

令和7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、鳥の群れの動きをシミュレーションしてアニメーション表示するものであり、詳細は次のとおりである。

- それぞれの鳥は各時点において、自分の視界内の鳥と同じ方向を向くように方向を変える。
- 鳥 A から見て鳥 B が視界内にあるとは、鳥 A と鳥 B の距離が `sightMax` 以内であることを言う。
- シミュレーションする空間は、画面の左端と右端、上端と下端がそれぞれつながっている。例えば、ある鳥が左から画面の右端に到達すると画面の左端に現れる。なお、距離の計算はこの回り込みを考慮して行われている。

#### Java Mode

```
1  int n = 30;
2  float [] x = new float [n];
3  float [] y = new float [n];
4  float [] vx = new float [n];
5  float [] vy = new float [n];
6  float vMax = 1;
7  float sightMax = 20;
8  float r = 5;
9
10 void setup() {
11     size(400, 400);
12     for (int i = 0; i < n; ++i) {
13         x[i] = random(width);
14         y[i] = random(height);
15         vx[i] = vMax * cos(random(TWO_PI));
16         vy[i] = vMax * sin(random(TWO_PI));
17     }
18 }
19
20 void draw() {
21     background(255);
22     drawBirds();
23     update();
24 }
25
26 void drawBirds() {
27     fill(0);
28     for (int i = 0; i < n; ++i) {
29         pushMatrix();
30         translate(x[i], y[i]);
31         rotate(atan2(vy[i], vx[i]));
32         beginShape();
33         vertex(r * cos(0), r * sin(0));
34         vertex(r * cos(radians(150)), r * sin(radians(150)));
35         vertex(r * cos(radians(210)), r * sin(radians(210)));
36         endShape(CLOSE);
37         popMatrix();
38     }
39 }
40
41 boolean inSight(int i, int j) {
42     for (int a = -1; a <= 1; ++a) {
43         for (int b = -1; b <= 1; ++b) {
44             if (dist(x[i], y[i], width * a + x[j], height * b + y[j]) < sightMax) {
45                 return true;
46             }
47         }
48     }
49     return false;
50 }
51
52 void update() {
53     for (int i = 0; i < n; ++i) {
```

```

54     float vxSum = 0;
55     float vySum = 0;
56     int count = 0;
57     for (int j = 0; j < n; ++j) {
58         if (i != j && inSight(i, j)) {
59             vxSum += vx[j];
60             vySum += vy[j];
61             count += 1;
62         }
63     }
64     if (count > 0) {
65         vx[i] += vxSum / count;
66         vy[i] += vySum / count;
67         float d = dist(0, 0, vx[i], vy[i]);
68         vx[i] = vx[i] / d * vMax;
69         vy[i] = vy[i] / d * vMax;
70     }
71 }
72 for (int i = 0; i < n; ++i) {
73     x[i] = (x[i] + vx[i] + width) % width;
74     y[i] = (y[i] + vy[i] + height) % height;
75 }
76 }

```

### Python Mode

```

1  n = 50
2  x = [0] * n
3  y = [0] * n
4  vx = [0] * n
5  vy = [0] * n
6  vMax = 1
7  sightMax = 20
8  r = 5
9
10
11 def setup():
12     size(400, 400)
13     for i in range(n):
14         x[i] = random(width)
15         y[i] = random(height)
16         vx[i] = vMax * cos(random(TWO.PI))
17         vy[i] = vMax * sin(random(TWO.PI))
18
19
20 def draw():
21     background(255)
22     drawBirds()
23     update()
24
25
26 def drawBirds():
27     fill(0)
28     for i in range(n):
29         pushMatrix()
30         translate(x[i], y[i])
31         rotate(atan2(vy[i], vx[i]))
32         beginShape()
33         vertex(r * cos(0), r * sin(0))
34         vertex(r * cos(radians(150)), r * sin(radians(150)))
35         vertex(r * cos(radians(210)), r * sin(radians(210)))
36         endShape(CLOSE)
37         popMatrix()
38
39
40 def inSight(i, j):

```

```

41     for a in [-1, 0, 1]:
42         for b in [-1, 0, 1]:
43             if dist(x[i], y[i], width * a + x[j], height * b + y[j]) < sightMax:
44                 return True
45     return False
46
47
48 def update():
49     for i in range(n):
50         vxSum = 0
51         vySum = 0
52         count = 0
53         for j in range(n):
54             if i != j and inSight(i, j):
55                 vxSum += vx[j]
56                 vySum += vy[j]
57                 count += 1
58         if count > 0:
59             vx[i] += vxSum / count
60             vy[i] += vySum / count
61             d = dist(0, 0, vx[i], vy[i])
62             vx[i] = vx[i] / d * vMax
63             vy[i] = vy[i] / d * vMax
64     for i in range(n):
65         x[i] = (x[i] + vx[i] + width) % width
66         y[i] = (y[i] + vy[i] + height) % height

```

p5.js Mode

```

1  const n = 50;
2  const x = new Array(n);
3  const y = new Array(n);
4  const vx = new Array(n);
5  const vy = new Array(n);
6  const vMax = 1;
7  const sightMax = 20;
8  const r = 5;
9
10 function setup() {
11     createCanvas(400, 400);
12     for (let i = 0; i < n; ++i) {
13         x[i] = random(width);
14         y[i] = random(height);
15         vx[i] = vMax * cos(random(TWO_PI));
16         vy[i] = vMax * sin(random(TWO_PI));
17     }
18 }
19
20 function draw() {
21     background(255);
22     drawBirds();
23     update();
24 }
25
26 function drawBirds() {
27     fill(0);
28     for (let i = 0; i < n; ++i) {
29         push();
30         translate(x[i], y[i]);
31         rotate(atan2(vy[i], vx[i]));
32         beginShape();
33         vertex(r * cos(0), r * sin(0));
34         vertex(r * cos(radians(150)), r * sin(radians(150)));
35         vertex(r * cos(radians(210)), r * sin(radians(210)));
36         endShape(CLOSE);
37     }

```

```

38 }
39 }
40
41 function inSight(i, j) {
42   for (let a = -1; a <= 1; ++a) {
43     for (let b = -1; b <= 1; ++b) {
44       if (dist(x[i], y[i], width * a + x[j], height * b + y[j]) < sightMax) {
45         return true;
46       }
47     }
48   }
49   return false;
50 }
51
52 function update() {
53   for (let i = 0; i < n; ++i) {
54     let vxSum = 0;
55     let vySum = 0;
56     let count = 0;
57     for (let j = 0; j < n; ++j) {
58       if (i !== j && inSight(i, j)) {
59         vxSum += vx[j];
60         vySum += vy[j];
61         count += 1;
62       }
63     }
64     if (count > 0) {
65       vx[i] += vxSum / count;
66       vy[i] += vySum / count;
67       const d = dist(0, 0, vx[i], vy[i]);
68       vx[i] = (vx[i] / d) * vMax;
69       vy[i] = (vy[i] / d) * vMax;
70     }
71   }
72   for (let i = 0; i < n; ++i) {
73     x[i] = (x[i] + vx[i] + width) % width;
74     y[i] = (y[i] + vy[i] + height) % height;
75   }
76 }

```