Virtual Renaissance Museum: A Interactive and Educational Experience through Google Cardboard

Yuhe Nie
yn2273
yn2273@nyu.edu

Abstract

Google Cardboard is a very cheap VR HMD. It can be purchased for less than $10. It is an alternative to providing a virtual education experience. However, the existing Google Cardboard Apps provide limited interaction for users and lack educational interactive and gamified applications that allow people to actively explore the virtual world and stimulate their interest in learning. In this project, we designed and developed a Virtual Renaissance Museum app that can be combined with Google Cardboard to provide fantastic VR experiences. It offers easy-to-learn inputs, uses high-performance rendering techniques, contains fun and rich interactions, and provides an immersive and free environment where users can virtually learn about Renaissance paintings and sculptures.

1. Introduction

The VR industry is growing rapidly. Mainstream VR devices now cost hundreds to thousands of dollars, and the software and ecosystem developed around VR are still limited. This makes people still deciding whether to buy VR devices. Google Cardboard [2] (provided by Google in 2016), which costs less than $10 and its software is purely based on existing mobile phone systems like Android and iPhone, provides an additional option for economically disadvantaged families.

However, many Google Cardboard Apps [3] only provide basic 3D texture environments and very limited text and voice interaction, which lack more features for users to explore and interesting interactive experiences actively. In this project, we developed a Virtual Renaissance Museum to explore the potential of Google Cardboard for better visualization, engaging interactions, and more content for users to experience.

We introduced global illumination and reflection probes to offer a high-quality environment. For a fluent user experience, we used only three inputs that required no additional physical devices (head rotation, eye gazing [5], and a button to tap and hold), and combined them to produce rich interactions. We also designed 4 specific sections that contain abundant content and knowledge.

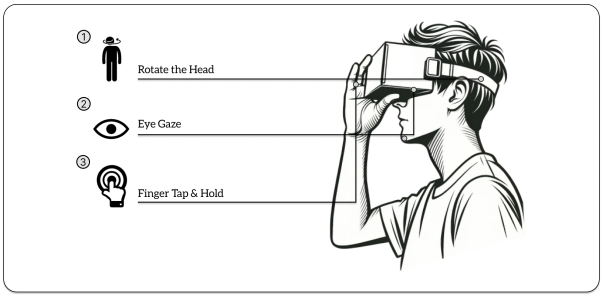

Figure 1. Supported User Inputs in the Renaissance Museum (1. Head Rotation, 2. Eye Gazing, and 3. Button Tapping and Holding)

2. Results and Demonstration

2.1. Inputs and Interactions

To enhance interactivity, we have considered using physical input devices, such as keyboards, mice, controllers, etc. However, after the experiment, a large number of users have indicated that they prefer to wear Google Cardboard directly, without the need to hold additional equipment. So we offered non-tangible input devices. As shown in Fig.1 shows, we support three inputs: **Head Rotation**, **Eye Gazing**, and **Button Tapping and Holding**.

We combine these inputs to provide different ways of interacting. We support the following interactive experiences:

- **Scene Roaming**: When the users turn their heads to look in a certain direction and press the button long, they slowly move in that direction. Release the button to stop the movement. Users can adjust the direction of their movement at any time.
- **Event Trigger**: When the users look at certain *interactive objects*, these objects display an outline indicating

that the user is firing an event for that object, at which point the user taps the Rebuffing button to trigger the event. (Different events will be elaborated in the following section.)

- **Access to Personal Album**: User can open their phone album, and they can select the photos they like and upload them to the app, which will load their photos in real time.

2.2. Renaissance Museum: Four Sections

As shown in Fig.2, we have developed a real-time graphics application: **Virtual Renaissance Museum**. It is mainly divided into four parts. Each section provides different interactions and content.

- **Section1: Renaissance Artwork**: In this section, we have specifically selected seven classic Renaissance paintings and two sculptures. Users are free to roam the scene and view these works. At the same time, users can interact with Trigger Events to get voice descriptions of these works.

- **Section2: Dive into Renaissance Period**: In this section, we've picked a video with a specific introduction to the Renaissance. When the user clicks on the screen using Trigger Events, the video plays. The user can pause or continue at any time.

- **Section3: It's your Time to be an Artist**: In this section, when the user clicks the empty frame in front of it using the Trigger Event, the user will jump out of the app, select a Renaissance painting that they have drawn in the album and upload it back to the app. The app will display the photo in a frame.

- **Section4: Quick Test**: In this section, the users are asked a question about Section 1's paintings and need to select the correct painting.

In contrast to other Google Cardboard educational apps, our app allows users to freely roam the entire scene, interact with various objects, and even present real things in a virtual museum. This increases the willingness of users to actively use our application to learn knowledge.

3. Implementation

3.1. Rendering and Visual Effect

For an immersive and realistic environment, we leverage the capability of **Global Illumination** [6] and **Reflection Probe** [1] (Fig.4). They are offline rendering methods in computer graphics that store rendering information in advance on a texture and directly sample the texture during rendering. This allows our application to perform no additional lighting calculations on the mobile side. To achieve smooth operation of the mobile phone at the same time, produce a higher quality picture.

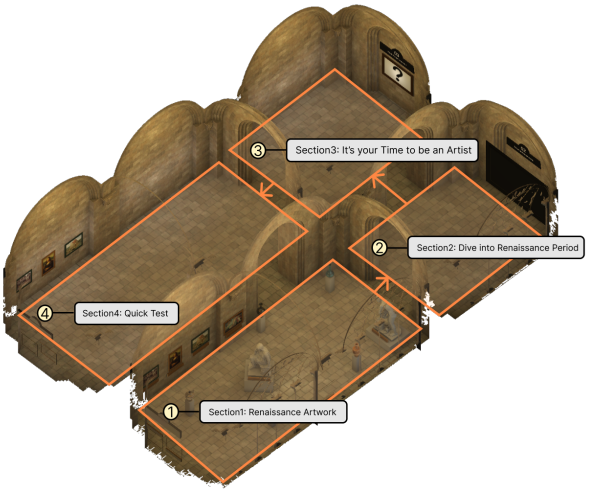

Figure 2. Virtual Renaissance Museum Layout.

Figure 3. Virtual Renaissance Section 1 Screenshot

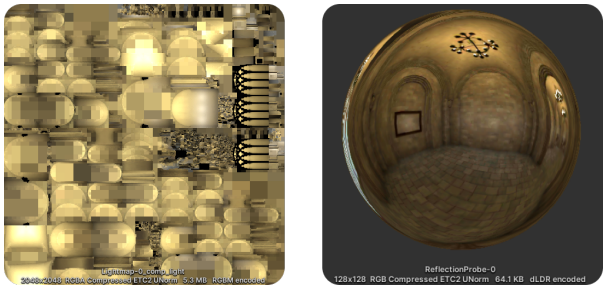

Figure 4. Global Illumination Texture (left) and Reflection Probe (right) in Unity

3.2. System Design

The whole script was written in C#. We include the unity **Input System** to tackle different input signals. For the interactable object, we integrated a unified event transceiver in-

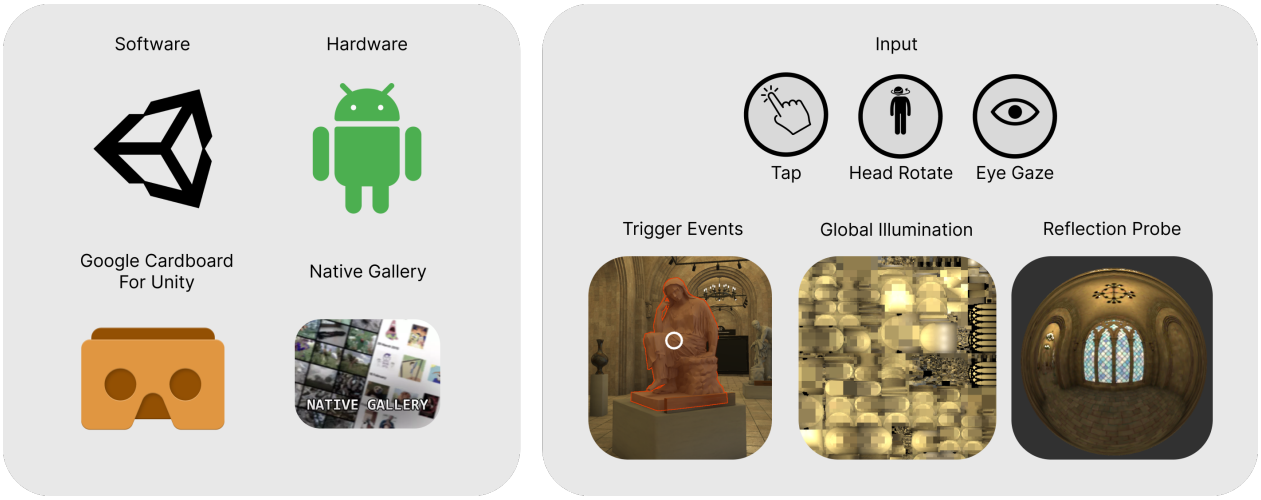

Figure 5. System Implementation Architecture. (The left block is the hardware, software, and SDKs we integrated. The Right block is input system, event system and rendering techniques we used)

terfaces **OnPointerEnter()** and **OnPointerExit()**. We also reported an existing bug to the Google Cardboard SDK, provided a sample repair script, and eventually fixed it. For most interactive objects, we use an **outline shader** to indicate that the 3D object is captured by the human eye, at which point the trigger event can be overwritten to trigger a different event. Events that can be triggered include audio introductions, playing and pausing videos, opening a user's photo album, quizzes, etc.

3.3. SDK Integration

In this project, we successfully integrated different SDKs into our project. These include the **Google Cardboard SDK for Unity** [4], which provides the functionality of dual eye view rendering and eye gazing detection, and **Native Gallery** [7], which offers mobile phone photo album access methods.

3.4. Configuration

The whole project was built by Unity 2022.2.12f1 and distributed on Android Moto G power. We offered an additional Google Cardboard for experiment purposes. The source code is available from this link.

4. Discussion

In this project, we tried to combine the normal workflow of Unity development with the Google Cardboard SDK to develop an educational and interactive museum app. We tried a variety of offline rendering techniques to enhance the graphics while still running smoothly on the phone. We have tried to design and develop a variety of interaction methods within the constraints of non-physical input devices. Future work may consider expanding the scale of the museum and adding more interactive and rendering effects. Such as using Shader to travel from a museum to a Renaissance world, users using Google Cardboard to draw works and sculptures in the app, and more 3D animation and interaction.

References

[1] Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. *Real-Time Rendering, Fourth Edition*. CRC Press, 2018. 2

[2] Google Cardboard arvr.google.com/cardboard/, 2014. 1

[3] Google LLC. Cardboard apps. https://play.google.com/store/search?q=cardboard%20apps&c=apps&hl=en_US, 2023. 1

[4] Google LLC. Google cardboard for unity. https://developers.google.com/cardboard/develop/unity/quickstart, 2023. 3

[5] Junichi Shimizu and George Chernyshov. Eye movement interactions in google cardboard using a low cost eog setup. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, page 1773–1776, New York, NY, USA, 2016. Association for Computing Machinery. 1

[6] Martin Thomas. Realtime global illumination techniques collection, 2016. 2

[7] yasirkula. Unitynativegallery: A native unity plugin to interact with gallery/photos on android & ios. https://github.com/yasirkula/UnityNativeGallery, 2023. 3