

ビジュアルプログラミング学習システム

仁科智晴*1・安岡航*1・石川智久*2・吉川桂太郎*2・鎌田洋*2
Email: b1436309@planet.kanazawa-it.ac.jp

*1: 金沢工業大学情報 フロンティア学部 メディア情報学科
*2: 金沢工業大学工学研究科システム設計工学専攻

◎Key Words プログラミング, 学習

1. はじめに

近年様々なことがコンピュータシステム上で行われるようになり、プログラマの需要が高まっている。2020年に小学校プログラミング教育必修化が検討されている程、プログラミング学習が重要となっている。しかし、プログラミング学習を難しいと感じている人も多く、プログラミング学習をサポートするシステムが必要である。

本研究ではプログラミングで図形を表示させ視覚的かつ本格的に学習ができるシステムの開発を行った。また、比較的簡単にプログラミングの学習ができる Processing⁽¹⁾をベースとした。

2. 従来の学習システムと課題

従来のプログラミング学習方法に教科書やe-learningがあるが、プログラミングが苦手な人にとってはそれらによる自学自習は難しく、入力したパラメータによってどのような処理が行われるのかを視覚的に捉えにくい。そのため、分かりにくさから苦手意識に繋がることが考えられる。

学習方法の一つに「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」⁽²⁾というものがある。プログラミングの授業を実験形式で行い、アニメやゲームなどを題材とし、学生が興味を持って学習を行うことができる。しかし、この方法は興味を持ってもらうことには効果的だが、実践的なプログラミング技術を身に付けることができない。

学習システムとして、自動的に穴埋め問題を生成する自主学习システム⁽³⁾がある。このシステムは用意されたプログラムから自動的に穴埋め問題を作成するシステムである。このシステムでは問題を回答するごとに学習者が苦手としている傾向を分析し、毎回異なるデータを用いて学習者に合わせた問題を作成する。これによって効果的に学習が行える。しかし、このシステムはプログラミングがある程度できる人には効果的であるが、プログラミングが苦手な人には難しいと思われる。

図1に自学自習かつ視覚的に学習できるScratch⁽⁴⁾というシステムを示す。これはキャラクターの動きを選択し実行するだけでキャラクターを動かすことができ、思い通りに動かすためにはどのように動作を組み合わせればよいかといったアルゴリズムの学習を行うことができる。そのため関数を覚えることなく誰でも

簡単に使用することができる。しかし、このシステムでは実際のプログラムに触れることができない。本格的にプログラミングを学習するためには次の段階の学習をする必要がある。

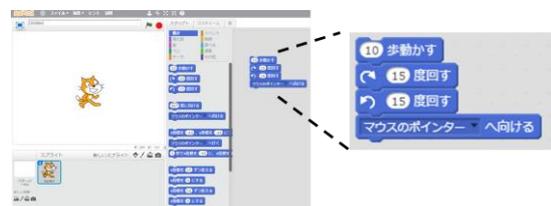


図1 Scratch⁽⁴⁾

3. 前研究と課題

前研究⁽⁵⁾⁽⁶⁾でビジュアルプログラミング学習システムを開発した。プログラミング学習をするにおいて、図形を出力するプログラムなら結果を見て分かりやすく学習意欲も維持できると考えた。そのため、図形をプログラミングで描く問題が出題し、指示通りの図形を出力するプログラミングを行うシステムとした。図2に、システムの基本処理を示す。このシステムは学習者が問題文に沿ってプログラミングを行い、出力された画像と正解のプログラムの画像を比較し正否判定をする。学習者が書くプログラムはプログラムの主要部分のみで、初期化と終了化や画像出力に関するプログラムはあらかじめ記述されている。

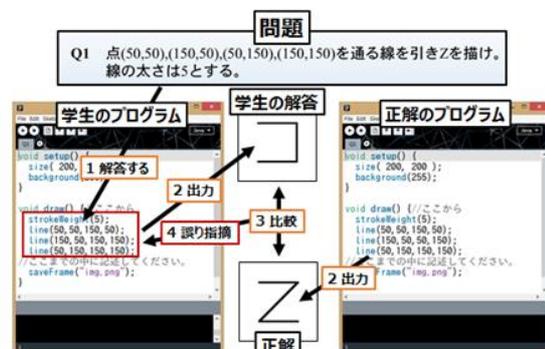


図2 システムの基本処理

図3に画面は学習者の回答に対する判定画面の例を示す。この画面では2つの画像を1ピクセル単位で比較し、それぞれの画像の違いを出力する。これによって学習者が出力した画像と正解の画像で異なる部分が

赤く表示され、どこが間違っているのかを見ることができると効果的に学習することができる。



図3 回答の判定画面例

しかし、このシステムは少しプログラミングが分かってきた人が更なる学習をするためのものである。そのため、問題の難易度も少し高めであり、プログラミングのやり方が全く分からない人には使うことが難しい。これからプログラミング学習を始める人に対してはより初心者向けの問題やプログラミング自体が簡単に行えるシステムが必要である。

4. 本システムの基本的な考え方

プログラミング学習に苦手意識を感じる理由として、プログラミング言語に馴染みが無くコードの書き方がよく分からないことや難しくそうに見えてしまうことが挙げられる。また、プログラミングでは一文字打ち間違えただけでエラーが発生してしまい正常に動作しなくなる。ミスが無いかをチェックしたり修正したりすることの面倒くささからも学習意欲の低下や苦手意識に繋がる。

そのため、前研究のシステムで学習する前段階としてより初心者向けのプログラミングのビジュアル学習システムを新たに作成した。図4に学習システムを示す。この学習システムはパラメータを選択するだけでプログラミングをすることができる。キーボードから直接入力する必要がないため入力ミスによるエラーが無く、プログラミングのやり方が全く分からない人でも操作することができる。また、指示された通りに図形を描き正解の画像と比較する機能も実装し、正しくプログラミングができていないか確認することができる。

また、前研究のシステムで用いられている問題は応用問題が多く、自由記述形式のプログラミングの練習に用いるのは困難である。そのため、より簡単な問題を用意し自分でプログラムを記述することの練習となるようにした。

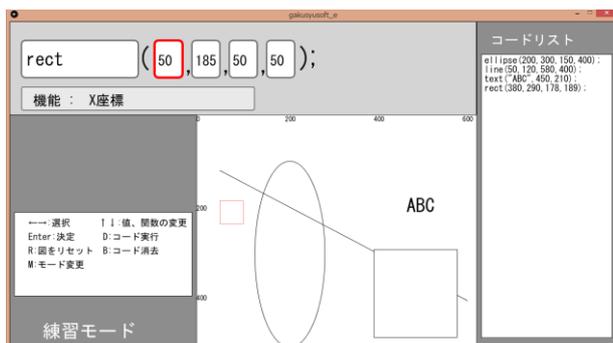


図4 新しい学習システム

5. 本システムの構成

5.1 入力方法

関数やパラメータを選択し実行することができる。現在選択している項目は赤い縁で表示され、方向キー左右で移動することができる。移動の様子を図5に示す。また、方向キー上下を押すことによって現在選択している項目の値や関数を変化させることができる。また、数値を選択している際には数字キーによる入力も可能である。関数、値の変更を図6に示す。



図5 選択の移動

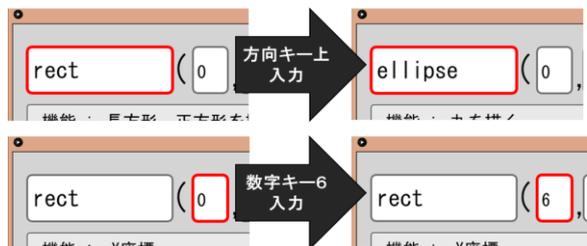


図6 関数、値の変更

入力後、ENTERキーを押すと図7に示すようにプログラムがコードリストに送られる。その後、実行することでコードリスト内のプログラムを実行画面で実行する。また、コードリストに複数のデータがある場合は上から順に1つずつ実行されていく。

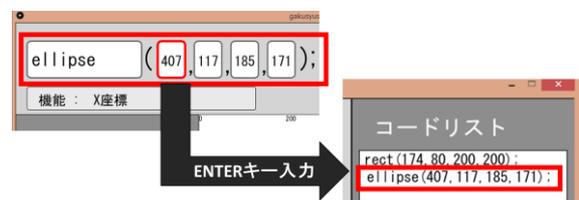


図7 コードリスト

5.2 練習モード

自由にプログラミングを行い、プログラミングの仕方を練習するモードである。始めにこのモードでプログラムを実行する練習をしてもらい感覚をつかんでもらうのが目的である。テストモードよりも実行画面が広く、多くの図形を一度に表示が可能だ。

5.3 テストモード

テストモードは、練習モードである程度慣れてきた人に向けたモードで、指示通りにプログラミングができるかを確認するモードである。図8にシステム概念図を示す。学習者は問題を見て入力と実行をし、正解の画像と比較する。答え合わせには前研究のシステムを使用しており、実行画面と見本画面を1ピクセルずつ比較し、一致した部分は黒で一致しない部分は赤で表示する。比較用の画面を別に用意したことにより確認しやすくなり、間違っていた部分が明確に分かるため学習がしやすくなったと考えられる。また、図形が完全に一致していた場合は正解と表示され、間違っ

いた場合は不正解と表示される。

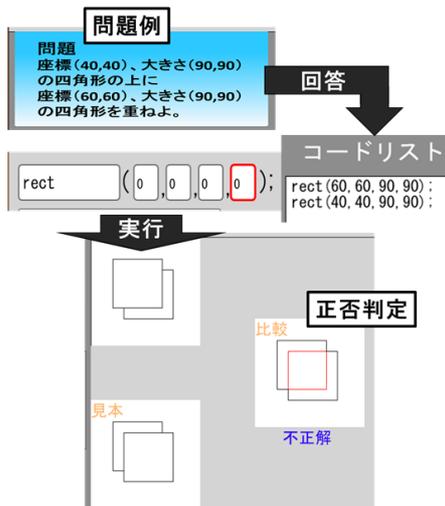


図8 テストモードのシステム概念図

5.4 テストモード問題例

テストモードの問題は基本的な機能や座標に関する問題である。また、図形を表示する順番や途中で線の太さを変えることによってプログラミング記述の順序によって結果が変わることの学習となっている。以下に問題例を示す。

(1) 図形を一つ表示する問題

図形を一つだけ表示し、基本的な使い方や座標、大きさ指定の練習をする問題を出題する。

問題：

座標 (50,50)、大きさ (40,40) の四角形を作成せよ

解答例：

```
rect(50,50,40,40);
```

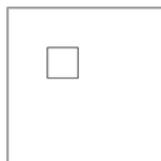


図9 解答図

(2) 複数の図形を表示する問題

図形を複数同時に表示させる問題を出題する。

問題：

点 (40,60) (160,60) (40,140) (160,140) を通る線を引き「Z」を描け。

解答例：

```
line(40,60,160,60);
line(160,60,40,140);
line(40,140,160,140);
```



図10 解答図

(3) 正しい順番で表示する問題

図形を表示や線の太さ指定を行う順番によって結果が変わることを練習する問題。

問題：

座標 (120,40)、大きさ (40,40)、線の太さ 2 の四角形と座標 (40,90)、大きさ (70,70)、線の太さ 5 の四角形を作成せよ。

解答例：

```
strokeWeight(2);
rect(120,40,40,40);
strokeWeight(5);
rect(40,90,70,70);
```

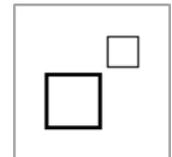


図11 解答図

5.5 前研究の問題の改善

前研究はフラクタル図形や漫画表現を用いた取り掛かりやすく楽しく取り組める内容であった⁽⁶⁾。しかし、プログラミング自体の難易度は高く、学習者のモチベーションが維持できないため、基礎から発展に至る問題をきめ細かく充実することが課題となった。そこで、新たに初学者でもモチベーションを維持しながら学習が可能である、テキスト表現を用いた簡単な問題を作成した。問題は空欄に当てはまる正解を 4 つの選択肢からを選択させる選択形式や記述問題である。問題には用いている関数の種類が少ない上複雑な問題がないため初心者にも容易に問題に取り組むことが出来る。以下に問題例を示す。

(1) 文字の表示に関する問題

解答図を図12に示す。

問題：

下図の実行結果になるように選択肢①～④から□に当てはまる関数を選択せよ。

```
background(255);
fill(0);
□("abcde", 0, 0, 40, 30);
```

選択肢

①textBox ②spot ③text ④printf

解答例：

```
Background(255);
fill(0);
text("abcde", 0, 0, 40, 30);
```

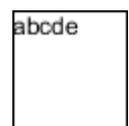


図12 解答図

(2) フォントサイズを指定する問題

問題：

(1)において20のフォントサイズを指定するため選択肢①～④から□に当てはまる関数を選択せよ。

```
□(20);
Background(255);
fill(0);
text("abcde", 0, 0, 40, 30);
```

選択肢

- ①textSize ②fontSize
③textFont ④sizeFont

解答例：

```
textSize(20);
Background(255);
fill(0);
text("abcde", 0, 0, 40, 30);
```

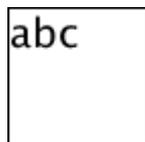


図13 解答図

(3) 文字の入力範囲に関する問題**問題：**

(2)の答えではdとeが表示されなくなりました。その理由は文字の大きさが指定範囲を超えてしまったからである。指定範囲を広げるためには「text("abcde", 0, 0, 40, 30);」の□で囲まれた数字を一定以上大きくする必要がある。dとeを表示できる最低限以上の数字に変更せよ。

解答例：

```
textSize(20);
Background(255);
fill(0);
text("abcde", 0, 0, 60, 30);
```

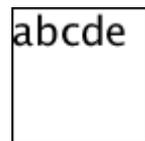


図14 解答図

6. 評価実験

9名の被験者にシステムを使用してもらいアンケートを回答してもらった。評価段階は5段階とし「そう思う」を5点、「少しそう思う」を4点、「どちらとも言えない」を3点、「あまりそう思わない」を2点、「そう思わない」を1点とした。結果を表1に示す。

表1 アンケート結果

評価レベル	Q1	Q2	Q3	Q4
そう思う 5点	5人	6人	5人	2人
すこしそう思う 4点	3人	2人	4人	6人
どちらとも言えない 3点	0人	1人	0人	1人
あまりそう思わない 2点	1人	0人	0人	0人
そう思わない 1点	0人	0人	0人	0人
平均点	4.3点	4.6点	4.6点	4.1点

- Q1 システムの入力方法を使いやすいと感じる
Q2 学習モードの画面比較は分かりやすい
Q3 プログラミングを学習できていると感じる
Q4モチベーションを維持しながら学習ができる

7. 考察

アンケートをとった結果、全体的に高い点数となったが、入力方法の使いやすさに関しては低い評価もあった。低い評価となった理由として、関数を選ぶ際に何回もキーを押して選択するのが面倒であることが考えられる。また、キー操作以外にもマウス操作にも対応したほうが良いといった意見も挙げられているため、更に使いやすく改良していく必要がある。

8. まとめ

視覚的かつ本格的にプログラミング学習ができるシステムを試作した。

システムは入力がしやすく画面比較も見やすいものにできた。使用できる関数や機能もより増やしていきたいが、機能を増やし過ぎることによって複雑になり使い方が難しくなることも懸念されるため慎重に制作をする必要がある。今後はより学習しやすくなる機能の追加を行い改良していきたい。

参考文献

- (1) Processing.org, <https://processing.org/>, 2017年2月1日取得。
- (2) 今仁順也, 長名優子, 菊池真之, 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).
- (3) 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).
- (4) Scratch -Imagine, Program, Share, <https://scratch.mit.edu/>,
- (5) 西川和隆, 鎌田洋: 「プログラミングのビジュアル学習システム」, 平成28年度 金沢工業大学 大学院工学研究科 システム工学設計専攻修士論文, pp.1-25 (2017)
- (6) 宮谷内雄太, 高橋涼, 鎌田洋: 「プログラミングのビジュアル学習システム～フラクタル図形と漫画表現の応用～」, 平成28年度 金沢工業大学 情報フロンティア学部 メディア情報学科 金沢工業大学卒業論文, pp.1-65 (2017)