Software Requirements Specification

for

InArt VR Project (Ver 3.0)

Version 3.0

Prepared by Joseph Chong, Jimmy Hernandez, Edwin Hernandez, Jaquan Jones, Alberto Landeros, Tony Lee, Jennelle Maximo, Eduardo Meza, Dean Nguyen, Anthony Viramontes

The Institute for Interactive Arts, Research, and Technology (*InArt*)

December 10, 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
First Draft	10/22/2021	Initial Draft of Document	1.0
Second Draft	11/5/2021	Second Draft of Document	2.0
Third Draft	12/10/2021	Third Draft of Document	3.0

1. Introduction

1.1   Purpose

The purpose of this document is to define and explain the requirements of InArt's VR immersive storytelling game within each section. We outline the software requirements in the following sections, including the programs, libraries, and custom-designed software we incorporate. The requirements outlined in this document will enable a working demo of an interactive VR experience that conveys immersive storytelling and interactive capabilities.

1.2   Intended Audience and Reading Suggestions

This document is intended for developers who want to see an overview of InArt's development process, and how we improve or remove features from development. Project testers can take advantage to understand the system components and the expected outcome to test if it's functional. Advisors will be able to evaluate and oversee the project's expected outcome.

1.3   Product Scope

The purpose of the InArt VR project is to produce a VR game demonstration that immerses a new medium for audiences and for storytellers. The game will help advance storytelling techniques by immersing the user in virtual environments and by conveying a storytelling experience in which participants will have the chance to engage through gestures and physical actions that represent actions, i.e., miming and other gestures that represent a variety of interactions.

Once completed, this interactive VR game will be playable on a VR headset with controller peripherals. This interactive medium will allow users to experience an immersive storytelling experience in which they will be able to participate in and interact with through movement and gestures.

1.4   Definitions, Acronyms, and Abbreviations

Institute of Arts, Research and Technology (InArt): an interdisciplinary institute at California States University, Los Angeles to synthesize and synergize artistic research.

Virtual Reality (VR): an artificial environment that is experienced through sensory stimulation provided by a headset and hand controllers, and in which the user's actions partially determine what happens in the environment.

Augmented Reality (AR): an enhanced version of reality created by the use of technology to overlay digital information on an image of something being viewed through a device.

Player/User: The person using the game will be referred to as a player or a user.

SRS: Software Requirement Specification

Heads-Up Display (HUD): a heads-up display or status bar, which contains information that is visually displayed to the player during gameplay.

Motion Gestures: Specific gestures that will cause an in-game action, allowing the user to simulate real-life actions.

1.5 References

- Job Simulator: the 2050 Archives: Owlchemy Labs, Accessed 22 October 2021.

2. Overall Description

The virtual immersive story project is a cinematic-like experience designed in the Unity engine to run on a virtual reality system. The project will enable an audience to perform motion gestures and inhabit a virtual space curated by a team of creative storytellers and developers. A virtual reality system will be used for the user experience. This project will consist of a base level and three interactive levels that will expose the player to a unique perspective of the immigrant experience.

2.1 System Analysis

This section provides the general system requirements and project hurdles for an immersive virtual story with The Institute of Arts, Research and Technology (INART*). The software has an easy-to-use system interface, natural motion gestures, and human interactions inside a virtual computer-generated environment. The main objective of this project is to showcase a new medium for immersive storytelling utilizing the functions of VR. Electronic hardware made for the purpose of VR will enable an audience to feel as if they were physically in the story. Therefore, we can utilize this hardware to achieve new methods of discovering and interacting to give the audience a truly immersive experience.

- The user is involved in a cinematic-like experience in which a story is being told around them. This is in contrast to conventional linear storytelling in many games today that tells the audience what to do. We strive to implement an open-world experience with virtual reality. The Unity game engine has the necessary tools to build this software. VR and AR will be used to enhance the user experience so that the audience feels like they are proactively creating the story rather than following along.
- The development process will come with inherent hardware limitations and technical obstacles, which will be overcome with iterative development and user testing. The project has a small budget, therefore a limited number of VR headsets will be provided for development. An early major technical hurdle to overcome is the development team's limited experience with VR development and Unity. The team will invest a sufficient amount of time in learning how to use Unity game engine software and methods of VR development. The goal is to have the team be proficient in Unity and VR development in order to ensure project delivery.
- We will assign roles in the development team to efficiently work on the project. The team consists of 10 group members. Each member can be assigned to multiple roles depending on the project's need in order to speed up development. The roles we plan to establish are Project Lead, Level Designer Lead, Lead Artist, Concept Designer, Document Lead, Level Designer, Sound Engineer, and Programmer, QA Lead, and a Tester. Each person will have a role and will overcome any technical issues at hand.

- Non-technical hurdles include a limited development budget. Development hardware such as VR headsets will be organized via a sign-up sheet with a check-in-check-out system due to limited quantity. VR headsets will enable developers to proactively test the software for issues and feedback. A system will be set in place so any developer on the team that needs a VR headset will have access in order to not bottleneck development.

2.2   Product Perspective

In terms of gameplay and themes, we plan to implement an independent style of development using Unity. The game is not a continuation of any previous InArt projects. We are not taking any examples or ideas from any larger systems, as we plan to create this game through meetings and storyboarding for visual examples of what we want. We do have a game that inspires the way we plan to implement in-game mechanics. The gestures we plan to implement are inspired by a simulation game called Job Simulator: The 2050 Archives by Owlchemy Labs because of their use of gestures. Our intention is to allow a user to maximize the use of motion gestures. Job Simulator: The 2050 Archives successfully implements motion gestures allowing the user to grab and interact with anything in front of them. We want to enforce unique gestures to make the user feel they have an impact on the story being told around them. We chose unity because it allows us to immerse the player into the environment they are in.

2.3   Product Functions

The description below represents a summary of the major components in our VR game. They will be separated into sections based on the importance they hold in the game. There are five major components to our VR immersive experience:

- Check List System
- Pause Menu
- Artifacts to be transferred to a level
- Arm/Hand Gestures
- Objective Tracker

2.4   User Classes and Characteristics

- College students will be able to go through our game with ease, and will be able analyze the theme of immagration.
- College students will not have access until InArt releases the finished product.
- The advisor will be able to run the project at any time.
- The advisor will be analyzing how we develop gestures and the environment.

2.5   Operating Environment

We will be using multiple operating environments to develop and publish the project. The operating environments used are:

- Microsoft Windows 10/11
- Oculus Quest system software (based on Android version 10)

The project will be built on the Unity game engine running on Microsoft Windows 10 and published to Android using the Android SDK. The VR headsets and VR controllers will run on a proprietary Android-based software called the Oculus Quest system software.

For team collaboration software, we will be using Google Drive, Atlassian JIRA, and Unity Teams to ensure that all members have access to a shared project file.

2.6   Design and Implementation Constraints

There are hardware limitations to the virtual reality headset provided for development. We set the baseline for the experience using the hardware specification shown below for the Oculus Quest 2 Virtual Reality Headset.

Hardware specifications of Oculus Quest 2 Virtual Reality Headset:

- Qualcomm Snapdragon XR2 5G Platform (8-core, 8-thread, 2.5Ghz Turbo)
- Adreno 650 Graphics
- 6 GB LPDDR4X RAM
- LCD 1832 x 1920 per eye @ 120 Hz
- 4 Cameras, 6DOF inside-out tracking
- 64 GB storage

It is noted that the Oculus Quest 2 virtual reality headset has the ability to run games standalone and untethered from a host PC. If the Oculus Quest 2 is untethered from a host PC, the graphical power limitations of the Adreno 650, limited 2-3 hour battery life and smooth user experience will hinder the virtual reality experience if the game is not carefully designed to account for the hardware.

An Oculus Quest 2 tethered to a high-end PC shall lift the inherent limitations of running the game standalone as explained above. The limitations of this setup will be limited motion and movement due to physical wires and lack of vision. This compromises user safety and might risk disengaging the audience from the immersive experience if the user gets tangled in the wires.

Other safety issues and limitations will be potential dizziness and bumping into real objects in the area around the user.

2.7   User Documentation

- **Software Requirements Specification**
    - PDF Format
- **Software Design Document**
    - PDF Format
- **User Manual: User will have access to this**
    - PDF Format
    - Print Format (similar to a booklet found in video game cases)
- **Tutorial: User will have access to this**
    - Youtube Video

2.8   Assumptions and Dependencies

It is assumed that the user in the experience is in capable health to perform the intended motion gestures required through the experience. It is assumed the software and hardware used for the project are:

- Oculus Quest 2 Virtual Reality Headset
- Unity
- Microsoft Windows 10.0+
- Unity Teams

If any of the aforementioned requirements have changed or been updated, the requirements of this document will be updated to adapt to new requirements.

2.9   Apportioning of Requirements

A requirement that will be delayed is additional sublevels within one of our three main levels. For instance, in the farm level, we would like to have two possible endings based on if the player produces enough gestures in the last scene. This will be delayed unless we have extra time at the end of the time span to implement this.

3 levels of importance:

- Priority 1
    - Highest priority. This needs immediate attention and must be implemented. It must be verified and tested thoroughly.
- Priority 2

- ○ Not the most important priority, but requirements must be implemented and failure to meet expectations can result in an unacceptable state.
- ● Priority 3
  - ○ Lowest Priority. If expectations are met and all other priorities have been completed, then we can implement the lowest priority.

3. External Interface Requirements

3.1   User Interfaces

- **Class Level:** Upon accessing the VR experience, the user interface will display a school classroom.
    - User starting location, will preview a menu system on a chalkboard .
        - Will be able to change any settings we have to offer.
    - Will contain various artifacts/objects to transfer the player to one of three game levels.
- **Kitchen Level:** Artifact will be a bowl with a soup spoon inside.
    - It will start the player off in a kitchen.
    - They will have access to the menu option to leave the game at any time, or they can look at the checklist to complete objectives.
- **Farm Level:** Artifact will be a Family photo of the famer you are playing as.
    - It will start the player off by looking at the back of a pick up truck on a farm.
    - They will have access to the menu option to leave the game at any time, or they can look at the checklist to complete objectives.
    - This particular level will have a HUD for the boss.
    - There will be a red bar on top of the top left screen, which will signify how mad the boss is. If it's at 100%, the player will restart the level and all objectives complete will restart.
- **Festival Level:** Artifact will be a ticket used to play mini games.
    - It will start the player off in a festival.
    - They will have access to unique festival games.
- **Menu Screen:** Can be accessed in game to pause if needed.
    - The user will be able to pause the game when they are in control of the player.
    - The user will not be able to pause the game during a cutscene.

3.2   Hardware Interfaces

- The following VR devices are supported (e.g. Oculus Quest 2, HTC Vive)
- VR system with a head mounted display (HMD)
- VR system with two handheld controllers
- VR system with stereo speakers
- Audio peripheral if not utilizing headset's speakers
- Computer system with internet access

3.3   Software Interfaces

- Unity Game Engine
- Computer running Microsoft Windows 10+ or MacOS

3.4   Communications Interfaces

- Facebook account required, if the user is utilizing a Facebook owned VR headset, such as an Oculus Rift or Oculus Quest 2.

4. Requirements Specification

4.1 Functional Requirements

4.1.1 Functional Requirements

- The system shall be able to run smoothly on hardware used.
- The system shall be able to interpret the programmed motions gestures and perform the subsequent actions.
- The system should be able to offer two modes of in-game locomotion, so that the player can choose to be playing standing up or sitting down.
- The system shall be free of any game-breaking errors or bugs.
- The system shall have sound effects that add to the immersion.
- The system shall have at least 3 different stories that the player can play.
- The system shall have validity checks on any inputs performed by the player.
- The system shall have one output for every one input.
- The system should have an options menu so that the player can tweak the game to their preference.
- The system shall allow the player to interact with the environment.
- The system shall be able to respond to the player's input in a timely manner.

4.1.2 Environment

- The System shall have objects that the user can interact with
- The System shall load up scenes corresponding to the level

4.1.3 Sound

- The system shall allow the user to adjust the volume
- The system shall have sound corresponding to each level and environment.

4.1.4 Lightning

- The system shall allow the user to adjust the brightness

4.1.5 Inventory

- The system shall allow the user to store and retrieve items

4.2   External Interface Requirements

List of Inputs:

- VR headset to display visuals to user
    a) Need user height to properly display height in game so the user can get the best experience
- VR hand held controllers
    a) Each button on the remote will have actions and each input will have a determining factor in the game. E.g. Grabbing, trigger, ect.

4.3   Logical Database Requirements

Not applicable.

4.4   Design Constraints

- The system is limited to the power of the headset to be used.
- The system is limited to the time constraint of the class which is two semesters.
- The system is limited in what gestures are possible based on the headset and controllers utilized.
- The system is limited to using C#.
- The system is limited to the Users/Developer hardware in order to run the game smoothly.

5.   Other Nonfunctional Requirements

5.1   Performance Requirements

- The number of users to be supported is one since this is a single player, linear-story based video game.
- Most users will be able to use the game if they have either played games before, or if they have experience using VR.

5.2   Safety Requirements

- Safety for installing:
    - A Unity license and Unity Student plan are used to program the game. The user does not need to worry about the safety of their devices running our game.
- Safety for the User:
    - One important safety requirement is the playing area for the user's body movement. Since the game requires a Virtual Reality headset, we require the user to have a suitable playing area to prevent them from getting hurt. The most optimal space for the user is a couple feet from any wall or objects from all angles. Second, the headset must be properly strapped to the user's head to prevent irregular movement from the headset. Lastly, controllers must be strapped to the user's wrist to prevent exterior damage to the environment in case the controller slips from the user's grasp.

5.3   Security Requirements

Security requirements are entirely based on the software of the Virtual Reality headset. We have no control over the hardware the user decides to use. However, the user might be required to make an account with the respective VR headset company, such as a Facebook account for the Oculus Quest.

5.4   Software Quality Attributes

The software should be available to anyone with the right compatible products (Virtual Reality headset, smartphones, computers, etc.). The software product should be able to function with a smartphone, computer, or a virtual reality headset. The program created for the software may not be changed by the user, nor should it allow the authorization for the user to make any changes in the program or encrypt/add any modifications to the software. The software should be portable as long as there is a screen that can display the images of the software project. Having the right equipment that can operate with the supporting machine that displays the image.

5.5 Business Rules

User's will have full access to the features and will be able to download the software. Once it is Spring of 2022, the senior design team will turn in the finalized work to InArt.

6. Legal and Ethical Considerations

Appendix A: Glossary

- **Institute of Arts, Research and Technology (InArt):**
  - An interdisciplinary institute at California States University, Los Angeles to synthesize and synergize artistic research.
- **Virtual Reality (VR):**
  - An artificial environment that is experienced through sensory stimulation provided by a headset and hand controllers, and in which the user's actions partially determine what happens in the environment.
- **Augmented Reality (AR):**
  - An enhanced version of reality created by the use of technology to overlay digital information on an image of something being viewed through a device.
- **Player/User:**
  - The person using the game will be referred to as a player or a user.
- **SRS:**
  - Software Requirement Specification
- **Heads-Up Display (HUD):**
  - A heads-up display or status bar, which contains information that is visually displayed to the player during gameplay.
- **Motion Gestures:**
  - Specific gestures that will cause an in-game action, allowing the user to simulate real-life actions.

Appendix B: Analysis Models

This project does not have an analysis model.