タブレット端末を活用したプログラミング教育（２）

Programming Education Using the Tablet Device（II）

田近　一郎，本多　一彦，杉江　晶子[1)]，森　博
Ichiro TAJIKA，Kazuhiko HONDA，Akiko SUGIE，Hiroshi MORI

前回本紀要で報告した「タブレット端末を利用したプログラミング教育」では，プログラミング演習やそれに関連する講義においてタブレット端末を補助的に用いることが有効であることを示した．しかし，当時は利用するアプリケーションプログラムと学内ネットワークの相性の悪さから，その活用がかなり制限されていた．現在その相性の悪さは解決され，タブレット端末が演習で本格的に使えるようになっている．本稿では，改善されたネットワーク環境の下でプログラミングに関連する講義において新たな試みを行ったことにより，プログラミング能力の上達やアルゴリズムの理解の増進に効果があったので報告する．なおタブレット端末は，マルチタッチ入力や重力センサーといったインターフェースが特徴であり，これらのインターフェースを利用する iPad 専用のソフトウェア開発環境がリリースされている．これらのアプリを活用したプログラミング教育を拡げることが次の目標であり，その概要を紹介する．

In the previous report entitled “Programming Education Using the Tablet Device” for this journal, we have shown that the complementary uses of the iPad are very effective for both programming exercises and lectures related to programming. However, the application was considerably restricted due to incompatibility between the used software on the iPad and the network configuration on our campus. At the present time, the network compatibility is solved, and it enables the full-scale application for programming exercises. In the current study, we show that both the improvement of network environment for programming exercises and the further attempt for lectures ensure the progress of programming capability and the enhancement of understanding of the algorithmic concepts. The tablet device is characterized by the specific interfaces such as multi-touch input, gravity sensor and so on, and several software development environments on the iPad using these modern interfaces are started to release. Our next target is to disseminate programming education using these software applications, and a review about them is briefly mentioned.

キーワード：プログラミング教育，タブレット端末，iPad，プログラミング言語，アルゴリズム，
iPad アプリ，プログラミングオンモバイル
programming education，tablet device，iPad，programming language，algorithm，
iPad app，Programming on mobile

[1)] 名古屋文理大学短期大学部

１．はじめに

平成23年度より，名古屋文理大学情報文化学部情報メディア学科（平成24年度から学部名も情報メディア学部．以降情報メディア学科と記す）では，入学者全員に iPad を無償配布し，教育に活用してきた．iPad を利用した教育の成果は，様々なメディアを通して発表している[1)-9)]．プログラミング教育においても平成24年度から iPad の機動性を活かす試みを行ってきた[10)]が，平成25年度は，より多くの演習・講義にその試みを拡げ新たな知見を得たので報告する．

２．プログラミング教育でのアプリ利用の状況と学生の ICT 環境について

2.1　プログラミング教育でのアプリ利用の状況

平成24年度は iPad の機動性をプログラミング教育に活かす試みとして，iPad 上で複数言語によるプログラミングが可能な「CodeToGo」[11)] アプリを小人数ゼミと２年次生演習科目「プログラム演習 I」で導入した．ただし，「CodeToGo」はプログラムのコンパイル／実行時に外部サーバとの接続を必要とするため，それが可能なインターネット接続環境になかった平成24年度における情報実習室や講義室での導入は限られており，そうした接続の影響を受けない自宅等での利用と，接続制限のないゼミ室での実施が主であった．平成25年度には，学内ネットワークが一新され，「CodeToGo」は学内のどこからでも利用できるようになり，学生は演習・講義中に即座にプログラムの作成とコンパイル・実行が可能となった．そこで「CodeToGo」を「プログラム演習 I」で本格的に導入した．また，２年次生講義科目「アルゴリズムとデータ構造」でも，簡易なスクリプト言語「Lua」[12)] によるプログラミングができる「TouchLua」[13)] アプリを導入しアルゴリズム理解の補助とする試みを始めた．

さらに最近「Codea」[14)]，「Pythonista」[15)] などの iPad 独自のソフトウェア開発環境といえるプログラミング用アプリがリリースされ，iPad による本格的なプログラミング教育の環境が整ってきている．これらのアプリの小人数教育への導入の可能性について調査をおこなった．

本論文の構成は以下の通りである．2.2ではプログラミング教育をとりまく ICT 環境に関する調査結果を報告する．３．と４．では，「CodeToGo」，「TouchLua」を利用した大人数対象の演習・講義での活用事例および調査結果を報告する．５．では「Codea」，「Pythonista」アプリの詳細を解説し，これらのアプリの小人数教育への導入の可能性について展望を述べる．

2.2　プログラミング教育をとりまく ICT 環境調査

情報メディア学科入学者全員に iPad の無償配布をはじめた平成23年度以来毎年度，iPad を半年間使用した1年次後期のはじめに，「プログラミング入門」科目受講者に対して ICT 環境調査を継続して実施している[2)]．平成25年度は iPad 導入３年目にあたり，学生の ICT 環境も導入当初とは大きく変化した．その変化の状況を把握するためアンケート調査の結果を簡潔にまとめ考察する．

平成23年度と比較すると，自宅でのインターネット環境では Wi-Fi 環境が充実している（図１）．また，所有ケータイではフィーチャーフォン所有率が激減した一方で8割以上の学生がスマートフォンを所有しており，しかもその約半分が iPhone 所有者である（図２）．パソコン環境では Mac パソコン所有率が倍増（８％→15%）している（図３）．プログラミング未経験者率は75%前後（図４）と平成23年度と変化はほとんどない．

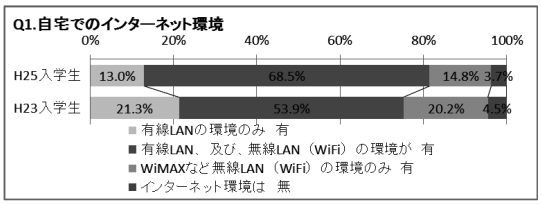

図１　自宅でのインターネット環境

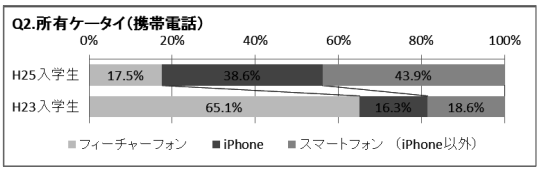

図２　所有ケータイ（携帯電話）

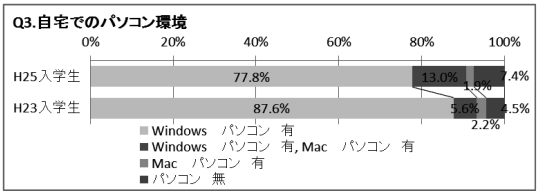

図３　自宅でのパソコン環境

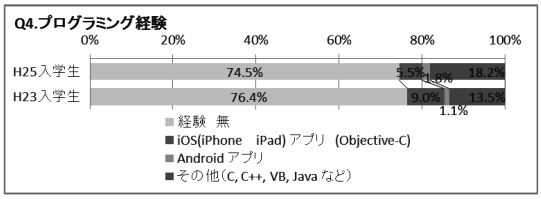

図４　プログラミング経験

ただし，平成23年度は「プログラミング入門」が必修科目で入学者95人中回答数89人，平成25年度は選択科目で入学者103人中回答数54人である。なお，図５以降では，平成25年度入学生の調査結果のみを示す．

SNS 利用頻度では，mixi が激減し８割を超える学生が Twitter や LINE を利用している（図５）．

配布された iPad mini の活用については，無料アプリを10個以上インストールした学生が4割近くいる反面，半数を超える学生が有料アプリをインストールしていない（図６）．活用頻度では Web 閲覧と授業中の教材利用で活用する頻度が高く，メールの送受信にはほとんど利用されていない（図７）．メールの送受信はスマートフォンでおこなっていることがうかがえる．学生は授業での iPad 活用には概ね満足しているといえる（図８）．

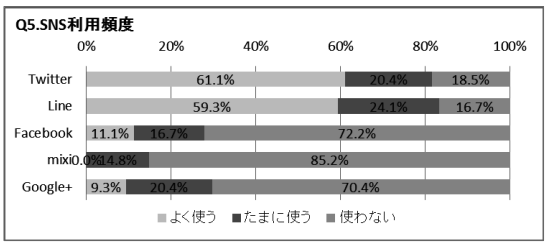

図５　SNS 利用頻度

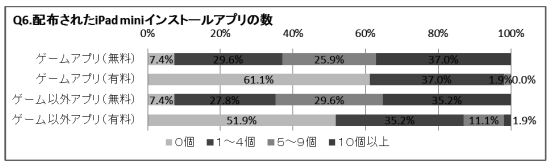

図６　iPad mini インストールアプリの数

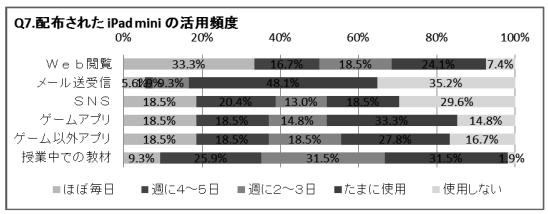

図７　iPad mini の活用頻度

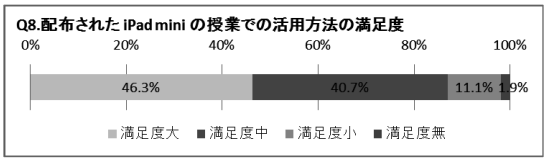

図８　iPad mini の授業での活用満足度

配布希望の情報機器として，半数近くは iPad を希望している一方，３割近くが Mac パソコンを希望していること（図９）が興味深く，iPad アプリの開発等を考慮すると今後，情報機器導入に関して検討の必要性が感じられる．

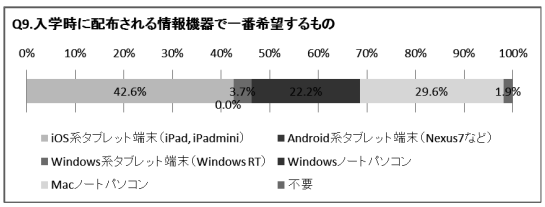

図９　配布希望情報機器

３．活用例の実際

大人数対象の演習および講義科目における iPad 活用の事例を以下に2つ示す．

3.1　大人数対象の演習「プログラム演習Ⅰ」での活用

２年次生前期選択科目「プログラム演習Ⅰ」では１年次生後期選択科目「プログラミング入門」の Visual Basic 言語に引き続いてＣ言語を使ったプログラミング教育を行う．後期に開講する「プログラム演習Ⅱ」と併せて１年間でＣ言語の修得を目的とする．言語教育環境としては，Borland C++ コンパイラをコマンドラインから使う方法で行っており，Windows プログラミングではなくＣ言語の文法とＣ言語を使う基本的なアルゴリズムの教育が主な内容である．

昨年同様，補助的教育環境として「CodeToGo」を導入した．昨年度は Wi-Fi 環境の問題から情報実習室で「CodeToGo」が使用できず，自宅とゼミ室のみ使用という試験的なものであったが，本年度は Wi-Fi 環境が整ったため，本格使用となった．

授業では，PC による演習を中心に行い，iPad の活用は「CodeToGo」用専用課題を毎回1問解かせるという形態をとった．あくまで補助的活用なので専用課題の提出は必須とせず，学生の自主性にまかせた．

3.2　大人数対象の講義「アルゴリズムとデータ構造」での活用

2年次生前期選択科目「アルゴリズムとデータ構造」は，各種言語に共通するデータ処理の手順（アルゴリズム）を，データ整列アルゴリズムやデータ探索アルゴリズムなどの解説を通して修得することを目的にしている．平成24年度は，アルゴリズムの概念を理解し慣れるための方策として，アルゴリズムを実装したソースコードを資料として配布し，iPad 上での編集・作成から実行まで可能なプログラミング用アプリにソースコードを取り込むしくみについて提案した[10]．このしくみを利用し，学生がコードの修正と実行を繰り返しながらアルゴリズムの理解を深める可能性について言及した．

今年度は実際に「TouchLua」アプリによる iPad プログラミングを導入した．この iPhone・iPad 向け無料アプリ「TouchLua」はプログラミング言語「Lua」によるプログラミング／コンパイル／実行がオフラインで可能なアプリケーションソフトウェアである．「Lua」はスクリプト言語の一種で文法構造が簡易で学習しやすい特徴があり，C 言語との親和性が高く PC でのゲーム開発，特にゲーム AI の実装のために国内外で広く利用されている．今回の導入でもプログラミング言語そのものの学習の容易さを考慮し「Lua」を選択した．また，アプリケーションソフトウェアの選択について，iPhone・iPad 上で「Lua」プログラミングが可能なアプリは複数あるが，プログラムの新規作成，編集，実行，コマンドラインからのデータの入力などの操作が，PC での「Lua」プログラミングとほぼ同様であるアプリが望ましいとの観点から今回は「TouchLua」アプリを選択した．なお，iPad を持たない４年次生には学校貸与のノート PC に「Lua」開発環境「SciTE」[16] を学内ネットワーク内からダウンロードしてもらった．この開発環境のインストールに複雑な操作は不要で学生は任意のファルダに開発環境のフォルダを配置して直ちに使用できる．

次に，「TouchLua」による演習の進め方について述べる．あらかじめ「アルゴリズムとデータ構造」用の Web サイトを開設しておき，各回の授業で解説するアルゴリズムを実装した Lua プログラムを Dropbox 経由でダウンロード可能とした．また，「TouchLua」の操作の手引きも PDF ファイルで配布した．学生が授業中にアルゴリズムの動作を確かめるとき，上記サイトの該当するアルゴリズムのリンクから Dropbox に

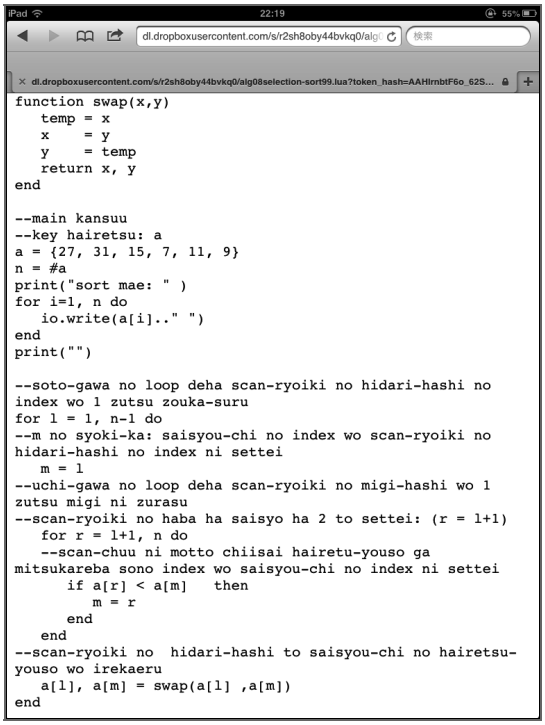

図10　DropBox 上で表示されたソースコード

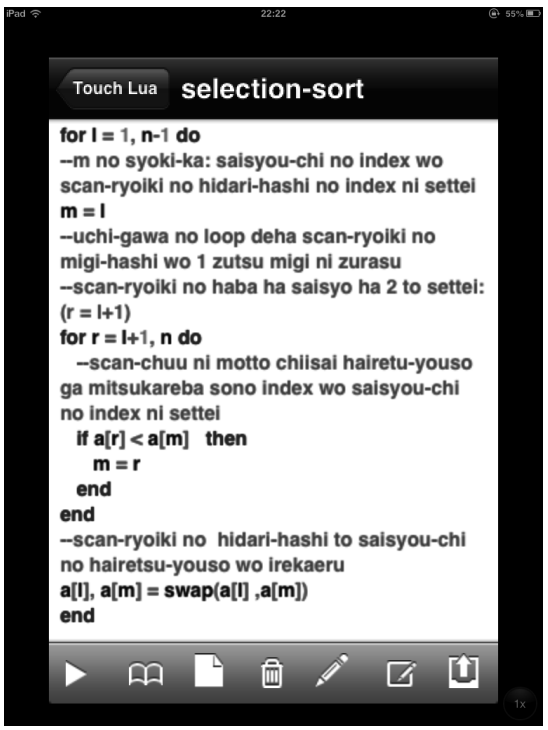

図11　「TouchLua」編集画面上のソースコード

図12 「TouchLua」コマンドラインに出力されたソートの結果

保管したLuaプログラムを開き，ソースコードを画面表示する(図10)．同時に起動しておいた「TouchLua」のトップ画面の「New」をタップしてファイル名入力用の画面に遷移し，その「File Name」欄にLuaプログラムのファイル名（ここでは例としてselection-sort）を入力し，さらに「Accept」をタップして編集画面に遷移しておく．ここに先ほどの画面表示させたコードをコピー＆ペーストし（図11），画面左下にある実行アイコン（三角形）をタップしてプログラムを動作させる（図12)．

講義では，学生にアルゴリズム中の特に重要な変数やコードの役割を理解してもらうため，その変数の値の変更やコードのコメントアウト／変更等を指示し，その結果アルゴリズムの動作がどのように変化するかを確認してもらった．また，レポート提出の場合には，半完成品の形でLuaプログラムを配布し，学生に穴埋め問題として残りのコードを考えてもらい完全なプログラムとして完成させたうえでメールでの提出を指示した．ただし，この作業は授業内で実施するには時間がかかりすぎるため半期を通して3回のみ実施した．

4．アンケート調査と考察

4.1 大人数対象の演習「プログラム演習Ⅰ」における「CodeToGo」アプリ導入に関する調査

今年度（平成25年度）も平成24年度と同様，「プログラム演習Ⅰ」の受講者に対し，半期の講義がほぼ終了した時点でアンケート調査を行った．アンケート調査対象者は，22名であった．平成24年度に比べ受講者が減少しているのは，カリキュラム変更によりこの科目が必修から2年次生選択科目に変更されたためと思われる．

QA1：「CodeToGo」ダウンロードの有無

ダウンロードした学生は22名中17名と77%であった．昨年と違って，情報実習室のWi-Fi環境が整備されたので，受講者全員にダウンロードするよう指示したが，実際には100%には至らなかった．

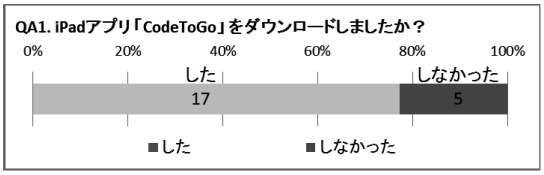

図13 「CodeToGo」ダウンロードの有無

QA2：「CodeToGo」ダウンロードしなかった理由

ダウンロードしなかった理由をたずねてみると5人中4人が有料(250円)という点をあげた．ただ，1名が，「CodeToGo」を使わず，アプリ「Textastic Code Editor for iPad」[17]でコードを書き，webサイト「ideone.com」[18]にコピーペーストして実行という形をとっていた．この方法だと，1つのアプリでコーディングと実行ができない欠点があるが，コードの中に日本語でコメントを記述するなど全角文字を使ってメールを送信しても文字化けしないメリットがある．また，入力にやや難があるiPadにおいて，自分好みのエディタを使用したいという欲求も理解できる．

QA3：「CodeToGo」がPC環境より優れている点

ダウンロードした17名に対して，平成24年度同様，「CodeToGo」を使った方がPCを使うよりどの点が優れているかを図14に示した8項目について質問した．「どちらでもない」を含めない4段階評価で調査した結果，「そう思う」と「ややそう思う」の2つで60%を越えた項目は8項目中6つと，全体に好意的な結果であった．その中でも「そう思う」と「ややそう思う」が合わせて80%弱と特に評価が高かったのは，①「iPad上だけで入力・コンパイル・実行できる」，④「RUNするのに長いコマンドを打ち込まなくてよい」と⑤「完成課題の提出をメールで送信するのが楽」の3項目で

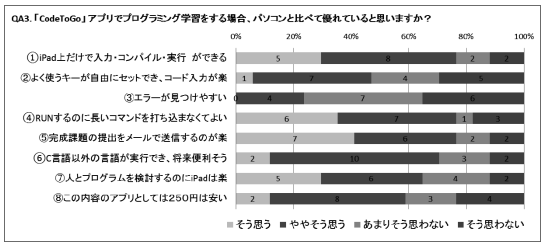

図14 「CodeToGo」がPC環境より優れている点

あった．これら３項目はPCのコマンドライン操作に比べて「手軽にプログラミング」ができて，できた課題プログラムを「簡単にメールで提出できる」点を評価したものと思われる．その２つのメリットは我々の狙いの一つであり，その効果がアンケートで裏付けされた．

一方，断然低い評価は，昨年同様，③「エラーが見つけやすい」で「あまりそう思わない」と「そう思わない」が合わせて70%を越えている．これは「CodeToGo」のエラーメッセージが，PC上のBorland C++コンパイラ日本語版のものより分かりにくいこと，つまり，「CodeToGo」のエラーメッセージが英語表示である点と、エラーの場所の指摘がPCのコンパイラに比べ分かりにくいことが原因と思われる。

QA4：その他の「CodeToGo」の長所（自由記述）

その他，「CodeToGo」の長所と思われる点を自由に記述してもらった．記述した学生は５名であった．２名がPCより手軽にプログラミングが学習できる点をあげ，１名が自宅での学習に便利と答えていた．１名はDropboxとの連携の良さを指摘しており，それを使うとPCでの学習とプログラムの移行がシームレスになり，学習効果の向上が見込まれる．iPadでのプログラムコード入力の効率の悪さをカバーすると思われるが，iPadで完結したプログラミングの学習をしようと思う学生にとって，Dropboxの使用は難しい選択といえる．

QA5：「CodeToGo」の欠点

最後に「CodeToGo」の欠点は何かとたずねる質問には，やはり①「Wi-Fi環境がないと『RUN』できない」と答えた学生が突出して多い．「CodeToGo」がWi-Fiにつながることが必須である（iPad単独で実行できない）点が「いつでもどこでも気軽にプログラミング」できる便利さを大きく阻害している．平成24年度はこの点が最大の欠点と指摘されていたが，このアプリを使う限りこの欠点からは解放されない．平成25年度は大学キャンパス内のWi-Fi環境が改善され，「CodeToGo」がキャンパス内のどこでも実行可能になったため，②や③の不満がほぼ解消された．

また，④「日本語を使用してメールを送信すると文字化けする」を指摘した学生も10名いた．「CodeToGo」の便利な機能として，プログラミングの結果を担当教員等にメールで簡単にソースコードと実行結果を送ることができる仕掛けがあり，学生も長所として評価しているが，日本語を使うと文字化けしてソースコードが読めなくなる欠点が存在する．やむを得ず，メールで結果を提出したい学生は漢字を使用しないように指導した．「CodeToGo」の中では漢字は使えるのでなんとか対処方法を見つけたいが，現在のところ見つかっていない．

4.2　大人数対象の講義「アルゴリズムとデータ構造」における「TouchLua」アプリ導入に関する調査

「アルゴリズムとデータ構造」の受講者に対し，前期最終回の2013年７月23日にアンケートを実施した．iPadを所有する情報メディア学科２年生，３年生の計47名の受講者（アルゴリズムとデータ構造全体で受講生54名）のうち37名から回答を得た．以下に各質問項目とその結果および考察をまとめた．

QB1：「TouchLua」ダウンロードの有無とその理由

講義の初期にこの科目では無料の「TouchLua」アプリを用いてプログラムを作成しレポート提出をおこなう旨を説明した．アンケート結果から多くの学生がダウンロードしている一方，一部の学生は「授業の内容が多すぎて手が回らない」からダウンロードしていないと回答している（図16）．この科目は専門性の高い内容のため高い理解力を要し，また学習すべき内容も多いことから，これ以上内容を増やすと消化不良を起こしかねないと考える学生が一部に含まれることも避けられないと思われる．

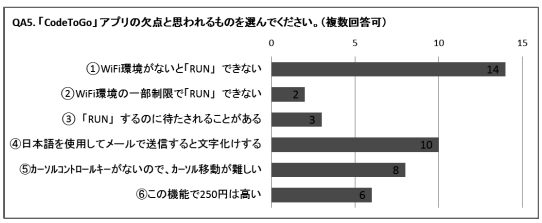

図15 「CodeToGo」の欠点（複数回答）

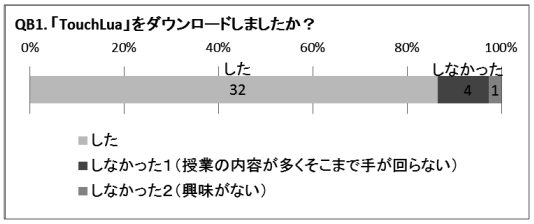

図16 「TouchLua」ダウンロードの有無とその理由

QB2：「TouchLua」によるプログラミングに関する所感

図17の質問項目：①「iPad 上だけで入力・コンパイル・実行できるのが楽」，②「プログラムのエラーが見つけやすい」，③「実行時に長いコマンドを打ち込まなくてよい」，④「友達などとプログラムを見せあって検討するのに向いている」，⑥「完成したプログラムをメールに添付して送信できるのが便利」は，昨年度・今年度の「プログラム演習 I」で実施された「CodeToGo」演習の受講生に対しておこなわれた質問項目の一部と同じであり，アンケート結果もそれらの結果と類似している．つまり，iPad 上では「手軽にプログラミング」できるが「プログラムのエラーを見つけるのはやや大変」という傾向が個別のプログラミングアプリによらず現れていると読み取れる．なお，この授業では Lua プログラムを半完成品の形で学生に配布したため，質問⑤の結果のように学生はプログラム作成の煩雑さを感じていない．

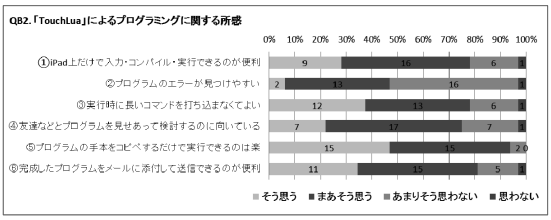

図17 「TouchLua」によるプログラミングに関する所感

QB3：授業時間外に「TouchLua」を起動した時間と場所

授業内に実施した Lua プログラミング演習の多くは授業内容を理解していれば５～10分あればできるものであった．そのため多くの学生は実際に授業時間内に演習を終わらせ，授業終了後まで演習を続けている学生は少ない．一方，大学での昼休み等の隙間時間や自宅で自主的に Lua プログラミングをおこなう学生も少数ながら見受けられる．

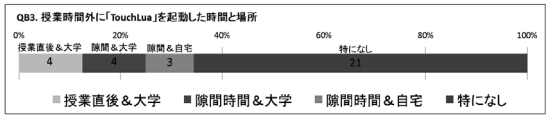

図18 授業時間外に「TouchLua」を起動した時間と場所

QB4：授業時間外に「TouchLua」を起動した頻度

QB3によると多くの学生は授業時間内に演習を終え，授業時間外に Lua プログラミングをおこなう学生は少ない．そのことが QB4の結果にも表れている．

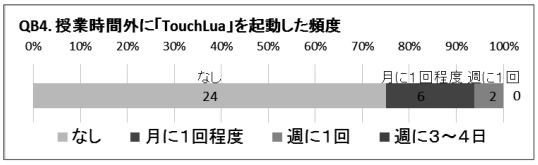

図19 授業時間外に「TouchLua」を起動した頻度

QB5：授業時間内外を問わない「TouchLua」の継続使用時間について

QB3で述べたように大半の学生にとって演習内容は時間がかかるものではないことがはっきりと表れている．

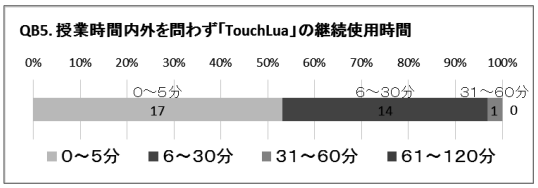

図20 授業時間内外を問わず「TouchLua」の継続使用時間

QB6：「TouchLua」によるプログラミング演習の望ましい実施頻度

小さなプログラムであっても，プログラムを完成させてレポート提出までをおこなう学生にとってそれなりに手間のかかるプログラム演習を頻繁におこなうことは学生から望まれてはいない．この科目では，平成25年度に導入したプログラムコードの提出以外に，従来からレポート用プリントにアルゴリズムを記述して提出する形式の課題を講義３回につき１回程度の頻度で実施しており，学生はこれ以上のレポートの増加に負担を感じていると思われる．

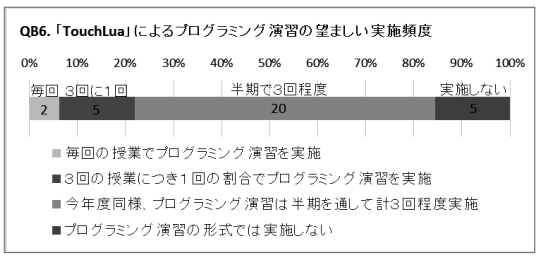

図21 「TouchLua」によるプログラミング演習の望ましい実施頻度

QB7：「TouchLua」プログラミングのアルゴリズム理解への役立ち

およそ３分の２の学生が「TouchLua」プログラミング演習がアルゴリズムの理解に一定程度以上役立っていると感じている結果となった．数理的な内容で理解にいたるまでのハードルが高い「アルゴリズムとデータ構造」であるが，今回導入した「ミニ」プログラミング演習の方法を工夫すればより理解度の向上が見込める結果と考えている．

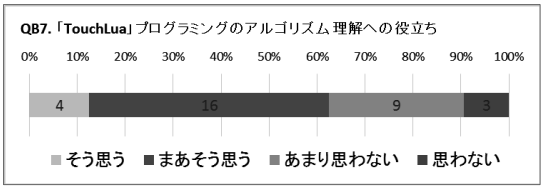

図22 「TouchLua」プログラミングのアルゴリズム理解への役立ち

５．iPad を活用するプログラミング教育のさらなる展開

３．で活用事例を紹介した「CodeToGo」は外部サーバを経由するプログラミング環境であった．この環境では，主だったプログラミング言語は一通り使えるものの，対話型のアプリケーションは作成できない．また，当然のことではあるが利用にたいして，ネット接続を前提としている．iPad の機動性を活かしたプログラミングを目指す場合，iPad のみで実行可能なプログラム開発環境が望まれる．iPad で利用可能なアプリケーションを調べてみると，ソースコードのダウンロードやアップロードの可搬性に制限があるものの，アプリケーション内インタープリタ言語を利用したプログラム開発環境がいくつか存在することがわかった．インタープリタ言語を単に移植したものや，iPad の特性を活かしながらも，プログラム開発環境として実用にはまだ課題が残るものが多く存在した．しかし少数であるが，１つのプログラム開発環境として完成度の高いアプリケーションを発見することができた．ここでは2つのアプリケーションについて，その機能を紹介し，将来にわたっての iPad を利用したプログラミング教育について展望したい．

まず，2011年 9 月に発表された「Codea」である．「Codea」は，ゲームやシミュレーションプログラム作成を目的にしたプログラミング開発環境で，プログラミング言語としては，「TouchLua」同様 Lua が使用されている．タッチセンサー，加速度センサー，カメ

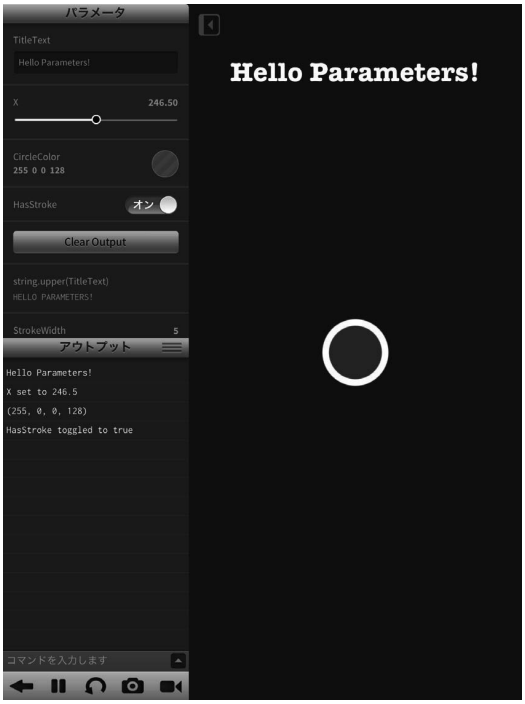

図23 Codea の実行画面

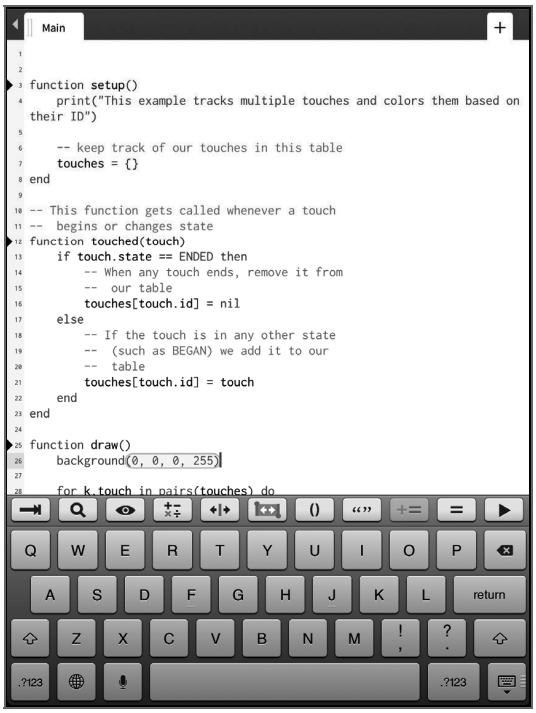

図24 Codea の編集画面

ラなどの制御が可能なライブラリが実装されており，これらの入力に対応したアプリケーションを，iPad 単独で作成することが可能である．一般にモバイル端末でのアプリケーション開発は PC で行い，プログラムをモバイル端末に転送する方法が取られる．しかし，センサーを駆使したアプリケーションでは，実際の動作を逐次モバイル端末で確認する必要があり，デバッグ作業が煩雑となる．「Codea」では，こうしたプログラミングを，直接おこない確かめることができるため，プログラミングにたいする興味維持の効果が期待できる．また物理エンジンライブラリも利用できるようになっており，ゲーム作成のみならず，物理シミュレーションでの活用でも効果的であると考えている．

「Codea」の仕様を詳細に見ると，極めて特徴的な仕掛けが随所に施されていることがわかる．例えばゲーム作成では，オブジェクト指向に基づくクラスの概念を利用してゲームでのパーツを作成することがある．Lua 言語ではクラスの概念が存在しないが，「Codea」では Class 関数をはじめとする「Codea」独自のビルドインされた関数群を利用することで，若干煩雑ながらクラスベース方式のオブジェト指向プログラミングを実現している．また，「Codea」では，エクセルの各シートに相当する複数のタブにプログラムを書くようになっており，その各タブがほぼ1つの関数またはクラスに相当する．

「Codea」の描出画面は，iPad に特化されており，実行のパラメータを変えるスライダー部，標準出力部，実行部に 3 分割されたもの（図23），もしくはフルスクリーンのいずれかで表示される．そのため，画面サイズ以外ほぼ同じハードウェア仕様の iPhone であっても「Codea」を利用することはできない．一方で，複数種のデバイスを考慮する必要がないため，画像描出のための設定の簡素化に成功している．iPad 上で稼働するプログラム開発アプリケーションでは，ソースプログラムの編集を考慮して，標準のソフトウェアキーボードが拡張されていることが多い．「Codea」においても，カーソル移動を含めて Lua 言語でのプログラミングを行いやすいように設計されている（図24）．加えてプログラミングエディタ上でライブラリ関数を選択して検索ボタンを押すと，該当するライブラリ関数について，オンラインマニュアルが表示されるようになっている．またライブラリ関数に渡される，色・音・スプライトの情報は，Lua 言語にとっては引数であるが，「Codea」からは，１つのオブジェクト

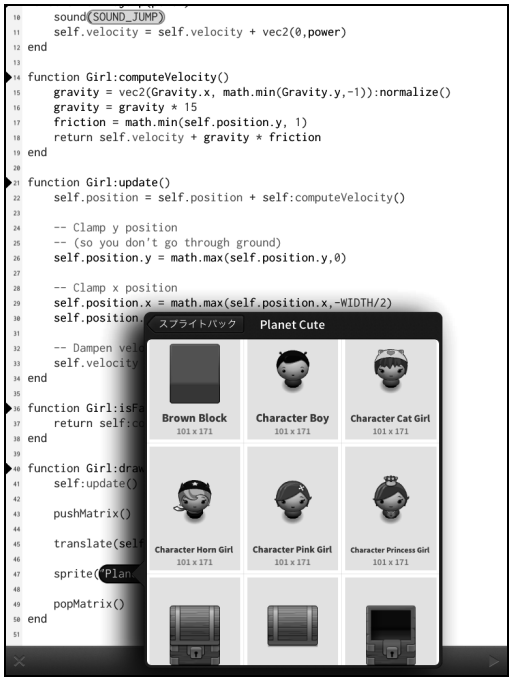

図25　Codea での sprite 引数の入力

になっており，その特性をグラフィカルに変更することができるようになっている（図25）．「Codea」では，チュートリアルを含め様々なオンラインマニュアルにアクセスできるが，マニュアルを精読しただけでは，解決できない問題が生じる可能性もある．ネットワークに接続された環境であれば，「Codea」から直接 Codea Forum へアクセスできるため，「Codea」の製作者を含め，エキスパートからコメントをもらえる可能性がある．現在のところ，マニュアルは英語で書かれているが，次期バージョンより日本語のローカライズがあるとのアナウンスがある．

「Codea」を利用した最大のデモンストレーションを見るには，「Cargo-Bot」[19] とよばれるゲームを体験するとよく，App Store より iPad 上に無料でダウンロードできる．この「Cargo-Bot」は，クレーンを使って荷物を指定された配置に並び替えるパズルゲームであるが，クレーンの動作は副プログラムを含む，順序・条件・繰り返しの制御プログラミングに依っている（図26）．「Cargo-Bot」は１つの独立したアプリケーションであるが，これはまず iPad 上の「Codea」を用いて作成され，そのコードをパソコンに移し，パソコン上で AppStore で配布可能なアプリケーションとして生成したものである．したがって「Codea」のサンプル

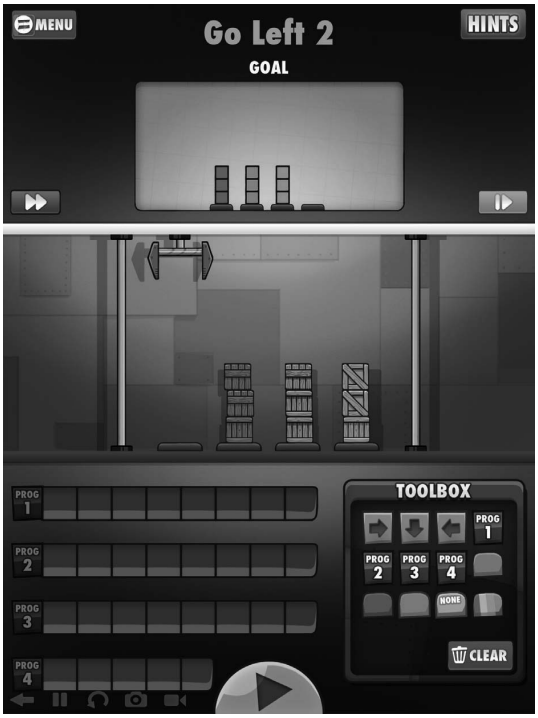

図26　Codea で作成された Cargo-Bot

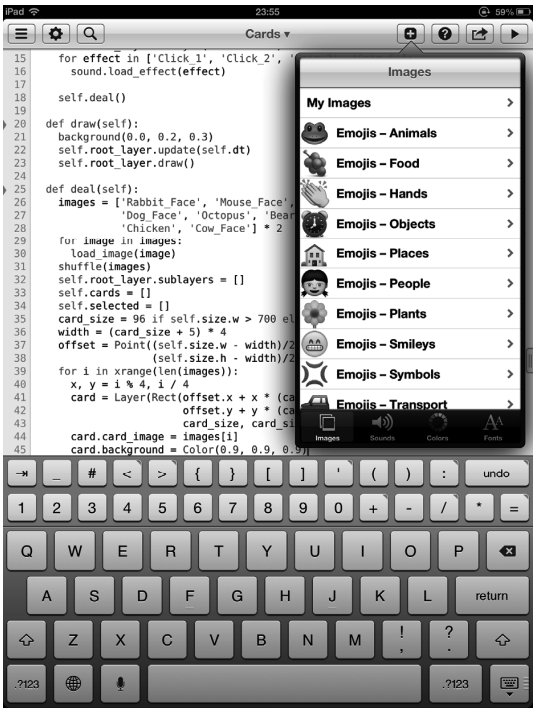

図27　Pythonista の編集画面

プログラムとして，「Cargo-Bot」がソースプログラムとして含まれており，「Codea」という環境で作成されたもう１つの環境が「Cargo-Bot」という，高次に展開された世界が iPad 上の１つのアプリケーションの中に実現しており，大変興味深い．

「Codea」の後，2012年７月に発表されたアプリケーションが，「Pythonista」である．「Pythonista」は，名前のとおりプログラミング言語として，Python を使用しており，「Codea」と同様，ゲームなどの対話型のアプリケーションも作成可能である．しかし，「Codea」が，ゲームとシミュレーションに特化したプログラミング開発環境であったのに対し，「Pythonista」は，より一般的なプログラム開発を目指した環境であるといえる．プログラミングに適した拡張されたソフトウェアキーボード，オンラインマニュアル，フォーラムへのアクセスなど，「Codea」と同等の機能を持つ（図27）．一方，コンソール画面を用いて，OS コマンドを逐次実行させることや，非対話型のグラフィックの描出など，GUI 登場以前のスタイルでのプログラミングも可能である（図28）．また，「Codea」と異なり iPhone 上でも動作する．

「Pythonista」の特徴の１つとして，データ通信機能

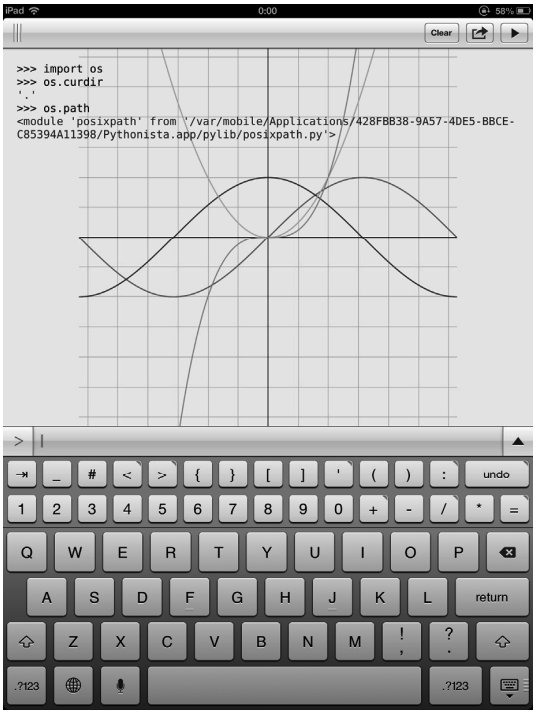

図28　Pythonista のコンソール画面への出力

が挙げられる. Dropbox へのデータアクセスや, クリップボードやブックマークレットを利用したデータの受け渡しは，モバイル端末のアプリケーションの中に封じ込められたデータの幾分かを，アプリケーション間連携で活用できることを示している[20].

「Codea」,「Pythonista」ともに iPad 上で稼働するプログラム開発環境として高機能であり，それぞれの特徴を踏まえたうえで使い分けることが重要である．ただし,「Codea」,「Pythonista」のいずれの場合も，例えばタッチイベントの取得に関して，アプリケーションがコールしなければならない特定の関数内に処理を記述する必要がある．つまり，PC 上で Java 言語を用いてゲームを作る場合, タッチイベントの処理を, キャラクタオブジェクトのメソッドとして記述できるのに対して,「Codea」「Pythonista」ではキャラクタオブジェクトの記述に制限が生じることになり，プログラムが煩雑になる問題がある．複雑な対話型のプログラムでは，作成が困難となっていくため，本格的なプログラム作成では，PC による作成にシフトしていかなければならないと考えている.「Codea」のように各種センサーを利用したプログラムを簡単に作成できること,「Pythonista」のようにネットワークを介したプログラムを iPad でも iPhone でも作成できることは，プログラミングオンモバイルの特徴であり，大いに強調されるべきものである．しかし，機動性のある機器を用いて，プログラミングへの興味を失わせることなく，その楽しさの機会を増やすことこそが，プログラミングオンモバイルの一番の利点ではないかと考えている．現在，３，４年次向けのプログラミングオンモバイルのゼミナールを計画しており，その成果はいずれ報告の予定である.

６．まとめ

本論文では iPad を活用するプログラミング教育の事例として,「CodeToGo」を利用するプログラミング演習と「TouchLua」を利用するアルゴリズムの理解を促進する試みを紹介し，これらが学生のスキルアップに有効であることを示した．また，今後 iPad を活用するプログラミング教育の実践で，効果的な利用が期待できる２つの iPad アプリ「Codea」,「Pythonista」の紹介をおこない，あわせて将来への展望を示した．今後も iPad を活用するプログラミング教育の試みを拡げ，学生の学習成果向上を目指したい.

７．引用文献

1）本多一彦：「モバイル機器の変遷から情報教育機器としての iPad を考察する」，名古屋文理大学紀要，11, 97-104 (2011)

2）森博，田近一郎，杉江晶子：「タブレット PC を活用したマルチメディア教育の試み」，名古屋文理大学紀要，12, 97-104 (2012)

3）佐原理，大橋平和，長谷川旭，長谷川聡，KAISER Meagan：「タブレット端末による学校教育現場向け多言語情報配信システム」，名古屋文理大学紀要，12，105-112 (2012)

4）長谷川旭，佐原理，尾崎志津子，本多一彦，山住冨也，長谷川聡：「名古屋文理大学における iPad 導入とアクティブラーニング」，モバイル学会研究報告集，7(2)，45-48 (2011)

5）長谷川旭，長谷川聡，本多一彦，山住富也，佐原理：「大学教育でのタブレット端末の利用とその効果―iPad を無償配布した名古屋文理大学における学生意識」, コンピュータ＆エデュケーション，31，70-73,（2011）

6）尾崎志津子：「iPad を活用したオンライン英語多読の導入―名古屋文理大学情報メディア学科における事例―」，コンピュータ＆エデュケーション . ; 32，49-52 (2012)

7）長谷川聡：「ソーシャルリーディングとソーシャルラーニング」，現代の図書館，50(2)，114-120 (2012)

8）長谷川旭，小橋一秀，山住富也，長谷川聡：「タブレット端末の教育利用と情報インフラ : 名古屋文理大の iPad 無償配布と大学図書館」，医学図書館，59(3)，186-191，(2012)

9）斉藤徹，河原潤，高下義弘：「教える！名古屋文理大学」,『iPad で現場を変える !』，日本経済新聞出版，132-147 (2011)

10）本多一彦，田近一郎，杉江晶子，森博:「タブレット端末を活用したプログラミング教育」, 名古屋文理大学紀要, 13, 85-92 (2013)

11）CodeToGo,
https://itunes.apple.com/jp/app/codetogo/id382677229より2013年10月28日検索

12）Roberto Ierusalimschy（著），新丈径（翻訳），「Programming in Lua プログラミング言語 Lua 公式解説書」，アスキー・メディアワークス，(2009)

13）TouchLua,

https://itunes.apple.com/app/touch-lua/id525273327 より2013年10月28日検索
14）Codea,
https://itunes.apple.com/jp/app/ codea/id439571171 より2013年10月28日検索
15）Pythonista,
https://itunes.apple.com/jp/app/pythonista/id528579881?mt=8より2013年10月28日検索
16）SciTE,
http://www.scintilla.org/SciTE.html より2013年11月８日検索
17）Textastic Code Editor for iPad,
https://itunes.apple.com/jp/app/textastic-code-editor/id383577124より2013年11月８日検索
18）ideone.com,
http://ideone.com/ より2013年10月28日検索
19）Cargo-Bot,
https://itunes.apple.com/jp/app/cargo-bot/id519690804?mt=8より2013年10月28日検索
20）Automating iOS: How Pythonista Changed My Workflow,
http://www.macstories.net/stories/automating-ios-how-pythonista-changed-my-workflow/ より2013年10月28日検索