

# 復興知2022 Processing-解答例

## 練習問題

1. 日本の国旗を描いてください

```
1 size(900, 600)
2 background(255)
3 noStroke()
4 fill(255, 0, 0)
5 ellipse(width / 2, height / 2, 350, 350)
```

2. イタリアの国旗を描いてください

```
1 size(900, 600)
2 background(255)
3 noStroke()
4 fill(0, 255, 0)
5 rect(0, 0, width / 3, height)
6 fill(255, 0, 0)
7 rect(width * 2 / 3, 0, width / 3, height)
```

3. スイスの国旗を描いてください

```
1 size(600, 600)
2 background(255, 0, 0)
3 noStroke()
4 fill(255, 255, 255)
5 rect(width // 5, 2 * height // 5, 3 * width // 5, height // 5)
6 rect(2 * width // 5, height // 5, width // 5, 3 * height // 5)
```

## アニメーションプログラム

- Step 0 | ウィンドウ設定

```
1 def setup():
2   size(600, 600) # ウィンドウのサイズ
3
4 def draw():
5   background(255) # 背景色を決める
```

- Step 1 | バーをマウスの位置で変更

```
1 def setup():
2     size(600, 600) # ウィンドウのサイズ
3
4 def draw():
5     background(255) # 背景色を決める
6     barWidth = 150 # バーの長さ
7     barHeight = 20 # バーの高さ
8     fill(0, 100, 255) # バーの色
9     rect(mouseX - barWidth / 2, 550, barWidth, barHeight) # バーを描画する
```

- Step 2 | ボールの動作

```
1 ballX = 50 # ボールのx座標
2 ballY = 50 # ボールのy座標
3 ballVx = 9 # ボールのx方向に対するの更新量
4 ballVy = 2 # ボールのy方向に対するの更新量
5
6 def setup():
7     size(600, 600) # ウィンドウのサイズ
8
9 def draw():
10    global ballX, ballY, ballVx, ballVy # グローバル変数を変更する
11    background(255) # 背景色を決める
12    barWidth = 150 # バーの長さ
13    barHeight = 20 # バーの高さ
14    fill(0, 100, 255) # バーの色
15    rect(mouseX - barWidth / 2, 550, barWidth, barHeight) # バーを描画する
16    fill(200, 0, 255) # ボールの色
17    ellipse(ballX, ballY, 30, 30) # ボールを描画する
18    ballX += ballVx # ボールのx座標を更新する
19    ballY += ballVy # ボールのy座標を更新する
```

- Step 3 | ボールの反射

```
1 ballX = 50 # ボールのx座標
2 ballY = 50 # ボールのy座標
3 ballVx = 9 # ボールのx方向に対するの更新量
4 ballVy = 2 # ボールのy方向に対するの更新量
5
6 def setup():
7     size(600, 600) # ウィンドウのサイズ
8
9 def draw():
```

```

10 global ballX, ballY, ballVx, ballVy # グローバル変数を変更する
11 background(255) # 背景色を決める
12 barWidth = 150 # バーの長さ
13 barHeight = 20 # バーの高さ
14 fill(0, 100, 255) # バーの色
15 rect(mouseX - barWidth / 2, 550, barWidth, barHeight) # バーを描画する
16 fill(200, 0, 255) # ボールの色
17 ellipse(ballX, ballY, 30, 30) # ボールを描画する
18 ballX += ballVx # ボールのx座標を更新する
19 ballY += ballVy # ボールのy座標を更新する
20
21 if ballY < 0: # 上端衝突
22     ballVy *= -1
23 if ballX < 0 or ballX > width: # 左右端衝突
24     ballVx *= -1

```

- Step 4 | ボールとバーの衝突判定

```

1 ballX = 50 # ボールのx座標
2 ballY = 50 # ボールのy座標
3 ballVx = 9 # ボールのx方向に対するの更新量
4 ballVy = 2 # ボールのy方向に対するの更新量
5
6 def setup():
7     size(600, 600) # ウィンドウのサイズ
8
9 def draw():
10    global ballX, ballY, ballVx, ballVy # グローバル変数を変更する
11    background(255) # 背景色を決める
12    barWidth = 150 # バーの長さ
13    barHeight = 20 # バーの高さ
14    fill(0, 100, 255) # バーの色
15    rect(mouseX - barWidth / 2, 550, barWidth, barHeight) # バーを描画する
16    fill(200, 0, 255) # ボールの色
17    ellipse(ballX, ballY, 30, 30) # ボールを描画する
18    ballX += ballVx # ボールのx座標を更新する
19    ballY += ballVy # ボールのy座標を更新する
20
21    if ballY < 0: # 上端衝突
22        ballVy *= -1
23    if ballX < 0 or ballX > width: # 左右端衝突
24        ballVx *= -1
25
26    leftBar = mouseX - barWidth / 2 # バーの左端座標
27    rightBar = mouseX + barWidth / 2 # バーの右端座標
28
29    # ボールとバーの衝突
30    if ballVy > 0 and ballX > leftBar and ballX < rightBar and ballY > 550 and

```

```
ballY < 550 + barHeight:
31     ballVy *= -1
```

- Step 5 | ブロックの設置

```
1 ballX = 50 # ボールのx座標
2 ballY = 50 # ボールのy座標
3 ballVx = 9 # ボールのx方向に対するの更新量
4 ballVy = 2 # ボールのy方向に対するの更新量
5
6 blockLife = 1 # ブロックのライフ
7
8 def setup():
9     size(600, 600) # ウィンドウのサイズ
10
11 def draw():
12     global ballX, ballY, ballVx, ballVy, blockLife # グローバル変数を変更す
    る
13     background(255) # 背景色を決める
14     barWidth = 150 # バーの長さ
15     barHeight = 20 # バーの高さ
16     fill(0, 100, 255) # バーの色
17     rect(mouseX - barWidth / 2, 550, barWidth, barHeight) # バーを描画する
18
19     blockWidth = 70 # ブロックの幅
20     blockHeight = 20 # ブロックの高さ
21     if blockLife > 0:
22         fill(200, 200, 255) # ブロックの
    色
23         rect((width - blockWidth) / 2, 30, blockWidth, blockHeight) # ブロックを
    描画する
24
25         fill(200, 0, 255) # ボールの色
26         ellipse(ballX, ballY, 30, 30) # ボールを描画する
27         ballX += ballVx # ボールのx座標を更新する
28         ballY += ballVy # ボールのy座標を更新する
29
30         if ballY < 0: # 上端衝突
31             ballVy *= -1
32         if ballX < 0 or ballX > width: # 左右端衝突
33             ballVx *= -1
34
35         leftBar = mouseX - barWidth / 2 # バーの左端座標
36         rightBar = mouseX + barWidth / 2 # バーの右端座標
37
38         # ボールとバーの衝突
39         if ballVy > 0 and ballX > leftBar and ballX < rightBar and ballY > 550 and
    ballY < 550 + barHeight:
```

- Step 6 | ボールとブロックの衝突判定

```

1 ballX = 50 # ボールのx座標
2 ballY = 50 # ボールのy座標
3 ballVx = 9 # ボールのx方向に対するの更新量
4 ballVy = 2 # ボールのy方向に対するの更新量
5
6 blockLife = 1 # ブロックのライフ
7
8 def setup():
9     size(600, 600) # ウィンドウのサイズ
10
11 def draw():
12     global ballX, ballY, ballVx, ballVy, blockLife # グローバル変数を変更す
    13     background(255) # 背景色を決める
    14     barWidth = 150 # バーの長さ
    15     barHeight = 20 # バーの高さ
    16     fill(0, 100, 255) # バーの色
    17     rect(mouseX - barWidth / 2, 550, barWidth, barHeight) # バーを描画する
    18
    19     blockWidth = 70 # ブロックの幅
    20     blockHeight = 20 # ブロックの高さ
    21     if blockLife > 0:
    22         fill(200, 200, 255) # ブロックの
    23         rect((width - blockWidth) / 2, 30, blockWidth, blockHeight) # ブロックを
    24         色
    25         描画する
    26         fill(200, 0, 255) # ボールの色
    27         ellipse(ballX, ballY, 30, 30) # ボールを描画する
    28         ballX += ballVx # ボールのx座標を更新する
    29         ballY += ballVy # ボールのy座標を更新する
    30
    31     if ballY < 0: # 上端衝突
    32         ballVy *= -1
    33     if ballX < 0 or ballX > width: # 左右端衝突
    34         ballVx *= -1
    35
    36     leftBar = mouseX - barWidth / 2 # バーの左端座標
    37     rightBar = mouseX + barWidth / 2 # バーの右端座標
    38
    39     # ボールとバーの衝突
    40     if ballVy > 0 and ballX > leftBar and ballX < rightBar and ballY > 550 and
        ballY < 550 + barHeight:
        ballVy *= -1

```

```

41
42     leftBlock = (width - blockWidth) / 2
43     rightBlock = (width + blockWidth) / 2
44
45     # ボールとブロックの衝突
46     if ballX > leftBlock and ballX < rightBlock and bally > 30 and bally < 30
+ blockHeight:
47         ballVy *= -1
48         blockLife -= 1

```

- Step 7 | 文字列によるゲームインタラクション

```

1 ballX = 50 # ボールのx座標
2 bally = 50 # ボールのy座標
3 ballVx = 9 # ボールのx方向に対するの更新量
4 ballVy = 2 # ボールのy方向に対するの更新量
5
6 blockLife = 1 # ブロックのライフ
7
8 def setup():
9     size(600, 600) # ウィンドウのサイズ
10
11 def draw():
12     global ballX, bally, ballVx, ballVy, blockLife # グローバル変数を変更す
    る
13     background(255) # 背景色を決める
14     barWidth = 150 # バーの長さ
15     barHeight = 20 # バーの高さ
16     fill(0, 100, 255) # バーの色
17     rect(mouseX - barWidth / 2, 550, barWidth, barHeight) # バーを描画する
18
19     blockWidth = 70 # ブロックの幅
20     blockHeight = 20 # ブロックの高さ
21     if blockLife > 0:
22         fill(200, 200, 255) # ブロックの
    色
23         rect((width - blockWidth) / 2, 30, blockWidth, blockHeight) # ブロックを
    描画する
24     elif blockLife == 0: # GameClear処理
25         fill(0)
26         textSize(30)
27         text("Game Clear !", 200, 200)
28
29     fill(200, 0, 255) # ボールの色
30     ellipse(ballX, bally, 30, 30) # ボールを描画する
31     ballX += ballVx # ボールのx座標を更新する
32     bally += ballVy # ボールのy座標を更新する
33

```

```

34 # GameOver処理
35 if ballY > height:
36     fill(0)
37     textSize(30)
38     text("Game Over...", 200, 400)
39
40     if ballY < 0: # 上端衝突
41         ballVy *= -1
42     if ballX < 0 or ballX > width: # 左右端衝突
43         ballVx *= -1
44
45     leftBar = mouseX - barWidth / 2 # バーの左端座標
46     rightBar = mouseX + barWidth / 2 # バーの右端座標
47
48     # ボールとバーの衝突
49     if ballVy > 0 and ballX > leftBar and ballX < rightBar and ballY > 550 and
ballY < 550 + barHeight:
50         ballVy *= -1
51
52     leftBlock = (width - blockWidth) / 2
53     rightBlock = (width + blockWidth) / 2
54
55     # ボールとブロックの衝突
56     if ballX > leftBlock and ballX < rightBlock and ballY > 30 and ballY < 30
+ blockHeight:
57         ballVy *= -1
58         blockLife -= 1

```