

Processing による高校生を対象としたプログラミング入門体験

土肥紳一^{†1} 今野紀子^{†1}

Processing はプログラミングの入門者のために開発された言語である。Java 言語で開発されており Windows, Mac, Unix 上で利用できる。高校生を対象としたプログラミング入門体験の中で、Processing を活用した。スケッチという概念を持った Processing は、プログラミング入門体験を行う上で、高校生の学習意欲の向上に効果的であると考えた。本論文では授業内容、授業中の観察、SIEM を活用したモチベーションの分析結果について述べる。

Using the Processing for Practice of Introduction to Computer Programming Education for High School Students

SHINICHI DOHI^{†1} NORIKO KONNO^{†1}

The Processing is programming language which was developed for novice programmer. It is developed by Java language, it operates on Windows, Mac and Unix operating systems. I introduced the processing to practice of Introduction to Computer Programming Education for high school students. The processing has unique idea of sketch, I thought that the processing is effective improvement of high school students' motivation in the practice of Introduction to Computer Programming Education. In this paper, we describe course contents, course observation, analysis result of students' motivation by the SIEM assessment standard.

1. はじめに

東京電機大学情報環境学部は千葉県の印西市に位置し、地域密着型のキャンパスとして高大連携型教育の取り組みを 2008 年度から続けている。2013 年度は、表 1 に示す講義内容での開講が決まった。全体で計 10 回の講義となっており、2013 年度は新任の先生方にも担当をお願いし、新しい内容を多く取り入れる配慮があった。各回共に開講時間帯は 10:00~11:50 であり、50 分を 2 コマ、1 コマ目の終了時に 10 分の休憩を入れる授業形態である。講義は学部内の教室で実施する。2013 年度の受講者は、近隣の 2 つの高校へ通う生徒 22 名であり、学年の内訳は 1 年生が 14 名 (63.6%)、2 年生が 5 名 (22.7%)、3 年生が 3 名 (13.6%) である。

表 1 講義内容

	実施日	担当講師「講義名」
第 1 回	5 月 18 日	上野先生「簡単なデジタル回路体験 足し算の仕組み」
第 2 回	5 月 25 日	土肥先生「プログラミング入門体験」
第 3 回	6 月 1 日	遠藤先生「コンピュータを使ったデザイン」
第 4 回	6 月 8 日	近藤先生「素数を作る不思議な世界」
第 5 回	6 月 22 日	バローズ先生「アメリカの教育からみる日米の文化」
第 6 回	6 月 29 日	今野先生「色彩心理学 - 色彩と心の不思議 -」
第 7 回	7 月 13 日	八槇先生「やっぱり危ないインターネット ~ 情報セキュリティのホントのところ」
第 8 回	9 月 14 日	酒井先生「簡単な生体情報計測器を自作してみよう」
第 9 回	9 月 21 日	伊勢先生「音を聴いて空間を知る ~ 最新の音響技術の紹介」
第 10 回	9 月 28 日	見正先生「大学数学入門 - “無限” について考えよう」

2. 授業内容について

(1) 授業内容の検討

筆者は第 2 回の「プログラミング入門体験」を担当する。この内容は、高大連携型教育では初めて担当するものであり、どのように授業を組み立て、どこに教育目標を定めるかについて検討した。1 コマ 50 分の授業を 2 コマ連続で実施する授業は、本学部のコンピュータプログラミングの授業と同じである。コンピュータプログラミングの授業では、Java 言語を利用しており、高大連携型教育の授業でも、学部で大学生が学んでいる環境に可能な限り近付けて体験してもらうことが重要であると考えた。さらに講義名から「体験」を目的としており、生徒が教わったことを直ぐに体験できるように主眼を置き、以下の目標を立てた。

- ① キーボードの操作が不慣れな生徒でも、授業に十分に付いて行けるよう配慮する。
- ② 可能な限り普段のプログラミングの授業に近付ける。
- ③ 「順次」「分岐」「繰り返し」の概念を体験してもらう。
- ④ 受講前後の生徒のモチベーションを測定する。

(2) プログラム入力から実行までの壁

高校生を対象に Java 言語を利用する事は、普段のコンピュータプログラミングの授業を担当する中で、無理があることを認識している。Java は文字を表示するだけのプログラムでも、多くの「おまじない」を入力する必要がある。さらに、「おまじない」の中には、小括弧、中括弧、大括弧と複数種類の括弧が出現し間違いやすい。図 1 と図 2 は、「コンピュータプログラミング A」の授業で、はじめてプログラムを入力した時のプログラム例とコンパイル回数を示したものである。1 回のコンパイルで実行できる割合は、平均すると約 50% である。初学者にとって、「おまじない」の敷居が高いことが伺える。さらに、エディタを使ってプ

^{†1} 東京電機大学情報環境学部
Tokyo Denki University, School of Information Environment.

プログラムを入力し、これを保存し、コンパイルし、実行する操作は、煩雑さを増す。これに拍車をかけて、コンパイラエラーや実行時のエラーが発生すると、授業崩壊になりかねない。

```
public class A2_0{
    public static void main(String[] args){
        System.out.print("Sie Taro");
    }
}
```

図 1 最初のプログラミングの例

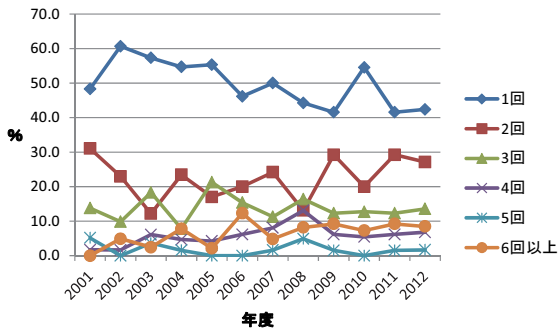


図 2 コンパイルエラーが出なくなるまでの回数

(3) プログラム言語の選択

プログラミングの入門者を対象に開発された言語として、日本語でプログラムを組めるドリトルやPENが有名である。ドリトルはオブジェクトを容易に作成でき親しみやすい[1]。筆者が担当する「情報科教育法」の授業の中で、受講者にドリトルを紹介している。主な感想には、「ドリトルなどのソフトがあれば、プログラミングの苦手意識なくプログラムを組む考え方が学べて良いなと思いました」「ドリトルは分かり易く、非常に興味深かった」等がある。PENはプログラムの動作や変数の変化などを視覚的に理解できる仕組みを持っており、プログラムがどのようにして動作しているのかが大変良く分かる[2]。PENに対する主な感想は、「プログラムを触ったことがない人にとっては、感覚的に操作できるためわかりやすいのではないかと思います」「PENがあれば、このプログラムの中身がどうゆう処理をされ、どう実行されるかが分かるので学習現場で使えると思いました」等がある。

「プログラミング入門体験」でも、ドリトルやPENの利用も考えたが、可能な限り普段のプログラミングの授業に近付ける目標を定めたことから、Processingを使う事に決めた。Processingは2001年にソフトウェアをスケッチ(試作)する道具として開発された[3]。グラフィックスやインタラクションに焦点を絞って開発されており、利用者にとって扱いやすい言語である。少ないキー入力でプログラムを実行でき、結果はグラフィックスを主体に表示できる。少ないキー入力は、キーボードに不慣れた初学者に最適である。

またJavaに類似した文法を持っており、Javaを利用したいと考えていた思惑と一致した。本番前に、「情報科教育法」の受講者にProcessingを試用してもらった。彼らの主な感想は、「Eclipseに似てると思いました」「Javaのほうが慣れているのでJavaが簡単な気がした」「非常に分かり易いと思った」等があった。Processingのサイトには豊富なサンプルプログラムが公開されており、これらを活用した[4]。

3. 1 コマ目の授業と生徒の観察

(1) 教室の形状

使用した教室は、筆者が普段コンピュータプログラミングの授業で使用している教室である。学生さんは個人所有のノートPCを持参して授業を受けるが、これを模倣するために大学所有のWindows 7のノートPCを貸し出した。教室は教材提示用のスクリーンが3面に分散配置され、テーブルが8箇所を設置されている。受講者と教授者の座席は、○と●で示した。この様子を図3に示す。1つのテーブルに約4名割り当て、教卓側の6個のテーブルを使った。

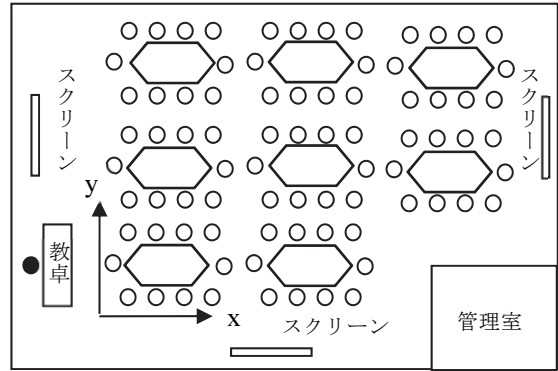


図 3 教室の形状(i-room)

(2) Processingの起動

Processingを起動すると、図4に示すウインドウが表示される。ウインドウは、ツールバー、タブ、テキストエディタ、メッセージエリア、コンソールから構成されている。

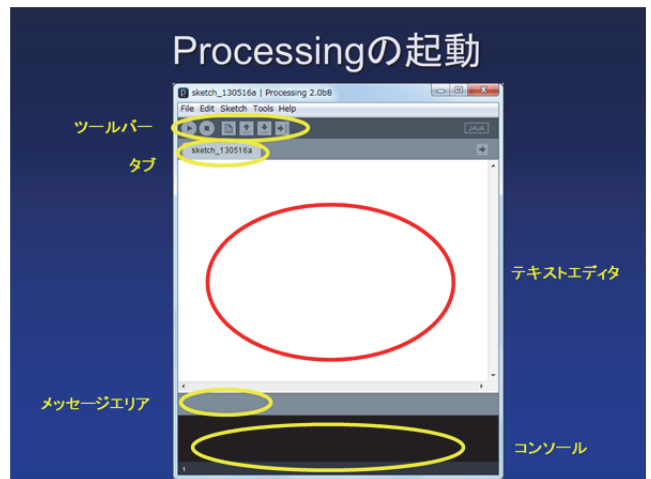


図 4 Processingの起動画面

(3) 初めてのプログラム

Processing における「Hello World」に相当するプログラムは、円の表示から始まる。テキストエディタに楕円を表示する関数を記述し、実行をクリックすると、その結果が即座に表示される。プログラムを保存すること無く、実行できる。最初のプログラムと実行結果を図 5 に示す。1 年生が 63.6%と多く、キーボードを初め、ノート PC 自体の操作に慣れていない事が想定されたため、スモールステップを多く取り入れ、全体の進行は細かく同期をとった。

```
ellipse(50, 50, 80, 80);
```

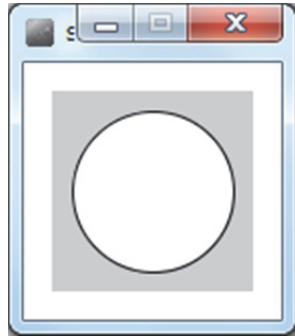


図 5 初めてのプログラムと実行結果

(4) 座標系について

Processing の実行結果は、グラフィックスが主体に表示されるため、プログラムを作成する上で座標系の概念が必要である。実行ウィンドウの座標系について補足説明した。この様子を図 6 に示す。

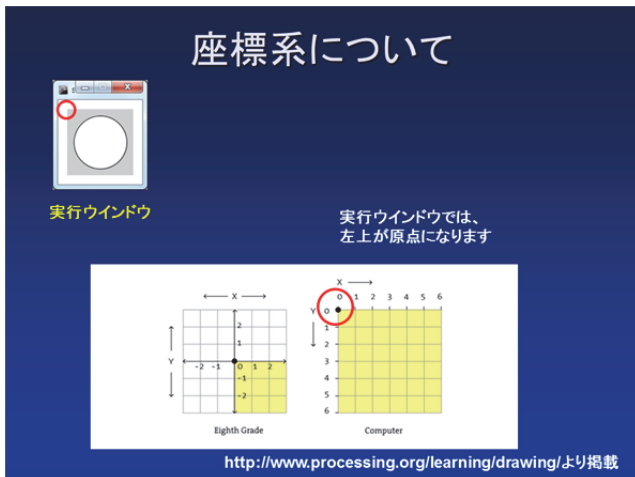


図 6 座標系の補足説明

(5) 円の表示

さらに ellipse()関数の引数について説明を加えた。この様子を図 7 に示す。わずか 1 行のプログラムであるが、生徒は記述した内容に従って計算機が動作している事を体験する。このプログラムで何名かの生徒から出た質問は、小括弧をどのように入力すれば良いのかと言った内容であった。このことから、中学校でキーボードの操作を十分に習得できていない生徒が居ることも分かった。一方、マウス

の操作に関する質問は無かった。

(6) ウィンドウのサイズ変更

実行ウィンドウが小さいので、これを大きくすることを試みた。具体的には、size()関数を ellipse()の前に追加する。多くの生徒たちが戸惑ったのは、ellipse()の前に入力することであった。単に行の挿入を行うことに他ならないが、テキストエディタに慣れていないためか、大半の生徒が行の挿入でつまづいた。ellipse()の e の前にカーソルを移動し、Enter キーを押すことで行の挿入ができることを説明した。

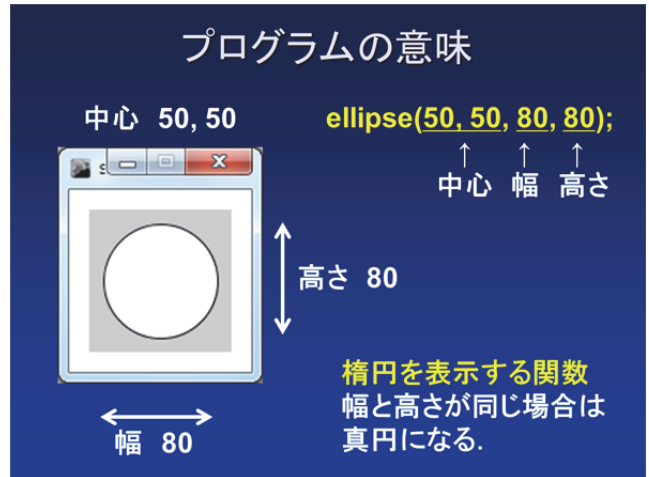


図 7 ellipse()関数の説明

```
size(480, 240);  
ellipse(50, 50, 80, 80);
```

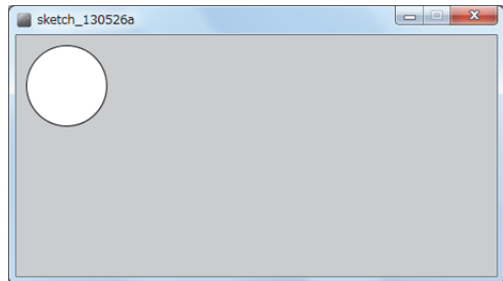


図 8 ウィンドウのサイズ変更

生徒の中には、ellipse()と size()を逆順に入力し、円が出て来ないと質問があった。プログラムがどのような順序で実行されているのかを説明する、良い質問であった。

(7) 複数の円の表示

複数の円を表示することを試みた。x 座標を変化させながら ellipse()を 2 つ追加した。その結果、x 軸の正の方向へ円が 3 つ表示される。この様子を図 9 に示す。この頃になると、引数の値を変更したり、さらに ellipse()を追加する生徒が出てくる。成功体験から、少し変更してやってみようという意欲が喚起される瞬間である。

(8) 繰り返しを使った複数の円の表示

この調子で円を表示し続けることは限界があることを

説明する。例えば 100 個の円を表示するためには、`ellipse()` 関数を 100 行記述することになり、大変である。そこで繰り返し(`for` 文)を使ってプログラムを書き換え、実行結果が同じになることを確認した。`ellipse()`関数は 1 行しか存在しないが、3 つの円が表示される事を体験する。ここでもキーボードからの入力に注意を払う必要があった。<や{}などは `Shift` キーを押さないと入力ができなことを補足した。実行結果は図 9 と同じため、プログラムのみ図 10 に示す。

```
size(480, 240);
ellipse(50, 50, 80, 80);
ellipse(60, 50, 80, 80);
ellipse(70, 50, 80, 80);
```

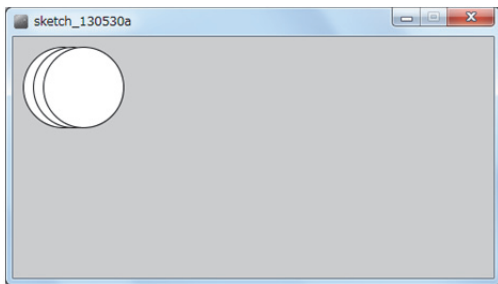


図 9 複数の円の表示

```
size(480, 240);
for(int i = 0; i < 3; i++) {
  ellipse(50 + i * 10, 50, 80, 80);
}
```

図 10 繰り返しを使った複数の円を表示するプログラム

for文の仕組み

```
for (int i = 0; i < 3; i++) {
  ④ ellipse(50 + i * 10, 50, 80, 80);
}
```

①が一回だけ実行されます(変数 `i` を0にする).

②の `i < 3` が成り立つ場合に④のブロックが実行されます.

③が実行されます(`i++`は、変数 `i` に1を加えます).

⋮

②の `i < 3` が成り立つ場合に④のブロックが実行されます.

③が実行されます(`i++`は、変数 `i` に1を加えます).

図 11 for 文の仕組み

繰り返しを使ったプログラムの実行を体験した後に、`for` 文の仕組みを解説した。この様子を図 11 に示す。この内容は、普段の授業で学生さんに説明する場合でも理解が難しい。1 回説明を聞いただけで理解できたとは考えにくい

が、数字の 3 を書き換えることによって、繰り返す回数を変えられることを説明した。変数 `i` は、陸上のトラック競技で何周したかを数えるための役割であることを補足した。さらに `y` 座標も `x` 座標と同様に、繰り返すに応じて増えるように記述すると、右下に向かって円が表示されることを体験した。この様子を図 12 に示す。この頃になると、「色を変えるにはどうすれば良いのか」との質問が出るようになった。成功体験から新しい創造が芽生え、色を変えてみたいとの興味も喚起されたようである。後で説明することを補足した。また、円の幅と高さを変える試みも始まった。`ellipse()`関数で得た知識を利用することによって、`rect()`関数と `line()`関数は容易に理解できる。この結果、自主的に関数を追加し、お絵かきが始まった。この様子を図 13 に示す。

```
size(480, 240);
for(int i = 0; i < 15; i++) {
  ellipse(50 + i * 10, 50 + i * 10, 80, 80);
}
```

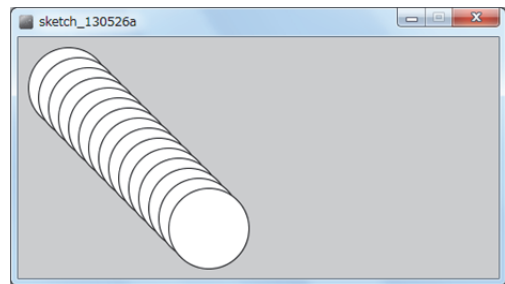


図 12 繰り返しを使った複数の円の表示

```
size(200, 200);
rectMode(CENTER);
rect(100, 100, 20, 100);
ellipse(100, 70, 60, 60);
ellipse(81, 70, 16, 32);
ellipse(119, 70, 16, 32);
line(90, 150, 80, 160);
line(110, 150, 120, 160);
```

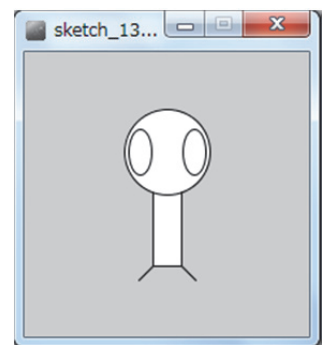


図 13 お絵かき

以上が、1 コマ目の 50 分の内容である。キー入力は不慣れなもの、プログラムの実行結果が図形で表示されることにより、その動作を十分に理解できることが観察された。

4. 2 コマ目の授業と生徒の観察

(1) マウス座標の活用

2 コマ目は、連続動作を活用したプログラミングを試みた。Processing は `setup()`関数と `draw()`関数が用意されており、非常に興味深い機能を提供してくれる。`setup()`関数は、

プログラムの実行時に1回だけ実行されるメソッドである。一方、draw()関数は、毎秒60回実行されるメソッドである。

これらを利用すると、1コマ目で学習した知識を拡張し、動きのあるプログラムを簡潔に記述できる。マウスのx座標とy座標は、変数 mouseX と mouseY が用意されており、ここに格納される。したがって draw()メソッドの中で ellipse()を実行する時の中心の座標に mouseX と mouseY を指定することにより、マウスカーソルの移動に応じて円が表示される。少ないプログラムの記述の割に迫力のある結果が表示され、生徒たちは興味津々であった。この様子を 図 14 に示す。このことは、計算機が非常に高速に動作し

```
void setup() {
  size(480, 240);
}
void draw() {
  ellipse(mouseX, mouseY, 80, 80);
}
```

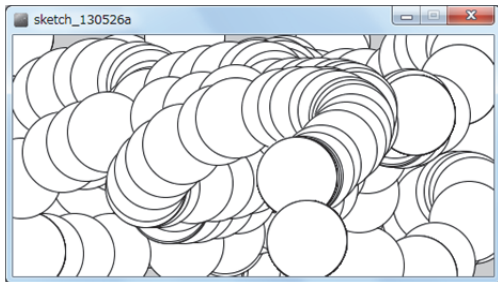


図 14 draw() 関数を使った複数の円の表示

```
void setup() {
  size(480, 240);
}
void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

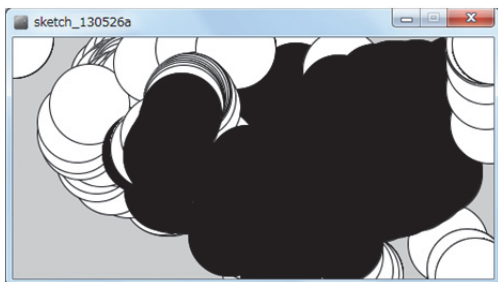


図 15 円の塗りつぶし

ていることを体験する良い例であった。実行ウインドウの大きさが物足りない生徒は、そのサイズを変更し始めた。また、マウスの座標によって、円の幅と高さを変更する試みも出てきた。手の動きに合わせて円が表示されることに、興味が喚起された様子であった。

(2) プログラムの分岐

分岐の概念を理解するために、マウスボタンが押されたときに黒く塗りつぶす例を試み、分岐(if文)によって処理を分けることを体験した。この様子を 図 15 に示す。if文の仕組みは 図 16 で説明した。このプログラムを試した結果、多くの生徒がウインドウ全体を黒く塗りつぶす操作に専念し始めた。マウスボタンが押されたか否かによって、動作が変わることの理解が目的であったが、生徒の目線では、ウインドウを黒く塗りつぶすことに興味が推移した。

if文の仕組み

```
if ( mousePressed ) {      条件式
  fill(0);                 条件式が成立した時に
                           実行されるブロック
} else {
  fill(255);               条件式が成立しない
                           時に実行されるブロック
}
```

図 16 if文の仕組み

(3) 色の表示

1コマ目でも出ていた質問であるが、「円に色を付けるにはどうすれば良いのか」との質問が多く出るようになった。円を表示できた後、着色することへの発想は、生徒にとって非常に素直な事と考えられる。塗りつぶしの色の変更は、fill()関数で行える事を説明した。この様子を 図 17 に示す。

fill(塗りつぶし)



モノクロ
色の濃さを0から255の数字で指定
fill(153);
rect(30, 20, 55, 55);



カラー
赤, 緑, 青の順に0から255の数字で指定
fill(204, 102, 0);
rect(30, 20, 55, 55);

<http://www.processing.org/learning/drawing/>より掲載

図 17 塗りつぶしの色の変更

(4) 円の回転

最後は、円が円周上を移動するプログラムで締めくくった。これはラジアンや三角関数を使うため、1年生での理解は難しい。理屈は、将来、理解できることを補足し、sinとcosを逆にすることによって、円の回転方向が逆になることを試した。ellipse()関数の幅と高さを実行毎に変化させることによって円を次第に大きくする事や、実行ウィンドウを大きくするなど、各自の工夫が始まった。

以上、2コマ目の50分の授業の中で筆者が観察したことを述べた。1年生の参加が多く、キーボードの操作が不慣れである印象を受けたが、当初の目標に全員が到達できた。

```
float rad = 0;
void setup(){
    size(120, 120);
}
void draw(){
    float x = 50 * cos(rad) + 60;
    float y = 50 * sin(rad) + 60;
    ellipse(x, y, 20, 20);
    rad = rad + 0.05;
}
```

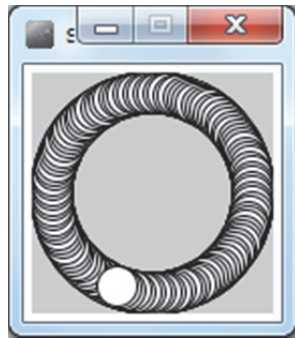


図 18 円の回転

表 2 SIEM アセスメント尺度

因子 1：授業構成因子	
(1) 成功機会度	授業中にできた・わかったという実感がありますか。
(2) 親性度	授業の内容は親しみやすいですか
(3) 愉楽度	このプログラミングの授業は楽しいと思えますか。
(4) 理解度	このプログラミングの授業は理解しやすいですか。
(5) 知覚的喚起度	自分が入力したプログラムの動作結果を見るのは楽しいですか。
(6) 意義の明確度	授業の意義や目的がはっきりしていますか。
(7) 好奇心喚起度	授業では好奇心を刺激されますか。
因子 2：自発性因子	
(8) 将来への有用度	将来に役立つと思いますか。
(9) 向上努力度	もっとプログラミングの勉強を努力しようと思いますか。
(10) 自己コントロール度	授業で学習したことを基にして、自分で工夫し勉強してみようと思いますか。
(11) 自己目標の明確度	自分の到達すべき学習の目標がはっきりしていますか。
因子 3：双方向性因子	
(12) コミュニケーション度	授業中、学生・教員などとのコミュニケーションはありますか。
(13) 所属集団の好意的反応度	教員やクラスのメンバーは好意的ですか。
(14) コンテンツの合致度	演習問題などは授業内容と一致していますか。
因子 4：参加性因子	
(15) 参加意欲度	休まずに出席しようという意欲が起こる授業ですか。
(16) 参加積極度	授業での自分の参加態度は積極的ですか。
モチベーション評価項目	
(17) 重要度	プログラミングを学習することは重要だと思いますか。
(18) 現状認知度	現在の時点で、プログラミングの知識・技術は身につけていると思いますか。
(19) 期待度	もっとプログラミングの知識や技術を高めたいと思いますか。

5. モチベーションの評価

(1) モチベーションの測定

「プログラミング入門体験」が、生徒のモチベーションにどのような影響を与えたかを評価した。評価方法は、本学部のプログラミングの授業で継続的に実践活用している SIEM アセスメント尺度を使い、アンケート調査を行った。SIEM アセスメント尺度は ARCS モデルを基に作成した尺度である[5]。これはプログラミング入門教育を対象に、モチベーションの向上を目指す研究の中で完成した[6]。SIEM アセスメント尺度を表 2 に示す。各項目について、5段階のリッカート尺度で回答を求め、モチベーションは、調査項目の「(17)重要度」と「(19)期待度」の積で算出する。なおアンケート調査は、スキャネットシートを使った。受講前の状況は、「(20) 授業を受ける前、プログラミングを学習することは重要だと思いましたか。」「(21) 授業を受ける前、プログラミングの知識・技術は身につけていると思いましたか。」「(22) 授業を受ける前、もっとプログラミングの知識や技術を高めたいと思いましたか。」を追加した。

(2) 分析結果

このクラスのモチベーションは、受講前(16.8)から受講後(19.9)にかけて 3.1 上昇した。この様子を表 3 に示す。

表 3 基本統計量

		受講前	受講後
全体 (標本数=22)	平均	16.8	19.9
	標準誤差	1.4	1.1
	中央値	16.0	20.0
	最頻値	25.0	25.0
	標準偏差	6.5	5.2
	分散	42.7	26.8
	尖度	-1.5	-0.9
	歪度	0.0	-0.5
	範囲	17.0	16.0
	最小	8	9
最大	25	25	
合計	369	437	
上位群	人数(%)	45.5	63.6
20 ≤ MV	MV の平均	62.3	74.4
中位群	人数(%)	31.8	31.8
10 ≤ MV < 20	MV の平均	26.3	23.6
下位群	人数(%)	22.7	4.5
MV < 10	MV の平均	11.4	2.1

さらに受講者のモチベーションに繋がる要因と満足度について CS(Customer Satisfaction)分析を施行した。CS 分析は、顧客満足度を調査する目的で使われる手法であり、授業評価でも活用されている[7]。CS 分析を SIEM の分析に加えることによって、モチベーションの満足度と改善すべき要因を明確にできるようになる。CS 分析の導入に伴い、目的変数をモチベーション、説明変数を SIEM アセス

メント項目とし、目的変数と説明変数との単相関係数を関連度、説明変数の評価値(平均値)を満足度とする。さらに、各々を偏差値化し、関連度偏差値(RLD: Related Level Deviation score)、満足度偏差値(SLD: Satisfaction Level Deviation score)と呼ぶ。

改善度指数(ILI: Improvement Level Index)は、CS分析によって得られた RLD と SLD を使って求めることができる。ILI の算出は以下の通りである。CS グラフから、RLD と SLD が 50 になる点を原点とする。原点から各 SEIM アセスメント項目までの距離 L と、直線 $y = -x$ とのなす角度 θ ($0 \leq \theta \leq 180$) を使い、(1)式で算出する。点 P_1 と点 P_2 を例に、長さ L_1, L_2 と $y = -x$ とのなす角度 θ_1, θ_2 を図 19 に示す。

$$ILI = \frac{L(90 - \theta)}{90} \quad (1)$$

$$L = \sqrt{(RLD - 50)^2 + (SLD - 50)^2}$$

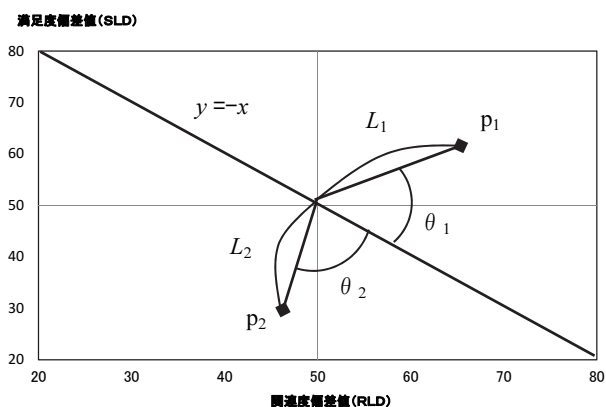


図 19 CS グラフと改善度指数(ILI)の関係

図 20 はそれぞれを偏差値化しプロットしたものであり、縦軸が満足度偏差値(SLD)、横軸がモチベーションとの関連性を表す関連度偏差値(RLD)となっている。CS 分析の結果、このクラス集団のモチベーションには、「向上努力度(67.1)」「愉楽度(64.9)」「自己コントロール度(63.9)」「好奇心喚起度(59.9)」が強く関与していることが判明した。また、このクラス集団の満足度では「知覚的喚起度(64.6)」「愉楽度(62.8)」「参加意欲度(62.8)」「成功機会度(59.1)」が高くなり、これらがモチベーションに繋がった。これらの様子を表 4 に示す。

(3) 今後への提案事項

表 4 は、CS 分析による授業改善度指数(ILI)も表している。改善度指数が高いものは改善すべき項目であり、特に 5 以上は要改善、10 以上は即改善項目と考えられる[8]。後期への提案事項では、改善度指数 5 以上である項目のうち上位 2 項目を目安にし、改善が効果的であると判断される項目について提案する。

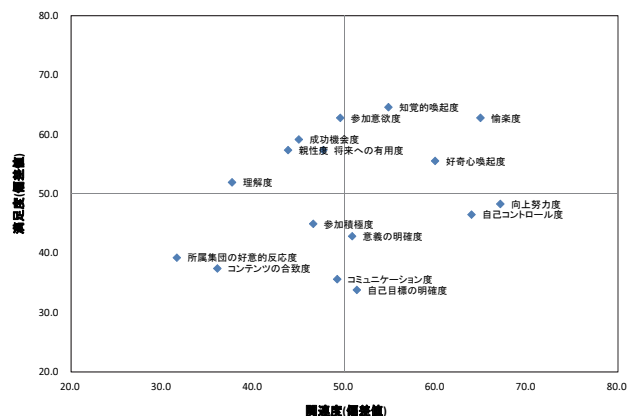


図 20 CS グラフ

表 4 CS 分析による改善度指数

SIEM アセスメント項目	満足度偏差値 (SLD)	関連度偏差値 (RLD)	改善度指数 (ILI)
成功機会度	59.1	45.0	-8.5
親性度	57.3	43.8	-9.1
愉楽度	62.8	64.9	1.0
理解度	51.9	37.7	-7.5
知覚的喚起度	64.6	54.8	-4.6
意義の明確度	42.9	50.8	4.1
好奇心喚起度	55.5	59.9	2.0
将来への有用度	57.3	47.7	-5.4
向上努力度	48.3	67.1	9.7
自己コントロール度	46.5	63.9	9.5
自己目標の明確度	33.8	51.4	9.0
コミュニケーション度	35.6	49.2	6.7
所属集団の好意的反応度	39.2	31.6	-3.5
コンテンツの合致度	37.4	36.1	-0.6
参加意欲度	62.8	49.5	-6.7
参加積極度	44.9	46.6	0.7

CS 分析の結果から、「親性度」「成功機会度」「理解度」「参加意欲度」「将来への有用度」「知覚的喚起度」「所属集団の好意的反応度」「コンテンツの合致度」においてモチベーションへの成果が認められた。さらなるモチベーション向上のためには、関連度は高いが満足度が低い項目、すなわち「向上努力度(9.7)」「自己コントロール度(9.5)」「自己目標の明確度(9.0)」の改善、工夫が効果的である。具体的には、「向上努力度」に関しては、プログラミングの勉強を努力することで今以上に向上できることを説諭し励ます。「自己コントロール度」に関しては、最初はできそうな課題で「やればできる」という感覚をつかませながら、馴れた頃にチャレンジ精神をくすぐるような課題に挑戦させることで、生徒に自らの工夫を生かした成功体験を与える。「自己目標の明確度」に関しては、今回、高大連携型教育の「プログラミング入門体験」という枠組みでの授業であり、まずは体験するといったことが目的となっているため、生徒に自らの到達すべき学習の目標を立てさせることはし

ていない。今後、彼らが自己の学習到達目標を明確化することで、さらに学習モチベーションは向上すると思われる。

マッチングは、表 2 に示した SIEM アセスメント尺度の (1) から (16) の項目についてレーダーチャートで示したものである。教授者と受講者の間で良好な関係が成り立っていれば、各項目の値は 5 に近づき、逆の場合は 1 に近づく。

「成功機会度」「親性度」「愉快度」「知的喚起度」「好奇心喚起度」「将来への有用度」が「参加意欲度」が「自己目標の明確度」「コミュニケーション度」「所属集団の好意的反応度」「コンテンツの合致度」が低目となった。

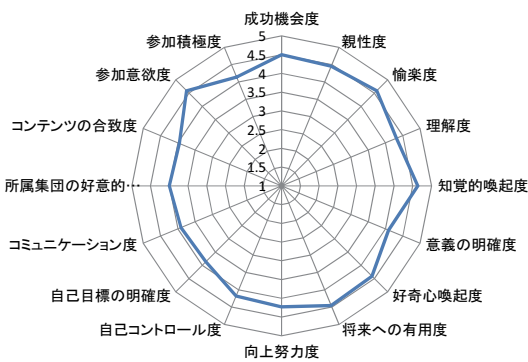


図 21 マッチングの状況

(4) モチベーションの推移

モチベーションの推移は、受講後のモチベーションから受講前のモチベーションの差を求め分析した。差が正になった(モチベーションの向上が見られた)生徒は 10 名(45.5%)、変化がなかった生徒は 11 名(50.0%)、低下した生徒は 1 名(4.5%)であった。さらにモチベーションの推移をパターン化するために、表 3 に示した上位群、中位群、下位群の 3 つについて H, M, L の記号を割り当て、受講前と受講後の状態を文字の連結で表現した。モチベーションの向上では MH が 4 名と最も多く、LH が 1 名、LM が 1 名であった。逆にモチベーションが低下したのは、HM の 1 名であった。LL が 1 名いた。これらの結果から、「Processing による高校生を対象としたプログラミング入門体験」がモチベーション向上の要因であると考えられる。

表 5 モチベーションの推移

推移のパターン	人数	割合(%)
HH	9	40.9
HM	1	4.5
LH	1	4.5
LL	1	4.5
LM	3	13.6
MH	4	18.2
MM	3	13.6
合計	22	100.0

6. まとめ

Processing を活用したプログラミング入門体験は、生徒のモチベーションを向上でき、効果的であることが分かった。一部の生徒の表情から、授業の最初はノート PC の操作に不安も見られたが、要領がつかめると成功体験が増え、不安が自信につながり、色々、思いついたことを試すようになった。生徒の独創性や創造性が、喚起された。また、引率の先生からいくつかの指摘を頂いた。偶然、引率の先生は「情報」と「数学」を担当されていた。「楢円は、高校 1 年生では教わらない」「三角関数は 2 年生で、ラジアンは 2 年生の後期で教わる」。また、「情報の授業は 2 年から行っている」等であった。キーボードやマウスの操作は、高校 1 年生の場合、中学校で教わったスキルが前提となる。これらのことは、授業実施前の準備段階で十分想定したことであるが、貴重なコメントをいただくことができた。今後の開催では、表現の工夫等が必要であることが分かった。

2013 年 4 月から、普通教科「情報 A」「情報 B」「情報 C」は廃止され、「情報の科学」と「社会と情報」に代わった。今後、履修割合がどのようになるのか、気になる所である。今回の授業を通じて、プログラミングに対する動機づけとして、Processing は十二分に役立つソフトであることを実感した。高校生を対象に初めてプログラミングの授業を担当したが、生徒たちはその面白さを実感できたようである。

またプログラミングを学ぶためには、数学や理科の基礎知識に基づく法則や理論の活用が必要な事、多くのプログラム言語は英語圏で開発されており、英語の基礎知識が必要であることも今回の体験を通じて理解できたと考えている。プログラミングを本格的に学ぶためには、情報系の大学へ進学する事を説諭し、授業を終了した。「情報の科学」の教科書では Visual Basic がプログラムの例として取り上げられている。生徒のプログラミングに対する興味が喚起されるのか、今後の様子を見守りたいと思う。

参考文献

- 1) 兼宗進, 久野靖: ドリトルで学ぶプログラミング第 2 版, イーテキスト研究所
- 2) 久野 靖, 辰己丈夫, 中野由章, 西田知博他: 情報科教育法改訂 2 版, オーム社
- 3) Casey Reas, Ben Fry 著, 船田 巧訳: Processing をはじめよう, オライリー・ジャパン
- 4) Processing
<http://www.processing.org/>
- 5) J.M.Keller, K.Suzuki : Use of the ARCS motivation model in courseware design (Chapter 16). In D.H. Jonassen(Ed.), Instructional designs for microcomputer courseware. Lawrence Erlbaum Associates, U.S.A, 1988
- 6) 土肥紳一, 宮川治, 今野紀子: SIEM によるプログラミング教育の客観的評価, 情報科学技術フォーラム情報科学技術レターズ, 3-3, pp.347-350, 2004
- 7) 南 学, 学生による授業評価への CS 分析の適用, 三重大学教育学部附属教育実践総合センター紀要, pp.29-34, 2007
- 8) 菅 民郎, らくらく図解アンケート分析教室, オーム社, 2007