

Soapbox VR



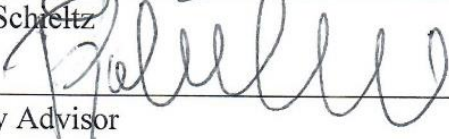
by

Colin Crowell and Brian Schieltz

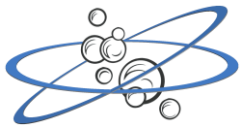
Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2017 Colin Crowell and Brian Schieltz

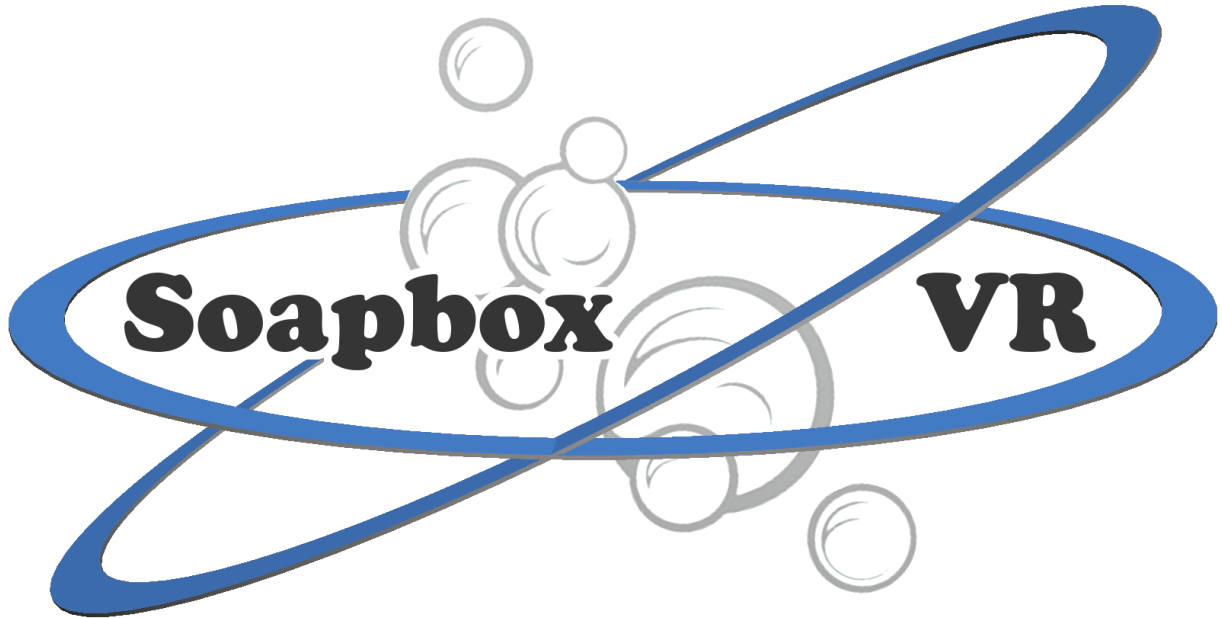
The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

 _____	<u>4/17/17</u>
Colin Crowell	Date
 _____	<u>4/17/17</u>
Brian Schieltz	Date
 _____	<u>4/17/17</u>
Faculty Advisor	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
April 2017



The University of Cincinnati
2600 Clifton Ave, Cincinnati, OH 45220
Tel (513) 556-6000
www.uc.edu



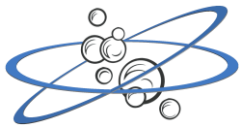
Prepared for:



The University of Cincinnati
2600 Clifton Ave
Cincinnati, OH 45220

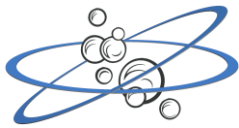
Prepared by:

Colin Crowell, Brian Schieltz
Students of University of Cincinnati
College of Education, Criminal Justice, and Human Services
cech.uc.edu/it

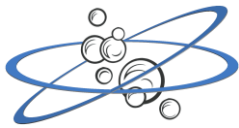


Contents

<u>Chapter</u>	<u>Page</u>
Abstract	1
1. Problem Statement	2
1.1. Introduction	2
1.2. Project Description	2
1.3. Problem	3
1.4. Solution	4
1.5. User Profile	5
1.5.1. Application Title	5
1.5.2. Potential Users	5
1.5.3. Software and Interface Experience	5
1.5.4. Task Experience	5
1.5.5. Frequency of Use	5
1.6. Technical Elements	6
1.6.1. Distribution	6
1.6.2. Software	6
1.6.3. Phone Storage	6
1.6.4. Server Storage	6
1.6.5. Application Install	6
1.7. User Interface	7
1.7.1. Categorical View	7
1.7.2. Thumbnail List View	7
1.7.3. Video Player View	7
1.7.4. Security	8
1.8. Use Case Diagram	9
2. Project Management	10
2.1. Project Timeline	10
2.2. Budget	11
3. Application	12
3.1. Overview	12
3.2. Configuration File	13



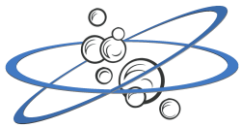
3.3.	Key Code Fragments	14
3.3.1.	Loading XML Into Object	14
3.3.2.	Object Rotation	16
3.3.3.	Full Screen Display Switching.....	16
3.5.	Screenshots	17
3.6.	Referenced Code	20
4.	Video Production.....	21
4.1.	Overview	21
4.2.	GoPro Omni	21
4.3.	Parallax.....	22
4.4.	Video Capture	22
4.5.	Audio Capture	23
4.6.	Stitching	23
4.7.	Editing	25
4.7.1.	Color Correction	25
4.7.2.	Graphics Insertion.....	28
4.8.	Publishing.....	30
5.	Testing	31
5.1.	Overview	31
5.2.	Scope	31
5.3.	Objectives.....	31
5.4.	Hardware Requirements.....	31
5.5.	Control Procedures.....	32
5.5.1.	Problem Reporting	32
5.5.2.	Change Requests.....	32
5.6.	Features to Be Tested	32
5.7.	Features Not to Be Tested	33
5.8.	Entry and Exit Criteria	33
5.9.	Logging Test and Reporting.....	34
5.10.	System Testing	34
5.11.	Testing Procedures	35
5.12.	Pass/Fail Conditions	35



5.13.	Schedule of Team Member Testing.....	35
5.14.	Schedule of Public Testing.....	36
5.15.	Risks / Assumptions	36
6.	2017 Tech Expo.....	37
6.1.	Overview	37
6.2.	Expo Posters.....	37
6.3.	Survey Results.....	40
6.4.	App Downloads.....	44
7.	Conclusion.....	45
7.1.	Overview	45
7.2.	Lessons Learned.....	45
7.3.	Skills Developed	47
8.	Bibliography	48
9.	Appendix	49
9.1.	Testing Results	49
9.1.1.	Bug List.....	49
9.1.2.	Table Key.....	49
9.1.3.	Testing Cases	50
9.1.4.	Test Results – Brian Schieltz	51
9.1.5.	Test Results – Colin Crowell	52
9.1.6.	Test Results – Experts.....	53
9.1.7.	Test Results – Novices.....	54

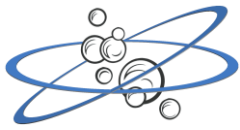
Tables

<u>Chapter</u>	<u>Page</u>
Table 1. User Profile.....	5
Table 2. Milestone and Deliverable Timeline.....	10
Table 3. Project Budget.....	11

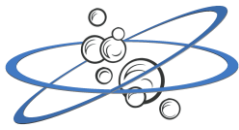


Figures

<u>Chapter</u>	<u>Page</u>
Figure 1. Application Intro Screenshot.....	3
Figure 2. Use Case Diagram.....	9
Figure 3. Gantt Chart Fall Semester 2016.....	10
Figure 4. Gantt Chart Spring Semester 2017.....	10
Figure 5. Application Full Screen View.....	17
Figure 6. Application Virtual Reality View.....	17
Figure 7. Video Selected.....	18
Figure 8. Video Intro With University Branding.....	18
Figure 9. Video Playback.....	19
Figure 10. GoPro Omni.....	21
Figure 11. Parallax.....	22
Figure 12. Color Harmonization.....	24
Figure 13. Stitching Error.....	24
Figure 14. Before Color Correction.....	26
Figure 15. After Color Correction.....	26
Figure 16. Lumetri Color.....	27
Figure 17. Waveform (RGB) Before.....	27
Figure 18. Waveform (RGB) After.....	27
Figure 19. White Balance.....	28
Figure 20. Flat Map View.....	29
Figure 21. Corrected Map View.....	29
Figure 22. Main Expo Poster.....	38
Figure 23. Supplemental Expo Poster.....	39
Figure 24. Survey Question 1.....	40
Figure 25. Survey Question 2.....	40
Figure 26. Survey Question 3.....	41
Figure 27. Survey Question 4.....	41
Figure 28. Survey Question 5.....	42
Figure 29. Survey Question 6.....	43
Figure 30. App Downloads From Expo.....	44



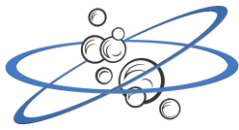
Soapbox VR
April 2017



Abstract

Soapbox VR is a 360° video platform that allows viewing in full screen or virtual reality. Virtual reality and 360° video are new marketing techniques, able to enhance the public message for a wide variety of industries. Virtual reality can take people to places and situations they could not normally go, from the fifty-yard line of a football field to the international space station.

Creating stand-alone 360° videos often does not get enough exposure, and creating a custom app is expensive. Only large companies can justify the cost. With Soapbox VR, any business can create a branded application customized to their specific needs, and published to all major platforms. Soapbox VR makes this possible with minimal technical expertise required and without the need for programming. Both large companies and small businesses can create branded apps with hundreds of videos, in custom categories and layouts, for a unique implementation each time.



1. Problem Statement

1.1. Introduction

Virtual reality is the act of wearing a headset that encompasses your field of vision, giving the illusion of being somewhere else. This can be a digital environment or a pre-recorded video.

The device tracks the users head movement, which allows the user to look in any direction he, or she desires. Specialized video recording rigs can be used to record in every direction at once.

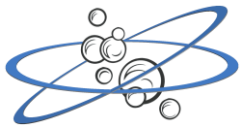
The University of Cincinnati, founded in 1819, is located in Ohio and ranked in the top 50 largest universities in the United States. UC is comprised of five campus locations, including Clifton, Medical, Clermont, Blue Ash and East Campuses. In addition to over 70 universities in Ohio and the surrounding area, UC competes with universities around the world to enroll students. UC's enrollment for 2016 was 44,338 with 76.7% being Ohio residents.

1.2. Project Description

We designed a 360° video platform, Soapbox VR, that allows customers to create a branded app unique to their business or organization. Both the content and layout are customizable.

Developed in Unity, the platform can be distributed to Google Cardboard, Google Daydream, iOS, Samsung Gear, Oculus, and HTC Vive.

Utilizing the Soapbox VR platform, we created an Android application that takes users on a virtual tour of the University of Cincinnati campus using 360° video. Our solution allows



viewers to watch the tours as standard videos by moving their tablet or smartphone, or by using Google Cardboard and entering a virtual environment. The university is able to use these videos and Android application for marketing purposes, reaching a wider audience of students. This includes individuals who may be incapable of visiting campus such as international and out-of-state students. Reaching these people presents a new way of increasing enrollment.

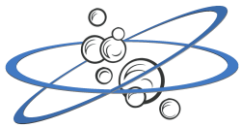


Figure 1. Application Intro Screenshot

1.3. Problem

Virtual reality and 360° video is expensive to create and distribute. Often, even for large companies, the benefits are not enough to justify the project, and for smaller businesses, the cost can be outright unaffordable. While large companies have the technical expertise to create it, their time is better spent elsewhere. Small companies routinely lack the resources and departments even to consider it.

The problem UC faces is that new student enrollment never ends. Every year students stop enrolling or graduate, meaning that new students must be found to replace them. UC Blue Ash

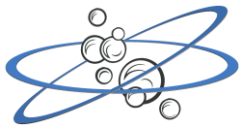


and UC Clermont specifically compete against Clifton for first year students. The videos also become a way to demonstrate the technical acumen of the university, reinforcing that students will get a quality education.

1.4. Solution

In order for companies of all sizes to create 360° video applications, there must exist a platform that is programming free and customizable. The platform must be cheaper than both in-house development and outsourced programming. Soapbox VR fulfills both of these requirements by being inexpensive and only requiring the configuration of an XML file to create the app.

The Assistant Director for International Admissions, Charlie Schwartz, believes that the best recruiting tool UC has is its campus; named by Forbes magazine as among the world's most beautiful. Once you see the campus, you will fall in love with it. 360° Virtual Tours, on the Soapbox VR platform, makes the campus more accessible to students who could never see it before. Prospective students will get a close up view of the dorms, campus buildings, and student events from the comfort of their home. Whether they are in another state, another country, or simply starting out in high school and not ready to make a college choice, UC will be there showing commitment.

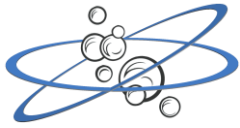


1.5. User Profile

Anyone interested in the University of Cincinnati is a potential user. This will most often be potential students or university faculty trying to recruit or fund raise.

1.5.1. Application Title
Soapbox VR
1.5.2. Potential Users
UC Recruiters, UC Marketing, UC Admissions, Current UC Students, Potential UC Students
1.5.3. Software and Interface Experience
Android app users will be presented with an introductory screen showing the categories: Main Campus, Medical, Blue Ash, and Clermont. Users will choose a category and then be presented with a list of videos. Tapping on a video thumbnail will start the video. The user can then watch the video, or press the Google Cardboard button and place the phone into a Cardboard headset.
1.5.4. Task Experience
Experience with standard mobile applications. Experience with the Google Cardboard headset is optional.
1.5.5. Frequency of Use
UC Staff will have repeated use as they continue to show prospective students and partners. Other users will have limited use, expected to be the viewing of 3-5 videos over 1 or 2 sessions.

Table 1. User Profile



1.6. Technical Elements

1.6.1. Distribution

The app will be made available on the Play Store.

1.6.2. Software

The videos can be viewed from almost any platform that can play YouTube and Facebook videos. PC, Mac, Android, iOS, and Windows Mobile.

1.6.3. Phone Storage

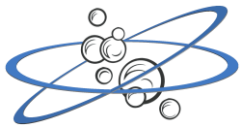
The best quality experience requires downloading the videos to the phone before watching. This means being mindful of video size, where the videos are downloaded to (SD card if available), and the ability to delete videos after viewing.

1.6.4. Server Storage

There is a server to store the videos for download by the app.

1.6.5. Application Install

The app is downloaded from the Play Store.



1.7. User Interface

1.7.1. Categorical View

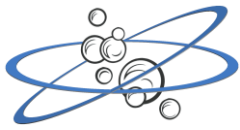
Upon starting the application, the user steps into a virtual space surrounded by category thumbnails. These categories are customizable, unique to every implementation. The universities categories will be for the different campuses and other major categories such as football. Upon selecting a category, the user is transported to another location, surrounded by thumbnails of videos in that category. Each video will have the option to stream or download, and once downloaded, the option to play or delete.

1.7.2. Thumbnail List View

Upon selecting a category, the user is transported to another location, surrounded by thumbnails of videos in that category. Each video will have the option to stream or download, and once downloaded the option to delete. After choosing to stream or play the video, the application will redirect to the video player.

1.7.3. Video Player View

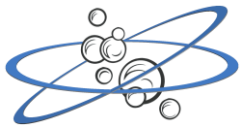
The video player view contains a video player widget that can be manipulated by the user, with controls such as pause and play. Cardboard view will split the video into stereoscopic 3D viewable through Google Cardboard or a similar VR headset for Android devices. Information about the location in which the video was recorded is displayed underneath the video player widget.



1.7.4. Security

The application is designed in such a way that no user will be required to enter any information.

This will prevent any unwanted code injection from being posted back to the content delivery server. The configuration file also allows for the addition of hash codes and file sizes for security checks.



1.8. Use Case Diagram

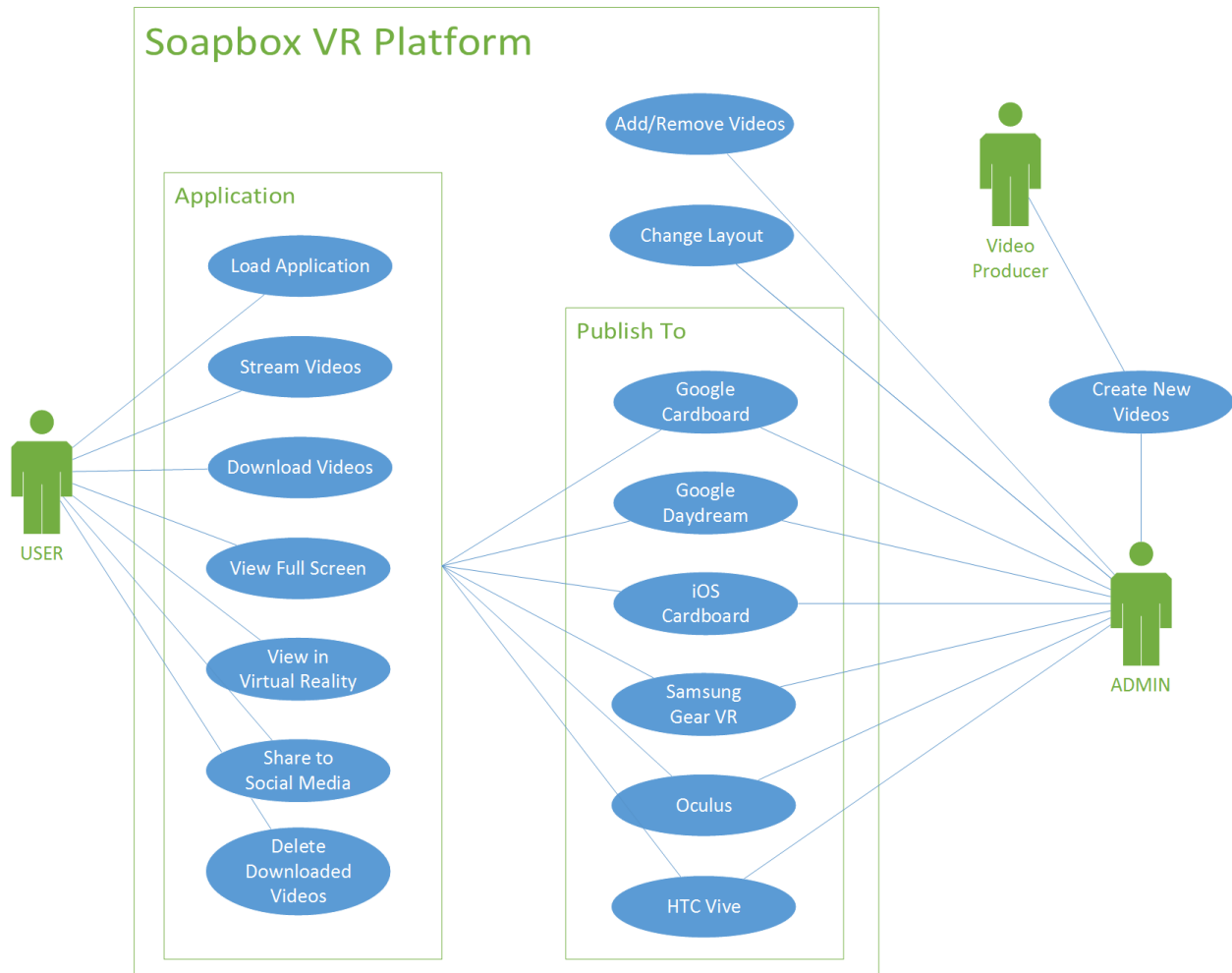
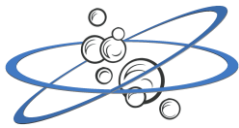


Figure 2. Use Case Diagram



2. Project Management

2.1. Project Timeline

Milestones & Deliverables			
Pre Planning Milestone	9/5/2016	Presentation Deliverable	11/14/2016
Team Building Milestone	10/1/2016	Final Draft Report Deliverable	11/28/2016
Obtain Camera Milestone	10/14/2016	Tour Videos Installment 2 Deliverable	1/31/2017
Software Alpha	10/28/2016	Software Testing Milestone	2/28/2017
Tours Videos Installment 1 Deliverable	10/31/2016	Tour Videos Final Installment Deliverable	3/31/2017
Software Working Beta Deliverable	11/7/2016	IT Expo Deliverable	4/11/2017

Table 2. Milestone and Deliverable Timeline

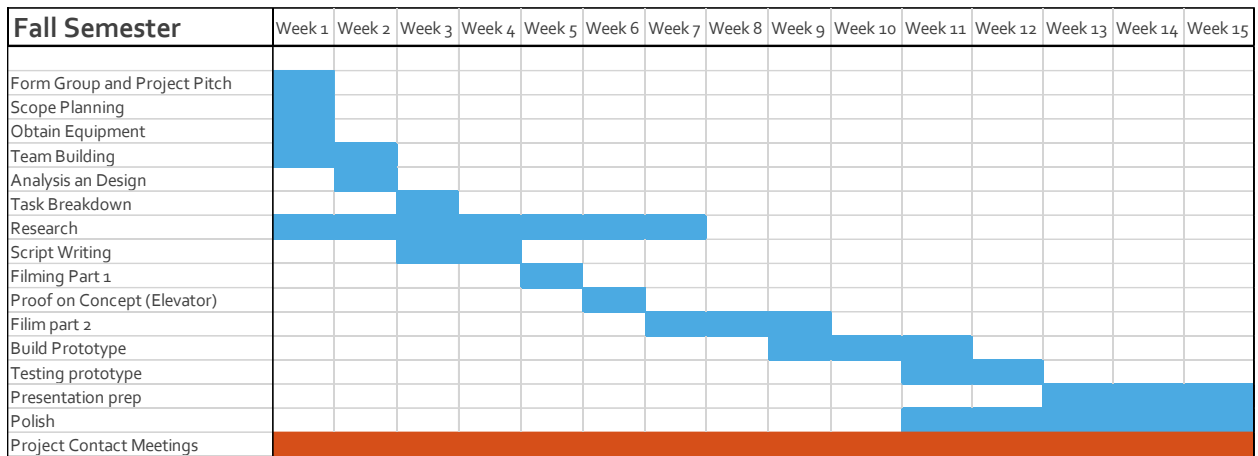


Figure 3. Gantt Chart Fall Semester 2016

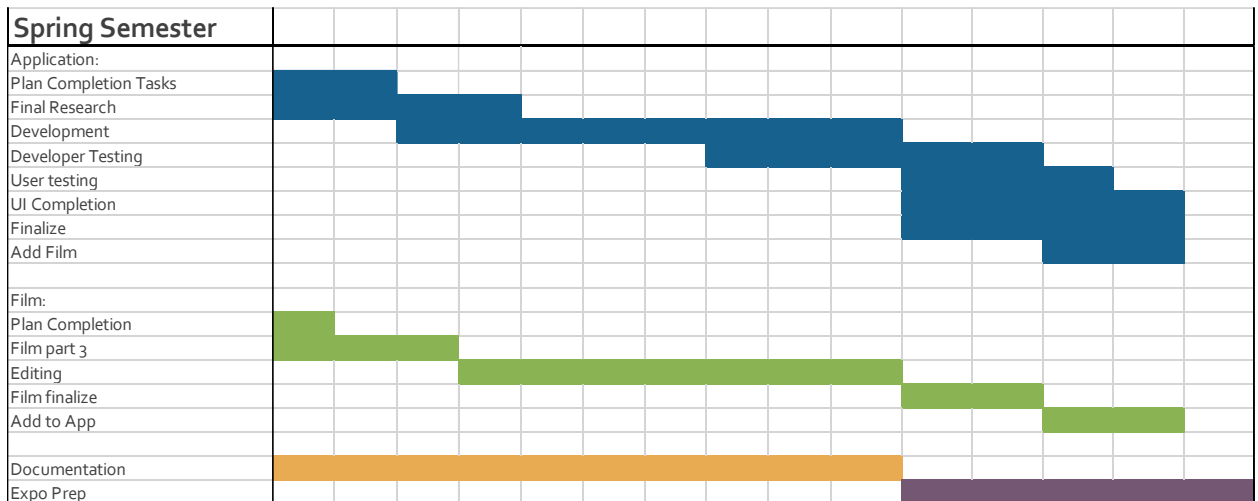
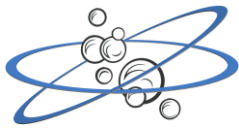


Figure 4. Gantt Chart Spring Semester 2017



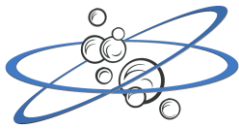
2.2. Budget

Table 3 represents the project budget. The Budget represents a real world cost of \$80,200 to create both the app and campus tour videos. The total cost obligated by UC is \$0 in exchange for experience and access to UC departments and facilities.

Because UC has elected not to own any part of the hardware or software, this budget represents only the initial deliverable at the end of our project. Continued creation of videos into future years may require investment on UC's part. Use of the platform will also require web hosting to serve the videos to the public.

No.	Item	Unit Count	Unit Price	Line Item Total
Software				
1	Labor – Programming (20 weeks at 40 hours each)	800	\$75	\$60,000
Video Capture				
2	Labor – Capture	120	\$75	\$9,000
3	Labor – Processing	80	\$75	\$6,000
Hardware				
4	Camera	1	\$5000	\$5000
5	Stabilization System	1	\$200	\$200
Subtotal				\$80,200
Total To University of Cincinnati				\$0

Table 3. Project Budget



3. Application

3.1. Overview

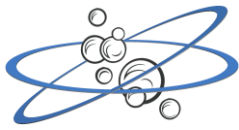
This project is written entirely in the Unity game engine in C# and the Google VR SDK.

Writing entirely in unity allows us to publish to multiple platforms from a single code base; including Google Cardboard for Android and iOS, Google Daydream, Gear VR, Oculus, and HTC Vive. The GVR SDK is required for Daydream support and has the added benefit of easily porting to Google Cardboard.

The application is a platform, utilizing an XML configuration file to generate the menus, thumbnails, backgrounds and videos. The configuration file allows for a variety of different layouts and the process of adding or subtracting videos is easy. The platform supports thousands of videos, but from a practical perspective, only so many thumbnails can fit on the screen at one time.

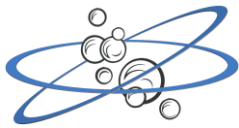
By creating a platform, the same codebase can be used for multiple clients. Each client also gets a unique, branded application to distribute as they see fit, that is different from all the other clients. Assets and videos will still need to be created, but no coding is necessary to get a new client to market. Assets and videos can also be pre-existing media the client already has.

This project utilized the platform to create a branded app for the University of Cincinnati.



3.2. Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<virtual_environment>
  <background_image_360></background_image_360>
  <face_camera_horizontal>False</face_camera_horizontal><!-- True/False - True: horizontal offset rotates around
  sphere to always face camera. False: moves straight -->
  <face_camera_vertical>True</face_camera_vertical><!-- True/False - True: horizontal offset rotates around sphere to
  always face camera. False: moves straight -->
  <!-- For future use
  <asset_graphic_player__play_button></asset_graphic_player__play_button>
  <asset_graphic_player__pause_button></asset_graphic_player__back_button>
  <asset_graphic_player__back_button></asset_graphic_player__back_button>
  <asset_graphic_player__home_button></asset_graphic_player__home_button>
  -->
  <category><!-- repeated for each category -->
    <name></name>
    <description></description>
    <thumbnail width="" height="" distance_from_viewer="" vertical_offset=""
    horizontal_offset=""></thumbnail>
    <background_image_360></background_image_360><!-- displayed when category selected -->
    <face_camera_horizontal>False</face_camera_horizontal><!-- True/False - True: horizontal offset rotates
    around sphere to always face camera. False: moves straight -->
    <face_camera_vertical>True</face_camera_vertical><!-- True/False - True: horizontal offset rotates around
    sphere to always face camera. False: moves straight -->
    <video><!-- repeated for each video -->
      <name></name>
      <description></description>
      <thumbnail width="" height="" distance_from_viewer="" vertical_offset=""
      horizontal_offset=""></thumbnail><!-- displayed on category screen -->
      <video_render_type>monoscopic</video_render_type><!-- choices: 2d | monoscopic |
      over_under | side_by_side --><!-- only monoscopic supported until future update -->
      <!-- For All Video Tags -->
      <!-- preferred format: h264 encoded as .mp4 -->
      <!-- sha1 is optional, but recommended for security and stability -->
      <!-- File size is used for display purposes. Allows the user to decide which files they want to
      download -->
      <streaming_url sha1=""></streaming_url>
      <high_quality_streaming_url sha1=""></high_quality_streaming_url>
      <download_url sha1="" file_size=""></download_url><!-- for compatibility, 1920 on the
      horizontal works best here -->
      <high_quality_download_url sha1="" file_size=""></high_quality_download_url><!-- typically 4k,
      only active (displayed) for high end devices -->
    </video>
  </category>
</virtual_environment>
```



3.3. Key Code Fragments

3.3.1. Loading XML Into Object

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;
using System.Xml;
using System.Xml.Serialization;
using System.IO;
using System.Text;

[XmlRoot("virtual_environment")]

public class VirtualEnvironment {
    public string background_image_360;
    public bool face_camera_horizontal;
    public bool face_camera_vertical;

    [XmlElement("category")]
    public Category[] categories;

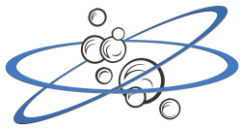
    public static VirtualEnvironment Load(string assetPath, string filename) {

        WWW www;
        www = new WWW(assetPath + filename);
        while (!www.isDone) { }
        string wwwtext = Encoding.ASCII.GetString(Encoding.Convert(Encoding.UTF8,
        Encoding.ASCII, www.bytes));

        //MonoBehaviour.print(assetPath + filename);

        var serializer = new XmlSerializer(typeof(VirtualEnvironment));

        using (var stream = new StringReader(wwwtext))
        //using (var stream = new FileStream(path, FileMode.Open))
        {
            return serializer.Deserialize(stream) as VirtualEnvironment;
        }
    }
}
```



```
public class Category {
    public string name;
    public string description;
    public float width;
    public string background_image_360;
    public bool face_camera_horizontal;
    public bool face_camera_vertical;

    [XmlElement("thumbnail")]
    public thumbnail_position[] thumbnail;

    [XmlElement("video")]
    public Video[] videos;
}

public class Video {
    public string name;
    public string description;
    public string video_render_type;

    [XmlElement("thumbnail")]
    public thumbnail_position[] thumbnail;

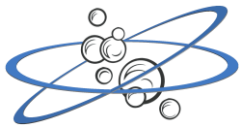
    [XmlElement("streaming_url")]
    public url_details[] streaming_url;

    [XmlElement("high_quality_streaming_url")]
    public url_details[] high_quality_streaming_url;

    [XmlElement("download_url")]
    public url_details[] download_url;

    [XmlElement("high_quality_download_url")]
    public url_details[] high_quality_download_url;
}

public class thumbnail_position {
    [XmlText(Type = typeof(string))]
    public string name;
    [XmlAttribute("width")]
    public float width;
    [XmlAttribute("height")]
    public float height;
    [XmlAttribute("distance_from_viewer")]
    public float distance_from_viewer;
    [XmlAttribute("vertical_offset")]
    public float vertical_offset;
    [XmlAttribute("horizontal_offset")]
    public float horizontal_offset;
}
```



```
public class url_details {
    [XmlText(Type = typeof(string))]
    public string url;
    [XmlAttribute("sha1")]
    public string sha1;
    [XmlAttribute("file_size")]
    public string file_size;
}
```

3.3.2. Object Rotation

```
public float width = 0;
public float height = 0;
public float radius = 340;
public float horizontalOffset = 0;
public float verticalOffset = 0;
public bool verticalFaceCamera = false;
public bool horizontalFaceCamera = false;

void SetPosition() {
    transform.localScale = new Vector3(width, 0, height);
    transform.localPosition = new Vector3(0, 0, radius); // sets starting
    position
    transform.rotation = Quaternion.identity; // resets rotation
    transform.Rotate(90, 0, 0); // faces them the correct direction

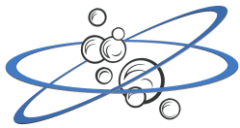
    if (horizontalFaceCamera) {
        transform.RotateAround(center, Vector3.up, horizontalOffset);
        //despite .up it rotates left or right
    } else {
        transform.localPosition += transform.right * horizontalOffset;
    }

    if (verticalFaceCamera) { // aligns tiles in sphere around center point
        transform.RotateAround(center, Vector3.left, verticalOffset);
        //despite .left it rotates up along a sphere
    } else { //if you want the tiles always vertical
        transform.localPosition += transform.up * verticalOffset;
    }
}
```

3.3.3. Full Screen Display Switching

```
private Camera camera;

public void SetVR(bool enabled) {
    camera = GetComponent<Camera>();
    VRSettings.enabled = enabled;
    camera.ResetAspect();
    PlayerPrefs.SetInt("VR Mode", enabled ? 1 : 0);
}
```



3.5. Screenshots

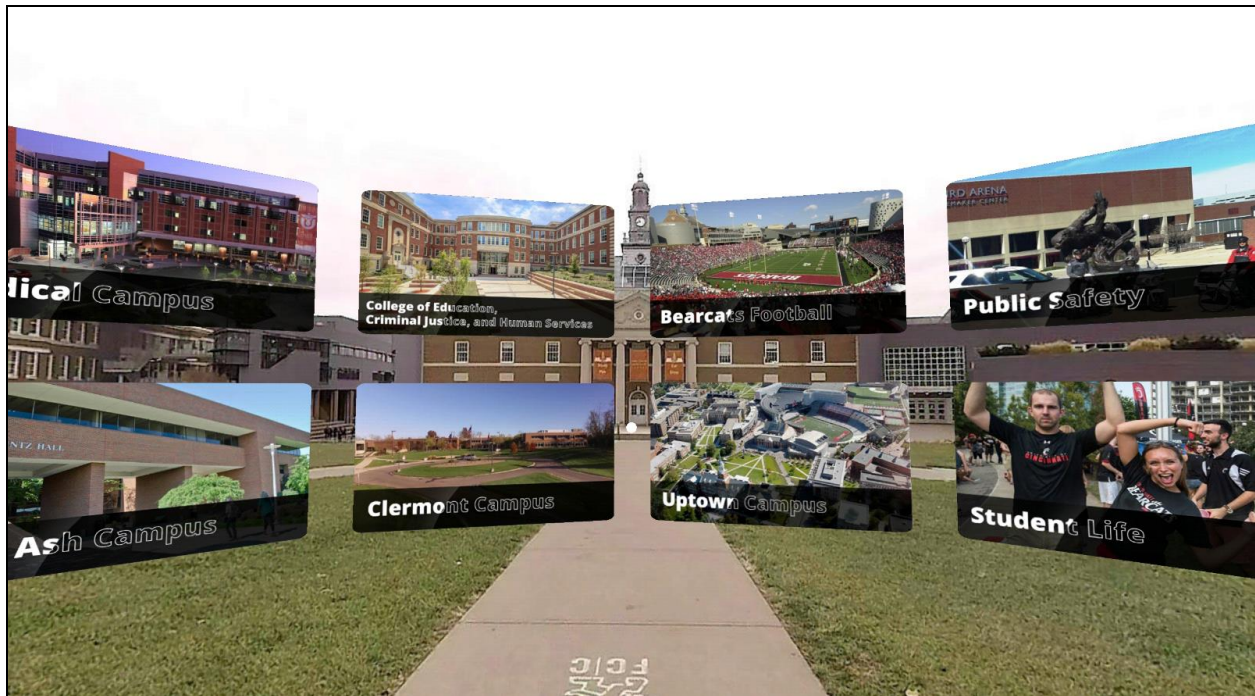


Figure 5. Application Full Screen View



Figure 6. Application Virtual Reality View

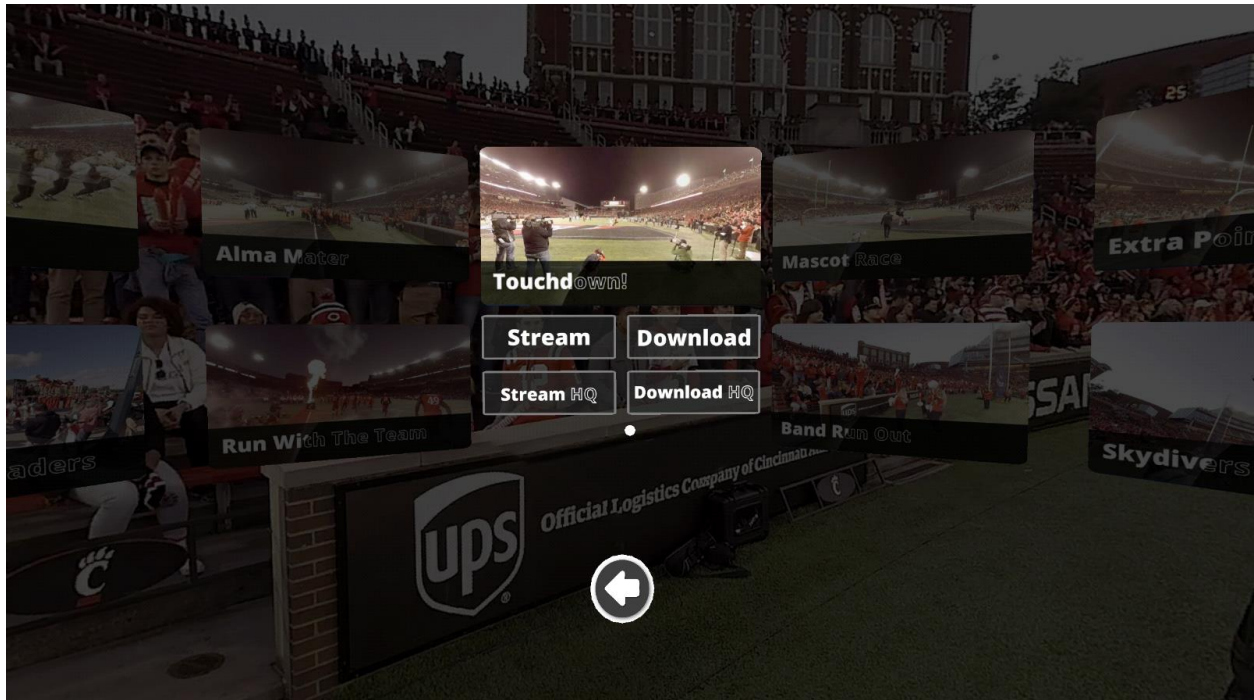
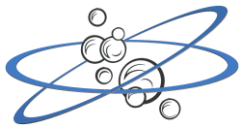


Figure 7. Video Selected



Figure 8. Video Intro With University Branding

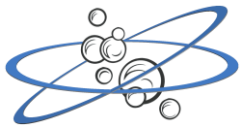
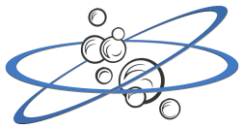
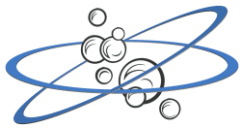


Figure 9. Video Playback



3.6. Referenced Code

- Unity 3d game engine (Unity 3d 2017)
- Google GVR SDK for Google Cardboard and Daydream compatibility (Google - GVR SDK 2017)
- MixedCameraControl class for controlling camera in full screen mode (Rosenberg 2016)
- Code for rounding corners on square objects (Bunny83 2015)
- Easy Movie Texture for playing videos. It is a wrapper for the Unity movie player for easy porting to multiple platforms (Lee 2017)



4. Video Production

4.1. Overview

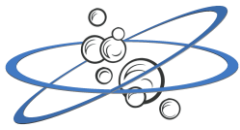
All 360 cameras are a combination of several smaller cameras. While some rigs use only two cameras, each with 270 degree field of views, others can have 24+ cameras. This means, after shooting the video, there are 2-24 separate videos for each scene that need merged into one equirectangular shot at a 2:1 ratio, or 1:1 for stereoscopic (specific view for each eye).

4.2. GoPro Omni

For this project, we used a GoPro Omni camera rig. It consists of six GoPro Hero4 Black cameras inside an aluminum spherical rig. It comes with Kolor software for stitching and is capable of shooting up to 8k resolution at 30 frames per second when the video feed is combined. All recording for this project was done at 6k to take advantage of the higher framerate offered, 60fps.



Figure 10. GoPro Omni



4.3. Parallax

Parallax is an effect where the position of an object appears different depending on the viewing angle. Astronomy uses it to calculate the distance to stars and galaxies. In this case, each camera sees the world around it slightly differently. For far objects, this effect is minuscule;

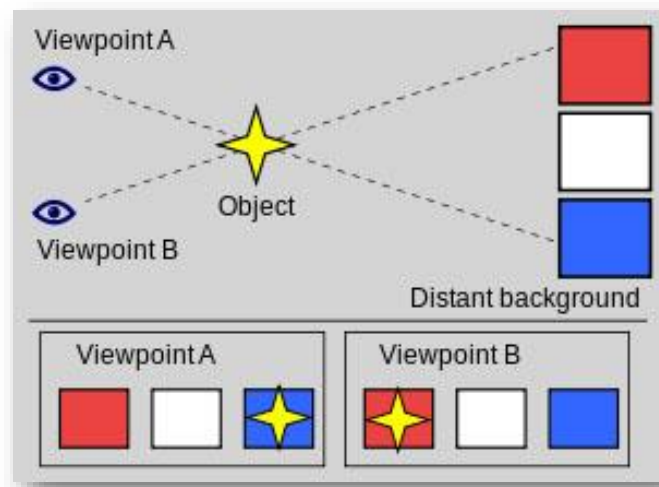


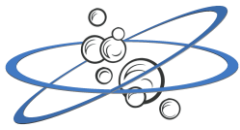
Figure 11. Parallax

however, for close objects it can be profound. You can observe this effect by looking at various objects around you, switching which eye is closed, or covering them with your hands.

Because the camera is recording both near and far items at the same time, controlling parallax becomes important. The more cameras you have, the lower impact parallax has, since the observed difference between the cameras is smaller. Lenses that capture wider field of views can also contribute to the parallax effect.

4.4. Video Capture

Unlike traditional video, a 360 camera captures everything. Meaning you have limited control over framing the shot. The starting direction can be controlled in editing. However, the only way to keep something or someone from being in the shot is to not be in view of the cameras.



Because of parallax, it is also important to be aware of where each camera is pointing. Avoid putting important things between two cameras, which causes a ‘stitching error’. For instance, a narrator in frame should have a camera pointed directly at him or her. Which camera, depends on which rig you are using. For the GoPro Omni use one of the upward facing cameras.

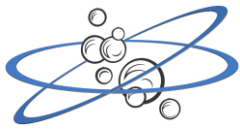
To reduce stitching errors, there should be a minimum four feet clearance in all directions from the camera. This buffer will help minimize the effect of parallax. Items far away, even if on the line between two cameras can be blended seamlessly because parallax is so small.

4.5. Audio Capture

The purpose of virtual reality is to transport the viewer into the scene being shown. For this reason, ambient noise is usually desired. It is advisable to mic any narrator, but background noise from the scene should not be scrubbed out. If adding narration during editing as voice over, merge with the audio captured from the cameras for a more authentic sound.

Sound effects added in are very noticeable and can distract from the scene, use with extreme caution. Even effects intended to help with immersion such as nature noises can easily sound out of place and distract from the message of the video.

4.6. Stitching



Stitching is the act of putting all the separate camera feeds into one single image. Because the images overlap, and because of parallax, this process is never perfect. Software is used to minimize this as much as the camera system allows.

Because they are physically separate cameras, they each end up with different levels of light. Even if the settings are the same, they will physically see different amounts of light. One may be pointed at the sun for instance.

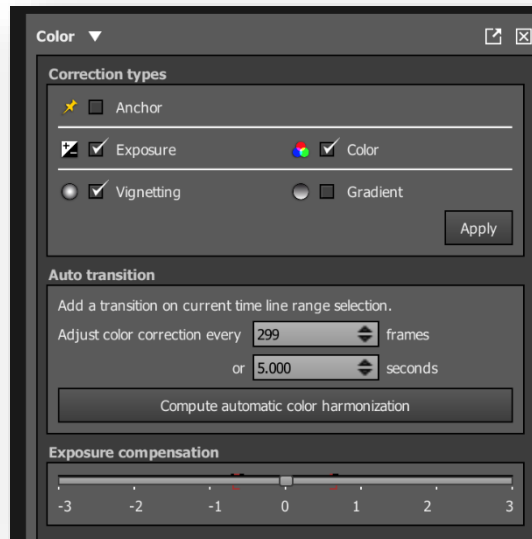
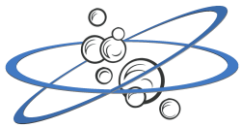


Figure 12. Color Harmonization



Figure 13. Stitching Error

Most stitching software should be able to correct for this. Kolor software, used for this project, has a color harmonization tool that can be used with its default settings in most cases.



4.7. Editing

4.7.1. Color Correction

In order to create a better stitch, cameras often record in a 'flat' light. This means they need to be color corrected during editing or the entire video looks washed out. See below before and after shots when color correcting.

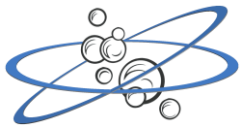
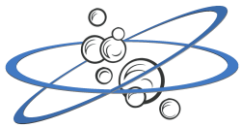


Figure 14. Before Color Correction



Figure 15. After Color Correction



In order to achieve this several filters must be applied to the scene.

First, and most important color correction tool, is 'Lumetri Color', or any filter that allows the limiting of RGB curves. This tool creates brighter whites and darker blacks by ensuring some pixels reach the visible threshold and has the added effect of removing the gray haze created by the cameras. Gray pixels become black.

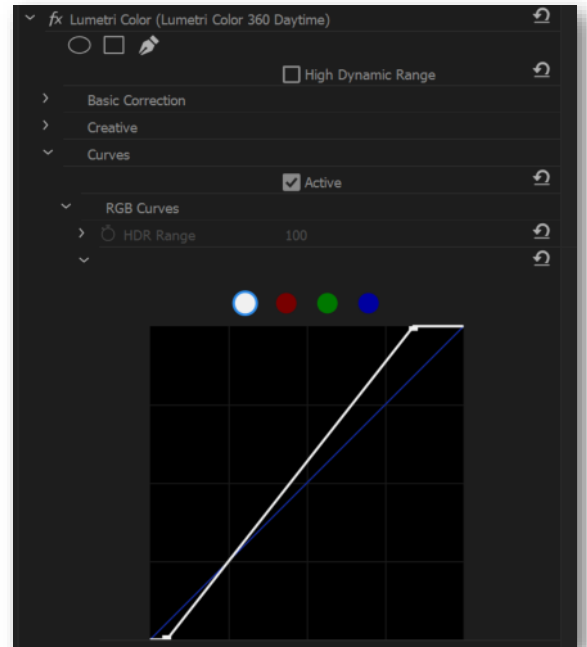
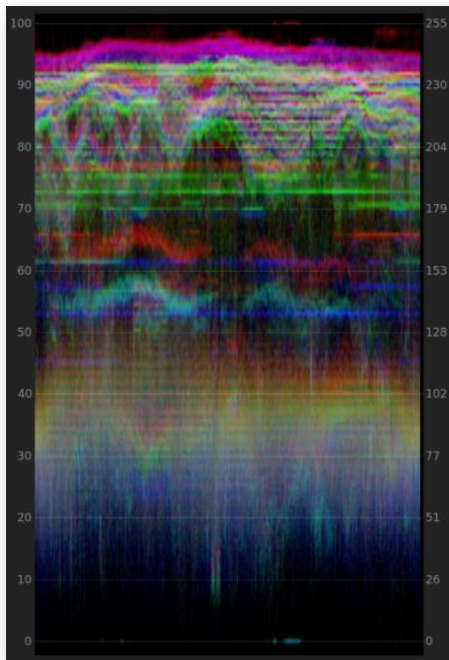
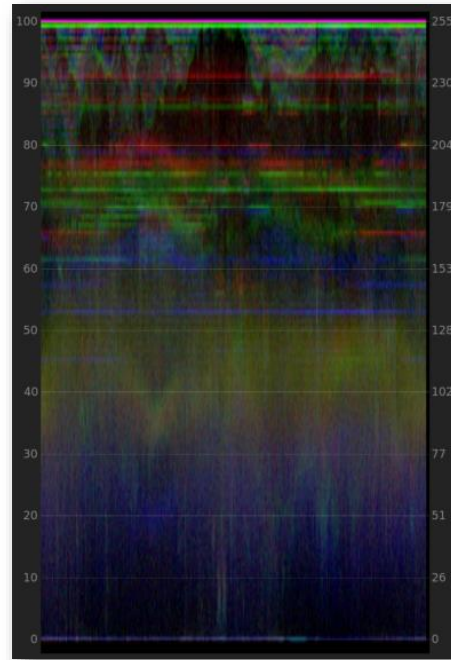


Figure 16. Lumetri Color

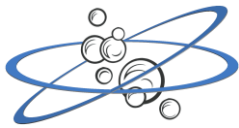
Every scene is different in the amount of adjustment it will need. Use the 'Waveform (RGB)' view as a guide for precise adjustment. See the below before and after shots, notice there are no color gaps at the top and bottom after adjustment.



**Figure 17. Waveform (RGB)
Before**



**Figure 18. Waveform (RGB)
After**



After this process, colors can still be flat or even wrong. The scene must still be white balanced and colors adjusted slightly.

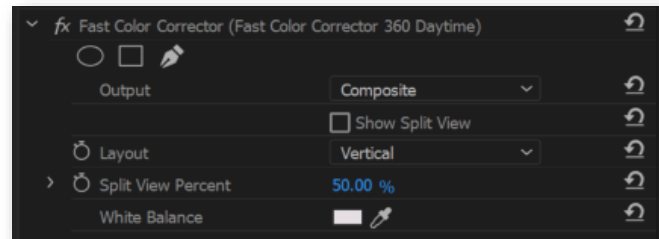


Figure 19. White Balance

White balances is achieved using ‘Fast Color Corrector’, selecting the dropper under ‘White

Balance’ and selecting something in the scene that is supposed to be white.

Finally, using the ‘Three-Way Color Corrector’, ‘Master Saturation’ and ‘Midtone Saturation’ can be adjusted between 100 and 125. Most scenes in this project are set to 100 or 110.

4.7.2. Graphics Insertion

Notice back in Figure 14 and Figure 15 that anything outside the middle 20% of the vertical frame is warped left and right. When viewing 360 video, this flat image is mapped to a sphere. The very bottom row of pixels, indistinguishable in these pictures, gets mapped to a single point on the sphere.

Because of this effect, any graphics inserted into the video must be adjusted for this warping or they will not appear as intended in the final 360 video.

Figures 20 and 21 show the difference between the flat image where the campus map is shown, and what it looks like mapped to the sphere for viewing.

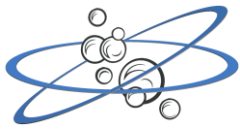
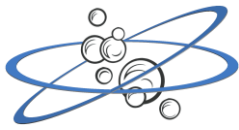


Figure 20. Flat Map View



Figure 21. Corrected Map View



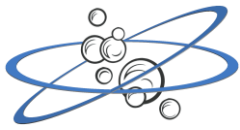
4.8. Publishing

The H.264 codec is capable of a max resolution of 4096 for width and height. While the camera is capable of 8k resolution, no mobile device can display it, and only a few codecs are capable of compressing it, such as H.265. Encoding time also drastically increases from around 30 minutes to 30 hours.

When publishing to YouTube, the full capability of H.264 can be used, 4096 pixels and a 24 Mb/s bitrate. YouTube will automatically create different versions of the video for a variety of resolutions allowing the viewer's device to decide what it is capable of.

When publishing to the app associated with this project, we created two resolution choices. 4096 x 2048 at 60fps and 24 Mb/s bitrate, for devices capable of playing 4k video; and 1920 x 960 at 30fps and 8 Mb/s bitrate for older devices, or any user limited on storage or bandwidth. Users with a device capable of 4k are given the choice which quality version they want.

Videos are uploaded to a publicly accessible server for downloading by the app and streaming copies are made in the HLS V3 format. No videos are included with the app at installation. All image assets and thumbnail views however are included in the compiled app. Downloading them at runtime causes unity to stutter and creates a poor user experience.



5. Testing

5.1. Overview

This section explains the testing methodology for the Soapbox VR platform. As this is a platform intended for multiple devices, testing will be in phases and appropriate for each device.

5.2. Scope

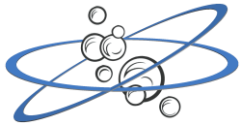
Testing will encompass the Android Cardboard and Daydream platforms. Future phases will cover iOS, Gear VR, Oculus, and HTC Vive.

5.3. Objectives

By running isolated and unit tests, we expect to find problems before consumer launch and verify that components are working as expected.

5.4. Hardware Requirements

- Google Cardboard requires an Android phone with OS 4.2 or newer.
- Google Daydream requires an Android phone compatible with Daydream and OS 7.0 or newer.



5.5. Control Procedures

We are using GitHub.com to manage all version control, bugs, and change requests.

5.5.1. Problem Reporting

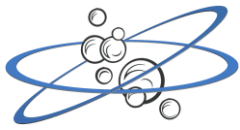
On the GitHub project page click the 'Issues' tab, then green 'New issue' button. Next to 'Labels' is a gear icon, click that and select 'bug' from the list to flag the issue as a bug report.

5.5.2. Change Requests

On the GitHub project page click the 'Issues' tab, then green 'New issue' button. Next to 'Labels' is a gear icon, click that and select 'enhancement' from the list to flag the issue as an enhancement / change request.

5.6. Features to Be Tested

- Application loads for Google Cardboard
 - View can be switched from full screen to cardboard mode
- Application loads for Google Daydream
- The correct graphics load on the intro screen
- Can select menu items
- Menu changes appropriately based on user decisions
- Monoscopic videos play



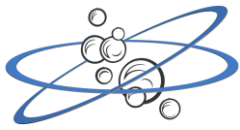
- Audio also plays and is in sync
- Normal video
- High quality video
- Various other UI specific expectations
- Video downloading
- Video streaming
 - Test in low and high bandwidth locations
- Changes to the XML configuration file are reflected in the application
- XML understandability to non-programmer

5.7. Features Not to Be Tested

- Apple iOS – future version
- Samsung Gear VR – future version
- HTC Vive – future version
- Stereoscopic Videos – future version
 - Side by side
 - Above and below
- 2d videos – future version

5.8. Entry and Exit Criteria

- Entry Criteria



- Build complete
- Self-testing complete
- Test environment is set up
- Exit Criteria
 - All tests are run and recorded
 - Bugs are documented and fixed

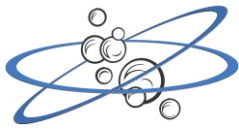
5.9. Logging Test and Reporting

When a bug is found during testing the tester will document the bug on the testing form. The developers will meet after the test and decide if it is a bug, a missing feature, or user error. Developers will enter bugs and missing features into the GitHub system for tracking and resource allocation.

User errors are examined by the human factors team for usability and UI design changes. Identified changes are submitted to GitHub for tracking and resource allocation.

5.10. System Testing

Soapbox VR will be tested as a complete application to ensure all individual components work together as a whole. The application is not considered working and publishable until this testing is complete.



5.11. Testing Procedures

The following outlines the steps needed to conduct a successful test.

1. Create all test scenarios and test cases.
2. Prepare a document of the steps for the tester to use.
3. Run listed tests and record results to the testing document.
4. Specify any bugs found and steps to recreate them if possible.

Tests performed:

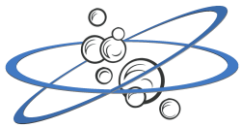
1. Stability Test – this test will focus on running through the application repeatedly checking for crash events.
2. Launch Test – Program must start and shutdown properly
3. User Interface Test – Program must be visually appealing. But users must also understand how to use the application using just the visual cues on the screen.

5.12. Pass/Fail Conditions

Soapbox VR must pass all tests to be successful, including expected failure tests. Tester will document all unexpected results.

5.13. Schedule of Team Member Testing

Developers tested weekly starting 10/09/2016 and conclude on 04/03/2017.



5.14. Schedule of Public Testing

Clermont IT Pros will tested on 03/06/2017 and concluded on 04/03/2017.

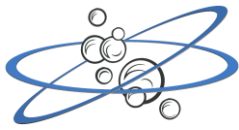
Project sponsors tested on 03/13/2017 and conclude on 04/03/2017.

Beta Testers tested on 03/20/2017 and conclude on 04/03/2017.

5.15. Risks / Assumptions

The following will impact testing:

- Tester availability
- Device availability
- Features not completed on time
- Bugs not fixed on time
- Meeting times for public testing groups to establish procedure
- Changes to any of the platforms being used or SDKs in use
- Video hosting availability
- Requires consistent internet use



6. 2017 Tech Expo

6.1. Overview

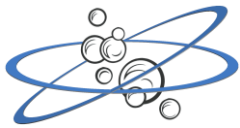
Our plan for the Tech Expo was to maintain enthusiasm for our booth throughout the entire event, which was successful. We did this by giving visitors many things to do at our booth. We had three VR headsets and one tablet for viewing the app. We also had plenty of information and visuals available to anyone curious to know more.

Before the expo, we published our app to the app store under UC's official account, as a beta version. This allowed us to have a QR code at the booth for anyone wanting to download our app. We were also able to put the QR code onto the Google Cardboard units we were giving away at the booth.

As mentioned, we secured four hundred Google Cardboard units to give away, of which we gave away around two hundred and fifty units. Each one had a sticker containing a QR code to download our app and a QR code to download google cardboard.

We did get a lot of interest in people wanting to know more about the platform and app and interest in people wanting to know technical details behind the technology. Finally, we had a survey of six questions we encouraged everyone to take.


6.2. Expo Posters






UNIVERSITY OF Cincinnati

- 4 Campuses
- 6 Categories
- 30+ Videos



Colin Crowell & Brian Schieltz • Technical Advisor: Professor Robin Carew • Expert Advisor: Mickey Mentzer & Professor Bill Nicholson


Virtual Reality Branded Video Platform

Problem

- Virtual reality and 360° video are expensive to create
- Benefits do not exceed ROI
- Can be unaffordable for small business

Solution

- A platform to easily create branded apps
- No programming required
- Customizable
- Inexpensive
- Configurable XML controls everything



Gather Analytics



Publish to all major platforms



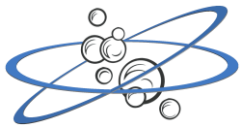
<XML>





University Of Cincinnati • College of Education, Criminal Justice, & Human Services • School of Information Technology

Figure 22. Main Expo Poster



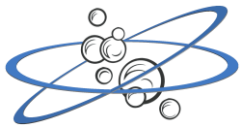
Financial Supporters

- Arvie & Rita Schieltz
- College of Education, Criminal Justice, and Human Services

Community Partners

- Charles Schwartz – International Admissions
- Jeff Bauer – Dean of Clermont
- Adam Niemeyer – Football Video Coordinator
- Brian Verkamp – Assistant Dean CECH
- Cass Johnson – CECH Director of Marketing
- Nicola Ziaddy – Marketing + Digital Communications
- Pete Gemmer – Blue Ash Director of Communications
- Bill Bangert – Medical Center Public Relations
- Michele Ralston – UC Public Safety
- Chris Collins – UC Virtual Environments
- Mae Hanna – UC Clermont Sr Assistant Dean
- Lori Carter – UC Clermont Admissions
- Melissa Baer – UC Admissions
- Darren Kuzma – Axiom General Manager

Figure 23. Supplemental Expo Poster



6.3. Survey Results

Thirty-three people responded to our survey, below are the results.

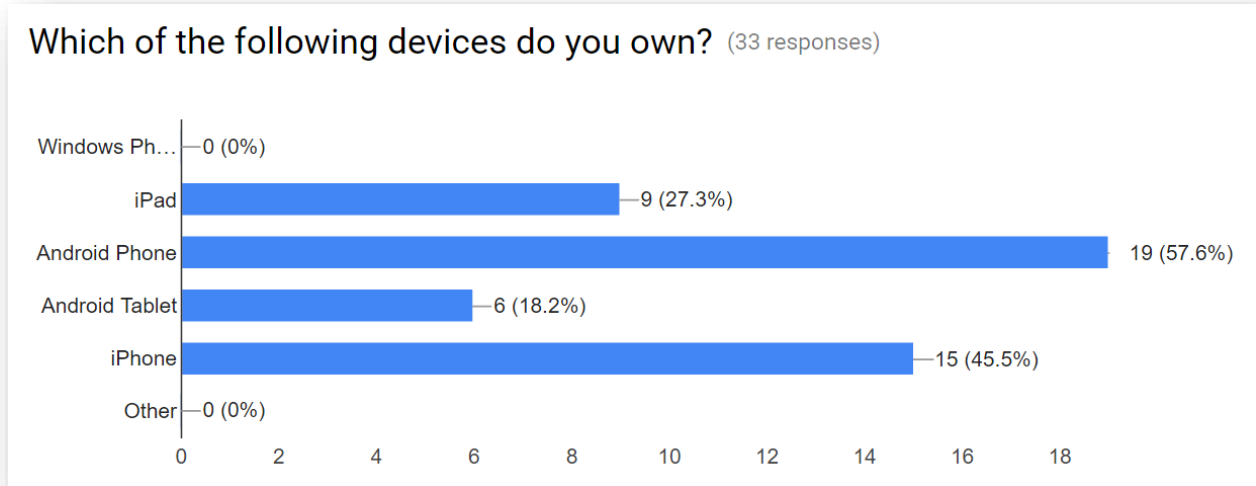


Figure 24. Survey Question 1

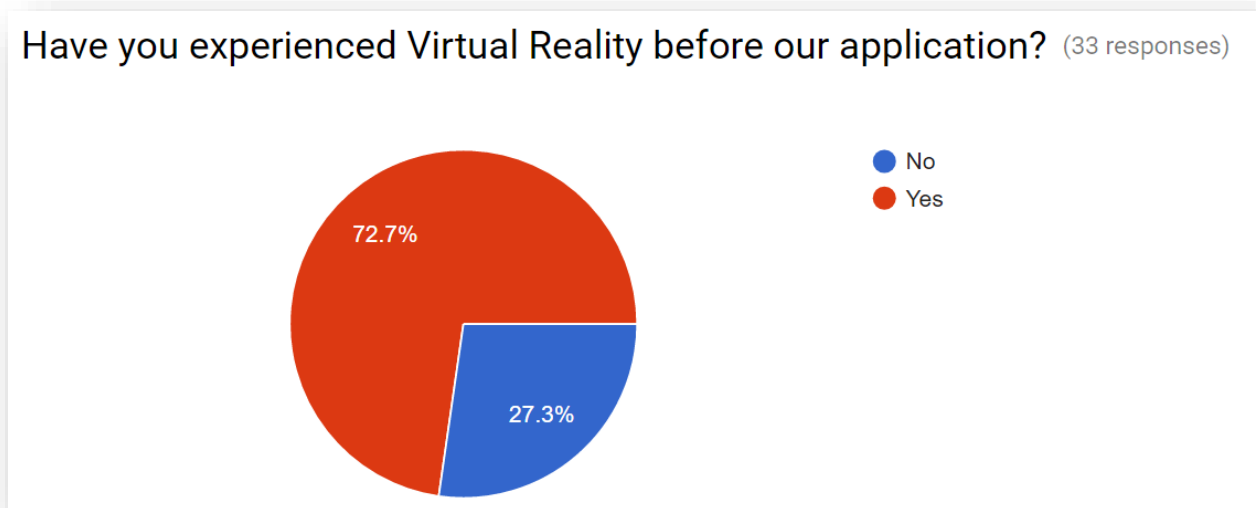


Figure 25. Survey Question 2

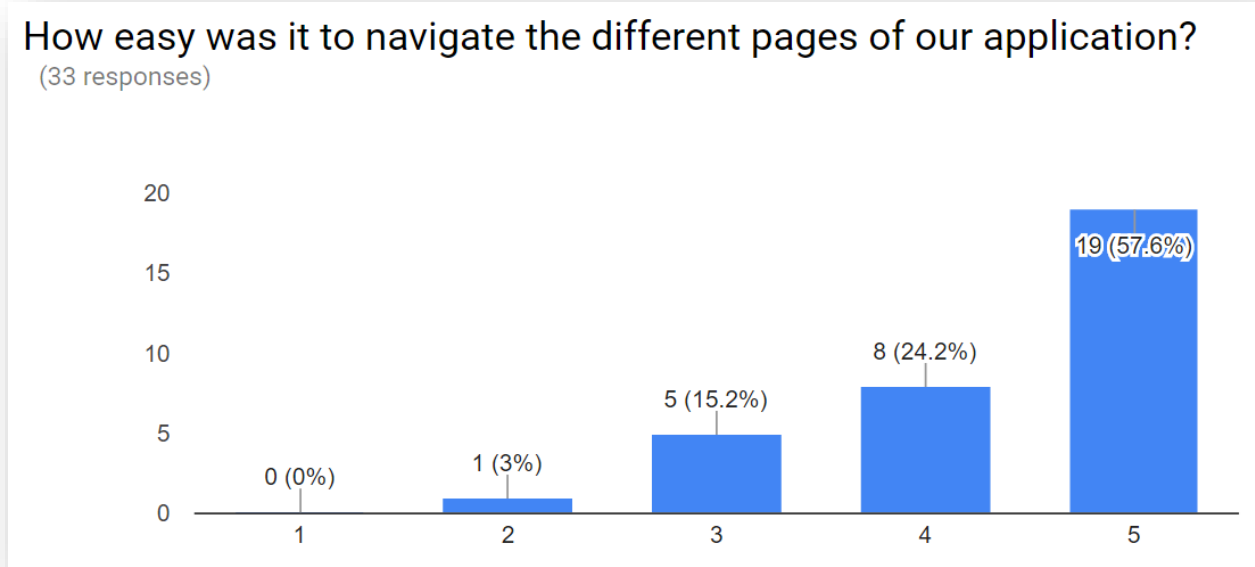
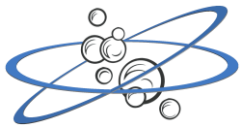


Figure 26. Survey Question 3

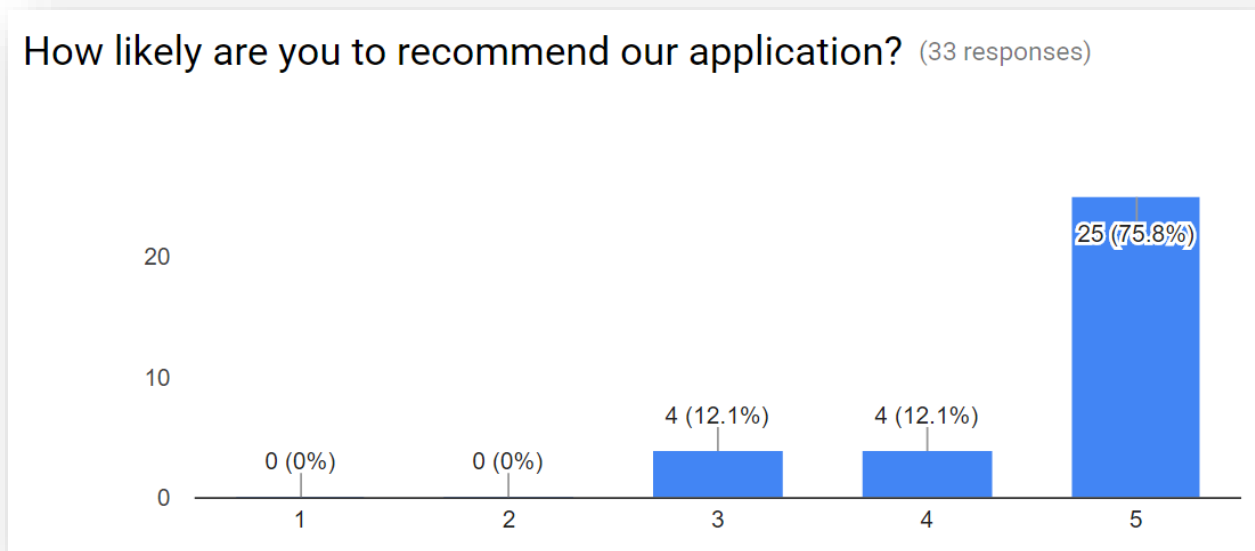
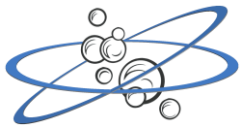


Figure 27. Survey Question 4



What changes would most improve our application? (12 responses)

This is Charlie's response. I was unable to navigate through the pages on my phone (Huawei Honor 5x). It would not rotate or let me select anything from the main screen. Perhaps my phone is incompatible? I will show you next time I see you.

Need more clear visuals

more initial info to the user on how to use it. "tutorial"

more apps that utilize space for VR

more events to watch

change location in vr instead of changing it outside. also kind of nauseating, I'm dizzy

havent used it enough yet

I can't download it to an apple phone

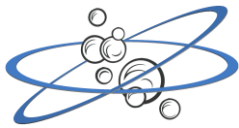
higher resolution

the UI is a bit clunky

add some games

Zoom!

Figure 28. Survey Question 5



What businesses or industries would you find this most helpful in? (28 responses)

everything

home buying / realty

Event marketing. Event space rentals. Unique Experiences. ie tourism companies

engineering, if you can "look" inside a system / machine before you fit it, it's easier to improve.

Construction and real estate

field trips and college students

video game industry, medical field

Google Maps

Marketing

tourist destinations

virtual touring

video games, architecture, art

video entertainment

virtual tours to visit places you cant get to

colleges, warehouse jobs tour

as a student it is nice

medical, gaming, engineering, and aeronautics

touring, home buying

film and video games

video games

gaming

giving virtual tours to us students, campus tours, google earth / streetview

recommend it to students trying to go to UC. really cool idea / app

nursing homes, therapy, real estate

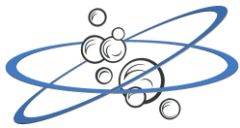
all institution that have a huge campus

therapy for amputees, people with PTSD, etc

For UC! Great Job!! :)

colleges, video games, marketing

Figure 29. Survey Question 6



6.4. App Downloads

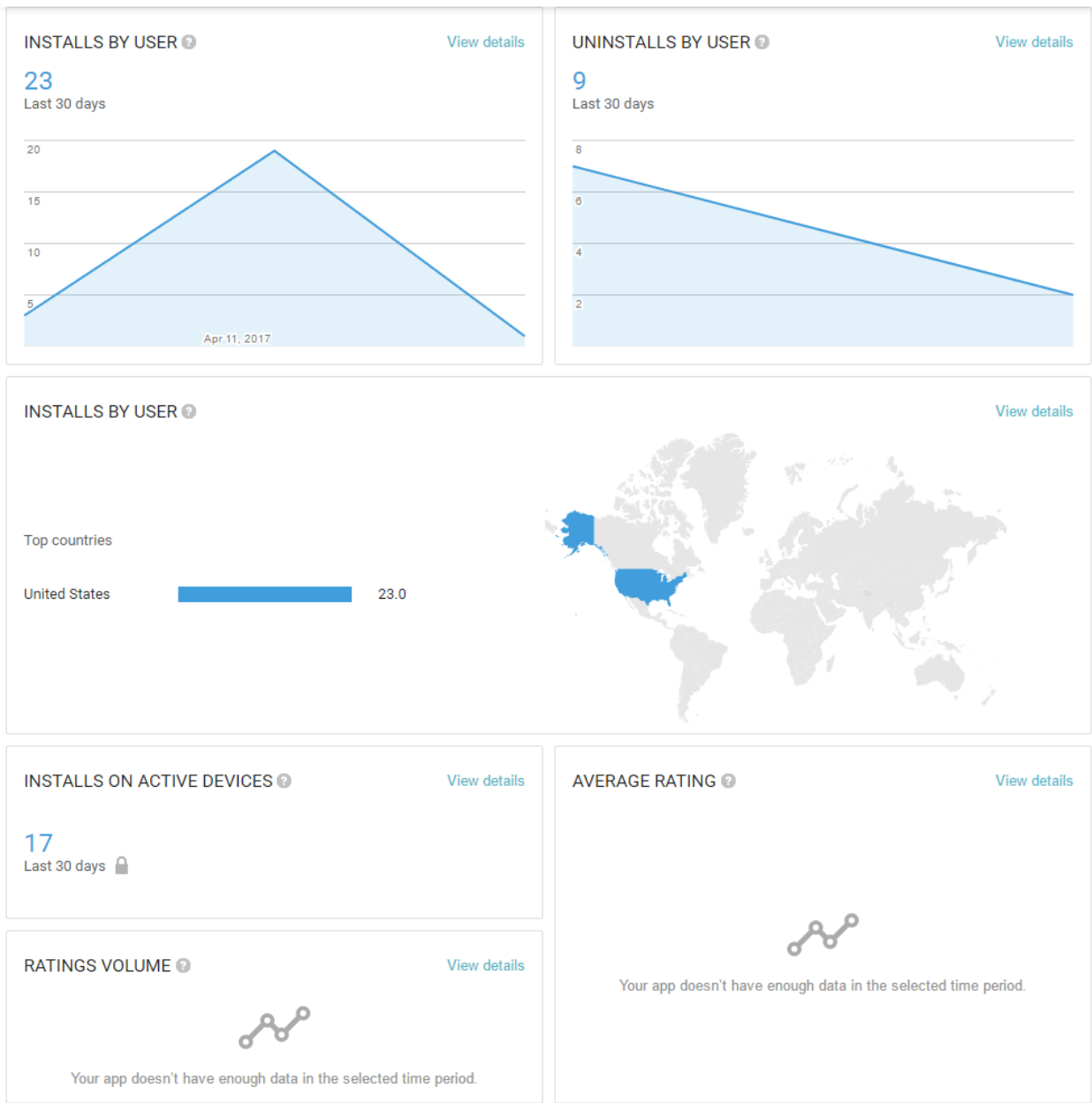
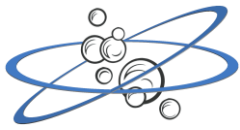


Figure 30. App Downloads From Expo



7. Conclusion

7.1. Overview

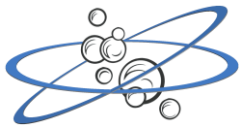
In the 2016 Fall Semester, the team acquired the necessary equipment to film and edit 360° panoramic video content, created a functional prototype application for Android OS, and recorded a series of 360-degree panoramic videos.

In the 2017 Spring Semester, the team completed an application using the Unity game engine, allowing both a full screen and VR experience. The app is currently available on both the Google Cardboard and Google Daydream platforms. In the future, the platform will support iOS, HTC Vive, Oculus and Gear VR. All of the steps we took during this project will make distributing to all platforms an easy task. However, as of writing this we were unable to get development units for Vive, Oculus and Gear. Moreover, because the Google VR SDK is still in the early stages of development, iOS is not supported yet.

Because the video content is expansive, it will exceed the bounds of what can be hosted on and downloaded from Google's Android Play Store. If the University is to adopt and use the product, arrangements must be made to establish a content delivery server either within the University's data center, or via third party.

7.2. Lessons Learned

This project ended up with a large scope. We could have easily made a small number of videos with a cheap camera, a simple android interface. We would have satisfied the requirements of



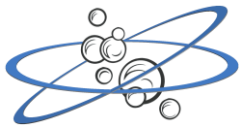
this class, and that was the original scope of this project. But, we also would have spent two semesters creating something no one would use.

When we started, we did not know if anyone at the university would even talk to us. If they did, would they help? And even if we did all of that, would the university want our product? This project started off as something ‘cool’, a virtual campus tour; and turned into a real product that fills a gap in the market no other product fills. The university also gets to be the first in the country with a dedicated VR app and substantially more 360 content than anyone else offers.

We learned that faculty at the university will answer their phones for students and actually listen. We were given support by many departments on campus, including CECH, Admissions, International Admissions, Football, the ROAR student group, Public Safety, Medical Campus, Clermont, and Blue Ash. All of which treated us with respect and support and without whom we could not have completed the project to the scale that we did.

We also learned that taking risks pays off. After finishing the fall semester, we made two drastic changes to our project. The first is to expand the scope of our project from a single app to a platform. While risky, and involving more work, it also completely changed the perception of our project. Instead of a student project that ended in April, we became a viable business opportunity. That distinction is important when the university considered whether to publish our app officially.

We also made the hard decision to completely scrap our original interface, an android app with a standard list view. A design common in the industry for a majority of Android apps, including VR apps. Giving up on the industry standard was a hard choice to make and many people questioned the decision. In addition, we had zero Unity experience; we had never even heard of



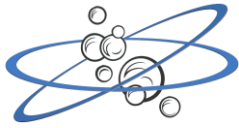
Unity before the fall semester. But, we realized that having to remove the device from a VR headset to navigate the interface was a very poor user experience. Putting our users first made us realize we had no choice but to re-write our interface from scratch. A decision that has proven to be correct by the amount of positive feedback we have received.

7.3. Skills Developed

Before the fall semester, we had never heard of the Unity game engine, or what it was capable of, and we had taken only a single class in C#. We used some existing skills, Colin uses C# regularly for work and Brian has experience with 3D modelling, Autocad and video editing in Adobe Premiere. However, this project required us to push our knowledge to new areas and utilize it in ways we never had before.

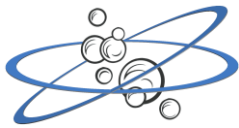
We also drastically increased our project management skills. The programming duties, split between two people was mostly as you would expect. However starting the fall semester we knew very few people on campus. Utilizing the UC directory, we called nearly 100 people and had dozens of in person meetings along with several different days and locations shooting video. Managing that many separate conversation threads was a challenge but the reward was worth it.

Our video editing experience was also very limited before this project. While we knew the basics of the Adobe Premiere interface, we had never created 360 content. We had never used stitching software or needed to color correct. Those were all new skills we had to learn on our own.



8. Bibliography

- Bauer, Jeff, interview by Brian Schieltz, & Colin Crowel. *Does UC want 360 degree virtual campus tours?* (August 28, 2016).
- Bunny83. *unity 3d*. 07 30, 2015. <http://answers.unity3d.com/questions/1007222/round-procedural-square-mesh-corners-c.html> (accessed 01 2017).
- Google - GVR SDK. *github*. 2017. <https://github.com/googlevr/gvr-android-sdk> (accessed 2017).
- Lee, Jae Yun. *Unity Asset Store*. 03 07, 2017. <https://www.assetstore.unity3d.com/en/#!/content/10032> (accessed 2017).
- Mentzer, Mickey, interview by Brian Schieltz, & Colin Crowel. *Brainstorming and Technical Questions* (October 6, 2016).
- Niemeyer, Adam, interview by Brian Schieltz, & Colin Crowel. *UC Football* (October 4, 2016).
- Rosenberg, Jeff. *Unity 3d*. 10 21, 2016. <https://forum.unity3d.com/threads/single-eye-render-mode-like-the-gvr-plugin-with-vr-mode-disabled.437274/> (accessed 01 2017).
- Schwartz, Charles, interview by Brian Schieltz, & Colin Crowel. *International Student Needs* (September 8, 2016).
- Unity 3d. *unity3d*. 2017. <https://unity3d.com/> (accessed 2017).



9. Appendix

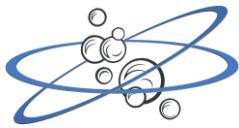
9.1. Testing Results

9.1.1. Bug List

Bug ID	Description	Severity	Status
1	Double clicking category thumbnail causes it to be selected like a video. Can crash app	High	Fixed
2	Large videos do not clear buffer fast enough when going back to video thumbnails	High	Not Fixed
3	Back button does not properly appear when selecting a category	High	Fixed
4	Downloading too many videos causes downloads to start failing until device is rebooted	High	Not Fixed
5	Back button not lined up with media control buttons when user looks down quickly	Low	Not Fixed
6	Screen will slowly rotate on its own, possibly sdk bug	Low	Not Fixed
7	Selection reticle will show highlighted on background items that can't be clicked	Low	Not Fixed
8	Blank value in config file for download would cause exception	High	Fixed
9	When video ends, it does not always automatically go back to category screen	Medium	Not Fixed
10	Download graphic sometimes displays in front of, instead of behind download button	Low	Not Fixed
11	On some phones, that have a high enough version of android, the screen does not rotate with the phone	High	Not Fixed
12	When video is selected, there is a graphics slowdown	Medium	Not Fixed

9.1.2. Table Key

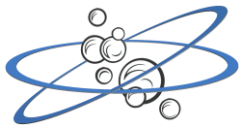
Status	Description
Passed	The test case applies for the current version and was run with the expected results
Pending	The test case applies for the current version but was not run yet
Failed	The test case applies for the current version and was run with errors
Not Run	The test case will not be run in the current drop
Deprecated	The test case is not valid any more due to change of requirements/priorities
Not Implemented	The test case was not implemented yet for the specified version
N/A	Feature will never exist for Environment selected



Feature	Environment	Device
Category	Android - Cardboard	Google Pixel
Videos	Android - Daydream	Samsung Note 4
Video Selection	iOS - Cardboard	Samsung Tab S2
Streaming	Gear VR	Huawei Honor 8
Downloading	Oculus	
Playing	Vive	
Deleting		
Video Controls		
Toggle Fullscreen		

9.1.3. Testing Cases

** Tests done with known bugs are considered "Passed" if no new bugs are found.					
Test Case ID	Device	Environment	Feature	Test Case Description	Expected Result / Comment
1	Google Pixel	Android - Cardboard	Category	1- Categories Display 2- Select Category	Videos of selected category appear
2	Google Pixel	Android - Cardboard	Videos	1- Video thumbnails appear for correct category 2- Select Video 3- Click Back	Clicked thumbnail details appears / back to category view
3	Google Pixel	Android - Cardboard	Video Selection	1- Video options appears 2- Can click various ways to play 3- Can download / delete if already downloaded 4- Click Back	All items are clickable / back to videos view
4	Google Pixel	Android - Cardboard	Streaming	1- Streaming and Streaming HQ	Selected option plays
5	Google Pixel	Android - Cardboard	Downloading	1- Download Video or Video HQ 2- Download changes to delete. Play is displayed	Video Downloads
6	Google Pixel	Android - Cardboard	Playing	1- Watch Video 2- Video Controls appear	Video plays in its entirety
7	Google Pixel	Android - Cardboard	Deleting	1- Delete Video 2- Delete Button changes back to download 3- Play disappears	Deletes video
8	Google Pixel	Android - Cardboard	Video Controls	1- Only appear when in video play mode 2- Can Pause, play, stop, jump ahead, jump back, exit video 2- Pause is displayed when playing, play is displayed when pauses or stopped	Control video playback

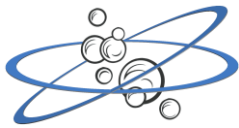


9	Google Pixel	Android - Cardboard	Toggle Fullscreen	1- Click Carboard Icon 2- Toggle Between fullscreen and VR	Field of view correctly changes
---	--------------	---------------------	-------------------	---	---------------------------------

9.1.4. Test Results – Brian Schieltz

Tester:		Brian Schieltz					
Test Case ID	Device	Environment	3/5/2017	3/12/2016	3/19/2016	3/26/2016	4/2/2016
1	Google Pixel	Android - Cardboard	Passed	Passed	Passed	Passed	Failed #8
2	Google Pixel	Android - Cardboard	Passed	Passed	Passed	Passed	Passed
3	Google Pixel	Android - Cardboard	Passed	Passed	Passed	Passed	Passed
4	Google Pixel	Android - Cardboard	Passed	Passed	Passed	Passed	Passed
5	Google Pixel	Android - Cardboard	Not Implemented	Not Implemented	Not Implemented	Passed	Passed
6	Google Pixel	Android - Cardboard	Failed #2	Passed	Failed #9	Passed	Passed
7	Google Pixel	Android - Cardboard	Not Implemented	Not Implemented	Not Implemented	Passed	Passed
8	Google Pixel	Android - Cardboard	Passed	Passed	Passed	Passed	Passed
9	Google Pixel	Android - Cardboard	Not Implemented	Passed	Passed	Passed	Passed

1	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed	Failed #3
2	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed	Passed
3	Google Pixel	Android - Daydream	Passed	Failed #7	Passed	Passed	Passed
4	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed	Passed
5	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Not Implemented	Passed	Failed #4
6	Google Pixel	Android - Daydream	Passed	Passed	Passed	Failed #5	Passed
7	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Not Implemented	Passed	Passed
8	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed	Passed
9	Google Pixel	Android - Daydream	N/A	N/A	N/A	N/A	N/A

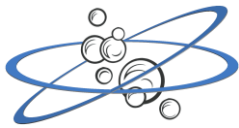


1	Samsung Note 4	Android - Cardboard	Passed	Not Run	Not Run	Failed #6	Passed
2	Samsung Note 4	Android - Cardboard	Passed	Not Run	Not Run	Passed	Passed
3	Samsung Note 4	Android - Cardboard	Passed	Not Run	Not Run	Passed	Passed
4	Samsung Note 4	Android - Cardboard	Passed	Not Run	Not Run	Passed	Passed
5	Samsung Note 4	Android - Cardboard	Not Implemented	Not Implemented	Not Implemented	Passed	Failed #10
6	Samsung Note 4	Android - Cardboard	Passed	Not Run	Not Run	Passed	Passed
7	Samsung Note 4	Android - Cardboard	Not Implemented	Not Implemented	Not Implemented	Passed	Passed
8	Samsung Note 4	Android - Cardboard	Passed	Not Run	Not Run	Passed	Passed
9	Samsung Note 4	Android - Cardboard	Not Implemented	Passed	Passed	Passed	Passed

1	Samsung Tab S2	Android - Cardboard	Passed	Not Run	Not Run	Passed	Passed
2	Samsung Tab S2	Android - Cardboard	Passed	Not Run	Not Run	Passed	Passed
3	Samsung Tab S2	Android - Cardboard	Passed	Not Run	Not Run	Passed	Passed
4	Samsung Tab S2	Android - Cardboard	Passed	Not Run	Not Run	Passed	Passed
5	Samsung Tab S2	Android - Cardboard	Not Implemented	Not Run	Not Run	Passed	Passed
6	Samsung Tab S2	Android - Cardboard	Passed	Not Run	Not Run	Passed	Passed
7	Samsung Tab S2	Android - Cardboard	Not Implemented	Not Run	Not Run	Passed	Passed
8	Samsung Tab S2	Android - Cardboard	Passed	Not Run	Not Run	Passed	Passed
9	Samsung Tab S2	Android - Cardboard	N/A	N/A	N/A	N/A	N/A

9.1.5. Test Results – Colin Crowell

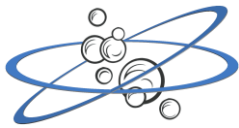
Tester:		Colin Crowell					
Test Case ID	Device	Environment	1/28/2017	2/17/2017	3/8/2017	3/20/2017	4/7/2017
1	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed	Passed
2	Google Pixel	Android - Daydream	Not Implemented	Passed	Passed	Passed	Passed



3	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Passed	Passed	Passed
4	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Passed	Passed	Passed
5	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Failed	Passed	Passed
6	Google Pixel	Android - Daydream	Hard Code Pass	Hard Code Pass	Passed	Passed	Passed
7	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Not Implemented	Passed	Passed
8	Google Pixel	Android - Daydream	Not Implemented	Passed	Passed	Passed	Passed
9	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented

9.1.6. Test Results – Experts

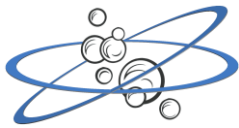
Tester:	Expert - Multiple People - Each Date column signifies a different person. Thus you can have the same date twice						
Test Case ID	Device	Environment	3/8/2017	3/8/2017	3/8/2017	3/8/2017	3/8/2017
1	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed	Passed
2	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed	Passed
3	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed	Passed
4	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed	Passed
5	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
6	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed	Passed
7	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
8	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed	Passed
9	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented



Tester: Expert - Multiple People - Each Date column signifies a different person. Thus you can have the same date twice						
Test Case ID	Device	Environment	3/8/2017	3/8/2017	3/8/2017	3/26/2016
1	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed
2	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed
3	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed
4	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed
5	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Not Implemented	Passed
6	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed
7	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Not Implemented	Passed
8	Google Pixel	Android - Daydream	Passed	Passed	Passed	Passed
9	Google Pixel	Android - Daydream	Not Implemented	Not Implemented	Not Implemented	Passed

9.1.7. Test Results – Novices

Tester: Novice - Multiple People - Each Date column signifies a different person. Thus you can have the same date twice				
Test Case ID	Device	Environment	3/5/2016	4/9/2016
1	Huawei Honor 8	Android - Cardboard	Not Run	Failed #11
2	Huawei Honor 8	Android - Cardboard	Not Run	Pending
3	Huawei Honor 8	Android - Cardboard	Not Run	Pending
4	Huawei Honor 8	Android - Cardboard	Not Run	Pending
5	Huawei Honor 8	Android - Cardboard	Not Run	Pending
6	Huawei Honor 8	Android - Cardboard	Not Run	Pending



7	Huawei Honor 8	Android - Cardboard	Not Run	Pending
8	Huawei Honor 8	Android - Cardboard	Not Run	Pending
9	Huawei Honor 8	Android - Cardboard	Not Run	Pending

1	Google Pixel	Android - Daydream	Failed #1	Not Run
2	Google Pixel	Android - Daydream	Passed	Not Run
3	Google Pixel	Android - Daydream	Passed	Not Run
4	Google Pixel	Android - Daydream	Passed	Not Run
5	Google Pixel	Android - Daydream	Not Implemented	Not Run
6	Google Pixel	Android - Daydream	Passed	Not Run
7	Google Pixel	Android - Daydream	Not Implemented	Not Run
8	Google Pixel	Android - Daydream	Passed	Not Run
9	Google Pixel	Android - Daydream	N/A	Not Run