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 (9)
多重配列

中村，高橋
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 で多重配列に挑戦！
  - 2次元のマス目に配置されたオブジェクトをどう扱っていくか？

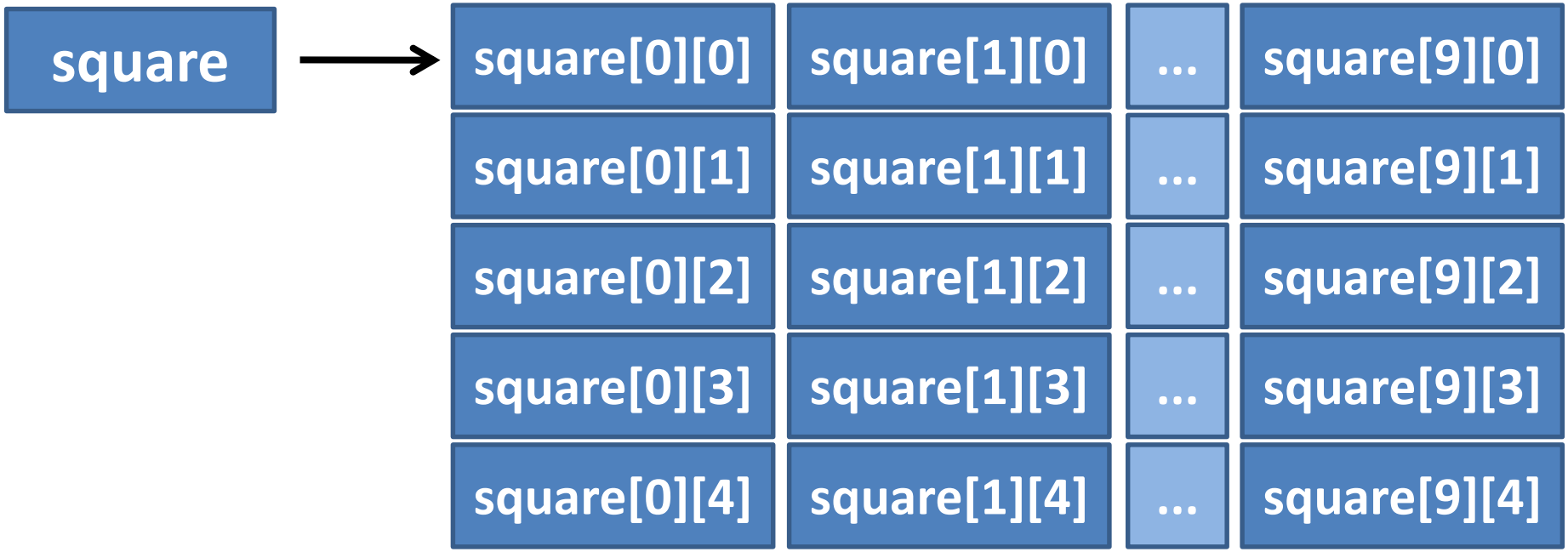
- 課題:
  - オセロゲームを作ってみる
  - ライツアウトを作ってみよう

2次元配列の定義

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[][] square = new int [10][5];
```

- 整数型で要素数が10x5個の square という配列を作成
- square の中に小箱が10x5できているイメージ

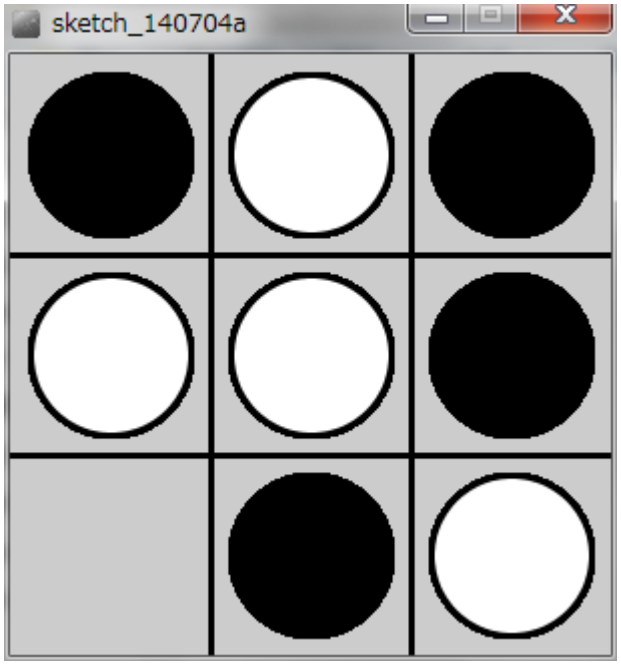


Tic Tac Toe

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

- Tic Tac Toe（マルバツゲーム）を作ってみよう！
  - ○と×の変わりに，白丸と黒丸で実現しよう！
  - クリックする度に**【何も置かれていない→白が置かれている→黒が置かれている】**と切り替える
  - どうやってマス目に何が置かれているかを管理する？
  - square という配列を用意し
    - 何も置かれていないか
    - 黒が置かれているか
    - 白が置かれているか

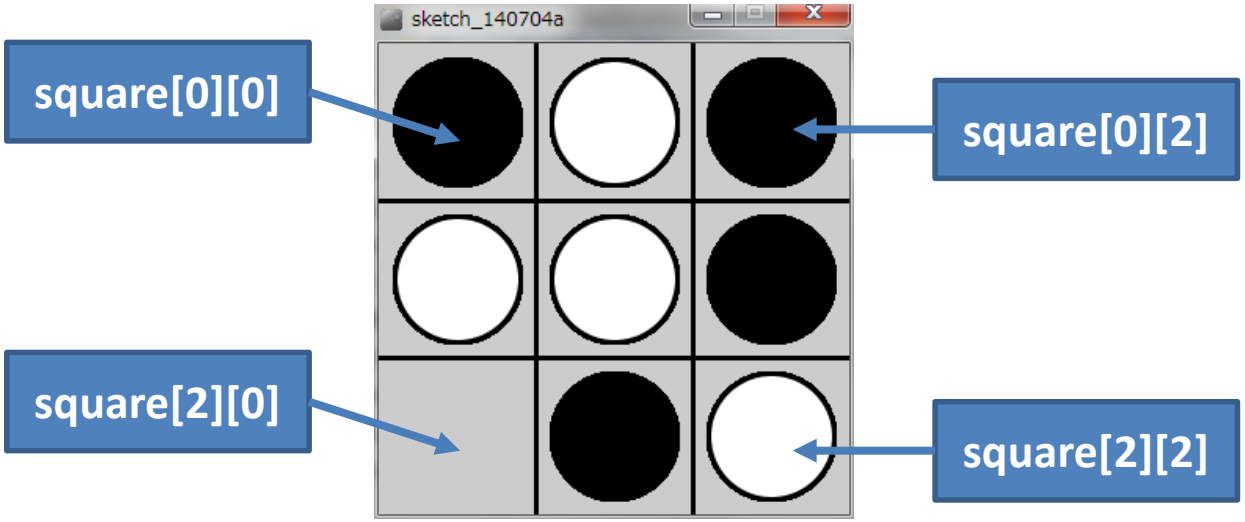
      を管理する



どう考えるか？

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

- まずはマス目を描画してみよう
  - line を4本引くことで描画することが可能
- 次に，1つずつのマス目を配列で定義しよう
  - square[3][3] という配列で管理する！
  - 最初に配列を0で初期化し，配列の値が0なら空白，1なら白色の丸，2なら黒色の丸を描画する.



まず初期化！

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

- 2次元配列の定義

変数の型 [][] 変数名 = new 変数の型 [要素数][要素数];

while

```
int  [][]  square = new int [3][3];

void setup(){
    size( 300, 300 );
    int x=0;
    while( x<3 ){
        int y=0;
        while( y<3 ){
            square[x][y] = 0;
            y++;
        }
        x++;
    }
}
```

3x3の2次元配列の定義

多重ループで0に初期化

for

```
int  [][]  square = new int [3][3];

void setup(){
    size( 300, 300 );
    for( int x=0; x<3; x++ ){
        for( int y=0; y<3; y++ ){
            square[x][y] = 0;
        }
    }
}
```

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

- square[x][y] の値に応じて描画
  - 多重繰り返しで描画する（線の描画は省略）

while

```
void draw(){
    background( 255 );
    // ここで線を描画しておく
    int x=0;
    while( x<3 ){
        int y=0;
        while( y<3 ){
            if( square[x][y] == 0 ){
                // 何も描画しない
            } else if( square[x][y] == 1 ){
                // 白丸を描画する
            } else if( square[x][y] == 2 ){
                // 黒丸を描画する
            }
            y++;
        }
        x++;
    }
}
```

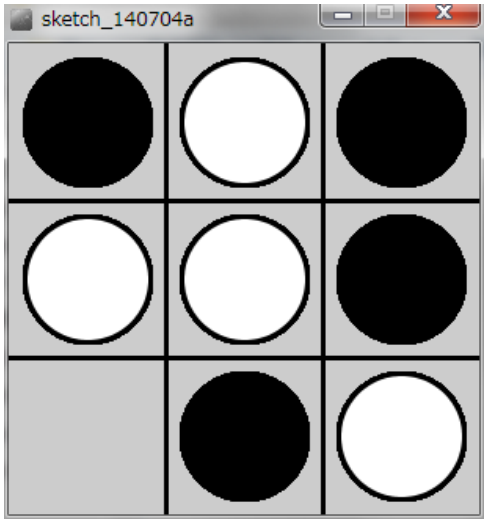
for

```
void draw(){
    background( 255 );
    // ここで線を描画しておく
    for( int x=0; x<3; x++ ){
        for( int y=0; y<3; y++ ){
            if( square[x][y] == 0 ){
                // 何も描画しない
            } else if( square[x][y] == 1 ){
                // 白丸を描画する
            } else if( square[x][y] == 2 ){
                // 黒丸を描画する
            }
        }
    }
}
```

マウスクリックでの変化

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

- マウスクリックされた時に，そのクリックされた座標に応じて，どの配列の値を変更するかを考える！
  - 1マスが100ピクセルなので，mouseX÷100や，mouseY÷100の値（値を切り捨てたもの）が配列の添え字（[]の中で指定する値）となる

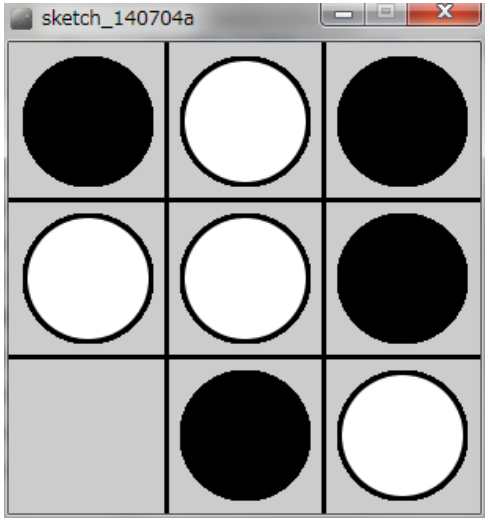



```
void mousePressed(){
    // どこのマス目をクリックしたか
    // マス目の大きさが100なので…
    int tx = mouseX / 100;
    int ty = mouseY / 100;
    square[tx][ty]++;
    if( square[tx][ty] > 2 ){
        square[tx][ty] = 0;
    }
}
```

Tic Tac Toe 改

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

- Tic Tac Toe（マルバツゲーム）を作ってみよう！
  - ○と×の変わりに，白丸と黒丸で実現しよう！
  - クリックするだけで白丸の後は黒丸，黒丸の後は白丸を自動でおけるようにせよ
  - 考え方
    - 初期化とかは問題ない
    - ターンという考え方を導入し，ターンに応じて置くコマを切り替える
    - Tic Tac Toe のmousePressedだけを変更！



Tic Tac Toe 改

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

- turn が0ならsquare[tx][ty]には1を，turnが1ならsquare[tx][ty]には2をセットする！
- クリックの度にturnの値を変化させる！

while

```
int  [][]  square = new int [3][3];
int turn = 0;
void setup(){
    size( 300, 300 );
    int x=0;
    while( x<3 ){
        int y=0;
        while( y<3 ){
            square[x][y] = 0;
            y++;
        }
        x++;
    }
}
```

for

```
void mousePressed(){
    int tx = mouseX / 100;
    int ty = mouseY / 100;
    if( turn == 0 ){
        square[tx][ty] = 1;
    } else {
        square[tx][ty] = 2;
    }
    turn++;
    if( turn > 1 ){
        turn = 0;
    }
}
```

もっと簡単にできないか？

Tic Tac Toe 改

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

- turn を1と2が変化する変数にする
  - turnが1ならsquare[tx][ty]には1を，turnが2ならsquare[tx][ty]には2をセットする！ つまりturnを代入するだけで良い！

while

```
int  [][]  square = new int [3][3];
int turn = 1;
void setup(){
    size( 300, 300 );
    int x=0;
    while( x<3 ){
        int y=0;
        while( y<3 ){
            square[x][y] = 0;
            y++;
        }
        x++;
    }
}
```

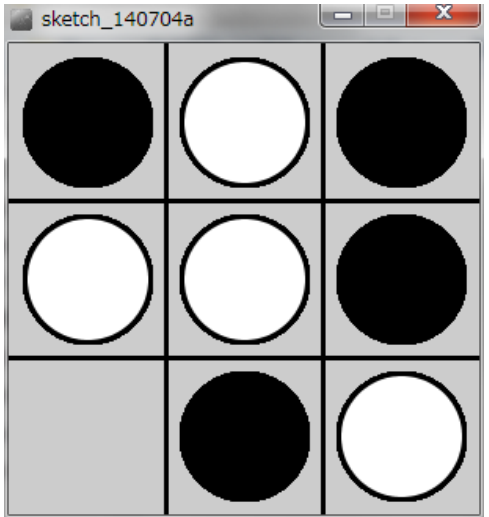
for

```
void mousePressed(){
    int tx = mouseX / 100;
    int ty = mouseY / 100;
    square[tx][ty] = turn;

    turn++;
    if( turn > 2 ){
        turn = 1;
    }
}
```

Tic Tac Toe 改二

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

- Tic Tac Toe（マルバツゲーム）を作ってみよう！
  - ○と×の変わりに，白丸と黒丸で実現しよう！
  - クリックするだけで白丸の後は黒丸，黒丸の後は白丸を自動でおけるようにせよ
  - 他のコマが置かれている場所には置けないようにせよ
  - 考え方
    - Tic Tac Toe改をさらに改良するだけで良さそう！
    - 置こうとしている場所に，すでにコマが置かれていたら（値が1以上だったら）置けないようにする！



Tic Tac Toe 改二

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

- そこに置かれているかどうかをチェックする！
  - 0なら置ける！ 1か2なら置けない！

```
void mousePressed(){
    int tx = mouseX / 100;
    int ty = mouseY / 100;
    if( square[tx][ty] == 0 ){
        square[tx][ty] = turn;
        turn++;
        if( turn > 2 ){
            turn = 1;
        }
    } else {
        // すでに置かれている場合は
        // なにもしないけど置けません！
        // と表示してもよい
    }
}
```

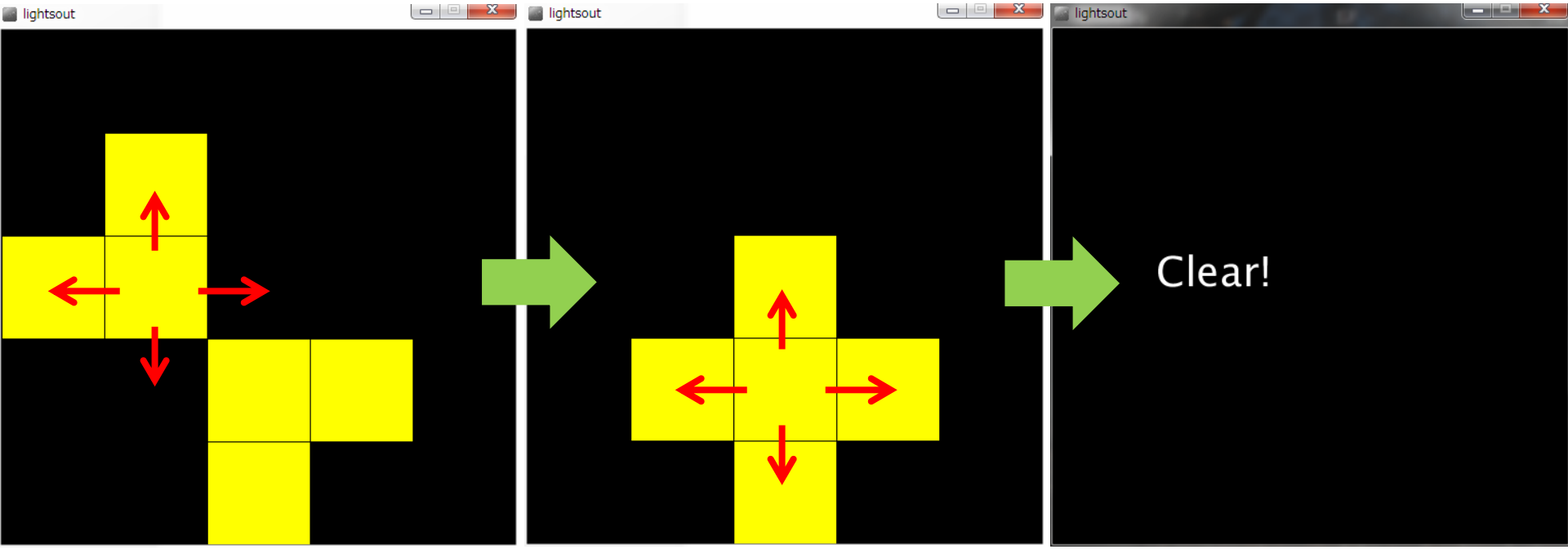
たったこれだけ！

勝敗判定をどうするかは
もう少し学んでから

[演習] Lights Out

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

- **横5マス、縦5マス** の盤面を作り、そのマス目上をクリックするとクリックされたマス目の上下左右とそのマス目自体の色を反転させるLights Outを作る
  - まずは，クリックすると上下左右反転させるものを目指す



Lights Out

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
  - 5x5のマス目にそれぞれ黄色または黒色の正方形を描画する
    - 5x5の配列を作る
    - 配列の値が0の場合は黄色，1の場合は黒色とする
  - マウスクリックされた座標がどのマス目に該当するかを計算する
    - そのマス目および，上下左右のマス目の値を反転させる
    - 0と1の値の反転は【square = 1 – square】を使うと簡単！だけど，もちろんif文で書いても良い

まず初期化！

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

- 2次元配列の定義

変数の型 [][] 変数名 = new 変数の型 [要素数][要素数];

while

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

void setup(){
    size( 500, 500 );
    int x=0;
    while( x<5 ){
        int y=0;
        while( y<5 ){
            square[x][y] = 0;
            y++;
        }
        x++;
    }
}
```

5x5の2次元配列の定義

for

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

void setup(){
    size( 500, 500 );
    for( int x=0; x<5; x++ ){
        for( int y=0; y<5; y++ ){
            square[x][y] = 0;
        }
    }
}
```

多重ループで0に初期化

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

- square[x][y] の値に応じて描画
  - 多重繰り返しで描画する

while

```
void draw(){
    background( 255 );

    int x=0;
    while( x<5 ){
        int y=0;
        while( y<5 ){
            if( square[x][y] == 0 ){
                // 黄色四角形を描画
            } else if( square[x][y] == 1 ){
                // 黒色四角形を描画
            }
            y++;
        }
        x++;
    }
}
```

for

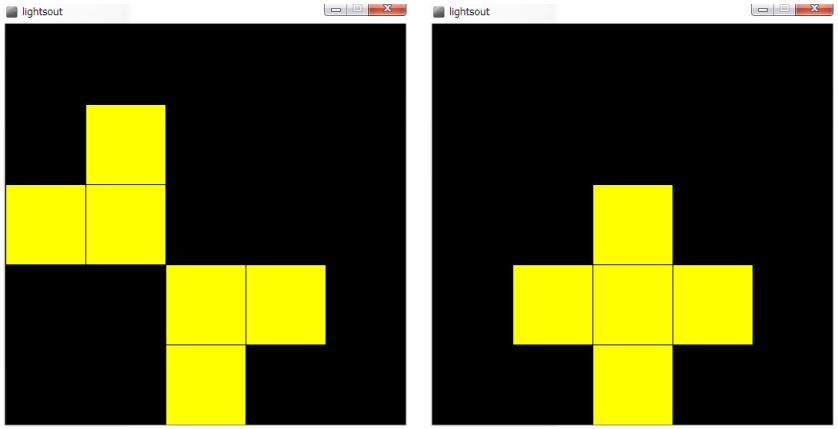
```
void draw(){
    background( 255 );

    for( int x=0; x<5; x++ ){
        for( int y=0; y<5; y++ ){
            if( square[x][y] == 0 ){
                // 黄色四角形を描画
            } else if( square[x][y] == 1 ){
                // 黒色四角形を描画
            }
        }
    }
}
```

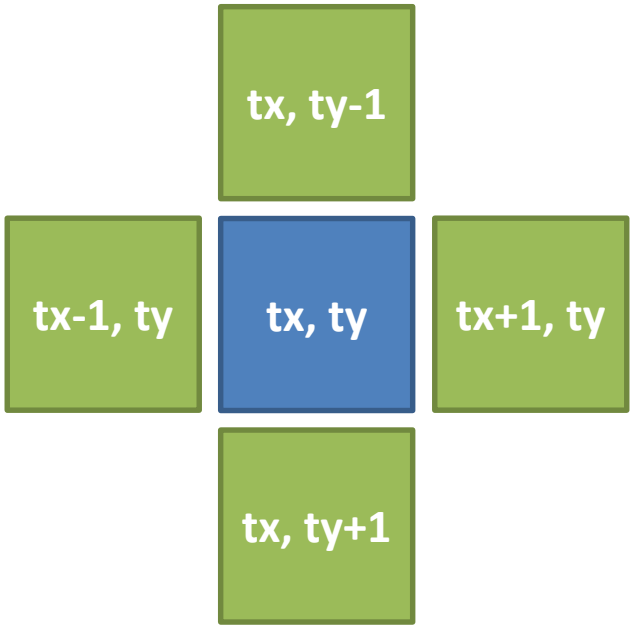
0か1かで描画内容を変更

マウスクリックでの変化

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

- マウスクリックされた時に，そのクリックされた座標に応じて，どの配列の値を変更するかを考える！
  - 1マスが100ピクセルなので，mouseX÷100や，mouseY÷100の値（値を切り捨てたもの）が配列の添え字（[]の中で指定する値）となる．また，その添え字の周辺が対象となる
  - tx, ty を求めて，そこの周辺を考える！





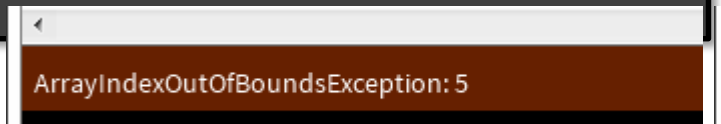
マウスクリックでの変化

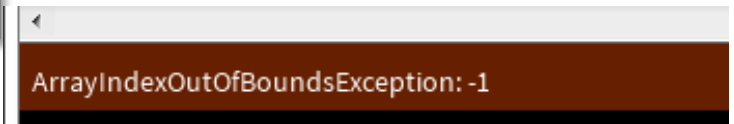
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- マウスクリックされた時に，そのクリックされた座標に応じて，どの配列の値を変更するかを考える！
  - 1マスが100ピクセルなので，mouseX÷100や，mouseY÷100の値（値を切り捨てたもの）が配列の添え字（[]の中で指定する値）となる．また，その添え字の周辺が対象となる

```
void mousePressed(){
    // どこのマス目をクリックしたか
    // マス目の大きさが100なので…
    int tx = mouseX / 100;
    int ty = mouseY / 100;
    square[tx][ty] = 1 - square[tx][ty];
    square[tx-1][ty] = 1 - square[tx-1][ty];
    square[tx+1][ty] = 1 - square[tx+1][ty];
    square[tx][ty-1] = 1 - square[tx][ty-1];
    square[tx][ty+1] = 1 - square[tx][ty+1];
}
```

求めた tx, ty の
周辺をそれぞれ0と1を反転させる
if 文で書いても良いがここでは
省略表記をしている



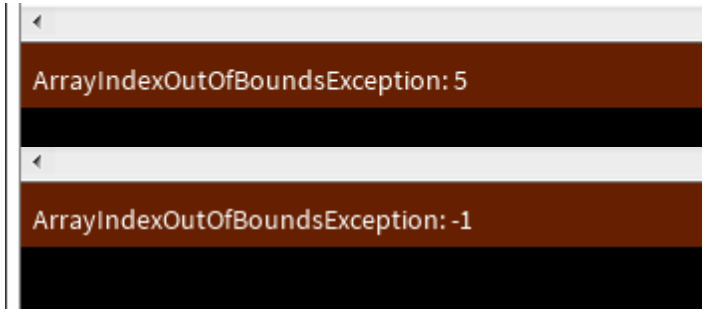


マウスクリックでの変化

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

- 上下左右を反転させようとするとエラーが出るのは配列の外を操作しようとしているから！
- つまり，tx-1やty-1，tx+1やty+1が，配列の外になっていないかをチェックする必要あり！
  - if 文で判定してチェック！

```
void mousePressed(){
  int tx = mouseX/100;
  int ty = mouseY/100;

  square[tx][ty] = 1 - square[tx][ty];
  if( tx-1 >= 0 ){
    square[tx-1][ty] = 1 - square[tx-1][ty];
  }
  if( tx+1 < 5 ){
    square[tx+1][ty] = 1 - square[tx+1][ty];
  }
  if( ty-1 >= 0 ){
    square[tx][ty-1] = 1 - square[tx][ty-1];
  }
  if( ty+1 < 5 ){
    square[tx][ty+1] = 1 - square[tx][ty+1];
  }
}
```

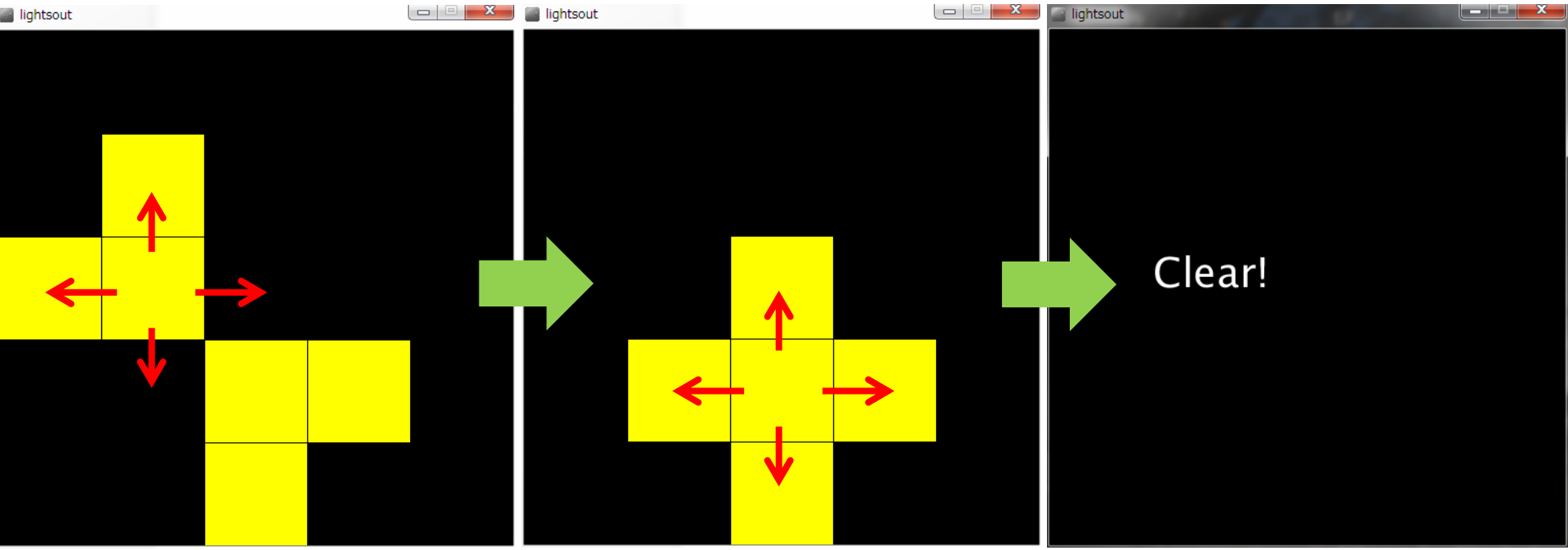


このままでもよいけれど
判定が少々面倒…

[演習] Lights Out 改

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

- **【横5マス、縦5マス】**の盤面を作り、そのマス目上をクリックするとクリックされたマス目の上下左右とそのマス目自体の色を反転させるLights Outを作る
- すべてが黒色になったらCLEAR!と表示する



考え方

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

- 黄色のマスの数を数えて，合計が1以上だったらクリアではなく，合計が0だったらクリアと表示したら良い！
- やり方としては例えば下記の2つ
  - mousePressedの時に数を数えて，その数が0だったらclearというフラグ変数を0から1にし，0の場合はゲームを表示し，1の場合は「Clear!」という画面を表示する
  - drawの度に黄色の数を数えて，その数が1以上だったらそのままゲームを表示し，0だったら「Clear!」という画面を表示する

clear変数で管理

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

- clearという変数を用意して，その変数の値を変化させる

while

```
int clear = 0;
void mousePressed(){
    // 反転とかもろもろの処理は省略

    int total=0;
    int x=0;
    while( x<5 ){
        int y=0;
        while( y<5 ){
            if( square[x][y] == 0 ){
                 total++;
            }
            y++;
        }
        x++;
    }
    if( total == 0 ){
         clear = 1;
    }
}
```

for

```
int clear = 0;
void mousePressed(){
    // 反転とかもろもろの処理は省略

    int total=0;
    for( int x=0; x<5; x++ ){
        for( int y=0; y<5; y++ ){
            if( square[x][y] == 0 ){
                 total++;
            }
        }
    }

    if( total == 0 ){
         clear = 1;
    }
}
```

clear変数で管理

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

- clear変数の値に応じて表示を切り替える

while

```
void draw(){
  background( 255 );

  if( clear == 0 ){
    int x=0;
    while( x<5 ){
      int y=0;
      while( y<5 ){
        if( square[x][y] == 0 ){
          // 黄色四角形を描画
        } else if( square[x][y] == 1 ){
          // 黒色四角形を描画
        }
        y++;
      }
     x++;
    }
  } else {
    text( "Clear!", 100, 200 );
  }
}
```

clearに応じて描画切り替え

for

```
void draw(){
  background( 255 );

  if( clear == 0 ){
    for( int x=0; x<5; x++ ){
      for( int y=0; y<5; y++ ){
        if( square[x][y] == 0 ){
          // 黄色四角形を描画
        } else if( square[x][y] == 1 ){
          // 黒色四角形を描画
        }
      }
    }
  } else {
    text( "Clear!", 100, 200 );
  }
}
```

drawに集約

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

while

```
void draw(){
  background( 255 );
  int total=0;
  int x=0;
  while( x<5 ){
    int y=0;
    while( y<5 ){
      if( square[x][y] == 0 ){
        total++;
      }
      y++;
    }
    x++;
  }

  if( total > 0 ){
    x=0;
    while( x<5 ){
      y=0;
      while( y<5 ){
        if( square[x][y] == 0 ){
          // 黄色四角形を描画
        } else if( square[x][y] == 1 ){
          // 黒色四角形を描画
        }
        y++;
      }
      x++;
    }
  } else {
    text( "Clear!", 100, 200 );
  }
}
```

数を数えて

数に応じて描画切り替え

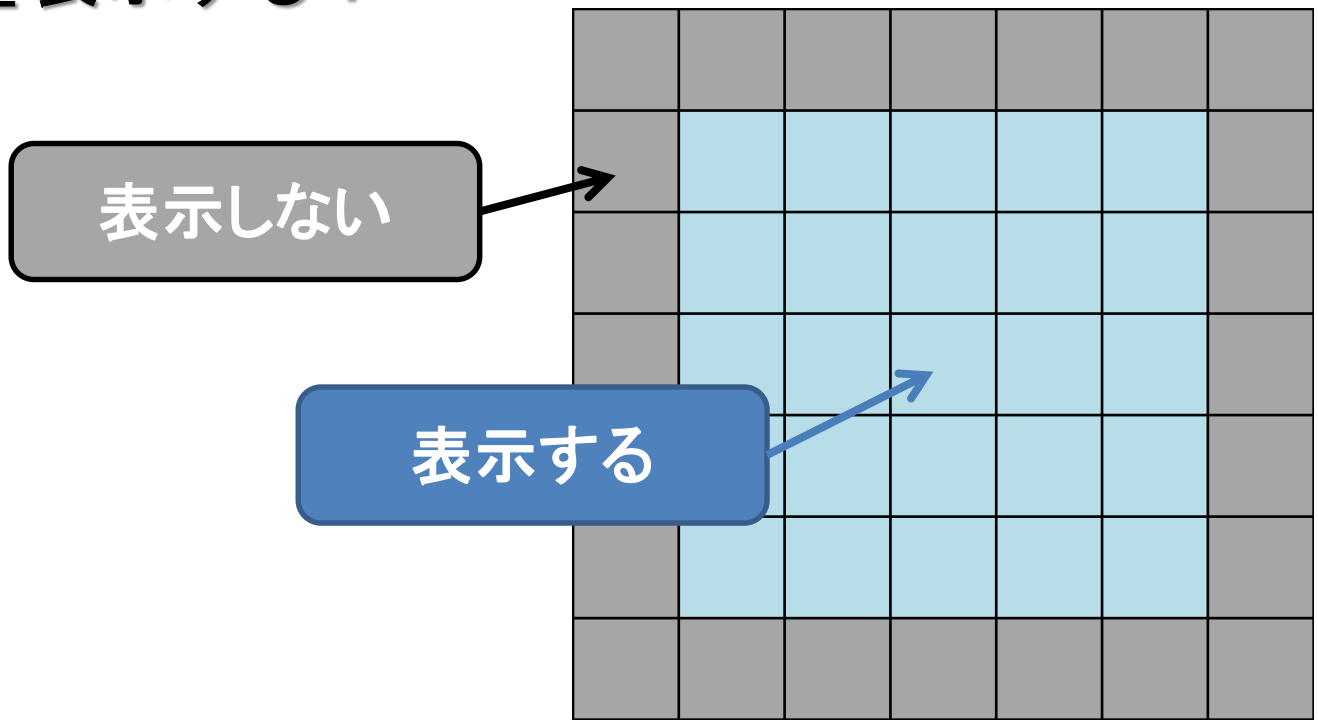
for

```
void draw(){
  background( 255 );
  int total=0;
  for( int x=0; x<5; x++ ){
    for( int y=0; y<5; y++ ){
      if( square[x][y] == 0 ){
        total++;
      }
    }
  }

  if( total > 0 ){
    for( int x=0; x<5; x++ ){
      for( int y=0; y<5; y++ ){
        if( square[x][y] == 0 ){
          // 黄色四角形を描画
        } else if( square[x][y] == 1 ){
          // 黒色四角形を描画
        }
      }
    }
  } else {
    text( "Clear!", 100, 200 );
  }
}
```

Lights Out を別の解法で

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

- 5x5のマス目からはみ出した処理が面倒！
- 配列を拡張して，7x7のマス目を作って，その一部を表示する！（配列の溢れを防ぐため）
- 左端右端，上端下端の部分は表示せず，そこを除いた部分だけを表示する！



初期化！

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

- 表示領域の外も0で初期化してしまう(どうせ表示しないからなんでも良いけれど)

while

```
int  [][]  square = new int [7][7];

void setup(){
  size( 500, 500 );
  int x=0;
  while( x<7 ){
    int y=0;
    while( y<7 ){
      square[x][y] = 0;
      y++;
    }
    x++;
  }
}
```

5x5の2次元配列の定義

多重ループで0に初期化

for

```
int  [][]  square = new int
[7][7];

void setup(){
  size( 500, 500 );
  for( int x=0; x<7; x++ ){
    for( int y=0; y<7; y++ ){
      square[x][y] = 0;
    }
  }
}
```

マウスクリックでの変化

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

- マウスクリックされた時に，そのクリックされた座標に応じて，どの配列の値を変更するかを考える！
  - 変更するマス目は，それぞれ+1したもの！
  - txとtyは1～5の値を取るので，-1や+1してもはみ出ない！

```
void mousePressed(){
  // どこのマス目をクリックしたか
  // マス目の大きさが100なので…
  int tx = mouseX / 100 + 1;
  int ty = mouseY / 100 + 1;
  square[tx][ty] = 1 - square[tx][ty];
  square[tx-1][ty] = 1 - square[tx-1][ty];
  square[tx+1][ty] = 1 - square[tx+1][ty];
  square[tx][ty-1] = 1 - square[tx][ty-1];
  square[tx][ty+1] = 1 - square[tx][ty+1];
}
```

描画するのはどこ？

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

- square[x][y] の値に応じて描画
  - 多重繰り返しで描画する

while

```
void draw(){
  background( 255 );

  int x=1;
  while( x<=5 ){
    int y=1;
    while( y<=5 ){
      if( square[x][y] == 0 ){
        // 黄色四角形を描画
      } else if( square[x][y] == 1 ){
        // 黒色四角形を描画
      }
      y++;
    }
    x++;
  }
}
```

for

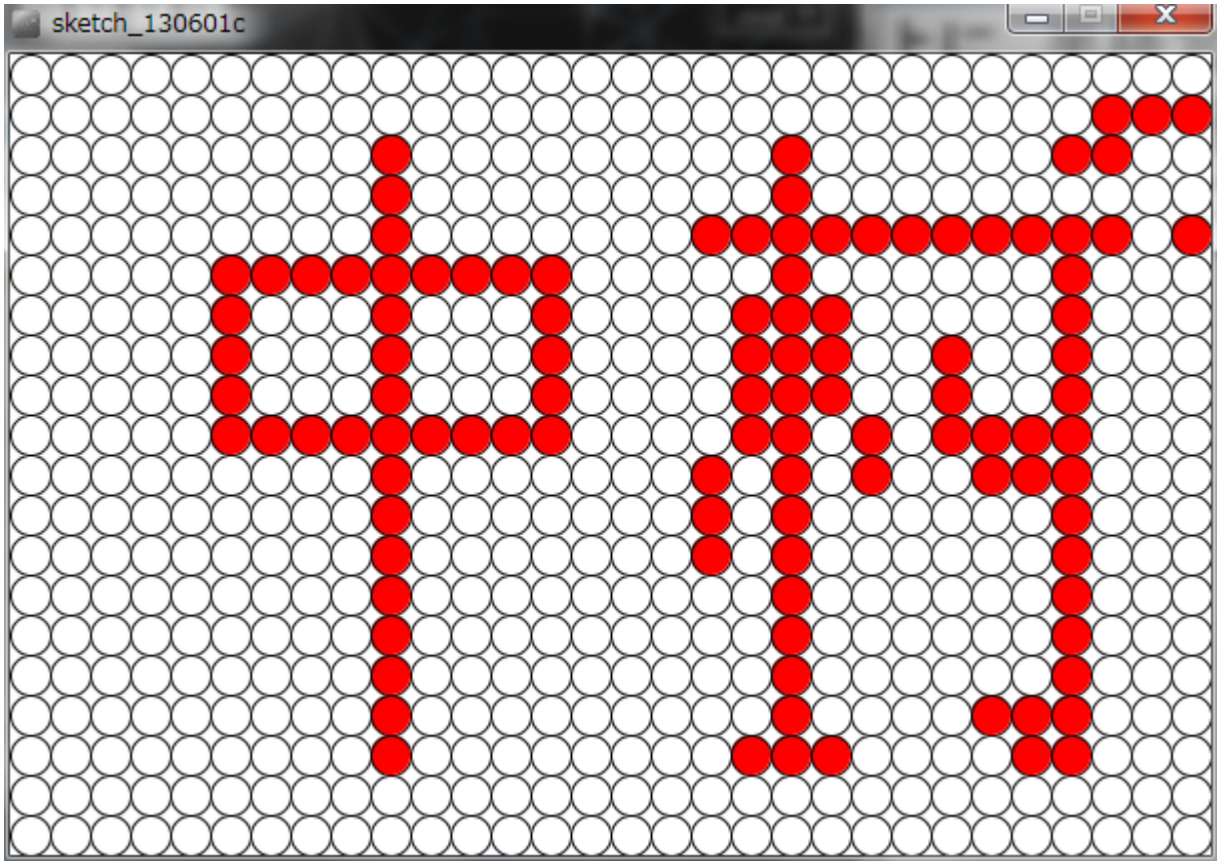
```
void draw(){
  background( 255 );

  for( int x=1; x<=5; x++ ){
    for( int y=1; y<=5; y++ ){
      if( square[x][y] == 0 ){
        // 黄色四角形を描画
      } else if( square[x][y] == 1 ){
        // 黒色四角形を描画
      }
    }
  }
}
```

0か1かで描画内容を変更

電光掲示板を作る

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) 600x400のウインドウの上に，半径10の円を敷き詰め，マウスカーソルが触れた場所が赤色に灯る電光掲示板のようなものを作るには？



電光掲示板を作る

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
  - 600x400に半径10の円を敷き詰める場合，横に30個，縦に20個敷き詰めることが可能
  - ライトが光っているか，光っていないかを管理する30x20の整数の配列を用意する
    - `int [][] light = new int [30][20];`
    - 光っていない場合は 0，光っている場合は 1 とする
  - 円の中心の座標は，変数 i と j を利用した場合 **(i*20+10, j*20+10)** となる（i, jは0から1ずつ増やす）
  - [i, j]番目の円の中心からマウスまでの距離を求めて中に入っていれば light[i][j] = 1; とする
  - light[i][j] が 1 なら塗りつぶしを赤，0なら白とする

電光掲示板を作る

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

```
int [][] light = new int [30][20];
void setup(){
  size( 600, 400 );
  // 最初にすべてを0にする
  int i=0;
  while( i<30 ){
    int j=0;
    while( j<20 ){
      light[i][j] = 0;
      j++;
    }
    i++;
  }
}
```

```
void draw(){
  background( 255, 255, 255 );
  int i=0;
  while( i<30 ){
    int j=0;
    while( j<20 ){
      // 円の中に入っていれば1にする
      if( dist(i*20+10, j*20+10, mouseX, mouseY)<10 ){
        light[i][j] = 1;
      }
      // 1なら赤色，0なら白色にする
      if( light[i][j] == 1 ){
        fill( 255, 0, 0 );
      } else {
        fill( 255, 255, 255 );
      }
      // 中心(i*20+10, j*20+10)の円を描画
      ellipse( i*20+10, j*20+10, 20, 20 );
      j++;
    }
    i++;
  }
}
```

電光掲示板を作る

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

```
int [][] light = new int [30][20];
void setup(){
  size( 600, 400 );
  // 最初にすべてを0にする
  for( int i=0; i<30; i++ ){
    for( int j=0; j<20; j++ ) {
      light[i][j] = 0;
    }
  }
}
```

for でも書ける！

```
void draw(){
  background( 255, 255, 255 );
  for( int i=0; i<30; i++ ){
    for( int j=0; j<20; j++ ){
      // 円の中に入っていれば1にする
      if( dist(i*20+10, j*20+10, mouseX, mouseY)<10 ){
        light[i][j] = 1;
      }
      // 1なら赤色，0なら白色にする
      if( light[i][j] == 1 ){
        fill( 255, 0, 0 );
      } else {
        fill( 255, 255, 255 );
      }
      // 中心(i*20+10, j*20+10)の円を描画
      ellipse( i*20+10, j*20+10, 20, 20 );
    }
  }
}
```

数が明確な場合は
forがわかりやすいかも

演習

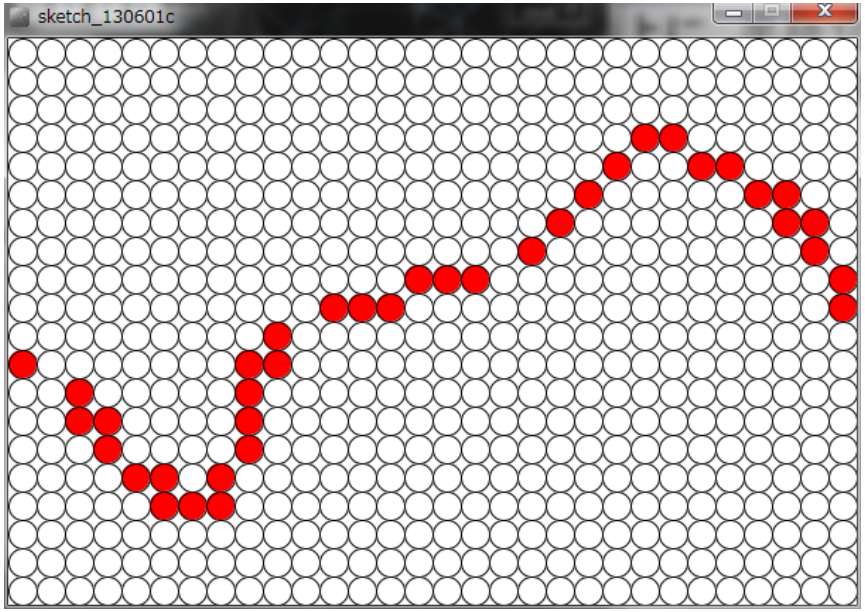
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

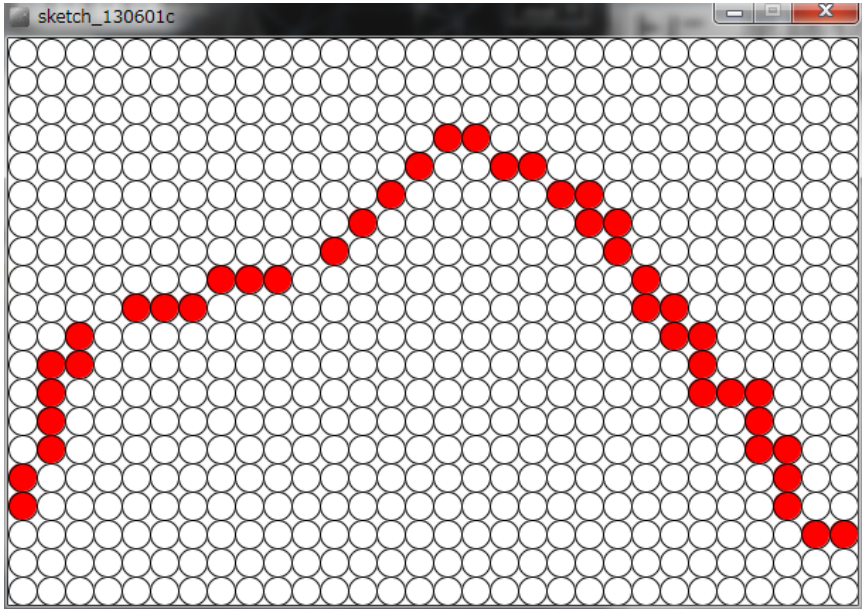
- 電光掲示板のプログラムで，マウスでクリックされるとON/OFFを切り替えるようにしましょう
  - ONなら赤色，OFFなら白色
  - マウスクリック mousePressed の時に距離の判定を行なって，ON/OFFを切り替えるライトを探す
- 電光掲示板のプログラムで，マウスのクリック回数で色を変えるようにしてみましょう
  - （例）白→赤→緑→青→白など

スクロールする電光掲示板

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) 600x400のウインドウの上に，半径10の円を敷き詰めた電光掲示板を，右から左にスクロールさせていくにはどうするか？





スクロールする電光掲示板

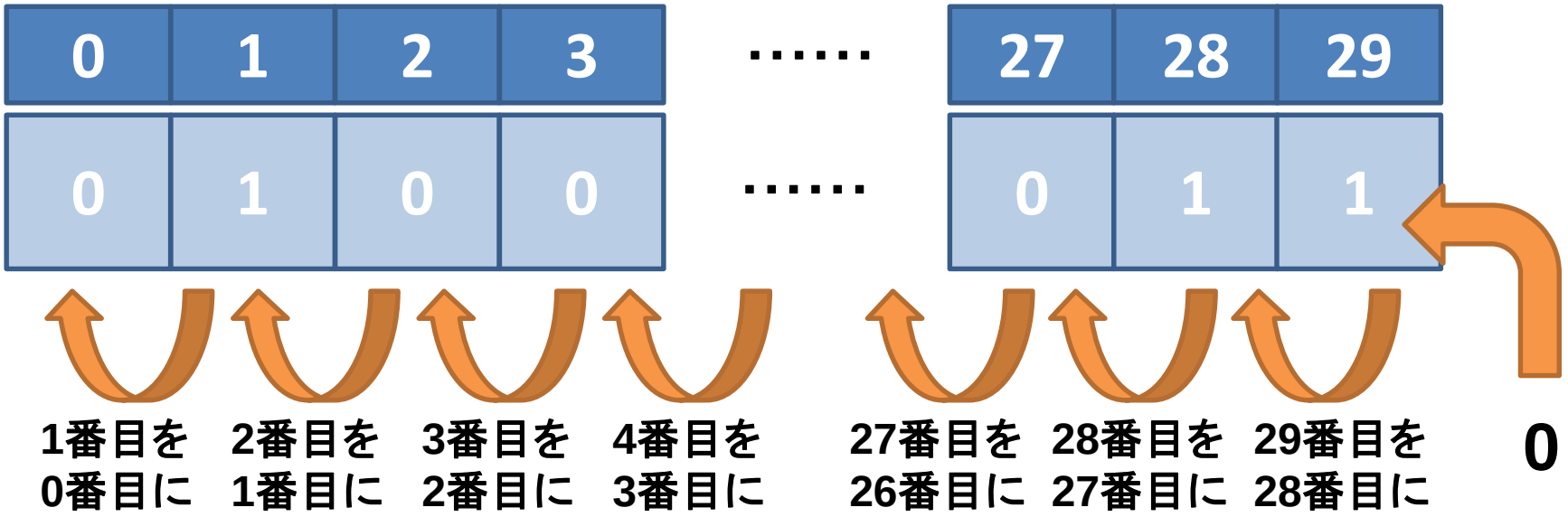
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 考え方
  - 30x20の整数の配列 light を用意して，まずは全てを 0 に設定する
  - すべての[i, j]について，マウスカーソルの下にある場合は，light[i][j] = 1; とする
    - dist(i*20+10, j*20+10, mouseX, mouseY)
  - light[i][j]が0なら白色，1なら赤色に塗り色を設定
  - [i, j]に相当する位置に円を描画する
  - 一回 draw される毎に，光っているライトの位置を右から左へ移動する（1つ左に，1つ右のを代入）
    - light[i-1][j] = light[i][j];

スクロールする電光掲示板

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

lightの配列のある j について…




```
light[i-1][j] = light[i][j];
```

1つずつ左へずらしていく
1番右には 0 を入れる

配列のエラーに注意

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

- light[i-1][j] = light[i][j] のため，iは1以上
  - iが0のとき，light[-1][j] = light[0][j] となってしまい，エラーになる！（-1番目なんて無い！）
- 右を左にコピーするので，左のものから順に処理をしていく
  - i は順に1ずつ増やしていく
- 多重ループは j を外のループにして，i を内のループにしたほうがわかりやすい

```
for( int j=0; j<20; j++ ){
    for( int i=1; i<30; i++ ){
        light[i-1][j] = light[i][j];
```

スクロールする

```
int [][] light = new int [30][20];
void setup(){
  size( 600, 400 );
  // 最初にすべてを0にする
  int i=0;
  while( i<30 ){
    int j=0;
    while( j<20 ){
      light[i][j] = 0;
      j++;
    }
    i++;
  }
}
```

```
void draw(){
  background( 255, 255, 255 );
  int i=0;
  int j=0;
  while( j<20 ){
    i=1;
    while( i<30 ){
      light[i-1][j] = light[i][j];
      i++;
    }
    light[29][j] = 0;
    j++;
  }
  i=0;
  while( i<30 ){
    j=0;
    while( j<20 ){
      if( dist(i*20+10, j*20+10, mouseX, mouseY)<10 ){
        light[i][j] = 1;
      }
      if( light[i][j] == 1 ){
        fill( 255, 0, 0 );
      } else {
        fill( 255, 255, 255 );
      }
      ellipse( i*20+10, j*20+10, 20, 20 );
      j++;
    }
    i++;
  }
}
```

スクロールする電光掲示板

明治大学総合数理学部
先端メディアサイエンス学科

```
int [][] light = new int [30][20];
void setup(){
  size( 600, 400 );
  // 最初にすべてを0にする
  for( int i=0; i<30; i++ ){
    for( int j=0; j<20; j++ ){
      light[i][j] = 0;
    }
  }
}
```

```
void draw(){
  background( 255, 255, 255 );
  for( int j=0; j<20; j++ ){
    for( int i=1; i<30; i++ ){
        light[i-1][j] = light[i][j];
    }
    light[29][j] = 0;
  }
  for( int i=0; i<30; i++ ){
    for( int j=0; j<20; j++ ){
      if( dist(i*20+10, j*20+10, mouseX, mouseY)<10 ){
        light[i][j] = 1;
      }
      if( light[i][j] == 1 ){
        fill( 255, 0, 0 );
      } else {
        fill( 255, 255, 255 );
      }
      ellipse( i*20+10, j*20+10, 20, 20 );
    }
  }
}
```

演習

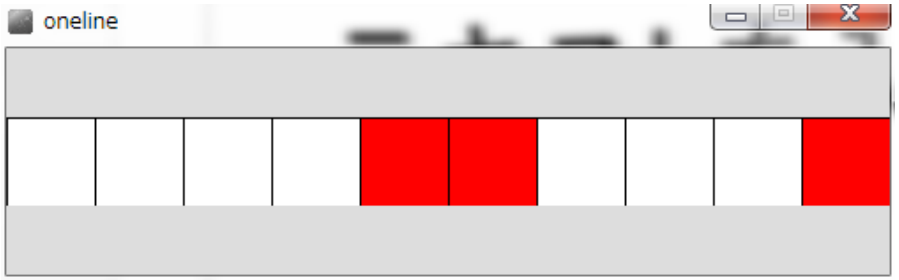
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 先述のプログラムのスクロール方向を左から右にしてみましょう
- 先述のプログラムのスクロール方向を上から下にしてみましょう
- 先述のプログラムで文字を表示してみましょう
  - どうやったら文字を表示できるでしょうか？

[演習] 左端から右端へ

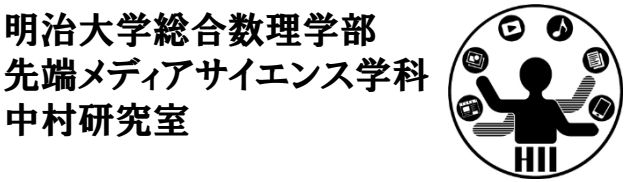
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- クリックによって赤／白の表示が切り替わる1次元の掲示板を，1秒毎に順に左側に移動し，左端のものを右端から登場させたい
- 単純に繰り返しで移動すると右端が白になってしまう

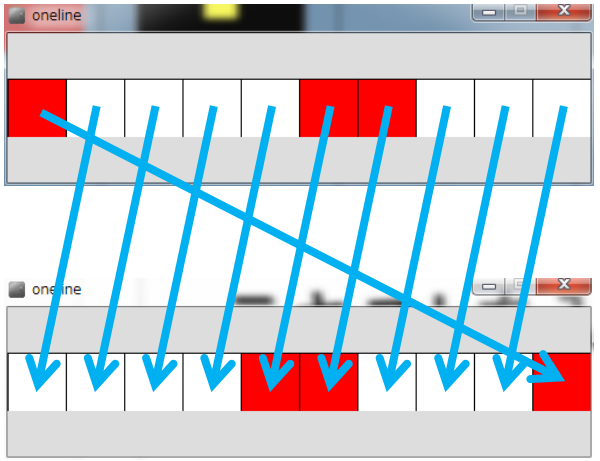


[演習] 左端から右端へ

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
  - 値を順に右から左へコピーしていく
    - status[0] = status[1];
    - status[1] = status[2];
      :
    - status[7] = status[8];
    - status[8] = status[9];
    - ここで，status[9] = status[0]; にするとすでに status[0] には status[1] の値が代入されてしまっているので駄目！

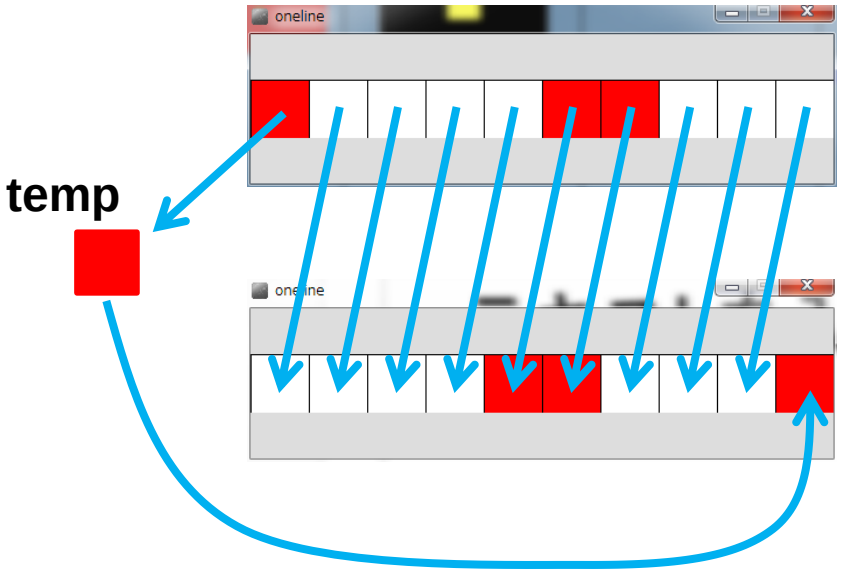


[演習] 左端から右端へ

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
  - 左端の値を，一時的に他の変数に保存しておいて，そこに保存していた値を右端に入れれば良い！

```
temp = status[0];
status[0] = status[1];
status[1] = status[2];
      :
status[7] = status[8];
status[8] = status[9];
status[9] = temp;
```



while

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

void draw(){
  int temp = status[0];
  int x=0;
  while( x < 9 ){
    status[x] = status[x+1];
    x++;
  }
  status[9] = temp;

  x = 0;
  while( x < 10 ){
    if( status[x] == 1 ){
      fill( 255, 0, 0 );
    } else {
      fill( 255 );
    }
    rect( x*50, 0, 50, 50 );
    x++;
  }
}
```

for

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

void draw(){
  int temp = status[0];
  for( int x=0; x<9; x++ ){
    status[x] = status[x+1];
  }
  status[9] = temp;

  for( int x=0; x<10; x++ ){
    if( status[x] == 1 ){
      fill( 255, 0, 0 );
    } else {
      fill( 255 );
    }
    rect( x*50, 0, 50, 50 );
  }
}
```

演習

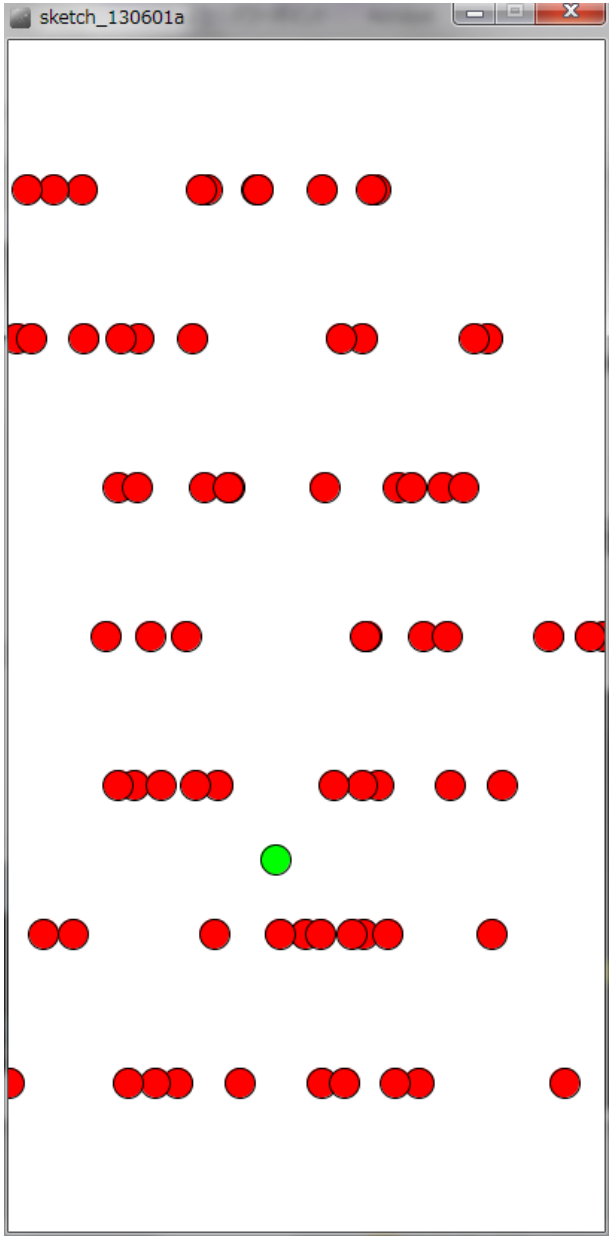
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 右端のものを，左端から登場させるにはどうするか？
- 2次元の掲示板のアニメーションを行い，左端から右端に登場させるようにするにはどうしたらよいだろうか？

[演習] 避けゲーム

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

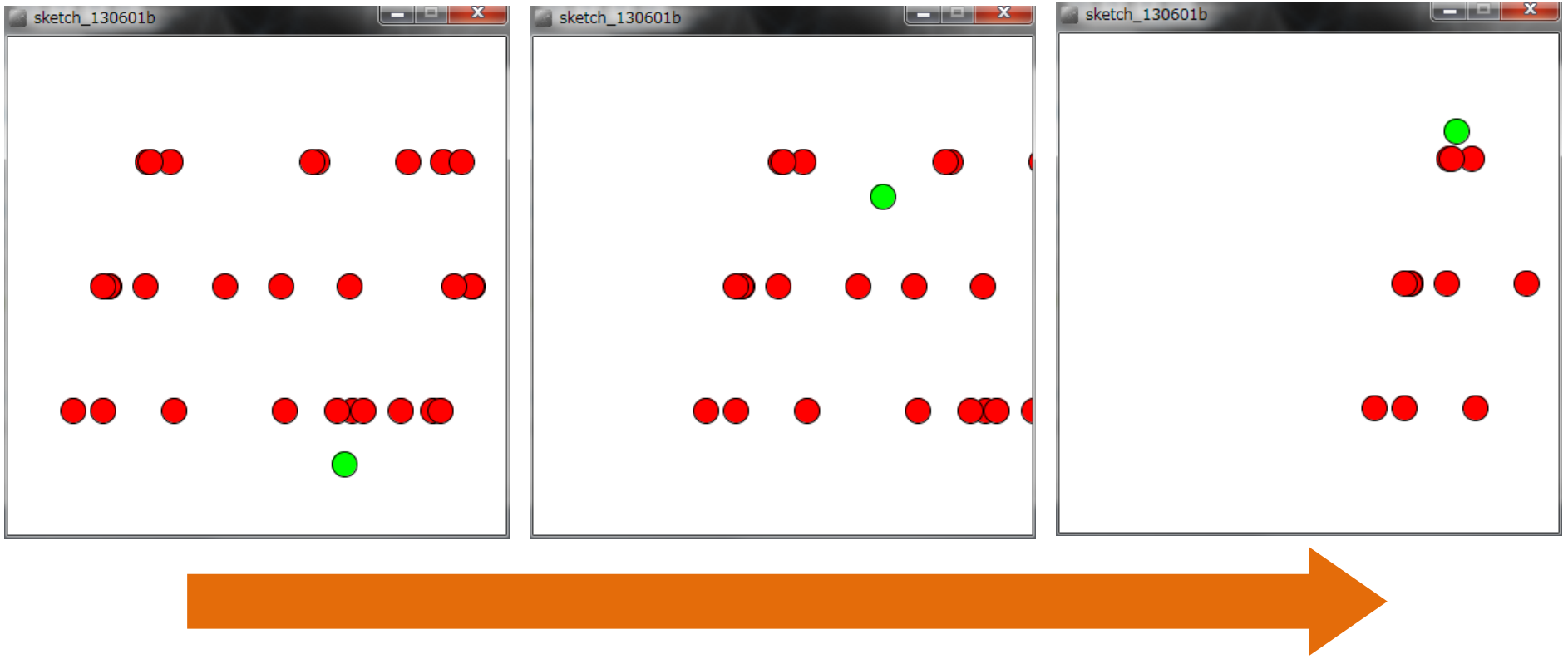
- 400x800のウインドウを作り，Y座標が100，200，300，400，500，600，700のところに，それぞれ円を10個ずつランダムに表示しましょう



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

(Q) 400x400の画面上の，Y座標100，Y座標200，Y座標300の場所に円を10個ずつ表示された円を左から右に動かそう（直径20）また，速度はランダムにせよ



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
  - 3x10個のX座標を格納する float 型の2次元配列を用意（float [][] posX = new float [3][10];）
  - 3x10個の速度を格納する float 型の2次元配列を用意（float [][] speedX = new float [3][10];）
  - 3x10個の配列に random で適当な値を代入
  - 3x10個の配列の値に応じて円を描画
  - 1回描画する度に，配列のそれぞれのX座標をspeedX分だけ右へ移動

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

- 3x10の2次元配列を作成し，それをそのまま変化させる！

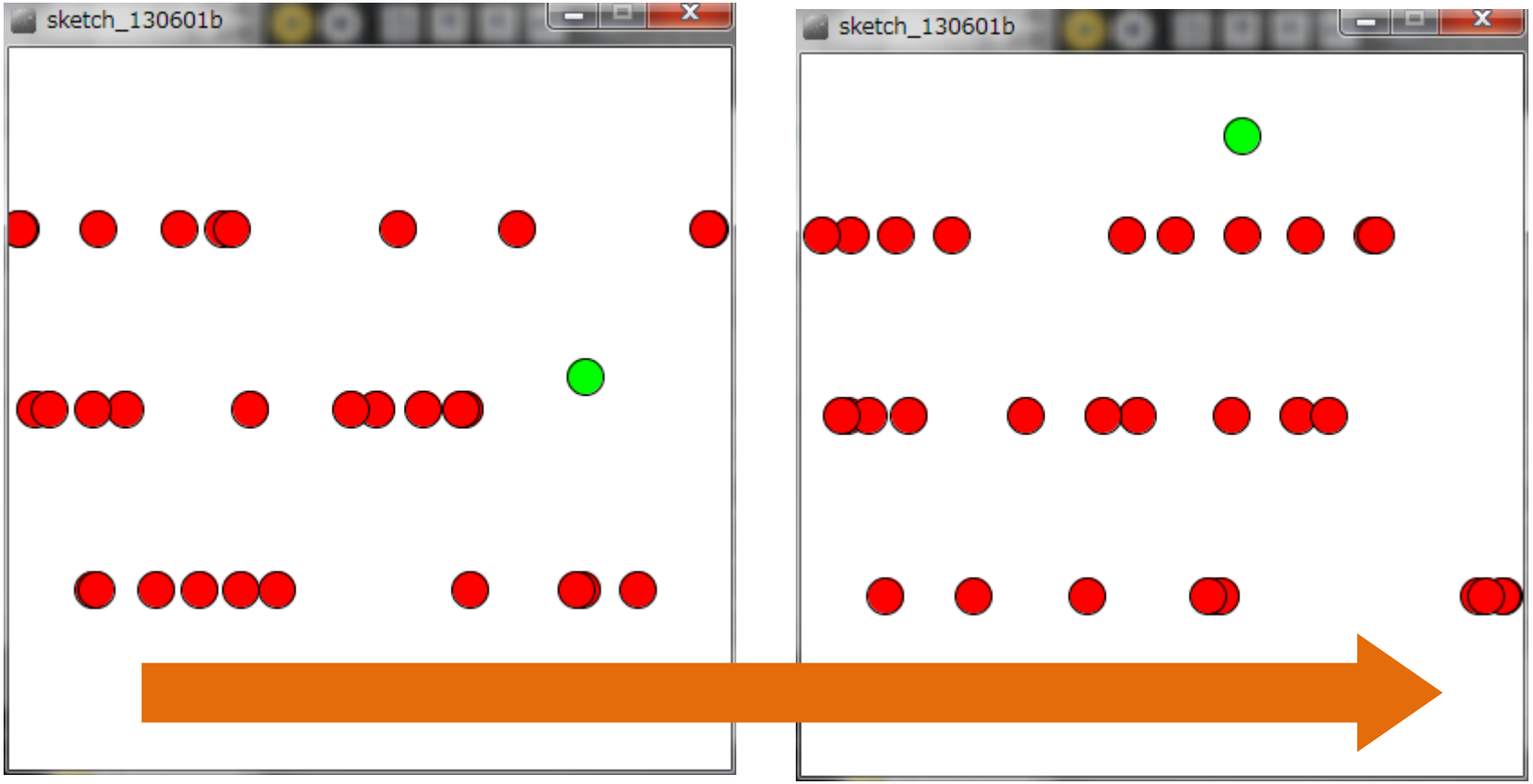
```
float  [][]  posX = new
float[3][10];
float  [][] speedX = new
float[3][10];
void setup(){
  size( 400, 400 );
  int i=0;
  while( i<3 ){
    int j=0;
    while( j<10 ){
      posX[i][j] = random(0,400);
      speedX[i][j] = random(5);
      j++;
    }
    i++;
  }
}
```

```
void draw(){
  background( 255 );
  fill( 0, 255, 0 );
  ellipse( mouseX, mouseY, 20, 20 );
  fill( 255, 0, 0 );
  int i=0;
  while( i<3 ){
    int j=0;
    while( j<10 ){
      posX[i][j] += speedX[i][j];
      ellipse( posX[i][j], (i+1)*100, 20, 20 );
      j++;
    }
    i++;
  }
}
```

円を一斉に動かす改

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400x400の画面に3列に配置された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



円を一斉に動かす改

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[i][j] > 400) なら posX[i][j] から 400引く！

```
float  [][]  posX = new float [3][10];
float  [][]  speed = new float [3][10];
void setup(){
  size( 400, 400 );
  int i=0;
  while( i<3 ){
    int j=0;
    while( j<10 ){
      posX[i][j] = random(0,400);
      speed[i][j] = random(0,5);
      j++;
    }
    i++;
  }
}
```

```
void draw(){
  background( 255 );
  fill( 0, 255, 0 );
  ellipse( mouseX, mouseY, 20, 20 );
  fill( 255, 0, 0 );
  int i=0;
  while( i<3 ){
    int j=0;
    while( j<10 ){
      posX[i][j] = posX[i][j] + speed[i][j];
      if( posX[i][j] > 400 ){
        posX[i][j] = posX[i][j] - 400;
      }
      ellipse( posX[i][j], (i+1)*100, 20, 20 );
      j++;
    }
    i++;
  }
}
```

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

- 一斉に動く円との衝突判定をしてみましょう
- 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！