CPSC 231 Tutorial #17

michael-hung.ca/teaching

Reminders

TODAY

Quiz 9

THURSDAY

Quiz 9 Review

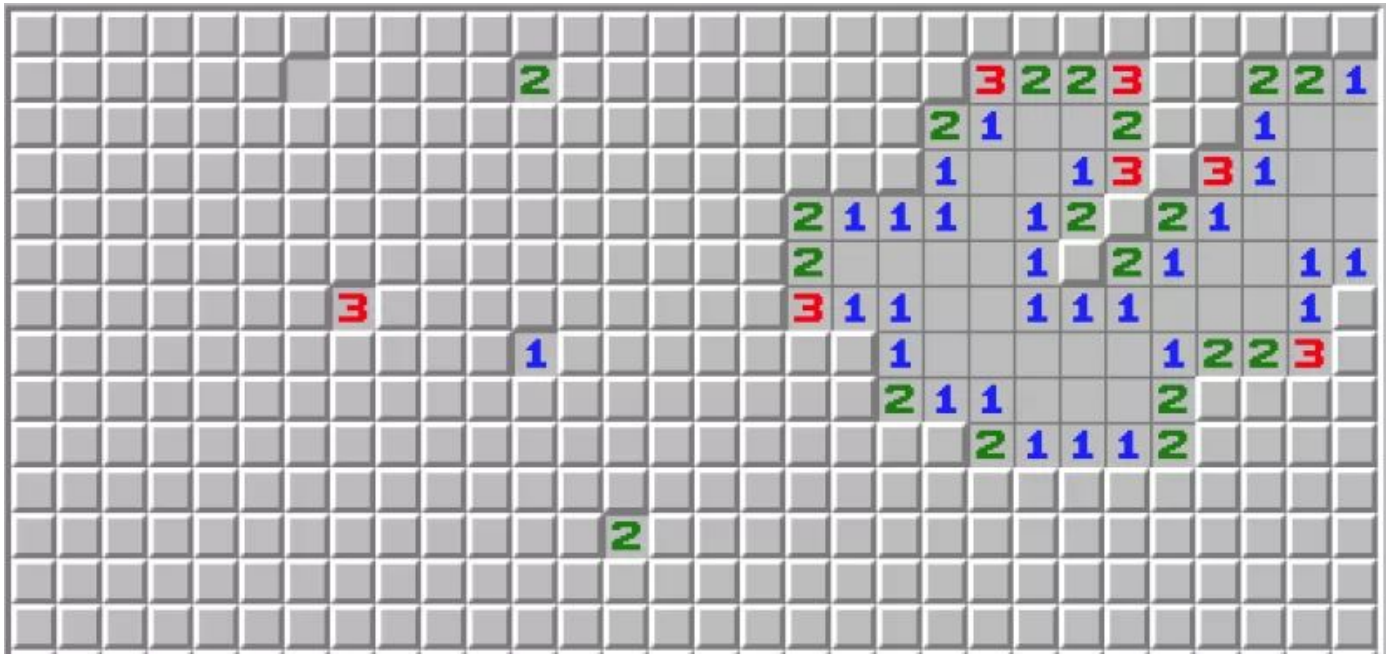
FRIDAY

Assignment 5 Individual Component

NEXT WEEK

Reading Week! No classes!

Minesweeper



Decomposing the Problem

---

1. Graphics
2. Game State
3. Game Logic

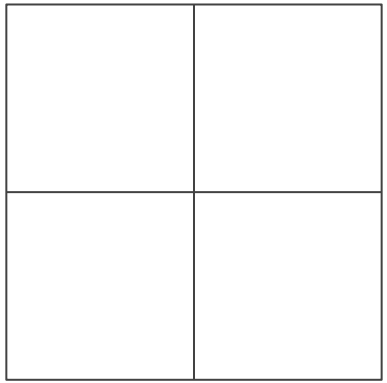
Minesweeper Graphics

---

1. Draw a **Board**
2. Reveal a **Tile**
3. Draw a **Mine**
4. Draw a **Number**

Drawing the Board

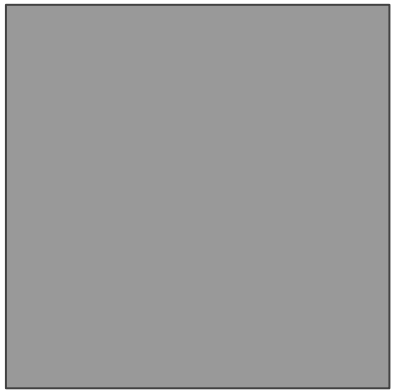
Based on width x and height y, draw a grid separating the board into x*y squares.



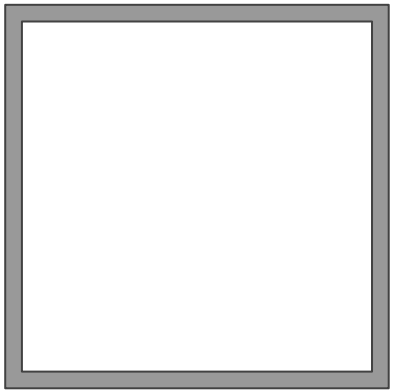
```
Width:   2        Vertical lines:      3
Height:  2        Horizontal lines:    3
```

Revealing a Tile

Start with a dark colour as the default state, then draw a lighter colour on top of it.



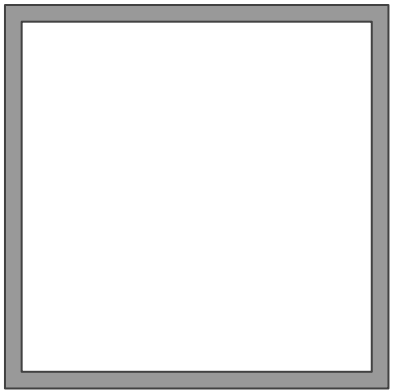
unrevealed



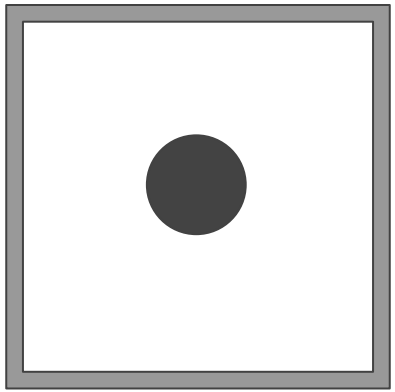
revealed

Drawing a Mine or Number

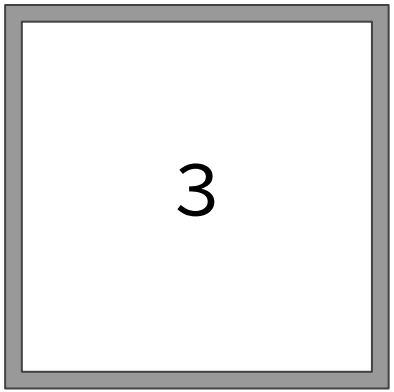
A tile can hide a mine, be blank, or display the number of adjacent mines (blank if 0). Draw on top of a revealed tile.



blank



mine



number

Minesweeper Game State

---

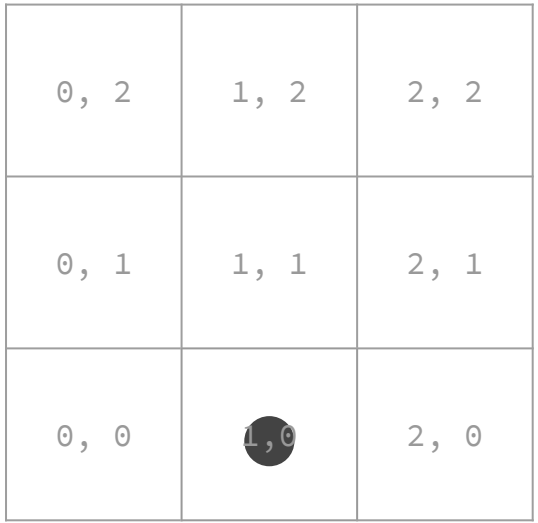
1. Setup the board
2. Identify a tile
3. Mines adjacent to a tile
4. Is the game won?

Setting up the Board

---

1. Generate a 2D array to hold the value of each tile.

e.g.



Fill the array with 0's and 1's; 0 for an empty tile and 1 for a mine.

```
[[0, 0, 0],
 [1, 0, 0],
 [0, 0, 0]]
```

Identifying a Tile

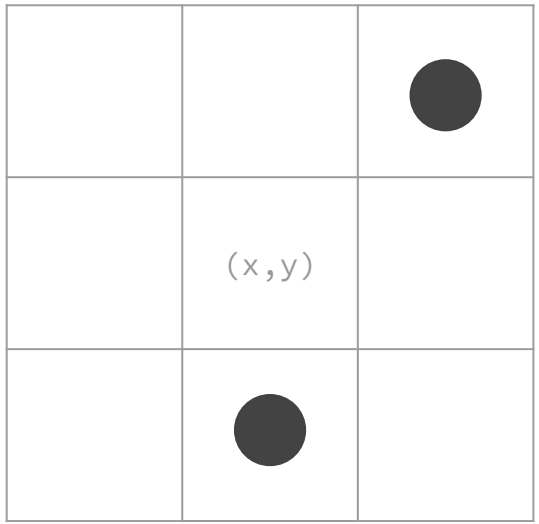
To make it easier to check a tile, make three functions that check if a given index [x][y] is **0** (empty), **1** (a mine), or **2** (a value used for how we'll reveal adjacent tiles later).

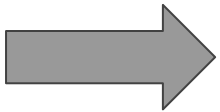
```
isClear(x, y)    isMine(x, y)    isRevealed(x, y)
```

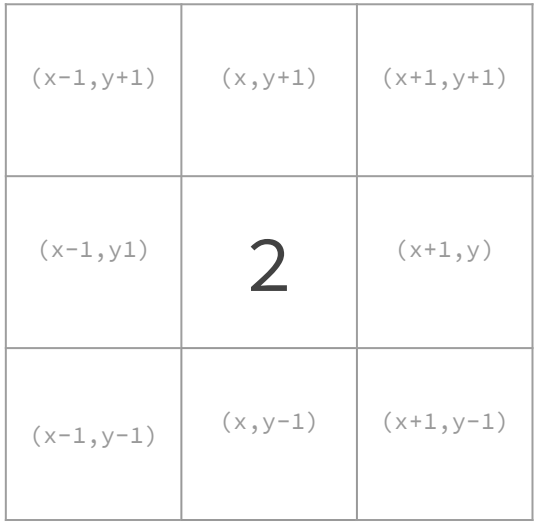
In all three, make sure to handle if x or y are out of bounds. Treat out-of-bound tiles as if they are **already revealed** and are **not mines**.

Number of Mines Near a Tile

For each tile revealed, if there is a mine within 1 tile away (diagonals included), display the number of mines.







Winning Condition

Constantly check the board to see if the game is won.

The game is won when **all non-mine tiles are revealed**.

i.e. there are no 0's in the 2D array.

Minesweeper Game Logic

---

1. Click on a tile
2. If tile is mine, game is lost
3. Reveal all tiles touching the clicked tile unless touching a mine

Clicking a Tile

When we click on a tile, we need to get that tile's position.

1. Set X and Y scales according to board dimensions. So a 3x5 board has X scale from 0 to 3 and Y scale from 0 to 5
2. Clicking on a tile, then, will give us two floats: mouseX and mouseY. Taking the floor will give us the index of the tile.

Clicking a Tile

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

Take this 3x3 board. Clicking somewhere on the middle tile might give something like:

(1.632…, 1.497…)

Simply taking the floor of each gives us the index [1][1]

Losing Condition

---

1. Use a boolean variable, **gameover**, initialized to False, to keep a While loop running the game.
   a. When **gameover** is **True**, the loop breaks.

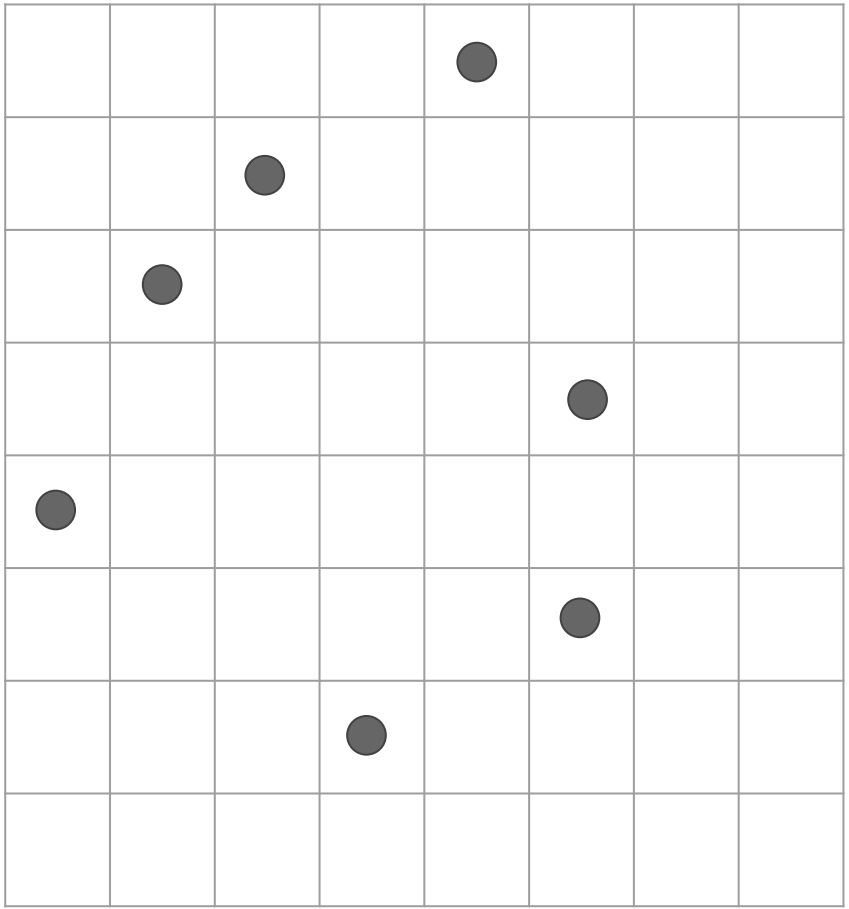
2. When you reveal a tile, check if it is a mine by calling **isMine**(*x*, *y*).
   a. If it is a mine, set **gameover** to **True**.

3. Similarly, a variable can be made for the winning condition.

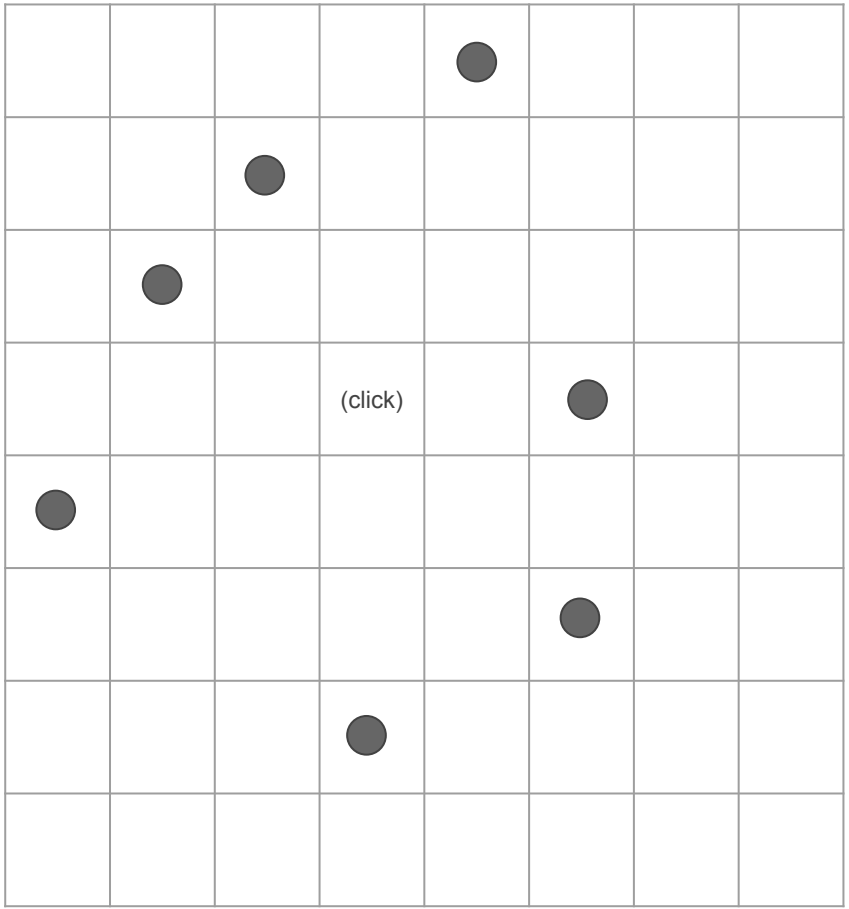
Revealing all Touching Tiles

---

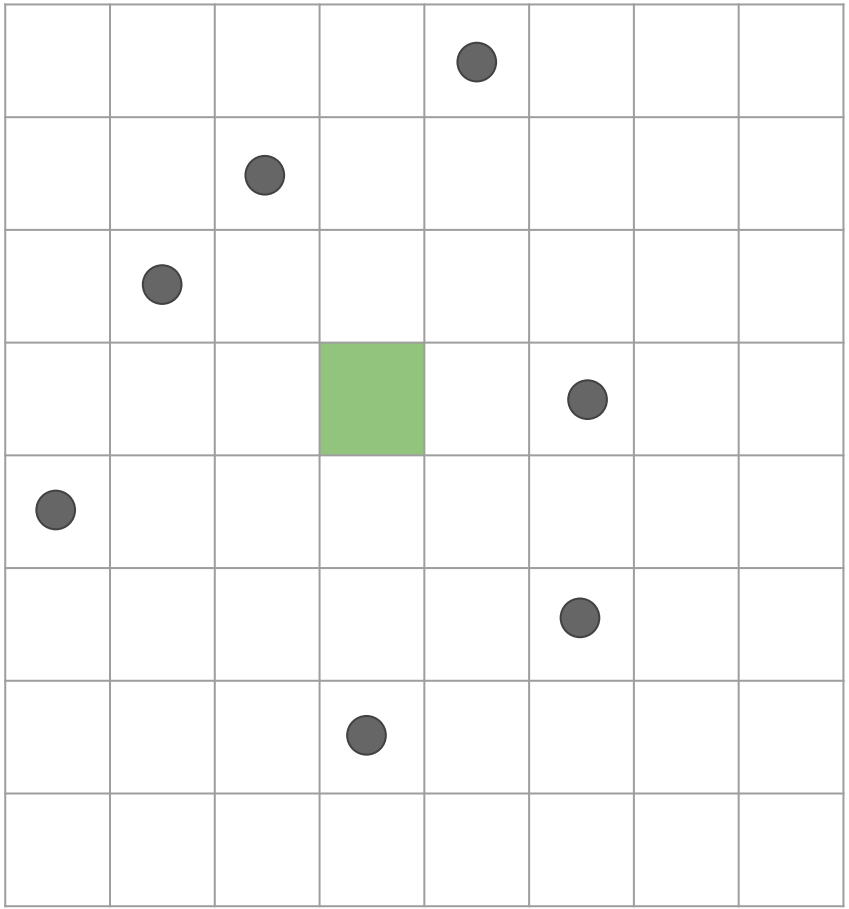
1. If you click on a tile that is not touching a mine, all tiles touching it that **are not a mine** are revealed.
2. If that tile is **also** not a mine *and* is also **not touching** a mine, then reveal all of *its* touching tiles.



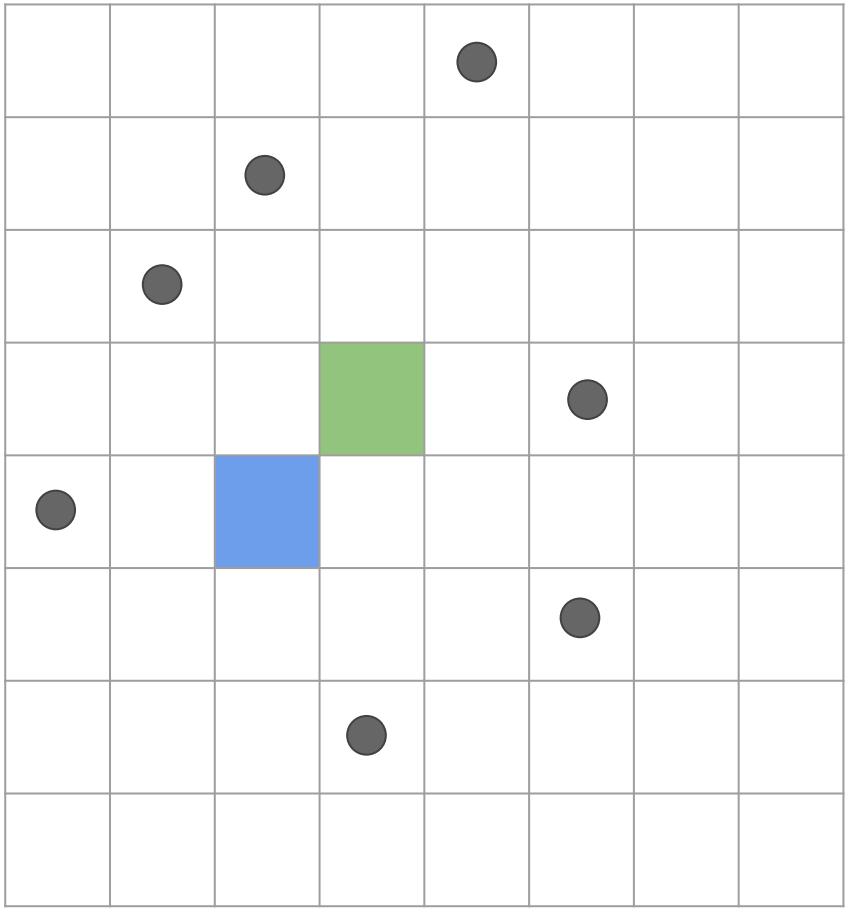
Revealed
To Be Checked
Mine



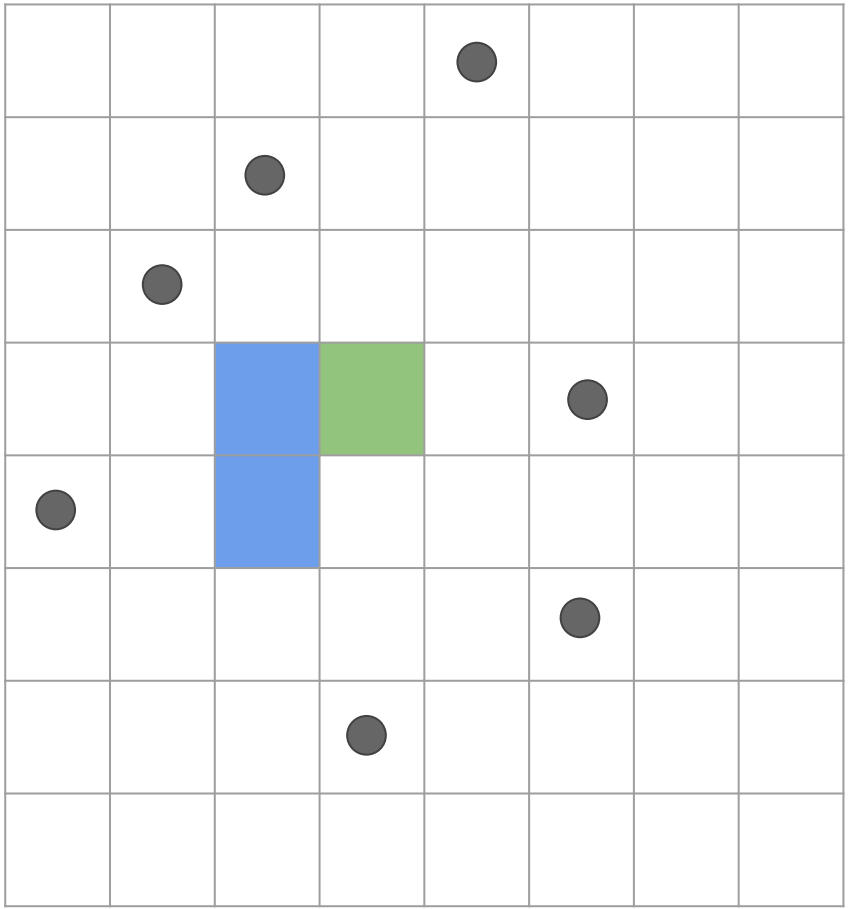
Revealed
To Be Checked
Mine



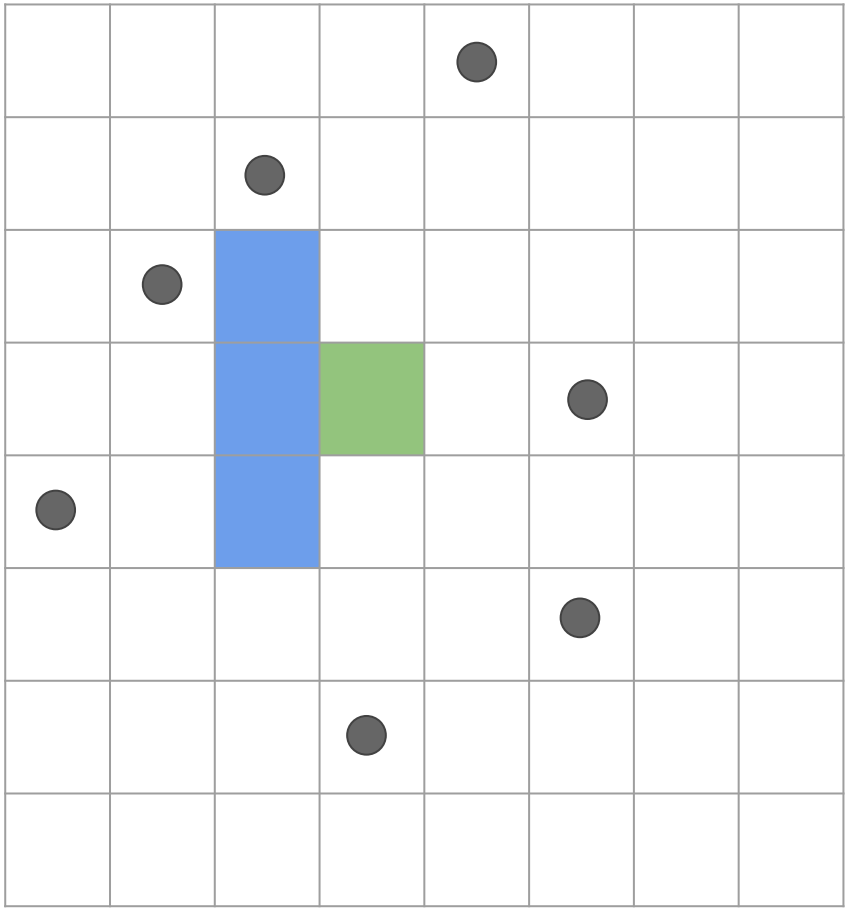
Revealed
To Be Checked
Mine



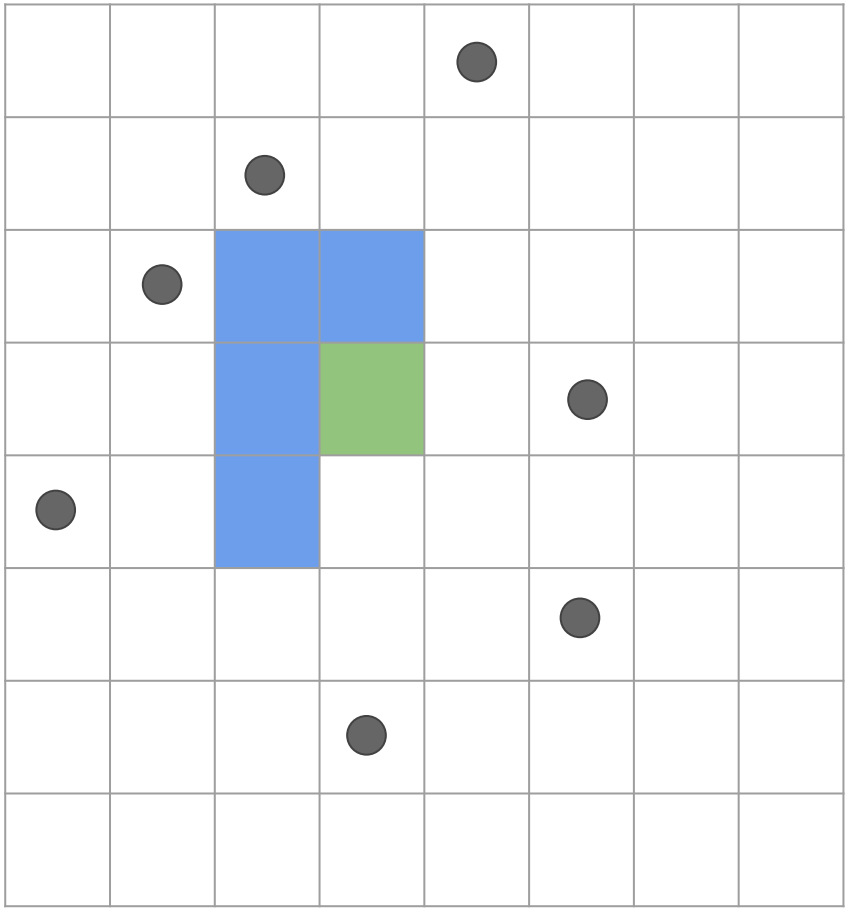
Revealed
To Be Checked
Mine



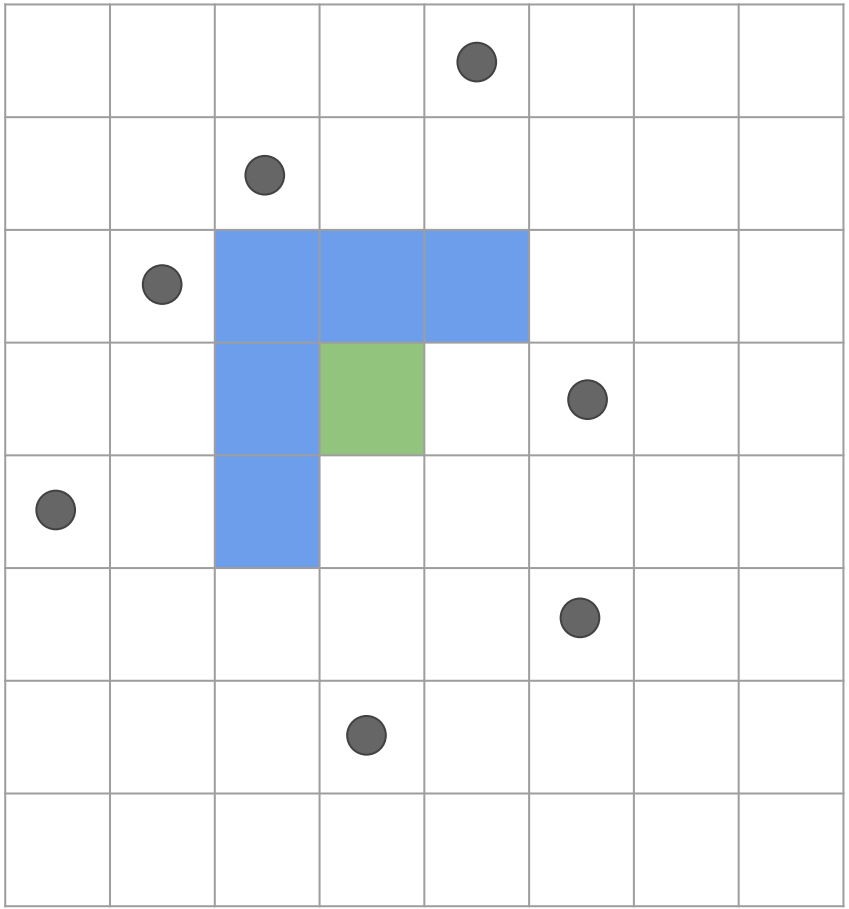
Revealed
To Be Checked
Mine



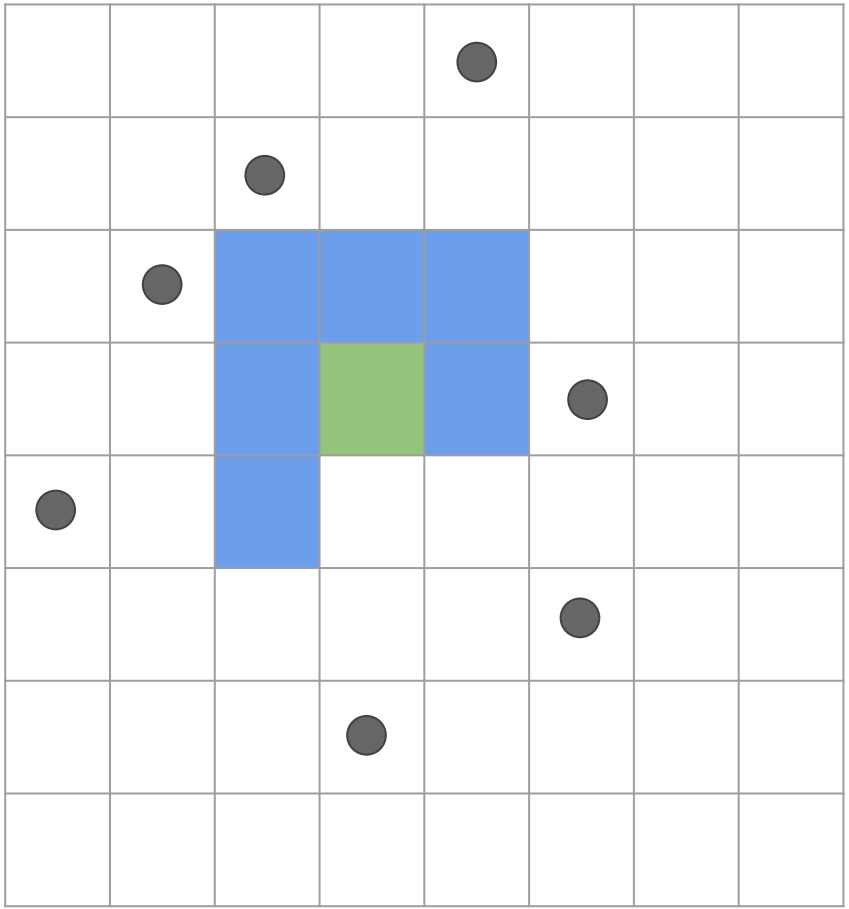
Revealed
To Be Checked
Mine



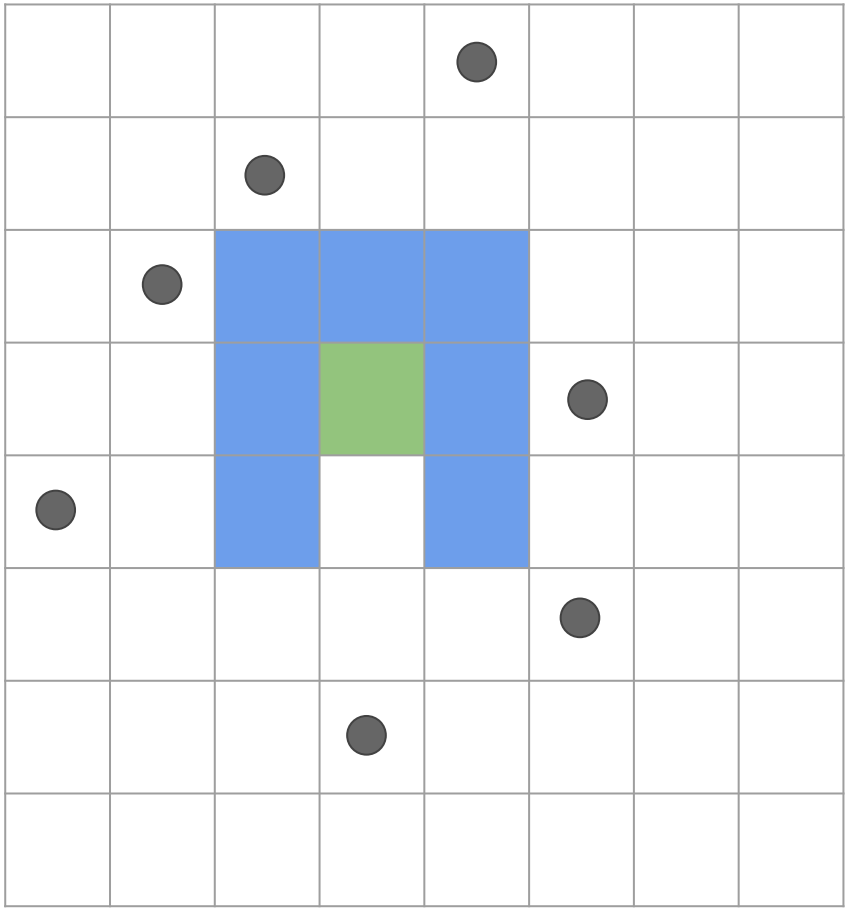
Revealed
To Be Checked
Mine



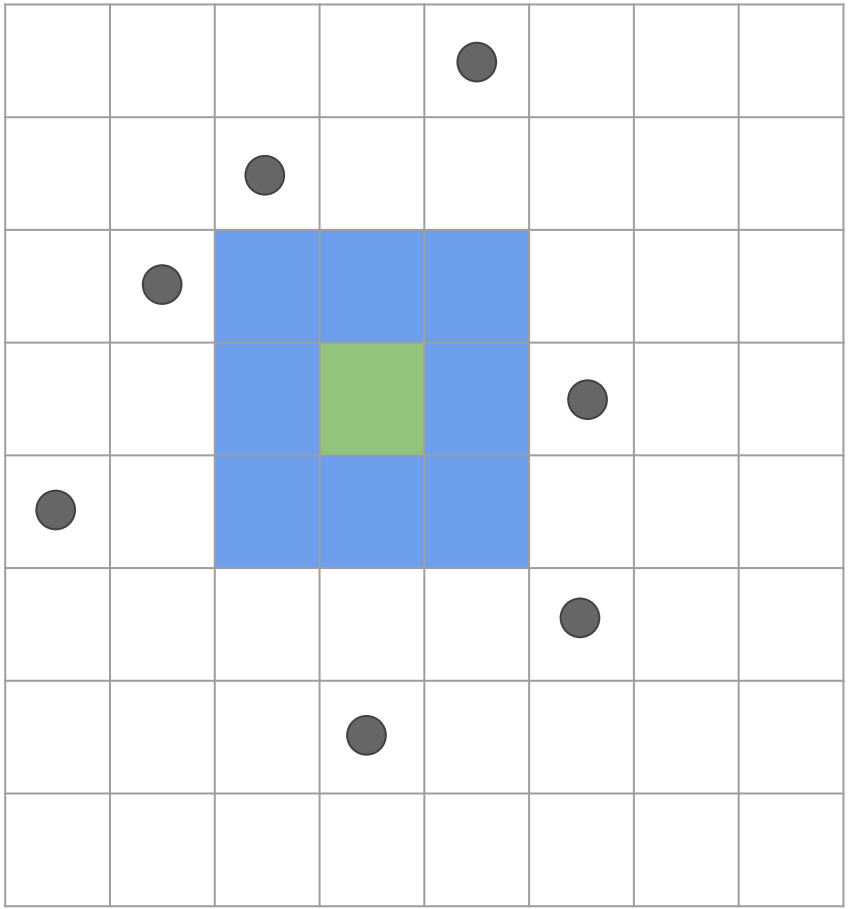
Revealed
To Be Checked
Mine



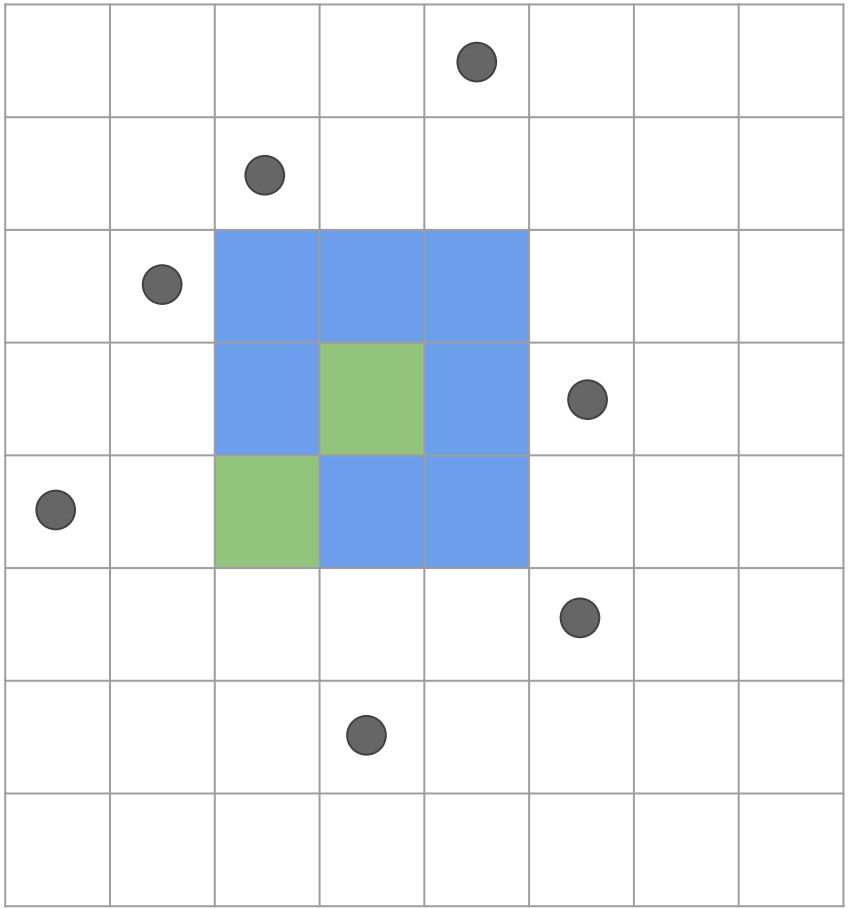
Revealed
To Be Checked
Mine



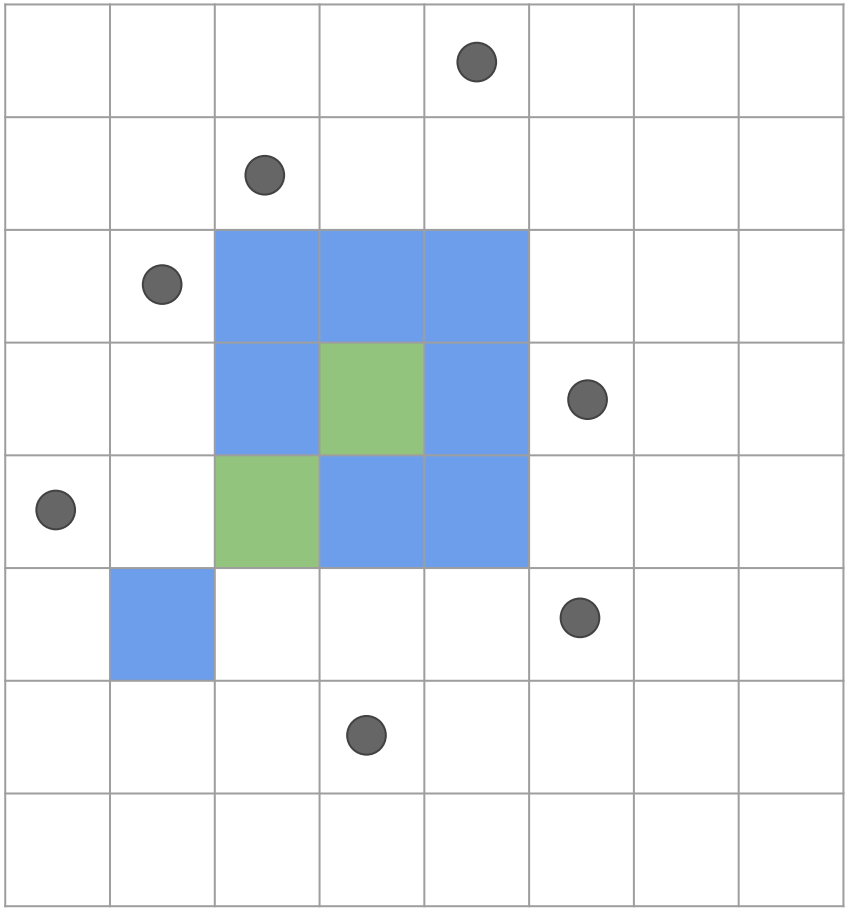
Revealed
To Be Checked
Mine



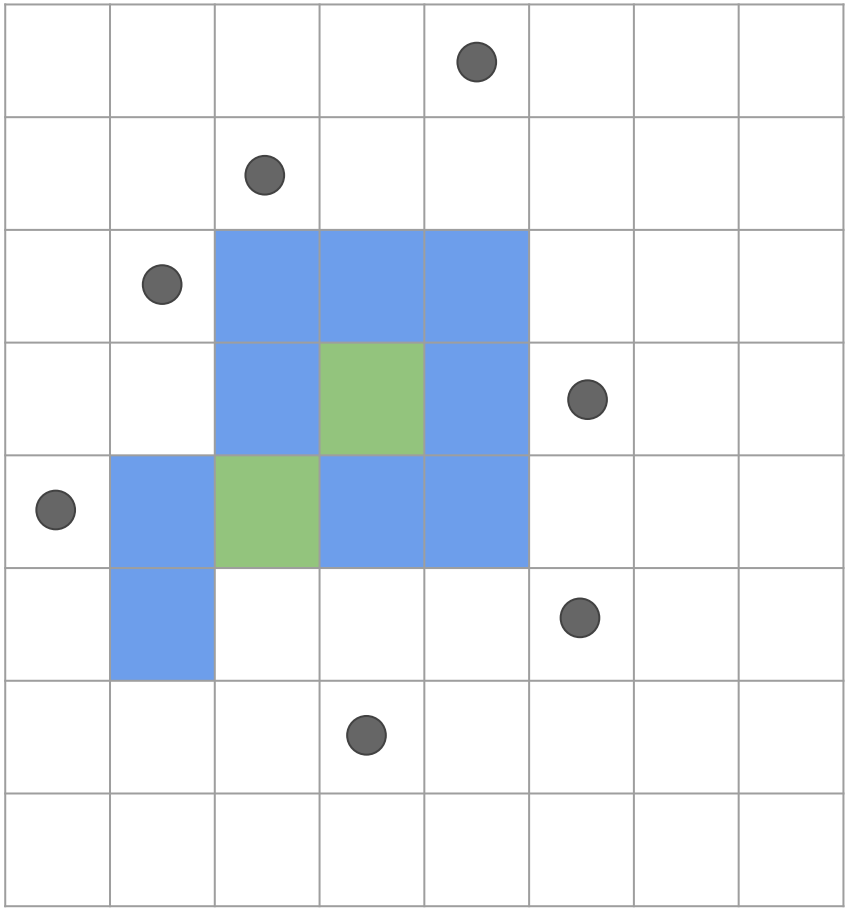
Revealed
To Be Checked
Mine



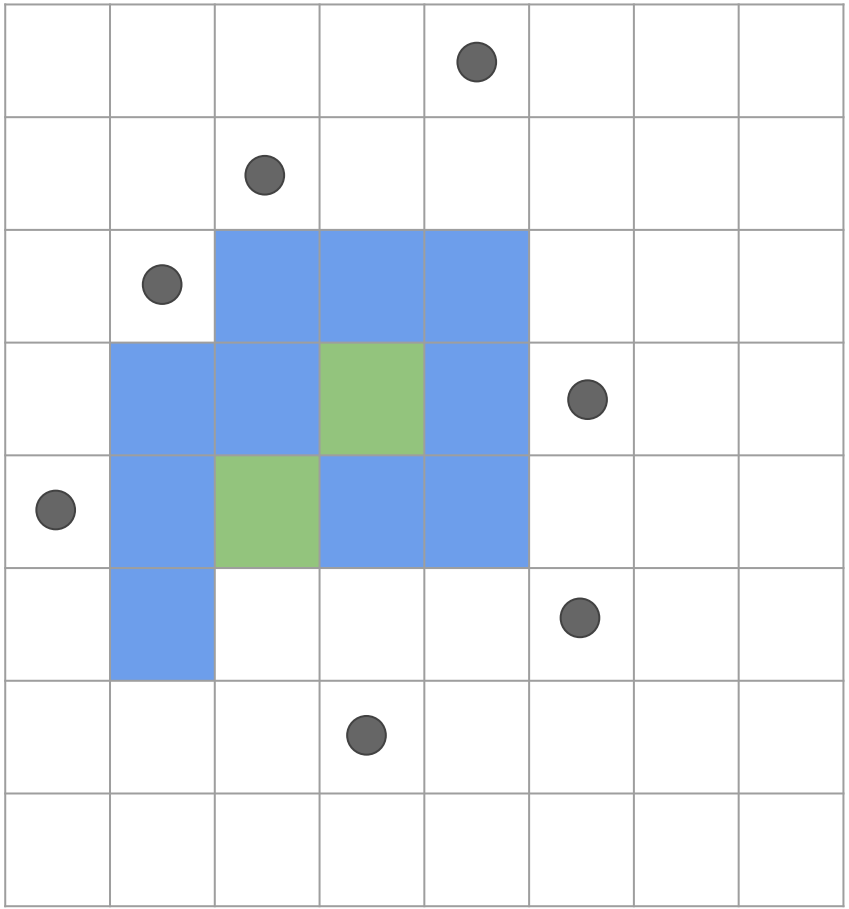
Revealed
To Be Checked
Mine



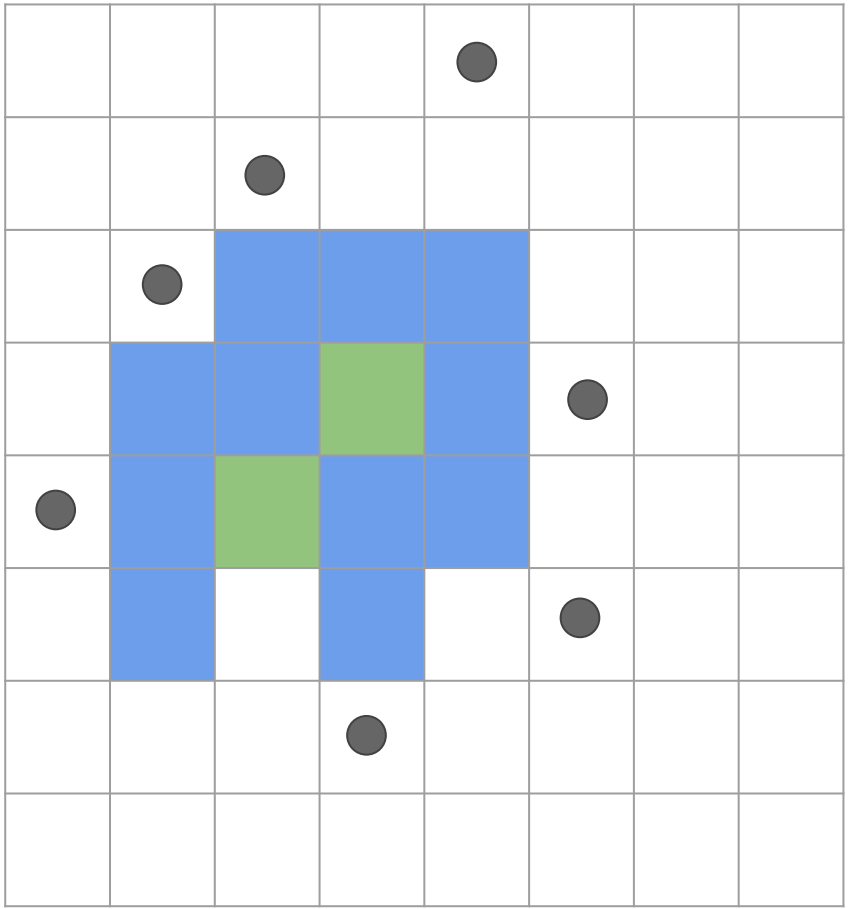
Revealed
To Be Checked
Mine



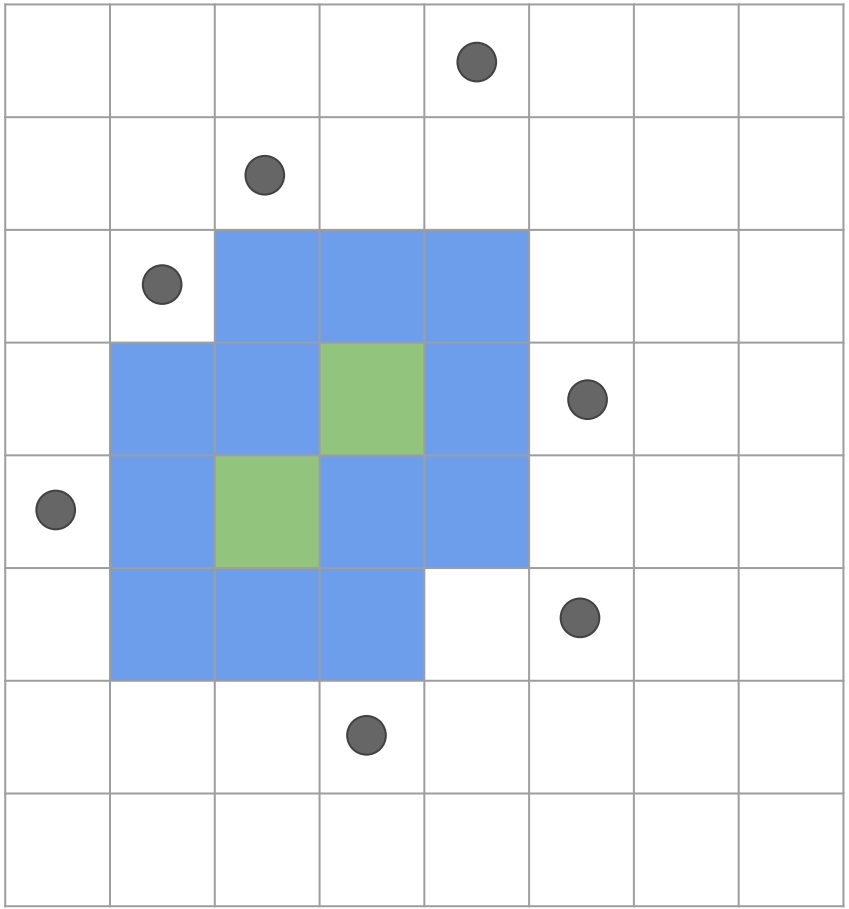
Revealed
To Be Checked
Mine



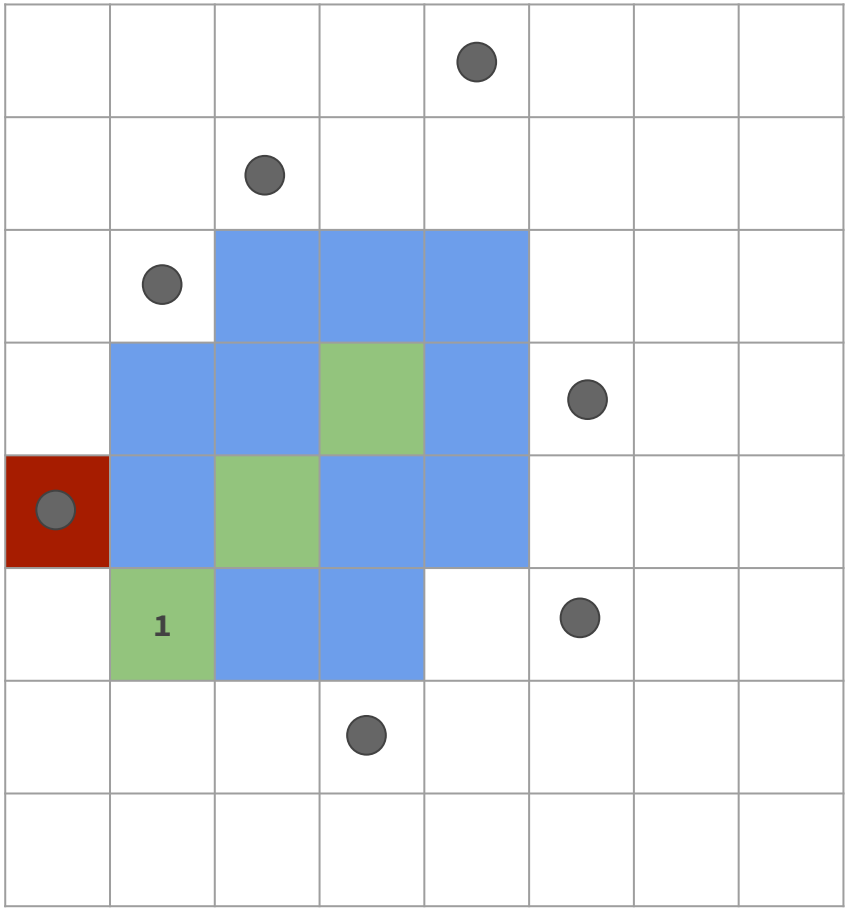
Revealed
To Be Checked
Mine



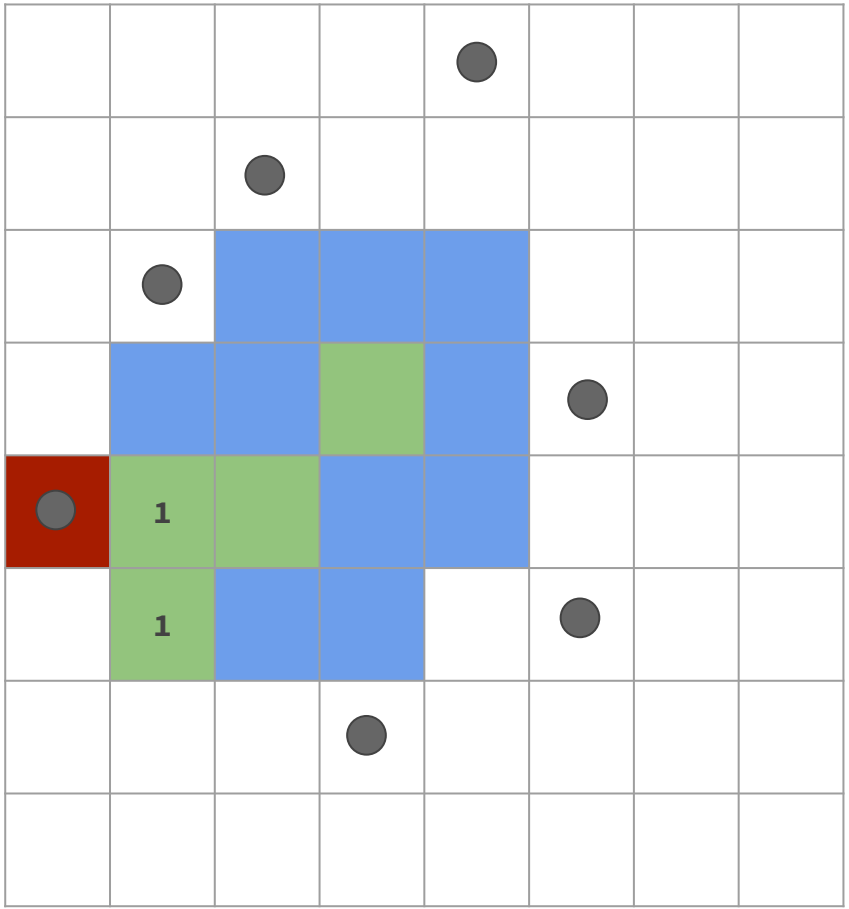
Revealed
To Be Checked
Mine



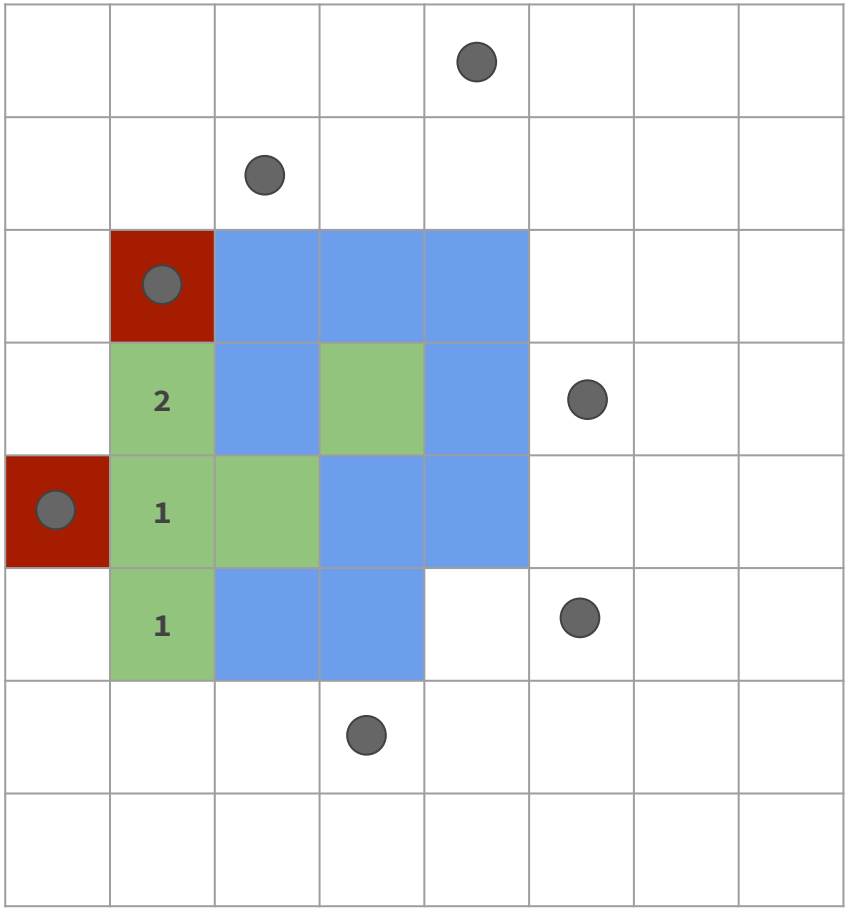
Revealed
To Be Checked
Mine



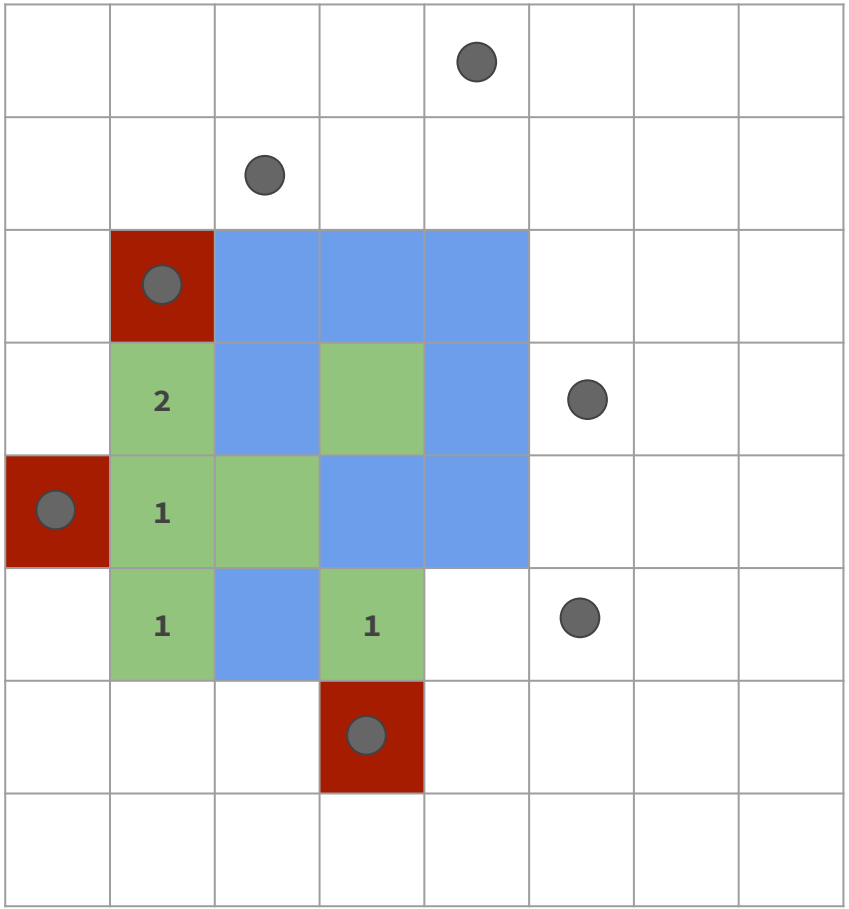
Revealed
To Be Checked
Mine



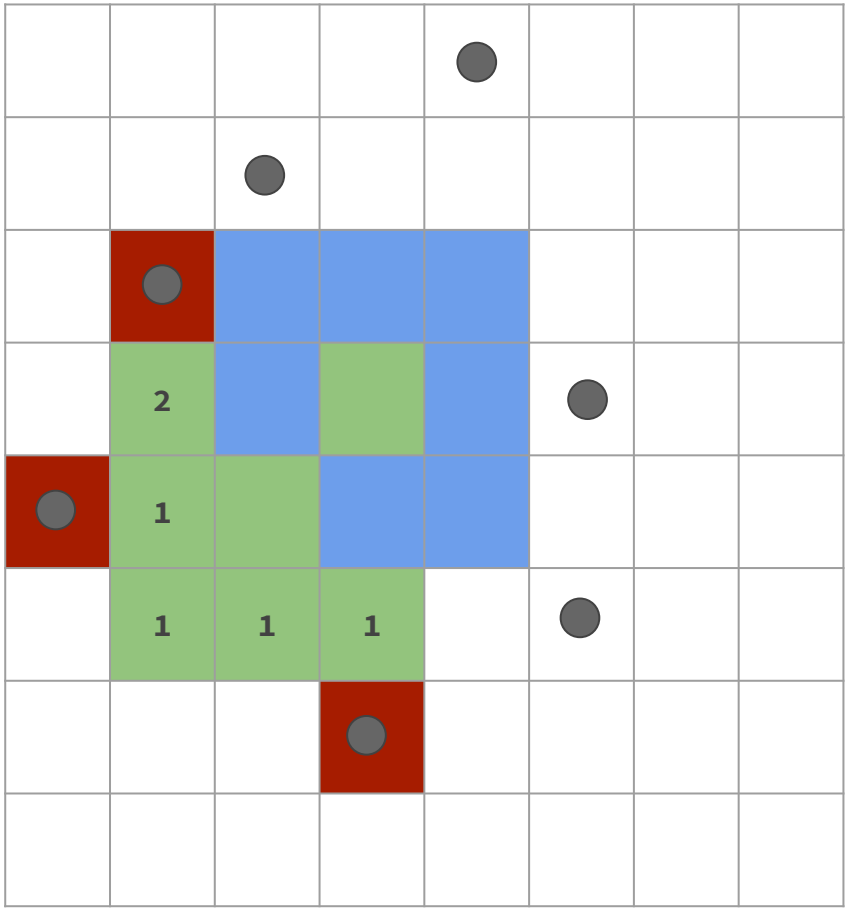
Revealed
To Be Checked
Mine



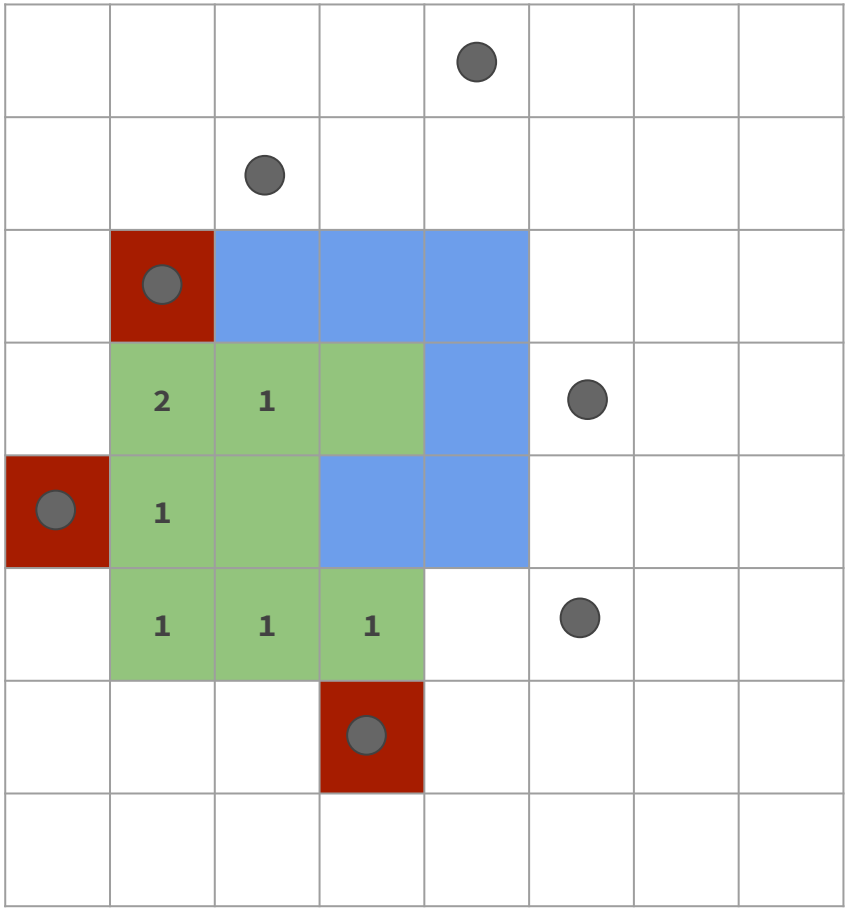
Revealed
To Be Checked
Mine



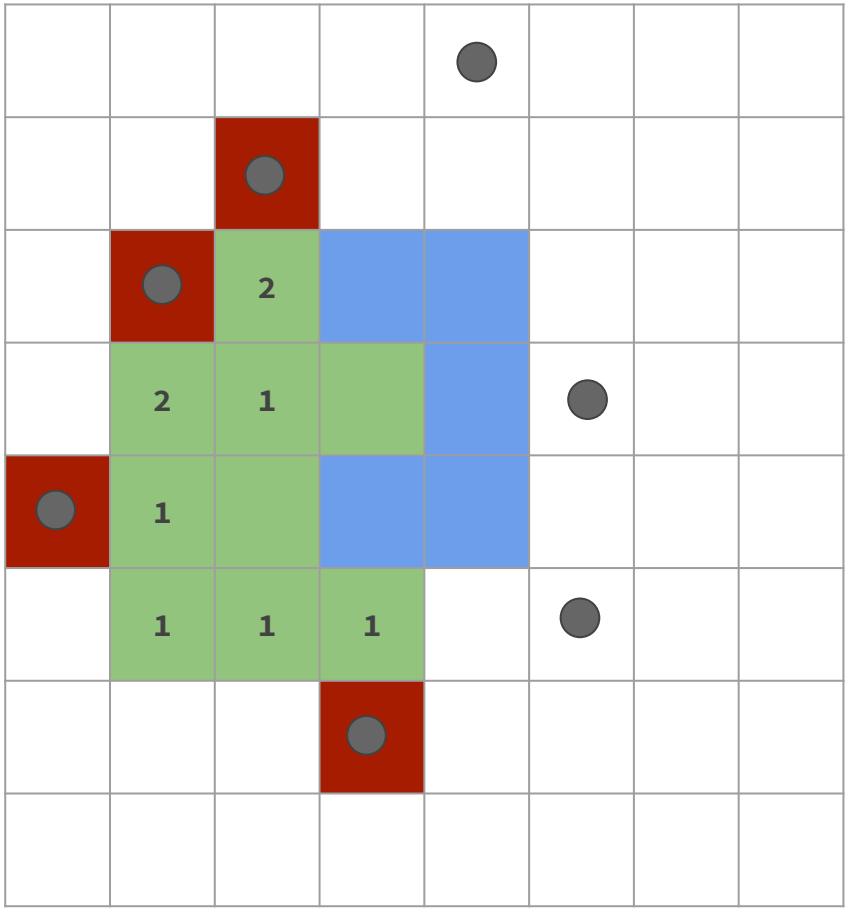
Revealed
To Be Checked
Mine



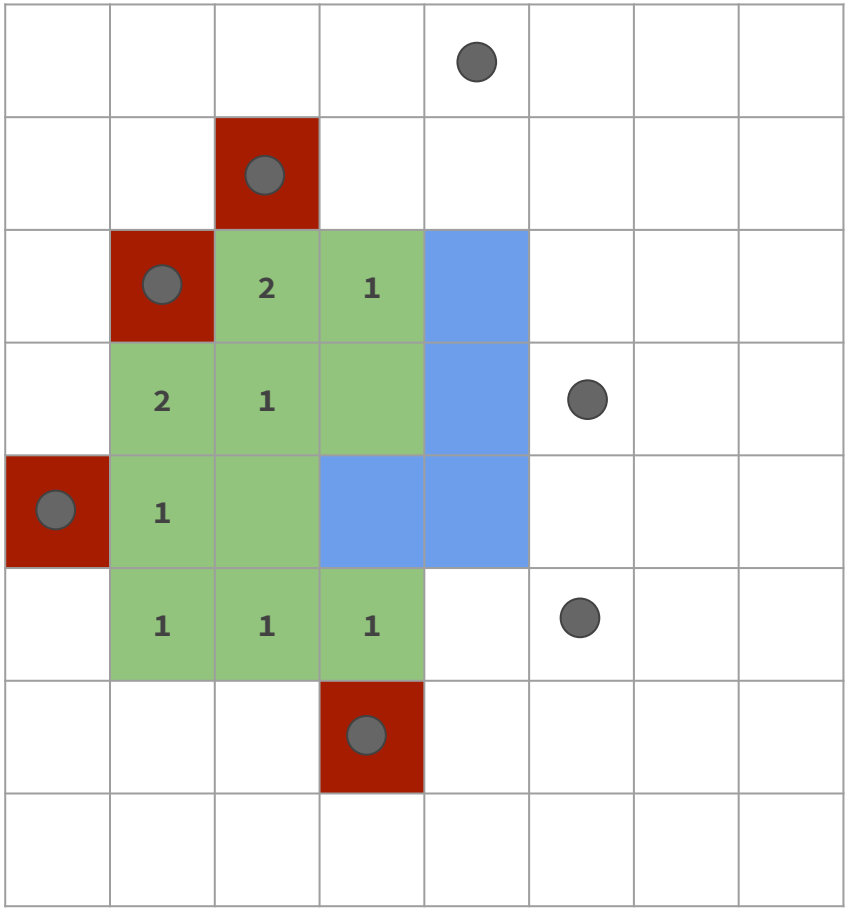
Revealed
To Be Checked
Mine



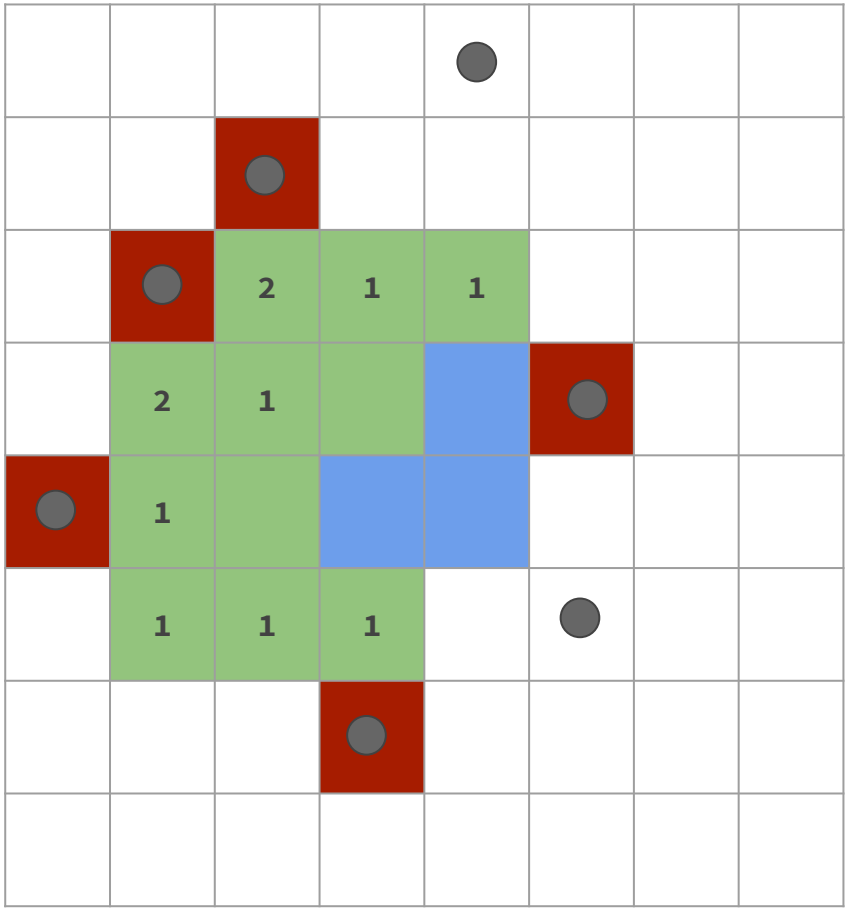
Revealed
To Be Checked
Mine



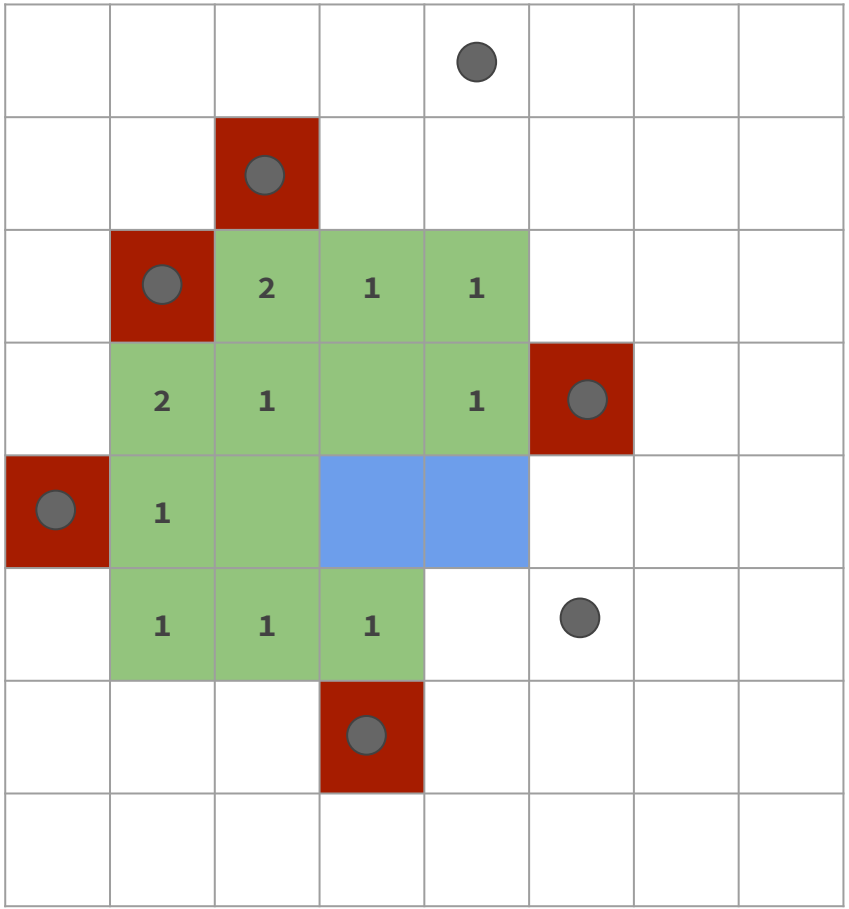
Revealed
To Be Checked
Mine



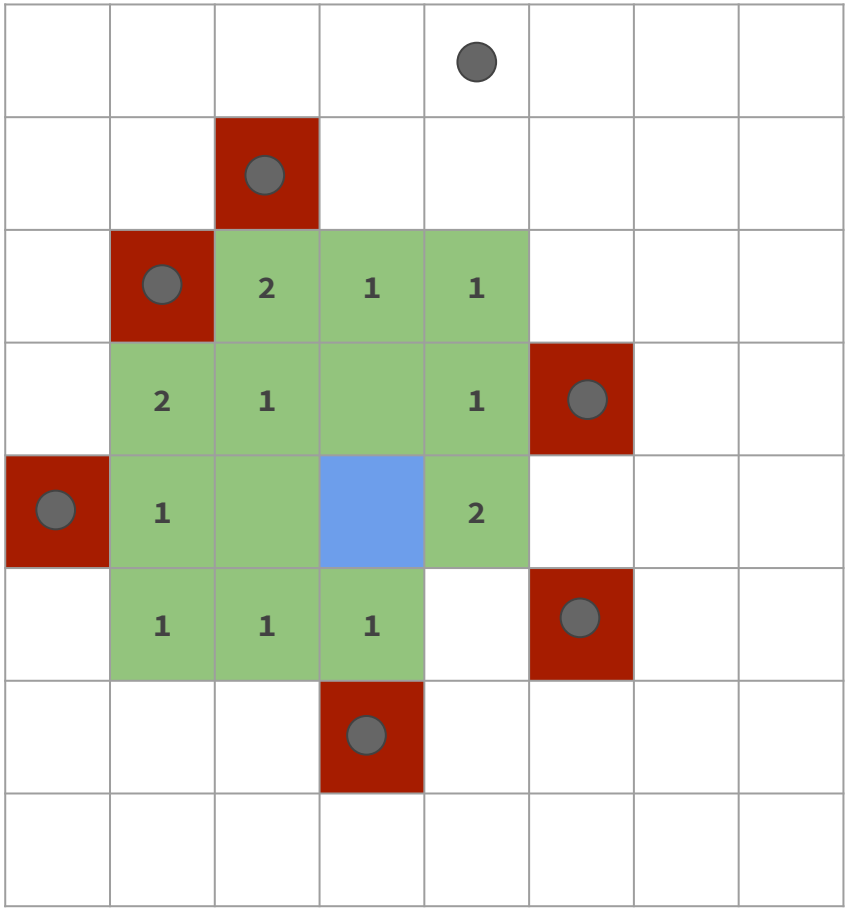
Revealed
To Be Checked
Mine



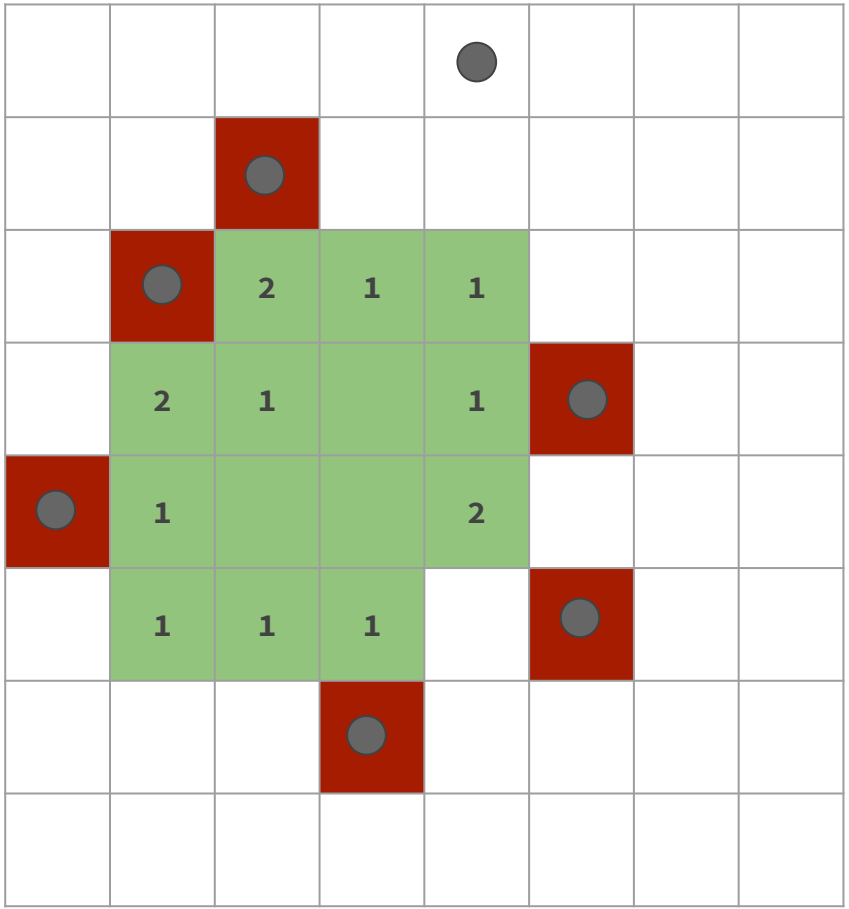
Revealed
To Be Checked
Mine



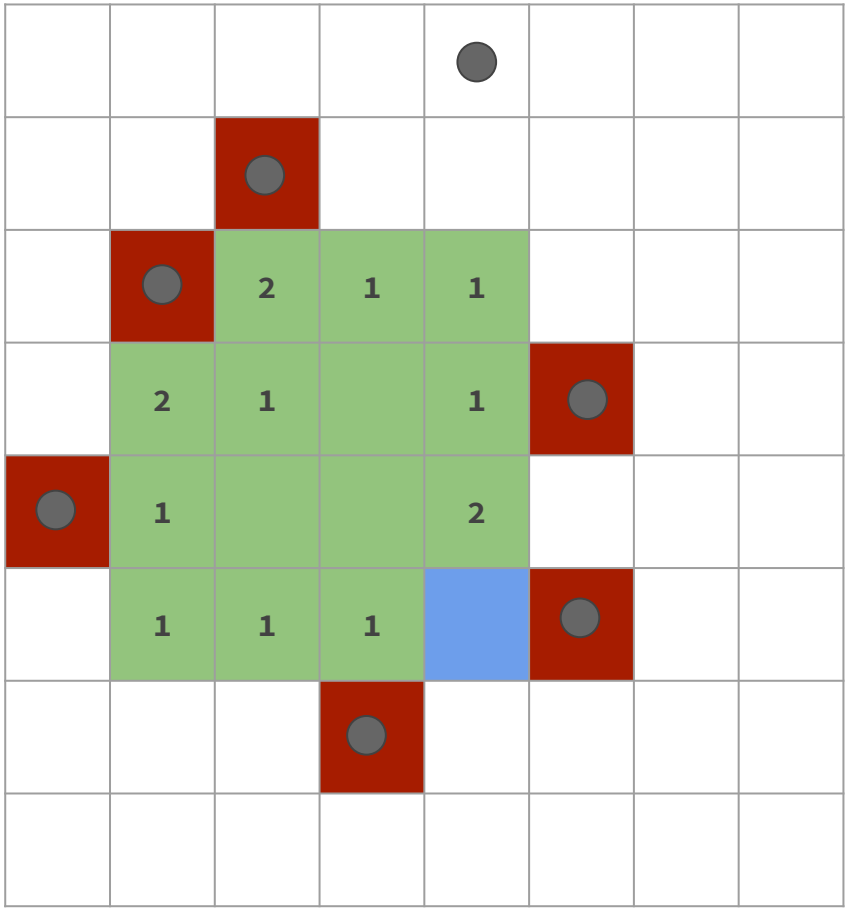
Revealed
To Be Checked
Mine



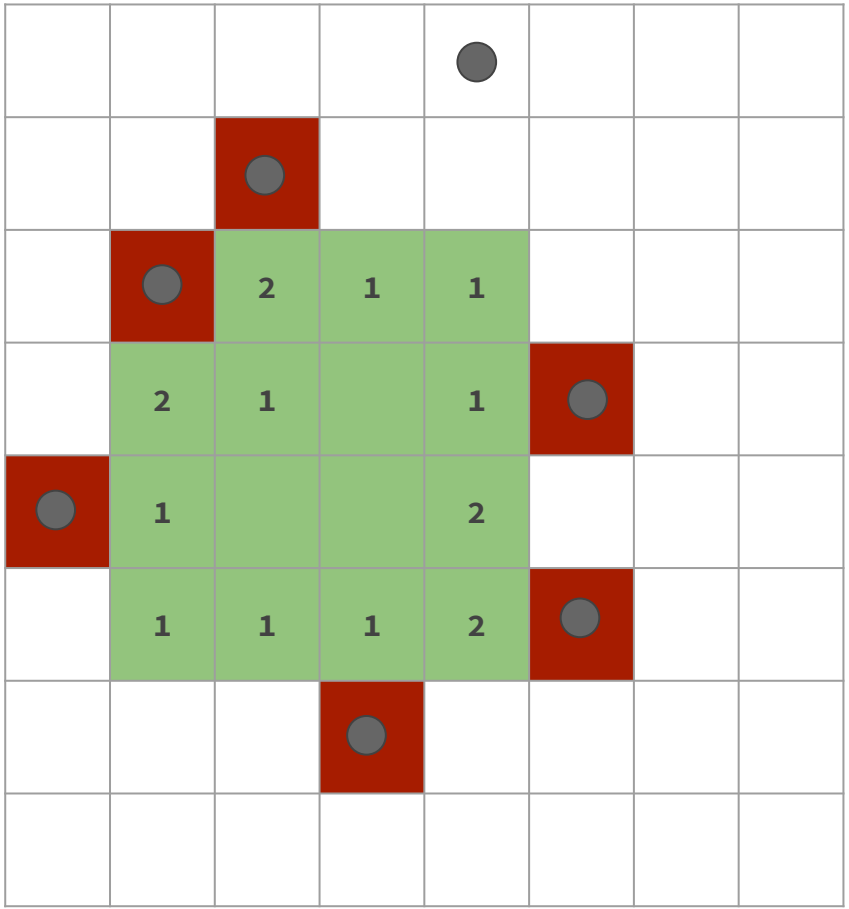
Revealed
To Be Checked
Mine



Revealed
To Be Checked
Mine



Revealed
To Be Checked
Mine



Revealed
To Be Checked
Mine