

# Software Requirements Specification

for

## INART VR Project

Version 2 approved

Prepared by Kevin Diaz-Lopez, Taha Kamran, David Hermsillo, Zudong Li,  
Kiet Hoang, Jessy Francisco, Steve Galvan, Noah Castro, Daniel Ramirez

Institute For Interactive Arts, Research, & Technology (INART)

May 14th, 2021

# Table of Contents

- Table of Contents..... 1
- Revision History..... 2
  - 1. Introduction..... 3
    - 1.1. Purpose..... 3
    - 1.2. Intended Audience and Reading Suggestions..... 3
    - 1.3. Product Scope..... 3
    - 1.4. Definitions, Acronyms, and Abbreviations ..... 4
    - 1.5. References..... 4
  - 2. Overall Description..... 4
    - 2.1. System Analysis ..... 4
    - 2.2. Product Perspective..... 5
    - 2.3. Product Functions..... 5
    - 2.4. User Classes and Characteristics..... 5
    - 2.5. Operating Environment..... 5
    - 2.6. Design and Implementation Constraints..... 6
    - 2.7. User Documentation..... 6
    - 2.8. Assumptions and Dependencies..... 6
    - 2.9. Apportioning of Requirements..... 7
  - 3. External Interface Requirements..... 7
    - 3.1. User Interfaces..... 7
    - 3.2. Hardware Interfaces..... 9
    - 3.3. Software Interfaces..... 9
    - 3.4. Communications Interfaces..... 10

4. Requirements Specification.....	10
4.1. Functional Requirements.....	10
4.2. External Interface Requirements.....	12
4.3. Logical Database Requirements.....	12
4.4. Design Constraints.....	12
5. Other Nonfunctional Requirements.....	12
5.1. Performance Requirements.....	12
5.2. Safety Requirements.....	13
5.3. Security Requirements.....	13
5.4. Software Quality Attributes.....	13
5.5. Business Rules.....	13
6. Legal and Ethical Considerations.....	14
Appendix A: Glossary.....	14
Appendix B: Analysis Models.....	14
Appendix C: To Be Determined List.....	14

## Revision History

Name	Date	Reason For Changes	Version
First draft	11/10/20		1
Seconds draft	12/09/20	Adjustments to the User Interface section	2

# **1. Introduction**

## **1.1 Purpose**

The purpose of this document is to define a full set of requirements for each of the INART VR's compositional modules. This document will provide an exposition of the purpose and features of the system, what the system will accomplish, and what will be required of this system. These requirements are to be used in the implementation of the software written as a part of this project.

## **1.2 Intended Audience and Reading Suggestions**

This document is intended for:

- Developers who can utilize this document to easily overview how to approach development and improve or add features.
- Project testers can review this document to know which system components to test and to see if they meet the required functionality.
- Liaisons who wish to overview the functional and nonfunctional requirements to determine if the product is being developed to meet their expectations and to further add or adjust requests.
- Advisors will propose this document for evaluation to determine if the document is sufficient.

## **1.3 Product Scope**

The INART VR Project is a 3D photo-realistic setting to be used by film students as their production sets utilizing virtual reality. INART VR will also be a filmmaking tool in which producers will be able to automate camera movements and easily adjust environment lighting. INART VR will create a virtual world and characters to film an interactive book. The actors and producers will be able to interact with their virtual environment and virtual characters.

## 1.4 Definitions, Acronyms, and Abbreviations

The acronyms included in this document are:

**INART** - Institute for Interactive Arts, Research, & Technology

**VR** - Virtual Reality

**SRS** - Software Requirement Specification

**UE4** - Unreal Engine 4

## 1.5 References

This application references Unreal Engine 4, an advanced real-time 3D creation tool.

- <https://docs.unrealengine.com/en-US/API/index.html>
- <https://www.udemy.com/course/unrealcourse/>
- <https://www.youtube.com/watch?v=1mihkGXuj8U>
- <https://www.unrealengine.com/marketplace/en-US/product/hotel-lobby-and-bar-modular-interior>
- <https://www.unrealengine.com/marketplace/en-US/product/unimodels-lamps>
- <https://www.unrealengine.com/marketplace/en-US/product/29d5a3af319748c78b10e8d3991014b7>

# 2. Overall Description

## 2.1 System Analysis

- a. A goal of this project is to create multiple 3D realistic movie sets for our client to use. This project also is involved with creating objects, characters, environments and cameras within the virtual world for the client to move and record scenes with.
- b. Starting out with this project we had our team get used to the Unreal Engine and create a basic world for the user to go around in. Implementing other objects into

the world and changing the lighting proved to be one challenge as well as trying to use a green screen in the virtual world. Implementing Ai for the virtual characters was another challenge that arose. Chroma keying actors into the virtual world was another obstacle as lighting reflections needed to be accurate with the actors' lighting.

- c. After some thorough research and tutorials, we were able to find a hidden setting in unreal, that controlled the universal lighting of the asset. Finding this setting was proven to be difficult but made the entire project much easier as lighting is the main component of the project to ensure it is realistic. This menu included being able to find the setting for the glare that was continuously ruining certain shots in the city.

## **2.2 Product Perspective**

This project fully utilizes The Unreal Engine by Epic Games to create the software needed. The virtual sets and camera are all constructed within the Unreal Engine software. With the entire project revolving around the use of Unreal Engine the interface is largely that of the engine.

## **2.3 Product Functions**

The main functions of our program are:

- Creating a virtual world where multiple films sets can be made from
- Create a camera object in which our client can move and record from inside the virtual world/set
- Import and Export video files
- Create a virtual character with automated movements and interactions
- Create multiple objects that the actors and the engine can interact with
- Have the camera be set on a course for pan shots or moving

## **2.4 User Classes and Characteristics**

- Actors can import their recordings into the environment.
- Production staff can customize the environment to fit their film (adjust lighting, change objects, etc.) and effectively record utilizing multiple cameras and camera movements.

## **2.5 Operating Environment**

The INART software will only reside within the Unreal Engine. Since this software project is running through a separate program there aren't any other hardware systems included. This system can be flexible in terms of operating systems used.

## **2.6 Design and Implementation Constraints**

When using the Unreal Engine the developers do have a hardware limitation in running the program. Developers need access to a computer with a graphics card to render and create the virtual sets within the engine. Since the interface of Unreal Engine allows any developer to use the product, there are no higher-order language requirements for our project. The use of Unreal Engine to produce virtual scenes and sets can set some memory constraints, however those constraints are fairly low. Another limitation is the financing of desired 3D components that can be used in the environment. Implementing AI into the virtual world also proved difficult as we needed to add different animations for walking and moving. Other applications may include Microsoft Visual studio where developers can use C++ to add to their project in Unreal Engine.

## **2.7 User Documentation**

**(Documentation that we give to Users not what we used)**

<https://www.udemy.com/course/unrealcourse/>

<https://www.youtube.com/watch?v=1mihkGXuj8U>

<https://docs.unrealengine.com/en-US/Engine/Editor/index.html>

<https://www.epicgames.com/site/en-US/privacypolicy?lang=en-US>

<https://www.unrealengine.com/en-US/eula/publishing>

## **2.8 Assumptions and Dependencies**

This product assumes the following:

- The user has access to a computer with the required hardware to run UE4.
- The user has access to video and audio equipment.
- The user has access to chroma key technology.
- The user has access to the internet.

## **2.9 Apportioning of Requirements**

Not available at this time.

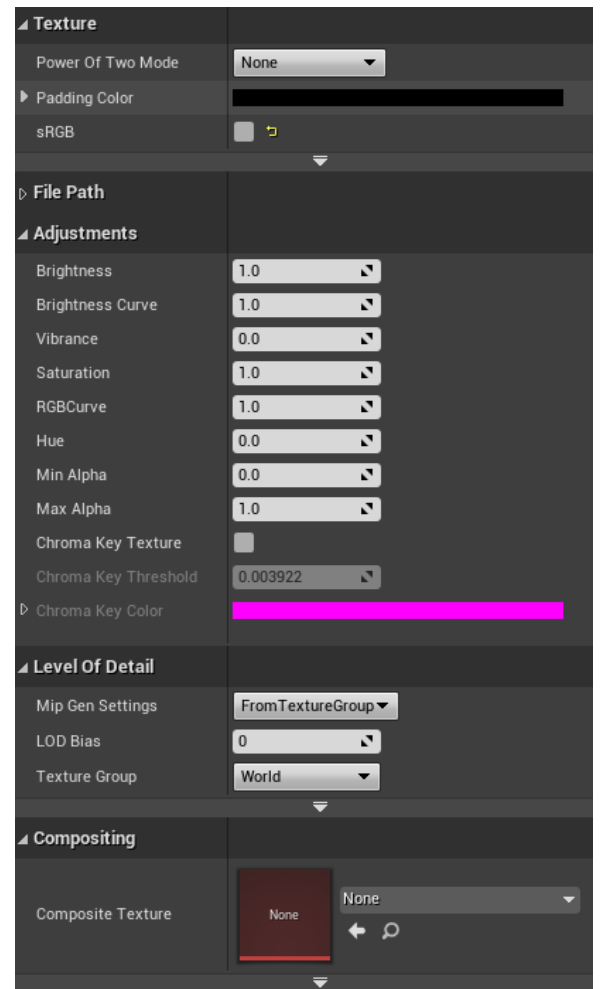
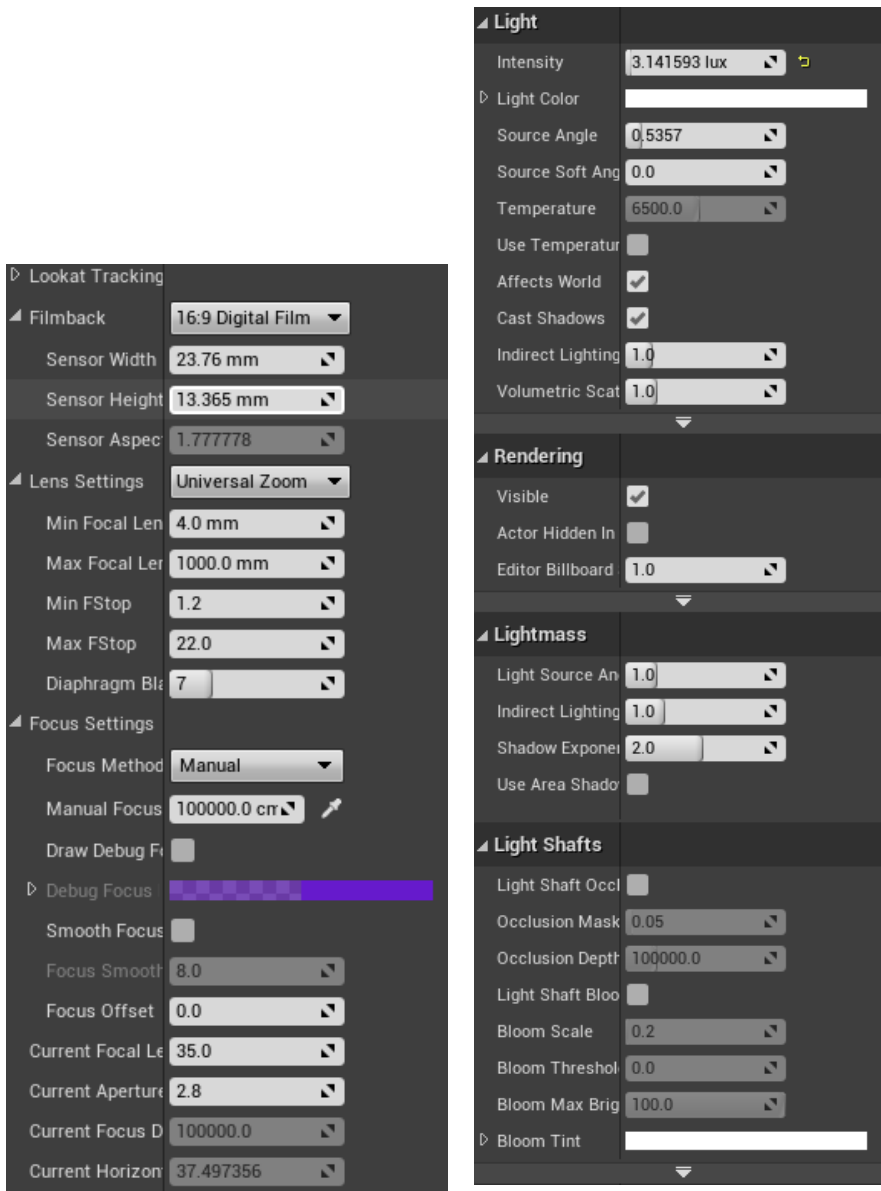
# **3. External Interface Requirements**

## **3.1 User Interface**

The user interface will users to access the following functions:

- Camera settings that allow adjustments to; camera depth, camera focus.
- Lighting settings that allow adjustments to; adjust lighting, adding lights.
- Environmental settings that allow adjustments to; adjusts environment textures.
- AI settings that change the movement of the virtual characters and other virtual objects





## 3.2 Hardware Interfaces

The supported devices are devices that run on Windows 7 64-bit or Mac OS X 10.9.2 or later. A functional standard keyboard and mouse in order to operate the system. The application will be able to interact with camera, audio, and lighting equipment.

## 3.3 Software Interfaces

The main software is the Unreal Engine version 4.25.3 to create the virtual backgrounds and sets. The Unreal Engine is installed through the Epic Games launcher and installs 2 prerequisites which are DirectX and VisualC++ redistributables. The Epic Games launcher requires users to sign in before launching the software. The use of OBS Studios is also included for capturing videos.

### 3.4 Communications Interfaces

INART VR will require internet connection in order to update new plugins and components in Unreal Engine.

## 4. Requirements Specification

### 4.1 Functional Requirements

4.1.0	<b>Virtual Character</b>
4.1.1	The system shall be able to place virtual character(s) in the virtual set
4.1.2	The system shall allow the user to change the character's position and other body parts individually
4.1.3	The system shall allow the user to animate the character
4.1.4	The character(s) shall be adjustable in scaling
4.1.5	The character should be able to be interacted with the user and actor
4.1.6	Virtual characters shall be capable of interacting with the user through artificial intelligence

4.2.0	<b>Recording</b>
4.2.1	The system shall allow screen recording
4.2.2	The system shall allow the client to move the recording camera within the software
4.2.3	The system shall allow the use to import video
4.2.4	The system shall allow the user to import audio

4.2.5	The system shall allow livestream recording
4.2.6	The system shall allow the use of multiple cameras
4.2.7	The system shall allow the user to switch between multiple cameras
4.2.8	The system shall have the option to chroma key videos in the virtual set

4.3.0	<b>Camera Movement</b>
4.3.1	The system shall allow camera movement on the X-axis
4.3.2	The system shall allow camera movement on the Y-axis
4.3.3	The system shall allow camera movement on the Z-axis
4.3.4	The system shall allow camera rotations
4.3.5	The system shall provide automated camera movements
4.3.6	The system shall allow the option to change camera movement speed
4.3.7	The system shall allow the camera to track a virtual object.

4.4.0	<b>Lighting</b>
4.4.1	The system should have the option to transfer the virtual environment's lighting onto the actors
4.4.2	The system shall allow the user to adjust the environment's lighting
4.4.3	The system shall allow the user to change light sources
4.4.4	The system shall allow the user to change the SkySphere
4.4.5	The system shall allow the user to add more light sources
4.4.6	The system shall allow the change of reflection of lighting

4.5.0	<b>Environment</b>
4.5.1	The system shall create objects that the actors can interact with
4.5.2	The system shall create objects that the client can interact with
4.5.3	The system shall allow the user to import levels

## 4.2 External Interface Requirements

For the INART system the only input we are receiving are video livestreams. This input is from the user side so that they can use the virtual set as the background. In return the output for the system is another video file but with the virtual objects and sets included.

## 4.3 Logical Database Requirements

Not applicable.

## 4.4 Design Constraints

- Design implementation is limited to C++ in order to work with UE4.
- Some users/developers have limited hardware capabilities that are required to smoothly run UE4 and load assets.
- Some developers lack camera and/or chroma key equipment in order to effectively test the system.
- Developers lack funds for potential assets that can be used in the environment.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

- The system must be simple to use for non-technical users such as actors, directors, etc.
- The environment must be a photo-realistic 3D setting,

- The system shall start users off in a camera perspective viewport.

## **5.2 Safety Requirements**

Since Unreal Engine is a licensed product that any developer and user can use there are no actions needed to be taken safety wise.

## **5.3 Security Requirements**

The Unreal Engine does require an account made from Epic Games and is only applicable to the developers and users that would like to edit in the virtual world. Users that receive the footage from the Unreal Engine aren't required to log into the engine unless edits or recordings are needed. As per the Unreal Engine privacy policy, any data collected from the user will be used to improve the Engine Code. Users do have the option to turn off that feature under the License.

## **5.4 Software Quality Attributes**

- Portability
  - The Unreal Engine project will be able to be transferred and downloaded to other users and retain all software features.

## **5.5 Business Rules**

All features are to be accessible to any user. As per the Unreal Engine policy, any revenue made from the use of this software includes a 5% royalty to Epic Games.

## 6. Legal and Ethical Considerations

### Appendix A: Glossary

**Chroma key** - A technique in which a block of color can be replaced by a video image. Often done through a green screen.

**3D** - Three dimensional

**UE4** - Unreal Engine 4

**OBS** - Open Broadcast Software

### Appendix B: Analysis Models

