7 月 17 日の授業中に作成したサンプルプログラム

情報メディア学科佐藤尚

```
//その 1
float xPos;
float yPos;
float xVel;
float yVel;
float rot;

void setup() {
  size(400, 400);
  xPos = width/2;
  yPos = height/2;
  float t = random(2*PI);
  xVel = 0.2 * cos(t);
  yVel = 0.2 * sin(t);
  rot = 0;
}

void draw() {
  background(255);
  xPos += xVel;
  yPos += yVel;
  rot += PI/180;
  pushMatrix();
  rectMode(CENTER);
  translate(xPos, yPos);
  rotate(rot);
  fill(255, 10, 10);
  rect(0, 0, 20, 20);
  popMatrix();
}


//その 2
```

```
float[] xPos;
float[] yPos;
float[] xVel;
float[] yVel;
float[] rot;
int numberOfRects = 100;

void setup() {
    size(400, 400);
    xPos = new float[numberOfRects];
    yPos = new float[numberOfRects];
    xVel = new float[numberOfRects];
    yVel = new float[numberOfRects];
    rot = new float[numberOfRects];
    for (int i=0; i<numberOfRects; i++) {
        xPos[i] = width/2;
        yPos[i] = height/2;
        float t = random(2*PI);
        float s = random(0.5,1.5);
        xVel[i] = s * cos(t);
        yVel[i] = s * sin(t);
        rot[i] = random(2*PI);
    }
}

void draw() {
    background(255);
    for (int i=0; i<numberOfRects; i++) {
        xPos[i] += xVel[i];
        yPos[i] += yVel[i];
        rot[i] += PI/180;
        pushMatrix();
        rectMode(CENTER);
        translate(xPos[i], yPos[i]);
        rotate(rot[i]);
        fill(255, 10, 10);
        rect(0, 0, 20, 20);
        popMatrix();
    }
}

//その 3
class Rectangle {
    float xPos;
    float yPos;
    float xVel;
    float yVel;
    float rot;
    color col;

    Rectangle() {
        xPos = width/2;
        yPos = height/2;
        float t = random(2*PI);
        float s = random(0.5, 1.5);
        xVel = s * cos(t);
        yVel = s * sin(t);
        rot = random(2*PI);
        col = color(random(255), random(50, 100), random(50, 100));
    }

    void display() {
        pushMatrix();
        rectMode(CENTER);
        translate(xPos, yPos);
        rotate(rot);
        fill(col);
        rect(0, 0, 20, 20);
        popMatrix();
    }
    void move(){
        xPos += xVel;
        yPos += yVel;
        rot += PI/180;
    }
}

Rectangle myRect;

void setup(){
    size(400,400);
    colorMode(HSB,359,99,99);
    myRect = new Rectangle();
}

void draw(){
    background(0,0,99);
    myRect.move();
    myRect.display();
}

//その 4
class Rectangle {
    float xPos;
    float yPos;
    float xVel;
    float yVel;
    float rot;
    color col;

    Rectangle() {
        xPos = width/2;
        yPos = height/2;
        float t = random(2*PI);
        float s = random(0.5, 1.5);
        xVel = s * cos(t);
    yVel = s * sin(t);
    rot = random(2*PI);
    col = color(random(255), random(50, 100), random(50, 100));
  }

  void display() {
    pushMatrix();
    rectMode(CENTER);
    translate(xPos, yPos);
    rotate(rot);
    fill(col);
    rect(0, 0, 20, 20);
    popMatrix();
  }

  void move() {
    xPos += xVel;
    yPos += yVel;
    rot += PI/180;
  }
}

int numberOfRects = 100;
Rectangle[] myRect;

void setup() {
  size(400, 400);
  colorMode(HSB, 359, 99, 99);
  myRect = new Rectangle[numberOfRects];
  for (int i=0; i<numberOfRects; i++) {
    myRect[i] = new Rectangle();
  }
}

void draw() {
    background(0, 0, 99);
    for (int i=0; i<numberOfRects; i++) {
        myRect[i].move();
        myRect[i].display();
    }
}

//その 5
class Rectangle {
    float xPos;
    float yPos;
    float xVel;
    float yVel;
    float rot;
    color col;

    Rectangle() {
        xPos = width/2;
        yPos = height/2;
        float t = random(2*PI);
        float s = random(0.5, 1.5);
        xVel = s * cos(t);
        yVel = s * sin(t);
        rot = random(2*PI);
        col = color(random(255), random(50, 100), random(50, 100));
    }

    void display() {
        pushMatrix();
        rectMode(CENTER);
        translate(xPos, yPos);
        rotate(rot);
        fill(col);
        rect(0, 0, 20, 20);
        popMatrix();
    }
```

```
  void move() {
    xPos += xVel;
    yPos += yVel;
    rot += PI/180;
  }
}

int numberOfRects = 100;
Rectangle[] myRect;

void setup() {
  size(400, 400);
  colorMode(HSB, 359, 99, 99);
  myRect = new Rectangle[numberOfRects];
  for (int i=0; i<numberOfRects; i++) {
    myRect[i] = new Rectangle();
  }
}

void draw() {
  background(0, 0, 99);
  for (Rectangle aRect:myRect) {
    aRect.move();
    aRect.display();
  }
}

//その 6
class Rectangle {
  float xPos;
  float yPos;
  float xVel;
  float yVel;
  float rot;
  color col;
  Rectangle() {
    xPos = width/2;
    yPos = height/2;
    float t = random(2*PI);
    float s = random(0.5, 1.5);
    xVel = s * cos(t);
    yVel = s * sin(t);
    rot = random(2*PI);
    col = color(random(255), random(50, 100), random(50, 100));
  }

  void display() {
    pushMatrix();
    rectMode(CENTER);
    translate(xPos, yPos);
    rotate(rot);
    fill(col);
    rect(0, 0, 20, 20);
    popMatrix();
  }

  void move() {
    xPos += xVel;
    yPos += yVel;
    rot += PI/180;
  }
}

ArrayList<Rectangle> myRect;

void setup() {
  size(400, 400);
  colorMode(HSB, 359, 99, 99);
  myRect = new ArrayList<Rectangle>();
```

```
}

void draw() {
    background(0, 0, 99);
    for (Rectangle aRect:myRect) {
        aRect.move();
        aRect.display();
    }
}

void mouseClicked(){
      for(int i=0;i<100;i++){
          myRect.add(new Rectangle());
      }
}

//その７
class MovingShape {
    float xPos;
    float yPos;
    float xVel;
    float yVel;
    color col;

    MovingShape() {
        xPos = width/2;
        yPos = height/2;
        float t = random(2*PI);
        float s = random(0.5, 1.5);
        xVel = s * cos(t);
        yVel = s * sin(t);
        col = color(random(255), random(50, 100), random(50, 100));
    }

    void move() {
        xPos += xVel;
        yPos += yVel;
    }

    void display() {
        stroke(col);
        fill(col);
        point(xPos,yPos);
    }
}

class Rectangle extends MovingShape {
    float rot;

    Rectangle() {
        rot = random(2*PI);
    }

    void display() {
        pushMatrix();
        rectMode(CENTER);
        translate(xPos, yPos);
        rotate(rot);
        fill(col);
        rect(0, 0, 20, 20);
        popMatrix();
    }

    void move() {
        super.move();
        rot += PI/180;
    }
}

class Circle extends MovingShape {
    void display() {
        fill(col);
        ellipse(xPos, yPos, 20, 20);
    }
}

int numberOfRects = 100;
MovingShape[] myRect;

void setup() {
    size(400, 400);
    colorMode(HSB, 359, 99, 99);
    myRect = new MovingShape[numberOfRects];
    for (int i=0; i<numberOfRects; i++) {
        if (random(1.0) < 0.5) {
            myRect[i] = new Rectangle();
        } else {
            myRect[i] = new Circle();
        }
    }
}

void draw() {
    background(0, 0, 99);
    for (MovingShape aRect : myRect) {
        aRect.move();
        aRect.display();
    }
}

//その8：じわじは画像が出てくる。詳しくはtint関数を調べて下さい。
PImage img;

void setup(){
    size(400,400);
    img = loadImage("m-riho.jpg");
}
```

```
void draw(){
  background(255);
  tint(255,255,255,millis()/1000.0);
  image(img,0,0);
}

//その 9：壁でボールが跳ねかえる
float xPos;
float yPos;
float xVel;
float yVel;

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

  xPos = width-20/1;
  yPos = 0+20/2;

  xVel = cos(radians(120));
  yVel = sin(radians(120));
}

void bounce(float nx, float ny) {
  float k = 2 * (xVel*nx+yVel*ny)/(nx*nx+ny*ny);
  xVel = xVel - k * nx;
  yVel = yVel - k * ny;
}

void draw() {
  background(255);
  xPos += xVel;
  yPos += yVel;

  // buttom
  if (yPos > height-20/2) {
        bounce(0,-1);
    }
    // left
    if (xPos < 20 /2) {
        bounce(1,0);
    }
    // right
    if(xPos+20/2 > width){
        bounce(-1,0);
    }
    // top
    if(yPos < 20/2){
        bounce(0,1);
    }

    fill(255, 10, 10);
    ellipse(xPos, yPos, 20, 20);
}
```