

# プログラミング通論および演習 最終レポート

学籍番号:2120026, 氏名:平野 晶也

8/20

## 1 構想・計画・設計

### 1.1 構想

私は十代のときにメダルゲームに熱を注いでいたときがあった。この授業を通して processing を学んだ今なら簡単なメダルゲームを開発できるのではないかと考えた。特に熱中した「ピンボールゲーム」を元に簡単なゲームを作っていきたいと思う。

### 1.2 計画

他の科目のテストや、課外活動等が忙しいという現実的な理由を鑑みて、冷静にスケジュールリングすることとする。コーディングに費やせる残り日数を 10 日とする。まず、重視したい面を優先順位の高い順番に箇条書きする。

- ゲームそのもの
- タイトル画面
- 終了画面
- 制限時間の実装
- スコアの実装
- 音楽の実装
- 有名アルゴリズムの活用

## 1.3 設計方法

上記の優先したいものを元に以下の順番で設計していくこととする.

- ゲームの枠組み
  1. タイトル画面の作成.
  2. ゲーム画面の作成.
  3. 終了画面の作成.
  4. それぞれの画面の推移を変数 `screen` を用いて作成.
- ゲーム性の追加
  1. 壁を反射しながら等速度直線運動するボールの作成.
  2. ボールと接触すると消える的の作成.
  3. 消えた的の数によってスコアが上昇するようにする.
  4. 制限時間の追加.
  5. スコアと制限時間をゲーム中に見られるようにする.
  6. 簡単なゲーム性を持たせるために接触するとスピードアップする的の作成.

以上の実装を済ませた. ここで, 課題を完了することも考えたが, 今までの授業内容を網羅したコードでは無かった. クラスの概念を1つも使わずにコードを書いてしまっていた. ここで, 今までの実装にクラスの概念を持ち込んでコードの簡略化を行うこととする.

### 1.3.1 第 10 回 クラスについて

第 10 回ではクラスについて学んだ. ここではピンボールの弾について `PinBall` クラスを作り, コードを修正した. しかし, このままではクラスを用いた意味が薄いため, 第 11 回, 第 12 回で学んだクラスの継承を利用して有用性を高めたいと思う.

```
PinBall [] pinball;  
class PinBall {  
    private color c; // color  
    private int d; // diameter  
    private int x, y;  
    private int vx, vy;
```

```

    PinBall() {}; // constructor
    PinBall(color c, int d, int x, int y, int vx, int vy) {
        setColor(c);
        setBall(x, y, d);
        setSpeed(vx, vy);
    }

    public void setColor(color c)
    {this.c = c;}
    public void setBall(int x,int y,int d)
    {this.x = x; this.y = y; this.d = d;}
    public void setSpeed(int vx, int vy)
    {this.vx = vx; this.vy = vy;}

    public void move(){
x += vx;
y += vy;
if (x < 0 || x + d > width) {
    vx *= -1;
    if (x < 0) {
        x *= -1;
    }else if (x + d > width) {
        x = 2 * (width - d) - x;
    }
}
if (y < 0 || y + d > height) {
    vy *= -1;
    if (y < 0) {
        y *= -1;
    }else if (y + d > height) {
        y = 2 * (height - d) - y;
    }
}

    public void draw(){ // draw ball
        fill(c);
        ellipse(x, y, d, d);
    }

```

```
}
```

### 1.3.2 第 11, 12 回 クラスの継承について

上述した PinBall クラスを継承したいと考える．利用できるところで真っ先に思いついたのは的である．そこで PinBall クラスを継承した Point クラスを生成することにする．

```
Point [] point;  
class Point extends PinBall{  
    Point() {}; // constructor  
    Point(color c, int d, int x, int y, int vx, int vy) {  
        setColor(c);  
        setBall(x, y, d);  
        setSpeed(vx, vy);  
    }  
}
```

## 2 まとめ

期間内にアニメーションにインタラクションを追加した作品を，構想から計画，設計までをスケジューリングして余裕を持って作ることができた．授業で習ったクラス概念は，受講しているときには理解が困難だったものの，継承等まで応用して最終課題に組み込めって良かった．まず冗長で無駄の多いコードを書いた後に，クラスなどの応用的な技術を適応して書き直すという方法を取ったが，一人でのソフトウェア開発では問題無かったもののチームでの開発では他者とのコミュニケーションが必要というのを実感した．