

# プログラミングのビジュアル学習システムの コースウェア化

西川和隆\*1・鎌田洋\*1

Email: b1107434@planet.kanazawa-it.ac.jp

\*1: 金沢工業大学大学院 工学研究科 システム設計工学科専攻

◎Key Words プログラミング学習、学習システム、コースウェア化

## 1. はじめに

プログラミングの学習をする人は少なくない。しかし、プログラミングに関心が深く学習意欲が旺盛である人でなければ継続して学習するのは難しい。プログラミングの初心者がモチベーションの維持できる学習方法が必要である。プログラミング学習において対象者が楽しんで学習でき、自らの成果がすぐに確認できるようにすることでモチベーションの維持に繋がるのではないかと考えた。プログラミング初心者に適した初歩的な問題から応用問題について、プログラミングの結果の正誤を目でみて容易に確認できる仕組みを導入したビジュアル学習システムを試作した。

## 2. 従来のシステム

従来のプログラミングを学習するシステムの一つに「プログラミングの楽しさ・魅力を体験的・実践的な方法で伝える」<sup>(3)</sup> という方法がある。これは座学と演習を中心とするスタイルではなく、実験という形態をとって授業を行うというものである。アニメーションやゲームなど学生が興味を持つ題材を課題として多く用いることで学習意欲を高めている。しかしこの方法では学生がプログラミングに興味をもってもらうことができても実践的なプログラミングの技術を身に付けることは難しいと思われる。

二つ目のシステムは自動的に穴埋め問題を生成する自主学習システム<sup>(4)</sup>である。このシステムは1つのプログラムを分解し、自動的に穴埋め問題を作成するシステムである。学習者が問題に解答するとその学習者のデータが更新され、学習者の苦手な知識を解析し、毎回異なるデータを用いて問題を作成する。これによって学習者は同じ問題を繰り返す必要がなくなり、学習者が答えを暗記してしまう問題を解決することができる。しかしこれはプログラミングに得意な学生には有効ではあるが、プログラミングに苦手意識をもっている学生には難しいと思われる。

## 3. 全システムと問題点

### 3.1 現状のシステム

本システムはプログラミング初心者にはプログラミングを学習してもらうために開発したがプログラミングといってもさまざまなプログラム言語がある。よって、

本システムの初心者がモチベーションを維持してプログラミングの学習を行えるという目的に合うようにビジュアルプログラミングを使用して対象者に学習してもらおうと考えた。ビジュアルプログラミングとは実行画面に図や文字が描き出されプログラムをした人が自分の成果を一目で確認できるようなプログラミングのことである。実行画面が目で見確認できるプログラミングは使用者がモチベーションを維持して学習できると考えた。概念図を図1に示す。

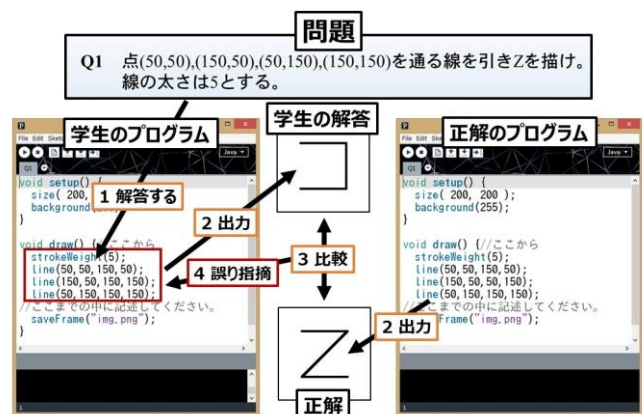


図1 システムの概念図

本システムでは答え合わせを画像や動画による正否判定で行う。答えを見るだけの正否判定より実際に間違えたところ分かる画像による正否判定のほうが使用者は理解しやすいのではないかと考えたからである。

静止画による正否判定は、画像プログラムの答え合わせのときに使用する。複数段階の成否判定は長文のプログラム問題の答え合わせで使用。動画による正否判定では、アニメーションのプログラム問題の答え合わせのときに使用する。

### 3.2 問題点

本システムに実装されているプログラミング問題は動作確認を目的に作成したものであり、実用的なものではなかった。より実践的な問題を作成し、学習者のプログラミングへの理解を向上させる必要がある。

## 4. 改善策と改善方法

### 4.1 改善策

プログラミング初心者でもわかるようにより基礎的な問題から構成し、体系的なプログラミング学習を行えるように問題のコースウェア化を行うことにした。問題をコースウェア化して出題することにより対象者のプログラミングに対する理解を段階的に向上させる。

### 4.2 改善方法

今回は対象者のモチベーションを維持するために比較的簡単な図形を描く問題を使って実験を行った。

今回からは問題を体系的に学習できるようにより実践的なプログラミングの問題を processing に実装されているサンプルプログラムから作成した。問題はジャンル分けされており、簡単な穴埋め問題から始まり、問題を解いていくと穴埋めの数が増えていくようにした。全体の学習内容を表1に示す。

表1 学習内容

サンプルプログラムの名称	領域	学習内容
Array	配列	配列の使い方
Color	色彩	色の指定
Control	制御	線や点の描き方
Form	形、形状	複雑な図形
Object	物体	アニメーション

まず Array では配列についての問題が出題される。最初は簡単な穴埋めから始まり、徐々に穴埋めを増やしていく。

Color では色彩についての問題を出題する。学習者には fill()の基本的な使い方を勉強してもらい、色の指定を覚えてもらう。

Control では点や線の描画についての問題が出題される。学習者には line()の中の数値を穴埋めして線を描いて使い方を学習してもらう。

Form では曲線を使った問題を出題する。問題の構成は基本的に前文の control の問題と同じで数値の穴埋め問題を中心にし、使い方を覚えてもらう。ほかにも曲線を使って複雑な図形の描き方の問題を出題する。

Object では図形を動かす問題を出題する。学習者には If 文などの条件分岐や変数でアニメーションの作り方を学び、使い方を覚えてもらう。

## 5. プログラム作成問題の例

問題のジャンルは processing に実装されているサンプルを参考に問題を設けた。

### • Array の問題

Q1 では配列の変数についての問題、Q2 では多次元配列についての問題、Q3 では繰り返しの配列についての問題を出題する。

**Q1** ①に当てはまるものを選択肢から選んで以下の図を完成せよ

```
float[] coswave;

void setup() {
  size(640, 360);
  coswave = new float[width];
  for (int i = 0; i < width; i++) {
    float amount = map(i, 0, width, 0, PI);
    coswave[①] = abs(cos(amount));
  }
}
```

(1)i (2)width (3)PI

答え：① i

**Q2** ①、②に当てはまるものを選択肢から選んで以下の図を完成させよ。ただし同じものは選ばないものとする。

```
void setup() {
  size(640, 360);
  maxDistance = dist(width/2, height/2, width, height);
  distances = new float[width][height];
  for (int y = 0; y < height; y++) {
    for (int x = 0; x < width; x++) {
      float distance = dist(width/2, height/2, x, y);
      distances[①][②] = distance/maxDistance *
    }
  }
}
```

(1)i (2)x (3)y (4)width (5)height

答え：① x ② y

**Q3** ①、②に当てはまるものを選択肢から選んで以下の動画を完成させよ。ただし同じものは選ばないものとする。

```
int wideCount = width / unit;
int highCount = height / unit;
count = wideCount * highCount;
mods = new Module[①];

int index = 0;
for (int y = 0; y < highCount; y++) {
  for (int x = 0; x < wideCount; x++) {
    mods[②++] = new Module.....
  }
}
```

(1)index (2)x (3) count (4)y (5)height

答え：① count ② index

### • Color の問題

Q1～Q3 で RGB を使った色指定の問題を出題する。

**Q1** ①、②、③に当てはまるものを選択肢から選んで以下の画像を作成せよ。ただし同じものは選ばないものとする。

```
inside 赤:102 青:0 緑:204
color inside = color(①、②、③);
```

(1) 0 (2) 30 (3) 204 (4) 100 (5) 102

答え：① 102 ② 204 ③ 0

Q2 ①～④に当てはまるものを選択肢から選んで以下の画像を作成せよ。ただし同じものは選ばないものとする。

```

a 赤:165 青:20 緑:167
b 赤:165 青:20 緑:89

a = color(①, ②, ③);
b = color(①, ④, ②);
    
```

(1) 165 (2) 20 (3) 204 (4) 89 (5) 167 (6) 146

答え : ① 165 ② 167 ③ 20 ④ 89

Q3 ①、②、③に当てはまるものを選択肢から選んで以下の図を完成させよ。ただし同じものは選ばないものとする。

```

線のグレースケール:255 透明度:100

for (int x = gridSize; x <= width - gridSize; x += gridSize) {
  for (int y = gridSize; y <= height - gridSize; y += gridSize) {
    noStroke();
    fill(255);
    rect(x-1, y-1, 3, 3);
    ①(②, ③);
    line(x, y, width/2, height/2);
  }
}
    
```

(1) stroke (2) strokeWeight (3) 180 (4) 255 (5) 100

答え : ① stroke ② 255 ③ 100

Q3 ①～④に当てはまる数値を書いて以下の図を作成せよ

```

b1 白:255
b2 黒:0
c1 赤:204 青:0 緑:102

b1 = color(①);
b2 = color(②);
c1 = color(③, ④, ②);
    
```

(1) 165 (2) 0 (3) 204 (4) 89 (5) 102 (6) 255

答え : ① 255 ② 0 ③ 204 ④ 102

・ Control の問題

Q1～Q2 では line() を使って線を描く問題、Q3 では stroke() を使った線の色や透明度を指定する問題を出題する。

・ form の問題

Q1 では曲線を描く問題、Q2 では円弧を描く問題を出題する。

Q1 ①～④に当てはまる数値を書いて以下の曲線を描け。

```

始点のx座標:50 終点のx座標:450
始点のy座標:250 終点のy座標:250
制御点座標 (410,20),(440,300)

bezier(①, ②, 410,20,③, ④, 450, 250);
    
```

答え : ① 50 ② 250 ③ 440 ④ 300

Q1 ①～④に当てはまるものを選択肢から選んで以下の図を完成させよ。ただし同じものは選ばないものとする。

```

最初の線:y座標が80から始まる長さ100の線
2番目の線:y座標が20から始まる長さ160の線
width=640, height=360

for(int i = 10; i < width; i += 10) {
  // if i divides by 20 with no remainder draw
  // the first line, else draw the second line
  if((i % 20) == 0) {
    stroke(255);
    line(i, ①, i, ②);
  } else {
    stroke(153);
    line(i, ③, i, ④);
  }
}
    
```

(1) 20 (2) 80 (3) 180 (4)width (5) height/2

答え : ① 80 ② height/2 ③ 20 ④ 180

Q2 ①、②に当てはまるものを文章から抜き出して以下の図を作成せよ

```

arc( 円弧の中心のx座標, 円弧の中心のy座標, 横の直径, 縦の直径, 弧を描き始める角度, 円弧を描き終える角度 );

void pieChart(float diameter, int[] data) {
  float lastAngle = 0;
  for (int i = 0; i < data.length; i++) {
    float gray = map(i, 0, data.length, 0, 255);
    fill(gray);
    arc(width/2, height/2, ①, ②, ②+radians(angles[i]),
        lastAngle += radians(angles[i]);
    );
  }
}
    
```

答え : ① diameter ② lastAngle

Q2 ①、②に当てはまるものを選択肢から選んで以下の図を完成させよ。ただし同じものは選ばないものとする。

```

y座標が180から始まる長さ340の線
width=640, height=360

else {
  stroke(102);
  line(i, ①, i, ②);
}
    
```

(1) 20 (2) 80 (3) 180 (4) height-20 (5) height/2

答え : ① height/2 ② height-20

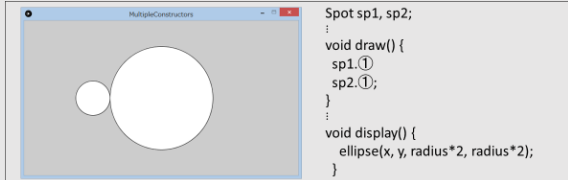
## ・Object の問題

Q1 はオブジェクト指向のプログラミング問題、Q2 は毎フレームごとに座標を増やしていき物体を動かすプログラム問題について出題する。

## 参考文献

- (1) 西川和隆、奥井康則：“プログラミングのビジュアル学習システム”、2015 PC Conference、pp317-320(2015)。
- (2) 西川和隆、奥井康則：“プログラミングのビジュアル学習システム”、CIEC 研究会報告集 Vol. 7、pp54-pp57(2016)。
- (3) 今仁順也、長名優子、菊池眞之、伊藤雅仁、石畑宏明：“プログラミングの初年度導入教育に関する実施報告”、一般社団法人 電子情報通信学会、ET2015-51、pp43-48 (2015)
- (4) 原 昂平、閻 宇、中野 廣人：“C プログラム初学者のための穴埋め問題生成システムの実装”、一般社団法人 電子情報通信学会、ET2015-29、pp37-42 (2015)。

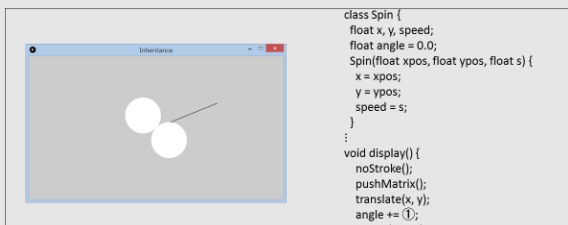
**Q1** ①に当てはまるものを選択肢から選んで以下の図を完成させよ。



(1) draw() (2) display() (3) xpos (4) ypos (5) 100

答え：① display()

**Q2** ①に当てはまるものを選択肢から選んで以下の動画を作成せよ



(1) x (2) y (3) xpos (4) ypos (5) speed

答え：① speed

## 6. 評価実験

問題が実戦的であるかを探るためにプログラミングが得意でない学生にシステムを使用させ実験後、前システムと同様にアンケート調査を行う予定である。評価の段階は「当てはまる」、「少し当てはまる」、「どちらでもない」、「あまり当てはまらない」、「当てはまらない」の5段階評価で行う。アンケート項目は以下のとおりである。

- 1：システムが使いやすい
- 2：ビジュアルプログラミングを学習できていると感じる
- 3：モチベーションを維持しながら学習できる

## 7. 考察

今回の研究で問題をコースウェア化するためにプログラミング言語 Processing に実装されているサンプルプログラムを参考に問題を制作した。関数の使い方から図形を動かすプログラム問題までの流れで段階的に学習できるようになっていると考える。

## 8. おわりに

本研究ではプログラミング言語 Processing に実装されているサンプルプログラムを参考にプログラミング初心者でもわかるようにより基礎的な問題から構成し、体系的なプログラミング学習を行えるように問題のコースウェア化の設計を行った。今後の評価実験で有効性を検証したい。