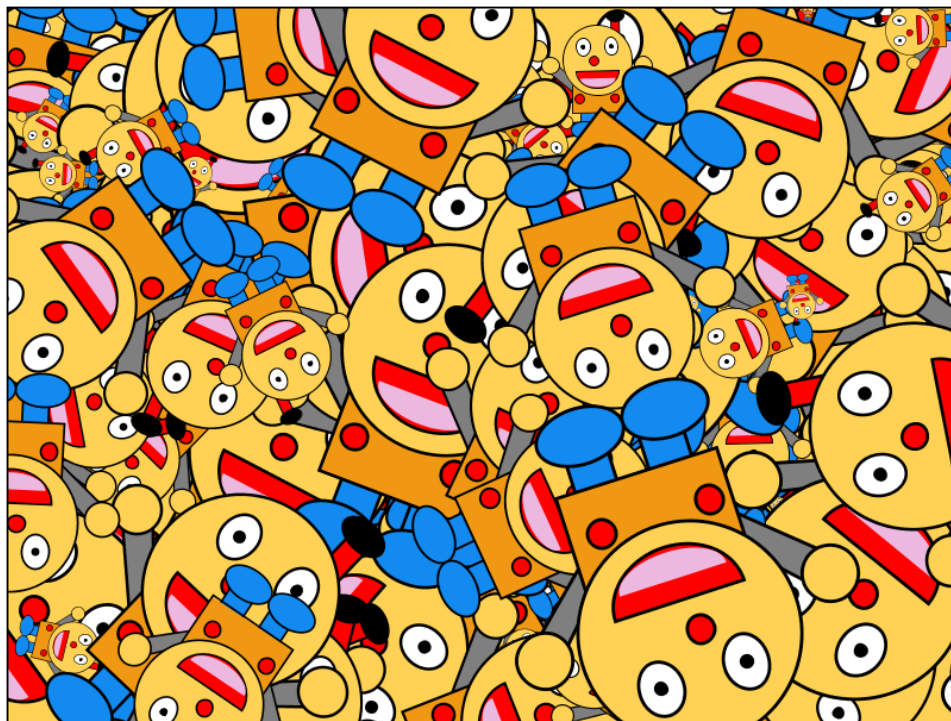


プログラミング演習I（第11回）課題

• 基本① スケッチ名 : charaRandom

- chara1で作成したキャラクタを画面内にたくさん描くプログラムを作成してください。
- 位置・大きさ・回転をランダムにしてください。



プログラミング演習I（第11回）課題

• 基本② スケッチ名 : lifegame_func

- 誕生、生存、過疎、過密によってセルが生まれたり死んだりするライフゲームを作ろう。
- ライフゲームでは、対象とするセルの周囲8マスが活着しているか死んでいるかを数え、その結果に応じてセルを活着している状態にするか、死んでいる状態にするかを切り替える。
- 100x100のマス目を用意し、セルが活着している場合は緑色の四角形を、死んでいる場合は黒色の四角形を描画するようにせよ。ライフゲームのルールは次ページで説明する。
- 配布する lifegame_func.pde の checkDeadOrAlive 関数を埋めてライフログゲームを完成させよ
- 下記URLの安定状態が幾つか観測されたら成功
<http://ja.wikipedia.org/wiki/ライフゲーム>

プログラミング演習I（第11回）課題

あるマス(赤フレーム)の縦・横・斜めの8マスの生死の状態(生の数)に注目する

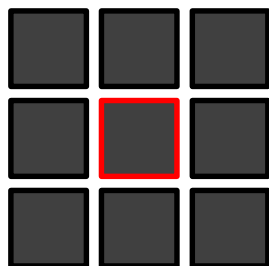
【誕生】死んでいるセルに隣接する生きたセルがちょうど3つならば次世代が誕生

【生存】生きているセルに隣接する生きたセルが2つか3つならば次世代でも生存

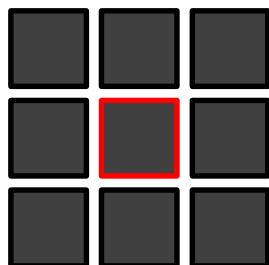
【過疎】生きているセルに隣接する生きたセルが1つ以下ならば過疎により死滅

【過密】生きているセルに隣接する生きたセルが4つ以上ならば過密により死滅

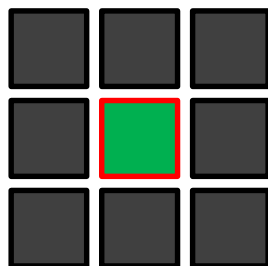
すべて死



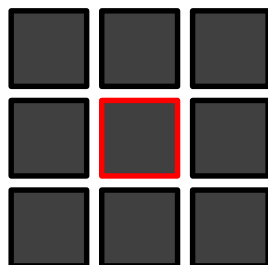
変化なし



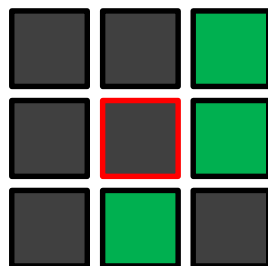
すべて死



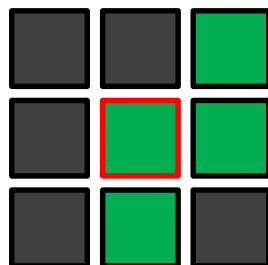
寂しくて死ぬ



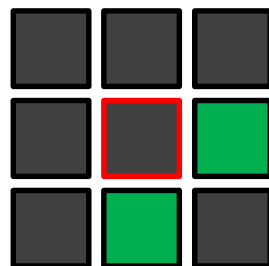
3つのマス



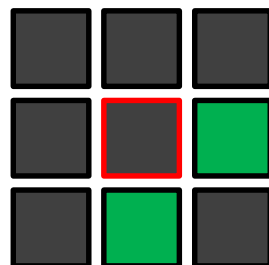
生まれる



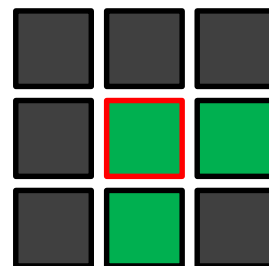
2つの生



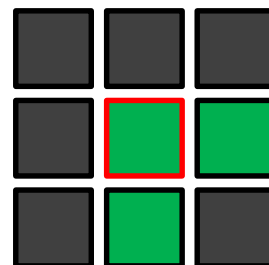
変化なし



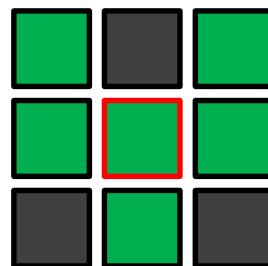
2つ以上の生



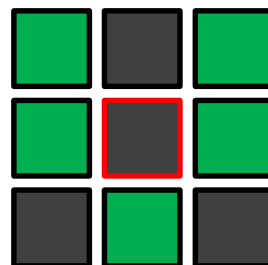
快適で変化なし



3つ以上の生



過密で死ぬ



プログラミング演習I（第11回）課題

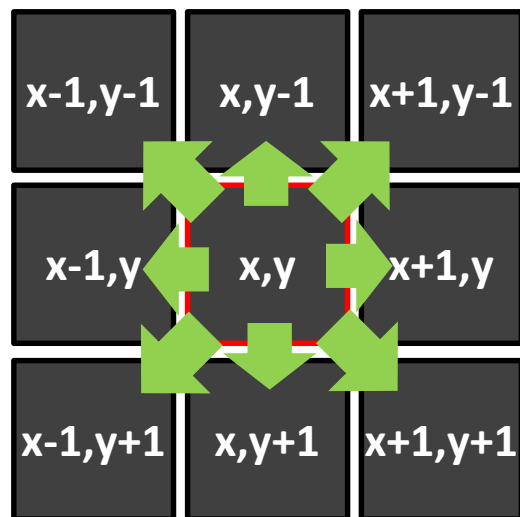
- 整理するとそのセルの周りの生きているセルの数が

3つなら生

2つなら維持

それ以外なら死

チェックする配列の添字は何になるか？



0から始めると
条件分岐が多くて面倒！

-1,-1	0,-1	1,-1
-1,0	0,0	1,0
-1,1	0,1	1,1

1から始めると
条件分岐が少なくなる

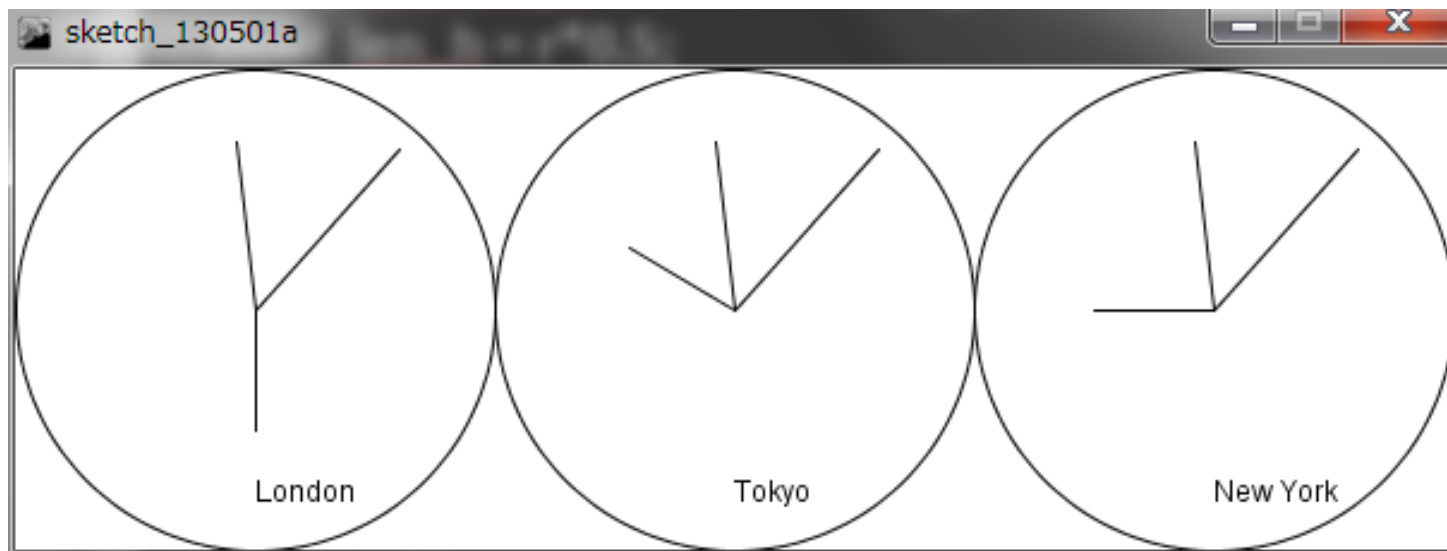
0,0	1,0	2,0
0,1	1,1	2,1
0,2	1,2	2,2

表示しない外周を用意して、周囲の「生」の数を数えると楽！

プログラミング演習I（第11回）課題

• 基本③ スケッチ名 : worldClock

- まずアナログ時計を表示する関数を作成せよ
- 次に, LondonとTokyoとNew Yorkの時計を表示するアナログ時計を横に並べてみましょう(時差考慮)
- サマータイムで日本とロンドン間の時差が+8時間, 日本とNYの間の時差が-13時間



プログラミング演習I（第11回）課題

• 基本④ スケッチ名 : TwinPrime

- 1000までの双子素数を求めて表示するプログラムを作成せよ
 - 双子素数とは, (3, 5), (5, 7), (11, 13), (17, 19)のように差が2の素数のペアのこと
 - 双子素数が無限に存在するか否かというのは, 数学の未解決問題
- 素数を求める関数を作成し, その関数を活用せよ

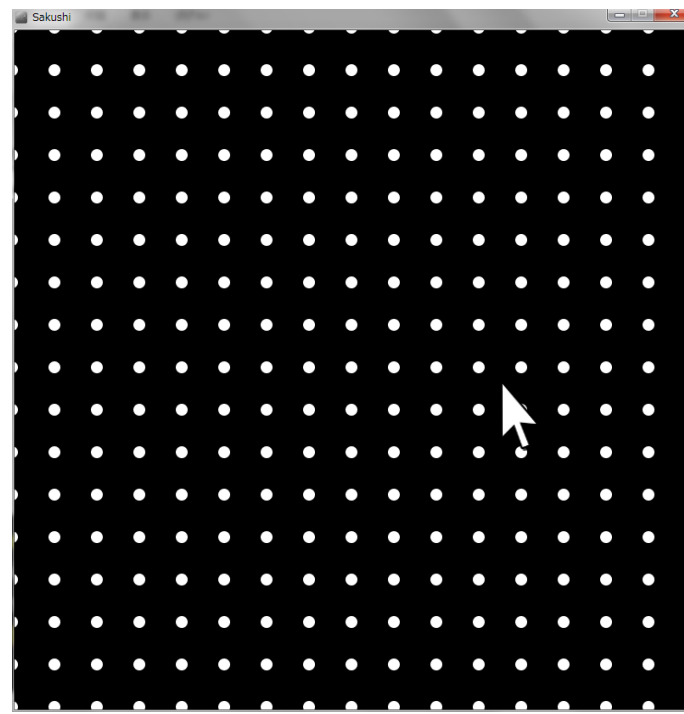
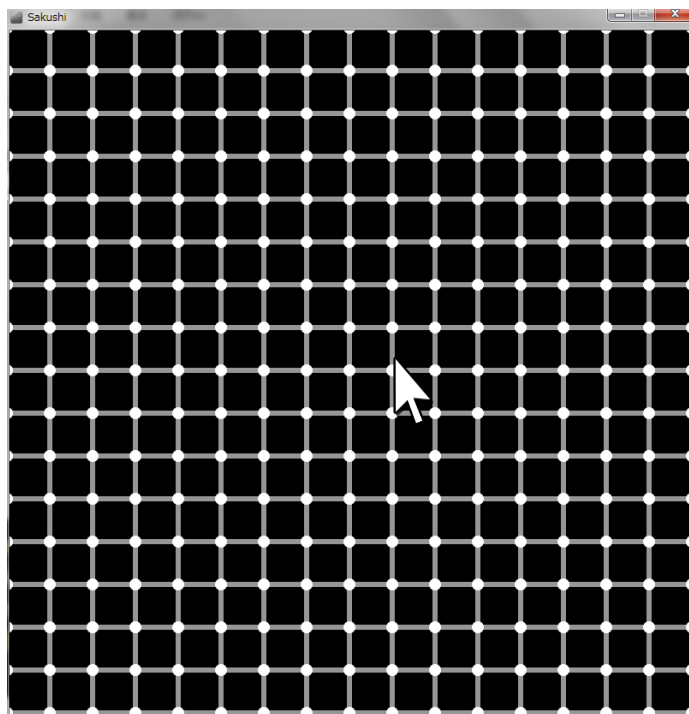
```
3 5
5 7
11 13
17 19
29 31
41 43
59 61
71 73
101 103
107 109
137 139
149 151
179 181
191 193
197 199
227 229
239 241
269 271
281 283
311 313
347 349
419 421
431 433
```

プログラミング演習I（第11回）課題

• 発展① スケッチ名 : kirameki

- きらめき格子錯視を作成せよ(黒背景, 灰色の格子, 枠無し白丸)
- ただし、マウスクリックによって格子の灰色の線を表示したり, 非表示にしたりと切り替えることが出来るようにせよ。

- <http://www.kecl.ntt.co.jp/IllusionForum/v/scintillatingGrid/ja/index.html>



プログラミング演習I（第11回）課題

• 発展② スケッチ名: LangtonsAnt

- ラングトンのアリは、黒と白のマス目が並んだ広い平面の中に棲んでおり、アリは東西南北のいずれかの方向を向いている。そして、時計の針が1つ進むと1マス進んで次のルールに従い行動する
 - 黒いマス目に入ったら左に90度回転する
 - 白いマス目に入ったら右に90度回転する
 - それまでアリがいたセルは白から黒、黒から白に色を変える
- このとき、最初の黒色がどのようなパターンであっても、最終的にハイウェイと呼ばれる道ができ、アリが脱出していくというものである（必ずハイウェイが出来て逃げるかどうかというのは未解決問題）
- 上記プログラムを完成させて、ラングトンのアリが脱出する様子を観測せよ

