EXPIERIENCE

Cal Poly Computer Engineering Senior Project

By: Emily Nguyen, Albert Chen, Seong Chang

Advisor: Professor Lynne Slivovsky

May 24, 2017

TABLE OF CONTENTS

PROJECT OVERVIEW/EXECUTIVE SUMMARY

The name of this project is called “exPIERience” and the technological field of focus is virtual reality. We will be working on creating a virtual reality experience using the Oculus Rift for the Center of Coastal Marine Sciences (CCMS). The purpose of this system is to enable people to experience diving at the Cal Poly Pier even during nonoptimal diving conditions which occur frequently. CCMS hopes to attract further interest and possible momentum in future outreach such as a “Live Dive” program, where live videos of divers under the pier can be streamed. A 360° video from the diver’s perspective will be recorded during the diver’s journey through the water. These captured images will be recreated into a 3D space that will simulate an experience as if the user of the headset is the actual diver. We will need to build a camera system that will properly record an entire 360° view of the space that the diver experiences first hand. In addition to this system, an easy-to use control system must be built for proper navigation throughout the 3D space. We hope to deliver a design that will enable users of all expertise to operate the device and have the same user experience.

CUSTOMER DEVELOPMENT

Clients/Customers

This project is developed for the Center for Coastal Marine Sciences (CCMS). Specifically, it should benefit the Cal Poly Pier by providing a tool that they can use to promote their goals of education and community outreach. CCMS provides special research opportunities to the students and faculty of Cal Poly, and what we build for them will help spread the benefit of the pier to a larger audience. The outcome of this project will given be to CCMS to be kept at the pier for visitors, and will also be able to be taken to off-pier events for people to experience scuba diving under the pier. CCMS as our client is represented by the following people:

- Jason Felton: jfelton@calpoly.edu - Diving Safety Officer/Pier Technician
- Crow White: cwhite31@calpoly.edu - Faculty and Researcher
- Maya Vavra: mvavra@calpoly.edu - Scientific Diver

Client's needs: Our clients need a portable system that captures everything under the Cal Poly Pier; this includes the ocean, the living species in it, the sediments, the pylons, and the experience that one perceives as if they were there. Specifically, the system's experience must highlight individual species at every strata along the pylons. This solution must be feasible, cost efficient, and intuitive for any type of user. At the minimum, our clients need an Oculus Rift with six GoPro cameras, and a computer system powerful enough to provide a positive experience for the Oculus user. Then, our clients also need an environment where they can run the demos repeatedly without any complications from the hardware and software components; the setup and the process should be as easy as turning on the program and putting on the equipment. Our clients should not be concerned about how the users will move under the water, but rather rely on the fact that the system will be intuitive for them. The experience ought to focus on bringing inspiration to all users including students, faculties, and any other visitors to the pier.

Stakeholders

The following people have an interest in the project but are not directly involved:

- SLO county public (i.e. students, teachers, educators)
- Cal Poly Community
- Cal Poly Pier Visitors
- Center for Coastal Marine Sciences

The Cal Poly Pier's virtual reality system serves as an academic tutorial for students to learn about the pier and marine science. The teachers can use this opportunity to expedite their teaching plans with "learn by doing" in mind. The Cal Poly community and visitors can use this system to fully experience the pier and the ecosystem under it. The CCMS can now have the virtual reality system added to their feature for the organization's needs and purposes for the future.

Stakeholder's needs: The stakeholders need an environment which provides real-life experience as a diver under the Cal Poly Pier and have the opportunity to be inspired by everything that goes on around it. But this environment must be burden-free, meaning the stakeholders should not have to dive to experience the Cal Poly Pier. They should be able to fully replicate this experience by putting on an Oculus and swim around at the touch of a controller or using a motion-detection sensor. The stakeholders require a relatively short demo but something meaningful to instill inspiration and knowledge. The stakeholders also need this project to provide ideas for future engineers who will implement another virtual reality system, and potentially serve as a tutorial with education in mind for future divers. The demo will be user friendly and easy to use since we are assuming that the stakeholders may not have much experience with the oculus and/or computers.

Framed Insights and Opportunities

After our visit to the pier, it's obvious that the pier is a special place that provides the students and faculty at Cal Poly with unique opportunities to research and experience the local marine environment. Despite being so great, it can lack exposure to the community at large due to its location and lack of capacity for regularly large amounts of visitors. The large number of visitors that do come on open house days, however, does show a great interest in what the pier is about. This project can help to bridge the gap between interest and accessibility by allowing people to experience a part of the pier no matter where they are. This appeals even to visitors at the actual pier, as they will be able to see something normally only licensed divers can have a chance to see.

This project was first inspired by the possibility of implementing a program similar to that of the Channel Islands Live Dive for CCMS. While an implementation of such a program is still in the works in years to come, this project will capture the inspiration that viewing and interacting with underwater environments can have. A successful implementation of this project will help to propel Live Dive and any other future CCMS outreach projects into realization.

Project Goals and Objectives

Based on conversations with our clients and our developing approach, we formulated the following goals and objectives:

Goals:

- To help CCMS raise awareness for the Cal Poly Pier and enable people to experience diving at the pier even during nonoptimal diving conditions.

Objectives:

- To build a Virtual Reality System using the Oculus Rift that simulates a diving experience under the Cal Poly Pier.
- Build a camera system that will capture an entire 360 video of the view from a dive underneath the pier.
- Use captured footage to recreate a 3D space that will simulate an experience as if the user of the headset feels like they are actually there.
- Develop an easy-to-use control system for navigating within the 3D space.
- Build a user interface for launching and running this program that does not require engineering expertise.
- Create a portable solution that can be moved and set up from place to place without special accommodations.

Project Outcomes and Deliverables

<u>Outcomes</u>: This project will help generate interest and momentum in future CCMS outreach such as a "Live Dive" program that allows for videos to be recorded from divers inside the water and live streamed on an external system. This will also provide both the Cal Poly community and the community at large with an understanding and appreciation of the marine environment and CCMS.

<u>Deliverables:</u> We will need to create a functional camera system that can fully capture the 360 view of the water from the diver's perspective. Our system will comprise of multiple cameras focused on a single area so we will need to combine multiple videos from different perspectives all together to create the view. When the project is finished there will be a virtual reality program with a user friendly interface that simulates an underwater diving experience at the Cal Poly Pier. Users will be able to explore the water with a 360 degree view. Users will also be able to navigate through the 3D space using a control system. CCMS will be given both the Oculus and the accompanying computer to run the program. Any documentation relevant to running this program will also be provided.

Duration

The planning of this project began in the middle of Fall Quarter 2016 as a Computer Engineering Capstone Project. During Winter Quarter, the Capstone group (which included two members not included in the senior project group) worked on creating and testing a 360 camera system, as well as designing and putting together the virtual world. This quarter, Spring Quarter, the Senior Project Team (Emily, Seong, Albert), worked on adding more features and improvements. The first milestone was constructing the image capturing system out of a matrix of go-pros. This is because the water is likely to be most clear in mid-winter, and a capture system must be ready when that happens. The second milestone was a proof of concept by creating 3D space using a footage of a classroom where the user will be able to explore with a 360 degree view it will help us transfer what we learned to building a similar system but underwater. Following the proof of concept, constructing the 3D space of underneath the pier, and overlaying it with the captured dive images, will be the next milestone. The third milestone was to ensure that the virtual world was as accurate as it could be, with proper sizing, spacing, and movement underneath the Pier. As the project draws to a close, packaging software into a user-friendly executable will be the final milestone.

BACKGROUND

Competitors include other organizations who seek to educate the public about diving. Some may do so through diving with a camera and streaming it live or simulating a dive through a video, virtual reality, or any other medium. A national park, Channel Islands, broadcasts a live video of an underwater dive by camera footage on their website. More similar to our system are other virtual reality programs whose goal is to provide a simulation of an experience that a person may not be able to experience without some degree of difficulty or risk such as rock climbing, sky diving, etc. The standard testing and validation procedures for the system are to ensure that we are able to record footage with a GoPro system and stitch it together to create a 360° degree video. We will also use photos and videos collected during the dive as a reference to render a virtual world where users can have a more interactive experience by enabling them to move up and down a pylon underneath the pier. The procedures that are tested will be that the user may start the demonstration, descend and ascend in the virtual ocean, look around and observe the pylons, and move around in the virtual world.

Camera Capture System

The initial goal of this project was to provide a realistic experience of an underwater environment for the user and promote education. The Channel Islands National Park recently launched a system that accomplished that experience through their live dive program. We began by analyzing the Channel Islands' live dive system as well as look through the information provided on their website to gain some insight as to what type of virtual reality experience we would like to provide to the user. From the live dive system, we designed a 360 camera system with 6 GoPros--one attached to each side. The rig that held the system together was based off of blueprints for 3D models found online. The goal was to stitch together the 6 views captured by the system to form a 360 video. This camera system went through three phases of testing--land, underwater in a swimming pool, and underwater in an aquarium. Although testing the camera system underwater was successful, stitching together the actual footage was not. There were many spots that our camera system did not capture, so the video stitching resulted in many blank spots. We realized that the underwater housings for the cameras blocked off some of the camera's visibility and to have the video stitching be more accurate we would need to add a few more GoPros to cover the missing areas.

Luckily, we acquired an actual 360 camera known as Vyu360 shortly after our final round of testing at the aquarium. This camera makes the video stitching much easier as it comes with a 360 video stitching application. Additionally, Vyu360 consists of two 180 cameras, back to back, so it fully captures a 360 view. Vyu360 comes with a mobile application that allows users to display what the camera is seeing in live time. However, this feature is not waterproof, because regular signals and waves cannot pass through the water. So whenever the footage is taken underwater, the live feed feature was never used. Instead, we relied on our diving team to capture a good footage before testing it. For demonstration purposes, the quality of the video is acceptable but definitely not optimal for professional use.

Computing System

Before deciding on a platform to run the Oculus, we visited the Cal Poly Innovation Sandbox to try their Oculus system and get an idea about performance and the specifics of owning and operating an Oculus Rift. They were running it on a Mac Pro with a dedicated graphics card, but the performance was underwhelming. The experience was not enjoyable because of the low framerate, and it was clear that we would need to select system specifications on our own platform that didn't fall short of the Oculus recommended specifications (as they did at the Sandbox).

After browsing pre-built, "VR ready" desktops, it was clear that they were overpriced and underpowered for our needs, and we would need to part out and build our own. We took our minimum specifications (from the Oculus website) to Robert Randle, the EE department IT analyst, and he assisted in selecting components that met these requirements, but also would perform reliably in the long-term in order to avoid future problems for our client as the computer ages.

Oculus Rift Operation

Besides the insight about computer specifications, the Innovation Sandbox also brought to our attention the need to have a procedure for introducing new users to the Oculus Rift. While there, we were given a short safety brief and instructions on wearing the headset, which we drew heavily from when we introduced our clients to the system for the first demo. This is especially important given that the users of our application can be of various ages and have little to no experience with virtual reality headsets.

Virtual Reality Content

After getting an idea of what the objective of our project was, we started researching how to create our own virtual reality content. Given that virtual reality is such a new technological space, there are currently two main types of content. One is creating your own 360° photos and videos. The other is by using virtual reality software development applications to render your own world. The first option is more photorealistic and straightforward to implement. The second option has a higher learning curve but provides a more immersive and interactive experience in comparison to the static photos and videos. To be able to provide both experiences and maximize our learning opportunity, we will be implementing both applications. We will be putting together a 360° photo and video experience as well as create a virtual experience that enables a user to swim up and down the pylons.

The 360° photos and videos will be created using a GoPro camera system where 6 GoPros will be mounted together in a spherical form. Then by stitching these photos and videos together, we can create custom 360° content for our users.

The rendered virtual will be created using a game engine in order to reduce the amount of time it takes to build our own VR experience. The following game engines provide Oculus integration: Epic, Unity, Crytec CryENGINE, Autodesk Stringray, MaxPlay, and Amazon Lumberyard. The virtual mode, also called the interactive mode, will have different kinds of sea organisms, pylons, a pier, rocks, plants, and the user. The ocean will have some type of underwater effect to simulate the real life visibility under the water. In essence, the visibility of the user will be reduced but only to an extent so that the user can still see sea creatures and other pylons. The user will be able to ascend and descend in the ocean to get to various places under the pier. The pier and the pylons in the beginning stages of development were not to scale. Currently, everything is resized to scale. The user should notice that the distance from one pylon to the next is about the same. The fish in the ocean moves around randomly in a given field of area. There is a seal now in the ocean but is a static object, unlike the fish. In later developments, the seal will be able to interact with the user or at least swim around in the ocean.

ENGINEERING SPECIFICATIONS

Customer Requirements

The customers require a safe environment free of unnecessary distraction when using the Oculus Rift. The process of putting on the gear needs to be effortless and straightforward, while the swimming process in the ocean should be intuitive and natural for anyone that uses it. The Oculus must be kept clean at all times for clear visibility through the lenses and provide comfort for every part that comes in contact between the user and the machine.

The Oculus must provide a real-life experience under the Cal Poly Pier that will enlighten the customers with interactions that occur beneath the pier. The experience that the Oculus brings to the customer should consist of entering, swimming and exploring the ocean. In each of these steps, the customer can observe every detail of the pier including the living species that roam about and be inspired by the world they are virtually immersed in.

The final system must also be well-documented and portable so that our client can transport it from place to place without special accommodation, and learn to set it up without help in a short (10 minute) amount of time.

Engineering Requirements

Spec. Number	Parameter Description	Requirement or Target with Units	Tolerance	Risk	Compliance
1	Weight	30lbs	Max	L	A, T
2	Size	19.45” x 8.27” x 19.57”	None	L	I, T, A
3	Production Cost	$1200	Max	L	A
4	Power	650W	Min	M	A, T, I
5	Environment	80 PLUS Gold Certified	None	L	A, T, S, I
6	Functionality	Virtual Reality	None	L	A, T, S, I
7	Usage Time (Health/Safety)	30 mins	Max	H	A, I
8	System Life Span	4 years	Min	L	A, T, S, I
9	Frame Rate	30 fps	Min	M	A, T, S, I
10	Steps/Ease of Use	4 steps	Max	M	A, T, S, I
11	Peripherals	Controller and Headset	N/A	L	A, T, S, I
12	Set up	10 mins	Max	L	A, T, S, I
13	Breakdown	10 mins	Max	L	A, T, S, I

Engineering Requirements Table

Specification #1: Weight

Computer Weight

One person should be able carry all of the components (though not necessarily all at once). Should be less than 30 pounds.

GoPro Rig Weight

One person should be able carry all of the components (though not necessarily all at once). Must be less than 10 pounds.

Specification #2: Size

GoPro Rig Size

This rig should be placed on some sort of pole or holder so that one person can simply hold it in his/her hand. 6x6x6 in.

GoPro Rig Pole Size

One person should be able to hold the GoPro Mount on a pole with a single hand. No more than 7x5x1.2 in.

Specification #3: Cost

GoPro Cost

There must be at least 6 GoPros in the system, depending on the model each costs anywhere from \$200-\$400. The cost should be no more than \$2400. The Computer Engineering Department had GoPros from previous projects. Thus, no additional purchases for these cameras were made for this project.

GoPro Rig Cost

The cost to create the rig should be relatively inexpensive. There are many models for 360 GoPro rigs available online that go up to nearly $2000. For the scope of this project and budgeting reasons, a rig was 3D printed following one of the models found online. The price of 3D printing was $15. The cost of the GoPro Rig for this project should be less than $2000.

Specification #4: Power

The power usage of the system should be 650 Watts.

Specification #5: Environment

The system should meet environmental standards, following 80 PLUS Gold Certified.

Percentage of rated load	20%	50%	100%
80 Plus Gold	87%	90%	87%

Representation of PSU Rating (And how well it converts AC power into DC power in percentage)

**Any AC power not translated into DC power is given off as heat so ideally, users want 100%

Specification #6: Functionality

The system should function as a virtual reality program.

Specification #7: Health and Safety

The system shall not harm the user in any way while using it given preconditions and precautions.

The headset requires an unimpaired sense of motion and balance. Do not use the headset when you are: Tired; need sleep; under the influence of alcohol or drugs; hung-over; have digestive problems; under emotional stress or anxiety; or when suffering from cold, flu, headaches, migraines, or earaches, as this can increase your susceptibility to adverse symptoms.

Remain seated unless your game or content experience requires standing. Serious injuries can occur from tripping, running into or striking walls, furniture or other objects, so clear an area for safe use before using the headset. Take special care to ensure that you are not near other people, objects, stairs, balconies, open doorways, windows, furniture, open flames, ceiling fans or light fixtures or other items that you can bump into or knock down when using—or immediately after using—the headset. Remove any tripping hazards from the area before using the headset. Remember that while using the headset you may be unaware that people and pets may enter your immediate area. Do not handle sharp or otherwise dangerous objects while using the headset. Never wear the headset in situations that require attention, such as walking, bicycling, or driving.

In total, the Oculus Rift should not be used for more than 10 minutes at a time. The maximum time spent exploring each of the modes should be no more than 5 minutes. For demonstration purposes at a booth where there is a long line of users waiting to use the program, each mode should take 3 minutes maximum.

Specification #8: System Lifespan

The system should last for more than 4 years.

Specification #9: Frame Rate/Performance

The frame rate of the VR should not distract the users from the virtual reality experience.

The required minimum target is 30 frames per second, although the ideal target is 60 frames per second. Energy consumption from the PC and VR headset will vary depending on usage load.

Specification #10: Usability/Ease of Use

The people who will be running this demonstration might not be engineers who are familiar with the system. As a result, the process of setting up the oculus and launching the demonstration needs to be as intuitive and easy as possible. In addition, this system is planned to be moved between multiple locations, set up and broken down each time. This process should not require special accommodations other than a car. This can be verifiable.

We can make it less ambiguous by reducing the steps necessary to get the system up and running. The high level of the steps look like the following:

Step 1: Turn on the PC.

Step 2: Turn on the program.

Step 3: Put on the VR.

Step 4: Play the program

Specification #11: Peripherals

The system requires two peripherals to function optimally. This requires a controller and a headset.

Specification #12: Setup

The person shall take no more than 10 minutes to set up for a user acquainted with the system.

The steps are as follows:

Step 1: Place the monitor, desktop, Oculus, keyboard, and a mouse on a desk.

Step 2: Plug all the peripherals (monitor, Oculus, keyboard, and the mouse) to the desktop.

Specification #13: Breakdown

The person shall take no more than 10 minutes to breakdown for a user acquainted with the system.

The steps are as follows:

Step 1: Unplug all the peripherals (monitor, Oculus, keyboard, and the mouse) from the desktop.

Step 2: Place all the parts into their original boxes.

Step 3: Place all the wires and cables into their original boxes.

Other Specifications: Computer System

Minimum PC requirements:

- Graphics Card: GeForce GTX 970 or AMD Radeon R9 290 or better.
- CPU: Intel Core i5 4590 or greater.
- RAM: 8GB or more.
- Video port: HDMI 1.3.
- USB port: 2 USB 3.0 ports.
- Windows 7 SP1 or newer.

Given that the PC remains dust-free for the remainder of its usage, it is expected to function for 5-6 years. The price should be around $950. Energy should be in between 550W to 650W. It must be Gold certified with 70% - 80% useful energy, 20% - 30% wasted energy.

Use Cases

Case ID	Case Name
1	Begin the exPIERience
2	User Descending
3	User Ascending
4	User Looking Around
5	Observing Pylon
6	User Moving Around

Use Case Summary

Case ID:	1
Case Name:	**Begin the exPIERience**
Created By:	Eva Chen
Last updated By:	Seong Chang
Date Created:	November 8, 2016 4:00 pm
Date Last Updated:	December 1, 2016 5:15 pm
Actors:	Operator
Description:	The operator starts the diving experience.
Preconditions:	Oculus must be correctly cabled to the PC that will be running the software. Documentation must be available to assist if needed.
Postconditions:	The Oculus is on and ready for use by the user.
Normal Flow:	Step 1: Operator launches software from an executable Step 2: The system responds by opening up an opening screen. Step 3: The operator begins the demo tour.

Alternative Flows:	N/A
Exceptions:	N/A
Assumptions:	Basic computer literacy on part on part of operator.

Case ID:	2
Case Name:	**User Descending**
Created By:	Emily Nguyen
Last updated By:	Seong Chang
Date Created:	November 8, 2016 4:00pm
Date Last Updated:	November 15, 2016 4:00pm
Actors:	User
Description:	User descends under the pier.
Preconditions:	The user is wearing the Oculus headset. In the virtual reality world, the user is in the water and not at the bottom of the ocean. The user must be near a pylon and has access to the controller.
Postconditions:	The user is deeper in the water than before.
Normal Flow:	Step 1: The user pushes button for descending on controller while near a pylon. Step 2: The system depicts the user descending.
Alternative Flows:	Step 1a: The user presses descend button not near a pylon, which is not supported. Step 1b: No vertical movement occurs.
Exceptions:	If the user is at the bottom of the ocean floor, then the user won't be able to sink further down.
Assumptions:	N/A

Case ID:	3
Case Name:	**User Ascending**
Created By:	Emily Nguyen
Last updated By:	Emily Nguyen
Date Created:	November 8, 2016 4:00pm
Date Last Updated:	November 15, 2016 5:00pm
Actors:	User
Description:	The user ascends towards the pier.
Preconditions:	The user in virtual reality must be towards the bottom of the sea. The user must be near a pylon and has access to a controller.
Postconditions:	The user will be closer to the surface of the water from before.
Normal Flow:	Step 1: The user presses a button on the controller for ascending. Step 2: The user floats up towards the surface of the ocean.
Alternative Flows:	Step 2a: The user will swim upwards relative to where he/she is facing.
Exceptions:	The user cannot swim upwards when already on the surface.
Assumptions:	The user's point of view will not be upside down. It is assumed that the point of view will be as if they are standing upright in the water.

Case ID:	4
Case Name:	**User Looking Around**
Created By:	Emily Nguyen
Last updated By:	Emily Nguyen
Date Created:	November 10, 2016 4:57pm

Date Last Updated:	November 15, 2016 4:57pm
Actors:	User
Description:	The user is looking around in the virtual world. The view should change in the direction that the user moves his/her head.
Preconditions:	User must have the program running, wearing the Oculus and looking into the lenses.
Postconditions:	The user will see the virtual world around them.
Normal Flow:	Step 1: As the user moves their head, the view changes based on the direction the user is facing.
Alternative Flows:	None.
Exceptions:	None.
Assumptions:	The user is able to move around their neck and have vision good enough to see through the lenses.

Case ID:	5
Case Name:	**Observing the Pylon**
Created By:	Emily Nguyen
Last updated By:	Emily Nguyen
Date Created:	November 10, 2016 4:57pm
Date Last Updated:	November 15, 2016 5:20pm
Actors:	User
Description:	User sees the pylon as well as the details and specimen along it.
Preconditions:	User must have the program running and the oculus on. User at least 5 feet away from pylon.
Postconditions:	User will be able to see the details of the pylon and specimen very clearly.

Normal Flow:	Step 1: The user can get a detailed view of the pylon by looking closely at it with the headset.
Alternative Flows:	None.
Exceptions:	Pylon must be seen in very high resolution, and very detailed. User must be at least 5 feet away from the pylon in order to get a clear image.
Assumptions:	The user is near the pylon, not above the water.

Case ID:	6
Case Name:	**User Moving Around**
Created By:	Emily Nguyen
Last updated By:	Emily Nguyen
Date Created:	November 10, 2016 4:57pm
Date Last Updated:	November 15, 2016 4:57pm
Actors:	User
Description:	The user moves around in the virtual world. The view should change in the direction that the user moves with the user interface (controller)
Preconditions:	User must have the program running, the oculus on and have a controller in hand.
Postconditions:	The user will travel in the virtual world from one spot to another.
Normal Flow:	Step 1: The user moves around the left analog stick. Step 2: The user swims in the direction that the left analog stick is pointing to.
Alternative Flows:	Step 2a: The user swims in the direction of the left analog stick relative to where they are facing. Step 2b: The user swims collides with the pylon and cannot move forward.

Exceptions:	None.
Assumptions:	User understands how to use controller. The user isn't colliding with an immovable object.

Personas

There are several personas that we can expect as end-users of this system. They include Cal Poly students, Cal Poly parents, and children. Students are roughly analogous to young adults, while parents can be compared to members of the general public in the central coast.

1. **Cal Poly Students:**

   A typical student at Cal Poly has a basic understanding of the organization of the university and is most likely aware of the existence of the pier (although not expected to have visited before). They would be interested in this project in order to learn a bit more about what the school has to offer, and also to see something interesting. They are the most likely group to have had VR exposure in the past, and adjusting to the system interface and control should be a simple task if given basic instructions. It can be expected that this group of end-users will push the limits of the system, attempting to go beyond the boundaries we design in.

2. **Cal Poly Parents:**

   Parents of Cal Poly students are expected to have similar knowledge about the university compared to students, but probably will not be aware of the existence of the pier until being exposed to this project. They would be interested in this project in order to see where all the tuition they pay is going, and to be able to tell folks back home about what kind of interesting things their child's university has to offer. They are not likely to have had VR exposure before, and may not even be intuitively comfortable with control schemes the system will offer. They are expected to be able to follow directions well, however, and probably won't push the limits of the system.

3. **Children:**

   Children are expected to be completely naive to both the university and the pier. They are probably not there by choice, but would nevertheless be interested in this system in order to see something cool and break the monotony of being dragged somewhere with their parents. They most likely have not experienced anything in VR yet, but will intuitively be more comfortable with it and any control schemes offered. Children will bore easily if the experience is not engaging enough, and they will almost certainly not be reading or following written instructions.

4. **The Public (Others in the community besides students such as Seniors, Parents, etc.) :**

   This section is added due to the experience we had at the Cal Poly Pier's Open House event. Many visitors checked out the exPIERience booth and tried the program. Most visitors had no background information on the pylons nor the organisms that lived on it before visiting the booth. The visitors became more interested in the purpose of the pier by the time they left the booth. A few of the visitors were retired workers that worked on or near the pier, and they were encouraged by the purpose and goals behind the exPIERience project.. We realized by the end of the day that the project not only informed the visitors about the pier and its purpose, but also brought more interests to the pier for people who did not care for it all before.

In order to meet the varying experiences and competencies of this group of users, care must be taken in creating a control scheme that is easy to explain and easy to start using. In addition, a protocol for guiding each user through the standard operation of a VR headset (such as safety, comfort, etc.) needs to be developed for the end delivery of this project. Making sure that the system is engaging and visually interesting is also key in capturing the interest of each of these personas.

DESIGN DEVELOPMENT

Oculus Rift

The Oculus Rift was available for the team since the beginning of the project and has been the base of our design concept. The system allows users to view things in virtual reality, and this satisfies the requirement for this project because our clients need a system that captures everything under the Cal Poly Pier in virtual reality.

The Oculus Rift allows other components to build off it, which provides options along the way. The set came with a headset, an external sensor, a remote control, and a Xbox controller. While every component excluding the controller is needed for basic usage of the headset, the controller opened up the option for the user to move in the ocean. Basic controls for movements like ascending, descending, and swimming forward in the water will be controlled by the Xbox controller.

We did not spend any expense for the headset along with its components due to its accessibility from our professor. We ran preliminary tests with the headset by running a demo with our PC system and looked for any lags, inconsistency in graphics, and smoothness of the program.

Base of the Oculus Rift Headset
(External Sensor and Remote Control Not Shown)

PC System

The Oculus headset required the team to purchase a powerful PC system that could run the Oculus program. The idea behind the PC build was meeting the PC requirements as listed on the Oculus website.

We first picked our core components of the build like the graphics card, CPU, and a motherboard. We made sure those components were compatible with one another. Then other parts were chosen, such as RAM, power supply unit, case, and CPU fan, which are easier to integrate into the system.

The main things we took into consideration were providing the system with enough power to prevent the computer failing, purchasing a powerful graphics card that can run the program smoothly for ultimate experience, and picking out stable components that can last the user for years to come without the system dying out quickly.

For preliminary testing, we ran the Oculus demo with the system, and looked for any lags, inconsistency in graphics, and smoothness of the whole program. The total estimate of the system came out to be around $1180.

Overview of the PC system

Item	**Notes**	**Potential Expense ($)**	**Actual Expense ($)**
PC Components			
Hard Drive			79
Case			110
Motherboard			105
Memory			76
Processor			235
GPU			250
Power Supply			90
Monitor			200
Cooling System			35
Subtotal			1180

Overall PC System Expenses

Six GoPro Cameras with a Mount

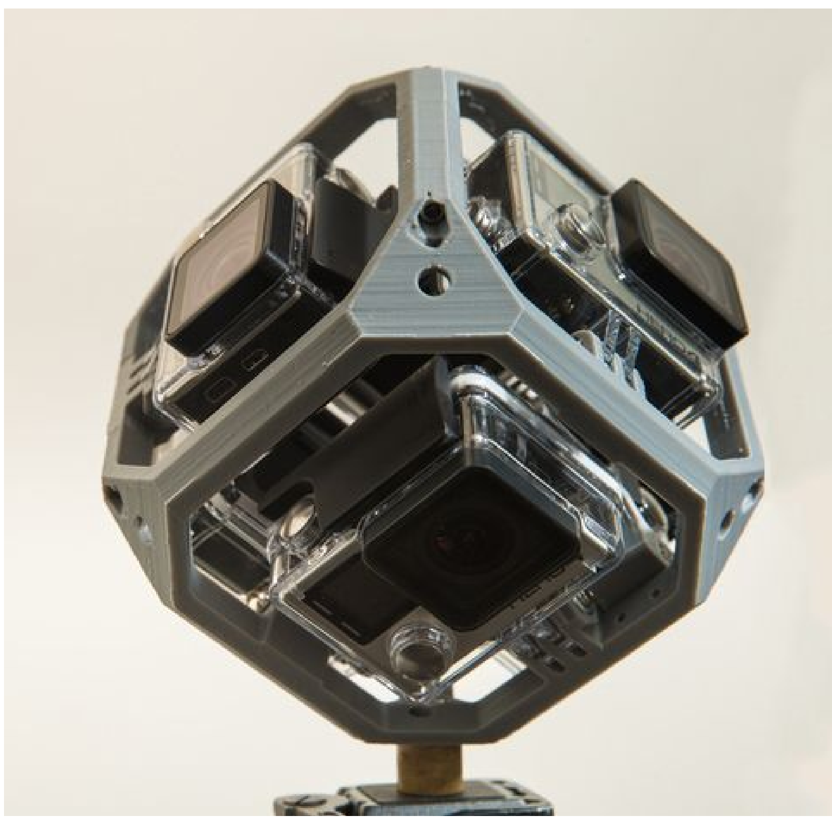
We chose to use six GoPro cameras for this project because we had six cameras to use at our disposal. Six cameras allow optimal view of the world (360^o by 180^o) with cameras facing at every core viewpoint: front, back, left, right, up and down. The six GoPro configuration also provides easier mount prototype due to its inherent geometric attributes: a cube with six faces.

Six camera configuration is also optimal for our clients since the expenses to require six GoPro is achievable compared to different configurations such as 12 or 18. Anything under six camera configuration is ideal for very low budget but this would harshly limit the user's viewing capabilities.

The mount design concept followed from the six GoPro configuration. Since we had access to 3D printer, we needed to find a model that satisfied our configuration. On top of making the mount to satisfy the six camera configuration, it needed to be light, sturdy, small, waterproof, and allow waterproof cases to fit.

Our first iteration of the mount prototype sufficed everything except allowing waterproof cases to fit. It satisfied enough requirements to capture the 3D world space of our class. Since it was an initial prototype, the first iteration was never 3D printed and thus never spent any expenses on it.
Our final iteration satisfied all the requirements including the waterproof capability. We used acrylonitrile butadiene styrene (ABS) plastic filament to 3D print our mount. ABS was used because it is tough and durable to hold six cameras underwater and it is cheap to recreate the prototype. The mount cost around $10 to 3D print with the ABS material. However, we need to run enough tests underwater with all six cameras and cases before diving into the pier for capturing videos.

First Iteration of GoPro Mount (Not Compatible with Waterproof Housings)



Final GoPro Mount (Compatible with Waterproof Housings)

Item	Notes	Potential Expense ($)	Actual Expense ($)
Camera System			
GoPros (x6)	We already have some that we can use. The price ranges based on which model we get. We ended up purchasing one more GoPro in order to get all of them to be the same model (for the purpose of syncing the settings).	$200 - $400 per camera	175
GoPro Mount	Mount is 3D printed, and we may require more prints as needs develop. In the end, we printed 2 mounts so that we could have a backup for the monterey dive.	15	20
GoPro Housings (x6)	Going to try and find some we can use. If we do need to buy some, it depends on if we want to buy third party accessories or not. http://shop.gopro.com/accessories-2/standard-housing---131-40m/AHSRH-401.html	50	0
SD Cards (x6)	We purchased 2 new 64GB micro sd cards. The rest were borrowed from the Computer engineering Department and the CCMS.		55.20
Subtotal			250.20

Overall GoPro and Mount expenses, including money spent and not spent for this project

Vyu360 Camera

While the GoPro rig is economically efficient, it has presented challenges in setup and post-processing. It takes a great deal of effort to charge up, start, synchronize, and then stitch each GoPro's respective video into a finished product. Midway through the project, Professor Slivovsky acquired a new camera system, "Vyu360", that is dedicated to capturing underwater spherical video. This system makes the setup and post-processing of videos almost negligible. The downside with this system is that it provides a video with lower resolution than what the GoPros can provide. This system may ultimately end up being our final mode of capture if the GoPros prove too cumbersome to be used effectively in our timeframe.

Item	**Notes**	**Potential Cost ($)**	**Actual Cost ($)**
Camera System			
Vyu360	Lynne Slivovsky pledged a kickstarter project for this camera and it arrived toward the middle of winter quarter.	$200	$200

Vyu360 Potential and Actual Expenses

Top Design Concept

We have decided that we are moving forward with a system that divides the user experience into two modes: a realistic diving mode that is minimally interactive but highly immersive, and an interactive mode that is less immersive, but allows the user control over their movements in the VR environment.

1. **Immersive dive mode:**

   This mode can be described as an "on-rails" experience. That is, the user is following a set movement path that they are not able to deviate from, but are still able to look around in all directions for the full VR experience. Other aspects of interactivity can be added in, such as optional information about what the user is seeing, or video options like pause and rewind. This mode is essentially a playback system of video runs our GoPro system captures in a dive underneath the pier. It is contingent on capturing good video runs during the dive, and being able to faithfully stitch high-quality spherical (360) videos for playback. The Oculus Rift development API has tools for spherical video playback, and they are what we expect to utilize here.

2. **Interactive dive mode:**

   In this mode, a user will be able to navigate between pylons and follow a pylon up and down, as described in the use cases. The pylons themselves will be objects constructed in the Unity engine and will be detailed with the different strata of marine life at each level. The background (and potentially textures of the pylons) will be derived from the GoPro images, but have the potential of seeming not-quite-realistic (though still visually interesting).

The reason for this split into two modes is that manipulating spherical videos seems to be a task beyond the scope of this class, and so altering them to create an interactive environment is off the table. However, it is still potentially possible to have the full interactivity we desire through the use of static images. That is, we can collect a set of spherical images in known locations underneath the pier. At any given time, a user is viewing a spherical image (with additions to make the surrounding environment seem dynamic, such as ambient sounds and the occasional passing creature overlaid in front). When the user is not moving, it is photorealistic.

The issue here is that the number of locations that have a captured spherical image is discrete, while a user's movement and resulting location are continuous. The challenge in this case would be making the transition from one real captured spherical image to the next (from a user's movement) appear seamless. A non-seamless example of this is the "street view" feature in google maps when moving along the street. They simply blur the image as you transition from one actual capture to the next. The environment under the pier would be more forgiving than this, however, and the space we need to image would be smaller as well. This technique would be contingent on how many static images could be collected on a dive, and how well we can perform the transitions between those images. Because we have relative uncertainty about both of those factors, we will be pursuing the two separate modes and techniques in order to achieve our baseline functionality, with us exploring this technique if we are easily able to implement the others.

PC System

We had 3 main options for our PC that runs the Oculus Rift: A custom build using an Intel processor and NVIDIA graphics card, a custom build using AMD processor and graphics card, and pre-built desktops similar to the ASUS G11CD-51 (falling within our price range and all having similar specs). The pre-builts all underperformed with most of their components in comparison to the custom builds, but still had similar or higher price points. In the end, we selected the Intel/NVIDIA custom build because the Oculus is optimized for those components, and thus earned slightly more points despite similar performance in non-VR tasks.

PC	Intel/NVIDIA custom build	AMD custom build	"VR ready", pre-built ASUS G11CD-51
(0.4)Cost	6	7	6
(0.5) Processor	8	7	8
(0.7) GPU	8	7	6
(0.3)Memory	9	9	7
(0.2)Hard Drive	9	9	6
Total	**16.5**	15.7	13.9

Computing System Decisions Table

There currently aren't any camera solutions that let you automatically take 360° photos and videos. The closest solutions either don't work very well or are extremely expensive. As a result, it's more cost efficient to take multiple images or videos and stitch them together using specialized software applications. Virtual reality is also a very new and specialized field; as a result, there aren't a lot of options when it comes to finding a tool to help create 360° content.

For video editing and stitching, after doing a bit of research it became apparent that most people use one of a couple options. Given how expensive the software can be, we chose to try out the cheaper solution which is VideoStitch Studio. At the moment, we are using the free trial version of the software.

Video Stitching Software	VideoStitch	Kolor
(0.3) Cost	8	3
(0.7) Ease of Use	8	8
Total	**8**	6.5

Video Stitching Software Decisions Table

For photo editing, there are a few more options available as panoramic photo stitching has been available for some time. However, for specifically editing 360° photos, some applications work better than others. For instance, while photoshop is easy to use and more available to us, it also isn't made to stitch together panoramic photos that overlap. As a result, before merging the photos together, you have to manually crop them. PTGui is made for 360° degree photos and is more streamlined for that specific purpose. It is however significantly more expensive and harder to use due to lack of experience and available resources. Krpano offers fewer features and is harder to use. It also costs quite a bit of money. I think it would be something nice to have on top of PTGui, but it wouldn't be an application that I would choose to use instead of PTGui. Right now we are currently using the free trial versions of PTGui and Krpano.

Photo Stitching Software	Adobe Photoshop	PTGui	Krpano
(0.3) Cost	9	7	7
(0.5) Ease of Use	7	8	6
(.2) Features	7	9	5
Total	7.6	**8**	6

Photo Stitching Software Decisions Table

The market for mounts that hold 6 GoPros in their underwater housing is actually quite small, and so the costs for any commercial ones were very high. While they did offer higher sturdiness and potential configurability, they ultimately couldn't compete with the affordability of a 3D printed design pulled off of a free 3D model sharing site. Ironically, the design we chose to 3D print (costing $8 total) was one we came across later that was ripped off the site, copied, and sold commercially for $650. The features are not weighted in the table below because they all equally contributed to our decision.

GoPro Rig/Mount	360RIZE 360H6 Mount	Freedom360 Kolor Abyss	3D printed model
Sturdiness	8	9	6
Cost	2	0	10
Configurability	3	2	4
Total	13	11	**20**

GoPro Mount Decisions Table

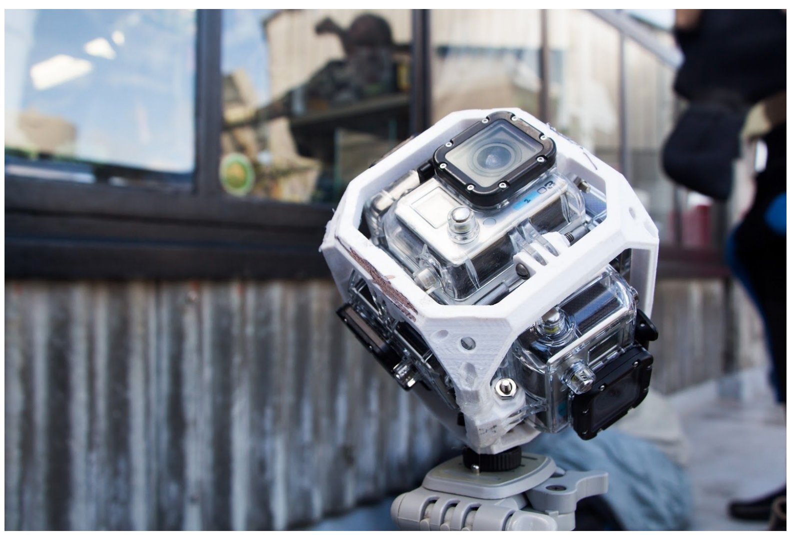
FINAL DETAILED DESIGN

Camera Systems

<u>GoPro Rig</u>

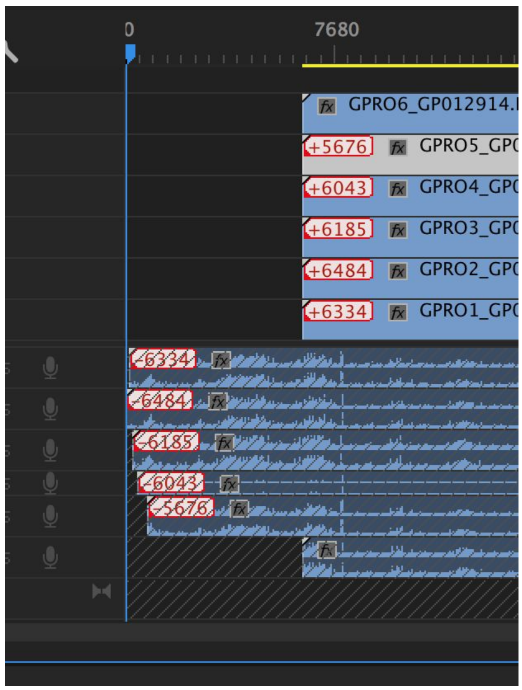
The final GoPro Rig had a 3D printed mount and 6 GoPro Hero cameras arranged in alternating directions. Settings that were used on all the GoPros include: 2K resolution, 4:3 aspect ratio, Protune on, Wide FOV, and 30 fps.

After testing this system during a dive at the Monterey Bay Aquarium, it has been determined that this system currently doesn't not meet the requirements needed for this project. Possible improvements to this system can be found in the "Overall System Analysis" section.



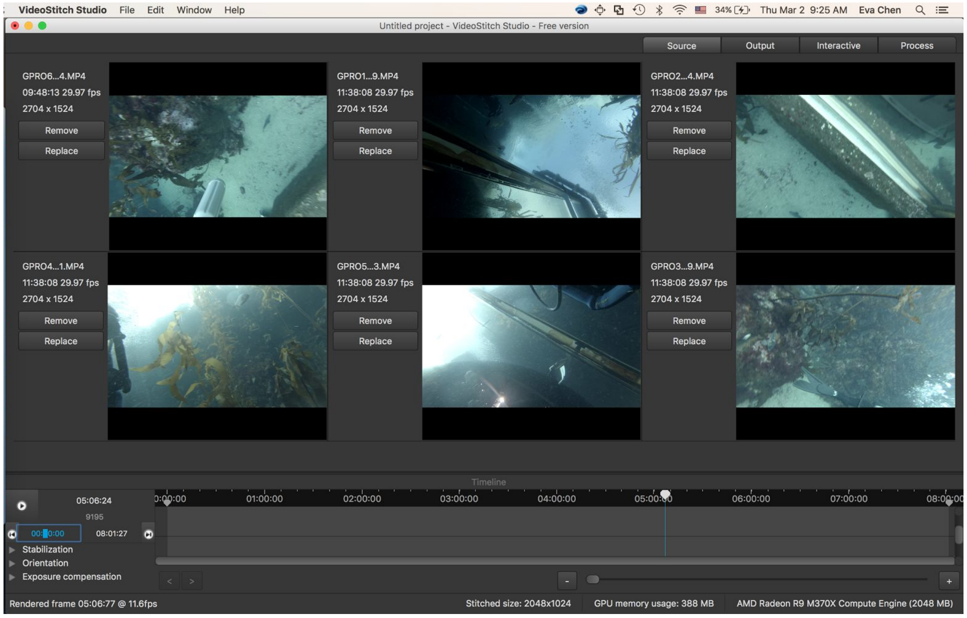
Completed GoPro Mount

- Post Processing
  - For processing the footage taken using the GoPro Rig, 3 different software programs were used.
    1. Adobe Premiere Pro: This program was used to obtain the frame offsets of each video clip needed to sync all of the footage together. The frame offsets were obtained by syncing the audio from the videos.



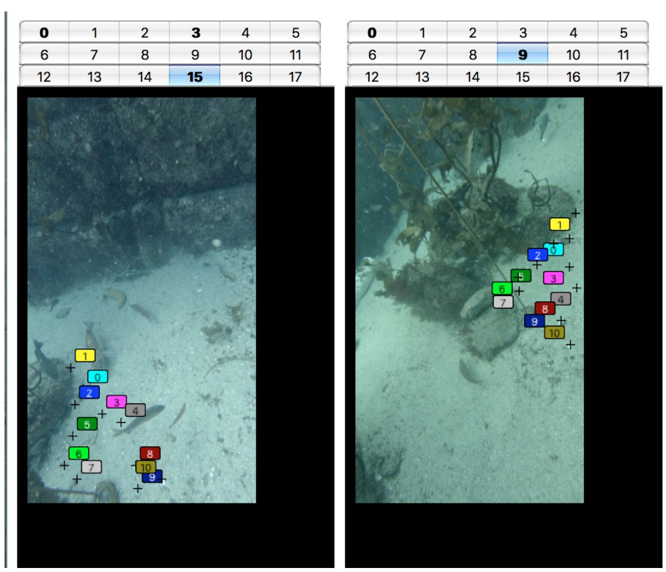
Frame Offsets Obtained by Syncing Audio

2. VideoStitch Studio: This program was used to stitch the video clips together into a 360 degree video. It has the capabilities of stitching the videos together on its own, but in the case that it couldn't stitch them together, another program could be used to create a calibration template.



VideoStitch Studio

3. PTGui: This program was used to create a calibration template that VideoStitch Studio could use to stitch videos together. This was done by telling the program where the images overlapped using control points. These control points identified identical points across a series of images.



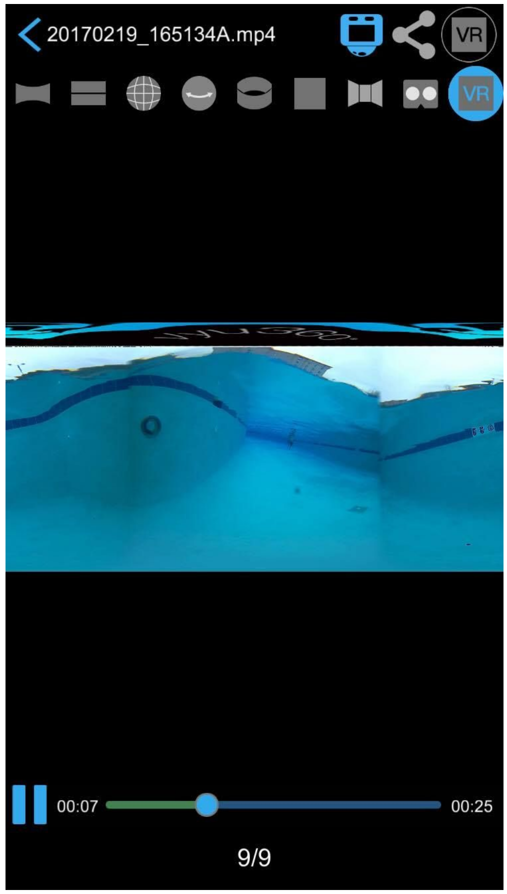
Setting Control Points in PTGui

Vyu360

The Vyu360 is the camera of choice in collecting footage since it arrived in the mail. It has a lot of benefits such as a smaller size, fewer buttons, less setup time, and significantly shorter post processing time. There are a few issues such as stitch lines and resolution, but it meets the minimum system requirements for our project.

Vyu360 Camera

- Post Processing
    - All the post processing done for the footage obtained using this camera is accomplished using the applications that the Vyu360 producers provide. Specifically we have been using their iOS application and their Mac Desktop Application.



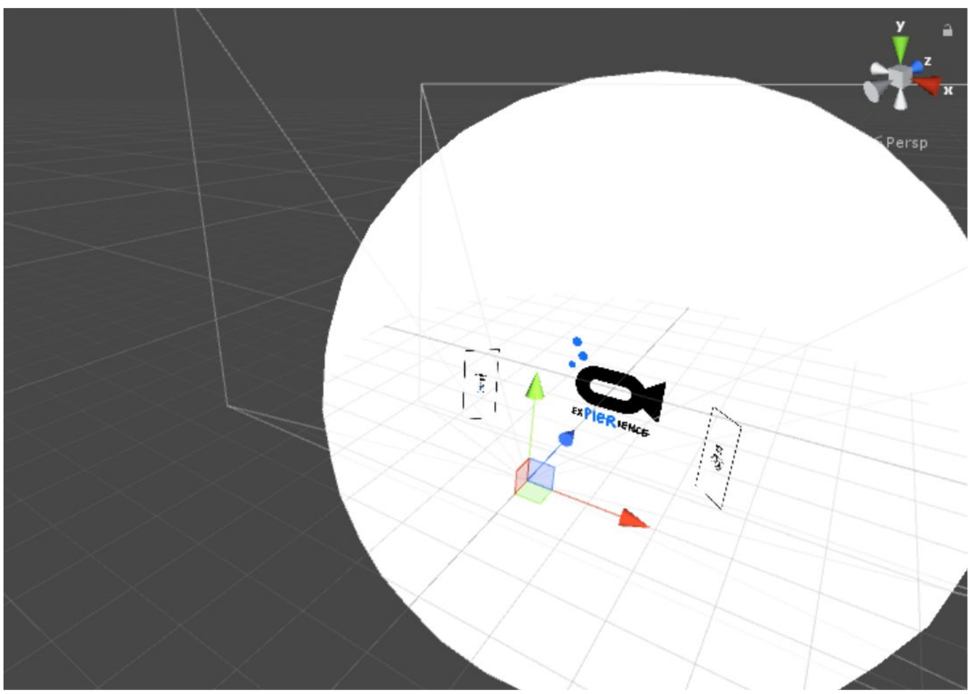
Vyu360 iOS Application

VR Application - The Multiple Unity Scenes

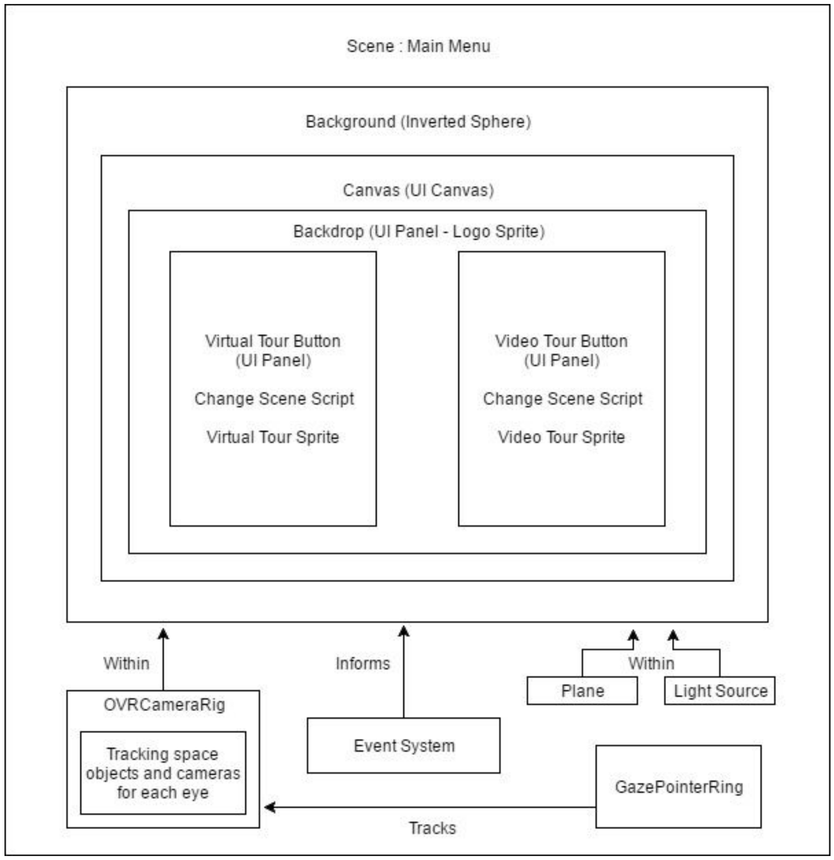
Main Menu Scene

This is the user's introduction to the application, and bridges between the interactive and immersive modes. The user has a blue ring in the center of their vision that moves as they move their head so that it remains static in their field of view. When the ring hovers over one of the options on either side of the logo, the option lights up with a blue color, and they can select to load the corresponding mode by pressing 'A' on the Xbox controller. Although visually simple, this scene has a number of complexities that make it a tractable VR menu.

Traditional UI elements are simply overlays on the screen, and require little effort. Since there is no conceptual screen with VR development, all UI elements must be put in the same world space as the user. In addition, they must have their traditional event systems that detect rays from a mouse pointer converted to use the user's gaze as a cast ray. Once this framework is put into place, the actual script that acts on the user pressing 'A' is fairly simple, initiating a scene change to each option's respective scene. Besides the menu interaction, the world in which the menu resides must also seem coherent to the user. We have gone with a cleaner aesthetic and chosen to make the surrounding area appear as white plane, to compliment our logo. To do this, we encapsulated the entire menu and user within a white sphere, inverting its normals with the same technique as the video player.



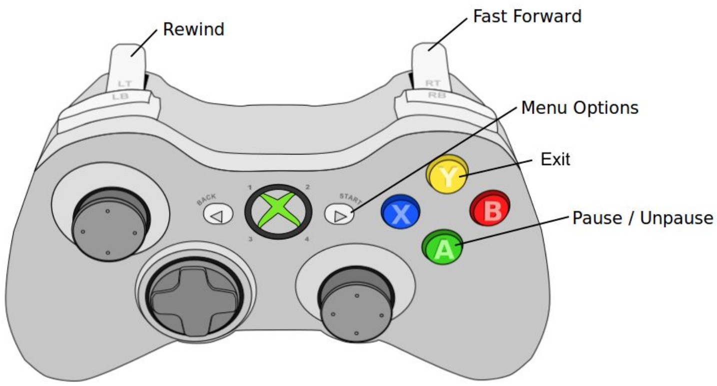
The Main Menu Scene from a Distance



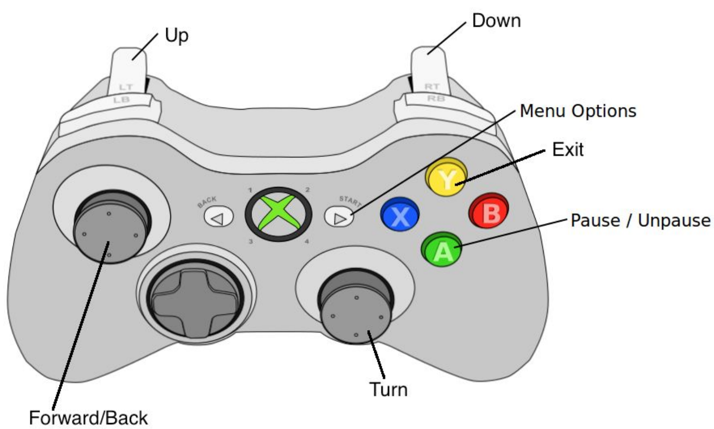
The Object Hierarchy of the Menu Scene

<u>Controller Scene</u>

Before entering each scene, there will be a scene that has a diagram of the Xbox Controller with labels as to which buttons will perform what action. Understanding how to use the controls is especially important in the Interactive Pier Scene, since navigating through the scene is done with the controller. It’s also important to note that the controls include an option to exit the scene and return to the main menu.



Controls Diagram for the Immersive Video Scene



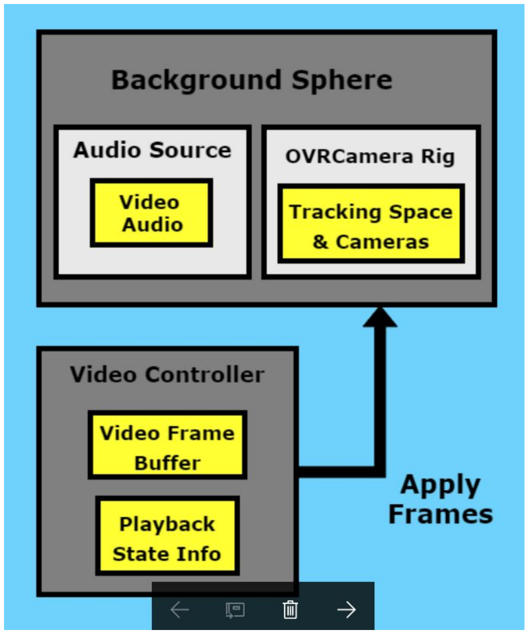
Controls Diagram for the Interactive Pier Scene

<u>VR Video Player Scene</u>

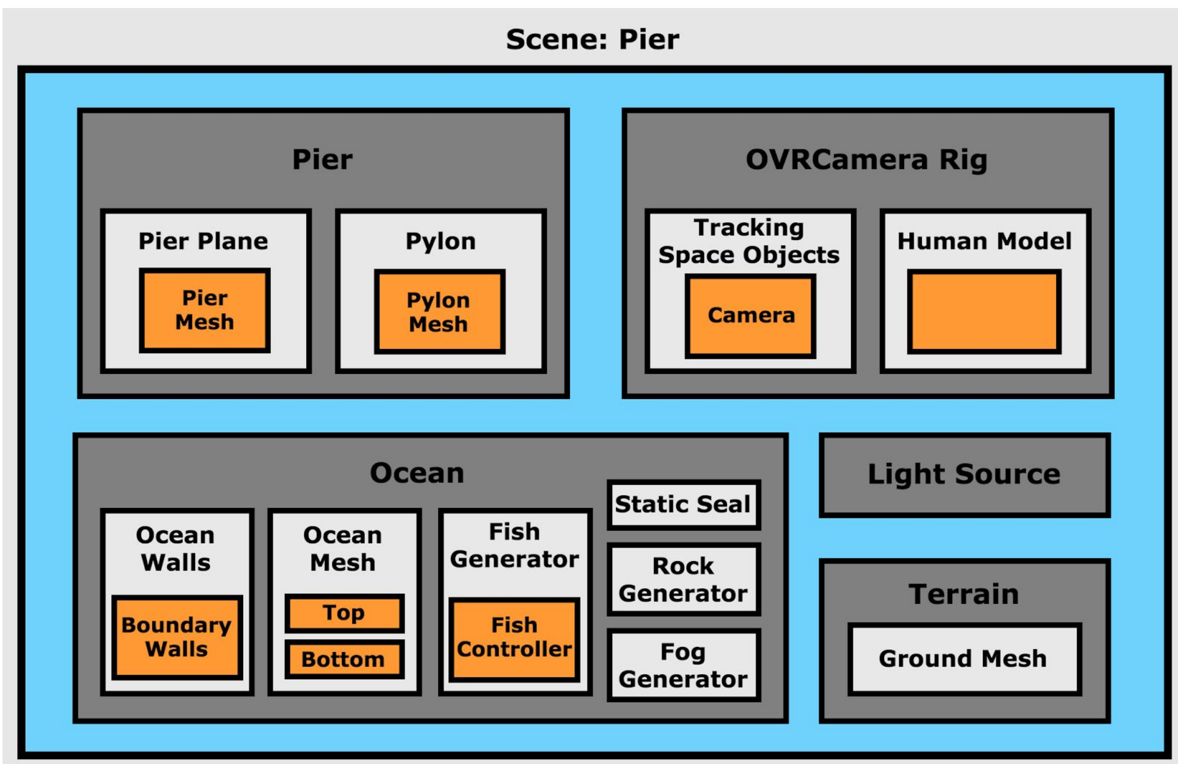
The immersive mode is achieved in a single Unity scene, “VRPlayer”. This scene is essentially a sphere with a viewable interior that the user observes from the inside. Frames from a spherical-format video are set as textures to this sphere, and sequenced at the framerate of the original capture. The result is that the viewer sees the scene from the viewpoint of the camera rig during the original video capture.

The video playback mechanism to project the frames onto this sphere surrounding the user and coordinate the audio involves several working components. First, the sphere must be modified to make its inside viewable. Normally, the Unity engine renders only the outside of objects in order to reduce computational time. Our approach for this is to modify the original shader of the sphere in order to turn all of the normals of each polygon of the sphere's mesh inward, effectively identifying the interior as the exterior, at least from Unity's point of view. An additional step is also to reverse the display of the sphere's texture because normally viewing it from the inside makes it appear reversed, as in a mirror.

We must then break up the original video into individual, accessible frames in an assets directory. A script in the scene then buffers these frames into an array. At scene updates, the array is indexed into based on the original frame rate and the time the video has been in a playing state. The fetched frame is then set to the texture of the sphere. The result is a seamless playback of the video, projected onto a sphere. It's important to note that the audio component must also be extracted and added as an AudioSource object, which can then be manipulated by the same script that sets the frames to the sphere. This script is responsible for synchronizing the video and the audio throughout pauses, rewinds, and fast-forwards.



The Object Hierarchy of the VR Video Player Scene



Object Diagram of the Pier scene

The interactive mode scene is called the “pier” scene and is composed of 5 major components. They are: the pier, ocean, OVRCamera rig, main light source, and the terrain. The grey boxes describe the components that each fulfill functionality at the highest level. Each component may contain one or more objects seen as the white boxes that add depth, such as pier platforms pylons, object generators, and ocean walls. The orange boxes represent specific tasks or actions within its role, which may describe adding material, textures, or providing access to controls, animations, or any other details.

The ocean has many sub components that give life and interactivity for the user. There will be the underwater effect with the fog particle systems that greatly limits visibility under the pier to give the user the appropriate atmosphere as a diver. The ocean will also have game objects such as rocks and fishes that populate the whole ocean. Fish are generated in a random position each time the program is started, they each have a goal position that they swim towards and a randomly generated speed. After they reach the position they receive a new goal position and a new speed. An animation is added to each fish that simulates swimming motion. If a fish collides with an object (another fish, floor, pylon, rock, etc) the fish will turn and another goal position and speed will be given to the fish. Rocks are generated with a fixed amount on the floor of the ocean. Each time the program is started the rocks are in a random position with a random size and random rotation. The ocean will also have kelps and other plants that divers would typically see under the pier.

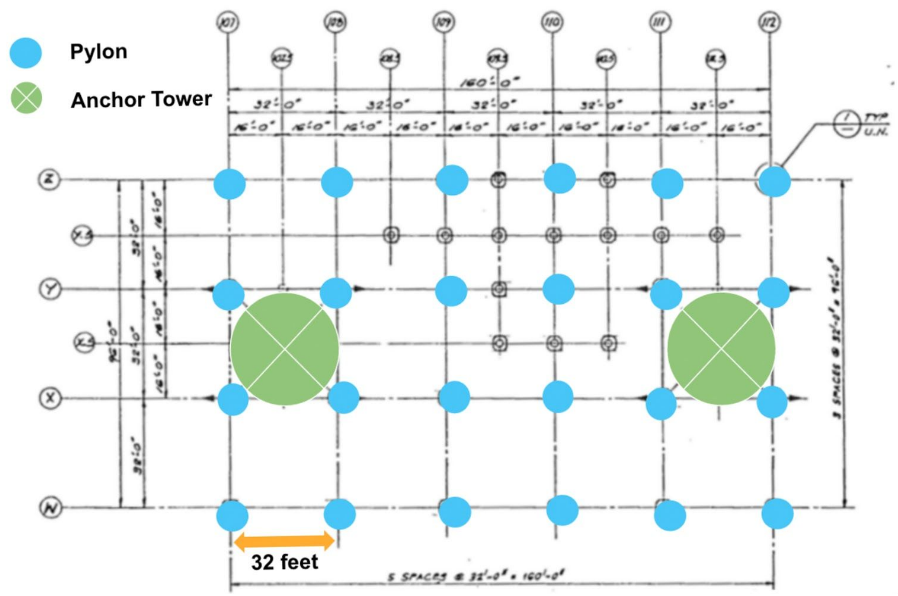
The OVRCamera rig is the swimmer that the user will be controlling in the scene. The user will have the ability to be idle, walk and jump on land. In the ocean, the user will be able to ascend and descent at the touch of a button with realistic buoyancy and gravity features. In theory, the user should be able to move around the ocean but in such a way that is convincing to the user that he or she is swimming under the pier. The actual 3D human model is an asset created in a different program called Blender. This model was shaped and rigged so that each joint can move like a real person can. Although the user won't be able to fully control the model down to every bone, it will give the user the idea that the human model is idle, walking, or swimming depending on its state.
The light source is a representation of a sun that gives light, and therefore create shadows for other objects. This allows user to not only see each object but also provide realistic depiction of the pier and under the pier.

The terrain is a partial representation of the earth so that every object has a reference during instantiation of the program. It gives a natural collider so that things do not fall through while being affected by gravity. This way, the user has a foundation for walking around or swimming around the pier.

Each object in the scene has a collider that prevents objects and the swimmer to pass each other. This is important since the scene is supposed to represent a realistic world that a diver is supposed to experience while swimming under the pier. If objects pass through each other while wearing the Oculus, the user may experience nausea and headaches. As of now, our program has numerous colliders so that user does not feel any sicknesses.

The pier represents the actual Cal Poly Pier with metal floors and 24 pylons added as supports. It will also have two anchor towers and boundaries for the user to know where to dive into the ocean. Each pylon is 32 feet away from the next in 6x4 configuration. The height of the pylon from the water top to the bottom of the pier is 40 feet. The rest of the pylon from the water top to the top of the pier is 25 feet. Each pylon will have metal textures with rust meshes that make it look like the actual pylon. In the future, the pylon will have species living and crawling on it for users to see and interact with it. The species will give the user an option to play the video playback that refers to the actual animal in real life.

The pier was constructed by analyzing the blueprint of the Pier, which can be seen below, provided by the Center for Coastal and Marine Sciences. The blueprint details the distances between the pylons, anchor tours, as well as the size of the pier. The pylons should be 32 feet away from one another, aligned into 6 columns with 4 pylons each, giving a total of 24 pylons beneath the pier. The sizing of the pylons were noted in a lengthy document with blueprints and additional details of the pier provided by CCMS as well.



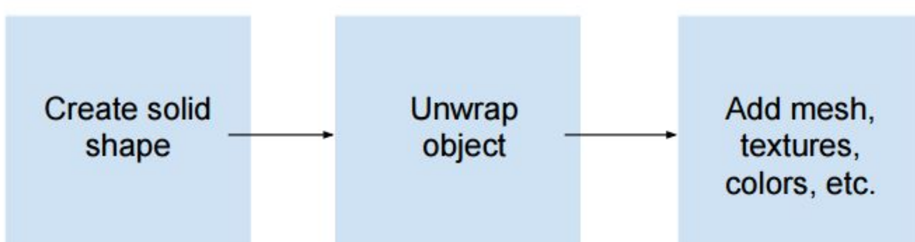
Blueprint of the Pier

The objects seen in the scene such as the pier, pylons, and rocks, were all created by 3D sculpting and modeling using the the Blender application.

The fish is from the free asset store in Unity. It acts as a placeholder fish until more research is done about what types of fish live in the ocean under the pier. But having these fish allows work to be done so that in the future it would be easy to add other types of fish. The movement of the fish was generated by a script written in C# and the swimming animation was created in Unity. There are two scripts that controls how the fish behaves one is to generate the fishes randomly and the other for movement.

The seal is a 3D modeled object that was created in the Blender application. It was colored with shades of grey and other textures but the application would not save the changes of the colors. Additionally, no movement of the seal has been created yet. Next steps for pushing this project forward will require implementation of a few scripts to add movement to the animal.

There are two levels of boundaries in the "pier" scene, and they are distinguished as ocean walls and boundary walls. The inner wall is the boundary wall which serves to restrict the users from swimming further into the ocean. This wall is needed so that the world space is confined into one box and restrict the user from running into the "void." If a user is able to go into the void, then the game will eventually crash, rendering the program as faulty. The outer wall is the ocean wall. Its purpose is to convince the user under the ocean that the ocean is endless. If it was any ordinary wall, then user can tell that they are actually confined into a box rather than an actual ocean. Its second purpose is to reflect light back into the ocean. This is important since the Unity Engine relies on lights and shades for the fog effect. For this scene, we rely on the fog effect for the underwater effect. Without the ocean wall, the fog effect won't be as effective, thus not giving the user the idea that they are underwater.



Process of 3D modeling

Process of adding objects into Unity

<u>Diver</u>

The diver is a human model created in the Blender, a 3D modeling program. The reason for the diver is to give the user the idea that they are actual divers in the ocean. When the user is swimming, the goal is to convince the reader that they are in fact swimming with hands and feet swimming in it. The model was created from a box model, and then sculpted down to different parts.

After the basic human form was created, joints were made for bone movement. The joints allowed us to "rigging" the system, meaning that the developer could manually bend the joints and save those profiles for animation in the Unity program. Essentially, it allowed the human model to move in real-time.

SYSTEM INTEGRATION & TESTING

Overview of FMEA

Functionalities that were testing during this analysis were broken down into three categories: physical interaction while wearing the Oculus Rift, interaction with the virtual environment, testing the GoPro system underwater.

Please access the table through this (public) link, as it is too large to fit well into this report.

https://docs.google.com/spreadsheets/d/1DRX-2IgbGGYUDGeUwiYD_ZUy5WxEC8GX3_PCM0nzSxo/edit?usp=sharing

DVP+R Table

Please access the table through this (public) link, as it is too large to fit well into this report.

https://goo.gl/aFVMjW

Sample Test Procedures

Buoyancy Mechanics Test

Determine: Whether the mechanics for applying positive and negative buoyancy operate only within the underwater portion of the pier scene, and only in a realistic (explained below) manner.

Materials: Project computer, Oculus Rift Headset and Sensor, Xbox controller

Safety: Operate the Oculus while sitting down in this test.

Procedure:

1. Start the “pier” Unity scene.
2. Ensure the user is in the underwater portion of the scene (relocate to it if necessary). The buoyancy mechanics are only affected by the ‘y’ axis transform of the user, so ‘x’ and ‘z’

positions should not be a concern, only whether the user is above or below the ‘y coordinate of the waterline.

3. Check the buoyancy indicator and make sure that it has defaulted to negatively buoyant without any application of the buoyancy controls.
4. Apply the negative buoyancy control. Make sure that it maxes out to -1 on the indicator value, and does not cause any clipping past the ocean floor. Also check that it applies a progressively stronger gravitational force downward as the indicator value increases negatively. If it does not max out, the value does not affect the strength of gravity, or there is clipping with the floor, this run fails.
5. Next apply the positive buoyancy control. Make sure that the buoyancy indicator maxes out at 1, and that whenever the user crosses waterline, normal (negative) gravity takes over and drops the user back into the water. Also ensure that the strength of the positive buoyancy increases with the indicator value. If normal gravity does not take over above the waterline, strength does not increase with the indicator value, or the value does not max out, this run fails.
6. This run passes if it reaches this point. Repeat 3-4 times at different x-z locations to ensure that the horizontal positions do that affect the mechanics.

Button Debounce Test

Determine: If "debounced" presses are only registered once, even if they are held down.

Materials: Project computer, Oculus Rift Headset and Sensor, Xbox controller

Safety: Operate the Oculus while sitting down in this test.

Procedure:

1. Insert a debug message into the script that prints every time the script registers a debounced button press.
2. Start the VRPlayer scene.
3. Select a button that has the debounce script on it.
4. Press the button, and keep it pressed. Make sure that it only prints one debug message for this press. If it prints more (or none), this run fails.
5. Rapidly (as quickly as a person reliably can) press the button 4-5 times (know how many). Make sure that there are exactly as many debug prints as there were presses. If there's not, this run fails.
6. Repeat for as many buttons as have this script.

Overall System Analysis

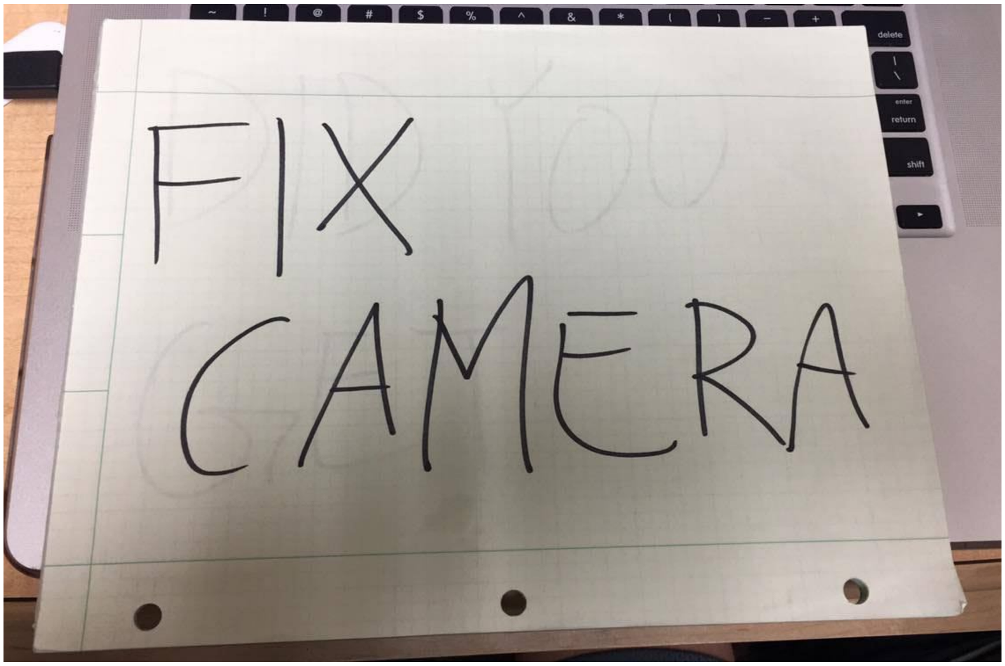
Camera Systems

<u>GoPro Rig:</u>

The GoPro rig as it is currently is not capable of recording 360 degree video underwater. This is caused by water distortion and will be discussed further below. The GoPro rig also has a few other issues that need to be addressed including camera stability, filming workflow, and post processing.

- Underwater Distortion:
  - Cameras lose around 30 degrees of their field of view (FOV) underwater.
  - Our rig current is a cube and as a result has a coverage of the following degrees (This is under the assumption that the GoPro settings are all set to a 4:3 aspect ratio with a Wide FOV):
    - One rotation has 2 horizontal cameras and 2 vertical cameras
      - Above water this covers: 434 degrees
      - Under water this covers: 314 degrees
  - Options for a new GoPro rig include the following camera configurations:
    - 5 vertical cameras (Not usable)
      - Above water this covers: 472 degrees
      - Under water this covers: 322 degrees
    - 6 vertical cameras
      - Above water this covers: 566.4 degrees
      - Under water this covers: 386.4 degrees
    - 4 horizontal cameras

        - Above water this covers: 490.4 degrees
        - Under water this covers: 370.4 degrees
    - If the new rig has only 6 vertical cameras and 1 camera on both the top and bottom, it won't be enough to cover 360. If you consider the rotation with the top and bottom, you essentially get 2 vertical cameras and 2 horizontal cameras. This would similar to our current setup, which means this would only cover 314 degrees instead of the full 360 degrees while underwater.
    - As a result, if a new rig is created, it would need a total of 10 GoPro cameras to get a full 360 view underwater. There would be 6 vertical cameras for the sides and 2 cameras for both the top and bottom of the rig.
- Camera Stability:
    - Some of the joints for the cameras weren't 3D printed super well. As a result, some of them started cracking before the dive and we had to glue them together to hold the cameras to the rig. This can be fixed by potentially altering the cad design of the mount and widening the end of the joints.
    - The cameras often would slide outwards and needed to be manually adjusted. The design of the joints is made so that the camera can rotate outwards and inwards, which is useful in putting the cameras in their cases and such. However, the plastic used in our joints either weren't a perfect fit, or the plastic was just too slippery. The screws were also hard to tighten as the hole left for a screwdriver was too small and the space between the mount and slot for a nut was too small to fit even a small wrench in.



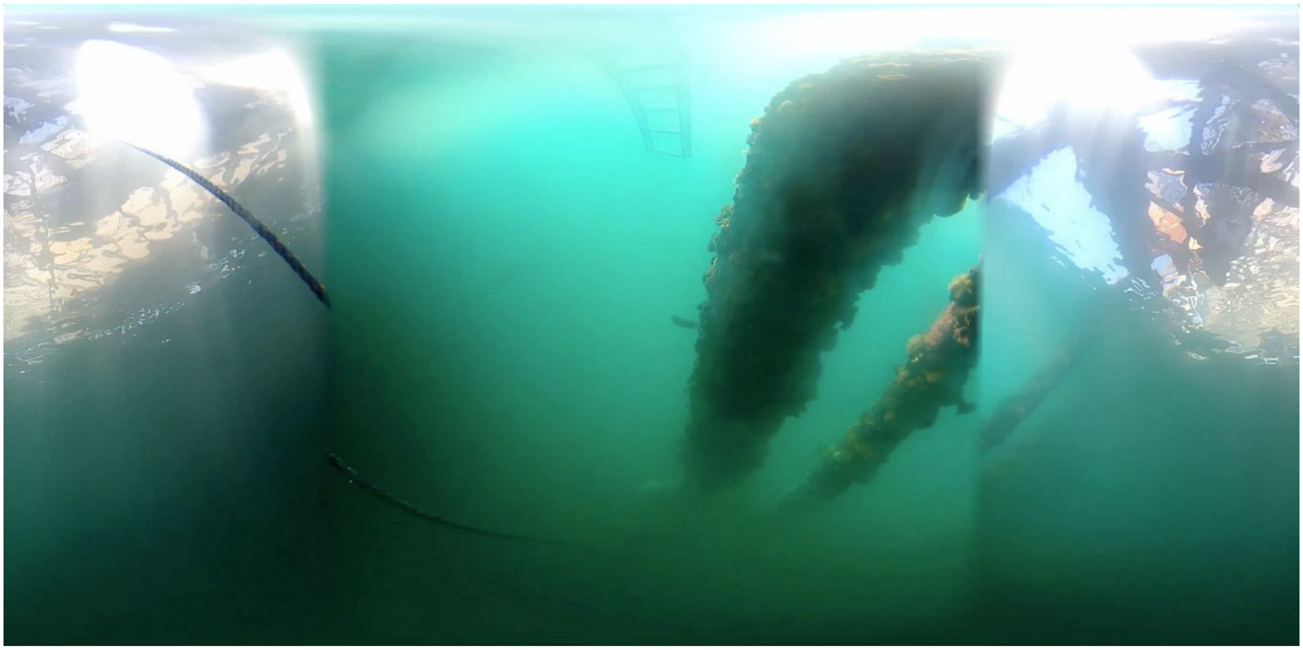
Sign Used to Tell Divers That Camera Slid Outwards During Monterey Dive

- Recording Workflow:
  - Directions for what to record and for how long can be more specific and structure. For instance, having a script written out for what needs to be done and how it should be done can be written and printed out onto waterproof paper for divers to take with them. This script could have a checklist to remind divers to make visual and audio cues.
- Post Processing:
  - The current post processing workflow hasn't been fully tested due to the lack of stitchable footage.
  - Post processing for the GoPro footage currently is extremely time consuming. This might be due to non-ideal footage, but looking into ways to reduce the amount of time this takes would be useful.

Vyu360:

The Vyu360 currently is the more optimal solution for recording 360 degree video underwater. There are a couple issues with this system such as resolution and stitch lines, but many of them cannot be addressed within the scope of this project. An option might be to contact the company that produces this camera and see what workarounds they have for some of our issues.

- Resolution:
  - 4K resolution on a GoPro and as an industry standard is defined as 3840x2160, while the 4K resolution on the Vyu360 camera is listed as 3040 × 1520. This difference in resolution is noticeable when being played back on the Oculus Rift. It also means that the 4K resolution isn't actually 4K, which is a bit disappointing.
- Stitch Lines:
  - The Vyu360 is convenient that it stitches the video output. This greatly improves the post processing workflow as in greatly reduces the amount of time needed to get a working product. Another benefit is that it for sure is capable of stitching together the videos it collects. This is in contrast to the GoPro system as if the overlap between the cameras isn't enough, then it is hard or impossible to stitch.
  - However, this convenience comes at a cost as it removes a lot of control in the stitching process. As a result, when things don't line up very well, there is no way to go in and adjust the output. This can be seen in the following screenshot where the pylons and rope are not lined up correctly.



Stitched Video Output from the Vyu360

Oculus Rift

<u>Immersive Mode:</u>

- Getting actual dive footage
    - The initial 360 GoPro Camera Rig was tested at the Monterey Bay Aquarium. Our clients held the system and swam around the kelp forest tank to try to get some footage for us to attempt to stitch together.
    - Stitching together the 6 different frames turned out to be incredibly difficult, since there were some spots that were not covered by any of the cameras. We did not take into account the fact that the underwater casings along with underwater distortion caused the scope of some of the cameras to be minimized a lot more than we expected.

- The footage that is currently integrated into the program was taken with the Vyu360 at the actual Cal Poly Pier. It is not captured by actual divers--instead, we held the camera beneath the water.
- Our next steps are to be able to capture footage of the pier through the perspective of an actual diver. Hopefully the weather conditions will be optimal.

- Exiting out of the video:
  - Our system is now able to exit to the main menu screen/scene via a button on the controller.

<u>Interactive Mode:</u>

- Sizing:
  - The virtual design of the pier and the pylon are now fixed and adjusted to scale.
  - The ocean walls and invisible walls are now adjusted to match the width and length of the rocks and fish's movement field in the scene.
- Return to Main Menu:
  - Our system is now able to exit to the main menu screen/scene via a button on the controller.
- Control Diagrams:
  - Our system now displays a button controller guide scene before the actual interactive mode to help the user become familiar with the peripherals.
  - All the necessary buttons to move, jump, look around, and return to main menu are now working.

- Physics Engine:
  - Our system’s physics effects are now more stable and adjusted for comfortable use. The visibility is adjusted for real life experience whereas the speed for moving straight is now smooth and at the right speed for most users. The ascending/descending speed of the diver is now adjusted to feel smoother.
- Content Creation:
  - Our scene has two different kinds of rocks to cover the bottom of the ocean. It also has one type of fish that moves around randomly in the water. About 100 fish are always randomly generated at run-time. A static seal is now placed in the scene to give more content to the scene.

MANAGEMENT PLAN

We will be following Agile Methodology, where we will be completing large goals every two weeks. Within this time frame, there will be subtasks leading up to that main goal.

Task	Completed by
How we want diver to capture the videos/photos (i.e. degree of position) Specifications as to where exactly the diver needs to be holding the GoPro Rig to fully capture the underwater environment	End of Fall Quarter (wk 12)
Creating objects in Unity, beginning with shapes to represent objects Create a shapes to represent the actual objects we would like to develop in the virtual reality space. We will first begin with creating a cylinder to represent the pylon.	Beginning of Winter Quarter (wk 1)
Adding objects into the 3D space that has been created After creating a static 3D space, we must successfully add on objects into this space.	Beginning of Winter Quarter (wk 1)

Here, we will have successfully created a single pylon. We will develop objects such as simple shapes to represent living species on the pylon.

Dive Plan	Beginning of Winter Quarter (wk 2)

Instructions for the diver must be made as to where the diver must be positioned underneath the pier in order to capture the videos/photos needed to develop the virtual reality space and objects.

Update Pylon Objects	Winter Quarter (wk 3)

Expand from 1 pylon to 24 pylons with an actual pier mesh added to it. We are incrementally updating number of objects in the scene along with textures to give a realistic depiction of the pier

Adding Mesh to 3D Objects	Winter Quarter (wk 4)

Different textures and meshes must be added to the pylon and the pier floor to give a realistic depiction of the pier.

Create a 3D Human Model — Winter Quarter (wk 5)

Using Blender program, create a human model in 3D for the user interface. This will give the user the idea that he or she is the actual diver. The model will be "rigged" so that joints can move to give the model a realistic movement scheme.

Create a Fish Model with Flock Generator — Winter Quarter (wk 6)

Import fish assets to the Unity scene, apply them with textures, meshes, and animators to give a more real-life experience as a diver under the pier. The Flock Generator will be a parent object that controls the number of fishes in the ocean with scripts that control them.

Create 3 Different Rocks with Generator — Winter Quarter (wk 7)

Import rock assets to the Unity scene, apply textures, colliders, and meshes.

This will be added to fill the scene up.

Collision Detection and Buoyancy Testing — Winter Quarter (wk 10)

A lot more testing will undergo to prevent system failures and inconsistencies. Physical objects should not pass each other nor should the user be able to swim off into the void. Gravity will be adjusted in water so the user can mimic the actual divers at the pier.

Exiting Functionality — Spring Quarter (wk 2)

The user should be able to exit out of the scene as they please. When exiting the scene, the user should be back at the 'Main Menu' where they are given the option to choose between the two modes.

Additional Controls Diagram — Spring Quarter (wk 3)

Diagrams depicting how the Xbox controller works per scene should be presented to the user with good labels right before the user enters the scene.

Creating Seal Model — Spring Quarter (wk 5)

Design and import a 3D model of a seal.

Correcting Physics — Spring Quarter (wk 6)

The speed and physics of the user swimming through the scene should be as accurate as possible. This will reduce motion sickness and bring a more realistic touch to the system.

Resizing, Rescaling, Spacing — Spring Quarter (wk 7)

The objects in the scene should accurately depict the sizing and spacing of the objects at the actual pier.

Fixing Diver's Initial Position After First Jump — Spring Quarter (wk 7)

When the diver jumps into the water from the top of the pier, the diver should be floating just below the surface of the water.

Planning information

The optimal time to record footage underwater is any non-rainy season. As a result, the camera system for recording 360 degree videos will need to be completed during the first half (fall and winter quarters) of our project. An established area beneath the pier that this project will focus on will also need to be decided on before any footage can be taken.

.

A system that meets the minimum requirements for developing applications on the Oculus also needs to be acquired before we can start development. Until a system has been purchased and assembled, tasks that can be achieved during this time include research on underwater simulations and development for virtual reality systems.

TEAM DEVELOPMENT

Team Mission and Team Objectives

After discussing our personal desires for this capstone project, we developed the following mission statement, goals, and objectives for our team:

Mission Statement: ENABLING CREATIVITY.

Goals:

- Gain knowledge in topics that we know *virtually* nothing about.
- Build something meaningful using virtual reality as a platform.
- Expand the possibilities of virtual reality applications.

Objectives:

- To build a working application that utilizes virtual reality.
- Incorporate sensors to implement intuitive control in a virtual environment.
- Gather input from current and future virtual reality users on their wants and needs.
- Create an effective workflow within the team for completing this capstone project.

Team Members and Roles

Emily Nguyen: Project Manager, Software Architect, System Interface Design Lead,

Albert Chen: Secretary, System Interface Designer, Software Architect

Seong Chang: Liason, Software Architect, Technical Production Officer

Description of Team Roles

Project Manager

- Determines milestones and timeline for the project over the two quarters.
- Schedules meeting times for group work, client meetings, and field trips. Makes sure everyone has rides to said events.
- Ensures that project deliverables such as project goals and objectives, requirements and use cases, etc are turned in by the required deadline.
- Checks in with other group members to track progress on deliverables and milestones.
- Ensures that work done is backed up using some form of version control, etc.

Secretary

- Record what is worked on, discussed, and decided during our group meetings, client meetings, and more.
- Maintains the organization of project documents in the google drive folder.

Liaison

- Main point of contact between our group, clients, and professor.

System Interface Designer

- Decides how users are going to interact with the virtual environment by deciding how to track a user's movement (Hardware like Leap Motion, Kinect, XBox Controller, etc.).
- Given a certain movement that is detected through some form of hardware, what should be the resulting control in the virtual world.
- Making sure said controls are intuitive and comfortable.

Software Architect

- Architecting the underwater environment in a physics engine such as Unity.
- Working with the System Interface Designer so that users can move through the virtual world.

Technical Production Officer

- In charge of designing a system to record footage underwater.
- In charge of procuring and building said system.
- Creates a guideline for what footage is needed and how much of it is needed.
- Processes footage afterwards to be used for the virtual reality system.

BIBLIOGRAPHY

United States. National Park Service, National Park Service. "Channel Islands Live."National Parks Service.

U.S. Department of the Interior, 2016. Web. 2 Dec. 2016.

GoPro Inc. "GoPro VR." GoPro VR. GoPro Inc., 2016. Web. 3 Dec. 2016.

Oculus VR. "Gear VR Powered by Oculus | Oculus." Oculus. LLC, 2016. Web. 3 Dec. 2016.

California Polytechnic State University. "Cal Poly Pier." Cal Poly Pier - Center for Coastal Marine Sciences -Cal Poly, San Luis Obispo. California Polytechnic State University, 2016. Web. 11 Dec. 2016.

VideoStitch, Videostitch. "VideoStitch Studio - 360 Video Post-production Software."360 VR Video Software | VideoStitch. VideoStitch, 2016. Web. 3 Dec. 2016.

Kolor. "Kolor | Image-stitching, Virtual Tour, 360-degree Video Experts." Kolor. Kolor, 2016. Web. 11 Dec. 2016.

Krpano. "Krpano Panorama Viewer." Krpano.com. Krpano, 2016. Web. 3 Dec. 2016.

Mito. "Learn 360 Photography - 360 Panoramic Photography / 360 Virtual Tour Tutorials." Learn 360 Photography. Learn 360 Photography, 2014-2015. Web. 11 Dec. 2016.

"GoPro | World's Most Versatile Camera | HERO4 Black Edition." HERO3+ Field of View (FOV) Information. N.p., n.d. Web. 24 Mar. 2017. https://www.google.com/url?q=https://gopro.com/support/articles/

Kintner, Michael. "CES 2017 : 360SeaDak 360 Video Dive Housing for KODAK PIXPRO SP360 4K." 360RIZE - Life Without Edges. N.p., 05 Jan. 2017. Web. 24 Mar. 2017.

APPENDICES

LIST OF APPENDIX DETAILED PERSONA NARRATIVE

LIST OF APPENDIX ENGINEERING REQUIREMENTS

LIST OF APPENDIX USE CASES

LIST OF DECISION TABLES/MATRIX