

ショートペーパー

フィジカル・インタラクションを使った プログラミング学習のための教材開発と実践

有賀 妙子^{1,2,a)} 森 公一² 大下 福仁³ 角川 裕次¹ 増澤 利光¹

受付日 2015年3月6日, 採録日 2015年9月5日

概要: プログラミングは一般情報教育において重要な要素の1つである。これまで筆者らは、音センサ、赤外線センサなどを用い、インタラクティブなメディアコンテンツ（インストールーション）の制作を通して、デジタル技術を表現手段とする造形基礎学習と、プログラミング学習との双方の目的を持つ教育プログラムを構築し、その教材を開発した（SV [Sensory Vision] 教育プログラム）。受講生が制作するのは、センサからの入力に反応して動的に生成されるグラフィックスと人の行為がインタラクションをとる作品である。制作にはプログラミングの理解が必要になるが、この教育プログラムでは、サンプルプログラムを読み解き、模倣することから始めて再生産、そして創作する方法で理解を進めた。今までの実践研究において一定の成果はあったが、プログラミング基礎要素の理解が十分ではないという問題点があった。そこで新たに本研究では、プログラミング学習を強化する目的で、主に自習に使用する Web 教材を開発した。この教材は、SV 教育プログラムとは独立した一般プログラミング学習の場においても活用できる。この教材を適用して、(1) 独立したプログラミング学習と、(2) 教材を適用した SV 教育プログラムの2つの教育実践を行い、従来の学習と比較した。本教材を使った学習実践が、プログラミングの基礎要素の理解に関して、従来の学習と同等以上の結果を示すことを確認した。

キーワード: プログラミング教育, インタラクション, Web 教材

On the Design of Learning Materials for Supporting Physical Interactive Programming

TAEKO ARIGA^{1,2,a)} KOICHI MORI² FUKUHITO OOSHITA³ HIROTSUGU KAKUGAWA¹
TOSHIMITSU MASUZAWA¹

Received: March 6, 2015, Accepted: September 5, 2015

Abstract: Programming is a significant element in informatics education for non-major students. In our previous study, we developed an educational program in which an interactive installation is created using a sound sensor and an infrared sensor as a project, which is called SV [Sensory Vision] program. It aimed at two points in learning: basic art and design using digital technologies as an expressive way, and computer programming. Students create interactive motion graphics that respond to audience's behavior. To create an interactive media artwork needs learning of computer programs. In the previous research, students read sample programs, imitated them as a starting point of learning, and then made reproduction. This way led students to create their works. However, it is not enough to understand programming basics. In order to enforce this point, we develop Web materials for learning them. The materials can be applied to a programming course isolated from SV program. We organized two courses using the Web materials: (1) a study meeting for a basic programming, (2) SV program. This paper describes the materials and results of practices in two courses as compared with normal courses without the developed Web materials.

Keywords: programming education, interaction, Web learning material

1. はじめに

人の創造性や知性を、コンピュータを使って実現するには、ユーザとしてコンピュータを操作するだけでなく、作り手としてプログラミングを学ぶことが重要で、それが可能性を広げる。プログラミング学習により、コンピュータがどう働くかといった基礎的な知識に加え、複雑な考えを単純な問題に分解して考える力、系統的に推論する力などを習得できる。そして、その考え方やスキルは理工学系の学生だけでなく、人文科学分野の学生にも必要である。多くの一般向けプログラミング教育の手法が提案、研究されているが、その1つが物理的な「もの」を用いて、それをプログラムで制御することで、プログラミング学習の楽しさや効果をあげようという工夫である。たとえば、Lego Mindstorms [1] や PicoCriket [2] があるが、これらは主たる対象が子供であり、本格的な制作に応用するには発展性に欠ける。

筆者らは、音センサ、赤外線センサなどを用い、インタラクティブなメディア造形作品（インストールーション）の制作を通して、デジタル技術を表現の手段とする造形学習と、プログラミング学習との双方の目的を持つ教育プログラムを構築し、その教材を開発した（センサリービジョン [SV] 教育プログラム）[3], [4]。SV 教育プログラムの中で受講生が制作するのは、センサからの入力に反応して動的に生成されるグラフィックスと人の行為がインタラクションをとることを体験する作品である。I/O モジュールやセンサ類を含むツールキット、展示環境などを開発し、教育実践に適用してきたが、それに加え、プログラミング基礎要素の学習面を強化するため、主に自習に使用する教材を開発した。その学習の対象となる基礎要素の内容を表 1 に示す。

2. 本研究の目的

SV 教育プログラムでは、センサによって検知する人の行為に応じ、動的にグラフィックスを生成するプログラムの作成を行い、プログラムの実行結果を手取るように、確認しながら、プログラミングを理解する。プログラミング環境には Processing を使った。Processing はグラフィックスの描画やアニメーションを容易に記述できる機能を持ち、プロトタイプ的な作品にとどまらず、実用的な作品にも

表 1 学習の対象となるプログラミング基礎要素

Table 1 Elements of basic programming as targets.

学習項目	Computer Science Curricula 2013 での対応する箇所
変数, データ型	SDF, Fundamental programming concepts, Core-Tier1
演算子, 代入	
条件分岐, 繰返し	
シンプルな入出力	
関数と引数渡し	
再帰の概念	SDF, Data structures, Core-Tier1
配列	
クラスの定義(フィールド, メソッド, コンストラクタ)	PL, Object-oriented programming, Core-Tier1
SDF と PL は Computer Science Curricula 2013 の Knowledge Area(KA)のひとつ。 SDF: Software Development Fundamentals, PL: Programming Language	

利用できるという長所がある。Processing は Java の上に開発されたシステムで、Java のコードをそのまま使えるのに加えて、描画のための関数群を用意しており、単純なプログラムを書くにも、多くの文法要素を理解しなければならない問題は解消されている。また、音声や画像処理などの機能が開発されており、拡張性、応用性が高い。

しかし、テキストでコーディングするという伝統的なプログラミング言語の範疇にあり、ビジュアルプログラミング環境と比較すると、「学ばずともプログラムできる」という簡便さはない。そこで、コンピュータ科学を専門としない学生を対象とすることを考慮し、模倣する → 再生産する → 創作するというフェーズを通して理解を進められるように、インタラクション実現の出発点となるサンプルプログラムのライブラリを開発した。プログラミング学習の観点から教育実践を考察した結果、8 割の学生はプログラミングに対する主観的理解度の認識を高めることができ、プログラムを表現手法の 1 つとして積極的にとらえる認識を持てた [4]。

SV 教育プログラム内では、十分な時間を直接的なプログラミングの説明にあてられないこともあり、上で述べたようにサンプルを出発点としてプログラム制作を進めた。しかし、それだけではプログラミング教育の観点からは十分でなかった。プログラミング基礎の理解がなくても、「作品」としてのプログラムを作ることはできる。模倣、再生しているが、模倣したサンプルに含まれない基礎要素は学ばないままになってしまう。Processing 自体が、プログラミングの基礎知識がなくても、グラフィックスを描くことができるプログラミング環境であるため、理解のないまま、とりあえずは動けばいいという傾向が増長される。

基礎要素を理解していれば、自作プログラムの修正、発展も容易であり、別の機会での問題解決にプログラムを有効に適用できる。この点を解決し、Processing で描画することと、プログラミング基礎要素の学びとを連動させられるような学習教材を開発するのが、本研究の目的である。具体的内容を次章で説明する。

¹ 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology,
Osaka University, Suita, Osaka 565-0871, Japan

² 同志社女子大学情報メディア学科
Department of Information and Media, Doshisha Women's
College of Liberal Arts, Kyotanabe, Kyoto 610-0395, Japan

³ 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of
Science and Technology, Ikoma, Nara 630-0192, Japan

a) tariga@dwc.doshisha.ac.jp

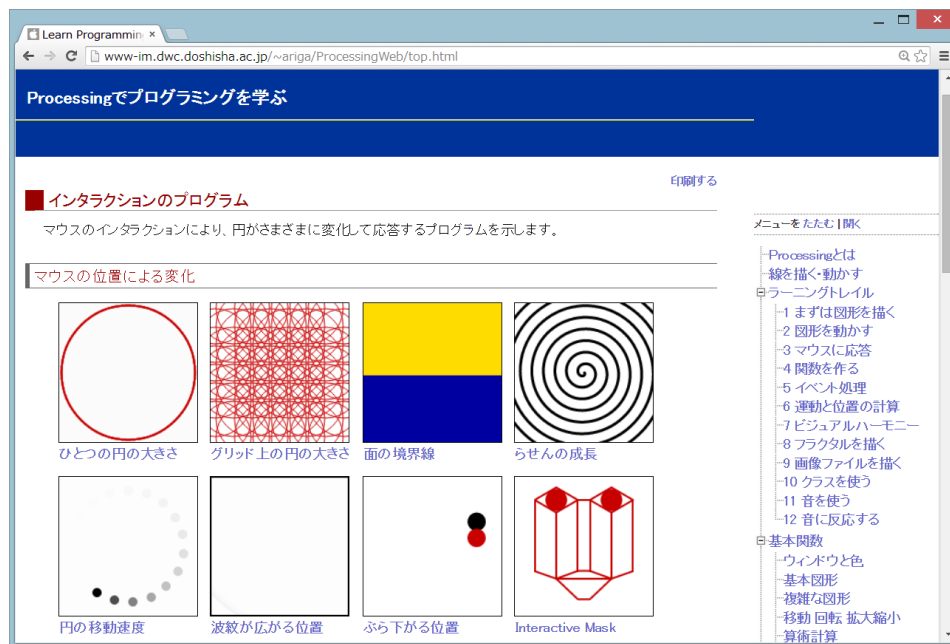


図 1 プログラミング学習用 Web コンテンツ

Fig. 1 Web contents for learning programming.

3. ビジュアルな表現を作り出すプログラミング学習教材

3.1 教材の位置づけ

本研究では、不可欠なプログラミングの基礎要素を学びながら、かつインタラクティブな環境で実行するプログラムを念頭に、モーショングラフィクスを段階を追って作る教材を Web コンテンツとして開発した (図 1)。パソコン内蔵のマイクの音声に反応するインタラクティブなグラフィックスを制作することを終着点に、プログラミング基礎教育で理解すべき点を教材化した。これを SV 教育プログラムの実践において、プログラミングを理解するために自習教材として使用する。加えて、この教材は、SV 教育プログラムとは独立した一般プログラミング教育の場においても活用できると考えている。SV ツールキットのようなセンサデバイスのない場面においても、たとえばパソコンの内蔵マイクを用いることにより、マウスやキーボード以外のインタラクションを持つモーショングラフィックスを制作することで、学びのモチベーションを高められる。

3.2 教材の構成と内容

教材は、3つの学びの流れを提供する。

- ラーニングトレイル (円を1つ描くところから始め、それを動かす、反応させるという流れの中で、モーショングラフィックスのプログラム作成を学ぶ)
- 基本文法知識を学ぶ
- サンプルプログラムから学ぶ

教材全体はこれら3つの中核部分に加え、導入部、Processing 基本関数のリファレンスの解説部からなる (図 2)。

Web の特性を活かして、各部分には相互にリンクが張られ、3つの学びの流れは相互に行き来できる。プログラムのソースコードを提示する部分では、コードとその解説だけではなく、Web ページ上でプログラムがインタラクティブに反応して動くようにしてある。つまり、Processing を起動し、ソースコードをコピーして実行しなくても、Web ブラウザ上で、直接実行結果をテストできる。

プログラミングの初学者は、導入部からラーニングトレイルへと進み、その中に出てくる基本文法項目についての解説を途中読みながら、音に反応するインタラクティブなグラフィックスを制作する最終部分に至る。「基本文法知識を学ぶ」流れは、プログラミング文法の基本項目の全体を見通すためのもので、プログラミング入門書と同等の構成である。ラーニングトレイルの中からリンクが張られており、学習者は必要に応じて参照する。それだけでなく、ラーニングトレイルでの一連の学びが終わった後に、「基本文法知識を学ぶ」流れを追うことで、基本文法知識全体の理解を整理できる。「サンプルプログラムから学ぶ」流れでは、SV 教育プログラムで開発した運動のサンプルプログラムの解説に加え、インタラクションを含んだプログラムのサンプルとその解説を示している。作品制作の企画段階で、これらサンプルの実行を見ながら、あわせてソースを理解することで、SV 教育プログラムでの作品作りのアイデアをつかむことを促す。

本研究がプログラミング学習の基礎要素と考え、教材に含めた内容を表 1 に示した。ACM と IEEE が共同で定めた Computer Science Curricula 2013 [5] の中で、これらがどのように扱われているかをあわせて示した。Software Development Fundamentals (SDF) 内の知識ユニット Fun-

導入	ラーニングトレイル	基本文法知識	サンプルプログラム
1 Processing とは 2 線を描く・動かす	1 まずは図形を描く 2 図形を動かす 3 マウスに反応 4 関数を作る 5 イベント処理 6 運動と位置の計算 7 ビジュアル・ハーモニー 8 フラクタルを描く 9 画像ファイルを描く 10 クラスを使う 11 音を使う 12 音に反応する	1 変数、式、文 2 算術演算子 3 条件分岐と繰返し 4 配列 5 関数の定義 6 クラスとオブジェクト 7 クラスの定義 8 クラスの拡張	1 運動 ・一方向 ・往復 ・振動 2 インタクション ・マウスの位置 ・マウスの移動速度 ・閾値によるスイッチ
基本関数			
1 ウィンドウと色 2 基本図形 3 移動回転拡大縮小 4 算術計算 5 タイポグラフィ 6 画像データ			

図 2 Web 教材の構成

Fig. 2 Structure of Web learning material.

damental Programming Concepts の Core-Tier1 (基礎段階で学ぶ内容) として提案されている全項目を含み、それを基本として、複数のグラフィックス要素を扱う際に欠かせない配列とクラスの基礎概念を加えた。クラスとオブジェクトの概念は、同じ振舞いをするグラフィックス要素を複数扱う際に必要となるのに加え、Processing が提供するサンプルでも使われており、その理解に必要となる。

アメリカ National Research Council はその提言 [6] で、Fluent with Information Technology (FIT) が求めるプログラミング経験として、条件分岐、繰返し、関数 (functional decomposition, functional abstraction) をあげており、FIT の要求を満たす内容となっている。

3.3 ラーニングトレイル部の詳細

センサからの入力に応じてグラフィックスを動かすプログラムを作成することが、3つの学びの流れの中で中核となるラーニングトレイル部の目的である。そのようなプログラムを書くために学ばべき要素として、表 1 に示した基礎要素に加え、基本描画、位置の計算、反応 (イベント処理) を含む。また SV 教育プログラムへの適用を考慮し、インタラクティブな反応に音声入力処理を加えた。サンプルを簡潔にするため、素材グラフィックスには単純な円を使った。項目ごとに取り上げる題材と、そこで学ぶプログラミング基礎項目、学習の目的について述べる。

(1) まずは図形を描く

「円を 1 つ描く」プログラムを通して、関数、引数渡しを理解する。次に「円を複数描く」プログラムに進み、規則性を発見し、変数を使って変更を容易にする工夫をする。さらに繰返しを理解し、プログラムを見通しよくしていく。あわせてデータ型、式、代入、演算について学ぶ。

(2) 図形を動かす

図形の大きさ、位置を変えることで、プログラムの実行に動きを作り出す。アニメーションを繰り返すため、条件分

岐を学び、応用できるようにする。

(3) マウスに反応

マウスに追従する図形を描くプログラムを通して、位置やその変化量に対応する処理を理解する。イージングの処理を理解し、その効果により、動きの印象が大きく変わることを知り、自分のプログラムに適用できるようにする。

(4) 関数をつくる

同心円を描く関数の定義を通して、関数による抽象化の概念を理解する。新しい関数を定義し、使えるようにする。

(5) イベント処理

マウスクリック、ドラッグ、キーボード入力に反応するため、イベント処理関数を定義する。SV ツールキットを使う教育プログラムでは、マウスやキーボード入力は不要であるが、一般の Processing の学習のことを考慮して、この項目を設けた。

(6) 運動と位置の計算

基本的な物理の計算式を使い、加速運動、ばねの運動をシミュレーションする。数式による位置の計算が新しい動きを生み出すことを理解し、自分のプログラムに応用できるようにする。

(7) ビジュアル・ハーモニー

古典的のモーショングラフィックスの作品である Whitney の Digital Harmony [7] を再現するプログラムを作成する。これを通して、シンプルな数式の表現が、驚きの効果を生み出す体験をする。段階を踏んだ作成の途中で、複数の円の位置を保持するために配列について理解する。

(8) フラクタルを描く

簡単なフラクタル画像を描くプログラムを書くことを通して、再帰処理を学ぶ。

(9) 画像ファイルの描画

あらかじめ用意した画像ファイルを描画することで、パラパラアニメを実現する。1,000 を超えるような多数のオブジェクトを描く場合、画像ファイルを使う方が描画処理が

早い。ファイルからのデータ入力とともに、画像ファイルによる描画方法を理解する。

(10) クラスを使う

クラスからオブジェクトを生成して使うことを学ぶ。UI部品としてのスライダクラスを取り上げ、そのAPIからコンストラクタ、メソッドについて知る。スライダクラスのオブジェクトを生成し、フラクタルの描画をスライダで操作するプログラムを作成する。自分でクラスを定義することを理解するとともに、Processingを拡張するために開発されているライブラリを使う準備となる。

(11) 音を使う

音声を扱うための拡張ライブラリ Minim を使い、あらかじめ準備した音声ファイルの再生と、その音量に応答して動くグラフィックスを描くプログラムの作成を行う。クラスのAPIから使い方を知って、利用することを学ぶ。

(12) 音に反応する

外部から得る値を描画の位置や大きさに反映させる実例として、パソコン内蔵マイクの音を入力して、それをグラフィックスの大きさの値に使うプログラムを取り上げる。入力値の範囲を、大きさや位置に使う別の範囲に割り当てる（マップする）ことで、入力値を大きさや位置に適切に反映させる方法を理解し、使えるようにする。SVツールキットを使う場合、センサからの入力値はセンサの種類や周辺環境によってさまざまなので、その値に応じて調整して入力値を使うことを知る。

4. 教育実践

4.1 Web教材を使ったProcessing勉強会

プログラミングの自習教材として開発したWeb教材 (<http://www-im.dwc.doshisha.ac.jp/~ariga/ProcessingWeb/top.html>で公開) を使ったプログラミング学習が、プログラミング基礎の学習に効果を生むかを、通常のJavaによるプログラミング授業と比較することで検証した。

Processingによる学習は2013年秋学期に自主的勉強会の形でを行い、プログラミング初学者8名(4年次:1名, 3年次:1名, 2年次6名)が参加した。週1回90分のセッションを15回行った。Web教材のラーニングトレイルを読んできた後、勉強会で質問に答え、演習問題のプログラムを作成した。一方、Java言語による学習は、2013年と2014年春学期の入門プログラミング科目で週1回90分2コマの授業を15週実施した。受講生は2013年19名, 2014年36名であったが、アンケート未回答、クイズを2回以上欠席した学生、Processingの勉強会に参加した学生(2014年の3名)を除き、2013年18名(4年次:1名, 3年次:2名, 2年次:15名), 2014年27名(3年次:7名, 2年次:20名)を比較の対象とする。いずれも情報メディア学科の学生を対象に実施した。

どちらの教育実践でも15週のクラスセッションの中で、

```
次のコードの後ろではaの値はいくつになりますか
int x = 50;
int y = 100;
int a = 0;
if (x >= 100 && y >= 100) {
    a = x+5;
} else {
    a = x-5;
}
```

図3 クイズの例(条件分岐)

Fig. 3 Sample of quiz (if-else).

基礎知識を問うクイズを5回実施し、また学習開始前と後で基礎知識への理解認識の程度をアンケートにより回答してもらった。クイズは、変数とデータ型、繰返し、条件分岐、配列、メソッド(Processingでは関数)の5つの基本的項目の理解を問う問題を、学習の進行に合わせて実施した。問題は、Java SE Bronze Oracle Certification Exam.で出題される内容と同等レベルとし、文法的な正誤を問う、プログラム実行時の変数の値を問う、コードの断片を記入する形の5-10の質問からなる。図3は、条件分岐の問題の一例である。

プログラミング基礎要素に対する理解の認識については、クイズで問うた変数、データ型、条件分岐、繰返し、配列、引数の6項目に対して、「よく理解している」から「まったく理解していない」の4段階で回答を求めた。学習終了後のアンケートでは、プログラミングに対する自己効力感(self-efficacy)についても尋ね、加えて「学習を通して何を学んだか」に対して自由に記入してもらった。自己効力感自分のプログラミング遂行能力に対する予測を問うもので、次の7項目に対して、「まったく自信がない」から「おおいに自信がある」の7段階で回答を求めた。

- (1) 文法的に正しいProcessing (Java) のプログラムを書ける。
- (2) 自分が書いたプログラムなら多少長くて複雑でもデバッグできる(エラーを直せる)。
- (3) 解説サイトや本、リファレンスがあれば、プログラムの課題を完成できる。
- (4) 私が詰まったとき、誰か助けてくれれば、課題を完成できる。
- (5) プログラムを完成させる十分な時間があれば、課題を完成できる。
- (6) 詰まったときでも、そのままプログラムの課題を続けられれば、解決する方法を見つけられる。
- (7) ソースコードを見て、処理がどう進むかを頭で追って考えられる。

自己効力感 Bandura [8] が提唱した概念で、具体的な目標をなしとげられる力があるという感覚を指す。教育やビジネスの幅広い分野で自己効力感と学習の達成度や意欲などとの関係が研究され、自己効力の信念が学業課題に対する努力とねばり強さを強化するという結果が報告

表 2 Processing 勉強会と Java 授業との比較

Table 2 Comparison between Processing study meeting and Java courses.

	Processing 勉強会*5 (N=8)	Java 授業*5		勉強会と Java 授業の値のウェルチの t 検定*6	
		2014 N=27	2013 N=18	2014	2013
(1) クイズの平均(各人の平均のクラス平均)*1 と中央値	6.3 (SE: 0.90) 7.0	4.5 (SE: 0.35) 4.6	4.1 (SE: 0.46) 3.8	t(9)= 1.87 p=0.09	t(11)= 2.14 p=0.06
(2) 基礎項目の理解認識の変化値のクラス平均*2 と中央値	6.9 (SE: 1.3) 6.5	5.9 (SE: 0.50) 6.0	6.1 (SE: 1.1) 5.0	t(9)= 0.71 p=0.49	t(16)= 0.44 p=0.67
(3) 学習後の基礎項目の理解認識のクラス平均*3 と中央値	3.2 (SE: 0.21) 3.2	2.8 (SE: 0.079) 2.7	2.8 (SE: 0.11) 2.8	t(9)= 1.74 p=0.12	t(11)= 1.58 p=0.14
(4) 自己効力感の平均(各人の平均のクラス平均)*4 と中央値	3.7 (SE: 0.33) 3.6	3.5 (SE: 0.21) 3.7	3.6 (SE: 0.21) 3.4	t(13)= 0.58 p=0.57	t(13)= 0.28 p=0.79

*1 各クイズの満点を 10 点に標準化し、それを各人ごとに平均した結果をクラス全体で平均した値。

*2 理解認識は 4 段階のリッカート尺度で回答された値 (4: よく理解している, 3: まあ理解している, 2: あまり理解していない, 1: まったく理解していない)。6 つの項目の回答を合計、学習前後の差のクラス平均。

*3 学習後の理解認識の回答値の各人の平均をクラス全体で平均した値。

*4 自己効力感に関する質問に、7 段階のリッカート尺度で回答された値を各人で平均した値のクラス全体での平均。7: おおいに自信がある, 6: けっこう自信がある, 5: まあ自信がある, 4: 50/50, 3: 少し自信がある, 2: ほとんど自信がない, 1: まったく自信がない。

*5 上段: 平均値 (標準誤差), 下段: 中央値

*6 リッカート尺度データの解析に関して、t 検定とマン・ホイットニー・U テストの検定値に差はないという de Winter らの研究報告[12]があり、本論文では両群の傾向を示すのに有用と考え、平均値の t 検定を示した。

されている。プログラミング教育においても、多くの研究があり、Ramalingam らは、プログラミング学習に関する Self-Efficacy Scale を考案し [9]、プログラミング教育の効果に関する研究で使われている (たとえば、文献 [10], [11])。Ramalingam らの Self-Efficacy Scale は 32 項目からなるが、学習内容や言語環境にまったく依存しない上記の 7 項目を選択し、学習終了後アンケートに使用した。

表 2 に、Processing 勉強会と Java プログラミング授業でのクイズ成績 (1)、プログラミング基礎要素に対する理解認識の学習前後での変化 (2)、学習後の値 (3)、自己効力感 (4) のクラス平均値と中央値を示した。さらに、個々人の認識や自信について詳しく見るために、「何を学んだか」に対する自由記入の回答から、コメントを切り出し、8 つのカテゴリに分類した (表 3)。

表 3 自由記入コメントの Processing 勉強会と Java 授業との比較

Table 3 Comparison of free-writing comments between SV workshop and Java courses.

	Processing 勉強会	Java 授業
自分で考えて作れる、工夫して応用できる。	—	6 (13.3%)
達成感があった	—	6 (13.3%)
楽しかった	5 (62.5%)	-
問題を冷静に分析できる。	—	12 (26.7%)
処理の流れを順序立てて考えられる。	—	4 (8.9%)
プログラミングの基礎、言語文法を理解した。	6 (75.0%)	27 (60%)
今後の学習に意欲がわいた。	5 (62.5%)	4 (8.9%)
プログラミングは難しい。	1 (12.5%)	4 (8.9%)
その他	・構造をしっかり把握していないとわからない。 ・プログラムには何通りもの書き方がある。 ・簡単な動作にも長いコードが要る。	

コメント総数: Processing 勉強会 18, Java 授業 66

括弧内の数字は、そのカテゴリのコメントを書いた人の割合

表 4 SV ワークショップの流れ

Table 4 Schedule of SV workshop.

1 週	導入・準備(目的の理解, ソフトウェアのインストール)
2	チュートリアル 1(プログラム動作確認, 入力実験)
3	チュートリアル 2(行為とグラフィックスとの関係性をセンサを変えて実験, 関数, 変数, 演算の理解)
4	チュートリアル 3(グラフィックスの大きさ, 位置を変化)
5	チュートリアル 4(音にตอบสนองする Interactive Mask[インタラクティブに仮面の表情が変化]の制作演習)
6	作品構想(行為と感覚のエクササイズ)
7	作品構想(動きのシナリオを作成)
8	制作
9	グループごとにプログラミングに関してのアドバイスをしながら進める
10	
11	プレゼンテーション

表 5 SV 教育プログラム前後でのプログラミング理解に対する認識の変化

Table 5 Change in understanding of programming before and after the workshop.

変化	Web 教材あり 2014 年 (N=9)			Web 教材なし 2008+2009 年 (N=49)		
増加	8 人 (89%)	4 以上	5 (56%)	39 人 (80%)	4 以上	6 (12%)
		3 増加	1 (11%)		3 増加	9 (18%)
		2 増加	2 (22%)		2 増加	11 (22%)
		1 増加	0		1 増加	13 (27%)
変化なし	1 人 (11%)			4 人 (8%)		
減少	0			6 人 (12%, -1: 5 人, -2: 1 人)		

・2014 年はアンケートに回答した 9 名を対象とした。

4.2 Web 教材を使った SV 教育プログラムの実践

SV 教育プログラムを、表 4 に示すような流れの 11 週のワークショップ形式（週 1 回 90 分のセッション）で 2014 年に実施した。Web 教材をプログラミングの自習に使い、チュートリアルでも参照した。受講生は 10 名（大学 3 年次：4 名，大学 1 年次：3 名，高校 3 年次：1 名，高校 2 年次：2 名）で、2 名ずつがペアとなって制作を行った。

今までに報告した 2008 年と 2009 年の SV 教育プログラムの実践 [4] では Processing 学習用の Web 教材はまだなく、個別にサンプルプログラムを実行しながらコードを参照することで、プログラミングの理解を進めた。これらの Web 教材なしの実践とプログラミング学習 Web 教材を導入した 2014 年の実践とを比較するため、「プログラムを書くことについての知識量や理解度はどの程度か」を 10 段階で学習前後に問うた結果を表 5 に示す。

5. 考察

5.1 本研究での実践結果の解釈

本研究で開発した Web 教材を使った Processing 勉強会と Java プログラミング授業を、表 2 に示した結果から比較する。クイズ成績、プログラミング基礎要素に対する理解認識、自己効力感ともに、Processing 勉強会は Java 言語による学習に対して同等以上の結果であった。両者の平均値に対するウェルチの t 検定では、クイズ平均値は Processing 勉強会の方が高い有意傾向が見られ、クイズで質問しているような基礎項目の理解は、Web 教材を使った Processing 学習でも、Java 授業と同程度以上に得られることを確認した。統計的には平均値に有意差が認められないが、どの平均値も Processing 勉強会の方が高い。これは人数が少ないことで、分からない部分をそのままにせずにとどまって考え、個々のペースで学習を進めたことも一因であると考えられる。Java プログラミング授業は、教師が説明し、学生は一部クラス内で、一部は宿題として演習を行う。それに対し、Processing 勉強会は事前に Web サイトを自習してきて、セッション内では、多くの時間を個別の演習に費やす。学習に使っている言語環境ではなく、学習のスタイルの違いが影響していると考えられる。

また、自由記入のコメントを比較する（表 3）と、Java 授業では「達成感があった」とのコメント（13.3%）があるが、Processing 勉強会では達成感という言葉は使われず、代わりに「楽しかった」というコメントが現れる（62.5%）。また「問題を冷静に分析できる」「処理の流れを順序だてて考えられる」というコメントは、Java 授業のみに現れる。Processing 勉強会では楽しく主体的に学べており、基礎項目については同等以上の理解に至っているものの、プログラミングの解説や演習問題に費やす時間が Java 授業に比べ短いこともあり、プログラミングの深い理解を得たという認識にまでは踏み込めていないことが示唆される。

一方、「インタラクティブな作品作りに使える」といった今後の学習への意欲を表す記述が Processing 勉強会では多い。本教材が、音に反応するモーショングラフィックスという絞られたテーマでの学びを誘導し、その延長線上での展開をイメージしやすいことを反映している、今後プログラミングを自分の問題解決手段の 1 つとして主体的に選択できるという点で、有効に働いている。

Web 教材を導入した SV 教育プログラムの実践（2014 年）と以前の実践（2008 年と 2009 年）とを比較した（表 5）。プログラミングに対する主観的理解認識の学習前後での変化は Web 教材を適用した方が大きかった。3 以上増加したと認識した割合が 2014 年は 67%であるのに対し、以前の実施では 30%で、自信につながる主観的理解認識の変化が大きい参加者が多かった。SV 教育プログラムにおいて、自習用 Web 教材を導入することで、プログラミング基礎要素の学習を強化する目的に対して、効果をあげたと判断できる。

実際、2014 年の実践では学習の場で、学生が構想した動きを実現するために、プログラムを書く段階において、何度も Web 教材を参照する場面が見られた。たとえば、サンプルプログラムを修正しながら意図する動きを探る段階で、Web 教材上で元にしたサンプルの解説や動きを確認する、文法の不明点を確認するなど、3 つの中核部分が互いにリンクする Web 教材の特徴が有効に活用されていた。また、Web 教材を学生同士、講師と学生が共有していることで、説明をする、教えあうプロセスが大変やりやすかった。これは Web ページ上でサンプルプログラムが動作する Web 教材である点で、単なるテキスト教材の共有とは異なる。開発作業をしている同じ画面上で、ソースコードと動きを確認しながら、それを発展させるためのアイデアについての話を進められた。プログラミング基礎事項とサンプルが解説されたコンパクトな内容であることで、制作中、確認したい項目、探している事柄を何度も振り返って確認できる点も、有効であった。

5.2 本 Web 教材と既存教材との比較

既存の Processing 学習のための Web コンテンツには、Processing 自身に含まれる多くのサンプルプログラム、リファレンスに加え、Processing の公式 Web サイトが提供するチュートリアルがある [13]。このうち、開発者である Reas と Fry による Getting Started と Shiffman のビデオ講座は初学者向けコンテンツである。Getting Started は本教材の導入部分の内容と一致し、また Shiffman のビデオ講座「Hello Processing!」は、形、色、マウスとの応答、条件分岐の項目からなり、本教材のラーニングトレイルの 1 ならびに 2 と 3 の一部の内容と同等である。

導入部分について見れば、既存チュートリアルの内容と重なり、それで十分といえる。しかし、その次の段階に進

むと、既存チュートリアルはその内容が多岐にわたり、どのように学習を進めていいかわからなくなるという欠点がある。公式 Web サイトのチュートリアルでは、座標、色、イメージとピクセル、データ、曲線の描き方や三角関数といった項目ごとに解説する Web ページが提供されている。これらは内容ごとに独立しており、必要に応じて学習者が選択して参照できるようになっている。そのため、たとえば色や配列など自分の知るべき観点が明確な場合は必要な内容を適切に選択ができるが、そうでない場合は難易度がさまざまな項目からどれを学習したらいいか判断しにくい。これに対して、本教材は導入に続く学習の道筋を示し、音に応答するサンプルプログラムの制作へ向けて必要な要素を学べるようになっており、初学者が迷うことがない。

また、Processing の開発グループとは別に公開されているチュートリアルには、Pazos [14] のビデオによる解説コンテンツがある。単純な図形の描画、音声や画像データの処理まで 150 のエピソードを提供している。プログラムで視覚的に楽しい画像を作り出す観点から必要となる内容をトピックスとして取り上げて解説しているが、プログラミング基礎要素を学ぶという点からは内容は整理されていない。加えて、個々のエピソードは独立しておらず、連続したいいくつかのシリーズとなって 1 つの項目を解説している。そのため、利用者が自分の問題に適切なエピソードを探しにくい。一方、本教材はインタラクティブなモーショングラフィックスを制作するために必要となる知識と方法に絞り、コンパクトにまとまっているのに加え、リファレンスとして使えるプログラミングの基礎要素や Processing 関数の説明をともに簡潔に示し、忘れたらただちに参照できるところに特徴がある。

6. 結論

本研究では、SV 教育プログラムでの適用を念頭に、プログラミングの基礎要素を理解しながら、モーショングラフィックスを段階を追って作る自習教材を Web コンテンツとして開発した。これを使ったプログラミング学習が、プログラミング基礎項目の理解と自己効力感の獲得に、Java 言語での入門科目での学習と同等以上に効果をあげることを確認した。基礎要素を理解していれば、学習の場で制作するプログラムの修正や発展も容易であり、別の機会での問題解決にプログラムを有効に適応できる。単にプログラムが動けばいいというだけではなく、基礎項目の理解を確実にするための、SV 教育プログラムにおいてこの教材を適用することは有効と考える。

さらに、この教材は、SV 教育プログラムとは独立した一般プログラミング教育の場においても活用できる。SV ツールキットのようなセンサデバイスのない場面においても、マウスやキーボード以外のインタラクションを持つモーショングラフィックスプログラムを制作することで、学

びのモチベーションを高められる。

謝辞 本研究は、2015 年度同志社女子大学研究助成金（国内研究助成 B）の助成を受けた。

参考文献

- [1] Lego Mindstorms, available from <http://mindstorms.lego.com/>.
- [2] PicoCricket, available from <http://www.picocricket.com/>.
- [3] Ariga, T. and Mori, K.: Sensory vision- development of a course for physical interaction and graphics, *Computers & Graphics*, Vol.34, No.6, pp.800-810 (2010).
- [4] 有賀妙子, 森 公一: フィジカル・インタラクションを使ったメディア造形基礎教育におけるプログラミング学習の実践, 情報処理学会論文誌, Vol.52, No.12, pp.3096-3105 (2011).
- [5] Association for Computing Machinery, and IEEE Computer Society: Computer Science Curricula 2013 (2013), available from <http://www.acm.org/education/CS2013-final-report.pdf>.
- [6] National Research Council: *Being Fluent With Information Technology*, National Academy Press (1999).
- [7] Whitney, J.: *Digital Harmony: On the Complementarity of Music and Visual Art*, McGraw-Hill Inc. (1981).
- [8] Bandura, A.: *Self-Efficacy in Changing Society*, Cambridge University Press (1997). 本明 寛, 春木 豊, 野口京子ほか (訳): 激動社会の中の自己効力, p.18, 金子書房 (1997).
- [9] Ramalingam, V. and Wiedenbeck, S.: Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy, *Journal of Educational Computing Research*, Vol.19, No.4, pp.365-379 (1998).
- [10] Ramalingam, V., LaBelle, D. and Wiedenbeck, S.: Self-efficacy and mental models in learning to program, *ACM SIGCSE Bulletin*, Vol.36, No.3, pp.171-175 (2004).
- [11] Sethuraman, S. and Dee Medley, M.: Age and self-efficacy in programming, *Journal of Computing Sciences in Colleges*, Vol.25, No.2, pp.122-128 (2009).
- [12] de Winter, J. and Dodou, D.: Five-point Likert items: t test versus Mann-Whitney-Wilcoxon. *Practical Assessment, Research & Evaluation*, Vol.15, No.11, pp.1-12 (2010).
- [13] Fry, B. and Reas, C.: A collection of step-by-step lessons covering beginner, intermediate, and advanced topics (online), available from <http://www.processing.org/tutorials>.
- [14] Pazos, A.: Fun programming (online), available from <http://funprogramming.org>.

推薦文

本論文は、著者らが推進している、センサを用いたインタラクティブなメディア造形作品の制作を通してデジタル技術を用いた造形学習とプログラミング学習を行うセンサリバービジョン (SV) 教育プログラムに対する、補完 Web 教材について述べている。具体的には、SV 教育プログラムについて、作品は作れるようになって、必ずしもプログラミングの基礎概念を十分身につけていない場合があることを指摘したうえで、この部分を学ぶための教材を開発

したものである (SV プログラムとは独立に使用することも考慮されている)。教材は興味を持たせる題材を順に学んでいくラーニングトレイル, プログラミングの基本知識, サンプルプログラム群の3系列に分かれており, Web 教材の特徴を生かしてこれら3系列を自由に行き来しながら学ぶことができる。本論文にはラーニングトレイルの構成が掲載されているが, 造形学習につながる興味深いカリキュラムだと考える。本教材を勉強会の形で使用したものと従来型の Java の授業を比較した評価を行っており, 統計的にははっきりした違いは得られていないものの, 両手法の比較という点で有用な知見が示されている。全体として, プログラミング教育およびそのための教材について興味を持つ読者にとり, 有益な情報を多く含む価値のある論文であるといえる。

(論文誌「教育とコンピュータ」アドバイザー
久野 靖)



有賀 妙子 (正会員)

1978 年立教大学理学部卒業。1980 年東京工業大学大学院理工学研究科修士課程修了。情報通信技術関連企業勤務を経て, 現在, 同志社女子大学情報メディア学科教授。プログラムを使った情報デザインコンテンツ制作の教材開発とその評価に関する研究に従事。ACM, 日本デザイン学会各会員。



森 公一

1984 年大阪教育大学大学院教育学研究科修士課程修了。現在, 同志社女子大学情報メディア学科教授。メディアアート作品の制作, 理論研究に従事。現在は脳の血行動態等の生体情報を用いたバイオフィードバックによる表現

の可能性を探索。日本映像学会会員。



大下 福仁 (正会員)

2000 年大阪大学基礎工学部情報科学科退学。2002 年同大学大学院博士前期課程修了。2003 年同大学院情報科学研究科博士後期課程退学。同年大阪大学大学院情報科学研究科助手。同大学院助教を経て, 現在, 奈良先端科学

技術大学院大学情報科学研究科准教授。並列アルゴリズム, 分散アルゴリズムに関する研究に従事。博士 (情報科学)。ACM, IEEE, 電気情報通信学会各会員。



角川 裕次 (正会員)

1990 年山口大学工学部電子工学科卒業。1992 年広島大学大学院工学研究科情報工学専攻博士課程前期修了。広島大学助手, 講師, 助教授を経て, 現在, 大阪大学大学院情報科学研究科准教授。分散アルゴリズムと教育工学の

研究に従事。博士 (工学)。IEEE Computer Society, 電気情報通信学会各会員。



増澤 利光 (正会員)

1982 年大阪大学基礎工学部情報科学科卒業。1987 年同大学大学院博士後期課程修了。同年同大学情報処理センター助手。同大学基礎工学部助教授を経て, 1994 年奈良先端科学技術大学院大学情報科学研究科助教授。2000 年

大阪大学大学院基礎工学研究科教授。2002 年大阪大学大学院情報科学研究科教授。1993 年コーネル大学客員准教授 (文部省在外研究員)。分散アルゴリズム, 並列アルゴリズムに関する研究に従事。博士 (工学)。ACM, IEEE, 電気情報通信学会各会員。