

2017年度情報メディア基盤ユニット 6月13日講義内で作ったサンプルプログラム

その1

```
float y; // 円の中心のY座標
float x; // 円の中心のX座標
float v; // 円の縦方向の移動速度
int radius;
void setup() {
  size(300, 400);
  radius = 10;
  v = random(1, 2); // 移動速度を乱数で決める
  x = random(radius, width-radius); // 出現位置をずらす
  y = -random(radius, 2*radius); // 出現タイミングをずらすため
}
void draw() {
  background(255);
  y = y+v;
  if (y -radius > height) {
    x = random(radius, width-radius); // 出現位置をずらす
    y = -random(radius, 2*radius); // 出現タイミングをずらすため
  }
  stroke(255, 10, 10);
  fill(255, 10, 10);
  ellipse(width/2, y, 2*radius, 2*radius);
}
```

その2

```
float[] x = new float[2]; // 円の中心のX座標
float[] y = new float[2]; // 円の中心のY座標
float[] v = new float[2]; // 円の縦方向の移動速度
int radius;
void setup() {
  size(300, 400);
  smooth();
  radius = 10;
  v[0] = random(1, 2); // 移動速度を乱数で決める
  y[0] = -random(radius, 2*radius); // 出現タイミングをずらすため
  x[0] = random(radius, width-radius);
  v[1] = random(1, 2); // 移動速度を乱数で決める
  y[1] = -random(radius, 2*radius); // 出現タイミングをずらすため
  x[1] = random(radius, width-radius);
}
void draw() {
  background(255);
  y[0] = y[0]+v[0];
  if (y[0]-radius > height) {
    x[0] = random(radius, width-radius); // 出現位置をずらす
    y[0] = -random(radius, 2*radius);
    ; // 出現タイミングをずらすため
  }
  stroke(255, 10, 10);
  fill(255, 10, 10);
}
```

```

ellipse(x[0], y[0], 2*radius, 2*radius);
y[1] = y[1]+v[1];
if (y[1]-radius > height) {
  x[1] = random(radius, width-radius); // 出現位置をずらす
  y[1] = -random(radius, 2*radius);
  ; // 出現タイミングをずらすため
}
stroke(255, 10, 10);
fill(255, 10, 10);
ellipse(x[1], y[1], 2*radius, 2*radius);
}

```

その3

```

float[] x; // 円の中心の X 座標
float[] y; // 円の中心の Y 座標
float[] v; // 円の縦方向の移動速度
int radius;
void setup() {
  size(300, 400);
  smooth();
  radius = 10;
  x = new float[2];
  y = new float[2];
  v = new float[2];
  for (int i=0; i<2; i++) {
    v[i] = random(1, 2); // 移動速度を乱数で決める
    y[i] = -random(radius, 2*radius); // 出現タイミングをずらすため
    x[i] = random(radius, width-radius);
  }
}

void draw() {
  background(255);
  for (int i=0; i<2; i++) {
    y[i] = y[i]+v[i];
    if (y[i]-radius > height) {
      x[i] = random(radius, width-radius); // 出現位置をずらす
      y[i] = -random(radius, 2*radius); // 出現タイミングをずらす
    }
    stroke(255, 10, 10);
    fill(255, 10, 10);
    ellipse(x[i], y[i], 2*radius, 2*radius);
  }
}

```

その4

```

float[] x = new float[50]; // 円の中心の X 座標
float[] y = new float[50]; // 円の中心の Y 座標
float[] v = new float[50]; // 円の縦方向の移動速度
color[] cols = new color[50]; // 円の色の情報
int[] radius = new int[50]; // 円の半径の情報

void setup() {
  size(300, 400);

```

```

colorMode(HSB, 359, 99, 99);
for (int i=0; i<x.length; i++) {
  radius[i] = int(random(5, 15));
  v[i] = random(1, 2);
  y[i] = -random(radius[i], 2*radius[i]);
  x[i] = random(radius[i], width-radius[i]);
  cols[i] = color(random(360), random(50, 100), random(50,
100));
}
}
void draw() {
  background(0, 0, 99);
  for (int i=0; i<x.length; i++) {
    y[i] = y[i]+v[i];
    if (y[i] > height) {
      x[i] =random(radius[i], width-radius[i]);
      y[i] = -random(radius[i], 2*radius[i]);
    }
    stroke(cols[i]);
    fill(cols[i]);
    ellipse(x[i], y[i], 2*radius[i], 2*radius[i]);
  }
}

```

その5

```

int pickedID = -1;
boolean picking=false;
float[] xBall = new float[10];
float[] yBall = new float[10];
color[] cBall = new color[10];
int radius = 10;

void setup() {
  size(400, 400);
  colorMode(HSB, 359, 99, 99);
  for (int i=0; i<xBall.length; i++) {
    xBall[i] = random(radius, width-radius);
    yBall[i] = random(radius, height-radius);
    cBall[i] = color(random(360), 99, 99);
  }
}

void mouseDragged() {
  if (picking) {
    xBall[pickedID] += (mouseX-pmouseX);
    yBall[pickedID] += (mouseY-pmouseY);
  }
}

void mouseReleased() {
  picking = false;
  pickedID = -1;
}

```

```

void mousePressed() {
  for (int i=0; i<xBall.length; i++) {
    if (dist(mouseX, mouseY, xBall[i], yBall[i]) <= radius) {
      picking = true;
      pickedID = i;
    }
  }
}

```

```

void draw() {
  background(0, 0, 99);
  for (int i=0; i < xBall.length; i++) {
    stroke(cBall[i]);
    fill(cBall[i]);
    ellipse(xBall[i], yBall[i], 2*radius, 2*radius);
  }
}

```

その6

```

int[] x = new int[100]; // マウスの X 座標の値を保存
int[] y = new int[100]; // マウスの Y 座標の値を保存
color[] cols = new color[100]; // 色の情報を保存
void setup() {
  size(400, 400);
  colorMode(HSB, 359, 99, 99);
}
void draw() {
  background(0, 0, 99);
  stroke(255, 10, 10);
  for (int i=0; i<x.length -1; i++) { // 一つずつ前に移動させる
    x[i] = x[i+1];
    y[i] = y[i+1];
    cols[i] = cols[i+1];
  }
  x[x.length-1] = mouseX; // 最後(99) に現在の情報を代入する
  y[y.length-1] = mouseY;
  cols[cols.length-1] = color(frameCount % 360, 99, 99);
  // 配列に保存されている情報を利用して、折れ線を描画する
  for (int i=1; i<x.length; i++) {
    stroke(cols[i]);
    line(x[i-1], y[i-1], x[i], y[i]);
  }
}

```

その7

```

String [] names = {
  "Riho",
  "Tomoyo",
  "Asuna",
  "Serval Cat",
  "Fennec",
  "Common raccoon",
  "Jaguar",
  "Asuna"};

```

```

PFont font;

void fadeToWhite() {
  stroke(0, 0, 99, 20);
  fill(0, 0, 99, 20);
  rectMode(CORNER);
  rect(0, 0, width, height);
}

void setup() {
  size(400, 400);
  font = loadFont("Serif-48.vlw");
  colorMode(HSB, 359, 99, 99);
  textFont(font, 48);
}

void draw() {
  fadeToWhite();
  // 表示する文字列を選択する
  int idx = int(random(names.length));
  fill(color(random(360), 99, 99));
  text(names[idx], random(width), random(height));
}

```

その 8

```

float y0; // 円の中心の Y 座標
float x0; // 円の中心の X 座標
float v0; // 円の縦方向の移動速度

float y1; // 円の中心の Y 座標
float x1; // 円の中心の X 座標
float v1; // 円の縦方向の移動速度

float y2; // 円の中心の Y 座標
float x2; // 円の中心の X 座標
float v2; // 円の縦方向の移動速度

int radius;
void setup() {
  size(300, 400);
  radius = 10;
  v1 = random(1, 2); // 移動速度を乱数で決める
  x1 = random(radius, width-radius); // 出現位置をずらす
  y1 = -random(radius, 2*radius); // 出現タイミングをずらすため
  v0 = random(1, 2); // 移動速度を乱数で決める
  x0 = random(radius, width-radius); // 出現位置をずらす
  y0 = -random(radius, 2*radius); // 出現タイミングをずらすため
  v2 = random(1, 2); // 移動速度を乱数で決める
  x2 = random(radius, width-radius); // 出現位置をずらす
  y2 = -random(radius, 2*radius); // 出現タイミングをずらすため
}

void draw() {
  background(255);
  y0 = y0+v0;

```

```

if (y0 -radius> height) {
  x0 = random(radius, width-radius); // 出現位置をずらす
  y0 = -random(radius, 2*radius); // 出現タイミングをずらすため
}
stroke(255, 10, 10);
fill(255, 10, 10);
ellipse(x0, y0, 2*radius, 2*radius);
y1 = y1+v1;
if (y1 -radius> height) {
  x1 = random(radius, width-radius); // 出現位置をずらす
  y1 = -random(radius, 2*radius); // 出現タイミングをずらすため
}
stroke(255, 10, 10);
fill(255, 10, 10);
ellipse(x1, y1, 2*radius, 2*radius);
y2 = y2+v2;
if (y2 -radius> height) {
  x2 = random(radius, width-radius); // 出現位置をずらす
  y2 = -random(radius, 2*radius); // 出現タイミングをずらすため
}
stroke(255, 10, 10);
fill(255, 10, 10);
ellipse(x2, y2, 2*radius, 2*radius);
}

```

その9

```

float[] y; // 円の中心のY座標
float[] x; // 円の中心のX座標
float[] v; // 円の縦方向の移動速度

```

```
int radius;
```

```

void setup() {
  size(300, 400);
  radius = 10;
  x = new float[3]; // x[0], x[1], x[2]
  y = new float[3]; // y[0], y[1], y[2]
  v = new float[3]; // v[0], v[1], v[2]
  v[0] = random(1, 2); // 移動速度を乱数で決める
  x[0] = random(radius, width-radius); // 出現位置をずらす
  y[0] = -random(radius, 2*radius); // 出現タイミングをずらすため

  v[1] = random(1, 2); // 移動速度を乱数で決める
  x[1] = random(radius, width-radius); // 出現位置をずらす
  y[1] = -random(radius, 2*radius); // 出現タイミングをずらすため

  v[2] = random(1, 2); // 移動速度を乱数で決める
  x[2] = random(radius, width-radius); // 出現位置をずらす
  y[2] = -random(radius, 2*radius); // 出現タイミングをずらすため
}

void draw() {
  background(255);
  y[0] = y[0]+v[0];
  if (y[0] -radius> height) {

```

```

    x[0] = random(radius, width-radius); // 出現位置をずらす
    y[0] = -random(radius, 2*radius); // 出現タイミングをずらすため
}
stroke(255, 10, 10);
fill(255, 10, 10);
ellipse(x[0], y[0], 2*radius, 2*radius);
y[1] = y[1]+v[1];
if (y[1] -radius> height) {
    x[1] = random(radius, width-radius); // 出現位置をずらす
    y[1] = -random(radius, 2*radius); // 出現タイミングをずらすため
}
stroke(255, 10, 10);
fill(255, 10, 10);
ellipse(x[1], y[1], 2*radius, 2*radius);
y[2] = y[2]+v[2];
if (y[2] -radius> height) {
    x[2] = random(radius, width-radius); // 出現位置をずらす
    y[2] = -random(radius, 2*radius); // 出現タイミングをずらすため
}
stroke(255, 10, 10);
fill(255, 10, 10);
ellipse(x[2], y[2], 2*radius, 2*radius);
}

```

その 10

```

float[] y;// 円の中心の Y 座標
float[] x;// 円の中心の X 座標
float[] v;// 円の縦方向の移動速度

```

```

int radius;

```

```

void setup() {
    size(300, 400);
    radius = 10;
    x = new float[3];// x[0],x[1],x[2]
    y = new float[3];// y[0],y[1],y[2]
    v = new float[3];// v[0],v[1],v[2]

    for (int i=0; i<3; i++) {
        v[i] = random(1, 2); // 移動速度を乱数で決める
        x[i] = random(radius, width-radius); // 出現位置をずらす
        y[i] = -random(radius, 2*radius); // 出現タイミングをずらすため
    }
}

```

```

void draw() {
    background(255);
    y[0] = y[0]+v[0];
    if (y[0] -radius> height) {
        x[0] = random(radius, width-radius); // 出現位置をずらす
        y[0] = -random(radius, 2*radius); // 出現タイミングをずらすため
    }
    stroke(255, 10, 10);
    fill(255, 10, 10);
}

```

```

ellipse(x[0], y[0], 2*radius, 2*radius);
y[1] = y[1]+v[1];
if (y[1] -radius> height) {
  x[1] = random(radius, width-radius); // 出現位置をずらす
  y[1] = -random(radius, 2*radius); // 出現タイミングをずらすため
}
stroke(255, 10, 10);
fill(255, 10, 10);
ellipse(x[1], y[1], 2*radius, 2*radius);
y[2] = y[2]+v[2];
if (y[2] -radius> height) {
  x[2] = random(radius, width-radius); // 出現位置をずらす
  y[2] = -random(radius, 2*radius); // 出現タイミングをずらすため
}
stroke(255, 10, 10);
fill(255, 10, 10);
ellipse(x[2], y[2], 2*radius, 2*radius);
}

```

その 11

```

float[] y;// 円の中心の Y 座標
float[] x;// 円の中心の X 座標
float[] v;// 円の縦方向の移動速度

```

```

int radius;

```

```

void setup() {
  size(300, 400);
  radius = 10;
  x = new float[3];// x[0],x[1],x[2]
  y = new float[3];// y[0],y[1],y[2]
  v = new float[3];// v[0],v[1],v[2]

  for (int i=0; i<3; i++) {
    v[i] = random(1, 2); // 移動速度を乱数で決める
    x[i] = random(radius, width-radius); // 出現位置をずらす
    y[i] = -random(radius, 2*radius); // 出現タイミングをずらすため
  }
}

void draw() {
  background(255);
  for (int i=0; i<3; i++) {
    y[i] = y[i]+v[i];
    if (y[i] -radius> height) {
      x[i] = random(radius, width-radius); // 出現位置をずらす
      y[i] = -random(radius, 2*radius); // 出現タイミングをずらすため
    }
    stroke(255, 10, 10);
    fill(255, 10, 10);
    ellipse(x[i], y[i], 2*radius, 2*radius);
  }
}
}

```


その 12

```
float[] y; // 円の中心の Y 座標
float[] x; // 円の中心の X 座標
float[] v; // 円の縦方向の移動速度
color[] c;
int radius;

void setup() {
  size(300, 400);
  colorMode(HSB, 359, 99, 99);
  radius = 10;
  x = new float[300]; // x[0], x[1], x[2], ...
  y = new float[300]; // y[0], y[1], y[2], ...
  v = new float[300]; // v[0], v[1], v[2], ...
  c = new color[300]; // c[0], c[1], c[2], ...

  for (int i=0; i<x.length; i++) {
    v[i] = random(1, 2); // 移動速度を乱数で決める
    x[i] = random(radius, width-radius); // 出現位置をずらす
    y[i] = -random(radius, 2*radius); // 出現タイミングをずらすため
    c[i] = color(random(360), 99, 99);
  }
}

void draw() {
  background(0, 0, 99);
  for(int i=0; i<x.length; i++){
    y[i] = y[i]+v[i];
    if (y[i] -radius> height) {
      x[i] = random(radius, width-radius); // 出現位置をずらす
      y[i] = -random(radius, 2*radius); // 出現タイミングをずらすため
    }
    stroke(c[i]);
    fill(c[i]);
    ellipse(x[i], y[i], 2*radius, 2*radius);
  }
}
```

その 13

```
PFont font;
String[] name;
/*
{"Akagi",
 "Kaga",
 "Souryu",
 "Hiryu",
 "Hokaku",
 "Zuikaku",
 "Suzuya"};
*/
void setup() {
  size(400, 300);
  font = loadFont("Serif-48.vlw");
```

```

    textFont(font, 48);
    name = new String[7];
    name[0] = "Akagi";
    name[1] = "Kaga";
}

void draw() {
    background(255);
    fill(0);
    int idx = second() % name.length;
    textAlign(CENTER, CENTER);

    text(name[idx], 0, 0, width, height);
}

```

その 14

```

float[] y; // 円の中心の Y 座標
float[] x; // 円の中心の X 座標
float[] v; // 円の縦方向の移動速度
color[] c;
int radius;

void setup() {
    size(300, 400);
    colorMode(HSB, 359, 99, 99);
    radius = 10;
    x = new float[300]; // x[0], x[1], x[2], ...
    y = new float[300]; // y[0], y[1], y[2], ...
    v = new float[300]; // v[0], v[1], v[2], ...
    c = new color[300]; // c[0], c[1], c[2], ...

    for (int i=0; i<x.length; i++) {
        v[i] = random(1, 2); // 移動速度を乱数で決める
        x[i] = random(radius, width-radius); // 出現位置をずらす
        y[i] = -random(radius, 2*radius); // 出現タイミングをずらすため
        c[i] = color(random(360), 99, 99);
    }
}

void draw() {
    background(0, 0, 99);
    for(int i=0; i<x.length; i++){
        y[i] = y[i]+v[i];
        if (y[i] -radius> height) {
            x[i] = random(radius, width-radius); // 出現位置をずらす
            y[i] = -random(radius, 2*radius); // 出現タイミングをずらすため
        }
        stroke(c[i]);
        fill(c[i]);
        ellipse(x[i], y[i], 2*radius, 2*radius);
    }
    drawCircleAtBalance(x,y);
}

```

```

void drawCircleAtBalance(float[] xPos,
                        float[] yPos){
    float gx=0;
    float gy=0;
    for(int i=0;i < xPos.length;i++){
        gx = gx + xPos[i];
    }
    for(int i=0;i < yPos.length;i++){
        gy = gy + yPos[i];
    }
    gx = gx / xPos.length;
    gy = gy / yPos.length;
    fill(0,99,0);
    ellipse(gx,gy,2*radius,2*radius);
}

```

その 15

```

int[] test;

void setup() {
    size(200, 200);
    test = new int[10];
    for (int i=0; i<test.length; i++) {
        test[i] = i;
    }
    down(test); //<>// //<>//
    int foo = 0; //<>// //<>//
    down1(foo);
}

void down(int[] h) {
    for (int i=0; i < h.length; i++) {
        h[i] = h[i] + i % 3;
    }
}

void down1(int x){
    x = x - 1;
}

```

その 16

```

int pickedID = -1;
boolean picking=false;
float[] xBall = new float[10];
float[] yBall = new float[10];
color[] cBall = new color[10];
int radius = 10;

void setup() {
    size(400, 400);
    colorMode(HSB, 359, 99, 99);
    for (int i=0; i<xBall.length; i++) {
        xBall[i] = random(radius, width-radius);
        yBall[i] = random(radius, height-radius);
        cBall[i] = color(random(360), 99, 99);
    }
}

```

```

    }
}
void mouseDragged() {
    if (picking) {
        xBall[pickedID] += (mouseX-pmouseX);
        yBall[pickedID] += (mouseY-pmouseY);
    }
}
void mouseReleased() {
    picking = false;
    pickedID = -1;
}
void mousePressed() {
    for (int i=0; i<xBall.length; i++) {
        if (dist(mouseX, mouseY, xBall[i], yBall[i]) <= radius) {
            picking = true;
            pickedID = i;
            break; // もう探す必要がないので、繰り返し処理を終了する
        }
    }
}

void draw() {
    background(0, 0, 99);
    for (int i=0; i < xBall.length; i++) {
        stroke(cBall[i]);
        fill(cBall[i]);
        ellipse(xBall[i], yBall[i], 2*radius, 2*radius);
    }
}

```

その 17

```

String [] names = {"Riho",
    "Tomoyo",
    "Nene",
    "Manaka",
    "Rinko",
    "Narumi"};
PFont font;

void fadeToWhite() {
    stroke(0, 0, 99, 20);
    fill(0, 0, 99, 20);
    rectMode(CORNER);
    rect(0, 0, width, height);
}

void setup() {
    size(400, 400);
    colorMode(HSB, 359, 99, 99);
    font = loadFont("Serif-48.vlw");
    textFont(font, 48);
}
void draw() {

```

```
fadeToWhite();  
// 表示する文字列を選択する  
int idx = int(random(names.length));  
fill(color(random(360), 99, 99));  
text(names[idx], random(width), random(height));  
}
```