

公立はこだて未来大学 2018 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2018 System Information Science Practice

Group Report

プロジェクト名

Underwater World

Project Name

Underwater World

グループ名

アプリケーション開発グループ

Group Name

Appication Development Group

プロジェクト番号/Project No.

19-A

プロジェクトリーダー/Project Leader

1016011 小橋元春 Motoharu Kohashi

グループリーダー/Group Leader

1016011 小橋元春 Motoharu Kohashi

グループメンバ/Group Member

1016003 今泉慧 Kei Imaizumi

1016011 小橋元春 Motoharu Kohashi

1016120 工藤祐弥 Yuya Kudo

1016155 酒田恭吾 Kyogo Sakata

1016170 渡邊建 Takeru Watanabe

1016218 長谷川涼乃 Ryono Hasegawa

1016223 松本魁人 Kaito Matsumoto

指導教員

長崎健 和田雅昭 高博昭

Advisor

Takeshi Nagasaki Masaaki Wada Hiroaki Taka

提出日

2019 年 1 月 16 日

Date of Submission

January 16, 2019

概要

本プロジェクトは、海に行かずとも、いつでも手軽に海を楽しむことのできるアプリケーションである Umistagram というアプリケーションの開発を行った。

近年、若者の海離れが顕著である。その原因として、海への親しみを感じていない若者が多いことが考えられた。私たちは、海に親しみを感じていない若者を対象として、新しい海の楽しみ方を提供するアプリケーションの開発を目的とした。

本プロジェクトでは 360°動画を用い、VR で海中を体験できるシステムを開発する。

前期の活動としては、主に動画撮影と、会議の 2 つの活動を行った。

動画撮影では実際に 3ヶ所の函館にある海岸へ行き、4つの撮影機材を用い海中の動画撮影を行った。

会議では、ブレインストーミングと KJ 法を用いることで、海に親しみを感じていない若者に対しどのような海の新しい楽しみ方を提供するかを検討した。

後期の活動としては、主に動画撮影と、アプリケーション開発の 2 つの活動を行った。

動画撮影では、VR アプリケーションの背景に使用するための動画撮影を行った。

アプリケーション開発では、Unity 班・画像処理班・ウェブサーバ班の 3 つに分かれて作業を行った。Unity 班では、Unity 上で球の裏側に背景画像を貼り、その中に視点を設置することで 360°画像を見られるようにし、付属のコントローラーを利用することで海の中の世界を体験できるようにした。また、リアルに表現した魚の群れや、泡や両手などといったオブジェクトの追加、視覚以外の振動や音を利用することで没入感を高めた。

次に画像処理班では、背景画像の輝度調整、撮影した画像の加工用フィルタの作成、簡単に加工できるフィルタアプリケーションの作成を行った。背景画像の輝度調性は、使用したカメラにレンズが 2 つあることで起きるレンズ毎の輝度差を、各レンズの輝度値の平均を算出し調整することで改善した。フィルタ作成では、グレイスケール化や斜線、ブルーフィルタなど、組み合わせることのできるフィルタを全部で 12 種類、python によって作成した。フィルタアプリケーションでは、GUI ボタンによって簡単に 12 種類のフィルタが適用でき、また画像の保存や加工前へのロールバック、90°回転、写真の消去などの機能を用意した。

次にウェブサーバ班では、サーバ構築とウェブサイトの作成を行った。サーバ構築は、MACmini のサーバを使い、そこに画像を自動で共有する設定をした。ウェブサイト作成は、HTML/CSS とギャラリー風ライブラリである photoswipe を用いることで、スマホ対応な画像一覧を表示させ、保存することのできる画像保存用サイトを作成した。

プロジェクトを行っての課題が 2 つある。1 つ目は、現在 2 度行っているファイル共有の際それぞれで別の方法を用いているためにそれらを統一する必要があること。2 つ目は、今回撮影した動画が波の影響で大きく揺れており、酔いの原因となるために VR アプリケーションに使用できなかったため、揺れないような撮影方法の確立が必要であること。以上の 2 点が今回プロジェクトを行っての課題である。

キーワード Umistagram, VR, アプリケーション開発, Unity, Python

(※文責: 渡邊建)

Abstract

In recently year, young people avoid the sea because they are not familiar with the sea. We aimed to propose the new method to enjoy the sea for them. In this project, we developed a VR application system using 360 degrees movies called “Umistagram” .

In the first semester, we shot underwater movies and discussions about the idea of the application. We went to coasts of Hakodate city to shoot underwater movies using four cameras. We also discussed new way to enjoy the sea for young people based on brainstorming and KJ method.

In the second semester, we shot underwater movies and developed VR application. We shot movies at Maehama beach to obtain the background image for the VR application. To develop the VR application,

In application development, we worked in three groups: Unity group, image processing group, web server group.

In the Unity group, they pasted a image on the back side of sphere on Unity and set a viewpoint in the sphere to enable viewing 360 degrees image. We also added objects such as bubbles, hands, schools of fish, sounds, and vibrations to raise a sense of immersion.

In the Image processing group, they performed the brightness adjustment of the background image, creating image processing filters, and created a filter application that can be easily processed. In the brightness adjustment of background image, the luminance tone of the background image was improved by calculating and adjusting the average of the luminance values of each lens. In created filters, we created 12 kinds of filters that can be combined, such as grayscaling, diagonal lines, and blue filters, in total using python. In the filter application, 12 types of filters can be easily applied by the GUI button. The application also has function of image saving, roll back before processing, 90 degrees rotation, and photo deletion.

The web server group constructed a server and developed web site. We used Mac mini to construct the server. The server was set up to share images between PC for processing and web site automatically. To developed web site, we used HTML, CSS, and photoswipe. Photoswipe is a gallery style library. In the web site, users can see images which were taken in the VR application and download images to their smartphones.

The developed VR applications has two improvements. First, it is necessary to unity file synchronization method between VR applications, filter GUI application and web server Second, it is necessary to establish movie shooting method to prevent shake due waves because the shake causes VR motion sickness.

Keyword Umistagram, VR, Application Development, Unity, Python

(※文責: 渡邊建)

目次

第 1 章	はじめに	1
1.1	背景	1
1.2	目的	2
1.3	従来例	3
1.4	従来の問題点	4
1.5	課題	4
第 2 章	プロジェクトの概要	6
2.1	問題の設定	6
2.2	課題の設定	6
2.2.1	撮影	6
2.2.2	アプリケーション開発	6
2.3	目標	7
2.4	課題の割り当て	7
2.4.1	前期	7
2.4.2	後期	8
第 3 章	課題解決のプロセスの概要	10
3.1	目標決定会議	10
3.2	撮影	10
3.3	中間発表	12
3.4	アプリケーション開発	13
3.4.1	Unity 班	13
3.4.2	画像処理班	13
3.4.3	Web サーバ班	14
3.5	最終発表	14
第 4 章	課題解決プロセスの詳細	15
4.1	会議	15
4.1.1	目標決定	15
4.1.2	動画の撮影場所	15
4.1.3	使用するカメラ	15
4.1.4	アプリケーションの仕様	16
4.2	撮影	16
4.3	中間発表	17
4.4	アプリケーション開発	17
4.4.1	Unity 班	17
4.4.2	画像処理班	22

4.4.3	Web サーバ班	28
4.5	最終発表	31
第 5 章	結果	33
5.1	成果	33
5.2	解決手順と評価	33
5.2.1	撮影	33
第 6 章	まとめ	36
	参考文献	40

第 1 章 はじめに

近年、海に親しみを感じていない若者が多い。そのため、海に親しみを感じていない若者に海の魅力を伝えるため、若者の興味を引くことが必要である。そこで、我々のプロジェクトは VR と SNS を用いて、「新しい海の楽しみ方」を提案する。全天球画像を用いることで、VR を用いて海の中に潜っているかのような体験をすることができ、海に興味を持つきっかけを作ることのできる VR アプリケーションを作成する。全天球動画とは、360°全てを見渡すことができる動画である。

(※文責: 今泉慧)

1.1 背景

我々のプロジェクトがインターネットで調査した結果、近年、海に親しみを感じていない人が多く、特に若者は海に親しみを感じていない割合が高い。2017 年に日本財団によって行われた、現在の日本人の海への意識や行動の実態を明らかにするために、全都道府県の 15 歳-69 歳までの男女を対象として行われた「海と日本」に関する意識調査 [1] を用いた。調査によると、図 1.1 に示す通り、海に親しみを感じていないと答えた 10 代の若者は全世代の中で最も高く、4 割以上おり、海に親しみを感じていると答えた割合も最も低い 27.5% だった。海に親しみを感じる割合は年齢が高い世代ほど高く、若くなるにつれ、比例的に低くなっていることが分かる。また、同調査から図 1.2 に示す通り、海水浴に行く意欲がなく行っていないと答えた 10 代は、約 55% と過半数を超えていた。図 1.1 と図 1.2 の結果より、若者は海に興味がなく、海に行くこと自体を好んでいないと考えられる。10 代が海に行く意欲がない理由として挙げているのは、割合が高い順に、「泳ぐこと自体が好きではない」、「海まで距離があり、時間がかかる/気軽に行けない」、「日焼けが嫌だ」などが挙げられている。そして、それぞれの理由が高くなった理由を考察した。

大学生 137 名に対して行われた、中野・大庭 (2004) の分析 [2] によると、水泳を嫌いな人は、その理由として、「泳げない・へたくそ」を挙げた割合が 55.6% と最も高く、泳げないから泳ぐことを嫌いになることが分かった。また、日本テレビのプログラムの一つである「zip!」によって 2003 年に行われた調査では、25 メートルを泳ぎ切ることができない人の割合が、30 代以上では約 9% だったのに対して、10 代から 20 代の約 52% が泳ぐことができないと回答した。これは、2002 年から実施された授業時間の削減によって水泳の授業が削減されたことなどが影響していると言われており、泳ぐ機会がなくなったことにより、泳げない若者が増えたことが分かる。次に、海まで距離がある、気軽に行けない等と感じる理由として、レジャーや集客施設の増加が考えられる。近年、交通が発達して海へのアクセスは良くなった。しかし、それはほかのレジャーや集客施設へ行くアクセスも良くなったことを意味し、泳げなかったり、日焼けを嫌う若者は屋内のプールや、服を着たまま遊ぶことができる遊園地や動物園、美術館などに行くことを選択しがちであると考えられる。最後に、日焼けを嫌がる理由として、紫外線による肌の老化や、がんの対策があると考えられる。国内の紫外線量は、紫外線観測を開始した 1990 年以降、長期的な増加傾向にある。さらに気象庁は 2005 年よりホームページ上に毎日の紫外線の強さの情報提供を始め、紫外線の強さを数値化した UV インデックスを参考に対策を取るよう呼び掛けている。これによって、日本でも紫外線に対する意識が高まった世代の子供にあたる 10 代から 20 代の若者は日焼けをひどく嫌うの

だと考えられる。

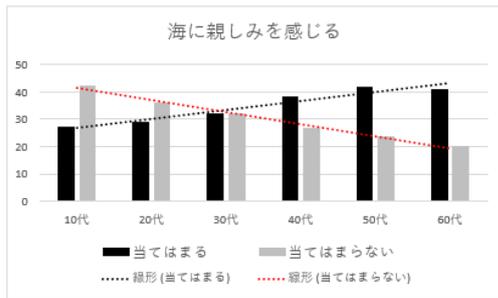


図 1.1 海に親しみを感じない人の割合 (日本財団, 「海と日本」に関する意識調査)

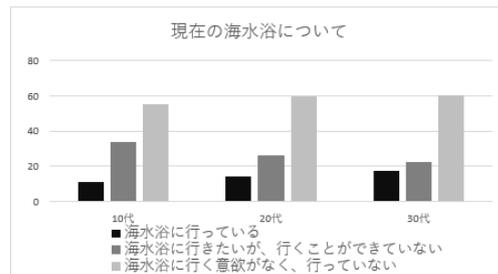


図 1.2 現在の海水浴について (日本財団, 「海と日本」に関する意識調査)

(※文責: 今泉慧)

1.2 目的

現在行われている海の楽しみ方としては、海に実際に行く海水浴や海辺の散歩やドライブ、魚を見ることができる水族館、映画館や家で実際には行くことが難しい海の映像を見ることができるドキュメンタリー映像などがある。しかし、1.1 より、現在海に興味のない若者は現在行われている上記に挙げた海の楽しみ方では、新しく興味を持ってもらうのは難しいと考えた。そこで、我々は海以外の新しい要素で引き付け、海を体験して楽しんでもらうことで海に興味を持つきっかけを作ることができると考え、海に行きたくない若者に自らやってみたくなるような「新しい海の楽しみ方」の提案をすることを目標とした。そこで、我々のプロジェクトは海に親しみを感じていない若者に対して、興味を引くことができる技術として、VR と Instagram の 2 つの要素を用いることに決定した。それぞれの要素を採用した理由を説明する。図 1.3 は、TesTee によって 10 代、20 代を対象として行われた、VR/AR に関する調査である。図 1.3 によると、VR は約 8 割の若者がやってみたくて答えているため、VR を用いることに決定した。次に、図 1.4 は、株式会社ジャストシステムによって、男女 15 歳-69 歳までを対象として行ったアンケートと、総務省統計局による人口推計から概算したデータをまとめ、ユーザー数の多い 3 つの SNS を抜き出したものである。表 1 は App Ape Report による国内の 3 万台の Android 端末を分析したであり、Instagram のアクティブ率が 84.7% と最も高かった。図 1.4 から、Twitter と Instagram の 2 つが国内において若者に流行している SNS であることが分かる。そして、表 1 から、アクティブ率が高いため多くのユーザーの目に情報が入りやすく、写真を共有することに特化している Instagram が、我々の目的である海の魅力を伝えることを、ユーザーが自発的に行うことを望むことができるため、今回採用した。この 2 つの要素によって、距離や日焼けなどの海に行きたくない理由を解決することができ、若者の興味を引くことができる海に行かない「新しい海の楽しみ方」を提案することを目的としている。

(※文責: 今泉慧)

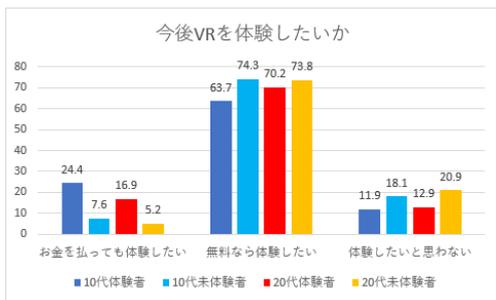


図 1.3 VR/AR に関する調査 (TesTee)

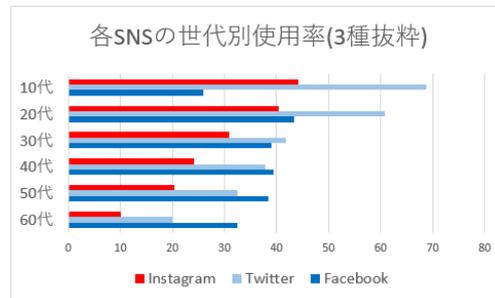


図 1.4 11 のソーシャルメディア最新動向データまとめより抜粋 (SocialMediaLab)

表 1.1 各 SNS の月間アクティブ率

SNS	月間アクティブ率
Instagram	84.7%
Twitter	70.2%
Facebook	56.1%

1.3 従来例

従来、海に行く以外で海の魅力を伝えるための海の楽しみ方として、様々なことが行われてきた。海のなかでも魚にフォーカスを当てた水族館と、自然のままの海と海に生息する生物に焦点を当てたドキュメンタリー映像の例を挙げる。まず水族館について説明する。沖縄県の美ら海水族館では、世界で初めてジンベイザメが複数匹同時に泳ぐことができる大きな水槽を作っており、迫力のある大型の魚が泳ぐ姿を見ることができ、世界初という真新しさから成功した。神奈川県の新江ノ島水族館では、夜間にガイド付きのツアーを行うことによって、普段の営業時間には見ることができない夜行性の魚や睡眠中の魚を見ることができ、次に、海のドキュメンタリー映像は、有名なものに 2010 年に公開された OCEANS や DEEPBLUE がある。OCEANS は、海の生物に焦点を当てたドキュメンタリー映像で、ロボットを使って魚と同じ目線の映像を楽しみ、自分が一緒に泳いでいるかの感覚を味わうことができる。DEEPBLUE は、世界中の海の生物を見せることに焦点を当てた動画で、総製作期間は 7 年に及び、世界 200 箇所以上の海に生息する、幅広い海の生物の姿を楽しむことができる。世界中の海に生息する生物や自然の綺麗な部分だけでなく、食物連鎖などの残酷な部分も見ることのできる自然をより感じることができる。

次に、VR では、近年 VR で用いる全天球映像を撮影することができるカメラが段々と一般的になっており、新しく集客や、業務効率の上昇、人件費や移動時間等のコストの削減などの効果を見込んで様々な業界で活用が始まっている。例えば、不動産会社で部屋の内見を実際に見に行くのではなく、VR で行うことで、従業員と客がともに移動に掛かる時間や経費などの負担を減らすことができ、その分の時間を顧客に使うことができるため、より満足度の高いサービスを行ったり、客の回転率を上げるなどことができ、顧客の増加が見込める。ほかにも、Google ストリートビュー室内版というサービスがあり、これは、レストランやホテルなど有料で導入するサービスで、室内の様子を VR で実際に見ることができ、客は実際のレストランやホテルの外見だけでなく、店の中の様子も見ることができるため認識の差が生じにくい。Google が第 3 者を雇って行っ

た調査によると、Google ストリートビューを導入したレストランは予約率が約 2 倍に上がり、特に 18-34 歳のユーザーは、Google ストリートビューを見た後に予約をする可能性が約 2.3 倍に上がっている。

(※文責: 今泉慧)

1.4 従来の問題点

海の魅力を伝えるには実際に海に関する物事に触れたり、海の映像を見て楽しんでもらうことが必要である。1.3 節で挙げた 2 つの水族館の方法では、これまでの水族館ではすることができなかったことを体験できるという新鮮さが客の興味を引くことができる。しかし、水族館はどこにもあるものではなく、海よりも若者に対して魅力的な水族館が近くにあることが少ないと考えられる。巨大な水槽は建設費や維持費がかかり、夜間ツアーは人件費が多く掛かるため、簡単には新設できない。そのため、距離の問題で、現在水族館に興味のある人を除くと、海よりも水族館よりもほかのレジャーや集客施設に行ってしまうことが考えられ、水族館に来ない人にはアプローチができない。そして、海のドキュメンタリー映像では、高性能なカメラなどの機材と、多くの時間をかけて撮影された、自分だけでは見ることのできない壮大な海の中の世界を、映画館や家のテレビなどで楽しむことができる。しかし、しかし、綺麗な風景や、動物の貴重なシーンを見ることはできるが、遠くの音はカメラで拾いきれないため、スタジオで録音した音を使用するため、「このドキュメンタリー映像はフェイクだ」などと問題になることもある。また、撮影の前後をバラバラに編集することで物語性を無理やり作ることによって、「食べられた動物がかわいそう」などの映像の作成者の狙った感想を持つ視聴者が多くなってしまうことが考えられる。

VR は、体験したことがある若者が少なく、体験者も自分が VR 機器を持っていない場合は、十分体験していることが少ないため、VR 体験者と未体験者を問わずに若者の興味を引くことができる。しかし、不動産会社や Google ストリートビューを利用する店は実際の姿を見せることを目的としているため、定点撮影のパノラマ写真を見ているのと同じである。そのため、写真を見てきれいだと感じても、感じた本人までで止まってしまい、海の魅力をより多くの若者に伝えることができない。

(※文責: 今泉慧)

1.5 課題

1.4 節から、これまで行われてきた海の楽しみ方では、海に親しみを感じる若者の増加は見込むことができない。海の魅力を伝えるには、まず海に関する物事に触れてもらわなくてはならない。しかし、そのためには我々は海に行きたくない理由を解決することのできる新しい要素を持った、「新しい海の楽しみ方」を創造する必要がある。そこで、1.3 節で挙げた海の魅力を伝えるために行われた従来例とは別の客層を狙った「新しい海の楽しみ方」を考える必要がある。別の客層というのは、現在海に行きたくないと感じている若者であり、図 1.2 より、10 代、20 代の半分以上という大きな開拓層に対してアプローチを行うことが必要である。そのためには、彼らの海に行きたくない理由を解決するため、「海で泳がない」、「すぐにできる」、「室内でできる」の 3 つの要素を兼ねた「新しい海の楽しみ方」を考える必要がある。また、VR に関しては、海の魅力を伝えるため

Underwater World

に、若者に広く使ってみてもらふ必要がある。現在の 360°を見渡すことができるということに加えて、オリジナリティのある写真を撮影し、SNSなどで世界と共有することで、似たような嗜好の若者に海の魅力を体験した若者自身が広めたいくなるような工夫が必要である。

したがって、海に行きたくない若者を対象として、海に行きたくない理由を解決することができる要素を持った「新しい海の楽しみ方」考えることと、VR アプリケーションに、海の魅力を若者が周りと共有したくなる要素を加えることの2つを解決すべき課題とした。

(※文責: 今泉慧)

第 2 章 プロジェクトの概要

2.1 問題の設定

今年度では、海への親しみや興味を失いつつある今の若者に、新しい海の楽しみ方を提案する。海に行きたくない理由を解決することができる要素を持った「新しい海の楽しみ方」を考えると、若者が回りと共有したくなる要素を加えることの2つを解決すべきとして設定した

(※文責: 松本魁人)

2.2 課題の設定

近年、海に親しみを感じていない若者が多いことがわかっている。その理由として、海が遠い、泳ぐことが好きではない、日焼けが嫌だ、などの理由が挙げられている。これらを解決するために、若者の意欲を刺激するような VR アプリケーションの開発を目指す。そして、そのアプリケーションに触れた人々が、VR 技術を用いて海の中という簡単には体験できないものを映像で見て触れて、海に対して興味を持ってもらう。前期では後期でアプリケーションを作成するための映像を撮影するために、全員がカメラを扱えるようにすることを課題と設定した。後期では、アプリケーション開発に取り組むために、最低限の知識を身に着けることを課題とし、インターネットや情報ライブラリーを利用するようにした。また、プロジェクト内の有識者とコミュニケーションをとり、時間をかけて勉強して、協力して開発に取り組む。

2.2.1 撮影

足場の不安定さや海の透明度などを考慮し、撮影しやすい場所を見つける。その場所が VR の動画で使える撮影場所なのか考える。撮影後、海の魅力を伝えるには不十分だったものは何かを話し合う反省会を行う。その後、次の撮影場所はどこがよいか案を出し合い、これらを数度繰り返す。そして、全員がすべてのカメラの最低限の操作ができるように慣れておく。撮影場所が決まれば、海の中に入れる期間は長くないので、早急に VR の動画に使う動画を撮影する。その後、画像処理班による動画の調整を行う。

2.2.2 アプリケーション開発

Unity 班

Unity 班の課題は、作ることに決めたアプリケーションを Unity で開発することである。海中を体験するための VR 空間や、その VR 空間への没入感を高めるための構成要素の作製をする。Unity 班のメンバーには、Unity に関する知識が無い人もいるので、情報ライブラリーやインターネットサイトを利用して技術の習得を行う。

VR 空間では、スマートフォンに実際に写真を撮影できる機能や、魚を掴むことができる機能を付ける。

画像処理班

画像処理班の課題は、2つあるカメラで撮影すると発生してしまう輝度差を調整することである。そして、Unity 班が作製した VR 空間で撮影した写真を加工するためのフィルタの作成、またその作成したアプリケーションを開発する。フィルタとアプリケーションは Python を使って作成する。

Web サーバ班

課題：VR アプリケーション内で撮影した写真を画像処理班が作製したフィルタアプリケーション用のパソコンにデータを転送し、フィルタアプリケーション用のパソコンとサーバの写真ファイルを Drop Box によって同期できるようにすることである。

また、フィルタアプリケーション用のパソコンで加工した写真を閲覧することができる Web サイトを作成することも Web サーバ班の課題となる。

(※文責: 松本魁人)

2.3 目標

本プロジェクトでは 2.2 節で設定した課題を解決すれば、アプリケーション開発を始めることが出来ると考えた。

海の中の様子を 360°見渡すことのできるカメラで撮影し、その全天球画像を VR によって海中を体験できるようなシステムの開発が、最終目標である。また、目標達成のために、以下の要素が実装されていることとした。

- VR アプリケーション内で魚の写真を撮ることが出来ること。
- 撮影した写真をフィルタアプリケーションで加工が出来ること。
- 加工後の写真を Web ページで見ることが出来ること。

(※文責: 松本魁人)

2.4 課題の割り当て

グループメンバーの得意分野などを基準に以下のように割り当てた。カメラの詳細な説明は 3.2 章で取り上げている。

2.4.1 前期

今泉

作業量の多い小橋のサポートを担当した。撮影では VR360 というカメラを扱い、撮影のために必要な装備の買い出しも行った。

また、中間発表用のスライド作成を担当した。

工藤

プロジェクトメンバー唯一の函館出身として、最初に撮影する場所決めを担当した。その後の撮影場所の決定も担当した。

小橋

去年作成されたゴルダックを扱える状態にするなど、VR360 の使用を担当した。撮影した動画の処理を行った。

また、グループリーダーを担当し、担当教員との連絡など、グループ全体のまとめや進行役を担った。

酒田

主に PowerRay という水中ドローンと GoPro というカメラを扱った。中間発表用のポスター作成も担当した。

長谷川

インターネットなどを利用して、撮影場所を探す役割を担った。PowerRay という水中ドローンと GoPro というカメラを扱った。

また、撮影のために必要な装備の買い出しも担当した。

松本

撮影では VR360 というカメラの使用を担当した。中間発表のスライドの作成に関わり、ポスターの英訳を担当した。

渡邊

PowerRay という水中ドローンと GoPro というカメラを扱った。中間発表のポスターの作成方法から勉強して、大部分を担った。

(※文責: 松本)

2.4.2 後期

今泉

画像処理班の中の輝度差調整を担当した。課題は2つあるカメラで撮影すると発生してしまう輝度差を調整することである。

また、最終発表用のスライド作成の大部分を担当した。

工藤

Unity 班の中の構成要素を作製する担当である。課題は没入感を高めるための泡や写真を撮影するためのスマートフォンを作製することである。

小橋

VR 空間やアプリケーション開発の大部分を担当した。海中を体験するための VR 空間を作成することが主な担当になる。

酒田

画像処理班の中の画像加工用フィルタの作成と、フィルタアプリケーションの作製を担当する。長谷川とともに Python を使って作成した。その後、最終発表用ポスターの作成を担当した。

長谷川

画像処理班の中の画像加工用フィルタの作成と、フィルタアプリケーションの作製を担当する。酒井とともに Python を使って作成した。最終発表でのポスターの作成も担当した。

松本

Unity 班の中の構成要素を作製する担当である。課題は主に、VR 空間を泳ぐ魚群の設定をすることである。知識がない状態で取り組むので、情報ライブラリーやインターネットサイトを利用して技術の習得を行う。スライド作成にも関わった。

渡邊

Web サーバ班を担当する。VR アプリケーション内で撮影した写真をフィルタアプリケーション用のパソコンにデータを転送できるようにし、フィルタアプリケーション用のパソコンとサーバの写真ファイルを Dropbox によって同期できるようにすることが課題である。

また、フィルタアプリケーション用のパソコンで加工した写真を閲覧することができる Web サイトの作成も担当する。

(※文責: 松本)

第 3 章 課題解決のプロセスの概要

3.1 目標決定会議

このプロジェクトは海に親しみのない若者に向けて海を楽しむ新しい方法を提案するという目的のプロジェクトである。まず初めに目的を実現するために必要な目標を考えるとところから始めた。実際に自分たちで撮影した動画を参考に、最終成果物のアイデアをブレインストーミングで自由に出し合った。そのアイデアを KJ 法で分類し、アイデアを絞った。その結果、最新技術であり、若者の興味を引くことができると考えられる VR を使用した、海中体験アプリケーションの開発に決定した。ただし VR の動画を作るだけでは既存のものを使っているだけという指摘が出たので、追加要素としてインスタ映えできるような写真を撮影することができる海中体験アプリケーションという目標に決定した。

(※文責: 酒田恭吾)

3.2 撮影

撮影機材として、担当教員に GoPro HERO6 Black(以下 GoPro とする)、PIXPRO SP360 4K(以下 SP360 とする) という 2 つのカメラ、PowerRay という水中ドローンを借りた。また補助機材として、GoPro を取り付けることができ、最大 7 m まで伸びる棒 BiRod、SP360 を沈める時に必要であり、去年のプロジェクトで制作したゴルダックを借りた。それぞれの機材を図 3.2、図 3.3、図 3.4、図 3.5、図 3.6 に示す。GoPro は、カメラ自体が防水のため高画質な 4K 動画の撮影が可能であり、とても軽量なため初心者でも取り扱いやすいという特徴がある。SP360 は、広画角レンズがついており半球動画の撮影位が可能であるという特徴がある。PowerRay は、スマートフォンと接続してリアルタイムの映像を確認することができ、水深 30 メートルまで潜水可能なので、人の行くことのできない所の撮影が可能という特徴がある。ゴルダックへのカメラの取り付け方法は、図 3.8 のように結束バンドを使用した。

この機材でどのような動画を撮影することができるかを確認したかったため、実際に海に行く必要があるということがわかった。そこで函館近郊の海で水質が良い場所の調査を行った。その結果、工藤の推薦で講義時間内に行き帰ることができる入船町前浜海水浴場に決定した。ここでは、SP360 と GoPro で撮影を行った。Power Ray は浅瀬のため使用できなかった。SP360 はゴルダックに取り付け鎖で沈め、GoPro は BiRod に取り付けて撮影を行った。撮影は小橋、今泉、松本の SP360 班と、工藤、酒田、長谷川、渡邊の Go Pro 班に分かれて行った。

2 回目は函館市国際水産・海洋総合研究センターで撮影を行った。ここでは SP360 と PowerRay を使用した。SP360 はゴルダックに取り付け鎖で沈め、PowerRay はスマートフォンに接続して撮影を行った。撮影は小橋、今泉、松本の SP360 班と工藤、酒田、長谷川、渡邊の PowerRay 班に分かれて行った。2 回目撮影後に目標が決定したので、PIXPRO 4KVR360(以下 VR360 とする) を購入した。図 3.7 である。VR360 は、前後に 1 つずつ広画角レンズがついているので、全天球動画の撮影が可能という特徴がある。VR360 を購入した理由は、目標である VR を使用した海中体験アプリケーションの制作において、VR の動画を撮影するには全天球カメラが最も適して

Underwater World

いるからである。それに加え、海に入って撮影する必要があると考え、胴長を購入した。

3 回目は黒岩岬で撮影を行った。ここを選んだ理由は水質が良いからである。ここでは VR360 と GoPro を使用した。波が激しかったため PowerRay は使用できなかった。また SP360 は VR360 の下位機種であるため使用しなかった。VR360 はゴルダックに取り付け鎖で沈め、GoPro は、BiRod に取り付けて撮影を行った。撮影は小橋、今泉、松本の VR360 班と工藤、酒田、長谷川、渡邊の GoPro 班に分かれて行った。

4 回目は入船町前浜海水浴場で実際に VR アプリケーションに使用する動画の撮影を行った。プロジェクト時間外だったため小橋、今泉の 2 人が担当した。使用した機材は VR360 である。

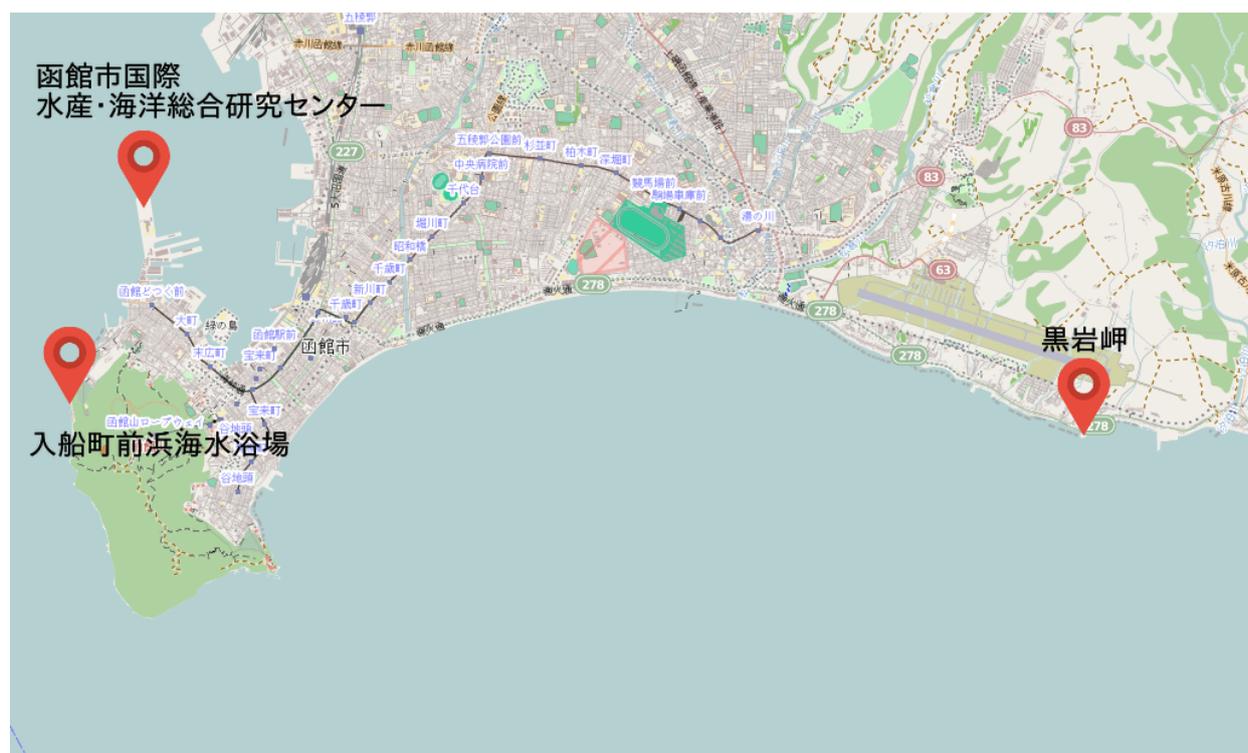


図 3.1 撮影場所の地図 (© OpenStreetMap contributors)

表 3.1 カメラのスペック

名前	GoPro HERO6 Black	PIXPRO SP360 4K	PowerRay	PIXPRO 4KVR360
画質	3840x2160 60/30/24fps	2880x2880 30fps	3840x2160 25fps	3840x1920 24fps
特徴	高画質・軽量	半天球動画撮影可能	水深 30 メートルまで潜水可能な水中ドローン	全天球動画撮影可能
撮影方法	Bi Rod に取り付けて撮影	ゴルダックに取り付けて撮影	スマートフォンに接続して撮影	ゴルダックに取り付けて撮影

(※文責: 酒田恭吾)



図 3.2 GoPro HERO6 Black



図 3.3 PIXPRO SP360 4K



図 3.4 Power Ray



図 3.5 BiRod



図 3.6 ゴルダック



図 3.7 PIXPRO 4KVR360

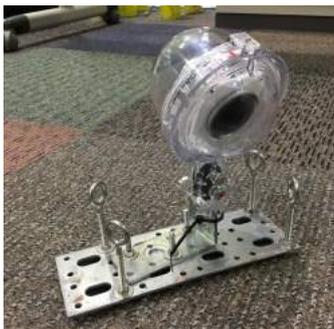


図 3.8 ゴルダックへのカメラの取り付け

3.3 中間発表

中間発表では、ポスター作成を小橋、酒田、松本、渡辺が担当し、スライド作成を今泉、工藤、小橋、長谷川が担当し、原稿作成を今泉、小橋が担当し、動画編集を小橋が担当して、松本以外の6人が15分ずつプレゼンテーションをした。

3.4 アプリケーション開発

後期からは、Unity を用いて VR 空間を開発する Unity 班、Python を用いて VR 空間内の背景画像を見やすくすること、VR 空間内の仮想スマートフォンで撮影した写真の見栄えを良くするフィルタの開発をする画像処理班、フィルタをかけた画像をスマートフォンで保存するための Web サイトを作成する Web サーバー班の 3 つの班に分かれて活動をした。最終的に私たちのプロジェクトは Umistagram というアプリケーションを開発した。このアプリケーションは、まず Unity 班が開発した VR 空間で海中を体験してもらいながら、仮想スマートフォンで撮影してもらい、その写真を One Drive を使用して PC と共有する。そして画像処理班が開発したフィルタアプリケーションを使用して PC 上でフィルタをかけ、保存した画像を Web サーバ班が作成したサーバに Dropbox を使用して画像ファイルを共有して、そこから Web サイトに自動でアップされるという仕組みである。

3.4.1 Unity 班

Unity 班は、Oculus Rift を使用して海中を体験できるアプリケーションの開発を目指した。VR アプリケーションの開発を小橋、工藤、松本の 3 人が担当した。まずはじめに、各人が Unity の入門書を勉強して、基礎を身につけた。それから、3 人で協力して開発を進めた。小橋が開発した魚の魚群アルゴリズムや魚を掴む機能、掴んだ時に振動する機能、工藤が開発した泡のオブジェクトにより、リアルな海中を再現することに成功した。また、仮想スマートフォンで VR 空間内を写真を撮影する機能も実装した。

(※文責: 酒田恭吾)

3.4.2 画像処理班

画像処理班は、VR 空間内の背景として使用する画像の調整、インスタ映えすることができるフィルタの開発を目指した。VR 空間内で使用する背景の画像の輝度差の調整を今泉が担当し、VR 空間内の仮想スマートフォンで撮影した写真を加工するフィルタの開発を長谷川が担当し、フィルタを扱うための GUI の開発を酒田が担当した。当初の予定では背景画像に動画を使用する予定だったが、VR 酔いの解決策を見つけれなかったため画像を使用することにした。VR360 は前後の 2 つのレンズで撮影しているため輝度差が生まれてしまうので、それを軽減するために Python で OpenCV を使用して軽減することに成功した。フィルター開発では、インスタ映えするようなフィルターを目指して 12 種類のフィルターを開発した。当初の予定では、OpenCV のような画像処理ライブラリよりも、よりインスタ映えをさせることができるツールを使用しようと考えていたが、学習に時間がかかるため断念した。UI 開発では、加工前と加工後をプレビュー表示したり、ドラッグアンドドロップ、回転、加工前に戻る、保存、重ねがけなど、様々な機能を実装したことでユーザーが使いやすいようにした。

(※文責: 酒田恭吾)

3.4.3 Web サーバ班

Web サーバ班は、加工した画像をユーザーが保存できるような Web サイトの開発を目指した。すべての仕事を渡邊が担当した。最初はサーバに関しての知識がなかったため思うような進捗はなかったが、先生の助力を借りながら着実に開発していき完成させた。Mac mini をサーバーとして使用した。photoswipe というライブラリを使用することでスマートフォンのギャラリーのような Web サイトを作ったり CSS を用いて背景色、フォントを変更して、ユーザーが使いやすいようにした。Web サイトの動作が重いという問題は、画像処理班と協力して、画像を png から jpg に変更と、画像のリサイズをしてファイルサイズを小さくすることで解消した。

(※文責: 酒田恭吾)

3.5 最終発表

最終発表では、メインポスター作成を酒田が担当し、サブポスター作成を長谷川が担当し、スライド作成と原稿作成を今泉が担当して、酒田以外の 6 人が 15 分ずつプレゼンテーションをした。

(※文責: 酒田恭吾)

第 4 章 課題解決プロセスの詳細

4.1 会議

会議の議題は、目標決定について、動画の撮影場所について、使用するカメラについて、アプリケーションの仕様についての 4 つであった。

4.1.1 目標決定

目標決定は、プロジェクト開始後すぐに行った。本プロジェクトの目的は、水中や魚の映像を用い、海中の様子やその魅力を伝えることである。対象となるユーザは決められていなかったため、大学生を含む若者を対象とすることを決定した。その後、若者が海に興味を持っているかを調べると、海に興味を持っていない若者が多いことが分かった。そのため、若者がなぜ海に興味を持っていないのか調べた。その結果、海に行くことが好きではない、泳ぐことが好きではないなどの問題点が挙げられた。そこで、プロジェクトの最終目標は、海に行かずとも海を楽しむことのできる新しい方法を提案することに決定した。その方法として、VR アプリケーションを開発することを決定した。VR を用いる理由は、最新技術であり、若者の興味を引くことができるであろうと考えたからである。アプリケーションのアイデアを出す手法として、BSKJ 法を用いた。

4.1.2 動画の撮影場所

動画の撮影場所の決定は、目標決定会議と並行して行った。撮影場所の条件として、きれいな映像を撮るために、水の透明度が高いことを条件とした。工藤の提案により、1 回目の撮影場所は入舟町前浜海水浴場に決定した。撮影後、水中に入る装備がなかったため、水中に入らずとも深い場所を撮影できる場所にする必要があると分かった。その結果、2 回目に函館市国際水産・海洋総合研究センター前の岸壁で撮影を行うことにした。だが、水の透明度が低く、良い映像が取れなかった。そのため、水中に入るための装備として胴長を購入し、水の綺麗な場所での撮影を行うことにした。渡邊の提案により、3 回目の撮影場所は黒岩岬に決定した。だが、撮影日は風が強く、うまく撮影ができなかったため、VR アプリケーションに使用できる映像が手に入らなかった。後期に入り、もう一度水の透明度が高い場所での撮影を行う必要があると考え、4 回目の撮影場所は再び入舟町前浜海水浴場に決定した。

4.1.3 使用するカメラ

使用するカメラの決定は、2 回目の撮影終了後、最初のプロジェクト活動日に行った。まず、担当教員の提案によって、黒岩岬での撮影に使用するために、全天球動画を撮影することのできる VR360 を購入した。その後、黒岩岬での撮影を行った。そこで撮影した映像を確認し、VR アプリケーションと相性の良い全天球動画を撮ることのできる、VR360 を使用することに決定した。

4.1.4 アプリケーションの仕様

アプリケーションの仕様についての会議は、後期に行った。VR アプリケーションのほかに、撮影した写真を加工するフィルタ、および加工した写真をスマートフォンに転送するための Web サイトを開発することにした。仕様が決まった際、PC の写真をスマートフォンに転送する必要があると判明した。スマートフォンは iPhone、Android の 2 種類がある。それぞれ PC の画像ファイルを共有する方法が異なるため、どのスマートフォンとでも写真を共有できるように、VR アプリケーションで撮影した写真を Web サイトにアップロードし、そこにスマートフォンでアクセスし、保存する方法を用いた。その後、それぞれの開発手法について決定した。VR アプリケーションはデスクトップ PC と Unity、フィルタはノート PC と Python、Web サーバは Mac mini と Apache を使用して開発することにした。

(※文責: 小橋元春)

4.2 撮影

撮影は、4 回に分けて行った。カメラは 3.2 節において説明した 4 つのカメラを使用した。

1 回目の撮影は、水の透明度の高い場所で、どのような映像が撮影できるか確認するために、入舟町前浜海水浴場で行った。使用したカメラは、GoPro と SP360 である。当初は Power Ray も使用する予定であったが、撮影場所が岩場であり、岩に衝突すると故障する危険があるため、使用は見送った。GoPro は工藤、酒田、長谷川、渡邊の 4 人が使用した。GoPro を BiRod の先端に取り付け、陸から海中を撮影した。SP360 は今泉、小橋、松本の 3 人が使用した。初めは SP360 をゴルダックに取り付け、鎖によって水中に沈め、浅瀬を撮影した。しかし、鎖で沈めるとレンズが上を向くため、水中があまり映らなかった。そのため、鎖を外し、ゴルダックを手で持ち、水中に横向きに沈めて撮影した。

2 回目の撮影は、深い場所での撮影を行うことと、障害物のない場所で Power Ray を使用することの 2 つを目的として、函館市国際水産・海洋総合研究センター前の岸壁で行った。使用したカメラは、SP360 と Power Ray である。SP360 は今泉、小橋、松本の 3 人が使用した。入船町前浜海水浴場での撮影と同じく、ゴルダックに SP360 を取り付け、鎖によって海中に沈めた。海の透明度が低く、SP360 の状態を陸から確認できないため、アンテナを用いて iPad と Wi-Fi 接続し、iPad の PIXPRO SP360 4K というアプリによって映像を確認しながら撮影した。Power Ray は工藤、酒田、長谷川、渡邊の 4 人が使用した。船などの障害物のない岸壁の外れに Power Ray を沈め、スマートフォンで操作し、撮影した。

3 回目の撮影は、水の綺麗な場所で撮影を行うことと、VR360 によって撮影した映像を確認するために、黒岩岬で行った。使用したカメラは、GoPro と VR360 である。また、水中に入るための装備として、胴長を使用した。しかし、撮影当日は強風、高波であり、波に流されてしまう危険があるため、両カメラとも浅瀬での撮影しか行えなかった。GoPro は工藤、酒田、長谷川、渡邊の 4 人が使用した。入船町前浜海水浴場での撮影と同じく、BiRod の先端に取り付けて撮影した。VR360 は今泉、小橋、松本の 3 人が使用した。SP360 での撮影と同じく、ゴルダックを用いた。しかし、VR360 は全天球動画の撮影ができるため、鎖が映ってしまう。そのため、鎖ではなく、透明な釣り糸を用いた。また、SP360 とは違い、iPad によって映像を確認できないため、撮影ボタンを押してから水中に沈めた。

4 回目の撮影は、VR アプリケーションに使用することのできる動画を撮影するために、入舟町前浜海水浴場で行った。使用したカメラは、VR360 である。この日はプロジェクト時間外の撮影だったため、今泉、小橋の 2 人が参加した。黒岩岬で撮影した映像には足が映っていたため、足が映らないように、海中の岩の上からカメラを沈めて撮影した。VR360 の各レンズの映像間の輝度を抑えるため、岩の日陰で撮影を行った。

(※文責: 小橋元春)

4.3 中間発表

中間発表は、スライドによる発表をメインに行った。15 分間の発表を 1 人が担当し、松本以外の 6 人がプレゼンテーションを行った。また、プロジェクトの前期の活動をまとめるためのメインポスターを 1 枚作製した。

発表スライドの作成は今泉、小橋、工藤、長谷川が担当した。前期の時点での成果物は動画のみであったため、スライドによる説明に時間をとることにした。また、本プロジェクトの特徴は海の動画を撮影することであるため、動画の撮影について詳細に書くことにした。使用したカメラは 4 種類あるため、それぞれのカメラの特徴と撮影した映像を見せながら、撮影についての説明を行った。

発表ポスターの作製は小橋、酒田、松本、渡邊が担当した。前期の時点で行った活動は会議と撮影であった。特に撮影は前期の時点で 3 回行っており、撮影した動画を成果物として扱うことができるため、撮影をメインに書いた。初めにレイアウトを作成し、文章量に合わせてレイアウトを微調整しながら作成した。

(※文責: 小橋元春)

4.4 アプリケーション開発

アプリケーション開発は後期から行った。まず、Unity 班と画像処理班の 2 つに分かれ、活動を行った。その後、Web サイトを使用することに決定したため、Web サーバ班を追加し、計 3 つの班で開発を行った。Unity 班の担当は小橋、工藤、松本の 3 人、画像処理班の担当は今泉、酒田、長谷川の 3 人、Web サーバ班の担当は渡邊の 1 人である。以下、各班の活動について詳しく説明する。

(※文責: 小橋元春)

4.4.1 Unity 班

Unity 班は、Unity と Oculus Rift を用いて VR アプリケーションを開発した。

Unity はゲームエンジンであり、3D 空間の描画が得意である。また、VR コンテンツの開発に対応している。Unity は、ゲームオブジェクト (以下オブジェクトとする) という単位で空間内の物体を管理する。そのゲームオブジェクトに位置や動き、形を設定することで、簡単に 3D 空間を作成することができる。また、Unity にはアセットというものが用意されており、ほかの人が作成した 3D モデルやオブジェクト、テクスチャなどを使用することができる。これらの利点があるた

Underwater World

め、開発に Unity を用いることに決定した。また、スクリプトというオブジェクトを動かすプログラムは、C#によって作成した。開発の環境は Unity2018 2.9f である。

Oculus Rift(以下 Rift とする) は VR ヘッドセットの 1 つである。Oculus Touch(以下 Touch とする) という両手に装着して使用するコントローラが付属しており、VR 空間に触れているかのように操作ができる。Rift および Touch の位置をトラッキングする方法として、赤外線センサが用いられており、同じ VR ヘッドセットである HTC VIVE に対し、設置が容易である。また、Unity での開発にも対応しているため、Rift を用いることに決定した。

Unity の勉強は、主に夏季休業期間中に行った。前期に課題を出し、夏季休業後に提出させることで、Unity の基本的な知識を習得した。工藤、松本の 2 名は、10 月に「Unity2018 入門 最新開発環境による簡単な 3D&2D ゲーム制作」という本を用いて勉強した [3]。それにより、Unity のより発展的な使い方を習得した。

Unity 班の仕事は 3 人に割り振った。小橋の担当は、全天球動画を Rift で鑑賞するための設定や、VR 空間の作成などである。工藤の担当は、サウンドに関するスクリプトの作成や設定、泡の作成などである。松本の担当は、魚を自動生成するスクリプトの作成である。

(※文責: 小橋元春)

小橋

小橋の担当は、全天球画像を鑑賞するための設定、Touch により VR 空間に干渉するための設定、魚の作成、写真撮影機能の実装、映像の調整である。

まず、全天球画像を鑑賞するための設定を行った。法線が逆である球オブジェクトに全天球画像をテクスチャとして貼り付け、その中心にカメラを設置することで、Rift によって全天球動画を鑑賞できるようにした。球オブジェクトは、インターネットで配布されていた sphere100 というものを使用した。

その後、Oculus Integration という、Oculus 社が作成したアセットを使用し、Touch によって VR 空間に干渉できるようにした。OVRCameraRig というオブジェクトを設置することで、Rift と Touch の位置を Unity 上で取得することができる。また、LocalAvatar というオブジェクトを設置し、Touch の位置に手を表示することで、実際に自分の手を動かしているような感覚で操作できる。

魚は、計算量の軽減、およびリアルな動きを再現するために、群れ単位で動作を管理することにした。魚の 3D モデルは、Unity のアセットを購入して使用した。魚の自然な動きの実装のために、Boids を使用した。Boids とは、Craig Reynolds 氏が発表した、群れのような動きを行うオブジェクトの集合体のことである [4]。各オブジェクトは、近隣の仲間との衝突を避ける、群れの重心に向かって移動しようとする、群れの平均移動ベクトルに従おうとする、という 3 つのルールに従って動こうとする。これらを魚のオブジェクトに設定することで、リアルな魚の群れの動きを表現した。このうち、群れの重心の計算および平均移動ベクトルの計算は群れを管理するオブジェクトに行わせた。それにより、それぞれの魚が計算を行う場合に比べ、計算回数が減るため、パフォーマンスが向上した。

魚の群れの目的地の計算方法は 2 種類用意した。1 つはランダムに移動する方法で、ランダムな位置を目的地に設定し、群れはそこを目指すように移動する。目的地は一定時間ごとに変わる。魚の動きが不規則的になるため、非常に単純な計算で自然な動きを表現できる。もう 1 つは円周上を移動する方法で、半径を設定し、その円周上を目的地に設定し、群れは円に沿って移動する。ラン

Underwater World

ダムな移動ではユーザの近くに行かない可能性があるため、円周上を移動する群れを配置することで、確実にユーザが魚に触れられるようにした。また、ランダム移動の場合、目的地によっては魚が不自然な動きを行う可能性がある。円周の場合、計算量は多くなるものの、より自然な魚の動きを再現できる。

写真撮影は、工藤が作成したスマートフォンオブジェクトによって行うようにした。スマートフォンに設置されたカメラの映像をテクスチャに変換し、そのテクスチャを png 形式の画像ファイルに変換し、保存することで、まるで本物のスマートフォンで写真を撮影しているかのように操作することができる。LocalAvatar の手の形と角度を調整し、実際にスマートフォンを持っている感覚で操作できるようにした。また、スマートフォンにズーム、フラッシュの機能を追加した。ズーム機能は、カメラの画角を調整することによって実装した。フラッシュ機能は、光源をカメラに追加し、撮影の瞬間だけ光るようにすることで実装した。それぞれの設定は Touch によって直感的に行える。オブジェクトを組み合わせて UI を作成し、現在の状態を簡単に確認できるようにした。

また、映像がより綺麗になるように、レンズフレア、フォグ、モーションブラーの3つのエフェクトを設定した。エフェクトの設定には、PostProcessingStack という Unity の標準アセットを使用した。PostProcessingStack とは、フォグやモーションブラーなどの映像を加工するエフェクトを一括して管理するものである。レンズフレアとは、きわめて明るい光源を見た際に生じる暗部への光の漏れのことである。主にカメラで撮影した際に起こる現象であるが、肉眼でもまつげや髪の毛の乱反射で起こるため、太陽を見た際のまぶしさの表現のために用いた。レンズフレアは、PostProcessingStack ではなく、LensFlare という Unity のアセットを用いた。背景画像上の太陽の位置にレンズフレア用の光源を設置することで、ユーザが Rift によって太陽を見た際に光がにじむエフェクトがかかるようにした。フォグとは、カメラからの距離に応じてオブジェクトに色を重ねるエフェクトである。青緑色のフォグを設定することで、水中の光の吸収を表現した。モーションブラーは高速で動く物体をカメラで撮影した際に生じるブレのことである。人間が視点を高速で動かした際に見えるボケを表現した。これらのエフェクトによって、肉眼で見た時のような自然な映像を作成した。

上で述べたエフェクトの有無や時間帯を変更できるオプションも作成した。これにより、ユーザは自分の好みの環境で写真撮影を行うことができる。時間帯の変更は、PostProcessingStack の ColorGrading という機能を使用した。ColorGrading とはレンダリング後の画像の輝度や色彩を調整する処理である。これにより、全体の色彩を橙色に調整することで、夕方の時間帯を作成した。オプションの調整は、Touch によって簡単に行うことができるようにした。

(※文責: 小橋元春)

工藤

Unity を自分のパソコンにダウンロードし、基礎的な本を用いて操作方法を学んだ。

次に、班のメンバーがそれぞれ何を担当するかを決め、海中を舞台とした VR アプリケーションにおいて欠かせない要素を Unity で作成した。担当したことは、スマートフォン型オブジェクトの作成、音を再生するスクリプトの作成、および泡オブジェクトの作成である。作業の流れは、下記の4段階から成る。

1. スマートフォン型オブジェクトの作成
2. 1. の制作物にカメラ機能や音声を追加
3. 泡の形を作成

4. 3. の制作物に動きや音声を追加

スマートフォン型オブジェクトは、本体、画面、カメラから成るオブジェクトである。まずは空ゲームオブジェクトを作成して SmartPhone という名前に設定し、そこに他のオブジェクトが収納できるようにした。次に、本体を作成するため Cube オブジェクトを 1 つ用意して Body という名前に設定し、それを先ほど作成した SmartPhone に収納した。その後には Plane オブジェクトから作成した Display という名前のオブジェクトを作成した。Display は縦横比が 3:2 となる平らな板のような形にしており、スマートフォン型オブジェクトにおいて画面の役割を果たすため、Body の表側に貼るような形で接合させた。この制作物をカメラとして使えるようにするため Camera オブジェクトで PhoneCamera という名前のオブジェクトを作成し、Body の裏側の中央部分に搭載した。最後に、スマートフォン型オブジェクトのボタン部分を作成するために Cylinder オブジェクトで Button という名前のオブジェクトを作成し、ボタンの形に変えて Body の表側に貼るような形で接合させた。作成した Display などのオブジェクトは、Body と同様に SmartPhone に収納することで、SmartPhone が多数のオブジェクトを持つ形に設定した。このようにして作成したスマートフォン型オブジェクトは単純な形だけで済むため、基礎的な本から得た知識のみで容易に作成することができた。作成中は、最終発表の聴講者がなじみ深く感じるようなものにするを考え、iPhone のような形で作成するように心掛けたため、実際に存在しているようなスマートフォンの形を作成することができた。

作成したスマートフォン型オブジェクトにおいて先ほど搭載した Phone Camera にカメラ機能を付け加えるため、Unity のフォルダ内に RenderTexture を新しく追加し、DisplayTexture という名前に設定した。次に、DisplayTexture を Phone Camera において TargetTexture に指定することで Phone Camera の映像をテクスチャとして扱い、それを Display に映すシステムにしてカメラ撮影をしているように見せたことによって、スマートフォン型オブジェクトにカメラ機能を付け加えることができた。この機能は Button の部分を押すことでカメラ機能を使うことができるシステムにしてあるため、実際にスマートフォンで写真を撮ることができるような臨場感を加えた。

この機能にはシャッター音が欠かせないため、インターネット上に配布されているフリー素材である音声を Unity に読み込み、新たなスクリプトを加えることによってシャッター音が鳴る動作が反映させるソースコードを作成する必要がある。そこで、Sublime text という名前のテキストエディタを用意し、指定された操作が行われたら音声が鳴る仕組みのソースコードを書き、それを Sound.cs という名前のスクリプトとして Sublime text に保存した。この際には、Audio Source というコンポーネントをオブジェクトに設定する機能を使用した。

泡の形を制作するには、3D オブジェクトとして制作すると複雑な構成のオブジェクトを作成する必要が出てくるので、スマートフォン型オブジェクトの場合よりも格段に難しくなる。そこで、作成する際の難易度を抑えるために 2D オブジェクトとして作成することに決定し、インターネット上に掲載されていた泡の画像を参考にしながら泡の画像を 10 種類描いた。これらの画像はすべて大きさや描き方が違っており、画像の一部が透過の状態になっているため、本物の泡であるような印象が出るようになった。また、泡の画像が透過の状態になるように保存するには、png 方式として保存する必要がある。これらを完成させた後は Unity に新しいオブジェクトを作成し、Sprite として読み込んだ。10 種類用意した泡の画像を搭載させるオブジェクトを Bubble_aa、Bubble_bb、Bubble_cc、Bubble_dd から成る 4 つのグループに分割させ、Bubble_aa と Bubble_bb には 2 種類ずつ、Bubble_cc と Bubble_dd には 3 種類ずつの泡の画像を Sprite として読み込んだオブジェクトを収納させることで、Unity のシーンを動かした際に発生する泡のパターンを多数用意することができた。この 4 グループから成るオブジェクトに Rigidbody を追加

して質量をそれぞれ 1 にすることで、泡のオブジェクトに物理特性によって制御することも可能にした。

この際に作成する泡は、ランダムな場所と時間において出現して左右に揺れながら上昇し、一定の高さにまで達したら消滅する動きをつける必要があったので、泡の動きは時間とともに y 座標の位置を高くしたうえで +sin 関数によって左右に振動させることによって実装をすることにした。そのため、シャッター音での作業と同様に新たなスクリプトを加えて一連の動作が反映されるソースコードを作成した。それを行うための準備として、2つのスクリプトを作成して名前を Makebubble と BubbleEffect に設定した。Makebubble ではランダムな場所と時間に泡を出現させる処理となるソースコードを作成するため、public GameObject [] bubbles を設定し、先ほど完成させた4つのグループから成る泡のオブジェクトをその bubbles に格納させる処理を行った。必要としているランダムな範囲を表現するために float 型の変数である generateTimeMin と generateTimeMax を作成し、指定された範囲の乱数を返却する Random.Range の最小値と最大値として設定して Random.Range を float 型の変数 t に代入した。void GenerateBubble(GameObject bubble) において float 型の変数である x,y,z を呼び出し、x と y には先ほど作成した Random.Range を代入して y には 0 を代入した。この後は Vector3 型の変数である position に new Vector3(x,y,z) を代入することによって、泡が発生する場所を x 軸と z 軸から成る平面のみに限定させることができた。このような処理は一定時間待機してから行われる流れを繰り返し起こす必要があるため、float t への代入が行われた後に StartCoroutine ("WaitRandomTime"); を設けることによって、コルーチンを実行させた。BubbleEffect では左右に揺れながら上昇する処理と一定の高さで消滅する処理となるソースコードを作成した。泡の形をしたオブジェクトに物理特性を加えた Rigidbody を動作させるために GetComponent<Rigidbody>() を変数 rigid に代入し、物体の座標変更を可能とするために変数の型が Vector3 である pos に transform.position を代入した。このオブジェクトにはランダムな頻度で泡の音を出現させる機能も必要であるため、gameObject.GetComponent<AudioSource>() に audioSource を代入し、audioClip1 を audioSource.clip に代入し、audioSource.Play(); によって音を鳴らす処理を施した。また、この音が鳴る処理はコルーチンによって繰り返し実行される仕組みにした。泡の形をしたオブジェクトが左右に揺れながら動いて途中で消える動作は void FixedUpdate() によって行った。その中で pos.y+speed を pos.y に代入し、rigid.MovePosition によって泡を上下左右に動かすことができる形とした上で、Mathf.PingPong(Time.time, 1) という関数によって泡が左右に揺れるように動くシステムにした。一定の高さまで上昇したら泡が消えるようにするため、もし pos.y の y 座標の値が 15 を上回ったら Destroy(gameObject) を働かせるシステムにした。スクリプトによって設定したこれらの動作を泡の形をしたオブジェクトにすべて適用可能にするために BubbleManeger という名前のオブジェクトを作成して MakeBubble をそこに読み込み、Bubble_aa から Bubble_dd までの4つのオブジェクトには BubbleEffect を読み込んだ。必要としている泡の音を鳴らすためにフリー素材である音声を Audio Source に新しく読み込ませ、先ほど BubbleEffect を読み込ませた4つのオブジェクトにこの Audio Source を搭載することによって音が出るシステムにした。このように、泡の形をしたオブジェクトに音声や動きをつける際には現実には存在する泡と変わらないような動作にするよう心がけることで、備え付けた設定を VR アプリケーションに反映させたときには現実的に見せるようにした。

(※文責: 工藤祐弥)



図 4.1 泡オブジェクト

松本

Unity に関する知識を全く持ち合わせていなかったため、基礎的な操作方法が書かれた本を使って、学ぶところから始めた。その際に、Unity の操作に慣れるために、人モデルを動かし、迷路を抜ける 3D ゲームを作製した。

そのあと、Unity 班のメンバーで話し合い、VR 空間への没入感を高めるための構成要素である、魚を作製することに決まった。最初に、魚のモデルを作製するために、Cube の 3 つを頭、胴体、尾に見立てて作成し、動かして魚のような動きをする物体を作ろうと挑戦した。しかし、魚が泳ぐ動きが出来ず、平行移動してしまう状態から作業が進まなくなってしまった。

そこで、情報ライブラリーやインターネットを利用して、魚に近い動きをさせるようなスクリプトの作成方法を探していたが、良いものを見つけることができず、自分の力で試行錯誤を繰り返すことになった。Animation を利用したりなど、多くの方法を試したが、最後まで魚のモデルを作製することができなかった。そこで、話し合いの結果、Asset Store から魚のモデルを購入することに決定した。

Asset Store で手に入れた魚のモデルそれぞれを、配列にして呼び出せるように設定し、ランダムな魚のモデルを返す関数を作成した。使用した魚のモデルは、alligator gar(アリゲイターガー)、bass(バス)、bluegill(ブルーギル)、brook trout(カワマス)、catfish(キャットフィッシュ)、longnose gar(ロングノーズガー)、salmon(サケ)、salmon female(サケ族メス)、trout(マス)、walleye(ウォールアイ) の 11 種類である。そして、ユーザの近辺に 5 体の魚が出現するように Instantiate を使って作成した。これは、ユーザが魚に触れやすいように工夫している。Prefabs を利用して、与えられた魚のモデルと同じものが数多く生成されるように設定し、簡単に呼び出せるようにすることができた。

次にユーザの遠方に魚が泳いでる様子を見せるために、魚の群れが 15 個出現するようにスクリプトを作成した。ユーザに近い場所も遠い場所も、ランダムに出現するように設定し、アプリケーションを起動しなおした時に、全く同じ景色にならないようにした。

(※文責: 松本魁斗)

4.4.2 画像処理班

画像処理班の役割は、大きく分けて 2 つある。1 つは背景画像処理、もう 1 つはフィルタアプリケーション作成である。

背景画像処理とは、VR アプリケーションの背景に使用されている画像の輝度を調整する仕事である。今泉が担当した。

フィルタアプリケーション作成とは、VR アプリケーション内で撮った写真を加工するためのアプリケーションを作成する仕事である。フィルタ自体の作成は長谷川が行った。フィルタを写真に適用するためのアプリケーションの作成は酒田が行った。

以下にそれぞれの仕事の詳細を書く。

(※文責: 小橋元春)

背景画像処理

今回使用したカメラは VR360 である。このカメラの詳細は前述しているが、2つレンズがある。

画像処理のした仕事は、画像を全体でつなぎ目のない1枚の画像として見えるように全体の輝度を調整して VR 越しに見た時に違和感がないような画像にすることであった。VR では画像がかなり引き延ばされていたおり、四角状のブロックノイズが発生しやすかったため、正確に調整しなくてはならなかった。

初めは、動画を VR アプリケーションの背景として使用する予定であったため、動画を1フレームごとに分割し、それを1枚ずつ輝度調節し、再度動画化しようとした。しかし、前期の反省として、VR は第1人称視点であり、視点であるカメラが海中で揺れているとひどい動画酔いが起きてしまう。そのため、動画酔い対策を施す方法を探したところ、私たちが使用したカメラでは、ステッチという2つのレンズで撮影した動画を違和感が小さくなるように1つの動画として張り合わせる作業を使用したカメラ専用のアプリケーションである「PIXPRO 360 VR Suite」の機能で自動で行っているため、今回は撮影自体をカメラが揺れないように行う以外では VR 酔いの対策はできないことが判明した。その結果、動画ではなく動画から切り抜いた静止画像を使用することに決定した。

その後の作業の内容は大きく分けて、「マスク画像の作製」、「画像の分割」、「各画像の平均の輝度値の算出」、「平均輝度値を基に γ 補正を用いて、各画像間の輝度差の最小化」、「画像を1枚にする」の5つである。画像処理の流れは以下の通りである。

1. マスク画像を表裏作製
2. マスクをそれぞれかけて画像を2つ作成
3. 2. の画像の黒の画素の数を算出し、全体の画素から引くことでマスクされた箇所以外の画素数を求める
4. 2. の画像をそれぞれグレイスケール化し、輝度値の合計を3. で求めた画素数で割ることでそれぞれの平均の輝度値を算出
5. 1枚の平均の輝度値をもう一枚の輝度値で割ることで、平均輝度値が何倍かを求める
6. γ 補正で2画像のつなぎ目が目立たなくなるように片方の明るさを調整
7. 完成した画像を Rift で確認して、修正箇所をまとめる
8. 1-7 を繰り返して、つなぎ目と全体の輝度が納得できるものになれば完成

これをまず2つのレンズに分けて行った。初めは片方の画像の輝度値をもう片方の輝度値に合わせるように調整を行ったが、それで完成した画像を Rift で見ると全体的に明るすぎ、目が疲れてしまうと感じた。また、2つの画像の平均輝度値を合わせたものの、1画像の中にもかなり輝度値の差があるため、つなぎ目の部分は輝度差がまだ有るままだったため、つなぎ目がはっきりわかってしまった。そこで、5% ずつ γ 補正で明るさを調節して画像を作成し、Rift で確認をしながら、つなぎ目が目立たない輝度の組み合わせを模索し、その後に全体の輝度を最も見ていて疲れないよ

Underwater World

うに調節した。その後、防水ケースにの防水機構の為に厚くなっている部分は他とは違う光りかたをしていたため、その部分に対しても同様に行った。ここで、各用語の解説をする。

マスク画像とは、指定した部分のみを表示し、それ以外の部分を表示しないようにする画像処理を行うための画像である。マスク画像の生成については、初めの2レンズ間のマスク画像は、カメラのレンズのうち片方にカバーをかけたまま撮影し、それを基に黒いカバーのかかっていた部分以外を白にすることで表のマスク画像を作成した。防水ケースの厚い部分のマスク画像は、違和感が大きい部分とそうではない部分があったため、デスクトップツールのペイントを用いて、違和感の多いところを手動で塗りつぶして表のマスク画像を作成した。そして、白と黒の値を変えるプログラムを作成し、表のマスク画像に使用して、裏のマスク画像を作製し、それを使用してそれぞれのレンズごとの2つに画像と、分けて輝度調整をした。

輝度とは、写真をグレースケールという白と灰色の明暗、そして黒のみで表示した際に、黒を0、白を255とした時の明るさの値を示すものである。2つの画像の輝度の平均を出すことでそれぞれが全体的にどれくらい明るいかが分かり、つなぎ目の輝度差が小さくなるように調整したり、全体の明るさを目が疲れないう程度まで減らす時の参考にすることができる。

γ 補正とは、ディスプレイなどに画像を表示する際の明るさを補正する機能のことである。これを使用して、2つの画像間の輝度値を合わせたりすることで一つの写真のようにつなげたり、全体の明るさを変更することで目が疲れないう画像を作成することができる。

画像処理の反省として、防水ケースの厚い部分のマスク画像の作製は手動で行ったが、カメラの揺れや時間の経過によって太陽の位置が変わると、ケースで光が屈折する度合いが変わり、作成したマスク画像を作成することができず、自動でマスク画像を作成する方法を探す必要がある。

(※文責: 今泉慧)

フィルタ作成

Instagramを意識した加工フィルタを作成することがプロジェクトの方針に決まり、まず「どのような技術を用いるか」「どのようなフィルタを何種類作成するか」などといったことを考えた。「どのような技術を用いるか」については、同じ画像処理班である酒田と話し合いGPUのプログラマブルシェーダーを用いて、画像のフィルタ処理を行うコンポーネント及びAPIであるCore imageを用いようとしたが、難易度が高く自分達には扱いきれないと考えた。その結果、画像処理でよく使用されるpythonとopencvを用いて酒田はwxpythonを用いたGUIアプリケーションの作成、自分はGUIアプリケーションの機能である加工フィルタの作成を主に行うことに決定した。

次に「どのようなフィルタを何種類作成するか」については、まず、初めに「Instagram」や「snow」などといった若者に人気の高い写真などの加工機能が搭載されたアプリケーションを調べ、傾向や加工の種類などを勉強するとともに「python」や「opencv」を用いた画像処理の学習をした。フィルタの作成に関しては、作業を進めていく過程でpythonとopencvでは「Instagram」などの加工に近づけるには限界があることが判明し「Instagram」には無いような映える加工フィルタを目指して開発を進めた。

結果、12個のフィルタを作成した。作成したフィルタの一覧と特徴は、表4.1に示す。フィルタの画像は図4.2から図4.13に示す。

(※文責: 長谷川涼乃)

表 4.1 フィルタ表

ルックアップテーブルフィルタ	画像を明るくするフィルタ。0 埋めの 256 の配列を作り、0 から 255 までの値を計算して配列に入れるガンマテーブルを作成し、ガンマを 1.8 に設定して、ガンマ補正をすることで作成した。
ルックダウンテーブルフィルタ	画像を暗くするフィルタ。ルックアップテーブルフィルタと同様に、0 埋めの 256 の配列を作り、0 から 255 までの値を計算して配列に入れるガンマテーブルを作成し、ガンマを 0.5 に設定して、ガンマ補正をすることで作成した。
グレイフィルタ	画像をグレースケール化するフィルタ。cv2.cvtColor(src,cv2.COLOR_BGR2GRY) と cv2.cvtColor(screen,CV2.COLOR_BGR2GRAY) の 2 つの式を用いることで作成した。
モノクロフィルタ	画像を白黒にするフィルタ。グレースケール化後、判別分析法を用いて 2 値変換を行い画像を白黒にした。
ネガポジフィルタ	画像の色を反転させるフィルタ。画像を読み込み、negaposhi = 255 - img の式を用いることで色反転を行った。
クリーンフィルタ	画像を鮮やかなエメラルド色にするフィルタ。青成分である B 値を 70、緑成分である G 値を 30 増やし、赤成分である R 値を 20 減らすことで作成した。
ブルーフィルタ	画像を鮮やかな青色にするフィルタ。青成分である B 値を 100 増やすことで作成した。
RGB フィルタ	画像を夕方風にするフィルタ。赤成分である R 値と青成分である B 値を入れ替えることで作成した。
エンボスフィルタ	画像中の物体の境界線がはっきりするフィルタ。グレースケール化し、画像に畳み込み処理をし、配列を与えてオフセット値を 128 に設定して OpenCV の「cv2.filter2D」メソッドを使用することでエッジ検出した。
漫画フィルタ	画像を漫画の 1 コマのようにするフィルタ。グレースケール化し、Canny アルゴリズムで輪郭検出し、色反転、三値化、その三値画像と輪郭画像と斜線のスクリーントーンを合成することで作成した。
水彩画フィルタ	画像を水彩画で書いたようにするフィルタ。減色処理を行った後、グレースケール化し、ノイズ低減、Canny アルゴリズムで輪郭抽出したものを RGB 色空間に変換。そして、画像の領域分割した差分を返すことで作成した。
ペンキフィルタ	画像をペンキで塗ったようにするフィルタ。単純な配列の参照処理で置き換えて効率化を図るため、ルックアップテーブルを作成し、グラフィックの階調（色数の表現段階）を下げる画像処理であるポスタリゼーションを行いペンキ風に仕立てた。



図 4.2 LUT フィルタ



図 4.3 LDT フィルタ



図 4.4 モノクロフィルタ



図 4.5 ネガポジフィルタ



図 4.6 グレーフィルタ



図 4.7 クリーンフィルタ



図 4.8 ブルーフィルタ



図 4.9 RGB フィルタ

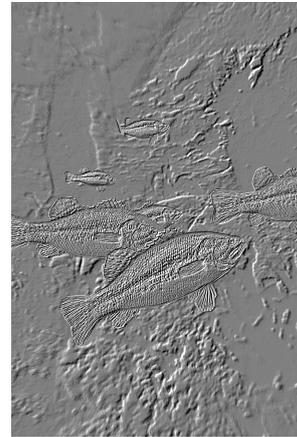


図 4.10 エンボスフィルタ

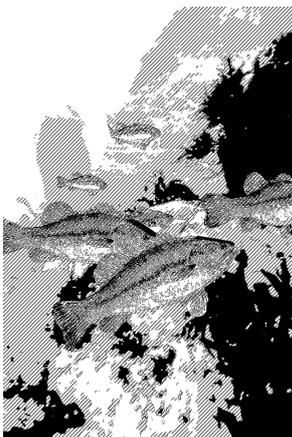


図 4.11 漫画フィルタ



図 4.12 水彩画フィルタ



図 4.13 ペンキフィルタ

フィルタアプリケーション作成

フィルタアプリケーション作成では、作成したフィルタを手軽に使えるようにするということが目標なので、どのような技術で開発するかについて話し合った。Web サーバ班が作成する Web サイトに、Web アプリケーションとして実装しようと考え、Django (Python で実装された Web アプリケーションフレームワーク) で Web アプリケーションを開発するための基礎を学習をしたが、学習に時間がかかりすぎてしまうので断念し、wxPython (Python で記述されたクロスプラットフォームなウィジェット・ツールキット) を使用しての開発に決定した。

フィルタアプリケーションの開発は酒田が担当した。まずは、wxPython についての勉強から始めた。wxPython の基礎についての学習はすぐに終わった。作成したフィルタアプリケーションの大まかな機能は、ボタンを押したらフィルタを適用する機能、画像をドラッグアンドドロップで選択できる機能、加工前の画像と加工後の画像をプレビュー表示させる機能、フィルタを重ねがけする機能、保存する機能、保存した画像を削除する機能、加工する前に戻る機能、回転する機能、フィルタの説明をするツールチップの 9 つである。ボタンを押したらフィルタをかける機能では、作成した 12 種類のフィルタを 1 つずつ関数にして wx.Button という関数で呼び出すことで実装した。画像をドラッグアンドドロップで選択できる機能では、wx.FileDropTarget というクラスを継承した MyFileDropTarget というクラスを作成した。しかし、ドラッグした画像を表示することはできるが、画像パスを取得できなかったため、フィルタをかけることができなかった。そこで、ドラッグしたときにもう 1 枚同じ画像を copy.png という名前で複製することで加工する関数の中で使用する画像の名前を copy.png に統一することができ、解決することができた。加工前の画像と加工後の画像をプレビュー表示させる機能では、wx.Panel というクラスを継承した ImagePanel というクラスを作ることで、ドラッグアンドドロップした画像を表示することに成功した。加工するごとに copy.png に上書きされそれを加工後の画像を表示する場所に上から貼り付ける処理を行ったことで実装した。フィルタを重ねがけする機能では、copy.png に毎回上書きをすることで実装した。保存する機能では、Web サイトにアップした際に画像サイズが大きくて通信が重くならないように、リサイズの処理と png から jpg への変換を行った。これにより画像サイズを半分以下にすることができた。削除する機能では、os という Python 用の os に依存した機能を使用できるライブラリを使用することで PC 内に保存した画像を削除できるようにし、誤作動がないようにダイアログで削除するかの確認をするようににした。加工する前に戻る機能では、ドラッグアンドドロップした際に copy2.png という名前のオリジナルの画像を保存することでそれを読み込むことで実装した。回転する機能では、numpy という Python 用の数値計算ライブラリの tranpose で転置し、numpy 配列の順番を逆順にする方法で実装した。フィルタの説明をするツールチップでは wx.ToolTip と wx.SetToolTip という関数を使用して実装した。フィルタアプリケーションのデザインは、wxPython ではあまり自由なデザインをすることができないということが後から判明したので納得のいくものはできなかった。ボタンのレイアウトなどは、wx.Sizer で配置するのが最適だが概念を理解できなかったため絶対位置で指定して配置した。完成したアプリケーションを py2app というツールを使用して Mac 用のアプリケーションにした。その際アイコンなども設定して Umistagram らしさを出した。

発表会では酒田の PC を使用するので酒田の PC の Finder と OneDrive (マイクロソフトが提供する Windows Live のサービスの 1 つである基本無料のオンラインストレージ) を同期して VR アプリケーションの画像ファイルと共有をした。酒田の PC と Web サーバの画像の共有は酒田の提案で Dropbox (Dropbox, Inc. が提供しているオンラインストレージサービス) を使用した。

表 4.2 フィルタアプリケーションの機能

機能	説明
フィルタを適応	ボタンを押すとフィルタが適応される。
画像をドラッグアンドドロップで選択	他のウィンドウからも画像をドラッグアンドドロップで選択できる。
加工前の画像と加工後の画像をプレビュー表示	左側に加工前の画像を表示して、右側が加工後の画像を表示することで比較しやすい。
保存	W ボタンを押すことで保存される。
削除	ボタンを押すことで確認画面が表示され完了するとフォルダ内の写真が削除される。
加工する前に戻る	オリジナルの画像に戻る。
回転	ボタンを押すことで 90°ごとに回転。

(※文責: 酒田恭吾)

4.4.3 Web サーバ班

Web サーバ班の仕事は、Web サーバの構築と、スマートフォンに写真を保存するための Web サイトの作成の 2 つである。担当は渡辺である。以下、それぞれの仕事の詳細を書く。

(※文責: 小橋元春)

Web サイト作成

まず最初にウェブサイトを作成するにあたって、python についてある程度理解があったため、python の web アプリケーションフレームワークである Django を利用しようとした。そこで Django について調べ、簡単なウェブサイトを作成したが、そのサイトを外部に公開する方法がわからなかったため、サーバについて調べることにした。

今回のサイトの内容としては、画像保存用のサイトであり、画像を一覧表示させそこから気に入った写真を選んでもらい保存するだけのサイトであったため、Django を利用しなくても作成することができると思った。また同じような理由で、bootstrap などの web アプリケーションフレームワークを利用しなかった。

まず最初に HTML、CSS、JavaScript、PHP といったウェブサイトを作成するうえで必要となる言語を学ぶことから始めた。そこで簡単な HTML を利用したウェブサイトを作成してみたが見た目が良くないという点があった。そこで、ウェブサイト用テンプレートというものがあったため、レスポンシブでギャラリー風ライブラリなテンプレートを利用しようと思った。調べた結果、photoswipe というライブラリがあり、それを使用しようと思った。

今回は、1design[5]にある、jQuery プラグイン化した photoswipe を利用したギャラリー風サイトのデモを使用した。jQuery とは、複雑な記述が必要な JavaScript を少ない記述で利用できるようにしたものである。また photoswipe とは、ギャラリー風ライブラリであり、スマートフォンに

対応したサイトを作成でき、タッチジェスチャーで画像の拡大・縮小、スワイプで次の画像に変更、画像の自動ローディングなど、さまざまな機能を持つライブラリである。photoswipe のデモを利用するにあたって、まず Windows 上でデモサイトをダウンロードし、scp コマンドを使いサーバへ送った。

今回、web サイトを作成するにあたっていくつかの問題点が発生した。

1 つ目の問題は、デモサイトでは画像を専用のディレクトリにあらかじめ入れ、その枚数の数だけ html に記述なのだが、自分が作成した web サイトでは画像が加工用 PC 側で写真を保存・全消しを頻繁にするため、ディレクトリ内の画像の枚数がすぐ変わってしまうという問題。

2 つ目の問題は、画像の枚数によって表示される際に状況が変わってくるという問題。

3 つ目の問題は、サイトの表示速度がサイトに表示される画像の枚数次第で非常に遅くなってしまいう問題。

その他の細かい問題。

それらの問題を解決するための具体的な方法を記述していく。

1 つ目：今回使用している拡張子で、ディレクトリ内にある jpg のファイルをすべて検索し、for 文を利用しカウントすることで、ディレクトリ内の画像を枚数分サイトに表示させることで解決することができた。

2 つ目：今回のサイトでは最大 4 枚、最小 2 枚横に並べ画像を表示させそれ以上になった場合は次の行へ画像を配置するという方法をとっている。CSS の記述で、画面の横のサイズが 641px 以上の時は画像を均等に横に 4 枚ずつならべ、4 の倍数の画像の時に `imageBox_last` というクラス（以下 last とする）を利用し、画像を次の行へ画像を並べ、641px 未満の時は `imageBox_left` という画像を左に配置するクラス（以下 left とする）と、`ImageBox_right` という画像を右に配置し次の行へ移動するクラス（以下 right とする）を用いた。デモの場合、もともと表示させる枚数が決まっていたため、一つずつ left、right、left、last、left…とクラスを変え記述することで画面の横サイズが 641px 以上の際は left、right を読み込まず last のみの読み込み、641px 未満の際は last は読み込まず、left、right のみの読み込みのため、問題がない。しかし、自分が作成した web サイトでは、PHP で for 文を利用していたため、ディレクトリ内の画像の枚数分ループさせようとしても記述内容が毎回一緒となり、left のみとなってしまう問題があった。そこでまず、ループ回数を半分にし、left、right を一回のセットとしループを行うという方法をとった。しかし、その際ディレクトリ内の画像の枚数が奇数だった場合、読み込む際に 2 枚セットで読み込んでいるため必ず偶数となり、一つ多く画像を読み込んでしまい、サイトが存在しないファイルを参照しようとするという問題が起きた。その問題を解決するために、if 文でディレクトリ内の画像の枚数が奇数と偶数の場合を考えた。偶数の場合はそのまま、left,right を 1 セットとしてループさせ、web サイトの表示数が横 4 枚のため 4 枚目の時に、right を last に if 分を用い変えることで、画面の横サイズが 641 以上と 640 未満に分けた表示ができた。奇数の場合は、もし画像が 1 枚だけの場合、まず for 分には入らず for 分とは別に一つだけ用意した left を用いることで 1 枚だけの際に 2 枚目を読み込もうとする問題をなくし、画像が 2 枚以上の場合は、3 枚であったら left,right を 1 セット、5 枚であったら 2 セット、7 枚であったら 3 セットのように枚数より一枚少ないセット数ループを行い、残りの 1 枚をセットでカウントしている for 文とは別に用意した left を用いることで奇数分の画像を表示させる際の問題を解決することができた。

3 つ目：本来は、サイトを開き画像をタップした時に画像が拡大表示され、スワイプすることで次の画像を表示させるなどといったことができるのだが、photoswipe の仕様として、サイトを開いた際にサムネイルと拡大した際の元の画像を全て読み込んでしまうため、サイトを開いた瞬間に

画像をタップすると全ての画像を読み込む関係上まだ読み込みが終わっておらず、画像だけを表示させた別タブで開いてしまうという状況にあった。その原因の一つに、png形式の画像を使用していたという点があった。png形式は可逆圧縮で画質を落とすことがないが、ファイルサイズが非常に大きくなってしまふ。Jpg形式なら非可逆圧縮で圧縮した際など画質を落としてしまふが、ファイルサイズが大きいものをとても小さくすることができるため変更することにした。

また、ImageMagickという画像を操作・表示させるためのソフトウェアスイートにあるqualityコマンドを使うことで画像サイズを落とそうとしたが、一回一回コマンドを入力する必要があった。crontabを用い、winpicのディレクトリに修正後の画像の追加を自動化したが、rsyncで自動同期するようになっていたため、winpicのディレクトリに修正後の画像を追加したと同時に修正後の画像が入っていない元のディレクトリと同期し、修正後の画像が削除されてしまうという状況が起きた。そこで元のディレクトリに修正後の画像を追加することで問題は解決したが、加工用PC側で全消しを行った際、元画像であるUmista****.jpgという画像のみをすべて削除というようなプログラムであったためあとから無理やり追加した修正後の画像を削除させるようなプログラムを打つ手間があり、また加工用PCでjpg形式にするだけである程度軽量化できていたため、最終的には使用しないということになった。

その他：他にも、サイトの見た目を改善するために、ヘッダーを作成し、Umistagramというタイトルを実際にInstagramで使用されているBillabongというフォントに変更した。その際、BillabongというフォントがPCに適用されていなかったときにそのフォントを見ることができない場合があったため、そのままフォントを使用するのではなく、文字を文字アイコンとして画像にし保存することができるサイトを利用し、BillabongフォントのUmistagramというタイトルを文字アイコンにし、画像としてヘッダーに貼ることで、画像として扱われるため、どのようなユーザーがwebサイトに訪れても、Billabongのフォントでタイトルを閲覧できるよう工夫した。

以上のような方法で、それぞれの問題を解決した。

他にも、背景を薄い青のグラデーションにすることで海をイメージしたサイトを作成した。ヘッダーはCSSを使い、上にスワイプした際にヘッダーが0.5秒かけてサイト上部に隠れ、下にスワイプした際に現れるというアニメーションを追加した。背景は青のグラデーションのかかった画像を横一面に埋まるまで張り続けるという記述をすることでサイトの横幅が変わっても対応できるよう作成した。

(※文責: 渡邊建)

サーバ構築

サーバ構築では、まずどのサーバを利用するか選ぶところから始めた。候補としては、渡邊のノートPC、教授のMac mini、レンタルサーバの3つがあった。はじめは渡邊のノートPCをサーバにしようとするところから始めた。そこでApacheやnginx、CGIなどといった様々な用語やシステムについて調べた。その際、Windows10の機能であり、WindowsでLinuxを利用可能にするWindows Subsystem for LinuxにLinuxディストリビューションであるUbuntuを導入し、それを利用しサーバを構築するという方法があった。しかし、知識・実力不足であったため、渡邊のPCをサーバにすることを断念することにした。

次にMac miniかレンタルサーバの二つの選択肢があったが、高先生の提案により、Mac miniをサーバとして利用することにした。その際、UltraVNCというコンピュータをリモート制御するツールを利用することでMac miniをGUIで操作することができたが、渡邊のPCでソフトを

起動した際不具合がありソフトを利用できなかったため、渡邊の PC をサーバにする際に導入した WSL の Ubuntu を CUI 操作し、教授の Mac mini に SSH 接続するという方法で操作を行った。

画像の共有は、VR アプリケーションと加工用 PC 間では One Drive、加工用 PC とサーバ間では Dropbox といったクラウドストレージを利用することで画像の共有を可能にした。VR アプリケーションでの画像の保存先を One Drive のプロジェクト用ディレクトリにし、加工用 PC でそのディレクトリから自分が加工したい画像を選んでもらい、加工し保存してもらおう。加工し保存した画像の保存先を Dropbox の output というディレクトリにすることで、自動的に画像ファイルをサーバに共有することができる。またその際、画像サイズを小さく変更したサムネイル用の画像も、Dropbox の thumbnail というディレクトリに保存するよう設定した。

次に、画像をウェブサイトに表示させる際、サイト用の PHP ファイルが root 権限上の階層にあり、その場所に画像用ディレクトリを置いていることにため、加工用 PC からサーバへ画像を転送する際に root 権限の関係から、一度ホームディレクトリに画像を送り、そこからさらに mv コマンドで画像を root 権限上の画像ディレクトリに移動させるという手間があった。そこで、その階層に winpic と thumbnail というディレクトリを作り、Dropbox の output と thumbnail にシンボリックリンクすることで、簡単に加工用 PC からサーバの root 権限上にある画像ディレクトリに転送できるよう工夫した。

画像を表示させる際に Dropbox を相対パスとして指定すると表示させることができなかった。その解決方法としては、DropboxAPI というものを利用することで表示させることができるのだが、自分の知識不足によりできず断念した。別の解決方法として、新しいディレクトリを作成し、crontab を用いて自動的にそのディレクトリと同期するという方法を用いた。まず、Pic というディレクトリを用意した。そのディレクトリと Dropbox の output を、crontab を用い rsync -av -delete を使うことで同期できるようにした。しかし、そのままでは最短でも一分毎に同期が最短となっているため、同期の時間を短くする必要があった。そこで、for 文と sleep、seqなどを組み合わせることによって毎秒同期することのできる設定にした。

(※文責: 渡邊建)

4.5 最終発表

最終発表は、スライドとアプリケーションのデモをメインに発表した。酒田以外の 6 人がプレゼンテーションを行った。ポスターはプロジェクトの活動をまとめたメインポスターと、Umistagram の使い方を説明するサブポスターの 2 種類を作製した。

メインポスターは酒田が作成した。後期の活動は、特にアプリケーション開発に力を入れていたため、開発について詳しく書くことにした。また、Umistagram は VR アプリケーション、フィルタアプリケーション、Web サイトの 3 つの工程に分かれており、使用するのが難しいため、Umistagram の使用方法についても書くことにした。しかし、メインポスター 1 枚に開発とアプリケーションの説明の両方を書くにはスペースが足りないと判断したため、サブポスターを作成することにした。そのため、メインポスターは特におこなった活動について詳細に書いた。

サブポスターは長谷川が作成した。こちらは Umistagram の使用方法をわかりやすく書く必要があるため、3 工程の流れがわかりやすいようなレイアウトで作成した。サブポスターのレイアウトは自由であったため、ポスターを横向きにし、矢印によって工程の流れを表現した。

スライドは今泉が作成した。デモの時間を長くとるために、可能な限り文章を少なく、画像を多

Underwater World

くし、短い言葉で説明しやすいようなスライドにした。原稿も今泉が作成した。専門用語を少なくすることで、用語の説明にとられる時間をなくした。

発表評価シートは工藤が作成した。後期には Umistagram という成果物があったため、その評価を行う必要もあると考えた。そのため、評価シートに Umistagram の完成度についての質問を設けた。

(※文責: 小橋元春)

第 5 章 結果

5.1 成果

H30 年度のプロジェクト学習において、前期の活動で目標を何にするか話し合った結果、海の興味のない若者に海に興味を持ってもらうきっかけを作るために「新しい海の楽しみ方」を提案することを目標にした。この目標での問題点を解決するため、VR を使用した海中体験アプリケーションの制作に必要な海中の映像を撮影することを考案した。そこで、プロジェクトの皆で意見を出し合って決めた 3 種類の場所で水中カメラを用いた撮影をした。場所は入舟町前浜海水浴場、函館市国際水産・海洋総合研究センター、黒岩岬の 3 か所である。

その際に撮影した海中の映像を後期で VR アプリケーションの制作に使用することを考案し、プロジェクトの皆で意見を出し合って決めた 3 種類の班でアプリケーションの開発をした。小橋、松本、工藤から成る Unity 班は VR 空間の作成を行うため、小橋は VR 空間内で写真撮影機能や魚に触れる機能の搭載を行い、松本は魚のオブジェクトを作成し、工藤はスマートフォン型オブジェクトと泡のオブジェクトを作成した。

今泉、酒田、長谷川から成る画像処理班は動画から抜き出した背景画像の輝度調整およびフィルタの作成を行うため、今泉は python を用いて画像の輝度差を調整する作業を行い、酒田と長谷川は wxpython を用いて GUI アプリケーションの開発を行った。その中で酒田はレイアウトやボタンの設定を、長谷川はレイアウトや画像を加工する際の設定を担当した。

渡邊から成る Web サーバ班はファイル共有設定および Web サイトの作成を行うため、WSL でサーバに SSH 接続してウェブページを公開し、ディレクトリ内で取得した画像を表示させた。

(※文責: 工藤祐弥)

5.2 解決手順と評価

5.2.1 撮影

前期では、海中を撮影することによって、後期の活動で使用できる素材を作り出すことができた。また、普段使わないような水中カメラを使ったことにより、活動する側も海の魅力を感じ取ることもできた。表 5.1 に、海中を撮影することになった 3 か所の場所における長所と短所を示す。

1 か所目である入舟町前浜海水浴場で水中カメラを用いた撮影をした結果、水の透明度が高いことが判明したため綺麗な水中の映像を撮影することができ、魚を 1 匹だけ確認することができた。しかし、撮影できる場所が浅瀬のみであり、深い海の中を表現するには適さない映像しか撮ることができなかった。また、足場には岩が多く、体のバランスを保ちながら撮影するには困難な場所だった。

これを踏まえ、2 か所目の撮影場所を函館市国際水産・海洋総合研究センターに変更して撮影した際には、水深が深かったことにより使用可能な機材は限られていたものの、範囲の広い水中の映像を撮ることができたうえに魚を多く確認できたため、プロジェクトの目標を果たすことに役立つ可能性があったものの、魚の種類を判別することはできなかった。そのうえ、1 か所目と比べると

Underwater World

水質が悪く、全体的に緑色となっている水中の映像が撮れたため、この映像を仮に使用するには特殊な技術を用いて綺麗な色に変換する必要があるといった意見があった。

3か所目の撮影場所である黒岩岬で行った撮影では、水の透明度が高く天気が良かったため質の高い映像を撮影できることが期待されたが、実際は風が強く波が高かったため立っただけで転倒しそうな状況だった。このときに撮影できた映像は泡が多く、魚が見当たらなかったため、VRアプリケーションに使用するには適さないものとなった。

これらの結果を踏まえ、一部のメンバーによって入舟町前浜海水浴場で再度撮影を行うことになった。この際に撮影した映像が一番優れていると判断されたため、VRアプリケーションに使用することに決定した。

表 5.1 撮影場所の長所・短所

撮影場所	長所	短所・問題点
入船町前浜海水浴場 撮影時間 7分 59秒	水の透明度が高い	撮影できる場所が浅瀬のみであった
函館市国際水産・ 海洋総合研究センター 撮影時間 22分 11秒	水深が深く・魚を多く確認できた	水質が悪く不透明
黒岩岬 撮影時間 22分 3秒	水の透明度が高い	強風・高波のため撮影が困難であった

4KVRは2つのレンズの映像をつなぎ合わせているので、両方に等しく光が当たらないと輝度差が生じてしまうことが判明した。GoProを用いた場合、鎖がカメラに映ってしまうことが判明したが、これを釣り糸で代用すると糸が切れてカメラを紛失する危険があるため、安全面を考慮して鎖がカメラに映ったまま撮影を行った。



図 5.1 入船町前浜海水浴場 (7分 59秒)



図 5.2 函館市国際水産・海洋総合研究センター (22分 11秒)



図 5.3 黒岩岬 (22分 3秒)

中間発表においては、発表を聞いてくれた人々に発表技術と発表内容をそれぞれ10点満点で評価してもらった結果、発表技術の平均点が6.61点で、発表内容の平均点が6.85点だった。発表技術ではコンセプトが具体的で分かりやすかったといった評価もあったが、声が小さくて聞き取れな

Underwater World

いといった評価もあった。また、発表内容では VR を使用するアイデアを高評価する意見があったが、撮影に使う機材が多かったため説明が長くなっていると指摘しているものもあったため、後期では再考する必要があった。

後期では、班ごとに分かれて作業することによって、効率の良い役割分担をすることができた。また、現在において最も影響力のある SNS から着想を得ることにより、若者に海の魅力を知ってもらうための手がかりができた。

しかし、ファイル共有を行う場合にはパソコンとサーバを用いることによって機能する状態だったため、データを転送するプロセスを円滑に行うにはこれらの方法を統一する必要があることが分かった。また、撮影した動画は波の影響で大きく揺れており、酔いの原因となるため VR アプリケーションに使用できず、発表で VR アプリケーションを体験させる際には静止画を使用することになった。そのため、カメラが揺れないような撮影方法を確立する必要がある。

最終発表においては、中間発表と同様の発表技術と発表内容に加えて Umistagram についてもそれぞれ 10 点満点で評価してもらった結果、発表技術の平均点が 7.00 点で、発表内容の平均点が 7.81 点で、Umistagram の平均点が 7.93 点だった。発表技術においては目的の分かりやすさを評価する意見があったが、中間発表において指摘された声の小ささに関してはあまり改善が見られなかったうえ、原稿を読んでいて正面を向いてなかったという指摘もあり、練習不足であったことが分かった。発表内容および Umistagram においては最新技術を駆使していることで海の魅力を分かりやすく伝えようとしていることが斬新で良く、映像にリアルな印象があってアプリとして完成度が高く出来ていると述べている意見が比較的多く、全体的に Umistagram の機能を高く評価する意見が大半だったが、VR 空間内でもフィルタをかけたほうがよいと述べている意見もあり、改善するべきところもいくつか指摘された。

(※文責: 工藤祐弥)

第 6 章 まとめ

6.1. プロジェクトの成果

私たちは、海への親しみを感じていない若者を対象に新しい海の楽しみ方を提案することを目的として海中を体験、インスタ映えできるアプリケーション Umistagram の開発を行った。その中でプロジェクトの成果として挙げられるものは主に、Umistagram の開発のために函館市内 3 か所で 4 種類の機材を使用して撮影した動画、制作物である Umistagram、Umistagram を開発する中で得た各々の習得技術の 3 つある。

(※文責: 長谷川涼乃)

撮影動画

函館市内の前浜海水浴場、黒岩岬、函館市国際水産・海洋センター前の 3 か所で動画撮影し、動画を収めた。動画撮影では水中ドローンの Power Ray、GoPro、SP360、VR360 の 4 種類の撮影機材を使用した。Power Ray では、深く遠い水中動画を GoPro では、画質の良い浅瀬のきれいな水中動画を、SP360 と VR360 では、360°の海中動画を収めることができた。GoPro、SP360 の 2 つの機材では、魚の姿を収めた動画を取ることに成功した。また、VR360 を用いて、成果物に実際に使用した動画を収めた。

(※文責: 長谷川涼乃)

制作物

制作物として海中を体験、インスタ映えできるアプリケーション Umistagram の開発を行った。Umistagram は、VR 空間内で海中を体験したり、写真を撮影するなどして海中を楽しむその後、GUI アプリケーションを使用し、VR 空間内で撮影した写真を加工・保存することで Umistagram のサイトから自分のスマートフォンに写真を保存することができるもので、海への親しみを感じていない若者を対象に新しい海の楽しみ方を提案することを目的として開発した。

(※文責: 長谷川涼乃)

習得技術

開発では、Unity 班、画像処理班、Web サーバ班の 3 つに分かれて行った。Unity 班では、Unity の使い方や C # のコーディングから VR アプリケーションを開発したことで Unity による VR 技術が身についた。画像処理班では、画像の輝度差を調節することで画像処理の知識やガンマ補正の技術が身についた。また、フィルタアプリケーションの作成では wxpython の知識や、python のコーディング力などといったことから様々なフィルタを作成することで、直線検出するハフ変換や色反転や三値化、ノイズ除去や領域分割などといった様々な画像処理に関する知識が身についた。Web サーバ班では、サーバを立てるために習得した apache の知識や画像を同期するために使用した shellscript や crontab などといったサーバ周りの技術を習得した。Web サイト作成では、PHP

の glob 関数などといった Web サイト作成に必要な技術を習得した。

(※文責: 長谷川涼乃)

6.2. プロジェクトにおける各人の役割

小橋 :

プロジェクトリーダーとして、スケジュール管理や意見のまとめ、プロジェクトメンバーへの仕事の割り振りなど他のメンバーを牽引していた。動画撮影では、SP360、VR360 を担当し実際に、魚群の動画を収めた。また、実際に Umistagram の開発に用いた動画を撮影した。開発では Unity 班に所属し、リーダーとして Unity 班のメンバーへの指示や助言などといったことから、Unity 内全般のアルゴリズムや機能、仕組みの開発を担当していた。Unity 班で全天球動画を観賞できるようにしたり、Oculus Rift で VR 空間空間に干渉できるようにしたりするなどといった基盤的なものから VR 空間内の魚の群れ単位のアルゴリズムや VR 空間内での写真の撮影、魚に触れる機能の実装やオプションによる環境の設定などの開発を行った。

(※文責: 長谷川涼乃)

今泉 :

通常のプロジェクト活動時では、プロジェクトリーダーを支える形で物品の管理などといったことなど様々なこととともにプロジェクトを牽引していた。動画撮影では、SP360、VR360 を担当し、魚群の動画を収めた。また、実際に Umistagram の開発に用いた動画を撮影した。開発では画像処理班に所属し、リーダーの役割を務めるとともに VR に使う画像を撮影するために使用した VR360 はレンズが 2 つあるため、2 つのレンズで輝度差が生じてしまう問題を python を用いて解決した。また、画像処理班のリーダーとして、他のメンバーのサポートや進捗確認を行っていた。最終発表では、原稿と発表スライドの作成を行った。

(※文責: 長谷川涼乃)

酒田 :

動画撮影では、Power Ray、GoPro を担当し、海に入って撮影することで水中のきれいな動画や魚の撮影に成功した。また、開発では、画像処理班に所属し、wxpython を用いて GUI アプリケーションの開発を行っていた。主に、GUI の機能（レイアウト、ボタンと関数、削除、加工前へのロールバックなど）の作成を行った。また、Web サーバ班との同期方法を渡邊と共に Dropbox によって実装させた。中間発表、最終発表ともに、メインポスターの作成を行った。また、発表で使用するために自分と PC を使用できるようにアプリケーション化や同期の設定を行った。

(※文責: 長谷川涼乃)

長谷川 :

動画撮影では、Power Ray、GoPro を担当し、海に入って撮影することで水中のきれいな動画や魚の撮影に成功した。また、開発では、画像処理班に所属し、python と opencv を用いて 1 2 種類の加工フィルタと回転機能を作成した、また、wxpython で GUI の機能を作成した（レイア

Underwater World

ウトや画像の回転や回転させた際の画像の縮尺、加工後の画像表示など)。また、高校生を対象とした発表ではスライドと原稿の作成と発表を一人で担った。中間発表ではスライドの作成、高校生への発表では発表スライドと原稿の作成と発表、最終発表では、サブポスターの作成を行った。

(※文責: 長谷川涼乃)

松本 :

動画撮影では、SP360、VR360 を担当し、魚群の動画を収めた。また、開発では、Unity 班に所属し、知識がない状態から始めた Unity での魚のモデルの作成で、情報ライブラリーなどを活用しながら、Unity を学習していった。最終的にはプレハブの生成と関数が上手く使えるようになった。Asset Store で手に入れた魚のモデルそれぞれを、配列にして呼び出せるような設定や、ランダムな魚のモデルを返す関数を作成した。また、ユーザが魚に触れやすいように、ユーザの近辺に 5 体の魚が出現するように Instantiate を使って作成したり、遠方に魚が泳いでる様子を見せるために、15 体の魚が出現するようにしたりなど没入感を高める機能を作成した。

(※文責: 長谷川涼乃)

渡邊 :

動画撮影では、Power Ray、GoPro を担当し、海に入って撮影することで水中のきれいな動画や魚の撮影に成功した。また、開発では、Web サーバ班を一人で担当し、サーバの構築、Web サイトの作成、画像の同期機能の作成を行っていた。WSL で、借りたサーバに SSH 接続し、apache を使うことでウェブページを外部に公開できるようにした。同期機能では、Dropbox とサーバのディレクトリを、shellscript と crontab を用いて毎秒同期するプログラムを組むことで、画像をサーバに自動で転送させ、ウェブページ表示させた。Web サイト作成では、PHP の glob 関数と for 文を用いて、ディレクトリ内にある画像を取得し、枚数文だけループするコードを書くことで、ウェブページにディレクトリ内にある画像をすべて表示させた。

(※文責: 長谷川涼乃)

工藤 :

動画撮影では、Power Ray、GoPro を担当し、海に入って撮影することで水中のきれいな動画や魚の撮影に成功した。また、開発では、Unity 班に所属し、Unity 内の泡やスマートフォンといったオブジェクトの作成を行った。泡の画像を 10 種類作成し、それらを読み込んで動きをつけた。上昇しながら左右に揺れて、一定の高さに達すると自動的に消えるシステムにすることで VR 空間の没入感を高めるようにした。

(※文責: 長谷川涼乃)

6.3. 今後の課題

本プロジェクトの今後の課題として、主にファイル共有方法の統一と動画の撮影方法の確立の 2 つある。ファイル共有方法の統一については、VR アプリケーションとフィルタアプリケーション間には OneDrive、フィルタアプリケーションとサーバ間は Dropbox を用いている為、データを

Underwater World

転送するプロセスを円滑に行うにはこれらの方法を統一する必要がある。撮影方法の確立については、今回撮影した動画は波の影響で大きく揺れており、酔いの原因となるため VR アプリケーションに使用できなかった為、カメラが揺れないような撮影方法を確立する必要がある。上記の2つの課題以外にも、最終発表において発覚した、加工機能の追加・向上、VR 空間内でそのまま撮影した画像を加工する仕組みに改善することや、Web サイトの有無などといった改善点がある。まだ、ファイル共有方法の統一と動画の撮影方法の確立などといった上記の課題の解決方法も明らかになっていない状態であるため、今後さらなる開発が必要となる。

(※文責: 長谷川涼乃)

参考文献

- [1] 日本財団 (2017), ”「海と日本」に関する意識調査”, [online]<https://www.nippon-foundation.or.jp/news/pr/2017/31.html> (参照 2018-7-18)
- [2] 中野裕・大庭昌昭 (2004), ”水泳の好悪に関する要因の分析”, 日本体育学会, 55, p.470
- [3] 荒川巧也・浅野祐一 (2018), ”Unity2018 入門 最新開発環境による簡単 3D&2D ゲーム制作”, SB クリエイティブ
- [4] Craig Reynolds(1987), ”Flocks, Herds, and Schools: A Distributed Behavioral Model”, Computer Graphics, pp.25-34
- [5] 佐藤まり子 (2016), ”1design” , [online]<https://1design.jp/web-development/1154>(参照 2018-11-9)