UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS 301 Capstone Project 2017

Vulknut Software Engineering

Compiled By

Peter Boxall - u14056136
Claude Greeff - u13153740
Marin Peroski - u13242475
Johan du Plooy - u12070794
Bernhard Schuld - u10297902

EPI·USE®
LABS
value through innovation

3D VR Presentations

GitHub Repository: Valknut Software Engineering

Contents

1 System Requirements and Design

1.1 Introduction

1.1.1 Purpose

This document serves to outline the overall description and requirements of the system. This document also serves as a guideline to the developers in order to ensure the final product meets these requirements, and indicates to the client what the required technologies are in order to be able to use this system.

1.1.2 Scope

The overall objective of this project is to provide any given user with a toolkit, with which the individual can create a 3D virtual reality presentation with ease. Our goal is to make it simple to use, enabling virtually any user to utilize the power of 3D, without having to build 3D objects completely from scratch. The user would custom build a 3D environment built upon a variety of available templates offered, or by selecting a set of 3D models and skyboxes when choosing to create a project from the ground up, taking user experience to a whole nother level.

1.1.3 Definitions, Acronyms and Abbreviations

1.1.3.1 MEAN

MongoDB, Express.js, AngularJS (or Angular), and Node.js

1.1.3.2 VR

Virtual Reality

1.1.3.3 MVP

Minimum Viable Product

1.1.3.4 MTBF

Mean Time Between Failures

1.2 Design

1.2.1 Software Methodology

We will follow the Agile development methodology. The principles this methodology is based on advocates planning, constantly evolving development, early delivery and continues improvements, and it encourages flexibility as well as maintainability.

The agile development process is built on four main principles:

1. Individual and team interactions over processes and tools.

2. Working software over comprehensive documentation.

3. Customer collaboration over contract negotiation.

4. Responding to change over following a plan.

The Agile development approach allows for frequent opportunities for clients to be involved in. Requirements are then reprioritized according to client specifications and they are elaborated on. The process of Agile development is based on the following actions:

- Short timeboxes of iterative development.

- Early and repeated client/user feedback.

- Reprioritization of work based on the client/user so that emergent requirements can be handled.

- Selecting a specific approach of which there are a variety of options including, Extreme Programming, Scrum, Lean Development, and Feature-Driven Development.

Some of the benefits of using the Agile development include stakeholder engagement, transparency, early and predictable delivery, predictable costs and schedule, allows for change, focus on the client, and ultimately improving the quality of the software.

For the above-mentioned reasons, we had chosen to utilize the Agile Software Methodology as it was the most applicable satisfying our needs as well as our client's.

1.2.2 Development Technique

During our first meeting with EPI-USE they had mentioned that we should make use of a development technique called MVP. A MVP is the most basic version of a product that can still be released. The point of this technique would be that early adopters would see the potential that the final product could offer, and give developers valuable feedback needed to guide them forward.

1.3 System Requirements

The Virtual Reality presentation software will have various requirements that will need to be fulfilled in order to deliver a viable product.

1.3.1 Functional Requirements

The following functional requirements will be met:

1.3.1.1 360 video export

The software must be able to export a presentation as a 360 degree video. The reason for this exporting format is to allow for a viewable format that will be as universal as possible to all VR capable devices ranging from mobile phones to VR headsets.

1.3.1.2 VR Device Viewable

The software must be able to be viewable and intractable with the use of a Virtual Reality headset such as the Oculus Rift and HTC Vive. A user must be able to preview his or her virtual reality presentation within a virtual reality environment if they own the appropriate hardware as well as edit the environment in virtual reality space.

1.3.1.3 Selectable Skyboxes

There needs to be a variety of skyboxes that a user can select from in order to match the theme of the environment. The skyboxes also need to be compatible with the Unity engine in order to be incorporated into the scene properly.

1.3.1.4 Template Environments

Template environments must be available to the user. These environments will vary to allow for a range of scenes to be catered for. A small set of template environments will be created by the development team and the rest by the community (community driven software approach).

1.3.1.5 Various Object Imports Into Scene

A variety of objects will need to be available for placement in the scene. These objects will need to be compatible with the Unity engine as well as relate to some possible environment. Users should be allowed to import their own objects as well as objects created by the community.

1.3.1.6 Editor Pont of View

The placement and control of the point of view of the user who is currently editing the scene (whether using a VR device or not) is vital. The user will need to be able to either walk on the terrain of the environment or be able to fly in the environment for better perspective of the presentation.

1.3.1.7 Native Interface Design

The interface for the software will need to run in the Windows environment as the application is being developed for the Windows operating system. It will also need to be easy to use and navigate.

1.3.2 Non-Functional Requirements

The following non-functional requirements will be met:

1.3.2.1 Image Importing

Images the creator wants to display in the scene will need to be able to be imported into it. Various image formats such as .jpg's, .png's and bit map images will need to be catered for, eliminating a formatting issue.

1.3.2.2 Slideshows

Imported images must be able to be incorporated into a slideshow or multiple slideshows. These will allow the creator to describe a topic in detail through various images that relate to the subject matter.

1.3.2.3 Video Importing

Videos allow for a rich and detailed explanation of the topics the user is trying to cover. The software must be able to incorporate .mp4 video formats into the presentation that can be viewable in the virtual reality space.

1.3.2.4 Audio Importing

The software will need to be able to incorporate audio snippets or songs into the scene. The required format type that needs to be catered for will be an .mp3 file. Voice recordings will also fall part of the audio file requirement as they are an important asset that can be incorporated into the presentation. This will allow the creator to direct a user's attention to a certain part of the scene. It will also allow the creator to describe parts of the presentation to the viewer. The software must be able to import voice recordings users make and save with third party applications.

1.3.2.5 Community Driven Content

The software will need to be able to allow users to create their own content and upload it to a central server which can then be viewed and downloaded by any user who wishes to use the created content. This will cover environmental scenes, 3D objects, presentation templates, skyboxes and other Unity based assets that are supported by the software.

1.4 Target Audience Characteristics

Our first focused audience would be targeted at the educational sector. Our initial goal would be to provide a toolkit for an individual to build a basic educational scene.

1.5 Constraints

There are several constraints needed to be taken into consideration.

Platform constraints:

- Mono, an open source development platform based on the .NET Framework. Mono's implementation is based on the ECMA standards for C# and the Common Language Infrastructure.

- For development:

  - Windows 7 SP+1, 8, 10; Mac OS X 10.8+.

- For running Unity applications/games (depending on the complexity of the project):

  - Windows XP SP2+, Mac OS X 10.8+, Ubuntu 12.04+, SteamOS+.

Device hardware constraints:

- Graphics card: DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.

- CPU: SSE2 instruction set support.

Video size:

- The exported video should be a realistic size, taking bandwidth and cap into consideration.

Community content needs to be a reasonable size (in community guidelines):

- Contributing to the complexity of a project will increase exported video size.

Community content needs to be relatively optimized (in community guidelines):

- Again, contributing to the complexity of a project will increase exported video size.

Other constraints that will be considered and in which further research will be conducted as implementation progresses include:

- Possible VR device constraints with regards to environment editing.

- Fixed set of templates.

- Unity assets only for community driven content.

1.6 Technologies

In our first meeting with EPI-USE we had discussed the use of various technologies. They had given us "free will" with regards to what technologies to use. After we had committed ourselves to extensive research we had selected the following, but did not limit ourselves to:

- Creating a 3D environment to design and bring to life a 3D presentation.
- Unity 3D virtual reality system tool kit library.
- HTC Vive virtual reality gear (already available).
- Import external models.
- Using plug and play libraries.
- Possibly include library templates for uses to build on.
- Community driven approach.
- Windows 10 environment.
- Docker.
- TravisCI.

2 Test Reports

2.1 Internal: Testing With Team Members

2.2 Usibility Testing

2.3 Examples of Target Audiences

2.3.1 Educational: Eg School Teachers

2.3.2 Corporate: Eg Board Members at a Project Proposal

3   User Manual

3.1   Introduction

3.2   Getting Started

3.3   Quick Start

3.4   Main Scenarios of Use

3.4.1   Features You Can Expect

3.4.2   Examples of Use

3.5   System Requirements

3.5.1   Minimum Requirements

3.5.2   Recommended Requirements

3.6   FAQs