

コンピュータグラフィックス(CG)におけるPBL教育の試み

An attempt of PBL education for the computer graphics

森 文彦*, 菅野 直敏*

Fumihiko Mori* and Naotoshi Sugano *

*玉川大学工学部機械情報システム学科, 194-8610 東京都町田市玉川学園6-1-1

*Department of Intelligent Mechanical Systems, College of Engineering, Tamagawa University,
6-1-1 Tamagawagakuen Machida-shi Tokyo 194-8610

Abstract

A simple computer graphics software: “Processing” is used for an example of PBL education. It was executed through three steps of the following: 1)fundamental image generation, 2)generation of the work using the fundamental images and 3)evaluation of the works by the students. The overview is reported in this paper.

Keywords: computer graphics, programing, PBL, Processing, image

1. はじめに

情報系の学生においては、コンピュータに仕事をさせる技術は必須である。コンピュータに仕事をさせるためには、何らかのプログラムを作成し、実行させることが必要である。

しかしながら、コンピュータの操作に慣れていない新入生に「プログラミムを書いて実行すること」を行ってもらうと、タイピングや画面操作に手一杯となるのみならず、級数計算など抽象的なものが多いためプログラミング習得に拒絶反応を示す学生も少なくない。すなわち、初めてプログラムを学ぶ大学生向けの授業で扱うプログラムは、「計算結果や文字列を出力するもの」が多い。基礎をしっかりと身につけるためには、この種の作業は重要であるが、学生には面白くない。

そこで、より学生の興味を引き出しやすい、視覚に訴える映像を作成するコンピュータグラフィックス(CG)に注目し、プログラミングの導入

授業としてCG用単純ソフト“Processing”(文献1))を採用することによって、c言語において比較的困難とされている「プログラムの形式、for文(繰り返し文)、if文(分岐文)、入れ子構造などの概念」に慣れることを目指した。

ここでは、①基本画像作成の簡単なプログラム群の作成、②それらを用いて静止画像や動画の作品を作成させ、③学生が出来栄を評価し、優劣を競う授業をPBL教育(文献2))の1つとして試みたので、その概要を報告する。

授業は、ガイダンスの後、6回の授業で基本プログラムを動かしてProcessingのプログラムの基礎を学び、その後7回の授業でオリジナル作品を作成して、最終回で発表会を行うという手順で行われた。

2. 基本画像作成プログラム例

作成した画像の表示ウィンドウの座標系は、左

上隅を原点 (0, 0)、右方向をX 軸、下方向をY 軸と定め、表示ウィンドウの大きさは、size メソッド (size(横サイズ, 縦サイズ);) で指定する (図 1)。

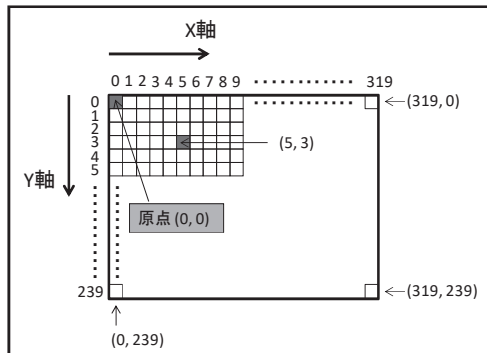
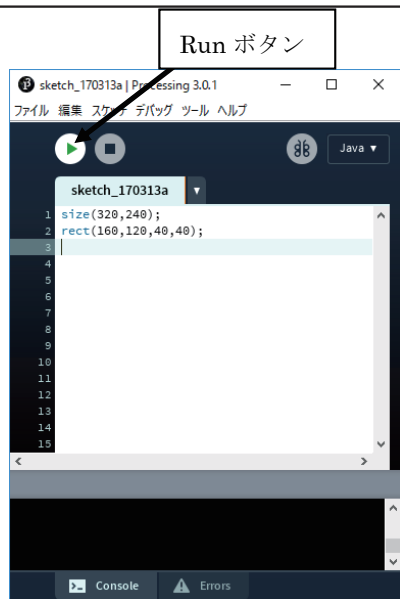
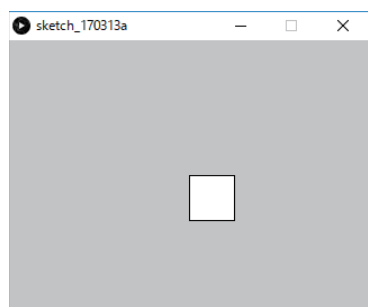


図 1 Processingの座標系

[プログラム例]
size(320, 240); // 表示ウィンドウ
rect(160,120,40,40); // 四角形を描く



a. プログラム例とエディタ画面



b. 実行結果の画面 (表示ウィンドウ)

図 2 エディタ画面と処理結果の画面

四角形を描くには、rectメソッド (rect(左上X座標, 左上Y座標, 横幅, 縦幅);) を用いる。エディタ画面例と実行結果の表示ウィンドウ例を図 2 に示す。

例えば、サイズ横320、縦240の長方形の表示ウィンドウにおいて、画面の中央を左上の角とする正方形を描くためには、図 2 aに示した 2 行のプログラムをエディタに入力して、エディタ画面の左上のRunボタンを押す。

繰り返し文(for)を理解するために、正方形を横に並んだ正方形 5 個を描く問題を与えた。その問題文と解答プログラム例と実行結果の表示例を図 3 に示す。

[問題文] 変数 i, x, y を宣言し, for 文を用いて一辺 40 の正方形を横一列に 5 つ描きなさい。

[プログラム例]
size(320, 240); // 表示ウィンドウ
int i; // 変数 i を宣言
int x, y; // 変数 x, y を宣言
y=0; // 変数 y に 0 を代入

for(i=0; i<5; i++)
{
x=i;
rect(40*x, 40*y, 40, 40); // 四角形描く
}

a. 問題文とプログラム例



b. 実行結果の画面

図 3 for文を用いて連続図形を描くプログラムと実行結果の例

分岐文 (if文) を理解するための練習問題として、図 4 bのような市松模様の様な複雑な出力図形を得るプログラムを作成する課題を与えた。縦

横偶数の場合と奇数の場合で位置がずれるように仕組まれた複雑なものであるが学生は直感的に理解したように思われた。

【問題文】 for 文と if 文を用いて一辺 10 の正方形を図のように描きなさい。

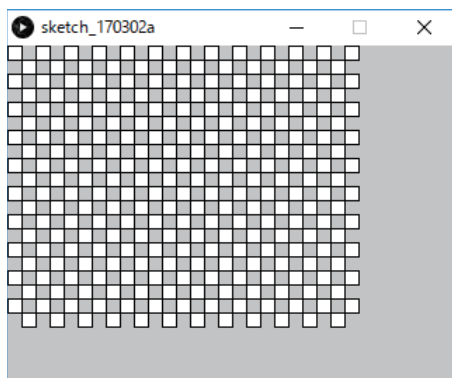
```

【プログラム例】
size( 320, 240 );
int i;
int j;
int x;
int y;

for(j=0; j<20; j++)
{
  for(i=0; i<25; i++)
  {
    if((i%2==0&& j%2==0) || (i%2!=0&& j%2!=0))
    {
      x = i;
      y = j;
      rect( 10*x, 10*y, 10, 10 );
    }
  }
}

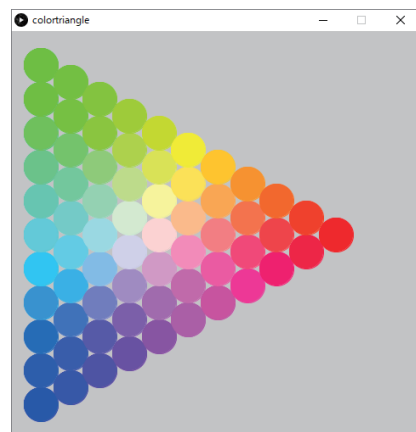
```

a. 問題文とプログラム例



b. 実行結果の画面

図4 if文を用いて連続で図形を描くプログラムと実行結果の例



a. カラートライアングル

```

【プログラム例】
size(480, 480);
colorMode(RGB, 256, 256, 256);
int rgb[] =
{
  {0, 0, 255}, {0, 51, 255}, {0, 102, 255},
  {0, 153, 255}, {0, 204, 255}, {0, 255, 255},
  {0, 255, 204}, {0, 255, 153}, {0, 255, 102},
  {0, 255, 51}, {0, 255, 0},

  {51, 0, 255}, {56, 56, 255}, {61, 115, 255},
  {69, 176, 230}, {74, 230, 255}, {74, 255, 230},
  {69, 255, 176}, {61, 255, 115}, {56, 255, 56},
  {51, 255, 0},

  {102, 0, 255}, {115, 61, 255}, {128, 128, 255},
  {140, 196, 255}, {153, 255, 255}, {140, 255, 196},
  {128, 255, 128}, {115, 255, 61}, {102, 255, 0},

  {153, 0, 255}, {176, 69, 255}, {196, 140, 255},
  {219, 219, 255}, {219, 255, 219}, {196, 255, 140},
  {176, 255, 69}, {153, 255, 0},

  {204, 0, 255}, {230, 74, 255}, {255, 153, 255},
  {255, 219, 219}, {255, 255, 153}, {230, 255, 74},
  {204, 255, 0},

  {255, 0, 255}, {255, 74, 230}, {255, 140, 196},
  {255, 196, 140}, {255, 230, 74}, {255, 255, 0},

  {255, 0, 204}, {255, 69, 176}, {255, 128, 128},
  {255, 176, 69}, {255, 204, 0},

  {255, 0, 153}, {255, 61, 115}, {255, 115, 61},
  {255, 153, 0},

  {255, 0, 102}, {255, 56, 56}, {255, 102, 0},

  {255, 0, 51}, {255, 51, 0},

  {255, 0, 0}
};

int i, j;
int r = 40;
int n = 0;
for(i=0; i<=10; i++)
{
  for(j=0; j<=10-i; j++)
  {
    stroke(rgb[n][0], rgb[n][1], rgb[n][2]);
    fill(rgb[n][0], rgb[n][1], rgb[n][2]);
    ellipse( (int)((i+1)*r/2.0*sqrt(3.0)),
              (int)((11-j)*r-r/2*i), r, r);
  }
  n = n + (11 - i);
}

```

【色指定のメソッド】
 stroke(r, g, b); // 図形の線の色指定
 fill(r, g, b); // 図形の塗りつぶしの色指定
 ただし, $0 \leq r, g, b \leq 255$
 【円を描くメソッド】
 ellipse(中心の横座標, 中心の縦座標,
 楕円の横幅, 楕円の縦幅);

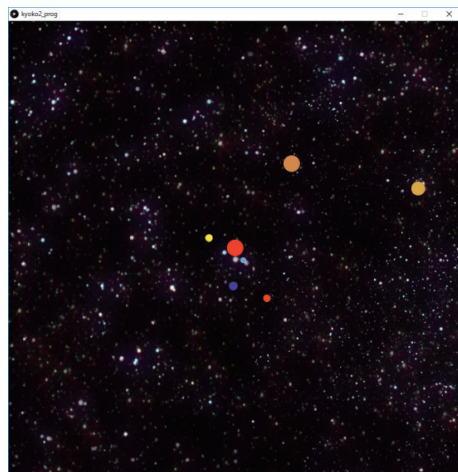
b. プログラム例と使用したメソッド

図5 カラートライアングルとプログラム例

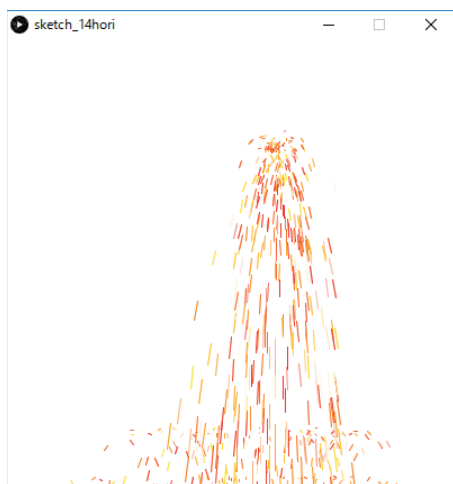
図5のカラートライアングル(文献3) や色立体を作成して色について学ぶ課題も試みた。

3. 作品例

Processingの基礎を身につけた後、応用としてオリジナル作品を作成してもらった。その作品例を図6に示す。



a. 太陽系の動きを描く動画



b. 地面に跳ね返る火花を描く動画



c. 複数のキャラクターの絵が上下左右のウィンドウ枠に跳ね返る動画

図6 学生の作成したオリジナル作品の例

一人ずつ各々の作品を作成した後、発表会を行い各自の作品を紹介し、学生自身と教員が全員の作品の評価を0点～10点の範囲で採点した。採点結果例を表1に示す。左端第1列のA～Mは学生名であり、上端の第1行のA～ゲスト2は、採点者名である。右端列は、各作品の平均値であり、この結果に基づいて上位者を表彰した。

表1 オリジナル作品の採点表

氏名	A	B	C	D	E	F	G	H	I	J	K	L	M	ゲスト1	ゲスト2	Ave.
A	10	10	10	10	7	7	9	7	6	5	7	8	8	5	5	7.6
B	10	10	10	10	7	9	9	9	8	3	8	9	4	4	6	7.7
C	9	9	6	8	6	7	9	6	4	3	6	5	9	2	6	6.3
D	9	9	10	10	6	6	9	7	6	5	8	7	4	6	6	7.2
E	10	10	10	10	8	10	9	7	6	8	9	8	8	5	7	8.3
F	10	10	10	10	9	9	10	9	9	8	10	8	10	8	8	9.2
G	10	10	10	10	7	9	9	8	8	4	6	6	8	4	7	7.7
H	9	9	10	10	6	9	9	10	9	5	6	6	9	8	7	8.1
I	10	9	10	10	10	10	10	10	10	10	10	10	10	10	10	9.9
J	10	9	10	10	7	9	8	9	10	6	6	6	8	8	7	8.2
K	10	10	10	10	6	9	7	10	9	10	10	8	9	10	10	9.2
L	10	10	10	10	10	10	7	10	10	10	6	8	10	9	10	9.3
M	10	10	10	10	9	10	7	10	10	10	9	10	10	10	10	9.7

4. むすび

ここでは、学生の興味を引き出しやすい、視覚に訴える映像を作成するコンピュータグラフィックスに注目し、①基本画像作成の簡単なプログラム群の作成、②それらを用いて静止画像や動画の作品を作成させ、③学生が出来栄を評価し、優劣を競う授業をPBL教育(文献1))の1つとして試みたので、その概要を報告した。

参考文献

- 1) Casey Reas, Ben Fry: Processingをはじめよう、オライリー・ジャパン (2011).
- 2) 箕輪功: PBL教育, 玉川大学工学部紀要, 第47号, pp. 1-2 (2012).
- 3) 菅野直敏: 解説 ファジィ理論の色彩への応用, 日本知能情報ファジィ学会誌, Vol. 26, No. 3, pp. 84-100 (2014).

2017年3月16日原稿受付, 2017年4月13日採録決定

Received, March 16, 2017, accepted, April 13, 2017