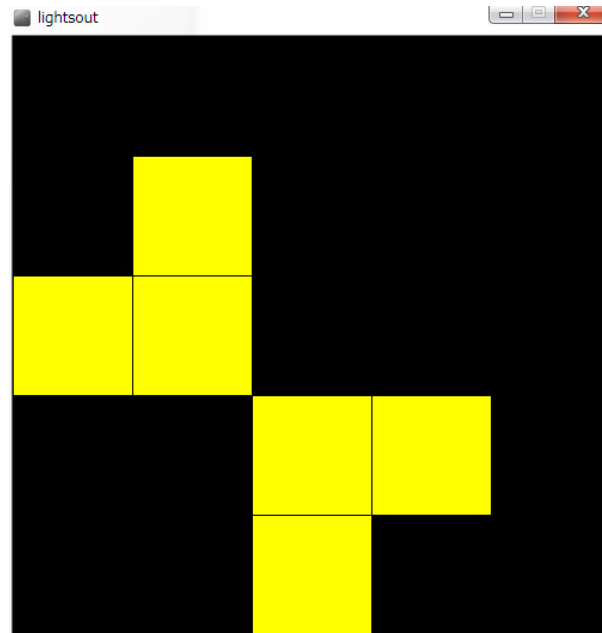


プログラミング演習I (第9回) 課題

• 基本① スケッチ名：LightsOut

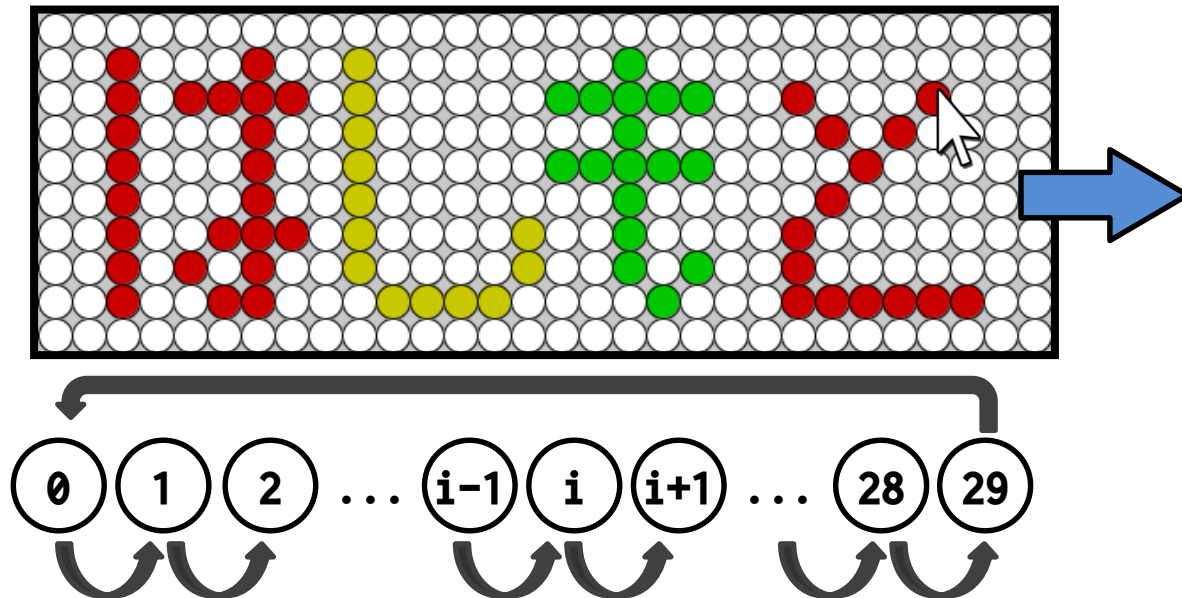
- 横5マス、縦5マスの盤面を作り、そのマス目をクリックすると、クリックされたマス目の上下左右とそのマス目自体の色を反転させるLights Outを作れ
- なお、すべてが黒色になったらCLEAR!と表示するようにせよ
- ただし、起動したタイミングで下記のような表示になっているようにせよ (予習した人はこれをやるだけ！)



プログラミング演習I (第9回) 課題

基本② スケッチ名: 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！！**

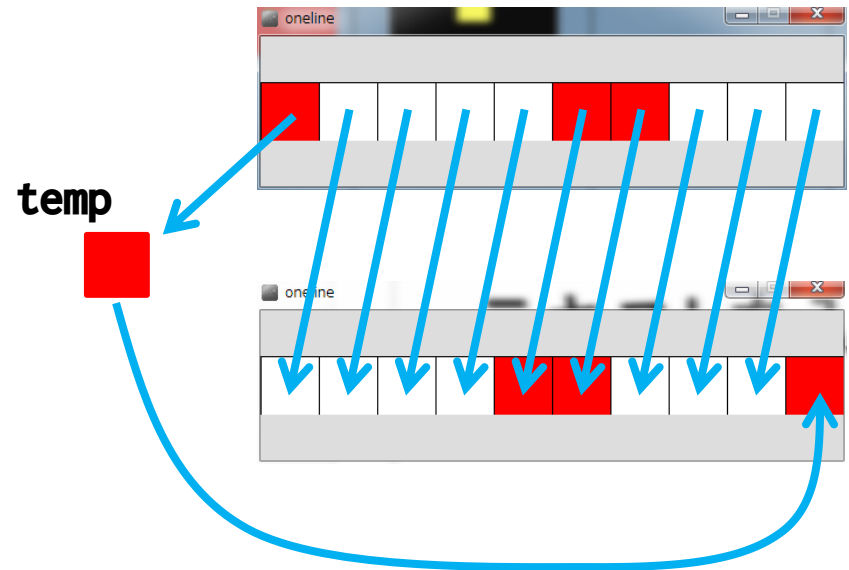


配列の値を循環させる

- 考え方

- 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;`

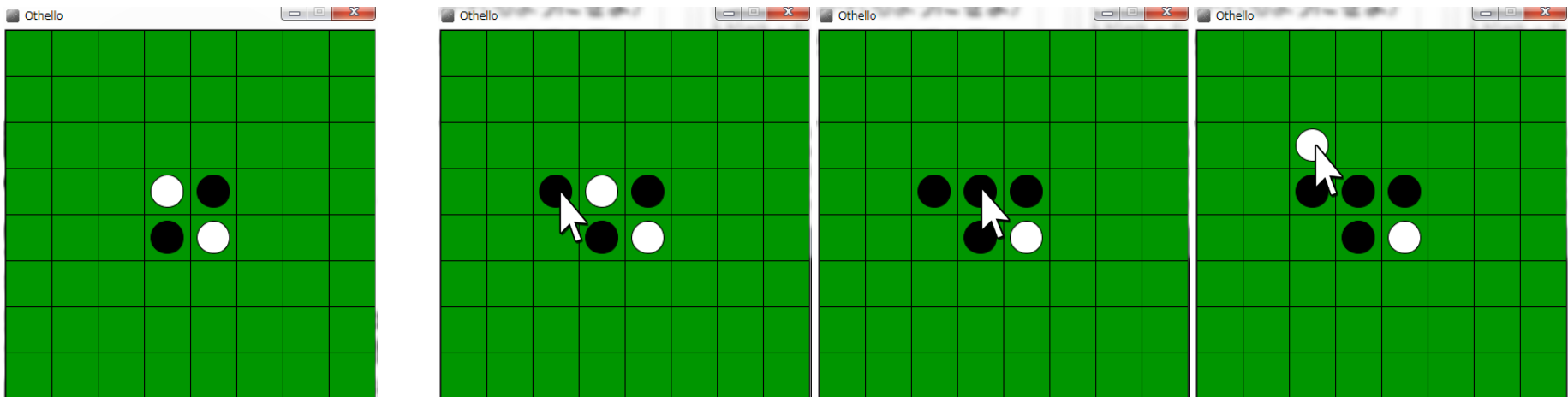


プログラミング演習I (第9回) 課題

基本③ スケッチ名：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

初期配置



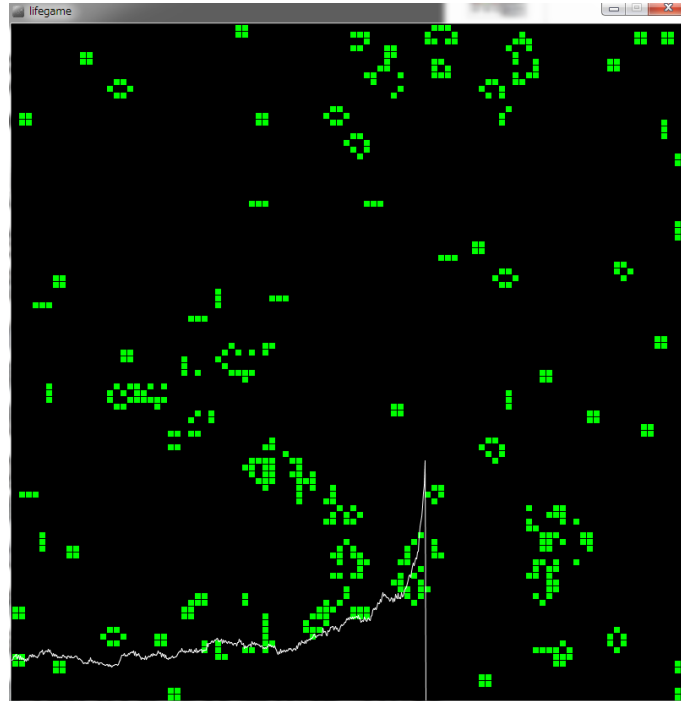
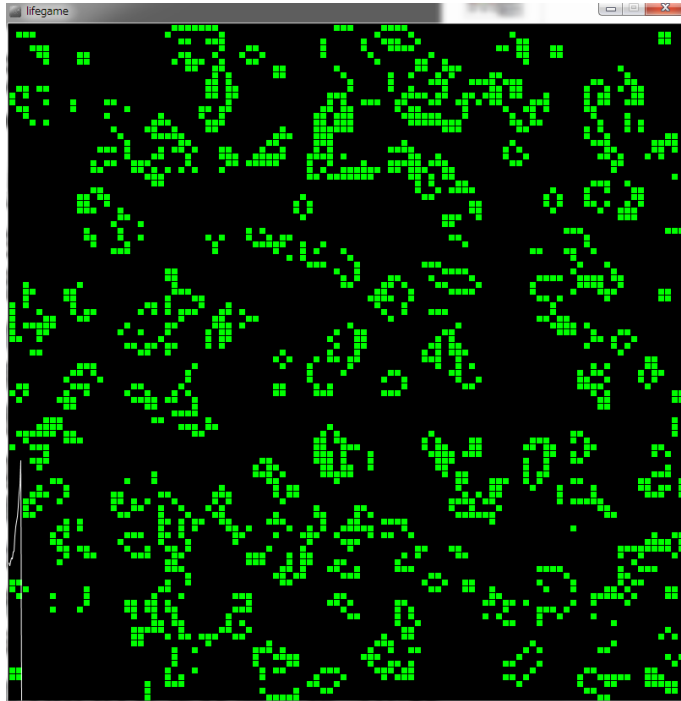
プログラミング演習I (第9回) 課題

• 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。
- **800x800のウィンドウ内に100x100**のマス目を用意し、セルが活着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



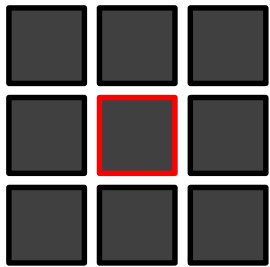
下記動画も面白いので参考までに
ライフゲームの世界 <http://www.nicovideo.jp/mylist/34610498>

プログラミング演習I (第9回) 課題

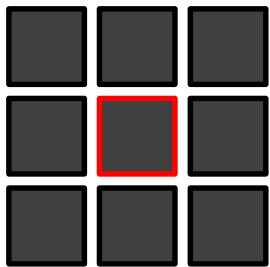
あるマス (赤フレーム) の縦・横・斜めの8マスの生死の状態 (生の数) に注目する

- 【誕生】 死んでいるセルに隣接する生きて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てきたセルが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てきたセルが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てきたセルが4つ以上ならば過密により死滅

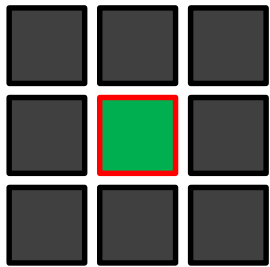
すべて死



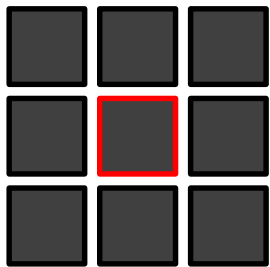
変化なし



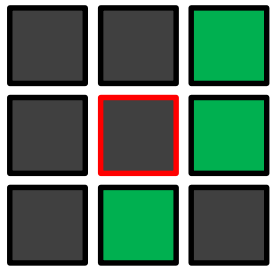
すべて死



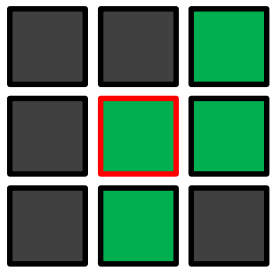
寂しくて死ぬ



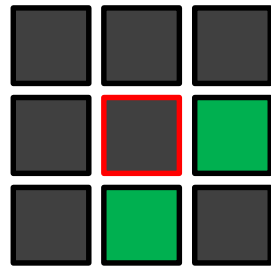
3つのマス



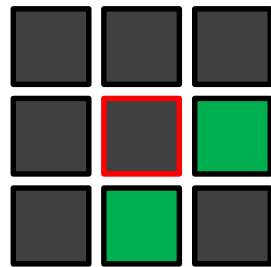
生まれる



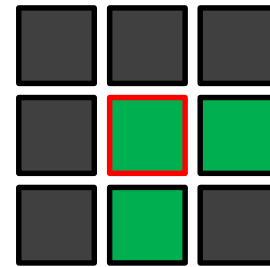
2つの生



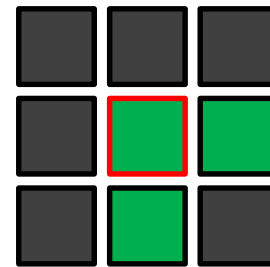
変化なし



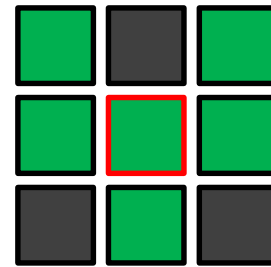
2つ以上の生



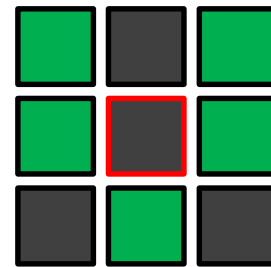
快適で変化なし



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

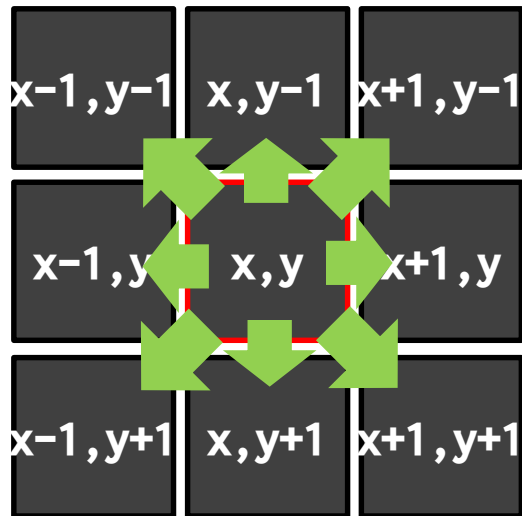
- ルールを整理すると . . .

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

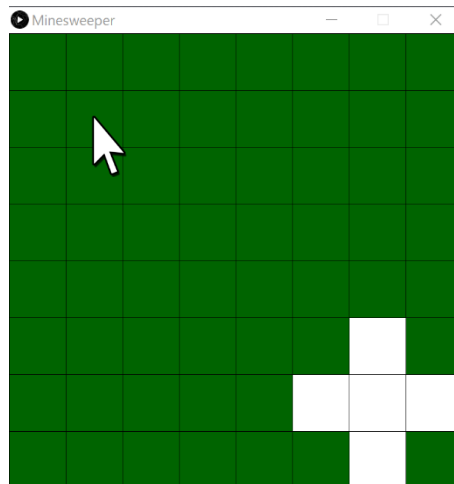
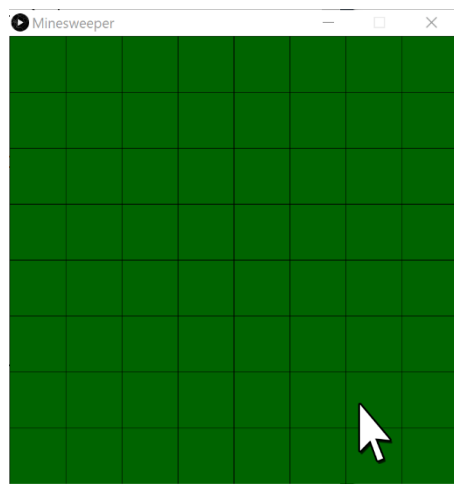
0,0	1,0	2,0
0,1	1,1	2,1
0,2	1,2	2,2

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

• 発展② スケッチ名: MinesweeperSimple

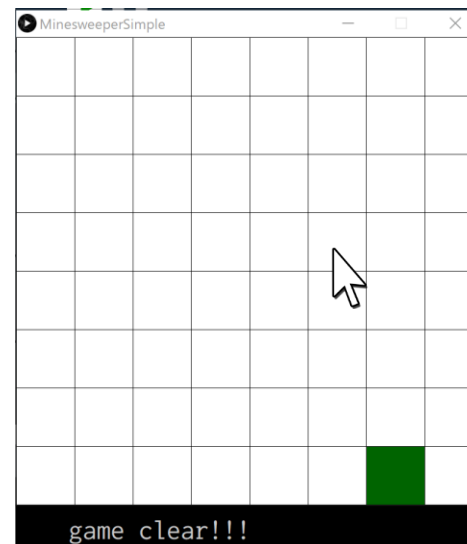
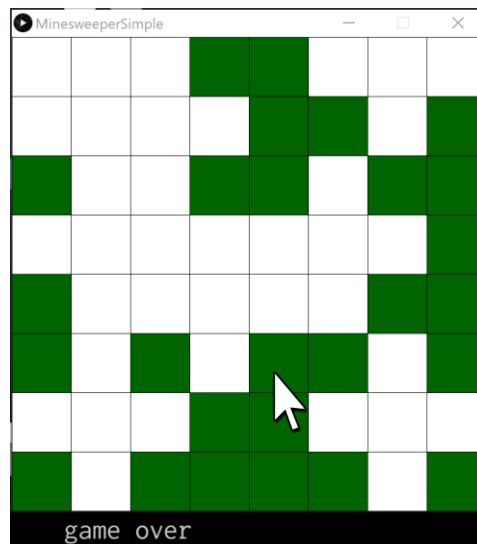
- ウィンドウ幅800x800に、横8マス、縦8マスのマス目を作成し、すべてのマス目を緑色にせよ
- またそのマス目にランダムに1個爆弾を配置せよ (どこに爆弾があるかはわからないようにせよ)
- マス目をクリックしたときに、マス目に爆弾があった場合はゲーム終了とし、マス目に爆弾がなかった場合はそのマス目を白色にせよ
- その上下左右4マス (端の場合は数が減る) すべてに爆弾がない場合は、その4マスを白色にせよ。上下左右のいずれかに爆弾が含まれる場合は何もしないようにせよ (これがヒントとなる)
- 全てマス目を白色 + 開けていない爆弾だけにするのができたらクリアと表示せよ
- ゲーム終了と、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



プログラミング演習発表会

- 7月17日 19:00-22:00 (5F ホール)
- 進め方
 - 研究室内でリーダーを決め、できるだけリーダーのPCにプログラムを集めてプレゼンを行う
 - 必ず事前にノートPCで動作を確認しておくこと (本番で動かなかつたら点数はありません)
 - その場で順序を決めその順序に応じて発表を行うこと
 - 発表時間は交代も含めて **【90秒厳守】**
 - 早く終わったからといって減点はしません
- 提出締め切り
 - プログラムはいつもの定められたフォルダに **【22日の17時まで】** に提出すること。
 - 提出が確認できない場合は発表点がつきません

レギュレーション

- Processingにより 「実世界のモノを模倣せよ」
 - 実世界の何らかの模倣であること **(3点)**
 - 動き (アニメーション) があり, マウスまたはキーボードの入力に対して, 何らかの反応 (動きなど) をするインタラクティブなものであること **(3点)**
 - プログラムは適切にインデントし, コメントを入れること **(3点)**
 - 一昨年度までの条件分岐, 繰り返し, 配列, 関数を使えというレギュレーションはなし (毎回のテストで判定)
 - 組, 番号, 名前, 何の模倣かと, こだわりポイントを発表すること **(3点)**
 - プログラムが動作し, 発表時間が交代時間含めて1人90秒で終わること (厳守&強制終了) **(3点)**
- 起動 + 自己紹介があるので見せる時間は60秒程度!
- 必ず研究室単位の発表用PCで動作確認をし, 発表練習をしておくこと. バックアップPCは用意しておくべし!