

タブレット端末を活用したプログラミング教育 (5)

— タブレット端末を用いたテキストプログラミングの新展開 —

Programming Education Using the Tablet Device (V)

— traditional text-based programming on mobile —

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

概要: 「タブレット端末を活用したプログラミング教育」と題した一連の報告で様々なプログラミング方法について実践・考察してきた。第4報では、ビジュアルプログラミングの方法がタブレット端末を用いたプログラミング教育では極めて有用であることを報告した。また、一連の報告の初期には伝統的なテキスト入力によるプログラミングをタブレット端末で行う方法についても報告した。ただし、iPad上でコンパイラ言語の実行可能コードの生成／実行が制限されているため、iPadでのC言語の実行はリモートサーバを介してしか利用できなかった。これはiPadをネットワークに常に接続し、旧来の大型コンピュータのようなバッチ処理的な利用をしなければならないことを意味している。しかし、近年C言語で記述されたプログラムをネットワークに接続することなくアプリ内のインタプリタで実行できるiPadアプリが出現した。本報告ではネットワーク接続を必要としない会話型処理ができるプログラミング学習アプリを用いた実践例を学内の教育環境との連携を含めた形で紹介する。最後にビジュアルプログラミングとテキスト入力によるプログラミングとの関係性と教育における両者のプログラミング方法の適切な使い分けについて展望する。

Abstract: In our successive reports entitled “Programming Education Using the Tablet Device”, the various kinds of programming methods are examined and discussed. We conclude that the visual programming method is one of the most effective tool on tablet device, and has been reported in the previous fourth report. The early stage in our reports, we also mentioned the traditional text-based programming on tablet device. Under the prohibition of generating and executing the codes for compiler programming languages on iPad, the execution of the typical programming language program as C is only possible on remote server. This restriction bears the very old batch processing for use of C language under network connection. However, the recent development of iPad application for learning programming enables us to run the compiler language program within the interpreter embedded in the application. In this report, the use of standalone interactive programming on iPad is shown with the recent development of the educational facilities in our university. Finally the relation between visual programming and text-based programming is considered, and the well-designed use of programming methods is expected.

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, iPad アプリ

Key words: programming education, tablet device, iPad, programming language, iPad application

1. はじめに

2011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（2012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2012年度からiPadの機

動性を活かす試みを行ってきた¹⁰⁾⁻¹⁵⁾。特に2013、2014年度の紀要では当時のiPadで利用可能なプログラミング学習アプリを活用した演習・講義の実践について報告をおこなった^{10), 11)}。それから約5年を経た2018年度、プログラミング教育環境はソフトウェア面、ハードウェア面で大きく変化している。その変化のもとでiPadを利用したプログラミング教育の演習・講義を実践し、一定の

¹⁾ 名古屋文理大学短期大学部

知見が得られたので本稿で報告する。

具体的には、プログラミング教育環境のソフトウェア面での主な変化としてネットワーク接続を必要としない iPad 用のプログラミング学習アプリの進展が挙げられる。2節ではこのプログラミング教育環境の変化について簡単に述べ、新規に導入した iPad 用プログラミング学習アプリ「モバイルC」¹⁶⁾によるプログラミング教育への活用事例を紹介する。一方、プログラミング教育環境のハードウェア面での変化として iPad 上でのテキスト入力によるプログラミングを支援する BYOD ステーションや Apple Pencil の新規導入が挙げられる。3節ではこれらの iPad 周辺機器についてその歴史的変遷に触れるとともに、iPad を利用するプログラミング教育において現時点で効果的と考えられるこれらの活用方法を提案する。4節はまとめと展望である。

2. iPad 用プログラミング学習アプリの活用について

2.1 プログラミング教育環境：ソフトウェア面の変化

iPad をプログラミング教育に活用し始めて2018年度にいたる約5年の間にプログラミング教育を取り巻く環境には、

- ・ iPhone・iPad 向けプログラミング学習アプリケーション（以下アプリと略記）の進展、
- ・ 教材を配信でき学習者の学習状況を把握できる学習管理システム（LMS）を含む e ラーニングシステム「WebClass」の全学的な新規導入、
- ・ 講義において何を学習するのかを示す複数の評価項目と学生が達成しているレベルを示す評価基準をマトリックス形式で表し、学生が自らの学修状況を確認し今後の学修に活かすために利用するルーブリック（ICE モデルに準拠）の全学的な新規導入、

などの変化があった。

以下では、iPhone・iPad 向けプログラミング学習アプリの進展とそれに直接かわるアプリ開発の環境の変化についてやや詳細に述べる。

Apple 社の iOS アプリ開発ツールキット「iOS SDK」の開発者向けのライセンス契約「iOS Developer Program License Agreement」^{17), 18)}（2010年9月改訂）によると、アプリはその内部にインタプリタを組み込んだ上でスクリプト言語のソースコードをインタプリタ実行することは許可されているが、コンパイラ言語の実行可能コードのダウンロード／インストール／ソースコードからのコンパイルによる生成／実行は許可されていない。そこで初期のプログラミング学習アプリの開発者は主にスクリプト言語の学習アプリの開発をおこなっていた。例えば、著者の一人が授業内のミニプログラミング演習用として利用したものでは「Touch Lua」¹⁹⁾、「iOS 用の Python 2.7」²⁰⁾がある。2013、2014年度に利用した「Touch Lua」はスクリプト言語「Lua」の学習アプリで、スタンドアロンで利用でき、無料で導入しやすいが、「Lua」の認知度が低く、発展性のある利用には難点があった¹¹⁾。2015～2018年度に利用した「iOS 用の Python 2.7」はデータサイエンス用言語として需要が高くプログラミングの初年次教育でも採用されることの多いスクリプト言語「Python」の学習アプリで、これもスタンドアロンで無料で導入しやすいが、iOS のバージョンアップで現在は利用できない。

一方、コンパイラ言語のためのプログラミング学習アプリも初期から存在した。例えば、著者の一人がプログラミング演習系科目内で2013～2017年度にかけて利用した「CodeToGo」²¹⁾は「C 言語」の学習アプリで、150円で C 言語を学ぶことができ、学生にとっても比較的導入しやすい^{10), 11)}。ただし、このアプリは、上述のライセンス問題によりユーザ（学生）が作成した C 言語のソースコードをアプリが搭載されたデバイス上でコンパイル／実行ができないため、ソースコードをインターネット経由でサーバに送り、サーバ上で実行し、その結果を再びインターネット経由で返すという方式を採用しておりスタンドアロンで利用できないという難点がある。

コンパイラ言語の学習アプリのこの難点は現在では次の方法で解消されている。現代的なコンパイラは、ソースコードを中間表現（IR）に変換するフロントエンドと中間表現を順次最適化し、指定した CPU に対応する機械語に変換するバックエンドから構成されており、Apple 社が iPhone・iPad 向けアプリ開発のために提供している統合開発環境 Xcode では、バージョン 4.2（2011年10月リリース）以降フロントエンドに C 言語・C++系のコンパイラ：CLang、バックエンドにコンパイラ基盤の一つである LLVM をそれぞれデフォルトとして採用している²²⁾⁻²⁸⁾。バックエンドの LLVM は中間表現を指定した CPU の機械語に変換する機能以外に、中間表

現表

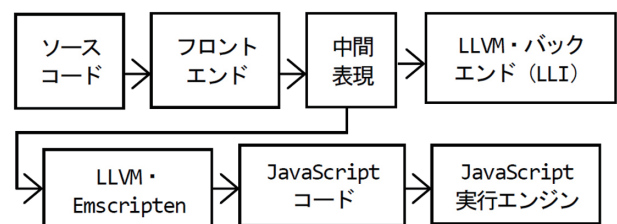


図1 ソースコードを LLVM で実行するしくみ

現を JavaScript コードに変換する機能、アセンブリコードに変換し内蔵インタプリタ LLI で実行する機能も備えている (図1)。

そこで、アプリ開発者は LLVM と JavaScript 実行エンジン²⁸⁾ をアプリの組み込みライブラリとして利用することにより、中間表現から JavaScript コード／アセンブリコードへの変換を組み込み LLVM が、JavaScript コード／アセンブリコードの実行を LLI / 組み込み JavaScript 実行エンジンのインタプリタが、それぞれ処理してくれることを利用して、自作プログラミング言語で記述されたソースコードを中間表現に変換するフロントエンド (コンパイラ) のみを作成するだけで比較的容易にプログラミング学習アプリを開発できるようになった。その結果、2012年以降「C 言語」「C++」「Swift」などのコンパイラ言語をスタンドアロンで学習できるプログラミング学習アプリが App Store に登場した。

プログラミング学習アプリを取り巻く上記の環境変化のもとで「C 言語」プログラミング学習アプリ「モバイル C」が2012年に登場した。このアプリは有料版／無料版があり、スタンドアロンで利用できる。無料版でもプログラミング演習で利用するには問題がなく2018年度に導入することとなった。次節以降で「モバイル C」の各種の特長とその特長を活かした活用事例を詳しく紹介する。

2.2「モバイル C」の特長

「モバイル C」は、C 言語のコード入力と実行がワンタッチで行える統合環境になっており、C 言語の学習者にとっては最適なアプリとなっている。また、我々が従来推薦してきたアプリ「CodeToGo」でプログラムを実行する場合、WiFi 環境が必須であったのに対し、「モバイル C」は一度インストールしてしまえば、オフライン (スタンドアロン) で使用できる点が大きなメリットとなっている。さらに、「CodoToGo」との大きな違いは、バッチ処理ではなく対話処理が可能な点である。「CodeToGo」では、入力データがある場合、あらかじめデータを入力データ専用の場所に記述しておかなければならないが、「モバイル C」では、対話的に入力することができる。バッチ処理と対話処理はそれぞれ一長一短あるが、PC の演習が対話処理なので違和感なく学習できる。

「モバイル C」はアプリ内にディレクトリを作成でき、そこにプログラムファイルを保存できる。また、シェル画面も有しており、代表的な UNIX コマンドを使って

```

cprog2> ls -l
.:
total 56
-rw-r--r--  1 mobile  mobile    0 Oct  8 22:02 3117xxxQ5_1.c
-rw-r--r--  1 mobile  mobile   756 Oct  2 14:28 3117xxxkadai1002.c
-rw-r--r--  1 mobile  mobile   767 Sep 25 13:51 3117xxxmax00.c
-rw-r--r--  1 mobile  mobile   465 Sep 25 14:12 3117xxxmax02.c
-rw-r--r--  1 mobile  mobile   294 Oct  2 13:21 3117xxxreidai5-1.c
-rw-r--r--  1 mobile  mobile   696 Sep 25 13:41 3117xxxsaidai.c
-rw-r--r--  1 mobile  mobile   141 Sep 25 13:16 bai10.c
-rw-r--r--  1 mobile  mobile   169 Sep 28 12:54 inp1.c
cprog2>

```

図2 シェル画面で ls -l コマンドを実行

ファイルの操作も行える (図2)。便利であると同時に UNIX コマンドの学習もできる。このように PC での演習とほぼ同等の感覚でファイルを扱うことができる。

その他、細かい仕様だが、プログラムの編集画面や出力画面のフォントサイズを自由に変更できる。プログラムの編集のしやすさや見易さが細かく調整できるだけでなく、演習課題の提出の際、実行画面のスクリーンショットの文字の大きさを自由に調節できる。iPad でスクリーンショットを撮った直後も保存する前に画面を編集できる機能が OS で提供されているが、フォントサイズが自由になるとさらに効率的に提出課題のレポート作成が行えるようになり学習者にとってのメリットも大きい。

2.3「プログラム演習 II」への活用

2018年度現在、情報メディア学科2年次生は、C 言語の修得を目的として前期に「プログラム演習 I」、後期に「プログラム演習 II」をそれぞれ必修科目として履修する。言語教育環境は、従来の Borland C++ コンパイラをコマンドラインから使う方法から、システム移行に伴い、2018年度から Borland C++ コンパイラの後継であるエンバガデロ社の BCC32C/BCC32X C++ コンパイラへ変更した。内容は Windows プログラミングを行わず、配列、文字列、ポインタ、構造体、関数、ファイル入出力などの C 言語の基本文法の修得と基本的なアルゴリズム教育が主なものである。

今年度初めて「モバイル C」を導入した。前期の「プログラム演習 I」の授業では「CodeToGo」と「モバイル C」の紹介を行い、自宅や自由時間での活用を勧めた。さらに後期の「プログラム演習 II」では第2週から第4週の3週間にわたり著者の担当する約35名の受講生に対し、基本的に PC を使わず、iPad 上の「モバイル C」のみで演習を行った。以下に「モバイル C」のみを使った演習を行うにあたり工夫した点を述べる。

「CodeToGo」を演習の補助に使ったとき、アンケート

から学習者にとって一番の欠点と指摘されたのはコード入力がPCに比較して効率が悪い点であった^{10), 11)}。そこで「モバイルC」の演習では例題のプログラムコードや提出課題のプログラムコード（一部分が空白で未完成のもの）を「WebClass」から配信するようにした。学習者は「WebClass」から配信されたプログラムコードをモバイルCにコピー&ペーストすることにより、コード入力に要する負担が少なく、文法やアルゴリズムの理解に時間を割けるようになった。

演習時間中は補助者が1名教室内を巡回して、学生の質問などに対応するようにしているが、「モバイルC」での演習中は、iPadを持って学生の方から教師や補助者に質問に来るようになった。質問を受けた場合、iPadを受け取り、即座に問題点を指摘したり、補助者で対応が難しい場合はそのまま教師へ対応を依頼するなど、演習をしているデバイスごと移動できるので、教師や補助者の巡回指導がより機動的な対応になった。

提出課題については、完成したソースコード画面と実行結果画面のそれぞれのスクリーンショットを撮り、その2個の画像をアプリ「Word」で編集させメールで教員のアドレスへ送付させることにした（図3）。最新のiOSでは、スクリーンショットを撮ると、保存する前に編集することができるようになったので、PCで同様の作業をするより簡単に提出課題ファイルが作れるようになった。また、「モバイルC」の文字フォント変更機能で文字サイズを変更することにより画面の編集はかなり柔軟に対応ができる。このようにiPadの中だけでプログラ

ミング、提出課題の編集、メールでの提出を完結することができ、慣れてしまえば、非常にスムーズにプログラミングから課題の提出までの作業ができるようになった。

PC演習の場合、課題の提出方法は、実行結果のスクリーンショット（コマンドプロンプトのウィンドウのみ）を「ペイントブラシ」のような画像編集ソフトウェアで適当な大きさに整形し、ソースコードファイルとセットで、学内で準備したレポートフォルダと呼ばれるネットワークドライブ上のセキュリティで管理されたフォルダに提出させている。このレポートフォルダのセキュリティは演習用にかなり強固に管理されており、他人の提出物を閲覧したり、コピーすることはできない。自分のファイルも上書きのみ許されており、削除もできないようになっている。また、教員は学内ネットワークからのみこのレポートフォルダにアクセスでき、学外からアクセスできない。さらに、たとえ学内ネットワークからのアクセスでも、Windows10の場合プロフェッショナルエディションでなければアクセスできない。

一方、「モバイルC」での演習の場合、提出物はメールで教員に送るため、教員は時間や場所に関係なくメールを受信できるので、学生が提出した課題に対しタイムリーに対応することができる。レポートに間違いがあれば直ちに指摘し、訂正後再提出を要求することもできる。これを両者納得がいくまで繰り返すことができる。

以上のように、iPadで課題制作、提出ファイル作成、レポート送信、間違いを指摘したメールの受信、直ちにプログラム修正、と教師と学生間のコミュニケーションが非常に円滑かつスピーディに行えるメリットがある。

2.4 アンケート調査と考察

「プログラム演習Ⅱ」の半期15回の演習のうち、第2回から第4回までの3回分の演習ではPCを使わず、すべて「モバイルC」のみを利用して演習を行った。第4週の演習（2018年10月9日）終了後に「WebClass」上でアンケートを実施した。その日の受講者30名すべてから回答を得た。以下に各質問項目とその結果および考察をまとめた。

Q1. iPadアプリ「モバイルC」でC言語プログラミング学習をする場合、PCで行う場合と比べて「長所である」と思うか否か

図4の質問項目（1）～（7）は、2013年度の紀要でiPadアプリ「CodeToGo」に対する質問とほぼ同じ内容の質問である。前回の調査と違って、今回は（1）～（7）の回

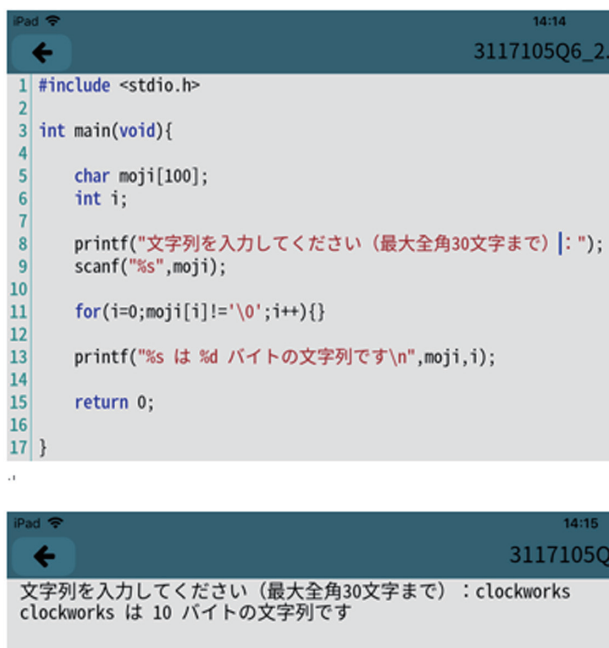


図3 アプリ「word」で編集された提出課題

答傾向に大きな差が生じなかった。すべての項目で「そう思う」と「まあそう思う」の肯定的回答が60%を超えており、「モバイル C」の使い勝手が PC と比べてもあまり遜色がないことがわかる。項目 (4) と (5) が特に高評価であったが、PC による演習では、コマンドラインから長いコマンドを打ち込んでコンパイルや実行をするのに比べ、「ラン」ボタンをワンタッチするだけで実行できる手軽さはやはり学習用アプリとしては大きなメリットである。また、課題プログラムの編集画面と実行画面のスクリーンショットを撮って「Word」アプリで2つの撮影画面をマージした後メールで提出させているが、スクリーンショットの撮影作業が PC より楽にできるという回答が多かった。PC での演習でも同様にスクリーンショットを撮らせているが、iPad の場合、iOS のバージョンアップでスクリーンショットを撮った後、即座に編集が可能になったことにより PC より一手間少なく、効率的に編集が行えると感じたのであろう。

否定的な意見で比較的多かったのが、(3)「エラーが見つけやすい」の項目で40%ほどが「あまりそう思わない」「そう思わない」であった。この点は「モバイル C」の性能向上に期待したい。

Q2.「モバイル C」の PC での演習と比べた長所 (Q1の①～⑦以外) と短所の自由記述

[長所]

- ・どこでもプログラムを作ることが出来ること。
- ・iPad なので持ち運びが楽。わざわざパソコンの電源をつけなくてもいいところ。
- ・エラーの場所がわかりやすいので対処がしやすい。

[短所]

- ・コード入力にソフトキーボードが使いにくい

- ・日本語の入力がやりづらい。全角文字と半角文字の切り替えが PC ほどスムーズにできない。
- ・PC より画面が小さいので見づらい場面がある。

上記の自由記述から、長所としてアプリの起動をはじめとしてすべての作業が PC より素早く手軽に行える点があげられており、タブレット端末の機動性がそのままプログラミング作業に反映していることがわかる。

短所では、コード入力の使いにくさがあげられている。特に全角・半角の切り替えの煩雑さが指摘されている。「モバイル C」のソフトキーボードには全角・半角の切り替え機能がなく、いったん iPad のソフトキーボードに切り替える必要があり複雑な使用方法になる。ぜひ「モバイル C」に切り替え機能を付加してほしい。ハードキーボードにくらべソフトキーボードの効率の悪さはタブレット端末共通の問題である。

PC 画面より小さく見づらいとの指摘もあったが、機動性とのトレードオフの問題であり、ある程度の見づらさはいたしかたなく、現状はフォントのサイズを変更するなどして対応するほかない。

Q3. iPad「モバイル C」を学習するとき、外付けキーボードは使用しましたか？

iPad によるプログラミングでやはりネックとなるのはソフトキーボードによるコード入力にあるということで、外付けキーボードに関する質問をした。

- ・ほぼ毎回使用した・・・13.3%
- ・時々使用した・・・6.7%
- ・ほとんど使用しなかった・・・6.7%
- ・持っていない・・・73.3%

外付けキーボードを所持している学生はアンケート回答者のうち30%弱であったが、持っていないでも使わない

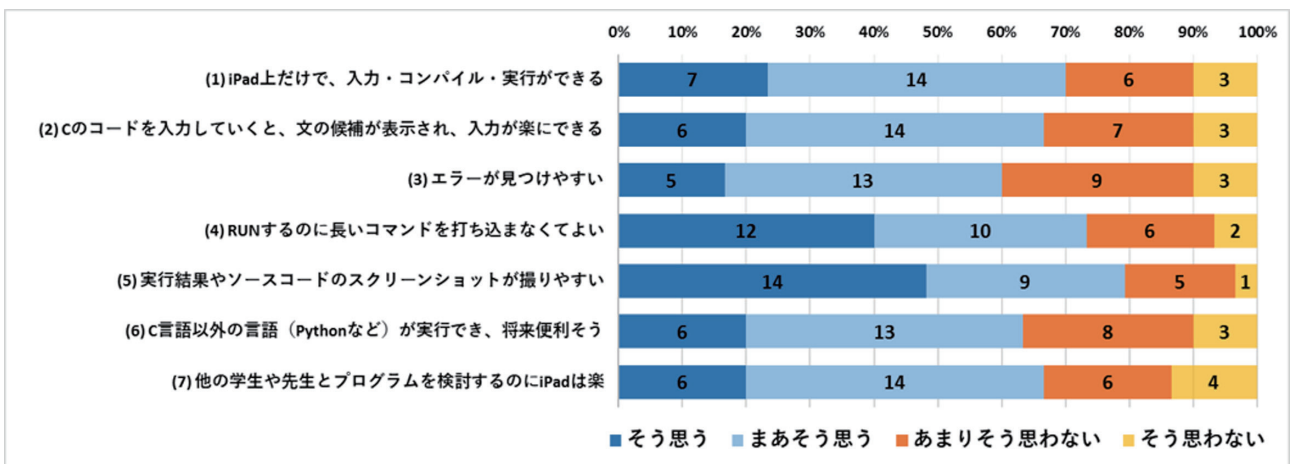


図4 「モバイル C」での C 言語プログラミング学習が PC に比べて優れている (長所である) と思う点

学生が7% 弱存在した。iPad 本体の他に外付けキーボードを持ち運ぶとなると機動性が損なわれるということであらう。

Q4. プログラミング学習に iPad を使うとき、外付けキーボードは必須のオプションだと思いますか？

キーボードを持っていない学生も含めて iPad でプログラミング学習にハードキーボードは必要かという質問をした。

- ・そう思う・・・約25%、
- ・まあそう思う・・・約25%、
- ・あまりそう思わない・・・約25%、
- ・そう思わない・・・約25%

授業内で作成した小規模なプログラムではキーボードの必要性を感じていない学生も半数ほどいると思われる。

それともう一つの問題は、iPad に最適な外付けキーボードがなかなか見つからないという問題もある。iPad ケースと一体化したキーボードというの也被考えられるが市販の物で最適なものはなかなか存在しなかった。ところが、この論文執筆中の2018年10月に Apple 社より新型 iPad Pro が発表された。ライトニング端子が廃止され、代わりに USB Type-C 端子が新設されたことにも驚いたが、新型の専用ペンと専用キーボードの出来の良さにはさらに驚いた。特に秀逸と思われるのが専用キーボードである。このキーボードはキーボード、ケース、スタンドの3つの機能を併せ持つもので、従来の外付けキーボードの欠点をすべて払しょくする大変魅力的なアクセサリのように感じられる。従来から多くの iPad ユーザーがなんらかのケースを装着することが多いが、そのケースにほとんど違和感なくキーボードとスタンドが追加されたという印象である。それほどこのキーボードは薄く感じられる。まだ実物を見ていないので断言できないが、新型 iPad Pro と専用キーボードの組み合わせは、いつでもどこでもプログラミング学習できる理想の組み合わせの一つではなかろうか。

2.5「アルゴリズムとデータ構造」への活用

本学2年次前期の情報システムコースのコース必修科目「アルゴリズムとデータ構造」は、各種プログラミング言語に共通するデータ処理の手順やデータ構造を、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としている。

2013年度は大人数対象の講義における iPad 活用の初めての実践としてこの科目の講義時間内に「TouchLua」

を用いたミニプログラミング演習を実施し、その結果を紀要にまとめた¹¹⁾。「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、その点を考慮し当時のミニプログラミング演習では「Lua」を選択した。また、演習に必要な資料は Dropbox 経由で配信した。

その後、「Lua」同様に文法構造が簡易で学習しやすく、データサイエンス用言語として需要が高く学生も将来的に用いる可能性が高いオブジェクト指向型/関数型プログラミング言語「Python」を選択し、学習アプリとして iPad 版の「iOS 用の Python 2.7」を活用することにした。しかし iOS のバージョンアップにアプリが対応しなくなり、2018年度は「モバイル C」を活用したミニプログラミング演習をこの科目の83名の受講者を対象に実施することにした。「モバイル C」は2.3節で述べた「C 言語」以外にも「C++」「JavaScript」「Python 3」「Lua」など複数のプログラミング言語によるプログラミング/コンパイル/実行がスタンドアロンで可能である。

「モバイル C」による演習の進め方について述べる。事前に e ラーニングシステム「WebClass」内に「アルゴリズムとデータ構造」用ページを開設しておき、各回の講義で解説するアルゴリズムを実装した Python プログラムを「WebClass」経由で配信可能とした。また、「モバイル C」で Python プログラムを作成し実行するための手引き(図5)も「WebClass」経由で配信可能とした。

学生はレポート課題としてプログラムを実装するときまたは講義の一部としてアルゴリズムの動作を確かめ



図5 モバイル C による Python プログラミングの手引き



図6 WebClass 上で表示されたソースコード

るとき、WebClass の該当する講義回のページに保管した Python プログラムを開き、ソースコードを画面表示する (図6)。同時に「モバイル C」で新規ファイル生成用のアイコンをタップしてファイル名 (ここでは例として1人用〇×ゲームプログラム marubatsu.py) を入力し、編集画面を表示する (図7)。ここで先ほどの WebClass に画面表示させたコードを「モバイル C」の編集画面にコピー&ペーストし、画面右下にある実行用アイコン「ラン」をタップしてプログラムを動作させる (図8)。

講義内のミニプログラミング演習では、アルゴリズムの設計上重要な部分が空欄となっている半完成品の Python プログラムを学生に配信し、プログラムの不足部分のソースコードを考察してもらいプログラムを完成

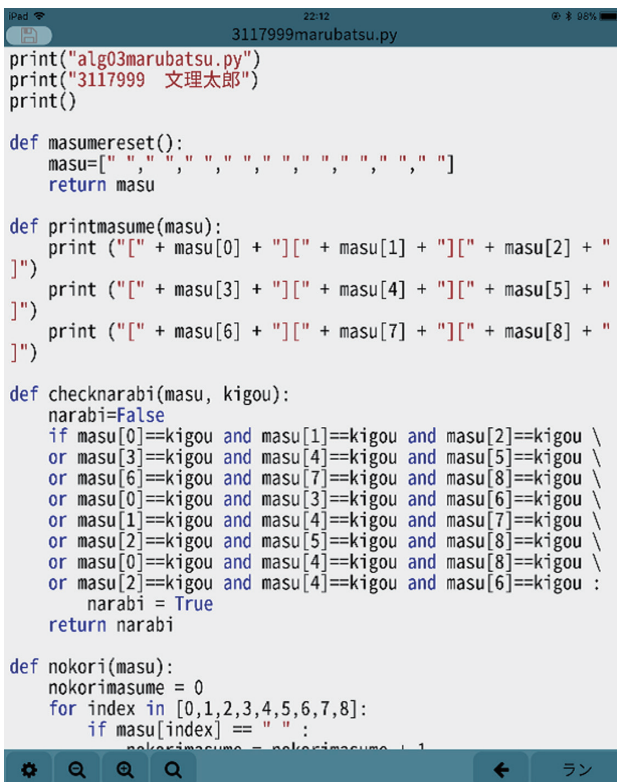


図7 「モバイル C」編集画面上のソースコード

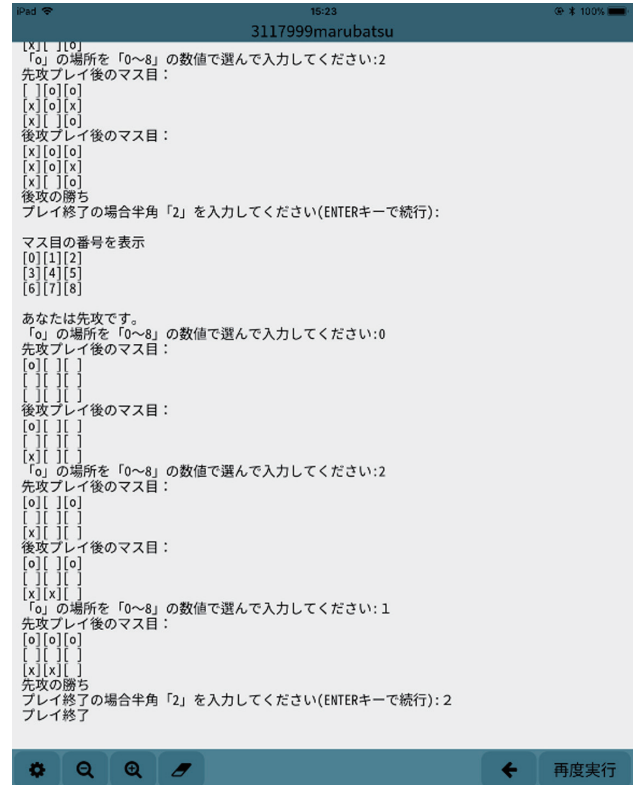


図8 「モバイル C」コマンドラインに出力された
〇×ゲーム結果

させ、動作確認をしたうえでメール提出するように指示した。なお、実施したミニプログラミング演習の内容は以下の6項目である。カッコ内は講義の回とプログラムをダウンロードしかつレポート提出した学生の人数である。

- ・モバイル C で1人用〇×ゲームプログラムを実行しこのツールの使い方に習熟 (第3回・65名)
 - ・変数への代入を利用するデータシフトアルゴリズムのコードを完成させ実行 (第4回・62名)
 - ・if 文を用いる4個の整数データの最大値を求めるアルゴリズムのコードを完成させ実行 (第6回・57名)
 - ・for 文を用いる、配列の中から最小値を求めるためのアルゴリズムのコードを完成させ実行 (第7回・60名)
 - ・2重の for 文を用いる選択ソートアルゴリズムの外側の for 文のコードを完成させ実行 (第8回・60名、これは第7回で作成したプログラムに for 文を1行追加するだけで完成できる)
 - ・マージソートと選択ソートのプログラムを実行し、両者が結果を出力するまでの時間をデータ数を変更しながら比較し時間計算量の概念を確認 (第11回・63名)
- ミニプログラミング演習の実施回数は2013年度の「TouchLua」利用時で3回であったのに対して、今回の「モ

バイル C」利用時では上記のように6回である。これは e ラーニングシステム「WebClass」の導入や「モバイル C」の操作性の高さにより、2013年度に比べて若干多い程度の時間でより効率的にミニプログラミング演習を実施できたことによる。なお、演習用のプログラムも含めて講義中に言及したすべてのアルゴリズムのソースコードを WebClass に保管しておき、学生に自由にダウンロードできるようにした。

2.6「ループリック」と客観的な学修達成度との関係

2018年度全学的に導入された「ICE モデルに基づくループリック」を「アルゴリズムとデータ構造」に適用し、学生に以下の学修評価のマトリックスを提示したうえで、各項目に自己評価による点数を付けてもらった。

「ループリック」のすべての評価項目は、ミニプログラミング演習レポートで出した課題およびこれらの課題に関連するアルゴリズムをより深く理解するための紙によるレポートで出した課題一つひとつに対応しており、各項目の学修達成度を1～5の値で自己評価して回答するものとなっている（図9）。おおむねミニプログラミング演習で出した課題の学修達成度に関する学生の自己評価が反映された値が記入されるとみてよい。

「ループリック」に回答した学生63名は、ミニプログラミング演習に取り組みレポートを提出した学生とほぼ重なっている。そのため、学生の自己評価の平均と、実際の（ミニプログラミング演習の評価の総計＋紙レポートの評価の総計）とを比較することで、学修達成度に関する学生の主観的な自己評価と客観的な評価との相関関係がある程度捉えることができるとと思われる。そこで、縦軸を（ミニプログラミング演習の評価の総計＋紙レポートの評価の総計）のポイントによる客観評価に、横軸を「ループリック」の各項目の学修達成度に関する自己評価をすべての項目にわたって平均したものに、それ

	idea	connection
アルゴリズムとデータ構造	概念がほぼ理解できている場合は自己評価3～4。この概念を用いた基本的なプログラムのレポート課題を解くことができる場合は自己評価5を付けて下さい。	他の概念と組み合わせたプログラムのレポート課題を解くことができる場合は自己評価5を付けて下さい。
変数と代入の概念	データシフトのプログラムのレポート	3.70
if文（条件分岐）、for文とwhile文（繰り返し）等のプログラムの「基本制御構造」	4個の整数の最小値を求めるプログラムのレポート	3.54
プログラムの実行時に何度も繰り返される処理の共通部分を取り出す「関数」の概念	関数Swapを用いる4個の整数の最小値を求めるプログラムのレポート	3.56
呼び出されている関数の中で、その関数と同じ名前の関数をさらに呼び出す「再帰呼び出し」の概念	マージソートの処理過程を関数レポート	3.69
アルゴリズムの性能をif文の使用回数で計測する「時間計算量」の概念	二分探索のプログラムの処理過程を関数レポート	3.62
二分木によるデータ構造の実現	二分木よりのデータの探索、挿入、削除の理解	3.53

図9 ループリックの評価項目

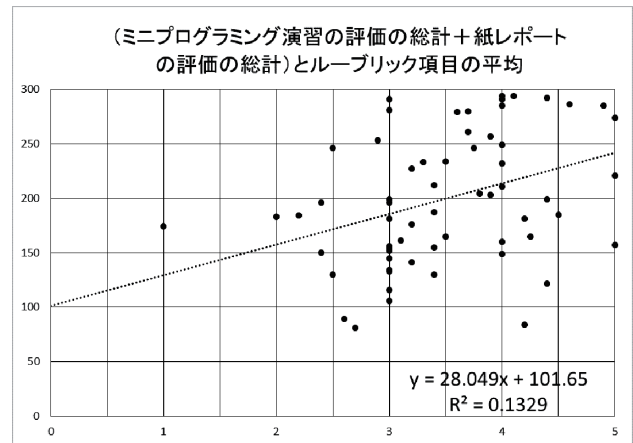


図10 学生毎の学習達成度に関する客観評価とループリックによる自己評価の平均の散布図

ぞれ対応付けた散布図を得た（図10）。散布図によると（ミニプログラミング演習の評価の総計＋紙レポートの評価の総計）のポイントによる客観評価と「ループリック」の各項目の学修達成度に関する自己評価を平均したものとの間の相関係数は0.365であった。この値から両者には「弱い正の相関関係」があることがわかった。

2014年度に実施したアンケートでは、回答した学生のおよそ3分の2が「TouchLua」によるミニプログラミング演習がアルゴリズムの理解に一定程度役立っている」と主観的に感じている結果となった¹¹⁾。2018年度はアンケート調査を実施しなかったが、「ループリック」により学習達成度に関する学生の自己評価と客観評価との間に弱いながら正の相関関係があることを直接捉えることができた。iPadを用いたミニプログラミング演習がアルゴリズムの理解にある程度寄与している結果が出ていると思われる。

3. プログラミング教育と iPad 周辺機器

3.1 iPad 周辺機器の変遷

iPad が発売された当初から外付けのキーボードは存在し、入力での効率化を図る周辺機器として知られていた。30ピンのドック端子にハードウェア接続する純正のキーボードも存在したが、その後 Bluetooth 接続のキーボードもしくは iPad のカバーを兼ねたキーボードカバーが主流となっていった。一方、GUI の操作として、画面を指でタップ、ピンチイン・ピンチアウトする方法が採られ、ペンの使用については考慮されていなかった。スタイラスペンがサードパーティーから発売されていたが、タッチパネルが静電容量方式であるため、充分なペンの反応を得るためにペン先を太くする必要があっ

た。そのため細かい画像の描出などには適していない。また指での操作と区別できず、パームリジェクション機能が使えないため手のひらを画面に着けて入力することができなかった。その後 Bluetooth インターフェースを用いパームリジェクション機能が可能なスタイラスペンも登場したが、OS レベルで機能をサポートしているわけではないので、対応するアプリが限定されていた。

2015年に iPad Pro が発売され、入力方式は 1 つの転機をむかえた。iPad Pro では Smart Keyboard と Apple Pencil が利用可能となった。Smart Keyboard は iPad Pro と Smart Connector を通してハードウェア接続される。さらにこの Connector を通して iPad Pro 本体から電源が供給されるため、周辺機器への接続や充電に関わる煩わしさがなくなることになった。また、Apple Pencil は OS レベルでサポートする Bluetooth ペンでありペン入力が正式にサポートされた。

こうして携帯電話（スマートフォン）でもパソコンでもないモバイル機としてタブレット端末が登場したのだが、現在はハードウェアキーボードの利便性を向上させ、また画面分割によりアプリ間のデータの参照が可能になった。さらにペンデバイスにも対応した。これは iPad という特定のタブレット端末に限ったことではなく、Android OS を搭載したタブレット端末においても同様な変遷をたどっている。つまり現在のタブレット端末は機動性を活かしつつ、パソコンの代わりにもなりうるし、紙と鉛筆の代わりにもなりうるということをアピールしているわけである。では実際にタブレット端末がその能力が教育の、特にプログラミングの場で活かすことができるかどうかを以下で議論していく。

3.2 プログラミング教育環境：ハードウェア面の変化

情報メディア学科では2018年度入学生にも iPad の無償貸与を行ったが、今回は Apple Pencil も同時に配布した。情報メディア学科の映像メディアコースでは映像技術、画像処理、アニメーションなどを学ぶためペンの活用が効果を上げることが容易に想像できる。一方プログラミングでの活用については、2節で報告した演習や講義が2年次で開講されているため現状では報告ができない。しかしコースによらない講義や演習でペンの利用例を見ることができれば、プログラミング教育へも適応できるはずである。例として1年前期の必修科目である「情報学概論」について取り上げる。講義内容は情報科学の基礎的なものであり、教科書指定をしている。教科書は紙の媒体に加え PDF 形式でも利用可となっているので

iPad 上でペンを利用して教科書に注釈を付けていくことができる。しかし、学生に「教科書は iPad で簡単に見ることができるので利用するように」と伝えるだけではどのようにすればよいか困惑してしまうであろう。そこで手順の流れを学生自らが書き込める教材を用意し講義でペンを利用してもらうことにした。一度利用の仕方がわかると自発的に利用が促進されると考えたからである。具体的には、キャッシュメモリの仕組みについてある程度まとめたものを PDF ファイルとし、「WebClass」上にアップロードした。この教材を講義中にダウンロードし、PDF ビューアに転送して利用してもらう。PDF ビューアは無償・有償を問わず多く存在するが、今回は PDF 形式に対応しており、またペンによる注釈機能を持っている電子書籍ビューアである「iBooks」を利用した。「iBooks」は純正のアプリであり、iPad が OS レベルでペンデバイスに対応したことから、ペンの動作の安定性やユーザーインターフェースの親和性の観点から安心して使用できる。キャッシュメモリの仕組みを解説しながら、重要と思った所は注釈を付けるように指示した。講義後「WebClass」を用いたアンケートに Apple Pencil の使用感が記されていた。この講義アンケートは「WebClass」を通して講義毎に行っているものであるが、特に iPad やペンの使用感について言及を求めたものではなかった。しかし以下のような感想があった。

- ・今日は、初めて授業で iPad を使いましたがとても使いやすく話したことをメモに取れるなど使い方によってはノートを取るより使いやすかった。
- ・今日は iPad を活用した新しいカタチの講義で、プログラム起動の流れや主記憶装置の事などを少しずつですが理解できるようになりました。
- ・キャッシュメモリの文字だけの説明はよくわからなかったけれど iBooks を使用した図式説明で理解できて、なるほどと思った。先生が口で言ったところも大切だと思った部分はメモするなど工夫した。
- ・Apple の電子端末システムを使い始めて3年以上経つが、iBooks を初めて使った。もっと様々なアプリを試して、使いこなせるようにしたい。

学生の多くはスマートフォンのエキスパートであり、どんな複雑な作業もスマートフォンの小さな画面を通してこなしてしまう。そのような学生にとっても、大きな画面と絵を描くのではなく注釈作業にペンが利用できるというのは新鮮に映ったようである。今回の試みは非常

に単純なもので、まだまだ工夫・改善の余地があると考えている。しかし次年度はペンの操作にも慣れた学生がプログラミングの演習・講義を受講することになるため、当初の想定であるプログラミングコードの注釈といった範囲を超えたペン利用を行ってくれる学生が現れることも期待している。

次にキーボードの利用についてであるが、3節の始めに述べたようにこの周辺機器の利用は当初から想定されていたが、オプションの域を超えてはいなかった。それが2節にあるように iPad を用いたプログラミング開発環境の発展に伴い、講義・演習での iPad のキーボードの活用に意義が見出されるようになった。こうした状況を見越し、本学では2018年度の教育システムの入替えに際して、演習室など3カ所で BYOD (Bring your own device) ステーションの導入を行った (図11, 12)。BYOD ステーションはディスプレイ装置、キーボード、マウスのみからなる環境であり、ユーザが各自のモバイ

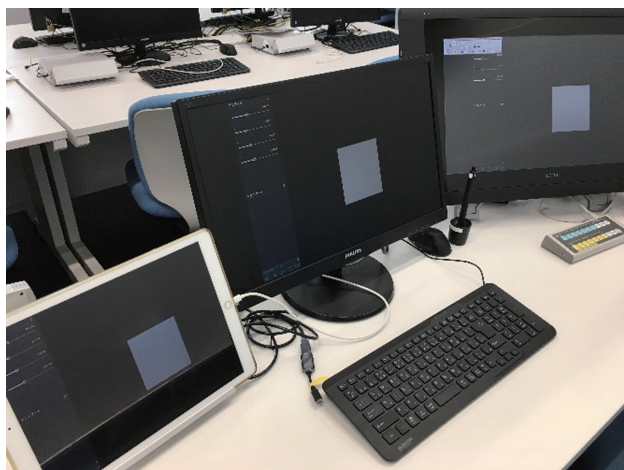


図11 教卓用 BYOD ステーション
右端は呈示用のペンタブレットモニター



図12 BYOD ステーションを設置した演習室の全景

ル端末を持ち込んで利用することを想定している。マウスは iPad では利用できないが、Android 端末の利用も考慮して導入した。モバイル端末とキーボードとの接続は有線で行うようになっている。キーボードに関しては汎用性を考慮して JIS 配列のものを導入したが、iPad は US キーボードとして認識するため、記号キーのキーアサインが異なり入力に困難をきたす問題が生じている。BYOD ステーションがある演習室にキーアサイン表を置き混乱のないよう努めているが、プログラミングでは記号の入力を多用するため不便なことには変わりがない。一方、かな漢字変換はテキストのコピー & ペーストは問題なく利用できる。現在この BYOD ステーションを設置した演習室でプログラミングのゼミナールを開始している。本格的な利用は始まったばかりであるので、学生の利用に関する感想については別の機会に譲ることにするが、あらかじめディスプレイやキーボードの接続法を説明しておけば学生は問題なく利用できている。ディスプレイへの出力は WiFi による無線出力であるが、プログラミングでのソースコードの表示では表示速度の問題を感じることはない。ディスプレイ装置の導入は iPad の画面でも十分な広さがあるが、プログラミングでは、更に大きな画面の方が利便性が向上すると考えたからである。学生用のディスプレイ装置は本来学生が自らのプログラムコードなどを見るために設置したものであるが、ゼミナールの指導では直接学生の隣に座り、学生のコードについて指示を与えることがある。その際学生は自分の iPad とディスプレイ装置の両画面を見てキーボードを操作し、教員は本来学生用であったディスプレイを見て指示を与える指導も可能であった。本学のパソコン演習室には教員のパソコンの内容を表示できる中間モニタを2台に1台の間隔で設置しているが、この中間モニタには学生のパソコンの内容を表示することはできない。しかしこの BYOD ステーションでは、学生が接続しているディスプレイ装置に限定されるが、パソコン演習室での中間モニタ的な利用を学生が接続している iPad から利用できることになる。

2.4節ではプログラミング学習における外付けキーボードの必要性についての質問 Q3, Q4を行い回答を得ている。それではこの Q3と Q4の回答をどのように解釈すればよいか考えてみることにする。Q3では外付けキーボードを所有している学生のおおよそ25%がキーボードを利用していないとの結果が得られた。モバイル機器としてキーボードを持ち歩く場合、キーボードの重量も大きな要因であろう。また Q4の外付けキーボードの必要

性では、意見が完全に4分されてしまっている。軽量のキーボードは入力時の快適性が損なわれることが多いであろう。一方、快適な入力可能なキーボードでは重量のためモバイルには不向きとなるであろう。以上の回答に答えるために BYOD ステーションの普及・活用が鍵となる。ただし1バイトコードが主流なプログラミングの世界であっても日本でのプログラミング教育を考えた際、日本語の入力が可能なことは非常に重要である。しかしアプリの日本語対応に加え、キーボードの JIS, US 規格の違いが問題を複雑にしている。ユーザーインターフェースに関しては“慣れ”の要素が大きいとはいえ、使いにくければ使用率は低下する。現状の BYOD ステーションを最も効率よく利用する方法についてはさらなる経験が必要になるが、これは実際の演習・講義を通じた経験によって得られるものである。有用な知見が得られた際は、別の機会に報告する。

4. まとめ

本稿では iPad を活用するプログラミング教育の事例として、「モバイル C」を利用するプログラミング演習の試みを紹介し、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。引き続き今後も iPad を活用するプログラミング教育の試みを広げ、学生の学習成果向上を目指したい。ただし、今回は手続き型プログラミング言語を文法の基礎から学ぶという伝統的なカリキュラムの中での iPad の活用について考察したものであった。一方で著者らはビジュアルプログラミング言語「ScratchJr」²⁹⁾ によるプログラミングが、プログラムを各機能を分担するクラス（エージェントまたはキャラクター）どうしが相互作用をおこなうものとみなすオブジェクト指向の概念³⁰⁾をきわめて具体的に提示してくれるという利点を活かし、オブジェクト指向プログラミングのための ScratchJr による教授法について提案した¹⁵⁾。

システム開発ではオブジェクト指向プログラミング言語により開発をおこなう方法が主流でありオブジェクト指向プログラミングの習得が欠かせない。著者らは研究の現段階で、あるオブジェクト指向言語上でデザインパターン³¹⁾や SOLID 原則³¹⁾などの設計原理を適用することの多い複雑なシステムのプログラムを設計・実装するためのオブジェクト指向プログラミングスキルを習得するには、ビジュアルプログラミングを活用する上記の教授法と、それとシームレスにつながるオブジェクト指向言語自体の上で構築されたプログラムの設計手法に関する教授法という2種類の教授法を実践することが望まし

いと考えている。一方、クラスを構成する各メソッドの内部のプログラムを設計・実装するためのオブジェクト指向プログラミングスキルを習得するには、本稿で実践したように手続き型言語のスキルを習得するための伝統的教授法をほぼそのまま適用するとよいと考えている。ただし、これらの教授法を適切に使い分ける体系化された方法論は具体化されてらず、今後の研究が期待される。

5. 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，山住富也，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) 齊藤徹，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) 田近一郎，本多一彦，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 (2)」、名古屋文理大学紀要, 14, 75-86 (2014)
- 12) 田近一郎, 本多一彦, 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 (3)」, 名古屋文理大学紀要, 15, 17-27 (2015)
- 13) 田近一郎, 本多一彦, 杉江晶子, 森博:「IT 教育のためのプログラミングオンモバイル」, モバイル学会研究報告集, 11, 143-146 (2015)
- 14) 本多一彦, 田近一郎, 杉江晶子, 森博:「プログラミング・オン・モバイル」, メディア教育シンポジウム (2016/2/6)
- 15) 田近一郎, 本多一彦, 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 (4)」, 名古屋文理大学紀要, 17, 11-23 (2017)
- 16) モバイル C,
<https://itunes.apple.com/jp/app/%E3%83%A2%E3%83%90%E3%82%A4%E3%83%AB-c-c-c-%E3%82%B3%E3%83%B3%E3%83%91%E3%82%A4%E3%83%A9/id467393915?mt=8>より2018年10月20日検索
- 17) iOS Developer Program License Agreement,
<https://developer.apple.com/terms/> より2018年10月27日検索
- 18) アップル, 「iOS」アプリ開発言語制限を大幅緩和,
<http://ascii.jp/elem/000/000/553/553560/> より2018年10月27日検索
- 19) Touch Lua,
<https://itunes.apple.com/jp/app/touch-lua/id525273327?mt=8>より2018年10月27日検索
- 20) iOS 用の Python 2.7
App Store で閲覧可能だがダウンロード不可. 2018年10月27日検索
- 21) CodeToGo,
<https://itunes.apple.com/jp/app/codetogo/id382677229>より2018年10月27日検索
- 22) Xcode,
<https://developer.apple.com/jp/xcode/> より2018年10月27日検索
- 23) Xcode Version Information,
<https://trac.macports.org/wiki/XcodeVersionInfo> より2018年10月27日検索
- 24) New Features in Xcode 4,
https://developer.apple.com/library/archive/documentation/Xcode/Conceptual/WhatsNewXcode-Archive/Articles/xcode_4_0.html#//apple_ref/doc/uid/TP40016147-SW10より2018年10月27日検索
- 25) The LLVM Compiler Infrastructure,
<https://llvm.org/> より2018年10月27日検索
- 26) Clang: a C language family frontend for LLVM,
<https://clang.llvm.org/> より2018年10月27日検索
- 27) 大学院生のための LLVM,
<https://postd.cc/llvm-for-grad-students/> より2018年10月27日検索
- 28) Duktape,
<https://duktape.org/> より2018年10月27日検索
- 29) ScratchJr,
<https://itunes.apple.com/jp/app/scratchjr/id895485086>より2018年10月27日検索
- 30) @ IT:FAQ UML 篇 UML はオブジェクト指向か?
http://www.atmarkit.co.jp/im/carc/serial/dstylefaq/uml/uml2_2/uml2_2.html より2018年10月27日検索
- 31) ロバート・C・マーチン, 瀬谷啓介:「アジャイルソフトウェア開発の奥義 第2版オブジェクト指向開発の神髄と匠の技」, SB クリエイティブ, (2008)