

錯視を題材にしたプログラミング学習システム

河合康一郎*1・砂長谷泰樹*1・下口尚佑*1・鎌田洋*1

Email: kamada@neptune.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

◎Key Words 錯視, プログラミング, 学習システム

1. はじめに

現代では、情報化が進み様々なところでプログラミングが使われている。そして2020年には、小学校でプログラミングが必修授業になる予定があるなど⁽¹⁾、身近なものになっている。しかし、プログラミングを使いこなすスキルを養うためには、自主的に学習することが必要だと考えられる。そこで、プログラミングを自主学習しやすいシステムを作成したいと考えた。

2. 従来のシステムと問題点

従来のプログラミング学習システム⁽²⁾では、提示された問題文に沿って直接プログラムを記述し、正誤を確かめるという、図1に示したシステムになっており、初心者でもわかるような難易度が低い問題が出題される。しかし、プログラミング学習は、初めて触れる人にとって興味を持ちにくいものであると考える。

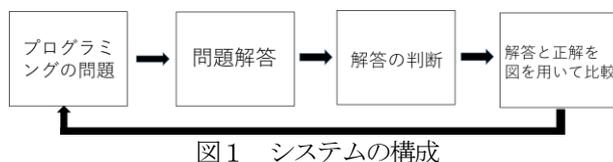


図1 システムの構成

3. 本システムにおける解決方法

プログラミングに興味を持ちにくいという問題点を解決するために、プログラミングの題材を錯視⁽³⁾にし、錯視を用いた図や色、画像をシステムに取り入れることで、視覚的にも楽しむことができ⁽⁴⁾、さらには錯視の事象も学習することができるシステムの作成をする。

4. 題材にした錯視

本システムに題材として使用した錯視の説明を表1に示す。

表1 錯視の説明

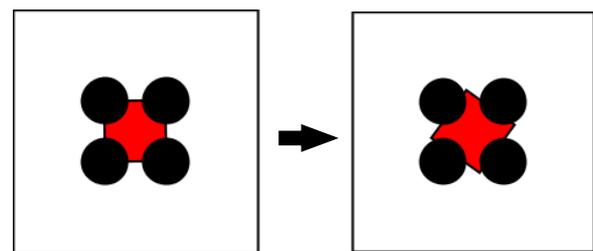
番号	名称	説明
1	四角形の拡大・縮小	四角形の周りに円を配置することで四角形が伸縮しているように見える錯視である。
2	フットステップ	縞模様の背景をオブジェクトが等速運動するとき、それが別の動きに見える錯視である。
3	ハーマン格子	細い白い格子の十字路のどこか1つの交差点を見ると周りの交差点で黒い影が見えるようになる錯視である。

4.1 「四角形の拡大・縮小」

四角形の周りに円を配置することで四角形が伸縮しているように見える錯視である。本システムでは、図2で示したパーツを使用し図3の錯視を作成している。



図2 使用されているパーツ



四角形が小さく見える 四角形が大きくなる

図3 「四角形の拡大・縮小」の錯視の現象

4.2 「フットステップ錯視」

縞模様の背景をオブジェクトが等速運動するとき、それが別の動きに見える錯視である。本システムでは、図4で示したパーツを使用し図5の錯視を作成している。

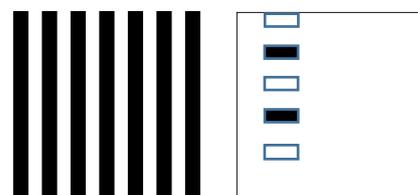
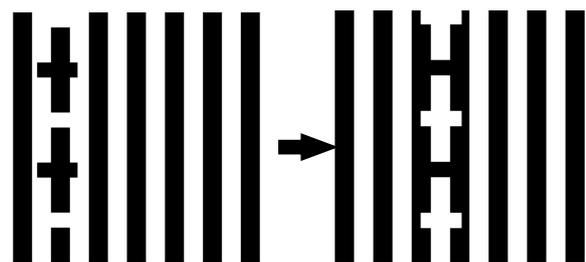


図4 使用されているパーツ



黒が白よりも前進している 白が黒よりも前進している

図5 「フットステップ錯視」の現象

4.3 「ハーマン格子」

細い白い格子の十字路のどこか1つの交差点を見ると周りの交差点で黒い影が見えるようになる錯視である。図6の右の図のように、中心の赤い点を見ることによって周りの交差点に点が見える。

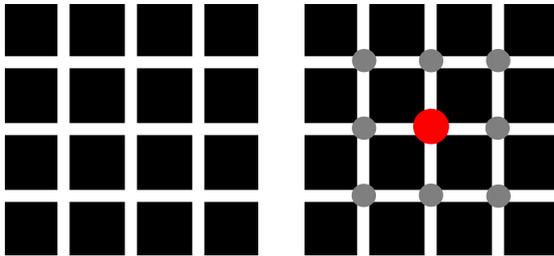


図6 「ハーマン格子」の錯視の現象

5. 本システムの構成

プログラミングで錯視を作っていく過程でプログラムについての問題を提示し、それを解答してもらうことによってプログラミングと錯視について学習することができるシステムである。問題の解答方法としては、錯視を作成するプログラムの1部を空欄にし、何が入るのかを選択肢の中から選び解答を入力してもらう。その解答結果と正解を比較できるようにし、解答の正誤を認識することができる。そして本システムの実現にはプログラム言語 Processing⁽⁵⁾を用いている。

6. 各錯視に関する問題

6.1 「四角形の拡大・縮小」

「四角形の拡大・縮小」の錯視を作成していく問題の手順を下記に示す。また、それらの実行結果を図7に示す。そして、すべての問を解答することで図8のプログラムが完成する。

- 問1 サイズが縦横に200ずつの画面の作成
- 問2 1つの四角形の作成
- 問3 四角形の色の変更
- 問4 四角形を回転させる
- 問5 四角形の4つの角に円を作成

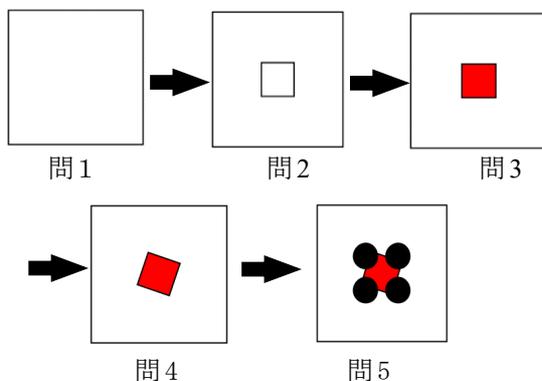


図7 「四角形の拡大・縮小」の錯視の作成過程

```

1 float xspeed = 2.5;
2 float r;
3 void setup() {
4   問1 size(200, 200);
5   smooth();
6 }
7
8 void draw() {
9   colorMode(HSB, 360, 100, 100);
10
11   stroke(0, 0, 0);
12   strokeWeight(1);
13   background(0, 0, 100);
14
15   pushMatrix();
16   translate( 100 , 100 );
17   問3 fill(0, 100, 100);
18
19   問4 rotate(radians( r ));
20   問2 rect(-25, -25, 50, 50);
21   popMatrix();
22
23   noStroke();
24   fill(0, 0, 0);
25
26
27   問5 ellipse
28     (80, 80, 37, 37);
29
30   問5 ellipse
31     (120, 80, 37, 37);
32
33   問5 ellipse
34     (80, 120, 37, 37);
35
36   問5 ellipse
37     (120, 120, 37, 37);
38
39   r = r + 1;
40   if( r > 360) {
41     r = 0;
42   }

```

図8 四角形の拡大・縮小のプログラム

6.2 「フットステップ錯視」

「フットステップ錯視」の錯視を作成していく問題の手順を下記に示す。また、それらの実行結果を図9に示す。そして、すべての問を解答することで図10のプログラムが完成する。

- 問1 サイズが縦横に200ずつの画面の作成
- 問2 1つの四角形の作成
- 問3 四角形の色の変更
- 問4 四角形を回転させる
- 問5 四角形の4つの角に円を作成

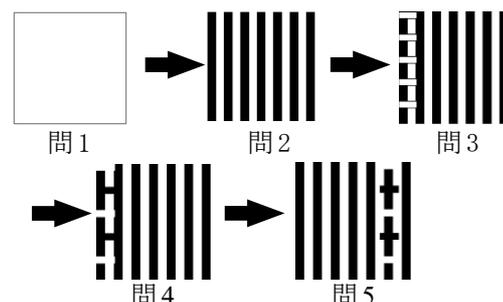


図9 「フットステップ錯視」の作成過程

```

1 int x=0;
2 int w=15;
3 void setup(){
4
5     size(200,200);
6
7     frameRate(30);
8
9 }
10
11 void draw(){
12     noStroke();
13     colorMode(HSB, 360, 100, 100);
14     fill(0,0,0);
15 問1 background(0,0,100);
16 問2 for (int i=0; i< width; i+=(w*2)){
17         rect(i, 0, w, 200);
18     }
19
20     fill(0,0,100);
21 問3 rect(x, 0, 32, 12);
22 問4 fill(0,0,0);
23 問3 rect(x, 40, 32, 12);
24     fill(0,0,100);
25 問3 rect(x, 80, 32, 12);
26 問4 fill(0,0,0);
27 問3 rect(x, 120, 32, 12);
28     fill(0,0,100);
29 問3 rect(x, 160, 32, 12);
30
31 問5 
32     (x++ > width ) x = 0;
33     if(frameCount==181)exit();
34     else saveFrame
35     ("../../../../比較/Quiz/Q5/data/output/img"
36     + frameCount + ".png");
37 }

```

図10 フットステップ錯視のプログラム

```

1 int w=20;
2 float r;
3 float y=0, x=0;
4 void setup(){
5
6 問1 size(200,200); //画面サイズ
7
8     frameRate(60);
9
10    colorMode(HSB, 360, 100, 100);
11
12    smooth();
13 }
14
15 void draw(){
16     stroke(0,0,0);
17     strokeWeight(1);
18     background(0,0,100);
19
20
21     stroke(0,0,0);
22     strokeWeight(1);
23
24
25 問3 for (y=0;y<=150;y=y+50){
26
27 問4 for (x=0;x<=150;x=x+50)
28     {
29         fill(0,0,0);
30         問2 rect(x, y, 40, 40);
31     }
32
33
34     if(frameCount==181)exit();
35     else saveFrame
36     ("../../../../比較/Quiz/Q4/data/seikai/img"
37     + frameCount + ".png");
38 }

```

図12 ハーマン格子錯視のプログラム

6.3 「ハーマン格子」

「四角形の拡大・縮小」の錯視を作成していく問題の手順を下記に示す。また、それらの実行結果を図11に示す。そして、すべての問を解答することで図12のプログラムが完成する。

- 問1 サイズが縦横に200ずつの画面の作成
- 問2 1つの四角形の作成
- 問3 四角形の色の変更
- 問4 四角形を回転させる

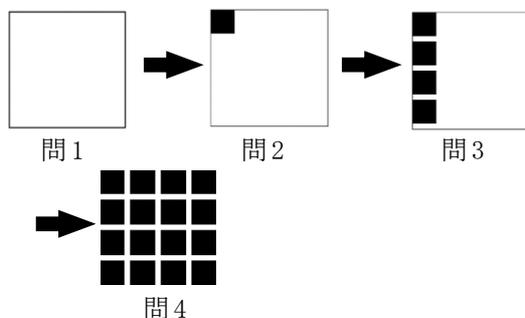


図11 「ハーマン格子」の作成過程

7. 本システムの操作

図13の「使用方法」をクリックすることで図14が表示される。また、「錯視1(四角形の拡大・縮小)」、「錯視2(フットステップ)」、「錯視3(ハーマン格子)」のいずれかをクリックすることで、それぞれの錯視の問題メニューに移動する。(錯視1の問題メニューを図15に示す。)図15の問題をクリックすることで問題を進めると、図16のように問題と「解答する」や「比較する」ボタンが表示される。解答するボタンを押すことでプログラムが表示され、図17のように、指定された箇所を図16の選択肢の1つを入力し実行することで解答することができる。また、比較するボタンをクリックすることで実行結果と正しい実行結果が表示され、図18のように比較することができる。



図13 メニュー画面

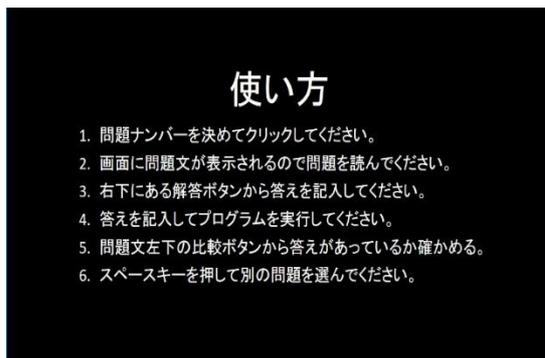


図 14 使用方法



図 15 四角形の拡大・縮小の問題メニュー

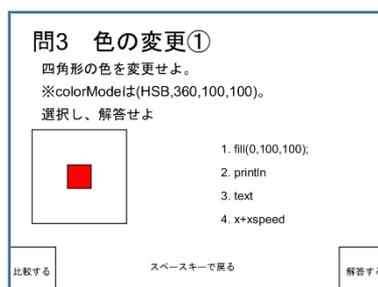


図 16 四角形の拡大・縮小の問3

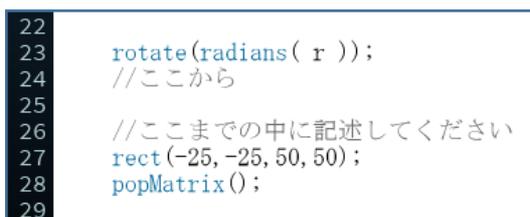


図 17 四角形の拡大・縮小の回答箇所

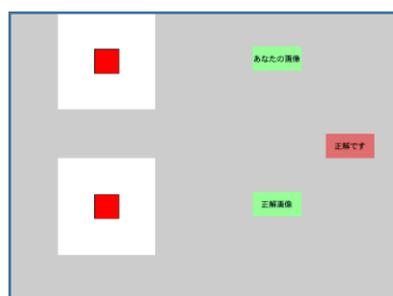


図 18 四角形の拡大・縮小問3の実行結果の比較

8. 評価実験

本システムの調査を大学生 15 人に、アンケート形式で行った。各項目の評価段階は 5 段階とし、「非常にそう思う」を 5 点、「全くそうは思わない」を 1 点とし、アンケート結果と点数の平均を表 1 に集計した。

表 1 評価実験の結果

評価項目	平均点	5点	4点	3点	2点	1点
Q1.取りかかりやすさ	3.5	2人	7人	3人	3人	0人
Q2.学習の促進	3.3	0人	7人	6人	1人	1人
Q3.楽しく学習できるか	3.2	0人	6人	6人	3人	0人
Q4.錯視の知識の習得	3.1	0人	5人	6人	4人	0人
Q5.システムの使用感	3.1	0人	6人	5人	3人	1人

9. 考察

表 1 の Q1.取りかかりやすさ(興味)の評価結果より、錯視をプログラミングの学習に取り入れることは、プログラミング学習を取りかかりやすいものとする上で、有用な傾向にあることが分かった。しかし、Q4.錯視の知識が身につく、の評価結果より、錯視についての知識を身につけるには不十分であることが分かった。その要因として、問題数の不足と、こういった錯視であるかという説明がされていないことが挙げられる。また、こういった錯視を作成しているのか、最初から分かっていた方がよいという意見があった。

10. おわりに

本システムでは、従来のシステムにはなかった、錯視という題材を用いることで、プログラミング学習に興味を持ってもらい、プログラミング学習に取りかかりやすいものにした。今後の課題は、問題数を増加させることと、設問終了後、錯視ごとにその錯視についての詳しい説明をすることで、よりよいシステムを構築できると考える。

謝辞

本研究は、JSPS 科研費 18K02836 の助成を受けたものです。

参考文献

- (1) 文部科学省：“プログラミング教育”，http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afidfile/2018/03/30/1375607_01.pdf(2019.6.13 取得)。
- (2) 井坂雄佑, 桂川大輝, 青木拓也, 仁多見光, 鎌田洋:“プログラミングの学習システム”, CIEC 研究会報告集, Vol.10, pp.47-52 (2019).
- (3) 北岡明佳:“錯視入門”, 朝倉書店(2010).
- (4) 坪田将輝, 峠大生, 鎌田洋:“錯視に関する学習システムの開発”, CIEC 研究会報告集, Vol.10, pp.35-40 (2019).
- (5) Processing.org: “Processing”, <https://processing.org/> (2019.6.13 取得)。