

演習 2 : 集合の知性を設計する

(05) 05/19 (06) 05/26

A | Unity環境の整備・簡単なルール設計

(07) 06/02 (08) 06/09

B | ボイドルール 1・2・3 の実装

(09) 06/16

C | 課題 1 : 集合知の解析

(10) 06/23

D 1 | SIR (感染モデル)

(11) 06/30 (12) 07/07 (13) 07/14

D 2 | 課題 2 : マイルール・感染ルール・視点操作

(14-15) 07/21

D 3 | 発表 (One-Minute Movie)

例題 1

いずれかのボイドからの距離が10以下となると、停止するルール9を加えてください。



Vector3

```
float *Distance(Vector3 p1, Vector3 p2)
```

p1とp2の距離を返すVector3のクラスメソッド

```
<Vector3> *zero
```

ゼロベクトル (0f, 0f, 0f)

例題 2

いずれかのボイドからの距離が30以下となると, 速さの大きい方の速度に合わせるルール8を追加してください。



Vector3

```
float *Distance(Vector3 p1, Vector3 p2)
```

p1とp2の距離を返すVector3のクラスメソッド

```
<float> *magnitude
```

ベクトルの大きさ・長さ ($|(\mathbf{x}, \mathbf{y}, \mathbf{z})|$)

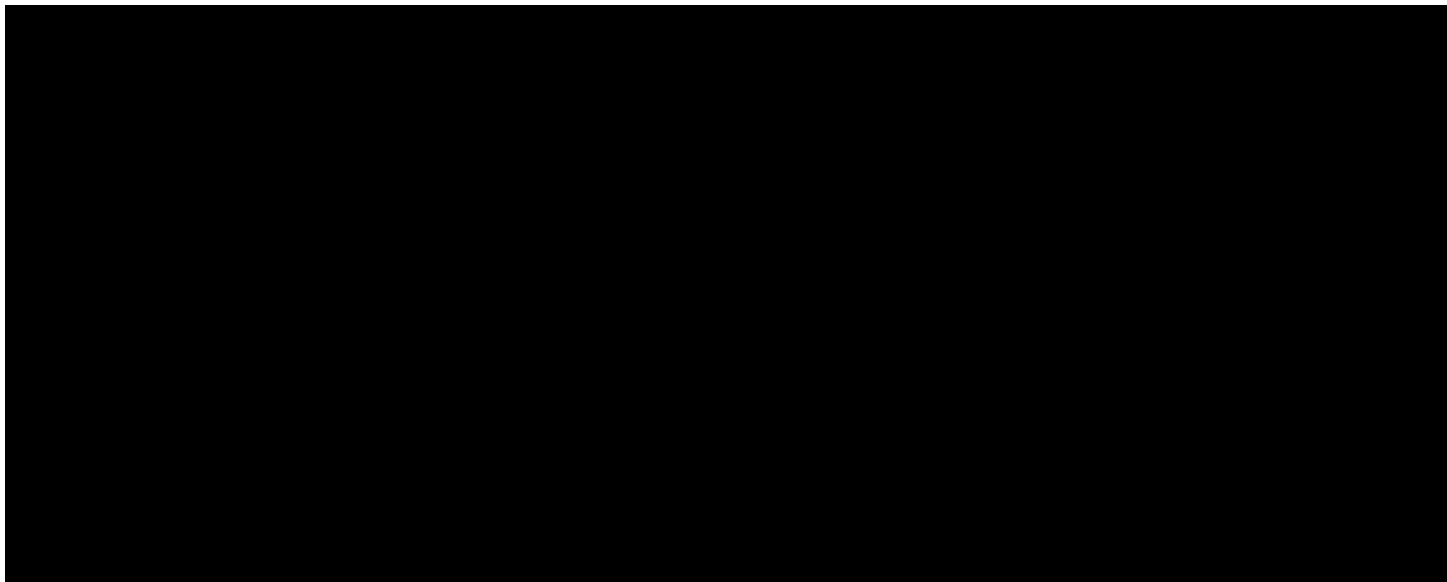
例題 2 (ヒント)

```
void ApplyRule8(){  
    for (int i = 0; i < pop ; i++) {  
        Vector3 ipos = boid [i].pos;  
        Vector3 ivel = boid [i].vel;
```

i 番目のボイドの位置・速度

```
        for (int j = i + 1; j < pop ; j++) {  
            Vector3 jpos = boid [j].pos;  
            Vector3 jvel = boid [j].vel;
```

j 番目のボイドの位置・速度



```
    }  
}  
}
```

BoidRule.cs