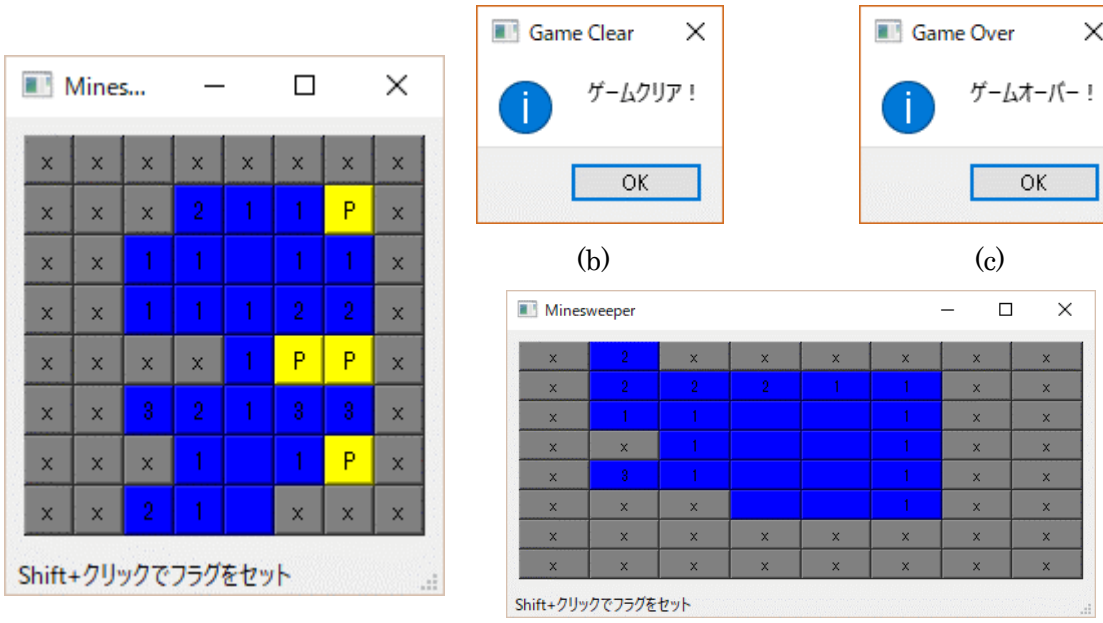
応用情報工学演習　“Minesweeper GUI” 課題

2019/10/10
2019/10/21 更新

1.　GUI 版 Minesweeper

Minesweeper に Graphical User Interface (GUI)を追加しよう．図 1 に GUI の仕様を示す．



(a)　(b)　(c)　(d)

図 1 GUI の仕様: (a) メインウィンドウ．クリックでセルを開く，[SHIFT]-クリックでフラグを立てることが可能．(b) ゲームクリア・ダイアログ，(c) ゲームオーバー・ダイアログ，(d) ウィンドウのサイズ変更．

2.　GUI 版 Minesweeper の作成

以下の仕様，手順に従ってプログラムを作成しなさい．

2.1.　Game クラス：地雷マップ，盤面状態の管理

前回課題で作成した Game クラスのコードをコピーする．

print_header(), print_footer(), print_mine_map()の各メソッドは不要なので削除する．

2.2.　MyPushButton クラス：盤面上のセル

QPushButton のサブクラスとして，盤面上の “セル”に対応する GUI 部品として MyPushButton クラスを定義する．

2.2.1.　__init__()：初期化

実装済みである．セルの位置(x, y)と親ウィンドウ(MinesweeperWindow クラスのインスタンス)を引数として与えて初期化を行う．初期最小幅を 20 ピクセルに設定し，サイズポリシーを

QSizePolicy.MinimumExpanding に設定する

2.2.2. set_bg_color()：色の変更

実装済みである．セルの色を変更する．QWidget.setColor()メソッドを使用する．

2.2.3. on_click()：セルをクリックしたときの動作

一部実装してあるので，未実装の部分を完成させなさい．処理概要は以下のとおり：

(1) クリックした場合はセルを開き, [SHIFT]キーを押しながらクリックした場合はフラグを立てる．
ヒント：キーボードの状態を調べるには，QApplication.keyboardModifiers()を使用するとよい．

(2) 地雷セルのクリックを検知したら，「ゲームオーバー！」のメッセージボックス（図 1(c)）を表示する．メッセージボックスで[OK]ボタンを押すとアプリケーションを終了する．
ヒント：アプリケーションを終了するためには親ウィンドウ(MinesweeperWindow のインスタンス)の close()メソッドを実行すればよい．

(3) セルの状態を表示する．単純に，次のステップで実装する MinesweeprWindow クラスの show_cell_status()を呼び出せばよい．

(4) ゲームが終了しているか確認し，終了していれば「ゲームクリア！」のメッセージボックス(図 1(b))を表示する．メッセージボックスで[OK]を押すとアプリケーションを終了する．

2.3. MinesweeperWindow クラス

CMainWindow クラスのサブクラスとして MinesweeperWindow クラスを作成する．図 1(a)のメインウィンドウに対応する．

2.3.1. __init__()：初期化

本メソッドは実装済みである．処理概要は以下のとおり：

(1) 親クラス(CMainWindow)の初期化メソッドを呼び出す,

(2) ゲームクラスのインスタンスを生成する,

(3) initUI()メソッドを呼び出す.

2.3.2. initUI()：ユーザインタフェース(UI)の初期化

一部実装してあるので，未実装の部分を完成させなさい．処理概要は以下のとおり：

(1) ステータスバーに”Shift+クリックでフラグをセット”と表示する

(2) ゲームボードを構築する．

(ア) 8x8 個のセルについて，以下を繰り返す．

① MyPushButton クラスのインスタンス button を作成する．

② ボタン button がクリックされたときのアクションとして，MyPushButton クラスの on_click()メソッドを呼び出すように設定する．

(イ) 図 1(a)に示すようにセルを 8x8 のグリッド状にレイアウトする．QVBoxLayout, QHBoxLayout クラスを使用する．
ヒント：デフォルト設定ではボタン間にスペースが入るので，QVBoxLayout(spacing = 0)とすると良い．

(3) 構築したゲームボードをメインウィンドウに設定

ヒント：ステップ(2)(ア)で生成した MyPushButton のインスタンスを次の show_cell_status()で参照するために記憶しておく必要がある．

2.3.3. show_cell_status()：盤面の表示

一部実装してあるので，未実装の部分を完成させなさい．画面仕様は以下のとおり：

表 1 画面仕様

盤面の状態	地雷マップの状態	表示テキスト	表示色
CLOSE	任意	“x”	灰色(“gray”)
OPEN	地雷以外	地雷数（8 近傍に地雷がある場合）または空白	青(“blue”)
FLAG	任意	“P”	黄色(“yellow”)

ボタンにテキストを表示するには MyPushButton.setText()メソッドを，色を変更するには MyPushButton.set_bg_color()を使うとよい．

2.4. main()関数

この関数は実装済みである．処理概要は以下のとおり：

(1) アプリケーション(QApplication)のインスタンスを作成する．

(2) MinesweeperWindow クラスのインスタンスを生成する．ウィンドウが表示される．

(3) メインループに入る．

3. オプション課題

余力があれば，プログラムに改良を加えて以下のようなことを実現してみてください．

(1) ボタンにアイコンを表示する．”P”, “x”などの文字を表示する代わりに，自分で作成した見栄えの良いアイコンを表示するように変更する．

(2) ゲームオーバ，ゲームクリアのメッセージボックスを表示する際に，地雷セルの位置を画面表示する．

(3) その他，ゲームを面白くしたり，見栄えを良くしたりするための改良を行う．

4. 動作確認

以下のチェックリスト（検査項目表）でプログラムの動作を確認しなさい．

表 1　プログラムチェックリスト

項番	入力・操作	結果	検査結果
1	minesweeper_gui.py を実行する	図 1(a)のウィンドウが表示される タイトルは “Minesweeper”である ステータスバーに” Shift+クリックでフラグをセット”が表示される セルが 8x8 表示されている セルには“x”が表示される セルの色は灰色である	

2	（開いていない）地雷以外のセルをクリックする	セルが開く（地雷が埋まっていない 8 近傍セルが開く） セルが青色になる 近傍セルに存在する地雷数が表示される	
3	任意のセル上で[Shift]-クリックする	フラグが設定される セルが黄色になる	
4	フラグが設定されたセルの上で[Shift]-クリックする	フラグが解除される セルが灰色になる	
5	フラグが設定されたセルの上でクリックする	~~セルが開かない（状態が変化しない）~~ セルが開く（地雷が埋まっていない8 近傍セルが開く），セルが青色になる，近傍セルに存在する地雷数が表示される	
6	開いているセルの上をクリックする	何も変化しない（周囲のセルが開くことがない）．	
7	ウィンドウのサイズを変更（マウスでウィンドウの端をドラッグ）	図 1(d)に示すように，セルが拡大する	
8	わざと地雷を踏んでみる	図 1(c)のダイアログ（ゲームオーバー！）が表示される [OK]を押すとアプリケーションが終了する	
9	ゲームをクリアする	図 1(b)のダイアログ（ゲームクリア）が表示される [OK]を押すとアプリケーションが終了する	

5. 課題提出方法

- LETUS の課題提出エリアに ~~10/24(木) 12:50~~ までに提出すること．
  11/3(日)18:00まで
- 提出物
  - ソースコード（処理概要が分かるようにコメントを入れてください）
  - メモ（追加課題に取り組んだ場合は機能概要について記述してください）
  - ファイル名は minesweeper_gui.py
- 注意点
  - 表 1 のチェックリストで全てが OK になることを確認してから提出すること

以上