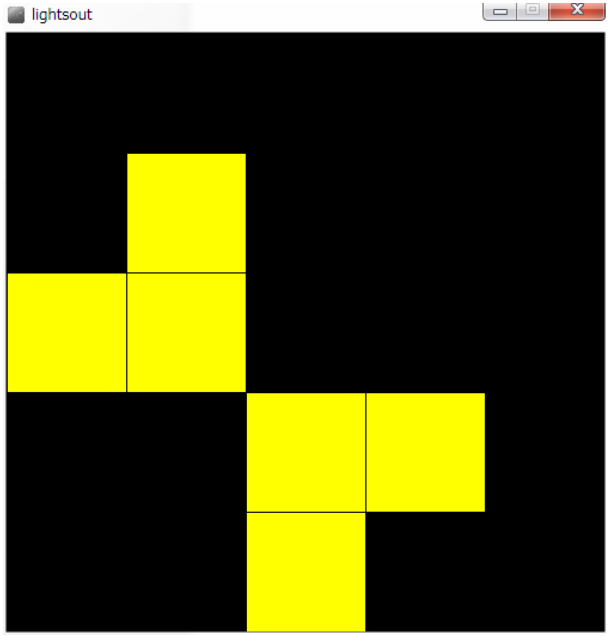
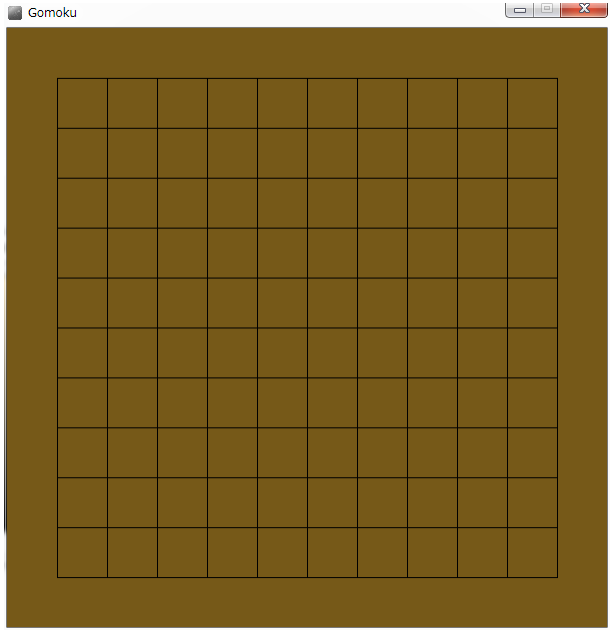
プログラミング演習I （第9回） 課題

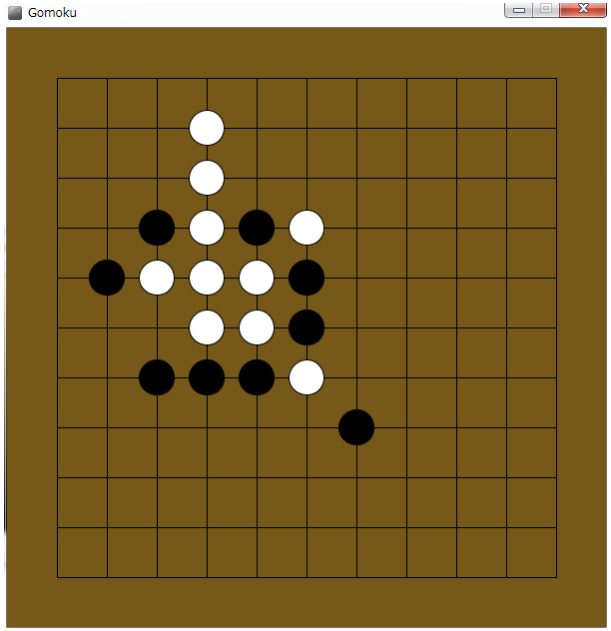
- **基本①　スケッチ名：basic_LightsOut**
  - 横5マス、縦5マスの盤面を作り、そのマス目上をクリックすると、クリックされたマス目の上下左右とそのマス目自体の色を反転させるLights Outを作れ
  - なお、すべてが黒色になったらCLEAR!と表示するようにせよ
  - **ただし、起動したタイミングで下記のような表示になっているようにせよ（予習した人はこれをやるだけ！）**



プログラミング演習I （第9回） 課題

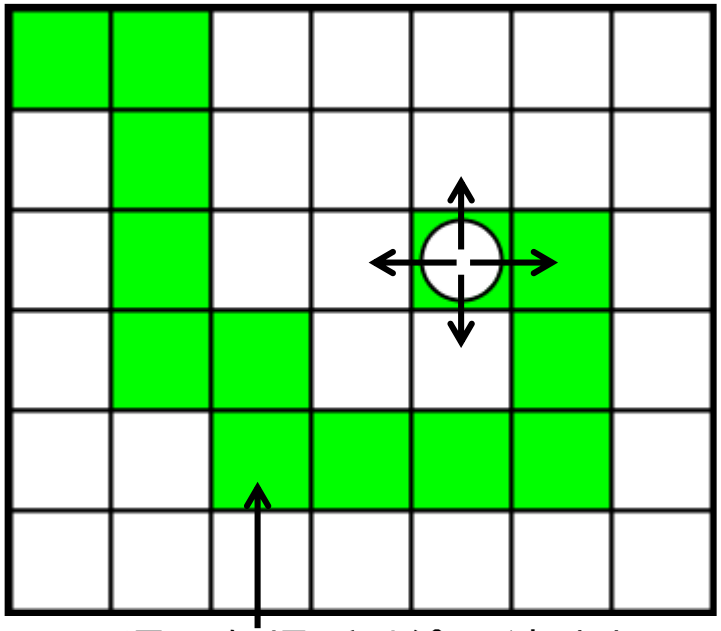
- **基本②　スケッチ名：basic_Gomoku**
  - 縦横11本の線が入った五目並べの盤面を作り、その格子の近くでマウスのクリックすることによって，白や黒のコマをおくことができるプログラムを作成せよ。
  - ただし、コマは自動で交互に打てるようにすること。
  - すでにコマが置かれている場所は置けないようにすること。





プログラミング演習I （第9回） 課題

- **基本③　スケッチ名：basic_Boardgame**
  - **横7マス、縦6マス** の盤面を作り、そのマス目上を方向キーの操作で円（またはキャラ）が移動するプログラムを作ってください。
  - ただし、すでに通ったマス目は色が変わるようにしてください。
  - 開始位置は左上として下さい
  - 画面の外に出ていこうとしたら（今回の課題的には）プログラムが落ちてもよいが，できるだけ落ちないようにせよ



既に通った場所は塗り潰される
（現在自分がいるマスも塗ること）

ヒント

縦横にマス目を作り、そこに方向キーで移動する円を作ろう。

マスに円が入ったかどうかのフラグ（例えば入ったら１、まだなら０）を格納する２次元配列を作ろう。円がいるマスのフラグを変更する処理はどう書いたらいいだろうか？

フラグの情報に基づいて、四角の色を塗り分ける処理を加えよう。

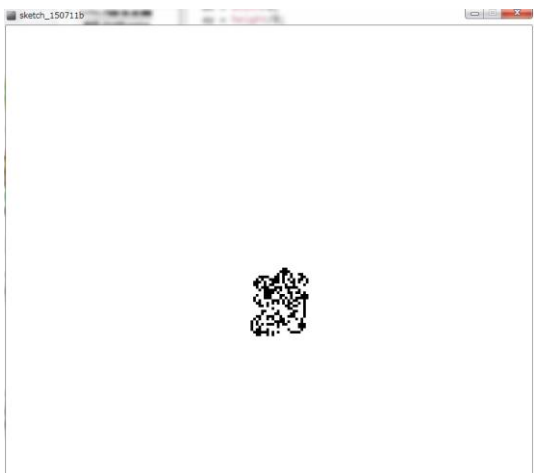
ヒント：up/down/left/rightで移動

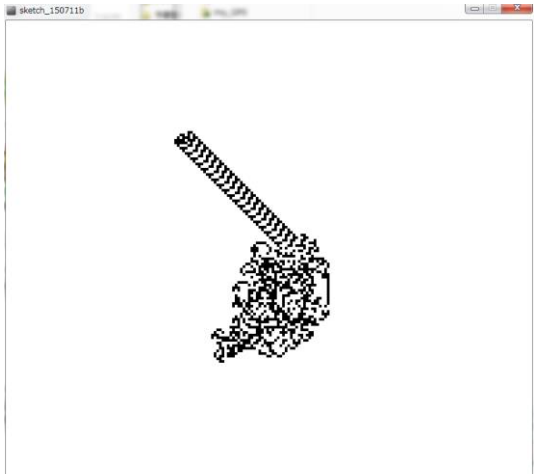
```
void setup()
{
  size(700, 600);
}

void draw()
{
}
```

```
void keyPressed()
{
  if(keyCode == UP)
  {
    println("上が押された");
  }
  else if(keyCode == DOWN)
  {
    println("下が押された");
  }
  else if(keyCode == LEFT)
  {
    println("左が押された");
  }
  else if(keyCode == RIGHT)
  {
    println("右が押された");
  }
}
```

プログラミング演習I （第9回） 課題

- **発展①　スケッチ名：advanced_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



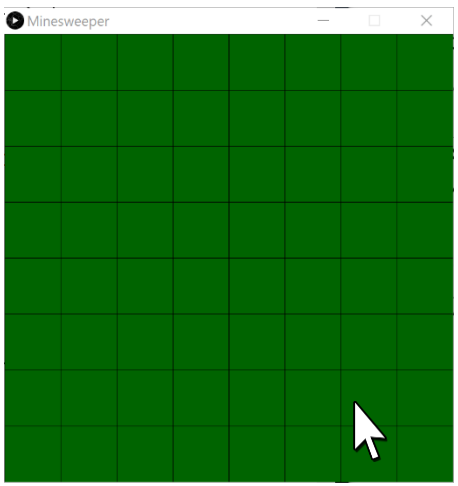


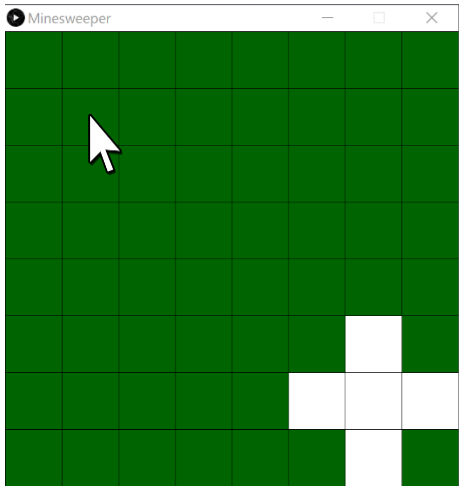
ヒント

- フィールドの白黒の情報を配列でもっておく
- アリの座標は？
- アリの向いている方角は？
- アリは次にどちらに動く？

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、横8マス、縦8マスのマス目を作成し、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





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