

公立はこだて未来大学 2020 年度 システム情報科学実習 グループ報告書

Future University-Hakodate 2020 System Information Science Practice
Group Report

プロジェクト名

ビーコン IoT で函館のまちをハックする - Beacon FUN 4

Project Name

Leverage the Beacon IoT for Our Smarter Life in Hakodate Real Downtown - Beacon FUN 4

グループ名

でももに!

Group Name

Demomoni!

プロジェクト番号/Project No.

7-B

プロジェクトリーダー/Project Leader

なし None

プロダクトオーナー/Product Owner

久米田羽月 Uzuki Kumeta

グループメンバ/Group Member

村石拓海 Takumi Muraishi

石澤大輔 Daisuke Ishizawa

宮田悠治 Yuya Miyata

荻ノ沢実佑 Miyu Oginosawa

山本雄平 Yuhei Yamamoto

久米田羽月 Uzuki Kumeta

指導教員

松原克弥 藤野雄一 鈴木昭二 奥野拓 鈴木恵二

Advisor

Katsuya Matsubara Yuichi Fujino Sho'ji Suzuki Taku Okuno Keiji Suzuki

提出日

2021 年 1 月 14 日

Date of Submission

January 14, 2021

概要

函館市には多様な観光資源があり、その魅力を活かす函館市総合計画という取り組みが行われている。本プロジェクトでは、ビーコンの特徴を活かすことで函館市による取り組みの一端を担うようなサービスを考案・開発することを目的としている。函館のまちをよく知らない状態でサービスを提案してしまうと、実際に潜んでいる課題を見逃してしまう可能性がある。そのため、サービスを提案するにあたり、函館のまちの課題を発見するためにフィールドワークを行った。発見した課題をもとに、函館の魅力を引き出せるようなアイデアを創出した。創出したアイデアをよりよくするために様々な観点に着目してブラッシュアップや絞り込みを行った。その結果、3つのサービスを提案することに決定した。

(※文責: 荻ノ沢実佑)

函館市は、歴史観光資源が多く存在しており、現在でも歴史的建造物が数多く残っている。それを目的に国内外問わずたくさんの観光客が函館を訪れている。それにより、観光産業は函館の主要な産業の1つとなっている。しかし、歴史的建造物といった観光地を訪れたいと考える観光客は、比較的年齢層が高めである。そのため家族連れの観光客の場合、子どもが十分に観光を楽しめなかったり、大人が本当は行きたいと考えていた観光地の来訪を妨げたりする可能性がある。これらを踏まえ、私たちのグループでは、大人も子供も函館観光を楽しめるようなサービス「でももに！」を提供することで、家族全員が函館観光を楽しみ思い出を残してもらえるようにする。本サービスは、スマートフォンで利用するアプリケーションである。函館のまちの各所に置いてあるビーコンに接近することで、ピースを入手することができる。これを集め、パズルパネルにはめることで、大きなパズルの一部分として思い出を残すことができる。入手したピースにはサインを書き残すことができる。後期の活動で、サービスの要件定義・設計・開発を行った。

(※文責: 荻ノ沢実佑)

キーワード 函館, IoT, ビーコン, 観光, 家族連れ, パズル, スマートフォン, アプリケーション, 思い出

Abstract

Abstract in English. Hakodate City has a variety of tourism resources, and made "Hakodate City Comprehensive Plan" for use the resources effectively. We aim to devise and develop a service that plays a part in the plan by utilizing the feature of beacons. If we propose a service without knowing the city of Hakodate well, we may miss the issues that are actually lurking. Therefore, for proposing the service, we conducted fieldwork to discover the issues in the town of Hakodate. We created ideas based on the issues we discovered, that could bring out the charm of Hakodate. In order to improve the ideas we created, we focused on various perspectives and refined and narrowed down. As a result, we decided to propose three services.

(※文責: Miyu Oginosawa)

Hakodate City has many historical tourist resources attractions, and many historical buildings still remain today. Many tourists in the world, visit Hakodate for this purpose. As a result, tourism has become one of the major industries in Hakodate. However, tourists who want to visit historical sites are relatively older. Therefore, in the case of tourists with families, children cannot fully enjoy sightseeing. Or parents may give priority to their children, which may prevent them from visiting tourist spots that they inwardly wanted to visit. In light of these issues, our group has developed a service that allows both adults and children to enjoy sightseeing in Hakodate, called "Demomoni!" so that both adults and children can enjoy sightseeing in Hakodate and leave with memories of their visit. The service is an application for use on smart phones. Users can obtain pieces by approaching the beacons placed at various locations in Hakodate. By collecting the pieces and fitting them into the puzzle panels, users can preserve their memories as part of a larger puzzle. Users can sign on the pieces they get. In the second semester of the project, we defined, designed, and developed the requirements for the service.

(※文責: Miyu Oginosawa)

Keyword Hakodate, IoT, beacon, sightseeing, family, puzzle, smartphone, application, memories

目次

第 1 章	本プロジェクトの活動と目的	1
1.1	背景	1
1.2	目的	1
1.3	ビーコンについて	2
1.4	本プロジェクトの活動の進め方	2
	1.4.1 本プロジェクトにおける開発着手前の手法	2
	1.4.2 本プロジェクトにおける開発手法	3
1.5	本プロジェクトのロゴについて	5
	1.5.1 ロゴ作成	5
第 2 章	提供サービスの考案プロセス	7
2.1	プロセス概要	7
2.2	フィールドワーク	7
	2.2.1 事前調査	7
	2.2.2 フィールドワークに関するレクチャー	7
	2.2.3 実地調査	8
	2.2.4 振り返り	8
2.3	サービスの考案	9
	2.3.1 アイデア考案に用いた手法	9
	2.3.2 中間発表時の外部レビュー	10
	2.3.3 プロジェクトでの内部レビュー	11
	2.3.4 アイデアのブラッシュアップ	12
	2.3.5 テーマの決定	12
2.4	その他	13
	2.4.1 ビーコンについての事前調査	13
第 3 章	提供するサービスについて	15
3.1	背景と課題	15
3.2	サービスの概要	15
	3.2.1 ユーザーストーリー	15
3.3	機能	17
	3.3.1 機能一覧	17
	3.3.2 チュートリアル機能	18
	3.3.3 ピース探索機能	19
	3.3.4 ピースの取得機能	19

3.3.5	ピース一覧機能	20
3.3.6	パネル状況閲覧機能	20
3.3.7	ピース情報閲覧機能	20
3.3.8	なぞなぞ一覧機能	21
3.3.9	サイン書き込み機能	23
3.3.10	ピースアップロード機能	23
3.3.11	パネルの機能	24
3.4	アイデアの発案から最終的なサービス内容決定までの経緯	27
第 4 章	開発	28
4.1	開発で用いたコミュニケーションツール	28
4.1.1	Slack	28
4.1.2	Discord	28
4.1.3	Zoom	28
4.2	開発環境整備	29
4.2.1	Git	29
4.2.2	GitHub	29
4.3	活動記録管理ツール	29
4.3.1	esa	29
4.3.2	YouTube	30
4.4	スクラム関連ツール	30
4.4.1	Google Jamboard	30
4.4.2	Miro	30
4.4.3	hatjitsu	30
4.5	デザインツール	31
4.5.1	Adobe XD	31
4.5.2	Adobe Illustrator	31
4.5.3	Adobe Photoshop	31
4.5.4	Adobe Premiere Pro	32
4.5.5	CLIP STUDIO PAINT	32
4.6	技術習得	32
4.6.1	夏季休暇期間中の勉強会	32
4.7	開発手法	33
4.7.1	開発の流れ	33
4.7.2	効果	34
4.7.3	課題	34
4.7.4	解決策	35
第 5 章	サービスの実装	37
5.1	システム構成	37

5.2	アプリケーション	37
5.2.1	iOS	37
5.2.2	Android	47
5.2.3	Web アプリケーション	52
5.3	デザイン	54
5.3.1	アプリケーションアイコン	54
5.3.2	テーマカラー	55
5.3.3	フォント	56
5.3.4	ピース探索, 取得画面	57
5.3.5	ピース一覧画面	58
5.3.6	なぞなぞ一覧画面	58
5.3.7	ピースサイン画面	58
第 6 章	成果報告会	59
6.1	中間発表	59
6.1.1	発表形式	59
6.1.2	発表技術の評価と反省	61
6.1.3	発表内容の評価と反省	61
6.2	成果発表	62
6.2.1	発表形式	62
6.2.2	発表技術の評価と反省	65
6.2.3	発表内容の評価と反省	66
第 7 章	各メンバーの振り返り	68
7.1	役割分担	68
7.2	久米田羽月の振り返り	69
7.3	萩ノ沢実佑の振り返り	70
7.4	村石拓海の振り返り	71
7.5	山本雄平の振り返り	72
7.6	宮田悠冶の振り返り	73
7.7	石澤大輔の振り返り	74
第 8 章	まとめと展望	76
8.1	前期の振り返り	76
8.2	後期の振り返り	77
8.3	今後の展望	78
	参考文献	80
	付録 A 中間発表で使ったプロジェクト概要のポスター	82

付録 B	成果発表で使⽤したプロジェクト概要のポスター	83
付録 C	オープンソースライセンス	84

第 1 章 本プロジェクトの活動と目的

1.1 背景

近年、通信技術の向上や通信機器の小型化・低コスト化のため IoT が普及している。[1]IoT デバイスの 1 つとして BLE ビーコンがある。BLE ビーコンとは、低消費電力の近距離無線技術「Bluetooth Low Energy」(BLE) を利用したデバイスのことである。スマートフォンなど BLE 対応デバイスを所持していれば、誰でも利用可能である。BLE ビーコンを利用したサービスとして、人やモノの位置情報を可視化する屋内位置情報サービス「Beacapp」や、自動販売機にスマートフォンを接続してドリンクを購入することでお得にドリンクを入手することができる「Coke On」などが存在する [2][3]。この他にも様々なサービスやアプリケーションが創出されている。そして、今後もその数は増加していくと考えられる。それに伴い、需要も高まっていくと期待できる。そこで、函館のまちの様々な場所にビーコンを配置することで、函館の魅力を伝える新しいサービスの提供が可能になる。

函館市には、自然、歴史、文化、食など幅広い観光資源が存在する。自然については、函館山からの夜景や、活火山の恵山などの雄大な自然の景色を楽しめる。歴史、文化については、五稜郭跡や軍事要塞跡をはじめ旧函館区公会堂など多くの歴史的建造物が存在するため、幕末から明治、大正と様々な時代の歴史を知ることができる。食については、函館市は海に囲まれているため、豊かな漁場から獲れる海産物を使った海鮮を食べることができる。また、函館朝市や地元の食材を使用したレストランなども存在する [4]。このように函館市は観光資源にあふれた観光都市であるといえる。この多様な観光資源をより多くの人に伝えるためにも、多角的に魅力を発信し、磨きをかけていく必要がある [5]。その方法の 1 つとしてビーコンを活用する。函館の様々な観光資源とビーコンを組み合わせることによって、函館の魅力をさらに引き出すサービス展開ができる。

(※文責: 荻ノ沢実佑)

1.2 目的

本プロジェクトの目的は、ビーコンを用いて函館のまちの様々な魅力や新たな価値を生み出すことである。ビーコンの特徴を活かしつつ、函館の魅力をさらに引き出せるようなサービスを考案する。

(※文責: 荻ノ沢実佑)

1.3 ビーコンについて

BLE ビーコンとは、BLE に準拠した Bluetooth の電波を発信する機器であり、端末ごとに固有の情報を保持している [6]。以後、本報告書で述べるビーコンは、BLE ビーコンを指す。スマートフォンなどの電子端末でビーコンの電波を受信・検知することにより、固有 ID や受信電波強度などの情報を取得することが可能になる。BLE ビーコンには Apple 社が提供する iBeacon、LINE 社が提供する LINE Beacon など様々な規格がある。iBeacon は、固有 ID や信号強度などの情報を持っている [7]。その情報を利用することで、距離測定や領域内の入出の監視などをを行うことができる。LINE Beacon は、コミュニケーションアプリ LINE がインストールされているスマートフォンに対して、広告やプッシュ通知を送信することができる [8]。

(※文責: 荻ノ沢実佑)

1.4 本プロジェクトの活動の進め方

1.4.1 本プロジェクトにおける開発着手前の手法

本プロジェクトでは、プロジェクトを円滑に進めるべく、開発着手前から独自の試みを多く行っていた。以降では、行った試みについて、それぞれ説明する。

(※文責: 大石晃平)

週ごとのファシリテーター制度

本プロジェクトではプロジェクトリーダーを決めなかった。今年度は前期の活動が全面オンラインであったため、当初はプロジェクトメンバーのことをお互いによく知らなかった。そのような状態で、本プロジェクトの責任を負うリーダーを決めるのは難しいと考えた。したがって、本プロジェクトでは、前期の活動を通じて互いのことを理解してから、プロジェクトリーダーを定めるかどうかを判断することに決定した。ただし、各週のプロジェクト活動の進行を担う人は決める必要があった。そのため、毎週最後のプロジェクト活動時に、次週の進行を担うファシリテーターを 2 人決定した。ファシリテーターは、公平性を保つためあみだくじなどを用いてランダムで決定していた。この決定方法には、全員に等しくプロジェクトをファシリテートする経験をさせることなどの狙いがあった。

しかし、ファシリテーターの役割が明確ではなかったため、仕事量が多くなり、メンバーがファシリテーターになることに抵抗感を覚えた。加えて、週ごとにファシリテーターが変わってしまうため、自分の担当した週以外ではあまりファシリテーターの活動に責任を持って活動ができていなかった。そのため、前週に行った活動内容の引き継ぎがされず、活動が右往左往してしまった。

これらの問題点から、開発着手後は、様々なことに担当大臣を任命し、その担当大臣が中心となって、任命された職務を進めていく「担当大臣制度」という独自の手法を取り入れた。ファシリテーターの役割は、「進行担当大臣」が担い、毎週のプロジェクト活動のアジェンダ作りや進行を

行った。上記の反省点を活かして、全体的にプロジェクト全体に対しての関心が深まり、全員がより責任を持って活動することができた。

(※文責: 大石晃平)

疑似的バックログの作成と全員での意思決定

「週ごとのファシリテーター制度」で述べたように、本プロジェクトでは週ごとのファシリテーター制度を取り入れていたため、全体の活動を統括するような役割を担う人が不在の状態であった。そのため、プロジェクトでの活動の見通しを全体で共有して持つことができるように、全員で疑似的バックログを作成した。週ごとにファシリテーターが変わるため、スケジュールを細かく決めても、その通りに物事が進む可能性が低い。したがって、週ごとで柔軟に対応可能にさせるため、バックログを採択した。また、なにかを決定する際には、極力全員での意思決定をして、活動を行った。全員での意思決定を大切にしたい理由には、プロジェクトリーダーの決めていないチームにおいて、全員の意思での決定を行えば不平不満が出さずに活動できるのではないかという狙いがあった。

疑似的バックログは、作成する際に全員で行い、活動の見通しを持つという意味では活かすことはできたが、それ以上は活用することはできなかった。そのため、すべての工程を計画し、その工程に沿ってプロジェクトを進めていってしまうこともあった。結果的に、疑似的バックログを作った理由である、柔軟に活動することができていない部分もあった。また、疑似的バックログでは、それぞれタスクにどのくらいの期間を有するかやそれに向けての心理的な問題点がどれくらいあるのかを数字などで表す、重みづけを行った。しかし、その重みづけもただけになってしまい、それぞれのタスクに対して振り返りをしなかった。そのため、その重みづけが正しいのかを判断することもできずに進んでしまった。全員での意思決定は、意識的に行っており、全体的にできていたため、全員が活動に対しての納得感や満足感があった。しかし、一回一回の意思決定で時間がかかってしまい、円滑に活動を進めることはできなかった。

これらの反省点を活かして、開発着手してからはそれぞれのサービスごとにバックログ作りに工夫を凝らし、より円滑に開発を進めることができた。また、意思決定の方法も「d」という言葉をチャットや体で表現をすれば、その意見に対して合意であるなどの独自の決定方法を作成することによって、円滑に進めることができた。

(※文責: 大石晃平)

1.4.2 本プロジェクトにおける開発手法

アジャイル開発

本プロジェクトでは、ソフトウェア開発の手法の1つである、アジャイル開発を導入した。アジャイル開発とは、プロセスやツールよりも個人と対話を、包括的なドキュメントよりも動くソフトウェアを、契約交渉よりも顧客との協調を、計画に従うことよりも変化への対応を価値とした開発手法である [9]。小さな単位での計画・設計・実装・テストを繰り返すことにより、継続的に動くソフトウェアを提供し続けることができる。また、できるだけ短い期間で上記のプロセスを繰り返

し、関係者からのフィードバックを継続的に得ることで、トラブルや仕様変更に対応することができる。これらのことから、開発初期から継続して価値のあるソフトウェアを作り続けることができると考え、この手法を採用した。

(※文責: 袴田結女)

スクラム

本プロジェクトでは、アジャイル開発を導入するにあたり、アジャイル開発の手法の1つであるスクラムを採用した。Ken Schwaber によると、スクラムとは「複雑な問題に対応する適応型のソリューションを通じて、人々、チーム、組織が価値を生み出すための軽量級フレームワークである」と定義されている [10]。スクラムでは、スクラムチームと呼ばれる、小さなチームを構成することが定められている。スクラムチームは、プロダクトオーナー1人、スクラムマスター1人、複数人の開発チームから構成される。プロダクトオーナーとは、プロダクトの責任者であり、プロダクトの価値を最大化する責任がある。後述するプロダクトバックログの管理の責任者でもある。スクラムマスターは、スクラムというフレームワークを円滑にまわすことに責任を持つ。スクラムのルールをスクラムチームに理解させることや、スクラムチームを妨害から守ること、スクラムチームが円滑に作業を進められるよう支援するというような役割をもつ。開発チームは、プロダクトの開発を行う役割であり、通常3人から9人までで構成される。開発チームは、機能横断的であり、プロダクトを作るために必要な作業をすべて行うことができなければならない。「コードを書く」や「UI/UX の設計を得意とする」というように、開発チームの各メンバーが所有する能力やスキルで、開発に必要なすべての作業を行うために協力し合う。また、開発チームにおける作業は、外部からの指示を受けることはなく、チーム全体が責任を持ち、主体的に行う。

スクラムでは、実現したい機能を、価値やリスク、必要性などを基準に並べ替えたプロダクトバックログと呼ばれるリストを作成する。プロダクトバックログ項目の作成や更新、並び順はプロダクトオーナーが最終決定権を持つ。プロダクトバックログの各項目は、ユーザーストーリー形式で書かれることが多い。ユーザーストーリーとは、実際に使う人たちの視点に立って、実現したいことを簡潔に記述したものである。プロダクトバックログを作成することによって、実現すべき機能の順番が明確化する。この順番が上位の機能から開発に取り掛かることで、成果を最大化することができる。

また、スクラムでは最長1ヶ月までの固定の期間に区切って、開発を繰り返す。この固定の期間をスプリントと呼ぶ。この期間内に、小単位での計画・設計・実装・テストを行い、プロダクトバックログの機能を完成させる。これを繰り返すことで、段階的に、そして継続的に動くソフトウェアを提供し続けることができる。スプリント内で機能の実装を目指すために、スプリントプランニング、デイリースクラム、スプリントレビュー、スプリントレトロスペクティブの4つのスクラムイベントを行う。

スプリントプランニングとは、スプリントを開始するにあたって行われる、スプリントで実行する作業を計画するミーティングである。このミーティングでは、「そのスプリントになぜ価値があるのか (why)」、「そのスプリントで何をするのか (what)」、「決めた作業をどのように成し遂げるか (how)」の3つのトピックについて話し合い、決定する。1つ目のトピックでは、スプリント

ゴールと呼ばれる今回のスプリントの目標を決定する。2つ目と3つ目のトピックでは、プロダクトバックログから今回のスプリントで開発できそうな機能を検討し、スプリントバックログを作成する。スプリントバックログとは、選択した実装したい機能と、そのための具体的な作業と計画のことである。次に、デイリースクラムとは、スプリント期間中、毎日同じ場所・同じ時間で行う15分のミーティングのことである。開発メンバー全員で「昨日何をしたか」、「今日何をするか」、「困っていること」を共有し、スプリントゴールが達成できるかを確認するためのものである。また、スプリントレビューとは、スプリント終了時にスプリントの成果をステークホルダーに披露したり、プロダクトの状況や進捗を報告したりするイベントである。ステークホルダーからのフィードバックに基づき、プロダクトバックログを見直し、今後やるべきことに取り組むことができるようになる。最後に、スプリントレトロスペクティブとは、今回のスプリントにおける活動を振り返り、次回のスプリント以降のアクションプランを考えるミーティングである。うまくいったことや問題点を話し合い、もっと成果が出せるよう改善を続けることで、スクラムチームの成長が期待される。

このように、ユーザーからのフィードバックをもとに、柔軟に計画を変更しながらより良い機能を実装し続けることができると考え、本プロジェクトではスクラムを採用した。

(※文責: 袴田結女)

1.5 本プロジェクトのロゴについて

1.5.1 ロゴ作成

本プロジェクトでは、プロジェクトの特徴やイメージを表現するため、加えて議論の練習と、メンバー全員の一体感を生むために、今年度のプロジェクトのロゴを制作した。ロゴの制作期間は3週間であった。まずはじめに、メンバー全員がロゴ案を1案以上考案してロゴ案の発表会を行い、それぞれの案についてレビューを行った。この段階ではそれぞれの案の良い点だけを伝える形式をとった。良い点のみを伝える目的として、作成者本人のモチベーションの維持、さらには他のロゴ案の良い点を各自のロゴ案に取り入れやすくするといった2つの目的があった。ここで受けたレビューと、他のメンバーのロゴ案を参考に、各自のロゴ案の改善をし、2回目の発表会を行った。この発表会のレビューでは、良い点に加えて改善点の指摘を含めたレビューを行い、その後投票によって4つのロゴ案に絞り込んだ。投票は「サービスの函館らしさ」、「ビーコンらしさ」の2つの観点を持って行われた。これらの観点は、函館のまちをコンセプトにしたサービスの開発をする宣言として、他のプロジェクトとの差別化を行うものとして設けた。発表会で得られたレビューをもとに、再度ロゴ案の改善を行い3回目の発表会を行った。この発表会后、前回と同じ観点を持ち、メンバー、TA、先生を交えて最終投票を行って今年度のロゴを決定した。その後、ロゴに関するワーキンググループを結成し、デザインの微調整や、色の確定を行うために、デザイン原案の改善を行った。最後に、ロゴデザインの完成版と、ロゴのカラーコードやフォント、フォントサイズ、使用上の注意点を記載した、ロゴ使用に関するガイドラインを作成した。

(※文責: 伊藤直樹)

Beacon FUN 4 Logo Manual

Color ver.



textsize : 18
font : Freude

- width : 10
RGB : 00, 3f, 8e
CMYK : 100, 55, 0, 44
- width : 8
RGB : 66, cc, ff
CMYK : 60, 19, 0, 0

Black and White ver.



- width : 10
RGB : 00, 00, 00
CMYK : 100, 100, 100, 0
- width : 8
RGB : 99, 99, 99
CMYK : 0, 0, 0, 40

Black Back ver.



- RGB : 00, 00, 00
CMYK : 100, 100, 100, 0
- RGB : ff, ff, ff
CMYK : 100, 100, 100, 100

図 1.1 ログガイドライン

第 2 章 提供サービスの考案プロセス

2.1 プロセス概要

本プロジェクトの提供サービスの決定までのプロセスは、大きく 2 つに分かれる。1 つ目は、フィールドワークに関わるプロセスである。函館やその周辺地域で実際にフィールドワークを行い、課題や気づきを発見・共有した。2 つ目は、サービスの考案に関わるプロセスである。フィールドワークでの結果をもとに、アイデアを考案した。考案したアイデアを 8 つに絞り込み、中間発表でレビューを実施した。そこで受けたレビューをもとに、再度アイデア出しを行った。考案したアイデアについて、プロジェクトメンバー、担当教員、OG・OB に向けてプレゼンを行った。そこで得られた評価をもとに、提供する 3 サービスを決定した。上記のプロセスのほかに、ビーコンの事前学習や、グループ C では、はこだて自由市場でのインタビューを行った。ここからは、フィールドワーク、サービスの考案、その他の 3 つに分けて、詳細を説明する。

(※文責: 袴田結女)

2.2 フィールドワーク

2.2.1 事前調査

函館やその周辺の地域に存在する問題や課題を把握することが目的であるフィールドワークを行うために、それらについての事前調査を行った。はじめに、フィールドワークを行う函館の地域を「五稜郭」、「赤川・美原・昭和・富岡」、「函館駅・西部地区」、「七飯・北斗」、「空港・湯の川」の 5 つに分けた。地域ごとにグループを分け、調査を通して得られた地域のイメージを、ブレインストーミングを行って書き出した。そして、書き出した各イメージについて様々な視点からグルーピングを行い、実地調査を行う際に注目する点を確認した。その後、各地域の実地調査を行うグループを編成し、実地調査を行った。COVID-19 の感染拡大を抑制するために、3 人 1 グループとして、人数を抑えてグループを編成した。

(※文責: 石澤大輔)

2.2.2 フィールドワークに関するレクチャー

フィールドワークの実施にあたり、40 分ほどの動画にて南部美砂子先生よりフィールドワークについてのレクチャーを受けた。レクチャーでは、まずフィールドワークとはどのようなものなのかという説明があった。ただ街を歩いて見たり聞いたりするだけではなく、情報を得て整理し、考えて結論を出し、他者に伝達して行動することが大事であると学んだ。次に、COVID-19 の影響の下でのフィールドワークについての説明があった。COVID-19 の影響でフィールドに行けないと

ということではなく、この COVID-19 が蔓延しているという環境そのものがフィールドであると学んだ。また、フィールドワークは目的ではなく手段の一つであり、フィールドワークの目的の一つは他者の合理性の理解であるということも学んだ。最後に、フィールドワーカーとしてのあり方についての説明があった。調査者は基本的に特権的存在であり、迷惑な侵入者という意識を持って調査を行うべきであるということや、守秘義務や匿名性、調査の説明や同意などに注意するべきであるということも学んだ。

(※文責: 石澤大輔)

2.2.3 実地調査

実地調査は、プレ調査と本調査の2回実施した。プレ調査は、本調査に向けて調査情報の記録、共有の方法について確認することを目的として、6月19日に函館山で実施した。本調査は、「2.2.1 事前調査」で述べた5つの地区で5つのグループに分かれて6月21日に実施した。プレ調査の振り返りを行い、本調査での情報の記録は、メモだけでなく、リアルタイムで共有を行うことで、ほかの地域ではどのような発見があったのかを確認することができることから、Slackに各グループのチャンネルを作成し、写真や文章を貼ることとした。また、調査実施にあたり、COVID-19感染への対策として体調の管理やマスク着用必須、人混みや他人との接触・会話を避けるなど注意することとした。調査場所に関しては、我々のプロジェクトが函館の観光や街の課題を探索し、見つけた課題に対して、ビーコンを用いて解決していくということを目的としているため、これを満たすために、観光地として取り上げられる一部地域だけでなく、函館の街全体を調査の対象とした。五稜郭地区では、五稜郭公園、五稜郭タワー、五稜郭交差点地下横断歩道、千代台公園、JR五稜郭駅、函館市中央図書館、六花亭で主に調査を行った。赤川・美原・昭和・富岡地区では、四季の杜公園、笹流ダム、四稜郭、住宅街で主に調査を行った。函館駅・西部地区では、函館駅周辺、金森赤レンガ倉庫、朝市、西部地区で主に調査を行った。七飯・北斗地区では、男爵ラウンジ、なないろ・ななえ、新函館北斗駅、大沼公園、山川牧場で主に調査を行った。空港・湯の川地区では、あいす118、函館空港、トラピスチヌ修道院、香雪園、植物園、函館公園、十字街で主に調査を行った。

(※文責: 山本雄平)

2.2.4 振り返り

調査で得た情報をまとめてほかの班と共有するため、各グループで調査中に気づいた点について、Google Jamboardに写真や付箋を用いて書き出しを行った。その後、書き出した写真や付箋をグループ分けをした。グループ分けは、班員それぞれが気づいた点を調査グループ内で共有すること、発表を行うときに、何処に行き、どのような気づきを得られたかを共有しやすくすることを目的とした。グループ分けは、班員がそれぞれで気づいた点の書き出しを行った後、それぞれの気づいた点を五稜郭や朝市など場所ごとで行った。グループ分けが完了したあと、Zoomの画面共有機能を用いて、1グループずつ、調査結果をまとめたGoogle Jamboardを見せながら発表した。プレ調査では、函館山を訪れた。調査の結果、展望台やロープウェイにはビーコンが設置されてい

ること、山でも GPS が利用できること、登山道の一部ではソーシャルディスタンスが保ちにくいこと、登山道が多く、迷いやすいことなどがわかった。本調査では、「2.2.1 事前調査」で述べた 5 つの地区を訪れた。調査の結果、五稜郭や函館駅、湯の川周辺などの主に観光地で、時短営業や観光客の減少などの COVID-19 の影響が見られること、夏にマスクを着用しながら歩き回るのはつらいこと、函館駅・西部地区では多くの撮影スポットがあること、公園では年配の方が多いことなどがわかった。これらの気づきについて組み合わせを考えたり、グルーピングを行い、さらに議論を通じてアイデアの考案を行った。

(※文責: 山本雄平)

2.3 サービスの考案

2.3.1 アイデア考案に用いた手法

BSKJ 法 (ブレインストーミング KJ 法)

調査と振り返り後、ブレインストーミングを踏まえて BSKJ 法を行った。BSKJ 法とは、思いつく限り多くの量のアイデアを出すというブレインストーミングと、それらを付箋等にアウトプットして得られたアイデアを整序しグルーピングを行う KJ 法と組み合わせた方法である [11]。今回は 4 人 1 グループを 3 グループ作成した。また、1 グループを 4 人に設定したのは、グループごとに異なるアイデアを出すことができ、後述する「OST (オープンスペーステクノロジー)」を実施することから、ほかの班のアイデアを見ることで新たな発見ができると考えたためである。グループごとにフィールドワークの共有結果を見ながら、Google Jamboard で付箋にアイデアを書いて、貼り付けを行った。その後、乗り物、観光地、聖地巡礼、植物や動物などの様々な観点でグルーピングを行った。

(※文責: 山本雄平)

ハッカソン方式

ハッカソンはハックとマラソンを掛け合わせた造語である。本来のハッカソンの内容はチームで短時間でサービスやシステム、アプリケーションを開発し、成果を競うものとなっている。この方式は時間制限を設けることで、プレッシャーを与え、その結果、良いアイデアが出るのではという期待を込めて取り入れた。私たちは時間制限設けること、文字よりも絵の表現のほうがアイデアのイメージより相手に伝わると考え、アイデアの絵を描くことをルールとして設けて、これをハッカソン方式とした。この方式を中間発表前と夏季休業期間のアイデア出しの 2 回用いた。1 回目は BSKJ 法のグループと同時並行で 90 分という時間制限を設けて 4 人 1 グループでアイデア出しを行った。2 回目は参加者が順番に出題し、5 分の時間制限を設けてお題に沿ったアイデア出しを行った。実際に絵に起こすことで、アイデアを視覚的に理解できるので、イメージをつかみやすく、アイデア共有がしやすくなった。

(※文責: 山本雄平)

OST（オープンスペーステクノロジー）

BSKJ法とハッカソン方式を行った後に、関心のあるテーマについて考え、より深い意見にするためにOSTを実施した。OSTとは、参加者が自ら議題を設定、または、参加する議題について自発的に選択することで議論を進めていく方法である[12]。OSTの特徴は、参加人数、時間やスペースの制限がないということである。参加者自身で議題を選択することから、議題によって、人数にばらつきがある。また、ほかの議題への移動も参加者に委ねられているため、時間を自由に使って議論をした後、ほかに議題に移動することができる。今回はBSKJ法3グループ、ハッカソン方式1グループでそれぞれ部屋を作成し、プロジェクトメンバーは興味のあるアイデアを探し、議論を行った。興味のあるアイデアについて議論する時間に制限は設けず、考えがまとまったらほかのアイデアを探す。これを繰り返し、それぞれのアイデアを深めていった。また、他グループのアイデアの発想を持ち帰り、グループのメンバーに共有し、さらに議論を繰り返すことで、一つ一つのアイデアを具体的にすることができた。OSTを実施した結果、斬新なアイデアや何か要素を加えることで発展しそうなアイデアに関しては、議論が進んだものと、既に実装されているものに近いアイデアなど人が集まらず、議論がされなかったものとで明確に分かれたので、その後に行われたアイデアの絞り込みを容易に行うことができた。

（※文責: 山本雄平）

2.3.2 中間発表時の外部レビュー

中間発表時に提案する仮アイデアとして、OSTを用いて「俺を食べてくれえ!!」、「カラオケ×ビーコン」、「観光客×地元民」、「函館のここ、おすすめかも」、「目の前のイベントは何?」、「クーポン長者」、「五稜郭地下歩道の活性化」、「未来大のライブラリをよくしよう!」の8つに絞り込んだ。

1. 海産物自身が最適な調理法をアピールしてくる「俺を食べてくれえ!!」
2. カラオケ店で注文や清算をスマホを通じてできる「カラオケ×ビーコン」
3. 地元をよく知る人をビーコンで探して質問できる「観光客×地元民」
4. 行動履歴をもとに、おすすめのお店を紹介する「函館のここ、おすすめかも」
5. 大道芸などの突発的なイベントの情報を得ることができる「目の前のイベントは何?」
6. お店を訪れた人に他店のクーポンを配る「クーポン長者」
7. 地下歩道のマップや周辺店舗の情報を表示する「五稜郭地下歩道の活性化」
8. 見つけたい図書の位置を教えてくれる「未来大のライブラリをよくしよう!」

サービスの選定のため、これらの仮アイデアについて、中間発表で評価を依頼した。8つの仮アイデアについて、「使ってみたいと思えるものはどれか」、「使ってみたいと思った理由や気になった点」について、Google Formsでの発表評価シートで回答を依頼した。中間発表で得られた評価は以下が挙げられる。

- 俺を食べてくれえ!!：普段やったことない調理方法を見て買おうと思う主婦には響くアイデ

アだと思ったけど、そもそも目が肥えてる主婦はアプリ見なくても鮮度とか調理方法とかわかるかも。どんな人に使ってほしいのかをもう少し詳しく知りたい。

- カラオケ×ビーコン：時代に合ったアイデアで良いですね。函館のカラオケでは入店する前に駐車場に車を停める人も居るので、駐車場に居る段階でカラオケの混雑状況を知ることができれば感染リスクを避けるという意味では安全につながるかも。
- 観光客×地元民：地元の人と観光客を結びつけるという考えは興味深かったです。でも、突然観光客に話しかけられるのは少し怖い面もあるかなと思いました。
- 函館のここ、おすすめかも：函館に来て数年経っても行ったことがない場所がたくさんあるため、新しい場所を開拓するのに使ってみたいと思いました。
- 目の前のイベントは何？：よくわからないイベントも多く開催されているので情報は調べたい。
- クーポン長者：クーポンの力を利用して連鎖的にお店の売り上げを上げるという考えがナイスだと思いました。しかし、あそこのクーポンが欲しいからここで何か買わないといけないという展開になってしまう気がするので、多くの人がどうしたら利用するか深く考える必要があると思いました。
- 五稜郭地下歩道の活性化：五稜郭の交差点の地下歩道は、確かに、空間的には広いので、いろいろ活用が見えると面白いかも、と思いました。
- 未来大のライブラリをよくしよう！：未来大のライブラリーが、COVID-19の影響で、当初の使い方が出来なくなっているの、何か突破口になるようなアイデアがあると面白いかもしれません。

また、8つの仮アイデア全体を通じて、詳細な情報まで決定していなかったことから、「個々についてアイデアについて説明が少なく、正直、選択するのが難しいです。」や、「ユーザーがなぜそのアプリを使うのかというユーザーのインセンティブが強く訴えられているとよいと思いました。」という意見も得られた。

(※文責: 袴田結女)

2.3.3 プロジェクトでの内部レビュー

中間発表終了後、新たにアイデア出しを行った。アイデアを評価する際に、本プロジェクトの目的を達成するようなサービスの基準を話し合い、「サービスの函館らしさ」、「ビーコンである理由」、「サービスの必要性」、「サービスの新規性」、「サービスの不変的な魅力（継続性）」の5つに定めた。「サービスの函館らしさ」は、函館市による取り組みの一端を担うためである。「ビーコンである理由」は、ビーコンを用いたサービスを考案・開発することが本プロジェクトの目的であり、ビーコンの特徴を活かしたサービスを提供すべきであると考えたためである。「サービスの必要性」は、需要のあるサービスを提供したいと考えたためである。「サービスの新規性」は、ユーザーに今までにしたことのない体験をしてもらいたいと考えたためである。「サービスの不変的な魅力（継続性）」は、考案したサービスを継続して使ってもらいたいためである。以上の理由より、これらの5つをアイデアを評価する際の観点として定めた。考案した29個のアイデアに対し、プ

プロジェクトメンバーが、5つの基準を満たすかどうかを議論し、満たさないものはそれを補う案を検討した。また、29個のアイデアのうち、新たに考案した21アイデアに対し、担当教員とTAから、上記の5つの観点でフィードバックを得た。担当教員とTAからのフィードバックと、5つの観点を満たせるかどうかを踏まえ、29個のアイデアから21個のアイデアに絞り込んだ。

(※文責: 袴田結女)

2.3.4 アイデアのブラッシュアップ

プロジェクトの内部レビューの後、それぞれのアイデアが5つの観点をどのように満たしているのかを記述したドキュメントを作成し、それをもとに、KJ法などを用いて8つのアイデアに絞った。また、それぞれのアイデアに興味のあるプロジェクトメンバーを偏りの少ないように振り分け、各グループの詳細なサービス案についてのブラッシュアップを行った。各グループごとに詳細なサービス案や5つの観点に対するサービスの強みをまとめたスライドを作成し、プロジェクト内でプレゼンテーションを行った。評価の方法として各アイデアに対して「サービスの函館らしさ」、「ビーコンである理由」、「サービスの必要性」、「サービスの新規性」、「サービスの不変的な魅力」の観点から意見を出し合い、アイデアをさらに絞った。最終的なアイデアは、「ビーコンバトル」、「俺を食べてくれえ!!」、「造語シェア」、「函館のここ、おすすめかも」、「未来大のライブラリをよくしよう」、「函館を舞台としたADV」の6つでそれらの改善を行った。

(※文責: 大石晃平)

2.3.5 テーマの決定

アイデアのブラッシュアップを終え、最終的に出揃ったアイデアをスライドにまとめ、各アイデアについてのプレゼンテーションと評価を行った。また、担当教員、TAの方からも評価を受けた。評価の方法としては、各アイデアに対して、「サービスの函館らしさ」、「ビーコンである理由」、「サービスの必要性」、「サービスの新規性」、「サービスの不変的な魅力」の5つの観点について、1～10点の10段階で点数づけを行った。採点結果は、表2.1に示す。プレゼンテーションを行ったアイデアは、「ビーコンバトル」、「俺を食べてくれえ!!」、「造語シェア」、「函館のここ、おすすめかも」、「未来大のライブラリをよくしよう」、「函館を舞台としたADV」の6つであった。

1. 函館のまちを自由に散策し、観光をしながらパズルのピースを集め、観光地に思い出を残すことができるサービス「ビーコンバトル」
2. 函館の海産物自身が鮮度に応じて、適切な調理法をアピールしてくるサービス「俺を食べてくれえ!!」
3. 観光地や景色の良いところで、感じたことなどをまだない言葉で表現し投稿・シェアすることができるサービス「造語シェア」
4. 観光客が、函館の街を歩くことで、行動履歴をもとに、訪れた店との関連性や好みに合わせてお店を提案するというサービス「函館のここ、おすすめかも」

5. 未来大のライブラリの本を、楽しくかつ簡単に探すことができるサービス「未来大のライブラリをよくしよう」
6. 函館を舞台としたゲームを作成し、その場所に実際に訪れることで獲得できる追加のコンテンツによって、函館のまちの活性化を促すサービス「函館を舞台とした ADV」

これらのサービスについての得られた評価の結果を踏まえ、自分たちがどのアイデアの開発に携わりたいかという観点で投票を行い、「俺を食ってくれえ!!」、「ビーコンバトラー（のちに「でもものに!）」、「函館のここ、おすすめかも（のちに「MET.）」の3個のアイデアを今年度のプロジェクト学習で開発することに決定した。

表 2.1 各アイデアの評価の平均

アイデア名	函館らしさ	ビーコンである理由	必要性	新規性	不変的な魅力
ビーコンバトラー	6.63	8.74	7.16	7.47	7.21
俺を食ってくれえ!!	7.30	8.05	7.70	8.75	7.30
造語シェア	5.19	5.95	5.81	8.67	6.52
函館のここ、おすすめかも	6.80	6.10	7.45	7.55	7.35
未来大のライブラリをよくしよう	6.10	7.50	7.30	7.15	7.80
函館を舞台とした ADV	6.24	5.38	5.43	7.33	5.38

(※文責: 大石晃平)

2.4 その他

2.4.1 ビーコンについての事前調査

本プロジェクトではビーコンを使用した開発をおこなうために、ビーコンに関する知識を深める必要があると判断し、事前調査をおこなった。事前調査は「近接・位置測位」、「GPS との機能・特性比較」、「各種センサーとの連携」、「LINE Beacon」、「適用事例」の5つの項目についておこなった。ビーコンを扱う上で一番必要な知識であろう「近接・位置測位」や、似た機能をもつ GPS との違いを把握するために「GPS との機能・特性比較」を検討した。また、ビーコンの中でも、特徴の一つであるほかのセンサーとの連携についてより深く知り、サービスの特色を出すため「各種センサーとの連携」、iBeacon をはじめとしたほかのビーコンとは利用方法が異なる「LINE Beacon」についての検討も行った。さらに、実際の利用方法をもとにアイデアを膨らませるために「適用事例」も調べることにした。その際、本プロジェクトは15名で構成されるグループであったため、各項目にメンバーを3名ずつ割り振って調査を行い、調査した内容を各グループごとにスライドにまとめ、共有した。この事前調査を行うことによって、ビーコンの特徴やビーコン特有の機能な

ど、様々な知識を会得することができた。サービスを考案する中で、GPS や RFID タグでも実装できるアイデアが多くなり、"ビーコンらしさ"という点が薄れてしまったため問題となった。しかしながら、事前調査を行うことによって、圧力センサー付きビーコンを利用したアイデアの検討など、GPS や RFID タグでは実現できないサービスの検討に一役買ったと考えている。

(※文責: 増田翔)

第3章 提供するサービスについて

3.1 背景と課題

函館は外交の主要拠点として発展し、現在でも歴史的建造物が多く残る観光地である。建造物や夜景などを目的に国内外からも観光客が多く集まっており、今では函館の主要な産業の1つとなっている。函館市の調査 [4] によると、函館観光の同行者における家族・親族の割合は33.4%となっており、家族で訪れる層が観光客の多くを占めると言える。

しかしながら、このような観光地は対象とする年齢層が比較的高めであり、未就学の子どもや小学校低・中学年の子どもがいる家族連れでは、子どもが楽しめる要素が少ないために、本来大人が訪れたい観光地への来訪を妨げたり、逆に子どもが観光を十分に楽しめない可能性が考えられる。そこで、函館の歴史的建造物をはじめとした観光地で、大人と子どもの両方が楽しめるようなサービスを提供することができれば、函館のまちにより多くの人に観光客として足を運んでもらうことができる考えた。

(※文責: 久米田羽月)

3.2 サービスの概要

本サービスの目的は、比較的大人向けの観光地である函館で、子どもを含めた家族全員に函館観光を満喫してもらうことである。本サービスでは、観光客が自分のスマートフォンで函館のまちの各所にあるパズルのピースを集めてパズルを完成させる。そしてその体験を通して家族全員が観光を楽しみつつ思い出を残すことができる。スタンプラリーなどとは異なり、観光したい場所で観光することができ、集めたピースに思い出を書き込んで、大きなパズルに残すことができる。子どもはピースを見つけること、大人は観光することを1つの目的として函館を家族全員で楽しめる新たな体験が可能となっている。

本稿では、アプリケーション上で扱われる、ビーコンと1対1で対応したジグソーパズルのピースのようなものを「ピース」と表し、それが集まって1つの絵になるものを「パズル」、パズルが表示される物理的なデバイスを「パズルパネル」、あるいは「パネル」と表現する。

(※文責: 久米田羽月)

3.2.1 ユーザーストーリー

本サービスは、観光を目的に函館へ訪れた子連れの家族を主なターゲットとしている。ここでは、小学生の男女2人の子どもを連れている4人家族を想定したユーザーストーリーを述べる。

ユーザーは観光のため、函館空港に到着する。空港の手荷物検査場を出ると、図 3.1 のように、本サービスが提供するディスプレイが設置してあり、そこに「でももに！」のパズルパネル (以下、パネル) が表示されている。パネルにはすでにいくつかのピースがはまっており、未完成であることがわかる。ここで、パネルに興味を持った子どもが、両親をお願いをして、母親のスマートフォンにアプリケーションをインストールしてもらおう。インストールが完了したあと、実際に子どもがアプリケーションを起動させる。このとき、アプリケーション画面には「まちを歩いてみよう!!」と表示されており、なぞなぞ一覧からなぞなぞを選んで、ピースを探すことを促される。親はなぞなぞ画面から、訪れる予定のある地域を選択し、その中から1つを選んで子どもに見せる。このとき、都合に合わせて、地区のカテゴリーである程度なぞなぞを絞り込むこともできる。子どもだけで分からない場合は、両親も協力してなぞなぞを解き、ピースのある場所に検討をつける。



図 3.1 パネルを見つけ、なぞなぞを解く家族

家族はなぞなぞの答えと思われる観光地へ向かい、両親はアプリケーションを起動して子どもに渡す。子どもにとっては少し退屈な場所と感じられるかもしれないが、本サービスのアプリケーションがあれば、子どもはピース探しを楽しみ、親は後ろについて行きながらも、その場の景観を楽しむことができる。もし子どもがピースを発見すると、図 3.2 のように、ピースを取得したことが画面内で通知される。



図 3.2 ピースを見つける子ども

子どもはそれを両親に伝え、発見を一緒に楽しむことができる。取得したピースの裏面には、図 3.3 のように、任意の写真を貼り付け、自由にサインを書き込むことができる。観光地で撮った写真に、家族全員で思い出を書き残すことができる。

その後、家族は、ピースの有無に関わらず函館の様々な場所を観光する。なぞなぞを解かなくとも、ビーコンを検知した場合はピースが自動的に取得され、ピース一覧にピースが追加される。観光を終えて帰路のために空港に向かうと、家族は「でももに！」のパネルがあることを思い出す。



図 3.3 ピースにサインを書き込む家族

パネルに取り付けられた専用のビーコンを検知すると、アプリケーションの画面にはピースをパネルにはめることができる旨が表示される。そこで、両親は子どもにピースをはめることを促す。アプリケーション上でピースを送ることを選択すると、アプリケーションの画面上からピースが飛んでいくような演出が行われる。同時にパネルのディスプレイ上にピースが現れ、ピースがはまる演出が行われる。ピースがはまると、図 3.4 のように、ディスプレイ上にはピースを見つけたお祝いのメッセージが大きく表示される。家族が協力して集めた思い出の詰まったピースがパネル上に表示され、観光地に思い出を残すことができる。



図 3.4 パネルにピースをはめる家族

こうして、函館観光を子どもも含めた家族全員で楽しみ、思い出を残すという、新しい体験をすることができた。

(※文責: 久米田羽月)

3.3 機能

3.3.1 機能一覧

アプリケーションが提供する機能の一覧を以下に示す。

- チュートリアル機能
- ピース探索機能
- ピースの取得機能
- ピース一覧機能

- ピース一覧機能
- パネル状況閲覧機能
- ピース情報閲覧機能
- なぞなぞ一覧機能
- サイン書き込み機能
- ピースアップロード機能
- パネルの機能

3.3.2 チュートリアル機能

ユーザーがアプリケーションを起動させると、図 3.5 に示すように、後述するメイン画面であるピース探索画面に遷移する。ピース探索画面には、プロジェクトのオリジナルキャラクターである「うさ」が歩いている様子がアニメーションで表示されており、ユーザーにはまちを歩くように促すようなメッセージを示す。さらに、ユーザーがアプリケーションを初めて起動させた時には、「うさ」がなぞなぞのページを表すタブを指し示し、ナビゲーションバーからなぞなぞのページを開き、なぞなぞを解くとピースのありかがわかることをメッセージとして示す。このように、ユーザーにどのような体験ができるのかを知ってもらうための工夫として、チュートリアルを行っている。



図 3.5 チュートリアル画面

(※文責: 久米田羽月)

3.3.3 ピース探索機能

ユーザーがピースを探す時にこの機能が使われることを想定している。ピース探索画面には「うさぎ」が歩いている様子がアニメーションで表示されており、図 3.6 のように、ビーコンを検知していない時は通常通りまちを歩くように促すメッセージが表示される。もし、ピースと対応するビーコンが検知されたときには、図 3.7 のように、近くにピースが存在することをメッセージとして表示し、ユーザーがピースを探すためのヒントとして提示する。



図 3.6 通常時のメイン画面

(※文責: 久米田羽月)

3.3.4 ピースの取得機能

ユーザーがアプリケーションを起動させた状態でまちを歩いている時、近距離でビーコンを検知した場合にそれと対応するピースを取得することができる。その際、ユーザーはアプリケーション画面に表示されるページからどのようなピースを取得したかがわかるようになっており、ピースの名前、取得した地区、取得日の情報を閲覧することができる。このサービスではビーコンの固有番号を用いて判別を行っている。そのため、サービス提供側がビーコンの設置を行わなくとも、各地の公共の場に存在するビーコンを有効に活用することができる。さらにビーコンを用いることで、市電や人力車、ロープウェイなどをピースの場所として設定することができるため、より柔軟なサービス展開をすることができる。

(※文責: 久米田羽月)



図 3.7 ピースが近くにある時のメイン画面

3.3.5 ピース一覧機能

ユーザーは図 3.8 に示すピースの一覧画面より、これまでに取得したピースを視覚的に閲覧できるようになっている。また、表示されたピースをタップすることで、そのピースの情報を閲覧することができる。この機能は、今までにどのようなピースを取得したのか確認したい場合に使用されることを想定している。

(※文責: 久米田羽月)

3.3.6 パネル状況閲覧機能

ユーザーは各地にあるパネルについて、どのピースがすでにはまっているのかを閲覧することができる。この機能は、まだ誰にも発見されていないピースを探したいユーザーが、どのピースが発見されているのかを確認したい場合に使用されることを想定している。

(※文責: 久米田羽月)

3.3.7 ピース情報閲覧機能

ユーザーはピース一覧画面に表示されたピースをタップすることで、そのピースの情報を閲覧することができる。ピースの名前、取得した地区、取得日、対応するなぞなぞの情報を閲覧することができる。また、この画面からサインの書き込み画面を表示させることができる。この機能は、



図 3.8 ピース一覧画面

ユーザーがピースについての思い出を振り返りたい場合や、取得したピースにサインを書き込みたい場合に使用されることを想定している。

(※文責: 久米田羽月)

3.3.8 なぞなぞ一覧機能

ユーザーは図 3.9 に示すように、なぞなぞ一覧画面より、ピースの場所を示すなぞなぞの一覧を確認することができる。なぞなぞ一覧に表示された項目をタップすることで、なぞなぞの名前、本文、地区の情報が表示されるようになっている。また、ヒントボタンを押すことで、図 3.10 に示すように、なぞなぞのヒントを表示させることができる。さらに、図 3.11 に示すように、ピースの場所に関する地区情報などのタグを指定することで、合致するなぞなぞを検索して表示させることができる。この機能は、ピースを探すためになぞなぞを閲覧する場合に使用されることを想定しており、事前に行き先を決めていた場合には、都合に合わせてピースの存在する地区を絞り込むこともできる。

(※文責: 久米田羽月)

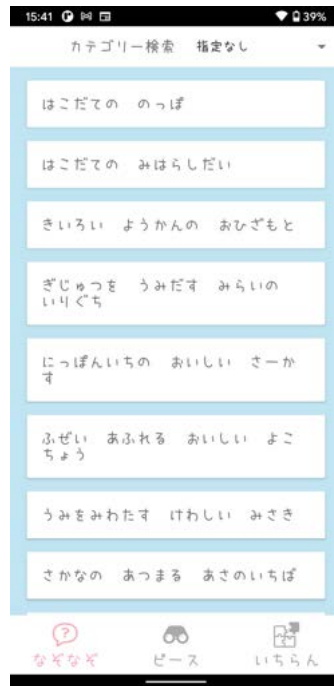


図 3.9 なぞなぞ一覧画面

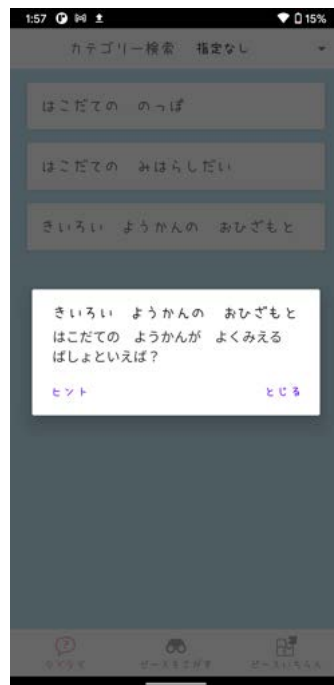


図 3.10 なぞなぞの情報

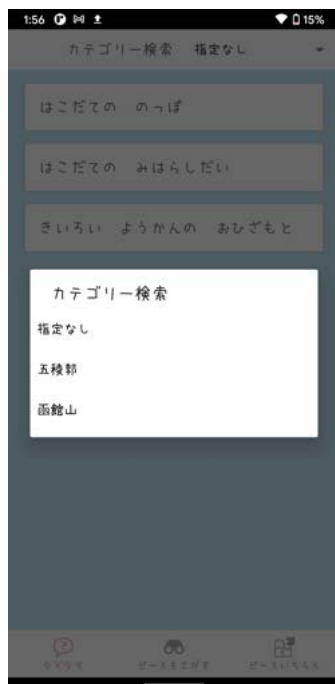


図 3.11 カテゴリ検索機能

3.3.9 サイン書き込み機能

ピース取得画面またはピース一覧画面より、取得済みのピースを選択することで、図 3.12 のように、サインの書き込みを始めることができる。ユーザーは取得したピースの裏面に、任意の写真を設定し、サインを書き込むことができる。サイン書き込み画面では、ペン・消しゴム・戻る・クリアの操作ができる。この操作は、パネルに送信されるまで何度でも行うことができる。テキスト情報の書き込みではなく、この機能を実装した理由として、写真と手書きのサインを残せることで、単なるデータではない思い入れのある体験を残せると考えたためである。その他の理由として、子どもでも直感的に書き込みの操作ができると考えたためである。

(※文責: 久米田羽月)

3.3.10 ピースアップロード機能

アプリケーションが起動された状態でパネルに近づくと、パネルに取り付けられたビーコンを検知することで、図 3.13 のようにピースがアップロードできることがユーザーに通知される。このとき、ピースを送信するかどうかを確認する画面が表示される。ユーザーはパネルのビーコンが検知されている間、任意のタイミングで、取得済みかつ未送信の任意のピースを送信することができる。ユーザーがピースを送信した場合、図 3.14 のように選択したピースがアプリケーションの画面外に消える演出が行われる。同時に、設置されたパネルの画面では、ピースがはまる演出が行われる。このような演出があることで、ユーザーの達成感を向上させることができると考え、この機



図 3.12 サイン書き込み画面

能を考案した。

(※文責: 久米田羽月)

3.3.11 パネルの機能

各地に設置されたパネルの画面には、図 3.15 のようにユーザーによって送信されたピースが表示され、誰でもパズルの状態を閲覧することができるようになっている。表示されたパネルは時間差で裏面が表示されるようになっており、ユーザーが書き込んだ写真とサインが表示される。これはパズルのピースが埋まっていく過程を通りがかる人に見てもらうためであり、それと合わせて裏面の記念写真やサインが公開されていることで、集客と同時に、観光地に思い出や足跡を残せるような効果を狙ったものである。

ユーザーがアプリケーション側で自分の見つけたピースをパネルに送信した場合、リアルタイムにピースがはまるような演出がなされ、図 3.16 のようにピースを発見したことを称えるメッセージが表示される。

(※文責: 久米田羽月)



図 3.13 パネルが近くに存在する時の画面



図 3.14 ピース送信画面



図 3.15 パネルの画面

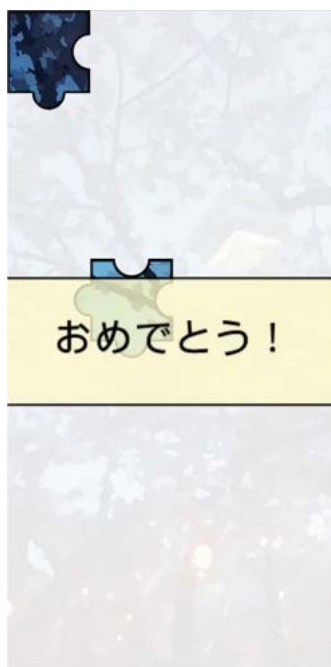


図 3.16 ピースの発見を讃える画面

3.4 アイデアの発案から最終的なサービス内容決定までの経緯

このサービスについて、発案当初は現在と異なるサービス内容を想定していた。発案当初は、1991年にエポック社から発売された「バーコードバトル」[13]に着想を得て、バーコードではなくビーコンのUUIDを用いた対戦ゲーム、「ビーコンバトル」として提案した。ビーコンのUUIDを収集し、友人や家族と対戦することで、より楽しく函館を散策してもらい、函館を活性化させることを期待して考案したものであった。

しかし、先生やTAからのレビューでは、UUIDではユーザーがゲームの法則を発見できずにゲーム性が弱いこと、観光客が観光地でゲームをしたいかどうかは分からないこと、ユーザー目線に立ったときにメリットが曖昧であること、函館で行うことに明確な意義が見出せないことなどが指摘された。

このような理由で、当初の案は採用されなかったものの、ビーコンのUUIDをコンテンツとして活用したことについては独自性が高かったことが評価された。ゲーム性には期待できなかったが、収集要素については悪くない評価が得られたこともあり、ジグソーパズルを集めるという改善案が得られた。函館を楽しく散策してもらおうという目的はそのままに、なぞなぞでピースの場所を出題され、パズルのピースを集めてサインし、空港のパネルにはめることで達成感を得られるという体験のアイデアは、このレビューで得られたものをもとにしている。

本サービスの名前である「でももに！」は、大人も子どもも家族一緒に楽しんでもらいたいという願いから、「パズルで子どもと一緒に函館観光！」という文のひらがな部分を抜き出したものに由来している。

(※文責: 久米田羽月)

第 4 章 開発

4.1 開発で用いたコミュニケーションツール

4.1.1 Slack

Slack のチャンネルでは、メッセージの送信、ファイル共有やテレビ電話機能だけでなく様々なソフトウェアとの連携も可能にしており、外部のソフトウェアに素早くアクセスできる。Slack はプロジェクトの活動においてメンバーだけでなく、教員や TA との連絡に用いた。私たちはサービス、プラットフォームやデイリースクラム用のチャンネルを作成することで、それぞれのチャンネルの役割を決めて連絡を取った。また、特定の人に通知を送ることができる、メンション機能はサービスで共有すべき重要な連絡に用いて通知を送ることで、早いレスポンスを実現し、スムーズに連絡を取っていた。実際の開発の場面では 4.2.2 節の GitHub との連携することで、Pull requests, Merge などを通知して、プラットフォームごとの進捗を把握することに利用した。

(※文責: 山本雄平)

4.1.2 Discord

音声・ビデオ通話を可能としたソフトウェアで、テキストチャンネルのチャットはもちろん、作成したボイスチャンネルごとに通話を行うことが可能である。開発の場面においては、作業している状況を画面共有することで、各メンバーが行っていることを視覚的に把握しながら、作業を進めていった。また開発以外にも、サービス会議やデイリースクラムなどで Discord を利用した。

(※文責: 山本雄平)

4.1.3 Zoom

クラウドコンピューティングを使用した Web 会議サービスであり、チャット機能、画面共有機能、録画機能、ブレイクアウトルームという部屋を複数作成できる機能と多機能なサービスである。録画機能に関しては、プロジェクト活動や勉強会を録画することに利用し、録画データを 4.3.2 節の YouTube にアップロードすることで、活動を振り返ることができるオンライン活動ならではの工夫をしていた。

(※文責: 山本雄平)

4.2 開発環境整備

4.2.1 Git

Git は分散型バージョン管理システムであり、Linux のソースコード管理のために開発されたものであるが、我々は、サービスのアプリケーション開発で必要となるソースコード管理に用いた。Git はファイルの状態を、コミットとして更新の記録を保存することができ、編集したファイルの差分を確認したり、ファイルの編集を取り消したり、ファイルの状態を過去の状態に戻すことができる、ソフトウェア開発において必須な、便利なシステムである。

(※文責: 村石拓海)

4.2.2 GitHub

GitHub は GitHub 社の Git を利用した Web サービスであり、複数人でのチーム開発を行う上で便利なサービスを提供している。今回の我々のサービス開発では、モバイルネイティブアプリケーション作成の iOS アプリケーション開発と Android アプリケーション開発、さらには、空港や駅に設置することを想定したパズルパネルを表示するための Web アプリケーション開発において利用した。具体的には、開発でのソースコード管理や、タスク管理用にカンバン機能である GitHub Projects の機能を利用してスプリントバックログを管理した。また、最終成果発表のために Web サイトを作成するために GitHub Action を利用した。担当教員から GitHub の招待をされて使い始めたことであるが、中間報告書の作成に利用したり、夏季休暇中に Git や GitHub の勉強会を行っていくことで、全員が GitHub の利用に抵抗を持たずに利用できるようになっていった。また、GitHub と Slack との連携を行うことで、GitHub の開発サービスリポジトリの更新を監視し、更新があった際には Slack に通知が来ることで、より他のチームメンバーが開発状況を把握しやすくし、オンラインでの開発でもなるべく円滑に開発が進むように工夫した。

(※文責: 村石拓海)

4.3 活動記録管理ツール

4.3.1 esa

情報共有サービスで、Markdown 記法を採用しているが豊富な入力補助があるため、初心者でも Markdown 記法で記事を書くことができる。そのほかの機能として、共同編集、ソフトウェア連携、途中でも記事を公開できる WIP 機能など様々なものを備えている。我々は前期からプロジェクト活動の議事録をとるために利用しており、開発においてはサービス会議の議事録や、エレベーターピッチやユーストリー等のインセプションデッキの記録など、各スプリントでの活動の議事録をとるために利用した。共同編集機能を用いることで、サービス会議の記録を一人ではなく、複数人で書くことで、会議の内容を見落とすことなく、詳細まで記事に残すことができた。これに

よって振り返り時に記事を見返すことでいつ何を行ったかなど、簡単に振り返ることができた。

(※文責: 山本雄平)

4.3.2 YouTube

動画共有サービスで、アカウントを作成することで、自分のチャンネルを開設することができ、動画のアップロードや公開が可能になる。我々は、プロジェクトで1つのアカウントを作成し、毎回のプロジェクト活動の録画をアップロードすることで、活動の振り返りを行うことや、夏季休暇中の勉強会の録画をアップロードすることで参加できなかったメンバーに共有することを行っていた。

(※文責: 山本雄平)

4.4 スクラム関連ツール

4.4.1 Google Jamboard

電子ホワイトボード機能を持つクラウドアプリケーションで、ペンで図を書くことや、付箋や画像の貼り付けはもちろん、共同編集も可能としている。前期の活動において前述したブレインストーミング、KJ法やバックログの管理に利用した。

(※文責: 山本雄平)

4.4.2 Miro

オンラインホワイトボードサービスで、共同編集を可能とした、豊富なテンプレートにより幅広い目的の利用を可能としている。我々は前述した Google Jamboard を用いて行ったバックログの管理を Miro に移行し、その他にスプリントバックログの管理、インセプションデッキの作成・管理、KPTA の作成に利用した。

(※文責: 山本雄平)

4.4.3 hatjitsu

Web ブラウザでプランニングポーカーができるツールで、ポイントの種類にフィボナッチ数列や T-shirts など用意されている。前期の活動では、バックログの重さ付けにおいて利用した。我々のサービスでは、スプリントバックログの重さ付けの際に利用した。

(※文責: 山本雄平)

4.5 デザインツール

4.5.1 Adobe XD

Adobe XD は Adobe 社による Web サイトやモバイルアプリケーション、音声デバイスの UI/UX のためのデザインツールであり、ワイヤーフレーム、デザインカンパ、プロトタイプを作成できるソフトウェアである。我々は、iOS、Android アプリケーションのモックアップの作成に利用した。利用できる UI/UX パーツに iOS、Android 用のものが用意されており、手軽にまとまったスマートフォン画面のモックアップができた反面、基本的に全員がツールの扱いに慣れていないため、作りたいモックアップを作るために、Adobe XD の使い方を調べながら行っており、作りたいモックアップを思い通りの形で表現することに苦労した。

(※文責: 村石拓海)

4.5.2 Adobe Illustrator

Adobe Illustrator は Adobe 社によるベクターグラフィックツールの業界標準といえるツールで、Web やモバイル用のグラフィックからロゴ、アイコン、本のイラスト、商品パッケージ、看板まで、あらゆるものをデザインできるソフトウェアである。我々は、アプリケーション内の各種素材作成に利用した。色々なデザイン機能が搭載されており、使い方に精通していれば作りたいものが作れる反面、ツールの扱いに慣れていないことから、他のデザインツール同様、機能を調べることから行う必要があり、作りたい素材を思い通りの形で作成することがなかなかできず苦労した。

(※文責: 村石拓海)

4.5.3 Adobe Photoshop

Adobe Photoshop は Adobe 社による AI を活用して画像を魅力的に仕上げることができるツールで、写真の編集や合成から、デジタルペインティング、アニメーション、グラフィックデザインのできるソフトウェアである。我々は、アプリケーション内の各種素材や最終成果発表の動画素材の透過編集を主として、素材の作成に利用した。画像素材の縁取りを自動で行うことができる機能があるため、簡単に素材の切り抜きができ便利であった。Adobe Photoshop もほかの Adobe 製ソフトと同様、プロ用で機能が非常に多くそのことが逆に初心者にとっては、うまく使うことができず苦労する点もあった。また、Adobe XD は共同編集機能があり、複数人でリアルタイムに共同編集することができる。アプリケーションのモックアップを作る上で、グループメンバーと話し合いながら作成する必要があるが、オンラインでの活動であることもあり、オフラインで集まって、話し合うことができなかった。それゆえ、Zoom や Discord を用いて、リアルタイムで通話しながら、Adobe XD で複数人でリアルタイム共同編集をすることで、グループメンバー内で、モックアップについて話し合いながら作成した。

(※文責: 村石拓海)

4.5.4 Adobe Premiere Pro

Adobe Premiere Pro は、Adobe 社による映画やテレビ番組、Web ビデオの制作に使用される最先端のビデオ編集ソフトウェアある。様々なクリエイティブツールと、Adobe の他のアプリケーションやサービスとの統合、Adobe Sensei の機能により、単一のシームレスなワークフローで、フッターを完成された映画やビデオ作品へと仕上げることができます。Adobe Premiere Rush との緊密な統合により、いつでもどこでも好きなデバイスでフッターのキャプチャと編集を開始できる。我々は最終成果発表のチームでの動画作成に利用した。

(※文責: 村石拓海)

4.5.5 CLIP STUDIO PAINT

CLIP STUDIO PAINT は CELSYS によるペイントソフトである。我々は、iOS、Android 両アプリケーションアイコンのアイコン作成、アプリケーション内の各種素材作成に利用した。また、サービスのユーザーストーリーについて話し合うときや、アプリケーションアイコン、その他について話し合いたいときに、お絵描きツールとしても利用した。オフラインでの活動であれば、集まって画用紙に書いたり、ホワイトボードを用いて、絵を書いて話し合うことができるが、オンラインでの活動であったため、Zoom や Discord を用いて、リアルタイムで通話しながら、CLIP STUDIO PAINT の画面共有することでお絵描きする画面をグループメンバーにリアルタイムで共有しながら、話し合うことで、サービス内で話し合う際のお絵かきツールとして有用であった。

(※文責: 村石拓海)

4.6 技術習得

4.6.1 夏季休暇期間中の勉強会

本プロジェクト内で、チームでの開発が未経験であるメンバーがいた。また、夏季休暇期間前に大まかなプラットフォーム分けと使用するプログラミング言語の決定を行った。その中で、各メンバーが今まで開発を行う際に使用していたプログラミング言語とは異なる言語を扱ってみたいという要望があった。そのため、後期からの本格的な開発に向けて、開発で必要となる知識をつけることと、各メンバーが主に開発するプラットフォームを決めることを目的として、夏休み勉強会を実施した。この勉強会の形式については、開発について初学者から経験者まで幅広い経験の差があったため、全員で内容を統一すると、初学者が理解しにくい内容になってしまうといった不都合が生じることが予想された。そのため、各メンバーが発表しやすく、内容に合わせやすいように、発表者が自由に形式を決定することにした。自分の持っている知識や、夏季休暇期間中に行ったことを LT 形式で発表したり、実際に見てもらっているメンバーに手を動かしてもらいつつ発表者がライ

ブコーディングを行ったりした。自分の興味のあるプラットフォームの勉強会には積極的に参加してもらい、1人約1時間ほどの勉強会を、メンバー全員がそれぞれ2回企画した。また、行った勉強会を録画し、その動画をプロジェクト内で共有することで、のちに復習しやすい環境を作ったり、参加できなかったメンバーが学習しやすい環境を作った。この勉強会を通じて、GitHubの基本的な知識や、サーバー、モバイル、Webといった各プラットフォームごとの知識が得られた。また、プログラミング関連以外にも、Adobe Illustratorを用いたポスターや資料を作る方法や、イラストを描くためのノウハウを学んだ。各メンバーが開発に携わりたいプラットフォームについての知識が深められ、円滑に開発に進められることができた。

(※文責: 石澤大輔)

4.7 開発手法

4.7.1 開発の流れ

「1.4.2 本プロジェクトにおける開発手法」でも述べたように、私たちはアジャイル開発手法の1つであるスクラムを実施した。開発の流れとしては、スプリントの期間を決定後、1回のスプリント内に行う開発のタスクを計画する。スプリント期間中は振り返りを行い、スプリントの最終日は1スプリントの振り返りと、ステークホルダーに成果物を報告会を実施した。

1スプリントの期間については、土曜日を開始とし、金曜日を終了とする1週間で決定した。開発のタスクの計画に関しては、プロダクトバックログから1スプリントのできるであろうバックログをメンバーと相談して決定して、取り掛かった。

スプリント期間中の振り返りとして、平日の12時15分から15分間の時間を設けてデイリースクラムを実施した。デイリースクラムでは、Slackに作成したチャンネルにおいて、12時15分までに「昨日行ったこと」「今日行うこと」「困っていること」の3つの項目について記述してもらい、時間になったら、Discordに集合して、Slackに書いた内容について発表した。

金曜日のスプリントの振り返りであるスプリントレトロスペクティブではKPTAとスプリントレビューを行った。KPTAは活動内容等を「Keep(目標の達成に向けて成果が出ているので継続して行うべきこと)」、「Problem(目標を達成するうえで問題となっており、改善する必要があること)」、「Try(出てきた問題を改善するために必要なこと)」、「Action(実際に行うこと)」の4つの項目で検討を行うことで、私たちのサービスでは、KPTAを用いて、1スプリントの活動内容について検討し、次のスプリントの方針を決定した。

成果物の報告会はスプリントレビューを実施した。スプリントで開発を行うことでできた、エレベータピッチなどの方向性から、実際のアプリケーションのデモなどのレビューを実施することで、次以降のスプリントの開発に活かした。

(※文責: 山本雄平)

4.7.2 効果

開発において、「4.7.1 開発の流れ」で述べたような手法を取り入れたことによる効果について、以下に記述していく。

デイリースクラムを導入することによって、1日に1回メンバーと話し合いの機会を設けることができ、困っていることの共有や問題発見をしやすくなった。オンラインでの開発を実施していたため、連絡は基本チャットで行っていたが、文字での表現が伝わりづらく伝え手と聞き手の間で言葉の意味がすれ違ってしまいうこともあった。また、通話に関してはメンバーの予定を調整しないと実現することが困難だった。そのため、デイリースクラムで1回話し合う時間を設けることで、チャットでは意味がすれ違ってしまいう内容でも、会話を通じて理解しやすくなった。個人のタスクで困っている事やサービス全体での問題点を挙げることで、すぐに解決できるものであれば、デイリースクラム終了後に時間を設けて、問題解決に向けて取り組むことができた。デイリースクラムでは各メンバーの進捗状況の確認ができるだけでなく、問題点の共有とその解決に向かわせるきっかけ作りができた。

スプリントの最終日に実施したスプリントレトロスペクティブのKPTAでは、メンバーそれぞれが考えていたことを書き出すことで自分自身では見つけることができなかった問題を網羅的に挙げるができ、その問題どのように解決していくかを考えることで、次のスプリントを効率的に行うことができた。また、良い点を書き出すことで、続けていこうというモチベーションの向上につなげることができた。

スプリントレビューは、開発の前半では、エレベーターピッチなどをレビューしてもらうことで、私たちがサービスで伝えたい点が伝えられているのかをアドバイスをもらい、何度も見直すことができた。開発の後半ではアプリケーションのデモのレビューを実施し、私たちのサービスで重要にしている「楽しさ」という点や対象が家族であるというサービスの方向性からずれていないかのアドバイスをもらい、アプリケーションの改善に活かすことができた。

(※文責: 山本雄平)

4.7.3 課題

スクラムや「4.7.1 開発の流れ」で述べた方法を導入したことにより、「4.7.2 効果」の効果が出ただけでなく、いくつか見つかった課題について以下に記述していく。

サービス開発全般では、開発の最初の2スプリントまでサービスの方向性についての話し合いをしていたため、開発に取り掛かることができず、開発のタスクが後に集中してしまった。デイリースクラムに関しては、本来、全員参加で決まった時間に実施する必要があったが、実際はチームメンバーの寝坊が主な原因で集合ができない場合や、Slackに「昨日行ったこと」、「今日行うこと」、「困っていること」を書き出している間に時間が来てしまい開始時間に始めることができなかった。これに伴い、15分という時間を守るできないことが多々あった。

KPTAに関しては、導入して間もない時に一つ一つの項目を検討する時間が曖昧でActionの内容を出すのが疎かになってしまった。また、Actionで挙げた内容を全て実行することができず、

挙げただけになってしまった内容が存在した。スプリントレビューは、アプリケーションのデモを見せる機会が2回しかなく、レビューをもらう機会が少なかったことである。これらの課題を如何にして解決したのかは次節で説明する。

(※文責: 山本雄平)

4.7.4 解決策

「4.7.3 課題」で挙げた様々な課題を解決するために行った対策を以下に記述していく。デイリースクラムの課題である、時間になっても開始できないことについての原因は、チームメンバーの寝坊と、Slackに予め記述するということが徹底されていなかった事である。この対策として、Slackの機能を使用することで、開始15分前にリマインダーを送ることや、寝坊しそうなメンバーには電話を使用して起こすこと実施することで、以前よりは減少したが、完全な解決には至らなかった。その理由として挙げられるのは、電話しても必ず起こすことができなかったことにある。また、デイリースクラムの時間通りに実施できない他の原因であった、Slackのデイリースクラム専用チャンネルへの書き込みが間に合わないことであったが、これは、メンバーがど忘れしていることが一因であったため、先ほども使用した、Slackのリマインダー機能を使用して、通知を行うことで、この原因によるものは解決することができた。

KPTAの課題に関しては、Keep, Problem, Try, Actionそれぞれの項目において検討する時間を決めていなかったことによって、順番的に最後にあたるActionを出す時間がなかったことである。この対策法として、各項目に対して具体的に時間を定義した。この方法は一度大学に来てオフラインでこの方法を実行したときに、上手くKPTAを行えたこともあり、この方法と以前の方法との違いを議論した際にオンラインのホワイトボード上に一斉に書き出しを行っていたときに自分の考えが他の人に引っ張られてしまい、考えることをやめてしまうことで考える時間が伸びているということがサービス内で挙げられたため、この方法をオンラインでも実施しようという提案をした。実際には、Keep, Problem, Tryのそれぞれを書き出す時間を3分で行い、5分でそれぞれが書き出したKPTを張り出す。3分でKPTの項目ごとにグルーピングを行う。5分でProblemとTryをつなげ、7分でTryからActionを挙げる。この時間に則って進めることによって、時間が足りないという課題を完全な解決に向かうことができた。Actionに挙げた内容がそのまま実行できなかったことに関しては、KPTAを作成してそのままになっており、Actionについて見直すことができていなかったため、対策として、スプリント中にActionについて見直す機会を設けて、作成するだけでなく、見るということを習慣づけを行うことで解決に向かうことができた。

スプリントレビューに関しては、TAや教員の方に空いている時間をいただいて、プロジェクト時間外のレビューを行うことで、レビューを機会を増やした。この結果として、より多くのアドバイスを開発するアプリケーションに活かすことができた。サービス全般の問題である開発期間後半のタスク集中においては、スプリントバックログのタスクを開発メンバーのベロシティになるべく近づけて、1スプリントで丁度できるか、少し早く終わるように調整することで、少しは改善することはできたが、開発の終盤になってくると、タスクが集中してしまうことを回避することはできなかった。

以上のことから、KPTAの振り返りと振り返り後の行動についてとレビュー機会の減少の2つ

に関しては解決することはできたが、デイリースクラムの開始時刻の遅延や開発タスクの集中の2 つについては完全な解決に至ることはできなかった。デイリースクラムに関しては、開始時間の変更等を考慮すること、開発タスクの集中に関しては、メンバーのベロシティの把握とそれに対応した、タスクの重さの見極めを行うことで改善していけると考える。

(※文責: 山本雄平)

第 5 章 サービスの実装

5.1 システム構成

本サービスは、ビーコン、モバイルアプリケーション、Web アプリケーションの 3 つで構成されている。ビーコンは、スマートフォンが特定の UUID, Major 値, Minor 値を有したビーコンの電波を検知する際の発信機として使用する。本サービスでは、観光地等の各所でピースを取得するためのビーコンと、取得したピースを空港や駅に設置したパネルを検知するためのビーコンの 2 つの用途で使用する。ピースを取得するための各所に置いてあるビーコンは、UUID と Major 値を一致させており、Minor 値ごとに各所のピースを割り当てている。パネルを検知するために設置しているビーコンは、ピースを取得するためのビーコンとは異なる Major 値を設定し、Minor 値ごとに各所のパネルを割り当てている。モバイルアプリケーションでは、ピースを取得するためのビーコンとパネルを検知するためのビーコンの検出と、ピースを取得する、ピースにサインを施す、なぞなどを表示する等のユーザーが行う一通りの機能の提供を行う。Web アプリケーションでは、ユーザーが取得したピースを空港や駅に設置している想定の大きなパネルに当てはめてもらうためのアプリケーションを提供する。データの保存等にサーバーは実装せず、それぞれのモバイル端末上に実装されているデータベースを用いて、位置情報や取得時刻情報等のピース情報を保存する。

(※文責: 石澤大輔)

5.2 アプリケーション

5.2.1 iOS

開発環境

本サービスの iOS アプリケーション開発では、Apple 社が提供している、アプリケーション統合開発環境である、Xcode を用いた。Xcode は、Mac, iPhone, iPad, Apple Watch, Apple TV 向け App を開発するためのデベロッパツールセットである。[14] 我々がアプリケーション開発に用いた Xcode バージョンは、Xcode 12 以降で、最小 OS 要件は Intel ベースの Mac については、mac OS 10.15.4, Apple シリコン搭載の Mac については、mac OS Big Sur 11.0 となっている。我々の iOS アプリケーション開発メンバーは、Xcode 12 を用いて開発した。開発環境の整備という点で、開発をする上で問題となった点について紹介する。Xcode のプロジェクトファイル内の、Signing & Capabilities 項目の Team について、開発メンバーが全て別の Team ID を用いていたことで、開発中に Team ID をそれぞれが設定する必要があるが、Git でのソース管理の上でも差分が出てしまう点になっており問題となった。この問題の対処として、開発メンバー全員が、公立はこだて大学用の iOS Developer University Program に参加させていただき、開発において、同様の Team ID を用いることができるようになり、問題が解消した。また、開発メンバー全員が、公

立はこだて大学用の iOS Developer University Program に参加させていただいたことで、ほかにあった 2 つの問題も解消することができた。1 つ目の問題に、Apple Developer の無料の開発用アカウントで 1 週間に 7 つ以上のアプリケーションをビルドすると、” Your maximum App ID limit has been reached. You may create up to 10 App IDs every 7 days.” とエラーが出てしまい、同一アカウントではその 1 週間はビルド制限がかかって、ビルドができなくなる問題があった。また、2 つ目の問題に、Apple Developer の無料の開発用アカウントでは同一端末に 3 つまでのアプリケーションをビルドすることができないため、Apple が提供しているサンプルアプリケーションを動かしたり、参考書や Web サイトをもとに勉強してアプリケーションを作ったり、ほかのグループのアプリケーションをレビューする際にビルド制限によりできなくなってしまう等の問題があった。これらの 2 つの問題に対しても、公立はこだて大学用の iOS Developer University Program に参加させていただいたことで、解消することができ、開発を円滑に進めることができるようになったため有益であった。

(※文責: 村石拓海)

アーキテクチャ

本サービスの iOS アプリケーション開発で用いたアーキテクチャについて説明する。アーキテクチャとは、アプリケーションをより綺麗に、見やすいコードで実装するための設計書といえるものである。アーキテクチャを用いずにアプリケーション開発を行うと、機能の追加や、機能の修正、テストを書く際もやりやすくなり、チーム開発をする際にも、コードが複雑で属人化が進む等の問題が発生する可能性がある。そこで、アーキテクチャを考慮して実装することで、1 つのクラスが肥大化することで、ロジックが複雑化し、わかりづらいコードになってしまうことを防ぐことができる。また、コードの可読性を保ち、同じ処理をしているところは、処理を使い回すことで、効率的な開発をすることができる。我々のサービスでは、チーム開発を行い、それぞれのメンバーが役割分担をして開発を行う必要があったので、アーキテクチャは必須であった。そこで、本サービスの iOS アプリケーション開発では MVP アーキテクチャを用いた。我々がなぜ MVP アーキテクチャを採用して iOS アプリケーション開発をすることにしたのか、説明する。

まず、MVP アーキテクチャの概要について説明する。MVP アーキテクチャの MVP は Model View Presenter の略であり、最近の iOS アプリケーションでは多く使われているアーキテクチャである。Model と View と Presenter に分け、それぞれについて説明する。まず、Model は主にデータの型を指定したり、データの処理、データの情報が更新されたことを Presenter に通知する役割がある。次に、View は主に UI のレイアウトを指定したり、ユーザーがスマートフォンに対して行う、ボタンのタップや画面のスクロールといったアクションを Presenter に通知する役割がある。最後に Presenter は、Model と View の中間にあり、それぞれと関係を持っている。主に、Model の操作を行ったり、Model から受け取った通知をもとに、処理を行い、View を更新するように通知する。また、場合によっては View の状態を保持するなど、アプリケーション内のビジネスロジックを Presenter が担っている。前述した通り、Presenter は Model と View にそれぞれ繋がっているが、Model と View は直接繋がっておらず、データのやりとりは必ず Presenter を介して行う。このような仕組みのアーキテクチャが MVP アーキテクチャである。次に、アーキテク

チャの選定理由について説明する。iOS アプリケーション開発をする上でよく使われるアーキテクチャについて、ほかに、MVC アーキテクチャ、MVVM アーキテクチャ、Clean アーキテクチャがある。ここでは、それぞれのアーキテクチャの特徴について簡単に触れ、なぜこれらのアーキテクチャではなく、MVP アーキテクチャを採用することにしたのかについて説明する。

まず、MVC アーキテクチャとは Model View Controller の略であり、Apple の公式ドキュメントでも、推奨されているアーキテクチャである。iOS アプリケーション開発における MVC は Model と ViewController に分けられ、Model に主にデータ処理やビジネスロジックの役割を持たせ、ViewController に MVP アーキテクチャにおける View と同じ役割に追加して、ビジネスロジックを担う Model の操作についても役割を持っている。MVC アーキテクチャは、シンプルなアーキテクチャでわかりやすい反面、ViewController で処理する量が大きく、それにより ViewController のコードが肥大化しやすい。また、チーム開発する上では、役割分担をしやすくする必要があったため、UI のレイアウトについて記述する swift ファイルと、Model を操作する裏側の処理を記述する swift ファイルをわけ、それぞれが干渉しあわないように開発をしたかったため、MVC アーキテクチャは採用しなかった。

次に、MVVM アーキテクチャとは Model View ViewModel の略であり、MVP の Presenter が ViewModel に差し代わった形態になっており、MVP と同じように、ViewModel は Model と View に繋がっているが、Model と View は直接繋がっておらず、データのやりとりは必ず ViewModel を介して行う。MVVM と MVP は非常に似ているが、Model と View を媒介する ViewModel と Presenter に違いがある。Presenter は View の UI と Model のデータの更新や、ユーザーのアクションがあった時の更新について、更新があったタイミングで都度行う役割を持っているが、MVVM は View の UI と Model のデータが DataBinding と呼ばれる、データソースと UI の間を橋渡しする役割を持つものでつなげる役割を持つ。Model の状態が更新されると自動的に View も更新される仕組みで非常に優れたアーキテクチャである。しかしながら、iOS アプリケーションで MVVM アーキテクチャによる開発を行うためには、基本的に ReactiveX 系の RxSwift や ReactiveSwift といったライブラリを用いる必要があり、これらのライブラリは、学習コストが非常に高く、プロジェクト学習という iOS 開発に慣れた人のみで開発を行なっていないことや、開発期間が短く、その上ある程度完成させたものを作り上げなければならなかったため、用いると学習コストが高くなってしまふ MVVM アーキテクチャについては、今回は採用しないこととした。

最後に、Clean アーキテクチャとは、RxSwift を用いて、MVVM よりさらに発展した、層ごとに役割と責任を分離したアーキテクチャである。ビジネスロジック、フレームワーク、UI、データ全て、依存関係がなく実装でき、全てのテストが容易であるため、非常に優れたアーキテクチャである反面、MVVM アーキテクチャ以上に学習難易度が高く、今回のサービス開発では用いないこととした。これらの選定理由から、今回のサービス開発では、チーム開発がしやすい点と、学習コストがあまり高くない点の 2 点を主に踏まえ、MVP アーキテクチャを採用した。実際に MVP アーキテクチャを用いて、約 1 ヶ月開発をした感想としては、iOS アプリケーション開発の初学者であっても、アーキテクチャの理解をすることができ、チーム開発をする上でも、開発の役割分担をして進めることができたため、当初の予定通りアーキテクチャを活用することができて有用だった。しかし、開発時間が短かったこともあり、後半についてはアーキテクチャ通り、綺麗な実装に

なっていないこともあったため、その点で悔いが残った。

(※文責: 村石拓海)

ライブラリ導入

本サービスの iOS アプリケーションを開発するに当たって、以下の 3 つのライブラリを導入して開発を進める。

- SwiftLint

SwiftLint とは、Swift の静的解析ツールである。Swift 標準のコンパイラー以上にソースコードの品質を保つことができるだけでなく、Swift のコーディング規約にのっとったプログラムを書くようにサポートしてくれるツールである。また、SwiftLint は、設定ファイルである “.swiftlint.yml” ファイルを設定することで、カスタマイズすることができる。今回は、デフォルトの設定から、line_length プログラムの一行の長さを長くなり過ぎないように制限する設定の値について、150 にカスタマイズして使用した。デフォルトの設定であると、プロジェクトファイル生成時にデフォルトで生成されるコメントアウト等が引っかかってしまうので、カスタマイズすることで避けた。今後活かせる点として、変数名の文字数を制限する identifier_name という値のカスタマイズがある。identifier_name の最小文字数を制限する min_length について、デフォルトの設定では 2 になっていて、2 文字以上の制限になっているのだが、これにより、for 文で用いることの多い 1 文字の変数で警告が出る状態となっていた。identifier_name の min_length の値を 1 に設定することで、1 文字以上の変数を許容することで for 文等で変数を用いやすくなるので、今後 SwiftLint を用いる際はカスタマイズして使用したいと考える。

(※文責: 村石拓海)

- SwiftGifOrigin (1.7.0)

GIF 形式のアニメーションを表示させることができるライブラリ。通常の画像であれば、UIImage を用いて表示することができるが、GIF 形式のアニメーションを表示することができないので、UIImage の Extension を利用して、アニメーションを表示する。

- Sketch (3.0)

ライブラリ導入のみで、ペンツールでお絵描きができる OSS の Swift ライブラリ。できる機能は、色を変える、ペンの種類を変える、お絵かきの全消去、ペン・消しゴム・スタンプなどのモードを変更する、Undo・Redo を行う、画像の読み込みが行える。Storyboard 上でも、コード上でも編集が行える。

(※文責: 石澤大輔)

データモデル

本アプリケーションを実装するにあたって、Beacon モデル、Piece モデル、Riddle モデルを作成した。Beacon モデルはビーコンの情報を、Piece モデルは各ピースの情報を、Riddle モデルは

各ピースの位置に対応したなぞなぞの情報を格納するためのモデルである。以下に、モデルの内容と概要を示す。

Beacon モデル

- beaconId
ほかのモデルと対応させるための ID
- uuid
ビーコンの UUID を識別する
- major
ビーコンの Major 値を識別する
- minor
ビーコンの Minor 値を識別する

Piece モデル

- pieceId
ほかのモデルと対応させるための ID
- name
ピースを取得した場所の名前
- hasGot
ピースを取得したかを識別する
- date
ピースを取得した日時
- latitude
ピースを取得した場所の緯度
- longitude
ピースを取得した場所の経度
- area
ピースを取得した場所の地域
- image
ピースの画像, 編集後のピース画像も保存する

Riddle モデル

- riddleId
ほかのモデルと対応させるための ID
- title
なぞなぞのタイトル
- content
なぞなぞの内容
- hint
なぞなぞのヒント

- area

なぞなぞの答えの場所の地域, Piece モデルの area と対応している

(※文責: 石澤大輔)

ビーコン検知機能

本サービスに使用しているビーコン検知機能は, Apple 社が提供している, Apple Developer 向けのサンプルコードを参考にして実装を行った。フレームワークとして, CoreLocation を用いた。CoreLocation は, 位置情報関連の機能がパッケージされたフレームワークである。このフレームワークでは, 位置情報のトラッキング, 設定した領域の監視 (モニタリング), ビーコンの領域測定 (レンジング), 方位の測定の 4 つが大まかな機能として扱える。本サービスでは, ビーコンを使用し, ビーコンの範囲内, または範囲外の測定を行って扱う目的で使用する。このフレームワークを使用する前提として, Xcode 内に自動生成される Info.plist という実行ファイルに必要な構成を保存するためのファイルにある "Privacy - Location When In Use Usage Description" という Key を設定する。これは, ユーザのモバイル端末に位置情報の使用を本アプリケーションに許可するためのものであり, ビーコンは位置情報の使用を許可して扱うものなので設定する必要がある。本機能は, 大まかに 3 つの機能に分かれる。

1 つ目は, ビーコンを検知する関数である。ビーコンの UUID を指定し, その UUID に対して検知を行う。指定した UUID が検知された場合, そのビーコンの Major 値, Minor 値, rssi 値等の情報を配列に挿入する。その中で, UUID, Major 値, Minor 値をもとに個々のビーコンの識別を行う。その後 startMonitoring() を用いて対象の UUID のモニタリングを開始, 継続する。ビーコンのレンジングに関して, 本アプリケーションではビーコンが計測可能範囲内にあるかどうかのレンジングを行う。CoreLocation のビーコンのレンジングには, rssi 値をもとに対象のビーコンがどのくらいの距離にあるか測定できる機能があるが, 今回はピースを取得するためにビーコン検知機能を使用するため, 距離の測定が必要なく, アプリケーションへの負荷を軽くするためにも実装をしていない。

2 つ目は, ビーコンの検知を停止する機能である。これは画面遷移等でピースを取得する画面から異なる画面に移る時に必要な機能である。作成した関数の中に, 検知した UUID に対して stopMonitoring() を記述し, 各画面の遷移前後で用いられるように public の関数とした。

3 つ目は, 事前に作成したビーコンの UUID や Major 値等が保存されている JSON データから, 検知したビーコンの beaconID を取得する機能である。このアプリケーションでは, 各所に設置しているビーコンに beaconID という値を設定しており, その beaconID を使ってピースの情報を扱う。この関数では, 検知したビーコンの UUID, Major 値, Minor 値を取得し, この 3 つが一致している JSON データを検索し, beaconID を取得するという実装になっている。また, JSON ファイルからビーコンのデータをデコードするための実装は, BeaconsList というデコード, ゲッター専用の Swift ファイルを作成した。デコードの機能として, guard let 文で読み込むための JSON ファイル (ここでは, 「データモデル」で説明した Beacon モデルに合わせた beacons.json) のパスを String 型として確立する。その後, do-catch 文を用いて先ほど確立したパスを utf8 コードでデコードを行った。もし, デコードを行えなかった時は nil を返して, クラッシュしないような策

を作った。ゲッターの機能として、変数に先ほどのデコードの関数を用いて JSON ファイル内の情報を格納する。その後、「データモデル」で記述した Beacon モデルにある beaconId と、引数で指定した beaconId を filter を用いて照合し、返り値として返すようにした。

(※文責: 石澤大輔)

ピース取得画面

ピース取得画面は、ピース探索画面とピース取得済画面の計 2 つの画面から構成されている。この 2 つの画面では、フレームワークとして UIKit を用いて実装した。UIKit は、iOS および tvOS app のアプリケーション・ユーザインターフェースの構築と管理を行うのに必要なフレームワークである。UIKit に含まれているオブジェクトを用いて、コンテンツをスクリーンに表示し、操作を行い、システムとのインタラクションを管理する。また、オブジェクトとして、UserDefaults を用いてデータの保存を行った。UserDefaults とは、Apple 社が Apple Developer に向けて作られた iOS, macOS, Mac Catalyst, tvOS, watchOS のアプリケーションで扱うことのできるキーバリュ型 of 簡易的なデータベースである。軽量なデータを扱うのに特化していて、Data, String, Number (Int, Float, Double), Date, Array, Dictionary, Bool のいずれかを 100MB まで保存することができる。また、アプリケーションのタスクを削除してもデータベース内に保存されたデータは消えないが、アプリケーション自体を削除してしまうとデータは消える。ピース探索画面では、前述したビーコン検知機能を用いてピースを取得するための画面を実装した。ここではユーザーにピースを探してもらうために、UIText を用いて探してもらうためのテキスト表示と、「ライブラリ導入」に記述した SwiftGifOrigin を用いて本プロジェクトのマスコットキャラクターであるビーコンうさ (以下、うさ) を表示した。うさの表示に関して、GIF を用いてうさが歩いているように見える表示をし、うさの足元に UIImage を用いて道路の表示することで、テキストのほかにも歩くことを促すための工夫を施した。また、ピース探索画面からピース取得済画面に遷移するときに、遷移するためのトリガーが足りなかったため、Bool 変数と Timer() を用いて、1 秒ごとに遷移できるかどうかの判断をする関数を呼び出した。遷移時、ピースに対応している beaconID を取得し、ピース取得済画面に渡す実装をした。

ピース取得済画面は UIStoryboardSegue を用いて表示する。これは画面遷移を行う上で使われるオブジェクトで、遷移後の画面表示方法にもいくつか種類がある。その中の Push を用いて、下からピース取得済画面を表示する。この画面では、ピースを取得したということをユーザーに伝えるための画面を実装した。この画面で表示している情報は、UIText を用いたピースを取得したという表記とピースの名前、取得場所、取得日時、UIImage を用いたピースの表示である。ピースを取得した表記とピースの表示には、ユーザーがピースを取得したという達成感を得られるように表示を実装した。取得したピースの情報は、ユーザーがその場所で取得したピースを確認するために表示する。この画面に遷移した時、ViewDidLoad() 内に Date 関数を使い、日付を取得する。また、再度ピースを探索する際に 2 度取得してしまうことを防ぐために、UserDefaults を用いて取得した場所のピースは既に取得したということと、取得した日付を保存する。この画面からピース探索画面に戻る際、ViewWillDisappear() 内に、再度ピース探索画面で用いた Timer() を記述している関数を呼び出し、ピースを探索できるように実装した。

ピース一覧画面

まず、ピース一覧画面の概要について説明する。ピース一覧画面は現在取得しているピースを一覧として見て確認することのできる画面であり、ピースを所持していることを示す、有色のピース部をタップすると、ピース確認画面に遷移し、現在持っているピースのシリアルナンバーや、ピースの見た目、ピースを取得した日時、場所、そのピースがあるエリアについて、函館を「函館駅前・大門」、「ベイエリア」、「元町・函館山」、「五稜郭」、「湯の川」、「美原・桔梗」、「七重・大沼」の7つのカテゴリに大きくわけ、ピースの場所について、エリアとして把握することもできる。また、ピース未所持であることを示す、灰色部をタップすると、ピースのシリアルナンバーや、ピースの見た目、ピースのエリアについてわかるようになっている。ピース一覧画面は以下の図 5.1 のようになっており、画面を開いた時にわかりやすいように画面全体にパネルを表示し、かつ子どもが楽しめるようなポップな UI になるように心がけて作成した。次に、ピース一覧画面の実装につ

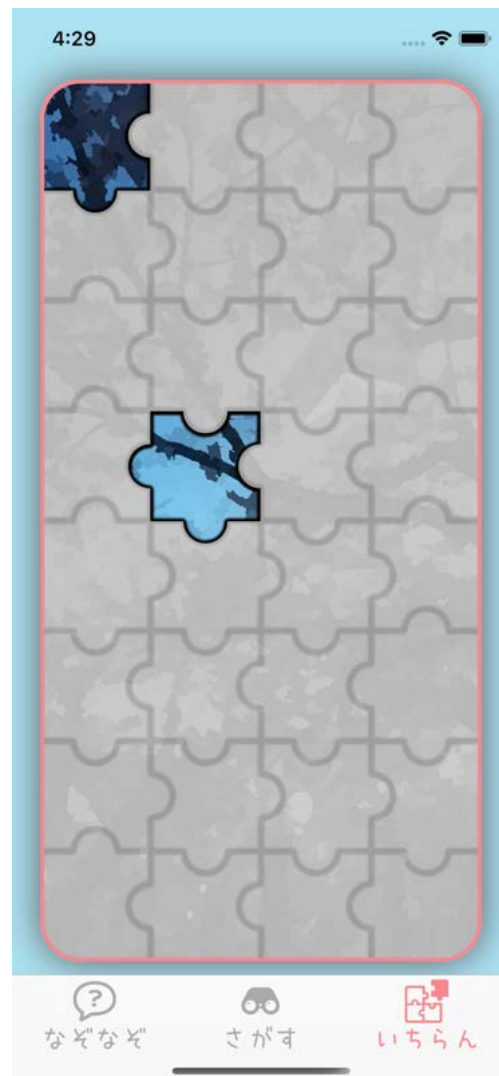


図 5.1 ピース一覧画面

いて説明する。UIKit フレームワークを利用しており、複数の View を重ねることで描画している。まず、背景色の水色を描画するために、UIView の Piece All View を作り、Piece All View の背景色に青のテーマカラーを適用した。背景色の青色が下部の UITabBar の箇所に干渉しないように、Piece All View は SafeArea を含む、UITabBar 以外の画面全体に配置した。次に、パズルの背景画像を描画するために、パズル全体の範囲を指定するための、大枠となる Piece All Image Area View を UIView を用いて作った。Piece All Image Area View のコンストレイントは、上部は、SafeArea の縁から 10px、下部は、画面下部の UITabBar から 10px の余白を開け、Piece All Image Area View の縦横比が 2:1 になるようにした。これは、パズルについて、縦に 8 ピース、横に 4 ピースの合計 32 ピースで構成されていることから、Piece All Image Area View の縦横比を 2:1 にすることで、Piece All Image Area View の上にパズルを描画するとちょうど良い画面比で描画できるようになっている。これにより、どの画面サイズの iPhone を使っていても、パズルがスマートフォンの画面で可能な限り大きく表示され、なおかつピース一覧画面での UI 崩れがなく、ユーザーにとって使いやすいように綺麗に描画することを可能にした。Piece All Image Area View は iPhone の様々な画面サイズに応じて柔軟にリサイズするように対応しているので、パズルに使われているパズルの画像や、一つ一つのピースといった要素は、Piece All Image Area View に依存して、Piece All Image Area View との相対的なサイズや相対的な座標で描画するようにした。

(※文責: 村石拓海)

ピース確認画面

ピース確認画面は 1 画面のみで構成されている。また、ピース確認画面は、ピースを所持している場合のピース確認画面とピースを所持していない場合のピース確認画面の 2 種類がある。まず、ピースを所持している場合のピース確認画面について説明する。ピースを所持している場合のピース確認画面は、ピース一覧画面の有色のピース部をタップすることで遷移できる。ピースを所持している場合のピース確認画面では、取得したピースのシリアルナンバーと、ピースの見え目、ピースを取得した日時、ピースのエリア、場所について確認することができる。ピースを所持している場合のピース確認画面は以下のような画面になっており、ピースのシリアルナンバーについては、上部の中央に表示し、ピースの見え目については、画面の中央にわかりやすく大きく表示するようになっている。中央のピースの画像の下部には、「サインをかく」と表示した、ボタンを作り、ユーザーがボタンをタップすることで、ピースサイン画面に遷移することができ、取得したピースに手書きでサインを書くことができるようになっている。また、ピースを取得した日時、ピースのエリア、場所については「サインをかく」と表示されたボタンの下部に並べて表示している。次に、ピースを所持していない場合のピース確認画面について説明する。ピースを所持していない場合のピース確認画面は、ピース一覧画面の灰色部のピース部をタップすることで遷移できる。ピースを所持していない場合のピース確認画面では、ピースのシリアルナンバーや、ピースの見え目、ピースのエリアについて確認することができる。

(※文責: 村石拓海)

ピースサイン画面

ピースサイン画面は、この画面一つのみで構成されており、ピース確認画面より画面遷移をして画面を表示する。この画面では、取得したピースにサインを残し、それぞれの思い出を書き込むことができる画面である。フレームワークとして UIKit を、ライブラリとして「ライブラリ導入」で説明した Sketch を導入し、実装した。画面の構成として、ペンでピースにサインを書く機能を 4 つの UIButton で、一つ前のピース詳細画面にもどる機能とピースへのサインの書き込みが完了した時にピース詳細画面に進む機能をそれぞれ UIButton で、実際にピースにサインを書き込むピース画像を UIImage で表示した。ピースにサインを書く機能として、Sketch のペンツール、消しゴムツール、書いたサインを全消しするツール、Undo ツールを採用し、それぞれを UIButton で実装した。これは、多くの機能を実装して表現を豊かにするよりも、誰でも直感的に扱えて、なおかつ思い出をサインとして書き込むことが簡単にできるということを重視していた。ペンツールでは、複数回 UIButton をタップすることでペンの色を黒、赤、青、緑の 4 種類に変えることができる機能を追加した。書けるペンの色が変わるごとに、タップする UIButton の羽ペンの画像の色をそれぞれの対応した色に変えられるように実装し、視覚的に色がわかるようにした。一つ前のピース詳細画面に戻る機能は、ピースに編集したものを破棄してピース詳細画面に戻るようにした。完了した時にピース詳細画面に進む機能は、ピースに編集したものを「データモデル」の Piece モデルにある image に保存してピース詳細画面に進むようにした。実際にピースにサインを書き込む画像は、ピース一覧画面で選択したピースに書き込めるようにした。ピースをにはめ込む画像には、個々の端末にあるフォトライブラリから選択する、または実際にカメラで撮影した後すぐにはめ込めるように実装した。また、ピースを表示する時、極力大きく映してサインを書きやすいようにするために、対応している iPhone の各端末の横幅の最大値をピースの横幅に合わせて表示し、縦横比を 1:1 にした。これにより、正方形のピースが画面の中央に表示される。

(※文責: 石澤大輔)

チュートリアル画面

チュートリアル画面は、この画面のみで構成されている。「ピース取得画面」で説明したピース取得画面の中にあるピース探索画面から初回起動時に画面遷移する。この画面では、本アプリケーションを初めて使用するユーザーに向けて、本アプリケーションはどのような風に遊ぶのかということ伝える。本アプリケーションを起動するときは、まず始めにピース探索画面が必ず表示される。その後、ピース探索画面の StoryBoard に Custom Class で連携されている Swift ファイルに、初回起動時であればチュートリアル画面に遷移するようにコーディングする。初回起動時であるかどうかの判断は、UserDefaults のデータベースに Bool 型の変数を作成して、初回であれば false、初回以降であれば true となるようにして実装した。チュートリアル画面では、うさが画面下部に表示している UITabBar のなぞなぞ部分を指さして、「ここをおしてなぞなぞをとくとピースのありかがわかるよ!!」という風に吹き出しを出して説明している。うさ、吹き出しは UIImage を用いて表示、テキストは UITextView を用いて表示した。また、ピース探索画面の UIView の上に不透明度 100 % のチュートリアル画面の UIView を置いてしまうと、何を指している画面かわからな

なくなってしまったので、不透明度を 50 %にし、アクセントカラーである黄色に設定した。加えて、なぞなぞ部分に白くて丸い UIView を重ねがけすることで、より視認性を高く実装した。戻るボタンとして、画面のどこかをタップすると一つ前のピース探索画面に遷移するようになっている。

(※文責: 石澤大輔)

ピース送信画面

ピース送信画面は 1 画面で構成されている。ピースを送信する先である、空港や駅に設置されたパズルパネル用のビーコンを iOS 端末で検知すると、自動でピース送信画面に遷移するようになっている。パズルパネル用のビーコンの UUID, Major 値, Minor 値によって、ビーコン検知機能と同様の方法によってビーコンを検知できるようになっている。パズルパネル用のビーコンを検知してピース送信画面に遷移すると、画面の上部に取得したピースの画像が表示され、ピースを送信するか確認するための「ピース送信」と表示されたボタンが下部に表示される。ボタンを押すことで実際に取得したピースをパズルパネルに送信することができる。ピース送信時は、上部のピースの画像が、上部にフェードアウトしていくことで、ピースが送信されたように見せている。コンストレイントについては、画面上部のピースの画像については、UIImage を中央に画像が表示されるように配置し、画面下部の「ピース送信」と表示されたボタンについては、UIButton を中央に表示し、titleLabel の Text を「ピース送信」と指定し、UIButton の角を丸くし、影をつけることで、子どもでも楽しめそうかつ立体的な見た目になるようにした。ピース送信時のピース画像の移動については UIImageView のクラスメソッドである、animate メソッドを用いて、あたかも上部にフェードアウトしているようにアニメーションを実現した。また、アニメーションの速度を加速度的に増えるようにすることで、よりピースが送信されている様子がユーザーに伝わるように工夫した。

(※文責: 村石拓海)

5.2.2 Android

開発環境

本サービスの Android アプリケーション開発では、Google が提供している Android 向けアプリケーション統合開発環境である Android Studio を用いた。JetBrains 社の開発した IntelliJ IDEA をベースに、Android アプリケーション作成の生産性を高めるために最適化されている。Android Studio は Gradle ベースのビルドシステム、高速なエミュレータ、優秀な Lint ツール、C++ と NDK のサポート、Google Cloud Platform の組み込みなどの多数の機能をサポートしている [15]。Android 開発には Android Studio ではなく Eclipse でも開発が可能であるが、利点がないため Android Studio を使用することにした。Android Studio のバージョンは開発開始時の最新バージョンが 4.1 だったことから 4.1 で開発した。また、サービスをまたいでコードレビューを行いやすくするため、サービス間でバージョンを 4.1 に統一し、最新のバージョンが公開されても更新せずに開発を続けることで決定した。

プログラミング言語は Kotlin を使用した。Android 開発は主に Java か Kotlin が用いられるこ

とがほとんどであるが、Kotlin には Java の完全な互換性があること、言語の仕様上 Java よりもコードの安全性が高まること、何より Java に比べてコードの記述量が減ることから Kotlin を採用した。

バージョンについては、開発開始時に最新であった 1.4.10 を用いた。ビルドツールには Gradle を使用した。バージョンは開発開始時点で最新のバージョンであった、4.1.1 を使用した。

また、最小 SDK バージョンは Android 8 (API Level 26) これは、できるだけ最新のバージョンを使おうとしたこと、大学側から借りた Android 端末の最新バージョンが Android 8 であったことが理由である。加えて、Android Studio 側でプロジェクトの新規作成をする際に表示されるバージョンのシェア率が 60.8% 以上の端末が対応ということであり、実際にサービスを Google Play にて配信する際に十分な数の端末に対応できるという判断からこのバージョンに決まった。

(※文責: 宮田悠治)

アプリケーション・アーキテクチャ

本サービスではアプリケーションのアーキテクチャとして MVVM アーキテクチャを採用した。これは 1 つのファイルが大きくなることを防ぐことや、Live Data を使いやすくするなど、様々な理由があるが、何より公式の推奨しているアーキテクチャが MVVM であること [16]、インターネット上で公開されているコードの多くが MVVM で実装されていることが大きい。また、Android 開発の MVVM には Data Binding という機能を使用しているものと、使用していないものがあるが、本サービスでは Data Binding を使用する方向で決定した。しかし本サービスの開発において、開発期間の逼迫からできるだけ早く実装することが必要になったこと、Android 開発初學者ゆえの MVVM、Data Binding についての学習時間の不足から、MVVM を模したファイル構造ではあるが、MVVM とは言いにくい実装部分が存在しているなど、本質的な MVVM での実装は間に合っておらず、Data Binding についても使用することなくボタンなどを実装しているという現状である。今後、Observe するデータの設定や Data Binding を用いてのボタンなどの実装をしていく予定である。

(※文責: 宮田悠治)

ライブラリ導入

本アプリケーションの開発には以下のようなライブラリを導入した。導入したライブラリ名とバージョン概要を示す。

- Fragment-ktx (1.2.5)
フラグメントの扱いを用意にするライブラリである [17].
- Live Data (1.1.1)
データの監視を行うためのライブラリである。ライフサイクルに変化が生じた際に Observer に通知をしてくれるため、画面の縦横が切り替わった際のデータ更新などをしてくれる [18].
- AltBeacon (2+)

Android デバイスで iOS のようにビーコンを使用できるようにするライブラリである [19]. バージョン表記の 2+ とはバージョン 2 以上の最新を読み込むための設定である. こうなった経緯としては, このライブラリ自体に既知の不具合が多数見受けられ, 本アプリケーションでも一部の不具合が見受けられることから, 楽観的ではあるが不具合の修正が進んで行くだろう最新のバージョンを持ってくるように配慮した結果である.

- Localbroadcastmanager (1.0.0)
AltBeacon を実装する際に読み込む必要のあるライブラリである. アプリケーション全体のイベントバス [20].
- Gson (2.8.6)
Android で JSON を扱えるようにできるライブラリである. JSON のパースを行うことができる [24].
- RecyclerView (1.0.0)
リストを表示するためのライブラリである. 大規模なデータセットを簡単かつ効率的に表示できる [21].
- Card View (1.0.0)
RecyclerView のリストをカードのように表示するためのライブラリである. プラットフォーム全体で一貫したデザインのカード内に情報を表示するための簡単な方法である [22].
- Glide (4.11.0)
画像やアニメーション GIF の表示をするためのライブラリである [23].

(※文責: 宮田悠治)

ビーコン検知機能

本サービスでのビーコン検知機能は上も示してある AltBeacon というライブラリを用いて実装した. このライブラリはアプリケーションのフォアグラウンド及びバックグラウンドでのビーコンの検知, 検知したビーコンの監視, Android 端末をビーコンとするなどの機能が備わっている. AltBeacon は領域内にビーコンがあるかどうか探索することをモニタリングと呼び, 領域内のビーコンを監視することをレンジングと呼ぶため以下そのように記載する. また, 本ライブラリを使用するにあたって AndroidManifest.xml というファイル内に位置情報の使用を許可するための記述すること, 及びアプリケーション起動の際にユーザーに対して位置情報の権限を催促する通知の実装を行った. また本サービスの運用, 及びそのデモにおいて使用したビーコンが Apple 社の提供する iBeacon であったため, それを検出するために BeaconParser という項目を iBeacon に対応するよう設定をした.

本アプリケーションではビーコンのレンジングに際して, モニタリングを実行し, 領域内にビーコンを発見した場合, レンジングに移行するように実装した. 本アプリケーションではビーコンをピースとして扱っているが, これの判別にはビーコン固有の UUID と Major 値, Minor 値を”,”で区切り連結した文字列を用いた. ピースとして扱うビーコンに関してはこの文字列を事前に strings.xml 内にそれぞれ StringArray として記載しておき, 検知したビーコンのものと比較する

ことで判別を行った。UUID, Major 値, Minor 値の対応については, StringArray の添字を対応させることで合わせている。

(※文責: 宮田悠治)

ピース探索, 取得画面

まず, 本サービスを担う主な 3 つの画面である, 本節で示すピース探索画面と, 後に記すなぞなぞ一覧画面とピース一覧画面を下部ナビゲーションバーでそれぞれ遷移できるように実装した。ピース探索画面はアプリケーションの起動時に最初に表示される画面である。ピース探索画面の実装について実装した内容を 3 つほど示す。

1 つ目に, ユーザビリティの向上のために, レンズングしている際のビーコンとの距離に応じて画面中央のマスコットキャラクターのセリフが変化するような実装をする必要があった。デモ前提の実装として, AltBeacon のレンズングの際, 予めライブラリで用意されている distance という変数を用いてビーコンとの距離を計測し, その距離が 3m 以内だった場合, ビーコンが近くにあるとの通知をし, 1m 以内になった時点でピースを獲得できるように実装した。しかし課題としてビーコンの情報が更新される頻度に問題があり, 人が素早く動くとき情報の更新が間に合わず, 1m 以内にビーコンがあるにもかかわらずビーコンの取得ができないなどの事例が起こる。これをユーザー体験が損なわれかねないが仕様であると割り切る。もしくは, バッテリーや処理の観点からライブラリから非推奨とされているが, レンズングの頻度をあげるのかなどが今後の議論すべき内容である。

2 つ目に, ビーコンを探索するための画面以外の画面ではビーコンの探索もモニタリングもしないという実装をする必要があった。これは探索する際のみ探索画面を開くことでバッテリーの節約や, ユーザーの意図したタイミングでピース獲得の画面に遷移させるためである。したがって, その画面以外に遷移する際にはモニタリングとレンズングを解除するように実装した。最後に, 画面デザインについては, 画面中央にプロジェクトのオリジナルキャラクターであるうさが可愛らしく歩いているアニメーション GIF を Glide ライブラリを用いて実装した。ビーコン検知を通知するセリフについては, 左上に吹き出しを用意し, オリジナルキャラクターが喋っているように演出した。

最後に, ピース取得画面について示す。ピースを取得すると, ピースを取得したという画面に遷移する。このときアプリケーション内では対象のピースを取得した旨が保存される。具体的には, Android アプリケーションの機能に標準で用意されている, Shared Preferences というデバイスにデータを保存するための仕組みを利用して保存されている。Shared Preferences は String 型や Int 型, Boolean 型を保存できる。本アプリケーションではビーコンの判別に使われる「UUID, Major 値, Minor 値」という文字列とそのビーコンが取得されているのかどうかを保存する Boolean 型のデータ, 加えて, ビーコンを取得したタイミングで日時を取得し, それを保存するように実装した。Room ライブラリという Android で SQLite を簡単に扱えるライブラリを用いてデータベースによる管理も実装方法として考えられていたが, 習得コスト, 開発期間短さという観点から, Shared Preferences で実装することで決定した。

(※文責: 宮田悠治)

なぞなぞ一覧画面

なぞなぞ一覧については、strings.xml 内になぞなぞ用の StringArray を実装することで端末内に保存した。またピースのビーコンの StringArray と添字を合わせることで対応を取らせた。具体的になぞなぞ用の StringArray はビーコンの場所を濁して示した quiz_titles と、なぞなぞの本文になっている quiz_contents, ほとんど答えを示すヒントが記載されている quiz_hints, ピースのビーコンの置かれている場所の函館でのエリア管理用カテゴリーの quiz_category, エリアそのものが記載されている quiz_area を用意した。エリア管理用のカテゴリーとは実際にサービス運用にあたって、エリアでの絞り込みを行う際の便宜上カテゴリーというように命名した。

画面構成としては RecyclerView View に Card View を適応して、quiz_titles を表示させるように実装した。わかりやすく示すと、quiz_title がカードに書かれているようにリストで表示されている。加えて、それがタップされた際に、ダイアログを表示させそのなぞなぞの詳細が表示されるように実装している。画面について詳しくは図 3.9 を参照されたい。

(※文責: 宮田悠治)

ピース一覧画面

まだ実装されていない機能であるため、今後実装することが目標である。ピース一覧画面はピースをはめ込むはめるパズルの台紙のような画像を中央に配置し、取得したピースに関しては ImageButton という画像をはめ込むことができるボタンを台紙の上に重ねて配置することで実現させる。未取得の画像に関しては、画像は配置されているが、非表示化することで見えないように実装する。

(※文責: 宮田悠治)

ピース送信画面

ピース送信画面はピース送信確認画面と、ピース送信画面で構成されている。まず、パネル用のビーコンを用意しておき、パネルのビーコンとして UUID, Major 値, Minor 値を登録しておく。ビーコンを検知し、それがパネルだった場合にピース送信画面に遷移するように実装されている。デモ版に限っては、デバック用としてピース探索画面に透明なボタンを用意しておきそこをタップした際にも遷移するように実装されている。ピースを送信数画面についてはまず送信前の確認画面を用意し、ピースを送信すると書かれたボタンを押すことでピース送信画面に遷移するように実装した。ピース送信画面では送信するピースが画面中央に配置され、それをタップすることでピースが画面上部の方に動いていくようなアニメーションを実装している。スマートフォンから送信されるような演出のためのものである。具体的には、画像を上方向に画面から十分にはみ出すように動かす動作の xml ファイルをタップされることをフラグにして読み込んでいる。

(※文責: 宮田悠治)

サイン書き込み画面

まだ実装されていない機能であるため今後実装することが目標である。取得したピースにサインを残す機能の実装については、Canvas というライブラリを導入し、Android 端末で手書きの字やイラストをかけるように実装する。描かれたサインなどについては Room ライブラリを導入し保存することを予定している。ピースを取得した際に表示されるピース取得画面、もしくはピース一覧の画面から遷移するように実装する。

(※文責: 宮田悠治)

チュートリアル画面

まだ実装されていない機能であるため今後実装することが目標である。チュートリアル画面の実装については、SharedPreferences 内に Boolean 型のフラグを用意して初回起動なのか、そうではないのかを管理し、初回起動だった場合、表示するように実装する。画面構成としては、起動時に表示されるピース探索画面において、その画面の上に重ねるように画面を表示する。下部ナビゲーションに表示されている画面の一覧を示しながら機能の説明を表示する。また、画面がタップされたらチュートリアル用の画面を非表示にするように実装する。

(※文責: 宮田悠治)

5.2.3 Web アプリケーション

開発環境

本サービスでは、パズルパネルの実装に Web アプリケーションを用いており、その開発には Microsoft が提供しているオープンソースのコードエディターである、Visual Studio Code を用いた。Visual Studio Code は、クロスプラットフォームで動作するソフトウェアであり、主に Web アプリケーションの開発で多く用いられている。実装は Microsoft が開発したオープンソースの JavaScript スーパーセットである TypeScript を用い、ビルド環境にはサーバーサイドで JavaScript を実行することができる、Node.js 12.19.0 を用いた。さらに、コードの整合性と保守性を保つため、静的検証ツールである ESLint、可読性を確保するために自動整形ツールである Prettier を採用した。

(※文責: 久米田羽月)

技術選定

本サービスにおけるパズルパネルは、ユーザーがピースをはめる際に達成感を得られるよう、アニメーションやエフェクトを多く用いる。そのため、グラフィカルな表現を行うのが比較的得意であること、サーバーサイドの実装や、データベースとの連携が得意である Web プラットフォームを用い、その上でゲームライブラリを採用するのが適当だと判断した。

TypeScript の採用理由として、大きく 2 つの理由が挙げられる。1 つはキーワードなどの入力

補完のサポートが受けられること、もう1つはJavaScriptにはない型情報があることで、コードの保守性を高めることができ、より堅牢なシステムを作ることができるためである。

(※文責: 久米田羽月)

ライブラリ導入

本アプリケーションでは以下のライブラリを導入した。導入したライブラリ名とバージョン概要を示す。

- phina.js(0.2.3) phina.js[25] は国内のコミュニティで開発されている、オープンソースのゲームライブラリである。アニメーションなどが数行のコードで実装できるほか、ライブラリの知識があるメンバーがいたため、採用した。

(※文責: 久米田羽月)

待機時の画面

待機時は、ピースをはめ込めるパズルの台紙のような画像が配置してあり、ユーザーによってすでにはめられたピースが台紙の上に表示されている。台紙およびピースは画像を表示させるためのSpriteクラスのインスタンスとして実装している。パズルの裏表は定期的に切り替わるようになっており、パズル面と裏面の写真とサインが書かれている面が交互に表示されるようになっている。

現在はモック状態であり、実際に裏表の切り替えなどはできないため、今後の開発で対応する予定である。

(※文責: 久米田羽月)

ピースが送信された場合の画面

ユーザーがアプリケーションを操作して、ピースを送信することを選択した場合、ピースに含まれる取得日、取得した地区、写真、サインなどの情報がアプリケーションからパネル側のWebアプリケーションに送信される。データの送信には、BLE通信を用いる予定である。ピースの情報が送信されたことを検知すると、ピースの情報をデータベースに登録し、対応するピースの画像を所定の位置に表示させる。この時、画面下からTweenを用いてアニメーションしながら所定の位置に移動し、その後Labelクラスを使い、「おめでとう！」と表示させる。

(※文責: 久米田羽月)

5.3 デザイン

5.3.1 アプリケーションアイコン

まずサービスの顔となるアプリケーションアイコンを決定するためにグループメンバー全員で案を出し合った。グループメンバー各々がアイコン案のスケッチを描いてそれらをレビューした。それを何度か繰り返し、サービスの特徴を捉えているかつグループメンバーから一番賛同を得られたものをアプリケーションアイコンとして決定した。サービスの特徴としては、「家族」「親子」「子どもと大人」「パズル」「ピース」「思い出」が挙げられる。

アイコン案にはサービスの特徴を表すパズルのピースや親子をはじめ、親子が手をつないでいる様子が描かれている案やプロジェクトのオリジナルキャラクターであるうさが描かれている案もあった。また、サービスタイトルであるでももに！をパズルのピースの中に収めたものや、でももに！のタイポグラフィもあった。タイポグラフィに関しては、サービスの対象が子どもがいる家族向けであることから、堅苦しい印象ではなく柔らかい印象をもたせるために、文字の角を丸くした案があった。何度かスケッチを重ねていく中で、プロジェクトのオリジナルキャラクターであるうさとサービスの特徴を表すものをうまく組み合わせる案はどうか、という意見が挙がった。子ども向けということもあり、うさの可愛らしさとサービスの特徴を表す何かを組み合わせることでよりサービスへの親しみやすさが増すと考えられたからである。その意見に賛同が得られていたため、先の意見を踏まえながらスケッチを行った。うさにパズルのピースを持たせた案、うさにスマートフォンを持たせた案、函館を背景にうさにパズルのピースを持たせた案などが挙げられた。その中のうさにパズルのピースを持たせた一案がグループメンバー間で好評であったため、これをブラッシュアップすることにした。うさの手にはパズルのピースではなくスマートフォンを持たせ、うさ



図 5.2 アイコン案



図 5.3 アイコンラフ画

をパズルのピースから体を乗り出しているものがアイコン案として決定した。5.3 その後、決まったアイコン案をもとに CLIP STUDIO PAINT を用いてアプリケーションアイコンを描きあげた。配色については「テーマカラー」で記述している。

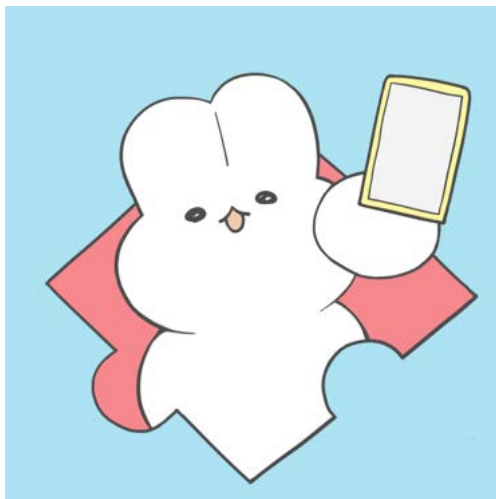


図 5.4 アプリケーションアイコン

(※文責: 荻ノ沢実佑)

5.3.2 テーマカラー

アプリケーションを製作するにあたって、サービスのテーマカラーを設定することに決定した。テーマカラーの決定において、サービスの雰囲気と主な対象の子どものいる家族に一致するような配色になるようにした。まず色を決める前に、大まかな雰囲気を決定するために、トーンをどのあたりにするのかを決めた。子ども向けということもあり、明るさ、楽しさ、元気さ、柔らかさなどが感じられるトーンが良いと考えた。したがって、ビビットトーン、ブライトトーン、ペイルトーンといった彩度や明度が高いトーンを中心に色を選定していくことにした。

次にどの色を使用するかについて考えた。あまりに多くの色を使用するとごちゃごちゃとした印象となり、ユーザーにとって使いにくいデザインになってしまうため、使用する色の数を絞ることにした。メインカラー、サブカラー、アクセントカラーの3色でアプリケーションをデザインすることにした。使用する色の候補としては、明るさや元気さを感じさせる赤、オレンジ、ピンクなどの暖色を始め、楽しさ、にぎやかさを感じさせる黄色、爽やかさやを感じさせる緑や水色など様々な色が挙げられた。なかなか絞り切ることができなかつたため、「3色だけでセンスのいい色」という書籍を参考にした [26]。まず、その中の2つのカラーバリエーションを参考にアプリケーションアイコンを配色した。1つ目は、クリーム・パウダー (#fff7c9)、キャンディー・ピンク (#f7c4d4)、アイルランド・ブルー (#c0e4f2) の3色を使用した。明るめの優しい柔らかさを感じるパステルな色使いで、親しみやすさを持ってもらえるようにこの3色を選んだ。メインカラーをアイルランド・ブルー、サブカラーをクリーム・パウダー、アクセントカラーをキャンディー・ピンクにしている。2つ目は、ペリカン・ピンク (#f3adc5)、バニラ・イエロー (#fff2b2)、アイス・ミント (#a7d9dd) の3色を使用した。先ほどの配色より濃い目の色合いになっており、ポップさや楽しさが感じられる色合いになっている。メインカラーをアイス・ミント、サブカラーをバニラ・イエロー、アクセントカラーをペリカン・ピンクにしている。色合いの明るさや柔らかさの観点で、1つ目の配色に決定して実装を進めていくことに決定した。しかし、実装を進めていく中

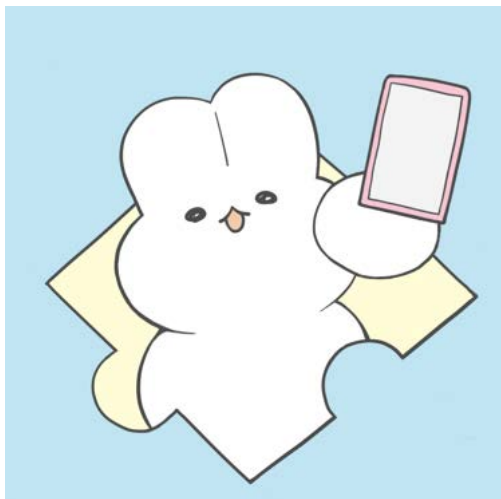


図 5.5 1つ目の色を適用したアプリケーションアイコン

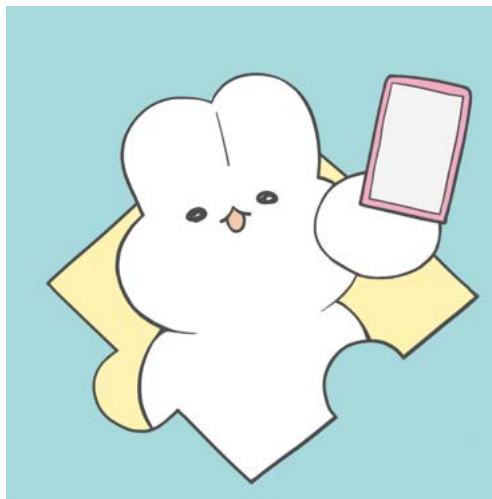


図 5.6 2つ目の色を適用したアプリケーションアイコン

で、全体的に色合いが薄いため見づらく視認性が悪いことが判明した。そのため、全体的に彩度を上げ色の調整を図った。最終的に、青 (#ace2f2)、ピンク (#f7888e)、黄 (#fff7a1) となり、メインカラーを青、サブカラーをピンク、アクセントカラーを黄と決定した。テーマカラーを適用したアプリケーションアイコンは「アプリアイコン」の図 5.4 に記載されている。

(※文責: 荻ノ沢実佑)

5.3.3 フォント

iOS, Android の両アプリケーションに統一感を持たせるために、サービスの雰囲気と合致しているフォントを設定することにした。サービスの主な対象が子どもがいる家族であり、子どもにも親しみを持ってもらえたり、楽しさ、柔らかさなどが感じられたりするようなフォントを選定した。候補に挙がったのは以下の 8 つである。8 つとも無料で利用可能なフリーのフォントで商用利用も可能なフォントとなっている。

1. じゅうちょうフォント

これは、太めのサインペンで書かれた温かみのあるゴシック体の手書きフォントとなっている [27]。太めも文字で角が丸みを帯びているフォントであるため、柔らかさや可愛らしさなどで親しみやすさが感じられるフォントになっている。しかし、太字であるため可読性にやや欠ける可能性がある。強調したい文字など部分的に使用するのがよいと考えられる。

2. ほにゃ字

これは、可愛い絵文字や文字を表現できる手書きフリーフォントとなっている [28]。名前の通り、字体がほにゃっとしており、柔らかさや可愛らしさを感じられるフォントとなっている。小さいフォントサイズでも可読性が高く保たれているところが特徴的となっている。

3. ゆるもじ

これは、手書き風の日本語ビットマップフォントで、ゲーム風なフォントになっている [29]. ゲーム風なフォントになっているため、わくわく感や楽しさが感じられるフォントになっている. 手書き風のフォントであるため可愛らしさも兼ね備えている.

4. ニコカフォント

これは、ニコ角 [30] フォントの仮名文字を、かわいいポップな場面で利用できるようにアレンジされた丸文字フォントとなっている [31]. やや字体が可読性に欠けるが、丸さと適度な角でポップで可愛らしさとカッコよさを感じられるフォントである.

5. プぷプ

これは、曲線が美しくディテールが凝った丸っこくてかわいいフォントになっている [32]. このフォントは字体が特徴的でポップさと可愛らしさが感じられる. 横線と縦線の太さが雲のようにもくもくとした緩急があり、柔らかさも感じられるフォントになっている.

6. JK ゴシック M

これは女子高生をイメージした、女子高生風のゴシックフォントになっている [33]. 角ゴシック体でありながら、少し太めの字体で少し崩れた形が可愛らしさやポップを感じられるフォントになっている.

7. うずらフォント

これは、小さくて丸っこいような少し角ばっているような、可愛い手書風フォントになっている. マーカーで書かれたような太さがあり、程よい太さで可読性が高いところも特徴的である [34]. 手書き風であるため、温かみを感じる. また、崩れた形の字体が可愛らしさと親しみやすさを醸し出している.

8. kawaii 手書き文字

これはちょっぴりクセのあるかわいい手書きペン字をひとつひとつフォント化したものになっている [35]. フォントのもとになる字が、ペンで書かれたフォントであるため、読みやすい太さになっている. 角が丸みを帯びており、可愛らしさ、柔らかさが感じられるフォントになっている.

この中の、うずらフォントをサービスのフォントとして使用することにした. 理由としては、以下の3つが挙げられる. 丸すぎず角ばりすぎない形が、可愛らしさを強調していること. 手書き風に適度に崩された形が親しみやすさを感じさせること. 文字の太さが程よく可読性も高いこと. 以上の3つの理由からこのフォントを iOS, Android の両アプリケーションで使用することに決定した.

(※文責: 荻ノ沢実佑)

5.3.4 ピース探索, 取得画面

この画面は、アプリケーションが起動されると一番最初に開かれる画面である. 基本的にはこの画面を見てピースを探してもらうため、ユーザーにとって見やすくかつ飽きのこないような画面にする必要がある. まず、ユーザーに函館のまちを歩いてもらいピースを探してもらう必要があるため、「まちをあるいてみよう!!」というテキストを画面に表示させ、ユーザーにまちを歩いて

もらうように促した。また、テキストだけでなくうさが歩いている GIF 画像を表示させることでユーザーがより楽しめるような画面にした。ビーコンとスマートフォンの距離によって吹き出しから表示されるテキストのセリフを変更することで、同じ画面を見ているも飽きさせない工夫もしている。実際の画面は図 3.6 と図 3.7 に記載されている。

(※文責: 萩ノ沢実佑)

5.3.5 ピース一覧画面

この画面は、ユーザーが取得したピースを一覧で確認できる画面である。ただ単にリスト形式で表示させるのではなくパズルの台紙があって、持っているピースがパズルにはまっているような形で表示されるようになっている。そのためどのピースを持っていてどのピースを持っていないのかが一目でわかるような UI になっている。実際の画面は図 5.1 に記載されている。

(※文責: 萩ノ沢実佑)

5.3.6 なぞなぞ一覧画面

この画面は、なぞなぞを一覧表示する画面である。なぞなぞはピースが函館のまちのどこにあるかを指し示している。なぞなぞがわからないユーザーでも楽しめるようにヒントも見れるようになっている。カードが並ぶように表示させ見やすくなるようになぞなぞ一つ一つに間が空くようになっている。実際の画面は図 3.9 に記載されている。

(※文責: 萩ノ沢実佑)

5.3.7 ピースサイン画面

この画面は、ユーザーが取得したピースにサインをする画面である。本アプリケーションの背景は大体が青であるが、サインするピースの背景はサインする際に背景色が目立ってしまわないように白色にしている。また、UIButton をシンプルかつわかりやすい見栄えにすることによって、子どもから大人まで直感的に扱える UI になっている。実際の画面は 3.12 を参照して欲しい。

(※文責: 石澤大輔)

第 6 章 成果報告会

6.1 中間発表

6.1.1 発表形式

2020年7月17日金曜日に、前期のプロジェクト活動の成果を発表するための中間発表が行われた。今年度はCOVID-19の流行により、オンラインでの開催になった。中間発表を行うにあたり、本プロジェクトの概要を説明する資料として、「A1サイズのメインポスター」と「プロジェクト紹介の動画」を作成した。

(※文責: 伊藤直樹)

ポスター

メインポスターは主に4人で作成し、Adobe Illustrator (Illustrator) を用いて作成した。Illustrator では同時に共同編集を行うことが不可能であるため、共同で作業するためにはZoom や Discord の画面共有を用いた。

メインポスターには、「プロジェクト概要」、「独自のタスク管理手法」、「開発サービスのアイデア出し」、「今後の予定」という大項目を設けた。A1サイズのポスター1枚に内容を収める必要があったため、本プロジェクトが行ってきたこと、これから行うことを上記の4つに絞って記述した。

「プロジェクト概要」では、本プロジェクトの目的とすること、ビーコンの特徴について記述した。「独自のタスク管理手法」では、アジャイル的なタスク管理手法を確立し、それに則ってタスクを管理していたことを説明した。「開発サービスのアイデア出し」では、まず函館のまちに潜む課題を発見するためにフィールドワークを行ったこと、それをもとにアイデア出しを行ったこと、そこから仮アイデアとして8つの案を選定したことを記述した。「今後の予定」では、中間発表以降の予定を月別に分けて、今後どのような活動を行っていくのかを箇条書きで説明した。

ポスターのデザインに関しては、文字やカラーはロゴのカラーと合わせて作成し、統一感を生み出した。またグリッドを作成することで、より整ったデザインを作り上げることができた。さらに文字幅などを調節することで、違和感のあるような部分での改行がされないように調整を施した。

(※文責: 伊藤直樹)

動画

動画の構成としては、順に「本プロジェクトの概要」、「今年度のプロジェクト活動の特徴」、「5月から中間発表までの活動内容」、「仮アイデア8つ」、「中間発表以降の活動」といった流れとした。

動画は、主に Adobe Premiere Pro (Premiere) のチームプロジェクトを用いて4人で作成した。Premiere のチームプロジェクトでは、複数人で同じプロジェクトに対する操作が可能であり、

効率的であると判断したため、このツールを選択した。

動画を制作するにあたって、他プロジェクトの動画や Web サイトにはない、奇抜さやこだわりを取り入れることで、多くの人が興味を持って見ることができる動画を目標にして取り組んだ。目標達成のための要素の 1 つとして、キャラクターの使用がある。今年度の本プロジェクトには「ビーコンうさぎ」というオリジナルキャラクターがいるため、そのキャラクターがプロジェクト活動について説明するという方法を用いた。もう一つの要素は、VOICEROID を用いたことである。人の声ではなく、機械音声を用いることで人々の注意を引き付けることができると考えた。キャラクターの使用と合わせて、YouTube で投稿されている動画の「ゆっくり実況」という形式を取り入れることで、より親しみやすい動画になると考えた。その他のこだわりとしては、全ての発言に対して字幕を付けたことがある。字幕のフォントも親しみやすいと思われるものを選択した。字幕をつけることで、機械音声が苦手な方や、聞き取りにくい部分があった場合でも、理解しやすい動画になると考えた。

動画制作における課題として、Premiere におけるチームプロジェクトの仕様を理解するまでに時間がかかってしまい、動画を作る以前の部分に時間を使ってしまったことがある。Premiere にはシーケンスというものがある。シーケンスに動画の素材を置いていくことで、シーケンス上に複数の素材が混ざった 1 本の動画が出来上がるというものである。チームプロジェクトでは 1 つのシーケンスに対して複数人で同時に作業することができず、同時に作業して全体に共有しようとするとどちらか一方の作業しか反映されない。同時にシーケンスを作業したい場合は動画を複数のシーケンスに分けると違うシーケンスで同時に作業することができるのだが、このことに動画制作後半になってから気がついたため、気がつくまでは 4 人のうち 1 人または 2 人が作業に参加できず、見ているだけになってしまうという事態に陥ってしまった。ほかにも、取り込んだはずのメディアがほかの端末で見た時にオフラインとなって見ることができなくなることや、正しくチームプロジェクトが読み込まれず、作業ができない人が出てきてしまったことがあった。

(※文責: 平山翔真)

中間発表当日

中間発表当日では、まず 15 時から 1 時間、各プロジェクトが作成した Web サイト、または動画の視聴と、メインポスターの閲覧時間が設けられていた。その後 10 分の休憩をはさみ、Zoom のミーティングルームを使用して、1 ターン 15 分の質疑応答を前半 3 回、後半 3 回の計 6 回行った。

また、質疑応答を担当する人数を均等にするために、前半 7 名、後半 8 名がミーティングルームに残るよう割り当てた。質疑応答では、ポスターの閲覧、動画の視聴が済んでいることを前提として、ターンが始まると同時に質疑応答を開始した。

さらに、本プロジェクト独自の取り組みとして、限られた時間外でもコメントや質問ができるように、Twitter のハッシュタグ (#BF4 中間発表) を作成して利用した。

(※文責: 伊藤直樹)

6.1.2 発表技術の評価と反省

評価アンケートは46件の回答が得られ、平均評価は10点中7.79点であった。そのうち、平均点以上の評価を得た回答を高評価、平均点未満の評価を得た回答を低評価と定義する。発表技術に関して、高く評価された点として、

- 他にはない動画で面白かった
- 動画の強み活かして、内容も理解しやすかった
- 動画にインパクトがあり、非常に目を引いた

などの意見があった。低く評価された点としては、

- すべての語尾に「うさ」がついていると聞きにくい
- 音声及早口で聞き取りづらかった
- ポスターとスライドの内容がほぼ一緒だったのが残念

などの意見があった。イメージキャラクターの使用と、そのアニメーションに機械音声を合わせた動画を作成したことによって、視聴者には今までにないような発表形式となり、強い印象を与えることができた。その反面、語尾に関してや音声の速度から、内容を理解しにくいといった意見も挙がった。

これらの評価から、今回の発表は、とくに動画に関しては好みが分かれるような発表になったと推測できる。

以上のことより改善点として、キャラクターの語尾の修正や、音声の速度の調整、また、動画に用いるスライドではとくに強調したい部分を抜粋すると良いといった点がある。

(※文責: 伊藤直樹)

6.1.3 発表内容の評価と反省

発表内容に関する平均評価は、プロジェクトの計画については6点中4.70点、目標設定については6点中4.74点であった。そのうち、上記と同様に、平均点以上の評価を得た回答を高評価、平均点未満の評価を得た回答を低評価と定義する。発表内容に関して、高く評価された点として、

- 今後に希望が持てる内容であった
- 前期の活動が細かく説明されていて、理解しやすかった
- 仮アイデアもよく考えられたもので、興味深かった

などの意見があった。低く評価された点として、

- 専門用語の説明がもう少し欲しかった
- 最初、質問してよい時間なのかがわからなかった
- 質疑応答がもう少し円滑だとなおよい

などの意見があった。このアンケートの回答から、発表の内容は理解しやすいが、専門用語の説明が不足していたといえる。また質疑応答の受け答えはできているが、いきなり質疑応答が始まり、聞き手を少し困惑させてしまった部分があった。

以上のことより改善点として、専門用語はなるべく噛み砕いて説明すること、よりスムーズに質疑応答に入るために、冒頭におおまかなプロジェクトの説明を行うと良いといった点がある。

(※文責: 伊藤直樹)

6.2 成果発表

6.2.1 発表形式

2020年12月4日金曜日に、本プロジェクト活動の成果を発表するための成果発表が行われた。中間発表と同様、COVID-19の流行により、オンラインでの開催になった。成果発表を行うにあたり、本プロジェクトの概要を説明する資料として、「A1サイズのメインポスター」と「プロジェクト紹介の動画」と「プロジェクト紹介のWebサイト」を作成した。

(※文責: 伊藤直樹)

ポスター

メインポスターは中間発表のときと同様に主に4人で作成し、Illustratorを用いて作成した。共同作業も行うときも中間発表と同じく、ZoomやDiscordの画面共有を用いた。

メインポスターには、「プロジェクト概要」、「活動内容」、「開発について」、「提案サービス」、「学び」という大項目を設けた。中間発表と同様に、A1サイズのポスター1枚に収める必要があったため、本プロジェクトの説明や行ってきたこと、サービスについて最低限の説明をするために、5つに絞って記述した。

「プロジェクト概要」では、中間発表に作成したポスターと同様、本プロジェクトの目的とすること、ビーコンの特徴について記述した。「活動内容」では、今年度の本プロジェクトの流れを「調査」、「企画・技術習得」、「開発」、「発表」の4つに分け、月別に、箇条書きで説明した。「開発について」では、本プロジェクトでは3グループすべてが「アジャイル手法」に挑戦したこと、開発の流れについての説明、チーム構成はアジャイル手法のスクラムに基づいて構成し、各チームにスクラムマスター、プロダクトオーナーを置いたことを記述した。「提案サービス」では、本プロジェクトが実際にどのようなサービスを開発し、提案するののかの簡単な説明と、各サービスのロゴを記載した。このポスターでの説明を簡単に記載した理由としては、メインポスターはA1サイズ1枚という制約があったこと、また、詳しい説明は動画で行われているためといった理由がある。「学び」では、約1年間のプロジェクト活動を通じて、メンバーが学んだことや、活動において大事であると感じたことを記述した。

ポスターのデザインに関しては、カラーや文章の配置、構成は中間発表のときと大きく変化させずに作成した。中間発表のときと変更した点としては、以前の反省を活かすために、文章のブロックごとの余白を少し大きめに確保し、フォントも少し線の細いものを使用した。この変更を施した

ことによって、詰められたように見える文章の問題を解消できた。また、左右の余白のスペースを少し小さくした。それを行うことによって、文章や図を記載できるスペースを増やし、より情報量を多くすることができた。さらに、中間発表のときよりも、図解表現を用いた説明を多くすることで、視覚的に認識しやすいようなデザインを作り上げることを心掛けた。そのほかに、文章の文字色を黒から明度を少し上げることで、下地の白とのコントラストを少し抑え、見やすくなるように施した。

(※文責: 伊藤直樹)

動画

動画の構成としては、順に「本プロジェクトの概要」、「今年度のプロジェクト活動の特徴」、「5月から成果発表までの活動内容」、「考案・開発した3サービス (MET.・でももに!・俺を食べてくれ!!) の紹介」といった流れとした。

動画は、中間発表の動画と同様に、主に Premiere のチームプロジェクトを用いて5人で作成した。中間発表動画作成の際の課題を意識し、あらかじめ動画の区切りごとで切ったシーケンスを5つ用意して動画制作を開始した。この試みにより、手が余ってしまう人は1人も出ず、効率的な動画編集を行うことができた。メディアがオフラインになってしまう問題は、正しい手法でメディアを取り込んでも起きてしまうことがあったため、Premiere 側のバグであると思われる。この問題はこまめに編集結果を共有し、動画作成者間で反映されていないことなどの報告を行うことで、早めに問題を取り除くことができた。

中間発表の際のアンケートでは動画に対し、合成音声について、聞き取りやすい、わかりやすいといったポジティブな意見もあったが、反対に聞き取りにくい、内容を理解しにくいといったネガティブな意見もあった。この結果から、合成音声は好みが分かれてしまい、大衆に向けての発表としては相応しくないと考えた。ただし、中間発表の際の、奇抜さやこだわりを取り入れて多くの人が興味を持って見ることができる動画を作るという目標は変えなかったため、キャラクターの使用はやめず、音声にはメンバーの声をキャラクターボイスとして採用した。中間発表の動画では、語尾に「うさ」とついていたが、これによって聞き取りにくくなっているという意見があったため、普段話す言葉に近い「だよ」に変更した。動画内での表現方法として、中間発表では文字が多かったが、期末発表では中間発表よりも文字での表現を減らし、図や表を用いることを心掛けた。

成果発表の動画は、中間発表の動画よりも5分程度長いため、キャラクターが最後まで話していると視聴者が飽きてしまうと考え、各サービスの発表では、それぞれのサービスのプロダクトオーナーが出演した。撮影はグリーンバックを設営し、各サービスの印象を伝えやすい背景を合成することで、視聴者が内容を理解しやすく、飽きにくい動画になった。音声に関しては、ラベリアマイク、ショットガンマイク、iPhone を使用して声を録音し、3つの音声を組み合わせることで、聞きやすい音になるように工夫した。各サービスの雰囲気や機能が視聴者に伝わりやすいように、アプリを使用している様子を表した動画やスマートフォンの画面を表示するといった工夫もした。

中間発表の動画では字幕を動画内に作成しており、概ね好評だったが、手間がかかってしまうことや画面の占有率が大きかったことを踏まえて廃止し、代わりに YouTube の字幕生成機能を使用した。YouTube の字幕生成機能はあらかじめ文章を書いたテキストファイルを読み込ませると、

ある程度自動で時間などを調整して、字幕を生成してくれるといったものである。これを使用することで、字幕を使うかどうかを視聴者に委ねることができ、画面を端から端まで使って動画を作成することができた。

課題としては、「でももに！」の部分の編集を担当したメンバーが、開発でも多くのタスクを抱えており、編集に十分な時間を割くことができなかつたため、編集を終えるのが遅くなってしまったことがある。この課題は、編集に対する時間や仕事量のマネジメントが行き届いていなかったことが原因であった。

(※文責: 平山翔真)

Web サイト

成果発表の際には、プロジェクトに関する情報を集約し、より効果的に活動内容を伝えるため、Web サイトを作成した。Web サイトの公開には、GitHub の Web ページホスティングサービスである GitHub Pages を用いた。

主な選定の理由として、すでに利用している GitHub の機能であるため、追加の知識が不要であったこと、Markdown ファイルや画像データを配置するのみで、面倒な環境の構築なしに作成に取りかかることができること、また、Markdown で作成したページには自動でテーマが適用され、簡単にサイトのデザインを選択することが可能であった点がある。Web サイトの作成に当たっては、プロジェクトメンバーのうち、主に文面や画像などのコンテンツを用意するメンバー、スタイルを修正するメンバー、全体を調整するメンバーの 3 人が担当した。

内容としては、発表動画やプロジェクトの目的、ビーコンの特徴、活動内容、サービスロゴの一覧を用意し、発表動画を視聴しながらも、活動の目的や内容、各サービスの特徴などをすぐに確認できるように工夫した。また、そのほかに工夫した点としては、GitHub Pages に用意されているテーマでは不要なボタンや不都合なスタイルが存在したため、スクリプトやスタイルを用いて見やすく改善したこと、favicon を設定して、ブラウザのタブ一覧からページを判別しやすくしたことがある。

その結果、成果発表の評価では、Web サイトがわかりやすく書かれていたというフィードバックが得られた。Web サイトの作成を行なった上で反省すべき点としては、作成開始が提出締め切りの直前になってしまったことから、作業時間が限られていたために完成イメージの共有が十分にできていなかったという点である。

(※文責: 久米田羽月)

成果発表当日

成果発表当日の段取りは中間発表と同じで、15 時から 1 時間、各プロジェクトが作成した Web サイト、または動画の視聴と、メインポスターの閲覧時間が設けられていた。その後 10 分の休憩をはさみ、プロジェクト WG によって用意された Zoom のミーティングルームのブレイクアウトルームに移動し、1 ターン 15 分の質疑応答を前半 3 回、後半 3 回の計 6 回行った。

また、質疑応答を担当する人数を均等にするために、前半 7 名、後半 8 名がルームに残るよう割

り当てた。また、どのサービスに対しての質問にも対応できるように、各サービス 1~3 名が残るように割り当てた。

質疑応答では、中間発表の反省を活かし、まず冒頭に本プロジェクトの説明と、提案する 3 つのサービス説明を、Google Slides を画面共有で映しながら 2 分程度説明を行った。その後、質疑応答の時間とした。

また、中間発表と同様に、Twitter のハッシュタグ (#BF4 最終発表) を利用し、コメントや質問を募った。

(※文責: 伊藤直樹)

6.2.2 発表技術の評価と反省

本年度の成果発表は、COVID-19 の影響もありオンラインで開催されることとなったが、動画担当大臣をはじめとした、各大臣を筆頭に、オンラインならではの効果的な見せ方ができたと感じている。上述でもあるが、中間発表では、本プロジェクトのイメージキャラクターである”うさ”と VOICEROID による合成音声を全編に使用したことで、聞き取りにくいといった声が複数上がった。これを解決させるため、成果発表では、プロダクトオーナーが実際に出演し、合成音声を使わず、プロジェクトメンバーがうさの声を担当する形へと変更した。また、クロマキー合成を用いることにより、プロダクトオーナーが説明しながら、実際に使用している場面を紹介するといった画面構成が可能となった。その際、少しでも発表者の声が聞き取りやすくなるように、ラベリアマイクとショットガンマイクを用いて、反響の多い教室内で撮影するための工夫を施した。

その結果、発表後のアンケートでは、

- 動画の音声が聞き取りやすかった
- 発表者の表情を見ながらサービスの内容を聞くことができてよかった

との声が上がリ、サービスごとのアプリケーションの使用方法がわかりやすかったという肯定的な意見が多数を占める結果となった。

中間発表終了後より、動画の見せ方について検討を重ねていたため、このような評価が得られ嬉しく思うと共に、聴取者に効果的に伝わっていることがわかり、大変良い結果となったと考えている。しかしながら、15 分という短い時間内にサービスの内容を全て詰め込むことが難しく、さらに説明が欲しいといった要望が見られた。また、キャラクターである”うさ”の語尾を”だよ”に変更し、親しみやすさを演出したつもりであったが、不快であったというネガティブな意見も得られた。

当日の発表時には、サービスの紹介とプロジェクトの紹介を 15 分の動画に全てまとめることが困難であったため、Web ページを作成した他、質問時間にできるだけサービスの詳細について理解してもらえるように努めた。その際、Twitter を用いて、#BF4 最終発表というハッシュタグを作成し、聴取者からの質問を募集した。また、プロジェクトメンバーは全員が Zoom のカメラをつけ、質問に対応した。

これについて、

- フレンドリーさを感じた

- 良い雰囲気の話だった
- しっかりと段取り等を取られており、質問もスムーズに答えられてよかった

などの意見があった。Twitter を利用したことで質問しやすい環境を作ることもこの要因であると考えている。質問に対しての受け答えについても、的確に答えられていたとの感想が得られた一方で、

- アンケートへの誘導や、Web サイトの提示がよりスムーズであるとよかった
- 質問がこない場合の対策が前半発表時には取れていなかった

などの問題点も指摘された。事前に予想される質問内容を各グループ内で考え、回答案を用意し、プロジェクトメンバー全員が答えられるように準備をしていたが、質問がこない場合の予測をしていなかったため、この結果になってしまったと考える。しかしながら、発表中盤より質問がなくても、動画やポスターに含めることができなかった各サービスの補足説明などを行うことができていた。

これらの事前準備等により、非常に良い成果発表に繋がったと考えている。

(※文責: 増田翔)

6.2.3 発表内容の評価と反省

発表内容について、集めた結果 6 点以下のコメントはネガティブな内容だったため、6 点以下の評価を低評価、7 点以上の評価を高評価として以下を記載する。また、高評価の意見、低評価の意見のどちらに関しても改善すべき点などの指摘があった、これらについては高評価、低評価とは別に記載する。

(※文責: 宮田悠治)

発表内容に関して、高評価な意見として

- 「子供の観点を考えたのは、とてもいいと思ったピース集めはかなり函館を楽しめるようになるのではと思った。」
- 「ビーコンを使うことで目的地付近で探索要素があるのは面白いと思います。」
- 「日常生活で使うことができそうなものを開発していて良かった。ぜひ使いたいと思った。」
- 「実際に使ってみたいなと思いました。」
- 「どのビーコンも活用してみたいなと思いました。」
- 「非常に函館に根付いたいい内容だったと感じました。」
- 「どの技術もそれぞれの狙った層に合っていたので、実装が楽しみです。」
- 「ビーコンをうまく絡めたアイデアばかりで良かったと思いました。」
- 「発表内容のクオリティも高くてすごく良かったです。全部のアプリが魅力的でユーザーによって使用するアプリを変えられますね。」
- 「成果が高いレベルに達していると思います。」

などが得られた。実際に使ってみたい、魅力的といった好意的な意見が多数見受けられた。このことから、このサービスにはユーザーに使ってみたいと思わせる魅力があることがあると考えられる。また、狙った層に合っていた、子供の観点を考えたのはとても良いと思った、という意見があった。このことから、私たちのサービスが想定していたユーザーと合っていたと考えることができる。

(※文責: 荻ノ沢実佑)

低評価な意見としては

- 「どこでビーコンが使われていたのか不明でした。」

が得られた。本サービスのユーザーストーリーを紹介する動画では「ピースと対応したビーコンを～」というように言及している。しかし、ビーコンについて言及している場所はそのみで、どこで、どのようにビーコンを使っているとはっきりと視聴者に対して説明している部分がなかった。そのため取りこぼされてしまい、視聴者に伝わらなかったのだと考える。ユーザーストーリーの前に、本サービスでのビーコンの活用方法を別途言及すべきだった。また、発表内容に低評価をつけている人のほとんどがコメントの欄が空欄であった。自分たちのサービスが動画、ポスター、Webサイトを通して全く伝わらなかった可能性がのではないかと考える。もしくは、自分たちのサービスに何も良いとも思わず、何も感想も浮かばなかったのだと考える。

(※文責: 宮田悠治)

また、コメントに寄せられた指摘に関しては、

- 「フィールドワークにおける、函館独自という観点は抜けているように感じた。」
- 「目標がぼやっとしている。」
- 「パズルを完成することで得られるメリット(クーポンを与える)などがあるとより利用するサービスが増えるのかなと思いました。」
- 「家族がバラバラに歩き回ってピースを集めるようなことはできないでしょうか？」
- 「最後のパズルがさらに函館らしいイラストになるとより良い。」

などが得られた。パズル完成後のメリットやパズルそのもののイラストについてなど本サービスをより良くするための意見や感想があった、サービスの今後の発展に生かしていきたい。また、家族がバラバラに歩き回ってピースを集めることができないのか、というアプリケーションの使い方に関する意見があった。これは、本サービス紹介動画のユーザーストーリー紹介の部分で大人は函館観光を楽しみ、子どもはピースを集めると言った紹介をした部分に対する意見だ。親子で揃って同じ観光地内でピース探しを楽しむことはもともとのユーザーストーリーには存在していなかったため、今後ターゲットにすべきか考慮すべき点だと言える。

(※文責: 宮田悠治)

第 7 章 各メンバーの振り返り

7.1 役割分担

本グループでは、スクラム手法に則り、プロダクトオーナー 1 人とスクラムマスター 1 人を決定した。また、技術的な視点からサービスの機能を考えるサブプロダクトオーナーと、スクラムマスターのサブスクラムマスターを 1 人ずつ決定した。その上で、サービス開発を行う際に、iOS 開発担当が 3 人、Android 開発担当が 3 人に割り振り、サービスのデザインを 1 人で行い、6 人全員が開発に関わるように分担を行った。報告書の執筆は、グループ全員で行った。以下に、各メンバーが担った役割を示す。

久米田羽月

- プロダクトオーナー
- Android アプリ開発
- Web ページの作成

萩ノ沢実佑

- スクラムマスター
- iOS アプリ開発
- サービスのロゴ作成とデザイン考案
- XD を用いたモックアップの作成

村石拓海

- サブプロダクトオーナー
- iOS アプリ開発
- UI の実装
- 開発環境整備

山本雄平

- サブスクラムマスター
- Android アプリ開発
- UI の実装

宮田悠治

- Android アプリ開発
- UI の実装

石澤大輔

- iOS アプリ開発
- UI の実装

(※文責: 石澤大輔)

7.2 久米田羽月の振り返り

今年度のプロジェクト学習は、COVID-19 の影響により、多くの活動がオンラインで行われることとなった。さらに、ほぼ全てのメンバーと面識がなく、上手く馴染めるかどうか不安な部分としてあった。その中で、プロジェクトの 1 メンバーとして、信頼関係を築くことができるように、初期の段階ではできる限り他のメンバーとコミュニケーションを取ることを心がけた。その結果、中間報告が終わるまでには、全てのメンバーと信頼関係を築くことができ、プロジェクトを良い方向に導くため、サービスについてのアイデアや、その良い点や課題点、あるいは活動全体に対する振り返りように、建設的な話し合いができるようになった。

また今回、私はプロダクトオーナーを務めた。その中でよくできた点もいくつかはあったが、至らない点も多くあったと感じている。グループが発足し、サービスの方向性を固める段階では、アイデア決定の直前に内容を変更したために、考案者でありプロダクトオーナーである自分でもサービスの内容や方向性をはっきりと認識できておらず、サービスの根幹を定めるインセプションデッキを上手く作ることができなかった。そのため、インセプションデッキを構成する要素の一つであり、サービスの内容やターゲットを定めるエレベーターピッチについても、1 から決める必要があった。方向性を決めるにあたって、グループメンバーの意見を取り入れようと考えたが、サービスの根幹が定まらないまま曖昧な意見を聞いても、メンバー側も何を答えてよいか困ってしまい、十分に建設的な話し合いができないという問題が発生した。実際に、サービスの内容や方向性を定めるのはプロダクトオーナーの責務であるため、ある意味でそれを放棄してしまっていたのは大いに反省すべき点であると考えた。この問題を解決するため、まずはプロダクトオーナーである自分が、サービスの内容と方向性を考え、それを 1 つの意見としてメンバーと共有することで、ある程度の共通認識を持つことができた。その上で良かった点としては、メンバーの考えるサービスの内容と方向性を話してもらうことで、お互いの意見の良い点や課題点などを整理しつつ、サービスの方向性を少しずつ定めることができたことが挙げられる。さらに、この作業を通してメンバーのユニークな意見を取り入れることや、メンバー全員でサービスの目的や目標を共有することができた。この作業には非常に時間がかかり、少ない開発期間に対してもさらに影響が出てしまったが、メンバー全員が納得するまで話し合うことで、良いものが出来上がったと考えている。

個人的な反省としては、もう少しアジャイルとスクラムの勉強をしておきたかったと思う。活動の中で、先生や TA の中田さんからアジャイルの考え方やスクラムイベントの中身について教えていただいたものの、全体の流れやプロダクトオーナーやスクラムマスターの役割など、完全に把握していたとは言い難い。当初からスクラムを使った活動を行っていくとは知っていたものの、enPiT からスクラムの教本である SCRUM BOOT CAMP を借り受けたのは後期の活動に入ってからだった。しかも、サービスの内容や開発に気を取られ、あまり中身に目を向けることができていなかったのは非常に残念に思う。しかしながら、そういった中での失敗などは逆に貴重なもので

あると考え、前向きに捉えた上で今後のアジャイルやスクラムの学習に役立てていきたい。

そのほかの視点では、サービスを考案するにあたって、そのサービスでどんな体験ができるのかを常に考えていくことが非常に重要であると思った。具体的には、サービスを開発するにあたってどんな機能を実装するのかが大切であるが、その機能は何のためであって、誰の、どのような体験に繋がるのかを考えないと、サービスの軸がぶれてしまうということである。さらに、サービスを説明するときには、どんな機能があるのかだけを説明してしまうと、聞く側にとってはそれ以上の興味を得られずに、サービスを使う価値が見えてこない。実際に、初めは機能ベースで考え過ぎていた節があり、先生にレビューしていただいた際にも、もっと体験できることを利用者の視点で説明してほしいと言われてしまったことが記憶に残っている。今後サービスやアプリを開発する際には、これをしっかりと意識して取り組んでいきたい。全体を振り返ってみると、サービスの内容と方向性の決定までに必要以上に時間がかかってしまったこと、初めのうちはメンバーとの対話が部分的に足りなかったことなど、プロダクトオーナーとして至らなかった点が多くあるように感じる。逆に、サービスの内容が定まっていなかった分、話し合いを繰り返してメンバーの意見を引き出し、全員が内容に対して納得のできるサービスにまとめ上げることができたのは、大きな成果だと考えている。

また、実装が全て完了していない部分も残っているため、今後の活動でもインセプションデッキを意識しながら仕様の確定と開発を続けていくべきだと思う。サービスの今後の展望としては、アプリをリリースして実際の観光客に使ってもらい、サービスの有用性を確かめたり、目的である、観光を家族で楽しむということを達成できるような改善を加えていくことが必要だと考える。

この1年で学んだ重要なことは、活動や達成したいことの目的をメンバー全員で共有しなければ、良いものを作るのは難しいということである。これは基本のことかも知れないが、これができなければ良い成果物を作ることはできないので、最も大切にしなければならなかったと思う。このように、1年間の活動を通して、多くのことを学ぶことができた。今後の活動でも、このプロジェクトで学んだことを活かし、仲間とうまく協力しあえるようにしていきたい。

(※文責: 久米田羽月)

7.3 萩ノ沢実佑の振り返り

前期、後期と1年間のプロジェクト活動を通して、非常に多くのことを学んだ。今年度は、COVID-19の影響によりオンラインで活動が執り行われた。これまでの対面でのプロジェクト活動とは一変して基本的にオンラインでの活動であったため、どのように活動が進んでいくのか全く見当もつかず不安であった。また、プロジェクトメンバーと打ち解けてうまく活動していけるかについても非常に不安に思っていた。しかしそんな予想とは裏腹に、最終的にはプロジェクトメンバーと打ち解けて非常に楽しく活動することができた。私たちのプロジェクトは、プロジェクトリーダーを置かないという選択を取り1年間活動してきた。これは非常に面白い取り組みであったと思う。前期は週替わりのファシリテーター制をとり、担当となったプロジェクトメンバーが進行を務めた。後期は担当大臣制度を作り、前期のファシリテーター制での問題点を改善し活動を行った。このプロジェクトリーダーを置かずに活動していたおかげで、誰かに仕事を押し付けすぎず、

自分の仕事に責任を持って取り組めた。また、この方式で活動したからこそ、プロジェクトメンバーの仲が深まり打ち解けて活動できたのだと考える。

開発の面では、主にサービスのデザインに携わった。デザインの分野に興味はあったものの、これまでデザインについてきちんと学んできたわけではなかったため、勉強しながら手探りで活動であった。どのような UI ならユーザーは使いやすいのか考えながらデザインすることは非常に難しかったが、ユーザー目線に立って開発を進めることを学ぶことができた。他にも、Adobe XD, Adobe Illustrator, Adobe Photoshop などのデザインツールの使い方に慣れることもできた。また、iOS アプリの開発にも少し携わった。私は、iOS アプリの開発が未経験であったため、開発経験のあるグループメンバーに助けられながらの開発であった。微力ながらも開発の助けになれている部分があれば嬉しい限りである。

開発を進めるにあたって、私たちはアジャイル開発手法のスクラムに則り活動してきた。その中で私は、スクラムマスター (SM) として他のグループメンバーを支援してきた。支援してきたなどと言ったものの、スクラムでの開発や複数人での開発は初めてであったため、これでやり方が本当にあっているのか不安に思いながら、探り探りで活動してきた。自分なりにスクラムについて勉強してきたため、スクラムイベントの際には少なからず何かしらグループメンバーを支援することができていたと考える。この1年間本当に実りのあるものであった。実力のあるメンバーに囲まれて、自分は何ができるのだろうかと悩むことが多々あったが、そのおかげで技術の面や精神面など様々な面で成長することができた。この活動を通して学んだことは、卒業研究、就職活動など多くの場面で活かしていきたい。

(※文責: 荻ノ沢実佑)

7.4 村石拓海の振り返り

まず、前期のプロジェクト活動でのコミュニケーションについて、COVID-19の影響でオンラインでの活動を余儀なくされ、プロジェクト活動が開始した直後は、なかなか他のプロジェクトメンバーや、先生方、TA との距離感を掴みづらく、グループで行うことが前提のプロジェクト活動をオンラインで行うことに、心配があった。しかしながら、初回のプロジェクト活動において、先生主体で他己紹介等のアイスブレイクを行ったり、オンラインでのスタートアップ飲み会の開催、また、初期から Slack を導入して、他プロジェクトメンバーだけでなく、先生方、TA も交えて、オンラインでのプロジェクト活動であっても気軽にコミュニケーションを取りやすくすることで、プロジェクト活動の最終成果発表時ほどではないが、早い段階でプロジェクトメンバー同士と打ち解けあい、1年間を通して和やかな雰囲気で行えたことは非常に良かったと考えている。オンラインでのプロジェクト活動で学んだ、コミュニケーションの取り方については、卒業研究や、就職後等の活動でも有意義なものであるため、生かしていきたいと考えている。次に、前期のプロジェクト活動全体での学びについては、プロジェクトリーダーを置かずに活動してきたことが、個人的に一番の学びであった。プロジェクトリーダーを置かずに活動した発端としては、例年は、初回のプロジェクト活動でプロジェクトリーダーを決定していたが、オンラインでの活動もあり、プロジェクト活動がどのように行われるか、良くわかっていなかったことから、一時的に設置せず

に、プロジェクト学習の進行役であるファシリテータ役の人をもうける方式であったが、結果的に開発サービスが確定して後期にグループが分かれるまで、この方式を採用して、プロジェクト学習を進行した。自分も、プロジェクト学習はプロジェクトリーダーを決めて、その人中心で進行して活動が進むものと考えていたため、結果的にプロジェクトリーダーを1年間置かずとも、良い活動ができたことが、自分にとって大きな知見であった。また、今年は例にもなくオンラインでのプロジェクト学習になり、学生はもちろんのこと、先生方やTAの方もどのようにプロジェクトを進めていけば良いか探りのところもあり、初対面の人も多く、人との距離感もつかみ辛いオンラインで、ここまでプロジェクト学習を完遂できたことが有益でした。

(※文責: 村石拓海)

7.5 山本雄平の振り返り

今年のプロジェクトは COVID-19 の影響もあり、オンラインでの実施となった。グループでの開発の経験があまり無いうえに、多人数でのオンラインでの会話について上手く成立するのかなどプロジェクト開始前から不安が募っていた。実際に始めると、アイスブレイクやキックオフとしてオンライン飲み会が設けられていたため、メンバーとの交流を深めることができ、プロジェクトにおける発話しやすい雰囲気づくりができたと思う。前期の進め方としては、週替わりでファシリテーター2人任命し、その2人がプロジェクトの進行を務めるという形で進めていた。また、アジャイル的な開発手法にも取り組んでおり、リーダーを決めることなく、意思決定はメンバー全員の合意をとるということが決まった。私もファシリテーターを務めたが、1回目は上手く進行を務めることができず、もう1人のファシリテーターやメンバーに助けをもらいながらプロジェクトを進めていた。この時の反省としては、進行役として、その週に行う活動のことは一番把握できていたのに関わらず、実際に作業を始めるときには曖昧な説明や指示を行ってしまい、自分の中で考えがまとまっていけないのにも関わらず、発言し、全体の進行を遅らせてしまった。このことから、自分の考えを人に分かりやすく説明できるように意識して活動に取り組むようにした。夏季休暇期間では、勉強会が設けられており、そこで私は、Golang で LineBot を作成したことについて発表した。この勉強会を通じて、開発時に LineBot を使用するときには活かす技術や知識を習得することができたと思う。また、メンバーの勉強会に参加することで、共同開発で重要になる GitHub の使用方法や興味があった Android や iOS についても学習することができたので有意義な時間を過ごすことができたと考えている。後期に入ると前期で出たアイデアと夏季休暇期間に出したアイデアの絞り込みを通じて、3つのサービスを開発することに決まりました。私は「でももに！」の開発メンバーになった。後期になり開発が始まると、スクラムを回すことになり、私はそのなかで開発メンバーであると同時に、サブスクラムマスターという役職に就くことになった。サブスクラムマスターはスクラムマスターをサポートするという役割であり、スクラムマスターは主に、サービスの会議などにおいて、進行役を務めることや、プロジェクトの全体の進め方において、誤った方向に進んでいないかを開発チームやプロダクトオーナーに伝える役割であった。本来のグループでの開発においては、スクラムマスターは開発に携わるべきでないのだが、私の場合は開発も同時に行いつつ、スクラムマスターを務める必要があったので、プロジェクト全体が正しい方向に進んでいる

かということについて、俯瞰的に物事を判断することにとっても苦労した。実際にサービスごとの活動では、すぐに開発に入る予定だったが、なかなかサービスのターゲットや方向性について決めることができず、予定の2週間ほど開発に入るのが遅れてしまった。こういった、方向性などについて悩んでいる場面においても、スクラムマスターが本来の目的とずれていないかなどを伝えるべきだったと反省している。私は、Androidのアプリケーション開発に携わり、画面遷移やUIの一部などを担当した。アプリケーション開発が初めての経験だったため、画面遷移やUIパーツの配置、アーキテクチャなど学習して、実際の開発に応用させるのは大変だったが、実際に想定した動作ができるまで、不具合の原因を特定し、実行を繰り返す過程は開発をしているという実感があつても楽しく開発を行うことができた。現段階でのアプリケーションはまだ開発の途中であり、これから様々な機能を追加するため、これからもAndroidの勉強を継続しつつ、開発に取り組みたいです。プロジェクト全体を振り返ってみると、オンラインで様々な不安はあつたが、メンバーをはじめ、TA、教員や外部の方にサポートしてもらい進めることができ、このプロジェクト学習を通じて自分の今後に活かすことができる技術や知識を習得することができた。

(※文責: 山本雄平)

7.6 宮田悠冶の振り返り

私は、本システム情報科学演習（以下プロジェクト学習）で様々なことを学ぶことができた。今年度のプロジェクト学習はCOVID-19の影響で、ほとんどがオンラインでの活動になり、プロジェクトのメンバーと顔を合わせたのはZoomにて、インターネットを介したものであつた。初めて現実で顔を合わせたのはフィールドワークの頃だった、あつて数分の知り合いではあるのだが、そうでないような絶妙な距離感を今でも覚えている。最初、何もわからない人の自己紹介（正確には他己紹介が行われた）を聞いて、始まったプロジェクト学習はととても不安だった。もともと知り合いだった人、そうでない人、編入生、十人十色のメンバーたちと毎週の活動、フィールドワークや議論をしていく中で、それぞれの人となりを知っていくのはとても新鮮な経験だった。前期の活動はあつと言う間に終わった。フィールドワークやサービス考案を通して函館の課題を見つけてそれをどうすれば解決できるのか考えた。ざっくり書くとこれだけでもまとめられるが、広げればいくらでも大きくなるほどに濃厚な前期だった。

後期の活動は前期のそれよりも早かつた。後期では、主にサービスの決定作業とそのサービスの開発を行った。サービスの決定作業ではエレベーターピッチの決定や、サービスの対象者の確定に難航した。また、私は、人よりできるとか得意というわけでもなく、ただ、たまたま触れていたし、楽しそうだったからAndroidアプリケーションの開発に携わつた。開発前にAndroidアプリをハッカソンで開発していたなどの経緯からAndroid担当大臣という役割にもなつた。Android大臣として置かれた立場もあり、必死に勉強したつもりだし、たくさんコードも書いたつもりだ。まだまだ足りないが、これからもAndroid開発を続けていきたいと思えるのはプロジェクト学習を通して楽しんで開発することができたからだと思うので、本当に感謝している。このプロジェクト学習を通して学んだ、コミュニケーションに関する技術や知見、開発技術を今後の卒業研究や就職活動にも生かして行きたいと思う。

7.7 石澤大輔の振り返り

今年のプロジェクト学習は、COVID-19の影響により例年とは異例のほぼオンライン下で行われる演習であった。今までは、対面で授業を受けたり友人と交流をしていたため、ZOOMを用いて人とはほぼ実際に会わずに1年間を共に過ごしていくということが先がイメージしづらかった。初めて顔合わせをして、どういう方針でプロジェクト学習を進めていくかなどを説明された後、自己紹介を通じて全員まではその場で覚えられなかったが、メンバーのことを知ることができた。しかし、知ることができたとはいえ、オンライン下でどのような距離感でメンバー、先生、Teaching Assistant（以下、TA）と接すれば良いかというのが掴めなかった。加えて、プロジェクトリーダーを決めずに最初は進めていくという方針に対して、今まで小中高で行ってきたチーム作業は全てリーダーを決めて進めてきたので、居ない中でどういう風に進めていけば良いのか手探りな状態であったため、不安がとても大きかった。前期の活動では、主に15人での話し合いが多かった。その話し合いを行う上で、プロジェクトリーダーを決めない代わりに、ファシリテーター制度を設けて進行した。最初の方に担当してくれたメンバーを見ていると、進行がスムーズに進んでいたり、しっかりと議事録を記録してくれたりとても頼りになっていた。これを見て自分も真似しようとしたが、進行に少し遅れが出たり、議事録の記録が追いつかなかったり、その場その場で決定することをメンバーに上手く誘導できずにと、思い返せばすごく散々なファシリテートだった。初めてファシリテーターを担当した時は、まだ実際に顔を合わせていない時だったので、メンバーへの強い申し訳なさを感じながらも、自分が（当たり前ではあるが）まだ未熟であると感じた。活動が進んでいく中で、手法についてよく学んだり、共有するためのソフトウェアの使い方を学んだりと進め方に慣れていった。また、フィールドワークや話し合いを通じてメンバーとの距離感や、オンラインでの活動の楽しみ方を知り、最初に感じていた不安は前期で解消された。話し合いを行ったり発表する上で、少人数での話し合いを行って、全体で話し合った結果を共有したり、全体で一つ一つについて意見を言っていくと、様々な方法を試して活動をしていった。そのため、オフラインで行っていたことがオンラインでは通用しない部分があったり、函館のまちの課題を解決するべく提案したサービスを発表する際にも、ジェスチャーや視線がいかに関手に伝えるためのコミュニケーションとして役立っていたかを実感したりなど、主にオンライン上とオフライン上での情報共有の違いという点で学ぶことが多くあった。また、ビーコンの特性を生かしつつ、置き換わることのない何かに役立つサービスを考案するということがどれほど大変か、ということも学んだ。

後期の活動では、夏季休業期間中で学んだことを生かし、各サービスの活動に入った。本来であれば、サービスの活動に移ってからは、エレベーターピッチやトレードオフスライダー等のファーストプリントをすぐに終わらせてプロダクトの開発に入る予定だった。しかし、それらの内容を決めていく中でターゲットや最終的にユーザーに体験してもらいたいことが曖昧になっていった。この内容を全員で納得のいくまで話し合いを行い、最終的に1ヶ月ほど期間がかかってしまった。時間はかかってしまったものの、全員が納得のいく状態になったのは良いことだと思うし、ユーザー視点を最優先に考えられた結果だと思う。開発の中で、自分はiOSアプリケーションの開発主任でUI設計やビーコン検知の担当をした。セカンドプリント以降の開発でも、UIやデザイ

ンなど、ユーザーが使う上でストレスのかからない設計をして、実機で触り、先生や TA から意見をいただくことで、ユーザーからの視点の追求がいかによりサービスを考える上で大事になってくるかを実感することができた。発表会では、動画作成を担当したが、これもまた自分たちが伝えたい内容を伝えつつ、視聴者が見やすい動画を作ることにかなり苦戦した。また、時間の制約がとて強かったこともあり、この1年の濃い内容を全て伝えることはできなかつたと感じるが、外部評価を見ても高評価をいただいていたので良かったと感じている。全体を通して、チームで何かを成し遂げようとしているときに、そのプロダクトやサービスの内容であつたり、進捗状況などの情報共有がいかにより大事かということがオフラインでやるときより顕著にわかつたのではないかと思う。また、その情報共有がメンバー同士の信頼を深めていったり、話し合いの中で相手のことがよくわかつてきたと思う。技術面でも精神面でも支えられることが多かつたと感じるので、ある種コミュニケーションが大事であることが実感できた。プロジェクト学習は終了してしまうが、今回作成したサービスをターゲットに使ってもらつた上でサービスの質を向上することができたら、その過程でもっと学べることがあると思うので、継続して追求していきたい。この1年間で、貴重な学びを経験できたので、今後の卒業研究や仕事でも活用していきたい。

(※文責: 石澤大輔)

第 8 章 まとめと展望

8.1 前期の振り返り

前期では、プロジェクトリーダーを設けず、週替りのファシリテーター制度でプロジェクト活動を進めた。これによって、特定の人に負荷が集中することを避けた。プロジェクトの最終意思決定については全員が納得するように運営した。これは、メンバーの全員がしっかりと責任意識を持ってプロジェクト活動に能動的に参加することを促すことにもつながった。

タスクの管理については、GoogleJamboard を用いてバックログを作成し、タスクを優先度順に並べて管理することで、今回取り掛かるべきタスクや、今後のやるべきことなどをメンバー全員で共有できるようにした。しかし、ファシリテーター交代の際、前週のファシリテーターと次週のファシリテーターで情報共有がうまくいかなかった。また、タスクの管理自体は GoogleJamBoard を用いて全員で共有し管理していたが、タスクの完了までの具体的な見通しをたてることに時間がかかってしまい、本質的な作業に取り掛かるのに遅れてしまう事があった。この問題の前者に対しては、議事録を複数人で書き、取りこぼしを防ぐこと、Zoom での会議を録画するなど、情報を残すことを徹底することで解決していった。また後者に対しては、全員でバックログの粒度を上げる作業や、タスクの重さづけを行う作業などを行うことで解決した。

前期における具体的な活動としては、ロゴ制作、フィールドワーク、サービス考案という活動を行ってきた。ロゴ制作では、メンバーのそれぞれがプロジェクトのイメージを考えロゴをデザインしてロゴに込められた想いや意味などを発表し合った。その後メンバーのレビューをもらい、ブラッシュアップを行っていった。メンバーによる投票を何度か行い、都度ブラッシュアップしていき、数を絞って入れていくことで現在のプロジェクトのロゴが確定された。

フィールドワークでは、函館の 5ヶ所でフィールドワークを行った。フィールドワークを行った場所は、「五稜郭」、「赤川・美原・富岡・昭和」、「函館駅・西部地区」、「七飯・北斗」、「空港・湯の川」である。この 5ヶ所を周り、各々気づいたことや思ったことについて GoogleJamboard や Slack のフィールドワーク用に作成されたチャンネル、手書きのメモなどを使用して書き出していたり、参考になりそうなものの写真を撮っていたりした。フィールドワーク終了後、GoogleJamboard を用いて意見を書き出し、KJ 法を用いてグルーピングを行い、関連性を明らかにした。サービスの考案では、フィールドワークのまとめや、自身の経験などからサービスを考えた。その際、BSKJ 法に加えて、ハッカソン方式を用いて、サービスのアイデアを洗練させた。また、Slack のワークスペースに誰でもいつでも自由にアイデアを投稿できるチャンネルを作成し、ユニークなアイデアを考案、発表できる機会を増やすことができた。また OST をしながら、自身の興味のあるアイデアに対し議論することでアイデアをより深めていくことができた。これらの活動によって、多くのアイデアの中から、メンバーのそれぞれの良いと思った部分を組み合わせ、ブラッシュアップを行った。

前期最後の中間発表では、ポスターと動画を作成し発表した。ポスターではメンバー全員で文章を添削し、より伝わりやすいポスターの作成に努めた。動画ではスライドを用いてプロジェクトを

説明した。より多くの人に興味を持ってもらうために、本プロジェクトのオリジナルキャラクターを解説役として起用した。さらに、動画内で合成音声を用いることでより印象に残る動画を目指した。この中間発表において現時点で上がっているサービスを発表し、聴衆にアンケートを取った。そして、このアンケートの結果を参考に後期でサービスの決定をしていくことになった。

(※文責: 宮田悠治)

8.2 後期の振り返り

後期の活動では、始めに、夏季休暇期間中にプロジェクトメンバー間で絞り込みをしたアイデアのレビューを担当教員と TA を交えて行った。それを踏まえサービスを決定した。その結果、「MET.」、「でももに!」、「俺を食ってくれえ!!」の3サービスを開発することとなった。その後、今後どのようにプロジェクト活動を進めていくのか今一度プロジェクトメンバー全員で話し合い、認識の共有を図った。

前期は週ごとにファシリテーターを設定して活動を行っていたが、ファシリテーターを担当している週と担当していない週とで責任感に偏りがあったり、ファシリテーターを担当している人に責任を投げたままがちになったりと、責任、平等という点で問題が発生していた。そのため、後期はプロジェクトリーダーを置いて活動するか否かについても、話し合った。話し合いの結果、プロジェクトリーダーを置いてしまったら前期の活動の否定になってしまうのではないかと、1人に責任を押しつけてしまうことになりかねないところが不安だといった、これまで通りリーダーを置かずに活動するべきと考える意見と、リーダーが責任を持ったほうが楽なのではないかなど、リーダーを置いて活動すべきと考える意見の両方が見受けられた。そこから新たに「大臣制度」を導入しようという結論に至った。大臣制度を導入することによって、これまでのリーダーを置かずに活動するというを守りつつ、メンバー一人一人が何かしらの担当大臣となり責任を担うことで誰か一人に責任が偏らないような形で活動していくことが可能となった。

ほかにも、担当教員や TA の方から自分たちは何をして欲しいのか、どういう立ち位置でいて欲しいのかについてやプロジェクト活動における主役は誰なのかなど活動の根幹となる部分について話し合うことができ、気持ちを新たにそれぞれのサービスでの開発を始めることができた。

後期も前期と同様に、アジャイル開発手法のスクラムに則り、活動と開発を進めていった。また、基本的にオンラインの活動で、必要に応じてオフラインで活動を行った。私たちのグループでは、開発を進めていくにあたって、デイリースクラムを毎日12時から15分ほど行った。また、プロダクトオーナー (PO)、スクラムマスター (SM) のほかにサブプロダクトオーナー (SPO) とサブスクラムマスター (SSM) を設定した。SPO を設定した理由は、PO はサービスのことを熟知しているが、実際に開発するにあたって、技術的な面で全てを熟知しているわけではないため、技術的な面でのサポート役として SPO がいたら開発が円滑に進むと考えたためである。SSM を設定した理由は、SM といえど、スクラムについて完全に熟知しているわけではなく、勉強しながら手探りの状態で進めていくため、状況を俯瞰して見ることができる人を SSM として設定することで、うまくスクラムを回していけると考えたからである。実際、SPO と SSM を設定して開発を行った結果、サービスを作り上げていくうえで細かく役割を分担することができ、開発が円滑に進

めていけることができたのではないだろうか。

私たちが開発を行う中で一番苦労したことは、サービスの方向性を決定することであった。当初は、ユーザーの対象が広く設定しており、ほかのユーザーと協力してパズルのピースを集める「協力」の部分が一番大事にしたいのか、「思い出を残す」という部分を大事にしたいのか、など方向性がぶれていた。この問題は、担当教員や TA から自分たちのサービスについてのレビューによって、対象を子どもがいる家族向けに絞ることで解決した。これによってサービスの方向性が決まり、達成したいことや搭載すべき機能を明確にすることができた。サービスの実装においては、Discord や Slack などのツールを用いてこまめに情報共有を図りながら全員で協力し開発を進められたと考えている。

(※文責: 荻ノ沢実佑)

8.3 今後の展望

私たちが実装したサービスは現段階では、課題がいくつか存在する。まず、iOS と Android で実装している機能が異なることである。私たちのサービスの機能を大きく分けると、「なぞなぞ一覧を表示」、「ビーコンを検知」、「取得ピース表示」の3つが存在するが、このうち iOS は「ビーコンを検知」と「取得ピースを表示」の2つを実装しており、Android は「ビーコンを検知」「なぞなぞ一覧を表示」の2つを実装している。また、詳細な機能についても幾つか実装できていないものがある。例えば、「取得したピースのデータをローカルデバイスに保存できる」という機能である。この機能はサーバーを使用すること、つまり、インターネットを使用することなくサービスを使用できるということを想定しており、ユーザーからも取得したピースの情報は観光の思い出を振り返るうえで重要だと考え、実用最小限の製品、MVP (Minimum Viable Product) を構成する1つの機能としてとして挙げていたが、実装することができなかった。今後は2つのプラットフォームで実装部分をそろえてだけでなく、詳細な機能についても実装を進めていくことで、全ての機能を実装するのではなく、それぞれの機能において、バグなどを少なくしていくことで品質を向上させることを今後の開発目標として挙げておく。

次に、パネル部分の実装が不十分であるということである。私たちのサービスは函館の各地を回って集めたピースをパズルを表示しているパネルへと集めることを想定しているが、このパネル部分の実装が現段階ではピースががはまるデモを Web アプリケーションで作成した状態で、実際にアプリケーションからパネルへとピースを送ることはできていない。今後は、パズル画面の表示方法を決定し、アプリケーションとパネルの連携方法について話し合いを行う。最終的に機能としてパネル専用のビーコンを検知している間はいつでもピースの送信を可能にすることを今後の開発目標として挙げておく。

また、パネルの設置場所についての地域との連携をとる必要がある。本サービスは函館に観光に来る家族を主に対象としているサービスのため、観光客が使用すると考えられる、函館空港や函館駅に設置することを想定していた。今後はパネルを函館空港や函館駅に設置して、実際に函館のまちを散策してピースを集めて、空港や駅に戻ってきて、ピースをはめるという一連の流れを体験できることを展望として述べる。

以上の機能を実装することや、サービスの環境整備を行い、本サービスを利用してもらうことで、家族全員での観光と思い出を残すことへのサポートをして、函館観光をより楽しむことができることを願っている。

(※文責: 山本雄平)

参考文献

- [1] 総務省 | 平成 30 年版 情報通信白書 | IoT デバイスの急速な普及,
<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd111200.html>
(2021 年 1 月 8 日閲覧)
- [2] Beacapp Here(屋内位置情報サービス), <https://jp.beacapp-here.com/> (2021 年 1 月 7 日
閲覧)
- [3] Coke ON (コーク オン), <https://c.cocacola.co.jp/app/> (2021 年 1 月 7 日閲覧)
- [4] 函館市観光基本計画策定調査平成 25 年 3 月,
<https://www.city.hakodate.hokkaido.jp/docs/2014032500399/files/siryou250515.pdf>
(2020 年 12 月 26 日閲覧)
- [5] 函館市総合計画 (2017~2026) について,
<https://www.city.hakodate.hokkaido.jp/docs/2016122100014/> (2021 年 1 月 8 日閲覧)
- [6] 【サルでもわかる BLE 入門】(1) BLE の基礎,
<https://www.musen-connect.co.jp/blog/course/trial-production/ble-beginner-1/>
(2021 年 1 月 13 日閲覧)
- [7] iBeacon とは | 電源設計の技術情報サイトの TechWeb,
<https://techweb.rohm.co.jp/iot/knowledge/iot02/s-iot02/04-s-iot02/3896> (2021 年 1
月 13 日閲覧)
- [8] 【サルでもわかる BLE 入門】(3) BLE ビーコンの基礎,
<https://www.musen-connect.co.jp/blog/course/trial-production/ble-beginner-3/>
(2021 年 1 月 13 日閲覧)
- [9] アジャイルソフトウェア開発宣言, <https://agilemanifesto.org/iso/ja/manifesto.html>
(2021 年 1 月 6 日閲覧)
- [10] Jeff Sutherland & Ken Schwaber, スクラムガイド - Scrum Guides,
<https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Japanese.pdf>
(2021 年 1 月 6 日閲覧)
- [11] 立命館大学, ブレーンストーミングと KJ 法, <http://www.ritsumei.ac.jp/yamai/kj.htm>
(2020 年 12 月 23 日閲覧)
- [12] OST(オープン・スペース・テクノロジー), HUMAN VALUE,
<https://www.humanvalue.co.jp/keywords/ost/> (2020 年 1 月 20 日閲覧)
- [13] 沿革 — エポック社, https://epoch.jp/epoch_info/outline.html (2021 年 1 月 9 日閲覧)
- [14] Apple Developer Xcode, <https://developer.apple.com/jp/support/xcode/> (2021 年 1 月
9 日閲覧)
- [15] Android Developers Android Studio の概要,
<https://developer.android.com/studio/intro?hl=ja> (2020 年 12 月 27 日閲覧)

- [16] Android Developers アプリ アーキテクチャ ガイド,
<https://developer.android.com/jetpack/guide?hl=ja#recommended-app-arch> (2020 年 12 月 27 日閲覧)
- [17] Android Developers Android KTX, <https://developer.android.com/kotlin/ktx?hl=ja>
(2020 年 12 月 27 日閲覧)
- [18] Android Developers LiveData の概要,
<https://developer.android.com/topic/libraries/architecture/livedata?hl=ja> (2020 年 12 月 27 日閲覧)
- [19] GitHub AltBeacon/android-beacon-library README,
<https://github.com/AltBeacon/android-beacon-library> (2020 年 12 月 27 日閲覧)
- [20] Android Developers Localbroadcastmanager
<https://developer.android.com/jetpack/androidx/releases/localbroadcastmanager?hl=ja>
(2021 年 1 月 20 日閲覧)
- [21] Android Developers RecyclerView
<https://developer.android.com/guide/topics/ui/layout/recyclerview?hl=ja> (2020 年 1 月 20 日閲覧)
- [22] Android Developers CradView
<https://developer.android.com/guide/topics/ui/layout/cardview?hl=ja> (2020 年 1 月 20 日閲覧)
- [23] Github bumptech/glide <https://github.com/bumptech/glide> (2020 年 1 月 20 日閲覧)
- [24] GitHub google/gson README, <https://github.com/google/gson> (2020 年 12 月 27 日 閲覧)
- [25] Home — phina.js, <https://phinajs.com/> (2021 年 1 月 20 日閲覧)
- [26] ingectar-e: 見てわかる, 迷わず決まる配色アイデア 3 色だけでセンスのいい色 (2020 年)
- [27] FONT FREE じゅうちょうフォント, <https://fontfree.me/1541> (2020 年 12 月 26 日閲覧)
- [28] FONT FREE ほにゃ字, <https://fontfree.me/1765> (2020 年 12 月 26 日閲覧)
- [29] FONT FREE ゆるもじ, <https://fontfree.me/1921> (2020 年 12 月 26 日閲覧)
- [30] FONT FREE ニコ角フォント, <https://fontfree.me/1393> (2020 年 12 月 26 日閲覧)
- [31] FONT FREE ニコカフォント, <https://fontfree.me/2353> (2020 年 12 月 26 日閲覧)
- [32] FONT FREE プぷプ, <https://fontfree.me/2977> (2020 年 12 月 26 日閲覧)
- [33] FONT FREE JK ゴシック M, <https://fontfree.me/1276> (2020 年 12 月 26 日閲覧)
- [34] FONT FREE うずらフォント, <https://fontfree.me/1248> (2020 年 12 月 26 日閲覧)
- [35] kawaii 手書き文字, <http://font.spicy-sweet.com> (2020 年 12 月 26 日閲覧)

付録 A 中間発表で使用したプロジェクト概要のポスター

2020.07.17 中間発表会



Project No. 07

ビーコン IoT で函館のまちをハックする

Leverage the Beacon IoT for Our Smarter Life in Hakodate Real Downtown

Members

鈴木 利武 Ryuji Suzuki	袴田 結女 Yui Hachimada	久末 瑠紅 Ruka Kuumatsu	熊谷 浩平 Kohpei Kumagai	増田 翔 Sho Masuda
野間 直生 Naoki Nomura	大石 晃平 Kohei Oishi	村石 拓海 Takumi Muraiishi	伊藤 直樹 Naoki Ito	萩ノ沢 実佑 Miyu Hagi-nozawa
平山 翔真 Shohei Hirayama	石澤 大輔 Daikichi Ishizawa	宮田 悠治 Yuuji Miyata	山本 雄平 Yuuhei Yamamoto	久米田 羽月 Hazuki Kunita

Teachers

松原 克弥 Katsuyuki Matsubara	藤野 雄一 Yuuichi Fujino	鈴木 昭二 Sho-ji Suzuki	奥野 拓 Takuro Okuno	鈴木 恵二 Kei Suzuki
------------------------------	-------------------------	------------------------	----------------------	---------------------

Advisors

南部 美砂子 Misako Nanbu	佐藤 生馬 Ikuma Sato
------------------------	---------------------

プロジェクト概要 Summary

目的 Purpose
ビーコンを函館のまちの様々な場所に設置し、ビーコンを活用したサービスを開発する。それによって、新たな価値を生み出し、利用する人にこれまでにない体験や魅力を提供することが目的である。
Install beacons in Hakodate real downtown, and develop services that utilize beacons. The purpose is to create new value and provide people unique experiences and attractions.

ビーコンの特徴 Feature of Beacon

- スマートフォンで受信可能な Bluetooth 電波を発している小さなデバイス
- バッテリー消費の少ない BLE (Bluetooth Low Energy) を使用
- ビーコンから受信端末までの距離の計測、個体の判別が可能
- A kind of device transmitting Bluetooth radio wave which can be received by smartphones
- Uses BLE (Bluetooth Low Energy) which is low battery consumption
- Able to measure distance between the beacon and the receiver, identify individual devices

開発サービスのアイデア出し Making Ideas of the Service

フィールドワーク Fieldwork
函館のまちの課題を発見するために、以下の場所でフィールドワークを行った。また、感染症対策として、メンバーは3人ずつ、計5グループに分けた。
- 赤川・美原・昭和・富岡 - 五稜郭 - 七飯・北斗
- 函館駅・西部地区 - 函館空港・湖川
We did fieldwork for finding issues in Hakodate local features. For preventing infection disease, we made five groups of three. Each group did fieldwork at the following locations:
- Akagawa, Mihara, Showa, Tomioka - Goryokaku - Nanae, Hokuto
- Hakodate Sta., Seibu-ward - Hakodate APT., Yunokawa

仮アイデアの作成 Making Candidate Ideas
メンバー全員が以下のいずれかの手法を用いて、仮アイデアを出した。
- ブレインストーミングと KJ 法を用いた手法
- ハッカソン風に、時間制限を設けて、アイデアを絞り出す手法
次に、Open Space Technology (OST) という手法を用いて、それぞれのグループで出たアイデアのブラッシュアップ、選定を行った。
We made candidate ideas using one of following methods:
- Brainstorming and KJ method
- Hackathon style; make ideas within a limit time
Then, we brushed up and selected the ideas using Open Space Technology (OST).



今後の予定 Future Plan

8月	9月	10月	11月	12月	1月
・設計 Design ・技術習得 Learn tech.	・技術取得 Learn tech. ・実装 Implement	・実装 Implement	・実装 Implement ・評価実験 Experiment for evaluation	・実装 Implement ・評価実験 Experiment for evaluation	・最終報告書提出 Submit final report

独自のタスク管理手法 Original Task Management Method
アジャイル的なタスク管理手法を確立
- プロダクトバックログを作成し、タスクを可視化して共有
- リーダーを決めず、週ごとに2人の進行役を決め、交代
- 1週間ごとにタスクを計画、実行、振り返りを行う
- 活動の流れを事前にすべて計画する手法に比べ、計画の変更や状況の変化に対して柔軟な対応が可能
We developed an original method for managing tasks following the agile method.
- Maintain product backlog to visualize and share our tasks
- Assign two facilitators every week instead of the leader
- Make a plan, do tasks and review a sprint every week
- Able to change contents of tasks flexibly compared to all-plan-decided management



フィールドワークに向けた事前準備 Preparing for Fieldwork
フィールドワークに関するレクチャ動画を視聴した。さらに、函館山にてプレフィールドワークを行い、課題を発見するための観察と記録の手法を学習した。
We took a video lecture about fieldwork. In addition, we did pre-fieldwork at Mt. Hakodate and learned ways to observe and record the local features for finding out issues.





仮アイデアの一覧 List of Candidate Ideas

1. 俺を食べてくれ! - Eat! Eat! Eat me!!!
海産物の状態・部位によって、最適な調理方法などを表示してくれる。
This app recommends how to properly cook the seafood depends on freshness or parts.
2. カラオケ × ビーコン - Karaoke × Beacon
部屋に入るだけで、注文や清算をスマホを通じてできる。
You can order something and check just using smartphone in Karaoke room.
3. 観光客 × 地元民 - Sightseer × Local people
観光客が地元をよく知る人を、ビーコンで探して質問できる。
Sightseer can ask any questions about Hakodate to local people found using beacon.
4. 函館のここ、おすすめかも? - Recommend Here in Hakodate
行動履歴をもとに、おすすめのお店を紹介する。
This app suggests you recommended shops based on places you visited.
5. 目の前のイベントは何? - What is the event in front of you?
大道芸などの突発的なイベントに対して、その情報を得ることができる。
You can get information about sudden events such as street performance.
6. クーポン長者 - Coupon Millionaire
お店を訪れた人に他のお店のクーポンを配り、函館のお店の活性化を図る。
You can get coupons of other shops when visit the shop toward regional shops revitalization.
7. 五稜郭地下歩道の活性化 - Revitalizing the Goryokaku Underpass
地下歩道のマップや、周辺店舗のキャンペーンを表示する。
This app shows underpass' map, and event information of shops around there.
8. 未来大のライブラリをよくしよう! - Get Improve FUN Library!
見つけたい図書を検索して、その位置を教えてくれる。
This app gives you location of the book shelf you are looking for.

図 A.1 中間発表で使用したポスター

付録 B 成果発表で使用したプロジェクト概要のポスター

2020.12.04 成果発表会

Project No. 07

ビーコンIoTで函館のまちをハックする

Leverage the Beacon IoT for Our Smarter Life in Hakodate Real Downtown

Members

鈴木 利武 Tom Suzuki 神田 結女 Yume Katsuda 久末 瑞虹 Ryoko Kuumoto 熊谷 浩平 Hiroshi Kumagai 増田 翔 Shou Masuda

野間 直生 Naoki Nomura 大石 昌平 Masahiro Oishi 村石 拓海 Takumi Murakami 伊藤 直樹 Naoki Ito 荻ノ沢 実佑 Mitsuhiro Hino

平山 翔真 Shouma Hirasawa 石澤 大輔 Daisuke Ishizawa 宮田 悠治 Yuzo Miyata 山本 雄平 Yuzo Yamamoto 久米田 羽月 Hazumi Kumeta

Teachers

松原 克弥 Kazuyuki Matsubara 藤野 雄一 Yuzo Fujiwara 鈴木 昭二 Shoji Suzuki 奥野 拓 Taku Okuno 鈴木 孝二 Takahiko Suzuki

Advisors

南部 美砂子 Misako Nanbu 佐藤 生馬 Naomasa Sato

プロジェクト概要 Summary

目的 Purpose

ビーコンを函館のまちの様々な場所に設置し、ビーコンを活用したサービスを開発する。それによって、新たな価値を生み出し、利用者にこれまでにない体験や魅力を提供することが目的である。

The purpose is to create new value and provide people unique experiences and attractions. We install beacons in Hakodate real downtown, and develop services that utilize beacons.

ビーコンの特徴 Features of Beacon

- スマートフォンで受信可能な Bluetooth で通信を行う小さなデバイス
- バッテリー消費の少ない BLE (Bluetooth Low Energy) を採用
- ビーコンから受信端末までの距離の計測、個体の識別が可能
- Beacon is a small device that communicate via Bluetooth which can be received by smartphones
- The devices uses BLE (Bluetooth Low Energy) which is low battery consumption
- The device is able to measure distance between it and the receiver, identify individual devices

活動内容 Activities

5: アイสบレイク Break the ice, ロゴの作成 Create logo mark
6: フィールドワーク Field work, ロゴの決定 Decide on logo mark
7: アイデア出し Brainstorm ideas, 中間発表 Mid-term presentation
8: アイデア出し Brainstorm ideas, 技術習得 Learn tech.
9: アイデア出し Brainstorm ideas, 技術習得 Learn tech.
10: サービス決定 Decide the services, 実装 Implement
11: 実装 Implement
12: 最終報告書提出 Submit final report, 成果発表会 Accomplishment presentation

開発について About Development

開発手法 Development method

小さな成果物を定期的に取り続ける「アジャイル手法」に挑戦した。また、オンラインでの活動が主であったため、Discord や Miro, Google Jamboard など、様々なツールを利用した。

We challenged the "agile method" which keeps making small deliverables. We also used various tools such as Discord, Miro, and Google Jamboard since our primarily activity was online.

開発フロー Development flow

1週間で「計画」、「実装」、「振り返り」を繰り返した。

We repeated planning, implementation, and review in one week.

チーム構成 Team formation

アジャイル手法のスクラムをベースにチームを構成した。

- 各サービスごとに3名、または6名のスクラムチームを形成
- 各チームごとにプロダクトオーナー (PO) とスクラムマスター (SM) を設置
- PO: プロダクトの価値を最大化し、プロダクトそのものの責任を持つ。
- SM: チーム外からの妨害の排除や、ファシリテーターなどを担い、チームを支える役割を果たす。

We teamed up based on scrum of agile method.

- Each service forms a scrum team of 3 or 6 people
- Each team has a Product Owner (PO) and a Scrum Master (SM)
- PO: Maximize the value of the product and take responsibility for the product itself.
- SM: Eliminates interference from outside the team, plays a role as a facilitator, and plays a role in supporting the team.

提案サービス Proposed Services

MET.
Meet Experiences on Trips.
思い出を交換して観光客に新たな旅行の楽しみ方を提供する。
Make a new way on trips through exchanging memories.

でももに! Demomoni!
バスルで大人も子どもと一緒に函館観光!
家族みんなでバスルのピースを集めて、思い出を観光地に残す。
Collect the pieces of the puzzle with the whole family and leave the memories on the sights.

俺をくってくれえ!! Eat Eat Eat Me!!
おさかな選びに革命を!!
海産物自身が鮮度に合わせた調理法をアピールしてくれる。
Seafoods appeal the recipe depending on the freshness themselves.

学び Lessons Learned

- オンラインの活動では、オフラインの活動に比べて表情や反応が分かりにくいため、自分の意見を分かりやすく伝えることが特に大切である。
- 心理的安全性を意識し、互いの意見を尊重した発言をすることで、コミュニケーションを取りやすい環境を作り上げることができる。
- リーダーを決めなくとも、担当者を細かく割り振ることで責任を分担することができる。
- メンバー同士で、提供するサービスのイメージをしっかりと共有することによって、作るものが明確化し、開発に入っても迷いなく進めることができる。
- チーム内で問題が発生した場合は、チーム全体に共有し、話し合うことが重要である。
- In online activities, it is especially important to tell one's opinion in a clearer way than offline activities.
- Conscious psychological safety, and speaking with respect make the environment for having conversations.
- Weather not deciding on a leader, responsibilities can be distributed if assigning roles appropriately.
- We can make and proceed the products without hesitation with making clear the image of the services and sharing it.
- When some issue arise, it is important to share and discuss with the whole team.

図 B.1 成果発表で使用したポスター

付録 C オープンソースライセンス

AltBeacon

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2014 Radius Networks

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Gson

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"

replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

glide

License for everything not in third_party and not otherwise marked:

Copyright 2014 Google, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY GOOGLE, INC. 'AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL GOOGLE, INC. OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of Google, Inc.

License for third_party/disklru-cache:

Copyright 2012 Jake Wharton
Copyright 2011 The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

License for third_party/gif_decoder:

Copyright (c) 2013 Xcellent Creations, Inc.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for third_party/gif_encoder/AnimatedGifEncoder.java and third_party/gif_encoder/LZWEncoder.java:

No copyright asserted on the source code of this class. May be used for any purpose, however, refer to the Unisys LZW patent for restrictions on use of the associated LZWEncoder class. Please forward any corrections to kweiner@fmsoftware.com.

License for third_party/gif_encoder/NeuQuant.java

Copyright (c) 1994 Anthony Dekker

NEUQUANT Neural-Net quantization algorithm by Anthony Dekker, 1994. See "Kohonen neural networks for optimal colour quantization" in "Network: Computation in Neural Systems" Vol. 5 (1994) pp 351-367. for a discussion of the algorithm.

Any party obtaining a copy of these files from the author, directly or indirectly, is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

phina.js

MIT License

Copyright (c) 2015 phi and other contributors, <http://phinajs.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

うさライセンス

うさについて

<https://github.com/ogi-san/usa-license/blob/main/README.md>

「うさ」とはうさぎとビーコンを組み合わせたオリジナルキャラクターのことで、本キャラクターは荻ノ沢実佑が制作しました。

利用方法について

【利用規約】

改変なし・ありの再配布・再販売、自作発言は不可です。

非営利目的での利用である場合、自由です。

公序良俗に反しない範囲で利用可能です。

クレジット表記も不要です。

(アバター OK/二次創作 OK/暴力・グロテスク表現 NG/差別表現 NG/性的表現 OK/商用 NG)