明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

プログラミング演習 (8)
配列

中村，小松
小林，橋本

目標

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- Processing で配列に挑戦！
  - ゲームで沢山の敵を扱うにはどうするか？
  - 不均等に並べられた沢山の丸をどんどん移動するにはどうしたらよいか？
- 課題:
  - 避けゲームを作ってみる
  - パラパラアニメを作ってみよう
  - 占いアプリを作ってみよう

沢山の情報を扱うには？

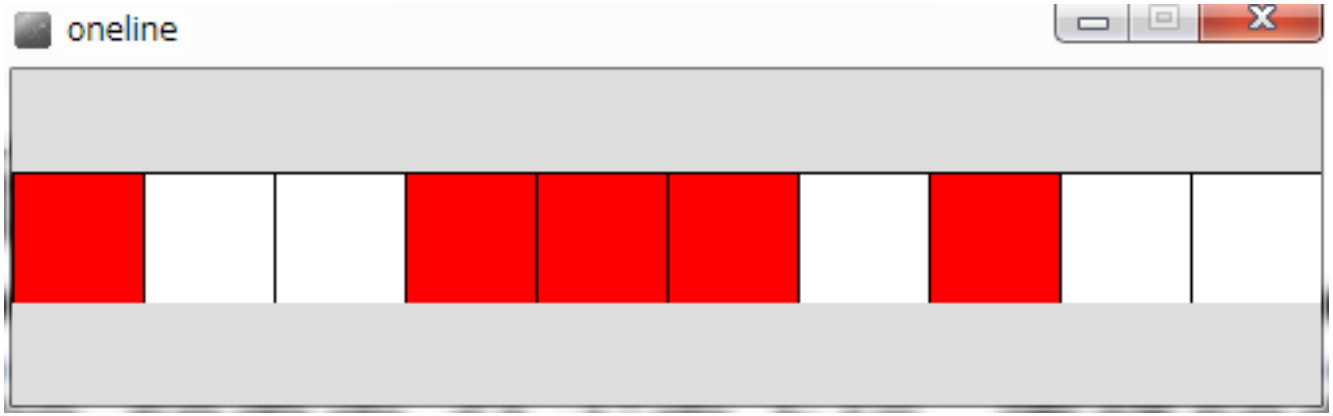
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ゲームの沢山のマス目をどう管理するか？
- トランプのカードや色々なスコアの管理
- 沢山の敵を同時に動かしたい
  - 沢山の敵の場所を管理するにはどうするか？
  - 沢山の円が一度に動き，その円をマウスで避けるような簡単なゲームを作ってみよう！

1次元の掲示板

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) 500x50のウインドウ上に，縦横50ピクセルの正方形を横に10個並べたボードを作成し，正方形の内部をクリックする度にその図形の色が【白→赤→白→赤】と変化させるようにする



1次元の掲示板

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方
  - 信号のプログラムの応用で，10個の正方形の状態記憶する整数（int型）の変数を10個用意
    - int status0, status1, status2, status3, ..., status8, status9;
  - クリック時のマウス座標（mouseX, mouseY)から，どの図形内でクリックされたかを判定し，そのstatusXの値を0→1→0→1と書き換える
  - draw()内で，statusXの情報に基づき0なら白色で，1なら赤色で正方形を描画する

```
int  status0 = 0;
int  status1 = 0;
int  status2 = 0;
int  status3 = 0;
int  status4 = 0;
int  status5 = 0;
int  status6 = 0;
int  status7 = 0;
int  status8 = 0;
int  status9 = 0;

void setup(){
    size( 500, 50 );
}
```

```
void draw(){
    background( 255 );
    if( status0 == 0 ){
        fill( 255 );
    } else {
        fill( 255, 0, 0 );
    }
   rect( 0, 0, 50, 50 );
    if( status1 == 0 ){
        fill( 255 );
    } else {
        fill( 255, 0, 0 );
    }
   rect( 50, 0, 50, 50 );

        （たくさん略）

    if( status9 == 0 ){
        fill( 255 );
    } else {
        fill( 255, 0, 0 );
    }
   rect( 450, 0, 50, 50 );
}
```

```
void mousePressed(){
    // どこをクリックしたか
    int tx = mouseX / 50;

    if( tx == 0 ){
        status0 ++;
        if( status0 > 1 ){
            status0 = 0;
        }
    } else if( tx == 1 ){
        status1 ++;
        if( status1 > 1 ){
            status1 = 0;
        }
    } else if( tx == 2 ){

            （たくさん略）

    } else if( tx == 9 ){
        status9 ++;
        if( status9 > 1 ){
            status9 = 0;
        }
    }
}
```

やってられん（怒）

配列

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 配列とは，ある同じ役割を持つ変数の固まり
- 変数名ではなく，変数名＋何番目かで値を代入したり取り出したりすることが可能
- 配列の定義

```
int [] status = new int [10];
```

変数の型 [] 変数名 = new 変数の型 [要素数];

- 変数への代入 `status[5] = 1;`
- 変数の値の出力 `println( status[5] );`

配列の注意点

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 配列の要素数が n の場合，**配列は0番目から始まり，n-1 番目まで**となる

  （例）int [] value = new int [10]; の場合

  value[0], value[1], value[2], ..., value[9] までとなる

- 配列の何番目かを指定する際に，指定できない数を指定するとエラーが出る

  （例）int [] value = new int [10]; の場合

  value[-1], value[10] などは存在しないのでおかしい

配列を使ってみる

```
int  []  status = new status [10];

void setup(){
    size( 500, 50 );
    status[0] = 0;
    status[1] = 0;
    status[2] = 0;
    status[3] = 0;
    status[4] = 0;
    status[5] = 0;
    status[6] = 0;
    status[7] = 0;
    status[8] = 0;
    status[9] = 0;
}
```

```
void draw(){
    background( 255 );
    if( status[0] == 0 ){
        fill( 255 );
    } else {
        fill( 255, 0, 0 );
    }
   rect( 0, 0, 50, 50 );
    if( status[1] == 0 ){
        fill( 255 );
    } else {
        fill( 255, 0, 0 );
    }
   rect( 50, 0, 50, 50 );

        ( たくさん略 )

    if( status[9] == 0 ){
        fill( 255 );
    } else {
        fill( 255, 0, 0 );
    }
   rect( 450, 0, 50, 50 );
}
```

```
void mousePressed(){
    // どこをクリックしたか
    int tx = mouseX / 50;
    status[tx]++;
    if( status[tx] > 1 ){
        status[tx] = 0;
    }
}
```

status[数字]とすることで楽！

配列を使ってみる

```
int  []  status = new status [10];
void setup(){
    size( 500, 50 );
    status[0] = 0;
    status[1] = 0;
    status[2] = 0;
    status[3] = 0;
    status[4] = 0;
    status[5] = 0;
}
```

```
void draw(){
    background( 255 );
    if( status[0] == 0 ){
        fill( 255 );
    } else {
        fill( 255, 0, 0 );
    }
    rect( 50, 0, 50, 50 );

        （たくさん略）

    if( status[9] == 0 ){
        fill( 255 );
    } else {
        fill( 255, 0, 0 );
    }
    rect( 450, 0, 50, 50 );
}
```

```
        status[tx] = 0;
    }
}
```

ただ，drawは相変わらず長いし
わかりにくいのでは？

status[数字]と
することで楽！

配列＋繰り返し

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 配列は繰り返しとの組合せにより能力発揮！
  - 繰り返しで増えていく数字を利用し，配列の何番目を指すかを指定する

配列＋繰り返し最強！

```
int  []  status = new int [10];
void setup(){
    size( 500, 50 );
    int i=0;
    while( i<10 ){
        status[i] = 0;
        i++;
    }
}
```

```
void draw(){
    background( 255 );
    int i=0;
    while( i<10 ){
        if( status[i] == 0 ){
            fill( 255 );
        } else {
            fill( 255, 0, 0 );
        }
        rect( i*50, 0, 50, 50 );
        i++;
    }
}
```

```
void mousePressed(){
    // どこをクリックしたか
    int tx = mouseX / 50;
    status[tx]++;
    if( status[tx] > 1 ){
        status[tx] = 0;
    }
}
```

配列＋繰り返し（for）

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 配列は繰り返しとの組合せにより能力発揮！
  - 繰り返しで増えていく数字を利用し，配列の何番目を指すかを指定する
  - for は配列との組み合わせでかなり有効！

```
int  []  status = new int [10];
void setup(){
    size( 500, 50 );
    for( int i=0; i<10; i++ ){
        status[i] = 0;
    }
}
```

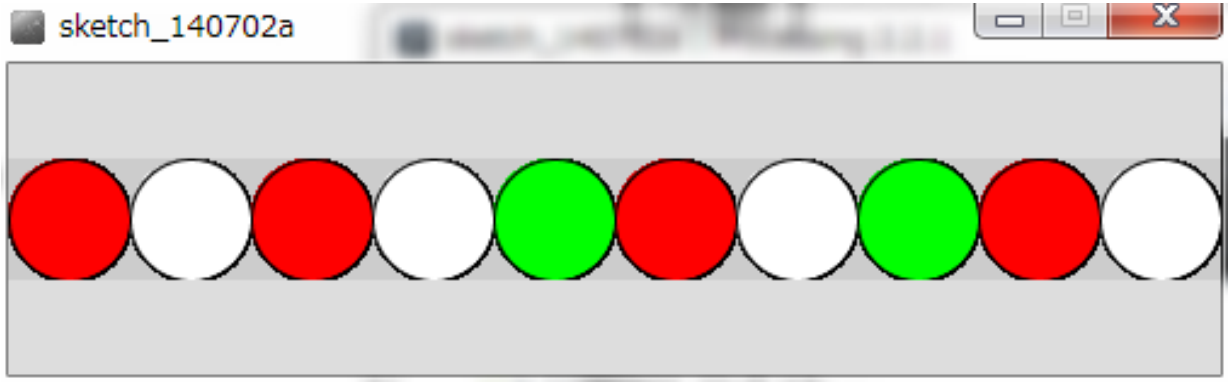
```
void draw(){
    background( 255 );
    for( int i=0; i<10; i++ ) {
        if( status[i] == 0 ){
              fill( 255 );
        } else {
              fill( 255, 0, 0 );
        }rect( i*50, 0, 50, 50 );
    }
}
```

```
void mousePressed(){
    // どこをクリックしたか
    int tx = mouseX / 50;
    status[tx]++;
    if( status[tx] > 1 ){
         status[tx] = 0;
    }
}
```

演習

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

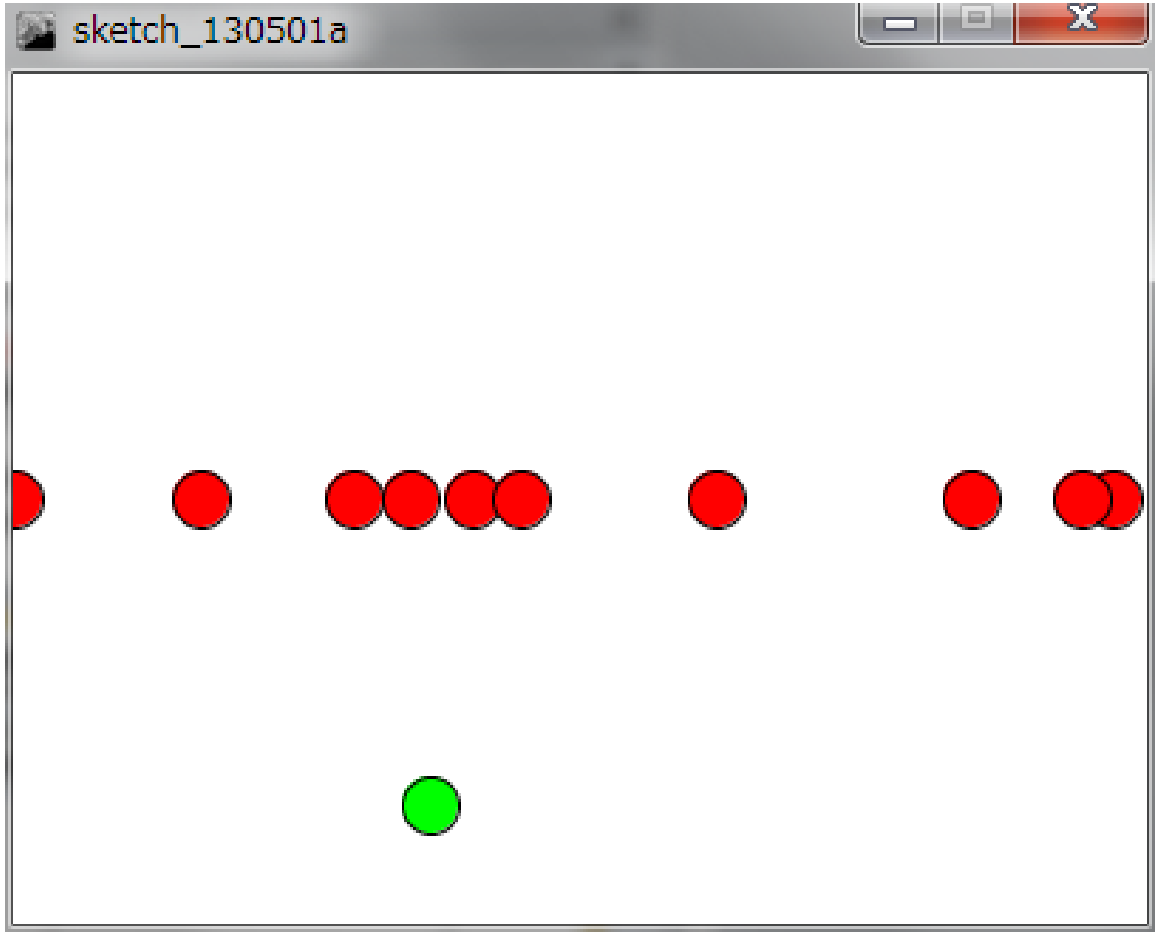
- 直径50の円を横に10個並べたボードを作成せよ。円の内部をクリックする度に、その円の色が【白→赤→黄→緑→白】と変化させるようにすること。



円を沢山描画する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) 400x300の画面上の，Y座標150の場所に円を10個，カーソルの下に円を描画しよう（直径20）



円を沢山描画する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方
  - カーソルの下に描画するには mouseX, mouseY
    - ellipse(mouseX, mouseY, 20, 20);
  - 10個の円のX座標を記憶する実数（float型）の変数を10個用意
    - float posX0; float posX1; float posX2; ... , float posX9;
  - setup() 内で，10個の変数に random を利用してx座標の値を代入
    - 横幅が400なので，posX0 = random(400); ...
  - draw() 内で，10個の変数の値を利用して円を描画
    - ellipse(posX0, 150, 20, 20); ...

円を沢山描画する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

100個ならどうする？

```
float  posX0;
float  posX1;
float  posX2;
float  posX3;
float  posX4;
float  posX5;
float  posX6;
float  posX7;
float  posX8;
float  posX9;

void setup(){
    size( 400, 300 );
    posX0 = random(400);
    posX1 = random(400);
    posX2 = random(400);
    posX3 = random(400);
    posX4 = random(400);
    posX5 = random(400);
    posX6 = random(400);
    posX7 = random(400);
    posX8 = random(400);
    posX9 = random(400);
}
```

```
void draw(){
    background( 255 );
    fill( 0, 255, 0 );
    ellipse( mouseX, mouseY, 20, 20 );
    fill( 255, 0, 0 );
    ellipse( posX0, 150, 20, 20 );
    ellipse( posX1, 150, 20, 20 );
    ellipse( posX2, 150, 20, 20 );
    ellipse( posX3, 150, 20, 20 );
    ellipse( posX4, 150, 20, 20 );
    ellipse( posX5, 150, 20, 20 );
    ellipse( posX6, 150, 20, 20 );
    ellipse( posX7, 150, 20, 20 );
    ellipse( posX8, 150, 20, 20 );
    ellipse( posX9, 150, 20, 20 );
}
```

円を沢山描画する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 配列で書き直すとこんな感じ

```
float  []  posX = new float [10];
void setup(){
    size( 400, 300 );
    posX[0] = random(400);
    posX[1] = random(400);
    posX[2] = random(400);
    posX[3] = random(400);
    posX[4] = random(400);
    posX[5] = random(400);
    posX[6] = random(400);
    posX[7] = random(400);
    posX[8] = random(400);
    posX[9] = random(400);
}
```

```
void draw(){
    background( 255 );
    fill( 0, 255, 0 );
    ellipse( mouseX, mouseY, 20, 20 );
    fill( 255, 0, 0 );
    ellipse( posX[0], 150, 20, 20 );
    ellipse( posX[1], 150, 20, 20 );
    ellipse( posX[2], 150, 20, 20 );
    ellipse( posX[3], 150, 20, 20 );
    ellipse( posX[4], 150, 20, 20 );
    ellipse( posX[5], 150, 20, 20 );
    ellipse( posX[6], 150, 20, 20 );
    ellipse( posX[7], 150, 20, 20 );
    ellipse( posX[8], 150, 20, 20 );
    ellipse( posX[9], 150, 20, 20 );
}
```

配列＋繰り返し

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 配列は繰り返しとの組合せにより能力発揮！
  - 繰り返しで増えていく数字を利用し，配列の何番目を指すかを指定する

```
float  []  posX = new float [10];
void setup(){
    size( 400, 300 );
    int i=0;
    while( i<10 ){
        posX[i] = random(400);
        i++;
    }
}
```

```
void draw(){
    background( 255 );
    fill( 0, 255, 0 );
    ellipse( mouseX, mouseY, 20, 20 );
    fill( 255, 0, 0 );
    int  j = 0;
    while( j<10 ){
       ellipse( posX[j], 150, 20, 20 );
       j++;
    }
}
```

配列＋繰り返し（for）

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 配列は繰り返しとの組合せにより能力発揮！
  - 繰り返しで増えていく数字を利用し，配列の何番目を指すかを指定する
  - for は配列との組み合わせでかなり有効！

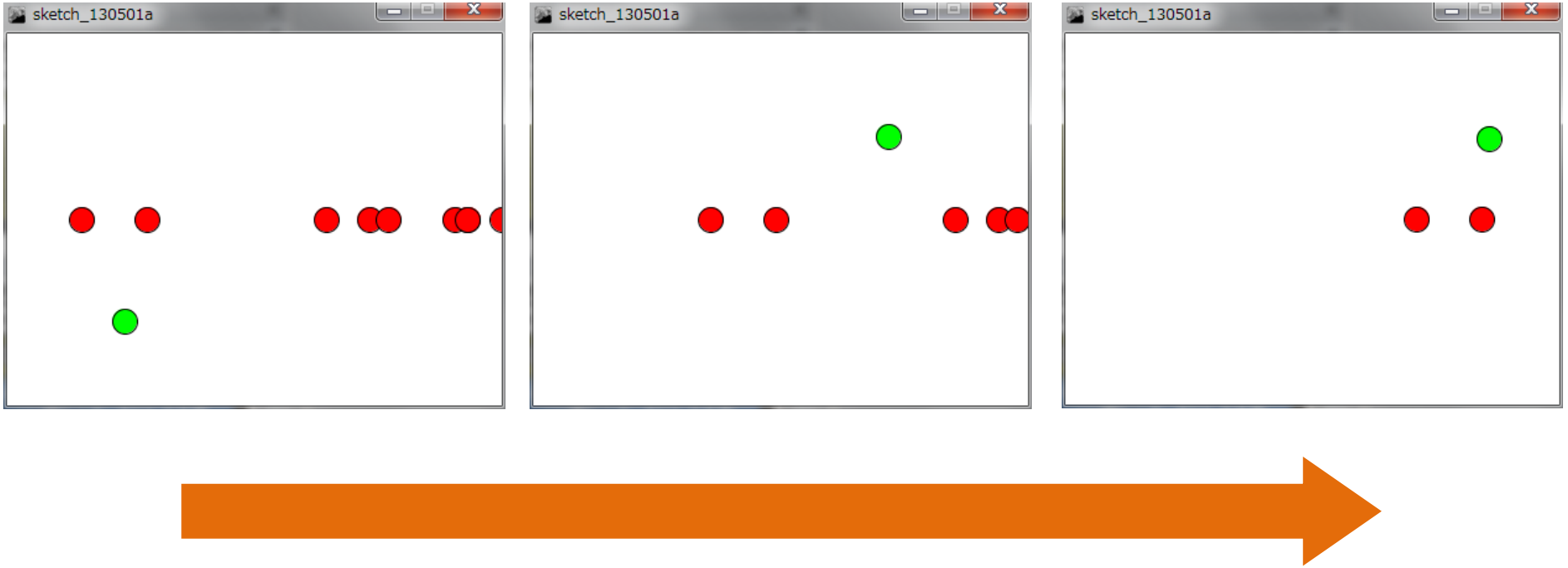
```
float  []  posX = new float [10];
void setup(){
    size( 400, 300 );
    for( int i=0; i<10; i++ ){
        posX[i] = random(400);
    }
}
```

```
void draw(){
    background( 255 );
    fill( 0, 255, 0 );
    ellipse( mouseX, mouseY, 20, 20 );
    fill( 255, 0, 0 );
    for( int  j=0; j<10; j++ ){
       ellipse( posX[j], 150, 20, 20 );
    }
}
```

円を一斉に動かす

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) 先ほど作成した400x300の画面に配置された10個の円を右方向に一斉に動かしてみよう



円を一斉に動かす

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方
  - 10個のX座標を格納する float 型の配列を用意
  - 10個の配列に random で適当な値を代入
  - 10個の配列の値に応じて円を描画
  - 1回描画する度に，配列のX座標を1つ右へ移動

円を一斉に動かす

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(A) 先ほど作成した400x300の画面に配置された10個の円を右方向に一斉に動かしてみよう

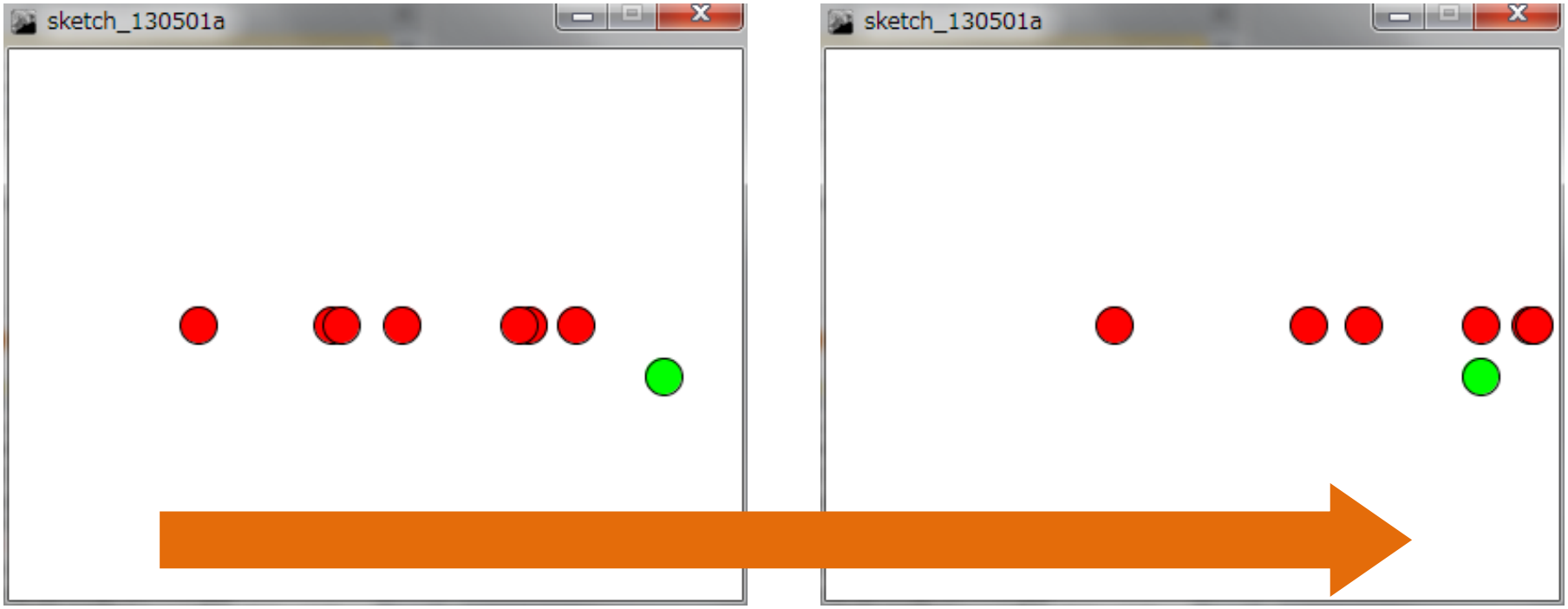
```
float  []  posX = new float [10];
void setup(){
    size( 400, 300 );
    int i=0;
    while( i<10 ){
        posX[i] = random(400);
        i++;
    }
}
```

```
void draw(){
    background( 255 );
    fill( 0, 255, 0 );
    ellipse( mouseX, mouseY, 20, 20 );
    fill( 255, 0, 0 );
    int  j = 0;
    while( j<10 ){
        ellipse( posX[j], 150, 20, 20 );
        posX[j] ++;
        j++;
    }
}
```

円を一斉に動かす ver. 2

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) 先ほど作成した400x300の画面に配置された10個の円を右方向に一斉に動かしてみよう．ただし円ごとにスピードを変えてみよう



円を一斉に動かす ver. 2

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方
  - 10個のそれぞれの円についてスピードを格納するfloat 型の配列を用意
    - `float  [] speed = new float [10];`
  - スピード用の配列に random で適当な値を代入
  - 10個の配列の値に応じて円を描画
  - 1回描画する度に，スピードの量だけ配列のX座標を1つ右へ移動
    - j 番目のX座標の値に，j 番目のスピードを足す

円を一斉に動かす ver. 2

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- それぞれのスピードで円が右に動いていく！
  - random(5) の5の値を大きくすると，スピードが速くなる！

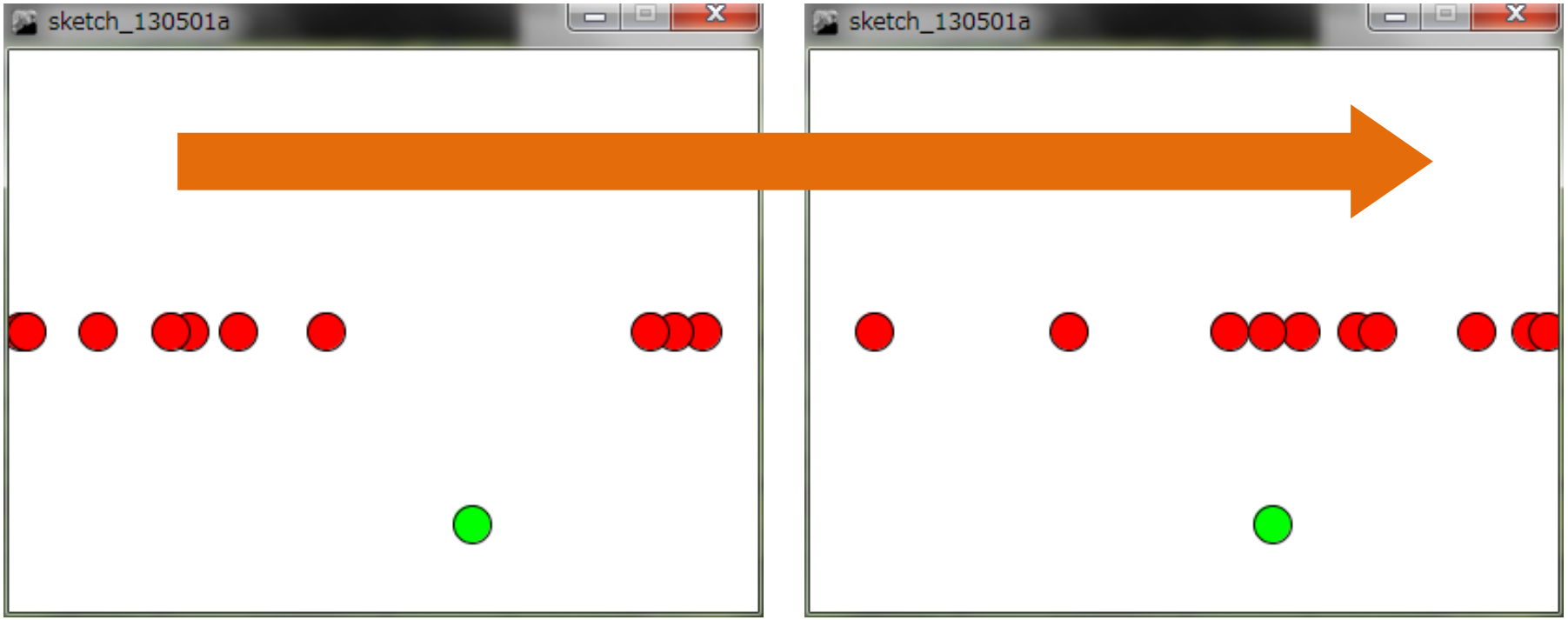
```
float  []  posX = new float [10];
float  [] speed = new float [10];
void setup(){
    size( 400, 300 );
    int i=0;
    while( i<10 ){
        posX[i] = random(400);
        speed[i] = random( 5 );
        i++;
    }
}
```

```
void draw(){
    background( 255 );
    fill( 0, 255, 0 );
    ellipse( mouseX, mouseY, 20, 20 );
    fill( 255, 0, 0 );
    int  j = 0;
    while( j<10 ){
       posX[j] = posX[j] + speed[j];
       ellipse( posX[j], 150, 20, 20 );
       j++;
    }
}
```

円を一斉に動かす ver. 3

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) 先ほど作成した400x300の画面に配置された10個の円を右方向に一斉に動く円は，右端に来ると左端から出てくるようにしよう



円を一斉に動かす ver. 3

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方
  - 円が右端に来た時，円を左端に登場させる
  - 右端のX座標は400，左端のX座標は0
  - 移動後の円の座標を計算し，400を超えていたら，0にする（または，座標から400引く）➔ if文！

円を一斉に動かす ver. 3

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- if(posX[j] > 400) なら posX[j] から 400引く！

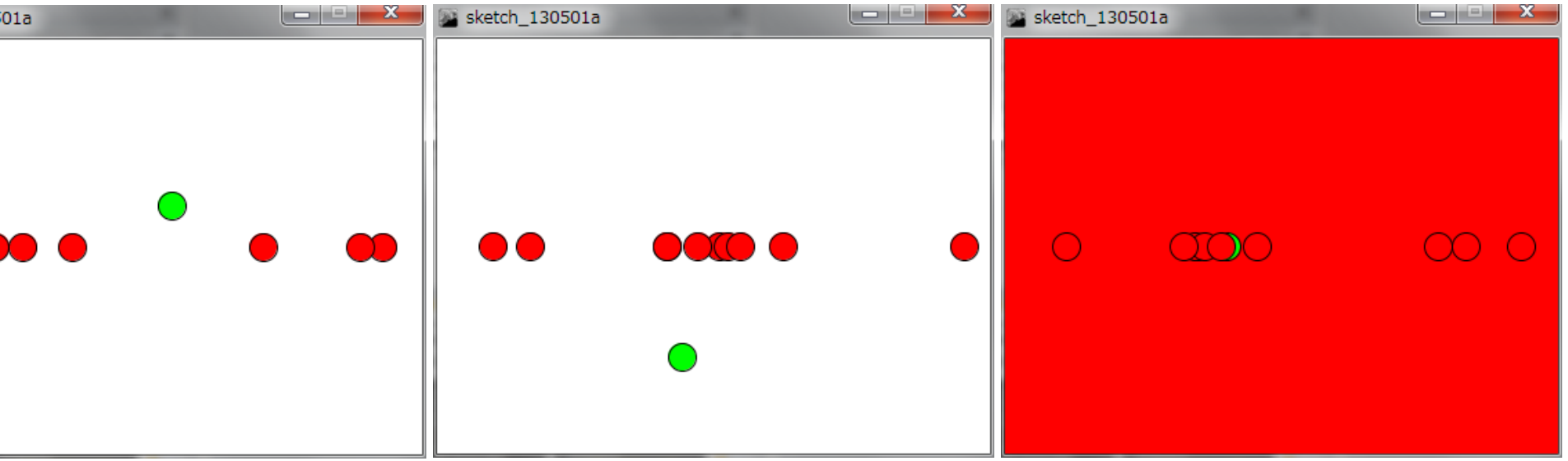
```
float  []  posX = new float [10];
float  [] speed = new float [10];
void setup(){
    size( 400, 300 );
    int i=0;
    while( i<10 ){
        posX[i] = random(400);
        speed[i] = random( 5 );
        i++;
    }
}
```

```
void draw(){
    background( 255 );
    fill( 0, 255, 0 );
    ellipse( mouseX, mouseY, 20, 20 );
    fill( 255, 0, 0 );
    int  j = 0;
    while( j<10 ){
        posX[j] = posX[j] + speed[j];
        if( posX[j] > 400 ){
            posX[j] = posX[j] - 400;
        }
        ellipse( posX[j], 150, 20, 20 );
        j++;
    }
}
```

一斉に動く円との衝突判定

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) 先ほど作成した400x300の画面に配置された10個の右方向に動く円と，マウスカーソルが衝突したら背景を赤色にしよう！



一斉に動く円との衝突判定

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方
  - カーソルとそれぞれの円の距離は dist で求める
  - j 番目の円とカーソルまでの距離は
    - `dist( posX[j], 150, mouseX, mouseY );`
  - dist が20以下なら円と衝突しているので，その場合に背景の色を赤色にする
  - 背景の色を変更するための変数を用意
    - int crash; など．通常は 0，クラッシュすると 1 にするなど
  - 背景色を変更の変数に応じて背景色を変更

一斉に動く円との衝突判定

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
float  []  posX = new float [10];
float  [] speed = new float [10];
int  crash  = 0;

void setup(){
    size( 400, 300 );
    int i=0;
    while( i<10 ){
        posX[i] = random(400);
        speed[i] = random( 5 );
        i++;
    }
}
```

```
void draw(){
    if( crash == 0 ){
        background( 255, 255, 255 );
    } else {
        background( 255, 0, 0 );
    }
    fill( 0, 255, 0 );
    ellipse( mouseX, mouseY, 20, 20 );
    fill( 255, 0, 0 );
    int  j = 0;
    while( j<10 ){
       posX[j] = posX[j] + speed[j];
       if( posX[j] > 400 ){
            posX[j] = posX[j] - 400;
       }
       ellipse( posX[j], 150, 20, 20 );
       if( dist( posX[j], 150, mouseX, mouseY ) < 20 ){
            crash = 1;
       }
       j++;
    }
}
```

演習

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 一斉に動く円の数を，10個から20個に増やしてみましょう
- 一斉に動く円の数を，10個から100個に増やしてみましょう
- 一斉に動く円の方向を左方向にしてみましょう
  - speed をマイナスにしたらOK！
- 一斉に動く円を，右方向にも左方向にも動くようにしてみましょう
  - speed を-5から5までにしたらOK！

状態の管理: アイテムガチャ

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) クリックする度にランダムに取得できる10個のアイテムについて，取得したアイテムの数をカウントし，その数を表示するプログラムを作成せよ

- 考え方
  - 要素数が10個の配列を作成する
  - クリックの度にランダムに値を生成する
  - 生成された値に対応する要素の配列の値を増やす
  - それぞれの数をprintlnで表示する

```
item0: 3
item1: 1
item2: 1
item3: 3
item4: 2
item5: 3
item6: 2
item7: 0
item8: 5
item9: 1
```

状態の管理: アイテムガチャ

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

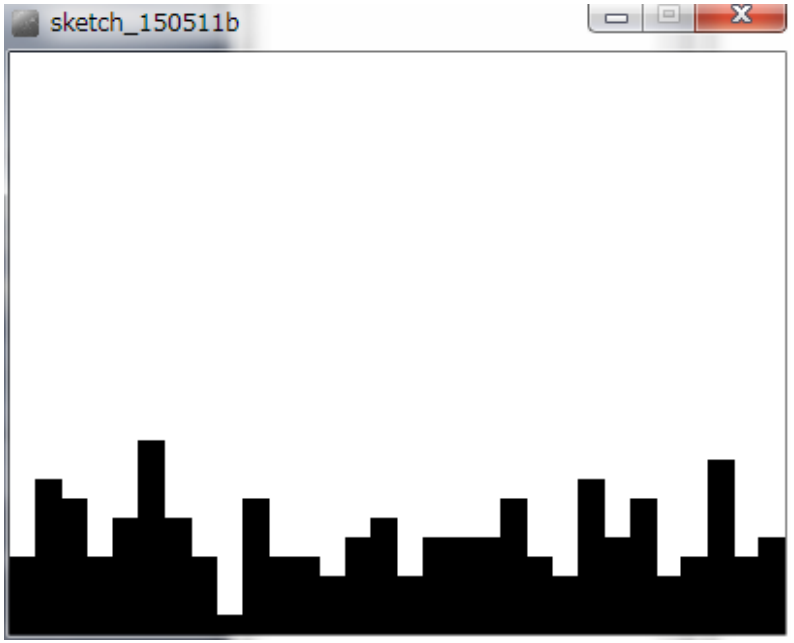

```
int [] items = new int [10];

void setup(){
    int i = 0;
    while( i<10 ){
        items[i] = 0;
        i++;
    }
}

void draw(){
}

void mousePressed(){
    int num = (int)random(10);
    items[num] ++;
    int i = 0;
    while( i < 10 ){
        println( "item"+i+": "+items[i] );
        i++;
    }
}
```

状態の管理: アイテムガチャ

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) クリックする度にランダムに取得できる30個のアイテムについて，取得したアイテムの数をカウントし，その数を表示するプログラムを棒グラフとして作成せよ

- 考え方
  - アイテム数を30個に変更
  - rectを使って表示しよう！



演習：コンプリートガチャ

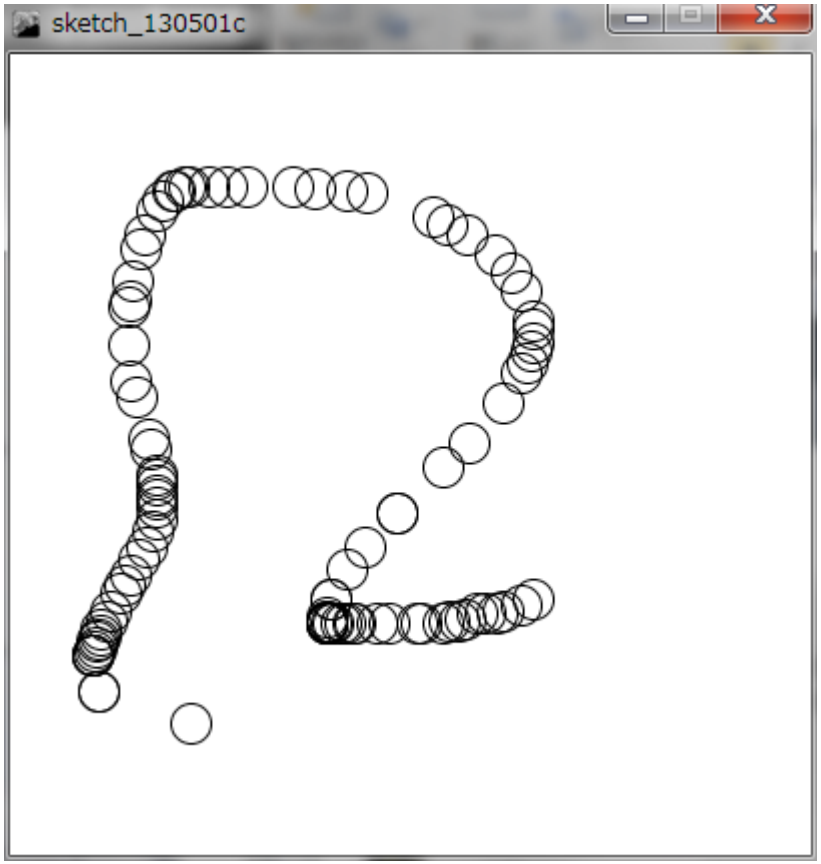
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

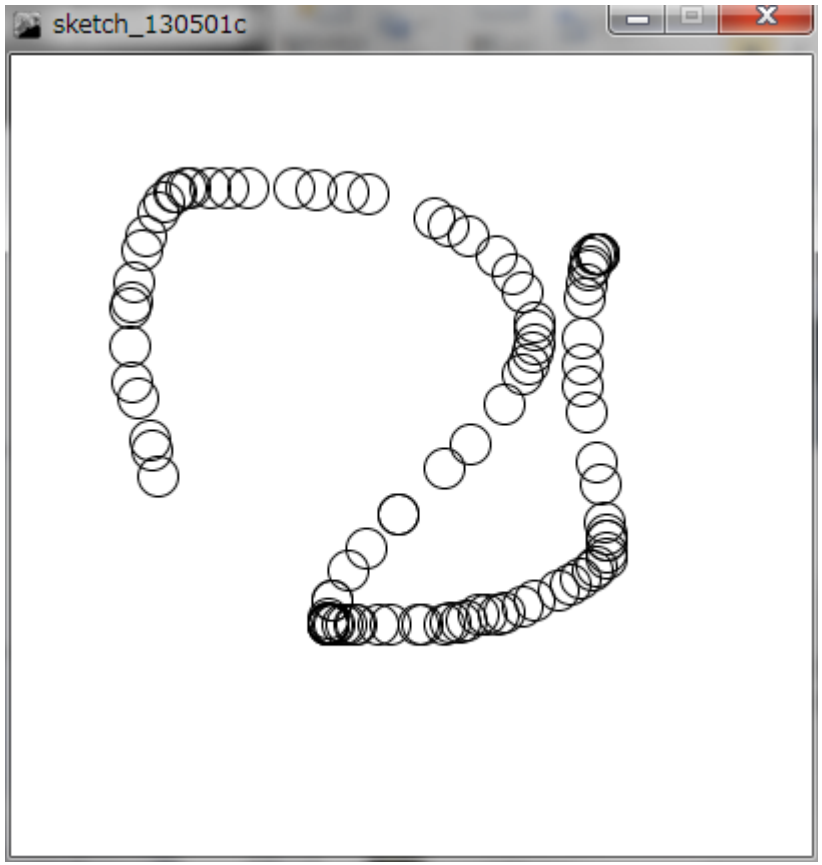
- 先述のプログラムを改良し，30個のアイテム全て1つ以上揃った時に「Complete」と表示するプログラムを作成せよ．また，その時に何回クリックしたかを表示せよ

マウスの軌跡を表示する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) マウスカーソルを追尾する100個の円を描画するにはどうするか？





マウスの軌跡を表示する

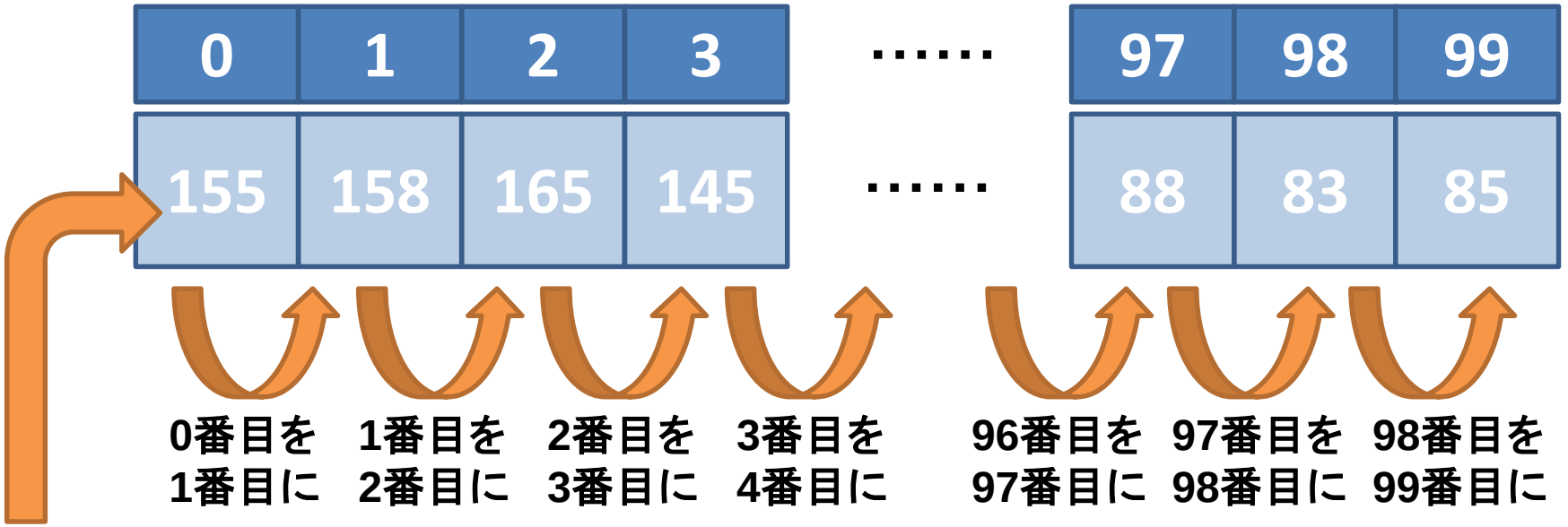
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方
  - マウスカーソルの座標は mouseX, mouseY
  - マウスカーソルの以前の座標を記録するため，要素数が100のXおよびYの配列用意
    - `int [] cursorX = new int [100];`
    - `int [] cursorY = new int [100];`
  - 現在のフレームの座標を0番目，1フレーム前の座標を1番目，2フレーム前の座標を2番目・・・とする
  - 配列cursorXとcursorYについて，0から99まで変化するすべての i についてcursorX[i], cursorY[i]を中心とした円を描画

マウスの軌跡を表示する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

cursorX



mouseX

```
cursorX[i] = cursorX[i-1];
cursorY[i] = cursorY[i-1];
```

1つずつ右へずらしていく

マウス軌跡を表示する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ありがちな間違い（setupと描画処理は省略）

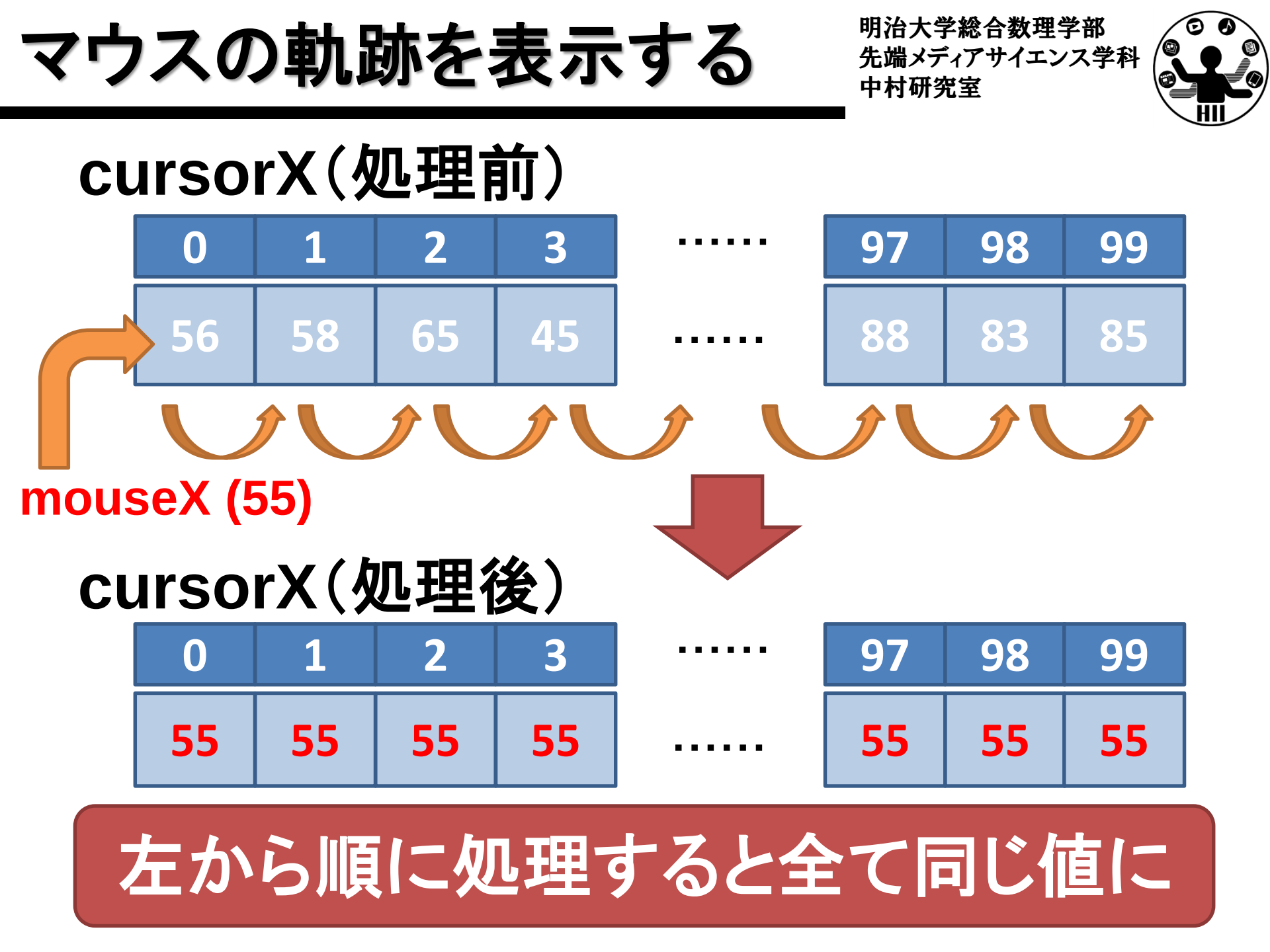
```
int [] cursorX = new int [100];
int [] cursorY = new int [100];
void draw(){
    cursorX[0] = mouseX;
    cursorY[0] = mouseY;
    int i=0;
    while( i <= 99 ){
        cursorX[i] = cursorX[i-1];
        cursorY[i] = cursorY[i-1];
        i++;
    }
    //  ここに描画の処理
}
```

何が間違っているか？

0番目を1番目に
1番目を2番目に
2番目を3番目に
3番目を4番目に
：

マウスの軌跡を表示する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



左から順に処理すると全て同じ値に

マウス軌跡を表示する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 右から順に処理する（setupと描画は省略）

```
int [] cursorX = new int [100];
int [] cursorY = new int [100];
void draw(){
    int i=99;
    while( i > 0 ){
        cursorX[i] = cursorX[i-1];
        cursorY[i] = cursorY[i-1];
        i--;
    }
    cursorX[0] = mouseX;
    cursorY[0] = mouseY;
    //  ここに描画の処理
}
```

98番目を99番目に
97番目を98番目に
96番目を97番目に
95番目を96番目に
：

```
int [] cursorX = new int [100];
int [] cursorY = new int [100];
void setup(){
    size( 400, 400 );
    noFill();
}
void draw(){
    background( 255 );
    int i=99;
    while( i > 0 ){
        cursorX[i] = cursorX[i-1];
        cursorY[i] = cursorY[i-1];
        i--;
    }
    cursorX[0] = mouseX;
    cursorY[0] = mouseY;
    i=0;
    while( i < 100 ){
        ellipse( cursorX[i], cursorY[i], 20, 20 );
        i++;
    }
}
```

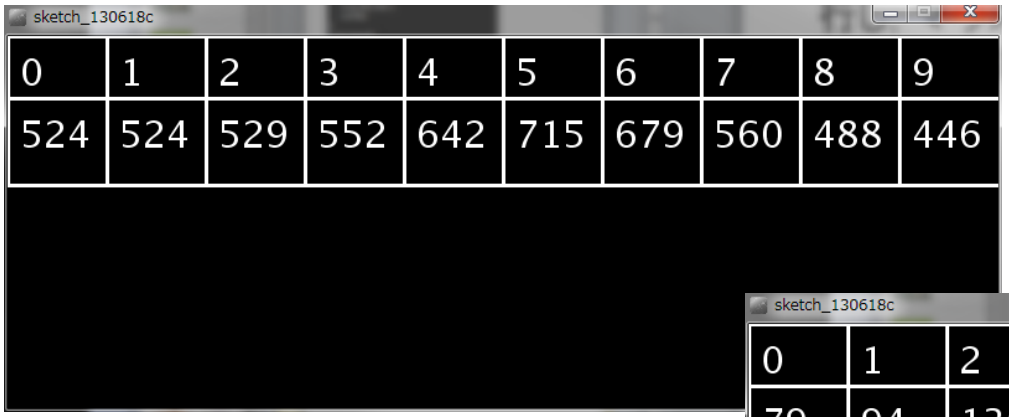
配列の内容を
1つずつずらす

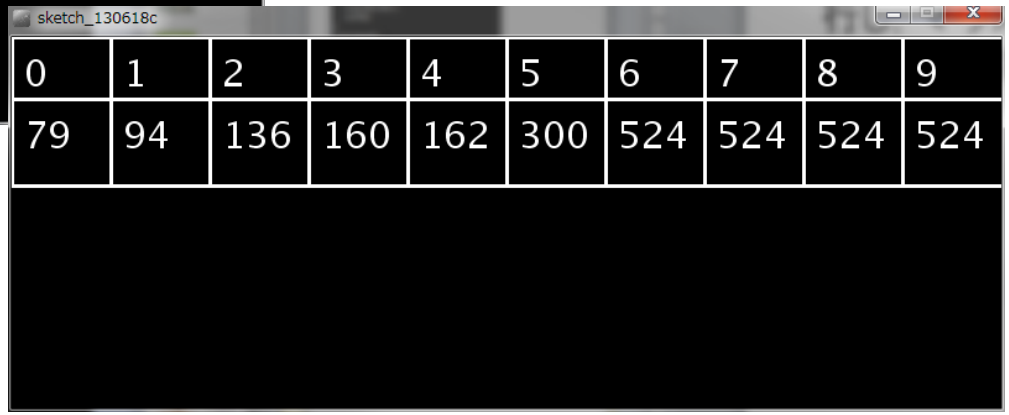
cursorX[i], cursorY[i] に
円を描画

配列の値がどう動くか？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 配列の値の動きがわかりにくい人のために，次ページにプログラムを用意したので入力して実行し，マウスを動かしてみましょう！（マウスのX座標が順に引き継がれます）





配列の値がどう動くか？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
void setup() {
    size( 800, 300 );
    frameRate(2);
    textSize( 30 );
    stroke(255);
    strokeWeight(3);
}
```

配列の内容を
1つずつずらす

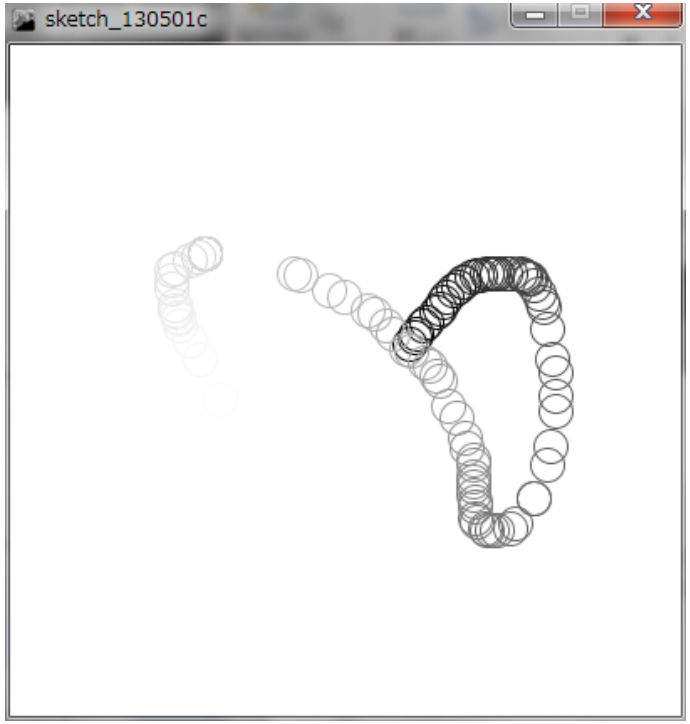
```
int [] arrayX = new int [10];
void draw() {
    background(0);
    for ( int i=9; i>0; i-- ) {
        arrayX[i] = arrayX[i-1];
    }
    arrayX[0] = mouseX;
    for ( int i=0; i<10; i++ ) {
        fill(0);
        rect( i*80, 0, (i+1)*80, 50 );
        rect( i*80, 50, (i+1)*80, 70 );
        fill(255);
        text( i, i*80+10, 40 );
        text( arrayX[i], i*80+10, 90 );
    }
}
```

予習問題

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- マウスの軌跡に円を描画するプログラムについて，古い軌跡ほど線を薄くするようにしてみましょう（strokeで色を変化）

  （ヒント）storke(255 * i / 100);



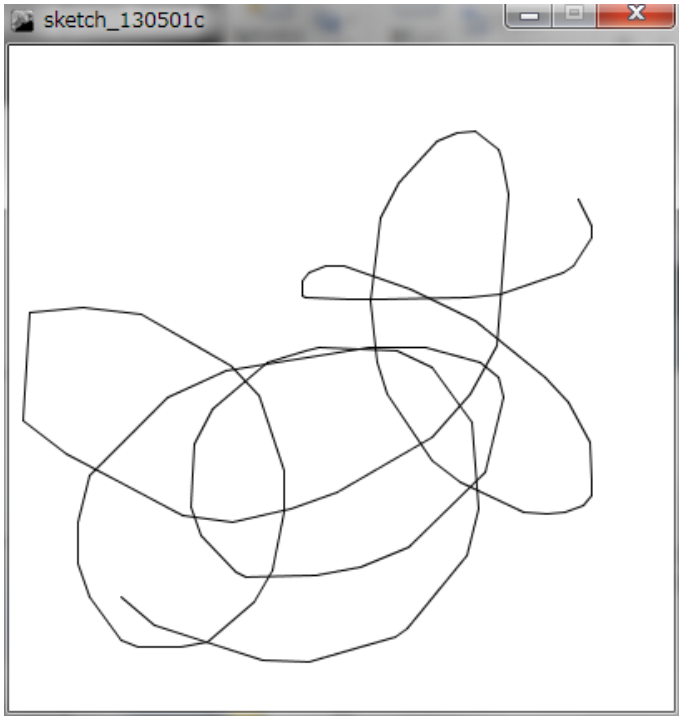
予習問題

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- マウスの軌跡に線を描画するプログラムを作ってみましょう

（ヒント）

```
line( cursorX[i], cursorY[i], cursorX[i-1], cursorY[i-1] );
```



配列のエラーに注意！

パラパラアニメーション

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

(Q) 用意した10枚の画像をパラパラ切り替えるアニメーションを作りたい

まず，画像の表示

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

PImage 画像用変数;

画像用変数 = loadImage("画像名"); で準備

image(画像用変数, x座標, y座標); で表示

画像はプログラムにドロップで利用可能に
（ドラッグアンドドロップしないと使えない）

```
PImage  mapImage = loadImage( "map.png" );
size( 640, 400 );
background(255);
image( mapImage, 0, 0 );
```

パラパラアニメーション

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方
  - 画像を10枚用意（ペイントで描いても，写真を撮影してもOK．名前は適当に順番を付けましょう）
  - 要素数が10の PImage 型の配列を作る
    - `PImage [] parapara = new PImage [10];`
  - setup で画像をすべて読み込む（loadImage）
    - `parapara[0] = loadImage( "gazo0.jpg" );`
  - draw で表示する画像番号を変数 i として準備
  - draw の度に i 番目の画像を表示
  - i が10になったら0に戻す
  - frameRate(10); で draw の更新速度を設定

パラパラアニメーション

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
PImage [] parapara = new PImage [10];
int  i = 0;

void setup(){
    size( 800, 600 );
    parapara[0]=loadImage("gazo0.jpg");
    parapara[1]=loadImage("gazo1.jpg");
    parapara[2]=loadImage("gazo2.jpg");
    parapara[3]=loadImage("gazo3.jpg");
    parapara[4]=loadImage("gazo4.jpg");
    parapara[5]=loadImage("gazo5.jpg");
    parapara[6]=loadImage("gazo6.jpg");
    parapara[7]=loadImage("gazo7.jpg");
    parapara[8]=loadImage("gazo8.jpg");
    parapara[9]=loadImage("gazo9.jpg");
    frameRate( 10 );
}
```

```
void draw(){
    image( parapara[i], 0, 0 );
    i++;
    if( i==10 ){
        i=0;
    }
}
```

ちなみに

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
parapara[0]=loadImage("gazo0.jpg");
parapara[1]=loadImage("gazo1.jpg");
parapara[2]=loadImage("gazo2.jpg");
parapara[3]=loadImage("gazo3.jpg");
parapara[4]=loadImage("gazo4.jpg");
      :
parapara[9]=loadImage("gazo9.jpg");
```

```
for( int j=0; j<10; j++ ){
   parapara[j]=loadImage("gazo"+j+".jpg");
}
```

予習問題

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

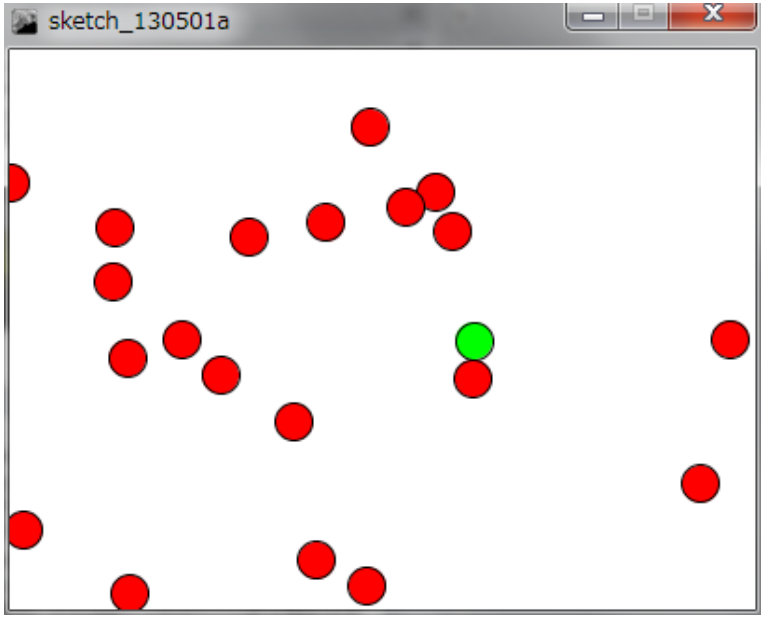
- 1回の試行で同一確率で1つのアイテムを得ることができるゲームにおいて，すべてのアイテムが揃った場合に何らかのスペシャルアイテムを得ることができる．さて，スペシャルアイテムを得るには何回の試行が必要だろうか？
  - 揃えるアイテムが2個の場合
  - 揃えるアイテムが10個の場合
  - 揃えるアイテムが50個の場合
  - 揃えるアイテムが100個の場合

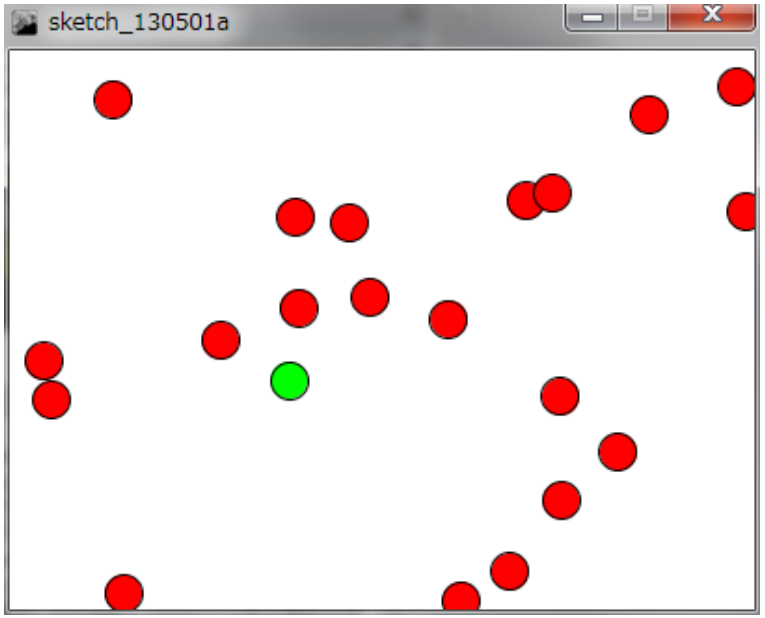
- 100回テストしてどれだけの数値になるか？

予習問題

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 400x300の画面内にX座標Y座標ともにランダムに配置された50個の円を描き，それぞれのX，Y方向へのスピードに応じて移動するようにせよ
  - posX, posY という座標の配列と，speedX, speedY というXおよびY方向の速度をもつ配列を導入！





予習問題

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 先述のプログラムで，端に行くと逆側から出てくるようにせよ
  - X座標が0より小さければ400プラス，X座標が400を超えれば400マイナス，Y座標も同じように
- マウスカーソルと衝突したらゲームオーバーにし，そこまで生き残った時間をスコアにせよ
  - ゲームオーバーかどうかを判定する gameover という変数を用意 gameover = 0; を衝突すると 1 にする
  - score という変数を用意して，gameover になるまでは draw の度に score を増やす！