

公立はこだて未来大学 2016 年度 システム情報科学実習  
グループ報告書

Future University Hakodate 2016 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

グループリーダー/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 梶勢也 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

2016 年 1 月 20 日

Date of Submission

January 20, 2016

## 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。

前期は、各メンバーが教育に関わるアプリを考え、中学生向けプログラミング入門アプリを作成することに決めた。しかし、中間発表での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ことにした。そのため後期は、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。最終成果発表会での、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.

In the previous period, each member considered the application of educating and the theme was decided in programming guide application for junior high school students. But 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 in the middle announcement, and the requirement definition was reconsidered again, and we redo design of an application. Therefore, 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 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” in an evaluation seat from teachers and other students in the final announcement 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	プログラミング教育の現状と課題	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	「子どものためのプログラミング入門 ワークショップ」への参加	5
4.1.2	リスク分析	6
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	9
4.1.7	KPT 分析	9
4.2	アプリ案の変化と内容	11
4.2.1	アプリ案の検討	11
4.2.2	大学生向けプログラミング入門アプリ	12
4.2.3	中学生向けプログラミング支援アプリ	15
第 5 章	前期の開発アプリについて	16
5.1	概要	16
5.2	プログラミング画面	16
5.3	戦闘画面	19
5.4	ゲーム性	19
5.5	教育性	19
第 6 章	前期の結果	21
6.1	プロジェクトの評価	21
6.2	プロジェクトの成果	21
第 7 章	後期の背景	22
7.1	公立はこだて未来大学の C 言語の講義について	22
7.2	プログラミング基礎の現状と課題	22
第 8 章	後期のプロジェクトの目標	23

8.1	開発アプリの目標	23
8.2	プロジェクト学習としての目標	23
<b>第9章</b>	<b>後期の活動</b>	<b>24</b>
9.1	アプリ案の推移	24
9.1.1	夏休みの活動	24
9.1.2	教育題材の見直し	27
9.1.3	デバッガーツールの企画	28
9.1.4	アニメーション教材の企画	29
9.2	イベント	29
9.2.1	HAKODATE アカデミックリンク 2015	29
9.2.2	最終成果発表会	29
9.2.3	振り返り	30
<b>第10章</b>	<b>後期の開発アプリについて</b>	<b>31</b>
10.1	概要	31
10.2	コンテンツ	31
10.2.1	概念の説明	31
10.2.2	例題	32
10.2.3	確認問題	33
<b>第11章</b>	<b>後期の結果</b>	<b>38</b>
11.1	プロジェクトの評価	38
11.2	プロジェクトの成果	38
<b>第12章</b>	<b>今後の課題と展望</b>	<b>39</b>
12.1	開発アプリの課題と展望	39
12.2	メンバーの課題と展望	39
<b>第13章</b>	<b>学び</b>	<b>41</b>
13.1	グループとしての学び	41
13.2	各メンバーの学び	41
13.2.1	熊谷優斗	41
13.2.2	梶勢也	42
13.2.3	新保遥平	42
13.2.4	中進吾	43
13.2.5	矢吹溪悟	43
<b>第14章</b>	<b>おわりに</b>	<b>44</b>
<b>付録A</b>	<b>新規習得技術</b>	<b>45</b>
<b>付録B</b>	<b>活用した講義</b>	<b>47</b>
<b>参考文献</b>		<b>49</b>

# 第 1 章 はじめに

当初、私たち教育系チームはこの 1 年間を通して、教育という視点でアプリの開発を行った。特にユーザーにプログラミング学んでもらうにという点に重点を置いて活動を行った。

私たちは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。しかし後期からは HTML や Processing を用いた Web アプリの開発に移行した。よって、前期と後期の背景や開発したアプリは異なっている。

第 2 章から第 6 章までが前期の活動である。前期では、フィールドワークを行い、ユーザーの視点を理解したつもりであったが、ユーザー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、実装を進めた段階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、アプリ案の再提案という手戻りが発生した。前期の教育系の活動はここまでであった。

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、ユーザーを意識したアプリ案の提案を行った。しかし、アプリ案の提案に慎重になりすぎてしまい、即座にアプリの実装に進むことができなかった。

( 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 Scratch や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

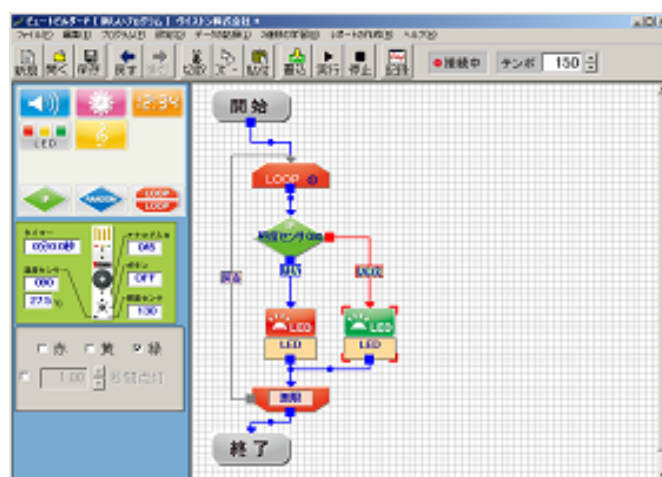


図 2.2 ビュートビルダーの画面

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入力して、プログラミングをするということを行っていない。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 プログラミング教育の現状と課題

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入力するソースコードを書く時、どのように書いていいかわからない。Web サイトやアプリなどのシステム開発を行う際、基本は文字を入力する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

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くアプリ開発し、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である Git を使い、効率よくアプリ開発する。第 2 章で述べた課題から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する。そして、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入力するようなソースコードの書き方を理解できるようになることを目標である。HAKODATE アカデミックリンク 2015 や成果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 App Store にリリースすることを目指している。

( 文責: 梶勢也 )

### 3.2 プロジェクト学習としての目標

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き、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。また、グループメンバーの得意分野や不得意分野を理解し、適切に活動できるように役割分担を行う。プロジェクト学習としての最終的な目標は、グループメンバーがプロジェクト学習が始まる前に定めた各々の到達目標を達成し、成功や失敗から多くのことを学ぶことである。

( 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 「子どものためのプログラミング入門 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、中学生と触れ合い、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年生の生徒達計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、メンバーは生徒の側についてプログラミングのアシスタントをした。

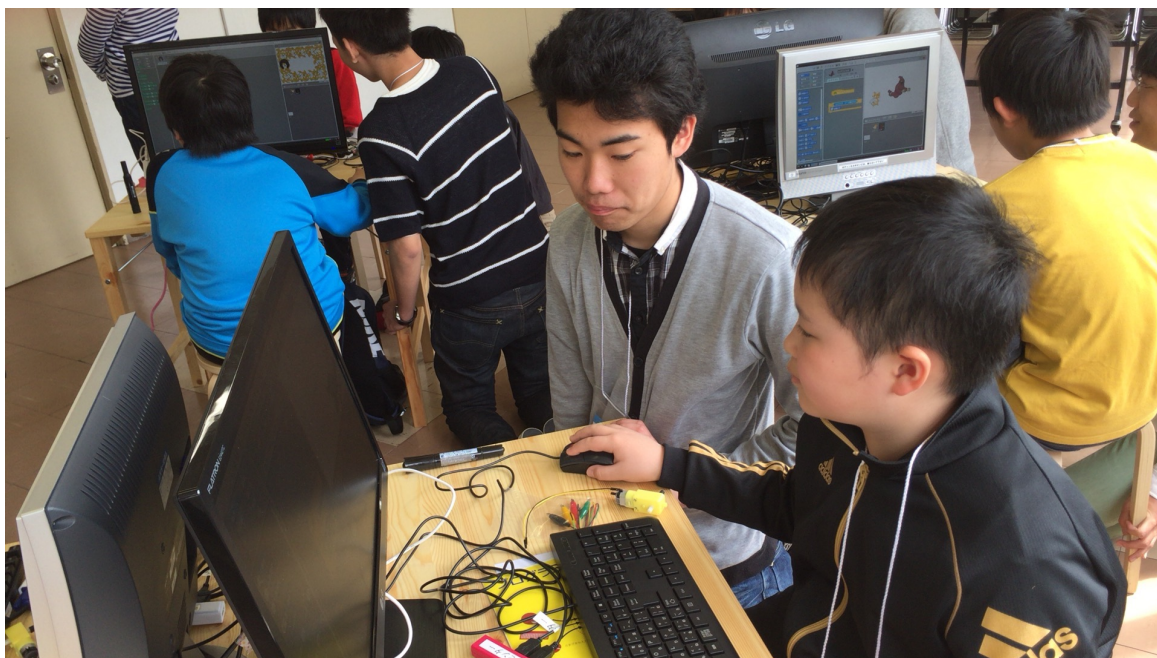


図 4.1 「子どものためのプログラミング入門 ワークショップ」の様子

ワークショップを通して、気づいた点は次の2点である。

- 生徒達は、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から引き続き参加している生徒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、生徒自らの点の修正を行っていた。生徒の成長の早さに驚いた。
- 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いない状態で作成したアンケートだったため、内容が適切なものではなく、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 4.2 は洗い出したリスクの一部である。

リスク分析を行ったことで、メンバー間の情報共有が適切にできていないことが明らかになった。例を挙げると、前述の「子どものためのプログラミング入門 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 人に連絡が繋がらなかった。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落ちる。	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	遅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落ち、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。	普段からに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 4.2 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

プロジェクト全体の活動として、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が開かれた。勉強会は、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 種類に分けて行われた。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、MapKit 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方法を学びつつ、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、作業完了までのタスクを分割することができず、上手くコーディングの役割分担を行うことができなかった。そのため、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。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、分割したタスクをメンバーにアサインすることに決定した。

( 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 4.3 は 6、7 月分のバックログの原案である。この時、バックログは Google スプレッドシートを用いて作成した。はじめに設計を行っていくことをしたかったため、「画面遷移図を作成」や「マップ画面設計」といった、設計に関するタスクの優先順位を高くした。

この原案を企業講師の方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」というアドバイスを頂いた。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」の機能をバックログに取り入れることにした。図 4.4 は実際に使用しているバックログである。図のように、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。

	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 4.3 6、7月分のバックログの原案

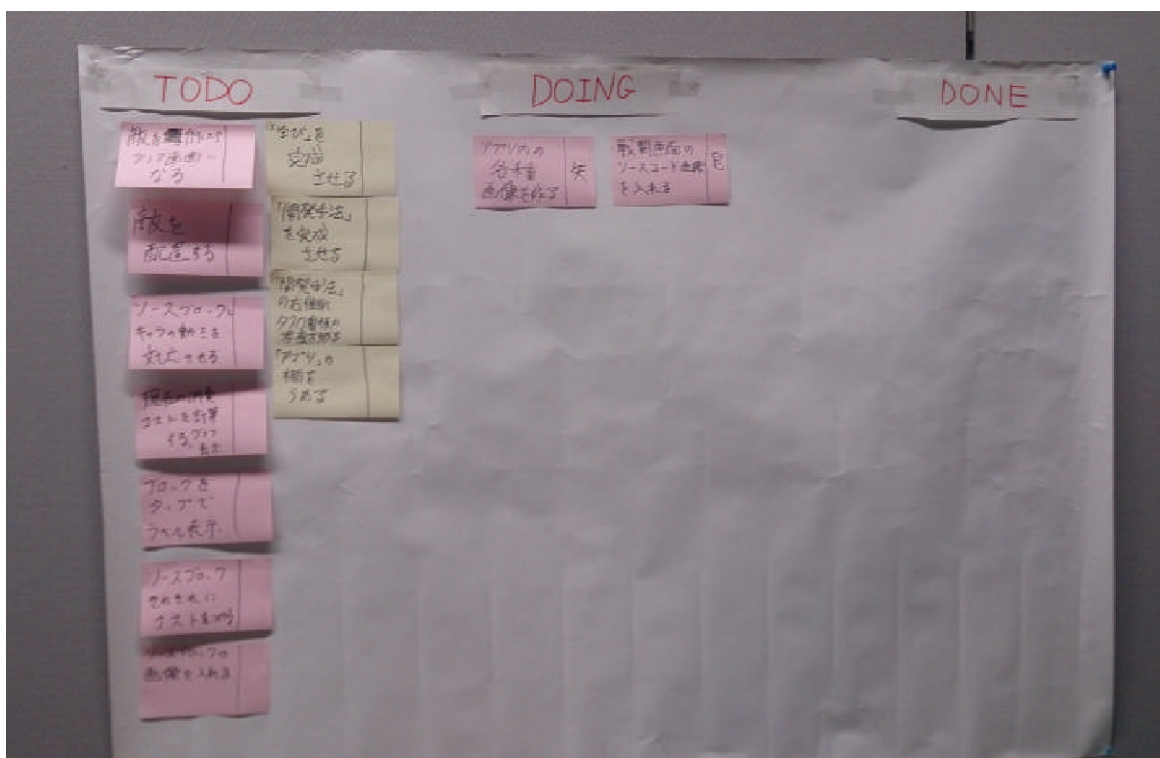


図 4.4 タスクかんぱんのシステムを取り入れたバックログ

( 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、実装班が忙しく、ポスターのレビューに時間を割くことが出来なかったため、メンバーそれぞれの意見をポスターに反映させることができなかった。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、というアドバイスを頂いた。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

見づらいというアドバイスを頂いた。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、というアドバイスを頂いた。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。

( 文責: 熊谷優斗 )

#### 4.1.7 KPT 分析

前期で行ってきた活動を振り返る際に、Keep(今後も続けていきたいこと)、Problem(課題)、Try(今後やっていきたいこと)の3つについて話し合い、KPT分析を行った。その内容を以下に記す。

- Keep(今後も続けていきたいこと)
  - 報告書をバージョン管理できた。
  - TA に積極的に質問をした。
  - ソースコードにコメントをたくさん書けていた。
  - GitHub-Flow を正しく運用できていた。
  - 中間発表で相手の顔を見て話せていた。
  - 実装力があつた。
  - 活動場所の確保ができていた。
  - プロジェクト全体を支える活動も行えていた。
  - 全員にやる気があつた。
  - それぞれの得意分野を理解できた。
  - 役割分担をした。
  - 前向きに取り組んだ。
  
- Problem(課題)
  - スクラムを用いた開発ができていない。
  - チーム内での情報共有がうまくできていない。
  - リーダー任せにしている部分がある。
  - 人の意見を批判するだけで、改善案を出さない。
  - メンバーの予定が合わない。
  - 教育についての理解が足りない。
  - 先生が話しているのに下を向いてしまっている。

## “Swift” Application Development Based on Field Research

- プロジェクトが始まる前にアジェンダを出していない。
  - 他の班との情報共有ができていない。
  - 報告書を完成させるのが遅かった。
  - タスクかんばんの進捗確認を行っていない。
  - 中間発表会での発表方法に問題があった。
  - 計画性が乏しい。
  - フィールドが定まっていない。
  - 要件定義を疎かにしてしまった。
  - プロジェクトの時間をうまく活用できていない。
  - 会議のファシリテーションがうまくいってない。
  - 自由勝手な行動をしてしまった。
- Try(今後やっていきたいこと)
    - 他の班から意見をもらうようにする。
    - 先生からさらに意見をもらう。
    - 教育についてもっと知る。
    - フィールドワークを行う。
    - 活動時間を有効に活用する。
    - 会議の始まる前にアジェンダを考えるようにする。
    - すぐに連絡取れるようにする。
    - 会議はいつまでどの内容を話すのか明確に決め、メンバーもそれに従う。
    - 必ず議事録を残す。
    - タスクかんばんの進捗確認を行う。
    - 要件定義をしっかり行う。

KPT 分析の結果をまとめると、GitHub を正しく運用できていたため、実装や報告書の作成をスムーズに行うことができていた。また Teaching Assistant(以下 TA) に頻繁にレビューを行って頂いたことで、間違いに気づくことができていた。しかし活動前にアジェンダを決めていなかったことやファシリテーターがいなくて計画的にミーティングを進めることができたかったことや、要件定義を正しく行えていないことがわかった。今後はミーティング毎にアジェンダを決めること、またファシリテーターを決め話し合いをうまく進めることや、教育についてのフィールドワークを行うことを目標にした。メンバー個人の問題や全員の問題などさまざまなことが振り返りを行うことで明確になった。

次に KPT 分析の結果から、夏休みの活動計画を立てた。フィールドがしっかり定まっていないという問題から、夏休みはメンバーそれぞれが教育に関するワークショップを調べることや他大学の教育についての PBL・研究などを調べ、夏休みが明けたらフィールドワークを行うことを目標にした。

( 文責: 皂勢也 )

## 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。図 4.5 にその流れを示す。まず、メンバーそれぞれが考えてきた案について、評価を行った。その結果、「プログラミングを学ぶゲーム」案を採用した。この案についての深く考えた結果、未来大学生を対象ユーザーとしたアプリを作成する方針に決まった。しかし、この案は背景に根拠が少なかったため、対象ユーザーを中学生へと変更した。次章から、そのプロセスを詳しく記す。

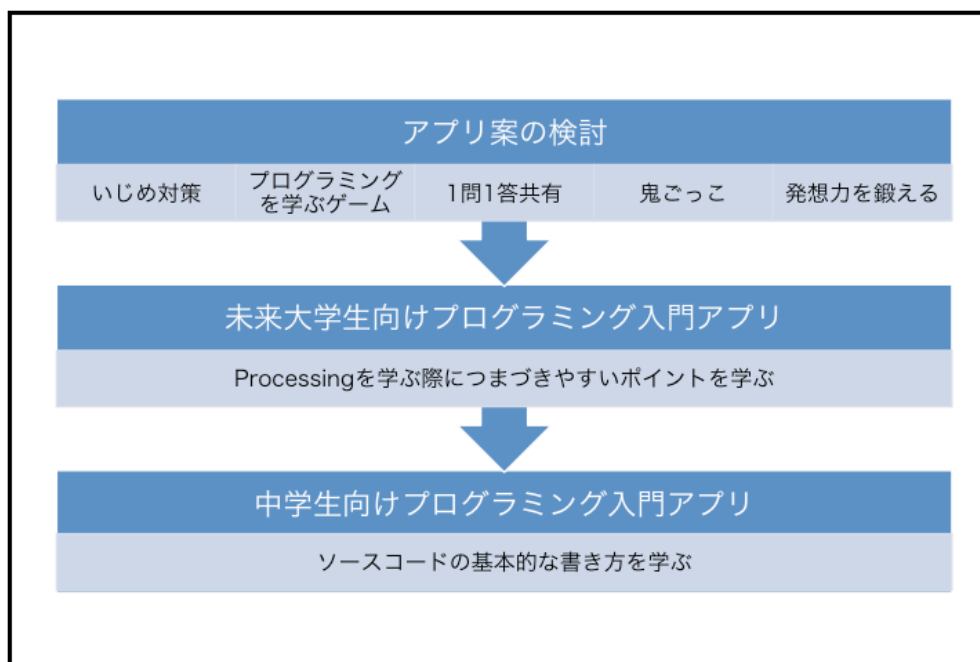


図 4.5 アプリ案の変化

( 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、次の 5 種類の案に絞った。

- 案 1 いじめ対策アプリ
- 案 2 プログラミングを学ぶゲームアプリ
- 案 3 発想力を鍛えるゲームアプリ
- 案 4 1 問 1 答共有アプリ
- 案 5 外遊び支援アプリ

5 種類それぞれの案をメンバー内で肉付けした後、TA と担当教員にレビューをして頂いた。それぞれの案の詳細とアドバイス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。



アドバイス内容 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。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。

アドバイス内容 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 発想力を鍛えるゲームアプリ ユーザーがプログラムの作成側とレビュー側の両面で機能することにより、全ユーザーと一緒に発想力を磨いていくアプリである。自分だけが納得するような発想ではなく、他人にも共感させるような発想ができる力を磨くことが目的である。

アドバイス内容 ユーザー依存型アプリは投稿が増えないとアプリが発展していか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 中学生や高校生が受験勉強やテスト勉強のために作った 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。

アドバイス内容 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が対象ユーザーである。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ことが目的である。その 1 つの案として挙げたのが、GPS 機能を使った鬼ごっこアプリである。

アドバイス内容 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、プログラミングについての教育を行いたい、という意見が合致したため、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。

( 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、函館に関連する要素を追加しよう」、「子供は地元の要素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学生にしよう」という理由から、公立はこたて未来大学(以下、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、未来大学 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 4.6 はアプリの全体の流れを表した図、図 4.7 はアプリ内画面の一部であ

る、「プログラミング画面」の図である。

「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、ユーザーに良いアルゴリズムを学んでもらうため、「コスト」という概念を定義した。ブロックそれぞれをコストで重み付け、1ターンで使用できるコストの上限を制限した。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と考えた。

このアプリ案を TA、担当教員、企業講師の方にレビューしていただいたところ、様々なアドバイスをいただいた。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、実際のプログラミングと異なり過ぎていて、未来大学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。

このアドバイスを受けて、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。

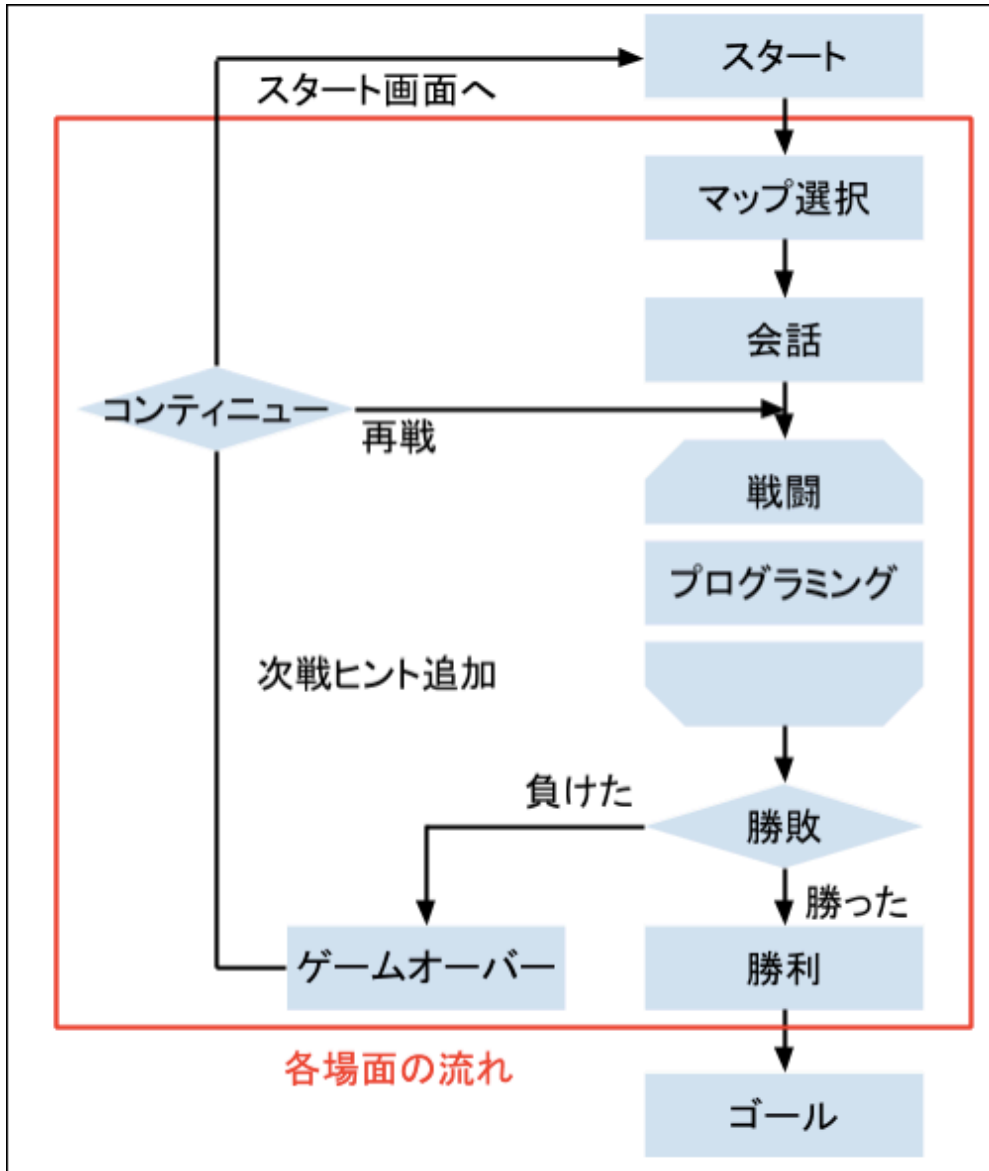


図 4.6 アプリの全体の流れ

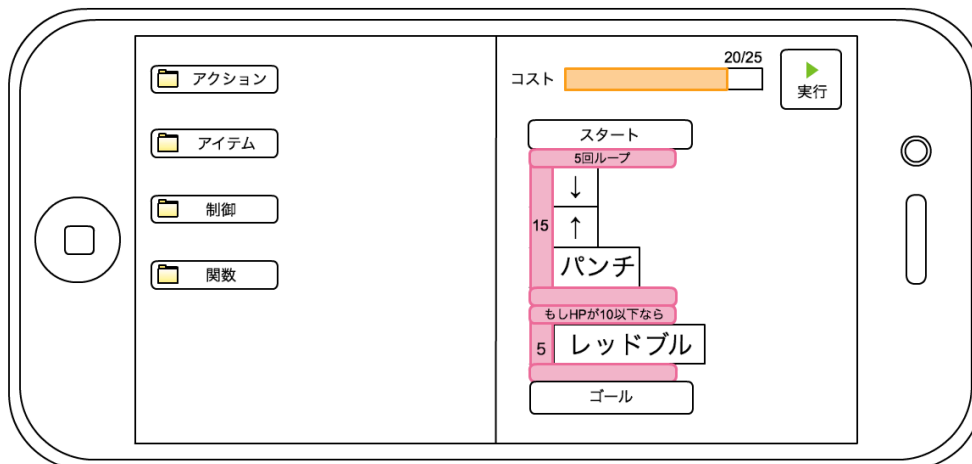


図 4.7 プログラミング画面

#### 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に根拠が少なく、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。話し合いの結果、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。その理由は2つある。1つは、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こと。もう1つは、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ことである。これにより、決定したのが第2章にて記載した背景であり、第5章にて記載するアプリ案である。

## 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、また興味を持った人を対象としたソースコードの書き方を学ぶゲームアプリである。図 5.1 のように、このゲームにはユーザーが操作するプレイヤーとして自機がある。またユーザーの敵プレイヤーとして敵機がある。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入力する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を目指すゲームである。

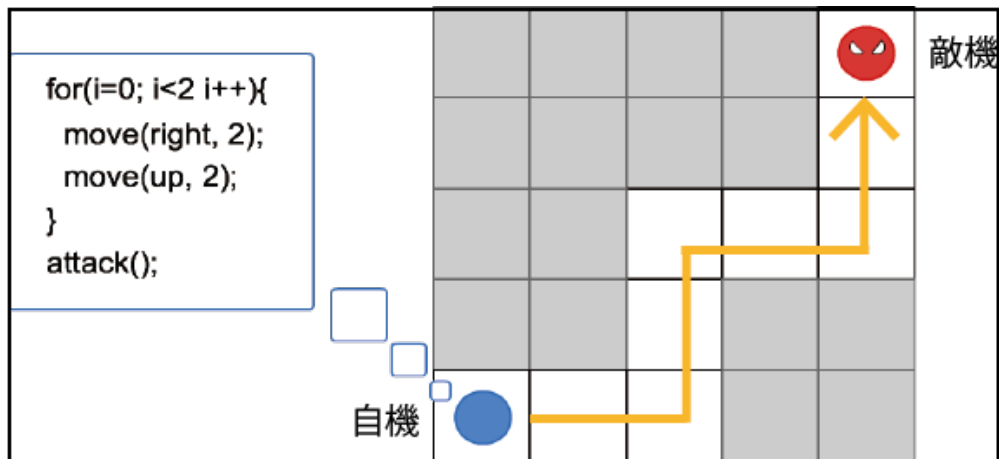


図 5.1 ゲーム概要

( 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 5.2 のように画面に配置されたソースボタンをタップする。

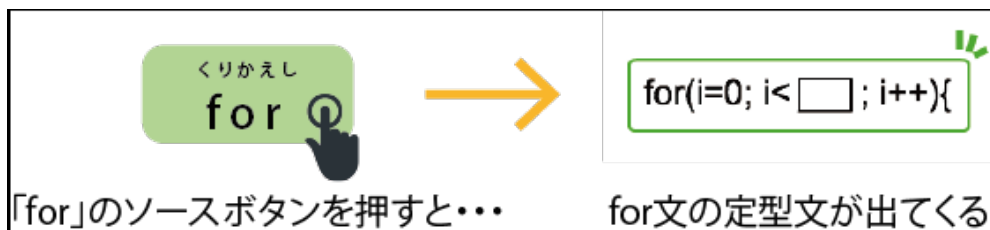


図 5.2 ソースコードの入力

ユーザーは図 5.3 のように画面左側に配置されたそれぞれのソースボタンをタップして、ソースコードを書いていく。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。

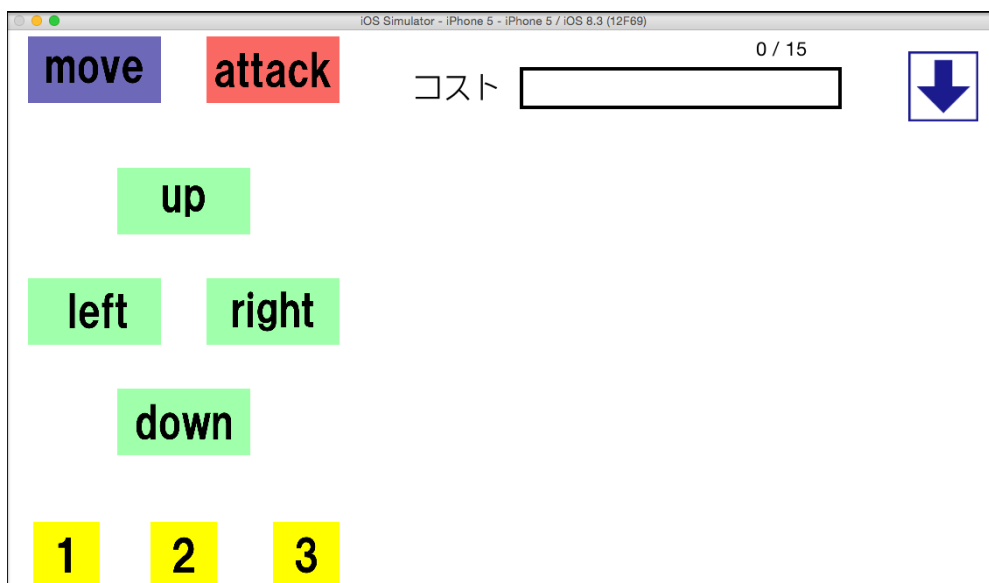


図 5.3 プログラミング画面

例えば下記のようなプログラムを書くとする。

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

このソースコードを図 5.4 のプログラミング画面に入力した。

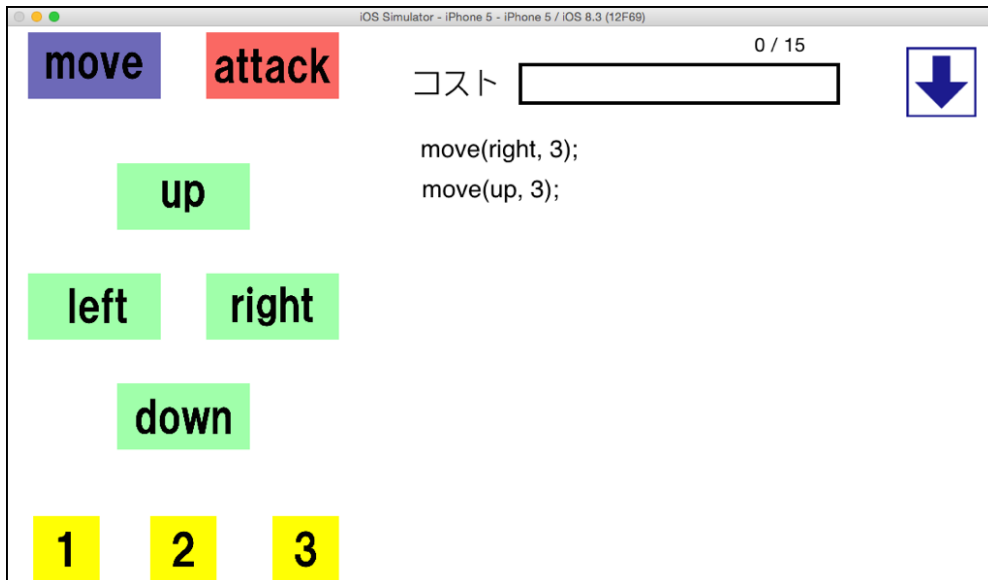


図 5.4 ソースコード入力後のプログラミング画面

図 5.4 のようにソースボタンを 1 つずつタップで入力することでソースコードを書くことが出来る。また、図 5.5 のように

`move(right,`

のような、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タップすると画面上にエラーが表示され、すぐに確認ができる。

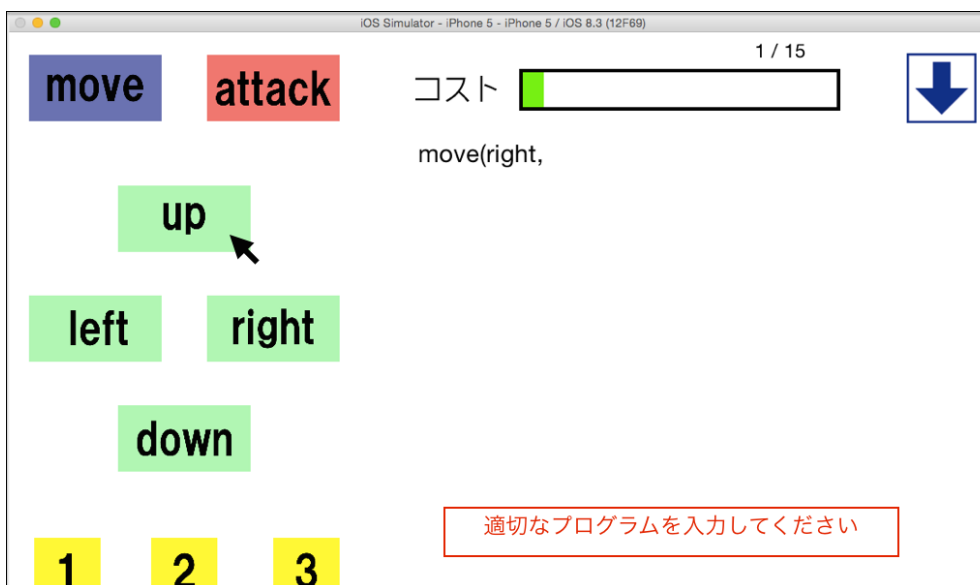


図 5.5 ソースコードがエラー時のプログラミング画面

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右上の「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 5.6 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できる。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。

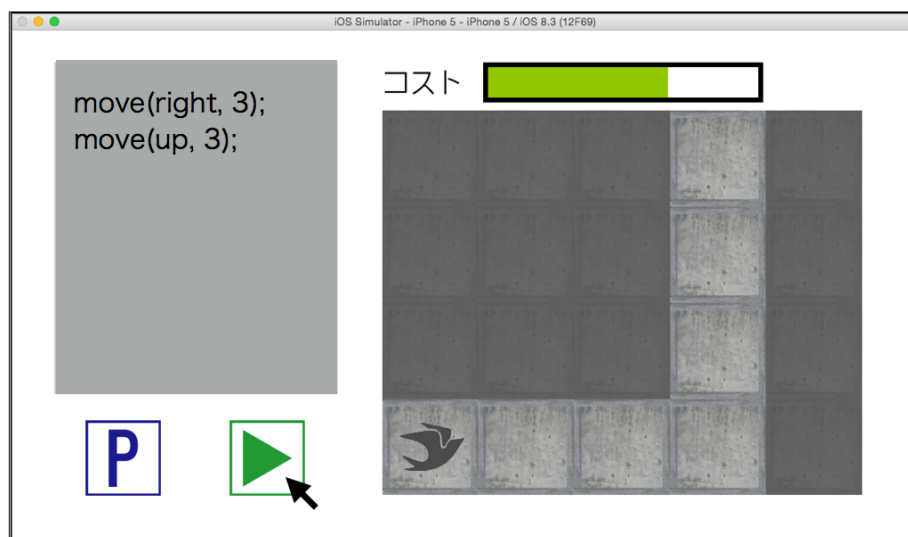


図 5.6 戦闘画面

( 文責: 新保遥平 )

### 5.4 ゲーム性

私たちはこのアプリで、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考えた。また実際にソースコードを書くことでプレイヤーである自機を思い通りに動かすことが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 )

### 5.5 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書けるようになるために、コストとランクがある。コストとはソースボタンそれぞれに設けられているコストのことである。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書くことができたと判定し、図 5.7 のように A ランクや B ランクなどのランクを与える。与えられたランクが低かった場合、ユーザーに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書



く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にしたものを図 5.8 に示す。

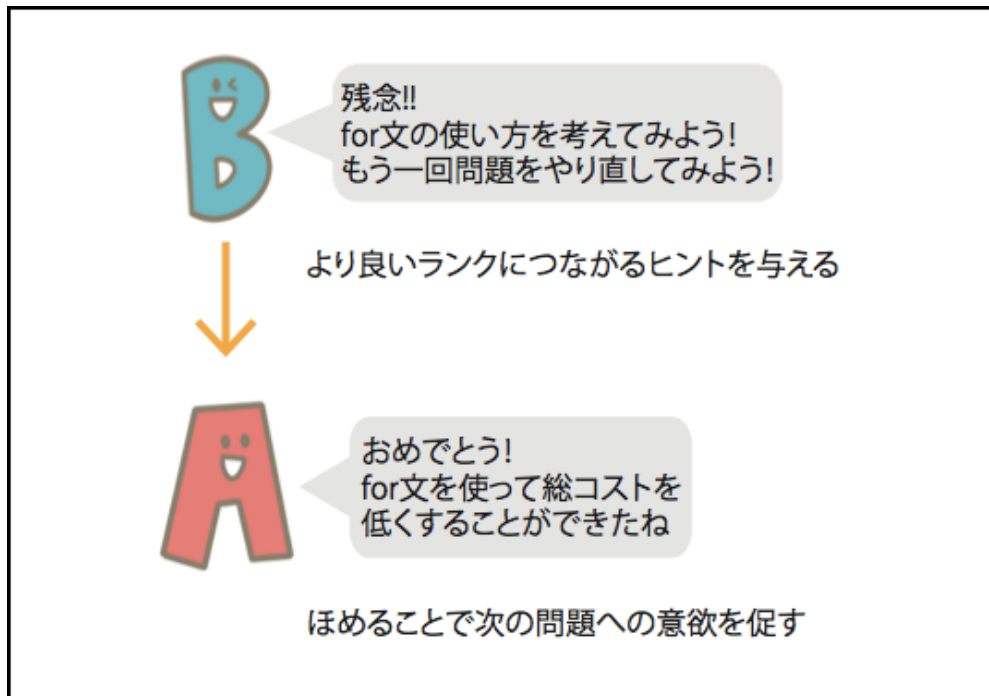


図 5.7 ランクとヒント

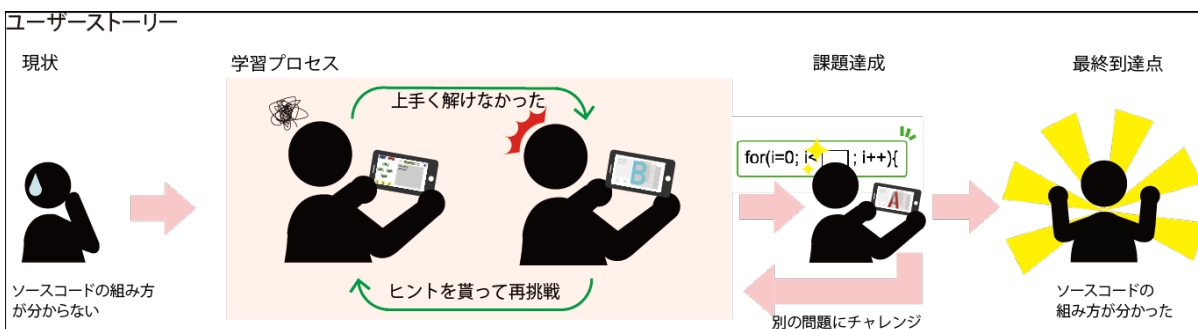


図 5.8 ユーザーストーリー

( 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いただいている。しかし、私たちが考えたアプリは課題を解決できるアプリとなっていない。そのため、要件定義をやり直す必要がある。

また、7月に行われた中間発表会では「声ははっきり聞こえた」、「声が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」、「既存のものとの比較がない」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、本プロジェクトの活動はわかりづらいので、背景、課題、目的につながるアプリ案が必要である。そして、第三者にアプリ案の背景、課題、目的が伝わる一貫性のあるもの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。

- 「子どものためのプログラミング入門ワークショップ」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ところ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 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。しかし、4.1.5 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会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

後期は、前期に考えたアプリ案を見直し、教育について考え直した。その結果、C 言語をアニメーションで教えるという以下の案に決めた。こういったプロセスで以下の案に決めたかは、9.2 で述べる。

( 文責: 中進吾 )

### 7.1 公立はこだて未来大学の C 言語の講義について

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の基礎である変数、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 ( プログラミング ) 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 [7]。しかし、例年「プログラミング基礎」を不合格になる 1 年生が多くいる。

( 文責: 中進吾 )

### 7.2 プログラミング基礎の現状と課題

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 にヒアリング調査を行った。

履修している 1 年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 1 年生が多くいた。実際に、配列が分からない 1 年生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おうと、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 1 年生は講義のどのような点を分かってい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いた。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。

( 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、講義で教える学習單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用い、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、ユーザーが学習単元の概念を理解し、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、ユーザーが適切なアルゴリズムを学べる機能を備える必要がある。迅速にプロトタイプを作成し、より多くの 1 年生にプロトタイプの評価を行ってもらい開発アプリをより良いものにしていく。また、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: 自勢也 )

### 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い、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: 自勢也 )

## 第 9 章 後期の活動

### 9.1 アプリ案の推移

#### 9.1.1 夏休みの活動

この章では、プロジェクト学習中間発表会及び中間報告書提出以降の活動から、夏休み中の活動を報告するまでの経緯を示す。この期間の主な活動内容は、2点ある。1つは、プロジェクト学習中間発表会の反省を踏まえて、一度提案を企画立案段階まで戻したことである。そしてもう1つは、企画立案段階に戻したことにより、夏休み中に調査・情報収集を行ったことである。

まず、プロジェクト学習中間発表会の反省を踏まえて、一度提案を企画立案段階まで戻したことについて述べる。私たちは前章でも述べた通り、プログラミングをするにあたって必要不可欠なアルゴリズムを、パズルゲームを通して学ぶことができる、中学生向けのプログラミング支援アプリケーションを開発した。しかし、このアプリケーションにはいくつかの問題があり、特に以下の3つが重大な問題となった。1つ目の問題点は、目的と提案が噛み合っていないため、アプリケーション全体に一貫性が薄れ、説得力に欠けるものになってしまったことである。2つ目の問題点は、提案物に対する調査を深く行っていなかったため、提案したゲームに酷似したアプリケーションがあることが後からわかり、新規性にかけてしまったことである。3つ目の問題点は、企画立案や要件定義にあまり時間をかけずに、設計工程に入ってしまったことである。

これらの問題から、中学生にプログラミングの教育を支援するという目的を解決することが非常に難しいゲームになり、さらに現状分析や先行事例の調査が不十分なため、結果的に新規性に欠けるゲームとなった。これらの理由により、プログラミングをゲームで解決することや、そもそもプログラミングという題材は教育において適切なのかという疑問が浮上した。したがって、メンバー内の会議の末に、提案を題材自体から見直す形で企画立案段階に戻した。

次に、企画立案段階に戻したことにより、夏休み中に調査・情報収集を行ったことについて述べる。企画立案段階に戻すに当たって、まず始めにやらなければならないことは、教育にふさわしい題材を見つけることである。よって、適切な題材を見つけるために、夏休み中にチームメンバーが各自で教育という観点からアプローチできる、ジャンルや事例の調査・情報収集を行うことが決定した。また、もし教育に関連するワークショップなどのイベントが開催された場合、積極的に参加するようにした。なお、ここで述べている教育にふさわしい題材というのは、教育を行うジャンルやカテゴリのことである。前期の提案物を例にあげると、プログラミングが題材にあたる。以降、教育にふさわしい題材を「教育題材」と呼ぶことにする。また、夏休み中に調査・情報収集を行った活動は、夏休みの前半の約3分の2の期間に行い、残った期間で共有や分析などを行う予定だった。しかし、チームメンバーそれぞれの予定の都合上プランを変更し、夏休み明けにチームメンバーの夏休みの活動内容を共有した。

夏休み中に行った、教育という観点からアプローチできる、ジャンルや事例の調査・情報収集、及び教育に関連するイベントの参加した結果を、以下に列挙する。まずは、教育に関連するワークショップに参加した結果について記述する。

メンバーの1人が、間々田先生という方が主催したワークショップに参加した。間々田先生は、盲学校で理科の教育を担当している先生である。参加メンバーは12人で、そのうち1人は小学校

5年生の全盲の少年であった。実施されたワークショップは3つあり、「天体の大きさを実感する」と「雨粒をつかまえる」、「1000個の積み木を積み上げる」というタイトルであった。ワークショップの詳細を順番に見ていく。

1. 「天体の大きさを実感する」ワークショップでは、紙粘土を用いて1cmの地球を作り、そのサイズを基準として月や木星、太陽の大きさを実感した。なお、太陽は地球の100倍の直径であり、これは風船を使うことでその大きさを体感した。
2. 「雨粒をつかまえる」ワークショップでは、小麦粉の中に入った水滴はその水滴と同じサイズの玉を作るという性質を利用して雨粒のサイズを実感した。ちょうど雨粒サイズの水滴を作るために、卓上箒に水をつけて振ることで、雨粒と同じぐらいの水滴を飛ばすことができた。また、小麦粉をふるいの代わりとしてガーゼを用いて玉だけを取り出し、さらにこれを焼き上げることで固まり、触って大きさを確かめることができるようになった。
3. 「1000個の積み木を積み上げる」ワークショップでは3cm立方の積み木を1つ、2つ、3つというように増やしていき積み立てていき、4チームそれぞれで1辺5の立方体を2個、これを合体させることで1000個の積み木を完成させた。

ここから、ワークショップを通して、間々田先生がどのような点に注意しながら進めているのか観察した。観察を踏まえわかったことは以下の5点である。

1. 作業をする際は生徒を立たせ、終わった人は座らせることだ。このようにすることで、大人数で作業していてもそれぞれの生徒の進捗が一目で把握することが出来る。
2. ワークショップで使ったもの、作ったものはビニール袋に入れて持ち帰らせることだ。持ち帰ることで、家に帰った時に親子間の会話の題材として使うことが出来る。さらにこの会話を踏まえて、子供は親に今日の経験を語るため、自然と振り返りを行うことができる。
3. 教材のにおいや触り心地に注意して選んでいることだ。これは、盲目の子は嗅覚や触覚をたよりにしているため、いい匂いや肌触りの良さは大切であるからだ。今回のワークショップでは手にくっつかない紙粘土を採用し、またひのきが材料の積み木を使用することで、心地の良い香りを演出した。
4. 触ることができる教材にすることだ。その場で作ったとしても、触ることができなければそれは映像教材となんら変わらないからだ。さらには、触って体験したものは忘れないという理由もある。
5. 間々田先生も最初からうまくワークショップを行えたわけではないことだ。先生自身がトライアンドエラーを重ねて、今回のような隔々にまで配慮されたワークショップを行えるようになった。

次に、ワークショップ中に全盲の少年がどのような反応をしているのかを観察した。観察を踏まえわかったことは以下の4点である。

1. 盲目だからといって特別変わった点はなく、一般的な小学5年生と同じように活発な男の子であった。
2. 「天体の大きさを実感する」ワークショップにて、直径110cmの風船を膨らませたが、「膨らむ時の感触が気持ちいい」と言っていたのが印象的であった。
3. 位置情報の記憶が鮮明であった。例えば机の上に置いておいたペットボトルがどこにあるのか、時間が経った後でも覚えていた。
4. 休み時間の際、「指スマしようよ」と持ちかけられ指スマをした。この時、全盲の少年に自

分が何本指を上げたのか、口で言うだけでいいのだろうかと戸惑ってしまったが、すぐに全盲の少年がこちらの指を触ることで確かめてきてくれた。

最後にワークショップ全体を通した感想をいくつか列挙する。

- 間々田先生のワークショップの進め方は参考になる点が多かった。例えば、生徒を立てて進捗を確認する点、教材を持ち帰らせる点、ところどころギャグを挟むことで場の空気を和ませる点などである。
- はじめは全盲の少年が盲学校の生徒なので、どのように接すればいいのだろうと不安に感じていたが、全盲の少年の方から手を繋いだり、指スマの件で指を触れてきたりしてくれたため、こちら次第に上手く接することができるようになったと思う。
- 間々田先生や全盲の少年に対して、教育系プロジェクトに関連する質問を1つも行うことができなかつた点は反省である。これは、事前に質問の準備を行うことをしなかつたことが最大の原因として考えられる。また、間々田先生や全盲の少年の発言に対して、客観的な視点から「なぜこうなるのか」と疑問に思っ、考えるべきであったとも感じている。
- こうしたワークショップを受ける側として参加することは様々な発見があり貴重であるため、機会があれば積極的に参加していくのがいいと感じた。

以上の内容が、教育に関連するワークショップに参加した結果についての記述である。続いて、教育という観点からアプローチできる、ジャンルや事例の調査・情報収集の結果について記述する。

まず、観光教育について調査した。観光教育とは、地元の魅力を地元の人たちに伝えるという活動のことを指す。このことから、観光に対して教育的アプローチがあることを発見する。例えば、「はこだて観光こども学習会」という、函館の魅力を函館の子供達に伝える活動が行われている。これに伴い、新幹線開通が背景の1つとして、子供達が観光大使となってくれることを目的としている。

次に、日本の教育に対する実態を調査した。現状の日本の教育問題として、自然体験のない青少年の割合が増加しているという問題点がある。例えば、子供の読書離れや子供の体力低下、朝食離れ、身近な伝統文化や現代の文化芸術に触れる機会などが挙げられる。

次に、世界の教育に対する実態を調査した。世界の教育問題として、世界には7億人の学校に通えない大人がいるという事実がある。教育を受けられないことで、その息子の代も教育を受けることができない可能性が非常に高く、負の連鎖にはまることが考えられる。

次に、教育に関する先行研究について調査した。以下に、他大学の研究から教育を題材にしているものを列挙する。

- 社会学教育分野では、地域活性化プロジェクトにより創造性を育む協働型体験学習（神戸松蔭女子学院大学人間科学部）や100人以上の大人数クラスでの学生参加型双方向授業（東海大学文学部）などがある。
- 統計学教育分野では、体験と主体的参加を特徴とする統計活用教育（同志社大学文化情報学部）やゲーム感覚を取り入れた統計教育の動機付けへの試み（実践女子大学人間社会学部）などがある。
- 教育学教育分野では、授業の実践力を促進するための協調型学修の工夫（創価大学教育学部）や授業アンケートを用いたアクティブ・ラーニングの検証（関西大学総合情報学部）などがある。
- コミュニケーション教育分野では、異文化コミュニケーション協働学修（早稲田大学商学

大学院)や多様性を育むための問題解決型学修の考察(常磐大学国際学部)などがある。

最後に、統計情報を扱っている Web サイトを検索した。その結果、生活定点と呼ばれる Web サイトがあがった [8]。

以上の内容が、夏休み中の主な活動内容である。ここから、これら情報をより詳細に分析をし、教育題材にしやすいものを選択し、テーマを確立し、企画を固めていくことが今後の活動となった。

( 文責: 矢吹溪悟 )

### 9.1.2 教育題材の見直し

前章では、前期の活動の反省から教育題材を見直し、夏休みに教育という観点からアプローチできる、ジャンルや事例の調査・情報収集、及び教育に関連するイベントの参加する活動を行った。この章では、夏休み中の活動を報告してから教育題材を再決定するまでの経緯を示す。この期間の主な活動内容は、夏休みの活動を踏まえて教育する題材を再検討したことである。

前章で述べた調査内容を分析する段階で、まず統計に着目し、分析を通して適切な教育題材があるかどうか検討した。以下に、統計の分析結果の例を列挙する。

- 家の中よりも野外で遊ぶことの方が好きな人が年々減少傾向にある。
- 1日に1時間以上ゲームをする小学生が年々増加傾向にある。
- 健康に気おつけた食事をしている人と健康に気をつけた生活をしている人が、ともに年々大きく減少傾向にある。
- お中元は毎年欠かさず贈っている人とお歳暮は毎年欠かさず贈っている人が、ともに年々大きく減少傾向にある。

また、これらの統計の分析結果と、ワークショップの参加や教育題材の調査・情報収集といった夏休みの活動の結果をもとに、教育題材についてブレインストーミングした。以下にブレインストーミングを通して考えられた教育題材の有力候補を記す。

- 観光教育(地元の魅力を地元民に教える)
- 日本の文化を教育
- こどもの健康支援
- 老人の健康支援

これらの結果、「観光教育」という教育題材が、チームメンバーの目に止まった。そこで、地元の函館に着目して、函館の魅力を函館に住む人に教える活動について検討した。しかしながら、この観光教育という教育題材に対して、2つほど大きな問題点もある。1つは、同プロジェクトの観光チームとの違いは何かという点である。これは、同プロジェクト内の別のチームに、木古内の観光を題材としているため、似たり寄ったりな、アプリケーションになってしまう可能性があったからである。もう1つは、根拠をもとに設定した教育題材ではあるが、チームメンバーが観光教育という教育題材で、最後までモチベーションを維持できるかという問題点である。これは、観光教育が本当にやりたいことなのかが疑問だったため、このような問題があがった。このような、問題点が残つつも、これらの問題点を解決する糸口を見出すために、担当教員に報告及びレビューをしてもらうことが決定した。

しかしながら、担当教員に報告する直前になり、教育題材を大幅に見直した。見直した理由の大



元は、上記の2つの問題点だ。見直した結果、再度「プログラミング」を教育題材にすることが決定した。その理由は2つあり、1つ目は、前期にプログラミングを題材にしていたため、少なくとも観光教育よりはノウハウが蓄積されているという点だ。もう1つは、チームメンバーが本当にやりたいことはプログラミングの教育であったということだ。このような理由から、プログラミングの方が観光教育よりも、経験値が豊富でかつチームメンバーの士気が向上するため、再度プログラミングを教育題材にするという結論に至った。なお、このことを担当教員に報告したところ、「やりたいことをやれ」という意見をもらった。

( 文責: 矢吹溪悟 )

### 9.1.3 デバッガーツールの企画

前章では、夏休みの活動を踏まえて教育題材を見直し、経験値が豊富でかつチームメンバーの士気が向上する、プログラミングを再び教育題材にしたことを述べた。この章では、プログラミングを教育題材にしてから、デバッガーツールという企画の有用性を判断するまでの経緯を示す。この期間の主な活動内容は、プログラミングのエラー文に着目し、デバッガーツールを企画したことである。

夏休みの活動を経て、再度プログラミングを教育題材にすることが決まったが、具体的な方向性はまだ決まっていなかった。そこで、プログラミングにおいて何が、教育の障壁となりうるのかメンバーと検討した。その結果、エラー文が読めないまたは読みにくい、仮に読めたとしても、そのエラー文の意図を把握しにくいという問題点があがった。この問題点を解決する糸口として、エラー文を日本語で出力するという方法と、エラー文をアニメーションで表現するという方法の2種類の解決方法が思いついた。特に、エラー文をアニメーションで表現するという解決方法は、新規性があり、直感的にかつ正確に意図を把握する事が期待できると考えた。また、対象は検証のしやすさや自分たちの考えやすさを優先し、今回は未来大学の特に1年生とした。さらに、デバッガーツールとして制作すれば使いやすいくと考えた。よって、プログラミングを教育題材として、教育の障壁をエラー文に絞り解決する、エラー文をアニメーションで表現するデバッガーツールが企画された。ちなみに、この企画はこの時点で、デバッガーツールを作った経験のある人がチームメンバーに居ないので、上手く作成することができるかという問題点もあった。

未来大学の1年生を対象とするため、対象言語は未来大学1年生が扱う processing と C 言語のどちらかを対象として検討するために、現状分析を行った。しかし、processing は、ver3 のエディターが「PDE X」という、エラー箇所を随時出力してくれるという高性能な機能を有していた。また、C 言語は「Clang」というコンパイラが、エラー箇所を示すだけでなく、改善案まで提示してくれることが判明した。これにより、エラー文をアニメーションで表現するデバッガーツールを作成しなくとも、既存のシステムでエラー文の難読性は解決可能という結論に至り、企画内容を見直す形となった。ただ、アニメーションでわかりやすく表現する手法は、他の局面でも用いることができると考えたため、次の企画へと持ち越すことにした。

( 文責: 矢吹溪悟 )

#### 9.1.4 アニメーション教材の企画

前章では、プログラミングを教育題材にしてから、エラー文の難読性からデバッガーツールという企画を提案したが、調査の結果、現状の技術で問題解決が十分可能と判明し、企画を見直したことを述べた。この章では、デバッガーツールという企画を見直してから、アニメーション表現という観点から別の企画を立てるまでの経緯を示す。この期間の主な活動内容は、アニメーション表現に着目し、未来大学の1年生がC言語の苦手意識を持つ学習単元を、学ぶことができるWeb教材を企画したことである。

前回のデバッガーツールの企画において、アニメーション表現というアイデアは良好だと判断した。そのため、アニメーションを有効活用できるようなプログラミング教育を考えた。その結果、我々は、未来大学の1年生が受講する「プログラミング基礎」の不合格者数の多さに着目した。そこで、ヒアリング調査の結果や講義資料を確認したところ、ヒアリング調査では、特に「配列」「構造体」「ポインタ」に苦手意識が持たれていることが判明した。また、講義資料は文字ばかりでこれを元にC言語を学んでいくことは、非常に大変であると結論づけた。そのため、C言語の苦手分野をアニメーションで教育するというアイデアを提案し、このアイデアを取り入れたWeb教材を企画することにした。Web教材にした理由としては、未来大学の1年生の所有するパソコンのOSは、MacとWindowsの2種類で混在していたため、あらゆるOSのパソコンであっても、問題なく動くようにするためである。

( 文責: 矢吹溪悟 )

## 9.2 イベント

### 9.2.1 HAKODATE アカデミックリンク 2015

2015年11月14日に「HAKODATE アカデミックリンク 2015」(以下、アカデミックリンク)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[9]。このアカデミックリンクには未来大学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。また、このアカデミックリンクには様々な分野の方が来場し、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。教育系グループは開発しているアプリのデモとポスターを来場者の方に見せながらポスターセッションを行った。高校生からは「こんなアプリが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聞くことができた。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も試すこともできた。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グループメンバーは自信をもって発表することができた。

( 文責: 新保遥平 )

### 9.2.2 最終成果発表会

最終成果発表会は、中間発表会と同様にメンバーを前半3人、後半2人に分けて発表を行った。また、中間発表会ではデモを交えて説明しなかったため、来てくれた人に私たちのプロジェクト

を分かってもらうことができなかった。その反省をふまえ、ポスター 2 枚と開発した Web アプリケーションのデモを交えて説明を行った。ポスターの内容は、1 枚目に 1 年間のスケジュールと後期の提案物について説明し、2 枚目は前期の提案物と私たちの学びについて説明した。実際に、発表をしているとき中間発表会と比べ内容が伝わっているように感じた。また、中間発表会とアカデミックリンクでの発表したため、上手く発表することができた。この発表会でいただいた評価は、第 11 章で述べる。

( 文責: 矢吹溪悟 )

### 9.2.3 振り返り

私たちは最終成果発表会が終わった後、今年 1 年間の振り返りを行った。メンバー全員で、付箋に学びや反省、展望を書いた。書いた内容は、以下に記す。

- 学び
  - 関係のない議論をしないためには、ファシリテーターが必要であると思った
  - プロジェクトが始まる前に、その日話す内容をメンバー全員で共有することが重要だということが分かった
  - どのようにプロジェクトを進めればいいのか知ることができた
  - 自分がレビューをする立場になって、レビューをする苦労を知ることができた
- 反省
  - メンバーそれぞれの進捗は、リーダーしか把握していなかった
  - 案を出してからその案に関する問題点を考えていなかった
  - 人任せにしていることが多かった
  - 適材適所で役割分担を変えることができなかった
  - 早い段階で先生や他の人に意見をもらうべきだった
- 展望
  - 1 人で何でも抱えこまないようにする
  - 人任せにしない
  - 学んだことを積極的に実践する
  - プロジェクトを始めるときは、あせらず慎重にやっていく
  - 議論する時間を適切に管理していきたい

振り返りを行ったことで、互いの学びや反省、展望を共有することができた。

( 文責: 矢吹溪悟 )

## 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学 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

概念の説明では、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。この概念の説明の部分は 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 10.1 のように、変数とはなんなのか、変数を家に例えて説明を行った。次に変数と配列の違いについて、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も、どのような点で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。

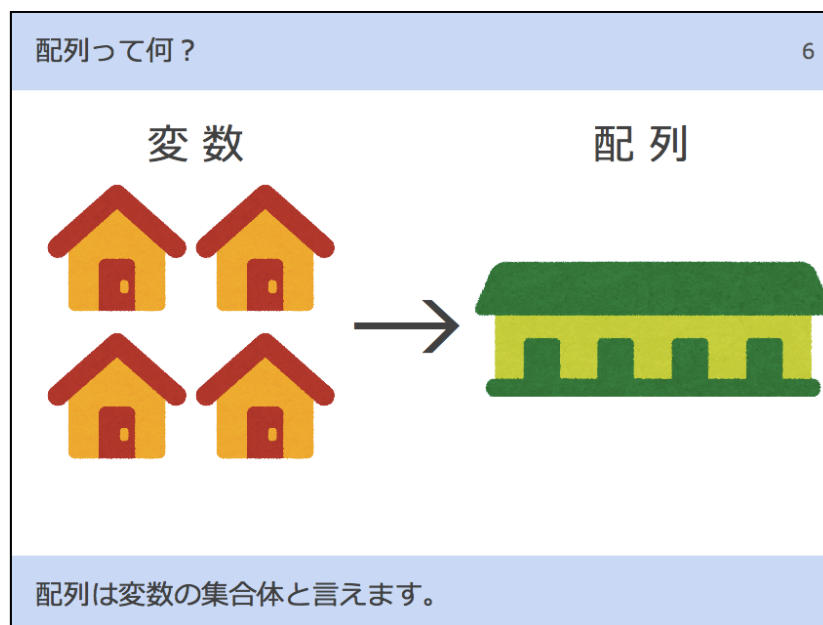


図 10.1 概念スライド

( 文責: 新保遥平 )

## 10.2.2 例題

例題の部分ではまず、図 10.2 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 10.3、図 10.4 のようにアパートとソースコードを組み合わせて説明を行う。具体的には、図 10.3 のソースコードの 1 行目では `apart[0]` には値を 2、`apart[1]` には値を 4、`apart[2]` には値を 3、`apart[3]` には値を 4 を入れ、アパートに住人が何人ずつ、住んでいるのか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]` の表示を行っている。この例題スライドでは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考えた。

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 10.5、図 10.6 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が 2 つある。1 つ目は、配列の値をランダムにしたことである。これはユーザーが何度も学習することができ、配列の理解が深まると考えたからである。2 つ目は、入力した値によってアニメーションが変化できるようにしたところである。これはユーザーが間違っ

どのようにプログラムが動くのかを確認させるためである。

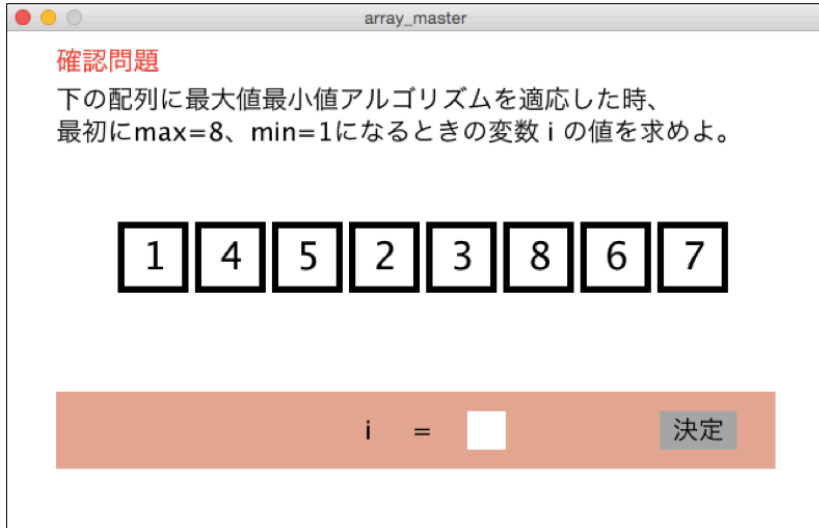


図 10.5 確認問題 入力画面 1-1

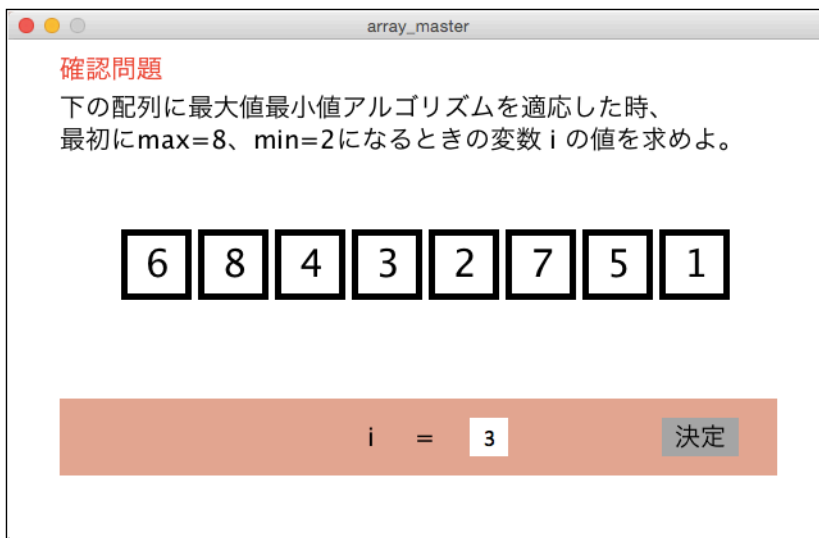


図 10.6 確認問題 入力画面 1-2

ユーザーには図 10.7、図 10.8 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学の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ふ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

その結果、図 10.9、図 10.10 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。次に入力した値が間違っている場合、このあとにプログラムがどのように動くかを説明する。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、確認問題ではユーザーが本当に配列を理解したのかを確認している。



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月に行われた最終成果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」、「円滑に説明を進めており、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、最終成果発表の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、最終成果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 [10]。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、ユーザー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ー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なることが今後の展望である。

( 文責: 自勢也 )

### 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、実装がスムーズにいかなかった。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した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ないうえを適宜確認する。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: 自勢也 )

## 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、設計をしっかりと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かったため、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: 自勢也 )

### 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の方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つ目に 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。しかし、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 )



## 第 14 章 おわりに

この 1 年間を通して私たち教育系チームは、多くの課題に直面した。特に困難だった課題が、前期に要件定義を何度もやり直したことである。この時、このプロジェクトをやめたいと全員が思った。しかし、チームの誰一人欠けずに何度も解決策を考えた。誰一人諦めなかったことが後期の成果物につながったのだと思う。当初、私たちが作りたいと思ったものと大きく異なるものになってしまったが、考えた要件定義につながる成果物を残すことができた。私たちのプロジェクトの目標は背景、課題、目的がつながる成果物を残すことだったため、達成することができてよかったと思う。そして、第 13 章で述べたようにこのプロジェクトを通して、私たちは失敗から多くの学びを得ることができた。これらの学びが私たちにとって 1 番の成果である。私たちは今回学んだことを無駄にせず、今後の研究活動や就職活動に活かしていきたい。また、学んだことをこれからプロジェクト学習を行う後輩たちに伝えていきたいと思う。

( 文責: 中進吾 )

## 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、物理的な動きをシュミ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
- 利用者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・スケージューリング・プレゼンテーション・ポートフォリオ・図解

#### 《授業内容》

- カバンのスケッチを通じた観察（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。

## “Swift” Application Development Based on Field Research

- 提案のユーザーストーリーを考えて、提案がユーザーにどのような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ィंगा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ィंगा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] ヴィストーン株式会社. 計測器プログラマー —ヴィストーン株式会社, 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)
- [7] 公立はこだて未来大学. 平成 27 年度 講義要項 平成 22 年度以降入学者用, 2015
- [8] 生活総研 ONLINE. 生活定点 1992-2014, 2014. <http://seikatsusoken.jp/teiten2014/> (2016/1/11)
- [9] HAKODATE アカデミックリンク 2015, [http://www.cc-hakodate.jp/blog/cch\\_blog/2015/10/post-44.html](http://www.cc-hakodate.jp/blog/cch_blog/2015/10/post-44.html), (2016/01/18 アクセス)
- [10] 柴田望洋、辻亮介. 新・明解 C 言語によるアルゴリズムとデータ構造. SB Creative, 2011