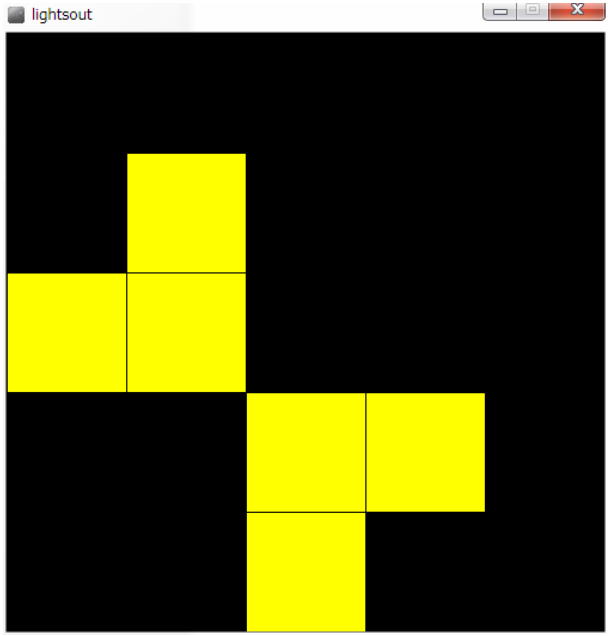
プログラミング演習I （第9回） 課題

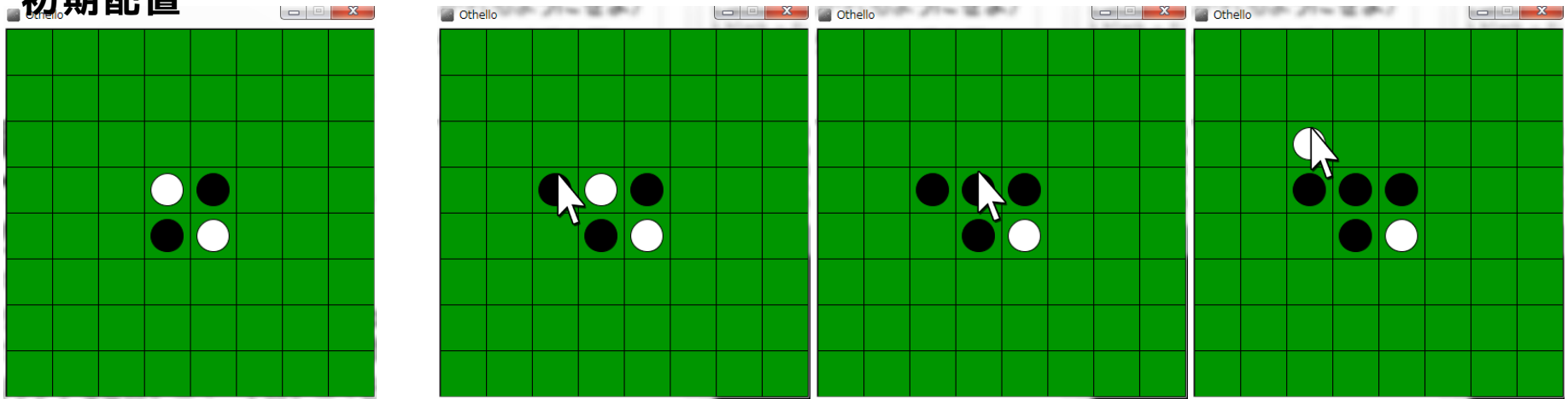
- **基本①　スケッチ名：basic_LightsOut**
  - 横5マス、縦5マスの盤面を作り、そのマス目上をクリックすると、クリックされたマス目の上下左右とそのマス目自体の色を反転させるLights Outを作れ
  - なお、すべてが黒色になったらCLEAR!と表示するようにせよ
  - **ただし、起動したタイミングで下記のような表示になっているようにせよ（予習した人はこれをやるだけ！）**



プログラミング演習I （第9回） 課題

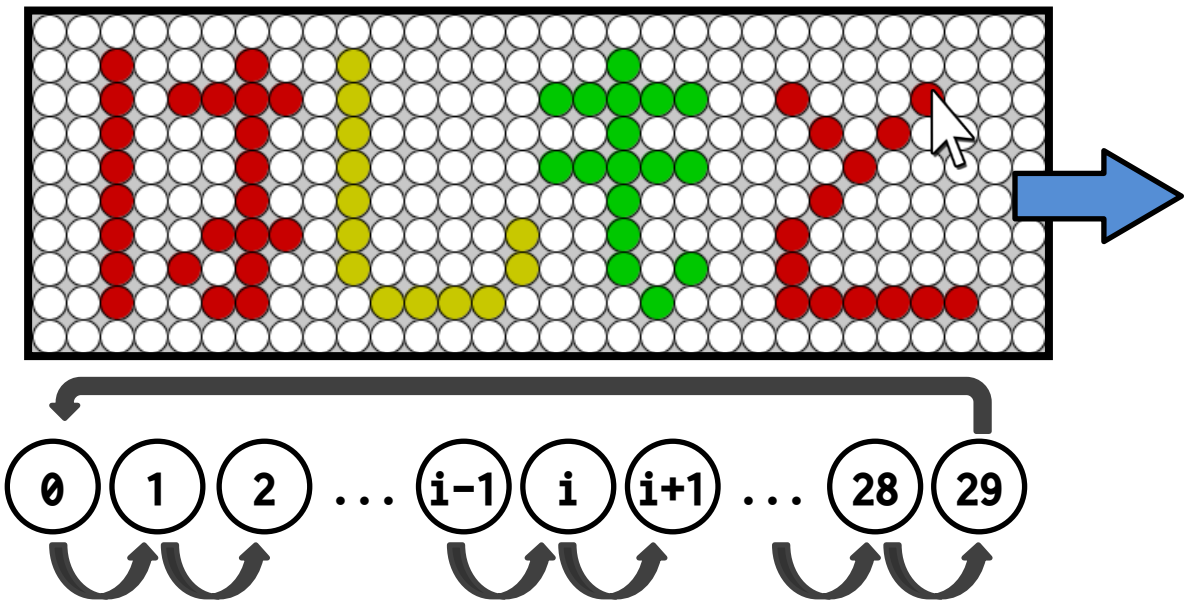
- **基本②　スケッチ名：basic_Othello**
  - 横8マス、縦8マスのオセロの盤面と左下のコマの初期配置を作れ
  - コマがないマスをクリックすると、ターンに応じて白いコマまたは黒いコマが置かれるようにせよ（白いコマ、黒いコマは交互に置かれるようにせよ）
    - ターンを管理する変数を用意して，あきマスに置かれたら値を変更する！
  - また、黒いコマをクリックすると白いコマへ、白いコマをクリックすると黒いコマへ変わるようにせよ
  - 余裕がある人はWhiteやBlackのコマの数を表示したり、コマを自動で反転したりするようにしてみよう。

初期配置



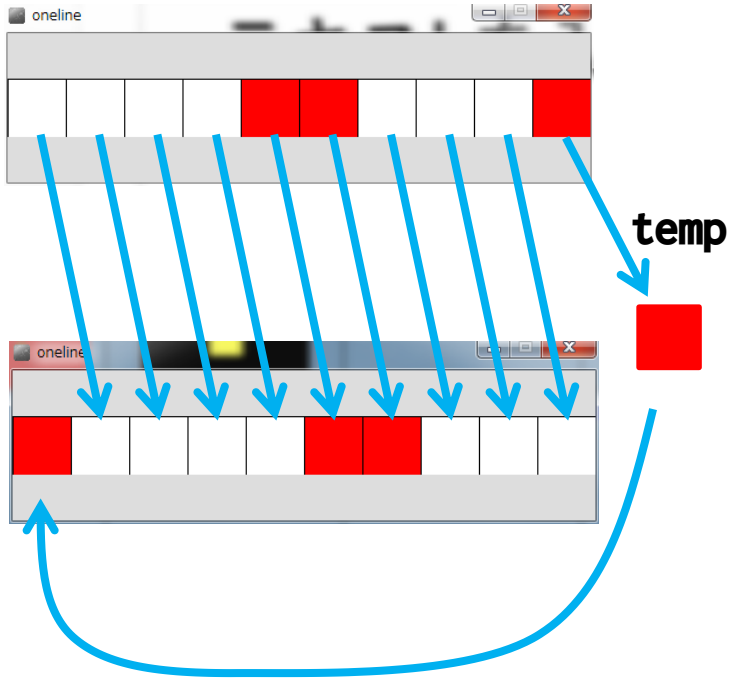
プログラミング演習I （第9回） 課題

- **基本③　スケッチ名：basic_Keijiban**
  - 直径30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** と繰り返し変化させること。
  - 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。
  - **右端のものは左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

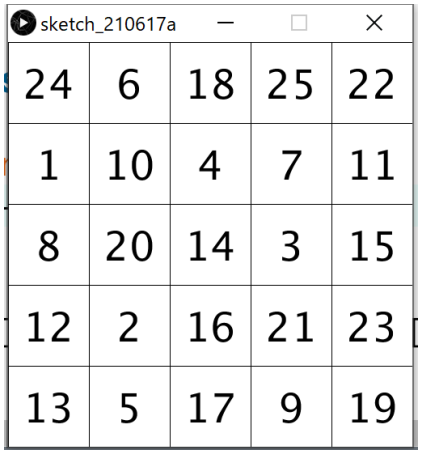
- 考え方
  - 右端の値を，一時的に他の変数に保存しておいて，そこに保存していた値を左端に入れれば良い！
    - `temp = status[9];`
    - `status[9] = status[8];`
    - `status[8] = status[7];`
    - `:`
    - `status[2] = status[1];`
    - `status[1] = status[0];`
    - `status[0] = temp;`



プログラミング演習I （第9回） 課題

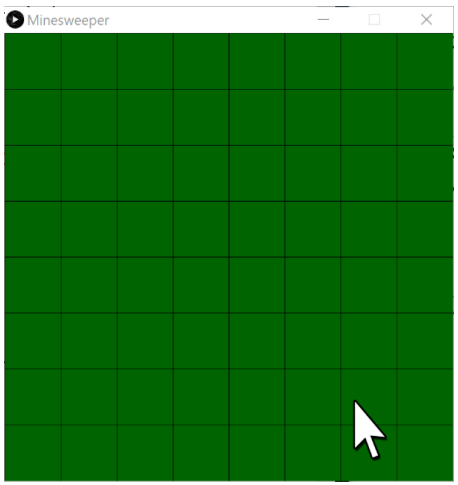
- **発展①　スケッチ名：advanced_Bingo**
  - 500x500のウインドウ内に横5マス、縦5マスのビンゴの盤面を作れ
  - 5x5のマス目に1から25までの数字を重複なくランダムに配置せよ
  - ウインドウ内でマウスをクリックするたびに、1から25までの数字をランダムに取得せよ（以前と同じ数字が出ても良い）
  - 選ばれた数字を標準出力するとともに、マス目を灰色にせよ

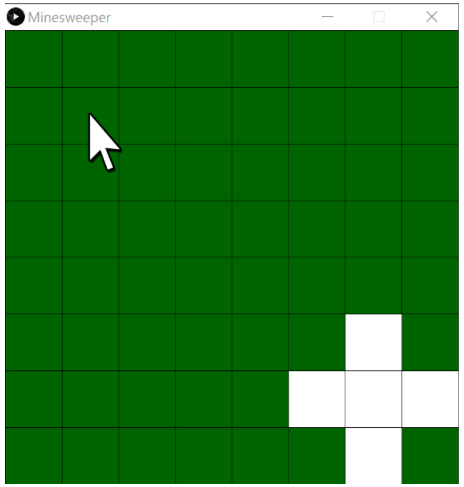
  - 以下は条件ではないが、チャレンジできる人は是非
    - 1から25までの数字がランダムに取得されるとき、重複しないようにせよ
    - ビンゴの条件がそろったらビンゴと表示したり、ビンゴが揃った場所を別の色で塗りつぶしてみよう！



プログラミング演習I （第9回） 課題

- **発展②　スケッチ名：advanced_MinesweeperSimple**
  - ウインドウ幅800x800に、8×8のマス目を作成し、すべてのマス目を緑色にせよ
  - またそのマス目にランダムに1個爆弾を配置せよ（どこに爆弾があるかはわからないようにせよ）
  - マス目をクリックしたときに、マス目に爆弾があった場合はゲーム終了とし，マス目に爆弾がなかった場合はそのマス目を白色にせよ
  - その上下左右4マス（端の場合は数が減る）すべてに爆弾がない場合は、その4マスを白色にせよ．上下左右のいずれかに爆弾が含まれる場合は何もしないようにせよ（これがヒントとなる）
  - 全てマス目を白色＋開けていない爆弾だけにすることができたらクリアと表示せよ
  - ゲーム終了と，クリアについては標準出力するだけでも良い
  - クリアの表示をした後に、クリックしても、反応してしまってもOKとして良い





プログラミング演習I （第9回） 課題

- ヒント
  - ベースの部分はオセロを流用する
  - 上下左右に開くことについてはLightsOutの仕組みを流用する
  - コマの状態として考えられるのは下記の3状態
    - 開けられている
    - 開けられていない＆爆弾なし
    - 開けられていない＆爆弾あり
  - 処理として工夫するのは，上下左右に爆弾が含まれるときには上下左右を開かず，爆弾が含まれないときには上下左右を開くということ
  - また，爆弾を残して全てを開けたときにゲームクリアと表示すること