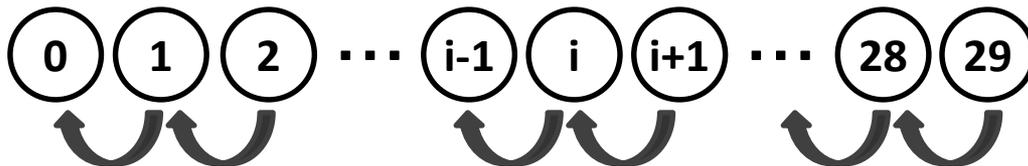
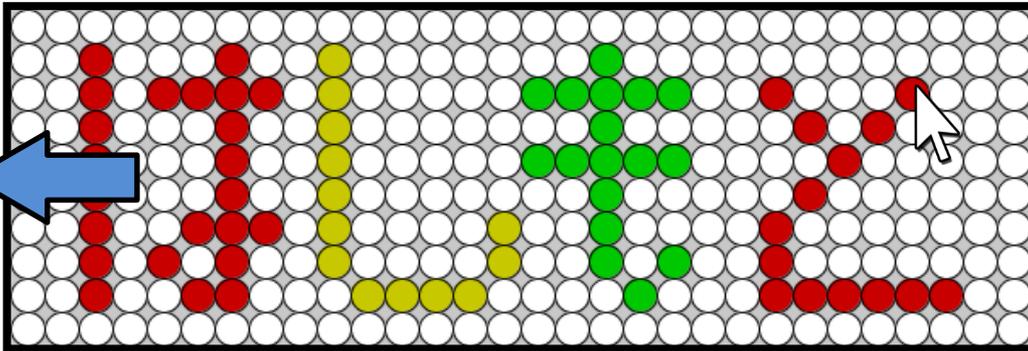


# プログラミング演習I (第9回) 課題

## 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。



### [step1]

まずはウィンドウの中に円を敷き詰めるプログラムを作ってみよう。

### [step2]

各円の状態を保存する2次元配列を作ろう。状態を示す値(フラグ)は、例えば、0だったら白、1だったら赤、2だったら黄など、自分で定義してみよう。

### [step3]

円をクリックしたら、その円のフラグが変わる仕組みを作り、値に基づいて色が塗られるようにしよう。

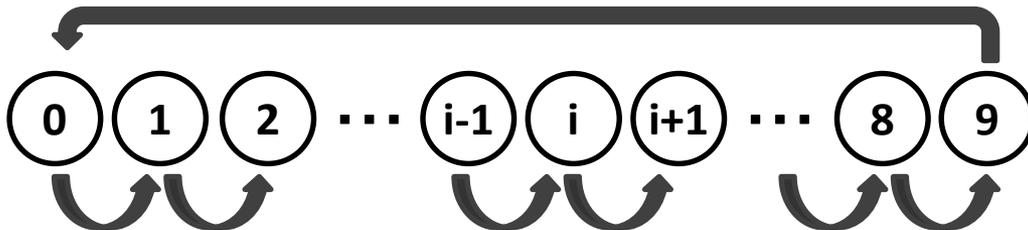
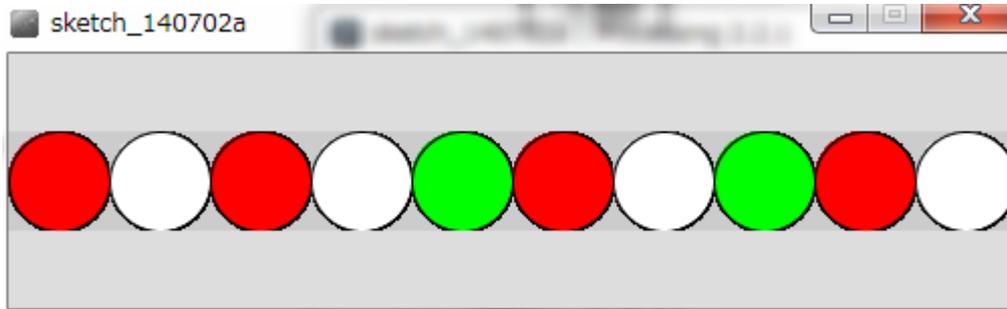
### [step4]

左キーが押されたら配列の値を右から左に移動するようにせよ

# プログラミング演習I (第9回) 課題

## 基本② スケッチ名: oneline2

- 直径50の円を横に10個並べたボードを作成せよ。
- 円の内部をクリックする度に、その円の色が【白→赤→黄→緑→白】と変化させるようにすること。
- また、キーボードの【右】キーを押すと、色が順に左から右へ移動するようにせよ。さらに右端まで来ると左端から出てくるようにせよ。



### [step1]

円を敷き詰め、各円の状態を保存する配列を作ろう。状態を示す値は、各自定義してみよう。

### [step2]

円をクリックしたら、その円の値が変わり、その値に基づき色を塗ろう。

### [step3]

右キーを押した時に値が左から右に移動されるようにしよう。

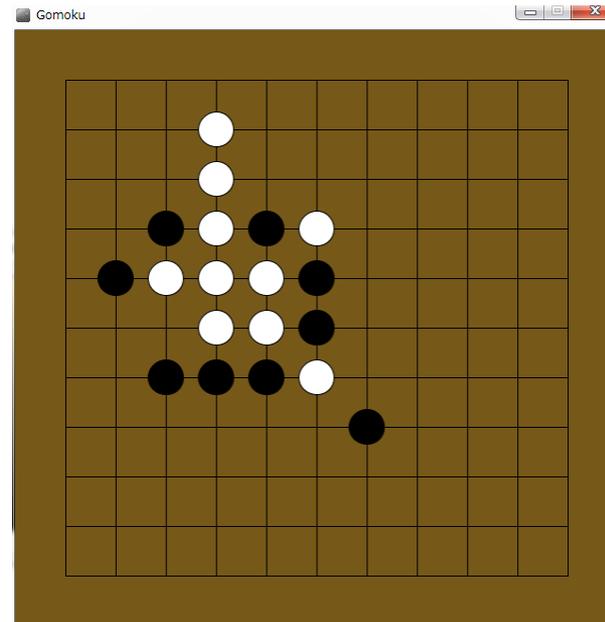
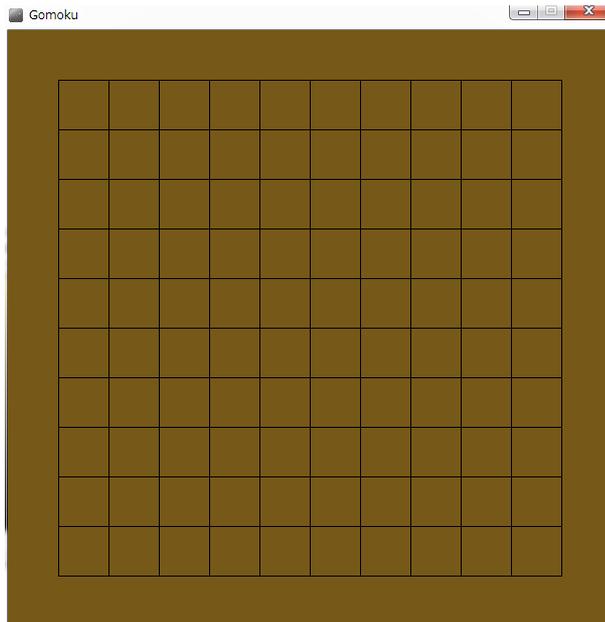
### [step4]

右端の値を左端に代入するようにしよう。値を代入するにはどうするか？

# プログラミング演習I (第9回) 課題

## • 基本③ 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。
- 余裕がある人は、勝敗の判定までやってみよう！



# 手順

---

1. 盤面を描いてみよう
2. 盤面のマス目を管理する配列を用意しよう
3. 配列の値に応じてコマを表示しよう
  - 配列の値を初期化しておこう
  - 配列に適切な値を代入し, 0なら表示しない, 1なら白, 2なら黒を表示
4. 盤面のクリックに応じて配列の値を変化させよう
  - 盤面のどの部分がクリックされかをまず計算しよう
  - その盤面の配列の値を変化させよう
5. 白いコマと黒いコマの数を数え, その数を表示させよう

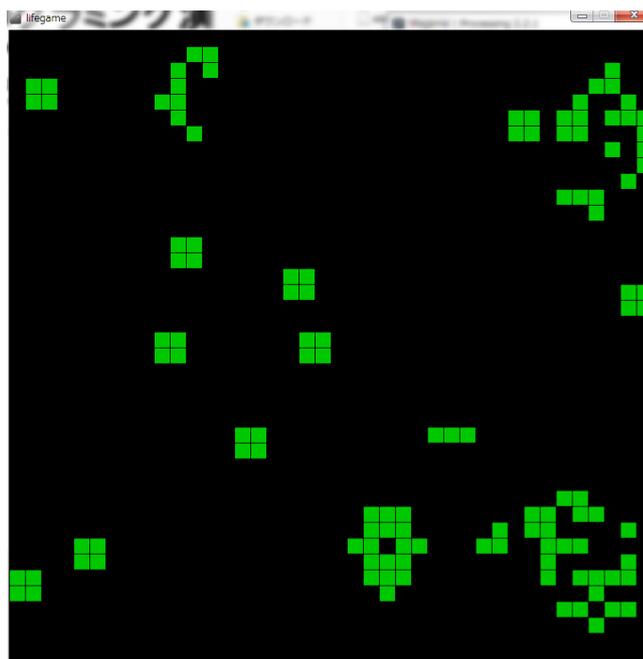
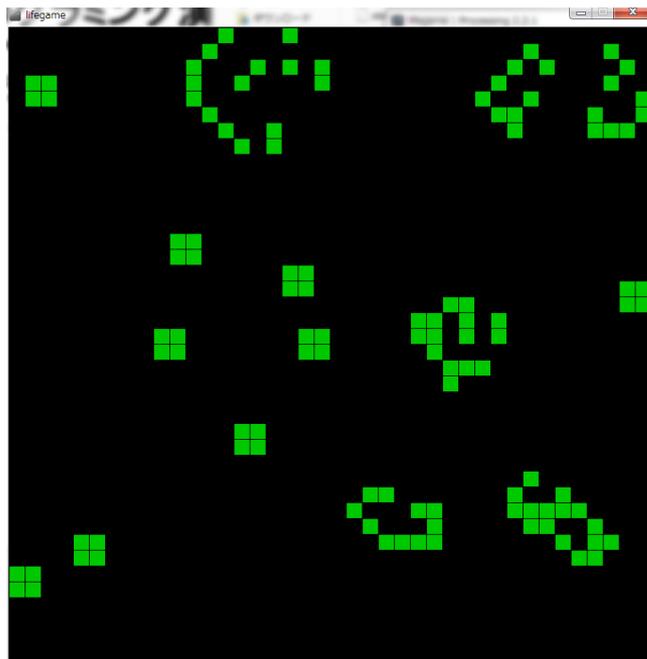
# プログラミング演習I (第9回) 課題

## • 発展① 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。
- 80x80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 lifegame1.pde や lifegame2.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

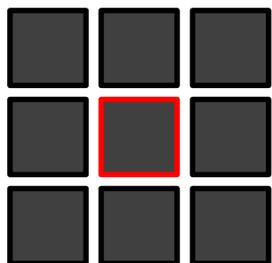
**【誕生】** 死んでいるセルに隣接する生きたセルがちょうど3つならば次世代が誕生

**【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

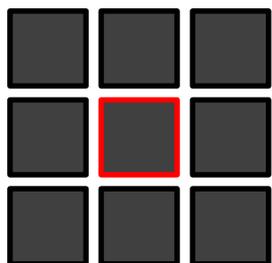
**【過疎】** 生きているセルに隣接する生きたセルが1つ以下ならば過疎により死滅

**【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

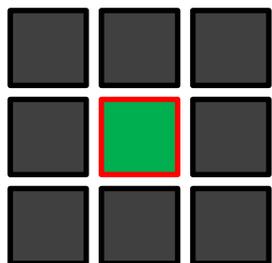
すべて死



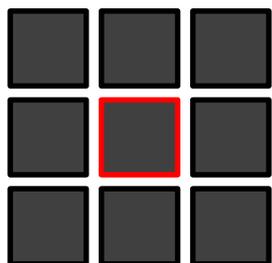
変化なし



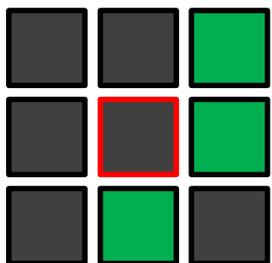
すべて死



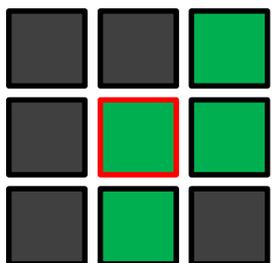
寂しくて死ぬ



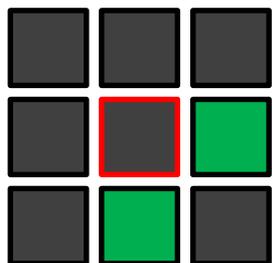
3つのマス



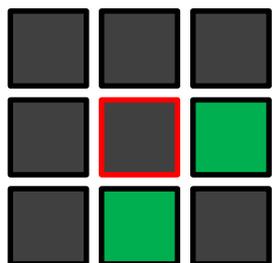
生まれる



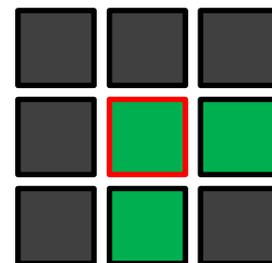
2つの生



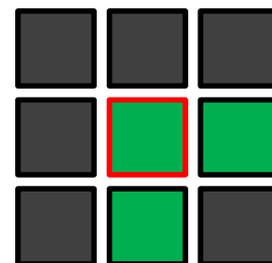
快適で変化なし



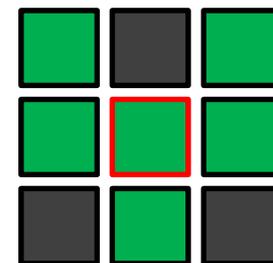
2つ以上の生



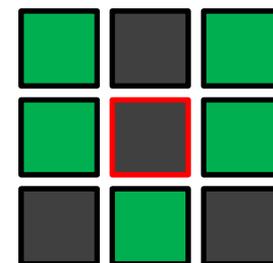
快適で変化なし



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

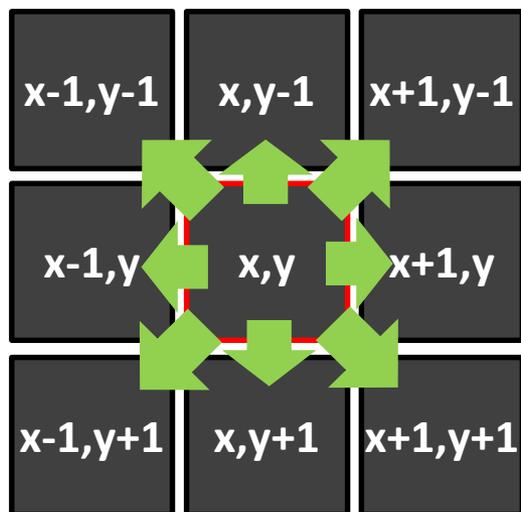
- ルールを整理すると...

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

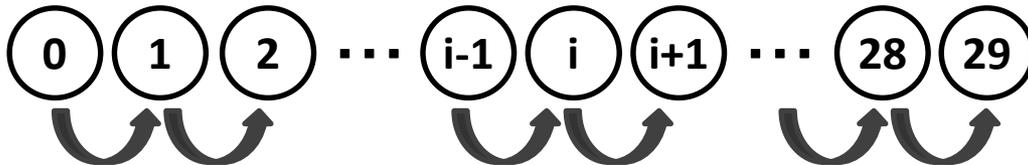
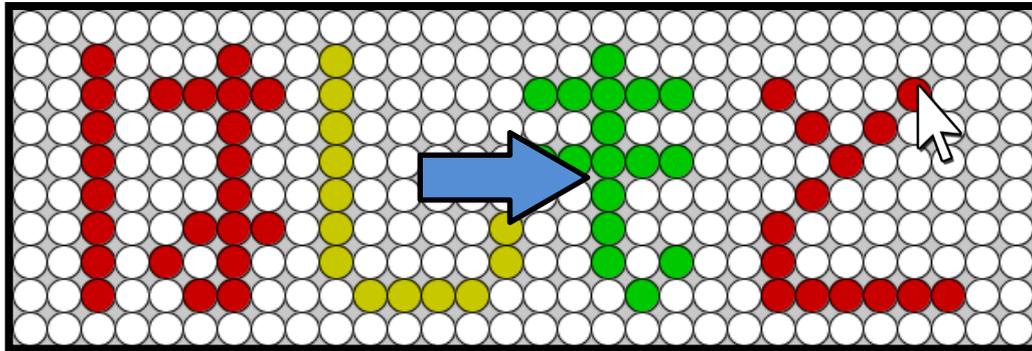
0,0	1,0	2,0
0,1	1,1	2,1
0,2	1,2	2,2

表示しない外周を用意して、周囲の「生」の数を数えると楽！

# 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【右】方向キーを押したら、右方向に1列円の色が動いていくようにせよ。



### [step1]

まずはウィンドウの中に円を敷き詰めるプログラムを作ってみよう。

### [step2]

各円の状態を保存する2次元配列を作ろう。状態を示す値(フラグ)は、例えば、0だったら白、1だったら赤、2だったら黄など、自分で定義してみよう。

### [step3]

円をクリックしたら、その円のフラグが変わる仕組みを作り、値に基づいて色が塗られるようにしよう。

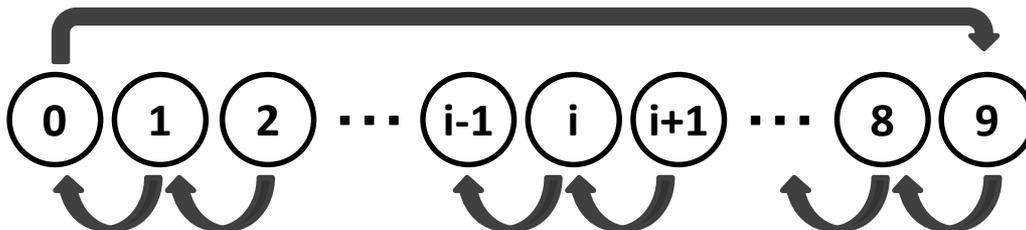
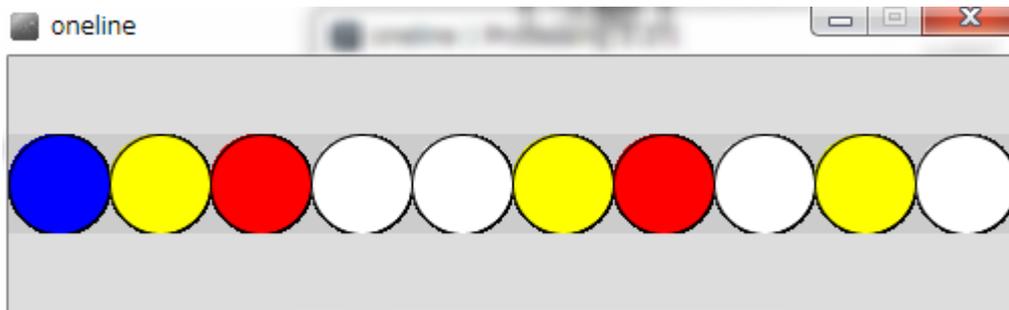
### [step4]

右キーが押されたら配列の値を左から右に移動するようにせよ

# プログラミング演習I (第9回) 課題

## 基本② スケッチ名: oneline2

- 直径50の円を横に10個並べたボードを作成せよ。
- 円の内部をクリックする度に、その円の色が【白→青→黄→赤→白】と変化させるようにすること。
- また、キーボードの【左】キーを押すと、色が順に右から左へ移動するようにせよ。さらに左端まで来ると右端から出てくるようにせよ。



[step1]

円を敷き詰め、各円の状態を保存する配列を作ろう。状態を示す値は、各自定義してみよう。

[step2]

円をクリックしたら、その円の値が変わり、その値に基づき色を塗ろう。

[step3]

左キーを押した時に値が右から左に移動されるようにしよう。

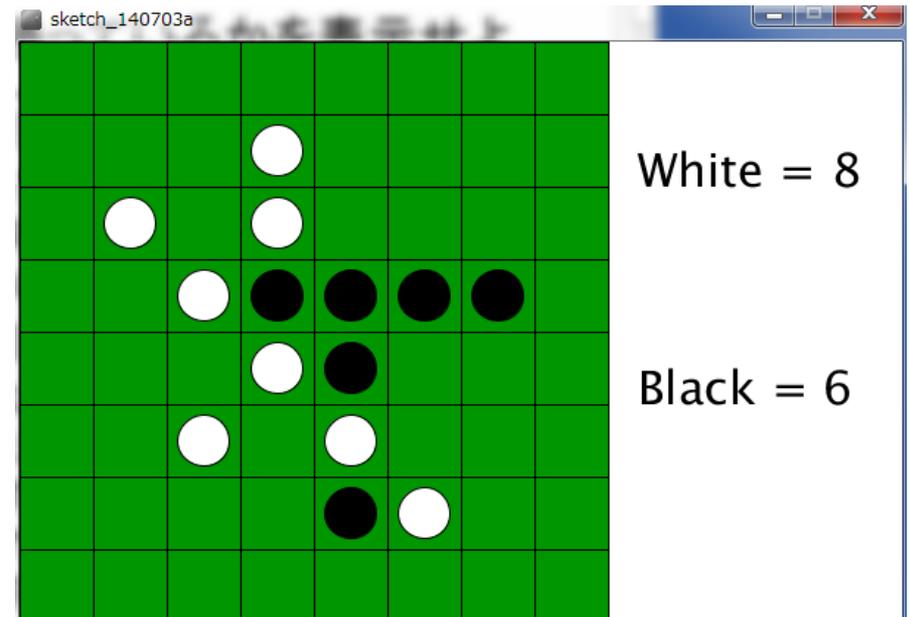
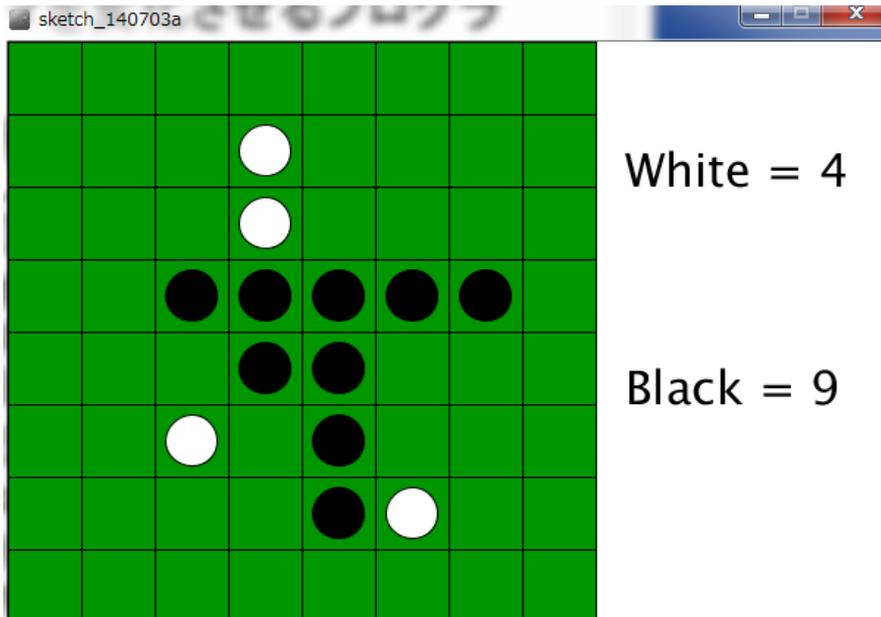
[step4]

左端の値を右端に代入するようにしよう。値を代入するにはどうするか？

# プログラミング演習I (第9回) 課題

## • 基本③ スケッチ名 : Othello

- 横8マス、縦8マスのオセロの盤面を作り、その上にマウスのクリックによってオセロがない状態、オセロの黒いコマ、白いコマを変化させるプログラムを作成せよ。
- また、常に白と黒のコマの数を表示せよ。
- 余裕がある人はターン管理をしたり、コマを自動で反転したりするようにしてみよう。



# 手順

---

1. 盤面を描いてみよう
2. 盤面のマス目を管理する配列を用意しよう
3. 配列の値に応じてコマを表示しよう
  - 配列の値を初期化しておこう
  - 配列に適当な値を代入し, 0なら表示しない, 1なら白, 2なら黒を表示
4. 盤面のクリック場所に応じて, どの配列に値を代入するかを考えよう
5. ターンという考え方(変数)を導入して, そのターンに応じて配列に値を代入しよう
6. すでに置かれている場所に置かないようにするにはどうしたら良いだろうか? すでに何か置かれていることの判定はどうしたらよいだろうか?

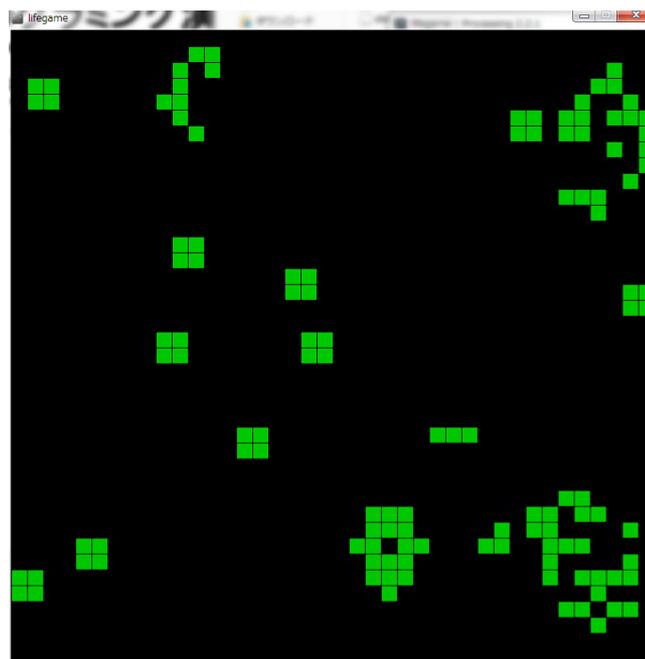
# プログラミング演習I (第9回) 課題

## • 発展① 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。
- 160x160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 lifegame1.pde や lifegame2.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

すべて死

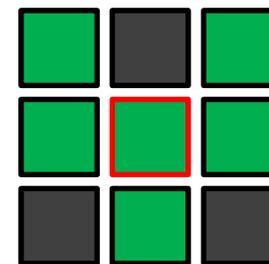
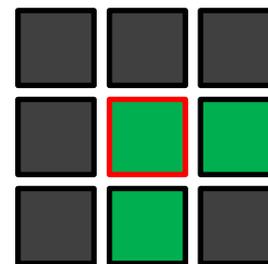
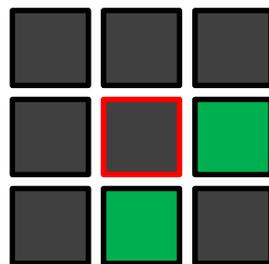
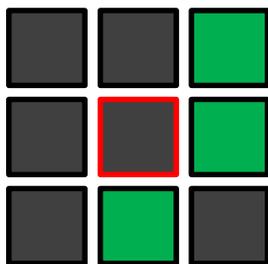
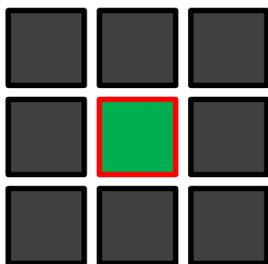
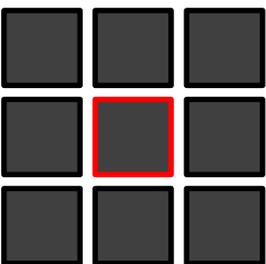
すべて死

3つのマス

2つの生

2つ以上の生

3つ以上の生



変化なし

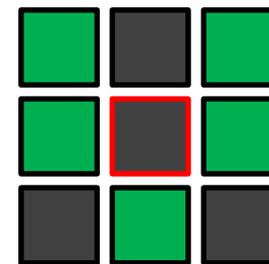
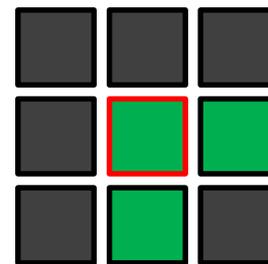
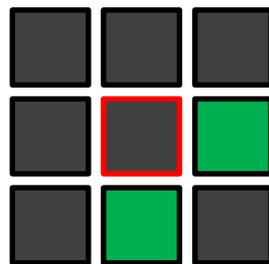
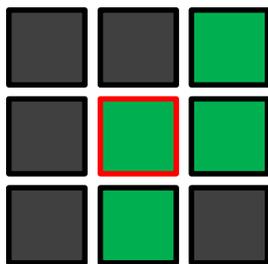
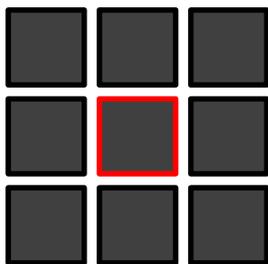
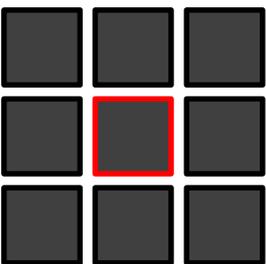
寂しくて死ぬ

生まれる

快適で変化なし

快適で変化なし

過密で死ぬ



# プログラミング演習I (第9回) 課題

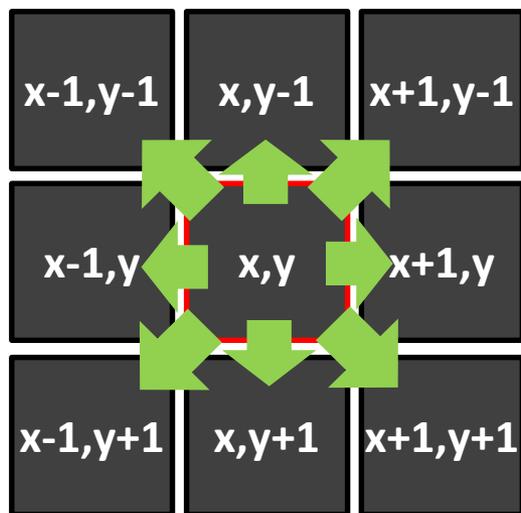
- ルールを整理すると...

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！