**VR Headset**

Software Design Document

Names:

Raafat Mazruq

Ido Guzi

Lab Section: Final Project

Date: January 2021

**TABLE OF CONTENTS**

[1. INTRODUCTION 1](#_Toc61627772)

[1.1 Purpose 2](#_Toc61627773)

[1.2 Scope 2](#_Toc61627774)

[1.3 Overview 2](#_Toc61627775)

[1.4 Reference Material 2](#_Toc61627776)

[1.5 Definitions and Acronyms 2](#_Toc61627777)

[2. SYSTEM OVERVIEW 3](#_Toc61627778)

[3. SYSTEM ARCHITECTURE 3](#_Toc61627779)

[3.1 Architectural Design 3](#_Toc61627780)

[3.2 Decomposition Description 3](#_Toc61627781)

[3.3 Design Rationale 4](#_Toc61627782)

[4. DATA DESIGN 4](#_Toc61627783)

[4.1 Data Description 4](#_Toc61627784)

[4.2 Data Dictionary 4](#_Toc61627785)

[5. COMPONENT DESIGN 5](#_Toc61627786)

[6. HUMAN INTERFACE DESIGN 5](#_Toc61627787)

[6.1 Overview of User Interface 5](#_Toc61627788)

[6.2 Screen Images 5](#_Toc61627789)

[6.3 Screen Objects and Actions 5](#_Toc61627790)

[7. REQUIREMENTS MATRIX 6](#_Toc61627791)

# INTRODUCTION

## Purpose

. This software design document describes the architecture and system design of Virtual Reality headset that will allow every low budget gamer who want to take there gaming experience to the next level but have limited budget and PC specs.

## Scope

Due to being open source the product that can be modified to the user desire unlike other VR headsets on the market, from hardware to software the user will be able to know exactly what is going on and to have full control over the device.

## Overview

VR is the way to give a user more immersed experience, by using a VR headset the user will see the game/video in the same manner we see the world (3d) with depth perspective.

The VR headsets that is on the market today are highly priced, and require a good PC to be able to play game with VR, but in order to have this experience they have to pay too much money for the best VR headset.

Our project goal is to make a VR headset with a limit budget and can be adjusted to the PC specs of a user.

## Reference Material

*This section is optional.*

None for now

## Definitions and Acronyms

*This section is optional.*

OS – Operating System

VR – Virtual Reality

API – Application Programing Interface

PC – Personal Computer

Driver – a driver is a software component that lets the operating system and a device communicate with each other.

# SYSTEM OVERVIEW

Hardware – monitor, 3d orientation sensor.

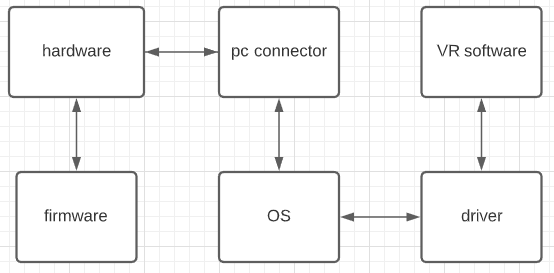
Firmware – arduino transferring and receiving hardware data and game data

Software (driver) – converting data to a game format.

# SYSTEM ARCHITECTURE

## Architectural Design

* Hardware – display, 3d orientation sensors.
* Software – drivers and interface to communicate with the hardware
* Testing environment – basic program to test and valuate the state of the product.
* Linked hardware – PC to connect the product to.
* Linked software – the end software (video game) the product should work with.



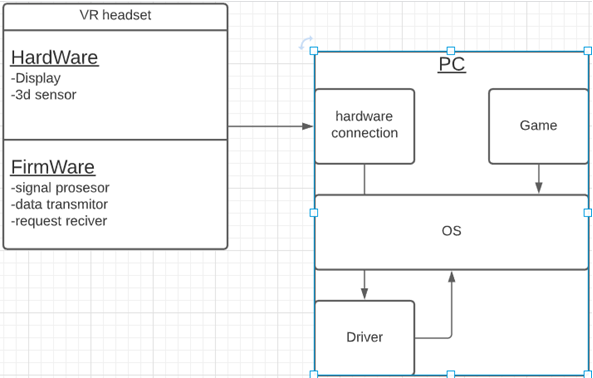
## Decomposition Description

Hardware will have the basic firmware installed in order to start the main process,

And will be connected to PC via Wire cable.

The driver will be running as part of the OS, receiving a sending data from the Hardware and process it.

VR supported Games will communicate with the drivers for the hardware data.



## Design Rationale

The structure of product is the standard of embedded systems today.

# DATA DESIGN

## Data Description

3d headset orientation data – real time data transformation from the headset to the PC then to the drivers and VR software

Headset settings – setting of the headset will be managed by the drivers and stored as a file in a specific location on the PC.

Drivers will receive the data from the headset process it accordingly to a specific API and will be sent to VR software asking for the data

## Data Dictionary

Driver

* Up/Down tilt – float
* Left/Right tilt – float
* Side tilt - float

FirmWare

* Up/Down tilt – float
* Left/Right tilt – float
* Side tilt - float

# COMPONENT DESIGN

Driver – not implemented yet (will update with development)

FirmWare – not implemented yet (will update with development)

In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.

# HUMAN INTERFACE DESIGN

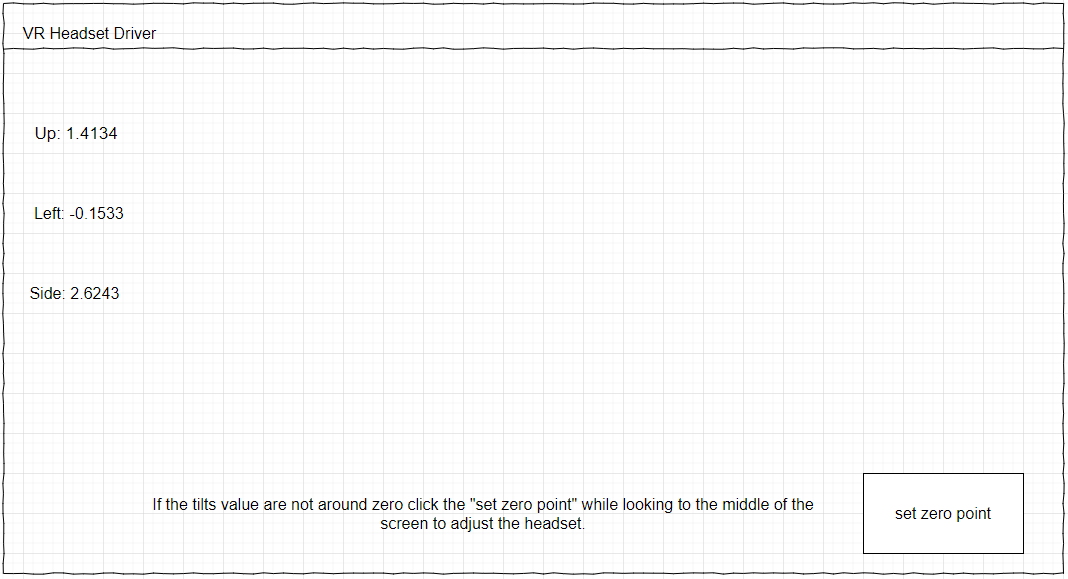
## Overview of User Interface

Connecting the device – the device should be connected to a PC with 1-2 cables.

Drivers – after installing the drivers, the user should be able to set the zero point (the dead center location when looking towards the screen), and can open a test to make sure the device works.

## Screen Images

How the driver mainly should look



## Screen Objects and Actions

Driver button – set zero point: adjust the zero point of the headset to the current orientation.

# REQUIREMENTS MATRIX

|  |  |
| --- | --- |
| requirements | components |
| 3d orientation detection of the headset | 3d sensor |
| Demo to present the abilities of the headset | Testing environment |
| Support in VR games | drivers |
| Calibration software | drivers |