5118006-03 Data Structures

Homework 3. Minesweeper

11 Apr 2024

Shin Hong

* Revised on Apr 26

Outline

- Due date: 7 PM, Apr 30 Tue
- Task
  - construct a Minesweeper game using Queue
  - make a video demo
- Submission: via LMS
  - source code files
  - video file (or URL)
- Evaluation
  - test (50%)
  - source code quality (20%)
  - usability (10%)
  - video presentation (20%)

Tasks

- Construct a text-based Minesweeper game by completing `mine.c`
  - properly use `gqueue_t` for implementing serial opening of non-mined cells

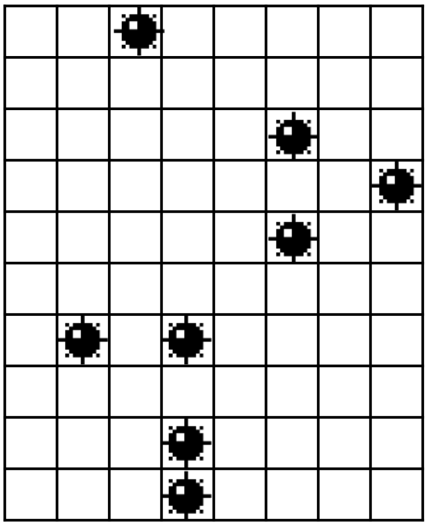
- Enhance user interface and add new features
  - add at least two features that make your Minesweeper game more fun and user-friendly
  - very welcome new ideas

- Record a video presentation less than 6 min to demonstrate your result and review code

Minesweeper Game (1/3)

- Reference
  - https://en.wikipedia.org/wiki/Minesweeper_(video_game)
  - https://minesweeper.online/
- A game board consists of *M* rows and *N* columns of cells, where mines are placed on K cells
  - $8 \leq M \leq 16$, $8 \leq N \leq 16$, $1 \leq K \leq 32$
  - the game board is given as a text file
    - the first line three integers M, N and K
    - each of the following K lines has two integers indicating the row and the column of a mine
    - example

```
10 8 8
0 2
3 7
6 3
6 1
4 5
2 5
9 3
8 3
```



Minesweeper Game (2/3)

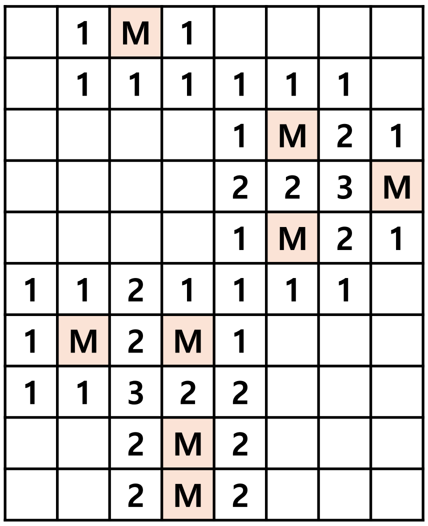
- Each cell is given as either *mined*, *numbered* or *safe*
  - a numbered cell is assigned with the number of mined cells among vertically, horizontally or diagonally adjacent cells
  - a safe cell is one that has no mined cells nearby
  - example

		●					
					●		
							●
					●		
	●		●				
			●				
			●				

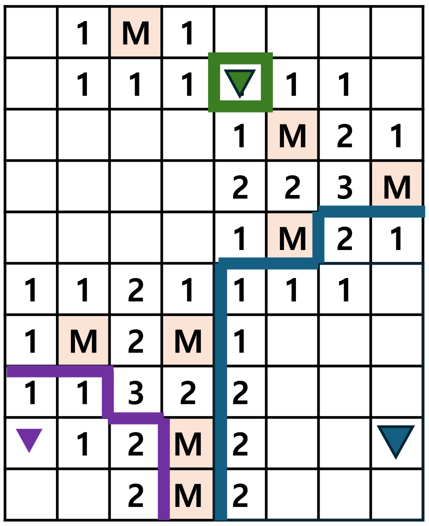
	1	M	1				
	1	1	1	1	1	1	
				1	M	2	1
				2	2	3	M
				1	M	2	1
1	1	2	1	1	1	1	
1	M	2	M	1			
1	1	3	2	2			
	1	2	M	2			
		2	M	2			

Minesweeper Game (3/3)

- Initially, all cells are closed
- The user wins the game if he/she marks all mined cells and opens all the other non-mined cells
  - at each turn, the user can mark (or unmark) one or multiple cells as mined
  - at each turn, the user must open a closed cell
  - the user loses if he/she opens a mined cell
  - when a safe gets opened, non-minded and unmarked nearby cells are automatically opened together, transitively.



	1	M	1				
	1	1	1	1	1	1	
				1	M	2	1
				2	2	3	M
				1	M	2	1
1	1	2	1	1	1	1	
1	M	2	M	1			
1	1	3	2	2			
		2	M	2			
		2	M	2			



	1	M	1				
	1	1	1	▼	1	1	
				1	M	2	1
				2	2	3	M
				1	M	2	1
1	1	2	1	1	1	1	
1	M	2	M	1			
1	1	3	2	2			
▼	1	2	M	2			▼
		2	M	2			

Make it More Fun and Usable

- `mine.c` gives only skeleton of a program
- UI and gaming features remain open
  - examples
    - scoring
    - scoreboard
    - time attack
    - board drawing
    - error handling

- Interesting and unique ideas will be appreciated!

Video Presentation

- Take a 6-min video for demonstrating your program reviewing the source code
  - explain in detail how a queue is used for opening adjacent non-mined cells if exist
  - demonstrate the new or enhanced features
  - the two team members must take video together and each one must take a part in presentation

- You can upload either one video file, an archive of multiple video files or a file indicating the URL of the video on the web

Remark

- Use of programming tools is not permitted
- No late submission will be accepted
- After submission, peer evaluation will follow
- The team members must work together at all activities for the homework
- Help desks will be offered