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/Project Leader

1013220　新保遥平　Yohei Shinpo

グループリーダ/Group Leader

1013068　中進吾　Shingo Naka

グループメンバ/Group Member

1013130　熊谷優斗　Yuto Kumagai
1013116　皀勢也　Seiya Kurokome
1013220　新保遥平　Yohei Shinpo
1013015　中進吾　Shingo Naka
1013104　矢吹渓悟　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 Nanbu

提出日

2015 年 7 月 29 日

Date of Submission

July 29, 2015

概要

日本語の概要を書く。ここに日本語の概要を書きます。

キーワード　キーワード 1, キーワード 2, キーワード 3, キーワード 4, キーワード 5

（※文責: 未来太郎）

Abstract

Abstract in English. you should write your English abstract in one page.

Keyword Keyrods1, Keyword2, Keyword3, Keyword4, Keyword5

（※文責: 函館花子）

目次

第 1 章　背景

1.1　日本のプログラミング教育について

日本では 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。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授業を行っている。また、プログラミングを学ぶのは中学 3 年生の時だけである。

（※文責: 中進吾）

1.2　現状と課題

中学校ではビジュアル言語を用いた授業を行っており、ソースコードを書く練習はしていない。また、中学校でプログラミングを学べる期間は短い。そのため中学校の授業だけでは、ソースコードを書こうとした時、どのように組んでいいかわからない。

（※文責: 中進吾）

第 2 章　本プロジェクトの目標

2.1　目的

本プロジェクトの目的は、中学校では実際にソースコードを書く練習をしていなく、中学生はソースコードを書こうとした時、どのように組んでいいか分からないため、ソースコードの組み方を学ぶゲームアプリを開発することである。

（※文責: 皀勢也）

2.2　目標

本プロジェクトの目的は、中学校では実際にソースコードを書く練習をしていなく、中学生はソースコードを書こうとした時、どのように組んでいいか分からないため、ソースコードの組み方を学ぶゲームアプリを開発することである。

（※文責: 皀勢也）

第 3 章　プロジェクトのこれまでの活動

3.1　イベント

3.1.1　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、まず子供達と触れ合い、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自分のものにしているようだった。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、その後に生きる学びであったと言える。

（※文責: 熊谷優斗）

3.1.2　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. ほげは洗い出したリスクの一部である。

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に、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 3.1　リスク分析の結果（一部抜粋）

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ながらない」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 人に連絡が行われておらず、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。

（※文責: 熊谷優斗）

3.1.3　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 TA 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 IBLabel や IBButton を用いた簡単なアプリ x 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 人で全てコーディングし、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、TA 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）

3.1.4　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 Scram という方法論を取り入れることに決まっていたため、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。バックログとは製品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. ほげは 6、7 月分のバックログの原案である。

	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 3.2　6、7 月分のバックログの原案

この原案を企業講師である高森満さんと木下実さん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. ほげは実際に使用しているバックログである。

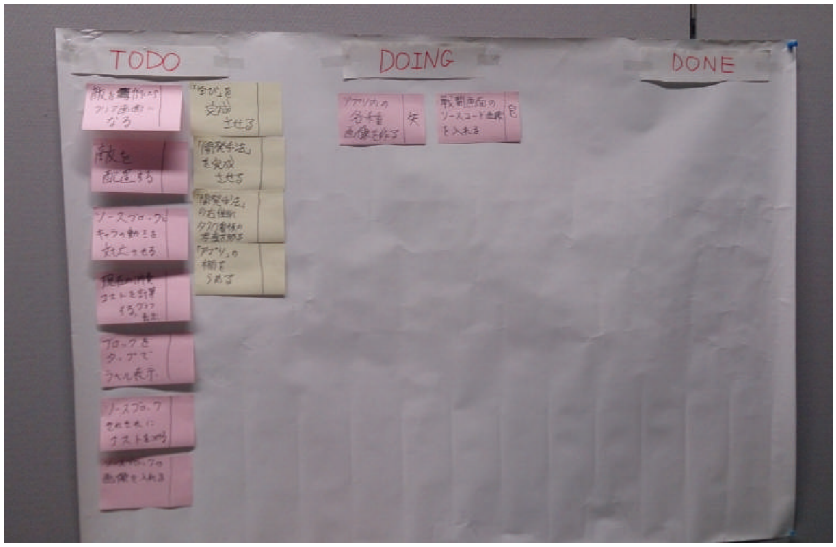

図 3.3　タスクかんばんのシステムを取り入れたバックログ

（※文責: 熊谷優斗）

3.1.5　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うまく行うことが出来なかった。そのため、ポスターを TA や担当教員見せたところ、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レビューを頂いた。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、どういった構成で文書を書いていくか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）

3.1.6　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、声が小さくて聞こえ辛い、アプリのデモを行わないと内容が伝わり辛い、というレビューを頂いた。これを受けて後半の発表では声を大きくし、デモを取り入れるように変更した。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プロダクトに魅力が少ないのではないか、という気づきを得た。

（※文責: 熊谷優斗）

3.2　アプリ案の推移

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。その推移を以下に記す。

（※文責: 熊谷優斗）

3.2.1　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. ほげに案を表す。

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 3.4　絞った 5 種類のアプリ案

5 種類それぞれの案をメンバー内で肉付けした後、TA と担当教員からレビューを頂いた。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　従来相談を乗るためには、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。

頂いたレビュー　このままだとチャットになりかねない。独自の機能が必要である。アプリの評価が難しい。

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。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。

頂いたレビュー　もし作るとしたら回答に手間がかかる、ユーザーに達成感があるものにすべき。似たようなもので何があるかは調査すべき。

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。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。

頂いたレビュー　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。目的に対しての手段を広く視野を保つべき。

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。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。

頂いたレビュー　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。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。

案 5 外遊び支援アプリ　遊びの教育。IT 化が進み、外で遊ぶことが少なくなってきている子供たちに対し、IT を活用することで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。

頂いたレビュー　開発者が楽しくてもユーザーが楽しいとは限らない。歩きスマホになりかねない。

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。

（※文責: 熊谷優斗）

3.2.2　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、ならばペルソナ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。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べることをアプリの目的とした。図 3. ほげ、ほげはアプリ内画面のイメージ図の一部である。

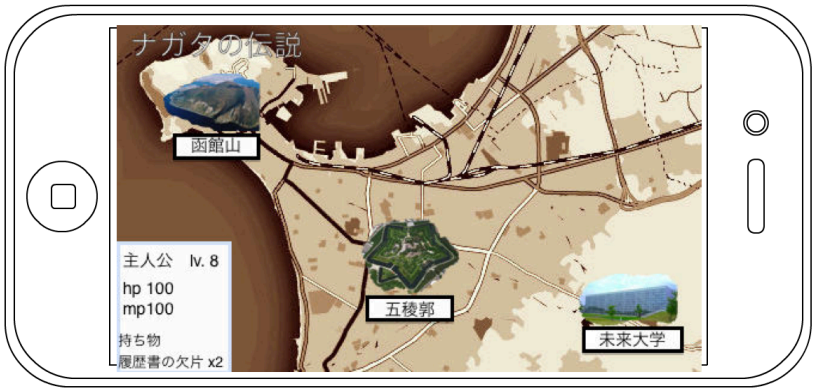

図 3.5　マップ画面

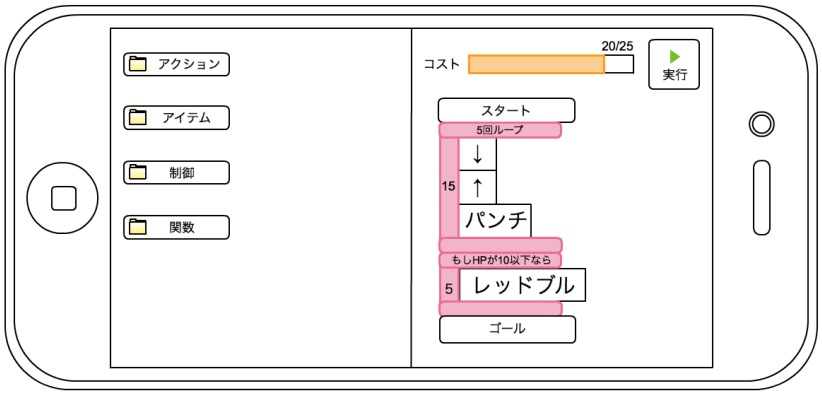

図 3.6　プログラミング画面

図 3. ほげ、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。主人公はこのブロックの通りに動くので、どのようなブロックを組めば敵を倒すことができるかを考える必要がある。更に、「コスト」というシステムを定義し、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総量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量で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量で済む。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。ということをアプリの押しの部分として定めた。

このアプリ案を TA、担当教員、企業講師の方々に見せたところ、様々なレビューを頂いた。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。
- 大学生が使うにしては、プログラミング画面の内容が低年齢向けである。
- もし AppStore 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。

（※文責: 熊谷優斗）

3.2.3 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内に、大学生向けプログラミング入門アプリは、プロジェクトとしての背景、フィールドが弱いことに気がついた。そこでメンバー内で検討した結果、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、ペルソナユーザーを変更すべきであるという結論が出た。この日の議論により生まれたのが現状の背景（第１章）であり、現状のアプリ案（第４章）である。

（※文責: 熊谷優斗）

第 4 章　開発アプリについて

4.1　概要

開発するアプリは中学生を対象としたプログラミングの組み方を学ぶゲームアプリである。マス目上のステージにある自機をプログラムを組んで動かし、敵機を倒しゴールすることでゲームがクリアとなる。

（※文責: 新保遥平）

4.1.1　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のではないかと考えた。また実際に自機をプログラム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のではないかと考えた。

（※文責: 新保遥平）

4.1.2　教育性

このアプリではユーザーがソースコードの組み立て方を学ぶことを目的にしている。これを実現するために、このアプリには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よいアルゴリズムでソースコードを組み立てることが出来たと判定し、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る。またより良いアルゴリズムでソースコードを組み立てることが出来るようになる。

（※文責: 新保遥平）

4.2　プログラミング画面

ユーザーは画面左側、図 4.1 に配置されたそれぞれのソースボタンをタップして、プログラムを組んでいく。主なソースボタンは 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。例えば下記のようにプログラムを組むことができる。

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

このようなにプログラムをタップで組むことができことが出来る。また、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。

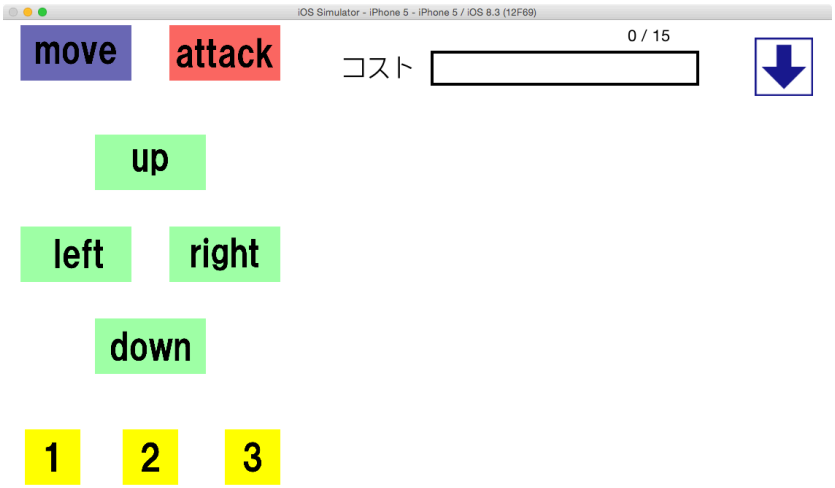

図 4.1　プログラミング画面

（※文責: 新保遥平）

4.3　戦闘画面

図 4.2 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を実際に動かすための画面である。ソースコードを入力後、戦闘画面にある実行ボタンを押すことで、実機を動かすことが出来る。1 回の実行で敵機を倒し、ゴールが出来るようなプログラムを組まなければならない。

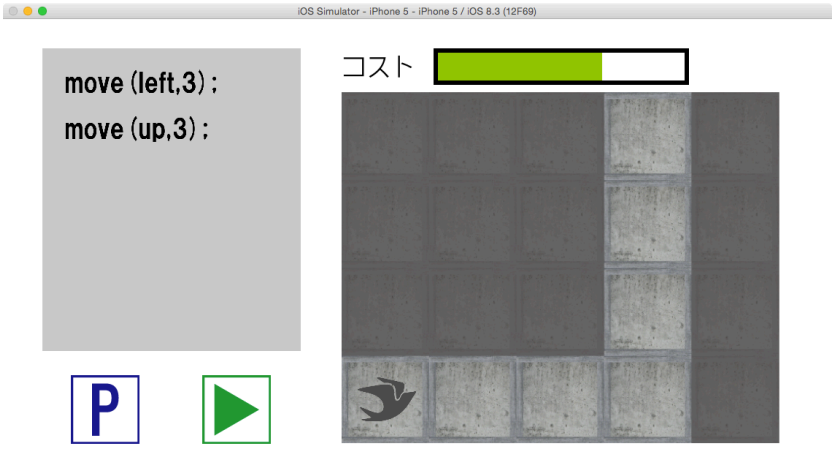

図 4.2　戦闘画面

（※文責: 新保遥平）

第 5 章　結果

5.1　プロジェクトの評価

7月に行われた中間発表会の評価シートの結果から、「声がはっきり聞こえた」、「声は大きく聞きやすかった」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?」、「まだ内容が決まっていないので評価不能」、「既存のもとの比較がない」などの意見をいただいた。これらの意見をまとめると、私たちのプロジェクトは目標が決まっていなく、内容がわかりづらいという評価であった。

（※文責: 中進吾）

5.2　プロジェクトの成果

小・中学生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、プログラミングでラジコンやロボット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。

（※文責: 中進吾）

第 6 章　まとめ

6.1　今後の課題と展望

今後は、現在のアプリ設計案を再考し、より具体的で一貫性がある設計案にしていく必要がある。また、実際にパズルゲームとしての問題を考え、それぞれの答えを用意することや実証実験や評価方法を適切に定める必要がある。

制御文のソースボタンやフィードバック機能を実装し、教育アプリとしての体裁を整え、11 月に開催されるアカデミックリンクにてワークショップを開き、そこで得たレビューを活かしてアプリの改善を行うことが今後の展望である。

（※文責: 皀勢也）

6.2　学び

要件定義を固めずに実装を行ったため、プロジェクトの目的を見失い要件定義を一から考え直すことになった。そのため、時間をかけて、要件定義をやり直すことの重要性を学んだ。また、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ことからドキュメントを残して、情報共有することの大切さを学んだ。

（※文責: 皀勢也）

付録 A　新規習得技術

付録 B　活用した講義

付録 C　相互評価

付録 D　その他製作物

参考文献

[1] 著者名. 書籍名. 出版社, 年号.
[2] ほげほげお. うんたらかんたら, 2003.