Requirements for Noctilucent VR

Taylor Fahlman, Joshua Bowen, Adam Puckette

Abstract

There are 4 main deliverables for this project. Creating an OpenGL test for the OSVR framework; writing a proposal to either use CloudCompare or other software; a simple documentation/summary of the final product; and the final product, where an OSVR headset can interact with the software.

1 Introduction

1.1 Purpose

The purpose of this addon is to allow any user of the open source software CloudCompare to be able to view point cloud data in Virtual Reality. The user, however, must be in possession of some Virtual Reality headware. In this regard this project is designed for anyone who utilizes CloudCompare for it's open source cloud data viewing capabilities, and is looking for a more immersive analysis of this data.

1.2 Scope

Noctilucent VR will function as an extension or addon of the already existent open source software CloudCompare. When utilized, this software will take the point cloud data as rendered by CloudCompare and display it in a fashion suitable for virtual reality. The software will render and update the display of the data in real time. Auxillery devices will also be usable to navigate the data. Benefits of this software include the viewing of point cloud data as it was intended to be viewed, in 3D. Our goal is to make this software as easy to use and keep a consistent framerate. While viewing the data the software should not stutter, should draw complete scenes, and should render millions of points. As point cloud data often contain millions of points it is important that this software is able to render as many points as possible.

1.3 Definitions

Free and Open Source Software: Also known as FOSS, is any software that can be classified as both free software and open source software. This means that anyone is freely licensed to modify, copy, or study the code as they see fit. users are encouraged to improve the software and share their work with others.
Virtual Reality: Also referred to as VR, is an existing technology which allows the user to put on a headware and have an immersive experience in which turning your head turns the camera and moving your head moves the camera. Additionally, visuals are rendered seperatly for both eyes allowing for different perspectives which when put together create a 3D environment.
CloudCompare: Also referred to as CC, is a FOSS program freely available to users who need to view point cloud data.
Open Source Virtual Reality: Also referred to as OSVR, is an existing Open Source VR headset and framework. OSVR is designed so that it can be developed on and modified by the the public.
Oculus Developers Kit 2: Also called the DK2, is another existing VR headset, however it is not open source and has since been replaced with a standard production model.
Graphics Processing Unit: Referred to as a GPU, is the piece of hardware inside of a computer responsible for performing microprocesses often associated with graphics rendering or deep learning.

1.4 References

The following links are our references: http://www.danielgm.net/cc/ http://www.osvr.org/

1.5 Overview

The user of selected software will be able to plug in an OSVR-supported headset and view their cloud point data in a 3D VR view. They will be able to examine and manipulate the data in 3D space. They will be able to rotate and zoom, crop, etc. large data sets at a constant framerate, ideally 60 FPS, with 30 as the minimum.

2 Overall Desciption

2.1 Product Perspective

2.1.1 User Interfaces

Noctilucent VR will make a mild modification to the current CloudCompare interface by adding an option to activate VR mode. When activated CC will begin rendering data for VR instead of for a mintor. When in VR mode, CC will utilized a different interace from the default version. The new user interface will be designed with VR in mind

and will only appear at the need of the user upon the push of a button. The VR Interface will include functionality to easily navigate the data and mark areas of interest.

2.1.2 Hardware Interfaces

Noctilucent VR will interface with the GPU of the computer to render graphics. //NOTE: I don't know what this will look like

2.1.3 Software Interfaces

Name: CloudCompare Mnemonic: CC Specification Number: Version Number: Source.
CloudCompare will be the software that handles the actual analysis and input of the cloud data information. In relation to our software product we will take what exists already with CloudCompare and add the ability for the software to render the data for VR.

2.2 Product Functions

Noctilucent VRs Functions are as follows

2.3 User Characteristics

Someone using this software is expected to already be familiar with point cloud data and softwares used to navigate it. Familiarity with CloudCompare is expected before jumping intot he VR aspects of it. This might be a users first experience with VR if they purchased it for this purpose.

2.4 Constraints

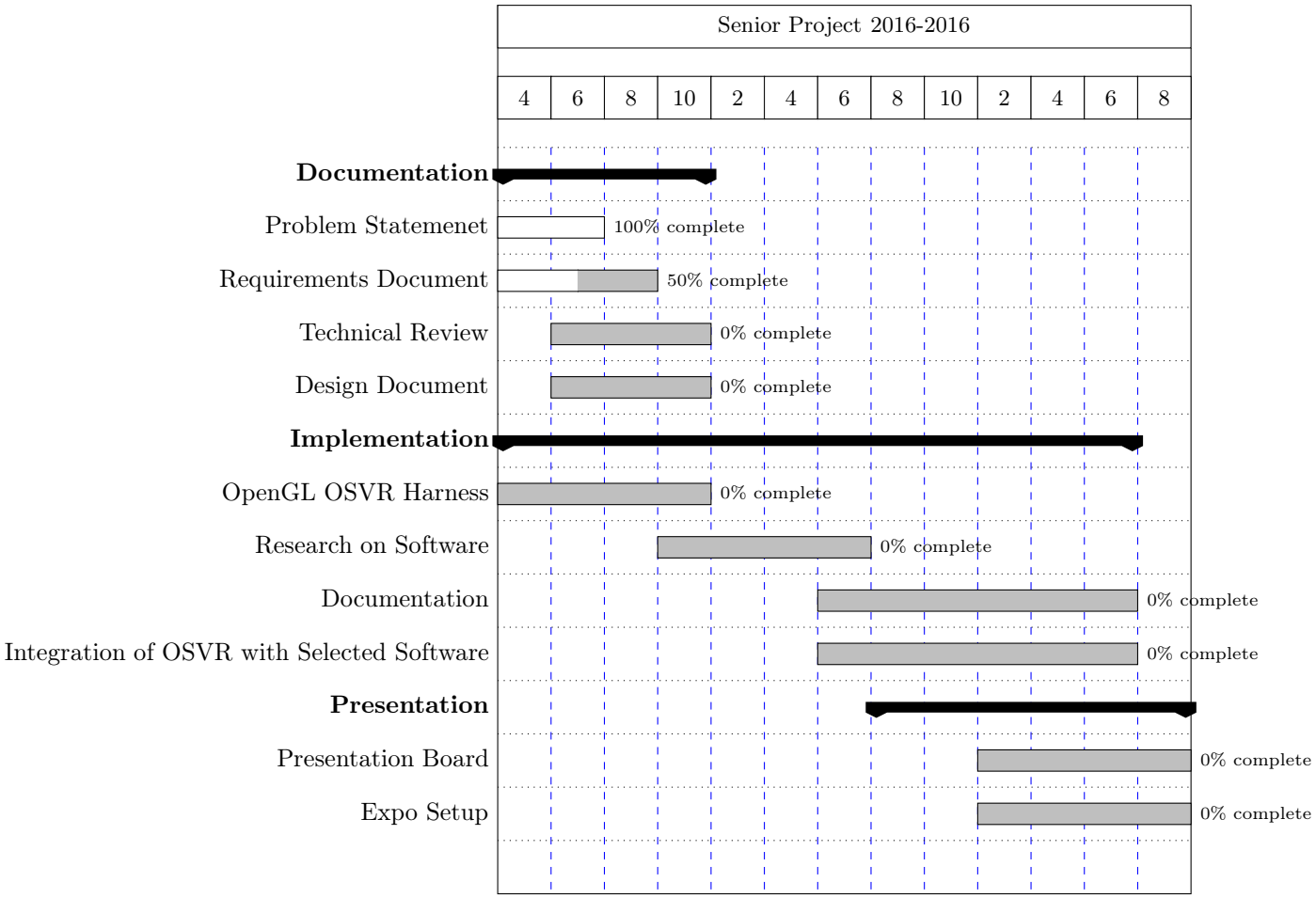
The possible constraints mostly rely in the way CC uses OpenGL. There have been repoted issues with the software in which stuttering was found with larger data sets. With this possible limitation already present in 2D view, enabling 3D support would likely make the situation worse. However, the lead developer has stated he will help in this case, and some of these problems may be able to be fixed. In the case that theses cannot be fixed, there is also a linux-only research program which can be used instead that already supports 3D viewing, but wi

2.5 Assumptions and Dependencies

It is our assumption that CC will be capable of rendering a million or more points of cloud data in realtime. In the event that CC is not capable of such we will need to considering creating a seperate software instead of an extension.

3 Specific Requirements

Specifically, an OpenGL program written to test the OSVR framework will be needed. This will simply involve an OBJ viewer in OpenGL, and some 3D OBJ file to view.
Reasearch will be needed on the constraints of OpenGL. Specifically, the viability of CC as the medium of delivery will need to be assesed. A comparison against the research linux-based software will be included in this reasearch.
Once a solution is decided on, a simplified documentation of the implementation will be written. This will address what parts of the architechture was needed to implement this change, as well as setting and other relevant information for future developers and users of the final product.
Finally, either a plugin or change to the core code will need to be written and presented to the client, as well as the reposity or stakeholder of the software we choose. The ideal situation is that the finaly product would be integrated into the core of the software. However, an external plugin would suffice.


Senior Project 2016-2016
4
6
8
10
2
4
6
8
10
2
4
6
8
Documentation
Problem Statemenet
100% complete
Requirements Document
50% complete
Technical Review
0% complete
Design Document
0% complete
Implementation
OpenGL OSVR Harness
0% complete
Research on Software
0% complete
Documentation
0% complete
Integration of OSVR with Selected Software
0% complete
Presentation
Presentation Board
0% complete
Expo Setup
0% complete