



プログラミング演習 (8)

配列

中村, 橋本, 小松, 渡辺

目標

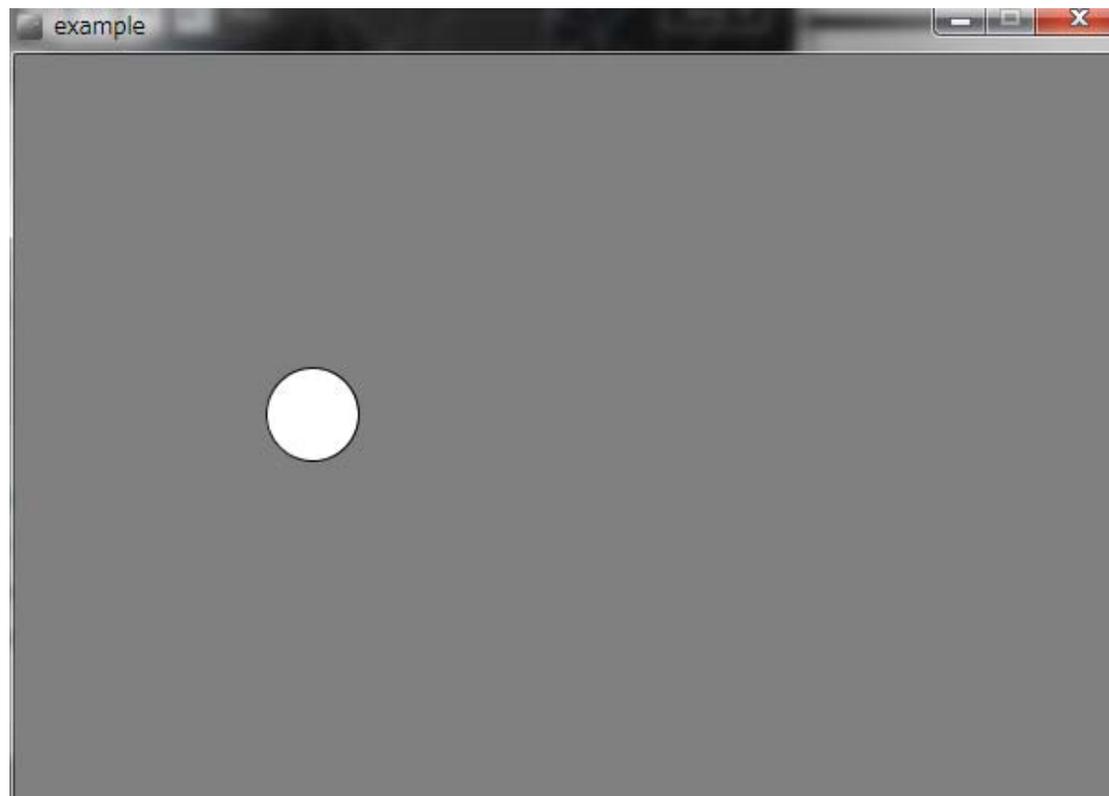


- Processing で配列に挑戦！
 - ゲームで沢山の敵を扱うにはどうするか？
 - 不均等に並べられた沢山の丸をどんどん移動するにはどうしたらよいか？
- 課題：
 - 避けゲームを作ってみる
 - パラパラアニメを作ってみよう
 - 占いアプリを作ってみよう

最初に



- 川連くん制作(協力原くん)のValueViewerを使いこなそう!



fill	true
str	hello
x	190
y	8.299983
hoge	1.0
fuga	50000000
bar	[0][0] 0
	[1][0] 0
	[2][0] 0
bar	[0][1] 0
	[1][1] 18
	[2][1] 0
bar	[0][2] 0
	[1][2] 0
	[2][2] 0
mouseY	3
x%height	189
mouseX	129

まずダウンロードしよう！

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



GitHub <https://github.com/nerunerunerune/ValueViewer> Sign in

nerunerunerune / ValueViewer

★ Star 0 Fork 1

～FMSプログラミング演習 I～ 中村先生のshowValueを別ウィンドウで半自動でやってくれる感じのアレ。

19 commits 1 branch 0 tags 2 contributors

branch: master ValueViewer / +

Merge pull request #4 from mactkg/add-gitignore

	nerunerunerune authored 20 hours ago	latest commit e59e14e47f
	.gitignore	add gitignore file 2 days ago
	README.md	change usage 2 days ago
	ValueViewer.pde	change usage 2 days ago
	compressor.sh	Ad 2 days ago
	example.pde	cha 2 days ago

ここをクリックしてダウンロード

- Code
- Issues 2
- Pull Requests 0
- Pulse
- Graphs
- Network

HTTPS clone URL
<https://github.com/>

You can clone with HTTPS, Subversion, and other methods.

- Clone in Desktop
- Download ZIP**

ファイルを解凍しよう



- ダウンロードした「ValueViewer-master.zip」をダブルクリックし、中から ValueViewer.pde を取り出しましょう

 compressor.sh	2013/06/25 23:24	SH ファイル	1 KB
 example.pde	2013/06/25 23:24	PDE ファイル	1 KB
 README.md	2013/06/25 23:24	MD ファイル	2 KB
 ValueViewer.pde	2013/06/25 23:24	PDE ファイル	12 KB

- 何度も使うので、適当なフォルダにコピーしておこう！

プログラムを書こう！

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



```
ValueViewer viewer;
```

```
void setup(){  
  // サイズは必要に応じて  
  size( 600, 400 );  
  viewer = new ValueViewer(this);  
}
```

```
void draw(){  
}
```

```
sketch_130627a | Processing 2.0  
File Edit Sketch Tools Help  
Java  
sketch_130627a  
ValueViewer viewer;  
  
void setup(){  
  size( 600, 400 );  
  viewer = new ValueViewer(this);  
}  
  
void draw(){  
}
```

10

登録しよう！



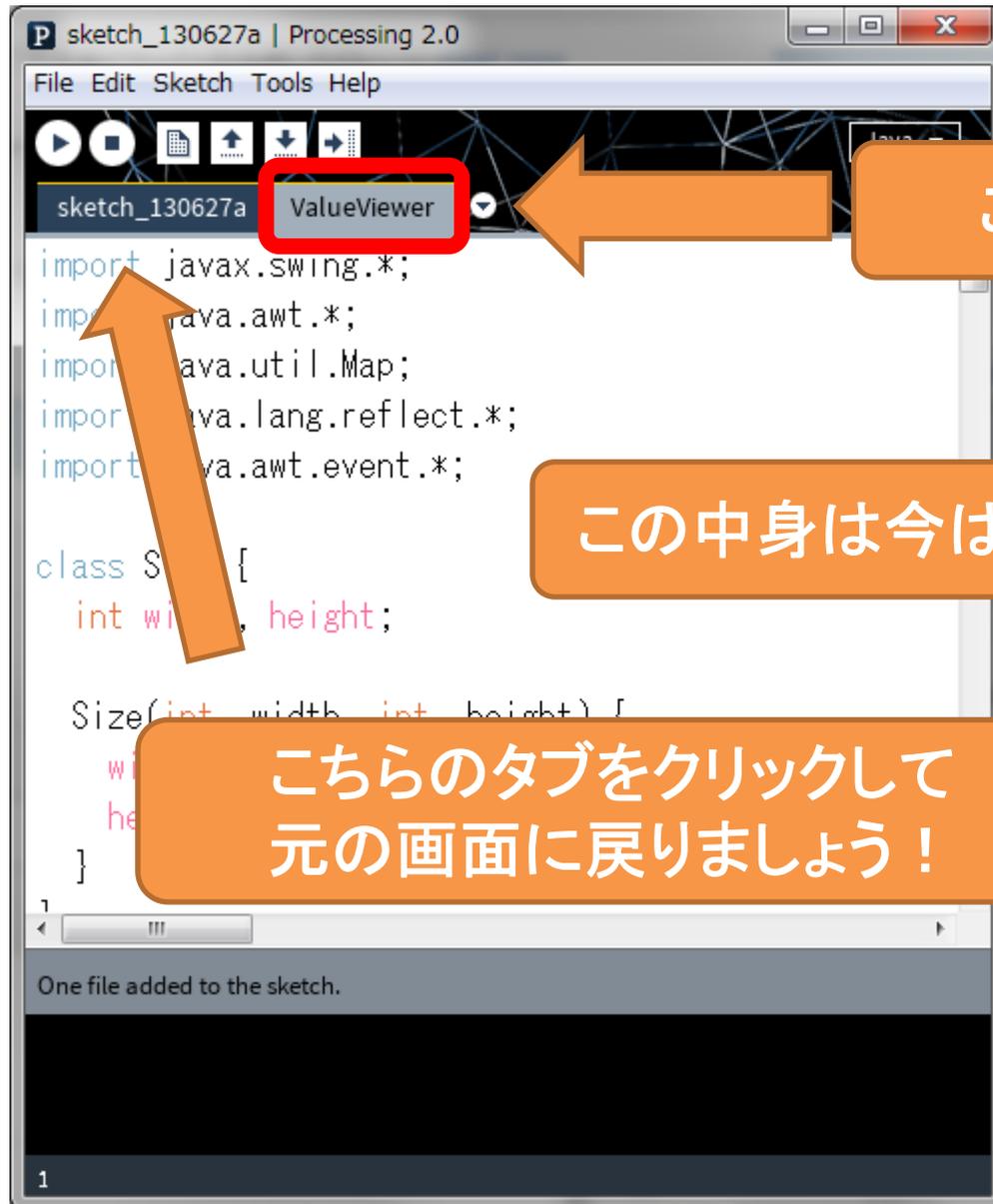
ドラッグ & ドロップ！

```
ValueViewer viewer;  
  
void setup(){  
  size( 600, 400 );  
  viewer = new ValueViewer(this);  
}  
  
void draw(){  
}
```

名前	更新日時	種類	サイズ
example	2013/06/26 14:43	ファイルフォル...	
compressor.sh	2013/06/25 23:24	SH ファイル	1 KB
example.pde	2013/06/25 23:24	PDE ファイル	1 KB
README.md	2013/06/25 23:24	MD ファイル	2 KB
ValueViewer.pde	2013/06/25 23:24	PDE ファイル	12 KB

wValueを、別ウィンドウで表示&変数の自動挙動

登録された！



こんな感じで追加されます

この中身は今は気にしなくてOK!

こちらのタブをクリックして
元の画面に戻りましょう！

実行しよう！



Processing 2.0 window showing code and a visual output:

```
ValueViewer viewer;  
  
float [] posX = new float [10];  
float [] speed = new float [10];  
void setup(){  
  size( 400, 300 );  
  viewer = ValueViewer viewer;  
  int i=0;  
  while( i < 10 ){  
    posX[i] = random( 400 );  
    speed[i] = random( 100 );  
    i++;  
  }  
}
```

The visual output shows a green dot at the top center and several red dots at the bottom, representing the initial state of the simulation.

	2372601307809332500674924584
speed	35400589312721063934842774

必要に応じて



- 横に伸ばしたり
- 縦に伸ばしたりしましょう

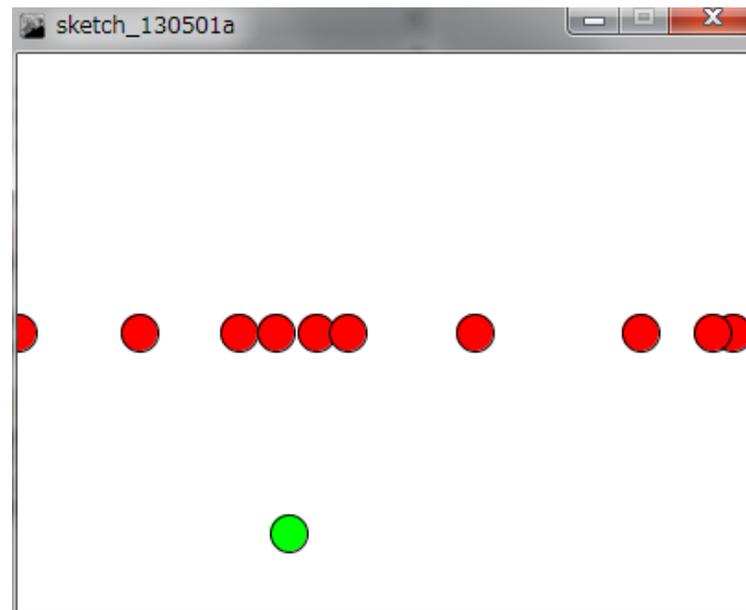
	330.2546	224.24352	80.610115	369.84348	230.6277	289.26904	273.03964	144.79568	66.84259	145.3119
speed	3.540584	2.0859404	0.32372087	1.8700641	2.3389134	3.1254637	4.494454	0.3278184	0.6399977	2.860970

- 値が増加しているとき, 赤色になります
- 値が減少している時, 青色になります
- 画面クリックで停止します

沢山の情報を扱うには？



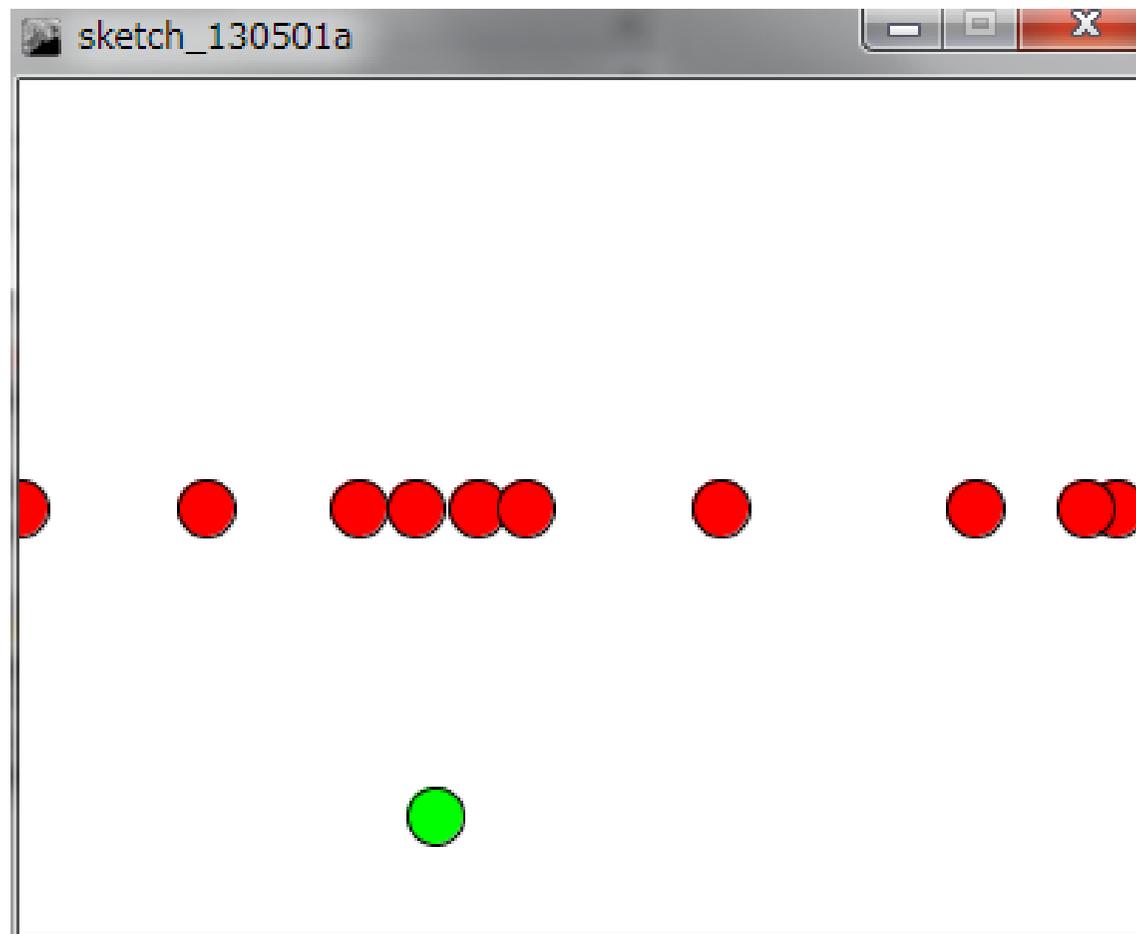
- トランプのカードや色々なスコアの管理
- 沢山の敵を同時に動かしたい
 - 沢山の敵の場所を管理するにはどうするか？
 - 沢山の円が一度に動き, その円をマウスで避けるような簡単なゲームを作ってみよう！



円を沢山描画する



(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



- 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

```
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);  
}
```

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？

```
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 );  
}
```

配列



- 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

float [] posX = new float [10];

変数の型 [] 変数名 = new 変数の型 [要素数];

- 変数への代入 **posX[5] = 20.5;**
- 変数の値の出力 **println(posX[5]);**

配列の注意点



- 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

配列の定義



```
int[] score = new int [5];
```

- 整数型で要素数が5個の score という配列を作成
- score の中に小箱が5つできているイメージ



円を沢山描画する



- 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 );  
}
```

円を沢山描画する



- 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);  
}
```

あまり楽になってないのでは？
100個ならどっちにしる大変

```
void draw(){  
  background( 255 );  
  fill( 0, 255, 0 );  
  ellipse( mouseX, mouseY, 20, 20 );  
  fill( 255, 0, 0 );  
  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



- 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)



- 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！

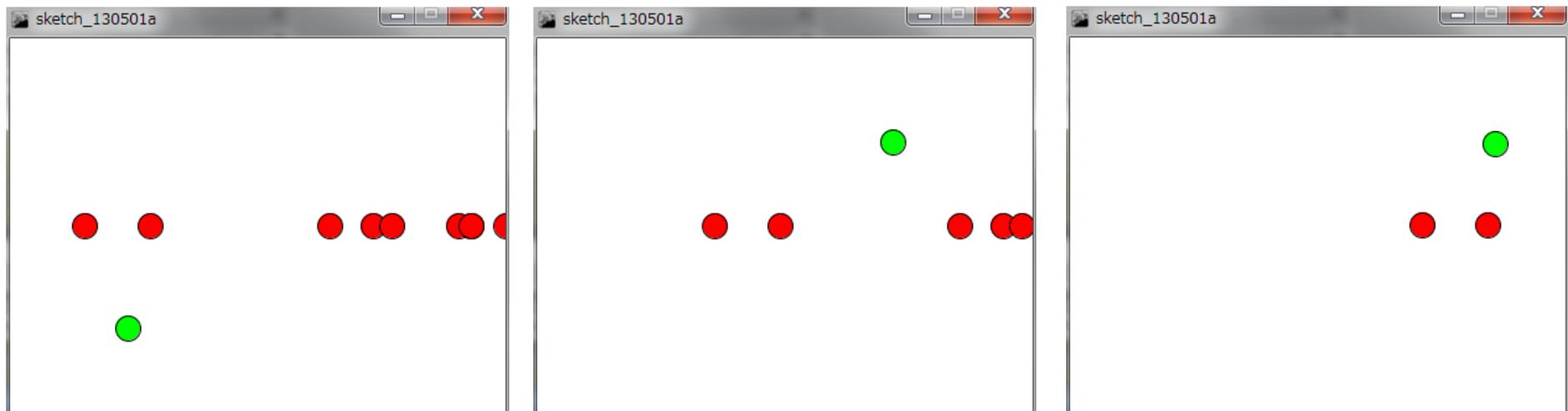
```
float [] posX = new float [10];  
void setup(){  
    size( 400, 300 );  
    for( int i=0; i<10; i++ ){  
        posX[i] = random(400);  
    }  
}
```

```
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



(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



- 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



(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



(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

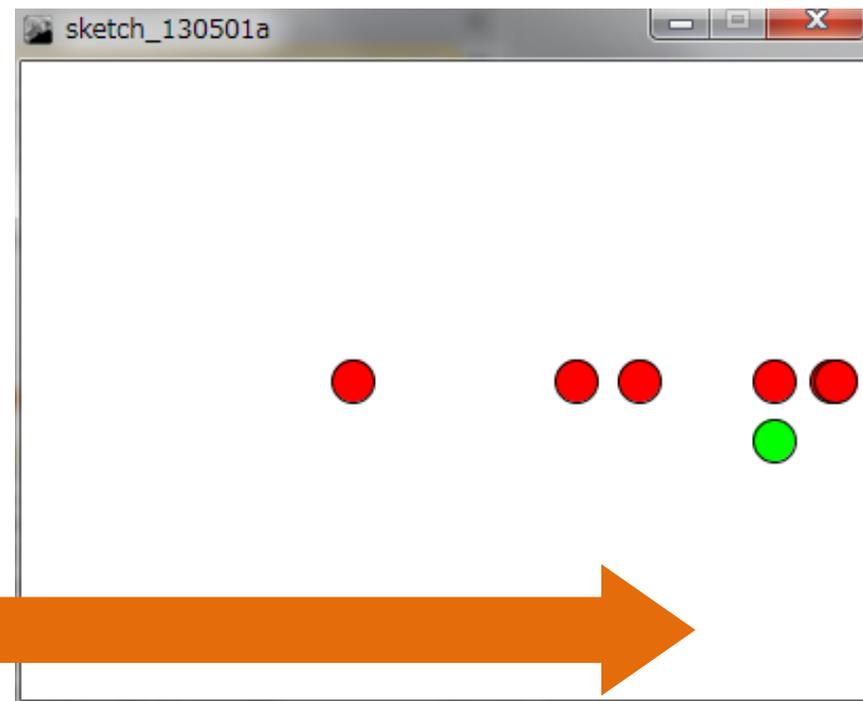
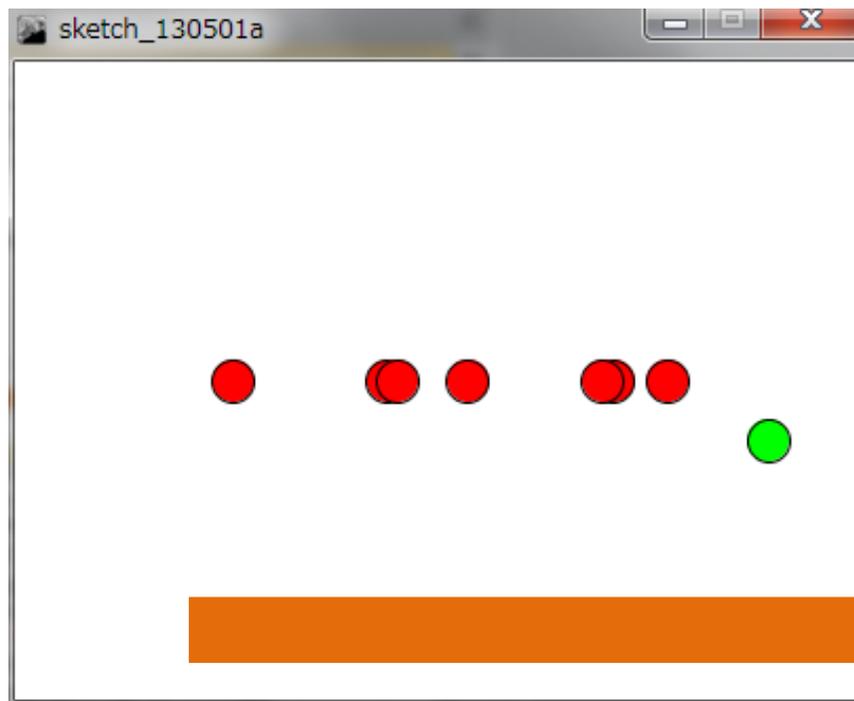
```
ValueViewer viewer;  
float [] posX = new float [10];  
void setup(){  
    size( 400, 300 );  
    viewer = new ValueViewer(this);  
    int i=0;  
    while( i<10 ){  
        posX[i] = random(400);  
        i++;  
    }  
}
```

```
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



(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



- 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



- 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！

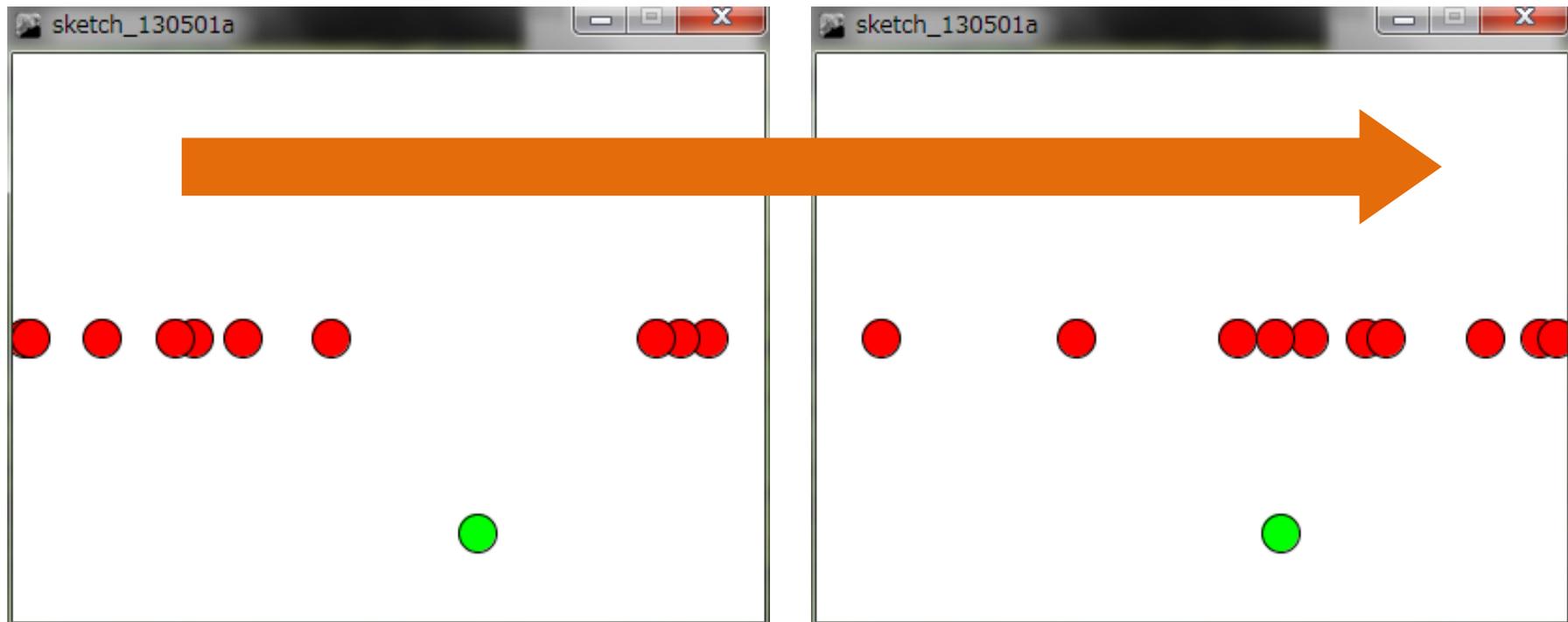
```
ValueViewer viewer;
float [] posX = new float [10];
float [] speed = new float [10];
void setup(){
  size( 400, 300 );
  viewer = new ValueViewer(this);
  int i=0;
  while( i<10 ){
    posX[i] = random(400);
    speed[i] = random( 5 );
    i++;
  }
}
```

```
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



(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



- 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



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

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

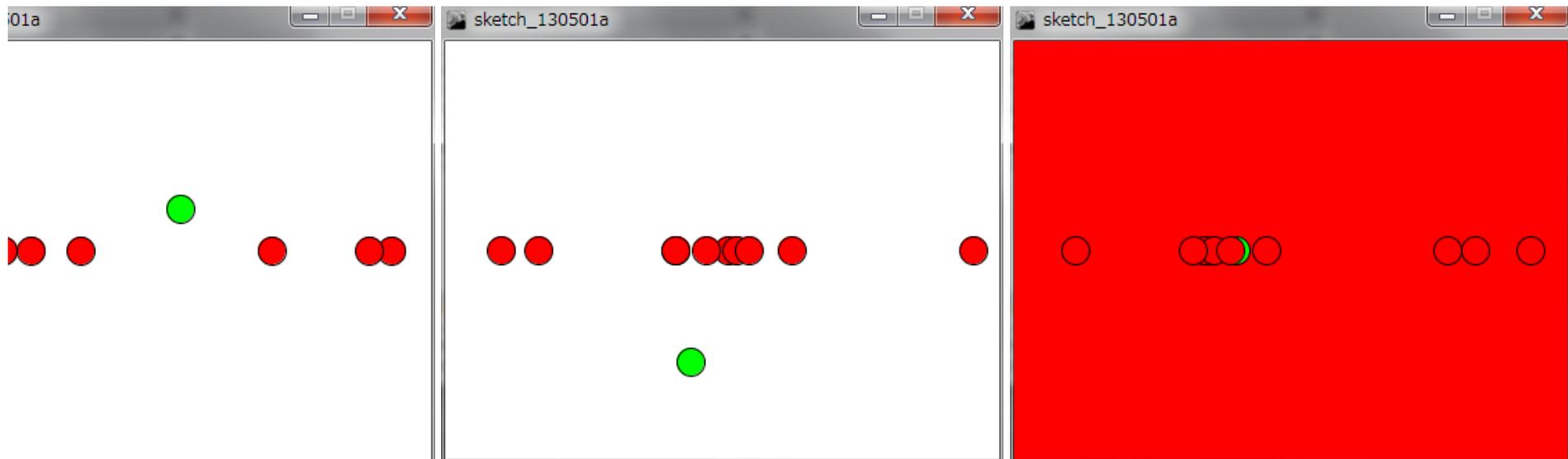
```
void draw(){
    background( 255 );
    fill( 0, 255, 0 );
    ellipse( mouseX, mouseY, 20, 20 );
    fill( 255, 0, 0 );
    int j = 0;
    while( j<10 ){
        posX[j] = posX[j] + speed[j];
        if( posX[j] > 400 ){
            posX[j] = posX[j] - 400;
        }
        ellipse( posX[j], 150,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



- 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

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

void setup(){
  size( 400, 300 );
  viewer = new ValueViewer(th
  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

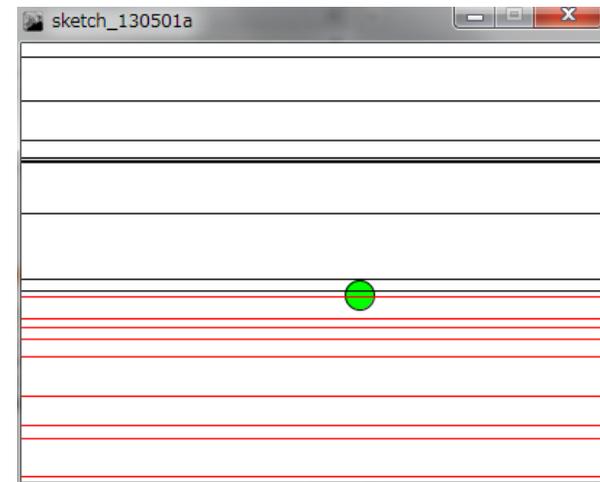
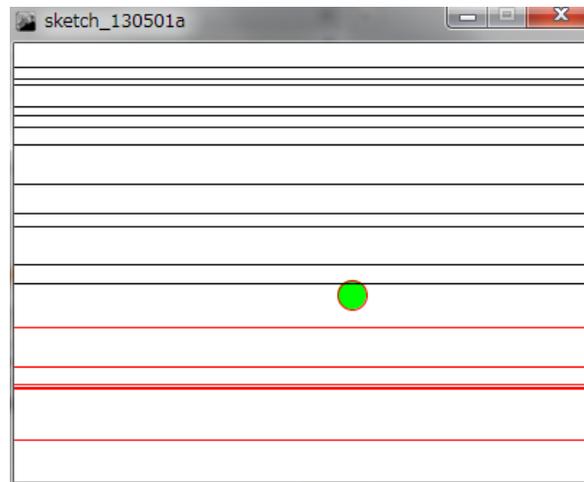
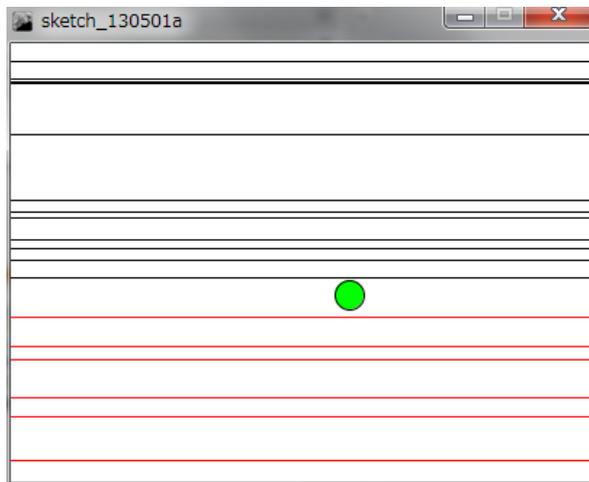


- 一斉に動く円の数を, 10個から20個に増やしてみましよう
- 一斉に動く円の数を, 10個から100個に増やしてみましよう
- 一斉に動く円を左方向にしてみましよう
 - speed をマイナスにしたらOK!
- 一斉に動く円を, 右方向にも左方向にも動くようにしてみましよう
 - speed を-5から5までにしたらOK!

線を一齐に動かす



(Q) 400x300の画面上に、横線を20本ランダムな場所に引き、その線を上から下に動かす。なお、下端に来ると上端から出てくるようにするには？



線を一齐に動かす



- 考え方

- 要素数が20個で, Y座標を格納する配列を用意する
 - `float [] posY = new float [20];`
- Y座標の値は0~300なので, `random(300);` を利用して繰り返しで20個の配列に値を代入
- 1回描画する度にそれぞれの配列のY座標を増やす
 - 横線は(0, `posY[j]`)から(400, `posY[j]`)まで描画する
- それぞれの配列のY座標が下端 ($Y > 300$) の時に, Yの座標を0に戻す(または, 300マイナスする)

線を一齐に動かす



- `float [] posY = new float [20];` で20個の要素を持つ配列を作成し, 値を代入, `line` で線を引く

```
float [] posY = new float [20];
```

```
void setup(){  
  size( 400, 300 );  
  int i=0;  
  while( i<20 ){  
    posY[i] = random(300);  
    i++;  
  }  
}
```

```
void draw(){  
  background( 255 );  
  int j = 0;  
  while( j<20 ){  
    posY[j] = posY[j] + 1;  
    if( posY[j] > 300 ){  
      posY[j] = posY[j] - 300;  
    }  
    line( 0, posY[j], 400, posY[j] );  
    j++;  
  }  
}
```

予習問題



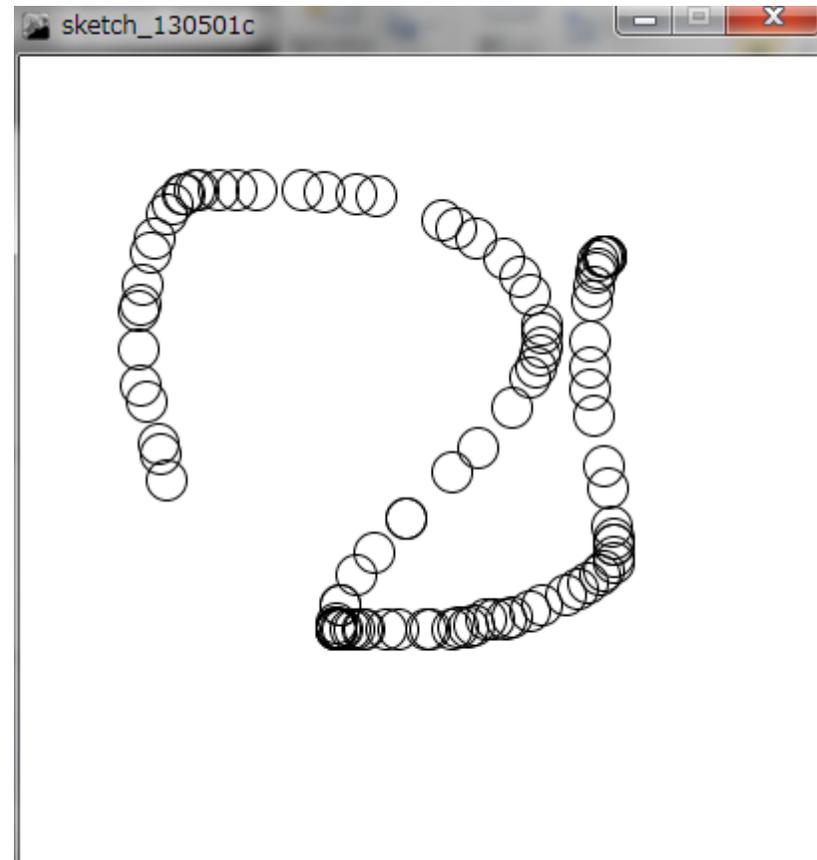
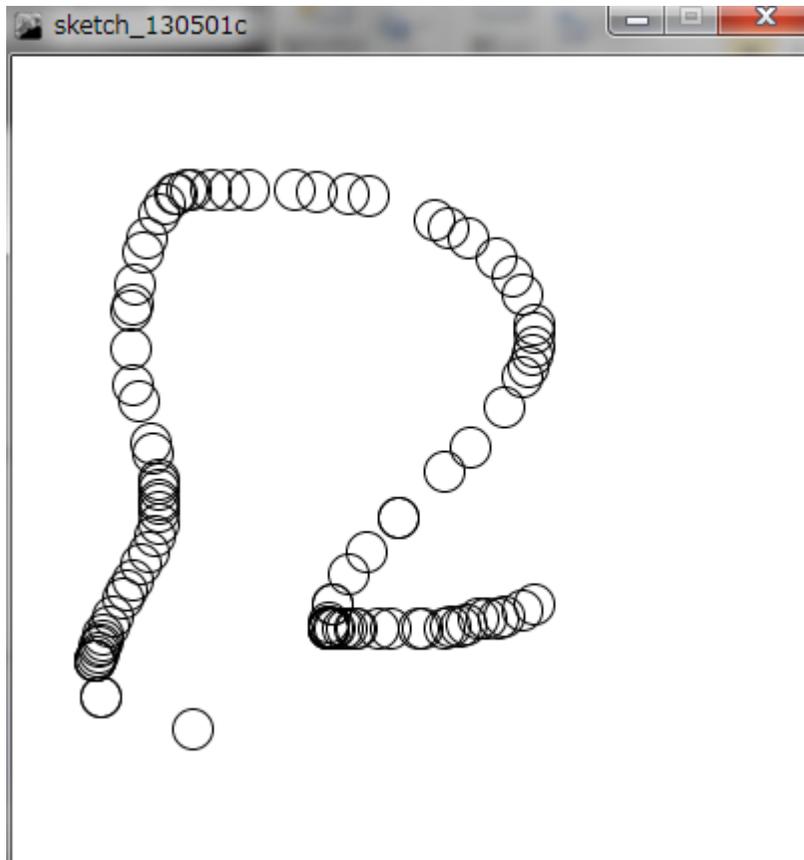
- 400x300の画面上に表示される20本の線が、それぞれ独自のスピードをもって下に落ちていくように変更せよ
 - speed 配列の導入
- 400x300の画面上に表示され動いている20本の線が、マウスカーソルより下にある場合は赤色、上にある場合は黒色で描かれるようにせよ
 - 場所を判定して位置に応じて stroke で色を変更

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



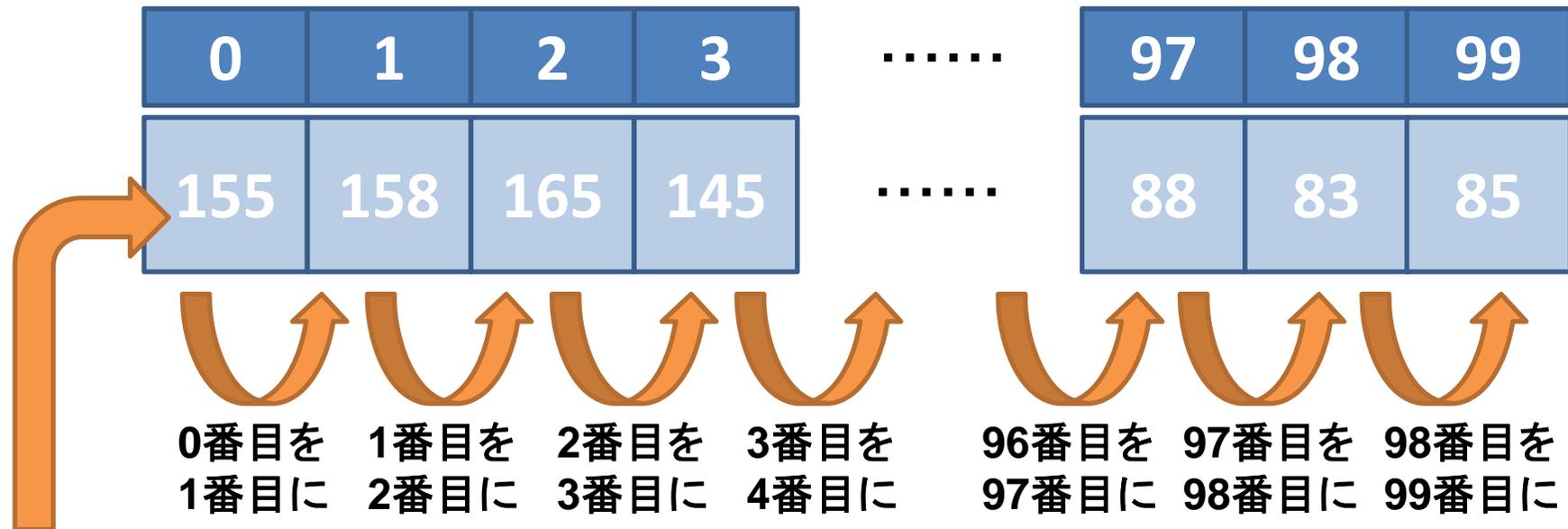
- 考え方

- 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



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



- 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  
}
```

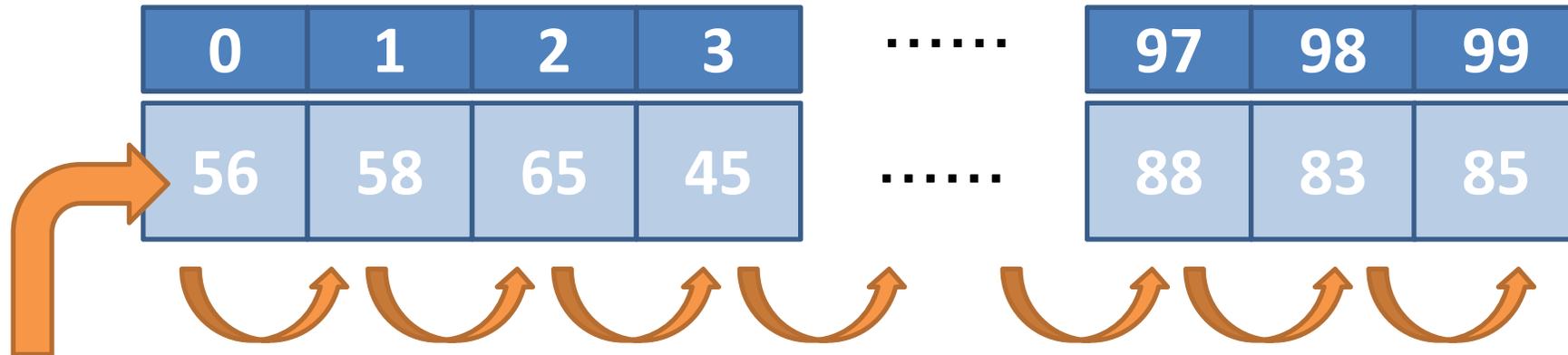
何が間違っているか？

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

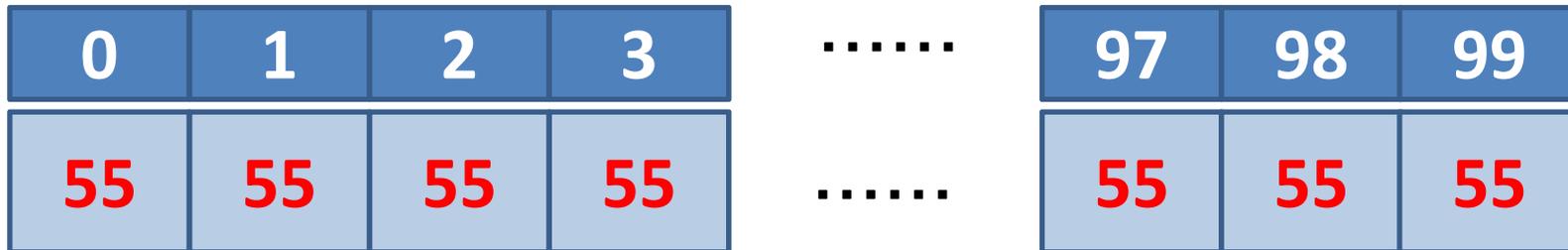


cursorX (処理前)



mouseX (55)

cursorX (処理後)



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



- 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
:

A

```
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++;
  }
}
```



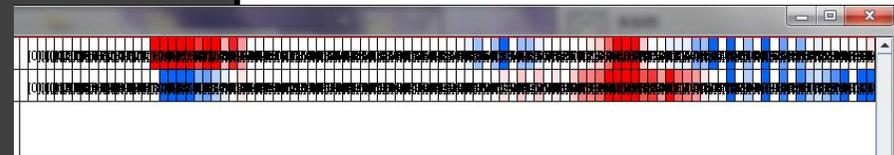
配列の内容を
1つずつずらす

cursorX[i], cursorY[i] に
円を描画

A

```
ValueViewer viewer;  
int [] cursorX = new int [100];  
int [] cursorY = new int [100];  
void setup(){  
    size( 400, 400 );  
    viewer = new ValueViewer(this);  
    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++;  
    }  
}
```

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



配列の値がどう動くか？



- 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よう！（マウスのX座標が順に引き継がれます）

0	1	2	3	4	5	6	7	8	9
524	524	529	552	642	715	679	560	488	446

0	1	2	3	4	5	6	7	8	9
79	94	136	160	162	300	524	524	524	524

配列の値がどう動くか？



```
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

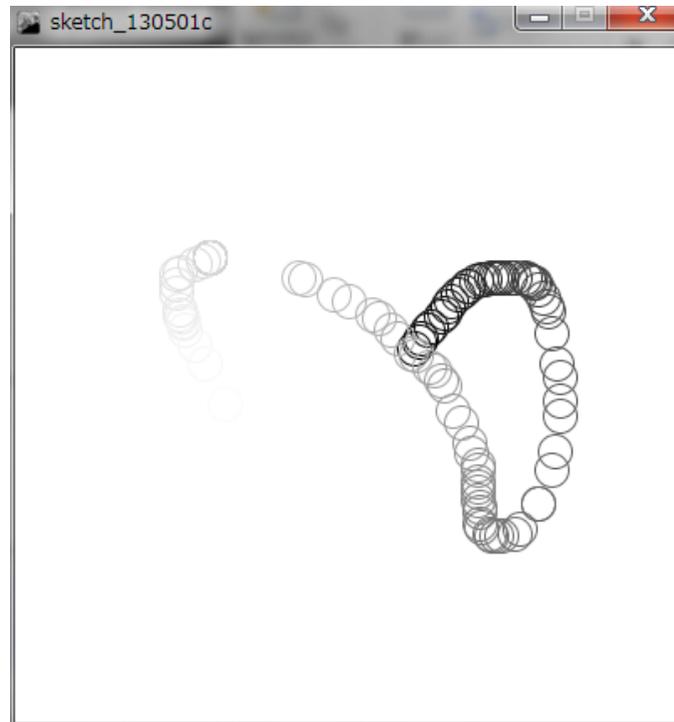
```
int [] arrayX = new int [10];  
void draw() {  
  background(0);  
  for ( int i=9; i>0; i-- ) {  
    arrayX[i] = arrayX[i-1];  
  }  
  arrayX[0] = mouseX;  
  for ( int i=0; i<10; i++ ) {  
    fill(0);  
    rect( i*80, 0, (i+1)*80, 50 );  
    rect( i*80, 50, (i+1)*80, 70 );  
    fill(255);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

予習問題



- 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) `stroke(255 * i / 100);`



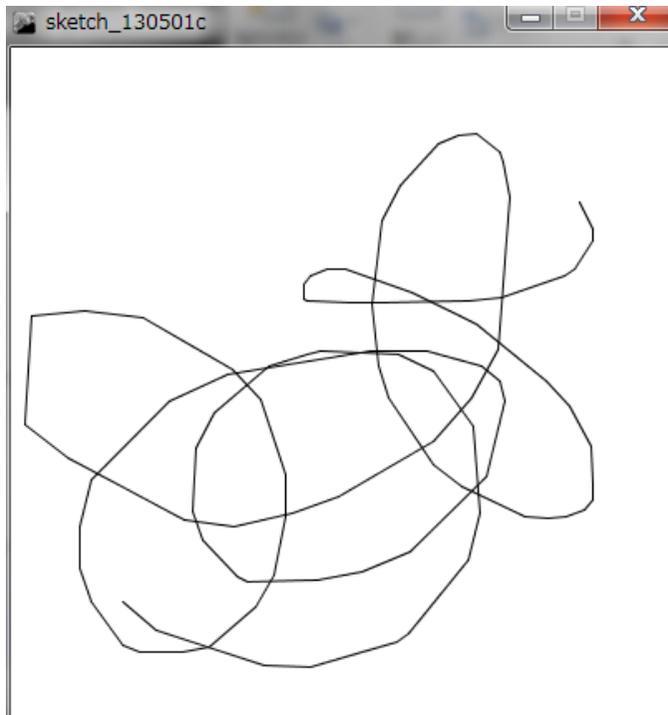
予習問題



- 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



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



- 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 [10] Pimage;
- 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;
```

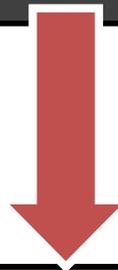
```
void setup(){  
  size( 800, 600 );  
  parapara[0]=loadImage("gazo0.jpg");  
  parapara[1]=loadImage("gazo1.jpg");  
  parapara[2]=loadImage("gazo2.jpg");  
  parapara[3]=loadImage("gazo3.jpg");  
  parapara[4]=loadImage("gazo4.jpg");  
  parapara[5]=loadImage("gazo5.jpg");  
  parapara[6]=loadImage("gazo6.jpg");  
  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



```
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");
```

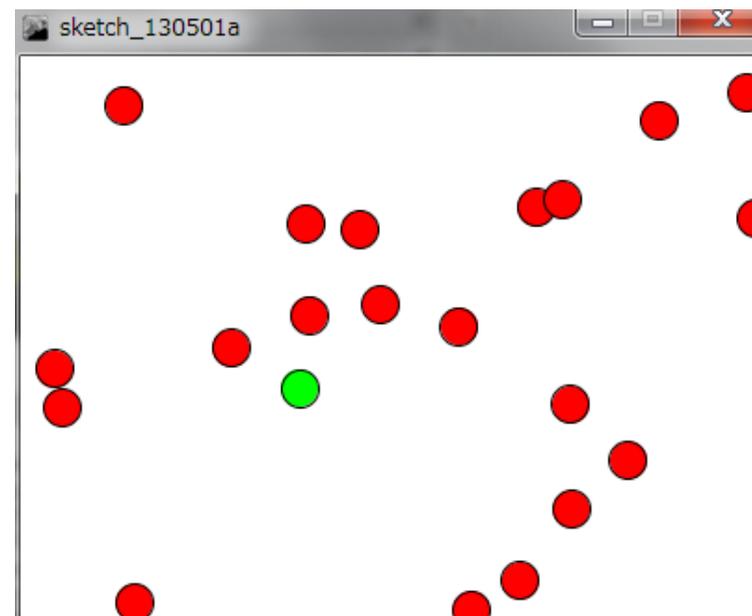
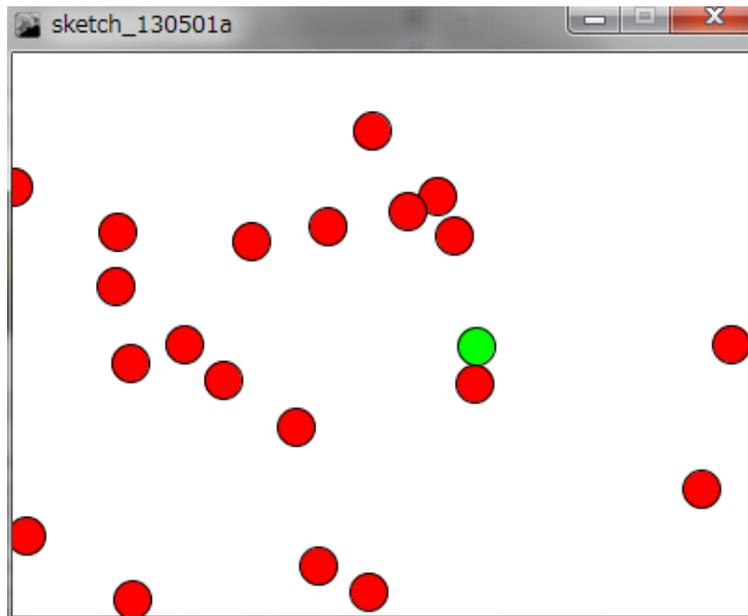


```
for( int j=0; j<10; j++ ){  
    parapara[j]=loadImage("gazo"+j+".jpg");  
}
```

予習問題



- 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



- 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！