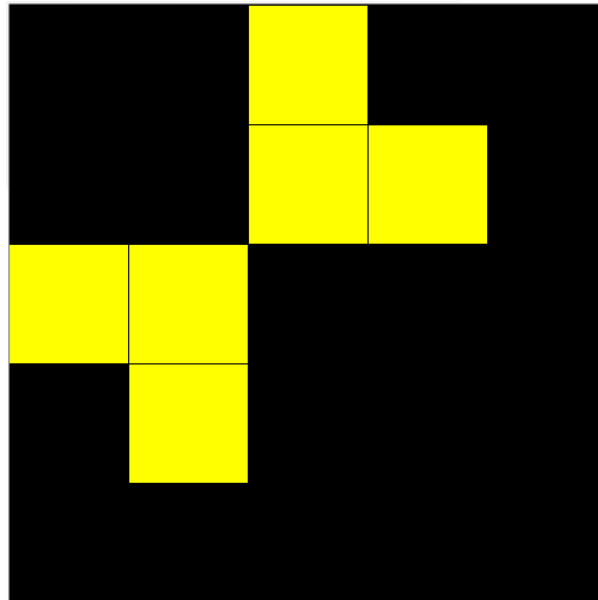


プログラミング演習I (第9回) 課題

• 基本① スケッチ名：**basic_LightsOut**

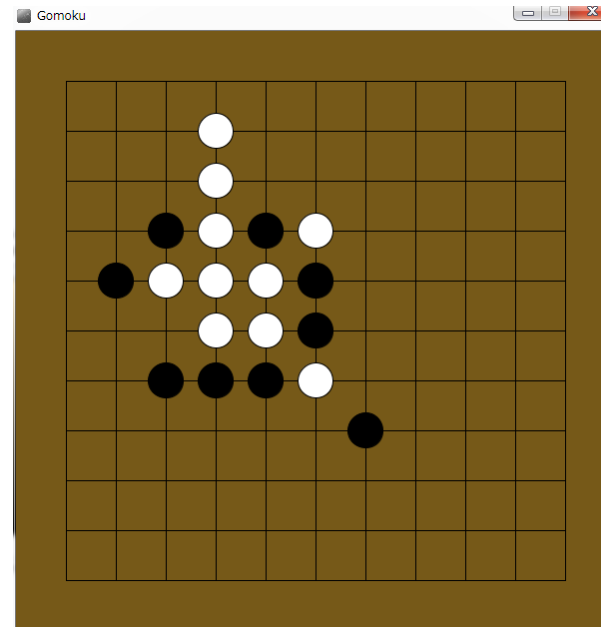
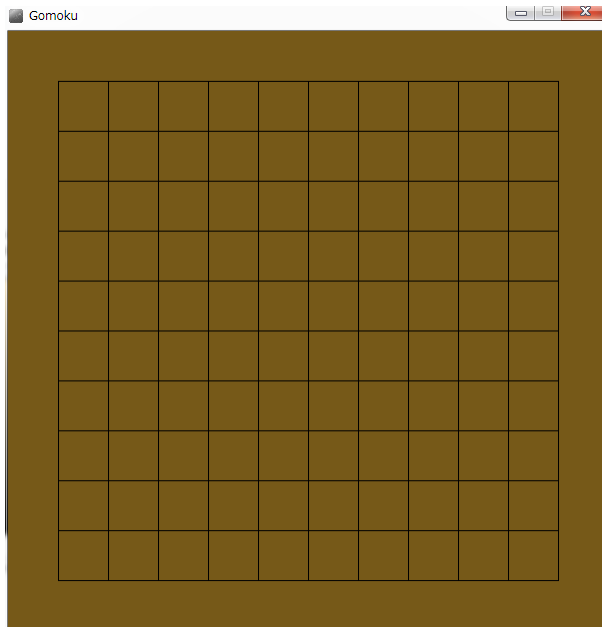
- 横5マス、縦5マスの盤面を作り、そのマス目をクリックすると、クリックされたマス目の上下左右とそのマス目自体の色を反転させるLights Outを作れ
- なお、すべてが黒色になったらCLEAR!と表示するようにせよ
- ただし、起動したタイミングで下記のような表示になっているようにせよ (予習した人はこれをやるだけ!)



プログラミング演習I (第9回) 課題

• 基本② スケッチ名：**basic_Gomoku**

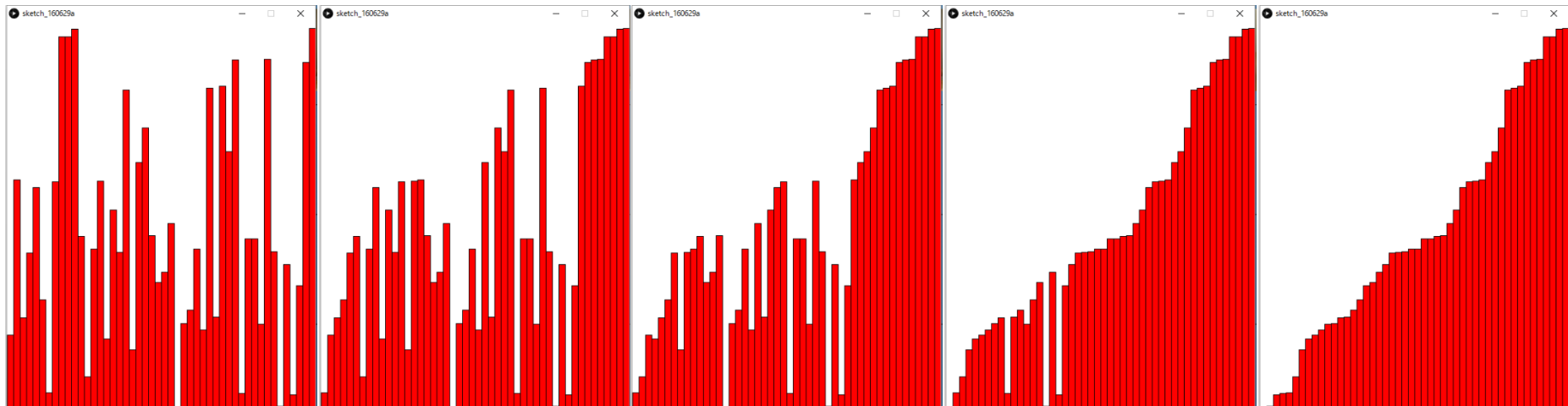
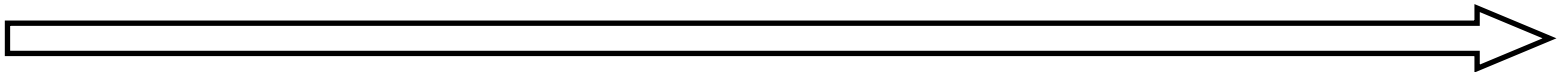
- 縦横11本の線が入った五目並べの盤面を作り、その格子の近くでマウスのクリックすることによって、白や黒のコマをおくことができるプログラムを作成せよ。
- ただし、コマは自動で交互に打てるようにすること。
- すでにコマが置かれている場所は置けないようにすること。



プログラミング演習I (第9回) 課題

• 基本③ basic_SortChart

- 480x600のウィンドウを作成し, 48個の配列の中に起動時に0~600の値をランダムに設定せよ
- また, その配列の中の値に応じてチャートを表示せよ
- さらに, マウスがクリックされる度にその配列の値の左右 (i 番目と $i+1$ 番目の値) を比較し, 左側の値のほうが大きい場合 (i 番目の値が $i+1$ 番目の値より大きい場合) に, i 番目と $i+1$ 番目の値を入れ替えるようにせよ。なお, i は0から順に増やして処理せよ
- 実行結果は下記のようにクリックのたびに階段状に変化していく



a, bの値を入れ替えるにはどうする？

aとbの関係を考える

a = b;

b = a;

とすると、1行目でaがbに値で上書きされるので、aの元の値をbに代入できない

temp変数を作る

tempに一度値を入れて
入れ替える

```
int temp;  
temp = a;  
a = b;  
b = temp;
```

配列の値を入れ替えるにはどうする？

i番目とi+1番目の関係を考えtemp変数を導入

tempに一度値を入れて
入れ替える

```
int temp;  
temp = num[i];  
num[i] = num[i+1];  
num[i+1] = temp;
```

ソートを行う場合は、forループで繰り返しを行いつつ、temp変数を利用して入れ替え！

```
for( int i=0; i<num.length; i++ ){  
    if( num[i] と num[i+1] を比較 ){  
        int temp = num[i];  
        num[i] = num[i+1];  
        num[i+1] = 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_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）
- 上記プログラム時計の針を1フレームとみたとて完成させて、ラングトンのアリが脱出する様子を観測せよ
- 脱出直後に落ちてもよい
- 800x800のウィンドウに100x100マスで実現せよ
- アリは赤色で示しましょう



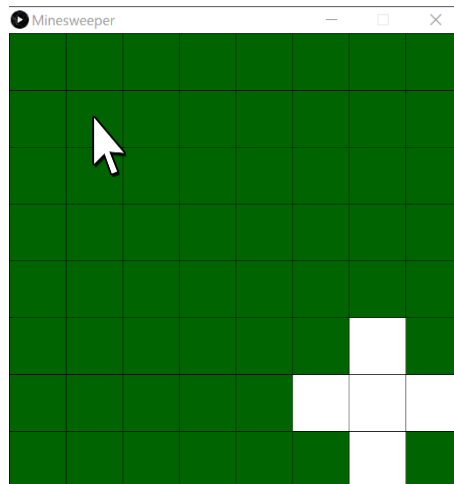
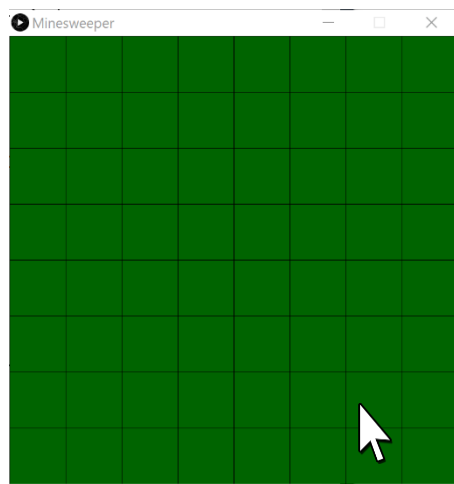
ヒント

- 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

- ウィンドウ幅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の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

