MINESWEEPER

By:

Abdel Rahman Ahmed Mohamed Abdel Fattah

&

Ahmed Khaled Abdel Saied

Introduction

Minesweeper is a single player puzzle game.it was published by Microsoft Corporation in the last century. After that many companies published their own editions. Editions varied, but the main rule of the game is still the same “mark all locations of mines and uncover all other locations”. What makes an edition different are the features it provides and the flexibility of it. Thus, the focus in this edition was on the features, in addition to the main idea of course.

For example:

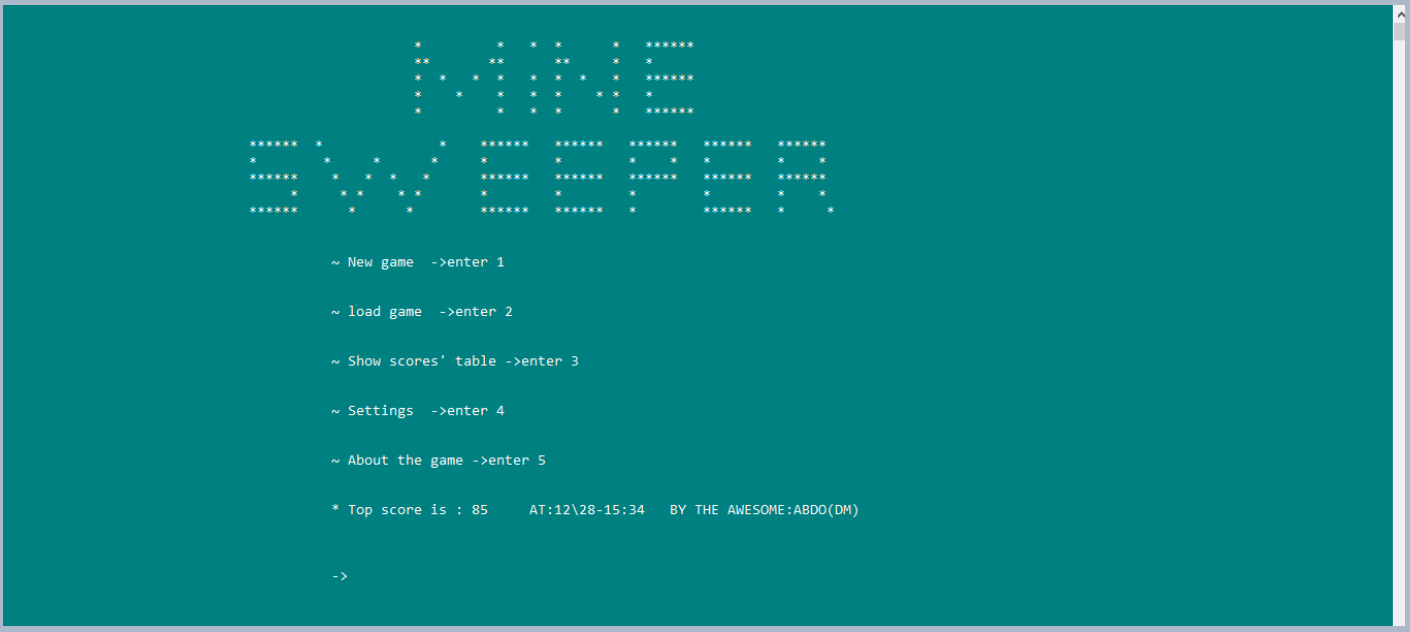
1. To start a game, the player have 3 different levels of difficulties to choose from.
2. Player can even customize his how lever by choosing the height, width and even number of mines before starting the game.
3. In settings, the colour of the text can be chosen, as well as game shapes.
4. Does the player want to know how the game works? Easy! Choose developer mode and the solution will be printed on the screen next to the game.
5. Arrow mode is also available to make the game more flexible and funny.
6. Save and load is available for more than one user.

And so many other features…

Game design

The game design contains 2 main categories: main menu and game layout

1) The “Main Menu”



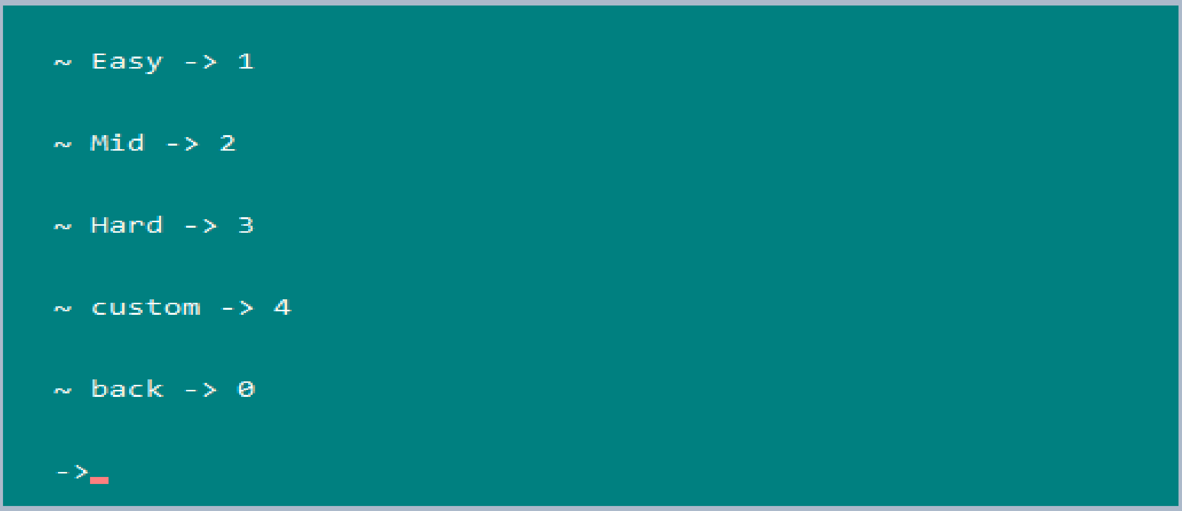
```
~ New game  ->enter 1

~ load game  ->enter 2

~ Show scores' table ->enter 3

~ Settings  ->enter 4

~ About the game ->enter 5

* Top score is : 85     AT:12\28-15:34   BY THE AWESOME:ABDO(DM)


->
```

User interface or the “main menu” is the first printed screen on the game. It consists of 5 sub-menus. And After them, top score is printed, the date the player scored it and the player name. The menus are:

1. New game:

By choosing “new game” new menu is opened containing options to start a game as shown in figure.

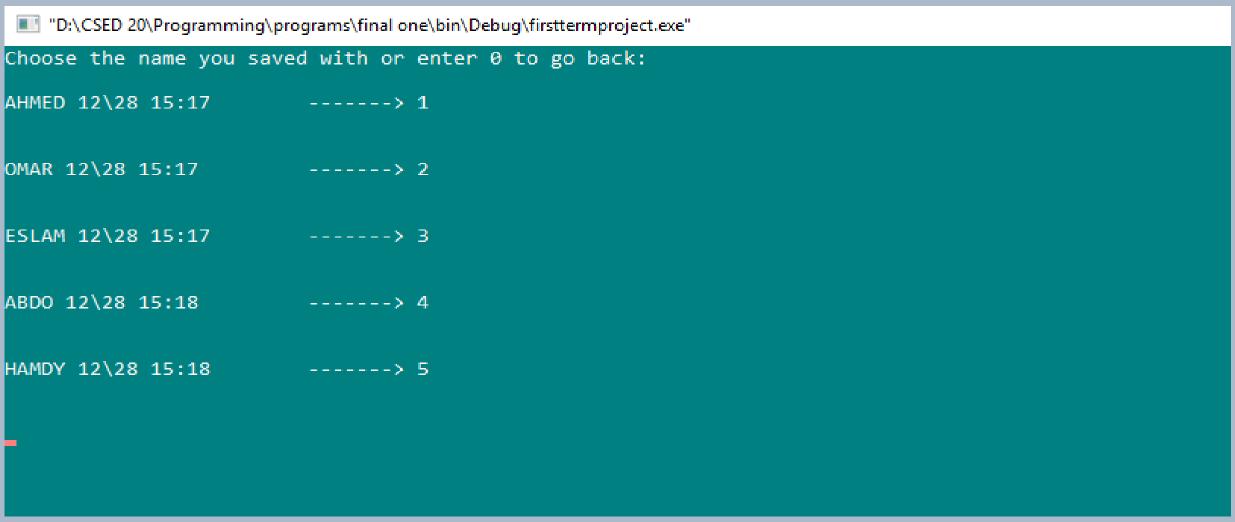
a. Easy:
   Start a game of height & width=10 and default number of mines =11 mines.
b. Medium:
   Start a game of height & width=15 and default number of mines =23 mines.
c. Hard:
   Start a game of height & width=20 and default number of mines =41 mines.
d. Custom:
   In this options user can choose any width, height and number of mines or user can just use the default number of mine.



```
~ Easy -> 1

~ Mid -> 2

~ Hard -> 3

~ custom -> 4

~ back -> 0

->
```

2. Load game:

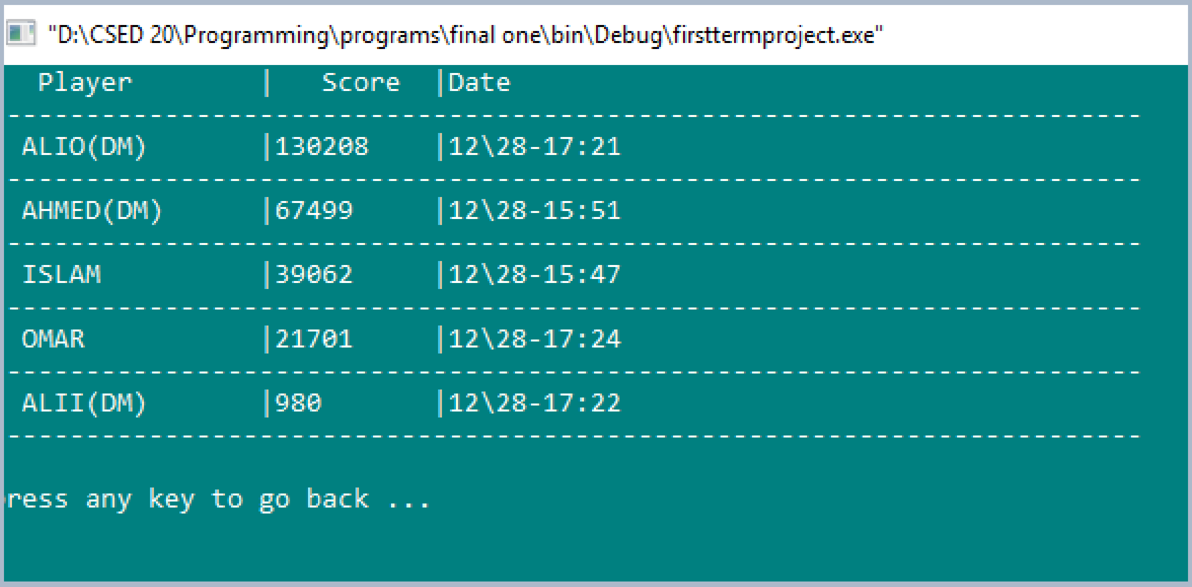
In this option user can load previously saved game. A numbered list of all saved games is printed, as shown in figure, and the user should choose the number next to his name, or just go back to main menu.



```
"D:\CSED 20\Programming\programs\final one\bin\Debug\firsttermproject.exe"
Choose the name you saved with or enter 0 to go back:
AHMED 12\28 15:17        -------> 1

OMAR 12\28 15:17         -------> 2

ESLAM 12\28 15:17        -------> 3

ABDO 12\28 15:18         -------> 4

HAMDY 12\28 15:18        -------> 5
```

3. Show scores' table

This option shows the rank table of the players who won. Each row of the table contains player name, player top score and the date of this score. If player won using "developer mode", the string (DM) will be printed next to his name.



```
"D:\CSED 20\Programming\programs\final one\bin\Debug\firsttermproject.exe"
  Player         |    Score   |Date
-------------------------------------------------------------------------
 ALIO(DM)        |130208      |12\28-17:21
-------------------------------------------------------------------------
 AHMED(DM)       |67499       |12\28-15:51
-------------------------------------------------------------------------
 ISLAM           |39062       |12\28-15:47
-------------------------------------------------------------------------
 OMAR            |21701       |12\28-17:24
-------------------------------------------------------------------------
 ALII(DM)        |980         |12\28-17:22
-------------------------------------------------------------------------

ress any key to go back ...
```

4. Settings

In this menu user can change and edit many features to fit his own desires. In the setting menu user can:

1. Change Game colour

User can change the background and printed text colour from a list of different colours.

2. Activate "full screen" mode

This options centers the printed text to make it easier to read

3. Activate developer mode

By activating this option the solved matrix of the game will be printed next to the main matrix. This option helps debugging and understanding the rules of the game.

4. Change game shape

In this option user can change the shapes printed on the screen for example: user can change the shape of unopened cell from ‘X’ to ‘Q’. So any unopened cell will be printed as ‘Q’. At any time user can reset default shapes.

5. Activate arrow mode

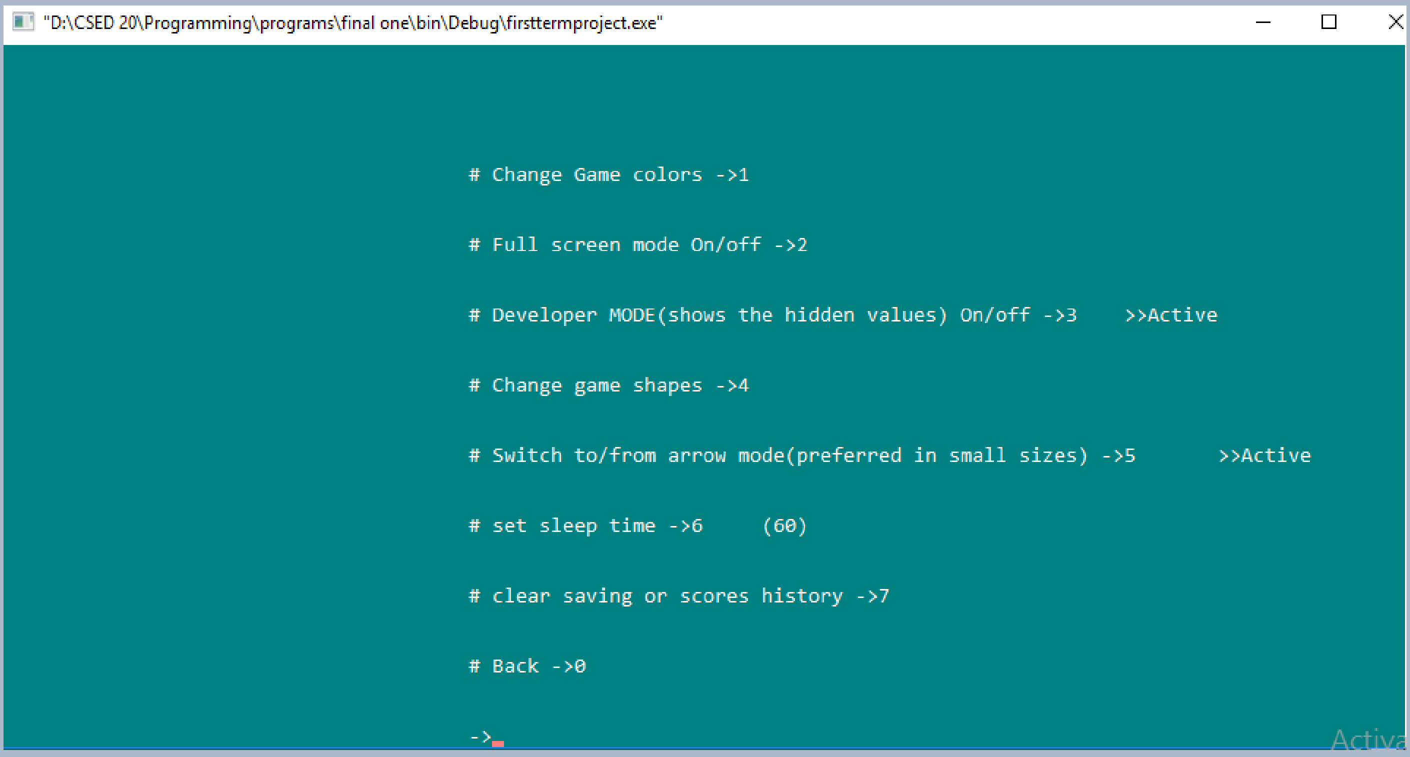
Actually this option is a masterpiece.by activating this option player can move through the matrix using arrows. Instead of entering the number of the cell the user wants to open or flag, this option makes it easier to reach it by just arrows. It is preferred to use this option in small sizes.

6. Set sleep time

In this option user can change the time after which the time will be updated in the game.

7. Clear history

As well as loading the saved games, deleting it is available to. Player can delete all saved games, all saved scores, or just delete both of them and start over.

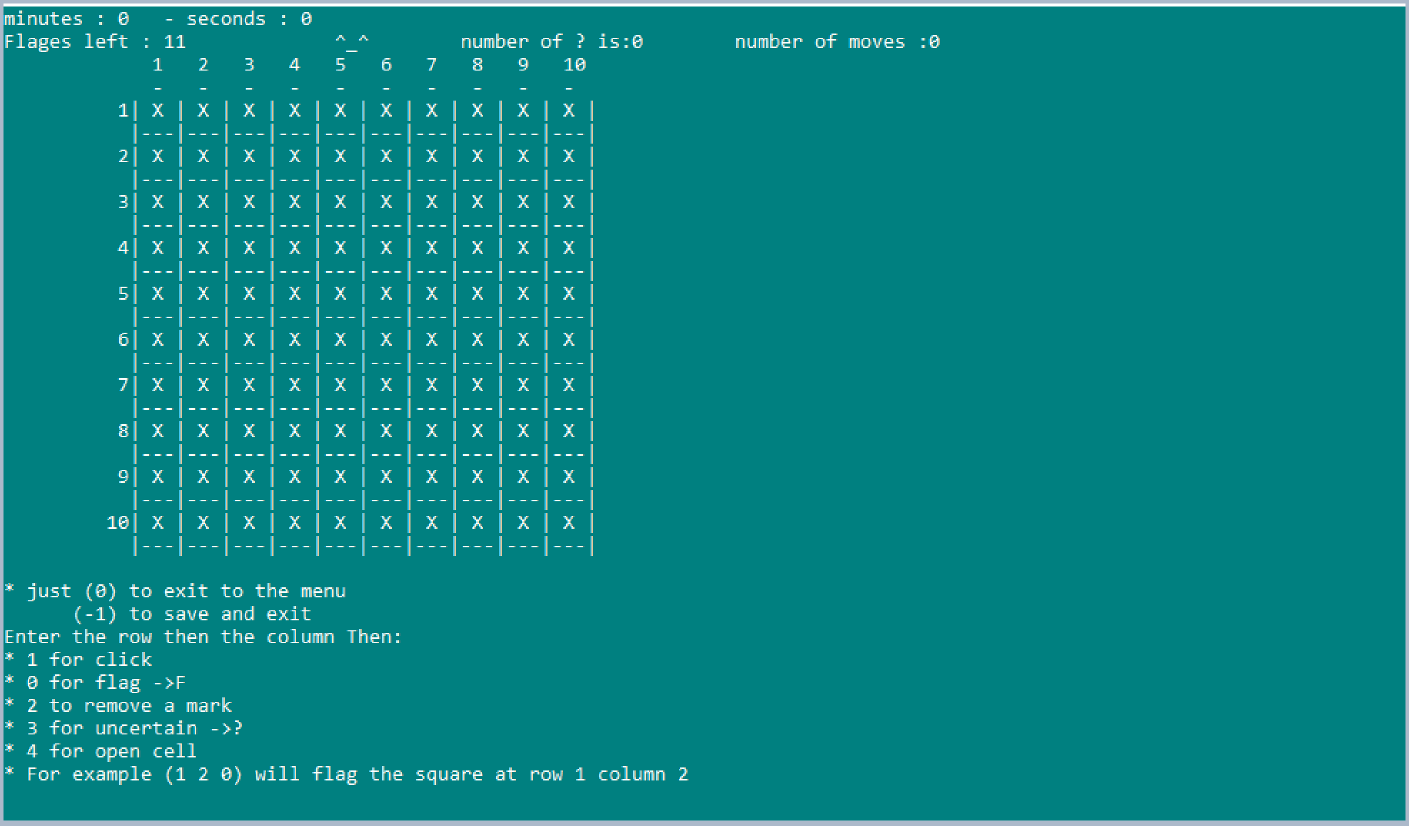


```
"D:\CSED 20\Programming\programs\final one\bin\Debug\firsttermproject.exe"

# Change Game colors ->1

# Full screen mode On/off ->2

# Developer MODE(shows the hidden values) On/off ->3    >>Active

# Change game shapes ->4

# Switch to/from arrow mode(preferred in small sizes) ->5      >>Active

# set sleep time ->6     (60)

# clear saving or scores history ->7

# Back ->0

->
```

5. About the game:

This menu contains an abstract about the game, its history, how to play and the game features.

2) Game layout



```
minutes : 0    - seconds : 0
Flages left : 11                ^_^        number of ? is:0        number of moves :0
             1   2   3   4   5   6   7   8   9   10
             -   -   -   -   -   -   -   -   -   -
          1| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          2| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          3| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          4| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          5| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          6| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          7| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          8| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          9| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
         10| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|

* just (0) to exit to the menu
      (-1) to save and exit
Enter the row then the column Then:
* 1 for click
* 0 for flag ->F
* 2 to remove a mark
* 3 for uncertain ->?
* 4 for open cell
* For example (1 2 0) will flag the square at row 1 column 2
```

After starting the game, the main layout is printed

The first line contains time spent playing the game in minutes and seconds. Time is updated every time the player makes a move. It is also updated every 1 minute. The second line contains:

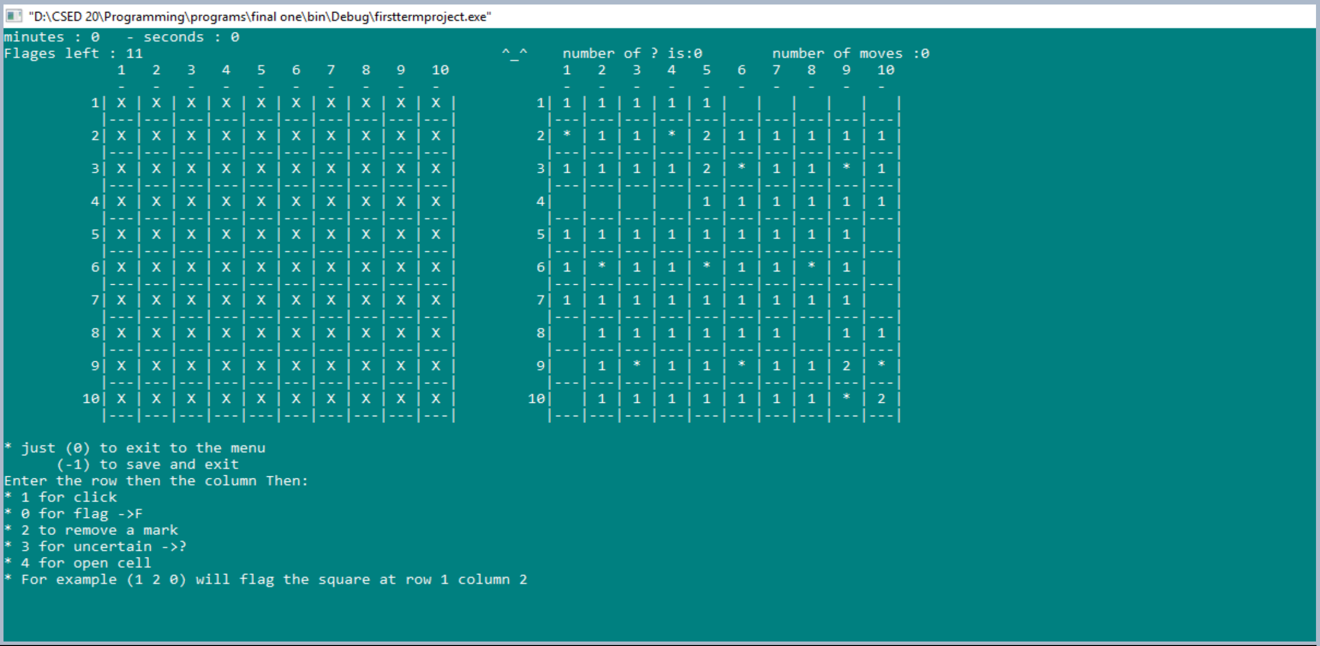
Number of flags left.

Number of question marks made by the user.

Number of moves made by the user.

After that, the main matrix is printed and updated after every move. Under the main matrix some information about how to play are printed.

If developer mode is activated, the solved cell is printed as shown in figure:



```
"D:\CSED 20\Programming\programs\final one\bin\Debug\firsttermproject.exe"
minutes : 0    - seconds : 0
Flages left : 11                                         ^_^    number of ? is:0        number of moves :0
             1   2   3   4   5   6   7   8   9   10                1   2   3   4   5   6   7   8   9   10
             -   -   -   -   -   -   -   -   -   -                 -   -   -   -   -   -   -   -   -   -
          1| X | X | X | X | X | X | X | X | X | X |            1| 1 | 1 | 1 | 1 | 1 |   |   |   |   |   |
           |---|---|---|---|---|---|---|---|---|---|             |---|---|---|---|---|---|---|---|---|---|
          2| X | X | X | X | X | X | X | X | X | X |            2| * | 1 | 1 | * | 2 | 1 | 1 | 1 | 1 | 1 |
           |---|---|---|---|---|---|---|---|---|---|             |---|---|---|---|---|---|---|---|---|---|
          3| X | X | X | X | X | X | X | X | X | X |            3| 1 | 1 | 1 | 1 | 2 | * | 1 | 1 | * | 1 |
           |---|---|---|---|---|---|---|---|---|---|             |---|---|---|---|---|---|---|---|---|---|
          4| X | X | X | X | X | X | X | X | X | X |            4|   |   |   |   | 1 | 1 | 1 | 1 | 1 | 1 |
           |---|---|---|---|---|---|---|---|---|---|             |---|---|---|---|---|---|---|---|---|---|
          5| X | X | X | X | X | X | X | X | X | X |            5| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |   |
           |---|---|---|---|---|---|---|---|---|---|             |---|---|---|---|---|---|---|---|---|---|
          6| X | X | X | X | X | X | X | X | X | X |            6| 1 | * | 1 | 1 | * | 1 | 1 | * | 1 |   |
           |---|---|---|---|---|---|---|---|---|---|             |---|---|---|---|---|---|---|---|---|---|
          7| X | X | X | X | X | X | X | X | X | X |            7| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |   |
           |---|---|---|---|---|---|---|---|---|---|             |---|---|---|---|---|---|---|---|---|---|
          8| X | X | X | X | X | X | X | X | X | X |            8|   | 1 | 1 | 1 | 1 | 1 | 1 |   | 1 | 1 |
           |---|---|---|---|---|---|---|---|---|---|             |---|---|---|---|---|---|---|---|---|---|
          9| X | X | X | X | X | X | X | X | X | X |            9|   | 1 | * | 1 | 1 | * | 1 | 1 | 2 | * |
           |---|---|---|---|---|---|---|---|---|---|             |---|---|---|---|---|---|---|---|---|---|
         10| X | X | X | X | X | X | X | X | X | X |           10|   | 1 | 1 | 1 | 1 | 1 | 1 | 1 | * | 2 |
           |---|---|---|---|---|---|---|---|---|---|             |---|---|---|---|---|---|---|---|---|---|

* just (0) to exit to the menu
      (-1) to save and exit
Enter the row then the column Then:
* 1 for click
* 0 for flag ->F
* 2 to remove a mark
* 3 for uncertain ->?
* 4 for open cell
* For example (1 2 0) will flag the square at row 1 column 2
```

Design assumptions

In this part some ideas used in coding and some information about program will be clarified

1) Point structure

This type of structure is used to store the cells of the main matrix of the game. It consist of 3 elements:

1. Hidden:

   This integer variable is the solved value of the cell. Its value is a number of the surrounding mines, form 0 to 8, or 9 if it is a mine. It is generated randomly before the user opens any cell. And if the first cell the user opens is a mine, the hidden matrix is re-generated taking into consideration that the opened cell mustn't be a mine.

2. Value:

   This character variable is the printed value of the cell. It can be an unopened cell, a flag, or if the cell is opened it can be a number or a space.

3. Pointed:

   This variable is whether 1 or 0. If it is 1, so this is the last cell the user did an operation into. So in the main matrix only one cell have the value of 1. This variable is used in 2 situations only: if the user lost, this cell must have a mine so it will be printed as "!", or if arrow mode is activated, the arrows will be printed before and after the pointed cell.

2) Saving the game

When a game is saved, many data should be saved with it. In this part the saved data will be clarified.

Saving a game is done using 2 files: "names" and "saves".

1. Names file

   It contains all the names of the user who saved a game before .So if a new game is saved, first the new user name is compared with the names in this file. In addition, if anyone wants to load a game, a list of the names of this file is printed.

2. Saves file

   This file contains the main important data that need to be saved. It contains the "array.value" and "array.hidden" arrays. And also number of moves, mines, question marks, array height and width, whether "developer mode" is activated or not, and time passed before saving. Before every lump of data, player name is printed preceded by the character "@".

And this way makes it available for more than one user to save and load, only if his name is not repeated.

3) Score

To form the leaderboard, every winner score is saved. With the score, player's name is saved and the date of winning up to minutes. If a player won more than one time, the program compares the old score with the new one and saves the larger score. However, the time is updated every time the player win. This option is made to let the player know when he won last time. In addition, if developer mode is activated, the string (DM) will be printed next to player's name and it have all properties of normal names.

4) Developer mode (DM)

Developer mode is an option for developers and new players. What happens when user choose DM is that: the solved matrix will be printed next to the unsolved one during the game. However, if the game's width is larger than 20, it is automatically turns itself off and the solved array is printed in a file called "solved.txt".
This option is not only made for debugging and tracing the program, But it is also mate for new players to know how to play and learn the game rules before playing without any help. Developer mode can be switched on/off any time from settings.

5) Arrow mode

This part of the game is an option made to make the game easier. Instead of reading the cell number from the user. We use the pointed variable to get it. To move through the matrix use "w" for up, "s" for down, "a" for left and "d" for right. This way is easier and its probability to have bugs is less than the ordinary way.

6) Counting number of moves

In this player can make many moves, some of them are counted as a move and the others are not. The following list includes the counted moves:

1. Opening a cell
2. Opening an opened cell
3. Flagging a cell
4. Question marking a cell
5. Removing a mark from a cell

And the following list includes some uncounted moves

1. Moving through the matrix using arrow mode
2. Opening a marked cell
3. Opening an empty cell

Or any other move that does not change the matrix.

Data structure

In this game, various data structure are used to store and manipulate data.

1. Point structure:

This type of structure is used to store the main game matrix. It consists of 3 variables:
Value: the value of the cell printed on the screen
Hidden: the value of the solved game matrix cell.
Pointed: this variable takes the value of 0 or 1 to indicate if the user chose this cell or not. To know more about it go to Point structure.

2. Point screen array:

It is an array of structures of type point which is the main game matrix.

3. Playerscore structure

This type of structure is made to store players score and contains of 3 variables
Name: a string of the player's name
Score: the number of the top score the player scored
Date: a string contains the date the player scored his high score

4. Playerscore array:

Array of structures of Playerscore used to store score in order to print rank table or to insert new score to the list

In this program 3 different text files are used to store data too:

Saves: contains all the information of the saved games each with the player name
Names: contains the names of the players who saved games before. This files makes it easier to print the load list. To read more about saving go to saving the game.
Score: in this file, top score of each player is saved. Scores in this file are sorted so that the top player is in the beginning of the file.

5. Variables:

   So many variable are used in this program. In this part, the important variables only will be clarified.

   i. Width & height: dimensions of the game matrix.
   ii. Numofmines: number of mines in the game.
   iii. Flgn: number of flagged cells, it can't be more than numofmines.
   iv. Numofmoves: number of moves done by the player
   v. Numofqm: number of question marked cell in the game
   vi. DM: determines if the developer mode is on '1' of off '0'.

vii. Arrowmode: determines if the arrow mode is on ‘1’ of off ’0’.

viii. Gameover: determines the state of the game:
0----- The game is still in progress
1----- The player lost
2----- The player won

All the previous variables are integer global variables.

ix. Sqr: a character that determines the value of any unopened cell. In the ordinary state it equals to “X”.

x. Mn: a character that determines the value of the mined cell. In the ordinary state it equals to “*”.

xi. Flg: a character that determines the value of any flagged cell. In the ordinary state it equals to “F”.

xii. Uc: a character that determines the value of any question marked cell. In the ordinary state it equals to “?”.

Those 4 global character variables are made for the option of changing game shapes and for saving the game more efficiently.

6. Threads

A thread is used update time every 1 minute. It is like another main function working at the same time the main working.

Important functions

There are lots of functions used in this programs. Many of them are used as extra features, and some of them are the main core of the program that configures the game. In this chapter of the report, only the main functions will be clarified.

1. Startup:

   This function is the main function that takes the input from the user and lead him to the other functions.

   a. It contains the function that prints the main menu
   b. Defines the height, width and number of mines of the game
   c. Contains the option menu
   d. It calls the function that prints the rank table.

2. Setminespos:

   This function sets the position of the mines in the matrix.

   A random function depending on the time is used to set the mines randomly within the main matrix. This function is called in 2 position: when the game stars and if the first cell opened by the user was a mine. If so, the function rearrange mines so that the first cell opened is not a mine.

3. Printscreenval:

   This function is responsible for printing the main matrix and everything around it during the game.

4. Updateinf:

   This function is like the brain of the game. When the user tries to open, flag or mark a cell, this function is called.

   a. It makes sure if the process the user wants to make is available or not. For example: user can not open a flagged cell.
   b. It preforms the process on the cell chosen by the player
   c. After preforming the process, the function checks if the player won, lost or still on the game.
   d. It calls the function that saves the game.

5. Opencell

   This function is called when user wants to open an opened cell. It checks if number of flagged cells around it equals to the cell number. If so, it opens all the surrounding cell except the flagged and the marked ones

6. Spacecase:

   This function is used to open an empty cell by opening all the surrounding cell except flagged and marked ones. If one of the surrounding cells is empty, the function calls itself again and so on.

7. Setallhiddentoval:

   This is the last function which is used when the player loses or wins the game. It opens all cells and print the solved matrix with a specific format depending on the conditions.

8. Savescore:

   If the player win, this function is called. Its job is to save the player score. But at first all the old scores are read and saved in an array of structures. The function read the new player name and searches the array for it. if it is found it saves the bigger score after placing it in the right position in the array. Then the array is printed again in the file.

9. Savet:

   This function is called every time the user want to save a game. Its rule is to search for his name in the “names” file and make sure the name is accepted and not repeated. If so, it calls another function called “saveIt” to save the important data in files.

10. deletIt:

   Actually we stared and finished writing this function but some problems occurred during debugging it, so we got rid of it. This function is made to delete an only one saved game. It was proposed to delete any loaded game before it is loaded. If it was applied, the user could overwrite his own saved games by deleting the old one and saving the new game.

PS* the function is left as a comment if you want to take a look.

Flowcharts

All flowcharts are fond in the flowchart folder flowcharts

Pseudo code

This pseudo code is for the main function only:

```
Initialize global variables
Check files existence
Print start up menu
Read user input
Start up
If load equals to 1
        Load height and width
        Load screen array
        For(i=0,steps 1 to height)
        {
        For j=0 steps 1 to width
        {
        Set screen[0][0].value to zero
        }
        }
        }
Else
        Set screen.value values
        Set mines in screen.hidden array
         Set numbers in screen.hidden array
If load =1
        Set old time to current time – time loaded
        Initialize first time to zero
        Initialize load to zero
Else
        Set old time to current time
        Initialize first time to one

While the game is on
{
Print current time – old time
Print screen.value array
Update printed information on the screen
Read input
If menu equals to 12
        Set menu to zero
        Set flag number to zero
        Go to the start-up menu
}
If the player lost

        Set entered to zero
        Modify screen.hidden array
        Print screen.hidden array

If the player won
        Print screen.hidden array
        Calculate score
        Print you won
        Print score
        Save score
        Print the rank table
Print click 1 to play again
Print click 2 to go to main menu
Print click 0 to exit
```

```
Input decision

If decision equals to one
    Set reply flag to 1
    Go to start-up menu
If decision equals to 2
    Set reply flag to 2
    Go to start-up menu
If reply flag equals to 0
    End the program
```

User manual

User manual for this game have 2 conditions: arrow mode is activated or not

User manual for ordinary mode:

Enter “number of row” followed by “space” then “number of column” followed by “space” then the number of the operation you want to do on the cell.
* 0--- for flag ->F
* 1--- for click
* 2--- to remove a mark
* 3--- for uncertain ->?
* 4--- for open cell

For example (1 2 0) will flag the square at row 1 column 2

If you want to save the game enter “-1” and if you want to quit without saving enter “0”.

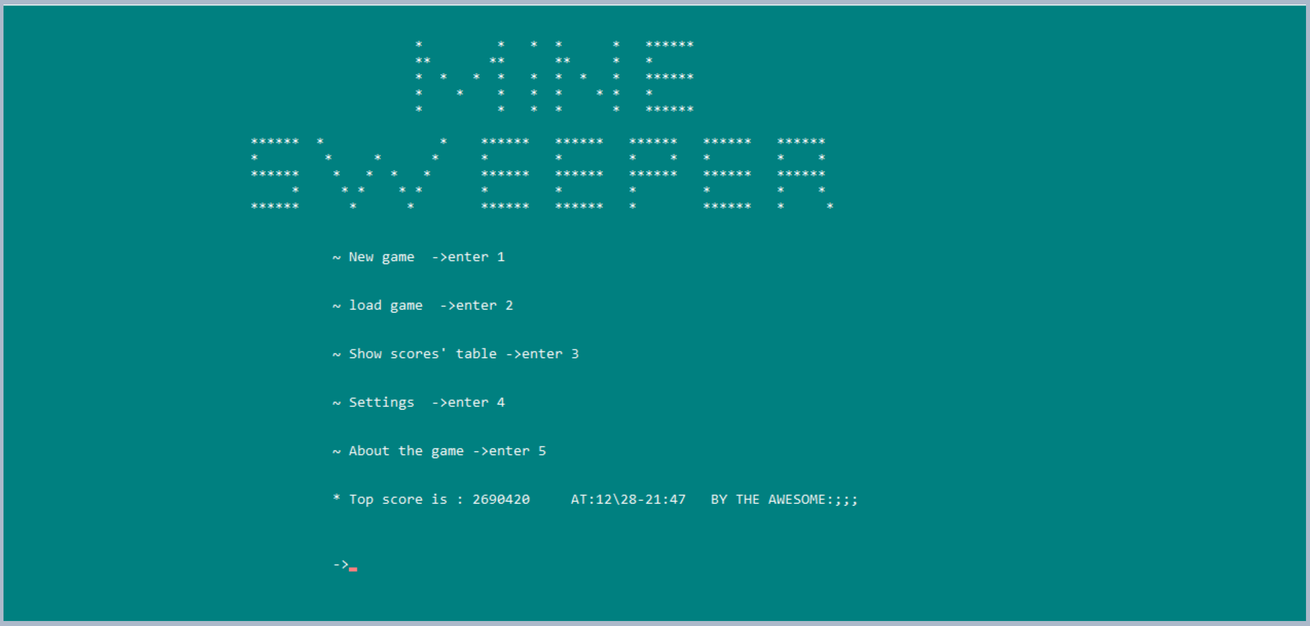
User manual for arrow mode:

To move the arrow
“**W**”----->up
“**S**”------>down
“**D**”----->right
“**A**”----->left
To do a process on the pointed cell press:
“**O**”----->open cell
“**F**”------>flag
“**P**”----->uncertain sign
“**U**”-----> remove mark

“**R**”----->solve near cells
“**Q**”----->Exit
“**V**”----->save and exit

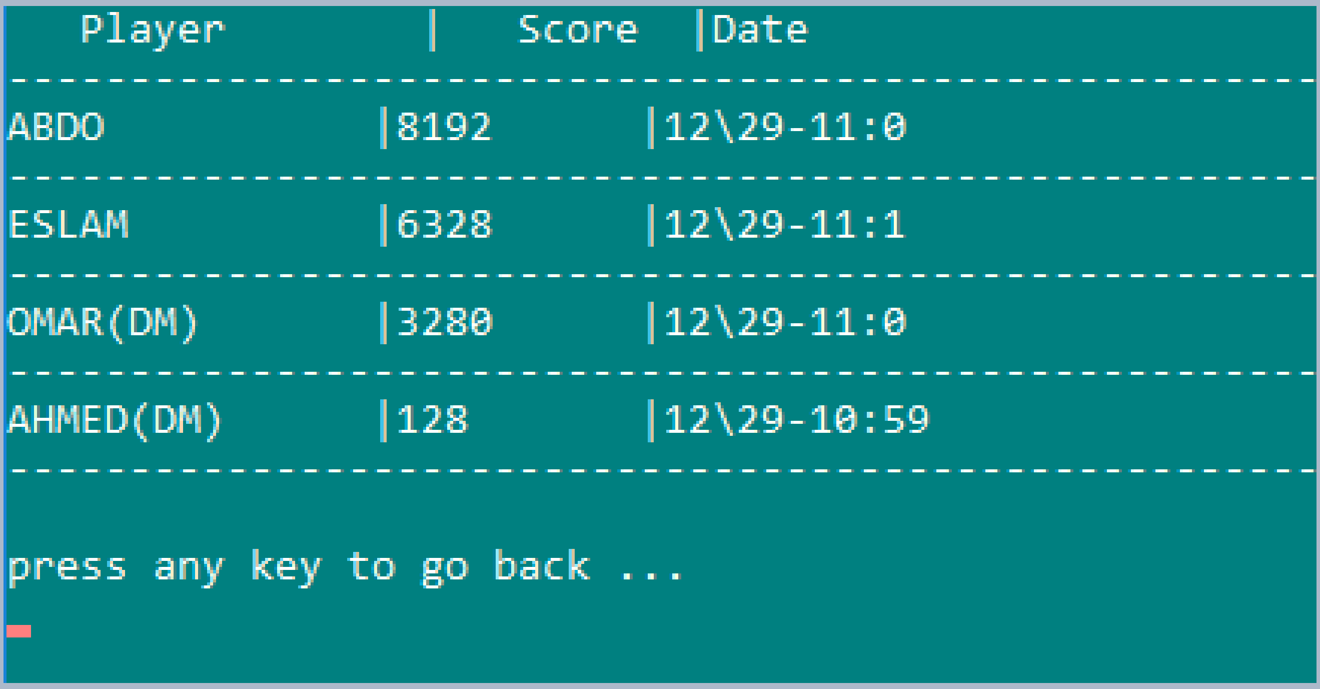
Sample run

Startup menu:



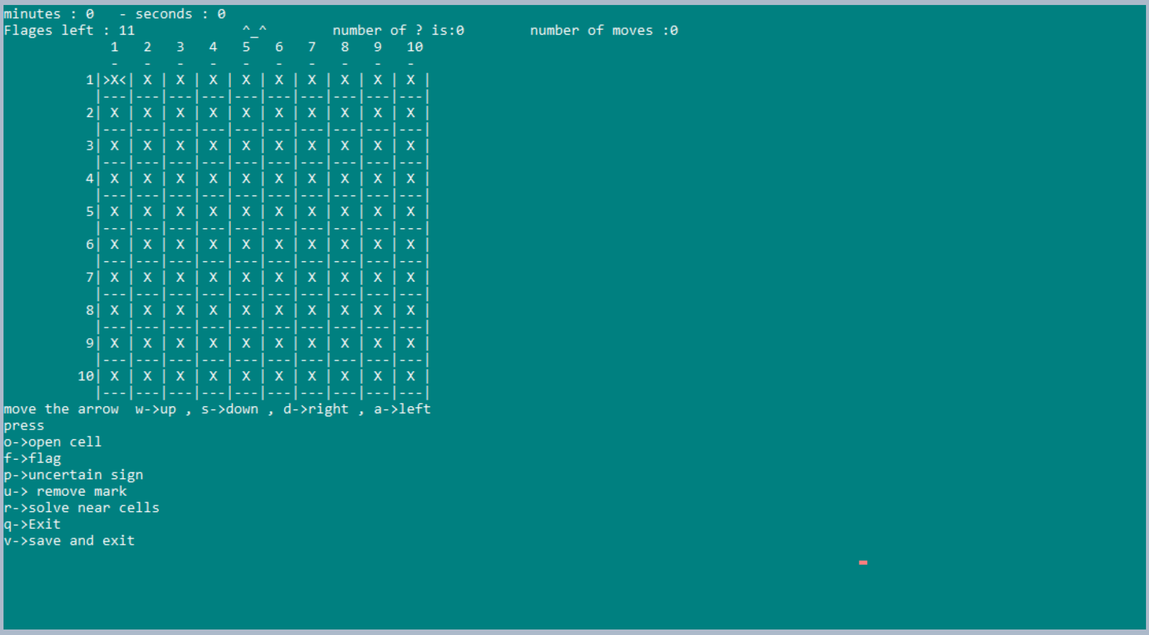
```
                    *         *   *  *      *   ******
                    **       **      **     *   *
                    *  *   *  *   *  *  *   *   ******
                    *    *    *   *  *    * *   *
                    *         *   *  *      *   ******

******  *              *    ******   ******   ******   ******   ******
*        *     *      *     *        *        *    *   *        *    *
******    *   *  *   *      ******   ******   ******   ******   ******
     *     * *    * *       *        *        *        *        *    *
******      *      *        ******   ******   *        ******   *     *

          ~ New game  ->enter 1

          ~ load game  ->enter 2

          ~ Show scores' table ->enter 3

          ~ Settings  ->enter 4

          ~ About the game ->enter 5

          * Top score is : 2690420     AT:12\28-21:47   BY THE AWESOME:;;;


          ->
```

Leader board:



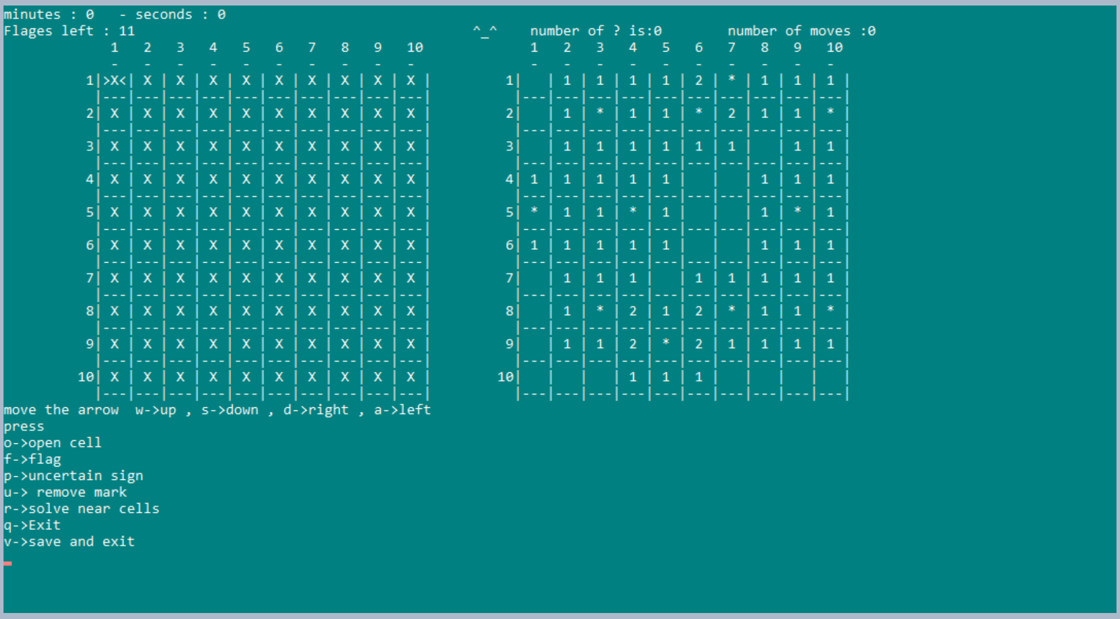
```
   Player                |    Score   |Date
------------------------------------------------------
ABDO                |8192       |12\29-11:0
------------------------------------------------------
ESLAM               |6328       |12\29-11:1
------------------------------------------------------
OMAR(DM)            |3280       |12\29-11:0
------------------------------------------------------
AHMED(DM)           |128        |12\29-10:59
------------------------------------------------------

press any key to go back ...
```

Starting a game:



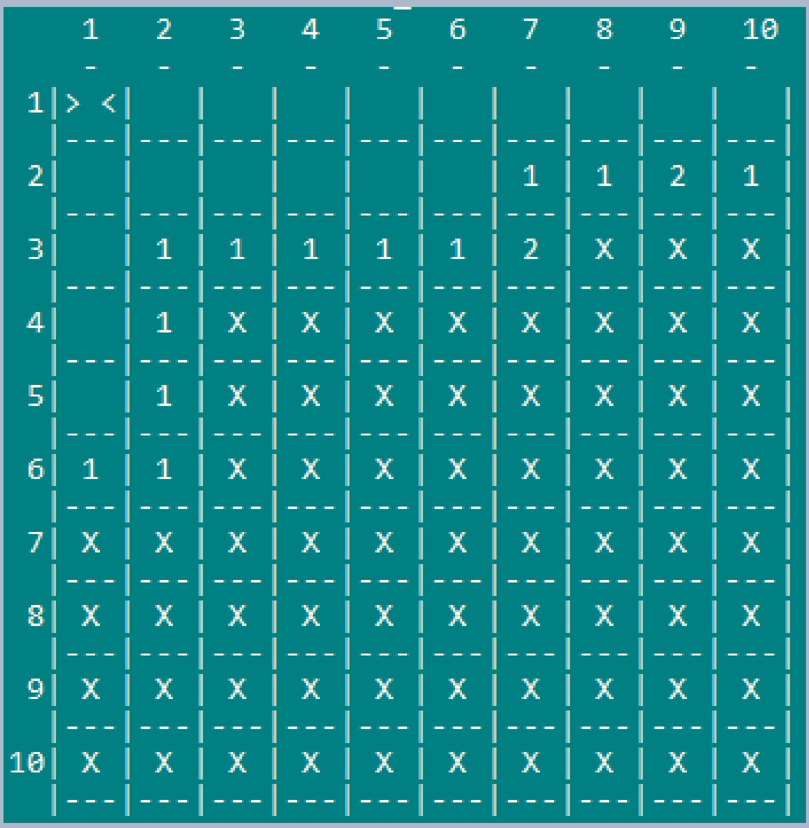
```
minutes : 0   - seconds : 0
Flages left : 11            ^_^        number of ? is:0        number of moves :0
             1   2   3   4   5   6   7   8   9   10
             -   -   -   -   -   -   -   -   -   -
          1|>X<| X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          2| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          3| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          4| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          5| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          6| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          7| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          8| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
          9| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
         10| X | X | X | X | X | X | X | X | X | X |
           |---|---|---|---|---|---|---|---|---|---|
move the arrow  w->up , s->down , d->right , a->left
press
o->open cell
f->flag
p->uncertain sign
u-> remove mark
r->solve near cells
q->Exit
v->save and exit
```

Starting a game with developer mode



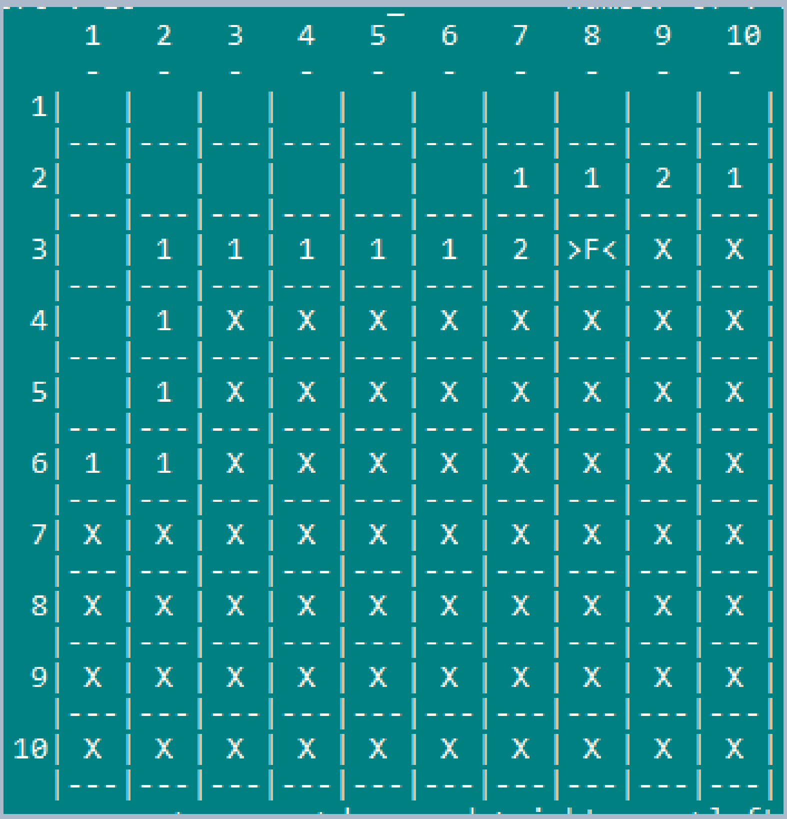
```
minutes : 0   - seconds : 0
Flages left : 11                                         ^_^    number of ? is:0        number of moves :0
             1   2   3   4   5   6   7   8   9   10                1   2   3   4   5   6   7   8   9   10
             -   -   -   -   -   -   -   -   -   -                 -   -   -   -   -   -   -   -   -   -
          1|>X<| X | X | X | X | X | X | X | X | X |           1|   | 1 | 1 | 1 | 1 | 2 | * | 1 | 1 | 1 |
           |---|---|---|---|---|---|---|---|---|---|            |---|---|---|---|---|---|---|---|---|---|
          2| X | X | X | X | X | X | X | X | X | X |           2|   | 1 | * | 1 | 1 | * | 2 | 1 | 1 | * |
           |---|---|---|---|---|---|---|---|---|---|            |---|---|---|---|---|---|---|---|---|---|
          3| X | X | X | X | X | X | X | X | X | X |           3|   | 1 | 1 | 1 | 1 | 1 | 1 |   | 1 | 1 |
           |---|---|---|---|---|---|---|---|---|---|            |---|---|---|---|---|---|---|---|---|---|
          4| X | X | X | X | X | X | X | X | X | X |           4| 1 | 1 | 1 | 1 | 1 |   |   | 1 | 1 | 1 |
           |---|---|---|---|---|---|---|---|---|---|            |---|---|---|---|---|---|---|---|---|---|
          5| X | X | X | X | X | X | X | X | X | X |           5| * | 1 | 1 | * | 1 |   |   | 1 | * | 1 |
           |---|---|---|---|---|---|---|---|---|---|            |---|---|---|---|---|---|---|---|---|---|
          6| X | X | X | X | X | X | X | X | X | X |           6| 1 | 1 | 1 | 1 | 1 |   |   | 1 | 1 | 1 |
           |---|---|---|---|---|---|---|---|---|---|            |---|---|---|---|---|---|---|---|---|---|
          7| X | X | X | X | X | X | X | X | X | X |           7|   | 1 | 1 | 1 |   | 1 | 1 | 1 | 1 | 1 |
           |---|---|---|---|---|---|---|---|---|---|            |---|---|---|---|---|---|---|---|---|---|
          8| X | X | X | X | X | X | X | X | X | X |           8|   | 1 | * | 2 | 1 | 2 | * | 1 | 1 | * |
           |---|---|---|---|---|---|---|---|---|---|            |---|---|---|---|---|---|---|---|---|---|
          9| X | X | X | X | X | X | X | X | X | X |           9|   | 1 | 1 | 2 | * | 2 | 1 | 1 | 1 | 1 |
           |---|---|---|---|---|---|---|---|---|---|            |---|---|---|---|---|---|---|---|---|---|
         10| X | X | X | X | X | X | X | X | X | X |          10|   |   |   | 1 | 1 | 1 |   |   |   |   |
           |---|---|---|---|---|---|---|---|---|---|            |---|---|---|---|---|---|---|---|---|---|
move the arrow  w->up , s->down , d->right , a->left
press
o->open cell
f->flag
p->uncertain sign
u-> remove mark
r->solve near cells
q->Exit
v->save and exit
```

Opening a cell:



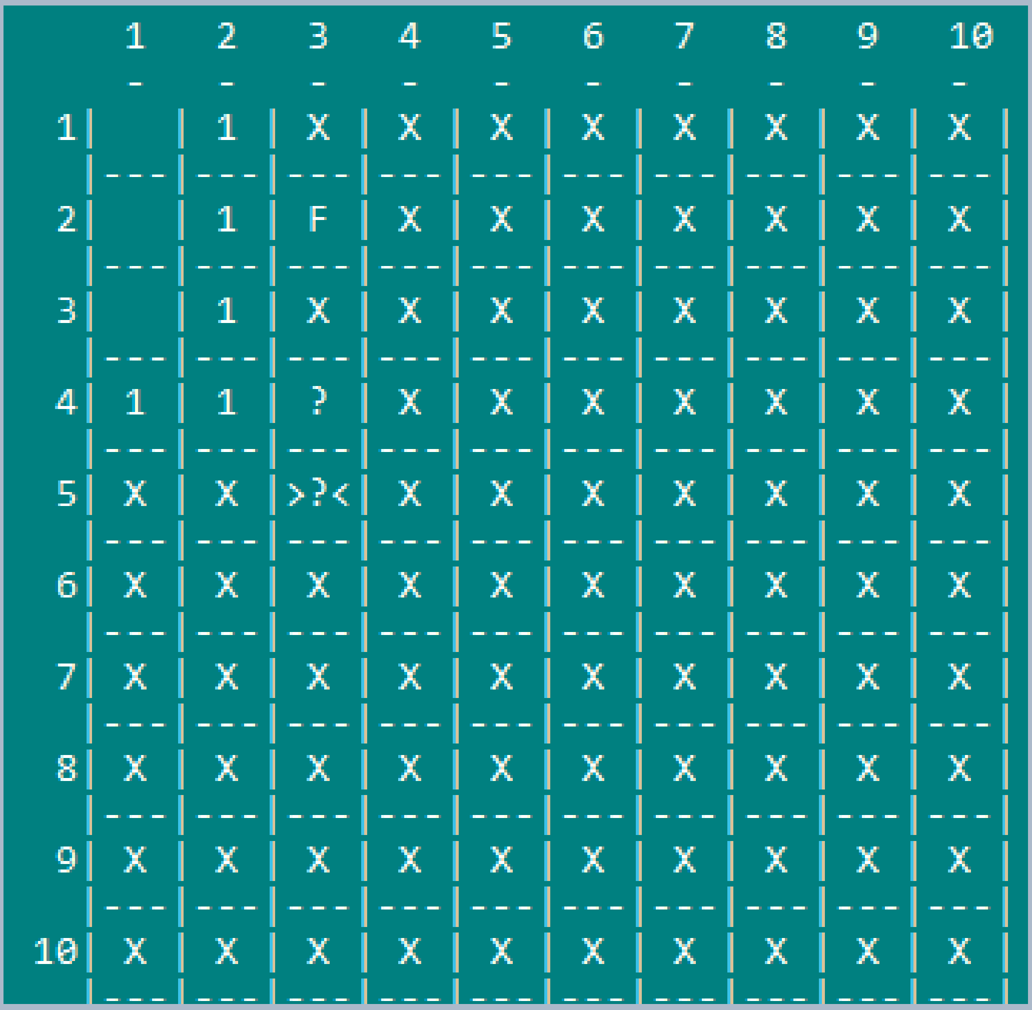
```
     1   2   3   4   5   6   7   8   9   10
     -   -   -   -   -   -   -   -   -   -
  1|> <|   |   |   |   |   |   |   |   |   |
   |---|---|---|---|---|---|---|---|---|---|
  2|   |   |   |   |   |   | 1 | 1 | 2 | 1 |
   |---|---|---|---|---|---|---|---|---|---|
  3|   | 1 | 1 | 1 | 1 | 1 | 2 | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  4|   | 1 | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  5|   | 1 | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  6| 1 | 1 | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  7| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  8| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  9| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
 10| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
```

Flag a cell



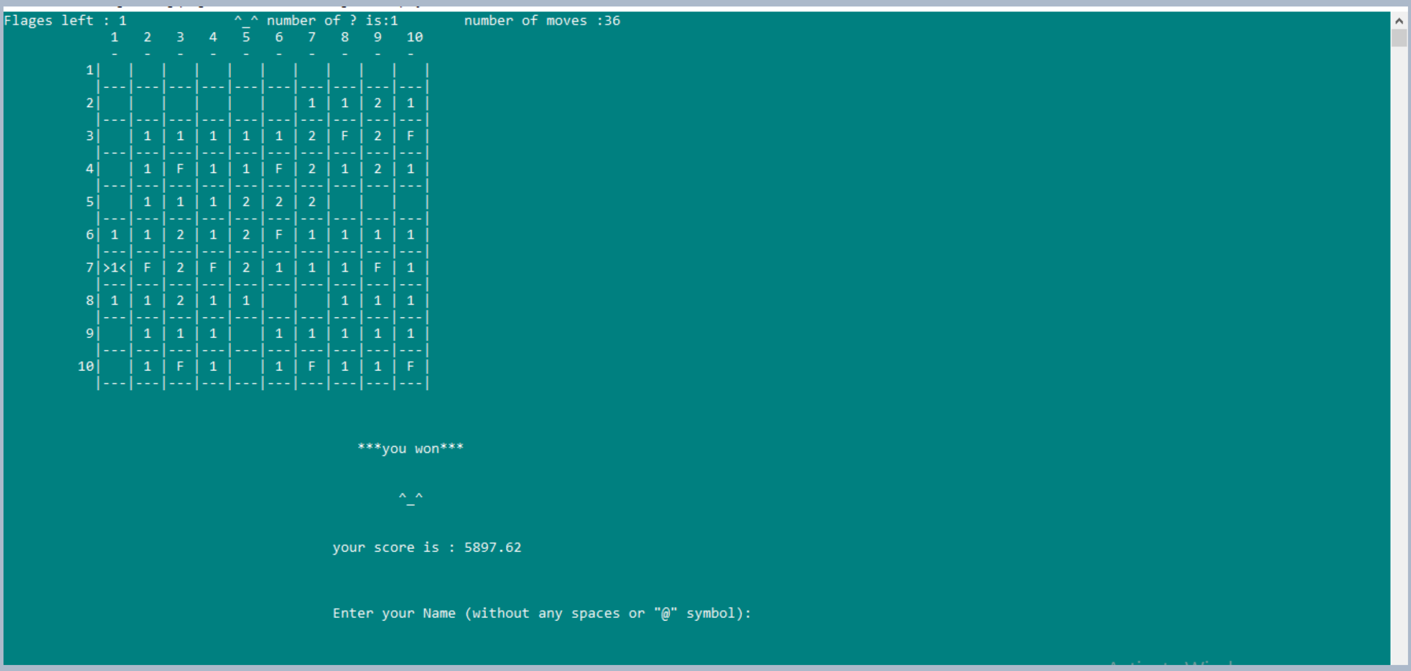
```
     1   2   3   4   5   6   7   8   9   10
     -   -   -   -   -   -   -   -   -   -
  1|   |   |   |   |   |   |   |   |   |   |
   |---|---|---|---|---|---|---|---|---|---|
  2|   |   |   |   |   |   | 1 | 1 | 2 | 1 |
   |---|---|---|---|---|---|---|---|---|---|
  3|   | 1 | 1 | 1 | 1 | 1 | 2 |>F<| X | X |
   |---|---|---|---|---|---|---|---|---|---|
  4|   | 1 | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  5|   | 1 | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  6| 1 | 1 | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  7| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  8| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  9| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
 10| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
```

Flagging and question marking a cell:



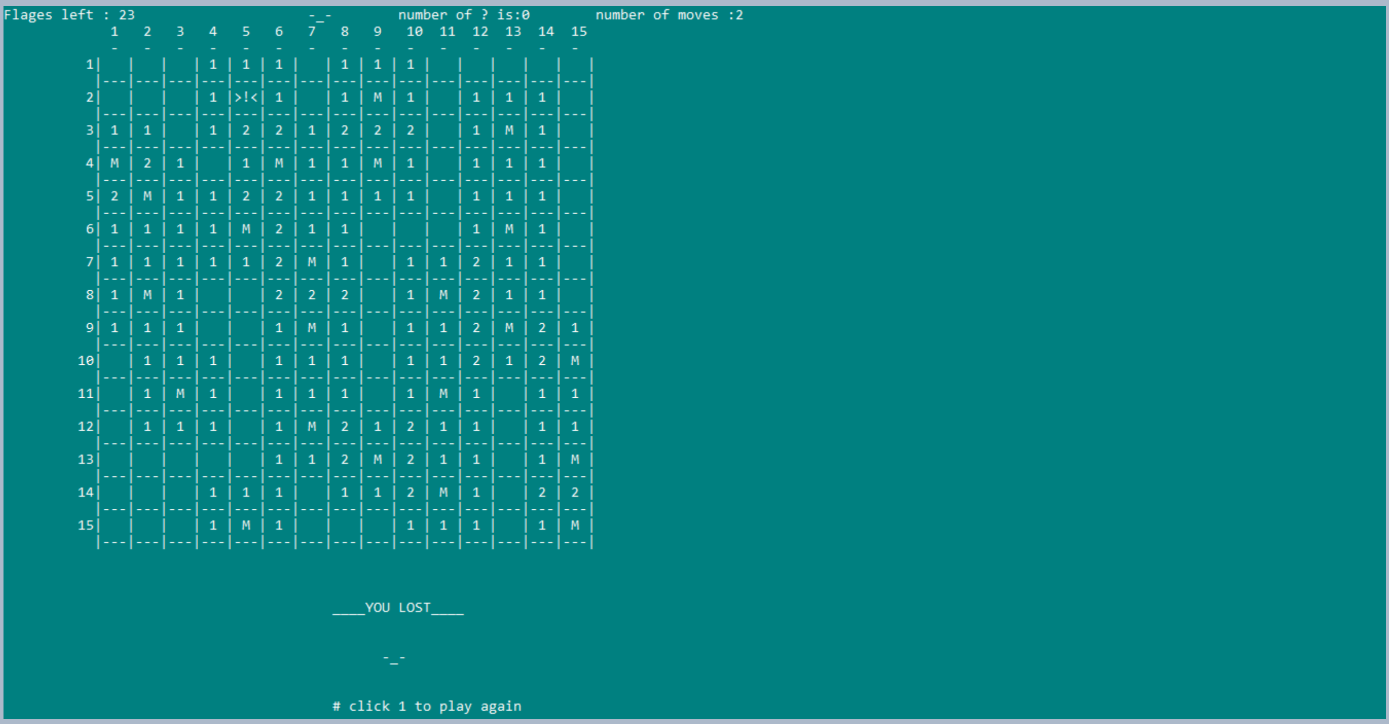
```
     1   2   3   4   5   6   7   8   9   10
     -   -   -   -   -   -   -   -   -   -
  1|   | 1 | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  2|   | 1 | F | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  3|   | 1 | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  4| 1 | 1 | ? | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  5| X | X |>?<| X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  6| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  7| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  8| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
  9| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
 10| X | X | X | X | X | X | X | X | X | X |
   |---|---|---|---|---|---|---|---|---|---|
```

Winning the game:



```
Flages left : 1              ^_^ number of ? is:1        number of moves :36
             1   2   3   4   5   6   7   8   9   10
             -   -   -   -   -   -   -   -   -   -
         1|   |   |   |   |   |   |   |   |   |   |
          |---|---|---|---|---|---|---|---|---|---|
         2|   |   |   |   |   |   | 1 | 1 | 2 | 1 |
          |---|---|---|---|---|---|---|---|---|---|
         3|   | 1 | 1 | 1 | 1 | 1 | 2 | F | 2 | F |
          |---|---|---|---|---|---|---|---|---|---|
         4|   | 1 | F | 1 | 1 | F | 2 | 1 | 2 | 1 |
          |---|---|---|---|---|---|---|---|---|---|
         5|   | 1 | 1 | 1 | 2 | 2 | 2 |   |   |   |
          |---|---|---|---|---|---|---|---|---|---|
         6| 1 | 1 | 2 | 1 | 2 | F | 1 | 1 | 1 | 1 |
          |---|---|---|---|---|---|---|---|---|---|
         7|>1<| F | 2 | F | 2 | 1 | 1 | 1 | F | 1 |
          |---|---|---|---|---|---|---|---|---|---|
         8| 1 | 1 | 2 | 1 | 1 |   |   | 1 | 1 | 1 |
          |---|---|---|---|---|---|---|---|---|---|
         9|   | 1 | 1 | 1 |   | 1 | 1 | 1 | 1 | 1 |
          |---|---|---|---|---|---|---|---|---|---|
        10|   | 1 | F | 1 |   | 1 | F | 1 | 1 | F |
          |---|---|---|---|---|---|---|---|---|---|



                                          ***you won***


                                               ^_^


                                       your score is : 5897.62



                                       Enter your Name (without any spaces or "@" symbol):
```

Losing a game:



```
Flages left : 23                     -_-        number of ? is:0        number of moves :2
             1   2   3   4   5   6   7   8   9   10  11  12  13  14  15
             -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
          1|   |   |   | 1 | 1 | 1 |   | 1 | 1 | 1 |   |   |   |   |   |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
          2|   |   |   | 1 |>!<| 1 |   | 1 | M | 1 |   | 1 | 1 | 1 |   |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
          3| 1 | 1 |   | 1 | 2 | 2 | 1 | 2 | 2 | 2 |   | 1 | M | 1 |   |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
          4| M | 2 | 1 |   | 1 | M | 1 | 1 | M | 1 |   | 1 | 1 | 1 |   |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
          5| 2 | M | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 |   | 1 | 1 | 1 |   |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
          6| 1 | 1 | 1 | 1 | M | 2 | 1 | 1 |   |   |   | 1 | M | 1 |   |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
          7| 1 | 1 | 1 | 1 | 1 | 2 | M | 1 |   | 1 | 1 | 2 | 1 | 1 |   |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
          8| 1 | M | 1 |   |   | 2 | 2 | 2 |   | 1 | M | 2 | 1 | 1 |   |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
          9| 1 | 1 | 1 |   |   | 1 | M | 1 |   | 1 | 1 | 2 | M | 2 | 1 |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
         10|   | 1 | 1 | 1 |   | 1 | 1 | 1 |   | 1 | 1 | 2 | 1 | 2 | M |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
         11|   | 1 | M | 1 |   | 1 | 1 | 1 |   | 1 | M | 1 |   | 1 | 1 |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
         12|   | 1 | 1 | 1 |   | 1 | M | 2 | 1 | 2 | 1 | 1 |   | 1 | 1 |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
         13|   |   |   |   |   | 1 | 1 | 2 | M | 2 | 1 | 1 |   | 1 | M |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
         14|   |   |   | 1 | 1 | 1 |   | 1 | 1 | 2 | M | 1 |   | 2 | 2 |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
         15|   |   |   | 1 | M | 1 |   |   |   | 1 | 1 | 1 |   | 1 | M |
           |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|


                                        ____YOU LOST____


                                              -_-


                                        # click 1 to play again
```

References:

https://www.youtube.com/playlist?list=PL6gx4Cwl9DGAKIXv8Yr6nhGJ9Vlcjyymq
http://stackoverflow.com/
https://www.youtube.com/playlist?list=PLCB9F975ECF01953C
K.N.KING, C programming a modern approach, 2nd edition
And of course the c lectures of this semester.
https://cplusplus.com/