# Open Virtual Exhibit Creator (OVEC): An Open Source Framework for Virtual Reality Museum Exhibit Creation

Brett Schumacher

Submitted under the supervision of Dr. Evan Suma Rosenberg to the University Honors Program at the University of Minnesota-Twin Cities in partial fulfillment of the requirements for the degree of Bachelor of Science, *summa cum laude*, in Computer Science.

Spring 2023

**Abstract**

There have been numerous innovative and impressive works in the current landscape of applying Virtual Reality in museums, yet there is a striking lack of community resources to aid in their development. Curators without the technical background in VR are then left with a significant barrier of entry. This paper presents the Open Virtual Exhibit Creator (OVEC), a new open-source Unity tool aimed at closing this gap. OVEC was developed over the last year in partnership with the Wangensteen Historical Library at the University of Minnesota Twin Cities as an initial effort in creating an open source framework for Virtual Reality exhibits. Despite not being able to conduct formal user evaluation, sufficient progress was made and informal feedback was received so as to measure its success. OVEC shows potential to assist in the creation of VR exhibits and has a flexible design to allow for easy future extension.

# Contents

# 1 Introduction

Virtual Reality (VR) technology has become an increasingly popular tool for museums to create engaging exhibits that allow visitors to explore their collections in new and innovative ways. However, creating VR exhibits can be a complex and time-consuming process, often requiring specialized knowledge and resources. As a result, numerous papers describe the process that specific museums have gone through when creating exhibits, but there has been limited work towards creating shared community tools.

Due to this lack of public tools, smaller collections and libraries may hesitate before jumping into VR due to limited time and funding, or a lack of staff with the necessary technical background. The Wangensteen Historical Library (WHL) at the University of Minnesota-Twin Cities, for instance, boasts a vast collection of medical texts and artifacts, many of which are too fragile to display regularly or allow visitors to handle. The library has already made efforts to digitize some of these artifacts through photogrammetry. With access to a VR space at the University, they saw an opportunity to bring these artifacts to life in a VR exhibit. After consultation with the lead curators at WHL, I partnered with them to develop the Open Virtual Exhibit Creator (OVEC), an open source tool inside the Unity game engine that uses OpenXR to aid in the development of virtual museum exhibits.

# 2 Related Work

## 2.1 VR Exhibits

Early research into the use of VR in museums goes back decades [7], and enough work has been published on their creation that the literature reviews have literature reviews [5]. Many virtual exhibits have taken advantage of the increased presence that VR offers in virtual spaces to do reconstructions of historical sites [4] [10] [12] [14]. Others focus on more traditional gallery-style exhibits [7] [8], or even full multi-user VR museums [9]. Each of these works goes through their own workflow and process for scanning or recreating the environments. Despite often using Unreal Engine or Unity, the development of the exhibits tend to start from scratch each time.

## 2.2 Creation Tools

Several attempts have been made at developing generalized VR exhibit creation tools. For instance, The Invisible Museum [16] offers numerous useful tools for setting up virtual exhibits, but as of writing, it is inaccessible to developers or users. Similarly, The Virtual Museum Framework (VirMuF) [3] contains exhibit creation tools for gallery display and artifact interaction that are extensive and quite impressive. However attempts to access VirMuF's code have been unsuccessful. In contrast, outside of exhibit creation, VREX [13] is a fully open-source Unity package for virtual environment creation. VREX is aimed at VR researchers setting up studies in the fields of psychology and neuroscience, and offers tools and UI for modifying various aspects of the study. It is worth noting that this discussion is not intended to pass judgment on tools that charge for their development effort, but to illustrate the possibility of public tools for these issues and the current gap that exists. Museums, which must consider funding, may hesitate to donate the tools they spent time and resources creating. Hence, universities and researchers may have a role in working on these open-source projects to contribute to the field.

## 2.3 Virtual Object Manipulation

In addition to the aforementioned virtual environments and tools, various VR interaction techniques have been proposed to enhance the user experience. In the realm of museums specifically, the Virtual Touch Museum [15] employs real-life proxies of artifacts to simulate the experience of handling fragile objects. Similarly, Ant-Man Vision [2] is a technique that enables visitors to resize themselves to view exhibits at different scales.

In the broader field of 3D user interfaces (3DUI), there are numerous other interaction techniques to be chosen from. In my work, I utilized the Spindle technique [6]. This technique allows users to manipulate and resize objects using two controllers, where the distance between the controllers scales the object, and it behaves as if an invisible "spindle" were spanning the user's hands that the artifact is attached to, rotating and moving it accordingly. Alternately there is the extension to spindle called Spindle + Wheel which allows the dominant hand to be used to rotate the object around the spindle. The two-handed technique iSith uses rays extending from the two hands, and those rays' closest point of intersection to handle moving the object [6].

For one-handed object manipulation, a simple virtual hand technique was chosen for OVEC, where the user's controller is mapped one-to-one to its VR counterpart. Then when grabbing an object, that object moves and rotates to match the hand. Other one-handed techniques include Go-Go which is similar to the simple virtual hand, but allows the virtual hand to extend past the user's real arm length, so that they can reach farther [6] There is interesting research into interaction techniques with fingers as well, however that was not explored in much depth since it would add the requirement of finger tracking to be used, which not all modern headsets support.

# 3 Methodology

## 3.1 Project Scope

For the scope of OVEC, I looked at what research had already been done in the area. Many workflows had been documented for exhibit creation, but the code itself tended to be proprietary. Therefore when designing OVEC, I made my top priority the ability for it to be easily understood and extended by others in the future in order to address the issue of many exhibits needing to start from scratch when going to VR. While a large tool like VirMuF [3] would have been ideal, within the scope of a single work I decided to narrow my goals. Thus I focused on streamlining the aspects of virtual exhibit creation that would be common to many types of exhibits.

One of the biggest strengths of VR exhibits is its ability to offer unique experiences that differ from a standard museum gallery. So instead of basing the system around a gallery style exhibit creation, I worked on automating the process of bringing artifacts into VR and setting up their interactions and displays. Ideally, a curator would be able to use OVEC to set up a basic exhibit with not only information on the artifact but also interaction systems to go along with them, all with little to no coding experience.

## 3.2 System Overview

OVEC as a whole is made of relatively few components so that it remains easy to understand. All the components can be seen in Figure 1.

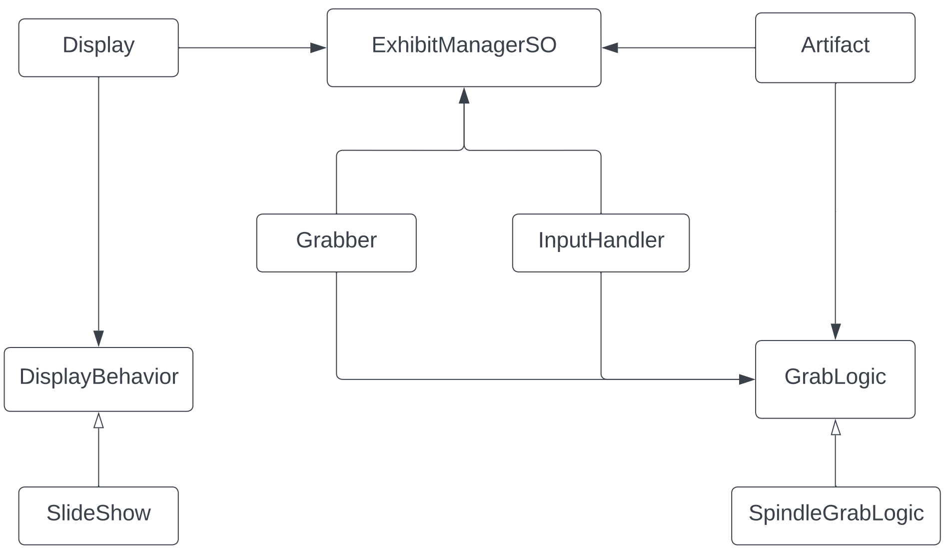


Figure 1: Diagram of the reference points used for Spindle

### 3.2.1 Exhibit Manager

The core of the system is the component at the top-center of Figure 1, the Exhibit-ManagerSO. “SO” stands for Scriptable Object, which is a specific Unity interface for an object that can be created independently of one scene but still referenced by the objects of any scene.

Its main use in the project is to offer a common point for all the different components of the exhibit to communicate through without referencing each other. By having various events declared publicly inside the Exhibit Manager, the different components can communicate by listening to and invoking the different events. In general this allows for easy testing of individual elements by using test scripts to intercept and trigger the events to see how the system reacts.

This not only allows for rapid testing and less interdependent code, it also makes future extension and modification much easier. For example, changing from VR inputs to keyboard and mouse now only requires adding a script that reads in the keyboard and mouse inputs and calls the relevant events inside the ExhibitManager rather than refactor the entire project.

### 3.2.2 Input Handler

The InputHandler has the job of being the bridge between Unity's Input System and the Exhibit Manager. It takes in the inputs by the user and translates them into the various events that need to be invoked for the rest of the system.

Most notably, it also handles the logic of detecting when the user starts an object interaction. It does this by running a sphere intersection around the player's hand when they use whatever has been designated as the grab button, by default this is the trigger button on either controller. Then if any colliders are found, the nearest one to the hand is passed to the ExhibitManager to be dealt with by other components.

The Input Handler also manages raising events for the button inputs of the controllers, which is currently only used for advancing slide shows in the display section of the project.

### 3.2.3 The Artifacts

The Artifact script itself does not have very much responsibility. Its main role is to let the system know when an object is an artifact, and to hold that artifact's display information to be passed on to the display part of the system when the artifact is interacted with.

### 3.2.4 Artifact Interactions

Interactions in the system are handled by two components. First is the Grabber script which listens to the events in the Exhibit Manager raised by the Input Handler when an object is grabbed. This script keeps track of which objects are currently being interacted with and handles the logic of notifying the system about picked up and released artifacts.

The other part of object interactions is the object's specific interaction logic, contained in the GrabLogic script. GrabLogic itself acts as an interface for object interaction behaviors. Any object that wants to support interaction needs to have both a collider, so that it can be detected, and a script that implements the GrabLogic interface so the object knows how to react to being interacted with. When an object is grabbed or released, the Grabber script manages the process of communicating with the specific GrabLogic script on that object so that it can update appropriately.

The only interactions that were implemented for this project, as discussed previously, wwere the Spindle grab logic for two-handed manipulation and a simple virtual hand for one[6]. This allows users to intuitively grab and move objects around with one hand, or examine them closer through freely scaling them up with two hands so that smaller details may be examined and appreciated.

To get the spindle grab logic to work, 2 things need to be initially calculated. As shown in Figure 2, these 2 things are: first, the spindle itself (shown in red) which spans between the two controllers (the red dots); and second, the center of the spindle relative to the center of the object (shown in blue). When the user first grabs an object, both of these vectors are saved in the object's local coordinates so that they may be transformed to global coordinates each frame to properly update the object. Additionally, the initial local size of the object is saved for proper scaling.

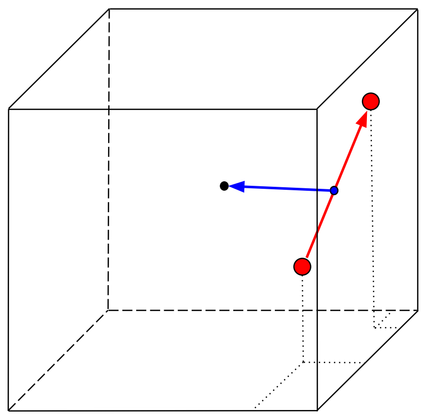

Figure 2: Diagram of the basis used for the Spindle technique

For the update each frame, first the new spindle is calculated in global coordinates by taking the displacement between the two controllers. Then the initially calculated spindle is transformed to global coordinates. The difference in scales between the magnitudes of these two vectors is used to set the new scale of the object. Then quaternions are used to get a from-to rotation from the initial spindle to the new spindle and this is applied to the object to align the spindles' rotations. Finally the center of the spindle is transformed into global coordinates and aligned to the center of the new spindle.

### 3.2.5 Displays

The final part of the system is how displays are handled which closely mirrors the artifact and interaction system structure. The Display class listens to events from the ExhibitManager so that when a new artifact is held, a Canvas prefab can be instantiated if there is a display available to render.

After creating the Canvas, the Display will search the new Canvas for a DisplayBehavior, which is similar to GrabLogic as it is an interface to be implemented by other classes. A DisplayBehavior only needs to describe how it reacts to the different controller inputs. This iteration of the project has only the SlideShow class available as a DisplayBehavior in which the main input button advances the slideshow and the secondary input button goes back in the slides.

### 3.2.6 The Artifact Import Menu

The Artifact Import Menu is a custom editor window created for the project. This allows curators to be able to set up their artifacts as long as they have the 3D model. After all the fields specifying import and save paths are filled out they can click the "Import Artifact" button which will automatically copy files into the project and save a prefab with all necessary scripts added and configured so it is ready to be added to an exhibit.

If the curator has a slideshow for the artifact they may attach it by first exporting it as a series of PNGs and then linking the folder of images. This type of exporting is supported by Microsoft PowerPoint, and Google Slides can export to a PowerPoint file format. If the curator would rather create their own display inside Unity using a Canvas, they can instead link their own prefab into the Artifact Importer as well.

The importer was programmed to try and catch as many errors as possible before they happened while importing so that it could give useful advice to the user. It will provide different messages depending on which part of the process failed, and as well as a clear message when the importing has succeeded. For example, custom responses will be given in each case if there are errors with loading the artifact, saving it, or generating the canvas, rather than the default error messages provided by Unity which may be confusing to new users.

## 3.3 Creating an Exhibit and Displaying an Artifact

As stated previously, one of the core goals was to create a tool accessible to people unfamiliar with Unity or VR. This next section provides a high-level overview of the current workflow for creating an exhibit using OVEC without getting too deep into the intricacies of the code base or VR setup. Assuming that OpenXR has already been installed, a brief overview of the process goes as follows:

1. Create an ExhibitManagerSO inside the project files.
2. Add an XR Rig to the scene.
3. Set up the Input Handler and Grabber scripts, likely attached to an empty game object.

4. Use the Artifact Importer Menu (shown on the right of Figure 3) to generate artifact prefabs from existing models and slide shows.

5. Set up the environment as desired and arrange the artifacts.

6. Add the Display script to the area that the slide shows should be shown.

After completing these steps, the user will have a simple exhibit with a single artifact that utilizes a SlideShow DisplayBehavior and Spindle GrabLogic. All these elements can be seen in the following example exhibit:

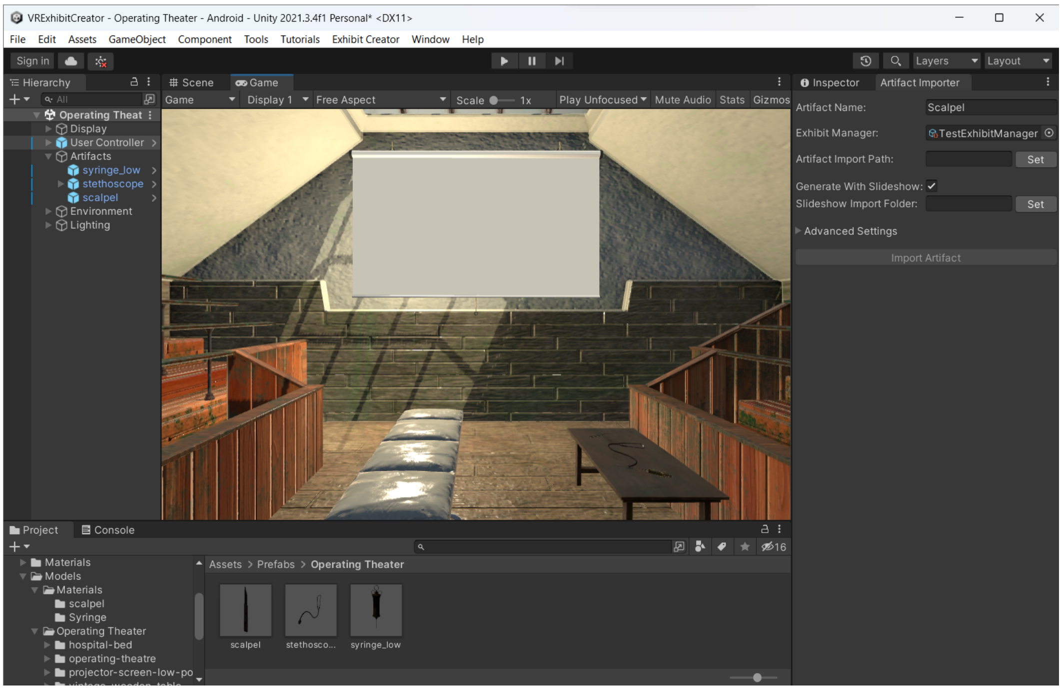


Figure 3: Screenshot of an operating theater created using OVEC

For this example exhibit models were used from Sketchfab[1], a website where people may upload their models for general use. Going through each of the main

[1]The models used for this oeprating theater example were: Operating Theater by Matt LeMoine, Hospital Bed by CobbleGames, Vintage Wooden Table by Poring, Projector Screen (Low Poly) by filththemutt, Medieval Syringe by Zoopa225, Rigid Stethoscope by Simon Pasi, and Scalpel by Lucy Greenhill

elements discussed in the exhibit, the artifacts can be found on the table to the right of Figure 3, and are shown more closely below in Figure 4.

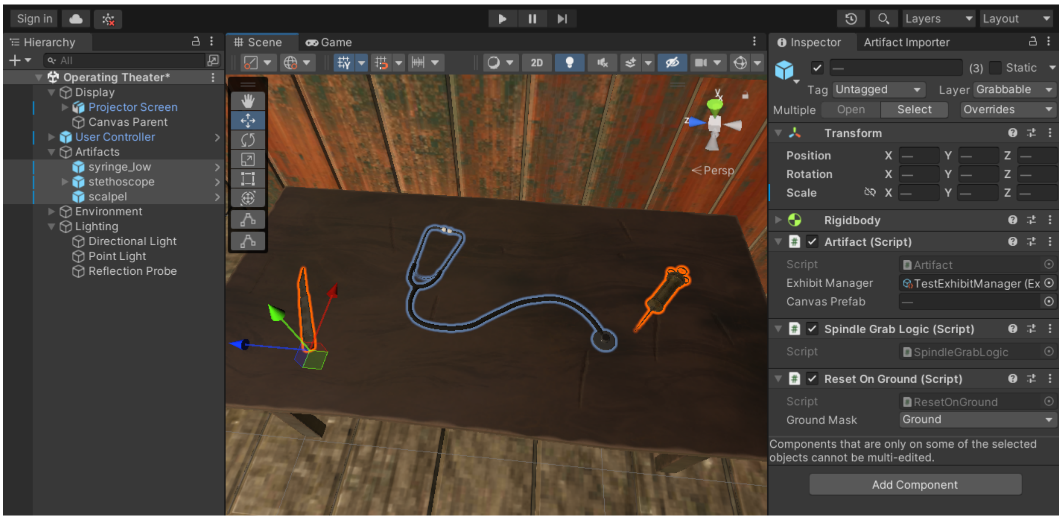

Figure 4: Artifacts used in the example operating theater exhibit

Next, the display can be seen mounted to the back wall of the exhibit in Figure 3, and once one of the artifacts has been grabbed this is where the slide show will appear. An example interaction with the syringe artifact is shown in Figure 5.

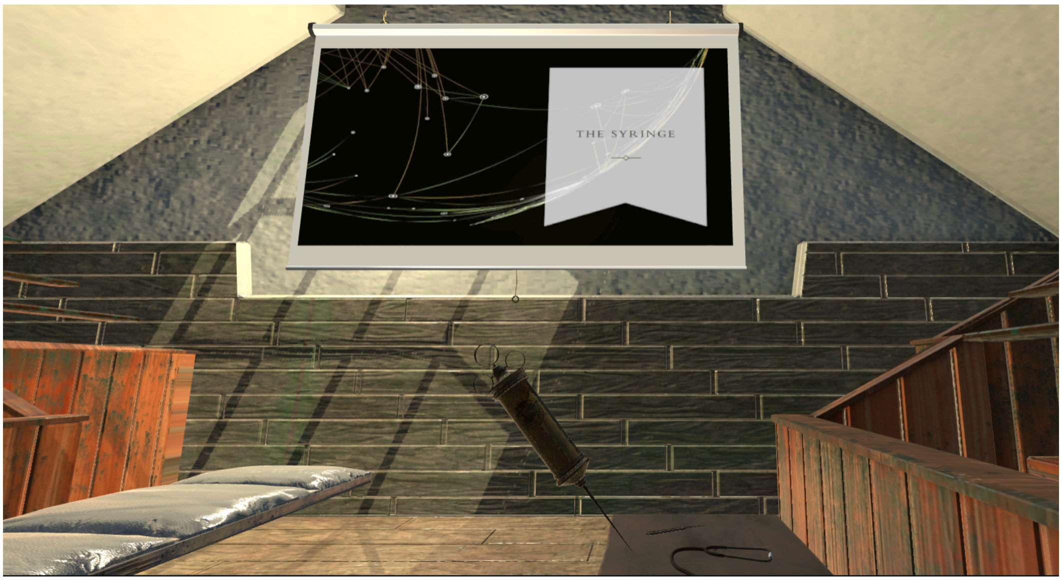


Figure 5: Slideshow for the syringe shown from the user's perspective

## 4 Informal Evaluation

To get feedback on OVEC, it was initially envisioned to recreate a surgical chamber filled with the previously made 3D scans from the WHL's collection. A curator would have gone through the process of creating this exhibit using OVEC to give feedback. Then after its creation visitors would go through the exhibit and give their feedback on the experience. Unfortunately, there was no time to create or run a test exhibit to this extent.

Rather, the demo shown previously in figures was created, and informal feedback was received by the two head curators at the WHL, as well as a three others associated with the WHL and health science education. For gathering this feedback I first had the participants engage with the demo exhibit in VR, noting their interactions and comments about the artifacts. Afterwards I guided them through the use of the tool, showing the steps to go from creating a new project to getting the package running, and basic use of the artifact importer.

For the demo, those who interacted with the artifacts in the scene were positive about the spindle interaction behavior. If slightly confused after the initial explanation, they quickly gained an intuitive understanding on how to scale and manipulate the objects as they wanted. But the simplicity of the spindle interactions did leave the curators slightly worried as to how/if the system could handle more complex behaviors, such as examining individual tools of an old surgical kit. Thankfully, as discussed, OVEC was designed to be expanded to deal with new interaction behaviors. So while the surgical kit example may require a future developer to implement the specific interaction, it would be on top of the existing code, instead of replacing or modifying it.

It was also suggested that the artifact displays could appear above the artifacts themselves as small popups to interact with. With the system as it currently is, this would require either changes to the display system, or a separate display system to be created. Optionally, the canvas could be attached as a child to one of the controllers instead of an element of the scene, so it would display above the wrist of the user, however this may be uncomfortable to read.

Because the demo artifacts were all example models, the WHL staff also expressed concern over how photogrammetry models would function with OVEC. While the Artifact Importer is limited to the 3d model formats that Unity supports, there are other tools such as Blender that exist that could aid in going between formats. Tools such as Blender could also help simplify geometry of high resolution photogrammetry models that may be too detailed for real time VR, which was another concern raised.

In general, OVEC seemed to not raised any large concerns for the curators, and sparked many ideas for exciting directions the tools could take in the future, many of which will be discussed in the following sections.

# 5 Discussion

Taking this feedback into account, as well as looking at the final system compared to the initial goals, we can see how successful the project was overall.

The artifact importer was largely successful. As shown, it can take in the model and a slideshow and output a prefab that contains all necessary components to be immediately used in an exhibit. There were a few shortcomings, however. Most glaringly, model materials are not automatically assigned so they need to be separately imported and added to the object models. This is not too burdensome, but it adds a few more steps to the process and could be confusing to someone who does not know where all the different input textures should go.

Additionally, the fact that importing a slideshow requires a series of PNGs is cumbersome, as it either requires going through Microsoft PowerPoint which not every user may have a license for, or manually exporting and naming each slide individually to work with the importer. Importing as a PDF was explored, however Unity does not support PDF rendering by default, and the tools others have created to do so cost money to use.

The systems for interacting and displaying artifacts were both present and successful as well, but the largest drawback was the fact that I ran out of time to support anything but Spindle interaction and slide show displays. These will be left for future work, but for now curators using OVEC would not have any alternate interaction or display choices unless they had the coding knowledge to extend OVEC themselves.

The primary goal of OVEC was ease of extensibility for future developers. This is the area in which I am most satisfied with, in regards to the final state of the project. Using the Exhibit Manager as a connection point between the systems leaves it very open for adding new behaviors and modifying existing ones. While no interaction or display classes were created besides the basic ones, the use of interfaces to integrate them in the system means that future extension will require no modification of the existing code. The only exception would be if future interactions required more input events to be added, such as more than primary and secondary input buttons, but this would only involve the setup of a couple new Unity Events.

# 6 Conclusion

There are many potential extensions that could be made to the project that have been discussed with the WHL staff. These extensions fall under two main categories: interactions and systems upgrades. Most commonly discussed with the curators at the WHL were interactions with texts, due to the WHL's large medical text collection. By adding more interaction behaviors, various special types of artifacts would be added. Specifically discussed were books, scrolls, and flap anatomies. Scrolls being the easiest, only have a single image that needs to be imported. Then gripping both ends of the scroll, the user could turn their controllers in sync to turn the scroll forward or backward, simulating the way one reads scrolls in real life. Books would be a series of images and a more complicated interaction behavior could allow the user to hold the book or turn the pages one at a time. Flap anatomies would each have to be customly created, as they would allow the user to flip each layer of the anatomy which are customly shaped to the anatomy that is being displayed.

As has been pointed out by others, [11], the possibility for narrative experiences in exhibits is a major draw for VR. Promising future applications could be features to enable narrative into the exhibit or features to help tailor the object manipulations to the exhibit. This could involve adding a system to track "progress" through an exhibit or facilitate branching choices the user can make while going through it. Examples inspired by the VirMuF application [3] include viewing a dynamic reconstruction of the artifact extrapolated back in time or a way to see slices of the artifact to look inside it.

Locomotion is also an important aspect in the design of any VR experience and has been investigated by others [1], but was not the focus of this work. Thus it could then make for a good extension in the future. This could include support for different locomotion styles such as simple one-to-one real life to virtual walking, joystick walking, teleportation, or even redirected walking.

In terms of system upgrades, addressing the previously raised issues of the texture assignment of imported artifacts is a clear next goal. Additionally figuring out a better way to import the slide shows would be a more difficult, but useful improvement. For a very quick creation of an exhibit, a default exhibit template could be made where the user selects artifacts and a fully set up scene is created. This was discussed during early talks with the curators at the WHL, but was too out of scope for the timeline of the project, so it would make another promising extension. Using the artifact importer to get the artifacts, and then the exhibit creator to generate a template exhibit, the amount of work for the curator could be drastically reduced.

Overall, the project successfully implemented the design goals it set out to accomplish, and received a small amount of external feedback. From this I hope that the documentation and open source nature of the code means that the project will not end with this work, but be used and extended by the WHL and others in the future.

# 7 References


[1] Ariel Caputo, Federico Borin, and Andrea Giachetti. A Comparison of Navigation Techniques in a Virtual Museum Scenario. *Eurographics Workshop on Graphics and Cultural Heritage*, page 8 pages, 2019.

[2] Yu-Ling Chen, Yan-Xuan Cheng, I-Yun Hung, and Shih-Wei Sun. Ant-Man Vision in the Museum with Interactive and Immersive Surreal Experience Based on Machine Learning. In *2022 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 242–244, CA, USA, December 2022. IEEE.

[3] Mohammed Elfarargy and Amr Rizq. VirMuF: The Virtual Museum Framework. *Scalable Computing: Practice and Experience*, 19(2):175–180, May 2018.

[4] T. P. Kersten, F. Tschirschwitz, and S. Deggim. Development of a Virtual Museum Including a 4D Presentation of Building History in Virtual Reality. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W3:361–367, February 2017.

[5] Demetrius Lacet, Maria Van Zeller, Paulo Martins, and Leonel Morgado. Digital Storytelling approaches in Virtual Museums: Umbrella review of systematic reviews. *Journal of Digital Media & Interaction*, pages 23–44 Páginas, December 2022.

[6] Joseph J. LaViola, Ernst Kruijff, Ryan P. McMahan, Doug A. Bowman, and Ivan Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley usability and HCI series. Addison-Wesley, Boston, second edition edition, 2017.

[7] George Lepouras, Dimitrios Charitos, Costas Vassilakis, Anna Charissi, and Leda Halatsi. Building a VR-Museum in a Museum. In *Proc. of VRIC Virtual Reality International Conference*, May 2001.

[8] Jean-Luc Lugrin, Florian Kern, Ruben Schmidt, Constantin Kleinbeck, Daniel Roth, Christian Daxer, Tobias Feigl, Christopher Mutschler, and Marc Erich Latoschik. A Location-Based VR Museum. In *2018 10th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)*, pages 1–2, Würzburg, Germany, September 2018. IEEE.

[9] Antonella Poce, Alessio Caccamo, Francesca Amenduni, Maria Rosaria Re, Carlo De Medio, and Mara Valente. A Virtual Reality Etruscan Museum Exhibition – Preliminary Results Of The Participants' Experience. *European Distance and E-Learning Network (EDEN) Conference Proceedings*, (2):40–49, 2020.

[10] Guy Schofield, Gareth Beale, Nicole Beale, Martin Fell, Dawn Hadley, Jonathan Hook, Damian Murphy, Julian Richards, and Lewis Thresh. Viking VR: Designing a Virtual Reality Experience for a Museum. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pages 805–815, Hong Kong China, June 2018. ACM.

[11] Maria Shehade and Theopisti Stylianou-Lambert. Virtual Reality in Museums: Exploring the Experiences of Museum Professionals. *Applied Sciences*, 10(11):4031, June 2020.

[12] Ovidia Soto-Martin, Alba Fuentes-Porto, and Jorge Martin-Gutierrez. A Digital Reconstruction of a Historical Building and Virtual Reintegration of Mural Paintings to Create an Interactive and Immersive Experience in Virtual Reality. *Applied Sciences*, 10(2):597, January 2020.

[13] Madis Vasser, Markus Kängsepp, Murad Magomedkerimov, Kälver Kilvits, Vladislav Stafinjak, Taavi Kivisik, Raul Vicente, and Jaan Aru. VREX: an open-source toolbox for creating 3D virtual reality experiments. *BMC Psychology*, 5(1):4, February 2017.

[14] Alexander P. Walmsley and Thomas P. Kersten. The Imperial Cathedral in Königslutter (Germany) as an Immersive Experience in Virtual Reality with Integrated 360° Panoramic Photography. *Applied Sciences*, 10(4):1517, February 2020.

[15] Yuchen Zhao, Maurizio Forte, and Regis Kopper. VR Touch Museum. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 741–742, Reutlingen, March 2018. IEEE.

[16] Emmanouil Zidianakis, Nikolaos Partarakis, Stavroula Ntoa, Antonis Dimopoulos, Stella Kopidaki, Anastasia Ntagianta, Emmanouil Ntafotis, Aldo Xhako, Zacharias Pervolarakis, Eirini Kontaki, Ioanna Zidianaki, Andreas Michelakis, Michalis Foukarakis, and Constantine Stephanidis. The Invisible

Museum: A User-Centric Platform for Creating Virtual 3D Exhibitions with VR Support. *Electronics*, 10(3):363, February 2021.