

## 実装形態の差異に着目した大学生対象スマートフォンアプリ開発の試行的実践

## Trial Practice of Programming Workshop of Smartphone Application Development for University Students Focusing on Differences Between GUI and CUI Programming in Implementation Targets

福井昌則\* 青木 徹\*\* 萩倉 丈\*\*\* 佐々木 雄 司\*\*\*\*  
 FUKUI Masanori AOKI Toru HAGIKURA Jo SASAKI Yuji  
 黒田昌克\*\*\*\*\* 平嶋 宗\*\*\*\*\* 森山 潤\*\*\*\*\*  
 KURODA Masakatsu HIRASHIMA Tsukasa MORIYAMA Jun

本稿は、実装形態の差異に着目した、数学的ゲーム・パズルを題材とする大学生対象スマートフォンアプリ開発の試行的実践について報告することを目的としている。大学生を対象とし、数学的ゲーム・パズルの「12-9（偶数引く奇数）」、「セブンアップ」を用いて、iPhone で実装した場合（GUI ベースのプログラミング）と他のプログラミング言語で実装した場合（CUI ベースのプログラミング）の差異を考えさせるような iPhone アプリケーション開発の試行的実践を行なった。その結果、iPhone プログラミング未経験者のみならず、プログラミング初学者である受講生全員が実装することができ、それまでの実践と同様に主体的にプログラミングに取り組ませることができた。以上のことから、本稿の有用性が示唆された。

キーワード：プログラミング教育, iOS, 数学的ゲーム・パズル, Swift, リフレクション

## 1. はじめに

本稿は、実装形態の差異に着目した、数学的ゲーム・パズルを題材とする大学生対象スマートフォンアプリ開発の実践について報告することを目的としている。

現在我が国は、人工知能の台頭による 2045年問題（技術的特異点, シンギュラリティ）<sup>1), 2)</sup> や、加速する高度情報化社会への対応と対策, 第4次産業革命の本格的な到来への対応, 少子高齢化時代における労働生産性の向上など様々な課題を抱えており、今まで通用してきた対策や対応では解決できない問題が山積となっている。そして、上述の問題を解決できるような人材の育成が急務となっている。厚生労働省によれば、我が国の総人口は、2010年の1億2,806万人から長期人口減少過程に入り、2030年の1億1,662万人を経て、2048年には1億人を割って9,913万人、50年後の2060年には8,674万人になることが見込まれると報告されており<sup>3)</sup>、少子化時代において社会を支えるために、労働生産性をあげていくことが求められている。日本生産性本部は、我が国の労働生産性は、加盟35カ国中18位で加盟国平均を下回って低迷が続いていると指摘しており<sup>4)</sup>、人工知能やITなどを積極的に活用することや、労働生産性が高いイノベティブな活動を展開する必要がある。

一方、経済産業省によれば、2020年には36.9万人、2030年には78.9万人のIT人材が不足すると指摘されており<sup>5)</sup>、ITスキルを有する人材育成が急務となっている

現状が伺える。ここで述べられているIT人材は、プログラミングやシステム開発を専門に行う人材だけではなく、情報システムやIT活用を行う人材も指しており、ITスキルを有する人材育成は喫緊の問題である。

その現状において、義務教育段階では2020年から次期学習指導要領が完全実施されることとなった。その中では、現行の学習指導要領と比して様々な変更点が見られるが、義務教育段階におけるプログラミング教育の必修化、高等学校段階におけるプログラミング教育の拡充など、プログラミング教育の充実化が図られることとなった<sup>6), 7)</sup>。大学教育においては、プログラミングを推進して行くことに対して明記されていないが、未来投資会議には、IT・データスキル人材育成のために、大学等の制度改革を通じた大学教育の質の向上やイノベーションの推進などが記載されており<sup>8)</sup>、ITスキルの重要性が指摘されている。

ITスキルを高めるために、知識を身につけるだけではなく、プログラミング教育を推進していくことが重要な方略の一つであると考えられ、多くの生徒・学生が単にコーディングできるようになるだけではなく、ITスキルを活用して労働生産性を高め、イノベティブな活動を展開する必要がある。

現在、プログラミング教育の方向性は様々なものがある。例えばScratchの実践<sup>9)</sup>といったブロック型プログラミングを用いた入門教育、Excel VBAを用いたプロ

\*広島大学大学院生 \*\*立命館大学大学院生 \*\*\*関西学院大学学部生 \*\*\*\*慶應義塾大学学部生

平成30年7月11日受理

\*\*\*\*\*兵庫教育大学大学院博士課程教科教育実践学専攻生活・健康系教育連合講座

\*\*\*\*\*広島大学工学研究科

\*\*\*\*\*兵庫教育大学大学院教科教育実践開発専攻生活・健康・情報系教育コース, 教育実践高度化専攻授業実践開発コース 教授

プログラミング教育の実践, ロボットを動かすプログラムを組むもの, Java や C 言語を用いた本格的なプログラミングなど, 様々なアプローチからプログラミング教育が行われている<sup>10)</sup>. つまり, 実践の形態や実践対象者, どのプログラミング言語で行うか, 扱う題材や素材などが多様であることは, 筆者らが言うまでもない. その中で, スマートフォンアプリ開発は, 最も注目を集めているプログラミングの一つである. 様々なスクールが開講されており, iPhone や Android に実装することを目的としたプログラミング講座が多数開講されている. また, いくつかの実践事例が報告されている<sup>11)</sup>. スマートフォンに実装するようなプログラミングは, 一般的に GUI (Graphical User Interface) プログラミング (以下 GUI と略記) と呼ばれており, 視覚的にわかりやすくすることや, アニメーションなどを用いてユーザを惹きつけることができる. その一方で, グラフィカルな要素にこだわることは大事でありながらも, そこに時間を取られることで, プログラミングの学習がおろそかになってしまう可能性があり, そのバランスをどのようにすればよいかについては, 明確な指針はなく, どのようにするかについては千差万別である. 一方, GUI に対し, いわゆるコマンドベースでプログラミングを行う CUI (Character User Interface) プログラミング (以下 CUI と略記) があるが, そのいずれを用いるのか, プログラミング導入段階において難しい選択であり明確な回答はない. 飯尾は, プログラミングの導入において GUI と CUI のいずれが望ましいかについての意識調査を, IT に携わる専門家群と, 受講側の学生群に対して調査研究を行なった結果, 専門家群の意見は CUI 派と GUI 派に二分されたこと, 学生群は信頼性の高い回答を集めることができなかったと報告しており<sup>12)</sup>, 今後より詳細に検討を行う必要があると考えられる.

しかし, GUI, CUI のいずれかという二者択一ではなく, 同じ題材を GUI で実装する場合と, CUI で実装する場合を比較することで, プログラミングに対してより俯瞰的な見方ができる可能性がある. 例えば, 平嶋・福井・林は, 実装するプログラムは同じでありながらも, 違ったリソースを準備することで, 学習者にリフレクションをさせることができ, プログラミング的思考を育成できることを指摘しており<sup>13)</sup>, GUI と CUI では, 同じ題材であってもどうプログラミングが異なるかについて受講者に考えさせることで, よりプログラミングについて深い理解を促し, 設計の重要性などを感じさせることが期待できる. しかし, 筆者らの知る限りにおいて, 先行研究を見かけることはない.

一昨年から, 筆者の 1 人は継続して iPhone アプリ開発講座 (90分×3 回の全 5 日間) を大学で担当しているが, その実践では, 参加者に自身で何を作るかをある制約の中で企画・立案させ, それを発表・実装するという活動を行なっている<sup>14), 15)</sup>. 本稿は, 上述の講座の内容を援用しながら, GUI と CUI で同じ数学的ゲーム・パズルをどうプログラミングするかについて受講者に考えさ

せることを取り入れた活動を試みる.

## 2. 実践のデザイン

### 2.1. 学習過程の設定

総務省は, プログラミング人材育成の観点として, 「(a) スキル習得型」, 「(b) モチベーション向上型」, 「(c) 課題設定・解決力育成型」の 3 つの方略をあげている. この中で「(a) スキル習得型」は, 基礎概念を習得し, 全体構成と実践を行い, そして, オリジナルアプリの開発の 3 ステップにより指導する学習過程である. その学習過程においては, Java や Objective-C, HTML5 といったテキスト言語を用いることが想定されている. また, 「(c) 課題設定・解決力育成型」では, 作りたい作品や目標を最初に具体化し, 自身でどのようなプログラムを作るかを考えた上で, どうやってプログラミングを作るかを考えること, そして自身で考えて成果を伝える力の育成の重要性を指摘している<sup>16)</sup>. 弓野は, プレゼンテーションといった表現的な創造性活動の有用性について指摘している<sup>17)</sup>. 以上のことから, 受講生が自身で考えた内容を自身で構成してまとめさせる. そして, 他の受講生に対してプレゼンテーションさせ, アプリとして実装させる活動を展開する.

### 2.2. 題材の設定

#### 2.2.1. iPhone アプリ開発の活用

スマートフォンには iPhone 以外にも Android, Windows Phone などが存在するが, 本稿では, iPhone を対象としたアプリ開発を行う. iPhone アプリ開発を行える環境はいくつか存在し, 例えば Xcode, AppCode<sup>18)</sup>, Unity<sup>19)</sup>, Xamarin<sup>20)</sup>, PhoneGap<sup>21)</sup> などがあるが, その中で今回は Apple が提供している Xcode を用いる (2018年 7 月 2 日現在, 最新バージョンは 9.4.1). Xcode は, Interface Builder を用いることで, 画面を視覚的に設計することができ, ソースコード側との連携も容易に行うことができる (Fig.1). また, ビュー間の遷移なども容易に実装することができる. Xcode7.x 以降から開発したアプリを無料で自身の iPhone や iPad などに転送できるようになり, より開発を行いやすくなった. 開発言語は, Swift を用いる. Swift は 2014 年に Apple が発表した新しいプログラミング言語で, それまで用いられていた Objective-C に比べ, モダンな文法を採用して

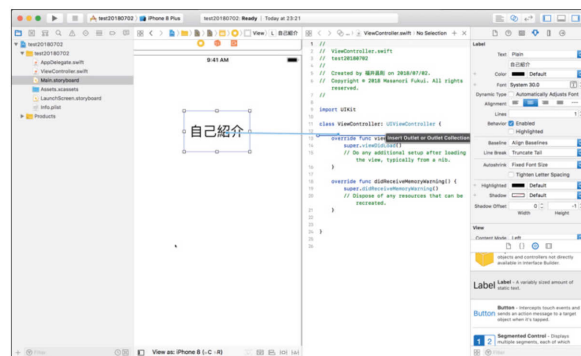


Fig.1 Interface Builder を用いた操作の様子



いる（例えば繰り返しは拡張 for 文に似た文法を採用していることや、文末に「;」を必要としないなど）。現在最新バージョンは4.2であり、積極的に言語の改善が行われている。以前のバージョンでは、バージョンアップのたびに言語仕様が変わるなど、初学者には対応が困難な部分が多く、指導も行いづらい側面があったが、現在はそのようなことも少なくなっており、初学者対象のプログラミングにも扱いやすくなってきている。

アルゴリズムの学習には、Playground<sup>22)</sup> を用いる (Fig.2)。Playground は Xcode に付属している、基本的なプログラミングを試すことができる環境であり、iPad などでも提供されている。コードを入力すると、すぐにコンパイルされ右側に結果がすぐに表示される。この機能により、インタラクティブに結果を確認することができ、アルゴリズムの学習に有用である。しかし、Playground がマシンによってはコンパイルに時間がかかることもあるため、オンラインコンパイラサービスなどの別の方法を準備しておくことも有用であると考えられる。

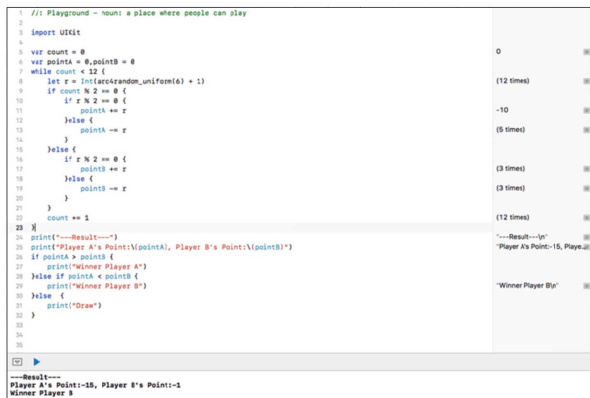


Fig.2 Playground を用いた操作の様子

### 2.2.2. 自己紹介, 数学的ゲーム・パズルの活用

本稿では、自己紹介アプリ作成および数学的ゲーム・パズルを題材とするアプリ作成を題材とする。筆者らは、前報において自己紹介アプリと丁半博打ゲームを用いた iPhone アプリ開発の実践を行ない、その有用性について報告している<sup>14, 15)</sup>。また、数学的ゲーム・パズルの題材として、サイコロやトランプを用いた実践について報告しており、その内容を援用する。以下、その内容について述べる。

#### (1) 自己紹介アプリの活用

自己紹介アプリは、複雑なプログラミングやアルゴリズムを用いずとも、画面を遷移させることおよびボタンをタップすることでアクションが発生することを理解すれば十分に開発することができる。一般的にスマートフォンアプリでは、一つの画面に多くの情報を入れすぎたりすると、画面の大きさや操作性の制約からユーザが混乱しやすく、操作に関して新たな学習を強いる結果となってしまう。このことから、画面に多くの情報を入れずに自分自身のことをわかりやすく、そして適切に伝えるために必要な設計の題材としても活用する。

#### (2) 数学的ゲーム・パズルの活用

自己紹介アプリで基本的な開発に慣れた後、題材として数学的ゲーム・パズルを用いる（本稿では数学的ゲーム・パズルを「必勝法が存在する、もしくは確率的要素を有するゲーム・パズル」と定義する）。山上は、「数独パズル」を用いた Excel VBA プログラミングを行い、その有用性について報告している<sup>23)</sup>。また筆者らは、数学的ゲームを用いたプログラミングの試行的実践を行い、その有用性について報告しており<sup>24, 25, 26)</sup>、その内容を援用する。数学的ゲーム・パズルは特徴として、①問題がより良く構造化されておりモデル化しやすいこと、②変数の数学的な処理によって解決方法を探究するためアルゴリズム的な思考を促しやすいこと、③ゲームの必勝法の探究など、興味・関心を高めやすいこと、④多くの予備知識を必要とせず、受講生が取り組みやすいこと、⑤具体的なプレイを通して、受講生間および受講生と教員とのコミュニケーションが促進できること、⑥変形・改良などが容易で、新しいゲーム・パズルを創出しやすいこと、などといった特徴を有しており<sup>27)</sup>、自身で画面遷移の設計および適切なアルゴリズムの選択を行わせ、それを実装させることが期待できる。

また、自身で作成した問題に対して興味を持ってもらい、完成させることを目指す。そして、具体的なプレイを通してアルゴリズムを考えて説明し実装することで、基本的なアルゴリズムの学習を促進することを目指す。数学的ゲーム・パズルには様々なものがあるが、「12-9 (偶数引く奇数)」と「セブンアップ」を用いる<sup>28)</sup>。その内容を Table.1 に示す。

Table.1 用いる数学的ゲーム・パズル

タイトル	内容
12-9 (偶数引く奇数)	2人対戦ゲームであり、サイコロの目が偶数のとき、その目を加算、サイコロの目が奇数のとき、その目を減算する。お互いが1回ずつ6個のサイコロを振って、点数が大きい方が勝ちとなるゲーム
セブンアップ	2人対戦ゲームであり、最初にお互いチップを21枚ずつ持っている。交互に2つサイコロを振り、出た目の合計値が7のとき、チップが7枚増える。出た目の合計値が7以外のとき、7との差の枚数のチップを捨てる。それらのプレイを繰り返し、先にチップがなくなった方が負けとなるゲーム

Table.1に示した数学的ゲーム・パズルを、iPhone 上でのように実装するかについて画面の設計をさせ、それに必要なアルゴリズムを、具体的なプレイを通して考えさせる活動を行う。以上のことから、本稿では、自己紹介と数学的ゲーム・パズル「12-9 (偶数引く奇数)」 「セブンアップ」を題材とした iPhone プログラミングの実践を展開する。

### 2.3. 学習内容

学習内容を設定する上で、本稿では基礎的なアルゴリズムを網羅することとした。iPhone プログラミングを行う際に、受講生はどうしても見た目のインパクトや動きを求める傾向が強いことが、2年間の実践を通して強く感じられたが、プログラミングにおいて必要な基本を身につけることも、興味・関心を高めることとともに、

重要である。以上のことから、分岐、反復、ネスト構造、配列（データ構造）を学習内容とし、それらの学習を段階的に進めていくこととした。学習内容について Table.2 に示す。

Table.2 学習内容

学習内容	
第1回	SwiftPlaygroundsで変数(var,let), 分岐, 反復, 乱数 Xcodeの使い方・操作方法, 自己紹介アプリの設計 UILabel, UIButton, UIImageView, 画面遷移の基礎
第2回	反復, 分岐, 配列, 乱数, 実機転送 自己紹介アプリの発表, 12-9(偶数引く奇数)の設計・実装
第3回	ネスト構造, 画面間の値の受け渡し, 複雑な画面遷移 12-9(偶数引く奇数)の発表, セブンアップの設計
第4回	アルゴリズムの総復習, セブンアップの実装・発表 まとめ

また本稿では、iPhone プログラミングと他のプログラミングでの実装における差異を感じさせる活動を行う。具体的には、iPhone に実装する場合（GUI、1つの画面で最初から最後までゲームを行うという実装や、複数画面を遷移しながらプレイを行うなどのプログラム）とそうでない場合（CUI、画面遷移などがなく、反復や分岐を用いて結果が出力されるようなプログラム）で、実装する数学的ゲーム・パズルは同じであっても、実装する対象によってどのように実装が異なるかということを考えさせ、iPhone プログラミングと他のプログラミングとの比較によって、アルゴリズムや全体の構成がどのように異なるかについて考えさせる活動を行う。その際、従来の反復などを用いた実装方法との差異を意識させることで、iPhone プログラミングを行う有用性が高まる可能性がある。以上の活動により、アルゴリズムやプログラミングの理解をより深める可能性があると考えられる。

## 2.4. 実践内容

以上の題材、学習内容を踏まえて、アルゴリズムの学習と自身で画面設計、アルゴリズムの選択などを行わせ、それを発表・実装させる。その際、ワークシートを活用する。それと並行して、同じ題材を GUI と CUI で実装する場合の差異について考えさせ、複数の観点から同一の数学的ゲーム・パズルについて考えさせる。そのような実践を展開することで、全体を俯瞰できるような力を育成することが期待できる。

## 2.5. 評価の手続き

本稿の第1日目、各回の授業後に調査票を用いたアンケート調査を実施した。

事前調査の項目として、将来どのようにプログラミングに関わりたいかを把握するために項目 A、プログラミングに対する意識を把握するための項目 B を準備した。項目 A は択一式で回答を求め、項目 B は「4: とても 3: まあまあ 2: あまり 1: まったく」の4件法で回答を求めた。

## 項目 A

将来的に、どのような形でプログラミングに関わりたいですか？

1. プログラマとして
2. 趣味として
3. 特に開発はしないが、よりアプリなどを活用するため
4. 特になし

## 項目 B

- 1) プログラミングに対するイメージは良い。
- 2) プログラミングを学ぶことは自分の役に立つと思う。
- 3) プログラミングを学ぶことは不安に感じる。
- 4) プログラミングの学び方を学びたいと思う。
- 5) プログラミングの文法や手法を身につけられる自信がある。
- 6) 本講座中の演習課題が自分でできる自信がある。
- 7) プログラミングの力を生活の中で活かす自信がある。

また、各回の講義後に、講義に対する理解度や満足度を把握する項目 C を準備した。

## 項目 C

- 1) プログラミングに興味・関心はありますか。
- 2) 今日の講義はよくわかった
- 3) 今日の講義は面白かった。
- 4) 今日の講義は興味深い内容だった。
- 5) 今日の講義は役に立った。
- 6) 今日の講義で自信を持てた。
- 7) 今日の講義は満足だ。
- 8) 自分なりに工夫できた。

さらに、森山らが作成した初学者のプログラミングにおける思考過程の自覚程度を把握する23項目<sup>29)</sup>、講義に関する感想などを質問する項目（自由記述）を準備した。初学者のプログラミングにおける思考過程の自覚程度を把握する項目は、「問題理解過程因子」「要素分解・統合過程因子」「局部的自己点検過程因子」「大局的自己点検過程因子」から構成されており、その尺度を用いて、プログラミングを全体的に見渡す力を把握することとした。

## 3. 研究方法

### 3.1. 対象

- ・場所：関西圏にある私立 R 大学
- ・人数：6 名（全員男性）、理系 4 名、文系 2 名
- ・実施時期および授業時間：1 日あたり 90 分×3 コマ、それを 2018 年 5 月末から 4 日間
- ・単位などはない課外授業の形で受講生を募集した。受講生の所属、プログラミング経験について、Table.3 に示す。

Table.3 受講生の所属とプログラミング経験

学部	学年	プログラミング経験
理工	1	なし
情報理工	1	Processing(大学の授業)、 Androidアプリ開発入門(スクール)
情報理工	2	C言語(大学の授業)
情報理工	2	C言語、Java(大学の授業)
経済	4	なし
スポーツ健康科学	4	なし



### 3.2. 実践内容

- ・システム要件：OSX 10.12 (Sierra) もしくは10.13 (High Sierra), Xcode 9.3以降, 言語はSwift4.x を利用.
- ・前章で述べた自己紹介・数学的ゲーム・パズルを題材とした授業を行う.
- ・第1回目の最初, および各回の授業後にアンケートによる調査を行う.

## 4. 結果と考察

### 4.1. 事前調査の結果

事前調査の結果, aの「プログラマとして」が5名, bの「趣味として」が1名であった. また, 項目Bはそれぞれ, 1) 平均3.83 (S.D.0.37), 2) 平均3.83 (S.D.0.37), 3) 平均2.17 (S.D.0.69), 4) 平均3.33 (S.D.0.75), 5) 平均3.17 (S.D.0.37), 6) 平均2.83 (S.D.0.37), 7) 平均3.17 (S.D.0.37) であった. 以上のことから, 基本的に全ての項目が高い平均値を示したものの, 4)「プログラミングの学び方を学びたいと思う」の項目については, 他の項目より相対的に平均値が低く, S.D.も高かったことから, 受講生は将来的にプログラミングに関わりたい受講生が多いながらも, プログラミングに対しての学び方を学びたいという意識にはばらつきがある実態が把握された.

### 4.2. 第1日目 (基本的操作の習得および自己紹介アプリの設計・開発)

本講座で初めて本格的にMacを使う受講生が4名いたため, まず最初にMacの操作方法について簡単に説明を行なった後, Xcodeの基本的な操作方法を指導した. Macはトラックパッドの使い方に特徴があるが, それを使いこなせることで大幅に操作が便利になることに, 受講生は有用性を強く感じていた. そして, iPhoneアプリ開発を本格的に開始する前に, 基本的なアルゴリズムを学習することとした. そのためにPlaygroundを用い, 変数 (var, let), 分岐, 反復を最初に指導した. しかし, Playgroundがマシンによってはコンパイルに時間がかかることもあったため, オンラインコンパイラサービスのpaiza.ioなどでも実行できることを指導した.

その後テキストを併用しながら<sup>28, 29)</sup>, Xcodeの基本的な使い方として, UILabelとUIButton, UIImage, UIImageViewについて指導を行ない, storyboardとの関連付けを行う方法について説明を行なった. Fig.1のように, 紐をむすぶように関連付けができることに, 受講生は皆大いに感動していた. そして, UIButtonに関しては, ボタンをタップするとアクションが生じるというイベントドリブンについて説明を行い, それが他のプログラミングとどのように異なるかについて説明を行なった.

午後からは, 自己紹介アプリを作成するために必要な画面遷移 (modal) について指導した. 画面遷移を用いることで, 一気にアプリらしい挙動となったことに, 受

講生は喜んでいて. その後, ワークシートを配布し, 画面設計を行わせた. この時点では, 基本的な分岐や反復しか学習していなかったが, 自己紹介アプリの作成にはそのような知識がほとんどなくても, 画面遷移を用いることでアプリを完成させることができる. そして, 多くの人が見てぱっとわかるようなユーザインターフェイス (UI) となるように指導した.

授業後の自由記述では, 感想として「1人でやってたら解決できないことも解決できてよかった」「IT企業に就職が決まっているのですが, そこで使えるスキルになるように頑張ります」「普段学校でやっているプログラミングよりずっとわかりやすかった」「必ずマスターしてみせます」などといったコメントが得られた.

### 4.3. 第2日目 (自己紹介アプリの発表およびオブジェクト指向の基礎, 数学的ゲーム・パズル実装に向けた導入)

午前中に, 前回の内容について簡単な復習を行なった後, 実機転送について指導した. 実機転送は, Apple IDがあれば転送できるアプリ数に制限があるものの, 無料で行うことができる. 実機転送を指導した後, iPhoneを所持している受講生には転送を行わせた. 全員がiPhoneユーザであり, 自身で作成したアプリがスムーズに動く様子に全員が大きな歓声をあげていた. また, 「みんなに自慢します」と言っている受講生もいた. その後, 完成させた自己紹介アプリについて発表させた. その様子を Fig.3に示す. 受講生のプレゼンテーションに対する抵抗感は少なく, 自身で考えて実装した自己紹介アプリを自信を持って紹介する様子が伺えた. また, 受講生同士のやりとりも活発に行われ, 和気藹々とした雰囲気を作ることもできた.



Fig.3 受講生による自己紹介アプリの発表

次に, 画面間での値の受け渡し, 任意の画面に遷移すること, 反復や分岐のネスト構造について指導した. その後, 前回と今回で作成した自己紹介アプリを, 画面遷移などを用いず, 他の言語ではどのように実装するかについて説明を行い, 今回実装したアプリとの差異を考えさせた.

その後, 数学的ゲーム・パズルの「12-9 (偶数引く奇数)」を紹介し, それをどのように実装するかについて, 最初に画面設計を行わせ, その内容をワークシートに記

入させた。ここで、紹介した12-9（偶数引く奇数）の内容を改良してもよいと伝えた。

画面設計が完了してから、12-9（偶数引く奇数）をGUIとCUIでどのような差が生じるかの一例について、スライドで説明を行なった。実際に用いたスライドをFig.4に示す。ここでは、全体の流れのみを示し、詳細については受講生に考えさせた。この活動を通して、受講生はiPhoneプログラミングを行う際には、従来のプログラミングと違った観点が必要となることや、画面設計の重要性を感じている様子が伺えた。

授業後の自由記述では、感想として「自分のiPhoneに転送できるところがアプリ開発のいいところですね」「就職先で何のプログラミングをやるかわからないので、今いろいろ学べるのはうれしい」「プログラミングと一言にいてもたくさんあることを知れてよかった」「復習をきちんとしないとイケない・・・」「いろんな創造性があるということがわかった」などのコメントが得られた。

#### 4.3. 第3日目（数学的ゲーム・パズルの実装(1)）

午前中に、前回の内容について簡単な復習を行い、アルゴリズムについて重点的に復習を行ない、配列について指導を行った。その後、記入させたワークシートを元に、数学的ゲーム・パズル「12-9（偶数引く奇数）」を実装させる活動を終日行なった。また、実装や変形・改良で自分なりに工夫した点についても記入させた。その後、受講生に完成させた「12-9（偶数引く奇数）」を発表させた。また、作成後の発表時に、他の受講生から意見を述べさせ、それを自身のアプリにどう活かすかについて述べさせた上で、さらに自身のアプリをどのように改良したらよいかについて考えさせ、それを追加して実装させた。

その後、数学的ゲーム・パズルの「セブンアップ」を紹介し、12-9（偶数引く奇数）のときと同様に、どのように実装するかについて、最初に画面設計を行わせた。その後、セブンアップをGUIとCUIでどのような差が生じるかの一例について、スライドで説明を行なった。実際に用いたスライドをFig.5に示す。授業後の自由記述では、感想として「少しずつプログラミングについてわかってきたような気がします」「他の人の作品が見れ

るのがいいし、それを参考にもっとよい作品にしたいと思える。アプリを作るときは、他のいろんなアプリを試したりすることも大事だと思えた」「繰り返しのところが最初よくわかりませんでした、先生がまず最初に具体的に書き出してみようというやり方をおっしゃっていて、それで理解できました。ありがとうございます。」「iPhoneプログラミング講座なのに、いろんなプログラミングが学べていいと思う」などといったコメントが得られた。

#### 4.4. 第4日目（数学的ゲーム・パズルの実装(2)）

午前中に、前回の内容について簡単な復習を行なった後、記入させたワークシートを元に、数学的ゲーム・パズル「セブンアップ」を実装させ、発表させた。ここでも、同じ題材を実装しているにも関わらず、全員違うアプリとなっていることで、受講生は他の作品を見て「自分のアプリに応用しよう」「もっと画面を減らせるんじゃないか」など、自身で作成したアプリと比較していろいろなことを考えていた。また、全員の画面設計および構成が違っていったことから、画面設計の重要性を強く感じ取っている様子が見られた。最後に受講生同士でお互いのアプリについて感想や、今後の抱負などについて発表させた。iPhoneアプリ作成にあたり、基本的なアルゴリズムの学習もおろそかにしてはいけないこと、自身のアイデアを実現するためにアプリ開発やプログラミングは強力なツールとなり得ることを伝え、講義を終了した。

授業後の自由記述では、感想として「他の参加者のアプリを見て刺激を受けました。もっといい作品を作れるようになりたいです」「1人でやっていたらこんなにスムーズに開発できなかったと思う。参加してよかったです」「プログラミングは一筋縄ではいかないですね」「先生がいろんな角度からプログラミングの例を出してくれて少しプログラミングのイメージが変わりました。もっと勉強しないとですね（笑）」「今後ともいろいろご指導下さい！」「次はどこかのコンテストであえるとうれいす！」などといったコメントが得られた。

#### 4.6. 実践の評価

事前事後でプログラミングに対する興味、関心を把握

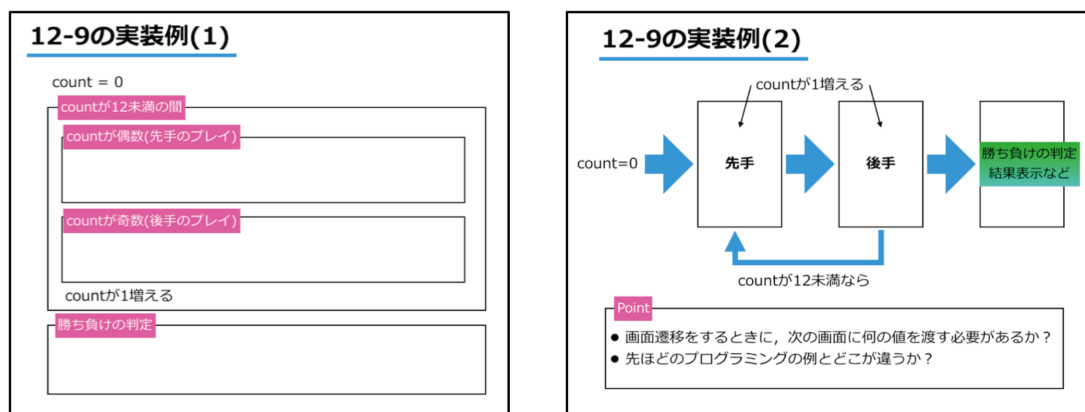


Fig.4 実際に用いたスライド（12-9（偶数引く奇数））



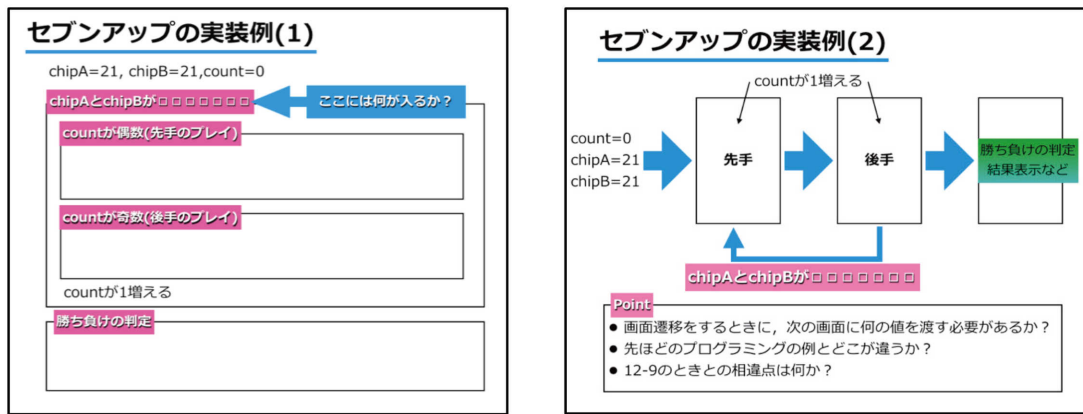


Fig.5 実際に用いたスライド (セブンアップ)

する項目について、そして各回の講義終了時に項目 C および初学者のプログラミングにおける思考過程の自覚程度を把握する項目のそれぞれについて、「4:とても 3:まあまあ 2:あまり 1:まったく」の4件法で回答を求めた。集計結果と各個人における推移を Table.4, 5, 6 に示す。

これらの結果から、「本講座中の演習課題が自分でできる自信がある」が5%水準で有意傾向を示した。他の項目については有意差は見られなかったが、事前事後を比較すると、いずれも事前より事後の値が高まっていた。しかし、「プログラミングを学ぶことは不安に感じる」の値が上昇していた。この理由として、受講生の1人が「プログラミングは一筋縄ではいかないですね」と言っていたように、これからプログラミングを学ぶにあたって、復習をしていかなければならないと強く意識したことや、これからより複雑な事柄が出てくることに対して取り組むことを想定することで、やや不安感が増したことが考えられる。他の項目が下がったということはなく、プログラミングの意味がわからなかったとか、自信を失ったということはなかったことから、単にプログラミングが理解できないため不安を感じたとは考えにくい。前報では不安感は減少していたことから、CUIなどの他の例を示すことは、単にプログラミングが楽しいと思わせることだけではなく、これからプログラミングに携わっていくにあたっての態度や姿勢をある程度育成できた可能性があると考えられる。

受講生の作成した作品のスクリーンショットおよび設計の様子について示す (Fig.6, 7, 8)。なお、以下の作品は受講生本人に許可が取れたもののみを示している。

全体として、画面レイアウトの作り込み度に差はあったが、それぞれ自身のアイデアを盛り込んでおり、同じ数学的ゲーム・パズルであっても同じものは一つとしてなかった。全てのアプリは、説明がそれほどなくとも遊べる作品となっており、値の受け渡しもうまくいっていることから、プログラミング未経験者であっても、学んだ内容を活かして最後まで完成させることができていた。

事後調査において、自由記述欄に記載された内容を以下に示す。

Table.4 項目 B の推移

因子		事前	事後	t 値
プログラミングに対するイメージは良い、	平均	3.83	3.83	0.00 n.s.
	S.D.	0.37	0.37	
プログラミングを学ぶことは自分の役に立つと思う	平均	3.83	3.83	0.00 n.s.
	S.D.	0.37	0.37	
プログラミングを学ぶことは不安に感じる	平均	2.17	2.50	1.00 n.s.
	S.D.	0.69	1.12	
プログラミングの学び方を学びたいと思う	平均	3.33	3.83	1.46 n.s.
	S.D.	0.75	0.37	
プログラミングの文法や手法を身につけられる自信がある	平均	3.17	3.33	1.00 n.s.
	S.D.	0.37	0.47	
本講座中の演習課題が自分でできる自信がある	平均	2.83	3.33	2.24 †
	S.D.	0.37	0.75	
プログラミングの力を生活の中で活かす自信がある	平均	3.17	3.50	1.58 n.s.
	S.D.	0.37	0.50	

(df=5, n=6)

Table.5 項目 C の推移

因子		1	2	3	4	F 値
プログラミングに興味・関心はありますか	平均	3.83	3.83	3.83	3.83	0.00 n.s.
	S.D.	0.37	0.37	0.37	0.37	
今日の講義はよくわかった	平均	3.83	4.00	3.83	3.67	0.74 n.s.
	S.D.	0.37	0.00	0.37	0.47	
今日の講義は面白かった	平均	3.83	4.00	4.00	3.67	1.41 n.s.
	S.D.	0.37	0.00	0.00	0.47	
今日の講義は興味深い内容だった	平均	3.83	4.00	4.00	3.83	0.67 n.s.
	S.D.	0.37	0.00	0.00	0.37	
今日の講義は役に立った	平均	4.00	4.00	3.83	4.00	1.00 n.s.
	S.D.	0.00	0.00	0.37	0.00	
今日の講義で自信を持てた	平均	3.50	3.67	3.67	3.67	0.11 n.s.
	S.D.	0.76	0.47	0.47	0.47	
今日の講義は満足だ	平均	3.67	3.83	3.83	3.83	0.22 n.s.
	S.D.	0.47	0.37	0.37	0.37	
自分なりに工夫できた	平均	3.00	3.67	3.67	3.67	1.11 n.s.
	S.D.	1.15	0.47	0.47	0.47	

(df=3,20, n=6)

Table.6 思考過程の自覚程度を把握する因子の推移

因子		1	2	3	4	F 値
問題理解過程	平均	3.00	3.14	3.14	3.39	0.32 n.s.
	S.D.	0.69	0.73	0.64	0.46	
要素分解・統合過程	平均	2.76	2.88	3.19	3.31	0.96 n.s.
	S.D.	0.56	0.67	0.56	0.54	
局部的自己点検過程	平均	2.72	2.97	3.08	3.25	0.64 n.s.
	S.D.	0.76	0.63	0.64	0.41	
大局的自己点検過程	平均	2.88	3.00	3.13	3.38	0.56 n.s.
	S.D.	0.85	0.66	0.49	0.47	

(df=3,20, n=6)

「1人だと解決できないことも解決できるようになった。受講してよかった。」

「コンテストで実力を発揮できるようになりたいので、今後ご指導よろしくお願いします。」

「Xcode はとても使いやすかったが、全部英語（筆者注：Xcode のエディタが全て英語表記ということ）なので、慣れるのはもう少し時間がかかりそうです。」

「iPhone プログラミングだけでなく、いろんなアルゴリズムを学べてよかったです。」

「他のプログラミングもやってみたいので、また教え

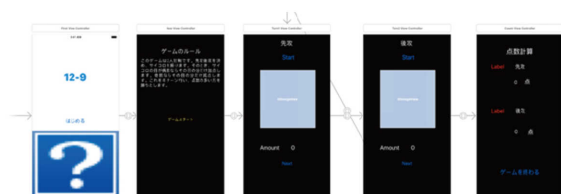


Fig.6 受講生 T 君の画面設計 (12-9)

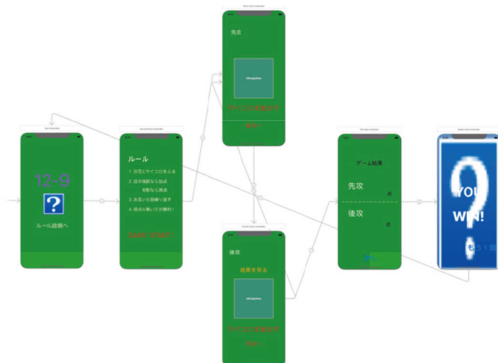


Fig.7 受講生 N 君の画面設計 (12-9)

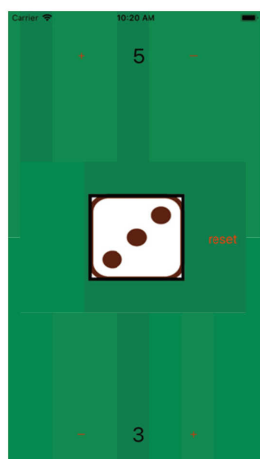


Fig.8 受講生 H 君による12-9 (偶数引く奇数)

て欲しいです。(最近、人工知能で Python とか話題になってますよね??)」

「今まで C 言語しかやったことなかったが, Swift は書き方が少し違って結局プログラムの基礎が大事だと痛感した. もっと大学でプログラミングを真面目にやりたい。」

「今回の授業では, 創造性がいろいろあるとわかったことが, 一番よかったことです。」

などのコメントが得られた。

以上のことから, 全体として, GUI と CUI の差を感じさせる本稿を通して, プログラミングに対する興味・関心を高めるだけにとどまらず, 有意差こそ見られなかったが, 全体として大半の項目の平均値を維持することができた。そして, これからプログラミングを学ぶにあたって, 復習をしていかなければならないと強く意識させ, より前向きにプログラミングをやっていくという態度や, 解決していこうとする姿勢をある程度育成できたと考えられる。よって, 本稿の一定の有用性が示唆された。

## 5. まとめと今後の課題

本稿では, 数学的ゲーム・パズル「12-9 (偶数引く奇数)」, 「セブンアップ」を用い, GUI と CUI における差を感じさせる実践を行なった。その結果, iPhone プログラミング未経験者の受講生全員が最後まで実装することができ, 主体的にプログラミングに取り組ませることができた。そして, これからプログラミングに携わっていくにあたっての態度や姿勢をある程度育成できた可能性があることが示唆された。

しかし, GUI と CUI の差を感じさせる授業がどこまで iPhone を開発する実践に寄与したかについては不明な点が多く, 一般的に行われている実践との比較・対照実験を行うなど, より詳細に検討する必要がある。また, 今回の実践では参加者が 6 人と少なく, 量的な評価を行うには不十分であったことは否定できない。今後, さらなる実践の精緻化を行い, 単に iPhone アプリを開発するだけにとどまらないような, アルゴリズム的な思考を育成するような実践を展開する必要がある。

## [文献]

- 1) レイ・カーツワイル: シンギュラリティは近い [エッセンス版] 人類が生命を超越するとき, NHK 出版 (2016)
- 2) 広口 正之: シンギュラリティとは~2045年問題, JNSA Press, 37, pp.1-5 (2014)  
[http://www.jnsa.org/jnsapress/vol37/2\\_kikou.pdf](http://www.jnsa.org/jnsapress/vol37/2_kikou.pdf) (2018.7.9アクセス確認)
- 3) 厚生労働省: 人口動態統計 (n.d.)  
<http://www.mhlw.go.jp/seisakunitsuite/bunya/hokabunya/shakaihoshou/dl/07.pdf> (2018.7.9アクセス確認)
- 4) 日本生産性本部: 日本の労働生産性の動向2017年版  
[http://www.jpc-net.jp/annual\\_trend/annual\\_trend2017\\_full.pdf](http://www.jpc-net.jp/annual_trend/annual_trend2017_full.pdf) (2018.7.9アクセス確認)
- 5) 経済産業省: IT 人材の最新動向と将来推計に関する調査 (2016) <http://www.meti.go.jp/press/2016/06/20160610002/20160610002.pdf> (2018.7.9アクセス確認)
- 6) 文部科学省: 小学校学習指導要領 (平成29年3月告示) (2017)  
[http://www.mext.go.jp/component/a\\_menu/education/micro\\_detail/\\_icsFiles/afldfile/2017/05/12/1384661\\_4\\_2.pdf](http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afldfile/2017/05/12/1384661_4_2.pdf) (2018.7.9アクセス確認)
- 7) 文部科学省: 中学校学習指導要領 (平成29年3月告示) (2017)  
[http://www.mext.go.jp/component/a\\_menu/education/micro\\_detail/\\_icsFiles/afldfile/2017/05/12/1384661\\_5\\_2.pdf](http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afldfile/2017/05/12/1384661_5_2.pdf) (2018.7.9アクセス確認)
- 8) 文部科学省: 第7回 経済・財政一体改革推進委員会 経済社会の活力ワーキング・グループ (2017)  
<http://www5.cao.go.jp/keizai-shimon/kaigi/special/reform/wg7/291114/shiryoul.pdf> (2018.7.9アクセス確認)



- 9) 阿部 和広：子供の創造的活動とプログラミング学習，情報処理，57(4)，pp.349-353 (2016)
- 10) 文部科学省：平成26年度文部科学省委託事業 情報教育指導力向上支援事業プログラミング教育実践ガイド (2015)  
[http://jouhouka.mext.go.jp/school/pdf/programming\\_guide.pdf](http://jouhouka.mext.go.jp/school/pdf/programming_guide.pdf) (2018.7.9アクセス確認)
- 11) M.Thomas: iOS app programming using an inverted classroom in a small department, *Journal of Computing Sciences in Colleges*, 29(5), pp.179-185 (2014)
- 12) 飯尾 淳：初学者向けプログラミング教育に関する意識調査 -GUI と CUI のどちらがよいか?-，情報処理学会 情報教育シンポジウム2015，pp.17-22 (2015)
- 13) 平嶋 宗，福井 昌則，林 雄介：異種リソースから作られた指示の対比によるプログラミング的思考の振返り，第43回教育システム情報学会全国大会講演論文集，pp.133-134 (2018)
- 14) 福井 昌則，森山 潤：課題設定・解決力を育成するスマートフォンアプリ開発を用いたプログラミング導入教育の実践 -大学生を対象とした課外教育の事例-，兵庫教育大学学校教育学研究，第30巻，pp.119-125，(2017)
- 15) 福井 昌則：情報活用能力の育成を目的とした初学者対象 iPhone プログラミング教育の実践，第1回国際 ICT 利用研究学会全国大会講演論文集，A3-1，(2016)
- 16) 総務省：プログラミング人材育成の在り方に関する調査研究報告書 (2016) [http://www.soumu.go.jp/main\\_content/000361430.pdf](http://www.soumu.go.jp/main_content/000361430.pdf) (2018.7.9アクセス確認)
- 17) 弓野憲一(編著)：世界の創造性教育，ナカニシヤ出版 (2005)
- 18) JetBrains: AppCode Smart IDE for iOS/macOS development (n.d.) <https://www.jetbrains.com/objc/> (2018.7.9アクセス確認)
- 19) Unity Technologies: Unity (2018)  
<https://unity3d.com/jp> (2018.7.9アクセス確認)
- 20) Microsoft: Xamarin.iOS (n.d.)  
<https://docs.microsoft.com/ja-jp/xamarin/ios/> (2018.7.9アクセス確認)
- 21) Adobe: PhoneGap Build (n.d.) <https://build.phonegap.com> (2018.7.9アクセス確認)
- 22) Apple: Playground (n.d.)  
<https://www.apple.com/swift/playgrounds/> (2018.7.9アクセス確認)
- 23) 山上 道恵：パズルの解法を通じて学ぶプログラミングの基礎，日本情報科教育学会誌，2(1)，pp.33-38 (2009)
- 24) 福井 昌則，宮寺 良平：課題解決力と創造力を育成する Java プログラミング教育の実践，関西学院高等部，論叢62号，pp.9-24 (2017)
- 25) 福井 昌則，黒田 昌克，森山 潤：ゲーム・パズルを題材に高校生の創造的態度の育成を図るプログラミング教育の試み，日本教育工学会論文誌，suppl-42，印刷中。
- 26) 福井 昌則，萩倉 丈，黒田 昌克，森山 潤，平嶋 宗：数学的ゲーム・パズルを題材にプログラミングのモチベーションを高める高校生対象ワークショップ型プログラミング講習の実践と評価，コンピュータ&エデュケーション，CIEC，45，印刷中。
- 27) 福井 昌則，森山 潤：サイコロを用いたゲームを題材とする初学者向け Java プログラミング学習の実践，第33回日本産業技術教育学会近畿支部研究発表会講演論文集，pp.27-28 (2016)
- 28) ライナー・クニツィア (正田 謙(翻訳))：ダイスゲーム百科，スモール出版 (2015)
- 29) 森山 潤，足立 明久，桐田 襄一：初心者のプログラミングにおける思考過程の自覚程度を測定するための尺度項目の作成，京都教育大学教育実践研究年報，12，pp.103-119 (1996)
- 30) 西 磨翁：本気ではじめる iPhone アプリ作り Xcode 9.x+Swift 4.x 対応，SB クリエイティブ (2017)
- 31) 森 巧尚：作って学ぶ iPhone アプリの教科書【Swift4&Xcode 9対応】 ~人工知能アプリを作ってみよう!，マイナビ出版 (2017)

## 付記

本研究の実施にあたり，株式会社 E-prost の鹿間正仁氏にご尽力いただきました。