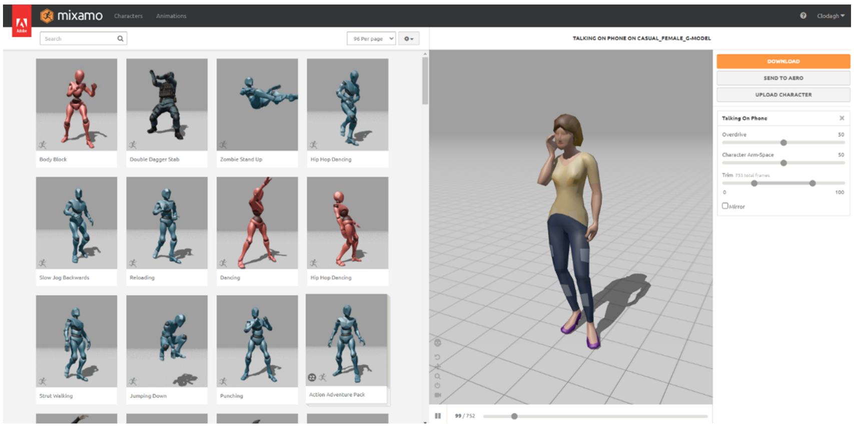



Figure 4.8: Mixamo Library of Animations

allowing for better control over each model's animations. This ensured they could be edited and optimised individually without affecting the overall scene.

Blender's Dope Sheet and Action Editor were indispensable tools in this process, providing a comprehensive overview and granular control over the animations. The Dope Sheet provided a comprehensive snapshot of all animations linked to a model. At the same time, the Action Editor allowed for detailed adjustments, enabling animations to be either "pushed down" into the Nonlinear Animation (NLA) tracks or "stashed" for future use (see Figure 4.9).

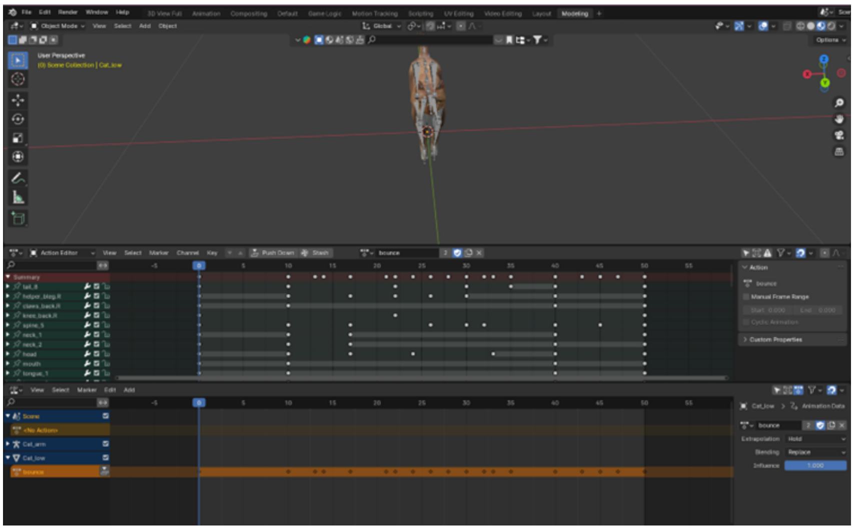


Figure 4.9: Action Editor and NLA in Blender

Pushing down an animation solidified it into the model's animation sequence for immediate use in the AR scene. In contrast, stashing kept the animation separate, preserving flexibility for adjustments or repurposing (see Figure 4.10).

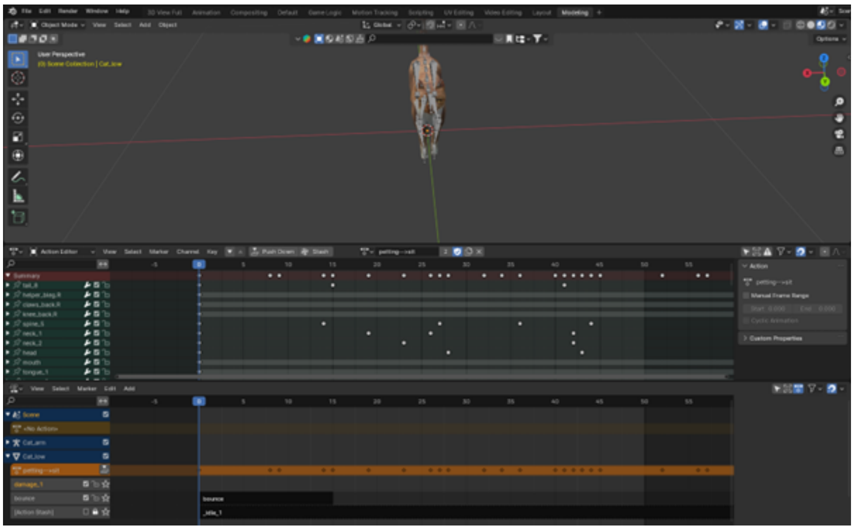


Figure 4.10: Using the Action Editor to Push Down and Stash Animations

The decision to push down or stash animations depended on how they would be used in the AR environment. This process required technical precision and creative vision to ensure the models were animated effectively. Additionally, the NLA Editor made it easy to transition seamlessly between animations and blend them automatically, resulting in smooth and dynamic movements that enriched the visual narrative.

After animating each model, the next phase involved preparing them to be integrated into a web-based AR scene. With the animations now refined and correctly applied to each model, the focus shifted to exporting these models in the glTF format (see Figure 4.11).

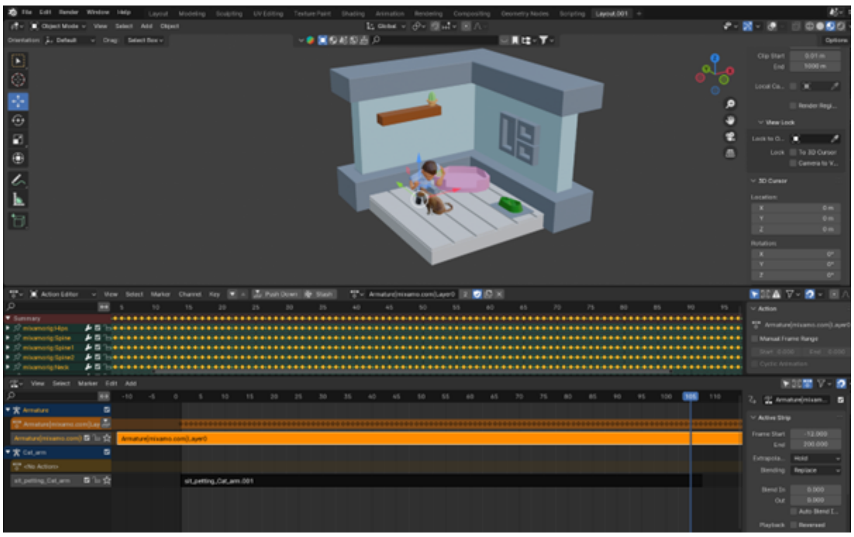


Figure 4.11: Models added to Scene and Prepared for Export.

This next step was crucial for optimising the models for web deployment, preserving the integrity of the animations, and ensuring compatibility with the AR framework.

### 4.2.3 Exporting the Models

Exporting models with animations from Blender for AR environments presented initial challenges, notably when animations failed to perform as expected in the final scene. At first, all animations played simultaneously, or only the first animation in the sequence would work. This led to a thorough investigation of the export settings and several attempts at animation baking. Eventually, it was found that the problem was with the export settings, which were set to "Actions." This highlighted a critical mismatch between the export configuration and the AR viewer's capabilities.

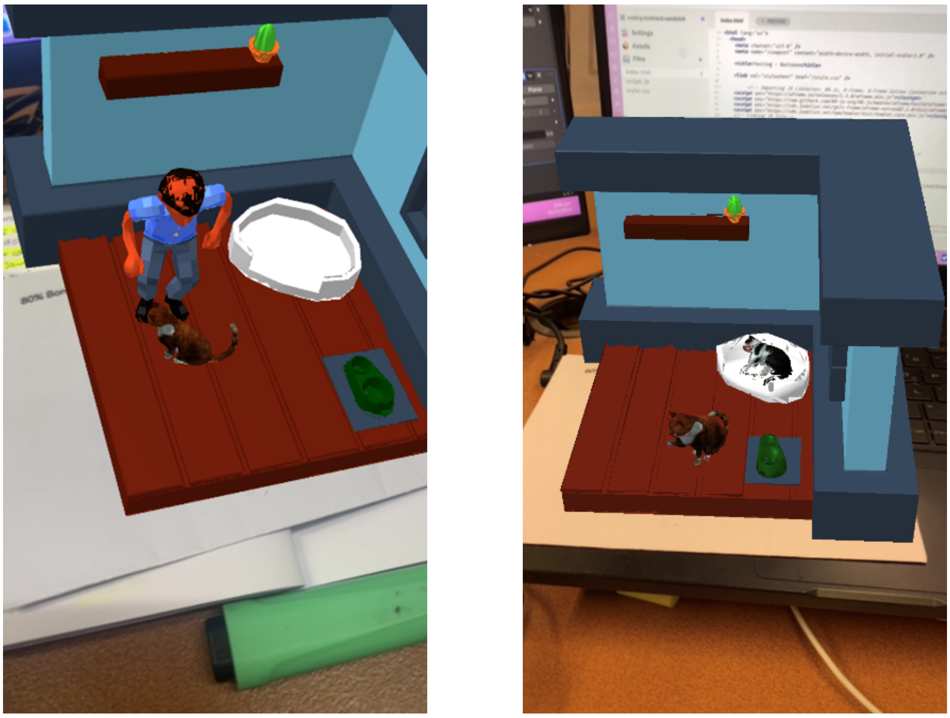
In addition to the animation problems in the initial exports, I faced an issue with z-fighting. This caused flickering in the models and made parts of the boy model, like his head, look patchy even though they were hidden by objects such as hair (see Figure 4.12). Z-fighting happens when the rendering system cannot determine which of two overlapping surfaces is closer to the viewer, resulting in visual glitches [66]. This issue was particularly problematic because it detracted from the visual quality and immersion of the AR experience.

To overcome the z-fighting problem, a logarithmic depth buffer was added to the `<a-scene>` attribute (see Listing 4.6). This technique improves depth accuracy, especially for faraway objects, by scaling depth values logarithmically [67]. It significantly reduced the occurrence of z-fighting, resulting in better stability and visual consistency of the models in the AR scene.

```
<a-scene embedded arjs="patternRatio: 0.56" renderer="
    logarithmicDepthBuffer: true">
```

Listing 4.6: Logarithmic Buffer

After addressing the z-fighting issue, the focus shifted to resolving the animation problems. Various export settings were experimented with to develop a customised approach that would cater to the distinct requirements of each model's interaction with the scene. For models with animations that didn't require movement within the scene, the "Scene" export setting was effective. This export method ensured that the animations were exported as they appeared in Blender's viewport, preserving their intended sequence and execution. However, for models that required in-scene movement, animations were



(a) Z-Fighting with Boy Model's Hair

(b) Z-Fighting with Cat Bed and Floor

Figure 4.12: Z-Fighting Issues

"stashed", allowing them to be exported as separate actions. This adjustment provided the necessary flexibility for the models to interact dynamically within the final AR scene.

The final export settings, as illustrated in Figure 4.13, were chosen with great care to optimise performance and preserve the visual quality of the models in the AR environment.

Limiting exports to "Visible Objects" included only essential elements, which helped optimise the file size and improve performance. The "+Y Up" transformation setting aligned the models with the AR environment's coordinate system, ensuring consistent orientation. Mesh settings such as "Apply Modifiers / UVs / Normals / Vertex Colors" preserved the models' detailed characteristics, ensuring accurate visual appearance. The "Animation mode: Scene" setting was critical for straightforward animations, maintaining expected behaviour in the final scene. Additionally, specifying "Rest and Ranges" to use the current frame as object reset transformations and setting all glTF animations to start at 0 aided in achieving animation synchronisation [68].

The decision to use these specific export settings was informed by the Blender documentation, which highlights the flexibility and control offered by different export options.

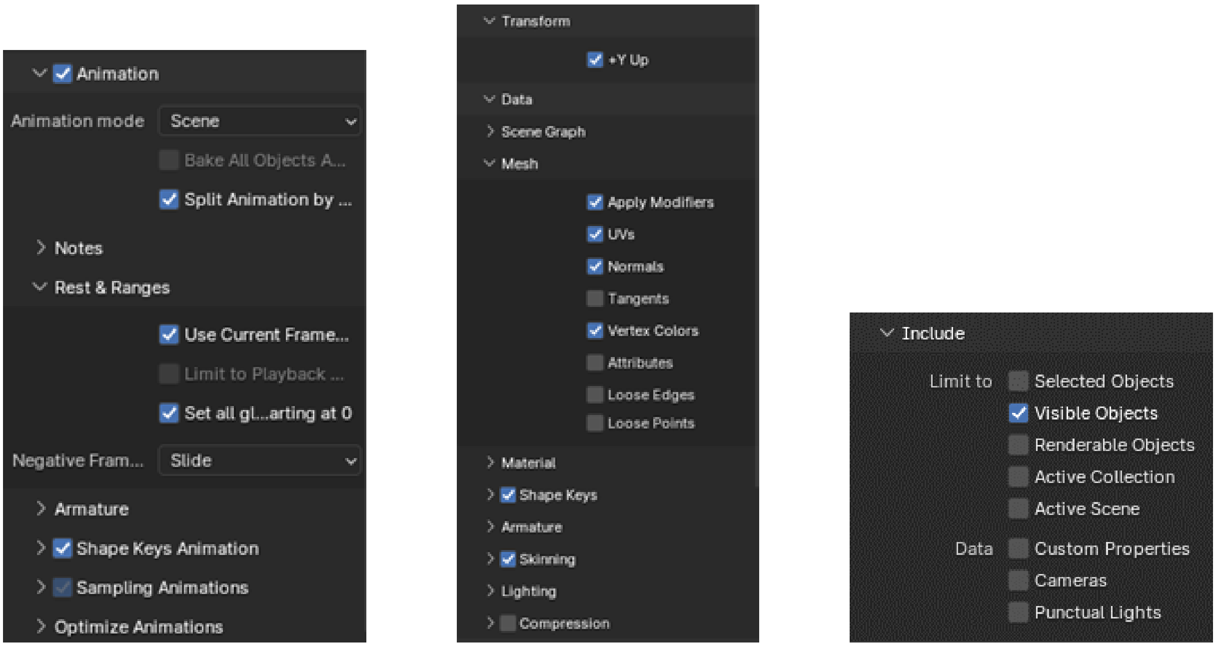



Figure 4.13: Blender Export Settings

For example, the “Actions” mode is ideal for game engines or scenarios where a character has an extensive animation library, requiring each action to be on its own NLA track. Alternatively, the “Scene” option exports animations precisely as they appear in the viewport, which ensures that the AR scene reflects the original Blender setup.

These carefully selected export settings, informed by Blender’s comprehensive documentation, addressed the unique demands of each stories AR environment. By understanding the implications of each setting, the models were optimally prepared for integration, ensuring animations played as intended and contributed to a rich, interactive AR experience. This thoughtful approach to animation and export solved technical challenges and enhanced the AR environment’s overall quality, making the virtual world more engaging and lifelike.

After texturing and animating the 3D models, the next step was integrating them into the AR scene. This step was crucial in creating the AR experience. It involved a meticulous approach to managing and embedding the models within the AR application hosted on Glitch. The use of Glitch not only facilitated hosting the models but enabled dynamic testing of the AR application in real-world scenarios.

After importing the models to Glitch, they were systematically organised within the application’s code using the <a-assets> tag, a standard A-Frame practice for managing external assets. This method ensures that all models are preloaded before the application runs, thereby reducing load times and improving user experience. Each model was assigned a unique identifier through the `<a-asset-item>` tag, with its source file location specified, making it easy to reference within the AR scene (see Listing 4.7).

```
<a-assets>
    <a-asset-item
          id="cat"
          src="https://cdn.glitch.global/link2model"
    ></a-asset-item>
</a-assets>
```

Listing 4.7: Asset Organisation

Transitioning from the initial Hiro marker, the application evolved to incorporate custom markers, enhancing the personalised aspect of the AR experience. Each custom marker was created using an online AR marker creation tool, generating a PNG image and a corresponding PATT file. These files were then uploaded to Glitch, with the PATT files associated with their respective markers in the application's code via the `<a-marker>` tag. This tag was configured to recognise the custom markers, specifying the "custom" preset and "pattern" type and pointing to the URL of the uploaded PATT file (see Listing 4.8).

The models were then integrated into the marker tag using the `<a-entity>` tag. Each model was identified by a unique identifier, and attributes such as scale, position, and rotation were finely adjusted to ensure the models appeared correctly within the AR environment (see Listing 4.8). The animation-mixer attribute played a crucial role in controlling the looping of the animations and bringing static models to life within the context of their respective markers.

```
<a-marker
    preset="custom"
    type="pattern"
    url=https://cdn.glitch.global/file.patt
>
    <a-entity
        gltf-model="#cat"
        scale="1.5 1.5 1.5"
        position="0 0 0"
        rotation="0 0 0"
        animation-mixer="loop: repeat"
    >
```

```
    </a-entity>
</a-marker>
```

Listing 4.8: Asset Organisation

After carefully assigning models to markers and configuring their animations, extensive testing was conducted to ensure that the correct models were triggered by the appropriate markers, resulting in a seamless and captivating AR experience. This successful testing phase marked a significant milestone in the development process, highlighting the successful integration of models into the AR scene.

## 4.3 Audio Integration

Integrating audio in AR is crucial to creating a fully immersive and interactive experience. Apart from the visual aspect, sound is essential in immersing the user in the digital world. It provides cues that guide interaction, evoke emotions, and enhance the application's storytelling. This section explores the tools, techniques, and considerations required to ensure a harmonious blend of sight and sound in the AR experience.

### 4.3.1 Sound Asset Sourcing and Preparation

To create a more immersive experience within the AR environment, the sound assets were selected based on the thematic elements of each story scene. The auditory experience is crucial in enhancing the visual experience, creating a multisensory engagement that complements the unfolding narrative. This endeavour led to exploring various cat and dog audio clips to bring the AR scenes to life and provide users with an even more immersive experience.

The search for high-quality, relevant sound assets brought me to Freesound.org, a platform known for its extensive collaborative database of audio snippets, samples, and recordings [69]. Freesound's mission to offer a wide array of sounds, all released under Creative Commons licenses, made it an ideal source. This allowed for the legal and ethical use of these sounds in the AR application, ensuring that the project remained compliant with copyright laws while benefiting from the platform's rich assortment of audio assets.

Initially, the downloads consisted mainly of .wav files. These files were of high quality but were also large in size and length. This posed a challenge, as the aim was to maintain a responsive and efficient AR application while not compromising sound quality. This challenge was overcome by using an open-source web-based audio editing tool called AudioMass [70]. AudioMass enabled the importing of the original audio files, trimming them to remove unnecessary parts and converting them into a more compact .mp3 format. This process was crucial in reducing the audio clips' length and file size, ensuring they were optimised for web deployment.

With the sound assets meticulously selected and optimised, the subsequent phase focused on integrating these audio elements into the AR scenes. Achieving a perfect synchronisation of visual and audio cues is crucial to creating an immersive AR experience for the user.

### 4.3.2 A-Frame Sound Component

Integrating sound into the AR environment with A-Frame initially seemed straightforward. The framework can attach audio to the scene through the `<a-entity>` tag. However, a significant platform-specific limitation emerged during the testing phase, particularly affecting iOS users.

According to the A-Frame documentation, a notable restriction for iOS devices requires physical user interaction for sound playback [71]. This constraint, stemming from iOS browser policies designed to prevent unsolicited audio playback, poses a unique challenge. It effectively prevents the auto-playing of sound, a feature critical to the seamless integration of auditory cues in the AR experience. The documentation further clarifies that A-Frame's internal events, such as those triggered by fusing cursors, do not meet the criteria set by iOS for initiating sound playback. This discovery highlighted a potential obstacle to achieving a truly universal AR experience, particularly in light of the project's commitment to accessibility across all devices.

The need for user interaction on iOS devices to trigger sound can significantly impact an application's usability and accessibility. Relying solely on A-Frame's sound component may inadvertently exclude a portion of the intended user base, which goes against the project's goal of inclusivity. Despite A-Frame's robust features, its sound component does not fully address the complexities of cross-platform audio integration, particularly iOS's stringent playback policies.

The limitations of A-Frame's built-in capabilities have prompted an exploration of alternative audio libraries that can bypass iOS's playback restrictions. The goal is to find a solution that enables effective, sound implementation across all platforms, ensuring that the AR application remains engaging and accessible to all users, regardless of their device.

### 4.3.3 Choosing an Alternative Audio Library

Difficulties when integrating sound directly through A-Frame, especially on iOS devices, highlighted the need for alternative solutions. This was essential for providing a more versatile and widely compatible audio experience across different devices. The A-Frame documentation suggested Howler.js, a robust audio library designed for the modern web.

Howler.js is a popular web audio library that offers advanced features for developers who want to add sound to their applications. Its documentation provides a detailed overview of its capabilities, including support for spatial audio, which is particularly useful for creating immersive AR environments [72].

One of the defining features of Howler.js is its reliance on the Web Audio API, with a fallback to HTML5 Audio, ensuring broad compatibility and reliability across all platforms. This dual approach addresses the critical challenge of audio playback restrictions on mobile browsers and platforms like Chrome and Safari. Howler.js implements a clever workaround to unlock audio on these platforms by playing an empty buffer at the first user interaction, effectively bypassing the limitations that hindered using A-Frame's sound component on iOS devices. This functionality underscores Howler.js's advantage in providing a seamless audio experience without requiring explicit user actions to enable sound playback.

Integrating Howler.js into the AR application was straightforward, thanks to its ease of implementation. Adding a script tag to the document's `<head>`, Howler.js could be easily used within the project, providing immediate access to its vast audio manipulation and playback features. With its extensive features and easy integration, Howler.js was an obvious choice over the native A-Frame sound component. Howler.js resolved the playback issues encountered on specific platforms and opened up new possibilities for creating a more dynamic and engaging auditory experience within the AR environment.

Howler.js's support for spatial audio further enhances its suitability for AR applications. Spatial audio, or 3D sound, plays a vital role in creating immersive experiences, allowing sounds to be perceived as originating from specific points in the space around the user. This feature is essential in achieving a sense of presence and realism in AR, making Howler.js an invaluable tool.

After choosing Howler.js as the audio library, the next step in the development process was to add sound to the AR scenes to enhance the immersive quality of the augmented reality world. Howler.js with its robust support for spatial audio and cross-platform compatibility, Howler.js provides the perfect foundation for embedding complex auditory experiences synchronised with the application's visual elements.

### 4.3.4 Adding Sounds to the App

Integrating sound into the augmented reality environment enhanced the application's immersion. This process was facilitated using JavaScript, with a significant role played by the Howler.js library. This library was chosen for its advanced audio capabilities, which created a dynamic interaction between the user and the AR environment.

The process began with integrating an interactive tap feature to activate animations and sounds simultaneously. This design choice was mainly influenced by the need to meet user interaction requirements for audio playback on specific platforms, such as iOS, thereby adding a layer of interactivity to enrich the AR experience.

The initial implementation focused on a single model, detailed in Appendix A. JavaScript code was embedded at the bottom of the `index.html` file, tasked with managing event listeners and utilising Howler.js for audio playback. This configuration provided a foundation to explore and understand the intricacies of synchronising audio with visual cues within the AR.js framework.

A Howl object is instantiated upon loading the document, verified by the `DOMContentLoaded` event. This object is configured to loop a specified audio file and control its volume to ensure it is audible yet non-intrusive. A variable, `modelVisible`, is established to monitor the model's visibility, an important factor in determining whether to trigger sound playback.

To augment the AR experience's interactivity, event listeners for both touch and click are added to the document's body. These listeners activate the `startExperience` function upon user interaction, which verifies the model's visibility. If visible, the function initiates sound playback if it is not active and synchronises the visual and auditory components by adjusting the `animation-mixer` attribute of the model.

Visibility changes are effectively managed through additional event listeners attached to the AR marker, specifically for `markerFound` and `markerLost` events. These listeners play a crucial role in setting `modelVisible` to true upon marker detection, enabling the immersive audio-visual experience. On the other hand, loss of marker visibility triggers the `stopExperience` function, halting audio playback and pausing animations (see Appendix B).

This structured implementation creates a dynamic interaction between the user and the AR environment, leveraging the capabilities of Howler.js to enhance the overall immersive experience. This approach addresses platform-specific user interaction requirements, such as those mandated by iOS for initiating media playback.

### 4.3.5 Spatial Audio

Incorporating spatial audio became a natural progression in the development process after successfully integrating Howler.js to manage audio playback for multiple models. Spatial or 3D audio is pivotal in replicating how humans naturally perceive sound in a three-dimensional space (see Figure 4.14). By mimicking the natural way sounds come from different directions and distances, spatial audio provides a more immersive and realistic auditory experience [73].

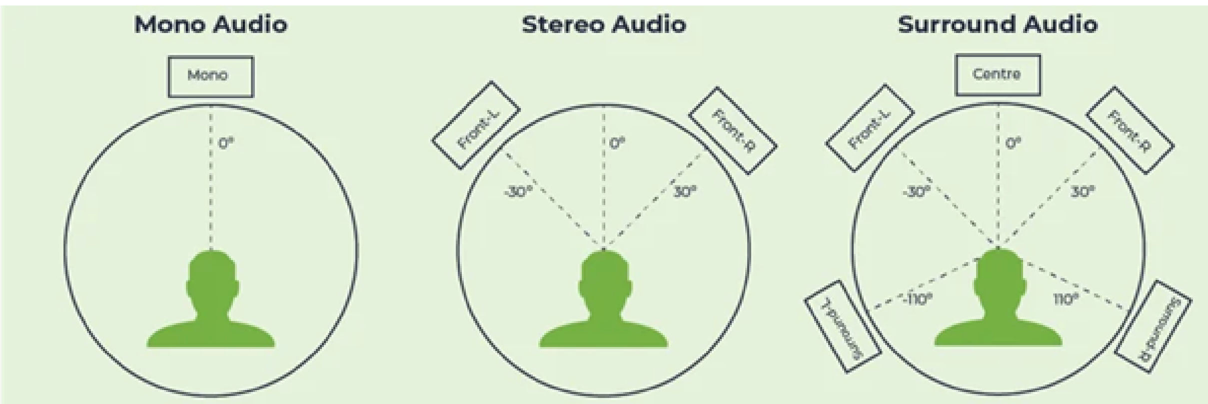



Figure 4.14: Different Audio Source Types [74]

The desire to enrich user immersion in AR drove the decision to incorporate spatial audio. Spatial audio's ability to simulate three-dimensional sound perception significantly elevates the user experience by [73]:

- **Increasing Immersion:** Spatial audio replicates how we hear sounds in the real world, contributing to a more immersive environment and making digital experiences feel more lifelike and engaging.
- **Enhancing Spatial Awareness:** It allows users to locate sound sources within the virtual environment, adding a layer of spatial understanding crucial for navigation and interaction.
- **Improving Navigation and Interaction:** Spatial audio can guide users toward specific actions or locations, adding to their sense of agency within the AR experience.
- **Adding Emotional and Narrative Depth:** Similar to the cinema, spatial audio in AR can evoke emotions and enhance storytelling, contributing to the overall impact of the experience.
- **Increasing Accessibility:** For users with visual impairments, spatial audio provides essential contextual information about their environment, helping them navigate and interact more effectively.

During Howler.js initialisation, spatial audio is enabled by setting the `spatialSound` property to true. This instructs the library to process audio with spatial considerations in mind. An essential component of implementing spatial audio is ensuring sound dynamically responds to the user's device orientation. To achieve this, an event listener for `deviceorientation` events was added (see Listing 4.9). To adjust the audio accordingly, this listener tracks the device's gamma (left to right tilt) and beta (front to back tilt) angles.

```
if (window.DeviceOrientationEvent) {
  window.addEventListener("deviceorientation", function(event) {
    // Early return if no sound is currently playing
    if (!currentBackgroundSound && !currentBackgroundSound.playing())
       return;

    // Mapping device orientation to sound position for spatial audio
       effect
    let soundX = event.gamma / 45;
    let soundY = 0;
    let soundZ = event.beta / 90;

    // Updating the sound position based on device orientation
    if (currentBackgroundSound && currentBackgroundSound.playing()) {
```

```
            currentBackgroundSound.pos(soundX, soundY, soundZ);
        }
    }, true);
}
```

Listing 4.9: Implementing Spatial Audio

The mapping of sound position based on device orientation, the gamma and beta values, simulates how sound sources might move or stay stationary relative to the user's perspective in the real world. By dividing these values by 45 and 90, the sound's position is normalised within a range that reflects realistic sound movement, enhancing the spatial audio effect.

The process of integrating spatial audio posed several challenges. At first, the spatial audio feature did not work as expected due to the absence of conditional checks for sound playback. After adding necessary checks and correctly setting the event listener's capture phase, the spatial audio features began functioning correctly, representing a significant milestone in the development process.

Integrating spatial audio using Howler.js improved the immersive quality of the AR experience and supported the goal of creating an accessible, engaging, and nuanced digital environment for all users.

## 4.4 UI and UX Considerations

In an environment where digital and physical realms converge, ensuring intuitive, seamless, and engaging interactions for the user is paramount. This section will explore the specific UI components and UX principles integrated into the application, including the design and implementation of on-screen prompts and buttons that facilitate user interaction with the AR app. Additionally, this section will explore how the UI elements can enhance the immersive quality of the AR experience, balancing functionality and aesthetic appeal.

### 4.4.1 Cardboard Mode

In the development of AR applications using A-Frame, a common feature is the VR cardboard mode button, prominently displayed in the right-hand corner of the device's screen as shown in Figure 4.15. This button is part of A-Frame's XR Mode UI component, designed to facilitate an easy transition into VR experiences for devices compatible with Google Cardboard viewers. Its presence is automatic, highlighting A-Frame's built-in support for immersive, cross-platform virtual reality experiences.


Figure 4.15: Cardboard Mode Button in Right-hand Corner

The button has a specific function: it allows users to switch the AR or VR application to a stereoscopic mode suitable for VR headsets. Though this button benefits VR applications, it may not be relevant to all AR projects. In cases where an AR experience is designed for direct interaction with mobile devices or VR immersion is not the primary goal, this button can be considered visual clutter. It can also add to the cognitive load on users by presenting an option that does not align with the intended use case, leading to potential confusion or distraction. A-Frame provides a simple solution to address this concern. The VR button can be removed from the UI by adding the attribute `vr-mode-ui=enabled:false` within the `<a-scene>` tag.

The decision to turn off the VR cardboard mode button in the AR application was driven by a desire to reduce visual clutter and cognitive load. Since the primary audience for this application is not expected to use VR headsets, removing the button is a logical step towards streamlining the user experience. This change ensures that users remain focused on the core elements and interactions of the AR experience without any unnecessary distractions.

While this decision enhances the application's usability by reducing cognitive load and visual clutter, it's essential to acknowledge the trade-off involved. Removing the button limits users' ability to explore the application's VR capabilities, potentially excluding a niche audience interested in such experiences. However, this compromise was deemed acceptable to achieve a more streamlined and focused user interaction model that prioritises the needs of the majority and the application's core objectives.

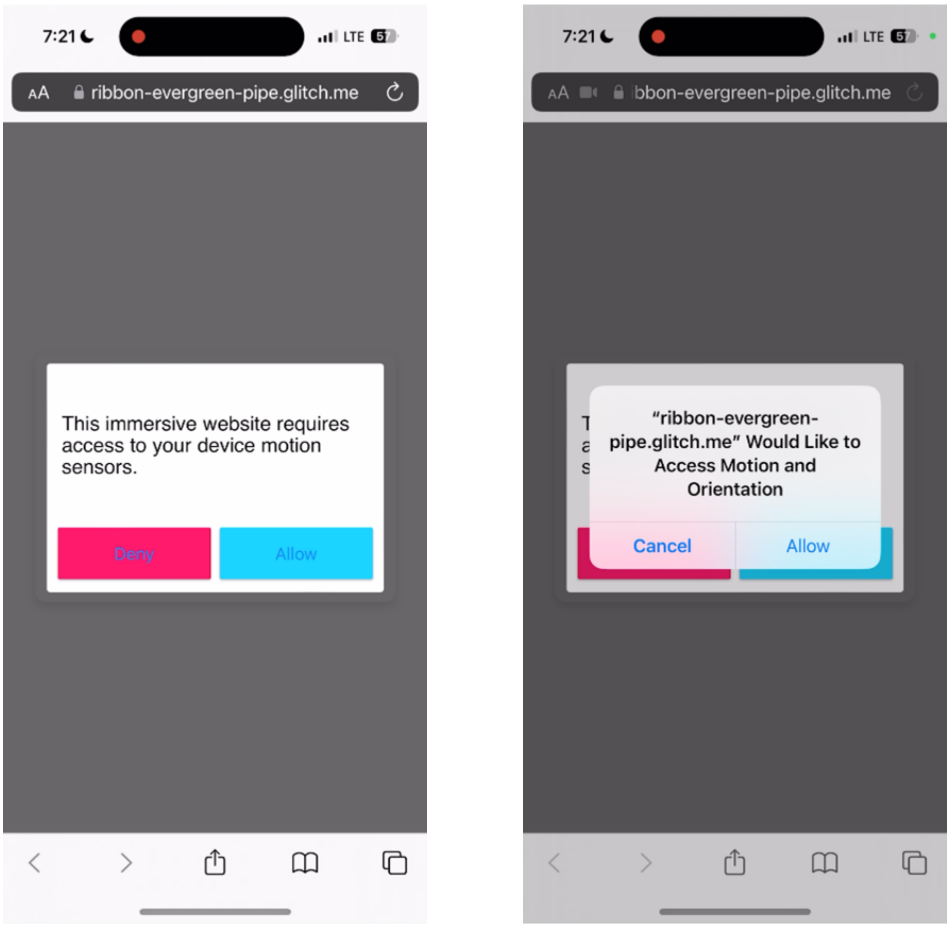
### 4.4.2 Orientation and Camera Prompts

When building AR applications using A-Frame, it is crucial to address the complexities of user permissions, such as device orientation. With the advent of Safari on iOS 13, a heightened emphasis on security and privacy has necessitated that sites not only be served over HTTPS but also explicitly request user permission to access DeviceOrientation events. This requirement introduces an additional layer of user interaction through permission dialogues. A-Frame's `device-orientation-permission-ui` component has been designed to manage these permissions within the `<a-scene>` element [75].

Upon accessing an A-Frame-based AR webpage, users are immediately greeted with a default orientation and camera prompt (see Figure 4.16a), an initial step designed to secure user consent for accessing the necessary device features. Following the A-Frame prompt, the device launches an additional prompt for camera access and orientation data (see Figure 4.16b). This device-specific prompt is mandatory and cannot be disabled, as it is mandated by the device's operating system for security.

This multi-step process requires users to navigate through two separate prompts before engaging with the AR application. From a UX perspective, this sequence is less than ideal. Requiring multiple interactions for access permissions can be perceived as cumbersome and disruptive, potentially detracting from the overall design quality and user engagement. Design guidelines universally advocate minimising friction within user interactions, emphasising that excessive prompts and clicks can lead to frustration and a diminished user experience.

The initial plan was to merge all the necessary permissions into a single prompt to address the issue, as illustrated in Figure 4.17. This approach aimed to minimise




(a) A-Frame Prompt (b) Device Prompt

Figure 4.16: Onboarding Prompts

the effort required from users at the outset, aligning with principles of UCD that prioritise simplifying interactions to reduce cognitive load and enhance usability [36]. The solution was to create a prompt that would cover both volume adjustment and orientation/camera settings, minimising the effort required from the user at the beginning.

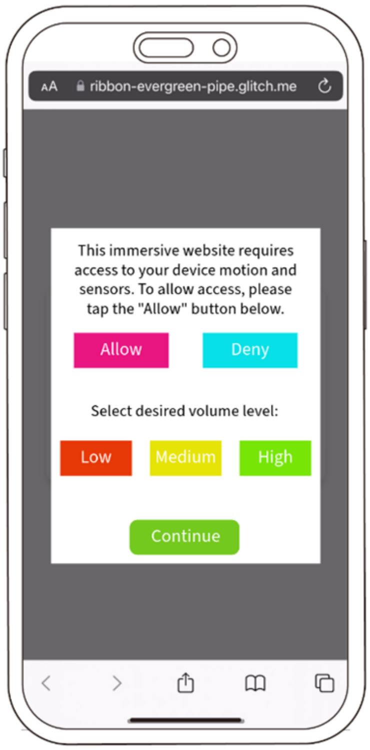



Figure 4.17: Concept of Streamlined Onboarding Prompt

Extensive research was undertaken to pursue this streamlined approach, exploring A-Frame documentation, forums, and other resources for guidance on customising the default prompt. While the documentation suggested altering the button's CSS and the dialogue to predefined options, it became clear upon reviewing A-Frame's source code that such modifications were limited by serialisation. Serialisation in this context refers to how A-Frame's configurations, including the permissions prompt, are preset and not easily altered without direct changes to the framework's source code.

Attempting to customise the prompt revealed several trade-offs. Customising the prompt by modifying the source code manually may result in a more streamlined and user-friendly initiation process. However, this method has certain disadvantages, including increased maintenance complexity, a higher risk of errors, and a tighter coupling between the application and a modified version of A-Frame, which negates the advantages of using a widely supported open-source framework.

Despite the UX benefits of reducing clicks and interactions, as underscored by the WCAG principle of making UI components operable and understandable [39], the constraints posed by A-Frame and device-specific security measures ultimately limited the feasibility of simplifying the onboarding process.

After careful consideration, it was decided to stick with the default A-Frame prompt configuration rather than customise it. This decision was made as a compromise between achieving the best user experience and adhering to necessary security protocols. Although imperfect, this solution underscores the difficulties and dependencies in developing AR applications and the ongoing effort to tackle these challenges to create enjoyable and accessible digital experiences.

### 4.4.3 Volume Prompts

The focus then shifted towards enhancing the application's volume control feature. The aim was to enable users to customise the AR experience's audio to fit their preferences. The initial volume prompt, as illustrated in Figure 4.18, was designed to allow users to easily manage their audio experience with options for Full, Low, and Mute sound levels.

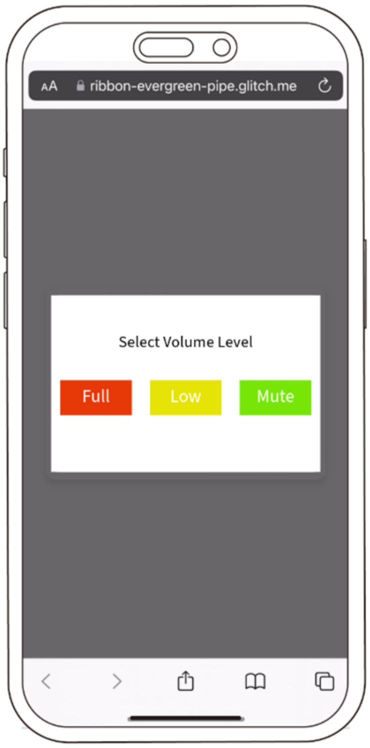



Figure 4.18: Volume Modal Mockup

The initial approach to incorporating a volume control feature was simple: a modal structure directly inserted into the HTML markup immediately after the `<a-scene>` tag closing (see Listing 4.10). This strategic placement ensured that the modal would be a part of the DOM but not interfere with the 3D scene rendered by A-Frame, allowing for it to be overlaid on top of the AR content when needed.

```
<div id="volumeModal" class="modal">
  <div class="modal-content">
    <h2>Select Volume Level</h2>
    <div class="button-container">
      <button onclick="setVolume(1)" class="full-button">Full</button>
      <button onclick="setVolume(0.3)" class="low-button">Low</button>
      <button onclick="setVolume(0)" class="mute-button">Mute</button>
    </div>
  </div>
</div>
```

Listing 4.10: Volume Button Modals

The volume control modal, structured with `<div>` elements, comprised an outer `.modal` container for page placement and an inner `.modal-content` section housing the title header and `.button-container` for the volume buttons. This compartmentalised design streamlined the management of styling and scripting. Complementing this, the CSS aimed to centre the modal on the user's screen, enhancing it with padding, a visually appealing background colour, and softbox shadows for depth. Buttons were traffic-light colour-coded for quick identification and styled with ample padding and rounded borders to create a user-friendly, interactive interface.

The JavaScript logic behind the modal was then created (see Figure 4.11). It featured a delayed display to ensure it appeared in sequence after orientation prompts using the `setTimeout` function. This strategic choice ensured that users were not immediately overwhelmed by prompts upon app initialisation. A global audio variable was set at the top of the script file to allow consistent volume control across the application, with functions defined to manage the modal display and volume settings.

```
document.addEventListener("DOMContentLoaded", function () {
  setTimeout(showVolumeModal, 1000);
  // additional code
};

let globalVolume = 1;

function showVolumeModal() {
  document.getElementById("volumeModal").style.display = "block";
}

function setVolume(volume) {
  globalVolume = volume;
  document.getElementById("volumeModal").style.display = "none";
}
```

Listing 4.11: Volume Button Modals

Significant visual issues were discovered during the first iteration that affected the user experience. The font was inconsistent with the rest of the application design, and the prompt box was off-centre (see Figure 4.19).

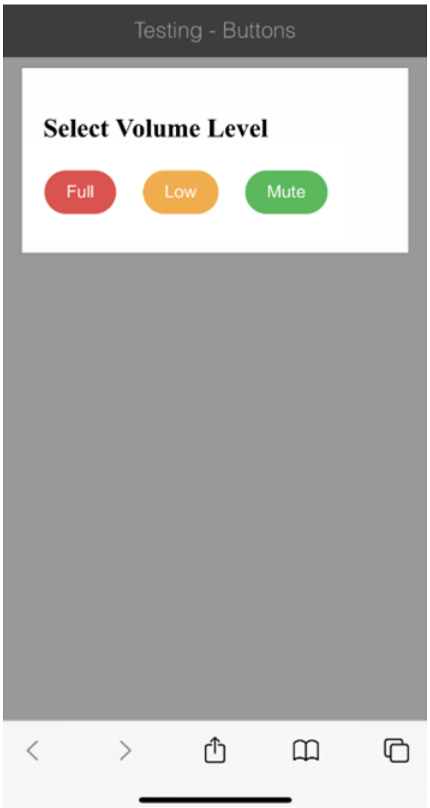



Figure 4.19: First Iteration of the Volume Modal

Several CSS changes were implemented to fix some problems. Specifically, the fonts were updated to match the application's overall design, and the modal's position and size were adjusted to centre it on the screen. After further testing, it was found that the box's visual appearance had improved, but it was still not fully centred and remained stuck to the side of the screen (see Figure 4.20).

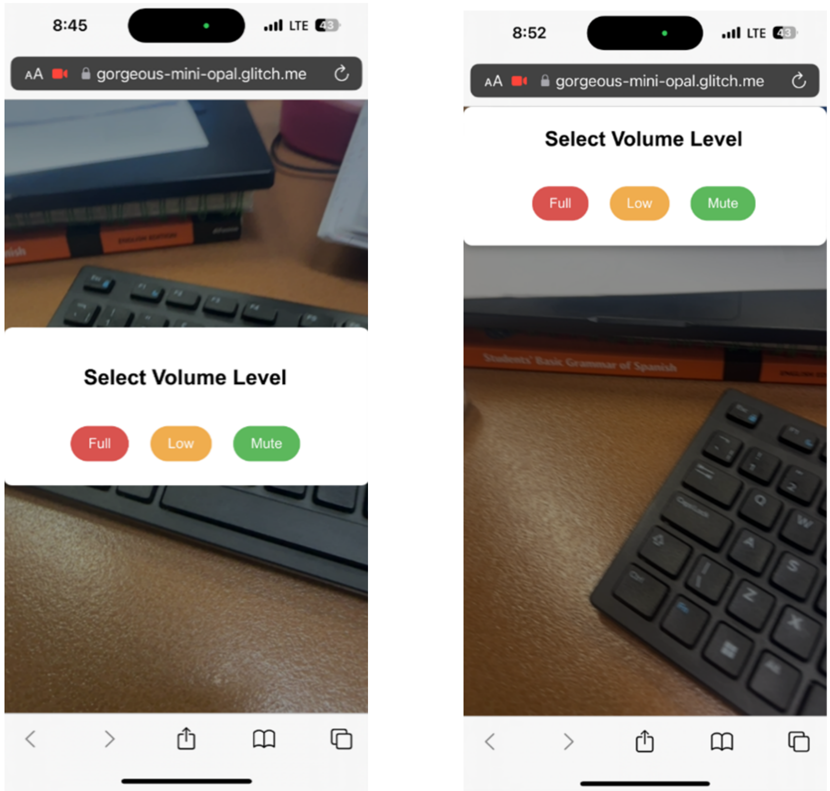



Figure 4.20: Volume Modal Sticking to Various Sides of the Page

The problem with positioning was quite challenging to solve. The usual CSS techniques for centring an element were not working correctly in this case. The modal did not stay centred and kept getting stuck to the top or side of the window. It seemed likely that CSS properties related to position, transform, and width were causing the issue. After changing the position to fixed, the width to eighty per cent, and adding a z-index value, the box finally appeared where it was supposed to be (see Figure 4.21).

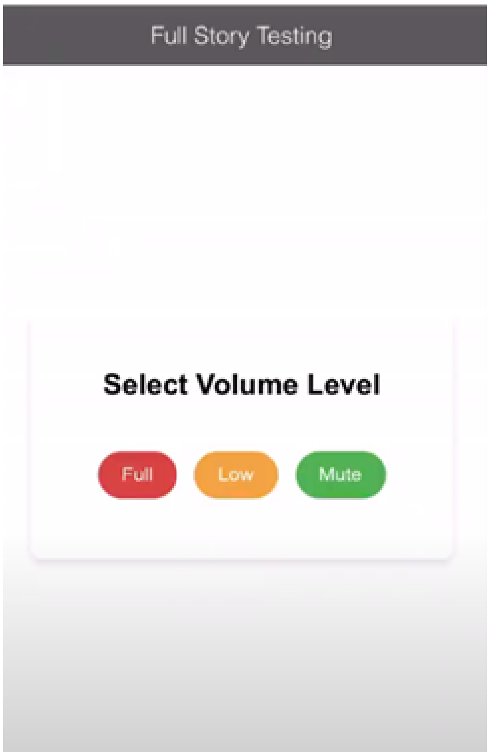



Figure 4.21: Correct Placement of the Volume Modal

After further tests and discussions, it was discovered that the initial design, despite its functional intentions, did not meet the critical accessibility requirements, particularly for ASD users. The colour-coded buttons, inspired by a traffic light system, and the labels used were found to be not as intuitive or accessible as initially intended.

To address these issues, the design approach was significantly changed. Instead of using button-based controls, a slider mechanism was adopted (see Figure 4.22). Adopting a slider mechanism was a strategic and usability-driven decision based on the initial volume control design. The implementation of the slider represented a more accessible approach, offering a seamless method for users to adjust volume within the AR application.

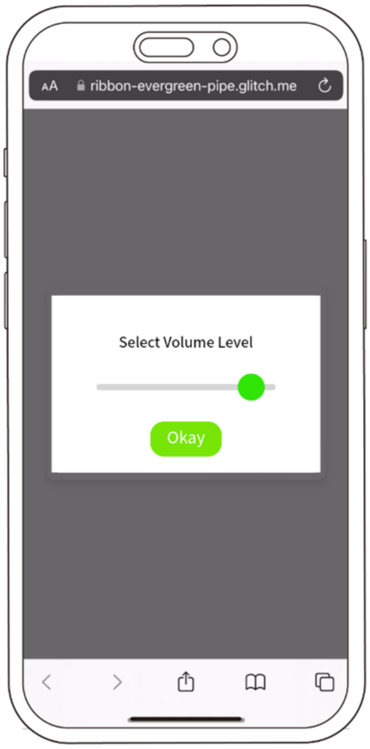



Figure 4.22: Slider Volume Modal Mockup

Adding a `<div>` for the volume slider below the `<a-scene>` tag in the revised HTML markup ensured a non-intrusive overlay when adjusting volume settings (see Listing 4.12). The slider was initialised with a step value of 0.01 for fine-grained control, which was later adjusted to 0.1 to simplify the user interaction and better match user expectations for volume adjustment increments.

```
<div id="volumeModal" class="modal">
  <div class="modal-content">
    <h2>Select Volume Level</h2>
    <!-- Volume Slider -->
    <div class="slider-container">
      <input type="range" id="volumeSlider" min="0" max="1"    step="0.1"
       value="1" role="slider"/>
    </div>
    <button id="confirmVolume" class="ok-button">Okay</button>
  </div>
</div>
```

Listing 4.12: Volume Button Modals

The corresponding CSS styles were then adapted from the previous modal design, ensuring the slider aligned visually with the application's aesthetic. This involved using the::-webkit-slider-thumb and ::-moz-range-thumb pseudo-elements to customise the appearance of the slider thumb across different browsers, ensuring a consistent user experience.

The JavaScript logic, encapsulated within the DOMContentLoaded event listener, managed the timing and interaction of the volume control modal. Most of the functions previously established to manage the volume settings remained the same, with the addition of a function to handle the confirmation action when the user selected their preferred volume level.

Overall, the shift to a slider-based volume control not only addressed usability and accessibility concerns but also demonstrated an agile response to the nuanced requirements of AR application development. The slider's ease of implementation was facilitated by the insights gained from previous challenges with modal positioning, which had already been overcome.

## 4.5 Story Adaptation

The AR application initially planned to explore two themes: "Being Nice to Pets" and "A Trip to the Hairdressers." However, a shortage of suitable models and animations made "A Trip to the Hairdressers" unfeasible. Consequently, the focus was redirected towards fully developing "Being Nice to Pets." This narrative was eventually divided into two stories, one dedicated to cats and the other to dogs. This division was based on the understanding that the distinct behaviours and needs of these pets required individualised stories to effectively convey the intended educational messages.

The decision to divide the information into two stories was based on several important factors. The main one recognised that cats and dogs have significantly different behaviours, methods of communication, and care requirements. Individuals with ASD must comprehend these distinctions to foster secure, respectful, and compassionate interactions with these animals. For example, while dogs may thrive on direct interaction, cats often prefer a more measured approach, highlighting the necessity for tailored educational content.

The next step involved rewriting and developing new storyboards for each segment after deciding to divide the story into separate segments for cats and dogs (see Figure 4.23 Figure 4.24). The main focus of this process was to refine the educational content and ensure that the narratives were more closely aligned with the accompanying AR experiences. The objective was to create complete social narratives that, combined with customised AR scenarios, would deliver a more engaging and immersive learning experience.

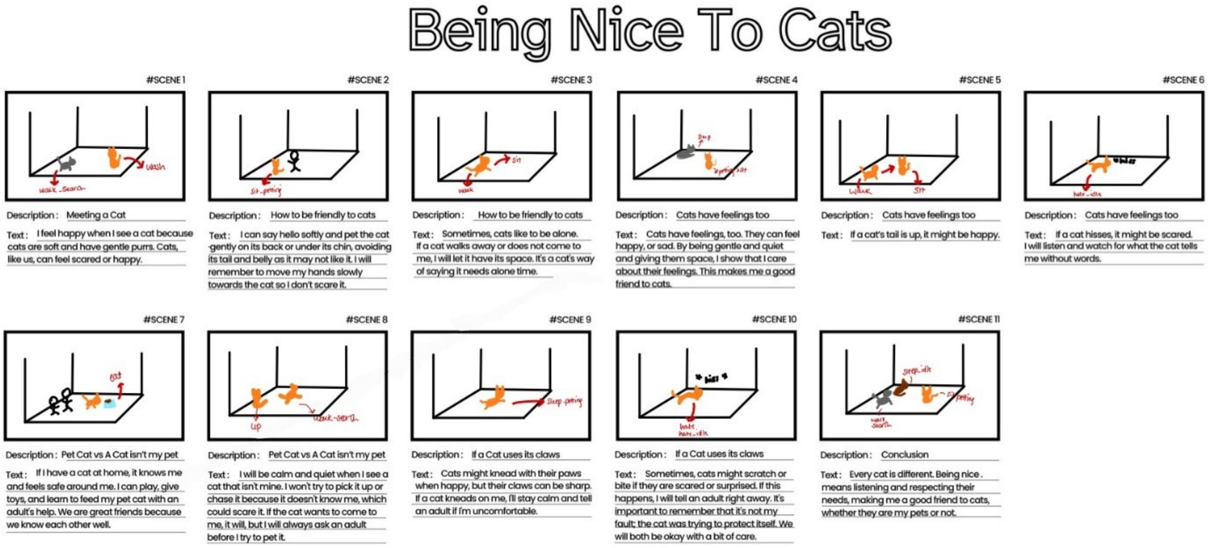



Figure 4.23: Being Nice to Cats Storyboard

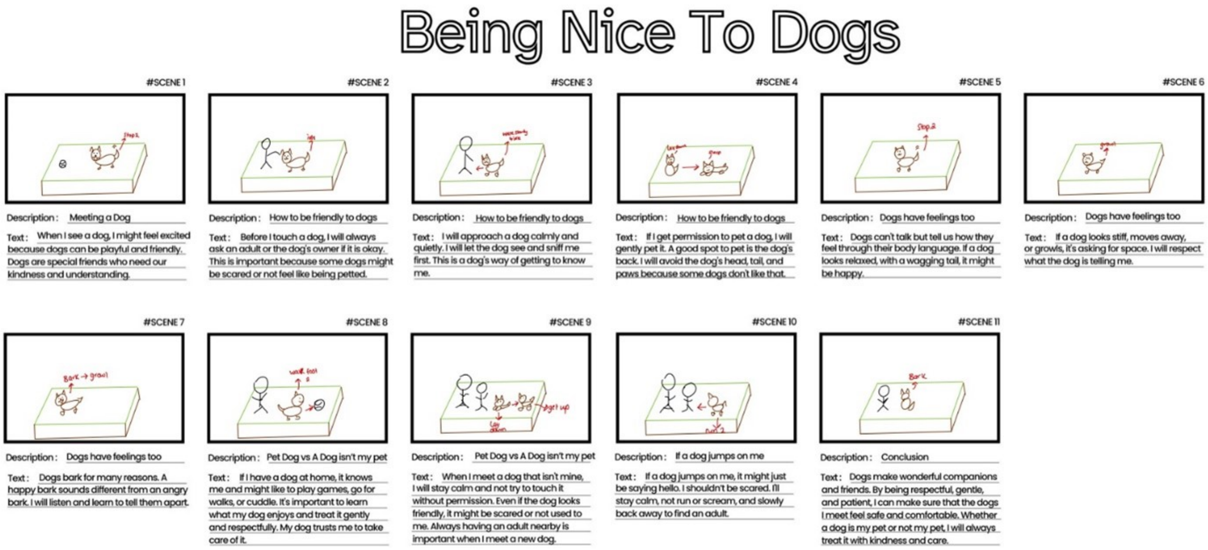



Figure 4.24: Being Nice to Dogs Storyboard

Resource limitations were effectively managed by strategically splitting "Being Nice to Pets" into two focused narratives. This resulted in creating engaging AR stories that offer nuanced insights into interacting with different pets, thereby accommodating the unique learning needs of individuals with ASD.

## 4.6 Text Boxes and Voiceovers

The original concept for the AR application was closely tied to the goal of complementing traditional social storytelling methods. It was envisioned as a dynamic companion to a physical storybook, designed to offer a unique and engaging learning experience for individuals with ASD. The idea was for another person, such as a parent or caregiver, to read aloud from the storybook while the individual interacted with the AR application. This would create a multisensory storytelling experience that could enhance their understanding and interaction with the narrative content.

However, a significant challenge emerged from this initial concept. Relying on another person to read the story could limit the application's accessibility and usability. In scenarios where an ASD individual might be alone, or another person wasn't available to read the story, switching between a physical book and the AR app could decrease the technology's benefits. The possibility of this dual requirement becoming cumbersome or tiring for the user was a concern that needed to be addressed, as it went against the advantages of using a smartphone-based AR experience for independent use.

A significant decision was made to solve a problem and make the application inclusive and self-sufficient. The decision involved integrating the story directly into the app, enabling users to engage with the story independently without requiring a physical book or an external narrator. Two distinct versions of the application were developed to cater to the diverse preferences and needs of the intended audience. One version includes text boxes accompanied by background sounds for those who prefer reading or may benefit from the visual reinforcement of text. The other version provides a VO narrative and background sounds for users who prefer audio narration or find it more accessible and engaging.

After adapting the "Being Nice to Cats" story, two versions were created to test which users preferred. One version was visually driven, while the other was audio-driven. This A/B testing would be crucial in gathering direct feedback from users to determine the most effective and preferred mode of delivery for the narrative content. The results of this testing phase would then determine which version would be used as the final iteration of the app.

### 4.6.1 Adding Text Boxes

The initial phase of integrating textual narratives into the AR application began with an optimistic outlook towards the simplicity of adding text boxes.

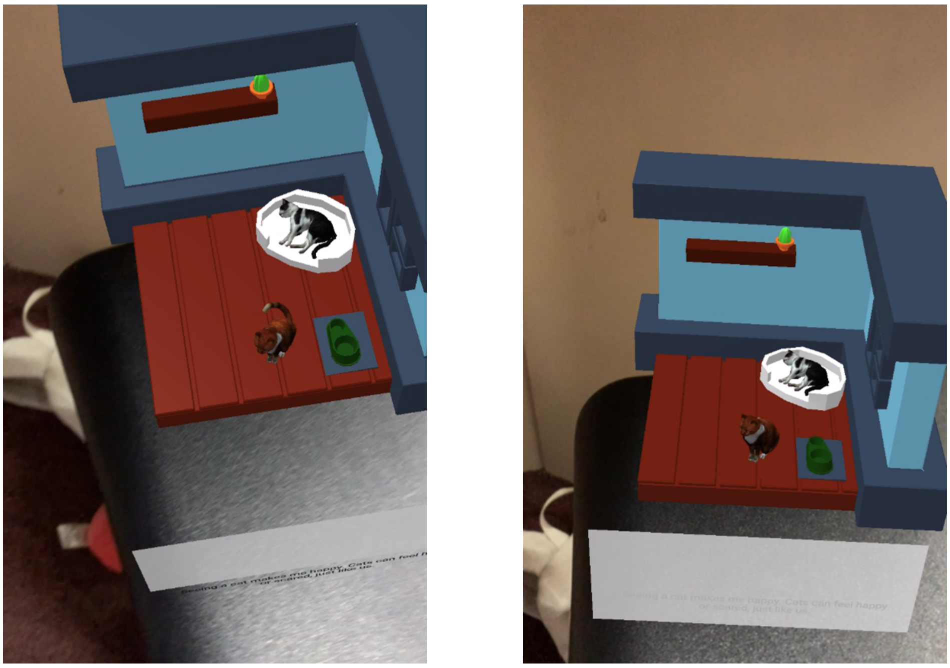

Figure 4.25: Text-box Mockup

The aim was to display the story text on a slightly translucent plane, creating a clear but subtle backdrop for the narrative, with the text positioned to float just in front of the plane for visibility (see Figure 4.25). To create this, I used A-Frame's `<a-text>` and `<a-plane>` primitives within a `<a-marker>` tag. This approach seemed straightforward, allowing easy adjustment and an engaging and straightforward story presentation alongside the AR visuals (see Listing 4.13).

```
<a-plane material="opacity: 0.8; transparent: true;" position="0 1 2"
    rotation="-90 0 0" width="2" height="1"></a-plane>

<a-text value="Seeing a cat makes me happy. Cats can feel happy or
    scared, just like us." position="0 1 2.1    width="2.2    align="
    center" visible="true" wrap-count="15"></a-text>
```

Listing 4.13: Text-boxes Using Planes

Initial tests of this method showed promise, appearing to work from an overhead angle. However, when viewed directly, there was a strange problem: the text appeared to recede behind the plane despite being positioned in front as shown in Figure 4.26.



(a) Text Appears Infront of Plane from Above

(b) Text Appears Behind when Viewed Directly

Figure 4.26: Text-box Issues

The struggle with adjusting the positioning of elements in the AR scene led to an in-depth search through documentation and various forums. It was discovered that the underlying issues were associated with the render order and the `logarithmicDepthBuffer` setting incorporated into the `<a-scene>` tag. Initially intended to improve visual depth handling, this setting inadvertently complicated the layering and visibility of 3D objects within the scene. A-Frame's rendering order primarily determines how various elements in a scene are visually layered. This mechanism usually operates seamlessly for most 3D objects. However, the default rendering order can lead to unexpected results when integrating 2D elements like text and planes, especially when these elements need to interact closely.

A-Frame's default handling of transparent objects based on DOM order rather than spatial logic resulted in the text appearing behind the plane from specific perspectives [76].

The `logarithmicDepthBuffer`, designed to enhance depth precision in scenes with significant scale disparities or at extreme distances, introduced another layer of complexity. While this setting effectively reduces z-fighting among 3D objects, it alters depth value calculations, affecting how 2D elements are perceived concerning each other and the 3D environment. This adjustment meant traditional depth-based sorting methods were less effective, complicating the intended visual hierarchy between text and planes. The issue's complexity was daunting, but it also presented an opportunity to delve deeper into the intricacies of A-Frame and AR technologies. To overcome this, modifications were made to the render order and depth testing properties of both the plane and text elements, hoping to improve the text's visibility.

Despite these efforts, the problem persisted, underscoring a complex interplay between A-Frame's rendering mechanics and the `logarithmicDepthBuffer` functionality. The challenge was not merely a coding error but a systemic limitation, affecting the feasibility of the initial text box approach. This realisation necessitated developing an alternative approach to effectively incorporate text boxes within the application.

New challenges emerged despite attempts to use CanvasUI elements for dynamically generating text textures. Several JavaScript functions were also created to update these text textures based on marker visibility. Nevertheless, this approach failed to deliver the expected results, as the AR models did not display at all. This led to significant frustration and uncertainty about potential solutions for effectively incorporating text boxes into the application.

After considerable trial and error, the breakthrough came from revisiting the `<a-plane>` concept, using SVG text images as textures. The decision to use SVG over other image formats like PNG or JPEG was driven by key factors that set SVG apart, particularly its scalability and quality. SVG files have the unique advantage of scaling infinitely without losing quality. This format is based on mathematical equations defining shapes and lines, ensuring images remain sharp and clear regardless of size. This feature makes SVG perfect for responsive web design and high-resolution displays, essential for maintaining image clarity across various screen sizes [77] [78]. Adobe Illustrator was used to create high-quality, scalable SVG images, ensuring textual clarity and maintaining visual fidelity (see Figure 4.27).


Figure 4.27: Using Adobe Illustrator to Create SVG Files

To use SVGs as plane textures, they were uploaded to Glitch as assets and added to the plane tag using the `src` attribute to specify the URL file to be loaded. Thankfully, this method was successful (see Figure 4.28), providing relief and satisfaction after a turbulent week of troubleshooting.


Figure 4.28: SVG Text Box

Although using SVGs as plane textures achieved the desired results, there were some drawbacks. While visually appealing, SVG images pose accessibility challenges as screen readers cannot interpret them. This could potentially exclude users who rely on such technologies. Additionally, making changes to the text within the SVG images requires manual edits, which can be time-consuming and complicate updating content and localisation efforts.

Although faced with multiple challenges, the SVG texture technique proved the most viable solution. This approach balanced visual clarity with the technical limitations of the AR platform. While imperfect, it emphasises the importance of creativity and flexibility in digital development, mainly when working on complex AR storytelling projects.

### 4.6.2 Adding Voiceovers

Integrating VOs into the AR application was a deliberate decision to improve accessibility and autonomy, especially for users with ASD. This consideration was crucial, recognising that individuals with ASD may have different preferences or needs when processing information. Reading may present challenges for some, or they may be more receptive to auditory content. As a result, VOs are essential to ensure that all users, regardless of their reading ability or preference for auditory learning, can fully engage with and benefit from the application.

Incorporating VOs into the AR application was pivotal in making the experience more immersive and accessible. This process started with recording the VOs on my phone. To ensure the audio files were manageable in size and format for web use, I converted them from m4a to mp3 using Audiomass [70]. This tool also helped to trim unnecessary pauses, making the audio concise and engaging for users.

The refined mp3 files were uploaded to Glitch and integrated within the `<a-marker>` tags through a newly created attribute, `voice-over-sound`. The JavaScript code also underwent adjustments to manage this new audio layer effectively. Updating existing conditions and loops to account for the VO sounds ensured that all background and VO audio elements were perfectly synchronised with the user's interaction with the markers (see Listing 4.14).

```
marker.addEventListener("markerLost", function () {
  const model = marker.querySelector("a-entity");
  model.setAttribute("visible", false); // Hide model and pause
     animation

  if (model.components["animation-mixer"]) {
    model.setAttribute("animation-mixer", "timeScale: 0");
  }

  if (currentBackgroundSound && currentBackgroundSound.playing()) {
    currentBackgroundSound.pause();
  }
  if (currentVOSound && currentVOSound.playing()) {
    currentVOSound.pause();
  }
});
```

Listing 4.14: Synchronising Animations with Background and VO Sounds

New Howl instances were initiated to manage the background and VO sounds, carefully adjusting the volume settings for each to optimise the listening experience. Personal tests determined that a background audio volume setting of 0.2 struck a perfect balance, ensuring VOs were clear without letting the background sound become distracting. Additionally, spatial audio configurations for the background and VO tracks were refined to enhance the overall audio blend. This careful calibration ensured a balanced and engaging audio environment for the users.

Implementing VOs was an uncomplicated process, especially when contrasted with the challenges of adding text boxes. It primarily involved introducing new variables and minor tweaks to the event listeners. This simplicity allowed for a more focused effort on fine-tuning the user experience, ensuring that the audio was integrated as a core component of the immersive experience rather than an add-on.

## 4.7 Accessibility

To enhance the AR application's accessibility, efforts were concentrated on implementing accessible HTML practices and ARIA (Accessible Rich Internet Applications) standards to ensure that the application is usable and inclusive for all individuals, including those with disabilities [41] [44].

One of the fundamental principles that was applied to enhance the accessibility of the AR application was good semantic HTML. The use of semantic elements, such as `<main>`, plays a crucial role in web accessibility, as it helps in clearly defining the structure of the content, making it easier for screen readers and other assistive technologies to interpret the page layout. Semantic tags provide a context to the content enclosed within them, which is invaluable for users relying on assistive technologies to navigate and understand the content [41] [44].

One attempt to improve accessibility was incorporating a `<main>` tag to identify the primary content of the application. However, this modification unexpectedly hindered the display of 3D models within the app, leading to its removal to maintain the app's functionality. This situation highlights the intricate relationship between accessibility improvements and the technical context of AR applications. In AR, specific HTML structures may not directly translate to an improved user experience due to the unique nature of AR content rendering.

ARIA labels were used strategically to improve the application's accessibility, especially for the volume control feature. ARIA is a framework that provides ways to make web content and applications more accessible to people with disabilities [44]. The volume slider is an interactive element crucial to the user's control over the application's sound experience. Therefore, ARIA attributes were added to enhance its functionality, making it transparent and accessible to users who rely on assistive technologies (see Listing 4.15).

```html
<div id="volumeModal" class="modal" aria-labelledby="volumeModalLabel"
    aria-hidden="true">
  <div class="modal-content">
    <h2 id="volumeModalLabel">Select Volume Level</h2>
    <!-- Volume Slider -->
    <div class="slider-container">
      <input type="range" id="volumeSlider" min="0" max="1" step="0.1"
          value="1" aria-valuemin="0" aria-valuemax="100" aria-valuenow="
          100" aria-valuetext="100% volume" role="slider" aria-label="
          Volume control"/>
    </div>
    <button id="confirmVolume" class="ok-button">Okay</button>
  </div>
</div>
```

Listing 4.15: Adding ARIA Attributes to Volume Slider

The JavaScript volume functions were also edited to dynamically update ARIA attributes based on the slider's position, ensuring that the slider is visually intuitive, understandable, and usable through screen readers and other assistive technologies.

The decision to limit ARIA labels to the volume slider instead of applying them to each AR scene or text box is based on the primary mode of interaction within the application. Since the VO feature provides auditory descriptions of the content, adding ARIA labels for screen readers to narrate already spoken content could lead to redundancy and overwhelm the user with simultaneous audio inputs. This consideration highlights a thoughtful balance between implementing accessibility features and maintaining a user-centric approach to application design. It ensures that enhancements contribute to a more accessible and navigable user experience.

This approach to accessibility underscores a commitment to inclusivity, ensuring the AR application is innovative, engaging and accessible to the broadest possible audience, including those with visual impairments or other disabilities. Integrating ARIA attributes and understanding how to apply accessibility standards within the context of AR demonstrates an ongoing effort to refine and adapt the application to meet the diverse needs of its users.

# Chapter 5

# Evaluation & Testing

## 5.1 Evaluation Objectives and Methodology

The main goal of the AR application evaluation was to assess its usability, educational impact, and overall user experience. This was particularly important as the application was developed for individuals with ASD who require engaging, accessible, and practical learning tools. A robust methodology was adopted for this evaluation to ensure a comprehensive understanding of the application's performance from the user's perspective.

Evaluation objectives include:

- **Usability:** To evaluate the AR application's user experience, including text and VO versions, for ease of use and efficiency.
- **Engagement:** Compare the effectiveness of text box and VO versions in engaging users and identifying preferences.
- **Educational Impact:** To evaluate the effectiveness of an AR application in conveying educational content and improving the learning experience for users.

To achieve these objectives, a mixed-methods approach was utilised, combining both quantitative and qualitative research methodologies:

- **Stability Testing and Performance Optimization:** Before user testing, stability testing was conducted to identify any technical limitations or issues within the system. This proactive step ensured that the application was optimised for performance, minimising variables that could affect the user experience during the evaluation phase.

- **User Testing - A/B Testing Approach:** The core of the user testing involved A/B testing, where participants tested both the application's text box and VO versions. This comparative analysis was essential to comprehend user preferences and the relative strengths of each version in terms of usability and engagement.

- **Standardized Questionnaires:** To quantitatively assess usability and user satisfaction, the survey incorporated two established standardised questionnaires: the System Usability Scale (SUS) and the User Satisfaction Evaluation (USE) questionnaire. These instruments were chosen for their reliability, validity, and ability to provide practical feedback on the application's performance. Microsoft Forms was chosen for surveying because it can quickly and easily create basic surveys and questionnaires [79]. It offers built-in analytics for viewing results as they are submitted and the capability to export data to Excel for additional analysis.

The mixed-methods approach aimed to provide a comprehensive analysis. Analysing quantitative data through descriptive statistics to quantify usability and satisfaction, while qualitative responses, particularly from open-ended questions, aim to offer deeper insights into user preferences, experiences, and suggestions for improvement.

## 5.2 Stability Testing and Performance Optimization

This section discusses the stability and performance testing before external user testing. This phase aims to identify and resolve any system limitations or issues. Specifically, the testing focused on the AR system's ability to recognise and track markers under different conditions. The tests were designed to simulate various real-world scenarios that users might encounter. This included different lighting conditions, background complexity, marker distance and orientation, and device performance to ensure a comprehensive system evaluation.

### 5.2.1 Findings from Stability Testing

In the pursuit of ensuring the reliability and usability of the AR application, stability testing uncovered several key findings regarding system performance. These findings include:

- **Lighting Conditions:** Natural lighting significantly enhanced marker detection, while artificial lighting sometimes caused delays due to the glare and reflections from the black border around the marker.

- **Marker Orientation and Movement:** Although the system can recognise marker movements and orientation changes accurately, it can experience detection delays when rapid or abrupt changes occur.
- **Background Variability:** The detection of markers was affected by the background against which a marker was presented, with specific backgrounds causing issues with detection efficiency. Marker detection was more efficient against lighter backgrounds.
- **Distance and Device Interaction:** The system's detection capabilities were influenced by the device's proximity to the marker and the interaction angle.

### 5.2.2 Implemented Changes

After performing stability testing, critical changes were implemented to address issues and enhance the AR system's performance and user experience.

- **Marker Design Modifications:** The findings from stability testing highlighted the need for improved marker detection under various conditions. As a result, the size of the markers' borders was increased to provide a higher contrast ratio against different backgrounds. This alteration was made physically and mirrored in the app's code to better accommodate the improved design. This change reduced the occurrence of glare and enhanced detection reliability.
- **Background Adaptability Guidelines:** Background variability significantly influenced marker detection. Guidelines were established for strategically placing markers within the physical book as a direct response. Ensuring a predominantly white or uniform background behind the markers helped mitigate issues related to complex or patterned backgrounds that previously reduced detection efficiency.

Some initial challenges with the AR application were identified during the stability testing phase. However, these issues were remedied by making targeted modifications that improved the app's responsiveness and reliability. These modifications addressed the immediate technical concerns and laid a stronger foundation for the upcoming phases of external user testing.

## 5.3 User Survey

### 5.3.1 Designing the User Survey

The design of the user survey was a crucial step in evaluating the AR application. The primary objectives of the user testing were to assess the application's ease of use and engagement and to determine which version - text boxes or VOs - users preferred. A mixed methods approach was used to achieve these goals, utilising qualitative and quantitative research methodologies to gather comprehensive feedback.

Two established questionnaires, the System Usability Scale (SUS) and the User Satisfaction Evaluation (USE) questionnaire, were chosen for their proven reliability and straightforward implementation in online forms. These specific questionnaires were selected to facilitate unbiased, structured feedback that could be easily analysed. The decision to utilise standardised questionnaires was made due to their numerous advantages [80]:

- **Objectivity:** These tools enable unbiased assessment by allowing evaluators to validate each other's findings independently.
- **Ease of Replication:** Utilising standardised methods simplifies the replication of research, enhancing the reliability of usability evaluations compared to ad hoc methods.
- **Detailed Reporting:** They provide a structured framework for detailed reporting beyond subjective judgment, facilitating the application of statistical analysis to understand the data better.
- **Improved Communication:** Using standard measures, practitioners communicate effectively, overcoming communication barriers that can stall progress.

SUS is recognised for its simplicity and efficiency. It consists of a ten-item questionnaire that provides an overall view of subjective usability evaluations (see Figure 5.1). It employs a Likert scale approach, which involves carefully selecting statements for respondents to evaluate beyond just gauging agreement or disagreement on a 5- or 7-point scale. This tool has garnered a reputation for its robustness and reliability, demonstrating solid correlations with other subjective usability measures [81].

The USE questionnaire, developed by Lund, focuses on capturing data regarding a product or system's Usefulness, Ease of Use, Ease of Learning, and Satisfaction. All items are positively worded, with responses ranging from 1 (strongly disagree) to 7 (strongly agree) (see Appendix E). While detailed psychometric information has yet

| | The System Usability Scale Standard Version | | Strongly disagree 1 | 2 | 3 | 4 | Strongly agree 5 |
|---|---|---|---|---|---|---|---|
| 1 | I think that I would like to use this system. | | ○ | ○ | ○ | ○ | ○ |
| 2 | I found the system unnecessarily complex. | | ○ | ○ | ○ | ○ | ○ |
| 3 | I thought the system was easy to use. | | ○ | ○ | ○ | ○ | ○ |
| 4 | I think that I would need the support of a technical person to be able to use this system. | | ○ | ○ | ○ | ○ | ○ |
| 5 | I found the various functions in the system were well integrated. | | ○ | ○ | ○ | ○ | ○ |
| 6 | I thought there was too much inconsistency in this system. | | ○ | ○ | ○ | ○ | ○ |
| 7 | I would imagine that most people would learn to use this system very quickly. | | ○ | ○ | ○ | ○ | ○ |
| 8 | I found the system very cumbersome to use. | | ○ | ○ | ○ | ○ | ○ |
| 9 | I felt very confident using the system. | | ○ | ○ | ○ | ○ | ○ |
| 10 | I needed to learn a lot of things before I could get going with this system. | | ○ | ○ | ○ | ○ | ○ |

Figure 5.1: SUS Questionnaire [81]

to be published, the USE is noted for its comprehensive development approach, which includes a large initial item pool and factor analysis, among other methods [82].

I could leverage their strengths in evaluating usability by utilising both the SUS and the USE questionnaire. These questionnaires offer quantifiable insights regarding the application's usability and satisfaction, and they complement each other nicely, resulting in a thorough assessment [82]. To prevent repetition and instead gain unique insights into user satisfaction and educational value, the usability section of the USE questionnaire was intentionally omitted while focusing on its distinct strengths.

The survey organisation was meticulously planned using Forms365, with questions divided into separate sections for clarity and cohesiveness. Open-ended questions were strategically included to gather detailed feedback, asking participants to specify their preferred application version, why, and suggestions for improvement. This approach aimed to balance the quantitative ease of SUS and USE data with the qualitative richness of personal user experiences, providing a well-rounded view of the application's impact.

The ultimate goal of the user survey was to gather valuable feedback that could further guide refinements to the AR application. The survey utilised standardised and open-ended questionnaires to gather valuable feedback on the application's usability, accessibility, and educational value. This method ensured a thorough evaluation of the AR application, guiding future refinements to enhance its usability, accessibility, and educational efficacy.

### 5.3.2 Gathering Results

The results of the AR application evaluation were gathered over three separate days. A diverse group of participants, including my workplace colleagues, coursemates, and friends and family, participated. This structured approach allowed for a wide range of feedback on the application's usability and effectiveness.

A consistent procedure was followed during each testing session to ensure that the data collected was reliable and comparable. Participants were first provided an overview of the application and the study's objectives. They were also given a physical copy of the storybook associated with the AR content (see Appendix C and Appendix D). This introduction effectively set the stage for the testing and ensured that participants were aligned with the study's goals and objectives.

Next, the participants interacted with the AR application, getting a firsthand experience of its interactive storytelling features. This engagement was crucial, marking the first test of the application's functionality across various device types, such as iOS and Android platforms. The diversity in hardware and operating systems would provide crucial insights into the app's performance and usability across different platforms.

After using the application, participants were asked to complete a survey that captured their impressions, preferences, and suggestions for improvement. The feedback mechanism was instrumental in identifying the strengths and areas for app development.

The approach taken to collect data aimed to closely replicate actual usage circumstances, guaranteeing that the obtained feedback was genuine and relevant. By employing a tangible storybook in conjunction with the AR app in an organic environment, participants could engage with the technology in a way that closely resembled its intended function, resulting in a well-rounded user experience.

## 5.4 Analysis of Survey Results

### 5.4.1 System Usability Scale (SUS) Results

The SUS provided a composite measure of the overall usability of the AR application. This section presents an aggregated view of the SUS scores obtained during the user testing phase, offering insights into the application's user-friendliness and functional integrity.

The SUS score is a composite measure calculated by adding the individual contributions from each of the ten items on the questionnaire. These scores for each item ranged from 1 to 5, with the odd-numbered items subtracted from 5 and the even-numbered items subtracted from 1. Subsequently, these adjusted scores were summed and multiplied by 2.5 to scale the result to a 100-point system [81].

The SUS scores generated from this process have provided a quantifiable understanding of the application's usability. The application's SUS score was 86.4, thus scoring the system well above the average threshold of 68. To ensure my calculations were correct, I utilised an online SUS calculator, which also stated that the average SUS score was 86.4 (see Figure 5.2).

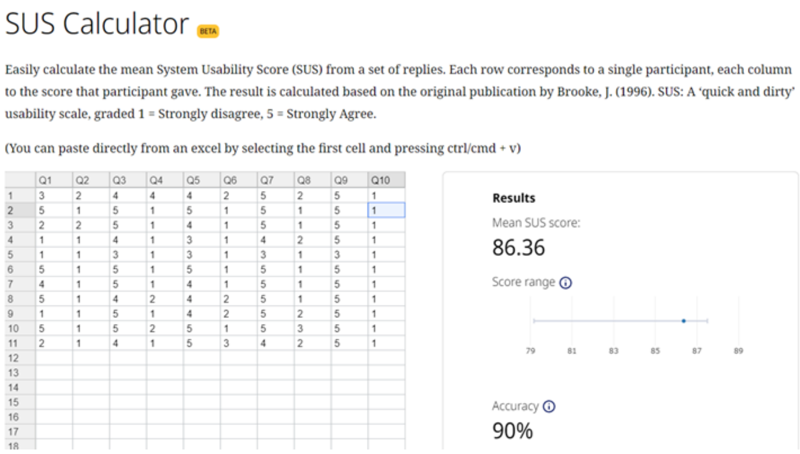



Figure 5.2: SUS Calculator Results

The SUS scores from the user survey quantify the app's usability and offer diverse insights into how users perceive the AR application. The overall SUS score can be interpreted in several ways [83]:

- **Percentiles:** With a score surpassing the 68th percentile, the app's usability is deemed satisfactory, suggesting a performance better than half of the comparative systems.

- **Grading System:** Employing a school grading scale from A+ to F, where C is average, the application achieved an "A+" grade, indicating exceptional usability.
- **Descriptive Adjectives:** Users often attribute adjectives like "Good," "OK", or "Poor" to a product's usability. The term "Best Imaginable" emerged as the descriptor for the application, reflecting the highest usability acclaim.
- **Acceptability Scale:** Usability can also be categorised as "acceptable" or "unacceptable". The application comfortably sits in the "acceptable" range, denoting a positive user experience.
- **Net Promoter Score (NPS):** SUS scores can inform the Net Promoter Score (NPS), predicting the likelihood of users recommending the app. Based on the responses, users fall into the categories of promoters (scores 9-10), passives (scores 7-8), and detractors (scores 0-6). The app primarily fell into the 'promoter' category, signifying a high probability of users recommending it to peers.

| Grade | SUS | Percentile range | Adjective | Acceptable | NPS |
|---|---|---|---|---|---|
| A+ | 84.1-100 | 96-100 | Best Imaginable | Acceptable | Promoter |
| A | 80.8-84.0 | 90-95 | Excellent | Acceptable | Promoter |
| A- | 78.9-80.7 | 85-89 | | Acceptable | Promoter |
| B+ | 77.2-78.8 | 80-84 | | Acceptable | Passive |
| B | 74.1 – 77.1 | 70 – 79 | | Acceptable | Passive |
| B- | 72.6 – 74.0 | 65 – 69 | | Acceptable | Passive |
| C+ | 71.1 – 72.5 | 60 – 64 | Good | Acceptable | Passive |
| C | 65.0 – 71.0 | 41 – 59 | | Marginal | Passive |
| C- | 62.7 – 64.9 | 35 – 40 | | Marginal | Passive |
| D | 51.7 – 62.6 | 15 – 34 | OK | Marginal | Detractor |

Figure 5.3: SUS Results Interpretations

These interpretations, as shown in Figure 5.3, indicate that the AR application is highly usable and well-received by its users. This suggests that the application has the potential to generate positive word-of-mouth and user satisfaction.

### 5.4.2 User Satisfaction Evaluation (USE) Questionnaire Results

Unlike the SUS, the USE Questionnaire has no standard scoring formula that converts responses to a single number. Instead, the USE questionnaire is typically analysed by calculating the mean (average) score for each of the questionnaire's subscales (Satisfaction, Ease of Use, Ease of Learning) and the overall average across all items.

Points based on their Likert scale response were assigned to each question to calculate the average score for each subscale. The points for each item within a subscale for each respondent were summed and then divided by the number of questions in the subscale to get the mean score. The average scores for each subscale were:

- Ease of Use: 6.1
- Ease of Learning: 6.7
- Satisfaction: 6.3

The overall average was then calculated by summing the mean scores of all subscales and dividing by the number of subscales, with the overall USE average coming to 6.4.

The results obtained from the USE questionnaire illustrate the effectiveness of the AR application from the users' standpoint. The Ease of Use, Ease of Learning, and Satisfaction subscales scored above 6, indicating that the application is easy to use and learn and provides a satisfactory user experience. These high scores across the subscales demonstrate not only the usability of the application but also the quick learning curve and overall satisfaction experienced by the users. The overall average score of 6.4 out of 7 suggests that the application is functionally practical and positively perceived by users.

### 5.4.3 A/B Testing Feedback

The A/B testing aimed to compare user preferences, ease of use, visual appeal, and storytelling effectiveness between the AR application's VO and text box versions. The survey gathered responses from various participants, including insights into what made each version easier to use compared to the other, along with overall enjoyment and storytelling effectiveness.

When evaluating the user's preference between the two versions of the application, it's essential to consider the factors that affect their choices. The preference indicates how users interact with the technology and their personal comfort and engagement levels with the content delivery method. Here are the results of the participants' feedback regarding their overall preference for the VO version versus the text box version:

- **Majority Preference:** A significant number of participants (7 out of 11) preferred the VO version. Common reasons for this preference included the ease of focusing on animations without the need to read text, making the story feel more immersive and enjoyable.

- **Text Box Preference:** Participants who preferred the text box version (4 out of 11) cited its visual appeal, ease of use in public locations, and a preference for reading.

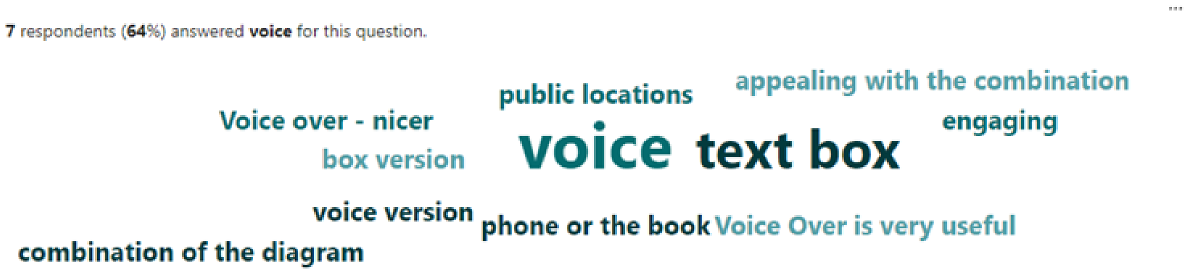



Figure 5.4: Word Cloud Representation of User Preferences Between Voice and Text Box Versions

The ease of use is an essential aspect of any application, as it significantly impacts user satisfaction and continued usage. A comparison between the VO and text-box versions can provide insights into which version users found more intuitive and easier to navigate. Here are the results of the participants' feedback regarding each version's ease of use as rated by our participants:

- **VO Version:** Participants almost unanimously rated it as "Very Easy" to use, indicating its intuitive nature and the convenience of listening over reading.

- **Text Box Version:** It is generally rated high but slightly lower than the VO version, with a few users finding it more challenging to read the text box under certain conditions.

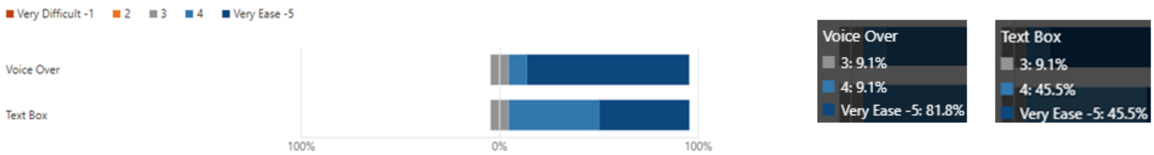



Figure 5.5: Bar Graph Comparing User Ratings on the Ease of Use for Voice Over and Text Box Versions (1 = Difficult, 5 = Very Easy)

An application's aesthetic appeal and enjoyability can significantly enhance the user experience, making the content more captivating and memorable. These factors are crucial in keeping the user engaged and interested in the application. Here are the results of the participants' feedback regarding the visual appeal and overall enjoyment of both versions of the application:

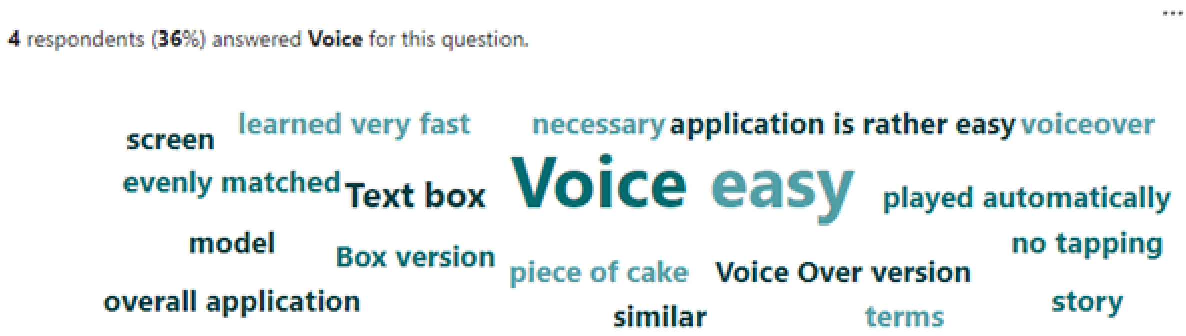



Figure 5.6: Word Cloud Highlighting User Feedback on the Ease of Use for Voice Over Versus Text Box Versions

- **VO Version:** A majority found it more enjoyable, attributed to the ability to listen to the story, making it feel like having company. The auditory element added an extra layer of engagement.
- **Text Box Version:** This was enjoyed for its immersive nature and the blend of visuals with text. However, some mentioned the challenge of managing the camera to read the text.


Figure 5.7: Bar Graph Showing User Ratings on the Aesthetic Appeal of Voice Over and Text Box Versions

The effectiveness of an AR application lies in its ability to tell a story. How the content is delivered - whether through voice or text - can significantly affect how well the user conveys and receives the story. Below are the results of the participants' feedback on which version was more successful in telling the story or conveying the message and why:

- **VO Version:** Seen as doing a better job at storytelling for most users due to its ability to minimise misinterpretation and maintain focus on animations.
- **Text Box Version:** A minority praises it for its immersive reading experience, but the VO version overshadows its storytelling capabilities.

Navigating and using a new application efficiently is essential for a positive user experience. This segment explores the learning curve associated with each version, focusing on

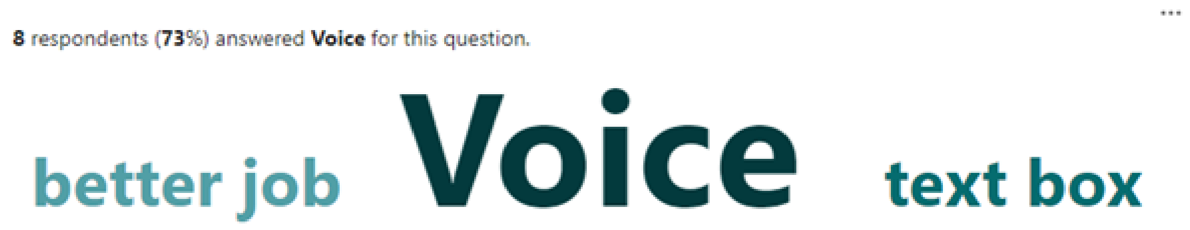



Figure 5.8: Word Cloud Depicting User Opinions on Storytelling Effectiveness of Voice Over and Text Box Versions

user feedback regarding which version they found more straightforward to learn and why:

- The application's design was intuitive and facilitated quick learning for both versions, with minor differences based on user interaction.

Feedback is an essential tool in the iterative design process of any application. Analysing users' suggestions helps developers identify areas where adjustments are required to improve usability, engagement, and content delivery. In this part of the analysis, the focus is on users' constructive feedback for both the VO and text box versions of the application to further refine it:

- **VO Version:** Suggestions include addressing issues with pattern recognition interruption and the desire for a play/pause button to manage the VO and animations.
- **Text Box Version:** Feedback highlighted the need for better zoom management on different devices, in this case, android, and the possibility of integrating text and VO features for enhanced flexibility.

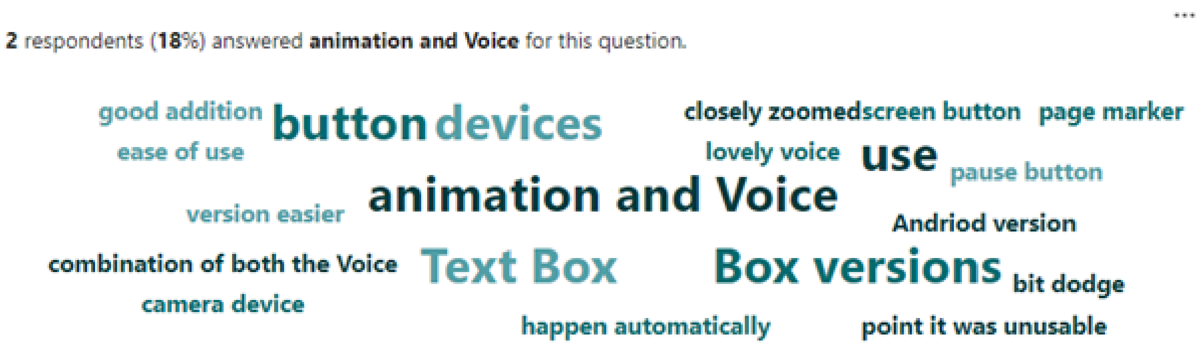



Figure 5.9: Word Cloud Illustrating User Suggestions for Improvements

During the A/B testing, it was found that users preferred the VO version of the application over the text box version. This preference was cited for ease of use, engagement, and storytelling effectiveness. Although the text box version was appreciated for its visual appeal and immersive nature, the convenience and simplicity of the VO version made it the favoured choice among users. These results played a significant role in shaping the final version of each story, ultimately leading to the creation of a VO version.

### 5.4.4 Overall Evaluation

The A/B testing revealed a distinct preference for the VO version of the application, highlighting the importance of auditory cues in enhancing user engagement and story comprehension. This preference underlines the effectiveness of auditory feedback in creating an immersive experience that allows users to fully concentrate on the animations and the storyline without the cognitive load of simultaneously processing text and visuals.

The analysis of the SUS scores underlined the application's high usability, with scores indicating that the application falls within the "excellent" usability category. The USE questionnaire results corroborated these findings, showing high scores across the Ease of Use, Ease of Learning, and Satisfaction subscales, affirming that the application is easy to use and learn and delivers a satisfying user experience.

Several critical areas for enhancement were identified through user feedback:

- **Android Compatibility:** Feedback from Android users revealed a zoomed-in view that detracted from the application's usability. To address this, a specific modification was implemented in the application's HTML code to correct the zoom discrepancy, ensuring consistency across diverse device types:

```html
<meta name="viewport" content="width=device-width, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0"/>
```

Listing 5.1: Android Compatibility Enhancements

  This change allowed Android users to use the app without issues and let them see the entire AR scene rather than just a zoomed-in portion.

- **Scene Size Adjustments:** "Sometimes the size of the room with the cats is bigger than the other, so then it doesn't fit into the frame on my iPhone 13." Feedback like this highlighted the need for scene size adjustments to ensure the application's visual content was comfortably viewable on smaller screens.

- **User Interaction:** Consideration is given to adding a "tap to start" button to prompt users to initiate the animations and VOs, making the application more intuitive and user-friendly.

The valuable feedback and insights obtained during the evaluation phase are crucial in determining the next steps for the application's development. These improvements, chosen to enhance user accessibility and improve the overall user experience, demonstrate a dedication to evolving the application based on actual user requirements and preferences.

## 5.5 Discussion

The testing process undertaken in this project was not only a necessary step for ensuring the functionality and usability of the AR application but also a critical phase for validating the design choices and technological strategies employed throughout the development. Through testing, several critical aspects of the project were validated, reinforcing the decisions made during the development phase and highlighting the application's potential impact on users with ASD.

### 5.5.1 Validation of Design Choices

The user feedback collected during A/B testing between the application's text box and VO versions highlighted a strong preference for the VO version. This preference emphasises the importance of auditory elements in educational tools for those who may find audio cues less distracting and more engaging than text. This finding supports the initial design decision to incorporate a VO feature to enhance the storytelling aspect and make the learning experience more intuitive and accessible.

### 5.5.2 Importance of Usability and Accessibility

The high usability scores obtained from the SUS questionnaire and the positive feedback on ease of use and learning in the USE questionnaire further validated the application's design from an accessibility standpoint. These results are particularly significant given the target user group, confirming that the application is practical and accessible. This aligns with the project's goal of making educational content more accessible through technology.

### 5.5.3 Enhancing Engagement through AR

The feedback reveals a clear preference for an AR-enhanced approach over traditional methods, indicating that AR can significantly enhance engagement and comprehension. This finding supports the project's theoretical foundation that AR technology can revolutionise educational experiences for individuals with ASD by making learning more interactive and immersive. The testing validated the effectiveness of this approach and underscored AR's ability to boost engagement levels, which is critical for achieving better educational outcomes in unique education settings.

# Chapter 6

# Conclusion

## 6.1 Summary

This project embarked on a journey to enhance social skills training for individuals with Autism Spectrum Disorder (ASD) by leveraging the capabilities of augmented reality (AR). The developed AR application aimed to transcend the limitations of traditional learning methods by integrating interactive multimedia elements that cater specifically to the sensory preferences and learning needs of individuals with ASD.

Over the months of development, I have deeply immersed myself in the intricacies of AR technologies, particularly AR.js, and their application in creating engaging educational content. This firsthand experience has deepened my understanding and also confirmed the potential of AR as a transformative tool in special education. Integrating sound, animation, and interactive storytelling within the AR environment significantly enhanced user engagement, as evidenced by the positive feedback and high usability scores obtained during the evaluation phase.

The project's success lies in its technical achievements and validation of AR as a valuable and practical educational resource for social skills training. This exploration into AR's application has been worthwhile, offering substantial insights into its capabilities and limitations. The practical benefits of the AR application, including enhanced user involvement and personalised learning opportunities, have proven invaluable, reinforcing my dedication to utilising technology to tackle real-world issues.

## 6.2 Future Work

Looking ahead, several avenues for further development could not only refine the existing application but also expand its scope and impact:

- **Expansion to Virtual Reality (VR) and Mixed Reality (MR):** Transitioning from AR to MR and VR could offer even more immersive learning experiences by catering to a broader range of sensory preferences and learning needs. It's an exciting prospect that could potentially revolutionise this educational tool.
- **Development of More Complex Storylines and Higher Quality Models:** As I continue to hone my skills in JavaScript and modelling software like Blender, I'm embarking on a journey to create more intricate narratives and animations. These enhancements are not just about making the stories more engaging and relatable but also about my personal growth and dedication to improving the educational and aesthetic value of the application.
- **Implementation of a Wider Range of Stories:** I am particularly interested in developing the original story about visiting a hairdresser, a common challenge for individuals with ASD. This story would enhance the app's relevance and utility, helping users navigate everyday situations more confidently.
- **Comprehensive User Testing with ASD Individuals:** Testing and refining the application based on direct feedback from users with ASD would ensure that it remains user-centred and effective. I am eager to explore how these stories impact their social skills and confidence.

# Chapter 7

# Appendix

## 7.1 4.3.4 Adding Sounds to the App

### 7.1.1 Appendix A

```
<!-- Script for Synchronising the Animations and Sounds on User tap and model visibility -->
    <script>
      document.addEventListener("DOMContentLoaded", (event) => {
        var sound = new Howl({
          src: [
            "https://cdn.glitch.global/3cc44f88-4d32-41a9-a495-bdcdb6f8ffa4/cat-purring-and-meow-5928.mp3?v=1707835432254",
          ],
          loop: true,
          volume: 0.5,
        });

        var modelVisible = false; // Track visibility of the model

        // Function to Play or Resume sound if model is visible
        function startExperience() {
          if (modelVisible) {
            if (!sound.playing()) {
              sound.play();
            }

            // Start or resume animation
            var animatedModel = document.getElementById("animated-model");
            if (animatedModel) {
```

```
              animatedModel.setAttribute("animation-mixer", "timeScale:1"); // Start or resume animation
            }
          }
        }

        //Stop the sound if it's playing and the model isn't visible anymore
        function stopExperience() {
          if (sound.playing()) {
            sound.stop();
          }

          // add pause reset animation
          var animatedModel = document.getElementById("animated-model");
          if (animatedModel) {
            animatedModel.setAttribute("animation-mixer", "timescale:0");
          }
        }

        // Listening for marker found and lost events to control model visibility
        document
          .querySelector("a-marker")
          .addEventListener("markerFound", function () {
            modelVisible = true;
          });

        document
          .querySelector("a-marker")
          .addEventListener("markerLost", function () {
            modelVisible = false;
            stopExperience();
          });

        // Add event listeners
        document.body.addEventListener("touchstart", startExperience);
        document.body.addEventListener("click", startExperience);
      });
    </script>
```

Listing 7.1: Adding Sound to One Model

### 7.1.2 Appendix B

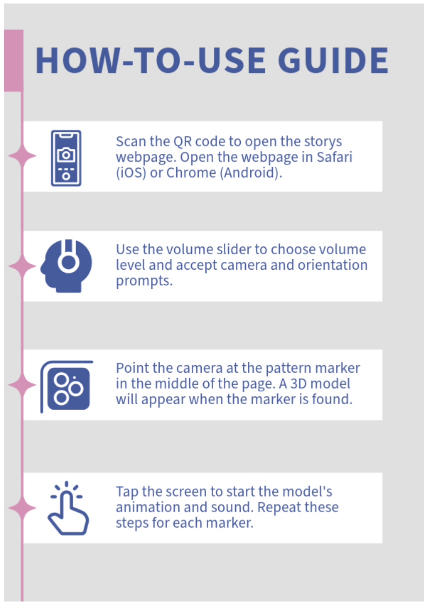
```
<!-- Script for starting model animations and sound when the user taps
```

```
the screen // model visiblity -->
<script>
  document.addEventListener("DOMContentLoaded", function () {
    const markers = document.querySelectorAll("a-marker");
    let currentSound = null;

    markers.forEach((marker) => {
      marker.addEventListener("markerFound", function () {
        const model = marker.querySelector("a-entity");
        model.setAttribute("visible", true); // Make model visible but animation paused
        model.setAttribute(
          "animation-mixer",
          "timeScale: 0; clampWhenFinished: true; loop: repeat"
        );
      });

      marker.addEventListener("markerLost", function () {
        const model = marker.querySelector("a-entity");
        model.setAttribute("visible", false); // Hide model and pause animation
        if (currentSound) {
          currentSound.stop();
        }
        // Animation pauses when the marker is lost
        if (model.components["animation-mixer"]) {
          model.setAttribute("animation-mixer", "timeScale: 0");
        }
      });
    });

    window.addEventListener("touchstart", function () {
      markers.forEach((marker) => {
        if (marker.object3D.visible) {
          const model = marker.querySelector("a-entity");

          // Resume animation if the model is visible and animation was previously paused
          if (model.getAttribute("visible")) {
            model.setAttribute("animation-mixer", "timeScale: 1"); // Resume animation
          }

          // Handle sound
          const soundSrc = marker.getAttribute("data-sound");
          if (currentSound) {
            currentSound.stop();
```

```
                }
                currentSound = new Howl({
                  src: [soundSrc],
                  autoplay: true,
                  loop: true,
                });
              }
            });
          });
        });
      </script>
```

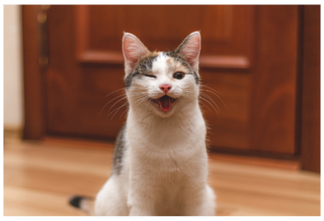
Listing 7.2: Adding Sound to Multiple Models

### 7.1.3 Appendix C

## STORY QR CODE

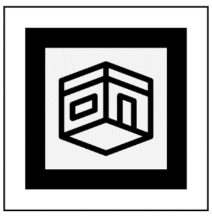
Hello, and welcome to the story “Being Nice to Cats”. This story will teach us how to be kind to cats and how to treat them.

Seeing a cat makes me happy. Cats can feel happy or scared, just like us.

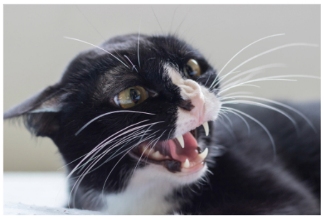
Say hello softly and gently pet the cat's back or chin, avoiding its tail and belly. Move slowly to not scare it.

Sometimes, cats like to be alone. If a cat walks away, give it space—it means it needs alone time.

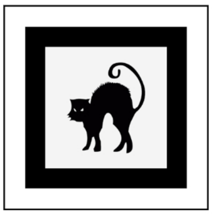


Cats have feelings, too. Cats can feel scared, happy, or sad. Be gentle, quiet, and give them space to show you care.

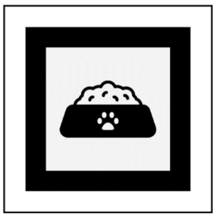
If a cat’s tail is up, it might be happy.



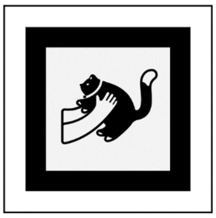
If a cat hisses, it might be scared. I will listen and watch for what the cat tells me without words.



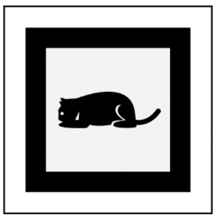
My pet cat knows and feels safe with me. We can play and share toys. I can learn to feed my pet cat with an adult's help.



For cats that aren't mine, I'll stay calm and quiet. I won't pick them up or chase them. Always ask an adult before trying to pet any cat.



Cats might knead with their paws when happy, but their claws can be sharp. If a cat kneads on me, I'll stay calm and tell an adult if I'm uncomfortable.

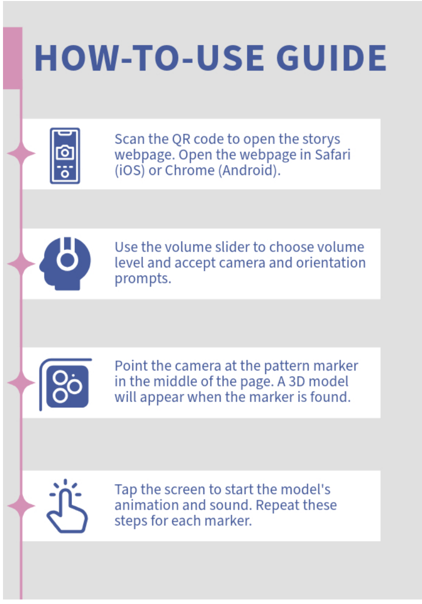


Sometimes, cats might scratch or bite if they are scared or surprised. If this happens, I will tell an adult right away. It’s not my fault – they're just scared or surprised.

Every cat is different. Being nice means listening and respecting their needs, making me a good friend to cats, whether they are my pets or not.

**PLEASE SCAN THE QR CODE BELOW TO GIVE FEEDBACK**

### 7.1.4 Appendix D

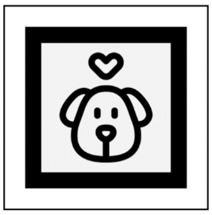




**STORY QR CODE**

SCAN HERE

Hello, and welcome to the story "Being Nice to Dogs". This story will teach us how to be kind to dogs and how to treat them.

Seeing a dog makes me happy. Dogs have feelings just like us.

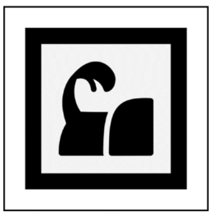


Before I touch a dog, I will always ask an adult if it is okay. Some dogs might be scared or not feel like being petted.

I will approach a dog calmly and quietly. I will let the dog see and sniff me first. This is a dog's way of getting to know me.

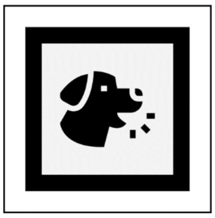
If I'm allowed to pet the dog, I'll do it gently, avoiding sensitive areas like the head, tail, and paws.

Dogs have feelings too. If a dog has a wagging tail, it might be happy.



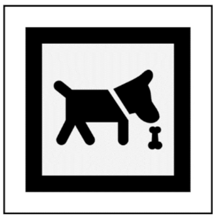
If a dog looks stiff, moves away, or growls, it might be scared. I will watch and listen to what the dog is telling me without words.

Dogs bark for many reasons. A happy bark sounds different from an angry bark. I will listen and learn to tell them apart.



My pet dog feels safe with me. We can play, and I can help take care of it with an adult's help.

For dogs that aren't mine, I'll stay calm and won't try to touch them without asking. Always having an adult nearby is important when I meet a new dog.



If a dog jumps on me, it might just be saying hello. I shouldn't be scared. I'll stay calm, not run or scream, and slowly back away to find an adult.

Every dog is unique. Being nice means understanding and respecting their needs, making me a good friend to dogs, whether they are my pets or not.

**PLEASE SCAN THE QR CODE BELOW TO GIVE FEEDBACK**

### 7.1.5 Appendix E

**USE Questionnaire: Usefulness, Satisfaction, and Ease of use**

**Based on:** Lund, A.M. (2001) *Measuring Usability with the USE Questionnaire.* STC Usability SIG Newsletter, 8:2. | About quest.cgi

Please rate your agreement with these statements.

- Try to respond to all the items.
- For items that are not applicable, use: **NA**
- Make sure these fields are filled in: **System:** **Email to:**
- Add a comment about an item by clicking on its icon, or add comment fields for all items by clicking on **Comment All**.
- To mail in your results, click on: **Mail Data**

**System:** **Email to:**

Optionally provide comments and your email address in the box.

Mail Data | Comment All | RETURN TO REFERRING PAGE

| **USEFULNESS** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | | **NA** |
|---|---|---|---|---|---|---|---|---|---|---|
| 1. It helps me be more effective. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 2. It helps me be more productive. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 3. It is useful. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 4. It gives me more control over the activities in my life. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 5. It makes the things I want to accomplish easier to get done. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 6. It saves me time when I use it. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 7. It meets my needs. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 8. It does everything I would expect it to do. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| **EASE OF USE** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | | **NA** |
| 9. It is easy to use. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 10. It is simple to use. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 11. It is user friendly. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 12. It requires the fewest steps possible to accomplish what I want to do with it. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 13. It is flexible. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 14. Using it is effortless. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 15. I can use it without written instructions. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 16. I don't notice any inconsistencies as I use it. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 17. Both occasional and regular users would like it. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 18. I can recover from mistakes quickly and easily. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 19. I can use it successfully every time. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| **EASE OF LEARNING** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | | **NA** |
| 20. I learned to use it quickly. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 21. I easily remember how to use it. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 22. It is easy to learn to use it. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 23. I quickly became skillful with it. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| **SATISFACTION** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | | **NA** |
| 24. I am satisfied with it. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 25. I would recommend it to a friend. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 26. It is fun to use. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 27. It works the way I want it to work. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 28. It is wonderful. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 29. I feel I need to have it. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| 30. It is pleasant to use. | strongly disagree | ○ | ○ | ○ | ○ | ○ | ○ | ○ | strongly agree | ○ |
| | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | | **NA** |

List the most **negative** aspect(s):

1.
2.
3.

List the most **positive** aspect(s):

1.
2.
3.

Mail Data | Comment All | Top of Form | RETURN TO REFERRING PAGE

# Bibliography


[1] National Institute of Mental Health (NIMH). *Autism Spectrum Disorders (ASD)*. 2024. URL: https://www.nimh.nih.gov/health/topics/autism-spectrum-disorders-asd (visited on 15. Oct. 2023).

[2] Applied Behavior Analysis Programs Guide. *What is Social Skills Training (SST) and When is it Used?* 2020. URL: https://www.appliedbehavioranalysisprograms.com/faq/social-skills-training/ (visited on 15. Oct. 2023).

[3] Carol Gray. *What is a Social Story?* No Date. URL: https://carolgraysocialstories.com/social-stories/what-is-it/ (visited on 5. Oct. 2023).

[4] Microsoft. *What is Augmented Reality (AR)?* No Date. URL: https://dynamics.microsoft.com/en-ie/mixed-reality/guides/what-is-augmented-reality-ar/ (visited on 5. Dec. 2023).

[5] Robert Sheldon, TechTarget. *Virtual Reality (VR) - Definition*. 2022. URL: https://www.techtarget.com/whatis/definition/virtual-reality (visited on 5. Dec. 2024).

[6] AR.js Documentation. *Marker-based Augmented Reality with AR.js*. No Date. URL: https://ar-js-org.github.io/AR.js-Docs/marker-based/ (visited on 10. Oct. 2023).

[7] Fred Churchville, TechTarget. *User Interface (UI) - Definition*. 2021. URL: https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI (visited on 16. Dec. 2023).

[8] Nielsen Norman Group. *The Definition of User Experience (UX)*. 1998. URL: https://www.nngroup.com/articles/definition-user-experience/ (visited on 16. Dec. 2023).

[9] Blender Documentation. *glTF 2.0 - Blender Manual*. No Date. URL: https://docs.blender.org/manual/en/2.80/addons/io_scene_gltf2.html#file-format-variations (visited on 10. Jan. 2024).

[10] Anthony Freda, Avast. *What Are QR Codes and How Do You Scan Them?* 2022. URL: https://www.avast.com/c-what-is-qr-code-how-to-scan (visited on 7. Mar. 2024).

[11] Xiaojie Lian et al. "Evaluating User Interface of a Mobile Augmented Reality Coloring Application for Children with Autism: An Eye-Tracking Investigation". In: *International Journal of Human-Computer Studies* 178 (Oct. 2023), p. 103085. DOI: 10.1016/j.ijhcs.2023.103085. (visited on. ).

[12] I.-Jui Lee. "How to Use the Advantages of AR and VR Technique to Integrate Special Visual Training Strategies in Non-Verbal Communication Skills Training for Children with Autism". In: *Types of Nonverbal Communication*. IntechOpen, 2020. DOI: 10.5772/intechopen.94587. (visited on. ).

[13] Chien-Hsu Chen, I-Jui Lee, and Ling-Yi Lin. "Augmented Reality-Based Video-Modeling Storybook of Nonverbal Facial Cues for Children with Autism Spectrum Disorder to Improve Their Perceptions and Judgments of Facial Expressions and Emotions". In: *Computers in Human Behavior* 55 (Feb. 2016), pp. 477–485. DOI: 10.1016/j.chb.2015.09.033. (visited on. ).

[14] Kamran Khowaja et al. "Augmented Reality for Learning of Children and Adolescents With Autism Spectrum Disorder (ASD): A Systematic Review". In: *IEEE Access* 8 (2020), pp. 78779–807. DOI: 10.1109/ACCESS.2020.2986608. (visited on. ).

[15] Viacheslav V. Osadchyi et al. "Features of implementation of modern AR technologies in the process of psychological and pedagogical support of children with autism spectrum disorders". In: *International Workshop on Augmented Reality in Education*. 2020. (visited on. ).

[16] MDN Web Docs. *HTML: HyperText Markup Language*. No Date. URL: https://developer.mozilla.org/en-US/docs/Web/HTML (visited on 7. Apr. 2024).

[17] MDN Web Docs. *CSS: Cascading Style Sheets*. No Date. URL: https://developer.mozilla.org/en-US/docs/Web/CSS (visited on 7. Feb. 2024).

[18] MDN Web Docs. *JavaScript*. No Date. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript (visited on 7. Feb. 2024).

[19] A-Frame. *A-Frame – Make WebVR*. No Date. URL: https://aframe.io (visited on 26. Oct. 2023).

[20] AR.js Documentation. *AR.js Documentation*. No Date. URL: https://ar-js-org.github.io/AR.js-Docs/ (visited on 26. Oct. 2023).

[21] Glitch. *Glitch: The Friendly Community where Everyone Builds the Web*. URL: https://glitch.com (visited on 26. Oct. 2023).

[22] Blender Foundation. *Blender*. URL: https://www.blender.org (visited on 11. Dec. 2023).

[23] Khronos Group. *glTF – GL Transmission Format*. No Date. URL: https://www.khronos.org/gltf/ (visited on 10. Jan. 2024).
[24] W3Schools. *HTML Introduction*. No Date. URL: https://www.w3schools.com/html/html_intro.asp (visited on 7. Feb. 2024).
[25] A-Frame. *Introduction to A-Frame*. Mo Date. URL: https://aframe.io/docs/1.5.0/introduction/ (visited on 7. Oct. 2023).
[26] Micheal Colonna, VR Wiki at Brown University. *AR.js - Related Technology*. 2023. URL: https://www.vrwiki.cs.brown.edu/related-technology/ar-development-software/ar-js (visited on 7. Oct. 2023).
[27] Glitch. *About Glitch*. No Date. URL: https://glitch.com/about (visited on 12. Jan. 2024).
[28] Martin, VR Wiki at Brown University. *Blender - VR Modeling Software*. 2023. URL: https://www.vrwiki.cs.brown.edu/vr-modeling-software/blender (visited on 7. Oct. 2023).
[29] Inglobe Technologies. *How to create Blender 3D models for Augmented Reality apps*. No Date. URL: https://www.inglobetechnologies.com/create-blender-3d-models-augmented-reality-apps/ (visited on 8. Jan. 2024).
[30] International Organization for Standardization (ISO). *ISO/IEC 12113:2022 - Information technology - Runtime 3D asset delivery format - Khronos glTF™ 2.0*. 2022. URL: https://www.iso.org/standard/83990.html (visited on 8. Jan. 2024).
[31] BR Softech. *Advantages and Disadvantages of Game Development Tools*. 2023. URL: https://www.brsoftech.com/blog/advantages-and-disadvantages-of-best-game-engines/ (visited on 12. Apr. 2024).
[32] Moldstud. *The Role of AR/VR in the Future of Native Mobile Apps*. 2024. URL: https://moldstud.com/articles/p-the-role-of-arvr-in-the-future-of-native-mobile-apps (visited on 12. Apr. 2024).
[33] Reydar. *Web-based Augmented Reality (Web AR): Explained, Benefits, Examples*. No Date. URL: https://www.reydar.com/web-based-augmented-reality-guide/ (visited on 12. Apr. 2024).
[34] Plutomen Hiren Kanani. *What is Web Augmented Reality?* 2019. URL: https://pluto-men.com/what-is-web-augmented-reality-benefits-and-the-future-of-webar-and-web-design/ (visited on 12. Apr. 2024).
[35] GeeksforGeeks. *Architecture of Augmented Reality and Its Components*. 2023. URL: https://www.geeksforgeeks.org/architecture-of-augmented-reality-and-its-components/ (visited on 10. Jan. 2024).
[36] Interaction Design Foundation - IxDF. *What is User Centered Design (UCD)?* 2016. URL: https://www.interaction-design.org/literature/topics/user-centered-design (visited on 10. Jan. 2024).

[37] GeeksforGeeks. *Introduction to UCD (User Centered Design)*. 2023. URL: https://www.geeksforgeeks.org/introduction-to-ucd-user-centered-design/ (visited on 10. Jan. 2024).

[38] CAST. *Universal Design for Learning Guidelines version 2.2*. 2018. URL: https://udlguidelines.cast.org (visited on 10. Jan. 2024).

[39] World Wide Web Consortium (W3C). *Web Content Accessibility Guidelines (WCAG) 2.2*. 2023. URL: https://www.w3.org/TR/WCAG22/ (visited on 11. Jan. 2024).

[40] Casandra Visser, Accessibility Checker. *Web Content Accessibility Guidelines (WCAG) Guide*. 2023. URL: https://www.accessibilitychecker.org/guides/wcag/ (visited on 11. Jan. 2024).

[41] W3Schools. *HTML Accessibility*. No Date. URL: https://www.w3schools.com/html/html_accessibility.asp (visited on 11. Jan. 2024).

[42] MDN Web Docs. *Accessibility - HTML: HyperText Markup Language*. No Date. URL: https://developer.mozilla.org/en-US/docs/Learn/Accessibility/HTML#text_alternatives (visited on 11. Jan. 2024).

[43] World Wide Web Consortium (W3C). *WAI-ARIA Standards and Guidelines*. 2024. URL: https://www.w3.org/WAI/standards-guidelines/aria/ (visited on 23. Jan. 2024).

[44] MDN Web Docs. *Web Accessibility: ARIA*. 2024. URL: https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA (visited on 23. Jan. 2024).

[45] National Autistic Society. *Social Stories and Comic Strip Conversations*. 2020. URL: https://www.autism.org.uk/advice-and-guidance/topics/communication/communication-tools/social-stories-and-comic-strip-coversations (visited on 18. Nov. 2023).

[46] Interaction Design Foundation - IxDF. *Hick's Law*. 2016. URL: https://www.interaction-design.org/literature/topics/hick-s-law (visited on 4. Feb. 2024).

[47] Interaction Design Foundation - IxDF. *What is Fitts' Law?* 2016. URL: https://www.interaction-design.org/literature/topics/fitts-law (visited on 4. Feb. 2024).

[48] Nicolò Carpignoli, Medium - Chialab Open Source. *AR.js - The Simplest Way to Get Cross-Browser AR on the Web*. 2018. URL: https://medium.com/chialab-open-source/ar-js-the-simpliest-way-to-get-cross-browser-ar-on-the-web-8f670dd45462 (visited on 14. Oct. 2023).

[49] AR.js Documentation. *AR.js Documentation - Marker-based*. No Date. URL: https://ar-js-org.github.io/AR.js-Docs/marker-based/ (visited on 14. Oct. 2023).

[50] Nicolò Carpignoli, Medium - Chialab Open Source. *10 Tips to Enhance Your AR.js App*. 2019. URL: https://medium.com/chialab-open-source/10-tips-to-enhance-your-ar-js-app-8b44c6faffca (visited on 14. Oct. 2023).

[51] AR.js Documentation. *AR.js Marker Training Generator*. URL: https://ar-js-org.github.io/AR.js/three.js/examples/marker-training/examples/generator.html (visited on 14. Oct. 2023).

[52] Flaticon. *Flaticon*. URL: https://www.flaticon.com (visited on 14. Oct. 2023).

[53] Figma. *Low-Fidelity Prototyping*. No Date. URL: https://www.figma.com/resource-library/low-fidelity-prototyping/ (visited on 27. Oct. 2023).

[54] Louise Bruton, UX Design Institute. *A Comprehensive Guide to Prototyping*. 2022. URL: https://www.uxdesigninstitute.com/blog/prototyping-guide/ (visited on 27. Oct. 2023).

[55] Sketchfab. *Sketchfab*. URL: https://sketchfab.com/feed (visited on 5. Jan. 2024).

[56] TurboSquid. *TurboSquid*. URL: https://www.turbosquid.com (visited on 5. Jan. 2024).

[57] CGTrader. *CGTrader*. URL: https://www.cgtrader.com (visited on 5. Jan. 2024).

[58] *Mixamo*. URL: https://www.mixamo.com (visited on 5. Jan. 2024).

[59] Gatis Zvejnieks, Overly. *How to Create 3D Content in Blender for Mobile Augmented Reality Projects*. 2023. URL: https://overlyapp.com/blog/how-to-create-3d-content-in-blender-for-mobile-augmented-reality-projects/ (visited on 11. Jan. 2024).

[60] Nathalie Fernandez, Meshmatic. *The experts guide to optimizing your 3D files for AR*. 2020. URL: https://meshmatic3d.com/technical/guide-to-optimizing-3d-files-for-ar/ (visited on 11. Jan. 2024).

[61] Cloudflare. *Why Use HTTPS?* No Date. URL: https://www.cloudflare.com/learning/ssl/why-use-https/ (visited on 4. Jan. 2024).

[62] vincentfretin, A-Frame. *Animation Loaders - Aframe Extras*. No Date. URL: https://github.com/c-frame/aframe-extras/tree/master/src/loaders#animation (visited on 29. Dec. 2023).

[63] Blender Foundation. *glTF 2.0 - Blender 4.1 Manual*. No Date. URL: https://docs.blender.org/manual/en/latest/addons/import_export/scene_gltf2.html (visited on 27. Dec. 2023).

[64] Blender Foundation. *Node Wrangler - Blender 4.1 Manual*. No Date. URL: https://docs.blender.org/manual/en/latest/addons/node/node_wrangler.html (visited on 6. Jan. 2023).

[65] Blender Foundation. *Decimate Modifier - Blender 4.1 Manual*. No Date. URL: https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/decimate.html (visited on 6. Jan. 2023).

[66] LearnOpenGL. *Depth Testing*. No Date. URL: https://learnopengl.com/Advanced-OpenGL/Depth-testing (visited on 21. Jan. 2023).

[67] A-Frame. *Renderer Component - Logarithmic Depth Buffer*. No Date. URL: https://aframe.io/docs/1.5.0/components/renderer.html#logarithmicdepthbuffer (visited on 21. Jan. 2023).

[68] Blender Foundation. *glTF 2.0 IO Scene Add-on*. No Date. URL: https://docs.blender.org/manual/en/2.80/addons/io_scene_gltf2.html (visited on 27. Dec. 2023).

[69] Freesound. *Freesound*. No Date. URL: https://freesound.org (visited on 3. Feb. 2024).

[70] AudioMass. *AudioMass*. No Date. URL: https://audiomass.co (visited on 3. Feb. 2024).

[71] A-Frame. *Sound Component*. No Date. URL: https://aframe.io/docs/1.5.0/components/sound.html (visited on 3. Feb. 2024).

[72] GoldFire Studios. *Howler.js - Mobile Chrome Playback*. No Date. URL: https://github.com/goldfire/howler.js?tab=readme-ov-file#mobilechrome-playback (visited on 3. Feb. 2024).

[73] Interaction Design Foundation - IxDF. *What is Spatial Audio?* 2023. URL: https://www.interaction-design.org/literature/topics/spatial-audio (visited on 7. Feb. 2024).

[74] Anton Venema, LiveSwitch. *Spatial Audio Techniques for Enterprise VR Applications*. 2021. URL: https://www.liveswitch.io/blog/archive/-spatial-audio-vr-guide (visited on 7. Feb. 2024).

[75] A-Frame. *Device Orientation Permission UI*. No Date. URL: https://aframe.io/docs/1.5.0/components/device-orientation-permission-ui.html (visited on 10. Feb. 2024).

[76] A-Frame. *FAQ: What Order Does A-Frame Render Objects In?* No Date. URL: https://aframe.io/docs/1.5.0/introduction/faq.html#what-order-does-a-frame-render-objects-in (visited on 10. Feb. 2024).

[77] Cloudinary. *SVG Format Features, Common Uses, and Pros & Cons You Should Know*. 2024. URL: https://cloudinary.com/guides/image-formats/svg-format-features-common-uses-and-pros-cons-you-should-know (visited on 7. Feb. 2024).

[78] Salman Ravoof, Kinsta. *SVG vs. PNG: What Are the Differences and When to Use Them*. 2023. URL: https://kinsta.com/blog/svg-vs-png/ (visited on 7. Feb. 2024).

[79] Microsoft. *Online Surveys, Polls and Quizzes*. No Date. URL: https://www.microsoft.com/en-ie/microsoft-365/online-surveys-polls-quizzes (visited on 7. Mar. 2024).

[80] Jeff Sauro and James R. Lewis. "Chapter 8 - Standardized Usability Questionnaires". In: *Quantifying the User Experience*. Ed. by Jeff Sauro and James R. Lewis. Boston: Morgan Kaufmann, 2012, pp. 185–240. ISBN: 978-0-12-384968-7. DOI: https://doi.org/10.1016/B978-0-12-384968-7.00008-4. URL: https://www.sciencedirect.com/science/article/pii/B9780123849687000084 (visited on. ).

[81] John Brooke. "SUS: A quick and dirty usability scale". In: *Usability Eval. Ind.* 189 (1995). (visited on. ).

[82] Meiyuzi Gao, Philip Kortum, and Frederick Oswald. "Psychometric Evaluation of the USE (Usefulness, Satisfaction, and Ease of use) Questionnaire for Reliability and Validity". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 62.1 (2018), pp. 1414–1418. DOI: 10.1177/1541931218621322. URL: https://doi.org/10.1177/1541931218621322 (visited on. ).

[83] Jeff Sauro, MeasuringU. *5 Ways to Interpret a SUS Score*. 2018. URL: https://measuringu.com/interpret-sus-score/ (visited on 7. Mar. 2024).