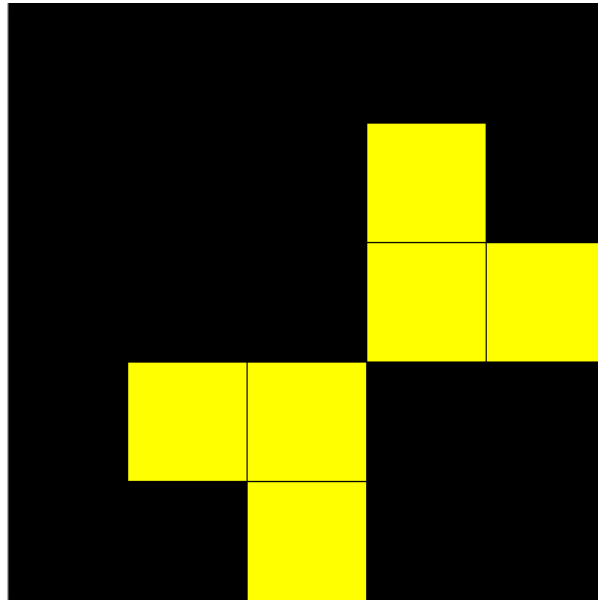


プログラミング演習I (第9回) 課題

• 基本① スケッチ名：**basic_LightsOut**

- 横5マス、縦5マスの盤面を作り、そのマス目をクリックすると、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を作れ
- なお、すべてが黒色になったらCLEAR!と表示するようにせよ
- ただし、起動したタイミングで下記のような表示になっているようにせよ（予習した人はこれをやるだけ！）

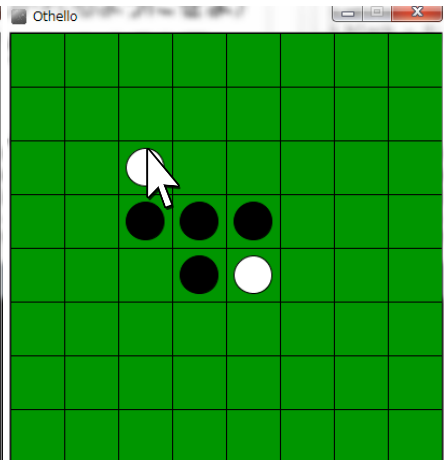
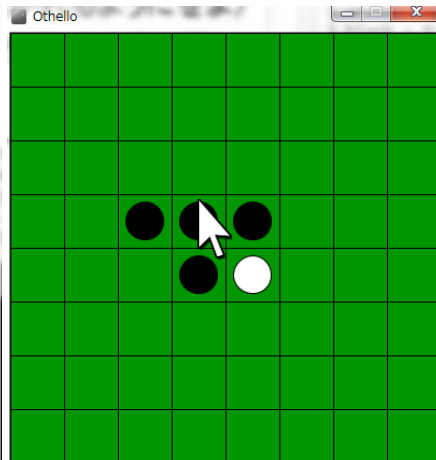
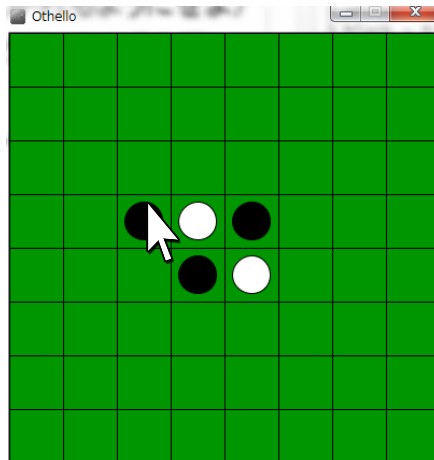
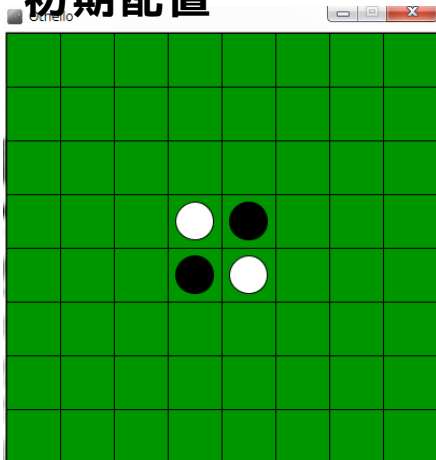


プログラミング演習I (第9回) 課題

• 基本② スケッチ名 : `basic_Othello`

- 横8マス、縦8マスのオセロの盤面と左下のコマの初期配置を作れ
- コマがないマスをクリックすると、ターンに応じて白いコマまたは黒いコマが置かれるようにせよ (白いコマ、黒いコマは交互に置かれるようにせよ)
 - ターンを管理する変数を用意して、あきマスに置かれたら値を変更する!
- また、黒いコマをクリックすると白いコマへ、白いコマをクリックすると黒いコマへ変わるようにせよ
- 余裕がある人は白や黒のコマの数を表示したり、コマを自動で反転したりするようにしてみよう。

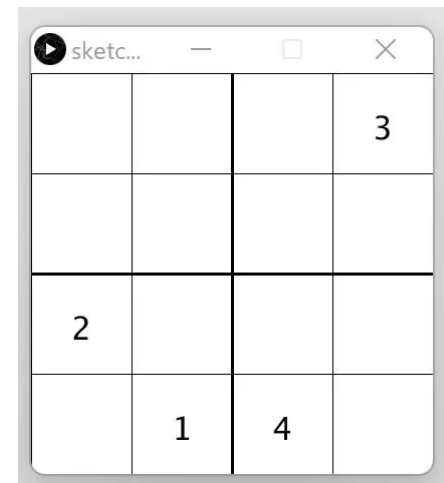
初期配置



プログラミング演習I (第9回) 課題

• 基本③ スケッチ名：**basic_Sudoku44**

- 下記に示す4x4マスの数独を作成し（起動時にこの数値が入っているものとせよ）、クリックすることで数字を埋め、手動で解けるようにせよ（解けたかどうかの判定は不要です）
- なお、空白マスをクリックすると1、1のマスをクリックすると2、2のマスをクリックすると3、3のマスをクリックすると4、4のマスをクリックすると空白になるようにせよ。
- また、もともとセットする値は下記のとおりとせよ。
 - クリックにより問題として設定されている数字が変わってしまっても良い（できるひとは変わらないようにせよ）
- 4つのブロックを分けるように太線を入れましょう。



課題例は下記より

<https://sudoku.tokyo/mini-sudoku/printable-4x4-easy-sudoku.php>

<https://gyazo.com/c938797453411cb0129bde8638a21e64>

プログラミング演習I (第9回) 課題

発展① スケッチ名: advanced_Bingo

- 500x500のウィンドウ内に横5マス、縦5マスのビンゴの盤面を作れ
 - 5x5のマス目に1から25までの数字を重複なくランダムに配置せよ
 - ウィンドウ内でマウスをクリックするたびに、1から25までの数字をランダムに取得せよ (以前と同じ数字が出てても良い)
 - 選ばれた数字を標準出力するとともに、マス目を灰色にせよ
- 以下は条件ではないが、チャレンジできる人は是非
- 1から25までの数字がランダムに取得される時、重複しないようにせよ
 - ビンゴの条件がそろったらビンゴと表示したり、ビンゴが揃った場所を別の色で塗りつぶしてみよう!

sketch_210617a				
24	6	18	25	22
1	10	4	7	11
8	20	14	3	15
12	2	16	21	23
13	5	17	9	19

sketch_210617a				
22	1	11	15	10
21	16	13	14	5
19	23	25	4	6
9	7	18	12	20
3	2	17	24	8

4が選ばれました
14が選ばれました

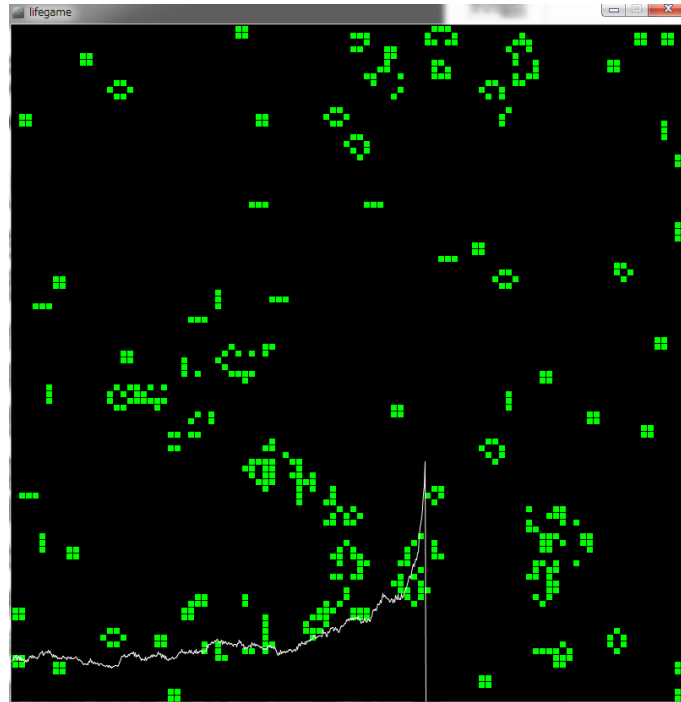
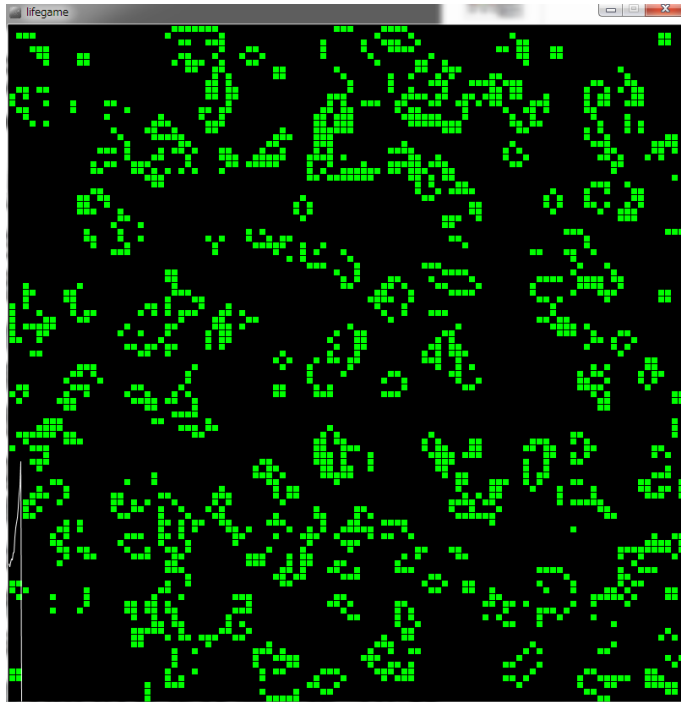
プログラミング演習I (第9回) 課題

• 発展② スケッチ名 : advanced_Lifegame

- 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。
- **800x800のウィンドウ内に100x100**のマス目を用意し、セルが活着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 advanced_Lifegame.pde をそのまま使い、drawの内部を書き換えよ（仕様を満たせば自分で1から作っても良いよ）
- 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9回) 課題

- ライフゲームとはこんなもの
 - ある種の生命のシミュレーション
 - 誕生, 生存, 過疎, 過密で生死を繰り返す



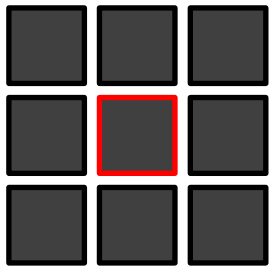
下記動画も面白いので参考までに
ライフゲームの世界 <http://www.nicovideo.jp/mylist/34610498>

プログラミング演習I (第9回) 課題

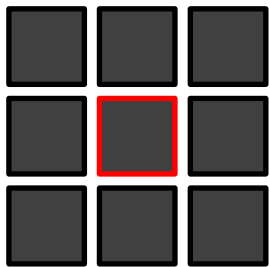
あるマス (赤フレーム) の縦・横・斜めの8マスの生死の状態 (生の数) に注目する

- 【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

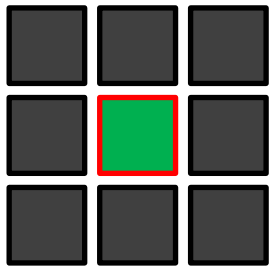
すべて死



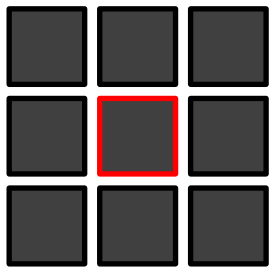
変化なし



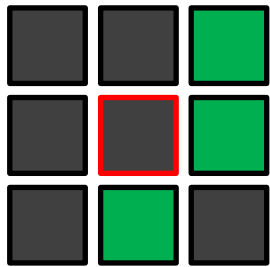
すべて死



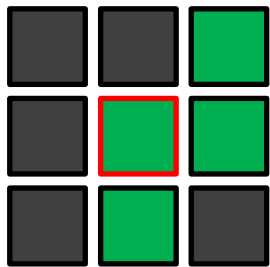
寂しくて死ぬ



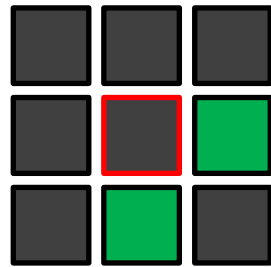
3つのマス



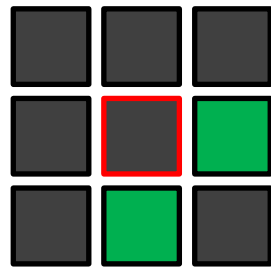
生まれる



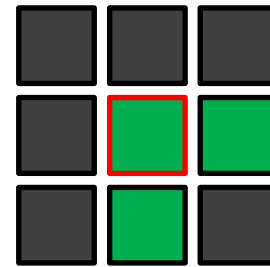
2つの生



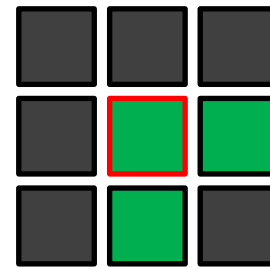
変化なし



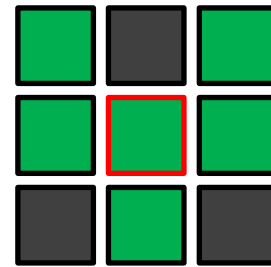
2つ以上の生



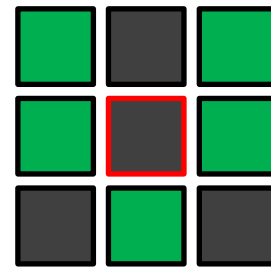
快適で変化なし



3つ以上の生



過密で死ぬ



プログラミング演習I (第9回) 課題

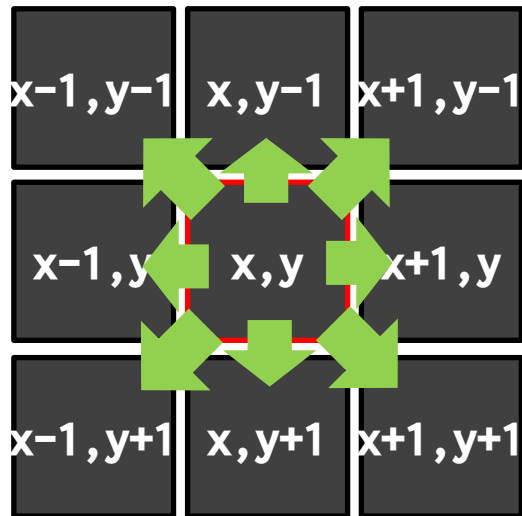
- ルールを整理すると . . .

数が3つなら生

数が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！