Using AR and VR characters for enhancing user experience in a museum

Kunal Shailesh Shitut
ks6807@rit.edu

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Using AR and VR characters for enhancing user experience in a museum

by

Kunal Shailesh Shitut

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

Department of Computer Science
B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York

March 29, 2021
The thesis "Using AR and VR characters for enhancing user experience in a museum" by Kunal Shailesh Shitut is approved by the following Examination Committee:

Dr. Joe Geigel,
Professor,
Thesis Committee Chair

Dr. Juilee Decker,
Associate Professor,
Reader

Dr. Reynold Bailey,
Professor,
Observer
Acknowledgement

This system was developed in collaboration with a team of faculty experts and student contributors at RIT. I would like to thank Dr. Joe Geigel, Professor in the Department of Computer Science, and Dr. Juilee Decker, Associate Professor, in the Department of History for giving me the opportunity to be a part of the project and guiding me through the design, development and testing of the system. I would like to thank the team at Genesee Country Village & Museum, especially Becky Wehle, the President and CEO, and Peter A. Wisbey, the Curator of Collections, for their support and contribution in deploying and testing the system at the Livingston-Backus house in the museum. I would like to thank Gary Jacobs, Assistant Professor at the School of Design at RIT, and Amanda Doherty, Adjunct Faculty at the College of Art and Design at RIT, for designing and modelling the character of Backus. RIT students, Brienna Johnson-Morris and Lizzy Carr, also played a crucial role in organizing the test for Hololens version of Backus at Frameless 2019. Finally, I would like to thank RIT student Koda Drake, for capturing and providing photographs of the onsite testing survey.
Abstract

Museums and cultural heritage institutions have used technology to create interactive exhibits and pedagogical tools that help spark visitors’ interests. The rise of Augmented, Virtual and Mixed Reality Systems has further enabled the creation of a new generation of immersive experiences that can engage and educate visitors. These technologies can be used to develop digital characters that can serve as virtual tour guides and improve user engagement by answering questions and forming social bonds with the users. While such tour guides have been deployed as exhibits at many museums, the implementation is usually limited to a single exhibit or a section of the museum space. We believe that visitors will be better served if the virtual guide not only enriches the onsite experience but also provides a take-home experience for users to encourage future visits. This thesis explores the enhancement in user experience that such a system can bring by offering onsite and offsite AR and WebVR technologies to create a virtual tour guide which assists visitors at the Genesee Country Village & Museum through interactive dialog as they explore the historic village on the museum campus.
# Contents

1 **Introduction** ................................................................. 6

2 **Background** ................................................................. 7
   2.1 Living History Museums ......................................................... 7
   2.2 GCV&M and the Livingston-Backus house ..................................... 7
   2.3 Technology in Museums ....................................................... 8
   2.4 AR, VR and MR .................................................................... 8
   2.5 AR, VR and MR in the Cultural Heritage Domain ......................... 9
   2.6 Computer Vision ............................................................... 10
   2.7 Related Work ..................................................................... 10

3 **Problem Statement and Hypothesis** .................................. 12
   3.1 Problem Statement ................................................................ 12
   3.2 Hypothesis .......................................................................... 12

4 **System Design and User Experience** .................................. 14
   4.1 System Design ..................................................................... 14
      4.1.1 Microsoft Hololens ......................................................... 14
      4.1.2 Smartphones .................................................................. 14
      4.1.3 WebXR ....................................................................... 15
      4.1.4 Back-end ..................................................................... 16
   4.2 User Experience and Interaction .......................................... 18
      4.2.1 Hololens ...................................................................... 18
      4.2.2 Mobile ........................................................................ 19
      4.2.3 WebXR/WebVR ............................................................ 20

5 **System Implementation** .................................................. 23
   5.1 Unity Engine ........................................................................ 23
      5.1.1 Design ......................................................................... 23
      5.1.2 Scripting ....................................................................... 23
      5.1.3 Editor UI ...................................................................... 24
   5.2 Hololens Development ......................................................... 25
      5.2.1 MRTK ......................................................................... 25
      5.2.2 Guide ......................................................................... 26
      5.2.3 Compiling and Deployment ............................................ 28
   5.3 Mobile .................................................................................. 28
      5.3.1 Login .......................................................................... 28
      5.3.2 Android version ........................................................... 29
      5.3.3 iOS version ................................................................... 30
   5.4 Web, WebXR/WebVR ........................................................... 31
      5.4.1 Login .......................................................................... 31
List of Figures

2.1 Livingston-Backus House .................................................. 8
2.2 The Reality-Virtuality Continuum by Milgram & Kishino [1] .... 9

4.1 System Design ................................................................. 15
4.2 Hololens Screenshot ......................................................... 16
4.3 ARKit (right) and ARCore (left) Screenshots ..................... 17
4.4 WebXR/Web version screenshots ....................................... 18
4.5 The Sequence for Login in Application .............................. 19
4.6 The Sequence for Speech Transcription in Mobile & WebXR .... 20
4.7 The Sequence for Object Scanning in Mobile & Hololens ..... 21
4.8 Hololens Air tap gesture .................................................. 21
4.9 Options for Backus in the Hololens version .................... 22
4.10 Mapped out environment in iOS & Android ..................... 22

5.1 Unity editor UI ................................................................. 24
5.2 MRTK Input Providers ..................................................... 25
5.3 MRTK Speech Commands ................................................. 26
5.4 The FSM for Backus’ animation transitions ....................... 27
5.5 Unity Hierarchy for Android (left) & iOS (right) ............... 29
5.6 User DB ER Diagram ....................................................... 35
5.7 Visualization of the Object Detection and Localization ....... 37

6.1 Trigger objects for scanning ............................................. 39
6.2 Users Interacting with the system at the Livingston-Backus house: Images by Koda Drake 40
6.3 Volunteers who had visited the Livingston-Backus house before 41
6.4 Volunteers who had interacted with costumed interpreters before 41
6.5 Comfort with technology in daily life ................................ 42
6.6 Comfort with technology in museums ................................ 42
6.7 Comfort with immersive technologies .............................. 43
6.8 Interaction with a virtual tour guide ................................. 43
6.9 Options unlocked by participants ..................................... 44
6.10 Subject of most interest to participants ............................ 44
6.11 Subject of least interest to participants ......................... 45
6.12 Experience was up to expectation .................................. 45
6.13 Enhanced interest in museum after the onsite experience ..... 46
6.14 Ease of use of the system .............................................. 46
6.15 Interest in expansion of the system ................................. 47
6.16 Platform of offsite experience ........................................ 47
6.17 Smoothness of the offsite experience .............................. 47
6.18 Additional value added by the experience ...................... 48
6.19 Preference for WebXR over mobile .................................. 48
List of Tables

6.1 Objects and Corresponding options .................................................. 46
Chapter 1

Introduction

Museums have been using interactive technology to spark visitor interest and improve understanding since the early 2000s [2] [3]. These technologies can play a crucial role in helping visitors understand more about an exhibit or site by using multimedia to provide information. The advances in Computer Graphics and 3-D rendering have particularly useful in enabling the creation of a new generation of pedagogical aids and immersive exhibits with the use of technologies such as Augmented Reality (AR), Virtual Reality (VR) and Mixed Reality (MR).

Virtual Reality is defined as an advanced human-computer interface that simulates a realistic environment [4]. An Augmented Reality system, on the other hand, superimposes or composites the real world with virtual objects rather than completely replacing reality [5]. Augmented Reality, is part of the Reality-Virtuality (RV) continuum, that was defined by Milgram & Kishino [1], which stretches from the Real World to the Virtual Environment. Mixed Reality is the class of displays/devices in which both real and virtual objects are displayed together [1]. Many technologies within the Reality-Virtuality (RV) [1] continuum have been used in the cultural heritage domain [6] [7]. MR Technologies have been particularly popular due to the ability to combine real world exhibits and sites with additional information, virtual reconstructions, or scenes etc. The progressive increase in computation power for small devices such as HMDs (Head-mounted Displays) and handhelds [8], has enabled development of applications that can provide an immersive experience while being untethered from a workstation or laptop computer. Furthermore, the ubiquity of smartphones, which contain all the sensors and computation power that is needed for running an immersive MR application, means that the user can access the experience on their own personal device and it itself can be tailored to the user’s preference, tastes and understanding. At the same time, advances in graphics and input technology have facilitated the creation of virtual characters that can be interacted with in various ways from natural language text [9] or voice [10] to gestures. The goal behind making such virtual assistants is to make computer interactions more seamless [11]. Such 3-D virtual characters have been found to be preferred by users in AR or non-AR scenarios [11].

Considering the positive impact that interaction with virtual characters can have on users, we have previously developed a digital docent: a 3D avatar presenting virtual and augmented reality, as a means for providing interactive storytelling experiences at the Genesee Country Village & Museum (GCV&M), a living history museum in Rochester, NY [12]. Living history museums often use costumed interpreters who convey daily life in a particular historic period [13]. Our work employs a similar model with a digitally rendered avatar based upon an actual individual. For this study we have improved and iterated upon our previous system to enable users to scan real world images to unlock more dialog options so that users are encouraged to explore various parts of the museum. It is believed that the user’s ability to interact with real world objects to continue the conversation will improve user engagement. At the same time, the offsite versions of the application will provide a way to extend a museum visit beyond the physical museum space forming a virtuous circle, similar to one described by Ailsa Barry [14], and encourage future visits for users. The system is multi-modal and deployed on three platforms: a Hololens version for onsite use, a mobile version that can be for onsite and offsite, and a web browser/VR version that can be for offsite interaction.

Users interact with our system either through their smartphones or through our HMDs. We analyze the users’ interaction with the system to determine the improvements in the increased interest in the content and improvements in learning as a result of the engagement. We believe that the immersiveness and interactivity of the application improves learning, retention and incentivizes future visits.
Chapter 2

Background

In this section we would like to shed some light on the technology and environments that are essential to our work. We will begin with a brief history behind Living History Museums and then look at the effectiveness of technology in museums, the technologies in the RV continuum and their use in cultural heritage, and Computer Vision.

2.1 Living History Museums

Living history museums have been defined as an attempt to simulate life in another time [15], usually this is the past. An attempt is made to live as other people once did. According to Jay Anderson [15], the three most common reasons for this are: to interpret material culture more effectively, usually at a living museum; to test an archaeological thesis or generate data for historical ethnographies; and to participate in an enjoyable recreational activity that is also a learning experience. In North America, one of the first living museums was established by John D. Rockefeller, Jr. when he agreed to provide the funds for the restoration, reconstruction, and refurnishing of Williamsburg, the colonial capital of Virginia. The museum had 500 structures and interpreters dressed in clothing appropriate to the Colonial period. Since then, more than 800 living museums have been established across the United States. They serve as an enjoyable recreation for families, enthusiasts, and researchers as well as learning experience. The main role of living history museums has been in democratizing historiography by focus not on the unusual but rather the everyday experiences of men, women and children [15]. Given the nature of living history museums, they can provide a unique challenge in terms of integrating learning technology as care must be taken to not replace the intended experience but rather augment it.

2.2 GCV&M and the Livingston-Backus house

Founded in 1966 by John L. Wehle, Genesee Country Village & Museum was created to help preserve the vanishing rural architecture of the Genesee Valley in New York State. The Historic Village in GCV&M depicts how a small town in the region might have changed over time, providing insight into the origins of current customs, traditions, and social values. The Livingston-Backus (Fig. 2.1), house is a part of the Historic Village at GCV&M. The house was built by the entrepreneur James Livingston in 1827. It was one of the first grand mansions in Rochester’s Third Ward. The house was sold by Livingston in 1835 and then purchased by Dr. Fredrick Backus in 1838. Dr. Backus was a prominent figure in civic and cultural affairs and an elected official when the City of Rochester was formed in 1834. Backus made substantial structural alterations to the house, employing Greek Revival elements and detailing. A one-story ell attached to the main block permitted the doubling of the parlor, while the entrance hall and stairway were shifted from the front to the side. Stylish decorative alterations were made on the interior. These changes, some very subtle, record an important phase in the history of the structure and were retained in its restoration.

Due to the historical and cultural significance of the house and Dr. Backus’s impact in Rochester’s formation, the house provides an ideal site that can be augmented with technology to capture a unique historical perspective that is educative and recreational at the same time.
2.3 Technology in Museums

Museums have been using media technologies since the early 60’s to enhance visitor experience. Acoustiguide [16] was one of the early systems which provided users with a portable tape playback machine containing a recorded description of the major works of art in the exhibition. In the early 90’s, as advances in computing gave rise to video games [17], these were used as educational tools in the museum space. In 1991, the Design Museum in London, developed and deployed a video game based on CAD software for young visitors to experiment with [18]. In 2002, Grinter et al. developed Sotto Voce, a electronic guidebook designed to encourage social interaction in the museum space for users [3]. Sotto Voce was a mobile system developed for enabling sharing of audio clips among users and its popularity among visitors underscored the importance of portability for museum technology. It showed that museums could use mobile devices to augment the entire museum space rather than develop a single compelling exhibit. These early successes in using technology to effectively convey cultural or historical information paved the way for better technologies that would become a deeply rooted part of the modern museum experience. Technology not only helps bring more visitors to the museum but also helps take the museum to millions more. Virtual museums and online collections have helped museums become accessible without the need of a physical onsite visit. Povroznik (2020) [19] has discussed the virtual history of museums in his paper.

2.4 AR, VR and MR

Augmented, Virtual and Mixed Reality technologies are part of the Reality-Virtuality (RV) Continuum which was first defined by Milgram & Kishino [1] (Figure 2.2) in 1995. The continuum stretches from entirely real world environments (Telepresence) at one end of the spectrum to entirely virtual environments (Virtual Reality) at the other end of the spectrum. Telepresence technologies present the real world as is to the user; the experience is expected to be constrained by the laws of physics and properties of the environment presented. On the other hand, Virtual Reality
(VR) technologies completely obscure the real world from the user, presenting them with an entirely virtual computer generated environment or imagery. The region between the extrema of the two ends of the continuum is defined as Mixed Reality (MR), where virtual world objects and the real world are presented as a single environment with varying degrees of overlap. While Milgram & Kishino [1] have defined multiple technologies that fall under the category of MR systems, the two most popular are Augmented Reality (AR) and Augmented Virtuality (AV). AR technologies augment the natural feedback or the natural world with simulated cues or virtual objects. These technologies can be used to add objects to the experience that blend with the real world, change the user’s perception of the real world through filters, or provide additional information about real world objects. Two most popular approaches of delivering an AR experience are “See-through” AR displays such as HMDs like the Hololens [20], Magic Leap [21] etc. or Monitor based AR displays which provide a “window-on-the-world” through a display system such as desktop monitor or mobile phone display [1]. technologies like Google’s ARCore and Apple’s ARKit have empowered millions of smartphones devices across the world to serve as monitor based AR displays. AV displays on the other hand, make it possible for virtual objects or environments to react to real-world cues such as hand gestures, physical objects or location data etc. Modern VR headsets, allow real-world hand actions to manipulate virtual objects thus enabling the creation of immersive AV experiences.

These technologies are fairly new in the consumer market, and have been rising in popularity in the last decade. But on the research front, immersive technologies have been under development since the 1960s. One of the first AR headsets was developed by Ivan Sutherland and his team at the University of Utah in 1968 [22]. The system, which is popularly known as The Sword of Damocles, was a head mounted display attached to a mechanical tracking system. The tracking system used a mechanical linkage attached to the HMD to track the position of the head and provide feedback to the display optical that worked using a miniature cathode ray tube.

This achievement lead to many innovations in the coming decades and finally in the 1990s, there was enough work for AR to qualify as a research field [23]. Ronald Azuma’s comprehensive Survey of Augmented Reality in 1997 [5] provides an expansive and brief description of the multiple applications, characteristics and limitations of the state of the art AR systems at the time. This was followed by a complementary survey in 2001, by Azuma et al. [23] that highlights the progress that was made in a short span of time. For those interested in a thorough and up to date review of the progress that has been made since 2001 in the AR domains of tracking technology, display technology, development tools, input and interaction technologies, design guidelines can refer to the comprehensive paper that was published by Bilinghurst et al. in 2015 [24].

2.5 AR, VR and MR in the Cultural Heritage Domain

Museums, historical sites and other institutions have extensively used immersive multi-media technologies to attract and engage visitors. The application of computing technologies to improving cultural heritage has been a robust field of inquiry [25]. In recent decades, mobile and desktop computers have enabled a wide range of applications in the cultural heritage domain. In this section we will look at the work that has happened with the heavy use of AR, VR and MR systems. A survey by Bekele et al. [6] in 2018, provides an overview of the various technologies and applications that have contributed significantly in this domain.

The Archeoguide system developed by Vlahakis et al. in 2001 used then state of the art visualization and mobile computing technology to provide Augmented Reality Tours to visitors [26]. It digitally reconstructed the archaeological site at Olympia in Greece and provided an AR interface which users can interact with while being in constant visual contact with the natural surroundings. This system had the advantage of allowing visitors and scientists to study and interact with the environment while respecting the sensitivity of the cultural heritage site by avoiding any
physical damage or disturbance to the site. The system was also built on a client-server model, with a site information server storing the audio-visual and textual information. The client devices were mobile units which provided the back-end system with user’s position and orientation information. Both the devices communicated through a wireless LAN. A major limitation of the system was its reliance on specialized hardware that had to be carried around in a heavy backpack. Users also expressed some concern with the availability and pricing of the system at the time [26].

Another system that used AR to project digital media onto a real world map was developed by Zöllner et al. in 2009. The system superimposed images of large-scale photos of archaeological sites in Rome with models of historical digital objects, seen through a static revolving AR displays or Ultra-mobile PCs [27]. The system was installed and tested at multiple locations including SIGGRAPH 2008. The major advantage of the system over comparable installations at the time was its cost effectiveness. Modern consumer VR headsets and the availability of easy to use development tools such as Unity3D has made it easier to developers to create such experiences as was demonstrated by Barsanti et al. [28]. The system was designed to provide users access to the Egyptian Funeral objects in Sforza Castle in Milan by making them accessible and interactive in a virtual VR experience. Ketchell et al. [29] successfully demonstrated the use of SLAM enabled mobile AR to develop a situated short story while Gunnar Liestøl [30] developed a mobile application that allows users to interact with a digital recreation of the landings at Omaha beach onsite. Another system, known as the Speaking Celt, uses mobile AR to guide users throughout the Museum of Celtic Heritage in Hallein, Austria [31]. The users scan a target which spawns 1 of 3 avatars that guide the user.

2.6 Computer Vision

Computer Vision (CV) deals with extracting useful information from images in a manner similar to the human visual system [32]. The challenge in CV stems from the complexity of visual data. An image depicting any common scene might contain hundreds of distinct objects that might be partially occluded or visible from a perspective where the entire shape or size of the object cannot be determined. For years, researchers and perceptual psychologists have been attempting to understand the working of the human visual system to develop algorithms that can perform similar perception and recognition tasks that humans seem to perform effortlessly [33]. One of the tasks that CV is widely useful for is in finding correspondences between two images of the same scene or object [34], which allows us to develop applications that can perform object detection among many other applications. These algorithms primarily function by finding interest points at distinct locations in each image, such as corners, blobs and T-junctions [34]. The interest points are then described using distinct descriptors and the descriptors from two images are matched in order to find objects between the images. For our system, we use CV to enable scanning of anchor objects in the museum space that serve as conversation topics with the virtual tour guide. For the detector and descriptor, we have used SURF (Speeded Up Robust Features) developed by Bay, et al. [34]. The detector is among the most popular ones and is widely used in the industry. The structure and working of our object detection system is explained in detail in the system design section.

2.7 Related Work

Inspired by the success of various technologies, attempts have been made to create virtual tour guides for cultural heritage sites. The benefits of a virtual tour guide over the use of media technologies such as video, audio or animations etc. is the ability to tailor the experience to the user’s personal preferences. By remembering user’s responses for future interaction, the character can form a bond with the users that make visits more engaging, informative and encourage future visits.

One such system, known as Tinker, was developed by Bickmore et al. and was installed in the Boston Museum of Science [35]. Tinker is an Embodied Conversational Agent (ECA) appearing in the form of a six-foot-tall animated robot projected on a screen, and communicates with visitors using synthetic speech and synchronized nonverbal behavior [35]. Tinker tracks interaction with users to build relationships through the use of hand geometry or names. Tinker was developed on the principles of incorporating principles from the social psychology of human personal relationships, such as using self-disclosure which is known to lead to increase in intimacy and trust in people. Tinker’s main purpose is to provide museum visitors with descriptions of and directions to museum exhibits, and to talk about her own implementation.
Another system for virtual museum guides, Ada and Grace, was developed by Swartout et al. [10], and also installed at the Museum of Science in Boston. Rather than using one character, The Ada and Grace system uses two life-sized, photo-realistic characters that interact with users in natural language, with gestures and other forms of non-verbal communication. To interact with the characters, an operator presses a push-to-talk button and speaks into the microphone. The users speech is parsed to determine to select a set of scripted responses best suited to the user’s query. The system then selects a response from the set based on recent utterances and the characters perform the response. Studies showed that interaction with the system had a positive impact on children of ages 7-14 [36].

These systems are powerful but rely on specialized hardware and large displays and are thus constrained to a single exhibit. But in a Living History Museum that is outdoors and spread out over a large area, constant visit to a fixed exhibit for information about various parts of the museum is counter-intuitive to the principle of rewarding exploration of the site. Taking this idea into consideration, our system is capable of providing onsite experience that is portable and is personalized on account of running on the user’s own device. The offsite experience is similar to the one provided by the systems mentioned above with the distinct advantage of being accessible from home where it can accompany users at any time as they study about the museum on their own.
Chapter 3

Problem Statement and Hypothesis

3.1 Problem Statement

Over the last few years, many museums have extensively expanded their online offerings in the form of virtual tours, videos and collections. Studies have found that online resources and experiences that museums provide are crucial in increasing user interest and disseminating knowledge about the areas that the museum focuses on [37] [38]. Experts such as Ailsa Barry have proposed the idea of a virtuous circle where the online and physical spheres are integrated so that the museum website, digital interfaces, kiosks allow users to take home bookmarks, links and media from their visits [14]. Petrelli et al. used this idea to develop a system which tracks a user’s visit and captures the experiences to create a tangible data souvenir [39]. The souvenir is then given to the user as a memento of their visit and gateway to further online resources.

At the same time, the virtual character systems that we discussed in the previous chapter have had a positive impact on user experience in museums they were deployed in. But limitations in hardware, software or system design have largely constrained these applications to the physical museum space, with limited research being done to make them accessible outside the museum.

Our current system consists of a virtual tour guide that visitors to the GCV&M can interact with using multiple portable devices: smartphones, laptops, HMDs and VR headsets with WebXR compatibility. One of the design goals for the system was providing users the flexibility of accessing the content on either the web, mobile or the Hololens from anywhere within or outside the museum [12]. The portability of the system ensures that a visitor does not have to return to a single site within the museum space for seeking assistance from the tour guide. We also believe that augmenting this system to provide conversational topics based on visitor’s scanning of the artifacts in the museum will increase user interest in the artifacts and their history. Furthermore, integrating Barry’s recommendations, we can convert the experience itself into a souvenir of their visit. While souvenirs are highly personal as they symbolize one’s own visit, an interactive conversation with the character can be shared with others as well. The interaction might help spread awareness and interest in the museum’s offerings. Given that the experience is served through an app or web, it can also upgraded to inform visitors about future events, new additions and new features or allow personalizing the character according to their interests.

3.2 Hypothesis

For this research, we have integrated an object detector with our existing Backus system, to allow visitors to scan real-world objects using the camera’s on their devices. The new system that we developed provides the following features to enhance visitor engagement in the museum:

• Provide visitors the flexibility to access the tour guide from a platform of their choice among the following: the Hololens, Android and iOS mobile devices, web, WebXR and WebVR compatible browsers.
• Integrate the virtual tour guide with the museum space through the use of trigger objects distributed around the museum that can be scanned and used to unlock new conversation options for the tour guide.

• Allow seamless transition between both onsite (mobile and Hololens) and offsite (web, WebXR and WebVR) experiences by tracking user interactions and incentivizing repeated visits to the museum.

We evaluated the effectiveness of our system in two use cases: the onsite and the offsite. For the onsite component of the study, volunteers were asked to scan predetermined objects in the Livingston-Backus house at GCV&M to unlock dialog options for Backus, the virtual tour guide, regarding the history of the object. Users can communicate with Backus either via voice or touch (selecting the dialog options displayed). It was hypothesized that a conversation with Backus would increase visitor interest in the history behind the object and the house.

For the second component of the study, we asked volunteers to use the web version of Backus within 48 hrs of the onsite experience. This was intended to happen offsite and volunteers were allowed and even encouraged to share the experience with those who did not experience the online section. The conversation options for the web version were based on the objects that the user scans during the onsite component of the study. This was meant to evaluate four things:

1. Does the ease of use provided by the portability of the system lead to increased engagement?
2. Does the object scanning feature invoke interest in the artifacts that are part of the museum?
3. Does the offsite experience complement the onsite experience to maintain interest in the museum and encourage a deeper dive into the museum’s history?
4. Does sharing the experience with those who have not used the onsite component, spark their interest in the house, the museum or the person, Dr. Fredrick Backus?

After collecting data gathered from surveys completed by the users after the two components, we believe that we will be able to answer the following research questions:

• Does a portable virtual tour guide that rewards exploration of the site, invoke an increased interest about the site’s history?

• Does the ability to share the experience with non-visitors improve awareness about the museum and its setting?
Chapter 4

System Design and User Experience

In this section, the design of our system and the components is explained in detail. The system is comprised of multiple front-end applications on various platforms and a back-end system that performs image recognition and transcribes user speech to text (Fig. 4.1). The front-end applications run on four platforms: Microsoft Hololens, Google Android powered mobile devices with ARCore support, Apple iOS mobile devices with ARKit support and a WebXR version that supports VR and browsers. These four platforms were selected due to their extensive support for augmented reality and accessibility to users. The back-end system uses two servers running on Google’s Cloud Compute service [40], one for Image Recognition using OpenCV and the other one using Google’s Cloud Speech-to-Text [41] for audio transcription. In the following sections, the design of the system and the functionality of each individual component is described in detail. This is followed with details of the user interaction and intended user experience.

4.1 System Design

4.1.1 Microsoft Hololens

Hololens is a head mounted holographic computer developed by Microsoft [20]. The device runs on the Windows Mixed Reality Platform and uses the Windows 10 operating system. Hololens does not need any external hardware for working as all the components: the sensors for tracking the user and the environment, the processor for performing any computations, and the display are all integrated into the headset itself. While Hololens 2 is currently available, we have only tested this application with Hololens 1. Though we believe that the common API will make the transition to Hololens 2 very smooth. The Hololens version of Backus (Fig. 4.2) is deployed on the device as a UWP app. The app is pre-loaded with all the interactions that users can have with Backus. The voice input component of Backus does not require any interaction with the server as Hololens’s built in Dictation provider [42] handles the transcription and keyword matching to select the correct dialog option for the user.

For the object detection feature, the application uses the Hololens webcam to capture an image from the users perspective and upload it to our object detection back-end server for detecting the appropriate object present in the captured image. If a trigger object is detected then the detection result is sent back to the user with the result of the detection so that the application can unlock the correct dialog option. Repeated detection of the same trigger object is ignored.

4.1.2 Smartphones

Smartphones are a ubiquitous computing platform equipped with the sensors that are needed to provide users with an immersive AR experience. The tracking, localization and movement sensors used in modern smartphones can be configured to provide camera transform estimation, environment detection and mapping data which can then be used to align virtual objects with the real world [43]. This provides a cost effective way to make AR accessible to a wide user base.
For this project, we support Apple’s iOS and Google’s Android smartphone operating systems because of their extensive market share (98% of all smartphones in 2021 [44]). Both Apple and Google support the development of AR apps through their software development kits, Apple’s ARKit and Google’s ARCore. The APIs that are part of these kits ease development of AR apps by providing an abstraction over the tracking and localization mentioned earlier.

The app’s design on both platforms is nearly identical (Fig. 4.3). Users are first required to sign up or login using alphanumeric codes selected by them, which are stored in a database in the object detection back-end server. These codes help track a user’s activity across platforms and onsite visits and once logged in, users can place Backus at any location in their environment to start the interaction. In order to place Backus in an appropriate location, the app scans the user’s environment to detect flat horizontal planes using the phone’s camera. Once placed, users can communicate with Backus via voice by tapping the "Press To Speak" button which turns on the microphone to record a 5 second audio clip of the user’s speech. The recorded audio is uploaded to the transcription server using an asynchronous WebSocket [45] connection to convert the result into text which is returned to the application. The text is then parsed to determine the topic it matches to find the appropriate response for Backus. An alternative to voice input is to press the button related to the topic that appear next to Backus when he is not speaking. The animations and audio for the conversations are already loaded in the app.

For the object scanning feature, users press the “Scan Object” button to capture an image from their smartphones camera. The image is uploaded to the object detection server through another asynchronous WebSocket connection and the results of the detection are returned to the app to unlock the dialog based on the topic. Thus, through the use of the two servers, the mobile apps provide the users with all the functionality that is required to ensure a smooth and immersive experience.

4.1.3 WebXR

The Hololens and smartphones are great platforms accessible to many users [44], but reliance on device specific APIs requires adherence to the platform’s restrictions and not every version of the OS might support the required APIs. On
the other hand, modern web browsers provide a much accessible and robust runtime platform for complex applications by abstracting away compatibility with native APIs. The development of standards such as WebGL, WebXR, and upcoming WebGPU seem to indicate a trend towards growing support for 3-D graphics and immersive applications in the browser. This would enable a single web application to run on smartphones, desktop PCs and HMDs that support compatible browsers. Taking this into consideration, we have developed a web version of Backus that uses the WebXR/WebVR API to provide users a web or VR based browser experience. This version is meant to serve as an offsite application for users who wish to share their experience with others or simply would like to engage in a VR version instead of the onsite AR version.

When executed, this version of Backus also requires the user to log in through an alpha-numeric code of their choice that was used in the onsite version. Once logged in, users are placed in a virtual 360° view of the interior of the Livingston-Backus house with Backus as he introduces himself. Similar to the other versions, users can converse with Backus through the microphone on their devices. The major advantage of this version is that the interaction does not require a button press for voice input. The system has been configured to detect the user’s voice and start recording once they start speaking. When the user stops speaking, the recording stops and the audio clip is uploaded to the transcription server using an Asynchronous WebSocket connection. The transcribed result text is sent back to the application for parsing.

The VR component enables the experience to be fully-immersive [6]. For accessing the VR component in the web version, the page must be opened in a browser that supports WebVR or WebXR at which point the user is shown an option to switch to VR mode. The VR mode can be experienced in smartphones through a VR viewer such as Google Cardboard [46].

Unlike the onsite versions of Backus, this version does not feature the ability to scan objects and neither provides users with new dialog options as it is intended to be used for exploring the experiences the user has already unlocked at the house. Also, since it is a web based application, it does not require any installation on the device and can be easily accessed through the URL.

### 4.1.4 Back-end

The back-end component of our system is the most component as it provides many essential services for multiple user facing components. The Back-end system is driven by 2 instances of Google’s Compute Engine cloud service. As mentioned previously, the two servers are the transcription server and the object detection server. Here we will give a brief overview of the two servers.
Transcription Server

The transcription server uses Google's Cloud Speech-To-Text service to process audio clips into its text transcription. The audio data can be uploaded to the server through a WebSocket connection. The WebSocket API [45] was used because it allows a two-way interactive communication session between the client and the server. The client can send messages to the server and receive event-driven responses without polling the server. The primary reason for using a server for transcription in contrast to integrating transcription into the on board application was to have a common transcription system, avoid security issues due to exposure of transcription API keys, and enable seamless switching of transcription libraries in the future if required.

Object Detection Server

The object scanning component of our system is core to the user engagement aspect. Thus, both accuracy and speed of the detection are essential for optimal user experience. For this, we have used SURF (Speeded Up Robust Features) [47] as the feature detector and descriptor which is part of OpenCV Computer Vision libraries [48]. The server performs two inter-dependent tasks: 1) It processes the images taken by a user and detects the relevant object if present in the scene, 2) Maintain a relevant database of all the objects processed by the user so that the unlocked options can be tracked across all user sessions on all platforms. For the processing, the main server uses a WebSocket connection to receive image data from the user apps and pass it to a child server which uses OpenCV to process the image. The results returned by the child server are added to the database if they haven’t been already and the result is then sent back to the parent. The user database stores a user’s code, and their unlocked options. The code is selected by the user when they sign up in any platform. The database is accessed by the applications using HTTP POST [49] requests which allows the app to add a user or verify their existence, check a user’s unlocked options or add new options to the database.
4.2 User Experience and Interaction

After providing an overview of the system’s design, this section elaborates on the user interaction and experience. By supporting multiple input devices, accessibility options, and platforms etc. we have attempted to accommodate every individual regardless of age, familiarity with technology, and disability. This section explains the user interaction process for each platform.

4.2.1 Hololens

When started, the app places Backus in front of the user and Backus will start with the introduction. Users can then interact with Backus using gesture or voice input. The gesture for input that we support is the Air tap gesture supported by the Hololens. The Air tap is an inbuilt supported gesture performed by the following actions (shown in figure 4.8):

1. Hold your hand straight out in front of you in a loose fist, and point your index finger straight up toward the ceiling. You don’t have to raise your whole arm. Keep your elbow low and comfortable.

2. Tap your finger down and then quickly raise it up again.

While interacting, at any time users can interrupt Backus by saying “Hold On”. Backus will stop talking and allow the user to select a new topic. To select a topic to discuss, the user can either read aloud an option (shown in Figure 4.9) from the ones shown next to them or perform the Air-Tap gesture upon which Backus will provide information about the topic. In the museum space, if users find a trigger object, they can simply perform an air tap while gazing at it and the Hololens will capture an image to perform the object detection. If a new trigger object is detected, the relevant
dialog option for Backus will be unlocked. Thus, interaction with Backus on the Hololens is simple and intuitive to make the conversation natural and engaging.

### 4.2.2 Mobile

For the mobile version, when users start the app, they are asked to login or generate an access code to track their unlocked options. Once entered, it is validated and stored on the device so that it is not needed for future sessions on that device. Users then start the AR experience. First, users have to scan the surrounding environment to map out all the horizontal surfaces. While mapping out, a mesh is created on the plane which is used to place the character (Figure 4.10). Users can tap on any visible part of the mesh and Backus is then placed on the mesh.

Once placed, Backus will start with the introduction of the house. At any point users can press the “Press To Speak” button to speak to Backus or interrupt him by saying “Hold On”. Upon a single press, only 5 seconds of audio can be recorded. Once interrupted, Backus will stop speaking and allow user to ask about any other topic. At any point users can use the “Scan Object” button to scan an object and capture a photo of the object they are looking at. The image is processed and if a new trigger object is detected then a dialog option is unlocked for user to ask Backus about it. At any point, users can log out of the experience and back to the login/sign-up screen by pressing the logout button. If users close the app without logging out, future sessions will automatically log them in to the experience using their last used code. The mobile experience is consistent across both mobile platforms: Android and iOS.
4.2.3 WebXR/WebVR

The web version is a hybrid experience that allows users to switch between the browser and VR on compatible devices and browsers. Users start the experience by opening the URL for the application in a compatible browser. Once opened, users are first asked to login to the experience using the same code that was used in the AR experience. Because the web version does not support scanning for options, users are only able to use the options that they unlocked in the AR version. The experience takes place in a 360° view of the interior of the Backus house with Backus standing in the center. Users can interrupt or communicate with Backus using the same voice commands as the other versions or by using the buttons that appear next to Backus. One advantage of the web version is the seamless integration of transcription as users can ask questions without a button press.
Figure 4.7: The Sequence for Object Scanning in Mobile & Hololens

Figure 4.8: Hololens Air tap gesture
Figure 4.9: Options for Backus in the Hololens version

Figure 4.10: Mapped out environment in iOS & Android
Chapter 5

System Implementation

We have described the design of our system and its various parts in the previous chapter, here the implementation of the components is detailed. For this project, we have used a combination of open-source and proprietary development tools, libraries, frameworks etc. Adherence to widely-used software engineering practices has been followed to ensure that the applications are maintainable and scalable in the future. We will start this chapter by providing a background of the Unity Engine. This will be followed by the various libraries and run-times that were used with Unity to develop and deploy the application on various platforms. After that, the implementation details of both servers on the back-end system are provided.

5.1 Unity Engine

Unity is a game engine developed by Unity Technologies and first released in 2005 [50]. The engine was initially targeted at hobbyists and independent game developers but over the years has become the development platform of choice for many studios that develop games for mobile and the browser [50]. Due to its popularity and support, Unity was used as the primary development environment for the system. This section provides a brief introduction to the essential components of the Engine.

5.1.1 Design

Unity follows a fairly intuitive design pattern where every component of the game/application is represented by the same base object known as a "GameObject" [50]. Every GameObject can have multiple components attached to it in order to perform it's intended functionality. For instance, a camera in Unity is a GameObject with the Camera component attached to it. The Camera component is then configured to adjust the camera properties. GameObjects can be hierarchical entities, so any GameObject could have multiple children attached to it. Transformations applied to the parent object are further applied to every child, so at any given time only the parent needs to be controlled for movement. Even UI elements on the overlay in a game or part of the game world are represented as GameObjects with the necessary text formatting, shading and mesh components. Relevant components are also used to attach shaders, materials, and physics properties, to GameObjects.

GameObjects can then be placed in the 3-D world space and each object also has it’s own local space which is referenced by it’s children and components. A collection of different GameObjects might be needed at different stages of an application, so Unity provides a Scene system as well. Every Scene is fundamentally a collection of objects and every application in Unity requires at least one scene.

5.1.2 Scripting

Users can easily develop their own GameObject components through the use of Unity’s comprehensive scripting support [50]. A script can provide various types of input fields for the editor to link other GameObjects or components, provide alphanumerical input, sliders, buttons etc. Scripts are written in C# [51] with every script intended to be a C#
class that extends Unity’s MonoBehaviour class. MonoBehaviour provides functions that can be used to perform any operation in synchronisation with the application’s lifecycle or frame updates. Every frame update call of the engine is passed to every active GameObject’s script and the relevant methods if found are executed. To ensure the appearance of smoothness, update calls differ in order for the game updates and the physics engine updates. Scripts can be used to change or modify scenes, affect the state of the application and add, modify or remove GameObjects from a scene. The use of C# also allows Unity to take advantage of a wide array of pre-existing Unity libraries that provide support for different functionalities.

5.1.3 Editor UI

Unity’s Editor UI provides 5 primary views for developers that are needed while developing an application (Figure 5.1).

1. The **Scene View** shows the placement of all the objects in a scene in relation to the world space. Objects can be transformed in any way before running the program.

2. The **Scene Hierarchy** shows the user all the objects that are placed in the Scene even if they are not visible or active.

3. The **Project View** shows all the directories in the application that contain the assets and other files.

4. The **Inspector** provides a selected GameObjects details with all the components attached to it.

5. The **Game View** provides a preview of how the application would look like from the camera’s perspective.

These UI components are easily understandable, customizable, and extendable for maximum flexibility.
5.2 Hololens Development

The Hololens version of Backus was developed using Mixed Reality Toolkit (MRTK). This section describes the various GameObjects, components and libraries that were essential to its development.

5.2.1 MRTK

MRTK is a Microsoft-driven project that provides a set of components and features [54], used to accelerate cross-platform MR app development in Unity. It can be easily integrated in any Unity scene through a simple import process and provides components, input providers etc. which form the building blocks for spatial interactions and UI. MRTK can be imported into Unity as a package and once imported it becomes available as a Unity menu which can be used to configure it and add it to the scene.

Once MRTK has been added to the scene, it shows up as two GameObjects: the MixedRealityToolkit (toolkit) and the MixedRealityPlayspace (playspace). The toolkit object contains the MRTK component which is used to configure the application. All aspects of the immersive experience can be configured using this component. For most scenarios users can work with the default configurations while custom configurations are also easy to perform. For our use, we have only edited two of the available configurations: the input and the spatial awareness aspect. The input configuration provided by MRTK supports various input types such as pointer devices, gestures, hand tracking, and gestures etc. Each type of input processes data provided by an input provider (Figure 5.2) that is available on the device. The providers can be changed independently from the input data processing and this provides flexibility to users for any device or platform. For the Hololens 1 we have not changed any provider. Even the input processing configuration was the default one for all inputs except the speech.

The speech input configuration allows users to configure the MRTK to trigger various methods on a provided GameObject when a specific keyword or key phrase is uttered. We added all the key phrases (Figure 5.3) that Backus could listen to for this setting and thus we were able to program Backus to respond to those key-phrases from the user.
The Playspace GameObject contains the camera for the AR experience. The camera is a child of the Playspace. While based upon a default Unity camera, this object has additional components that allow the camera in the application to be controlled and transformed by the gaze and tracking data from the Hololens device. This ensures that the camera in the scene moves similarly to the user’s head in the real world and thus provide a user the perception of blending of the virtual and physical space.

5.2.2 Guide

The “Guide” GameObject contains all the code necessary for Backus. This object contains the components that manage the animations, interactions, and server communication. Here we will explain the components and their functionality.

Audio Source

The audio source component is responsible for playing the audio during Backus’ conversations. The Audio Source component in Unity is essential for any GameObject to play audio clips. There are various configurations to modify the audio output but we have used the default configurations.
Animation Controller

The Animation Controller script handles all configurations related to Backus’ animations and the other UI components that are affected by the animation. The component is linked to every audio file that Backus can speak and the related animation. The caption data is provided to this component as well to manage the captions during animation. The captions are written in JSON \[55\] instead of standard SRT subtitle format to make it easier working with them using default library functions.

While the script manages the working of the audio and the captions, the model animations are managed by a Controller object that is part of the component. The Controller represents a finite state machine \[56\] (Figure 5.4) which plays the ideal animation that Backus should be playing next. The StartState is the initial state for the system. After an animation is triggered, the corresponding state is entered to play the animation. Once completed, the FSM enters the idle state, where an ideal animation for Backus is looped until the next interaction is triggered by the user. The synchronisation of the animation, audio and captions provide the engaging conversation experience for the guide. Once an animation is completed the component handles the creation of the buttons that appear next to Backus for the user to select the next option through a gesture if they so desire.

![Figure 5.4: The FSM for Backus’ animation transitions](image)

Speech Input Handler

The Speech Input Handler component is a part of MRTK that handles the result of MRTK’s speech recognizer. Here the GameObject component methods that must be triggered are linked to each option along with the required parameters. We use the Handle Speech Commands component to handle the correct animation and then pass it to the animator.
Handle Speech Commands

The Handle Speech Commands component runs the appropriate animation through the Animation Controller component but performs any necessary pre-processing if required. It also saves the last conversation option selected in case the user asks for the dialog to be repeated.

Object Detection

The Object Detection Script performs the process of initiating image capture, connecting to the back-end system and then sending the data to the back-end for processing and handling the result of the capture.

As soon as the application starts, a WebSocket connection is established with the back-end server through the use of WebSocketSharp [57], a .NET library for WebSockets in C#. Listeners are setup for handling messages or errors from the WebSocket. During runtime, the Gesture Handler component calls this scripts camera capture method. The callback for the camera result is recorded and the Stream Script component script is invoked. The resulting texture from that component is recorded and processed. Processing involves converting the image to grayscale and mirroring it due to the camera’s mirrored output. The image byte data is then uploaded to the server. The response from the server is received in the handler and then received to invoke the callback that was recorded while initiating photo capture (Figure 4.7).

Gesture Handler

The Gesture Handler component listens for the various gestures which are detected by MRTK and then responds to them by initiating the image capture process for object detection. When an object is detected, the handler adds the new option to the list of available options.

Stream

The Stream component processes the media stream from the Hololens Webcam to capture a photo of the user’s gazed object in order to process it for detection. It uses Windows’ inbuilt Webcam APIs to capture the stream.

5.2.3 Compiling and Deployment

For deploying a Unity application to Hololens, Unity is configured to build a UWP or Universal Windows Platform build [58]. The build is then executed with Visual Studio to be deployed to a connected Hololens device. While this process is used for a development app, the app can also be published on the Window’s store in order to provide it to other users.

5.3 Mobile

In this section, the implementation details of the mobile versions of the system are provided. We have developed the mobile version of Backus for two platforms, Android and iOS. While the functionality and design of the applications is similar, there are a few key differences in implementation on the two platforms which will be highlighted. The first part of the section covers the Login screen, which functions similarly in both the version, and then the implementations of the two versions are described separately.

5.3.1 Login

The login code provided by the user is used to track the conversation topics that they unlock for Backus through their onsite visit. Three essential steps are performed on this screen: verification of user if logging in, creation of a new user and getting the unlocked options for use in the AR experience. All the operations are carried out through HTTP POST requests that update the database on the back-end and fetch the required details. Once login is successfully completed, the details are stored in Application Preferences, a persistent key-value pair storage for the application,
and used in subsequent sessions to automatically log the user in the app. User’s can logout to wipe the preferences and use a different login.

5.3.2 Android version

The Android version of the system uses ARCore SDK for Unity. ARCore SDK [59] is a standalone development kit that enables access to ARCore’s features on a compatible Android device. The SDK is independent of Unity and must be imported through a package. Default ARCore provides a demo activity that can be used to place a sample GameObject on a mesh that is placed in planes detected in the user’s environment. The objects are placed via tap on the screen at a visible point on the mesh. We have used this template to place Backus on the mesh instead of a sample. The following components are used to provide the essential AR functionality of control and interaction.

![Figure 5.5: Unity Hierarchy for Android (left) & iOS (right)](image)

**ARCore Device**

The ARCore Device GameObject is one of the inbuilt GameObjects provided by ARCore (Shown in the hierarchy in Figure 5.5). This object replaces the default camera by providing a trackable version that is controlled by the device’s sensors and overlays the scene’s GameObjects on a real-world camera feed. The Canvas with the UI for interaction is also attached to the camera as a child object to ensure that it moves along with the camera and the UI is also in user’s sight. The three primary UI buttons are implemented using Unity’s UI system and are linked to GameObjects that carry out the intended functionality.

**AR Controller**

The AR Controller component handles the instantiation of the guide when the user clicks on a mapped out mesh area. The controller performs a raycast from the 2-D point on the screen to the 3-D world space and when the first instance of a mesh is detected, the rotation and transformation of the placement is calculated and the guide is placed.
**Plane Generator, Plane Discovery, Depth Menu**

These 3 components are inbuilt AR Core components that discover a mesh through depth calculation and generate the Plane mesh. The components guide the user via UI hints on properly scanning the environment in case the mesh is not generated. Users can turn off the depth calculation for placement. This will result in placed objects not being obscured by the environment.

**Guide**

The guide object is similar to the one used in Hololens. The MRTK specific components have been removed. Only the Animation Controller and Handle Speech components from the MRTK version are present as the speech input is now done by a different component connected to the server.

**Speech Manager**

The Speech Manager handles the uploading of voice data to the server and handling the response. At the start of the scene, the component establishes a WebSocket connection with the server for the voice data communication. The user’s microphone permissions are also checked, and requested if required. Once the user presses the record button, the component records a 5 second audio file and saves it using a SavWav open source script. The file is then read from another thread and uploaded to the server, after which it is deleted. Once the transcribed text is received it is forwarded to the Handle Speech component of the Guide GameObject.

**Camera Manager**

The Camera Manager initiates a WebSocket connection to the Object Detection server to upload the image data captured from the camera, and receive the processed response. The process response is then directly passed on to the Speech Handler for adding the new option to the user’s existing ones. The camera capture is invoked when the scan object button is pressed.

**Game Manager**

The Game Manager component is invoked by the logout button for initiating the logout process for the user. The user preferences are deleted to prevent automatic future logins and enable logging in with new credentials.

**UnityThread Class**

The UnityThread class is a third-party open source plugin script that simplifies code execution from other threads on the Unity main thread. Unity does not allow access to texture or other GameObject data from the main thread. Thus, in order to perform the texture creation after processing the pixel data, the UnityThread class is used to carry out the execution on the main thread.

**Build and Deploy**

For the development and deployment of Android apps, Unity requires the installation of Android SDK, NDK, and build tools. This enables Unity to export the developed app as an APK file ready for deployment on a mobile device or as a project for integration with other Android codebase. There is no requirement for external development or build tools like Hololens.

**5.3.3 iOS version**

For iOS, the ARKit runtime is required for AR applications. We have used the AR Foundations pipeline instead of working directly with ARKit API. The Speech Manager, Camera Manager, and Game Manager GameObjects are common to both the iOS and Android version. The key difference between the two versions are the APIs used for developing the AR experience. Unity’s AR Foundation pipeline supports multi-platform development with a
common code base. Code reuse from the Android version is made possible due to ARCore extension’s library for the new pipeline. Here the core components that are unique to this version are described.

**AR Session Origin**

This GameObject is the parent for the Camera and also contains the components that use the camera data to perform plane and point cloud visualization for the user’s environment.

**AR Session**

This GameObject manages the frame rate and tracking for the AR session and also provides the input handlers that are necessary to capture user data.

**ARCore Extensions**

The ARCore Extensions GameObject references the other GameObjects which manage the operations that provide an abstraction similar to ARCore in Android.

The two mobile versions therefore are nearly similar in their implementation except the few differences highlighted here.

**Deployment**

Unlike the Android version, Unity cannot build the iOS app file (.ipa) as the process requires Apple developer credentials and XCode, Apple’s proprietary IDE available on macOS devices. Unity exports a project that can then be opened in Apple’s XCode which is used to build and deploy the app to the device or .ipa file. The app can then be exported for distribution or uploaded to an App Store Connect account where we deploy it for testing through TestFlight, Apple’s proprietary beta testing platform for app developers.

### 5.4 Web, WebXR/WebVR

For the Web version of Tour Guide system, we have used Mozilla’s Unity WebXR Exporter [63] to export our Unity project to a WebXR/WebVR compatible experience. The Exporter exports the application to run in a WebXR Polyfill. The Polyfill is an implementation of the WebXR Device API and WebXR Game pad. A major advantage with the Polyfill is that it falls back to the now deprecated WebVR protocol if support for WebXR is not available in a browser and when neither APIs are supported, a standard web application is provided. Thus, the use of the Exporter makes it possible to run a single version to handle compatibility with most available browsers. The exported base application is combined with native JavaScript code for recording user audio and transcribing it with the backend. In this section we will first describe the implementation of the Unity application and then describe the JavaScript code that is added to the application. Finally we will describe the details of deploying and hosting the application.

**5.4.1 Login**

The login of the user to determine the available dialog options is implemented similar to other versions. The key difference is that since this version does not support object scanning, new options cannot be unlocked from within the web version.

**5.4.2 Building in Unity**

The Mozilla WebXR Exporter converts a Unity VR or standard application to WebXR by replacing the Camera GameObject. This is done by providing three cameras under a single parent, 2 for the left and right eye when in VR view and one for the normal browser view. The activation and switching of the correct cameras is handled by the WebXRCameraSet component as it determines the compatibility of the device running the application.
Web XR Camera Set

This object is an inbuilt component of the Exporter provided by Mozilla. The Camera Set is a parent that contains the three different cameras. The camera tracking and movement is handled by the Manager component in the Camera Set object. Different UI canvasses are added to the different view cameras and switched between depending on the mode. The canvasses contain the caption and transcription views.

House

The House GameObject is a simple 360° model of the interior of the entrance hallway of the Backus house. It was created by texture mapping a 360° photo of the house on the interior of a cylinder. The interior mapping was done by using a simple GLSL texture mapping shader [64]. Backus is placed at the center of the room and users can look around to see the house.

Guide

The Guide object in this version is similar to the one used in other versions with minor differences. The key difference is unlike the previous versions, the transcription is not handled by the Unity Player and hence the method for handling transcription results in Handle Speech Commands component is invoked from the browser JavaScript. In order to start recording the audio, a call to the recorder method in the browser is sent using a .jslib [65] plugin. A .jslib plugin is JavaScript code that is part of the Unity application and is part of the application’s package.

Deployment

The deployment from Unity to a web application is performed by Unity’s WebGL builder. The WebGL builder exports the application as a single Polyfill Unity Player. This is done by using Emscripten SDK [66] to compile the application to WebAssembly [67], a portable binary-code format for executable programs, which can run on browsers among other platforms.

5.4.3 Integrating Voice Transcription in WebXR

To integrate voice transcription into the exported web build, we use three open source JavaScript libraries that perform the speech detection, audio capture, and transmission. This section describes these libraries.

WebRTC and RecordRTC

WebRTC (Web Real-Time Communication) is a technology which enables Web applications and sites to capture and optionally stream audio and/or video media, as well as to exchange arbitrary data between browsers without requiring an intermediary [68]. The set of standards that comprise WebRTC makes it possible to share data and perform teleconferencing peer-to-peer, without requiring that the user install plug-ins or any other third-party software [68].

RecordRTC is an open source WebRTC JavaScript library for audio/video as well as screen activity recording [69]. The library supports recording in multiple browsers. We use the library’s StereoAudioRecorder to record audio in wav format. The audio is recorded at 16000hz and sent to the server as a blob file.

Hark

Hark is a tiny open source browser/JS module that listens to an audio stream, and emits events indicating whether the user is speaking or not [70]. Hark uses the WebAudio API to get the power of the audio using FFT (Fast Fourier Transform) in the audio stream. If the power is above a threshold, it’s determined to be speech. Our application uses Hark to indicate when the user starts and stops speaking. The RecordRTC recorder starts recording when the speaking starts and stops and uploads the blob to the server after the user stops speaking. Thus, Hark makes it possible to provide a natural conversational experience.
Socket.IO

Socket.IO is a library that enables real-time, bidirectional and event-based communication between the browser and the server [21]. It consists of both server and client implementations. The client will try to establish a WebSocket connection if possible, and will fall back on HTTP long polling if not. Although Socket.IO indeed uses WebSocket as a transport when possible, it adds additional metadata to each packet. That is why a WebSocket client will not be able to successfully connect to a Socket.IO server, and a Socket.IO client will not be able to connect to a plain WebSocket server either. Thus for transcription for the web application, we use Socket.IO on the back-end and front-end. Socket.IO listens for messages from the server and then sends them to the Unity application.

5.4.4 Deployment

The three libraries mentioned above are added to the index.html file of the application using methods which are called by the .jslib libraries created in Unity. Once the libraries are integrated, the application is uploaded to our back-end Node.js server where it is hosted using Express.js.

5.5 Back-end

This section describes the Back-end of our system. As mentioned in the previous chapter, the Back-end system is comprised of the audio transcription server and the object detection server. Both server’s are hosted on Google Cloud’s Compute Engine service. The virtual machines run on Debian Linux and utilize various open source framework and services to provide maximum reliability, scalability, and extensibility. The first subsection describes the audio transcription server which is followed by a description of the image recognition server.

5.5.1 Audio transcription server

The audio transcription server runs on Node.js, using Express.js for it’s functionality. This subsection explains how the system uses these technologies to provide concurrency, reliability and scalability.

Node.js and Express.js

Node.js is an asynchronous event-driven JavaScript runtime designed to build scalable network applications [22]. Each connection to a Node.js instance fires a callback but if there is no work to be done, the process will sleep and conserve resources. It presents an event loop as a runtime construct instead of as a library. Node.js enters the event loop after executing the input script and exits the event loop when there are no more callbacks to perform. Due to these advantages Node.js has been a preferred back-end platform for more than a decade. We manage the packages and version of our Node.js instance through Node’s supported package manager NPM. We have also used PM2, a process manager that ensures reliable working of Node.js instances and logs and attempts to handle any failures or errors.

Express.js [72] is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It provides mechanisms to: Write handlers for requests with different HTTP verbs at different URL paths (routes), integrate with "view" rendering engines in order to generate responses by inserting data into templates, set common web application settings like the port to use for connecting, and the location of templates that are used for rendering the response, Add additional request processing "middleware" at any point within the request handling pipeline.

We instantiate our Node.js runtime by importing and initializing the necessary libraries, including Express.js. Once initialized, the Backus web version is added as a static webpage to express. This redirects the user to Backus when they open the URL on their browser. The connection to Backus is secured using HTTPS in order to authorize it to request user microphone permission. An HTTPS Server is instantiated to listen at the decided port. The next step is to configure Socket.IO to listen on the same HTTPS Server. All the Socket.io requests to the url will be redirected to the Socket.IO listener. The first HTTPS server is the executed. After this we execute another HTTPS server on the same Node.js runtime for handling the mobile transcription requests. The separate HTTPS server instance on a different port
is used to redirect to a WebSocket connection instead of the Socket.IO connection. The second HTTPS Server is also executed then.

**Socket.IO**

The Socket.IO server is configured to listen on the HTTPS Server. Upon receiving a Socket.IO connection, the socket callback is executed. Upon receiving a message, the data is processed into a base64 string and added to a JSON object along with its configuration. The JSON is then forwarded to Google Cloud’s Speech-To-Text service and the response is sent back through the same socket to the client. This process is performed independently for every connection on both HTTPS servers with the only exception being that the second server uses a default WebSocket connection for mobiles.

**Google Speech-To-Text**

Google Cloud’s Speech-To-Text service is a proprietary API provided by Google. It supports transcription for multiple languages and provides reliable and robust results. We use Google’s client API to send our data to their server and receive a text transcription. The data is received from a client as a LINEAR16 encoded audio at 16000Hz and converted into a base64 string. The string is then passed to the service along with key phrases that might be present. The phrases include names of the topics that can be discussed with Backus. This provides an overview into the working of the Audio transcription server.

**5.5.2 Object Detection Server**

The Object Detection Server processes the images captured by users in the onsite version and stores the results in a database along with their code in order to track the options unlocked by every user. The database can then be accessed by the Unity applications to obtain a user’s unlocked options. The implementation of this system is explained in this subsection.

**Node.js and Express.js**

The server runs on Node.js runtime with Express.js similar to the Audio transcription server. After initialization, the Express application enables the use of Cross Origin Requests so that requests originating from the web version of Backus will also function smoothly. After this the database is created or connected to in case it has already existed. SQLite is used for the database due to its light footprint and ease of use. Once the database has been defined, the POST Requests are also configured for the front-end applications to acquire and add the user’s unlocked options. After this, the WebSocket connection listeners are initialized to listen for client requests. The final step in the creation of the server is to spawn a ZeroRPC process that starts another local server for processing the image in OpenCV.

**SQLite**

SQLite is a software library that provides a relational database management system [74]. Normally, a database requires a separate server process to operate. The applications that want to access the database server use TCP/IP protocol to send and receive requests. This is called client/server architecture. In contrast, SQLite database is integrated with the application that accesses the database [74]. The applications interact with the SQLite database read and write directly from the database files stored on disk. With these advantages in consideration, we have adopted SQLite to store the user data.

The layout of our SQLite database is provided in Figure 5.6. The database has two tables, one for keeping a track of the user’s database id in relation to their login code, and the other for a mapping of the database id to the option number unlocked.
HTTPS Requests

We have configured four HTTPS requests in Express for the application to interact with the database. The first is the 'check-user' request which verifies whether or not a user with a given code already exists in the database or not. The second is the 'create-user' request which adds a user with a given code to the database or returns an error if they already exist. The third request, "add-option", adds the option provided in the JSON request to the database for the user whose code is provided. The "get-options" request then returns all the options unlocked by the user in JSON format for a given user code or returns an error if it is invalid. Every request uses the database to extract results through an SQL query.

WebSocket

The WebSocket server listens for WebSocket requests originating from the client. After connection, every request from the client contains the request data and its type. The first request after the connection is of the type "user_code" which provides links the socket to the user’s code so that every image received afterwards is added to the user’s code. When ever images are received, the data is passed to the Zerorpc server to obtain the result of the detection.

Zerorpc

Zerorpc is a light-weight, reliable and language-agnostic library for distributed communication between server-side processes [75]. It is an implementation of the RPC protocol that is used to communicate between the Node.js process and the OpenCV process which runs in Python. Zerorpc enables Node.js to call a method on the hosted object and pass the image data to it. The Server when initialized, processes the template images and stores the keypoints and descriptors calculated using the SURF detector [34]. The detector and the matcher are also stored to process the images that will be uploaded to the server for object detection.

OpenCV

The processing of images captured during scanning, and object detection is performed using OpenCV (Open Source Computer Vision), a library for real time computer vision [76]. OpenCV was selected for its robustness, efficiency and wide array of algorithms provided. When an image is uploaded for image processing to the Zerorpc server, lists of features are extracted using the the SURF (Speeded Up Robust Features) algorithm. SURF, is a scale-invariant feature based algorithm, that results efficient in the extraction of features of an image both in terms of strength and in terms of computation [34]. The SURF interest point detector uses the Hessian Corner detector for which we have used a Hessian threshold of 300 which provided the best results. Before matching, the image is scaled down depending on its original size in order to ensure a faster match. The image is then matched with every stored template using a brute force L1 Matcher. The results of the match are then filtered using Lowe’s ratio test [77]. For this test, we have used a ratio threshold of 0.7 which provided the best results for our template images. Once the best match has been determined, the object is localized in the image using RANSAC (Random Sampling Consensus). RANSAC is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are
to be accorded no influence on the values of the estimates. This helps RANSAC detect outliers between the matches and find the location where the object exists in the image [78]. We have shown a visualization of the matched and localized object in the image in Fig. [5.7]. The result of RANSAC is a Homography matrix that is used to calculate the location of the object in the scene. The matrix represents an affine transformation for the template in the scene. In order to determine if the transformation is valid so that false positives can be removed from the detection, we use the determinant of the Homography matrix to check if the transformation is orientation preserving [79]. According to our tests, we found that determinant above 0.03 usually indicates a good detection. Another test that we perform to detect if the detection is valid, is to take the SVD of the first to last singular value and check if the value is not too large [80]. Based on the results of these calculations we determine which object exists within the scene and return the results.

Result

Once the object is detected, the result is returned to the Node.js process. The process checks if the option has already been added to the user’s unlocked options or not. If not, it is added to unlock the new option and the client application is notified by sending the option number to unlock. The unlocked option number is added to the list of available options and the visitor is shown the new available option.
Figure 5.7: Visualization of the Object Detection and Localization
Chapter 6

User Study and Data Collection

The Tour Guide was designed with the objective of enhancing visitor interest in the museum space by providing onsite and offsite engagement. While long-term engagement can only be measured through deliberate tracking of difference in visitation patterns of users over multiple visits, we devised a brief two phase study of the prototype of our system to determine which features were of interest to users. This section describes the user study that was performed onsite and offsite and discusses the data that was gathered from users through survey questionnaires.

6.1 Two Phase User Study

The two phase user study was conducted at the Livingston-Backus house in GCV&M. The onsite component was conducted on 2 consecutive days: on the first day, museum staff was given an opportunity to interact with the system, and on the second day the system was made available to volunteers who had signed up for the testing through our registration forms. The museum staff interaction was conducted with the intent of gathering valuable insight from primary enablers for visitor’s interaction with the museum. The expertise of the staff in the historical setting of the museum and their experience with visitors would enable them to successfully gauge whether the system would meet visitor expectations. The volunteers, on the other hand, were a group of diverse individuals ranging in factors such as age, familiarity with immersive technologies especially AR, VR & MR, comfort with technology in daily usage, and familiarity with the museum itself. It was believed that the data gathered from the volunteers would provide insight into making the system accessible and appealing for all groups of visitors. In the next subsections, the layout of the testing site and the triggers is explained. This is followed with the protocol for user interaction that was practiced onsite.

6.1.1 Test site layout and triggers

The user’s interaction in the Livingston-Backus house begins at the main (front) entrance of the house (Figure 2.1). As a visitor enters the main doorway, they find themselves in a large hallway that leads to the back of the house with a staircase on right that leads to the upper floors, and rooms on their left and right. This is where, after starting the application, Backus will describe the architecture of the house, referencing it’s influence from the Greek revival style prominent in New York at the time. In this area of the house, volunteers can encounter the first set of objects in the hallway and the room on the left. These are a newspaper, medical bag, the inkwell and the family Bible (All the trigger objects are shown in Figure 6.1). Scanning each object using the application’s object scanning feature unlocks the relevant topic for Backus to interact with. As the move towards the kitchen, they will also find a pitcher in the dining room.

The experience continues towards the kitchen of the house in the back, where a medicine bottle, herb and wall clock serve as triggers. The final trigger is the exit sign as volunteers exit the house through the back door into the garden. While moving through the house, Backus can be moved and placed into the desired position

By selecting trigger objects that are relevant to the history of the house and Dr. Backus, the layout and look of the
house is preserved to be accurate to the conveyed time period. Thus, through optimal distribution of triggers within
the space, exploration of the house is incentivized and an engaging user experience is created.

6.1.2 Interaction Protocols

The interaction protocols for the staff did not require any sign up as they were free to participate for 15-20 minutes
doing the onsite component depending on their availability during the hours of the onsite test. Volunteers were asked
to fill a pre-visit survey and consent form to provide some idea of their familiarity with the museum, and the consent to
use their anonymous interaction data. Volunteers were suggested to complete these forms and install the apps to their
devices before arriving at the museum. Upon arrival, all users were given an introductory overview of the system. They
were shown a demonstration of the interactions with Backus and performing object scanning. After the introduction,
they were allowed to look around the house, scan objects and unlock the desired options, and interact with Backus.
While we were present onsite to help with any issues or doubts that might arise regarding the system, face-to-face
interactions were minimized to reduce the risk of COVID-19 transmission and also to enable the experience to happen
as intended. While initially the experience was designed to end once users exited the house after scanning the goodbye
sign at the exit door, the onsite survey ended in the kitchen once all the objects inside had been scanned. This was done
to provide an introduction to the second offline component and also answer any questions that users might want to ask.
Once completed, users were asked to complete the survey for the onsite component. For the offsite component, users were given the URL for the WebXR version and asked to interact with Backus using the same code as the one they had created for the onsite version. They were also told encouraged to share the experience with others who were not part of the onsite visit. The various features of the WebXR version such as VR on smartphones through compatible browsers and seamless conversation with Backus without needing to press a button. Once users has completed the offsite component they were asked to fill out the form offsite survey form. A window of 48 hours was given for completion of the offsite component from the completion of the onsite component. Once both the components of the study were completed the data from the study was collected and analyzed to determine the feedback from the system.
6.2 Survey Questionnaire and Data Extraction

For the study, we tested the system with 9 GCV&M staff members and 17 volunteers who enrolled through the registration form. Out of the total 17 external volunteers, 15 completed the pre-visit survey. Out of the total 29 participants in the onsite and offsite study, 17 completed the onsite survey while 12 completed the offsite survey. Many staff volunteers chose to provide verbal feedback about the system that answer the surveys. The onsite and offsite surveys also had open-ended questions at the end for participants to share any opinions they had about the system. In this section, responses obtained from all the three surveys are discussed.

6.2.1 Pre-visit Survey Responses

The Pre-visit survey was designed to gain insight into volunteers’ familiarity with the museum, museum technology, and technology in general. 7 questions were asked as part of the survey (The questions are provided in Appendix A and the responses are in Appendix D). The responses show that 9 out of 15 of the volunteers had previously visited the Livingston-Backus house as shown in Figure 6.3. This shows that a majority of the users were familiar with the house and it’s layout. In the following question, when asked about their interaction with costumed interpreters, over 10 out of the 15 volunteers indicated that they had interacted with the costumed interpreters in the historic village before as shown in Figure 6.4. These responses indicate that most volunteers were already familiar with the theme and historical setting of the museum. When measuring volunteers’ familiarity with technology, we used a Likert scale [81] with ratings ranging from 1 (strongly disagree) to 5 (strongly agree) for volunteers to indicate their response.
Out of the 15 volunteers, 13 indicated that they were comfortable with using technology in work and/or daily life (Figure 6.5). On the other hand, only 10 volunteers indicated that they were comfortable with technology in museums, while 11 indicated that they were comfortable with immersive technologies such as AR or VR (Figures 6.6 and 6.7). Finally, a majority of volunteers (8 out of 15) had not interacted with a virtual tour guide of any type before (Figure 6.8). Overall, 9 participants who had interacted with the costumed interpreters had also indicated being comfortable with immersive technologies to at least some degree. Thus, looking at the data, it can be determined that a majority of participants had some familiarity with the museum and immersive technologies. This is very helpful as it indicates that the volunteers would be able to evaluate our system in context of its use case and user experience more effectively.

6.2.2 Onsite Survey Responses

The Onsite survey was designed to for users to evaluate the onsite system. While two platforms for onsite use have been designed: the Hololens and Mobile, we were only able to test the mobile due COVID-19 restrictions mandated users to bring their own devices. 8 questions were included in the survey to measure the quality of the overall experience and the UI. The last question was an open-ended question where users could provide any feedback that they wanted to add about the experience. All the questions from this survey can be found in Appendix B and the responses are made available in Appendix E. The objects that were available in the online component for scanning are provided in Figure 6.1 while the mapping of topics to ask Backus against the trigger objects is provided in Table 6.1. From the responses (Figure 6.9), the most commonly unlocked option was the Senate (13 out of 15 users), followed by Homeopathy (10...
out of 15 users) and Refuge (9 out of 15 users). The option least unlocked was Cholera (3 out of 15 users), followed by Charlatans (4 out of 15 users). The reason for this seems to stem from the detector working successfully while scanning 2D surfaces with good contrast while facing issues with scanning objects that did not have many contrasting points on its surface or whose appearance was affected due to lighting changes. The next two questions were designed to gauge which aspect of the setting was most appealing to users.

It is seen (Figure 6.10) that most users found the time: 1850s, Rochester, Corn Hill neighborhood more appealing than the character of Backus or the architecture itself. None of the three options was the least liked with a three way equal split for all the options (Figure 6.11). The next questions were regarding the actual experience. 12 of 15 users indicated that the experience met their expectations while 2 disagreed (Figure 6.12). In contrast, 15 out 17 users expressed an enhanced interest in the history of the museum due to the experience (Figure 6.13). This could be taken as an indication that the system was successful to some degree in enhancing interest in the museum but it could have been boosted by other factors such as most users being familiar with the museum already.

In regards to the user experience of the system, the ratings seem to be mixed. 8 users indicate that the system was easy to use and follow while 6 expressed disagreement (Figure 6.14).

Given that most users were familiar with immersive technologies, this can be taken as an indication for modifying the design of the user experience to become more intuitive or according to standard. Despite issues in UI, all 17 users agreed that they would like to see the system expanded to more buildings in the museum (Figure 6.15). This can be considered as success of the system’s appeal in augmenting the experience at the museum. In the open-ended feedback, most users suggested an improvement in the Wifi architecture to improve smoothness of the museum. Some users did mention that the experience would be improved by adding hints to which objects in the museum space could be scanned. In terms of verbal feedback, most of the museum staff believed that the system would be help in improving
engagement in the museum, though some did indicate that some UI elements could be improved. Overall, the feedback for onsite experience seems to be positive and seems to have enhanced some interest in the museum site and its history.

6.2.3 Offsite Survey Response

The offsite survey was designed to evaluate the effectiveness of the offsite component of our system after users had completed the onsite component. The offsite system was designed with two objectives: maintaining engagement by adding offsite value for visitors, and increasing awareness about the museum by enabling the experience to be shared. To evaluate this, volunteers were given 48 hours to use the offsite experience and complete the survey. The questions of the survey are provided in Appendix C while the responses are included in Appendix F.

Only 12 of the 29 volunteers completed the offsite survey. 10 of the 12 volunteers used the application on a Desktop or Laptop browser while only 2 used Google Cardboard or a VR headset (Figure 6.16). Out of 12 users, 8 have indicated that the experience was smooth on their device while 2 have expressed the opposite (Figure 6.17). This could have stemmed from the fact that the WebXR experience relies on WebGL 2.0 standard which is not yet supported by all browsers on all platforms. In terms of adding value, which was one objective of the offsite experience, 7 users have agreed to the experience providing additional value and enjoyment while 2 have disagreed (Figure 6.18). The mobile version is also enabled for offsite use alongside the WebXR version. In the survey, a large majority of users, 6 out of 10 indicated that they would prefer using the WebXR version instead of the mobile version for offsite use (Figure 6.19).

The next two questions collected data about the sharing of the experience with users who did not visit the museum. Out of 12 respondents, only 4 shared the experience with someone else while 8 did not (Figure 6.20) and 3 indicated
that the people with whom the experience was shared were engaged by it and interested in visiting the museum (Figure 6.11). For the last question, 2 users described the issues they faced, while 1 indicated that though the system could be effective for sharing even though they found the onsite experience more interactive. Overall, it is difficult to gauge the effectiveness of the offsite system due to limited feedback received.
The experience has enhanced my interest in the history of the museum
17 responses

Figure 6.13: Enhanced interest in museum after the onsite experience

<table>
<thead>
<tr>
<th>Topic</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cholera</td>
<td>Medical Bag</td>
</tr>
<tr>
<td>Senate</td>
<td>Inkwell</td>
</tr>
<tr>
<td>Canal</td>
<td>Pitcher</td>
</tr>
<tr>
<td>Governor</td>
<td>Newspaper Title</td>
</tr>
<tr>
<td>Refuge</td>
<td>Wallclock</td>
</tr>
<tr>
<td>Charlatans</td>
<td>Castor oil bottle</td>
</tr>
<tr>
<td>Homeopathy</td>
<td>Roots Pillbox</td>
</tr>
<tr>
<td>Goodbye</td>
<td>Exit Sign</td>
</tr>
</tbody>
</table>

Table 6.1: Objects and Corresponding options

The system was easy to use and follow
17 responses

Figure 6.14: Ease of use of the system

46
Figure 6.15: Interest in expansion of the system

Figure 6.16: Platform of offsite experience

Figure 6.17: Smoothness of the offsite experience
Figure 6.18: Additional value added by the experience

Figure 6.19: Preference for WebXR over mobile

Figure 6.20: Experience shared with others
Figure 6.21: Enjoyment of those with whom the experience was shared
Chapter 7

Discussion

Based on the onsite and offsite studies that were conducted with the prototype version of the system, in this section we present our findings and the issues that we encountered during testing. We evaluate whether the data from the surveys indicates if the system was successful in answering our research questions and validating our hypothesis. This is discussed in the next section.

7.1 Findings

The main objective for developing this system was to increase the interest of users in the Livingston-Backus house by integrating the virtual tour guide into the real world through the object scanning feature. We believed that by allowing users to scan trigger objects in the museum space and asking Backus about them, visitors would be incentivized to explore the museum space for more conversational topics.

Based on the observations that can be made from data gathered in the onsite version, it is seen that most respondents indicate that the experience did enhance their interest in the history of the museum. This seems to directly answer our first research question as stated at the end of Chapter 3. Also, as stated in the previous chapter indicator of the system’s success can be seen in user’s expressing their interest in seeing the system expanded to other buildings within the museum. Volunteers were enthusiastic in making suggestions for integrating the system into other buildings or expanding with new features. Here are quotes from some of the volunteers,

- “I see this as being more for younger generations (mainly kids or families). You could make a game/scavenger hunt/quiz for prizes. You could make interactive demos for (example) the tin smith”.
- “I love the idea that visitors could meet Backus and have a better time learning more information when they step into the house. I would absolutely love to see additional characters developed, as well as additional houses represented.”

Thus in terms of engagement, the prototype does seem to be on a successful track but we believe that a more definitive survey that tracks visitation patterns of a large number of museum visitors would provide deeper insight about the pros and cons in the system. For instance, the question “The system was easy to use and follow” met with mixed response and two avenues of improvement are evident to us from onsite observations: better network speeds to speed up the transcription and object detection, and clear indicators for trigger objects; These issues could be explored further to find other issues that hamper user experience by combining multiple sets of trigger objects, with larger group of visitors in different buildings.

For the offsite version, while the data gathered is inadequate to definitively state whether the experience truly adds value and engagement to form the virtuous circle mentioned earlier, responses seem to indicate positive feedback. With 6 out of 10 users expressing preference for the WebXR version over mobile for offsite use, the application has potential to enhance the museum’s virtual outreach. While virtual tours have been adapted in many museums, integrating it with a virtual tour guide like Backus provides much needed guidance to users relying on these platforms. The system could definitely be improved in sharability. Only 4 users out of 12 shared the experience and only 3 found it to be engaging.
for the person with whom it was shared. An extended study of the online version can provide useful insight into which aspects of the web experience can benefit from improvements and provide increased engagement.

While both versions of the system have been designed to maximize visitor engagement, it should be noted that they are not intended to replace the museum experience. The virtual tour guide is designed as a complement to the costumed interpreters and the events that are regularly organized onsite. Thus, future components of the system should be designed to integrate the user’s environment by blending the guide with environmental objects as is now possible with both ARKit and ARCore. This will ensure that the experience becomes an effective means of incentivizing exploration rather than becoming the end goal of the user’s interaction. UI elements can be designed to further encourage exploration of the user’s environment. While testing the system onsite, we also encountered a few issues, which should be tackled to improve the user experience and effectiveness of the system. We discuss these issues in the next section.

7.2 Issues Encountered

During the onsite and offsite testing, issues in the implementation and design of the system were encountered through verbal feedback from users and the user surveys. Here we discuss some of these issues and suggest possible solutions to resolve or bypass them in future versions of the system.

One of the major issues that users encountered in the museum was the need for a good networking infrastructure that is needed for a reliable connection with the server. Due to the Livingston-Backus house being located away from the buildings that have wifi, and the location of the museum outside the city affecting mobile networks as well. While the primary solution to this problem would be to improve the network in the museum space, another would be to perform both object recognition and speech transcription on the device. Libraries such as Vosk [82], an open source speech recognition library for on device and server side speech recognition, and OpenCV4Android SDK [83] could be used to substitute or complement the server side APIs.

Another issue that affected the user experience was with the Object Detector. While the detector was successful with many 2D surfaces, matching 3D objects that did not have a high number of good points for the descriptor and feature detector and were susceptible to lighting changes in the environment prevented some users from unlocking all the options. A good solution for this issue would be to select objects that provide good candidates for the detector, or to train and optimize the detector to detect the objects accurately. Objects that have unique 2D faces should be preferred.

The third issue that was mentioned by users is lack of hints to signify scannable objects. This can be done by providing photos of objects to find, indicate location of nearest objects within a range through GPS etc. It is necessary to ensure that indicators are either deployed through the app, or deployed in the real world in a manner that they blend effectively with the museum space.
Chapter 8

Future Work and Conclusion

Our system expands upon similar systems that were deployed in the museum space earlier, utilizing the progress made by advances in mobile and computing technology along with increasing support for AR in mobile devices. Though innovative, there is a lot of potential for expansion to this system. In the next section, potential areas for improvement to the system are suggested. The final section provides conclusion to the report.

8.1 Future Work

The Backus we have developed currently has only a few options for conversation with users. On the other hand, previous tour guides such as Ada and Grace [10] and Tinker [35] provided a better system where user speech was parsed to select dialog which would seem similar to a natural conversation. Such a character would occupy vast amount of disk space to account for all the audio and animation, which would exceed application storage limitations on most mobile devices. One avenue of research would be to apply existing compression or streaming algorithms to provide the natural conversation experience while constraining the application size. A more natural virtual tour guide could improve the user engagement.

Another avenue of research would be to develop characters from different time periods and settings and enable them to have an interactive multi-agent conversation with the user. Strong et al. [84] have already described a generative conversation tool for writers for conversations. The conversations between characters from different settings and times would be a great educational opportunity for users.

Currently, the creation of a single character requires significant effort and collaboration between programmers, designers, and writers etc. Building tools that can help these teams by automating the generation of some content would reduce the budget constraints on teams while enabling quicker development and deployment of characters.

Our system thus provides many avenues for future researchers to explore and build upon.

8.2 Conclusion

This project began with the objective of development of a virtual tour guide system that could provide onsite and offsite engagement. Through collaboration with a team of experts from multiple domains, we developed a system with the following features:

- Provide visitors the flexibility to access the tour guide from a platform of their choice among the following: the Hololens, Android and iOS mobile devices, web, WebXR and WebVR compatible browsers.
- Integrate the virtual tour guide with the museum space through the use of trigger objects distributed around the museum that can be scanned and used to unlock new conversation options for the tour guide.
- Allow seamless transition between both onsite (mobile and Hololens) and offsite (web, WebXR and WebVR) experiences by tracking user interactions and incentivizing repeated visits to the museum.
Through a user survey with museum staff and participants we attempted to answer the following questions:

• Does a portable virtual tour guide that rewards exploration of the site, invoke an increased interest about the site’s history?

• Does the ability to share the experience with non-visitors improve awareness about the museum and its setting?

Through our onsite testing and offsite testing of the prototype, we believe that the system does successfully increase interest about the site’s history by rewarding exploration. The system enhances interest in the site’s history by integrating real world objects with the virtual tour guide. For the offsite system though, we cannot definitively state if sharing really does improve user interest in the system due to lack of enough volunteers to generate data. But from the feedback we have received we believe that the system can contribute to increasing awareness about the museum among non-visitors. We believe that through research in this domain through deliberation and conducting extensive surveys with more volunteers, we can integrate the desired improvements in future versions.
Appendix A

Pre-visit Survey

1. Are you a member of GCV&M?
   - Yes
   - No

2. Have you visited the Livingston-Backus house before?
   - Yes
   - No

3. If you have visited the museum before, have you interacted with the costumed interpreters in the historic village?
   - Yes
   - No
   - Not Visited Before

4. I am very comfortable with using technology, generally, in work and/or everyday life.
   - 1 (Strongly Disagree)
   - 2
   - 3
   - 4
   - 5 (Strongly Agree)

5. I am very comfortable with using technology in museums.
   - 1 (Strongly Disagree)
   - 2
   - 3
   - 4
   - 5 (Strongly Agree)
6. I am very comfortable with using immersive technologies such as AR or VR
   • 1 (Strongly Disagree)
   • 2
   • 3
   • 4
   • 5 (Strongly Agree)

7. Have you ever interacted with a virtual tour guide before?
   • Yes
   • No
Appendix B

Onsite Survey

1. Please check all the topics that you asked Backus about (Multiple options selectable)
   - Governor
   - Senate
   - Refuge
   - Canal
   - Cholera
   - Charlatans
   - Homeopathy
   - Goodbye

2. What subject interested you the MOST about the Livingston-Backus House?
   - I was most interested in learning about the place: the house, the architecture, the decor
   - I was most interested in learning about the time: the 1850s, Rochester, Corn Hill neighborhood.
   - I was most interested in learning about the person: Fredrick Fanning Backus, a medical doctor and lawyer.

3. What subject interested you the LEAST about the Livingston-Backus House?
   - I was least interested in learning about the place: the house, the architecture, the decor
   - I was least interested in learning about the time: the 1850s, Rochester, Corn Hill neighborhood.
   - I was least interested in learning about the person: Fredrick Fanning Backus, a medical doctor and lawyer.

4. The experience met my expectations
   - 1 (Strongly Disagree)
   - 2
   - 3
   - 4
   - 5 (Strongly Agree)
5. The experience has enhanced my interest in the history of the museum
   • 1 (Strongly Disagree)
   • 2
   • 3
   • 4
   • 5 (Strongly Agree)

6. The system was easy to use and follow
   • 1 (Strongly Disagree)
   • 2
   • 3
   • 4
   • 5 (Strongly Agree)

7. I would like to see the system expanded to more buildings in the museum with more characters
   • 1 (Strongly Disagree)
   • 2
   • 3
   • 4
   • 5 (Strongly Agree)

8. Is there anything more you would like to suggest or share?
   [TEXTBOX FOR OPEN ENDED ANSWER]
Appendix C

Offsite Experience

1. Check the box that corresponds to your interactions with the WebXR version of Backus Overall: (Multiple Options Selectable)
   • I used the experience on a Desktop or Laptop Computer
   • I used the experience on a Google Cardboard with my smartphone

2. I feel that the experience was smooth on my device
   • 1 (Strongly Disagree)
   • 2
   • 3
   • 4
   • 5 (Strongly Agree)

3. I enjoyed the additional value and enjoyment provided by the offsite experience
   • 1 (Strongly Disagree)
   • 2
   • 3
   • 4
   • 5 (Strongly Agree)

4. I would prefer the WebXR version to the Mobile version for offsite use
   • 1 (Strongly Disagree)
   • 2
   • 3
   • 4
   • 5 (Strongly Agree)

5. Did you share the experience with those who did not visit the museum?
   • Yes
   • No
6. Those who I shared the experience with were engaged by it and are interested in visiting the museum
   • 1 (Strongly Disagree)
   • 2
   • 3
   • 4
   • 5 (Strongly Agree)

7. Is there anything you would like to add about this experience?
   [ TEXTBOX FOR OPEN ENDED ANSWER ]
Appendix D

Pre-visit Survey Responses
Figure D.1: Pre-visit Survey Responses Part. 1
Figure D.2: Pre-visit Survey Responses Part 2

I am very comfortable with using technology in museums
15 responses

I am very comfortable with using immersive technologies such as AR or VR
15 responses

Have you interacted with a virtual tour guide before
15 responses
Appendix E

Onsite Survey Responses
Figure E.1: Onsite Survey Responses Part 1
Figure E.2: Onsite Survey Responses Part 2
Is there anything more you would like to suggest or share?

11 responses

i had issues with getting anything to scan on site. i still liked everything that i was able to view from the other participants

My Android phone didn't scan everything, but it was probably a network error

I would suggest reworking the backus model to have a full right foot, and have a consistent vertex density on the mesh (Are the normals inverted on the right foot, causing Unity to back face cull and not display the foot properly?). I would suggest adjusting the placement code to not allow backus to be placed on vertical surfaces, and perhaps rework the controls to allow for manual x,y,z placement of backus in the scene. (I had problems with backus appearing several feet above the floor and I couldn't adjust his y position). Another idea I had was having pictures of the items themselves to scan, that way you could put tracking markers on the pictures to allow for easier image recognition in case the object scanning doesn't work.

The only reasons that I gave the experience low scores are because (1) there seemed to be issues with Android and (2) the internet connectivity issues HAVE to be fixed

It was very cool though. REDACTED PERSONAL DETAILS With this system, she could "visit" the museum. I see this as being more for younger generations (mainly kids or families). You could make a game/scavenger hunt/quiz for prizes. You could make interactive demos for (example) the tin smith.

You need an overall guide that can walk you around the museum and provide basic info on each building.

I'm not sure how well this will scale with population on busy days. I can see a tourbus full of kids showing up leading to 20 people running around a house trying to scan things and listen to an avatar tell them things.

Good job. I really loved the idea!
It might be nice to include a list of objects that you can scan within the application itself to prevent confusion, similar to a scavenger hunt. Having barcodes or signs within the building that indicate what can be scanned may be obstructive to the immersion/histoty.

The WiFi was slow. Some objects did not scan. Great ideas and great delivery of info.

I was wondering that, can this be used as a plugin or something like google earth for museums and other historic buildings.

There needs to be a screen prompt for signalling to users that an object is scan-able or not. Although glitchy, some of the issues seem to come from the wi-fi infrastructure of the museum and not from the project avatar itself.

Better connection for devices in the museum would enhance the experience. It was a novel and interesting way to explore it!

Overall - awesome project! However, there are still some kinks to iron out. The wifi in the house is very bad (not your fault), the app continuously glitched and shut down (possibly my phones fault), and the app had a difficult time scanning the objects or showed up that they were scanned previously, when I had never successfully scanned them. Despite all of the difficulties - the app idea is awesome! I love the idea that visitors could meet Backus and have a better time learning more information when they step into the house. I would absolutely love to see additional characters developed, as well as additional houses represented. This is a super cool project - and all from the little smart phone in your hand! Well done!
Appendix F

Offsite Survey Responses
Check the box that corresponds to your interactions with the WebXR version of Backus
Overall:
12 responses

- I used the experience on a Desktop or Laptop: 10 (83.3%)
- I used the experience on a Google Cardboard: 2 (16.7%)

I feel that the experience was smooth on my device
12 responses

- 1 (16.7%)
- 2 (16.7%)
- 2 (16.7%)
- 6 (50%)

I enjoyed the additional value and enjoyment provided by the offsite experience
12 responses

- 1 (8.3%)
- 3 (25%)
- 6 (50%)

I would prefer the WebXR version to the Mobile version for offsite use
10 responses

- 9 (90%)
- 1 (10%)
- 3 (30%)
- 3 (30%)
- 3 (30%)

Figure F.1: Offsite Survey Responses Part 1
Figure F.2: Offsite Survey Responses Part 2

Did you share the experience with those who did not visit the museum?
12 responses
- Yes: 66.7%
- No: 33.3%

Those who I shared the experience with were engaged by it and are interested in visiting the museum
6 responses
- 0 (0%)
- 1 (16.7%)
- 3 (50%)
- 1 (16.7%)
- 2 (33.3%)

Is there anything you would like to add about the experience?
5 responses
- Having a way to view the dialog and artifacts I might have missed would be wonderful, as it feels like I didn’t get anything extra out of using the web version.
- The offsite experience never loaded on my computer (a one-year-old Macbook Pro running Chrome and 180Mbps down/5Mbps up). I sat there for several minutes watching the “Loading” screen.
- Hard time getting it to open in My phone. User error I think. Saw it demonstrates at the museum. Very cool. Keep up the good work.
- The web experience works well. It could be good to share our experiences with other people that we know. Also could be used to hold competitions for who can find more of the events!
- The in-person experience is more interactive and found it fun to explore with.
Bibliography


[38] Joan Beaudoin. Art museum collections online: Extending their reach – MW20 | Online.


[42] MRTK. Dictation.


