

# 公立はこだて未来大学 2015 年度 システム情報科学実習 グループ報告書

Future University Hakodate 2015 System Information Science Practice  
Group Report

## プロジェクト名

フィールドから創る地域・社会のためのスウィフトなアプリ開発

## Project Name

“Swift” Application Development Based on Field Research

## グループ名

教育系グループ

## Group Name

Education Group

## プロジェクト番号/Project No.

3-C

## プロジェクトリーダー/Project Leader

1013220 新保遥平 Yohei Shinpo

## グループリーダ/Group Leader

1013015 中進吾 Shingo Naka

## グループメンバ/Group Member

1013130 熊谷優斗 Yuto Kumagai

1013116 皂勢也 Seiya Kurokome

1013220 新保遥平 Yohei Shinpo

1013015 中進吾 Shingo Naka

1013104 矢吹溪悟 Keigo Yabuki

## 指導教員

伊藤恵 奥野拓 原田泰 木塚あゆみ 南部美砂子

## Advisor

Kei Itou Taku Okuno Yasushi Harada Ayumi Kizuka Misako Nambu

## 提出日

2015 年 7 月 29 日

## Date of Submission

July 29, 2015

## 概要

本プロジェクトは教育というフィールドを調査し、教育に関する問題を解決するアプリケーションを開発することを目的としている。

各メンバーが教育に関わるアプリを考え、メンバーと担当教員にプレゼンテーションを行った。メンバー間では情報の共有を行い、担当教員からはレビューを受けた。その後、担当教員から受けたレビューを基にお互いにアイデアを広げ、テーマを1つに絞り込み大学生向けプログラミング入門アプリに決まった。

テーマが決まった後、アプリの設計を行った。しかし、要件定義を固めずにアプリの設計を行ったため、一貫性のないアプリ設計になってしまった。そのため、要件定義を1からやり直すことになった。要件定義をやり直すことは、1度で終わらず何度も行った。その結果、中学校でプログラミングを学んだ人、興味を持った人を対象にしたゲームアプリというテーマに決めた。

現在日本の中学校では、2012年から中学校の技術家庭科でプログラミング教育が必修項目となっている。しかし、今の中学校のプログラミング教育ではソースコードを打ってプログラミングをするということを行っておらず、プログラミングの内容を深く取り上げていない。そこで、私たちは中学で学んだプログラミングと実際のプログラミングの間のプロセスを支援するゲームアプリを開発することを決めた。

中間発表では、私たちが考えた提案をポスターにまとめ、ポスターセッションを行った。教員や他学生からの評価シートには「最終的なゴールは?」、「まだ内容が決まっていないので評価不能」、「既存のものとの比較がない」などの意見をいただき、もう1度要件定義を見直しアプリの設計をやり直す必要があることに気付かされた。

後期は、夏休みにメンバー各自で調べてきた教育のアイデアを出し合い、後期のプロジェクトのテーマについて何度も話し合った。その結果、C言語を学んでいる大学生を対象としたC言語のWebアプリケーションの学習教材を開発するテーマに決めた。

現在、C言語を学んでいる公立はこだて未来大学の学生とその講義を担当しているTA(Teaching Assistant)にヒアリング調査を行った。その結果、C言語の配列などの概念やアルゴリズムが分かっていないことが分かった。また、現在使われている講義資料を確認してみたところ、ほとんどが文字で構成されており、専門用語が多くあり、C言語を初めて学ぶ学生にとっては分かりにくい講義資料だと思った。そこで、私たちはこれらの課題を解決するためアニメーションで、概念やアルゴリズムを教えるWebアプリケーションを開発することを決めた。

最終発表では、中間発表と同様に私たちが考えた提案をポスターにまとめ、ポスターセッションを行った。教員や他学生からの評価シートには「イメージがわくので、分かり易かったです」などの意見をいただき前期と比べて評価は高かった。一方で、「実際に1年生に使ってほしかった」、「他の班と比べてiOSアプリを作らなかったメリットを知りたい」などの意見をいただき、今後の改善点を見つけることができた。

今後の展望としては、私たちが作成した提案物と酷似したソフトウェアが本学の2年次の科目である「アルゴリズムとデータ構造」の教科書の付属CDにあったため、そのソフトウェアとの差別化を図っていきたいと思う。また、ユーザーに評価していただくために本学の1年生やメタ学習センターに実際に使っていただきたいと思う。

( 文責: 中進吾 )

# Abstract

This project is having for its object to investigate a field as education and develop the application useful for education.

Each member considered the application of educating and presented members and teachers. We shared information among the members and received reviews from teachers. After those ideas was expanded each other and the theme was narrowed down to 1 based on the review we received from teachers. The theme was decided in programming guide application for college students.

After the theme was decided, the application was designed. But the design of application had inconsistent because the application was designed without making the requirement definition hard. Therefore we changed the requirement definition from one. We did not finish changing the requirement definition and went many times. As a result, it was decided in the theme as the game application that made the person who learned a programming at junior high school and interested people the subject.

Programming education is the compulsory item at technical and homemaking course of Japanese junior high school from 2012. But neither to hit source code with programming education of the junior high school and program be being performed nor the contents of programming be taken up deeply now. So we have decided to develop the game application which supports process between the programming learned at junior high school and the actual programming.

The proposition that we thought was gathered in a poster and a poster session was performed in the middle announcement. We received opinions of which “what is last goal”, “having no comparison it exists down”, “the contents are not decided yet, so evaluation is impossible” in an evaluation seat from teachers and other students, and the requirement definition was reconsidered again, and they made notice that it is necessary to redo design of an application.

In the latter period, educational ideas that members found in the summer holidays was shared and we discussed theme of project many times. As a result, The theme was decided in Web application teaches C language for college students learning it.

We investigated Future University HAKODATE students learned C language and Teaching Assistant took the lecture now. As a result, we knew that they do not know array’s concept of C language and algorithm. We think that the lecture material used was difficult for them because it was made of most characters and had professional words when we checked it. We decided that we made Web application teaches concept and algorithm in animation to resolve these problems.

The proposition that we thought was gathered in a poster and a poster session was performed like the first term in the final announcement. We received opinions of which “it is easy because I can imagine” in an evaluation seat from teachers and other students, and the evaluation of the latter period was better than it of the first period. Also, we received opinions of which “I hope this Web site was used by 1st grader”, “Compared with other groups, I want to know the merit of not making iOS application” and we knew the improvement points of future.

I would like to plan for differentiation with the software because the suggestion thing we made and software I resemble closely matched an accessory CD of the textbook of “algorithm and data structure” which is 2 grader of subject of the science as future’s view. I would like to ask the user to use it for 1st grader of this college and the META learning center actually to estimate.

( 文責: 中進吾 )

# 目次

第 1 章	はじめに	1
第 2 章	前期の背景	2
2.1	世界と日本のプログラミング教育について	2
2.2	現状と課題	3
第 3 章	前期のプロジェクトの目標	4
3.1	開発アプリの目標	4
3.2	プロジェクト学習としての目標	4
第 4 章	前期の活動	5
4.1	プロジェクト全体としての活動	5
4.1.1	スクラッチワークショップへの参加	5
4.1.2	リスク分析	5
4.1.3	アプリ開発のための勉強会	6
4.1.4	バックログの作成	7
4.1.5	中間発表会の資料制作	8
4.1.6	中間発表会	8
4.2	アプリ案の変化と内容	9
4.2.1	アプリ案の検討	9
4.2.2	大学生向けプログラミング入門アプリ	10
4.2.3	中学生向けプログラミング支援アプリ	12
第 5 章	前期の開発アプリについて	13
5.1	概要	13
5.2	プログラミング画面	13
5.3	戦闘画面	16
5.3.1	ゲーム性	16
5.3.2	教育性	16
第 6 章	前期の結果	18
6.1	プロジェクトの評価	18
6.2	プロジェクトの成果	18
第 7 章	後期の背景	20
7.1	プログラミング基礎の現状	20
7.2	現状と課題	20
第 8 章	後期のプロジェクトの目標	21
8.1	開発アプリの目標	21

8.2	プロジェクト学習としての目標	21
<b>第 9 章</b>	<b>後期の活動</b>	<b>22</b>
9.1	イベント	22
9.1.1	後期キックオフ会	22
9.1.2	アカデミックリンク	22
9.1.3	最終成果発表会	22
9.2	アプリ案の推移	22
9.2.1	夏休みの課題と調査報告	22
9.2.2	Processing デバッガツール	22
9.2.3	C-mation	22
<b>第 10 章</b>	<b>後期の開発アプリについて</b>	<b>23</b>
10.1	概要	23
10.2	コンテンツ	23
10.2.1	概念の説明	23
10.2.2	例題	24
10.2.3	確認問題	25
<b>第 11 章</b>	<b>後期の結果</b>	<b>30</b>
11.1	プロジェクトの評価	30
11.2	プロジェクトの成果	30
<b>第 12 章</b>	<b>今後の課題と展望</b>	<b>31</b>
12.1	開発アプリの課題と展望	31
12.2	メンバーの課題と展望	31
<b>第 13 章</b>	<b>学び</b>	<b>33</b>
13.1	グループとしての学び	33
13.2	各メンバーの学び	33
13.2.1	熊谷優斗	33
13.2.2	梶勢也	34
13.2.3	新保遥平	34
13.2.4	中進吾	35
13.2.5	矢吹溪悟	35
<b>付録 A</b>	<b>新規習得技術</b>	<b>36</b>
<b>付録 B</b>	<b>活用した講義</b>	<b>38</b>
<b>参考文献</b>		<b>40</b>

## 第 1 章 はじめに

私たち教育系チームはこの 1 年間を通して、プログラミング教育という視点でアプリの開発を行ってきた。自分たちのアプリでプログラミングを理解して楽しんでもらおうという目的であった。私たちのグループでは前期から後期を通して幾度となく、アプリ案の変更があった。前期では Swift 言語によるアプリ案を考えてきた。だが後期からは html や Processing を用いた Web アプリケーションの開発に移行していった。第 2 章から第 6 章までが前期の活動である。前期では、ユーザを意識したアプリ案が少なく自分たちの作りたいアプリ案を多く提案していた。また、実装まで進めた後でアプリのコンセプトの見直しなどが多くあった。そのため、アプリ案の再設計というのが前期の教育系の進捗であった。第 7 章から第 11 章が後期の活動である。後期は前期の反省を踏まえ、アプリ案の提案に慎重になりすぎてしまい、なかなかアプリ案の確定、実装の工程に進むことができなかった。だが最終成果発表会までには自分たちの納得のいく成果物を作ることができた。

( 文責: 新保遥平 )

## 第 2 章 前期の背景

### 2.1 世界と日本のプログラミング教育について

現在、世界中でプログラミング教育の必要性が高まっている。政府が公教育としてプログラミングを取り入れている、または取り入れようとしている国が増えてきている。イギリスでは、5 歳から 16 歳の義務教育の新カリキュラムにプログラミングが正式導入されており、エストニアでは小学校 1 年生からアプリ開発の授業を開始することになっている [1]。日本でも 2012 年から中学校の技術家庭科で、プログラミング教育が必修項目となっている [2]。

日本では、ビジュアルプログラミング言語の図 1.1 の Scratch や図 1.2 のビュートビルダーなどを用いて、ラジコンなどの機械を動かす授業を行っている [3][4][5]。



図 2.1 Scratch の画面

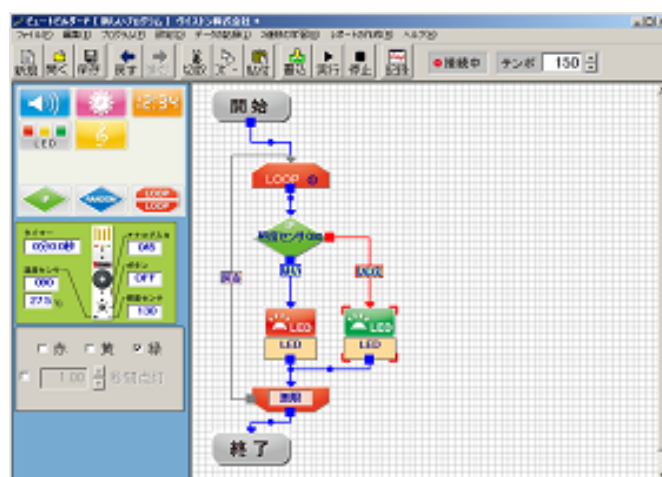


図 2.2 ビュートビルダーの画面

また、2013 年 6 月 5 日に安倍政権の経済政策「アベノミクス」の「第 3 の矢」として発表した成長戦略の素案には、「産業競争力の源泉となるハイレベルな IT 人材の育成・確保」という項目が

あり、その中には「義務教育段階からのプログラミング教育等の IT 教育を推進する」との記載があった [1]。今後、日本のプログラミング教育はさらに拡大していくことが予想される。

しかし、今の中学校のプログラミング教育ではソースコードを打ってプログラミングをするということを行っておらず、プログラミングの内容を深く取り上げていない。また、プログラミングを学べるのは中学校 3 年生の時だけで、イギリスやエストニアと比べるととても短い期間である。高校では、実際にプログラミングを教えているところもあるが、義務化されていないので誰もが学校でプログラミングを学べるわけではない [6]。

( 文責: 中進吾 )

## 2.2 現状と課題

日本の中学校ではビジュアルプログラミング言語を用いたプログラミングの授業を行っており、ソースコードを書く練習は行っていない。ビジュアルプログラミング言語は C 言語や Java のようなプログラミング言語と表記の仕方が大きく異なっている。そのため中学校の授業だけでは、C 言語や Java のように実際に文字を打ち込むようなソースコードを組もうとした時、どのように組んでいいかわからない。Web サイトやアプリなどのシステム開発を行う際、基本は文字を打ち込むプログラミング言語を用いるので、ビジュアルプログラミング言語はほとんど使用しない。今の中学校のプログラミング教育だけでは、産業競争力の源泉となるハイレベルな IT 人材の育成・確保をすることはできない。現状のままでは、イギリスやエストニアなどの他国との差が広がる一方である。

( 文責: 中進吾 )



## 第 3 章 前期のプロジェクトの目標

### 3.1 開発アプリの目標

背景で述べたように、世界中でプログラミング教育の必要性が高まっており、実際に小学生からアプリ開発の授業を行っている国もある。日本でも中学校の技術家庭科でプログラミング教育が必修項目となっている。しかし、現在の中学校のプログラミング教育ではソースコードを打ってプログラミングをするということをしておらず、プログラミングの内容を深く取り上げていない。

そのため、中学校でプログラミングを学んだ人、興味をもった人を対象として、中学で学んだプログラミングと実際のプログラミングの間のプロセスを支援し、ソースコードの組み立てを学ぶことが出来るゲームアプリを開発する。そこで、ビュートビルダーや Scratch のようなビジュアルプログラミング言語を学んだ中学生が、C 言語のように実際に文字を打ち込むようなソースコードの組み方を理解できるようになることが、開発アプリの目標である。

( 文責: 梶勢也 )

### 3.2 プロジェクト学習としての目標

新しいプログラミング言語である Swift 言語を使い、アジャイル開発手法の一つである Scrum という方法論を用いて素早いアプリ開発をする。また、短期間での開発とフィードバックを繰り返し、より良い品質のアプリ開発を目指している。さらに、バージョン管理システムの理念を学び、効率よくアプリ開発する。プロジェクト学習を通して、情報システムコース、高度 ICT コースや情報デザインコースなど異なるコースのメンバーで開発を進めていく。開発を進めていく中で、コミュニケーション能力、グループ開発力を養い、異なる分野の知識を吸収する。

プロジェクト学習としての最終的な目標はアカデミックリンク、成果発表会や課外発表会でアプリの発表を行いレビューを受け、受けたレビューを反映させたアプリをリリースすることである。

( 文責: 梶勢也 )

## 第 4 章 前期の活動

### 4.1 プロジェクト全体としての活動

#### 4.1.1 スクラッチワークショップへの参加

教育をテーマにするに当たり、まず小学生と触れ合い、教育の現状について考えるために、原田先生主催のワークショップに参加した。ワークショップは、2015 年 5 月 9 日に、函館市青年センターにて行われ、小学校 5 年生から中学校 1 年生の子供達計 10 人が参加した。ワークショップの内容は、ビジュアルプログラミング言語「Scratch」を用いて、動きに反応して音が鳴る不思議楽器を作るというものである。当日、メンバーは小学生の側についてプログラミングのアシスタントをした。

ワークショップを通して、気づいた点は次の 2 点である。

- 子供達は、一度得た知識はすぐに自分のものになっているようだった。今回のワークショップは、前回のワークショップ参加者から引き続き参加している子供が多いということもあって、メンバーが使い方を教えるまでもなく、自力でプログラミングを行っていた。更に、繰り返し文の使い方を教えたところ、「じゃあさ、ここもこうすればいいんじゃない？」と、子供自ら別の点の修正を行っていた。子供の成長能力の高さに驚いた。
- 前回から参加している子供に、どうして今回も参加したの？と尋ねたところ、「だって、これ (Scratch) 楽しいんだもん」と答えた。子供でもプログラミングに興味を持っていることに驚いた。

また、ワークショップの最後に、参加者の子供達と、その親に向けた簡単なアンケートを実施した。しかし、プロジェクトとしての方針が決まってない状態で作成したアンケートだったため、内容が建設的なものではなく、得たアンケート結果をその後に生かすことが出来なかった。むしろ、アンケート内容に子供にはわかりづらい表現がある、難しい漢字を使っている、子供用と大人用のアンケート用紙の区別がつかないといった問題を発見できたことが、その後に生かせる学びであったと言える。

( 文責: 熊谷優斗 )

#### 4.1.2 リスク分析

プロジェクトを進めるにあたって起こりうるリスクをメンバーそれぞれで洗い出し、それぞれのリスクに対して発生確率、被害の内容、対処方法を挙げた。図 3.1 は洗い出したリスクの一部である。

リスクの洗い出しをした時点で既に発生していたのが、「メンバーに連絡がつかない」というリスクだ。前述のスクラッチワークショップにてアンケートを実施したが、このアンケートを作成する際、メンバーの 1 人に連絡がつかず、ワークショップ当日になってそのメンバーにアンケート内容のレビューをしてもらった結果、いくつかの不備があることが発覚した。この不備は、

そのメンバーが前日にアンケート内容をレビューできていれば気づけたはずである。今後このようなリスクが発生しないよう、メンバー内で1日1回はSkypeやLINEを確認することを義務づけた。

リスク名	発生確率	被害の内容	対処方法
メンバー間の能力の差が顕著に表れる	大	メンバーの意識の差が広がり、プロジェクトの生産性が滞る。	メンバー全員で協力し成長し合うよう心掛ける。個々が活かせる能力を見つけて、発揮させる
1人で問題を抱え込んでしまう	大	プロジェクトの進捗に悪影響が出てしまう。また、その人のやる気を失わせることになる。	メンバー間で、問題を共有し、解決策を見出す。
メンバーに連絡が、つながらない	大	情報共有ができない。	1日に1回は、携帯かパソコンでメール、skypeを確認する。
タスクが1人に偏る	大	その人が、休むことになった時、開発が進まなくなる。	できるだけ、均等にタスクを割り振る。
報告書の提出が遅れる	大	怒られる、単位が貰えない、留年する	週報をしっかりと書く
メンバー間の仲が悪くなる。	中	プロジェクトの生産性が滞り、進捗に悪影響が出る。最悪の場合プロジェクトが破綻する。	普段からこやかにメンバーと接するように心がける。ミスは誰にでもあることなので激しく非難したりしない。
メンバー間のスケジュールの共有ができていない	中	リリースが遅れてしまうことにつながる。	スケジュールは、できる限り確認し合う。
システムのテストを十分に行わない	中	システムのバグに気づかず、リリースをしてしまう。	テスト期間は十分に設け、テスト項目を用意する。
想定したユーザー（ペルソナ）に合わないインターフェースや仕様になってしまう	中	想定されるユーザーがアプリケーションをうまく扱うことができなくなり、最悪、アプリを利用してもらえなかったり、ユーザーが想定外の操作をしてしまったりする。	作ったアプリを想定ユーザーに使ってもらって、使用中のユーザーの観察や事後アンケートを通して適切なフィードバックをもらい修正する。また、それを繰り返す。
アプリの設計が終わらない	中	永遠と要求定義を繰り返し全くアジャイルな開発を行えない。	普段からどのような問題が存在しているか考える。

図 4.1 リスク分析の結果（一部抜粋）

（ 文責: 熊谷優斗）

#### 4.1.3 アプリ開発のための勉強会

iOS アプリを開発するにあたって必要となる知識を学ぶ勉強会をプロジェクトのティーチングアシスタントが開催したため、これにグループ全員で参加した。勉強会では、Xcode や Swift 言語の使い方を学ぶ Swift 勉強会とバージョン管理システムである、Git と GitHub の使い方を学ぶ GitHub 勉強会の 2 種類が行われた。それぞれで行ったことを具体的に記述する。

Swift 勉強会は全部で 3 回行われた。第 1 回では、メンバーそれぞれの PC に Xcode を導入し、Swift 言語によって UILabel や UIButton を用いた簡単なアプリを作成した。その後、iPad にて作成したアプリをビルドするために、iOS Developer Program への登録を行った。第 2 回では、UIKit という Framework を用いた地図アプリを作成した。第 3 回では、サーバーからデータを読み書きすることのできるアプリを作成した。それぞれの回の終わりには演習問題が出され、これを解くことで学んだ知識の復習を行うことができた。

GitHub 勉強会は全部で 3 回行われた。それぞれの回を通して、バージョン管理システムの理念を学びつつ、Git の基本的な使い方を学んでいった。第 3 回では、Swift 勉強の演習問題を GitHub を用いてメンバー間で分担しながら作成せよ、という課題が出た。しかし、上手くコーディングの役割分担を行うことができず、1 人で全てのコーディングをし、残りのメンバーでコードレビューをするという形を取った。これに対し、ティーチングアシスタントから昨年度はもっと役割分担ができていた、という報告を受けた。今後、上手く役割分担をしていくために教育班では、GitHub の issue 機能を利用していくことを決定した。

（ 文責: 熊谷優斗）

#### 4.1.4 バックログの作成

プロジェクトの方針として、アジャイル開発手法の1つである Scrum という方法論を取り入れることが決まっていたため、プロジェクトのスケジュールをバックログを用いて管理した。バックログとは成果物を作り出すために必要な要素を項目に起こした一覧のことで、この一覧を上下に整頓することで項目の優先順位を表す。バックログには明確なスケジューリングをする必要はなく、優先順位の高いものから順番に行っていく。図 3.2 は 6、7 月分のバックログの原案である。

この原案を企業講師である高森満様と木下実様にお見せしたところ、「バックログの優先度を議論する際にもっと手軽に入れ替えることが可能なように、紙や付箋を用いたほうが良い」というレビューを受けた。そこで、せっかく紙と付箋を使用するならばと、ソフトウェア開発のツールの1つである、「タスクかんばん」のシステムをバックログに取り入れることにした。具体的にはタスクの状態を「TODO」、「DOING」、「DONE」の3つのステージに分割し、更に「TODO」欄のタスクの上下関係によってタスクの優先度を表すようにした。図 3.3 は実際に使用しているバックログである。

	やること	説明	締切日
優先度 高	画面遷移図を作成		2015/06/08
	スタート画面設計		2015/06/08
	会話画面設計		2015/06/08
	マップ画面設計		2015/06/08
	プログラミング画面設計		2015/06/08
	戦闘画面のレイアウトを決める		2015/06/08
	戦闘をどのようなものにするか決める		
	ゲーム作成のための技術調査	Swiftのゲームの技術調査	2015/06/08
	SpriteKitの学習	Swiftのゲーム用フレームワーク	2015/06/19
	SpriteKitの学習ための資料調査		2015/06/19
	SpriteKitの学習用資料の厳選、抜粋		2015/06/19
	Spritekitの実装		2015/06/19
	戦闘画面実装		2015/07/01
	戦闘画面を作る		2015/06/22

図 4.2 6、7 月分のバックログの原案

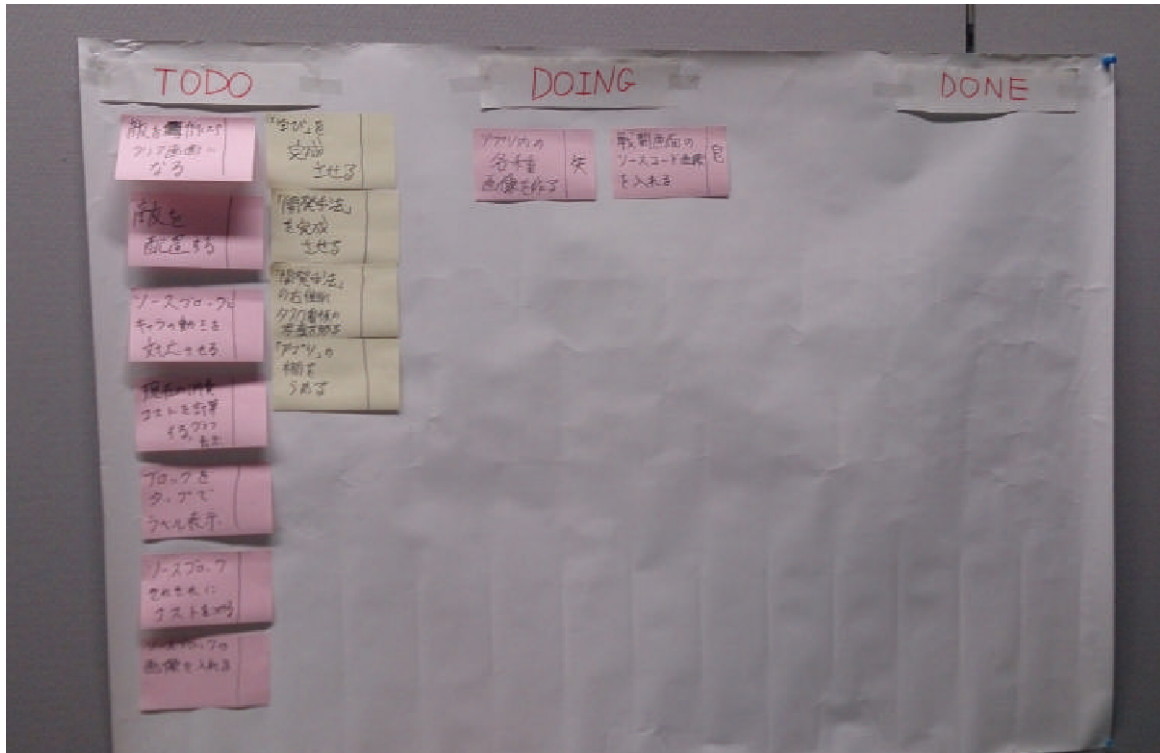


図 4.3 タスクかんばんのシステムを取り入れたバックログ

( 文責: 熊谷優斗 )

#### 4.1.5 中間発表会の資料制作

中間発表会に向けて、ポスターを制作した。制作にあたって、グループメンバーを実装班 3 人とポスター班 2 人に分け、ポスターの制作が終わったら実装班がレビューをする、という形式をとった。しかし、メンバー間の意識共有が上手く行われていなかったため、実装班がポスターのレビューを上手く行うことが出来なかった。そのため、ポスターをティーチングアシスタントや担当教員に見せたところ、目的と制作物がずれている、というレビューを受けた。これを受けて、メンバー 5 人全員で、一度背景、目的、課題の見直しを行い、ポスターの作り直しをした。しかし、短期間で急いでポスターの作り直しを行ったため、今度は文字が多すぎて見づらいというレビューを受けた。これを受けて、ポスター内の文字を少なくするため、もう一度ポスターの構成を見直すという作業を行った。結果、ポスターを作り上げることができたが、その制作に多くの時間を割くことになってしまった。ポスターやその他ドキュメントを作る際には、まずメンバー間の意識共有を行い、こういった構成で文書を書いていくのかを考えることに時間をかけるべきだ、ということを学んだ。

( 文責: 熊谷優斗 )

#### 4.1.6 中間発表会

中間発表会ではメンバーを前半 3 人、後半 2 人に分けて、発表を行った。前半の発表では、アプリのデモを行わないと内容が伝わりづらい、というレビューを受けた。これを受けて後半の発表で

は、デモを取り入れ、内容が伝わりやすいようにした。しかし、後半の発表では合計で 9 人しか見に来た人がいなかった。このことから、教育系が開発しているアプリに魅力が少ないのではないかと、という気づきを得た。

( 文責: 熊谷優斗 )

## 4.2 アプリ案の変化と内容

プロジェクトを進めるうちに、アプリ内容、対象ユーザーが変化していった。その内容を以下に記す。

( 文責: 熊谷優斗 )

### 4.2.1 アプリ案の検討

メンバーそれぞれが考えて来た案を評価し、5 種類の案に絞った。図 3.4 に案を表す。

5 種類それぞれの案をメンバー内で肉付けした後、ティーチングアシスタントと担当教員からレビューを受けた。それぞれの案の詳細とレビューの内容を以下に示す。

案 1 いじめ対策アプリ 従来相談を受けてもらうためには、電話をかけ、言葉で喋らなければならないので、ハードルが高い。一部の教育委員会では、メールの対応も行っている。そこで、少しでもハードルを下げるために、LINE のように教育委員会と会話ができるようにする。  
レビュー内容 このままだとただチャットをするだけのアプリになってしまうのではないかと。既存の SNS アプリと差別化を行うため、独自の機能が必要である。また、どのようにしてこのアプリの評価を行うかという点は要検討である。

案 2 プログラミングを学ぶゲームアプリ 子供がゲーム攻略を楽しみながら、いつのまにかプログラミングを覚えることができるアプリである。最終的なユーザーの到達点としては制御文が使えるようになることである。  
レビュー内容 ゲーム内容は、答えを導くのに手間がかかり、ユーザーに達成感があるものにすべきである。似たようなアプリは既にいくつも存在しているため、それらを調査し、どのように差別化を図るか検討する必要がある。

案 3 発想力を鍛える - 生産消費的なゲームアプリ ユーザーが生産側と消費側の両面で機能することにより、全ユーザーで一緒に発想力を磨いていくアプリである。自分よがりの発想力ではなく、他人にも共感できる発想力を身につけさせることが目的である。  
レビュー内容 ユーザー依存型アプリは投稿が増えないと開発が進まない可能性があるため、どうしたらユーザー同士で活発に活動してもらえるか考えるべきである。

案 4 1 問 1 答共有アプリ ユーザーが作った 1 問 1 答を共有するアプリである。問題を作る楽しさと問題を解く楽しさをシェアすることができる。  
レビュー内容 案 3 と同様に、どうしたらユーザー同士で活発に活動してもらえるか考える必要がある。投稿者が何か得をするシステムにしなければ問題の投稿数は増えていかないだろう。

案 5 外遊び支援アプリ 遊びの教育を行う。IT 化が進み、外で遊ぶことが少なくなっている子供たちに対し、IT を活用することで子供に外で遊んでもらう機会を増やす。その 1 つの



案として、GPS 機能を使って鬼ごっこを行う。

レビュー内容 このアプリを開発するのであれば、楽しく開発を行えるだろう。しかし開発者が楽しくてもユーザーが楽しいとは限らない。ユーザーがどのような遊びを求めているか調査する必要があるだろう。また、アプリを使いながら鬼ごっこをすると、歩きスマホのような状態になり危険なのではないか。

レビューを受け、グループ内で検討した結果、前期では「案2 プログラミングを学ぶゲーム」を作成することに決定した。

提案	内容
案1	いじめ対策アプリ
案2	プログラミングを覚えるゲームアプリ
案3	1問1答共有アプリ
案4	鬼ごっこアプリ
案5	発想力を鍛える-生産消費的なゲームアプリ

図 4.4 絞った 5 種類のアプリ案

( 文責: 熊谷優斗 )

#### 4.2.2 大学生向けプログラミング入門アプリ

前述の「案2 プログラミングを学ぶゲーム」についてより深く考えていった結果、「既存の類似アプリと相違点を持たせるため、函館要素を追加しよう」、「子供は地域性に対してあまり興味を示さない、ならば対象ユーザーを未来大生にしよう」といった理由から、未来大学に入学することが決まった高校生に対する Processing 導入アプリを作成する方針が決まった。アプリの目的は、未来大1年生が「情報表現入門」でプログラミング言語「Processing」を学ぶ際につまづきやすいポイントを、入学前にゲーム形式で気軽に学んでもらうことである。図 3.5 と図 3.6 はアプリ内画面のイメージ図の一部である。

図 3.6、通称「プログラミング画面」では、Scratch のように主人公の行動を表すブロックを組み立ててゆく。どのようなブロックを組めばいいかを考え、それにより主人公を操り、敵を全て倒すことがゲームの目的である。更に、「コスト」という概念を定義し、ブロックそれぞれをコストで重み付ける。1 ターンで使用できるコストの上限は決まっているので、ユーザーはどのようにブロックを組めば同じコストでも最も多く行動できるかを考える必要がある。例えば、ループ文を使うことで、単純に同じブロックを何度も使用するよりも少ないコストで済む。これによりユーザーは自然と良いアルゴリズムを学ぶことができる。

このアプリ案をティーチングアシスタント、担当教員、企業講師の方々に見せたところ、様々なレビューを受けた。一部を抜粋すると次のようなものである。

- このアプリをプレイしたところで、本当にプログラミングの教育になるのだろうか。肝心のプログラミング画面の内容が薄く、未来大生がつまづきやすいポイントを学べるとは思えない。どうすればユーザーへの「教育」になるかを練り直すべき。
- 大学生が使うには、プログラミング画面の内容が低年齢向けである。
- もしアプリを一般向けにリリースすることを目標としているのであれば、未来大学入学向けアプリというのは対象ユーザーが狭すぎる。

このレビューを受けて、もう一度メンバー内で教育要素について考え直し、対象ユーザーを全国の高校生、大学生向けへと変更した。また、プログラミング画面にて、「」「パンチ」といった簡単な記述ではなく、「move(right, 3)」「attack(up)」といった、より本物のソースコードに近い形で表示するようにし、そのソースコードはボタンをタップしていくことで組み立てることができる仕組みにした。

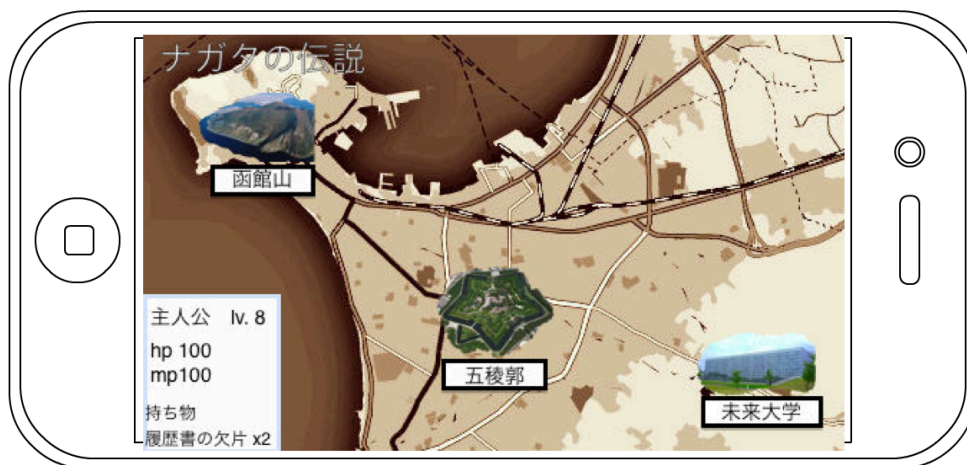


図 4.5 マップ画面

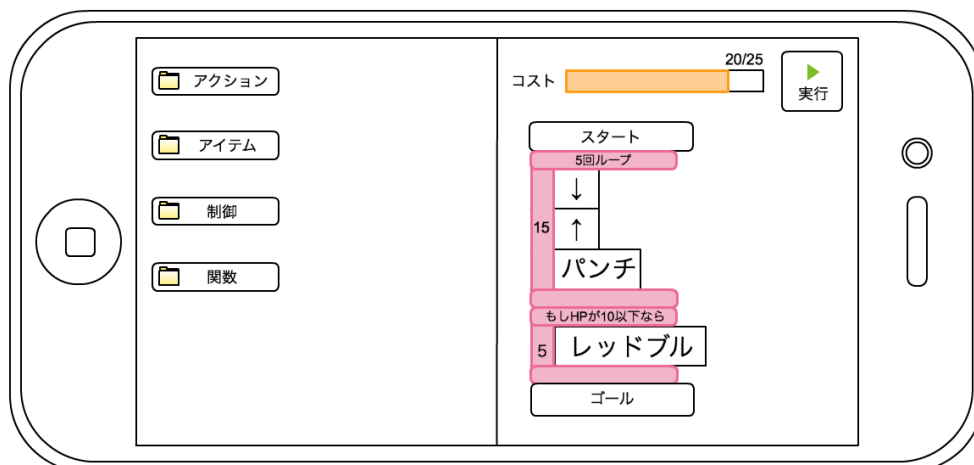


図 4.6 プログラミング画面

( 文責: 熊谷優斗 )



#### 4.2.3 中学生向けプログラミング支援アプリ

プロジェクトを進めるうちに、大学生向けプログラミング入門アプリでは、プロジェクトとしての背景が、客観的に見て共感されないような内容であることに気がついた。そこで、日本の中学校ではプログラミング教育が義務化されている、また、現状のアプリ内容であれば、小学生、中学生が利用しても問題がないといった理由から、対象ユーザーを変更すべきであるという結論にたどり着いた。この日の議論により生まれたのが現状の背景（第1章にて記載）であり、現状のアプリ案（第4章にて記載）である。

（ 文責: 熊谷優斗 ）

## 第 5 章 前期の開発アプリについて

### 5.1 概要

開発するアプリは中学生でプログラミングを習った人、興味を持った人を対象としたソースコードの組み方を学ぶゲームアプリである。図 4.1 のように、このゲームには自機と敵機があり、ユーザーはマス目上のステージにある自機をソースコードを組むことによって動かし、敵機を倒すことでゲームがクリアとなる。

- 各ステージのクリアまでの流れ
1. 自機を敵機の前まで移動して倒すソースコードを考える。
  2. ソースコードを入力する。
  3. ソースコードを実行する。
  4. ソースコードの通りに自機が動く。
  5. 敵を全て倒すとステージクリアとなる。
  6. 使用したコストに合わせてランクとコメントが表示される。

敵機まで移動して倒すまでをいかに短いソースコードで完了させるかをめざすゲームである。

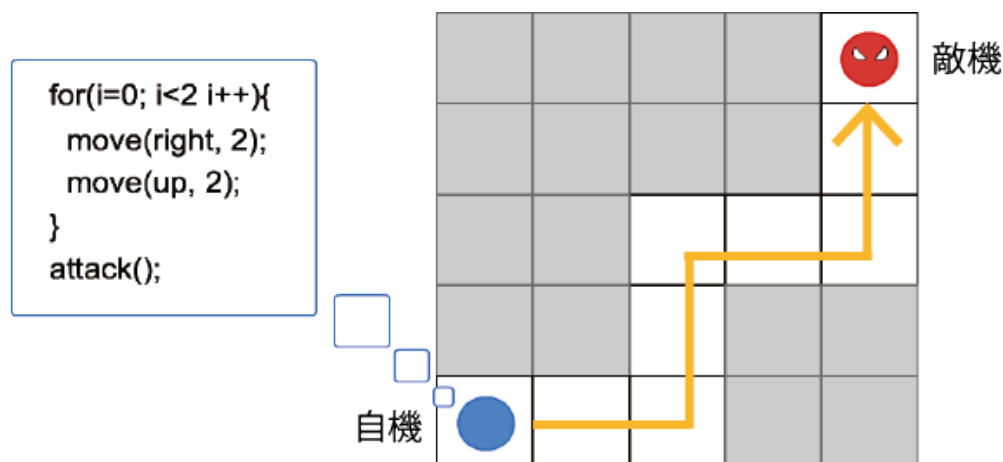


図 5.1 ゲーム概要

( 文責: 新保遥平 )

### 5.2 プログラミング画面

プログラミング画面ではユーザーが敵機を倒すためのソースコードを入力する。例えば for 文を入力したいときは図 4.2 のように画面に配置されたソースボタンをタップする。



図 5.2 ソースコードの入力

ユーザーは図 4.3 のように画面右側に配置されたそれぞれのソースボタンをタップして、ソースコードを組み立てていく。現状、実装したソースボタンは attack()、move()、left、right、0~9、; などである。タップされたソースボタンは順に、右側のスペースに記述される。

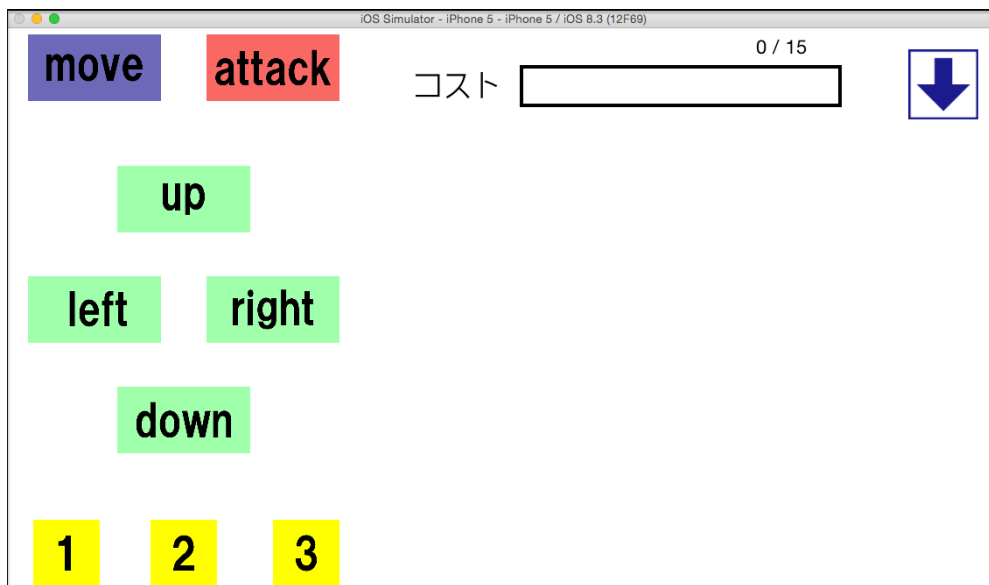


図 5.3 プログラミング画面

例えば下記のようなプログラムを組むこととする。

```
move(right,3);  
move(up,3);
```

このソースコードを図 4.4 のプログラミング画面に入力した。

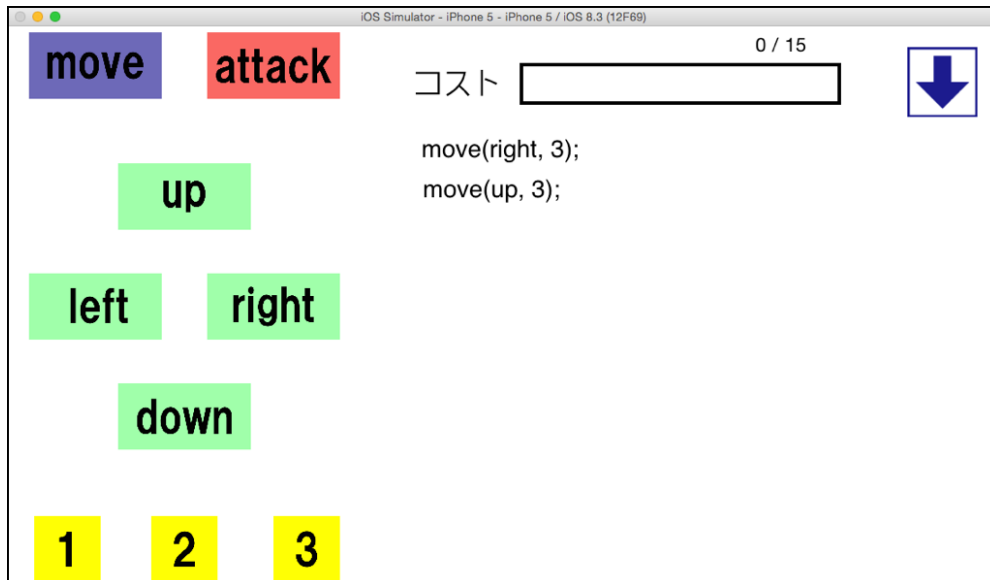


図 5.4 ソースコード入力後のプログラミング画面

このようにプログラムをタップで組むことが出来る。また、図 4.5 のように、次に引数である数字を入れるべきところに「up」をタップしてしまうなど、間違ったタイミングでソースボタンを押すと画面上にエラーが出て、すぐに確認ができる。

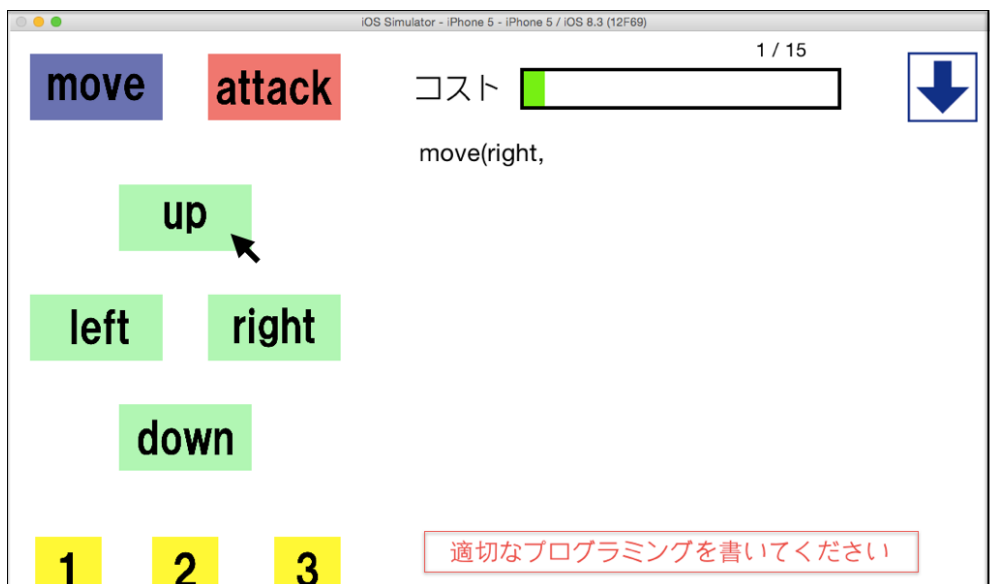


図 5.5 ソースコードがエラー時のプログラミング画面

ユーザーはソースコードを入力後、プログラミング画面の左上の「下矢印」ボタンで戦闘画面に移動する。

( 文責: 新保遥平 )

## 5.3 戦闘画面

図 4.6 の戦闘画面はプログラミング画面で入力したソースコードで自機を動かすための画面である。戦闘画面の左下にある三角の実行ボタンを押すことで、自機を動かすことが出来る。また戦闘画面、左下の「P」と書かれたボタンでプログラミング画面に戻ることができる。現状では、あらかじめ設定されたプログラムでしか自機を動かすことが出来ない。

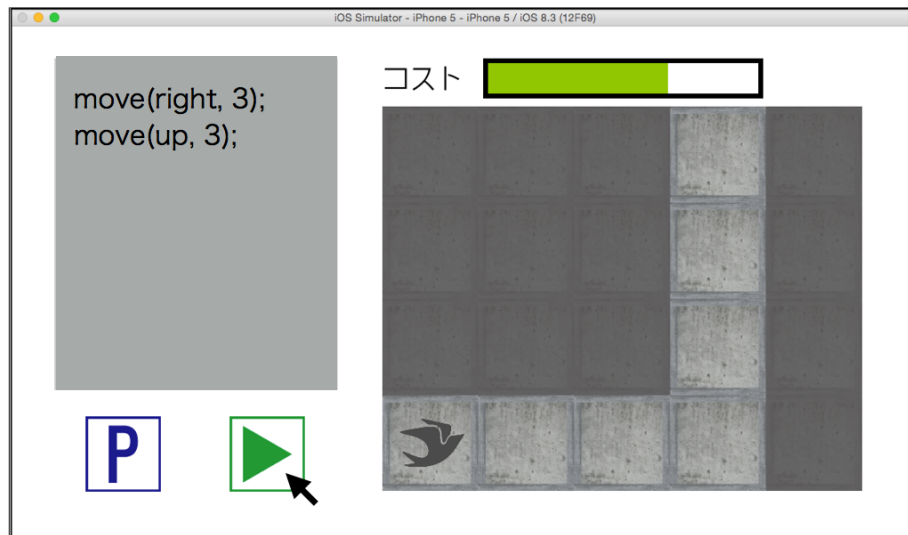


図 5.6 戦闘画面

( 文責: 新保遥平 )

### 5.3.1 ゲーム性

ただプログラミングを学ぶのではなく、ゲームを通してプログラミングを学ぶことでユーザーのモチベーションを保ちつつ、アプリを使ってもらえると考えた。また実際にソースコードを組むことで自機を思い通りに動かすことが出来たときにプログラミングの学習が深まると考えた。

( 文責: 新保遥平 )

### 5.3.2 教育性

このアプリではユーザーがより簡潔なソースコードを組み立てられるようになるために、コストとランクがある。ソースのボタンそれぞれにコストが設けられており、問題をクリアした際にコストの使用量が少ないほど簡潔にソースコードを組み立てることが出来たと判定し、図 4.7 のようにランクを与える。ランクが低かった場合、より良いランクにつながるヒントを与える。そして高いランクが与えられたときに、ユーザーを褒める言葉を表示する。このサイクルが次の問題への意欲につながり、より簡潔なソースコードを組み立てることが出来るようになる。この流れをユーザーストーリーとしたものを図 4.8 に示す。

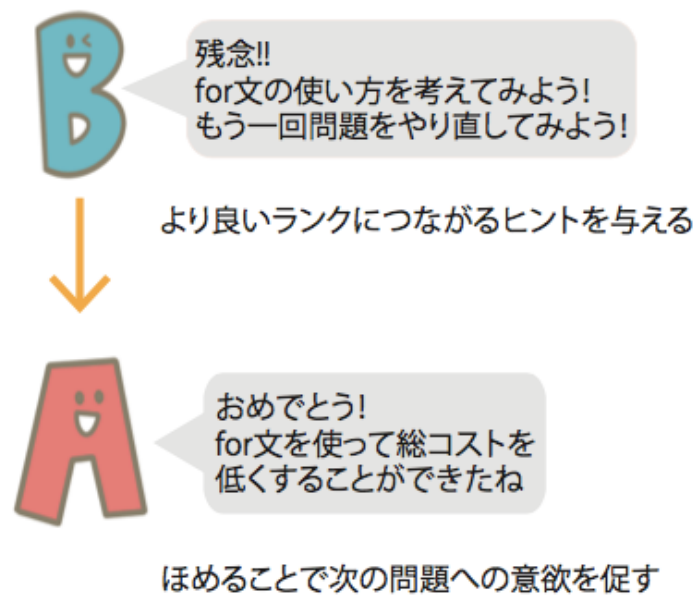


図 5.7 ランクとヒント

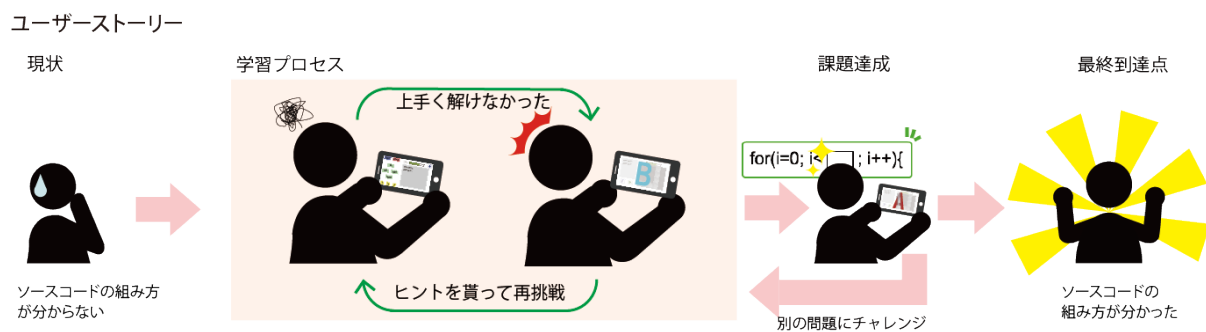


図 5.8 ユーザーストーリー

( 文責: 新保遥平 )

## 第 6 章 前期の結果

### 6.1 プロジェクトの評価

本プロジェクトは、多くの方からレビューを受け今の背景、課題、目的になっている。しかし、私たちが考えたアプリはその課題を解決できるアプリとなっていない。そのため、アプリの設計を見直す必要がある。

また、7月に行われた中間発表会の評価シートの結果からは、「声がはっきり聞こえた」、「声は大きく聞きやすかった」、「相手の顔を見て話してくれたので、聞き取りやすかった」などの意見をいただき、発表技術に関しては高い評価を得られた。しかし、発表内容に関しては「最終的なゴールは?」、「内容がわからないので評価不能」、「既存のものと比較がない」などの意見をいただいた。いただいた意見をまとめると、本プロジェクトは目標が決まっておらず、内容がわかりづらいという評価であった。

これらのことから、本プロジェクトは第三者の方に伝えられる内容になっていないので背景、課題、目的につながるアプリ案が必要である。そして、第三者の方に伝えられる内容にしていかなければならない。

( 文責: 中進吾 )

### 6.2 プロジェクトの成果

前期の活動の成果は以下の 3 点である。

- スクラッチワークショップに参加したことにより、プログラミング初心者の方にプログラミングを教える場合、C 言語や Java から始めるのではなく、Scratch のようなビジュアルプログラミング言語から始めた方が良いということがわかった。また、プログラミングで音声機器などの機械を動かしてもらうことにより、プログラミングに興味を持ってもらうことができるということがわかった。
- 本プロジェクトでは、Swift 言語を用いてアプリ実装を行うことになっていた。しかし、メンバー全員 Swift 言語は使ったことがなかったため、実装に不安があった。アプリを実装できる期間は短かったが、キャラクターを動かしたり、ボタンをタップすることでソースコードを打ち込めることができるまでアプリを開発することができた。これによって、今後のアプリの実装に対する不安がほとんどなくなった。また、アプリの実装を行えたことにより、実装に携わったメンバーは Swift 言語に自信を持つことができた。
- 中間発表会で展示するポスターは、Adobe Illustrator という描画ツールソフトを使用して作成することになっていた。このソフトを使用したことがあるのはメンバーの 5 人中 2 人だけだったため、ポスターを作成できるのは 2 人だけだった。しかし、3.1.5 で述べたように目的と制作物がずれていたため、メンバー 5 人全員で、一度 背景、目的、課題の見直しを行い、ポスターを作り直すことになった。その結果、今まで Adobe Illustrator を使ったことがない人も使えるようになり、メンバー全員がポスターを作成できるようになった。11

月に開催されるアカデミックリンクや最終成果発表でもポスターを使用するので、メンバー全員がポスターを作成できるようになったことは、非常に大きな成果である。

( 文責: 中進吾 )



## 第 7 章 後期の背景

### 7.1 プログラミング基礎の現状

公立はこだて未来大学には、学部 1 年生が後期に必修科目として履修する「プログラミング基礎」という講義がある。この講義は、1 年生が前期に履修した「情報表現入門」で学んだことをもとに、C 言語について学び、プログラミング（概念、考え方）への理解を深めることを目的としている。講義内容はプログラミング基礎概念である、変数、配列、条件分岐、繰り返し、関数、文字列、構造体などについて学ぶ内容となっている。また、講義だけでなく、演習として課題（プログラム）に取り組むことになっており、毎回の講義で課題を提出することになっている。成績の評価方法は、期末試験の結果を重視し、各回の課題の提出状況で評価されることになっている。しかし、例年「プログラミング基礎」の落第者は多くいる。「プログラミング基礎」は必修科目であるため、卒業するためには必ず単位を取らなければならない講義である。

（ 文責: 中進吾 ）

### 7.2 現状と課題

私たちは、例年「プログラミング基礎」でなぜ多くの学生が落第するのか知るために、現在履修している学生とその講義を担当している TA (Teaching Assistant) にヒアリング調査を行った。

履修している学生に講義の難しい点を聞いてみると、配列などの基本的な構文が分からないということが分かった。実際に、講義で出題された課題に取り組んでいる所を見せてもらうと、講義で覚えたことをそのまま書いており、そのコードがどのように動いているのか分かっていないことが分かった。一方、TA に講義のどのような点が分からないのか聞いてみると、課題で出題される問題に対するアルゴリズムが分かっていないことが分かった。

実際に講義で使われている資料を見ると、ほとんどが文字で構成されており、専門用語が多くあった。講義を履修する学生は、プログラミングを学んで日が浅い人が多いのでより分かりやすい表現が必要であると感じた。

（ 文責: 中進吾 ）

## 第 8 章 後期のプロジェクトの目標

### 8.1 開発アプリの目標

本学の 1 年生が履修する「プログラミング基礎」は、C 言語を学ぶ必修の講義でありながら、例年多くの 1 年生が落第しているという課題がある。また、1 年生に対してのヒアリング調査から配列などの基本的な書き方やアルゴリズムがわかっていないことが分かった。

そのため、後期の開発アプリでは、C 言語やアルゴリズムを理解できていない学生に対して、講義で教える学習单元ごとに概念の説明、例題、確認問題をアニメーションで教える Web アプリケーションを作成した。このアプリケーションを実際の授業で使い、1 年生に予習、復習用の教材として利用してもらう。そこで、アニメーションを用いることで、ユーザーが学習单元の概念を理解し、実際のソースコードのアルゴリズムの動きを把握できるようになることが後期の開発アプリの目標である。

後期のプロジェクトの活動では、概念の説明、例題、確認問題のアニメーションを完成させ、Web アプリケーションとしての体裁を整え、本大学の HOPE に掲載することを目標に活動していく。

( 文責: 梶勢也 )

### 8.2 プロジェクト学習としての目標

前期の提案内容では、目的と提案が噛み合っていなかったため、教育性が欠けている一貫性のない提案になってしまった。また要件定義を深く行っていなく、設計を先に進めたため、目的を見失ってしまった。さらに、提案内容に関する事前調査が不十分であったため、提案内容に酷似した iOS アプリが存在していた。そのため、プロジェクトの活動を要件定義・企画の段階へと戻すことになった。

後期の活動ではこれらの失敗を学び、前期のように後戻りが頻繁に起きないようにする。また、前期で行ってきた活動の経験から、グループメンバーの得意分野を活かし、メンバーの適切な役割分担を行う。さらに、本グループでは ICT 演習に参加しているメンバーと情報デザインコースに所属しているメンバーがいるため、前期では要件定義プロセスの進め方の考えに相違があり、話し合いに時間をかけすぎてしまい、作業の遅れが生じた。後期では、双方のプロセスを学び、適切にプロジェクトを進められるようにすることが目標である。

( 文責: 梶勢也 )

## 第 9 章 後期の活動

### 9.1 イベント

#### 9.1.1 後期キックオフ会

#### 9.1.2 アカデミックリンク

2015 年 11 月 14 日に「HAKODATE アカデミックリンク 2015」という函館市内などの教育機関による合同研究発表会が行われた。このアカデミックリンクには未来大の多くのプロジェクトが参加しており、我々のプロジェクトも参加した。教育系グループは C-mation のデモとポスターをお客さんに見せながらポスターセッションを行った。このアカデミックリンクには様々な年齢層の方がいらっしゃり、我々のアプリについて多くの意見をいただくことができた。高校生からは「こんなアプリあったらやってみたい!」プログラミングの経験者からは「このアプリは誰かに実際に使って評価を得ることができたの?」「本当に C 言語でつまづく部分は配列なの?」などの厳しい意見を得ることができた。このアカデミックリンクではアプリの評価だけでなく、自分たちの発表する力を試すこともできたと思われる。4 月にプロジェクト学習が始まって以来、何度も発表の機会があったため、メンバーの発表技術が上がり、アカデミックリンクの発表に関しては成功した。

( 文責: 新保遥平 )

#### 9.1.3 最終成果発表会

### 9.2 アプリ案の推移

#### 9.2.1 夏休みの課題と調査報告

#### 9.2.2 Processing デバッガツール

#### 9.2.3 C-mation

## 第 10 章 後期の開発アプリについて

### 10.1 概要

最終成果発表に向けて開発したものは「C-mation」という C 言語のプログラミング学習支援ツールである。対象ユーザーは未来大 1 年生である。ユーザーにはこの Web アプリを用いて C 言語のプログラミングを学んでもらおうと考えた。このアプリではアニメーションを用いて、概念の説明、例題の解説、確認問題の出題を行う。このように段階を踏んで教えることで、C 言語を理解してもらうことが目的である。今回は特に C 言語の配列の部分の開発を行った。

( 文責: 新保遥平 )

### 10.2 コンテンツ

#### 10.2.1 概念の説明

概念説明では、まず身の回りにあるものを例に配列の説明をアニメーションを用いて行った。この概念説明の部分は Google スライドを用いて作成した。配列を説明するにあたり図 9.1 のように、変数とはなんなのか、変数を家 (一戸建て) に例えて説明を行った。次に変数と配列がどこが違うのかを配列をアパートを例にして説明を行った。最後に配列を使うことによって変数より、どのような時に便利なのかを説明した。また、実際に配列の宣言の方法も説明している。

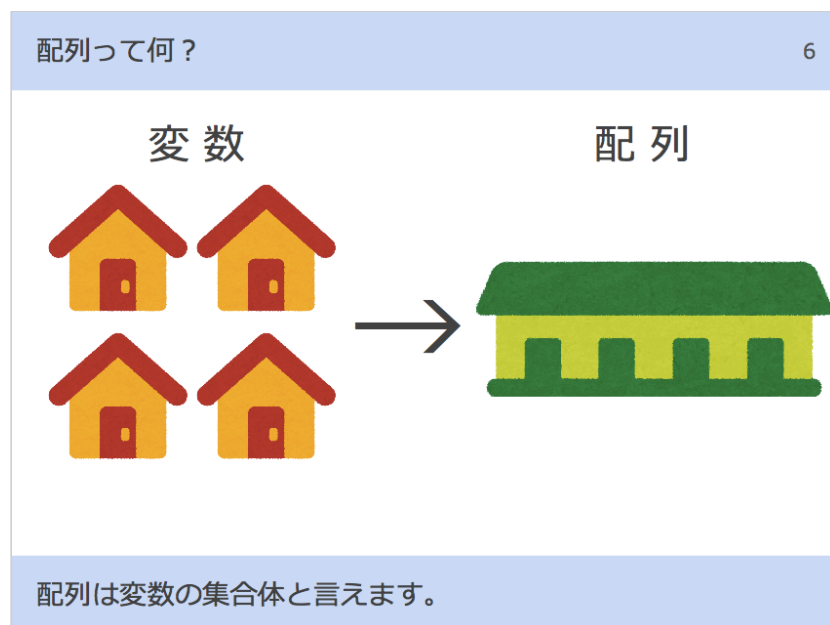


図 10.1 概念スライド

( 文責: 新保遥平 )

### 10.2.2 例題

例題の部分ではまず、図 9.2 のように例題のソースコードをユーザーに見せて、このソースコードがどのように動くのかを考えさせる。これによって、次の画面でユーザーが頭の中で考えていたソースコードの動きが合っているかを確認することができる。また、この例題の部分も Google スライドを用いて作成した。

例題その1

3

```
int apart[] = {2,4,3,4}; ...①  
printf("apart[2] = %d", apart[2] ); ...②
```

例題の 1 つ目です。  
このプログラムはどのように動くか考えてみてください。

図 10.2 例題スライド 1

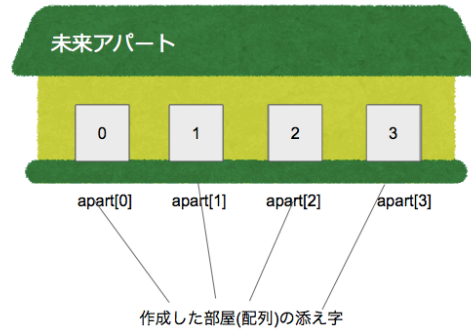
次に図 9.3、図 9.4 のようにアパートとソースコードを組み合わせで説明を行う。具体的には、1 行目のソースコードでは `apart[0]` には住人が 2 人、`apart[1]` には住人が 4 人、`apart[2]` には住人が 3 人、`apart[3]` には住人が 4 人が未来アパートに住んでいることをアニメーションで説明している。次に 2 行目のソースコードでは `apart[2]` の表示を行うことを示している。また、ソースコードが示している部分がアパートのどのような状態かを合わせて説明している。これによって、ユーザーがソースコードの動きを見ながら配列を理解することができると思った。

例題その1

5

```
int apart[] = {2,4,3,4}; ...①
printf(" apart[2]= %d", apart[2] );
...②
```

apart[0]には2人  
apart[1]には4人  
apart[2]には3人  
apart[3]には4人の住人が住んでいます



①配列を作成し、値を代入します

図 10.3 例題スライド 2-1

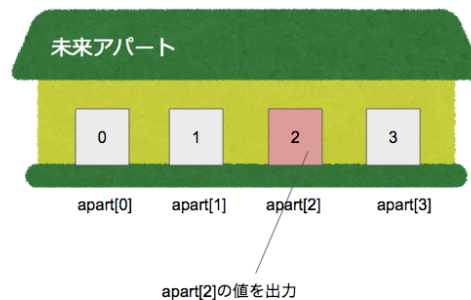
例題その1

6

```
int apart[] = {2,4,3,4}; ...①
printf(" apart[2]= %d", apart[2] );
...②
```

出力結果

apart[2]=3



apart[2]の値を出力します。これで例題1を終わります

図 10.4 例題スライド 2-2

( 文責: 新保遥平 )

### 10.2.3 確認問題

確認問題ではインタラクティブな学習を行ってもらうために、図 9.5、図 9.6 のように、実際にユーザーに値を入力をして問題を解いてもらうことを目的とした。また、この確認問題の工夫した点は配列の値をランダムにし、ユーザが飽きないように何回も学習することができ、入力した値によってアニメーションが変化することである。

array\_master

**確認問題**  
下の配列に最大値最小値アルゴリズムを適応した時、  
最初にmax=8、min=1になるときの変数 i の値を求めよ。

1 4 5 2 3 8 6 7

i =

決定

図 10.5 確認問題 入力画面 1-1

array\_master

**確認問題**  
下の配列に最大値最小値アルゴリズムを適応した時、  
最初にmax=8、min=2になるときの変数 i の値を求めよ。

6 8 4 3 2 7 5 1

i = 3

決定

図 10.6 確認問題 入力画面 1-2

ユーザーには図 9.7、図 9.8 のように確認問題のソースコードを見ながら確認問題を解いてもらう。この確認問題は未来大の 1 年生が「プログラミング基礎」の授業で扱う講義スライドを参考に我々が考えた問題である。問題の内容は、与えられた配列に入った 8 つの値の指定された大小の値を求めるときの繰り返す回数を考えさせるものである。流れとしては、ユーザーが答えだと思う値をキーボードから入力する。

```
////////// ここから最大値最小値アルゴリズム //////////

/*
 * 配列内の数字を順に探索していき、最大値、最小値を発見するアルゴリズム
 * 配列の i 番目の要素が現在の max より大きい場合、max に i 番目の要素を代入する
 * これを繰り返すことで、for 文を抜けた時、
 * max には配列の中の最大値が代入されていることになる
 */

for(int i; = 1; i < 8; i++){
    if(max < numbers[i]){
        max = numbers[i];
    }
    if(min > numbers[i]){
        min = numbers[i];
    }
}

////////// ここまで最大値最小値アルゴリズム //////////

// 最大値、最小値を出力
printf("配列の最大値は %d\n",max);
printf("配列の最小値は %d\n",min);

}
```

図 10.7 確認問題 ソースコード 1-1



確認問題ソースコード

```
#include <stdio.h>

int main(){
    int numbers[8]; //配列を宣言

    /*
     * 配列の i 番目の要素に入力した数字を代入する
     * for 文が 2 ループ目の際は number[2]に入力された値を代入
     */
    for(int i; = 0; i < 8; i++){
        scanf("%d", numbers[i])
    }

    /*
     * 入力された値の中から最大値、最小値を発見する
     * 最終的に、max には最大値、min には最小値を代入されているようにする
     */
    int max, min;

    //max と min に初期値を代入
    max = numbers[0];
    min = numbers[0];
```

図 10.8 確認問題 ソースコード 1-2

すると、図 9.9、図 9.10 のように入力した値が正しいか、間違いかをすぐに表示する。そして入力した値が間違っている場合、その後プログラムがどうなるかを説明する。これは、ユーザーがなぜ「この値が間違っているのか」を理解してもらうためである。このようにして、確認問題でユーザーが本当に配列を理解したのかを確かめている。



図 10.9 確認問題 正解画面



図 10.10 確認問題 不正解画面

( 文責: 新保遥平 )

## 第 11 章 後期の結果

### 11.1 プロジェクトの評価

12 月に行われた最終発表の評価シートの結果から「声が大きく、ポスターが見やすかった」、「円滑に話を進めており、聞き取り易い」、「具体的な説明をしながらデモができてた」などの意見をいただき、前期の中間発表と同様に発表技術は高い評価を得られた。一方、発表内容に関しては「アプリ内容が分かり易く、教育アプリは使えると思いました」、「イメージがわくので、分かり易かったです」、「前期と後期のつながりが分かり易く説明されていた」などの意見をいただき、後期と比べて高い評価をいただいた。また、「実際に 1 年生に使ってほしかった」、「教育アプリが他の班と比べて劣って見えた」、「他の班と比べて iOS アプリを作らなかったメリットを知りたい」などの意見をいただき、自分たちの活動の不十分な点を知ることができた。

これらのことから、後期のプロジェクトは前期と比べて第三者の方に伝えられる内容であったと思われる。また、改善する必要がある部分を知ることができたため、それらを改善していきたいと思う。

( 文責: 中進吾 )

### 11.2 プロジェクトの成果

後期の活動の成果は以下の 2 点である。

- 前期は、活動時間を無駄にすることが多く何度も居残りをしていた。後期はそれを防ぐために、プロジェクトが始まる前にグループリーダーが LINE でメンバーに今日のアジェンダを伝えた。その結果、メンバー全員がその内容を意識して話し合いを行ったことで、活動の時間を無駄にすることがほとんどなくなった。このことから、活動を始める前に話す内容を伝えることは重要なことだということがわかった。
- 本プロジェクトは中間発表、アカデミックリンク、最終発表の 3 つの発表会に参加し、いずれもポスターセッションで行った。3 度の発表会を経験した結果、最初はメンバーの数人しか発表することができなかったが、最終発表会ではメンバー全員が人前で発表することができるようになった。今後、研究成果発表など発表の機会は多く存在するため、メンバー全員が人前で堂々と発表できるようになったことは、大きな成果だと思われる。

( 文責: 中進吾 )

## 第 12 章 今後の課題と展望

### 12.1 開発アプリの課題と展望

最終発表会を終えてから、TA から本グループの開発アプリに酷似したソフトウェアが本学の 2 年次の科目である「アルゴリズムとデータ構造」の教科書の付属 CD にあるという意見を頂いた。その中身は、プログラムの実行に伴って、刻々と変化するプログラムの流れや変数などを、C 言語で書かれたプログラムリストと対比しながら、視覚的に体験学習できる「アルゴリズム体験学習ソフトウェア」である。その中に三値の最大値を求めるプログラムがあり、アニメーションの速度を調整できたり、変数の初期値を変更することや 1 コマずつ動かすことも可能になっている。本グループの開発アプリにはない機能や似たような機能が入っていることを踏まえ、これからグループで検討すべき課題である。

「アルゴリズム体験学習ソフトウェア」には、ユーザの入力によってアニメーションが変化するなどの相互作用がないため、本開発アプリでは今ある機能だけではなく、教育性を高めるためにユーザとアプリの相互作用が高い機能をつける。また、現状の開発アプリは、配列のアニメーションしか作成してなく、コンテンツとしては不十分である。そのため、配列に加えて、ポインタ、構造体などの 1 年生が難しいと感じる単元を調査し、アニメーションを加え、Web アプリケーションとしての体裁を整えていく。その後、本学の 1 年生に開発アプリを使って評価を行い、アプリを改善していく。

また、最終発表会で「メタ学習センターと連携してみては？」という意見を頂いた。本プロジェクトの最終目標である未来大学の授業の予習、復習用の教材として使ってもらうだけではなく、メタ学習センターと連携しプロジェクト学習、メタ学習センター、授業が連携して学生に教育できるようにすることが今後の展望である。

( 文責: 梶勢也 )

### 12.2 メンバーの課題と展望

前期の活動では、中間発表会に向けて、プロトタイプを制作する実装班と、ポスターを制作するポスター班で分かれたが、いくつか課題があった。アプリやポスターに使われる画像の制作は情報デザインコースのメンバーがほとんど一人で作っていたため、実装がスムーズにできなかった。なるべくタスクをメンバーで分散させるべきであった。また、ポスター班は実装に関わっていなかったため、実装班とのスキルの差があった。

また、アプリ案を決める際に本プロジェクトメンバー、TA、担当教員に毎回アプリ案の発表を行っていた。そこで頂くレビューに対し、グループメンバー誰一人しっかりと説明や対案をすぐに示すことができなかった。そのためレビュー内容の修正をするのではなく、要件定義、設計をやり直すことを何度も行っていた。

さらに、プロジェクト学習の授業外での作業が多く、メンバー全員が作業できない日があった。そのため、情報共有に時間がかかったり、効率良く作業をすることができなかった。

後期の活動では、最終発表会に向けて概念の説明、例題、確認問題を制作する実装班とポスター

のデザインを考え、制作するポスター班、Web サイトのレイアウトやロゴを決める Web 班に分かれて活動を行った。確認問題を制作するにあたって、残っている実装期間を考え 1 年生の頃に使用し慣れている Processing を用いて、実装を行った。その結果、新しい言語である Swift 言語を習得することができなかった。また、ほとんど一人のメンバーが Processing で実装を行っていたため、コード規約がなく他のメンバーが理解するのに時間がかかるコードになってしまった。

また、プロジェクト時間外に進捗報告をする機会がなかったため、メンバーが何の作業をしているか把握できずにメンバーのタスクの量に偏りが発生した。さらに、グループでの話し合いに時間をかけすぎてしまうことや、論点にずれが生じメンバーの共通認識に相違が発生してしまうことが多くあった。さらに、Web 班をメンバー 1 人に任せていたことやオンラインクラウドストレージに保存し、グループメンバーで進捗確認を行っていなかったため、Web サイトのレイアウトを構築を行っていたメンバーが作業データを紛失してしまい、スケジュールが大幅に遅れてしまったことがあった。

今後の展望は、メンバーの役割を適切に決めることや進捗確認を毎日行うなどし、お互いの進捗を確認し、タスクの量に偏りがいないかを適宜確認する。また、活動を行う前には活動計画を決めることや話し合いの際には、時間を決めファシリテーターを設定し、論点がずれていないか客観的に見るようにしていく。また、グループメンバー全員が開発アプリに対して共通認識を持ち、適切な説明ができるようになることである。

( 文責: 梶勢也 )

## 第 13 章 学び

### 13.1 グループとしての学び

本グループでは、前期と後期の活動が異なり失敗を多く経験してきた。以下の 4 点が失敗から特に学んだことである。

- 想定外のタスクが発生した際に、先の見通しを立てることができず、作業時間を無駄にした。このことから作業の変更や遅れが発生したとき、随時スケジュールの更新を行うことで効率よく作業を行うことの重要性を学んだ。
- 話し合いに時間をかけすぎてしまい議題が少しずつ脱線し、議論に集中できなくなって議論についていけなくなったメンバーがいたり、メンバー全員で最終的な議論の共有ができていなかった。そのため、作業の遅れが発生した。このことから活動する前に活動計画を決めることや話し合いの時には時間を決め、議論が正しい方向で進んでいるか客観的に見ることの重要性を学んだ。
- 前期ではアプリ案の要件定義を固めずに実装を行ったため、目的と提案が噛み合っていない、一貫性がない内容が分かりにくい提案となってしまったことや教育系グループとして教育という意味をしっかりと決めていなかったため、何度も提案内容の変更があった。このことから課題調査を深く行い、設計をしっかり行うことの重要性を学んだ。
- プロジェクト学習の授業外での作業が多く、議事録を残していないことがあり、情報共有がうまくできていなかった。さらに進捗確認をきちんと行っていないため、メンバーが何をしているのか把握できずにメンバーのタスクの量に偏りが生じた。このことからメンバーの役割を適切に決めることやタスク看板を用いることでお互いの進捗を確認し、情報共有することの重要性を学んだ。

前期での失敗を後期で振り返ることで、後期の活動をスムーズに行うことができた。開発アプリの目標は達成することができなかったが、多くの経験や学びを得ることができたため、プロジェクト学習としての目標は達成できた。

( 文責: 梶勢也 )

### 13.2 各メンバーの学び

#### 13.2.1 熊谷優斗

これまで大学での活動の中で何度か PBL は経験したものの、自ら課題を考え主体的に活動したことは初めての経験だったため、様々な学びを得ることができた。その中でも、今回は 2 つの学びをとりあげる。1 つ目の学びは、メンバーに技術を教えることの難しさだ。過去の PBL 経験から、Git に関する知識は持っていたため、メンバーに Git と GitHub の使い方を教える役目を受け持った。しかし、メンバーによって理解度の差があるにも関わらず、理解できているメンバーに合わせて話を進めていってしまった。そのため、メンバー全員に Git の使い方を理解してもらうこと

ができなかった。このことから、メンバーの理解状況を確認し、全員が理解した上で次のステップに進む、という教え方が大切であることを学んだ。2 つ目の学びは、適切な文言を使用することの重要性だ。前期の活動では、「基本的な制御文の書き方を学ぶ」という意味で、「簡単なアルゴリズムを学ぶ」という文言を使用していた。その結果、TA や教員によるレビューの際に、自分たちが作成したいアプリケーションの内容をうまく伝えることができず、苦労した。このことから、適切な文言を使用することは、他人に自分の意見を伝えるために重要であることを学んだ。

( 文責: 熊谷優斗 )

### 13.2.2 梶勢也

前期は本プロジェクトを進めるにあたって、Swift 言語を使ってプログラミングを行うため Xcode が入っている MacOS が必要であった。自分は Windows のノートパソコンしか持っていなかったため、enPiT から MacBook Air を借りて作業を行った。初めは、Windows と異なり上手く作業を進められなかったが、使っているうちに MacOS に慣れて上手く作業できるようになった。

また、グループで開発を行っていくうちに、他者にもわかりやすいソースコードを書くように意識した。さらに、Git と GitHub を使ってバージョン管理することや議事録を残すことの重要性を学んだ。

本グループのメンバーには ICT 演習に参加しているメンバーと情報デザインコースに所属しているメンバーがいたため、話し合いの進め方や要件定義プロセスなどの様々な技術と知識を吸収することができた。そのため、他にグループ活動をするときは自分がリーダーとなって活動を進めていくことができるようになった。

後期では、Processing を用いて実装を行ったことに加え、スライドを作って企業講師にグループの活動を発表したり、ポスター制作を行った。プロジェクトが始まった当初は、発表経験があるメンバーに発表を任せていたため、発表の技術がうまくならなかった。後期から人前で発表することや、話し合いに積極的に参加することで人に伝える技術が身についた。またポスター制作などでメンバーや TA からレビューを頂き、修正をスムーズに行うことができた。この経験からレビューの重要性を学び、積極的にレビュー行うようになった。

( 文責: 梶勢也 )

### 13.2.3 新保遥平

この 1 年間では多くのことを学んだ。技術的な点では主に 3 つの学びを得た。1 つめに GitHub をしっかりと理解して使えるようになったことだ。GitHub は以前に使ったことがあったが、曖昧な部分が多くわからない部分も多くあった。だが、プロジェクトが始まってからはグループ内でも GitHub を上手く使ってアプリの開発や報告書の作成を行うことができた。2 つ目に、ポスターの作成のために Adobe Illustrator の使い方を学ぶことができた。他プロジェクトではポスターの作成はデザインコースの人が作るプロジェクトが多い中、自分もポスター作成に携わることができ、いい経験になった。3 つ目は発表技術である。私は即席で人の前で話すことがとても苦手であったが、この一年、何度も即席で話す機会が多くあり、自分の意見をまとめて話すことができるようになった。

前期は自分一人で行うべきでないタスクも一人で行ってしまったため、自分のタスクの進捗が遅

れることがあった。このことから、プロジェクトリーダーとして、人に仕事を振り分けることの重要性を学んだ。

また後期は、プロジェクトが始まる前に今日の活動のアジェンダ、終わる際に、今日の進捗報告を行った。これは各グループの活動が不透明だったからである。実際に開発しているアプリの進捗を聞くことによって他のグループからの相互レビューを受け、アプリ案の修正を行うことができた。このことから、プロジェクトリーダーとして情報共有の重要性を学んだ。

( 文責: 新保遥平 )

#### 13.2.4 中進吾

私がこのプロジェクト学習で学んだことは以下の4つである。

1つ目は、メンバーに仕事を均等に振り分けることができていなかったため、進捗が遅れることがあった。このことから、メンバー1人を頼りすぎるとスケジュールが遅れてしまうことが分かった。

2つ目は、私はこれまでに2度PBLに参加してきたが、要件定義は先輩に任せて自分は実装ばかりを行っていた。今回、一から要件定義を考えたことは大きな学びであり、自分に足りないものを見つけることができた。今後、PBLに参加するときには自分から率先して要件定義を立て、この経験を後輩に伝えていきたいと思う。

3つ目は、後期は、今日のアジェンダをメンバー全員に伝え、時間の管理を行った結果、前期と比べプロジェクトをうまく進めることができた。アジェンダを流すこととタイムマネジメントをすることの重要性を知ることができた。

4つ目は、毎回のプロジェクト学習の終了前、進捗報告を行っていた。そのため、チーム全体が今日何をやってどこまで進んだのか、メンバー全員で共有することができた。このことから、進捗報告の重要性を学ぶことができた。

( 文責: 中進吾 )

#### 13.2.5 矢吹溪悟

デザインプロセスの大切さを再認識し、同時に自分の未熟さを痛感した。なぜなら、初期テーマの設定からブレインストーミングやフィールドの調査を怠ってしまったからだ。本来ならば、メンバーが一丸となって、話し合いや現状を調べながら、みんなでテーマを確立するべきだ。しかしながら、今回は個人個人がやりたいものを考え、そこから一つに絞ってしまったのだ。そのため、背景情報や対象ユーザーの設定が偏見や想像論になったり、後付けとなり、プロジェクトの根幹が揺らいでしまい、最終的にテーマの見直しまでに落ちてしまった。よって、今後テーマを見直す上で大切なのは、ブレインストーミングなどを通していろいろな可能性を吟味した上で、徐々に一つに絞り込むことが大切だと再認識した。

( 文責: 矢吹溪悟 )



## 付録 A 新規習得技術

### 【Swift】

#### 《キーワード》

Swift・Xcode・SpriteKit・iPhone・iPad・Mac・オブジェクト指向・iOS アプリケーション

#### 《概要》

- 2014 年に WWDC で発表された、新しいオブジェクト指向型のプログラミング言語
- Objective-C に代わる iOS アプリケーション開発言語
- 開発環境は Mac、実機テスト用に iPhone や iPad が必要で、開発エディタは Xcode が推奨される
- 開発する上で、iOS Developer Program への登録が必要
- SpriteKit という 2D カジュアルゲーム用のフレームワークなど、幾つかテンプレートが用意してある

#### 《Swift の特長》

- 速い：Objective-C より実行速度が高速
- モダン：プログラミングの書き方が新しい
- 安全：プログラミングにエラーが起きにくい仕組みが増える

#### 《Objective-C との主な相違点》

- 行末のセミコロンや制御文の ( ) が不要
- メソッドの記述方法が「. メソッド名 ( )」と一般的な記述に
- 変数に nil が代入されるとエラーが表示され、安全性が向上した
- 別クラスにアクセスするときも import が不要

#### 《Xcode とは》

- iPhone アプリを作るための開発ツール
- アプリを作るのに必要な作業を全て行う

( ex : アプリ画面のデザイン・プログラムの入力・実行ファイルの作成 )

- iOS シミュレーターで実機を使わなくても粗方のデモストレーションが行える

#### 《SpriteKit を使うメリット》

- 文字やグラフィックスを素早く表示させたり動かしたりすることができる
- 2D の物理エンジンがついているため、物理的な動きをシミュレートさせることができる

### 【Git / GitHub】

#### 《キーワード》

Git・GitHub・分散型・バージョン環境システム・リポジトリ・ローカル・保存 ( コミット )・

## オープンソース

### 《概要》

#### [ Git ]

- プログラムソースなどの変更履歴を管理する、分散型のバージョン管理システムのこと
- Linux の開発チームが使用していたことがきっかけとなり、徐々に世界中に広まった

#### [ GitHub ]

- Git の仕組みを利用して、世界中の人々が自分の作品（プログラミングコードやデザインデータ、ドキュメントなど）を保存・公開することができるウェブサービスのこと
- 運営は GitHub 社で、個人・企業を問わず無料で行うことができる
- 基本的にオープンソースだが、有料サービスを利用するとプライベートなりポジトリも作ることができる

### 《Git と従来品の比較》

#### [ Git ]

- 自分のパソコンなどのローカル環境に、全ての変更履歴を含むリポジトリが複製される
- 各ローカル環境がリポジトリのサーバーとなれる
- ローカル環境にもコードの変更履歴を保存（コミット）できるので、リモートのサーバーに接続する必要がない
- ネットワークに接続していなくても作業ができる

#### [ 従来 ]

- サーバー上にある一つのリポジトリを共同で使っていた
- 利用が増えると変更内容の衝突が頻繁に起きる
- 整合性を維持するのが大変

## 付録 B 活用した講義

### 【情報デザイン 1】

#### 《キーワード》

Adobe Illustrator・図解表現・ポートフォリオ

#### 《授業内容》

- Adobe Illustrator の使い方（個人）
- 市立函館博物館を題材とした図解表現（個人）
- 図解表現の説明も兼ねたポートフォリオの作成

#### 《活かせる技術・知識・経験》

- 画像やアプリ素材を Adobe Illustrator で作成すること。
- 議事録や発表ポスターの素材を図解を用いて表現すること。

### 【情報デザイン 2】

#### 《キーワード》

グループワーク・タンジブル・アクティングアウト（寸劇）・プロトタイピング・ストーリーテリング・プレゼンテーション・ポートフォリオ・図解

#### 《授業内容》

- 時計の分析（グループ）
- スマートウォッチの分析（グループ）
- タンジブルな提案（グループ）

#### 《活かせる技術・知識・経験》

- 机に座って話し合いするよりも、手足を動かして考える方がより良い提案になること。

### 【情報表現基礎 3】

#### 《キーワード》

グループワーク・観察・フィールドワーク・アクティングアウト（寸劇）・プロトタイピング・ストーリーテリング・スケージュERING・プレゼンテーション・ポートフォリオ・図解

#### 《授業内容》

- カバンのスケッチを通した観察（個人）
- オートマタ制作（個人）
- 西部地区のフィールドワークを通して、より西部地区に足を運べる物・事・企画の提案（グループ）
- 各ポートフォリオの作成

#### 《活かせる技術・知識・経験》

- 情報共有をしっかりと行わないと、メンバー間の考えのズレやタスクの進捗に影響すること。

- 提案のユーストーリーを考えて、提案がユーザーにどういう変化を与えるのかを常に考えながら、作成していくこと。
- 提案をプロトタイピングし、デモンストレーションやアクティングアウトなどを行う。それを通して、問題点・優良点・改善点などを発見しやすくすること。
- プレゼンテーションにアクティングアウト（寸劇）や提案のデモンストレーションを盛り込んで、発表に説得力を持たせること。

## 参考文献

- [1] Code 部. 5 歳からプログラミング必修化！？世界の最新 IT 教育トレンドまとめ —Code 部, 2015. [http://blog.codecamp.jp/programming\\_education/](http://blog.codecamp.jp/programming_education/) (2015/7/20)
- [2] TechAcademy. プログラミングが義務教育に！政府の成長戦略素案に盛り込まれたプログラミング教育の内容とは —TechAcademy マガジン, 2013. <http://techacademy.jp/magazine/736> (2015/7/20)
- [3] Code 部. 大人も子供も楽しめる！プログラミング入門ゲーム「Scratch」をやってみた—Code 部, 2015. [http://blog.codecamp.jp/try\\_scratch/](http://blog.codecamp.jp/try_scratch/) (2015/7/20)
- [4] 日本経済新聞. 中学の技術・家庭科で「ビジュアルプログラミング」を導入:日本経済新聞, 2012. [http://www.nikkei.com/article/DGXNASFK2701H\\_X21C12A2000000/](http://www.nikkei.com/article/DGXNASFK2701H_X21C12A2000000/) (2015/7/20)
- [5] ヴィストン株式会社. 計測器プログラマー —ヴィストン株式会社, 2012. <http://www.vstone.co.jp/products/mcprogrammer/> (2015/7/20)
- [6] コードアカデミー高等学校. コードアカデミー高等学校, 2015. <http://www.code.ac.jp/> (2015/7/20)