#プログラミング　#条件判断（if 文）　#論理演算子
#文字の表示　#アニメーションの停止　#ゲーム

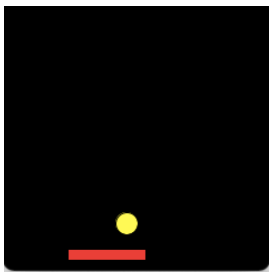
Processing でプログラミングしよう(5)

Processing でプログラミングしよう(4)の内容をアレンジして，簡単なテニスゲームをつくります。

材料（必要なもの）

Processing

1　簡単なテニスゲーム



画面上を跳ね返ってくるボールを，左右に動くバーに当てて，ボールが画面下に落ちないようにします。テニスゲームというよりもブロック崩しのブロックなしという感じでしょうか。

2　赤いバーを描きます

Processing でプログラミングしよう(1)で学習したことを参考にして，画面の下の方のどこかに赤く塗った長方形を描きましょう。

右のプログラムが赤いバーのプログラム例です。y 座標の位置は，画面下の方ということで 220 にしていますが，画面下の方であればどこでも構いません。

```
39  fill(255,  0,  0);
40  rect(0, 220, 70, 10);
```

3　マウスを利用して赤いバーを動かします

描画した赤いバーをマウスで動くようにしましょう。Processing には mouseX という変数が用意されていて，この変数はマウスの x 座標の値を取得します。そこで，この変数を赤いバーを描画する x 座標の位置とします。ただし，バーは横の長さが 70 あるので，マウスの位置がバーの真ん中になるよう調整します。

```
rect(mouseX - 35, 220, 70, 10);
```

0 を mouseX に変えることで赤いバーがマウスの動きに連動します

4　赤いバーへのあたり判定をします

赤いバーの y 座標の位置は，私のプログラム例だと 220 に固定しています。ボール（Procesing(4)では円と呼んでいました)の y 座標が 220 に触れるときで，x 座標の位置がバーのどこかと触れるときであれば，ボールの上下方向の向きを上に変えます。バーの長さが 70 あり，mouseX の値はバーの真ん中にしていたので，mouseX-35 以上で mouseX+35 以下の間にボールの x 座標の位置があれば，ボールの向きを上向きにするということになります。

ボールの y 座標の位置が 220 に触れるときが処理を行う条件の 1 つ目です

```
41  if(y > 209){
42    if(x >= mouseX - 35 && x <= mouseX + 35){
43      muki_t = 1;
44    }
45  }
```

さらに，ボールの x 座標の位置が mouseX-35 以上で mouseX+35 以下の両方を満たすとき（論理演算子の論理積を利用）が 2 つ目の条件です

5　ゲームオーバーの条件を加える

テニスゲームですから，ボールを赤いバーで打ち返せなかったときにゲームオーバーになります。しかし，このままだとボールが画面下まで行くと跳ね返ってしまいます。そこで，ボールが画面下に行ったらアニメーションをストップさせます。Processing では，draw 関数に記述した処理がずっと繰り返されていますが，この処理をストップさせる命令が noLoop( )という命令になります。

```
47    if(y > 229){
48      noLoop();
49    }
```

ボールが画面下まで行ったときに noLoop( )が実行されます

6 「GAME OVER」と文字を表示させます

ゲームがストップするだけではバグったのかなと思われるので，「GAME OVER」と表示させます。文字を表示する命令は text( )です。文字の色は fill( )で指定できます。文字の色が黒だと背景と重なって見えないので注意してください。文字の大きさは textSize( )で指定します。

```
47    if(y > 229){
48      noLoop();
49      fill(255);
50      textSize(14);
51      text("GAME OVER", 80, 120);
52    }
```

fill のカッコ内の値が１つのときはグレースケールになります

text のカッコ内は，表示する文字，表示する x，y 座標の位置です

7　完成したプログラム

```
1   void setup(){
2     size(240,240);
3     frameRate(24);
4   }
5
6   int muki_y = 0;
7   int muki_t = 1;
8   int x = 120;
9   int y = 200;
10
11  void draw(){
12    if(x > 229){
13      muki_y = 1;
14    }
15    if(x < 11){
16      muki_y = 0;
17    }
18    if(y > 229){
19      muki_t = 1;
20    }
21    if(y < 11){
22      muki_t = 0;
23    }
24    background(0);
25    if(muki_y == 0){
26      x += 3;
27    }else{
28      x -= 3;
29    }
30    if(muki_t == 0){
31      y += 3;
32    }else{
33      y -= 3;
34    }
35    fill(255,255,0);
36    ellipse(x, y, 20, 20);
37
38    fill(255, 0, 0);
39    rect(mouseX - 35,220,70,10);
40
41    if(y > 209){
42      if(x >= mouseX - 35 && x <= mouseX + 35){
43        muki_t = 1;
44      }
45    }
46
47    if(y > 229){
48      noLoop();
49      fill(255);
50      textSize(14);
51      text("GAME OVER", 80, 120);
52    }
53  }
```

コツ（留意点）

赤いバーにあたる，ゲームオーバーになる条件を，比較演算子や論理演算子を用いて表現することがポイントです。

作成者

北海道札幌北高等学校　前田健太朗

k_maeda@hokkaido-c.ed.jp