

演習 2 : 集合の知性を設計する

(05) 05/20 (06) 05/27

A | Unity環境の整備・簡単なルール設計

(07) 06/10 (08) 06/17

B | ボイドルール 1・2・3 の実装

(09) 06/24 (10) 07/01

C 1 | 集合知の解析

(12) 07/08 (13) 07/15

C 2 | マイルール (ルール 4) の実装・視点の操作

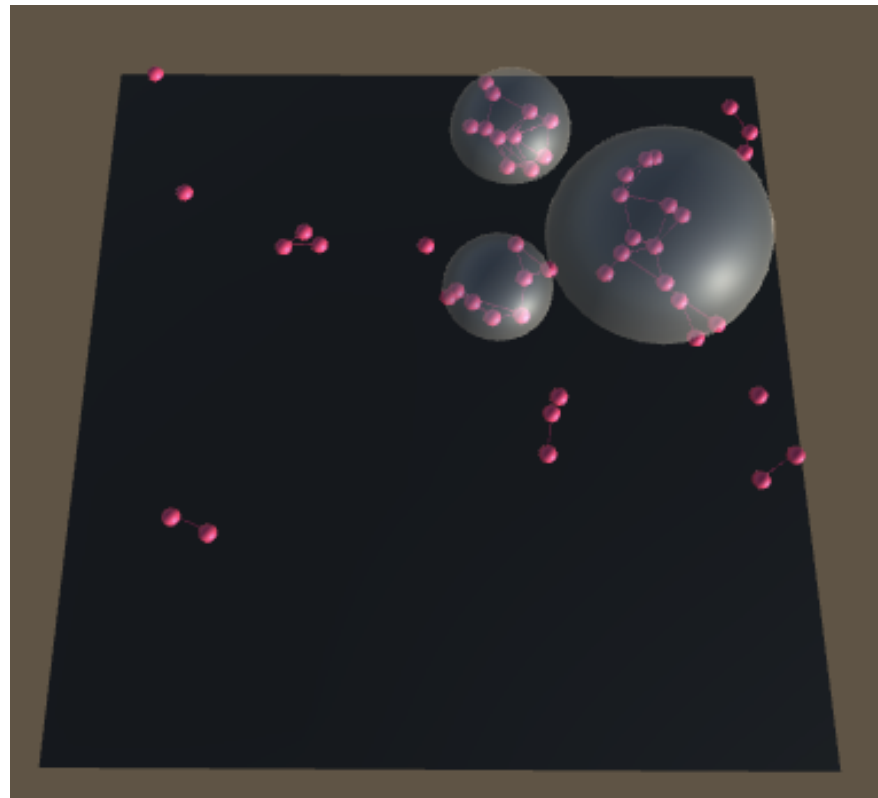
(14-15) 07/22

C 2 | 発表 (One-Minute Movie)

演習 2 - C1

集合知の解析

ボイドのパラメータを任意に変更し、
 集団の振舞の変化を観察してください。



宣言部

```
//解析オブジェクト  
BoidClusterAnalysis ana;
```

```
/* ボイド解析用の変数*/  
private float countmax = 3000f; //計測フレーム  
  
private float cls_sum = 0f; //クラスターの総和  
public float cls_count = 0f; //現在のフレーム数  
public float cls_mean = 0f; //クラスターの平均値
```

```
//新しい知り合いできたフレーム数  
// (10フレーム以上同一のクラスターにいるペアが新たに生まれた場合)  
public int make_friends_frame = 0;
```

```
/* 解析オブジェクトの生成 */  
ana = this.GetComponent<BoidClusterAnalysis> ();
```

Start()

解析の本体は、BoidClusterAnalysis.cs
に記述されています。

```
if (Input.GetKeyDown (KeyCode.I)) {  
    InitBoidPosition ();  
    InitBoidVelocity();
```

```
    cls_sum = 0f;  
    cls_count = 0f;  
    make_friends_frame = 0;  
    ana.InitFriendsHistory(this);
```

```
}
```

Update()

I ボタンを押すと、計算用
変数 (総和とカウンタ)
を0に初期化します。

```

/* 解析モードがON (Key A) のとき */
if (mode_analysis)
{
    ana.SetBoid(this);
    //個体数 (1/10) をクラスター成立の要件とする
    ana.SetMinimumPop((int)Mathf.Floor(bsum / 10f));
    //クラスタを計算
    ana.CountCluster();
    //新しく「知り合い」が成立したボイドの数
    int new_friends = ana.UpdFriendsHistory(10);
    //クラスタに半透明の球を描画
    ShowClusterBall();

    //カウンタが最大値 (初期値1000) となるまで、解析を続ける
    if (cls_count < countmax)
    {
        //クラスタのフレーム内総数
        cls_sum += ana.csum;
        //クラスタの1フレーム平均
        cls_mean = cls_sum / cls_count;
        //計算フレーム数
        cls_count++;

        //新しい知り合いが一件でも成立すれば、
        //「知り合い成立機会」を一つ増やす
        if (new_friends >= 1)
        {
            make_friends_frame += 1;
        }
    }
}

```

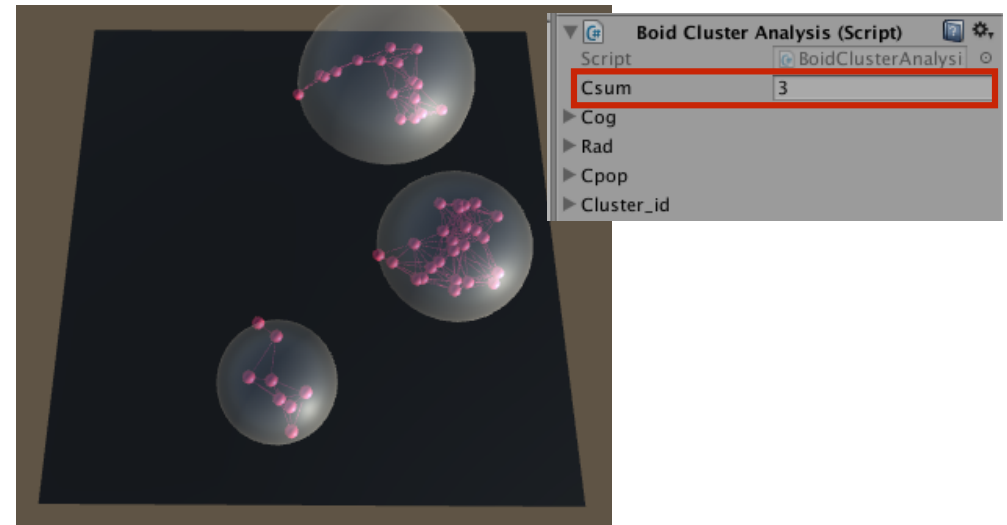
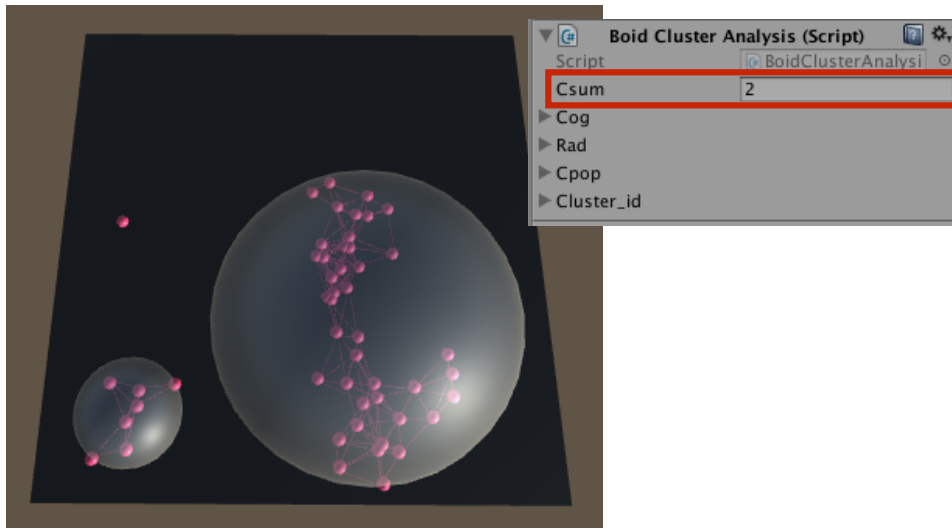
Update()

Aボタンで mode_analysis が true となると、クラスタのフレーム内平均 (cls_mean) ・知り合い成立回数 (make_friends_frame) をフレーム数 : countmax を上限として計算します。

A 解析モードの ON / OFF の切り替え

解析モードがONになるとクラスターが可視化され, Boid Cluster Analysis コンポーネントの<Csum>にクラスター数が表示されます。

(ここでは, クラスターを<個体数の1/10の可視距離内集団>と定義しています)

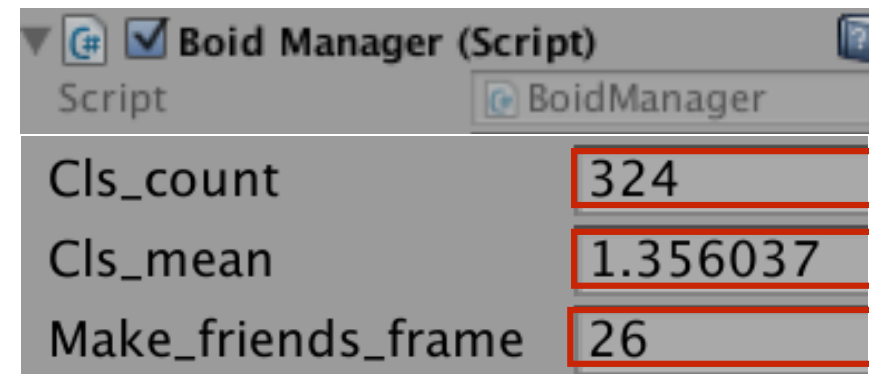


I ボイドの位置・速度の初期化

I ボタンを押した時点から, クラスターのフレーム内平均の計算・知り合い成立機会フレーム数のモニタが開始します。

Boid Manager コンポーネントに計算結果がリアルタイムで表示されます。

30フレームの間、同一のクラスターにいるボイドのペアについては「知り合い成立」とみなします。



cls_count 計算フレーム数 (最大1000)

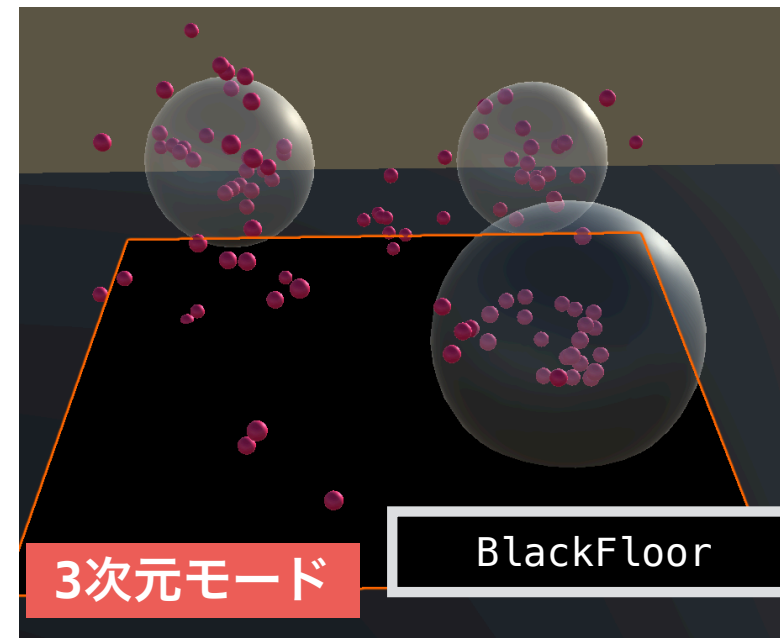
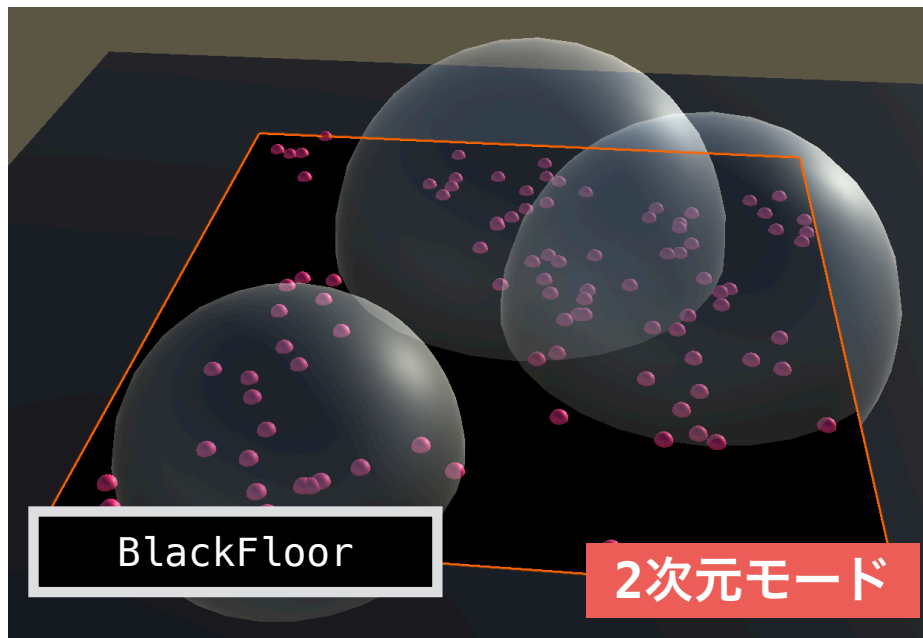
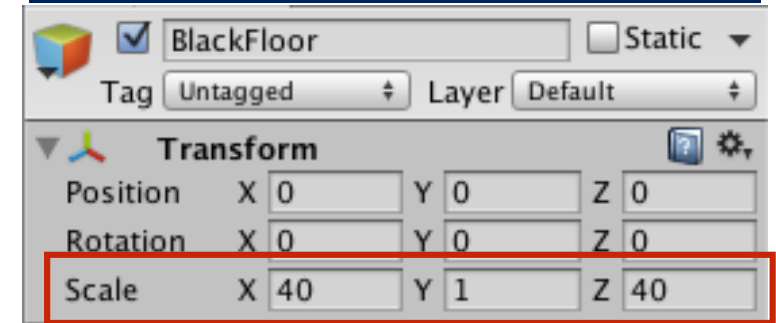
cls_mean クラスターのフレーム内平均

make_friends_frame 知り合い成立機会フレームの数

2次元モード・3次元モード

- デフォルトでは, 400×400 (xz) の空間をボイドが動き回りますが, 「D」 ボタンを押すと, 三次元モードとなり, $400 \times 400 \times 200$ (xzy) の空間を使うことができるようになります。
- 「I」 ボタンを押すと, すべてのボイドの位置と速度が初期化されます。

床 (Plane) も初期設定では, 400×400 のサイズとなっています。



2 / 3

二次元モード・三次元モードの切り替え

I

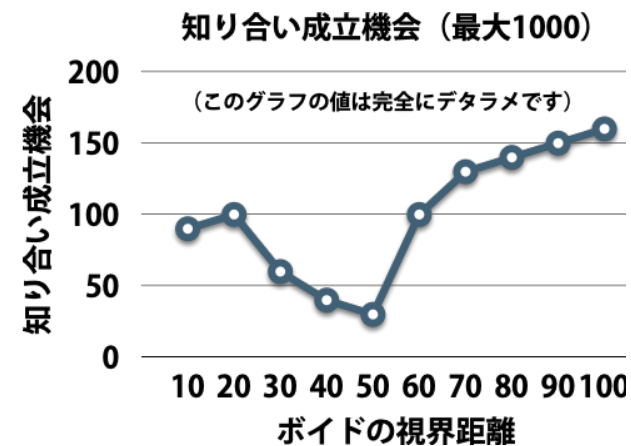
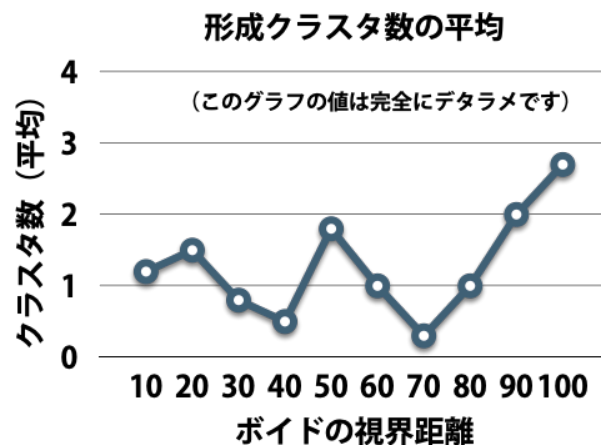
ボイドの位置・速度の初期化

共通課題

2

二次元モードにします。

二次元空間における、ボイドの視界範囲とクラスタの数（フレーム内平均）および、知り合い成立機会（フレーム内総和）の関係を調べて、以下のようなグラフを作成してください。



ただし、視界範囲以外の変数は以下の値に固定してください。

**neighbor_space = 10, bsum = 50,
c1 = 0.1; c2 = 15.0; c3 = 0.01;**

上記の条件でグラフを作成し、グラフから読み取れることを考察してください。そのなかで、自分が、この群衆のメンバーであることを想像して、「**豊かな関係性とは何か**」に着目して考察してください。

共通課題の詳細

2

二次元モードにします。

A

解析モードにします。

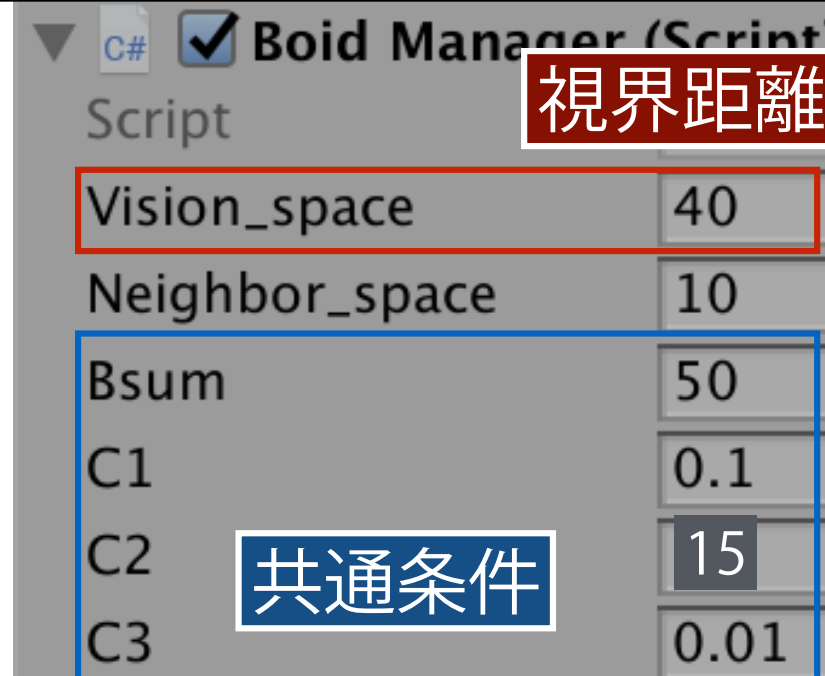
I

位置と解析を初期化します。

解析は、「Aキー」を押した後、「Iキー」を押してから約3000フレーム（計算機の特性に合わせて適当なところで切ってもらってOK）におけるフレーム内平均値（クラスタ）およびフレーム内総和（知り合い成立機会）として算出します。

全ての視界距離の値を調べる必要はありません。グラフの概形がわかるために必要なデータ量を各自で判断してください。

視界距離値は、直接にGUIで vision_space の値を設定してください。



Cl	324
Cls_mean	1.356037
Make_friends_frame	26

知り合い成立機会

「Iキー」を押した後、3000フレームが経過すると、自動的に計算が終了します。

自由課題

2

二次元モード

3

三次元モード

二次元空間または三次元空間ボイド（どちらでもよい）において、ルール1・ルール2・ルール3がクラスタ形成あるいは「豊かな関係性の構築」にどのように関わっているかについて考察してください。

目視による考察でも良いが、データに基づいた考察が望ましい。この際、目的に応じて、共通課題とは異なる変数の値を選び、 $c1 \cdot c2 \cdot c3$ の値を変化させた時の「クラスタ数」「知り合い成立機会」の変化を計測するなどしてみてください。

意欲があれば、「BoidClusterAnalysis.cs」に独自のメソッドを追記し、新たな指標を導入してもよい。

課題C1 (提出先)

提出ファイル

共通課題と自由課題に関する書類を「1850XX」という名前のフォルダに入れて、そのフォルダを圧縮したもの（「WorkC1_1850XX.zip」）を提出してください。

提出先

Dropbox のファイルリクエストにより提出します。投稿先のリンクは、以下のページから。

<http://lab.kenrikodaka.com/class/2020/mediaengineering.html#aC1>

06/24

BOID (5) | 集合知解析

[\[資料PDF\]](#)

[\[YOUTUBE\]](#)

[出席報告しなくていいです]

07/01

[\[課題C1提出 | 7月12日締切\]](#)

期限

7/12 (日)