

文学部におけるプログラミング教育の意義とその方法

—Squeak eToys を用いた論理的思考を養うためのプログラミング—

文学部 本村康哲

本稿は、文学部におけるプログラミング教育の意義と授業実践について述べたものである。この授業ではプログラミング言語の習得を目的とするのではなく、論理的に考えるためのスキルの獲得を狙っている。具体的な目標として、あらかじめ手順を記述し、検証する姿勢を養うことを目指す。プログラミング環境として、視覚的に操作が可能な Squeak の eToys を導入する。Java などと比較して、学生たちは最後までモチベーションを維持しながら積極的に授業にかかわり、論理的思考を身につけ、達成感を得ることが可能となる。

1. はじめに

この約 10 年で急速に普及してきたパーソナルコンピュータ（以下、PC）は、いまやコモディティ製品として多くの人々が利用する道具となった。大学においても電子メールとウェブサイト閲覧を中心とした利用はもはや日常的である。最近では、ほとんどの学生が大学入学までに PC に触れたことがある、あるいは自宅に PC を所有しているといった状況である。また、ウェブサイト閲覧はひとつの娯楽あるいは時間のすごし方としてもとらえられており、テレビと並ぶひとつのメディアとなりつつあることをうかがわせる。

その一方で、学生たちが PC を「思考の道具」「生産の道具」として利用できるようになってきたかという点、はなはだこころもとない。たとえば、ワードプロセッサでレポートや論文を作成する、ウェブサイト製作ツールでサイトを製作する、画像処理ソフトウェアで画像を作成するなどのコンテンツ作成に際して、ゴールまでの道筋を自ら探りあてて実行することが困難である。多くの学生はこちらが提示した道筋を辿るだけであり、自ら思考しながら手順を構築することを苦手としている傾向にある。

また、PC は特定用途専用の製品ではなく、作業にあたっては Microsoft Office をはじめとする多機能汎用のソフトウェアを導入して利用することになる。このようなソフトウェアの利用にはプログラミング言語による記述を必要としないが、期待する結果を予測・イメージしつつ、場面に応じて最適な手順を構築して操作することが要求される。しかしながら、PC の扱いが不慣れた学生のほとんどは、こういった手順を踏みながら一步一步目的に近づいていくための思考を苦手としている。

ソフトウェアの機能や操作方法を知ることでも必要ではあるが、道具を使って何かを生産する際には、目的とする作業の手順を組み立てて実行していくことが重要である。こういったことは、PC スキル（操作方法）の問題というよりも思考スキルの問題であり、PC を扱う以前に手続きが決定あるいは構築されなければならない。

そこで、本稿では上述のような論理的思考（一定のルールの下での手続き的思考）を養うために、文学部の専門科目「アルゴリズムとプログラミング」で Squeak eToys による授業実践を行った事例を紹介する。

2. プログラミング教育の意義

世の中の仕事のほとんどは手順を定めてから取り掛かる。つまりあらかじめ計画された「段取り」を順番に実行して仕事の目的を達成する。それを実現するまでのステップをひとつひとつ解決するために、制約条件下における一連の「手続き」をあらかじめ用意する。これはまさしくプログラム (pro-gram: あらかじめ記述されたもの) である。こういった手続きは、仕事の生産性を決定するものであり、PC の操作に限らず、あらゆる仕事の場面において重要である。

しかしながら、文学部における学生の多くは手続き的思考を極端に嫌う傾向にある。ものごとを順序立てて粘り強く考え続けるといった作業は、PC を操作する際に顕著に現れる。たとえば、ウェブサイト閲覧やメール送受信などの操作ではほとんど問題とならないが、生産的な仕事をする場合には、PC の操作方法に関する断片的知識を持っていたところでうまくしえるものではない。その原因は、PC の操作そのものよりも、達成すべき仕事のゴールまでの道筋が見えていないことに問題があるといえる。

コンピュータプログラミングを学ぶことは、一連の手続きを記述しコンピュータに実行させて検証することができる。こういったスキルを身につけることは、PC を使って仕事をす際だけでなく、大学での研究のみならず社会でのさまざまな仕事の現場で大きな生産性をあげるであろうことは想像に難くない。このため、将来コンピュータのエキスパートにならないまでも、プログラミングによって手続き的な思考を身につけることは大いに意味のあることであると考えられる。

3. 文学部におけるプログラミング教育の問題点

ところで、プログラミング教育を行う際には、その時代に普及している開発言語で行われることが多い。しかし、C や Java は文学部でプログラミング教育を実施するにはさまざまな問題をはらんでいる。実際、筆者はこういった言語を用いて教育を行ったことがあるが (表 1)、学生の多くは最後までモチベーションを維持することが困難であった。比較的扱いやすいといわれる簡易言語 (VBA など) においても同様の結果になると推測される。その理由として、以下のような事柄が考えられる。

(1) プログラムの実行にいたるまでの工数

C や Java などの言語は、ソースコードの編集、コンパイル、リンクといった複数の工程を経て実行可能となる。こういった作業の意味を理解して操作することは重要なことではあるが、初学者にとってはかなり煩瑣な作業であることは否めない。また、Eclipse の

ような統合開発環境（ソースコード編集、コンパイル、デバッグ、実行をひとつの画面で扱える）もあるが、その操作法を習得し慣れるまでに時間を要する。

(2) 膨大なライブラリ群

オブジェクト指向言語などは、多くのライブラリ（すでに作成されたプログラム群）を提供している。豊富なライブラリは、プログラミングに精通した者にとっては短期間でソフトウェア開発を可能としているが、前提として内容をよく理解している必要がある。そのため、多くのドキュメントを参照しなければならないが、これがプログラミングの初学者にとっては重荷となる。

(3) プログラミングパラダイム

プログラミングパラダイム（現実世界のモデル化の方法）はひとつの思想体系であり、抽象的な思考を要求される。考え方に馴染むまでにはかなりの時間がかかり、初学者にとってはハードルが高い。また、よく普及している言語のパラダイムには現実の問題に対応させるためにあいまいなところもあり、初学者に大きな混乱を生じることになる。

(4) 厳密な文法体系

プログラミング言語は人工言語であり、厳密な文法規則が存在する。ピリオドやコロンの打ち方をひとつ間違えるだけでまったく動作しなくなる。さらに、こういったケアレスミスを発見して修正するには、相当な集中力を要するため初学者には精神的な負担が大きい。学習者の多くがプログラムを実行させるまでに疲労困憊し、なかなか前に進まないと厭戦気分が漂い始め、投げ出してしまう。

表1 過去の「アルゴリズムとプログラミング」の内容

年度	プログラミング言語	開発環境	教科書
2002 ¹	C	TeraTerm 経由で UNIX の shell	『C 実践プログラミング』
2003	C	TeraTerm 経由で UNIX の shell	『C 実践プログラミング』
2004	Squeak (春)、Java (秋)	Squeak, Cpad for Java ²	『スクイークで遊ぼう』『やさしい Java』
2005	Java	Cpad for Java	『やさしい Java』
2006	Squeak	KotodamaOnSqueak-1.1.6	『Squeak で学ぶプログラミング』
2007	Squeak	KotodamaOnSqueak-1.1.6	『Squeak で学ぶプログラミング』

これらのハードルを越えてなおもプログラミングを習得したいというモチベーションを継続的に維持することは、ものづくりを志す理工学系の学生ならば可能であろう。しかしながら、文学部の学生が進む先にはコンピュータ・プログラミングとはほとんど縁がなく、上述

¹ 実際には履修登録がなかったために非開講であったが、自主講座として開講。

² エディタ、コンパイラなどが1画面で利用できる簡易統合環境。

<http://www.vector.co.jp/soft/win95/prog/se153698.html>

の困難を乗り越えてもなおプログラミングを習得したいというモチベーションを維持できる学生は皆無に等しい。

では、こういった学生たちがプログラミングを学ぶことが無意味かということ、そうではないと主張したい。プログラミングで学ぶことができるのは、役に立つソフトウェアを作成することだけではない。その過程でアルゴリズム（問題の解き方の過程）を学び、手続き的な思考をトレーニングすることができる。このため、文学部の学問を志す学生がプログラミングを学習する意義は大いにある。しかしながら、従来のプログラミング環境では、アルゴリズムを学ぶまでにほとんどが挫折を経験し、プログラミングは難しいもの、特殊な職種（プログラマ）だけが扱うもの、という考えにとどまってしまう。

そこで、上述の問題が顕在化しにくく、アルゴリズムに集中できるプログラミング環境として Squeak eToys を導入することにした。

4. Squeak eToys

Squeak 自体は SmallTalk を祖とするオブジェクト指向プログラミング言語であり、eToys は Squeak 上で動作可能なビジュアルプログラミング環境である。用意されたタイルを組み合わせるタイルスクリプティングによって作業を進めていく（図 1）。従来の言語的記号操作に比べると自由度も低く制限された環境ではあるが、初学者にとっては覚えることが少なく視覚的に操作可能なため、アルゴリズムの構築に集中しやすい。また、eToys 自体はプロトタイプベースオブジェクト指向であり、Java のようなクラスライブラリに関する知識がなくてもプログラミングが可能である。

前述の問題点と照らし合わせてみると、まず、(1)プログラム実行までの工数が非常に少ない。図 1 を参照されたい。プログラムの作成は、まずオブジェクトを右クリックすると出現するハロ³からビューアを選択し、スクリプティングのためのウィンドウを開く。そして、プログラムの編集はタイルをはめ込んでパラメータを設定していくだけである。実行するためには、時計マークのあるところをクリックして、「チクタク」にする。マウス操作の手数は多いが、視覚的な操作なので、GUI に慣れた最近の学生にとっては、比較的扱いやすいと考えられる

つぎに、(2)膨大なライブラリ群であるが、プロトタイプベースのため、クラスライブラリを使用せず、最初から自分でプログラムを作成できる。かつての BASIC がそうであったように、プログラミング言語としてはできることが限られているが、それがかえって全体像を把握しやすく、初学者にとってはプログラミング環境の世界観をイメージしやすい。新しいことを学び始める際に、全体像を俯瞰しやすいことは重要なことである。

(3)プログラミングパラダイムは、先にも述べたようにプロトタイプベースオブジェクト指

³ ハロ：オブジェクトの周囲に現れるボタン群。オブジェクト属性を定義できる。

向であり、また、ビジュアルプログラミング環境である。すでに用意されている部品や自分で描画したオブジェクトにスクリプトを貼り付けていく作業は、現実世界との対応がとりやすく、抽象的なパラダイム理解を必要としない。

(4)eToys の操作には文法体系 (ルール) は存在する。eToys 環境でできないことはマウス操作が拘束されており、ケアレスミスを未然に防止している。エラーメッセージもほとんど提示されることはない。こういったインターフェースは不自由を感じるとともに、できることの少なさを実感するであろう。しかしながら、自由度の高さを犠牲にしながらも扱いやすさを優先することによって、プログラミングのしきいを下げていることは重要である。

なお、Squeak eToys におけるスクリプトの制御構造は基本的に上から順番に実行され、一番下まで来ると再度上から実行するというように繰り返し構造 (ループ) が最初から組み込まれている。条件分岐はタイルで用意されている。これらの制御構造を組み合わせることによってゲームを作成したり、物理現象のシミュレーションさえも可能としている。

以上の検討から、Squeak eToys が初学者の環境として適していると考えられる。

なお、Squeak eToys は無料で配布されているが、授業で用いるプログラミング環境はフリーもしくはサイトライセンスによって無料で提供されていることが望ましい。なぜならば、プログラミングの学習は授業内だけで完結するものではなく、自宅などで継続して行われる必要があるため、手軽に自宅の PC で利用可能でなければならないからだ。また、学生にとっては授業料以外に少しでもコストを負担することは心理的抵抗が大きい。ましてや、プログラミングに対する強いモチベーションが望めない場合はなおさらである。

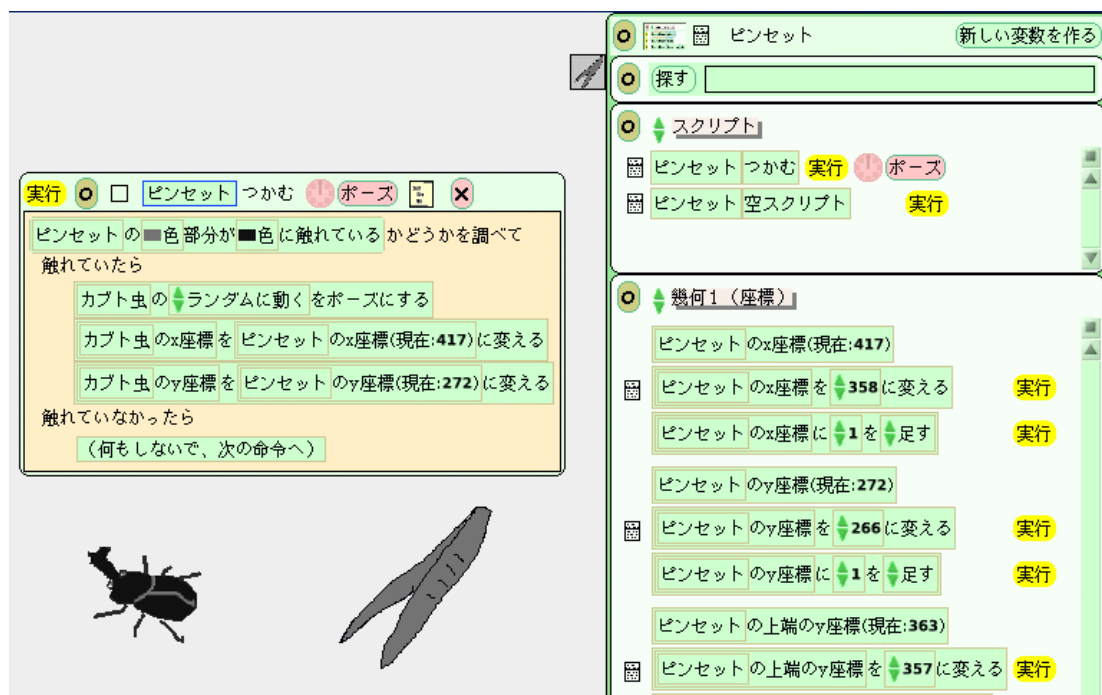


図1 Squeak eToys のタイルスクリプティング

5. 文学部専門科目「アルゴリズムとプログラミング」

5.1 科目の性格

この科目はもともと文学部の通年の必修科目として人文情報コースの学生向けに開講されていたが、2003年度からはオープン科目として文学部の全学生が履修可能となっている。

履修者は、ほぼ全員がプログラミングの経験を持たないといってもよい。こういった学生に対して、プログラミングの意義を説き、ソフトウェアを作成することの楽しさを伝えるとともに、コンピュータのふるまいと問題解決の手順を理解することを目的として、以下のような講義概要を提示している。

「プログラムがなければコンピュータはただの箱」といわれるように、コンピュータにはプログラムが欠かせない。プログラムとはコンピュータを動作させるための手順書であり、アルゴリズムとは手順そのものである。人間に意志を伝えるためには言葉や書き物が必要であるのと同じように、コンピュータに意志を伝えるためにはプログラムが必要なのである。

この授業では、オブジェクト指向言語 **Squeak Etoys** によるプログラミングをとおして、コンピュータを動作させるための手順（アルゴリズム）と方法（プログラミング）について講義と演習を行う。また、**Squeak** を用いたシミュレーションや計測・制御についても触れる。最終的にはコンピュータに対するより深い知識とプログラム作成技法を身に付けるとともに、論理的思考能力と問題解決能力を養うことを目標としている。

なお、練習課題が毎回3～4問提示されるので、必ず提出しなければならない。また、基本的な事項を積み上げながら学習を進めて行くため、一度でも欠席すると授業について行くのがつらくなる。このため、やむを得ない事情で欠席する場合は教科書を読んで自習しておくこと。

5.2 クラス

授業は第1学舎4号館4階「文学部情報演習室C」で行われる。PCが52台設置されており、学内LANに接続されている。2007年度の履修登録者は44名、総合人文学科インターディパートメント人文情報コースの学生が主体であり、ほぼ全員がプログラミングの初学者である。授業では毎回出欠確認を行い、可能な限り継続して授業に出席するように促している。出席率は、人文情報コースの学生にとっては必修科目のため平均して85.9%（＝（出席＋遅刻）／（出席＋遅刻＋欠席））と比較的高く、常時40名弱の学生が出席していることになる。

授業は講義と演習を半分ずつ実施した。テキストの章(**Project**)の最初には解説を行い、その後2～4問の練習課題を課した。練習課題は1週間後に提出期限を設けた。プログラミングの習得は継続して行わなければ効果がないが、毎回の課題が学生にとっては重荷のようである。しかし、全体を通しての提出率は78.8%（＝提出数／提出を課した課題の全数）であり、学生が熱心に取り組んでいる様子が伺える。



図2 授業風景

5.3 プログラミング環境

現在、Squeak eToys にはいくつかのバージョンがあり、日本語化も行われている。今回は、日本の Squeak コミュニティでは著名な慶応大学環境情報学部の大岩研究室⁴で無料配布されている”Squeak Ver.1.1.6 Kotodama Edition”を用いた。最初の授業で CEAS（後述）を通して環境を配布し、USB メモリにインストールさせる。学生は USB メモリを持っていれば、自宅や IT センターなど教室以外の場所でも学習可能である。

テキストには、プログラミング環境と同様に慶応大学の 大岩研究室で編集・配布されている『Squeak で学ぶプログラミング Vol.1』を用いた[1]。これは PDF 形式であり、ダウンロードして利用できる。

なお、これら教材の授業での使用に際しては、あらかじめ大岩研究室から使用許諾をいただいている。

5.4 CEAS

CEAS⁵は関西大学工学部の冬木研究室で開発された授業支援システムである。主に授業中に使用することを目的とした e-learning システムであり、資料配布、課題提示・収集、出欠管理、掲示板・チャットなどのコミュニケーション機能などを備えている。

授業では、CEAS の出欠管理機能で出欠を取り、資料配布機能で補足事項や考え方のヒントなどを適宜配布している。また、課題提示・収集は毎回の課題を課すことができ、学生はインターネットに接続されている PC があれば自宅からでも課題を提出できる。これはプログラミング学習のモチベーションを授業後に維持継続するために重要な機能である。なお、今回はコミュニケーション機能は用いなかったが、秋学期の作品製作プロジェクトでグループフォルダ機能を活用した。

⁴ <http://www.crew.sfc.keio.ac.jp/>

⁵ <http://www.atl.kansai-u.ac.jp/>

5.5 実施スケジュール

年間スケジュールを表 2 に示す。

春学期のはじめは、eToys の基本的な操作とスクリプティングの方法を解説し、環境に慣れることを目標とした。その後、eToys 特有のインタフェースであるオブジェクト、ハロ、ビューア、スクリプトといった基礎知識を習得させた。つぎに、場合分け（条件分岐）の考え方をフローチャートを使って解説し、つづいて変数と変数を使った場合分けの演習を行った。乱数を使ったプログラムの作成ができるように四則演算タイルや乱数タイルについても解説を行った。

秋学期は、変数を使って速度と加速度について説明し、車のアクセルとブレーキ、さらに自由落下のシミュレーションを行った。その後、1 グループ 4~5 名のグループに分かれてゲームを製作するプロジェクト活動に移行した。プロジェクトはテキストに沿って、企画書、仕様書、計画書、報告書、プレゼンテーションの順で作成させ、提出させた。授業時間外にも継続して協働作業を行うため、CEAS のグループフォルダを利用して、ファイルの共有を行わせた。

表 2 年間スケジュール (2007 年度)

	回	日	授業概要	提出物
春学期	1	04/06	ガイダンス	
	2	04/13	Project 1 : オブジェクト、ハロ	Project 1, 練習問題 2.1.1, 2.1.2, 2.1.3
	3	04/20	Project 2 : ビューア、スクリプト、命令	Project 2, 練習問題 2.2, 2.4, 2.7
	4	04/27	Project 3 : 場合分け	Project 3, 練習問題 3.1
	5	05/11	フローチャートの描き方(1)	フローチャート、練習問題 3.2, 3.3
	6	05/18	フローチャートの描き方(2)	練習問題 3.4, 3.5, 3.6
	7	05/25	Project 4 : 変数と値の変更	Project 4, 練習問題 4.1, 4.2, 4.3
	8	06/01	変数を使った場合分け	練習問題 4.4, 4.5, 4.6
	9	06/08	Project 5 : 他の変数を使った値の変更	Project 5, フローチャート
	10	06/15	2 つの変数を使った場合分け	練習問題 5.1, 5.2, 5.3
	11	06/22	演習	練習問題 5.4, 5.5, 5.6, 5.7
	12	06/29	Project 6 : 計算と乱数	Project 6, 練習問題 6.1, 6.2, 6.3
	13	07/06	演習	練習問題 6.4, 6.5
秋学期	14	09/21	Project 7 : 速度と変数	Project 7
	15	09/28	加速度 (加速と減速)	練習問題 7.1, 7.2
	16	10/05	演習	練習問題 7.3, 7.5
	17	10/12	グループ分け、企画書の作成	企画書
	18	10/19	仕様書の作成	仕様書
	19	10/26	作業手順書・スケジュールの作成	作業手順書、スケジュール
	20	11/09	実装作業 1	作業報告書 1
	21	11/16	実装作業 2	作業報告書 2
	22	11/30	実装作業 3	作業報告書 3、作品、スライド
	23	12/07	担当者出張のため休講	
	24	12/14	プレゼンテーション 1	発表評価 1
	25	01/11	プレゼンテーション 2	発表評価 2、アンケート

6. 授業アンケート

最後の授業の際に、以下のようなアンケートを実施した（表 3）。大きく分類して、作業手順、制御構造、テキスト、身についた能力について、学習者の主観で回答してもらった。なお、回答数は 35 であった。

表 3 アンケート項目

設問 1	Squeak の操作方法に慣れたのはどの時点ですか？
設問 2	Squeak のスクリプト作成に慣れたのはどの時点ですか？
設問 3	Squeak のスクリプトは上から順番に実行されることを理解していますか？
設問 4	Squeak のスクリプトを「チクタク」にすると、繰り返し実行されることを理解していますか？
設問 5	「場合分け処理」を使ったスクリプトを作成できますか？
設問 6	問題解決の「道筋」を考えるのに慣れたのはどの時点ですか？
設問 7	テキストはわかりやすかったですか？
設問 8	テキストを自学自習に利用しましたか？
設問 9	テキストに関する意見を書いてください。
設問 10	問題解決の道筋を考える能力は身につきましたか？
設問 11	報告書やディスカッションで説明などの表現力は身につきましたか？
設問 12	友人と相談したり一緒に協力して仕事をする能力は身につきましたか？
設問 13	その他、この授業を通して身についたと考えられる能力を書いてください。
設問 14	この授業を受けて、感じたことを書いてください。

6.1 作業手順

図 3 に作業手順に慣れるまでにかかった時間についての結果を示す。Squeak eToys 環境の操作方法（オブジェクト、ハロ、ビューアの扱い方）については、半数の学生が 1 ヶ月以内に慣れたと回答している。また、春学期が終わる 3 ヶ月後には 80% の学生が習熟している。このことから、Squeak eToys が非常に扱いやすい環境であることがいえる。

また、スクリプト作成（スクリプト、命令タイルの扱い方）については、春学期までに慣れたと回答した学生が 57% と、環境の操作に比べると大きく後退しているが、秋学期中には 89% が慣れている。タイルスクリプティングが簡単にスクリプト作成できるとはいえ、座標などの数値を扱う項目になると習熟度が落ちる傾向がうかがえる。

さらに、手順を考える段になると、秋学期を終えることになってようやく慣れたと回答する学生が 70% となり、習熟までに時間がかかっている。このことから文学部の学生が手続き的な思考（論理的思考）を苦手としていることがうかがえるが、この授業で少しは身につけることができたのではないだろうか。授業を終えてもなお慣れないと回答している学生が 30% もいることについては今後の課題としたい。

6.2 制御構造

図 4 に制御構造への理解についての結果を示す。スクリプトウィンドウの中で、どのような制御構造が使われているのかについては、授業中にも解説しており、フローチャートを描かせる課題や、スクリプト作成の課題も多く与えている。このため、ほとんどの学生が制御

構造について理解していることがうかがえる。しかしながら、条件分岐について 23%の学生が理解していないという結果となっており、これについては今後の対策が必要である。

6.3 テキスト

テキストに関する詳細な議論はここでは割愛するが、わかりやすいと答えた学生は 51%であり、どちらでもないと答えた学生が 37%とおおむねわかりやすいテキストであったといえる。特に対話形式の解説と図が多用されていたところは非常に好評であった。ただ、自由記述ではヒントの充実を望む声が多く、昨今の学生気質を反映している。

6.4 身についた能力

図 4 に身についた能力についての結果を示す。この授業の主眼は、問題解決に至るまでの手続き的思考を身につけることにあったが、これは 49%の学生が身についたと回答した。また、どちらでもないと回答した学生は 43%であったが、表 4 の自由記述からあわせて考えると、まだ自信を持って身についたとはいえないが、手続き的思考が重要であることを認識しているようだ。

表現力については、作品製作での報告書やグループワークでのディスカッションについて問うたものである。プログラミングは一種の表現記述であるが、それ以前にプログラムの記述以上に文章や会話での表現能力も重要である。このため、報告書やディスカッションでの表現能力向上を狙ったものである。この項目についても、42%の学生が身についたと解答している。

協調性は、グループワークが表現力と同様、プログラミングのスキル向上に副次的ながら重要な示唆を与えるものである。表 4 の自由記述にも現れているように、自分では気づかなかったことを他人から教えてもらったり、グループで作業することによって自分のプログラム記述に対する責任感も生じる。このことは、ドキュメントやプログラム記述の可読性の向上に寄与するものと考えられる。

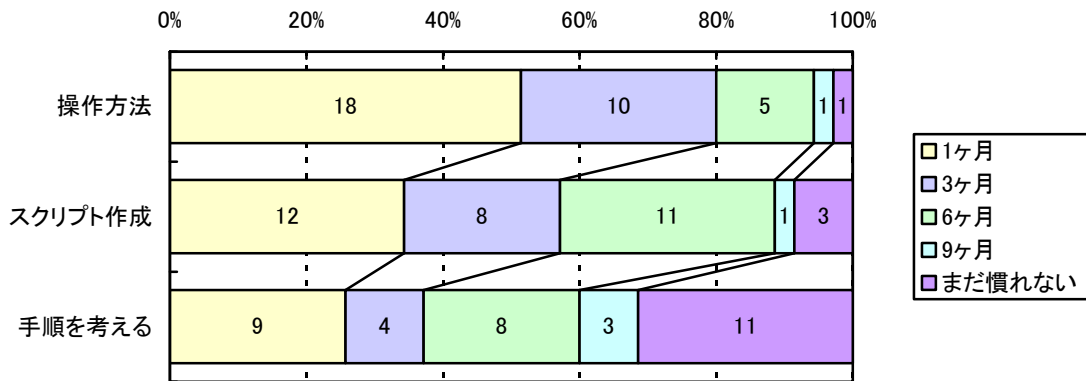


図3 作業手順に慣れるまでの時間

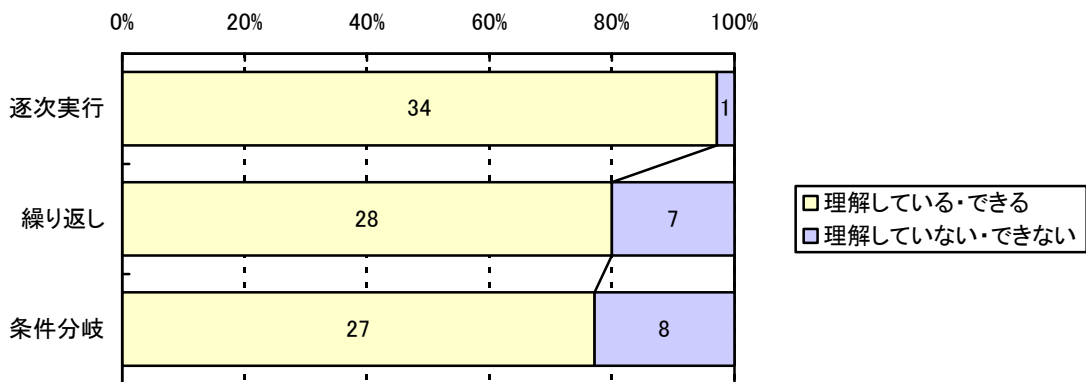


図4 プログラムの制御構造は理解しているか・扱えるか

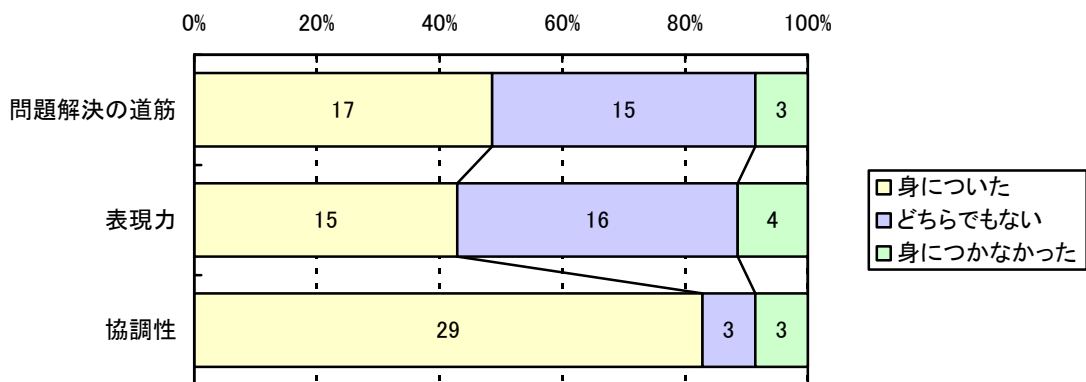


図5 どのような能力が身についたか

表 4 授業を終えての感想

1	コンピュータの動きの一つ一つがすごい量のプログラミングから成り立ってるんだなと感じました。
2	グループワークは、みんなの予定がそろわないので、個別にやらせてほしかった。
3	実際に自分でゲームが作れるので、やりがいがあった。課題に関しては、難しいものがあって苦勞したが、自分で考える力が身についたと思う。
4	私生活で簡単なゲームなどを見ると、そのゲームの構造はどうなっているのか、などふと考えるようになった。ひとつひとつ順番が大事だなと思った。
5	はじめのうちは使い方や構造がまったく理解できなかったが、だんだんと分かってくると楽しくなりました。完成したときにも達成感を味わえるしかなり面白く感じました。
6	試行錯誤しながら問題を解決していくことで、考え方に幅が出た気がします。
7	簡単にプログラムが作成できるのはいいけれど、できることが制限されすぎていて、やりたいことができない・逆に面倒で作りにくいと感じることが多かった。
8	本格的なプログラミングをやってみたかったです。どうもゲーム感覚から抜け出せませんでした。
9	パソコンというパーソナルなものを使いながらも、遊び心のある興味をそそるようなソフトを使って、一人でするのはなくみんなで作っていく、という点がすごく楽しかったしよかったですと思いました。
10	最初は、めんどくさいと思える授業でしたが、授業を受けているうちにどうしたら自分の理想とするスクリプトが作れるのだろうと必死になって考え、熱中していき自分がいました。最後の方は楽しめたと思います。
11	最初は難しいと思っていた問題も、色んなパターンを考えていく中で問題が解けたとき（出来上がったとき）、とても嬉しかったという達成感があった。
12	後半になってくると難しくなってきたが、完成させたときの喜びも大きかった。
13	難しくてなかなかスクリプトが完成せずいざまった時は辞めたくになりましたが、考え抜いてひらめいた時はとても感動しました。ゲームをつくるというのは、試行錯誤とアイデアと考える力といきずまってもあきらめない気持ち・・・など、たくさんの時間と労力がかかっているのだと実感しました。
14	普段の授業の程度のスクリプトだと全く問題はないのですが、自分たちで複雑なスクリプトを組むときにはスクイークは無理があると感じた。
15	一人作業もグループ作業もそれぞれの楽しさと大変さがあると感じた。
16	Squeak はスクリプトがわからなかったり複雑だったりした時はイライラしましたが（笑）、その他はとても楽しく学ぶことができました。
17	機械の性能について「難しい」「ややこしい」というだけの先入観を持っていたが、すべてにおいて、順序だててプログラミングされていることがわかった。
18	最後にグループでひとつの作品を作ったのですが、1人じゃ絶対にできないと改めて思いました。協力してやれば、けっこう難しいスクリプトもいろんな人のアイデアを組み入れるとできて楽しかったです。
19	企画し、自分たちの手で物をつくる楽しさが実感できた
20	いろいろなゲームを作ることで、いかに順序立てて物事を考えることが大事かを学べたように思います。
21	プログラミングは命令の集まりで、その命令を増やし、より複雑化することでゲームを使う側にとって面白みが増すものになる。命令の中身をみてみると、簡単だと思うものでも自分で考えて作るとなると結構難しいことがわかりました。しかし、ゲームを作る事だけでなく順序だてて物事を考える力やグループで協力する力など沢山の力がついたと思います。
22	最後のゲーム作りの段階で、班の4人のメンバーで協力して、電話をして相談したり、メールでのやり取りなどコミュニケーションを取れるよききっかけになりました。1つの課題を皆で完成させたという達成感があり、授業で発表があつてよかったです。
23	はじめは簡単なスクリプトさえ難しかったのですが、慣れて問題が解けるようになることが大変ではありましたが、とても勉強になりました。最後の共同作業では、時間をかけて凝ったゲームをつくり発表することで教材にはない工夫が凝らされた作品に触れることができ充実した授業であったと思います。一年間ありがとうございました。
24	先生の笑いながらの厳しいお言葉は勉強にもなりました。この授業がなければスクイークを知る機会などなかったと思うので、この授業がありよかったです。スクイーク自体そのものがゲームでもあるので授業を受けてて非常に楽しく取り組むことができました。良かったです。いろいろと質問などをしましたがきちんと対応してくれて良かったです。有難うございました。先生もお疲れ様でした。

7. おわりに

以上、文学部での「アルゴリズムとプログラミング」におけるプログラミング教育の実践事例を述べた。この授業では、従来の手続き型言語とは異なるビジュアルプログラミング環境 Squeak eToys を用いて、プロジェクト方式による作品製作をゴールとする手続き構築の過程を獲得することを目的とした。

問題解決の手順構築には、論理的かつ合理的な思考が必要である。しかしながら、従来のプログラミング学習は、問題解決以前に言語仕様を習得するまでの過程で学習者の多くが挫折を経験する。これでは、論理的思考の重要性やコンピュータに対する興味を削ぎ、物事に対してブラックボックス化ひいては思考停止につながる恐れがある。これでは、情報通信技術を用いたよりよい社会システムの構築や生産性の合理化などがごく一部の「専門家」の手にゆだねられてしまうことになるであろう。わが国の社会システムのさまざまな不具合や、コンピュータシステムの使いにくさわかりにくさはこのようなところに起因するのではないだろうか。分野はどうであれ、コンピュータシステムの大まかな概念を理解し、それを幅広い分野に適用することができる人材が増えれば、わが国の社会生活もより合理的なものになるであろうと期待している。そのために、思考トレーニングとしてのプログラミング教育が普及することを願ってやまない。

謝辞

授業に際して教材の使用を快くお許しいただきました慶応大学の岩元先生はじめ CreW Project のみなさまに感謝いたします。

参考文献

[1] CreW Project 編著『Squeak で学ぶプログラミング Vol.1 Kotodama Edition』, 2007.3.29.