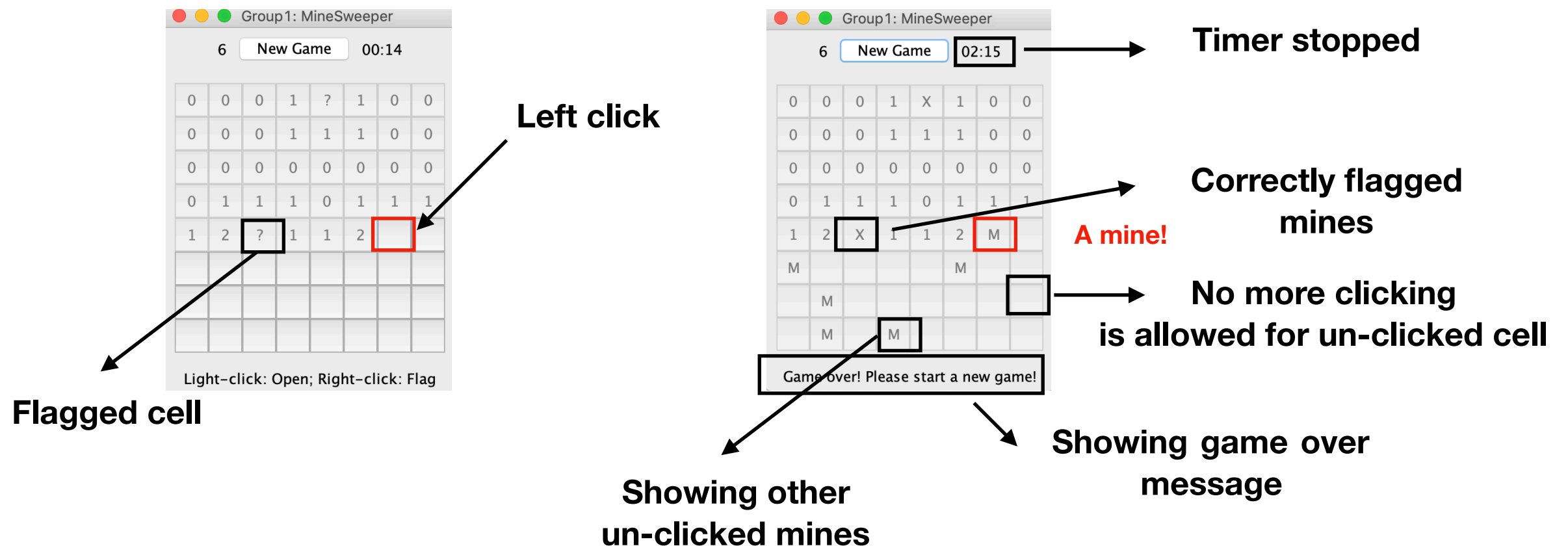


User Cases Testing And The Pattern

CS 151 Group project
Mine Sweeper

Case 1 Left Click

- Open a cell
 - Variations # 1
 - 1.1 It's a mine — end game.
 - 1.2 Timer stops
 - 1.3 Show the rest of the un-clicked mines.
 - 1.4 show cross signs on the correctly flagged mines.



Case 1 Left Click

- Open a cell
 - Variations # 2
 - 2.1 It's a mine — end game.
 - 2.2 Timer stops
 - 2.3 Show the rest of the un-clicked mines.



← Left click



→ Timer stopped

→ Showing other un-clicked mines

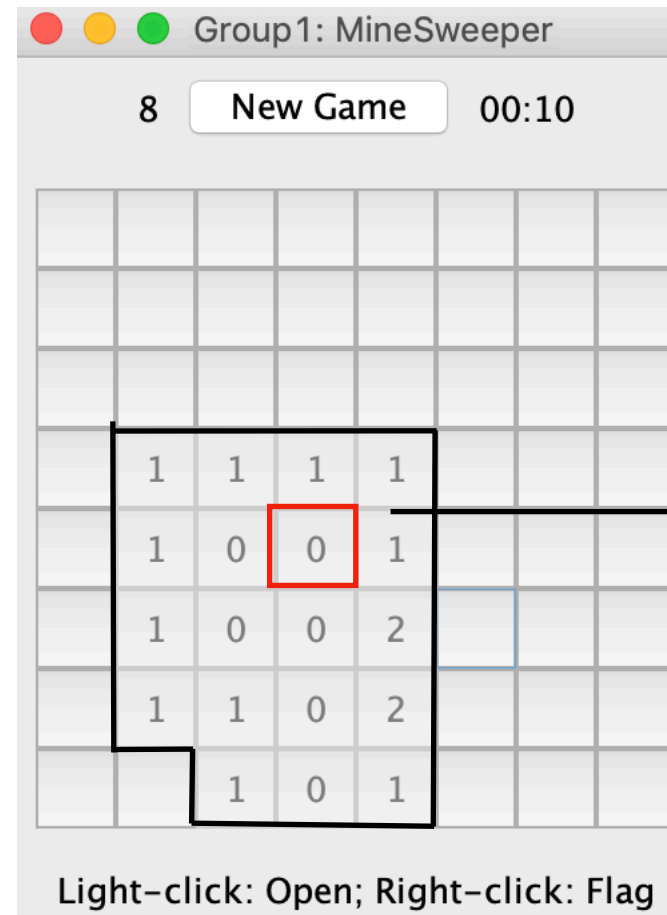
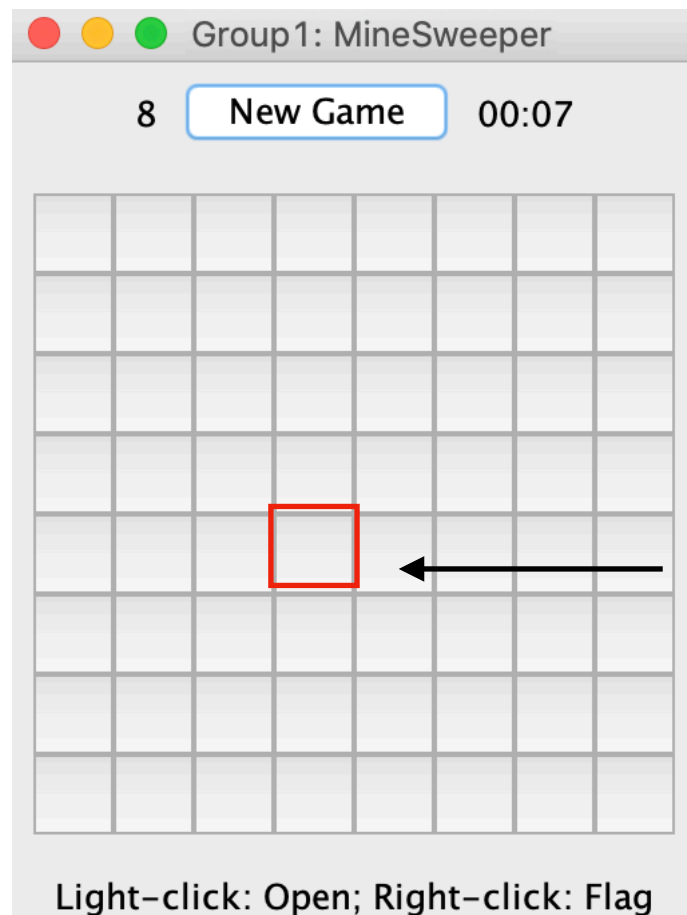
A mine!

→ No more clicking is allowed for un-clicked cell

→ Showing game over message

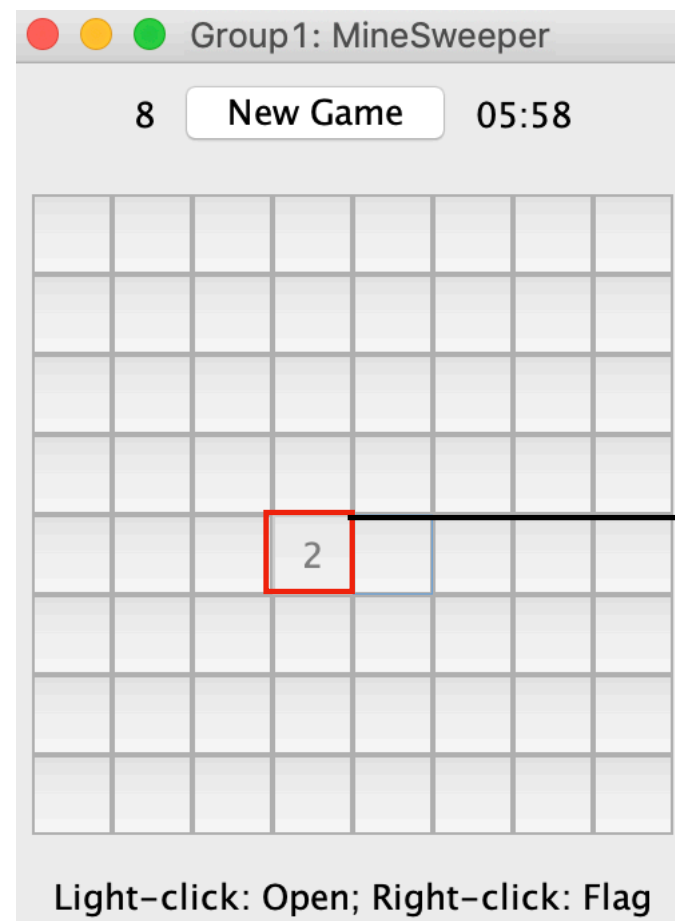
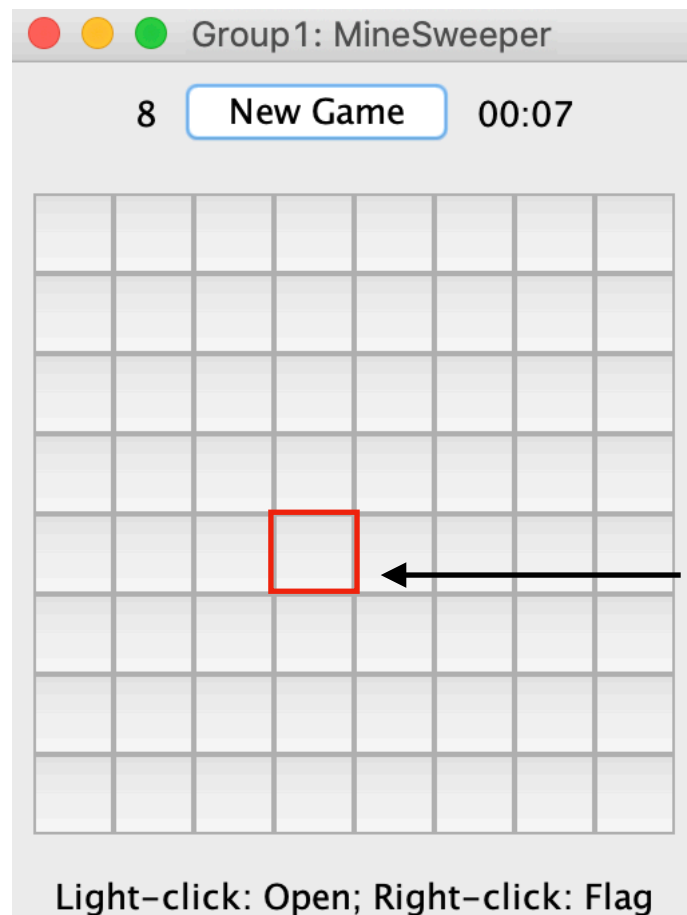
Case 1 Left Click

- Open a cell
 - Variations # 3
 - 3.1 it's not a mine, and there is no mine around it, open the surrounding area



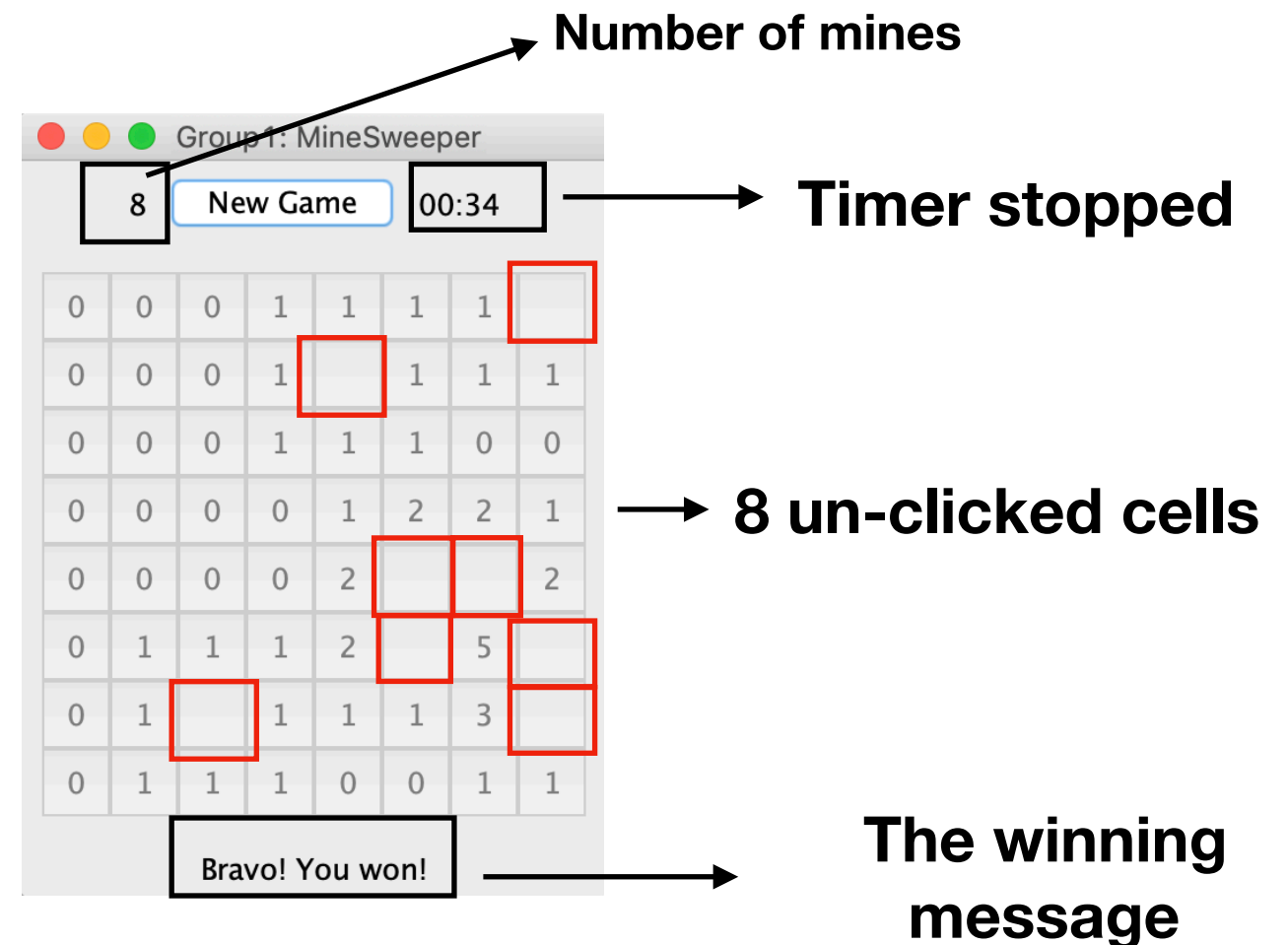
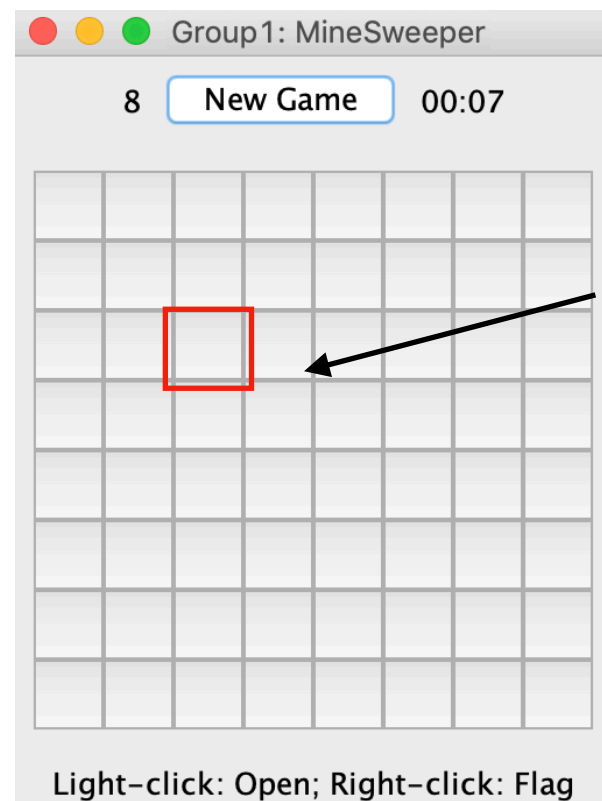
Case 1 Left Click

- Open a cell
 - Variations # 4
 - 4.1 Not a mine, and there are mines around it, show the number of adjacent mines



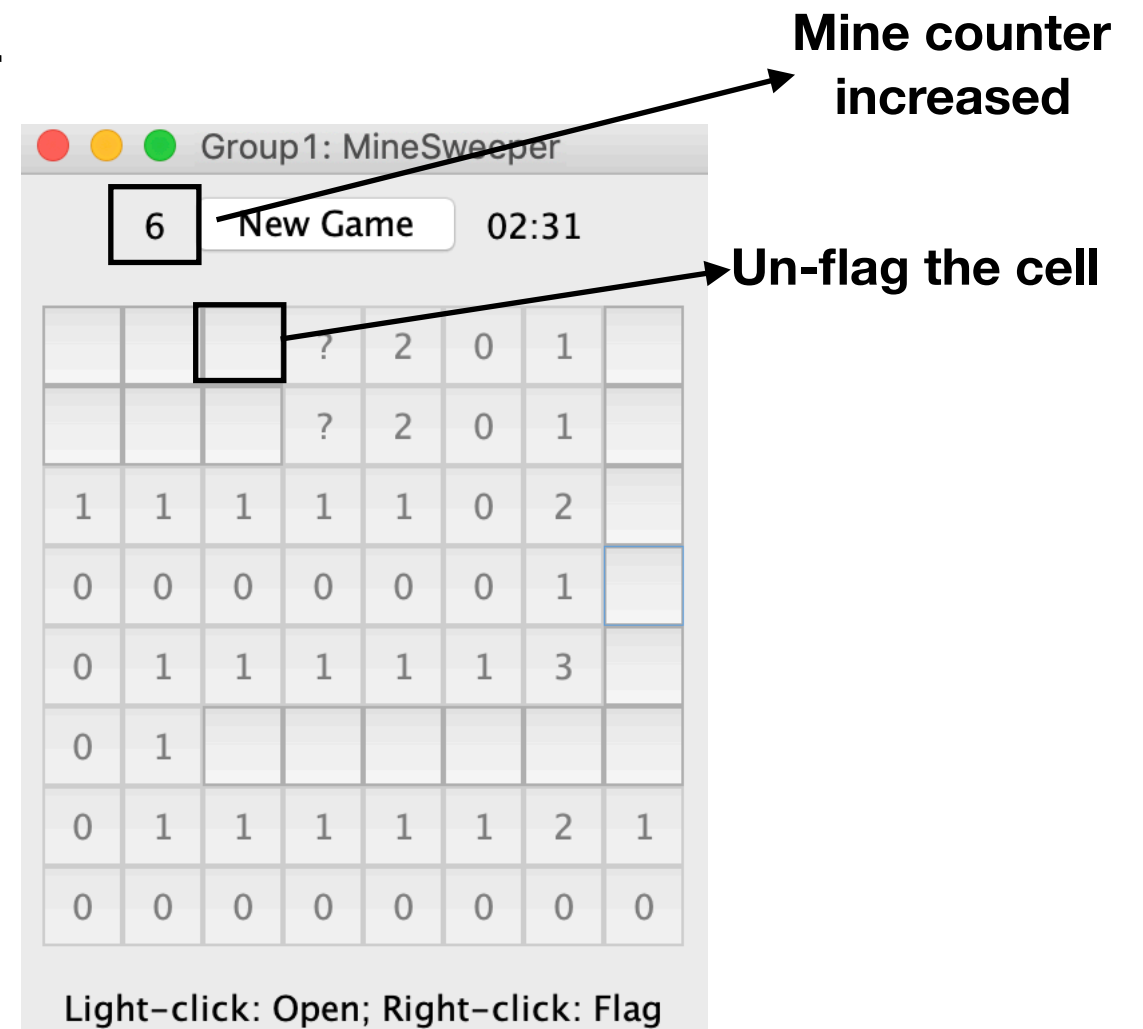
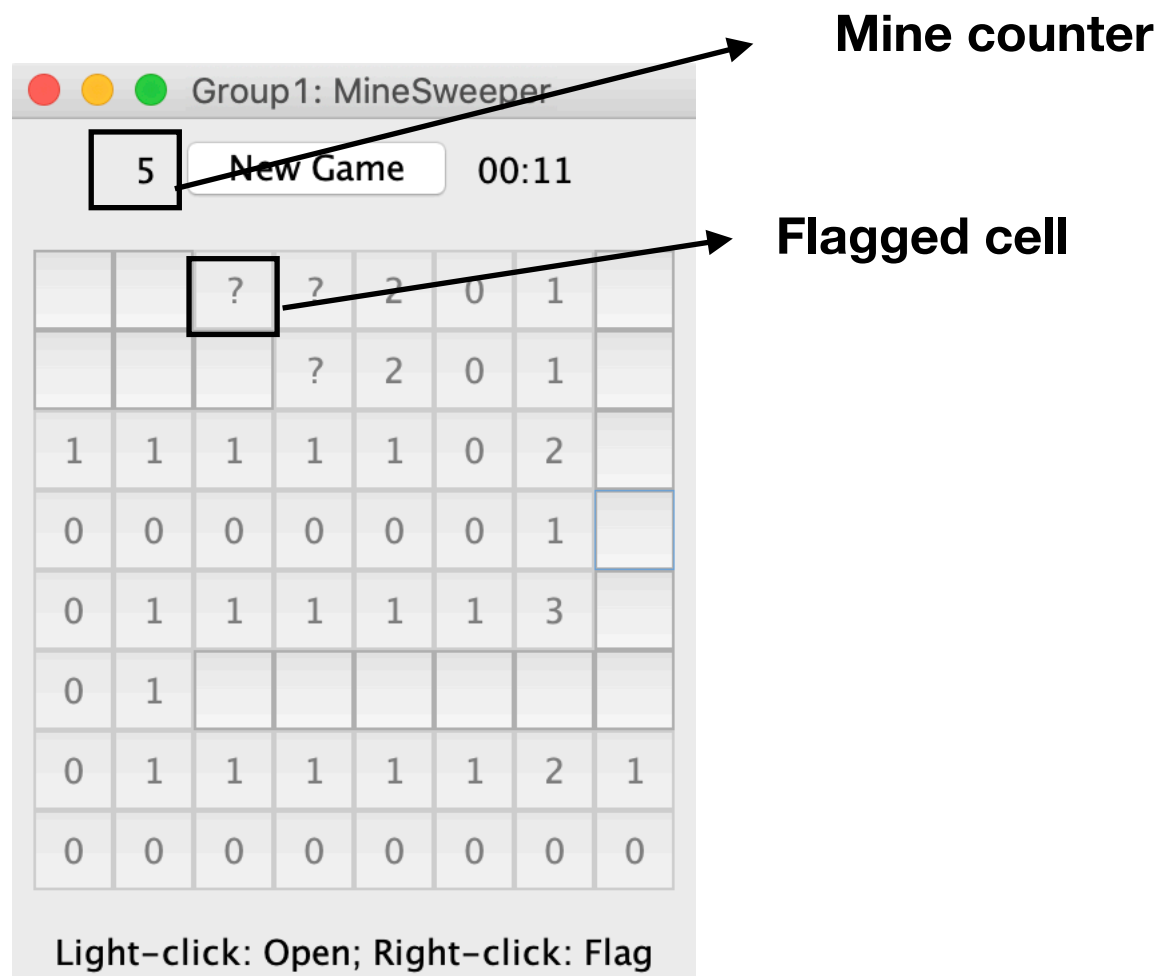
Case 1 Left Click

- Open a cell
 - Variations # 5
 - Continue from 4.1
 - Number of the un-clicked cell equals to the number of mines - user win.



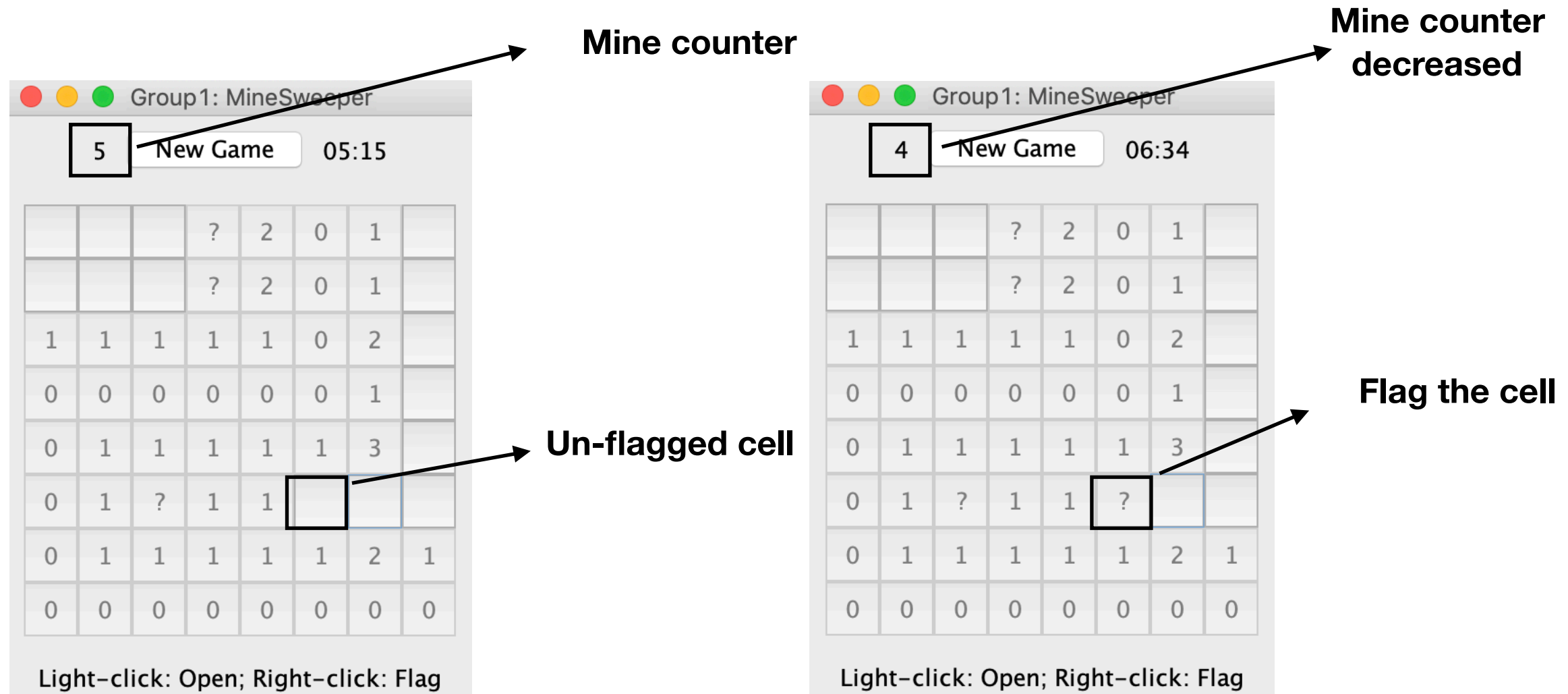
Case 2 Right Click

- Right-click mouse on a flagged cell
 - 1. Un-flag a cell
 - 2. Increase the numbers on mine counter



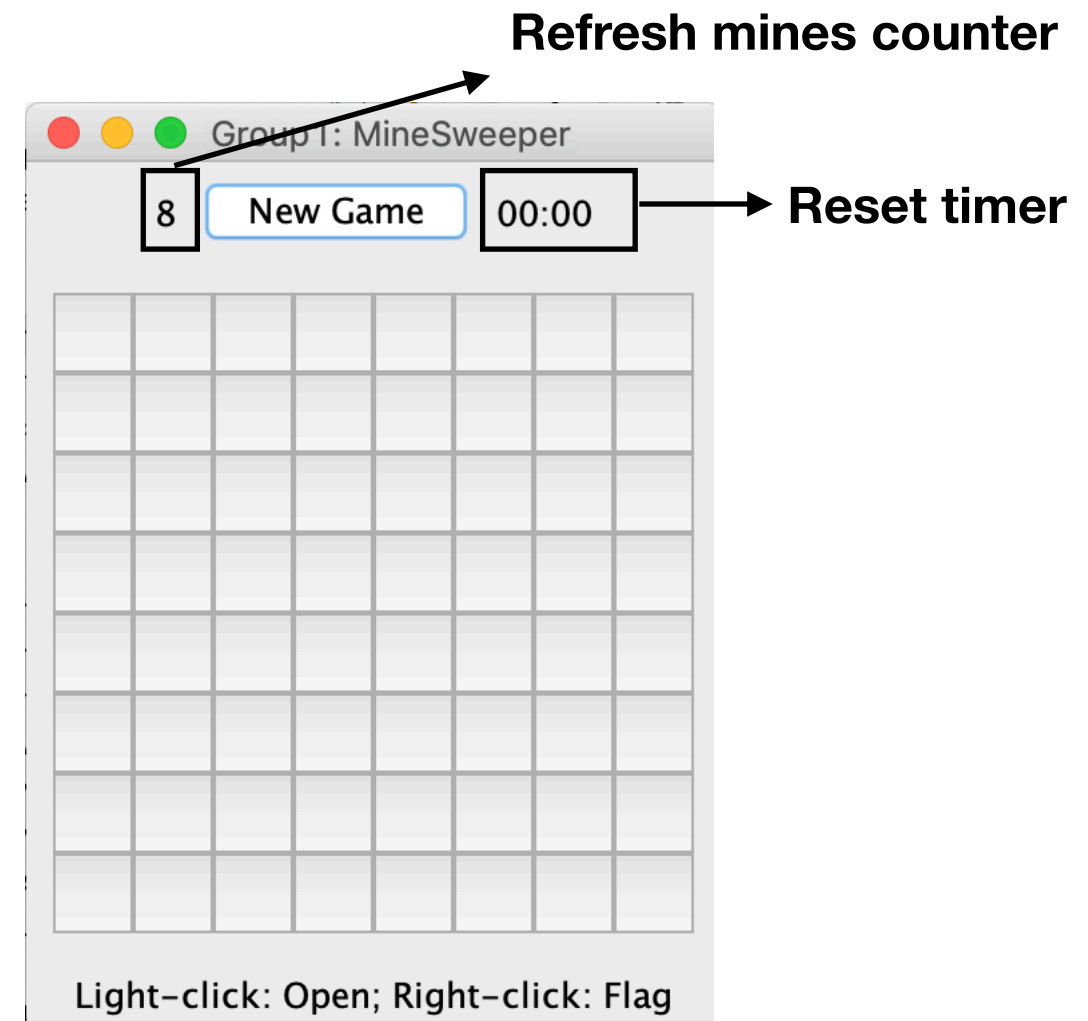
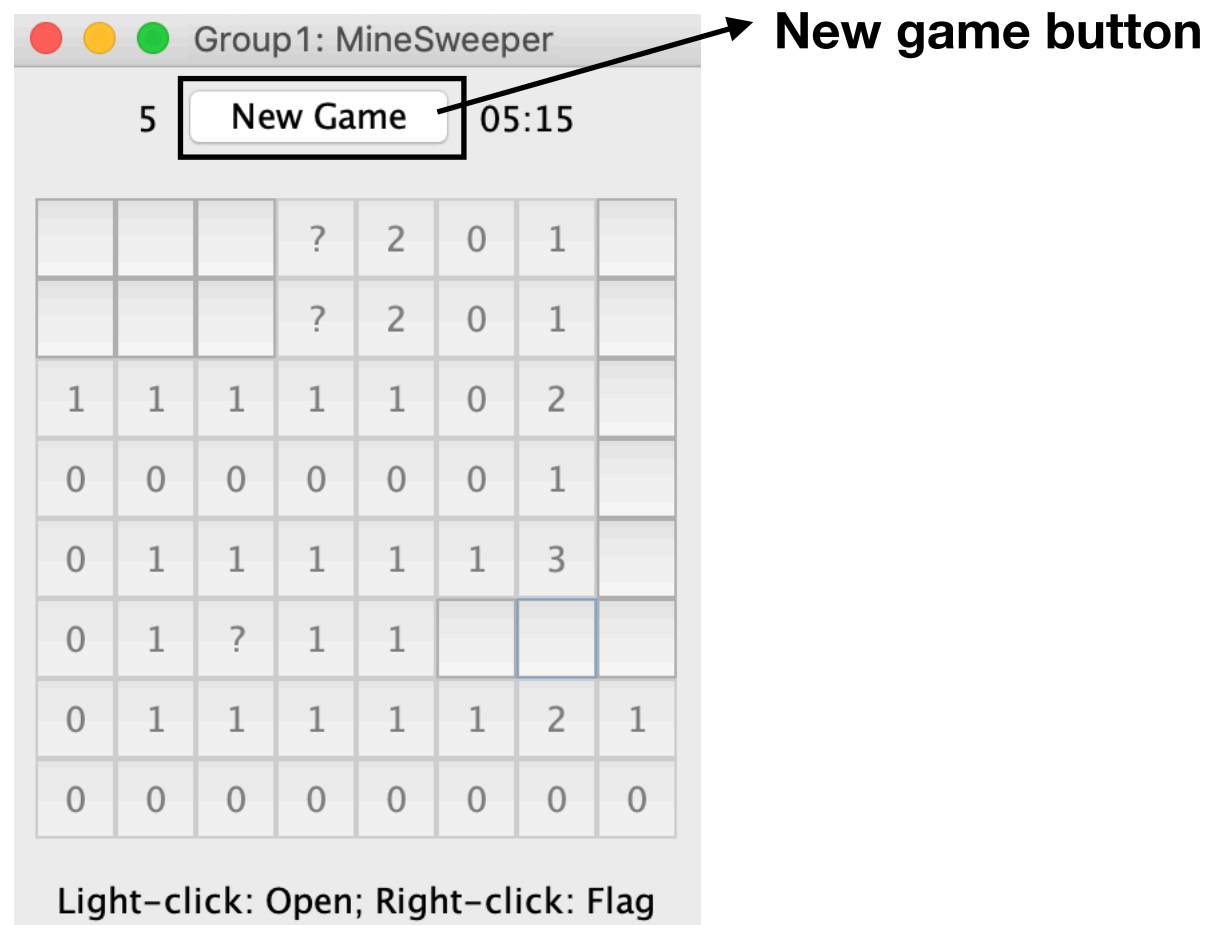
Case 3 Right Click

- Right-click mouse on a un-flagged cell
 - 1. Flag the cell
 - 2. Decrease the numbers on mine counter



Case 4 New game button

- New game button click
 - 4.1 User initiates a new game
 - 4.2 GameController set GameModel to restart the game.
 - 4.3 Generate new field.
 - 4.4 Refresh timer and mines counter



The Pattern

Components of the proxy pattern are client, proxy and real subject (subject is an interface). Under that frame, there are variations of proxy patterns: Virtual Proxy, Remote Proxy and Access Proxy etc., depending on the characteristics of the object associated with user's demand.

In the models of our MineSweeper game, we applied Access Proxy pattern because we used proxy as a gatekeeper for the most frequently called methods in the model. Function of the proxy in our case is to check whether the input height, width index is in the boundary range of the mine fields.

Client is `openCell(int h, int w)` and `toggleCell(int h, int w)` methods in the Model class; `boundaryProxy(int h, int w)` method is a proxy method implementing BoundaryChecker interface (this is our subject); and `hasMine(int h, int w)` and `openZeroArea(int h, int w)` methods in the MineField class are real subjects.