

# 力学を題材にしたプログラミングの学習システム

井坂雄佑\*1・桂川大輝\*1・青木拓也\*1・仁多見光\*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 プログラミング, 力学, 学習

## 1. はじめに

小学校でプログラミング学習を通じた論理的思考力の育成が求められており<sup>(1)</sup>, プログラミングに関する注目度は上がっている。しかし, プログラミングは複雑という印象を持っている人が多い。プログラムの初心者にはまずプログラムが簡単な命令, 処理の集まりだということを認識してもらうことが必要である。

## 2. 従来のシステムの問題点

従来の研究<sup>(2)</sup>ではプログラミング初心者向けのシステムとなっており, 提示された問題文に沿って直接プログラムを記述し出力した画像を正解の画像と比較して正誤を確かめるシステムとなっている。しかし, 初心者にとっては学習したプログラムを組み合わせることは難しく, 従来のシステムでは簡単なプログラムの命令や処理を組み合わせることを学びにくい。

## 3. 本システムにおける解決方法

本システムはプログラムの組み合わせを理解させることを目的とした。力学を題材にして運動する物体のプログラム作成の問題を段階に分けて, 順番に作成させることによりプログラムの組み合わせを学習できるようにした<sup>(3)</sup>。我々の身近にある物体の運動を題材にすることで興味を持ってもらうようにした。さらに, 選択肢から選んで記述する形式にすることで初心者を取り組みやすくした。

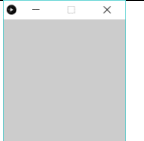
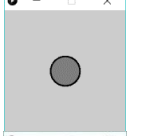
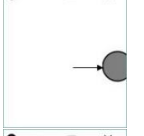
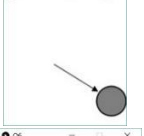
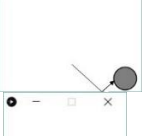
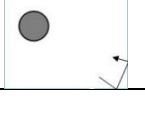
## 4. 本システムの構成

本システムは, 前システム<sup>(3)</sup>における学習教材を大幅に拡張した。本システムの学習教材を表 1 に示す。物体が停止した状態から段階的に等速直線運動, 跳ね返り, 重力や抵抗による加速と減速をプログラムで実現する教材にした。本システムで学習できる項目と具体的な処理, 正解画像を学習教材毎に表 2 から表 6 に示す。また各学習教材の完成プログラムを図 1~5 に示し, 各問において記述する部分を明示した。学習に用いるプログラミング言語は Processing<sup>(4)</sup>であり, 本システムも Processing で実現した。学習教材は, Processing の無償のサンプルプログラム The Nature of Code<sup>(5)</sup>を利用した。

表 1 学習教材一覧

名前	主な処理	難易度
壁に反射する物体	繰り返し処理で座標値を更新し, 条件文を用いることで等速直線運動している物体が壁に当たると反射する	低
壁に反射するクラス物体	PVector を用いて上記の処理を行い, 物体に残像を付ける	高
風と重力を受けた物体	PVector を用いて風と重力を再現し, 物体を減速させる	高
連続運動の物体	物体がウィンドウの端で座標変換する	低
加速する連続運動の物体	物体がウィンドウの端で座標変換し, 加速する	高

表 2 壁に反射する物体

問	学習項目	処理	正解画像
問 1	setup	ウィンドウのサイズを指定する	
問 2	円を表示する関数	座標値と大きさを指定して円を表示する	
問 3	水平方向の繰り返し処理	繰り返し処理を用いて x 座標のみを更新し, 円を等速直線運動させる	
問 4	水平方向と垂直方向の繰り返し処理	繰り返し処理を用いて x 座標と y 座標を更新し, 円を等速直線運動させる	
問 5	垂直方向の判定	円がウィンドウの上端か下端で跳ね返るようにする	
問 6	水平方向の判定	円がウィンドウ両端で跳ね返るようにする	

```

float x = 100;
float y = 100;
float xspeed = 2.5;
float yspeed = 2;

void setup() {
  size(200, 200);
  smooth();
}

void draw() {
  background(255);


  x = x + xspeed;
  y = y + yspeed;

  if ((x > width) || (x < 0)) {
    xspeed = xspeed * -1;
  }
  if ((y > height) || (y < 0)) {
    yspeed = yspeed * -1;
  }

  stroke(0);
  strokeWeight(2);
  fill(127);
  ellipse(x, y, 48, 48);
}
    
```

図1 学習教材1の完成プログラム

表3 壁に反射するクラス物体

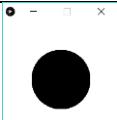
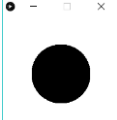

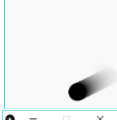
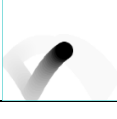
問	学習項目	処理	正解画像
問1	画面のサイズ	ウィンドウのサイズを指定する	
問2	円を表示する関数	クラスを利用した円の表示	
問3	円を移動	クラスを利用した座標の加算	
問4	一定条件で起こる動き	if文を利用して移動方向を反転する	
問5	プログラムを組み合わせていくこと	画面に白く透けた色を塗ることで残像のように見せる	

```

PVector position;
PVector velocity;
void setup() {
  size(200,200);
  background(255);
  position = new PVector(100,100);
  velocity = new PVector(1.5,3);
}
void draw(){
  noStroke();
  fill(255,10);
  rect(0,0,width,height);
  position.add(velocity);
  if ((position.x > width) || (position.x < 0)) {
    velocity.x = velocity.x * -1;
  }
  if ((position.y > height) || (position.y < 0)) {
    velocity.y = velocity.y * -1;
  }
  stroke(0);
  fill(175);
  ellipse(position.x,position.y,16,16);
}
    
```

図2 学習教材2の完成プログラム

表4 風と重力を受けた物体

問	学習項目	処理	正解画像
問1	円を表示する関数	円を表示する	
問2	クラスの利用	座標値にベクトルを使って円を表示する	
問3	繰り返し処理による円の等速直線運動	円の座標値を繰り返し処理で更新することで等速直線運動させる	
問4	条件文	条件文を用いて円を等速直線運動させる	
問5	繰り返し処理による円の減速	円の速さを繰り返し処理で更新することで円を減速させる	

```

PVector position;
PVector velocity;
PVector deceleration;
PVector gravity;
int a=30;
void setup() {
  size( 200, 200);
  background(255);
  deceleration = new PVector(0.998, 0.998);
  position = new PVector(width/2, height/2);
  velocity = new PVector(200.0/60, 10.0/60);
  gravity = new PVector(0.0/60, 9.8/60);
}
void draw() {
  position.add(velocity);
  velocity.x = velocity.x * deceleration.x;
  velocity.y = velocity.y * deceleration.y;
  velocity.add(gravity);
  fill(255,20);noStroke();rect(0,0,width,height);
  stroke(0); fill(0);
  ellipse(position.x, position.y, a,a);
  if( (position.y+a/2 > height) || (position.y-a/2 < 0) ){
    velocity.y = -1*velocity.y;
  }
  if( (position.x+a/2 > width) || (position.x-a/2 < 0) ){
    velocity.x = -1*velocity.x;
  }
}

```

図3 学習教材3の完成プログラム

表5 連続運動の物体

学習項目	処理	正解画像
問1 setup	ウィンドウサイズの指定	
問2 クラスを定義	Velocity と position の定義	
問3 変数の宣言	円の初期位置と速度を定義	
問4 円の速度を計算	円の速度を計算して x 座標と y 座標を求める	
問5 円を描く	円をウィンドウ上に描く	
問6 x 座標の判定	円の x 座標がウィンドウの左端にいったら x 座標を 0 にする	
問7 y 座標の判定	円の y 座標がウィンドウの下端にいったら y 座標を 0 にする	

```

Mover mover;
void setup() {
  size(640,360);
  mover = new Mover();
}
void draw() {
  background(255);
  mover.update();
  mover.checkEdges();
  mover.display();
}

class Mover {
  PVector position;
  PVector velocity;
  Mover() {
    position = new PVector(300, 0);
    velocity = new PVector(-1, 2);
  }
  void update() {
    position.add(velocity);
  }
  void display() {
    stroke(0);
    strokeWeight(2);
    fill(127);
    ellipse(position.x, position.y, 180, 180);
  }
  void checkEdges() {
    if (position.x > width) {
      position.x = 0;
    }
    else if (position.x < 0) {
      position.x = width;
    }
    if (position.y > height) {
      position.y = 0;
    }
    else if (position.y < 0) {
      position.y = height;
    }
  }
}

```

図4 学習教材4の完成プログラム

表6 加速する連続運動の物体

学習項目	処理	正解画像
問1 setup	ウィンドウのサイズを指定する	
問2 円を表示する関数	クラスを利用した円の表示	
問3 円の移動	クラスを利用した物体の移動	
問4 円の加速	クラスを利用して下方向へ加速する	
問5 条件文	移動する物体が画面端で上下逆の座標に移動する	
問6 クラスの利用	クラスを利用することで物体の加速に上限を設定する	

```

PVector position;
PVector velocity;
PVector acceleration;
float topspeed;
void setup(){
  size(200,200); 問1
  position = new PVector(width/2, height/2);
  velocity = new PVector(0, -1); 問3
  acceleration = new PVector(-0.003, 0.02); 問4
  topspeed = 10;
}
void draw(){
  noStroke();
  fill(255,99);
  rect(0,0,width,height);
  velocity.add(acceleration);
  velocity.limit(topspeed); 問6
  position.add(velocity);
  if (position.x > width) {
    position.x = 0;
  }
  else if (position.x < 0) position.x = width; 問5
  if (position.y > height) position.y = 0;
  else if (position.y < 0) position.y = height;
  stroke(0);
  strokeWeight(2);
  fill(127);
  ellipse(position.x, position.y, 48, 48); 問2
}

```

図5 学習教材5の完成プログラム

## 5. 評価実験とその結果

本システムを用い、学生9人にアンケートを行った。スクリーンで本システムを実行し、5段階評価(5:高評価~1:低評価)のアンケートに回答してもらった(表7)。

表7 本システムについてのアンケート

項目	5 良い	4 やや 良い	3 どちら らで もな い	2 やや 悪い	1 悪い	平均 点
学習意欲の向上	2	4	2	1	0	3.8
苦手意識の払拭	1	3	5	0	0	3.6
システムの使いやすさ	2	2	3	2	0	3.4
学習の継続性	0	4	3	2	0	3.2
学習の効率性	0	3	4	2	0	3.1
システム全体評価	1	4	3	1	0	3.6

## 6. 考察

システムの全体評価は3.6点であり、良好な評価が得られた。その中で一番評価の高かった項目は「学習意欲の向上」であった。この理由として前回よりも問題数が増えた事で学習範囲が増加したことや自分が過去に誤答した箇所を確認できるようになったことが考えられる。逆に一番評価の低かった項目は「学習の効率性」であった。この理由としては、間違えた際にヒントや解説が特にないことが挙げられる。今後はヒントや解説などの機能を充実させ、初心者が無理なく続けられるようにすることが必要である。

## 7. おわりに

プログラムの初心者のために力学を題材にして段階的に学習させるシステムを作成した。システム全体として一定の評価を得たが、問題のヒントや解説を追加するなどして今以上に初心者が分かりやすくするための工夫が必要である。

## 参考文献

- (1) 小学校学習指導要領, [http://www.mext.go.jp/component/a\\_menu/education/micro\\_detail/\\_icsFiles/afieldfile/2017/05/12/1384661\\_4\\_2.pdf](http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2017/05/12/1384661_4_2.pdf) (2018年6月15日取得).
- (2) 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).
- (3) 井坂雄佑, 桂川大輝, 仁多見光, 青木拓也, 仁科智晴, 鎌田洋: “力学を題材にしたプログラミングの学習システム”, 平成29年度 北陸地区学生による研究発表会講演論文集, F-1-6, p.105 (2018).
- (4) Processing.org: “Processing”, <https://processing.org/> (2018年6月15日取得).
- (5) ダニエル・シフマン: “NATURE OF CODE”, pp.23-52, ボーンデジタル (2014).