

令和5年度総合型選抜プログラミング実技試験
事前公開プログラム

日本大学文理学部情報科学科

このプログラムは、ボールが自由落下する様子をシミュレーションするものである。プログラム冒頭で定義された定数はそれぞれ下記のとおりである。

- GRAVITY: 重力加速度
- COEFFICIENT_OF_RESTITUTION: 反発係数
- FRICTION: 動摩擦係数

Java Mode

```
1 float GRAVITY = 9.8;
2 float COEFFICIENT_OF_RESTITUTION = 0.8;
3 float FRICTION = 0.9;
4
5 int ballSize = 30;
6 float ballX;
7 float ballY;
8 float ballVX;
9 float ballVY;
10 color ballCol;
11
12 void setup() {
13     size(600, 600);
14     ballX = random(0, width);
15     ballY = random(0, height);
16     ballVX = random(-5, 6);
17     ballVY = random(-5, 6);
18     ballCol = color(random(256), random(256), random(256));
19 }
20
21 void draw() {
22     background(128);
23     ballVY += GRAVITY / 60;
24     ballX += ballVX;
25     ballY += ballVY;
26
27     if (ballX > width){
28         ballX = width;
29         ballVX = -ballVX * COEFFICIENT_OF_RESTITUTION;
30     }
31     if (ballX < 0){
32         ballX = 0;
33         ballVX = -ballVX * COEFFICIENT_OF_RESTITUTION;
34     }
35     if (ballY > height){
36         ballY = height;
37         ballVY = -ballVY * COEFFICIENT_OF_RESTITUTION;
38         ballVX *= FRICTION;
39     }
40     fill(ballCol);
41     ellipse(ballX, ballY, ballSize, ballSize);
42 }
```

Python Mode

```
1 GRAVITY = 9.8
2 COEFFICIENT_OF_RESTITUTION = 0.8
3 FRICTION = 0.9
4
5 ballSize = 30
6
7
8 def setup():
9     global ballX, ballY, ballVX, ballVY, ballCol
10    size(600, 600)
11
```

```

12   ballX = random(0, width)
13   ballY = random(0, height)
14   ballVX = random(-5, 6)
15   ballVY = random(-5, 6)
16   ballCol = color(random(256), random(256), random(256))
17
18
19 def draw():
20   global ballX, ballY, ballVX, ballVY
21   background(128)
22   ballVY += GRAVITY / 60
23   ballX += ballVX
24   ballY += ballVY
25
26   if ballX > width:
27       ballX = width
28       ballVX = -ballVX * COEFFICIENT_OF_RESTITUTION
29   if ballX < 0:
30       ballX = 0
31       ballVX = -ballVX * COEFFICIENT_OF_RESTITUTION
32   if ballY > height:
33       ballY = height
34       ballVY = -ballVY * COEFFICIENT_OF_RESTITUTION
35       ballVX *= FRICTION
36   fill(ballCol)
37   ellipse(ballX, ballY, ballSize, ballSize)

```

p5.js Mode

```

1  const GRAVITY = 9.8;
2  const COEFFICIENT_OF_RESTITUTION = 0.8;
3  const FRICTION = 0.9;
4
5  const ballSize = 30;
6  let ballX;
7  let ballY;
8  let ballVX;
9  let ballVY;
10 let ballCol;
11
12 function setup() {
13   createCanvas(600, 600);
14   ballX = random(0, width);
15   ballY = random(0, height);
16   ballVX = random(-5, 6);
17   ballVY = random(-5, 6);
18   ballCol = color(random(256), random(256), random(256));
19 }
20
21 function draw() {
22   background(128);
23   ballVY += GRAVITY / 60;
24   ballX += ballVX;
25   ballY += ballVY;
26
27   if (ballX > width) {
28       ballX = width;
29       ballVX = -ballVX * COEFFICIENT_OF_RESTITUTION;
30   }
31   if (ballX < 0) {
32       ballX = 0;
33       ballVX = -ballVX * COEFFICIENT_OF_RESTITUTION;
34   }
35   if (ballY > height) {
36       ballY = height;
37       ballVY = -ballVY * COEFFICIENT_OF_RESTITUTION;

```

```
38     ballVX *= FRICTION;
39   }
40   fill(ballCol);
41   ellipse(ballX, ballY, ballSize, ballSize);
42 }
```