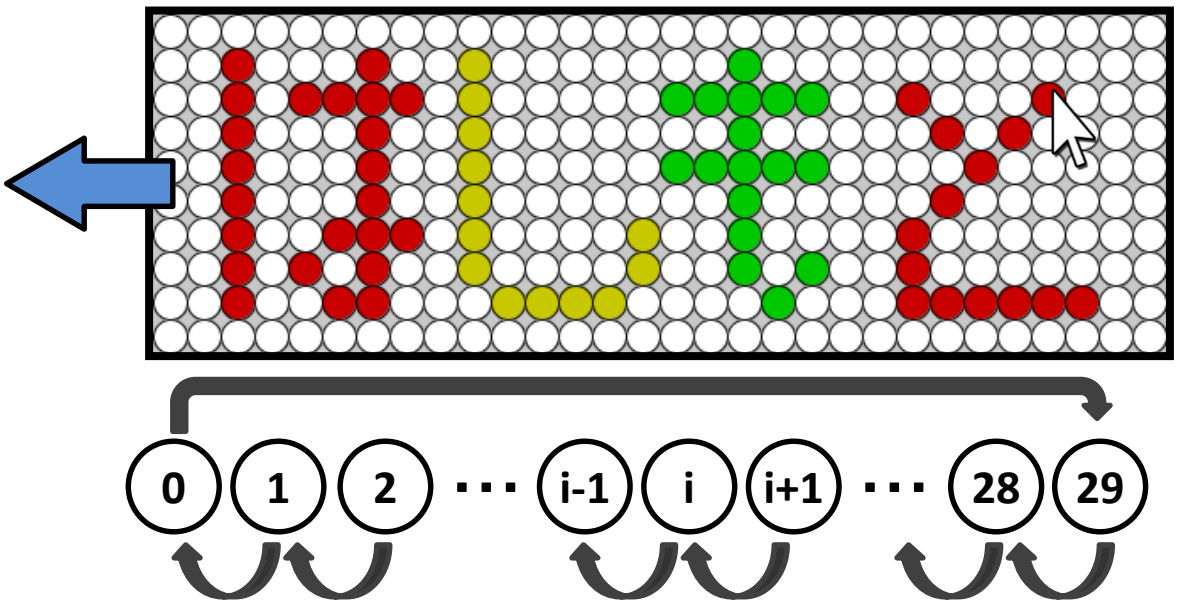
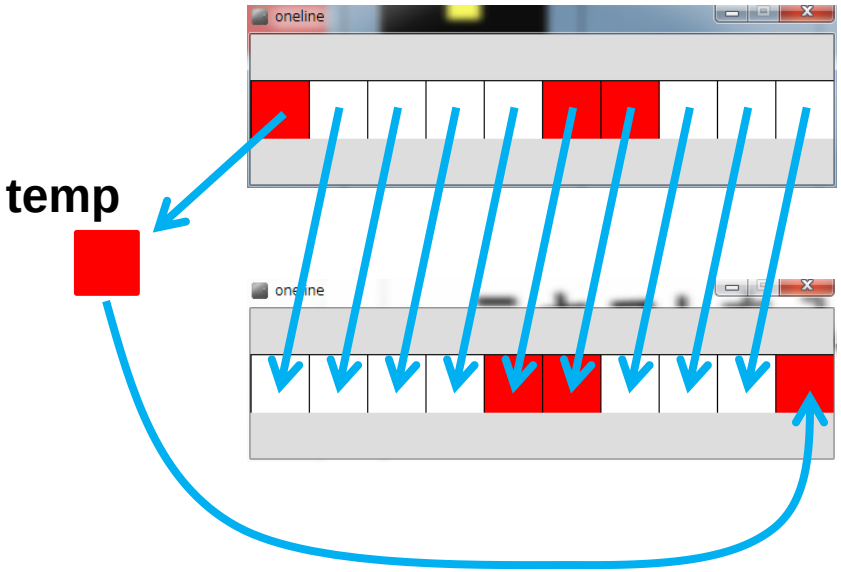
プログラミング演習I（第9回）課題

- **基本①　スケッチ名：keijiban**
  - 直径20の円を **横に30個、縦に10個** 敷き詰めて電光掲示板を作ってください。円をクリックすると、その円の色が変わるようにしてください。
  - クリックするたびに **白→赤→黄→緑→白** と変化させること。
  - さらに、キーボードで【左】方向キーを押したら、左方向に1列円の色が動いていくようにせよ。
  - **左端のものは右端から出てくるようにしてループするようにせよ！！**



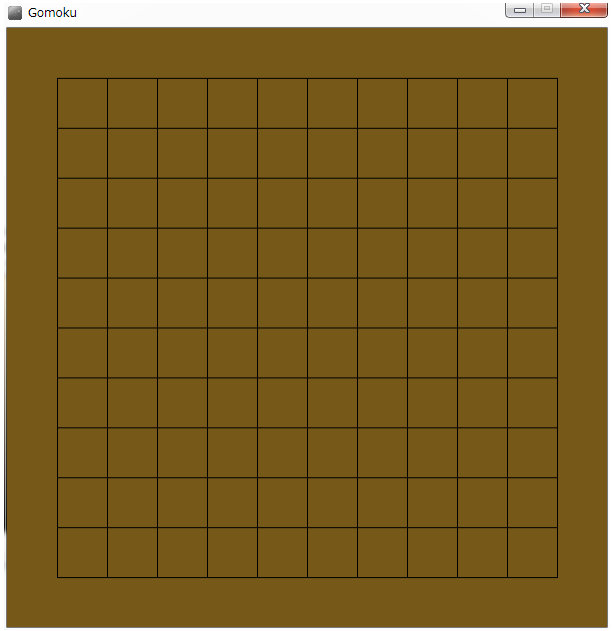
配列の値を循環させる

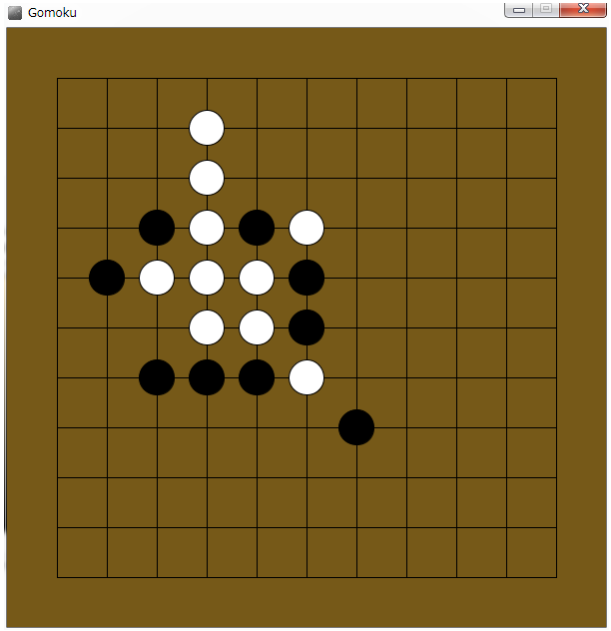
- 考え方
  - 左端の値を，一時的に他の変数に保存しておいて，そこに保存していた値を右端に入れれば良い！
    - temp = status[0];
    - status[0] = status[1];
    - status[1] = status[2];
      :
    - status[7] = status[8];
    - status[8] = status[9];
    - status[9] = temp;



プログラミング演習I（第9回）課題

- **基本②　スケッチ名：Gomoku**
  - 縦横11本の線が入った五目並べの盤面を作り、その格子の近くでマウスのクリックすることによって，白や黒のコマをおくことができるプログラムを作成せよ。
  - ただし、コマは自動で交互に打てるようにすること。
  - すでにコマが置かれている場所は置けないようにすること。

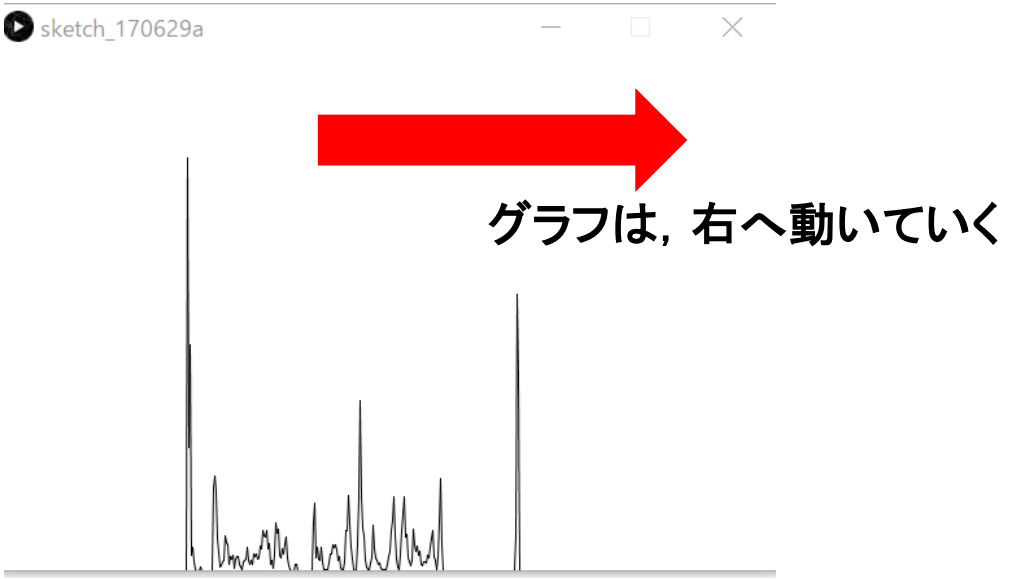




プログラミング演習I（第9回）課題

- **発展①　スケッチ名：mouseGraph**
  - 800x400のサイズのウインドウを作り，800フレーム分の各フレームにおけるマウス移動量をグラフとして表示せよ
  - ここで右端を800フレーム前，左端を現在のフレームとせよ
  - なお，前のフレームから今回のフレームまでのマウスの移動量は下記の命令で取得することができる

    dist(pmouseX, pmouseY, mouseX, mouseY);



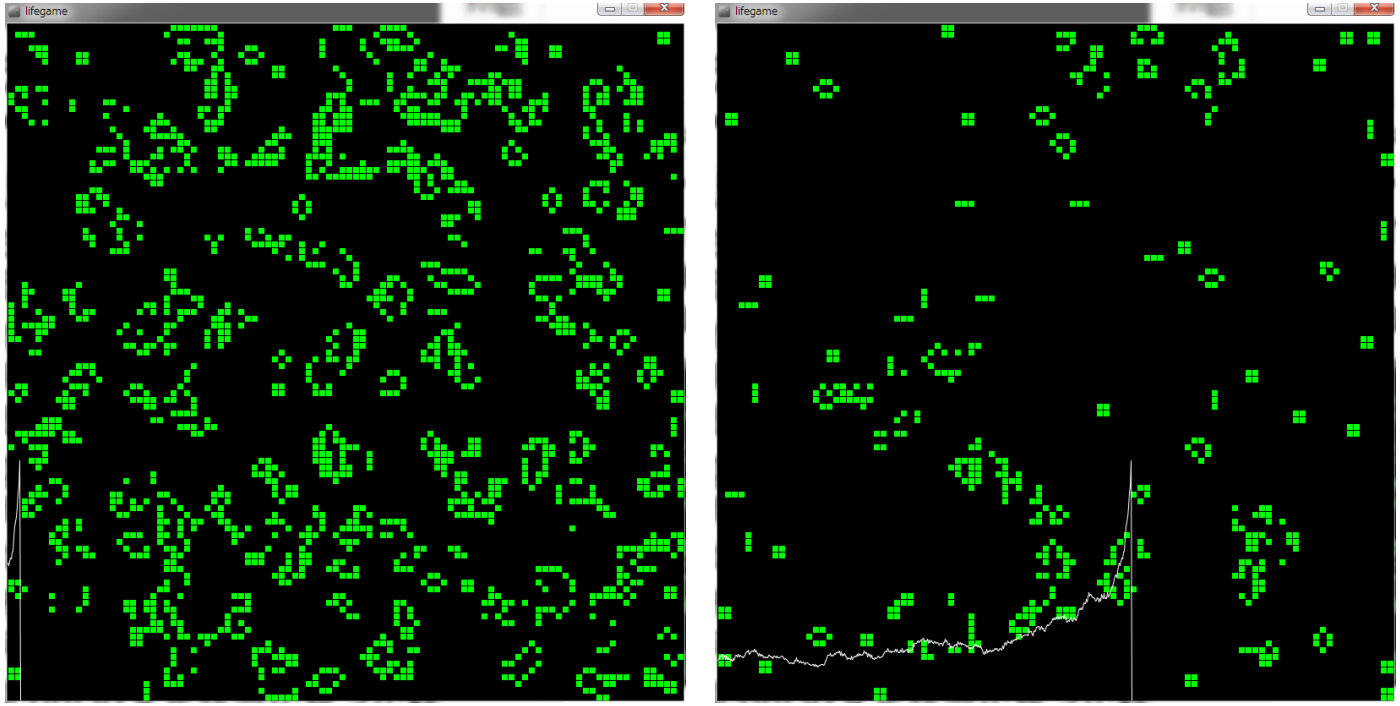
プログラミング演習I（第9回）課題

- **発展②　スケッチ名：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マスが生きているか死んでいるかを数え、その結果に応じてセルを生き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，セルが生き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.pde をそのまま使い，drawの内部を書き換えよ
  - 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

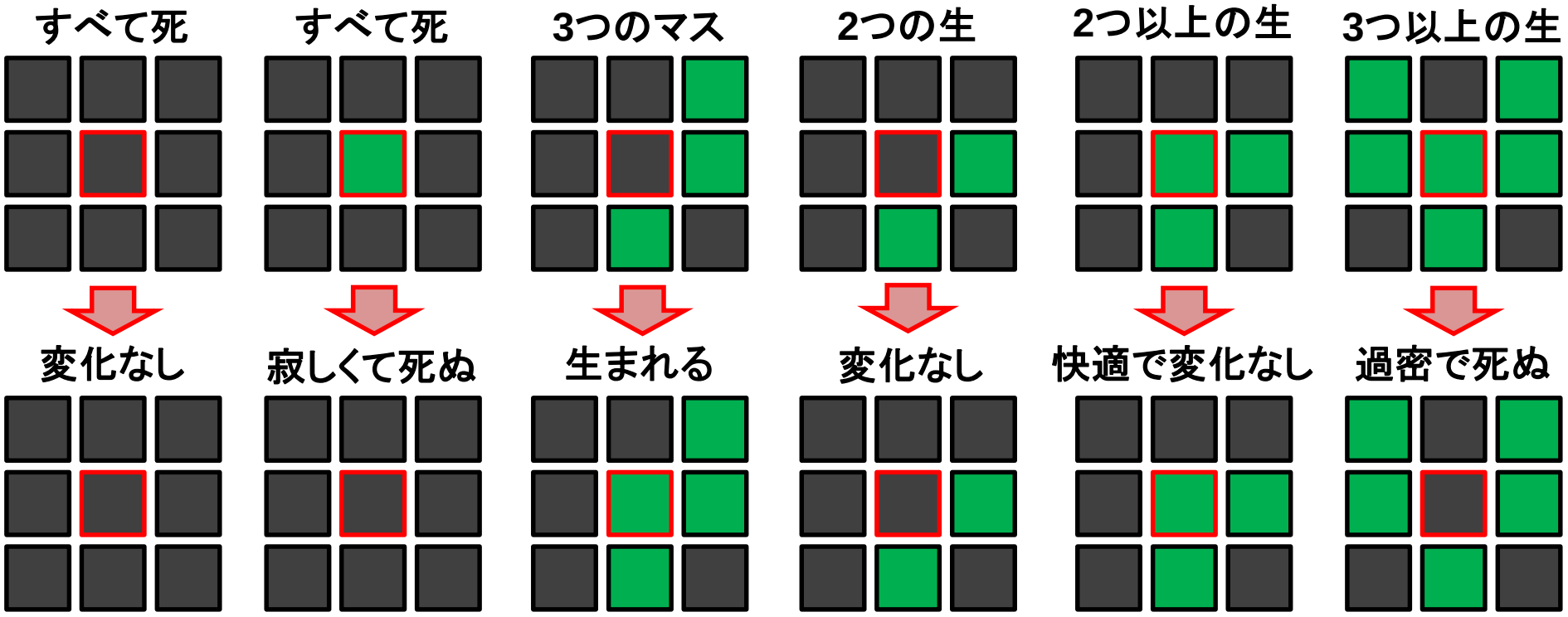
あるマス（赤フレーム）の縦・横・斜めの8マスの生死の状態（生の数）に注目する

【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

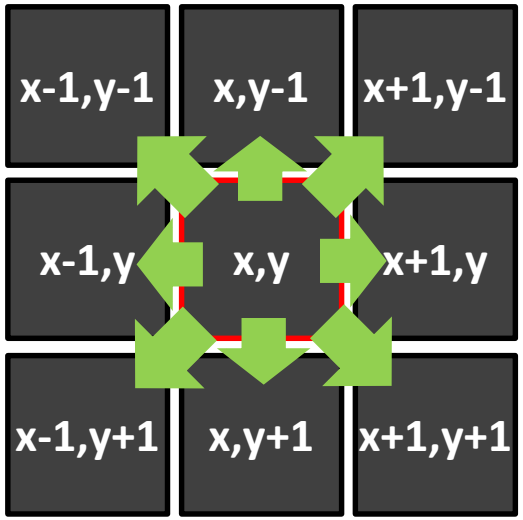


プログラミング演習I（第9回）課題

- 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！