N-031

誤りから学ぶプログラミング学習コンテンツの試作
Prototype for Programming Learning Content from Errors

石川　悠真†　　土肥　紳一‡
Yuma Ishikawa　Shinichi Dohi

1. はじめに

東京電機大学システムデザイン工学部デザイン工学科では 1 年次後期に初めてのプログラミング講義として「コンピュータプログラミング I」を受講する. この科目は必修科目となっている. 毎年受講生の 6 割以上が初回授業アンケートにてプログラミングが初めてと回答している. そして, 最終回授業アンケートでは受講生の半数以上がプログラミング学習の努力をしたいと思うと回答している[1]. 本研究では, 初学者が陥る誤りを調査, 分類し, そのサンプルコードと改良後のサンプルコードをコンテンツとして提供し, 学習者がそのコンテンツを理解し, プログラミング学習に繋がるか調査することを目的としている. つまり, ユーザがその誤りを知ることで, プログラミング学習に対するモチベーションへどう影響するかを調査する. そのためのコンテンツを試作する.

2. Processing について

Processing は, イメージ, アニメーション, そしてインタラクションを生み出すソフトウェアを書くためにあるプログラミング言語である[2]. 画面に楕円を描くとき, Processing の場合ソースコードを 1 行書くだけでできる. 視覚的なフィードバックが即座に得られ, スケッチブックに絵を描く感覚でプログラミングできるように設計されている. Processing は Java で作られており, オブジェクト指向プログラミングの基礎も学習できる. この言語を使用して, デザイン工学科はプログラミング授業の必修科目の中で扱っている[3].

3. 誤りについて

誤りは「実行結果は正しいが, ソースコードは誤っているもの」とする[4]. 例えば, 風船を表現するプログラムを制作する. この図形は授業では取り上げていないが, 授業で教わったことを応用できるように試作したものである. 図 1 で左のソースコードは内容の一部を示した状態である. strokeWeight 関数は線の太さを, fill 関数は図形の塗りつぶしを, ellipse 関数は円の表示を, line 関数は線の表示を, rect 関数は長方形の表示を行う. 一方, 図 3 は図 1 のソースコードの中で, 風船の円を表示するソースコード（図中文字を赤字で記載）とワイヤを表示するソースコード（図中文字を青字で記載）の場所を入れ替えている. 図 1 と図 3 の結果を比較すると, ワイヤの部分が前面に出ているか後面に隠れているかの違いに気づきにくい. 図 2 では, 図 1 の中でワイヤ部分の storkeWeight 関数の数値を 10 に上げた実行結果である. この時, 風船という図形を制作する上では誤りとする. この誤りは, 描画の順序の矛盾を利用している.

```
size(400, 400);
background(204);
ellipseMode(RADIUS);
//風船
strokeWeight(7);
fill(255, 141, 102);
ellipse(200, 150, 50, 50);
//ワイヤ
strokeWeight(2);
line(150, 150, 180, 260);
line(250, 150, 220, 260);
//箱
strokeWeight(2);
fill(141, 255, 102);
rect(180, 260, 40, 30);
```

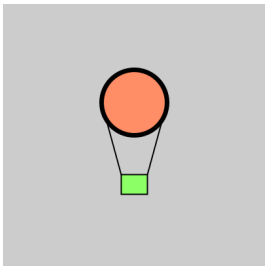

図 1　誤った風船の表示

```
略
//風船
略
//ワイヤ
strokeWeight(10);
line(150, 150, 180, 260);
line(250, 150, 220, 260);
//箱
略
```

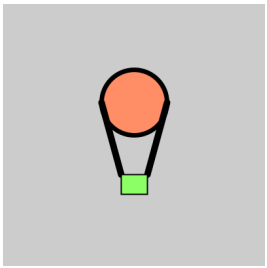

図 2 図 1 でワイヤの部分を強調した実行結果

```
略
//ワイヤ
strokeWeight(2);
line(150, 150, 180, 260);
line(250, 150, 220, 260);
//風船
strokeWeight(7);
fill(255, 141, 102);
ellipse(200, 150, 50, 50);
//箱
略
```

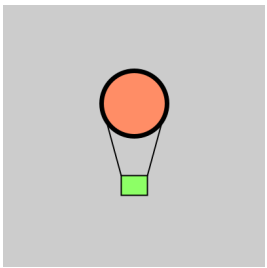

図 3　正しい風船の表示

4. コンテンツの試作

誤りのコンテンツを作成する上で, ソースコードでどういう誤りが想定されるかを分類する. 例えば, 図 1 から図 3 のように描画の順序という内容

† 東京電機大学大学院システムデザイン工学研究科
Graduate School of System Design and Technology, Tokyo Denki University
‡ 東京電機大学システムデザイン工学部　School of System Design and Technology, Tokyo Denki University

や，boolean 型の変数で，変数そのものに真であるか偽であるかを示し，真であるならば「keyPressed == true」と記載をしなくてもよいという内容の誤り，また比較演算子である「<」，「>」，「<=」，「>=」で条件式の中で無駄な演算を行った内容の誤りへと分類する．授業は，教科書の例を入力しながら進行する．教科書の内容と誤りの定義に沿い，過去に出題された試験や練習問題を基に改良後の試作を以下に示す．

```
int balloonX = 250;
int balloonY = 250;
int speed = 10;
void setup(){
  size(500, 500);
  ellipseMode(RADIUS);
}
void draw(){
  background(204);
  if(keyPressed){
    if(key == 'w'){
      balloonY -= speed;
    }
    if(key == 'a'){
      balloonX -= speed;
    }
    if(key == 's'){
      balloonY += speed;
    }
    if(key == 'd'){
      balloonX += speed;
    }
  }
  balloon(balloonX, balloonY);
}
void balloon(int x, int y){
  pushMatrix();
  translate(x, y);
  strokeWeight(2);
  line(-40, 0, -20, 80);
  line(40, 0, 20, 80);
  strokeWeight(4);
  fill(255, 141, 102);
  ellipse(0, 0, 40, 40);
  fill(141, 255, 102);
  rect(-20, 80, 40, 30);
  popMatrix();
}
```

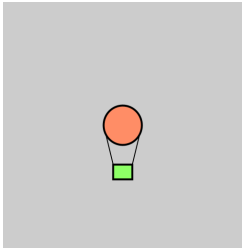

図 4 キーボード入力による風船の移動

図 4 は boolean 型の変数と計算式の短縮形という内容のコンテンツである．keyPressed 変数自身が真か偽か示すため，「==」演算子を使って明示的に比較する必要はない．そのため，「if(keyPressed == true)」と記載することは誤りとして扱う．また，宣言された speed 変数による風船の x 座標と y 座標を変化させる計算式では，短縮形として「+=」や「-=」を利用できる．そのため，「balloonX = balloonX + speed」という記載は誤りではないが，短縮形を推奨する．図 4 で注目した部分を赤字で示した．

```
void setup(){
  size(800, 250);
  background(204);
  ellipseMode(RADIUS);
  for(int i = 0; i < 8; i++){
    balloon(50 + 100 * i, height/2);
  }
}
// 図 4 の balloon 関数の略
```

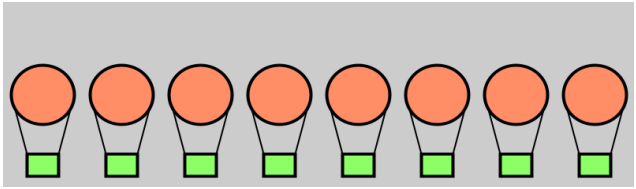

図 5 繰り返しを使った風船の表示

図 5 は条件式という内容のコンテンツである．for 文の中の条件式で「i < 8」を「i <= 8」と記載しても同じ実行結果であるが，ウィンドウの右の外側にもう 1 個風船が生成され誤りである．

5. まとめと今後の課題

本研究では，Processing 言語における誤りのあるソースコードを用いて，サンプルコードと改良後のサンプルコードをコンテンツとして提供することで，学習者がコンテンツを理解し，プログラミング学習に繋がるか調査することを目的としている．誤りとはあくまで「実行結果は正しいが，ソースコードは誤っているもの」とした定義に沿う．本論文では，そのためのコンテンツの試作段階を述べた．

今後は引き続きコンテンツとしてのサンプルコードの試作と，それらを閲覧する Web サイトの構築を行っていく．課題として，試作コンテンツの準備と今後学習者からもらう評価の実施方法の検討が挙げられる．

参考文献

[1] コンピュータプログラミング I，https://dohi.chiba.dendai.ac.jp/~dohi/computer- programming-1/ad/，2023 年 6 月 12 日閲覧

[2]Cassy Reas，Ben Fry，Processing をはじめよう，株式会社オライリージャパン，Vol.57，2016

[3] 土肥 紳一，Processing によるオブジェクト指向プログラミング入門教育の実践，情報教育シンポジウム論文集，pp.134-141，2018

[4] 石川 悠真，土肥 紳一，インストラクショナルデザインに基づいた誤りから学ぶプログラミング学習コンテンツの提案，情報処理学会第 85 回全国大会論文集(4)，pp.665-666，2023