

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 Nanbu

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 iOS アプリを開発することを目的としている。

プロジェクトの当初は、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 つに絞り込んだ。テーマは、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 から考え直すことになった。要件定義を考え直すことは、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まった。

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。

今後は、アプリの設計をやり直しアプリを実装していくことを考えている。また、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もとにアプリを改善していくことを考えている。

(文責: 中進吾)

Abstract

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

Each member considered the application of educating and presented a members and teachers on the first of a project. We shared information between the members and received reviews from teachers. After that 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 a 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 didn't finish changing the requirement definition and went many times. As a result, it was decided in a theme as the game application that made the person who learned a programming at junior high school and an interested person the subject.

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 aren't 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's necessary to redo a design of an application.

We are thinking the design of application is being redone from now on and an application is being mounted. A workshop will be opened in the academic link held in November and we get the review and are thinking an application is being improved.

(文責: 中進吾)

目次

第 1 章	背景	1
1.1	日本のプログラミング教育について	1
1.2	現状と課題	1
第 2 章	本プロジェクトの目標	2
2.1	目的	2
2.2	目標	2
第 3 章	プロジェクトのこれまでの活動	3
3.1	イベント	3
3.1.1	スクラッチワークショップへの参加	3
3.1.2	リスク分析	3
3.1.3	アプリ開発のための勉強会	4
3.1.4	バックログの作成	5
3.1.5	中間発表会の資料制作	6
3.1.6	中間発表会	6
3.2	アプリ案の推移	6
3.2.1	アプリ案の検討	6
3.2.2	大学生向けプログラミング入門アプリ	8
3.2.3	中学生向けプログラミング支援アプリ	9
第 4 章	開発アプリについて	10
4.1	概要	10
4.2	プログラミング画面	10
4.3	戦闘画面	13
4.3.1	ゲーム性	13
4.3.2	教育性	13
第 5 章	結果	15
5.1	プロジェクトの評価	15
5.2	プロジェクトの成果	15
第 6 章	まとめ	16
6.1	今後の課題と展望	16
6.2	学び	16
付録 A	新規習得技術	17
付録 B	活用した講義	18

付録 C	相互評価	19
付録 D	その他製作物	20
参考文献		21

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ことである。

(文責: 梶勢也)

2.2 目標

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、本プロジェクトの目標である。

(文責: 梶勢也)

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い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 3.1 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 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、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 Scrum 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 3.2 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の学習のための資料調査		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。

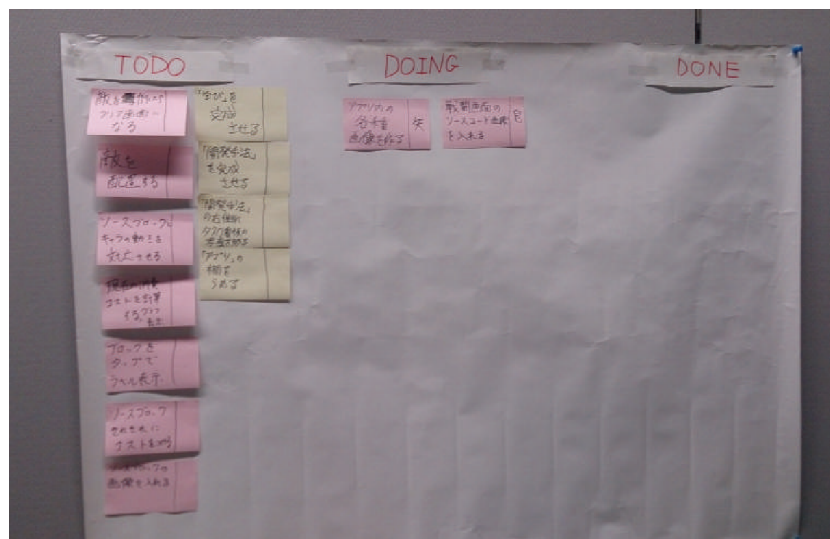


図 3.3 タスクかんばんのシステムを取り入れたバックログ

(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上手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、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)

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 3.4 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。
レビュー内容 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である。似たようなもので何があるかは調査すべきである。

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、全ユーザーと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。
レビュー内容 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。目的に対しての手段を広く視野を保つべきである。

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。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 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。
レビュー内容 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3.5と図3.6はアプリ内画面のイメージ図の一部である。

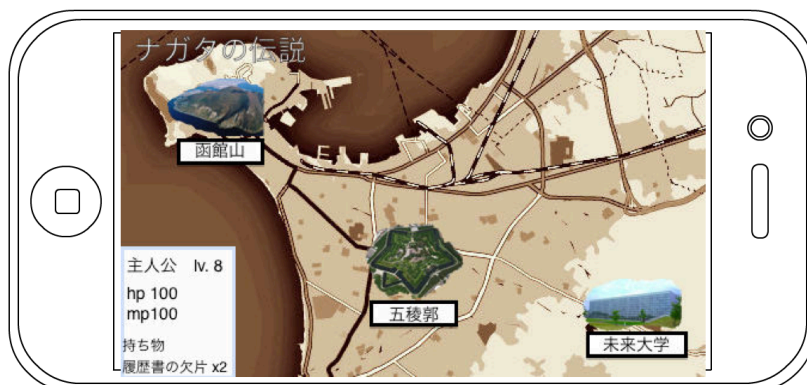


図 3.5 マップ画面

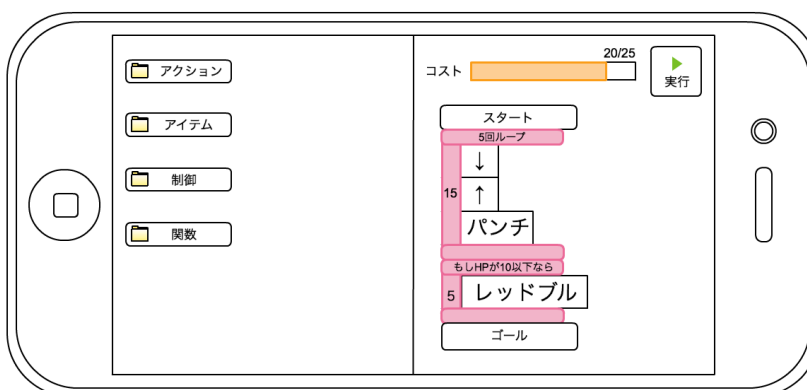


図 3.6 プログラミング画面

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。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。
- 大学生が使うにとしては、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。

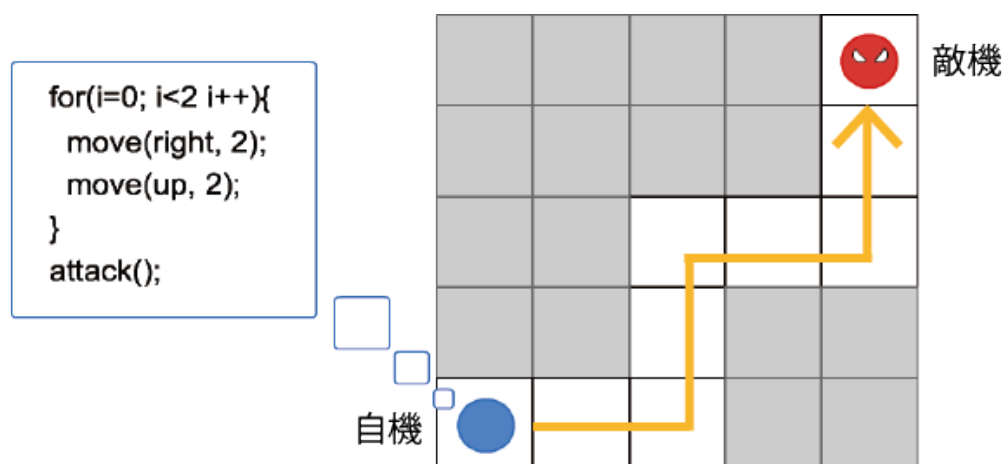


図 4.1 ゲーム概要

(文責: 新保遥平)

4.2 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 4.2 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。

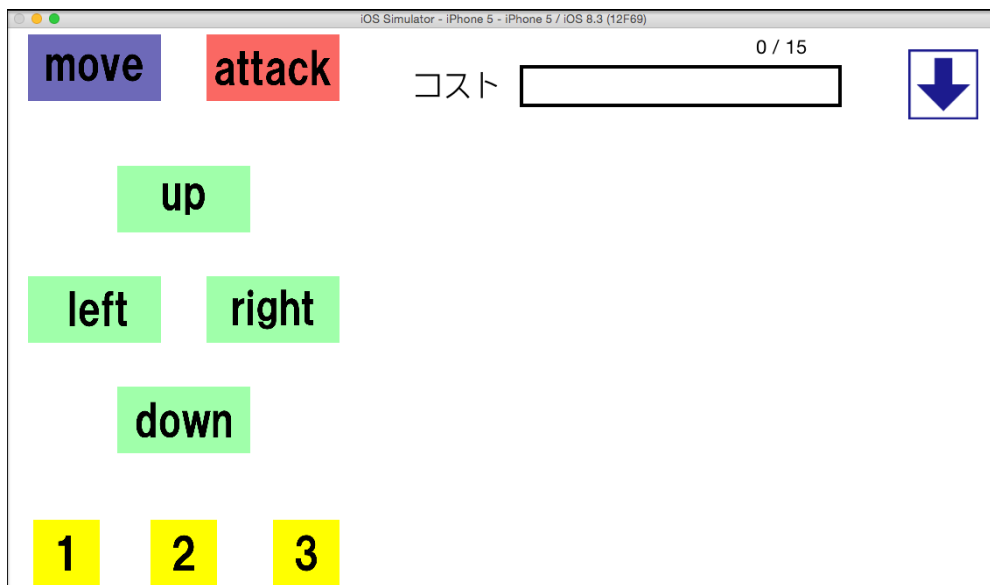


図 4.3 プログラミング画面

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。

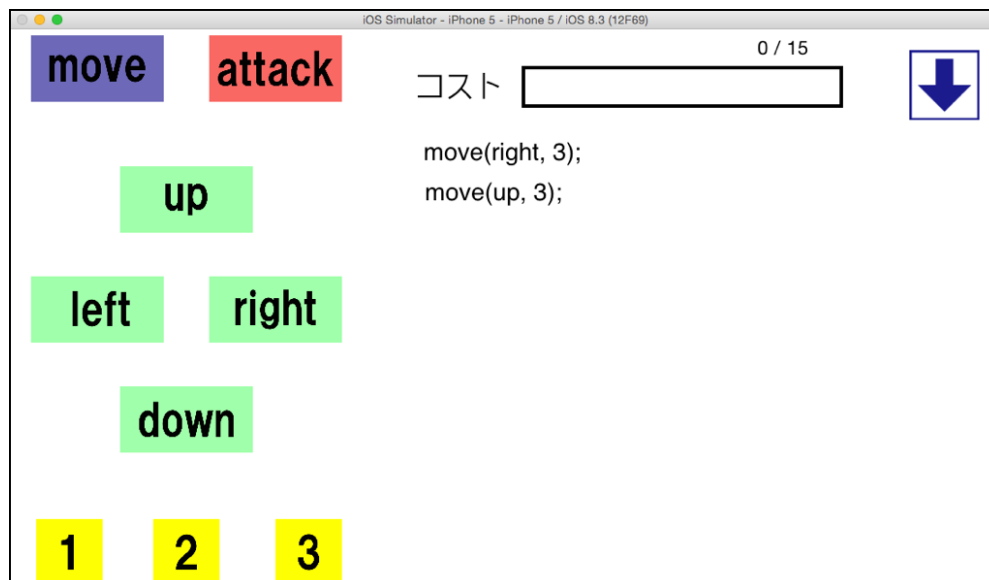


図 4.4 ソースコード入力後のプログラミング画面

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。

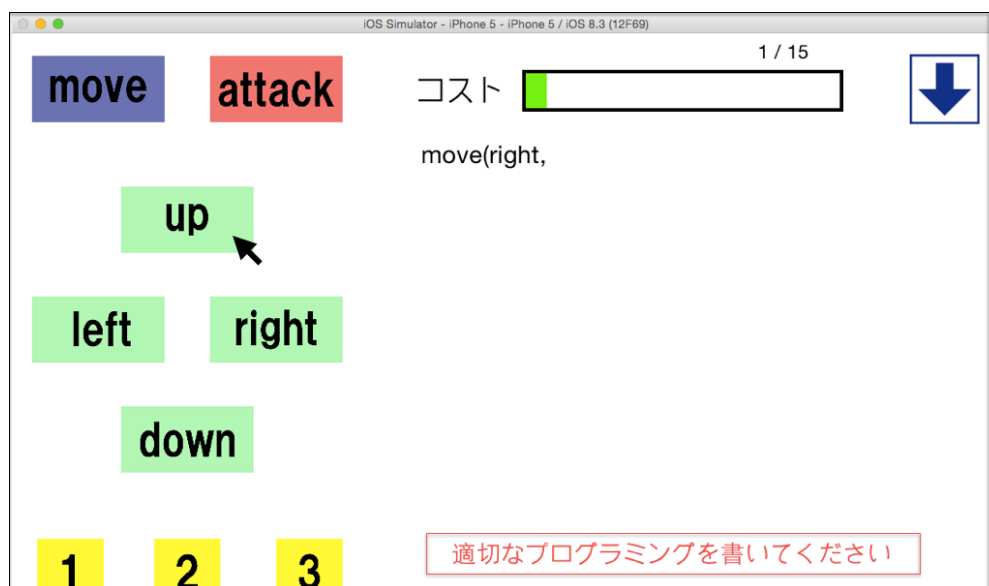


図 4.5 ソースコードがエラー時のプログラミング画面

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)

4.3 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。

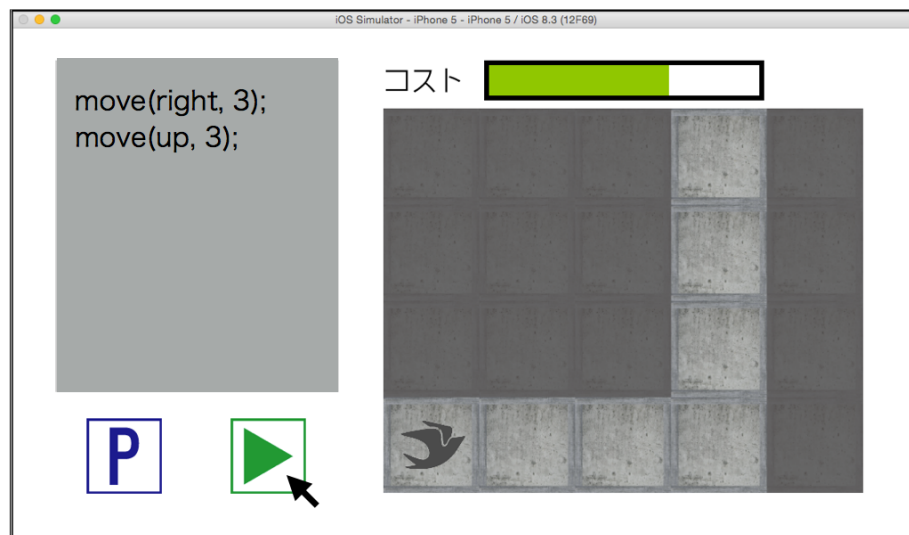


図 4.6 戦闘画面

(文責: 新保遥平)

4.3.1 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)

4.3.2 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る。また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。

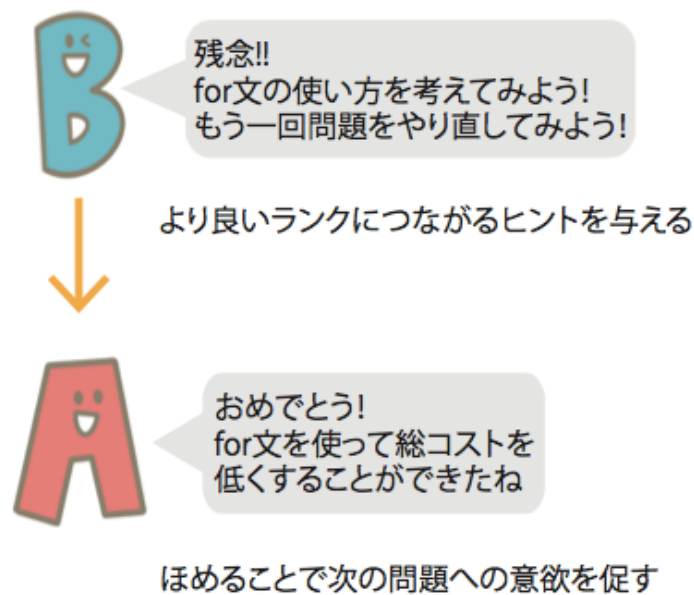


図 4.7 ランクとヒント

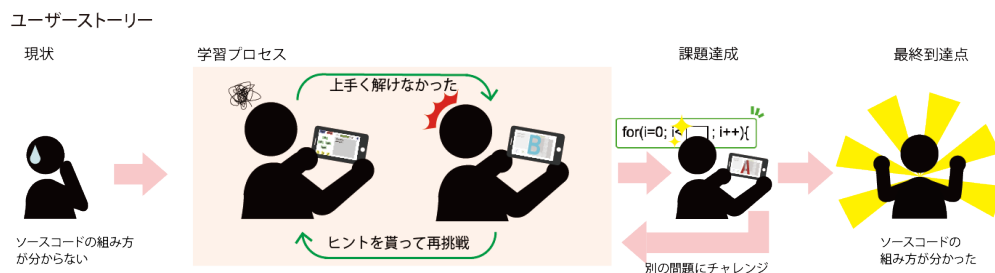


図 4.8 ユーザーストーリー

(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、「声をはっきり聞こえた」、「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」、「既存のものと比較がない」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、実際にゲームの内容を考え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: 梶勢也)

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、要件定義を行う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: 梶勢也)

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