

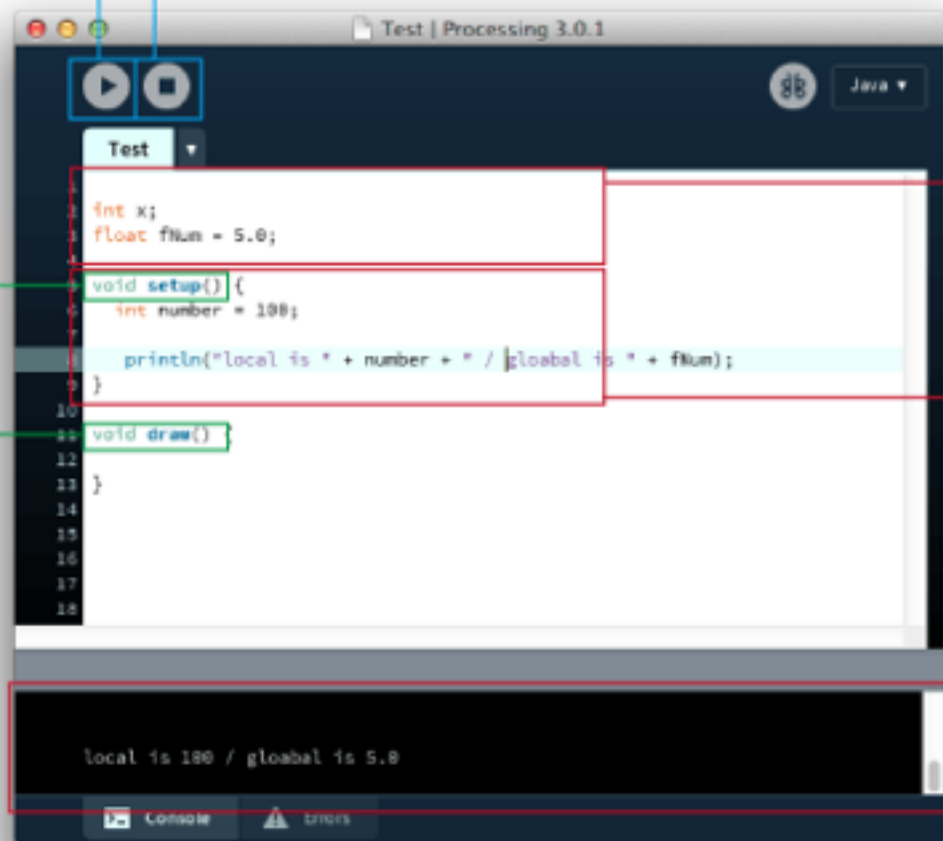
モバイルシステム演習

Processing

- 簡単なインタラクション -

株式会社GOCCO.
watanabe@gocco.co.jp

プログラムの実行ボタン
プログラムの停止ボタン



グローバル変数領域

ここで宣言された変数は、他の関数の中でも使用できる。
この領域で宣言された変数をグローバル変数と呼ぶ。

関数領域

{ } で囲われた領域内で宣言された変数は、その領域内でのみしか使用することが出来ない。この領域内で宣言された変数をローカル変数と呼ぶ。

コンソール

print 関数でログを出力した場合、ここに内容が表示される

setup 関数 (予約関数)

初期化を行う関数で、プログラム実行後に一番最初に 1 度だけ処理が行われる関数。

draw 関数 (予約関数)

プログラム実行後、毎フレーム処理が行われる関数。描画を更新したり、アニメーションさせたりするための関数。
プログラムの心臓部分

マウスポインタの位置を使う

`mouseX`... キャンバス上にあるマウスポインタのX座標

`mouseY`... キャンバス上にあるマウスポインタのY座標

`pmouseX`... 1フレーム前のマウスポインタのX座標

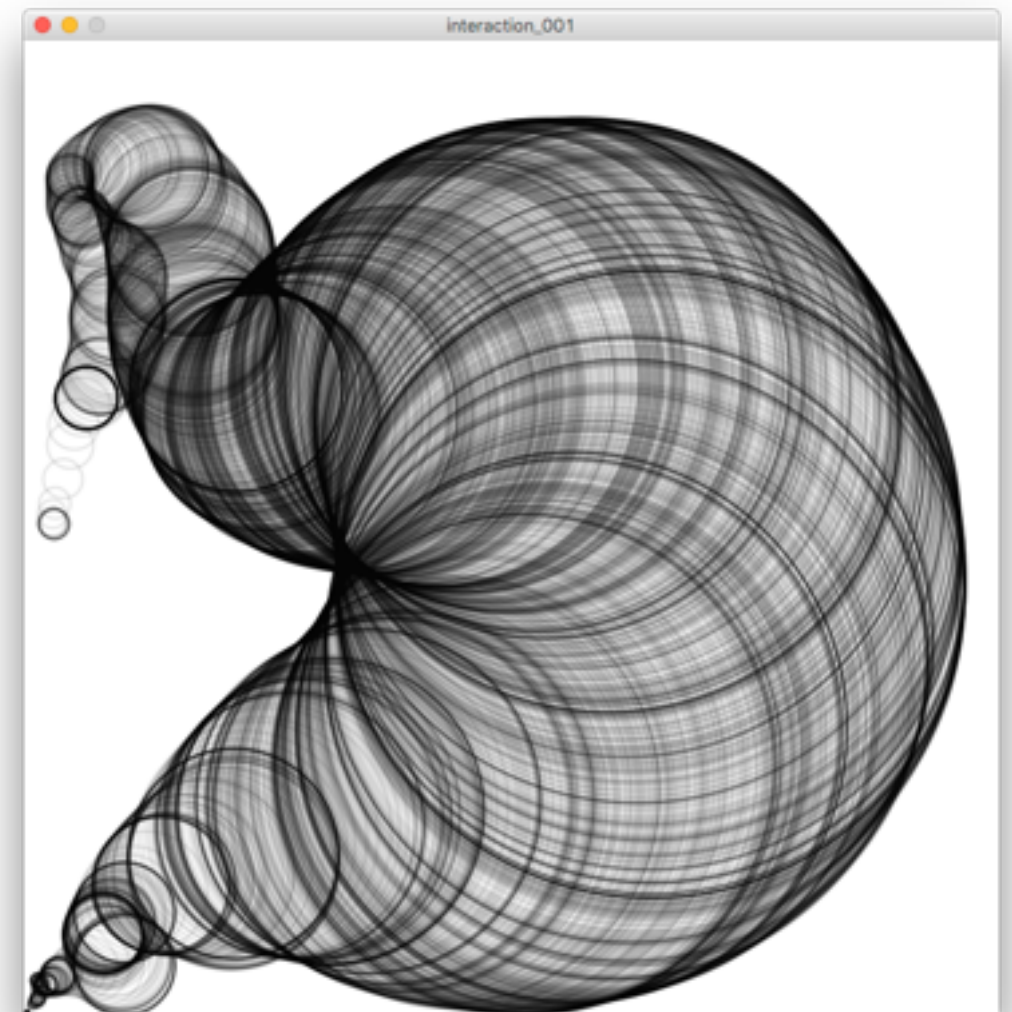
`pmouseY`... 1フレーム前のマウスポインタのY座標

マウスポインタの位置を使う

毎フレーム x , y 座標あるいは矩形の大きさの値にマウス座標を割り当てると、動きのある図形を作成することができる

マウスポインタの位置を使う

```
void setup(){  
  size(720, 720);  
  smooth();  
  noFill();  
  background(255);  
}  
  
void draw(){  
  float radius = mouseX;  
  strokeWeight(2);  
  stroke(0, 25);  
  ellipse(mouseX, mouseY, radius, radius);  
}
```



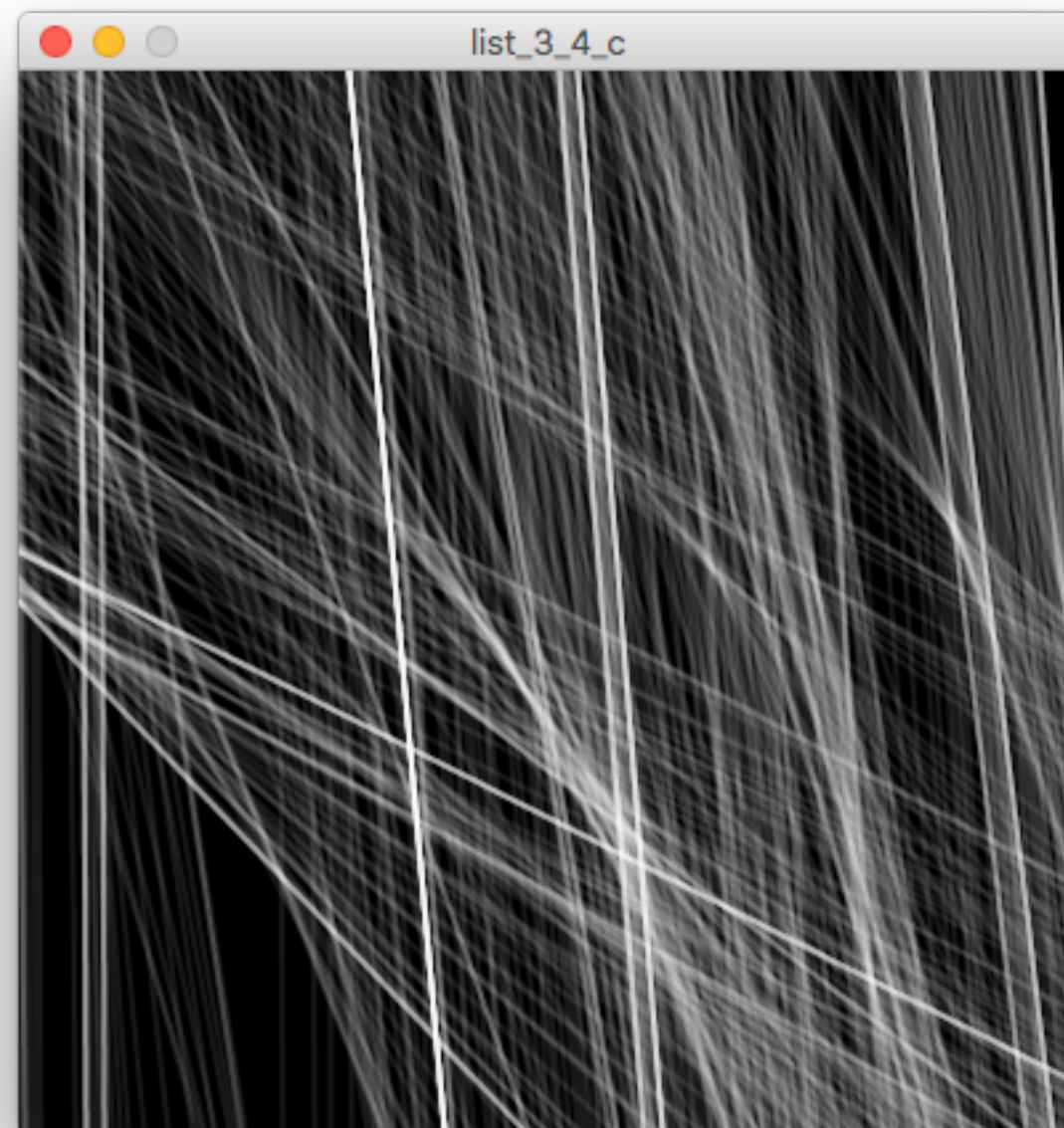
だんだん大きくなる

```
int radius = 0;
void setup() {
  size(800, 800);
  radius = 10;
}

void draw() {
  ellipse (width / 2, height / 2, radius, radius);
  radius++;
}
```

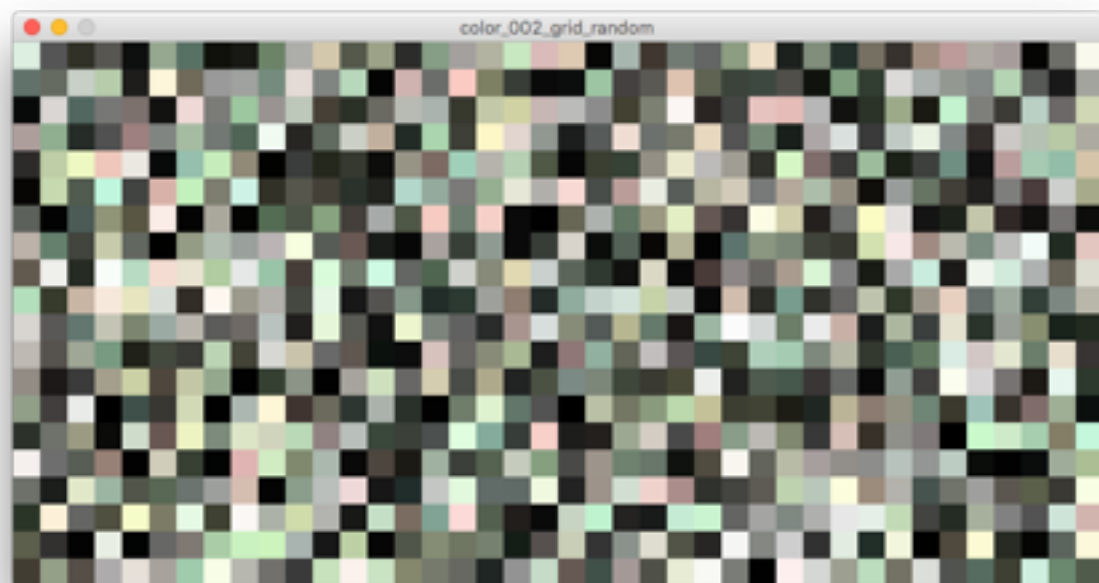
課題

同じ要領で、縦ラインの図形を
マウスに合わせて生成してみよう

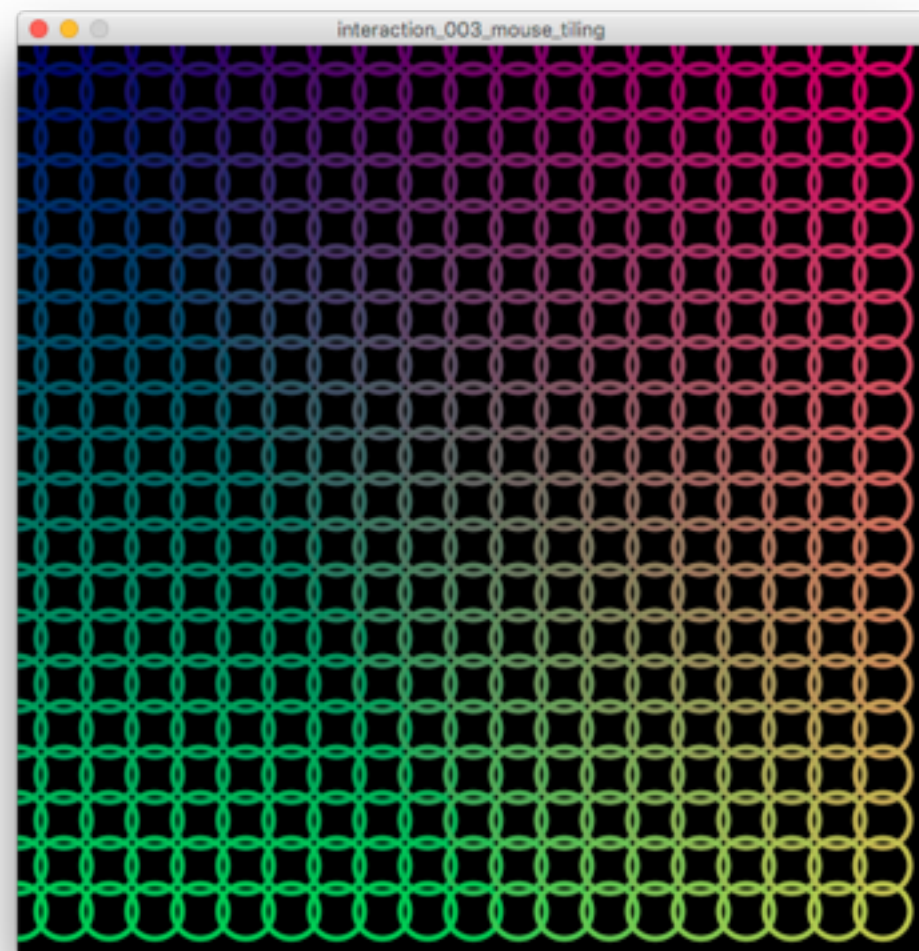


挑戦課題

マウス座標に応じて色や形のサイズを変えてみよう



カラーノイズ



タイリング