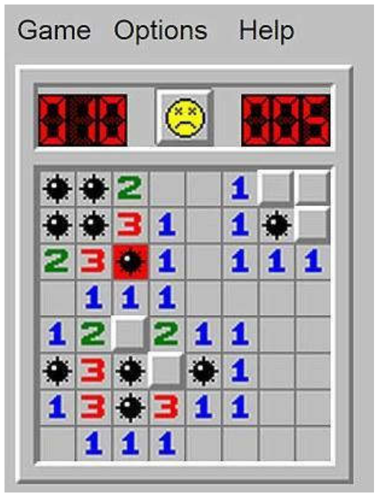
PROJECT

MINESWEEPER GAME

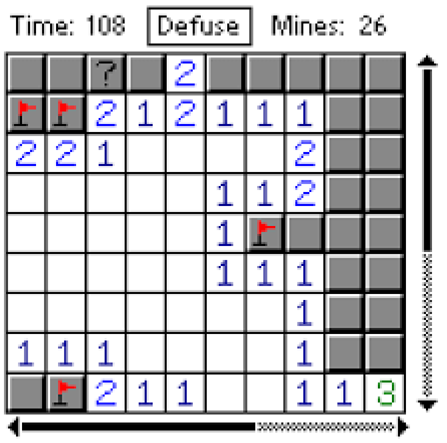
1. Introduction to Minesweeper Game

Minesweeper is a single-player puzzle video game. The objective of the game is to clear a rectangular board containing hidden "mines" without detonating any of them, with help from clues about the number of neighboring mines in each field. (Ref: https://en.wikipedia.org/wiki/Minesweeper_%28video_game%29)



To get a better feel of the game, it is highly recommended you play it at least once!

2. Descriptions



The playing area is known as the *grid*. The grid contains *w* * *h* number of tiles, where *w* (width) is the number of tiles horizontally and *h* (height) is the number of tiles vertically. Each tile has a *state*, which is one of the following: {Uncovered, Covered, Marked, Questionable}. In addition, a tile may contain a single mine. The total number of mines may not exceed the size (the number of tiles) of the grid.

The states are explained below:

- **Uncovered**: A tile is uncovered when the LEFT CLICK action is performed successfully. When uncovered, the tile will show the number of neighboring mine(s). (In the actual game, when there is no mine, the game appears as having one big flat

blank tile. Nothing happens when the user left clicks or right clicks on an uncovered tile.)
- **Covered**: When covered, the character '=' is shown. This is the initial state of every tile in the game.
- **Marked**: This state is toggled when a Covered tile is right clicked ONCE. Nothing happens when the user left clicks on a Marked tile.
- **Questionable**: This state is toggled when a Marked tile is right clicked ONCE. If right clicked again, the tile will revert back to the COVERED state. The user may uncover this tile normally by left clicking on the tile.

Besides marking flags, the player can make two kinds of moves to attempt to uncover squares:
- **Single guess:** The player clicks on a square with unknown state and no flag. Reveal the square, see if the player died, and put a number in it. If the square contains a 0, repeat this recursively for all the surrounding squares. This should be in a dedicated function, to separate it from the GUI's event handler, to make the recursion easy, and because it's reused in the multiguess.
- **Multiguess:** The player clicks on a square that is uncovered and known to be safe. If the number of flags surrounding equals the number in this square, we open up the unflagged squares using the same procedure as above.

Winning the game:

If the number of squares that are covered up is the same as the number of mines, then the player has won, even if they haven't placed a flag on every square.
When the player loses, it is customary to mark any incorrect guesses that they made, the remaining mines, and the mine that they stepped on.

3. Tasks

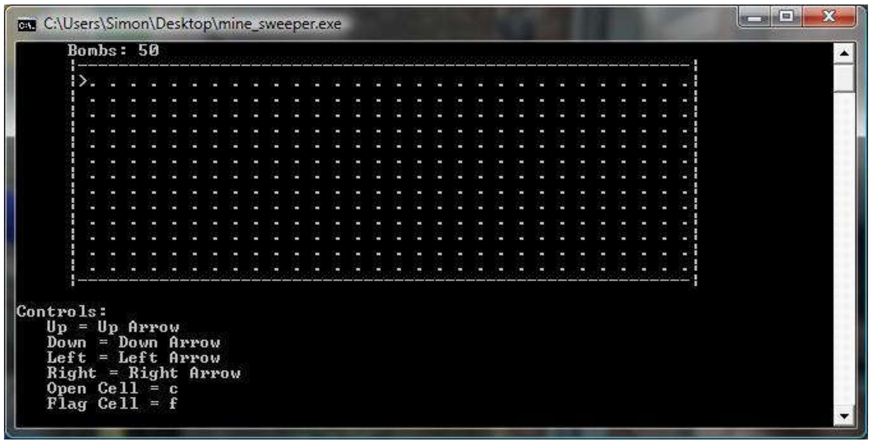
3.1. Level 1 (displaying game for user)

In this level, you will write a simple Minesweeper command line game program. You will notice that when you left click on a single empty tile with no neighboring mine, the area surrounding it is uncovered until a "border" with numbered tiles is reached. To keep things simple, you **do not** have to implement such automatic repeated uncovering of tiles. Hence, at every move you will only uncover **at most** a single tile. Furthermore, you will *only* display the grid and do not have to implement the other functionalities (such as the timer and number of remaining mines).
As such, you only have to consider the following cases.

- Left click (you may optionally use any other keyboard such as **L** character for left click, so on)
    - Tile must not be in a "Marked with a mine state" to uncover
    - Tile must be in either a "Questionable" or a "Covered" state
    - Tile must not already be uncovered
- Right click (you may optionally use any other keyboard such as **R** character for right click, so on)
    - The status is toggled in the following order:
      "Covered" => "Marked" => "Questionable" => "Covered"
    - You cannot toggle status on a tile that has already been uncovered

Winning or Losing:

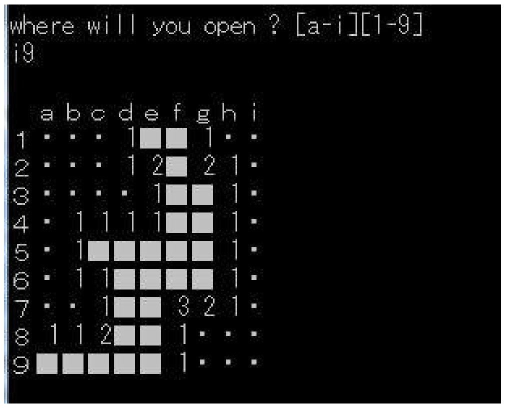
The game is won when ALL the non-mine tiles are uncovered. Therefore when winning, the remaining tiles are in one of the COVERED, MARKED, or QUESTIONABLE states, and they are all mines. The game is lost *immediately* when a mine is uncovered.
The following images illustrates some mine_sweeper program in console:



```
C:\Users\Simon\Desktop\mine_sweeper.exe
    Bombs: 50
    |-----------------------------------------------------------|
    |>. . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .|
    |-----------------------------------------------------------|

Controls:
    Up = Up Arrow
    Down = Down Arrow
    Left = Left Arrow
    Right = Right Arrow
    Open Cell = c
    Flag Cell = f
```

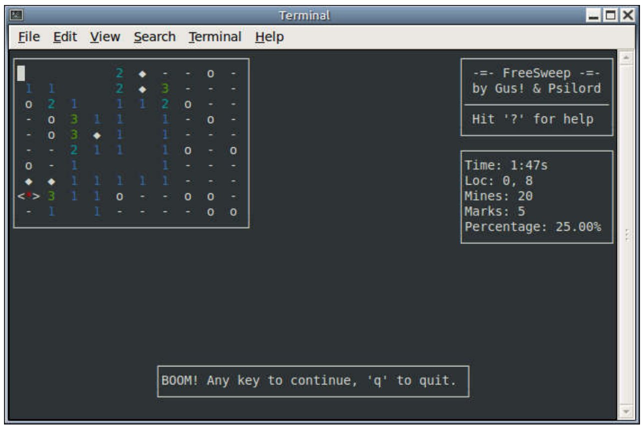
(Ref: http://x-volt.com/downloads/view.php?id=22)



```
where will you open ? [a-i][1-9]
i9

  a b c d e f g h i
1 · · · 1 ■ ■ 1 · ·
2 · · · 1 2 ■ 2 1 ·
3 · · · · 1 ■ ■ 1 ·
4 · 1 1 1 1 ■ ■ 1 ·
5 · 1 ■ ■ ■ ■ ■ 1 ·
6 · 1 1 ■ ■ ■ ■ 1 ·
7 · · 1 ■ ■ 3 2 1 ·
8 1 1 2 ■ ■ 1 · · ·
9 ■ ■ ■ ■ ■ 1 · · ·
```

(Ref: http://vivi.dyndns.org/tech/games/minesweeper.html)



```
█           2  ◆  -  -  o  -        -=- FreeSweep -=-
 1  1       2  ◆  3  -  -  -        by Gus! & Psilord
 o  2  1    1  1  2  o  -  -
 -  o  3  1  1     1  -  o  -       Hit '?' for help
 -  o  3  ◆  1     1  -  -  -
 -  -  2  1  1     1  o  -  o
 o  -  1           1  -  -  -       Time: 1:47s
 ◆  ◆  1  1  1  1  1  -  -  -       Loc: 0, 8
<*> 3  1  1  o  -  -  o  o  -       Mines: 20
 -  1     1  -  -  -  -  o  o       Marks: 5
                                    Percentage: 25.00%

            BOOM! Any key to continue, 'q' to quit.
```

(Ref: https://screenshots.debian.net/package/freesweep)

Write program with command line parameters:

Command line: GroupName_MineSweeper.exe level input.txt

where level = 1, input.txt contains game board.

Input:

- The first line of the input has two integers *w* and *h* separated by a space. The first integer *w* (width) is the number of tiles horizontally. The second integer *h* (height)is the number of tiles vertically. You may assume that $2 \le w \le 100$ and $2 \le h \le 100$ and *w* is not necessarily equal to *h*.
- The next *h* lines contain *w* characters in each line. Each character is a '-', a '*' or a number from 1 to 8, where '-' represents a safe and empty tile, '*' represents a tile with the mine, and the number indicates the number of adjacent mines. Characters are separated by a space. The first line is the top-most row of the grid and the last line is the bottom-most row of the grid.

We define the tile at the bottom-left corner as having the coordinates (1, 1).

For example:

```
9 9
* 1 - - - - - - -
1 1 - - - - 1 1 1
- - 1 2 2 2 2 * 1
- - 1 * * 2 * 2 1
1 1 1 2 2 2 1 2 1
* 1 - - - 1 1 2 *
1 1 1 1 1 1 * 2 1
- - 1 * 2 2 2 1 -
- - 1 1 2 * 1 - -
```

After read input into program, you need to cover all cells in preparation to start the game.

```
 |1|2|3|4|5|6|7|8|9|
1|_|_|_|_|_|_|_|_|_|
2|_|_|_|_|_|_|_|_|_|
3|_|_|_|_|_|_|_|_|_|
4|_|_|_|_|_|_|_|_|_|
5|_|_|_|_|_|_|_|_|_|
6|_|_|_|_|_|_|_|_|_|
7|_|_|_|_|_|_|_|_|_|
8|_|_|_|_|_|_|_|_|_|
9|_|_|_|_|_|_|_|_|_|
```

Progress:

- User will use mouse or keyboard to play game. Display guide to play before user starting. In progress, console screen will show board status reflecting the user's interaction. For example, if user click on any tiles, uncover it.
- The output is a single line containing the text "LOST" if any of the commands uncovered a mine. The output is "WON" if all non-mine tiles have been uncovered.

3.2. Level 2 (generating game for user)

Now, next task is to randomly generating the grid and placing mines in the field of cells. In generating the board, make sure you don't overlap mines. This takes some cleverness, and isn't done in most variants.
Because this algorithm could lead into creating a board with some mines grouped too much together, or worse very dispersed (thus boring to solve), you can then add extra validation when generating mine positions. For example, to ensure that at least 3 neighboring cells are not mines, or even perhaps favor limiting the number of mines that are too far from each other, etc

Command line: GroupName_MineSweeper.exe level input.txt

where level = 2, input.txt contains board size.
Input:

- The first line of the input has two integers *w* and *h* separated by a space.

3.3. Level 3 (uncovering all adjacent empty cells)

Uncovering an empty cell, is to be followed by uncovering all adjacent empty cells. Cells that have a mine or numbers should not be uncovered.

Command line: GroupName_MineSweeper.exe level input.txt

where level = 3, input.txt contains board size.

3.4. Level 4 (playing game with limited timer and multiple players)

You display countdown or countup timer while user playing. Game over if user has not finish yet when count to zero in countdown mode. Ranking user in count-up mode. Game allows multiple players, one player at a time. You also need to require user's name before starting.

Command line: GroupName_MineSweeper.exe level input.txt output.txt

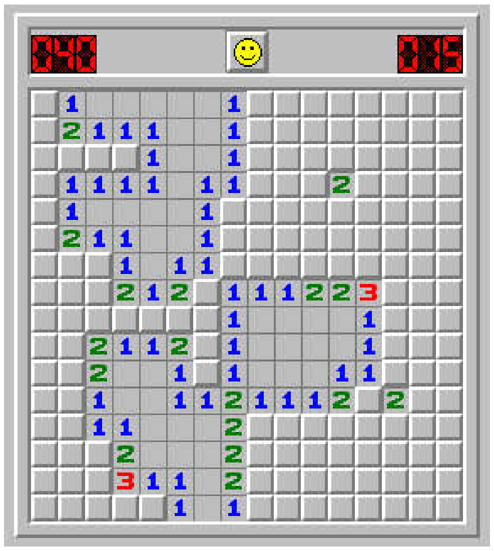
where level = 4, input.txt contains board size and list of previous users (name, score and timer), output.txt contains leaderboard of all user (include old users and new users). Score is calculated based on rate of detected mines (e.g. detect 20/40 mines, so score is 50 percent). The second is unit of time. Higher score is higher rank. If scores are equal, lower time is better.

For example:

John 50 120
Nam 32.2 80

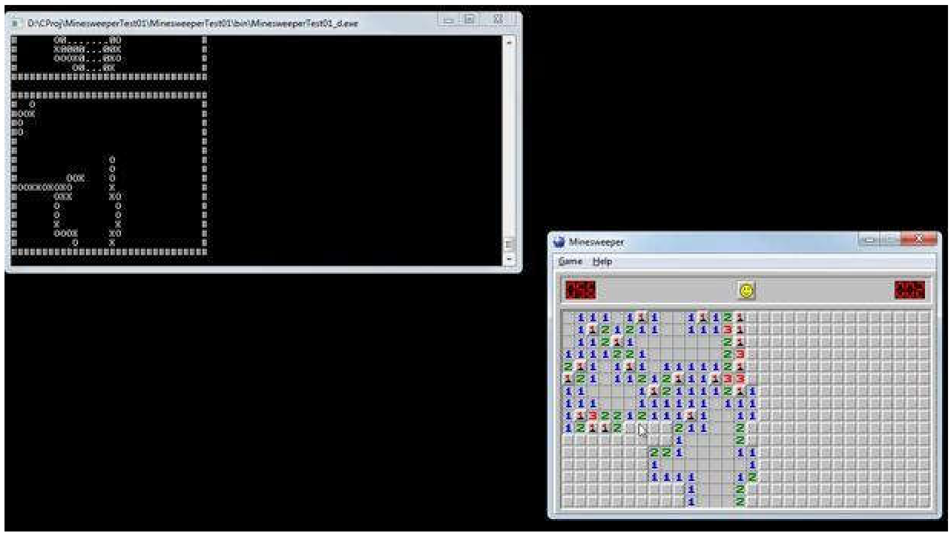
3.5. Level 5 (advanced level - displaying game with graphic)

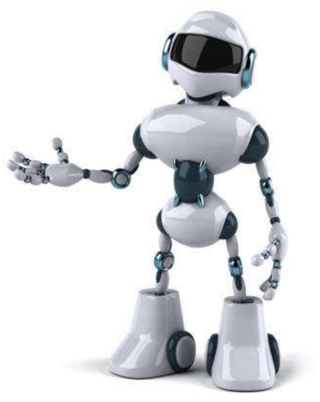
Write program in GUI (graphic user interface) mode. User can use mouse to click on tiles. For example:



3.6. Level 6 (super level - machine play game like a human)

With program to create a bot which it can play minesweeper on Windows OS automatically. When game is start, it detect and get information from game. After that, bot can click on game and solve game. You can use WinAPI to hook into the windows to click, and SDL to detect tiles.





Demo bot: https://www.youtube.com/watch?v=tcEd3Nigtok

4. Requirements

Each team include maximum of 4 members. The leader create a name for your team. Use teamwork skill to complete project.

Submit your works through Moodle by the due deadline. All files are compressed in zip/rar/7zip format.

Directory Structure:

- *GroupName-StudentId1-StudentId2-...*
  - *Document (.docx)*: report plan, the level of work accomplishment, the effort of each member, design, notes, guides, so on. More detail is better.
  - *Levelx*:
    - *Header*: .h
    - *Source*: .cpp
    - *Test Case*: inputs
    - ….

5. Assessment

Higher achieved level is higher score.
Some criterias and score ratio:

- The program is well documented: 30%
- Level 1: 30%
- Level 2: 20%
- Level 3: 10%
- Level 4: 10%
- Level 5: 20%
- Level 6: 20%

6. References

http://stackoverflow.com/questions/1738128/minesweeper-solving-algorithm
http://www.techuser.net/minecascade.html
http://www.dreamincode.net/forums/topic/44876-simple-minesweeper-c-source-code/
http://code.runnable.com/VWdt0iew3vhtRMRr/minesweeper-for-c%2B%2B
http://www.cs.ubc.ca/~acton/techTrek/MineSweeper/Minesweeper.pdf
http://www.cs.toronto.edu/~cvs/minesweeper/minesweeper.pdf
http://zetcode.com/tutorials/javagamestutorial/minesweeper/
https://www.codeproject.com/articles/439920/minesweeper
http://www.formauri.es/personal/pgimeno/compurec/Minesweeper.php
https://github.com/Workshop2/Minesweeper/tree/master/mine_sweeper
https://www.youtube.com/watch?v=c8wswUEfnrQ