Software Requirements Specification

for

VRCar

Version 1.0 approved

Prepared by Pallak Singh

Manik Kumar

Vinay Garg

Vineet Jain

The NorthCap University

26 April 2018

Table of Contents

Table of Contents iii

1. Introduction 1
   1. Purpose 1
   2. Document Conventions 1
   3. Intended Audience and Reading Suggestions 1
   4. Product Scope 2
   5. References 2
2. Overall Description 2
   1. Product Perspective 2
   2. Product Functions 2
   3. User Classes and Characteristics 3
   4. Operating Environment 3
   5. Design and Implementation Constraints 3
   6. User Documentation 3
   7. Assumptions and Dependencies 3
3. External Interface Requirements 4
   1. User Interfaces 4
   2. Hardware Interfaces 4
   3. Software Interfaces 4

Appendix A: Glossary 6

List of Abbreviations 6

# Introduction

## Purpose

The purpose of this Software Requirement Specification document is to provide a detailed description of the VRCar system requirements, design constraints, and functionalities. Through this document, the workload for development, validation and verification will ease.

## Document Conventions

IEEE standard 830 template for System Requirement Specification Documents is used to create this document.

In this document, flow charts are used to illustrate the specific details in various sections. Hand drawn illustrations are also provided as part of a ‘paper prototype’ of the mobile application.

Bold fonts are used to highlight the headings and the subheadings.

1. Headings:

Font – Bell MT, bold  
 Size - 18 pt

2. Subheadings:

Font – Bell MT, bold

Size - 14 pt

3. Content:

Font – Bell MT, regular

Size - 12 pt

## Intended Audience and Reading Suggestions

The following group of people have been identified as the target audience for this system:

* Gamer - Gamers can use the project to simulate a game environment in real life, for example: real life racing games between multiple similar cars where the gamer would have a more immersive and a real physical experience.
* Scientist - Scientists would be able to use to this explore alien or not easy to reach places with an immersive experience and further produce data for further research by adding customized sensors or add-ons as the designs will be open source.
* Differently Abled People - Differently abled can experience the world around them with little to no movements on their part

## Product Scope

There are a variety of applications that require quick and real time action-response in the world of robots. We require rapid information with the most meticulous observations from the robots sent in the remote areas we cannot reach ourselves. We propose to bring an immersive experience to make information gathering more efficient and lifelike. The main focus of this project is to create an economical system that combines a robotic car with integrated VR so that such devices can be used widely in such fields. VR focuses on creating an interactive and immersive experience that engages the body and mind. This project will utilize virtual reality for just this: an immersive and interactive experience.

## References

[1] SRS Document Template, IEEE.

[2]GitHub link: <https://github.com/mnk400/VRcar>

[3] Documentation for UV4L(video streaming service): <http://www.linux-projects.org>

[4] Yufeng Shan, UC Berkeley. Cross-Layer Techniques for Adaptive Video Streaming over Wireless Network 2005:2, 220–228

[5] Jingwei Huang, Zhili Chen, Duygu Ceylan, Hailin Jin. “6 DOF VRVideos with a 360 Camera”, March 18-22, 2017, Los Angeles, CA, USA 978-1-5090-6647-6/17

[6] Nadine E. Miner, Sharon Stansfield, “An Interactive Virtual Reality Simulation System for Robot Control and Operator Training”, Intelligent Systems and Robotics Center: <https://pdfs.semanticscholar.org/8bda/99c02224678484fa7d76020397f36eb1eb54.pdf>

# Overall Description

This section gives background information about specific requirements of the system to be developed detail. This section will describe the factors affecting the final product.

## Product Perspective

There exist certain areas in the world that are inaccessible to humans due to their dangerous or remote nature. Scientists require a way to gather information from such places and collect precise observations for research purposes. A system that provides them with an immersive experience of the field will allow them to make meticulous observations of the field as if they were there in person.

## Product Functions

1. Virtual reality experience of real time environment via a robotic car.
2. Control the directions using joystick movement.
3. Control the speed and accelerate using controller.
4. Control the directions using head movements.

## User Classes and Characteristics

1. Users with physical disabilities to explore their environment without having to move from their positions and only moving their heads.
2. Game developers to further explore this system to develop VR games for the people who prefer not to move around while playing.
3. Programmers who are interested in this project to further expand it and include more features.

## Operating Environment

1. Windows 8
2. Windows 10
3. OS X 10.11 El Capitan
4. macOS 10.12 Sierra
5. macOS 10.13 High Sierra
6. Debain Linux
7. iOS 10.0 and above

## Design and Implementation Constraints

1. VRCar is developed in Swift and obj-C using Xcode. The mobile application will only be available for iOS users.
2. The mobile application stores user information using core data, which means that the data is stored locally on the device itself. It cannot be transferred between different devices.
3. Latency and jitter
4. The experience may vary for users based on the type of hardware they are using, i.e VR headsets and controller as well as the functioning of the accelerometer, magnetometer and gyroscope.

## User Documentation

A user manual will be available on the official website and the GitHub page considering all aspect of user related tasks and general maintenance of the same. The user manual will include how to set up the car and the VR while navigating through the various user settings on the application.

A requirements page will be available for the user to gather the appropriate hardware.

## Assumptions and Dependencies

1. We assume that the users will already be informed of the basic functionality of the product.
2. VRCar is developed for iOS devices and therefore requires the users to have an iOS device with iOS 10.0 or higher and sufficient knowledge to operate the smartphone.
3. The user must have sufficient knowledge as to how to fix the smartphone to their VR headset and operate the VR headset and controller.
4. Latency and jitter does not occur
5. The user must not suffer from motion sickness in case of accelerated robot car speeds.

# External Interface Requirements

## User Interfaces

User interface will include a mobile application with view controllers that include the home screen that swipes to the display that shows the current view of the camera. A settings page allowing users to change image settings and camera settings such as screen size, lens separation, vertical center, chromatic and outer distortion. The user will have the option to save their settings that will be stored locally on their phones.

## Hardware Interfaces

1. The user will be wearing the virtual reality headset and experience the scenario provided by the camera mounted on the car.
2. We will be using the gyroscope, accelerometer and magnetometer of our smartphone to orient and control the speed of our robot car.
3. We are utilizing an Arduino board mounted to our robot car to send orientation data from our phone to the car,
4. Our raspberry pi will be powering our Arduino and facilitate the communication between the controller with the user and the robot car.
5. A game controller joystick will be used to orient the car and acceleration.
6. OV7670 low voltage CMOS VGA camera to be mounted on the Arduino to send image to the smart phone application.
7. DC 6V 4-wheel Robot Smart Car Chassis Kits car with Speed Encoder for Arduino with tachometer encoder.

The system will be using a VR headset with a smartphone with the mobile application fitted into it. The user will be handed the controller or the keyboard to control the robot smart car. The built-in gyroscope and accelerometer of the smartphone will be taken advantage of and employed to orient the robot car with the head movement of the user.

## Software Interfaces

1. VR headset is worn by the user to make the user reach places virtually using a robotic car.
2. MQTTis a publish-subscribe based messaging protocol. It is designed for connections with remote locations or the network bandwidth is limited. We will be using MQTT to send data from our arduino to our mobile phone and vice versa.
3. We will be using core data as the framework for storing user information locally on the user device.
4. The programming for the robot car orientation and speed will be done in Python and Arduino C while the mobile application is coded in obj-C and Swift.

**4. System Features**

**4.1 System Feature 1**

4.1.1 Description and Priority

The Mobile/PC software will provide the on-board computer on the car with sensor data so that the car could interpret the data and turn the car based on users head movements, it will also process the VR video and have network connectivity so that it could be transmitted to the user end.

4.1.2 Stimulus/Response Sequences

While setting up the product, the user will be greated with UI that will allow them to set the initial settings for the car like connecting the on board computer to the network. The controller will allow the user to control the car and the head movements of the user will change the direction in which the car is moving.

**5. Other Nonfunctional Requirements**

**5.1 Safety Requirements**

The hardware should be treated carefully as extensive damage to the hardware might hinder the end user experience, functionality of the VR car might also be affected because of the damage. Driving very fast while wearing the VR headset might induce dizziness in the users body, hence longer sessions or sessions with very fast driving should be avoided.

**5.2 Security Requirements**

The video transmitted should be over secure channels such that anybody connected to the same network would not be able to intercept the video.

Appendix A: Glossary

1. **Virtual reality headset:** A virtual reality headset is a head-mounted device that provides virtual reality for the wearer. VR headsets are used in simulators and trainers. They comprise of a stereoscopic head-mounted display, stereo sound and head motion tracking sensors. They may also include eye tracking sensors. The user will be wearing this headset and experience the scenario provided by the camera mounted on the car.
2. **Gyroscope:** A gyroscope is a device used for measuring or maintaining orientation and angular velocity. The gyroscope measure the rate of change of a particular axis at the current moment in time. This means that to keep track of our angle, we need to sum all of the rates of change over a given period of time. We're essentially looking for the integral of our gyro data.
3. **Accelerometer:** An accelerometer is a device that measures proper acceleration.
4. **Arduino:** An Arduino is an open-source micro controller developer board that can be used to read sensors and control things such as motors and lights. It allows you to upload programs to this board and hence, interact with things in real world and make them respond. We are utilizing an Arduino board mounted to our robot car to send orientation data from our phone to the car,
5. **Raspberry Pi:** A raspberry pi is a low cost, credit-sized computer that plugs into a monitor, and uses a standard keyboard and mouse. Our raspberry pi will be powering our Arduino and facilitate the communication between the controller with the user and the robot car.
6. **Game Controller:** A game controller is a device used with games or entertainment systems to provide input to a video game, typically to control an object or character in the game. A [joystick](https://en.wikipedia.org/wiki/Joystick) is a [peripheral](https://en.wikipedia.org/wiki/Peripheral) that consists of a handheld stick that can be tilted around either of two axes and (sometimes) twisted around a third. The joystick is often used for [flight simulators](https://en.wikipedia.org/wiki/Flight_simulator).
7. **VR:** VR is a computer-generated scenario that simulates a realistic experience. The immersive environment can be similar to real world in order to create a lifelike experience or grounded in reality. We aim to use a VR headset to make the user reach places virtually using a robotic car.
8. **MQTT:** MQTT is a publish-subscribe based messaging protocol. It is designed for connections with remote locations or the network bandwidth is limited. We will be using MQTT to send data from our Arduino to our mobile phone and vice versa.
9. **Core Data:** Core Data is a framework that you use to manage the model layer objects in your application. We will be using core data as the framework for storing user information locally on the user device.
10. **Latency:** Time taken for a single video frame to from the camera to display.
11. **Jitter:** Video jitter occurs when horizontal lines of video image frames are randomly displaced due to corruption of synchronization signals during video transmission.

List of Abbreviations

|  |  |
| --- | --- |
| VR | Virtual Reality |
| IoT | Internet of Things |
| MQTT | Message queueing telemetry transport |
| GUI | Graphical User Interface |