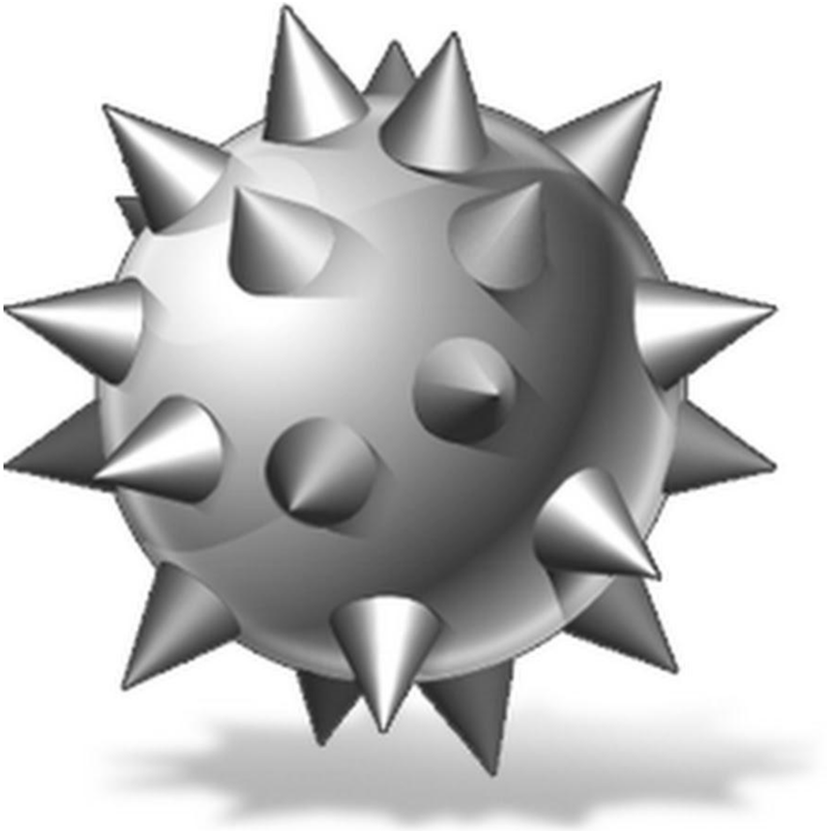


Minesweeper Project

CONSOLE-BASED GAME

Ahmed Hussein & Bassam Aiman | Programming-1 | CSED 20

Game Overview

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.

The game originates from the 1960s, and has been written for many computing platforms in use today. It has many variations and offshoots. This Computer game utilizes strategy and chance.

Design Overview

This project was totally designed, implemented, debugged and tested by Ahmed Hussein and Bassam Aiman.

The main purpose of the design was to implement every algorithm or method taught by the course staff throughout the year.

The game is divided into various sections, having a strong and vivid structure makes the code maintainable, easy to debug and understand and giving it the best performance according to the capabilities of the Programming-1 course student knowledge.

The project is divided into the main.c source file besides 6 header files (with their sources).

Each source file is divided into various code sections (functions), where each function has its own purpose and is as independent as possible from the rest of the code making the code divided into smaller semi-independent units guaranteeing the self-integrity for each function and for the whole project.

Also naming conventions and commenting was used to ensure the best readability experience.

Next to come is a list of every header used in the project and coded by us.

Main Source File

Here’s a list of the functions included in in main.c source file:

- main

  1. Parameters: NULL
  2. Return Type: int
  3. Description: The main function that runs the program, ensures the best flow and harmony between each function in the program.

- new_game

  1. Parameters: NULL
  2. Return Type: void
  3. Description: Executes when the user chooses a new game from the home page, it asks the user to enter the size of the grid and then prints it. Also it asks the user to choose which cell to open first.

- regen_user_grid

  1. Parameters: NULL
  2. Return Type: void
  3. Description: it reprints the game grid to the user and asks the user to input the next cell coordinates and the action to be applied on it. Also it prints the total time, moves, flags and question marks.

- open_first_cell

  1. Parameters: NULL
  2. Return Type: void
  3. Description: This cell is executed when the user enters the first cell coordinates. It initializes the starting time, place mines randomly, determines the number of mines beside each cell and apply the depth-first search algorithm to open the neighboring empty cells and stops and the cells with mines nearby.

- cell_action

  1. Parameters: NULL
  2. Return Type: void
  3. Description: This is where the program decides what to do with the inputted action by the user, whether to open, flag, question, save or exit.

- scan_home_input

1. Parameters: NULL
2. Return Type: void
3. Description: This function executes in the homepage, it asks the user what action he wants, whether a new game, load game, scoreboard, sounds, help, exit.

- Lose

1. Parameters: NULL
2. Return Type: void
3. Description: Executes when the user lose the game, it produces sounds when executed and regenerate the grid showing the user the mines, the mine that made him lose the game and any incorrect flags the placed.

- Win

1. Parameters: NULL
2. Return Type: void
3. Description: Executes when the user wins the game, it generates sounds, also regenerate the grid showing the mines with the letter “F”. After that, it redirects the user to enter his name to store his score in the scoreboard, then it redirects him to the scoreboard.

- Play

1. Parameters: NULL
2. Return Type: void
3. Description: Just a controlling function to ensure the smoothness of the program flow, redirects the compiler to each function during each step of the palying.

Headers

Here's a list of the headers implemented and their included functions:

➢ CellFunctions.h

❖ calculate_mines

1. Parameters: int x_coor, int y_coor
2. Return Type: int
3. Description: Returns the number of mines surrounding each cell in the grid.

❖ DFS_recursion

1. Parameters: int x_coor, int y_coor
2. Return Type: void
3. Description: A recursive algorithm to open every unopened cell in the grid and stops at cells having mines surrounding them.

➢ ExceptionHandlers.h

❖ exceptionHandledIntInput

1. Parameters: int lowerLimit, int upperLimit
2. Return Type: int
3. Description: Takes the input from the user rejecting any invalid input and any input outside the boundaries. Returns that integer after a series of filtering algorithms to ensure only integer is inputted then it returns that integer.

❖ exceptionHandledCharInput

1. Parameters: NULL
2. Return Type: char
3. Description: Takes the input from the user rejecting any invalid inputs and returns the entered char.

➢ FilesFuntions.h

❖ save_game

1. Parameters: NULL
2. Return Type: void
3. Description: Saves the current game and saves every parameter needed to resume the game afterwards.

❖ load_game

1. Parameters: NULL
2. Return Type: void
3. Description: Loads the game after verifying the name of the game, afterwards the game is resumed with the elapsed time throughout the game, moves, etc.…

❖ load_scoreboard

1. Parameters: NULL
2. Return Type: void
3. Description: Loads the score of the game winners displaying their names and scores.

❖ register_score

1. Parameters: NULL
2. Return Type: void
3. Description: Registers the score of the winner player, taking his name as an input and saves his score.

➢ IdleFuntionThread.h

❖ one_minute_idle

1. Parameters: void *arg
2. Return Type: void
3. Description: Imports pthread.h library to enable the program to excute multiple threads to detect if the user is idle and prints the total idle time on the console.

➢ MinorDebuggersAndTools.h

❖ print_test

1. Parameters: char array[ARRAY_SIZE][ARRAY_SIZE], int array_x_size, int array_y_size
2. Return Type: void
3. Description: Just a function for testing and debugging, prints the entered array.

❖ Power

1. Parameters: int x, int y
2. Return Type: int
3. Description: Calculates the power of x^y . Returns a power of type int instead of the stdlib.h pow() function that returns double.

❖ Clear

1. Parameters: NULL
2. Return Type: void
3. Description: Clears the console screen.

➢ TextOnlyFuntions.h

❖ print_logo

1. Parameters: NULL
2. Return Type: void
3. Description: Prints the header of the homepage.

❖ init_home

1. Parameters: NULL
2. Return Type: void
3. Description: Prints the options of the homepage.

❖ Help

1. Parameters: NULL
2. Return Type: void

3. Description: Prints the help screen.

Also a list of imported headers from mingw library:

- stdio.h
- stdlib.h
- conio.h
- Windows.h
- pthread.h

Data Structures

To ensure the best performance and to reduce coding complexity, one the basic and simple data types were used.

For determining each element in the game grid only a two-dimensional char array was used to define each element and further simplify the coding.

Also various type structures was used to simply implement time representing from Windows.h header and a structure for files (FILE) to make the program able to read and write from external sources.

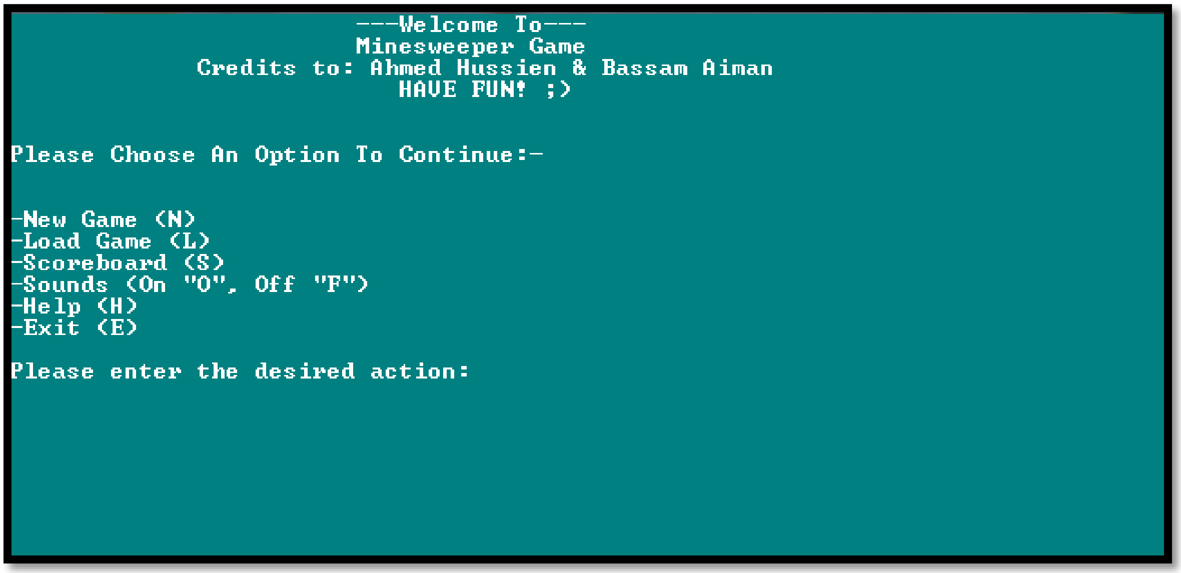
Another structure type was used for pthread.h header which is pthread_t, it's the header identified data type to represent a pointer to a specific thread running in the program.

Also various types were used to add colors to the text, text background and console background.

Other than this, basic data types were used, mainly of int and char data types which ensures the least memory size consuming.

Game Interface

➢ Home Page



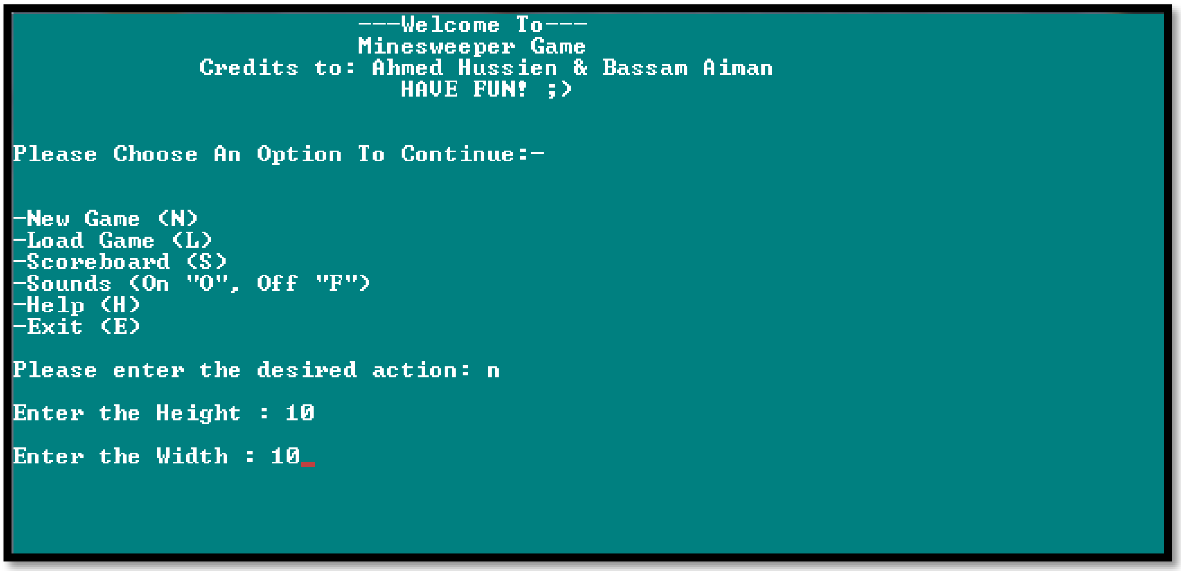
```
                        ---Welcome To---
                        Minesweeper Game
             Credits to: Ahmed Hussien & Bassam Aiman
                           HAVE FUN! ;)

Please Choose An Option To Continue:-

-New Game (N)
-Load Game (L)
-Scoreboard (S)
-Sounds (On "O", Off "F")
-Help (H)
-Exit (E)

Please enter the desired action:
```

Homepage of the game is made of six different options to choose from. New game, Load game, Scoreboard, Sounds (On/Off), Help and Exit.

You can simply choose any desired option by pressing the first initial of that option (N, L, S, [O/F], H, E).

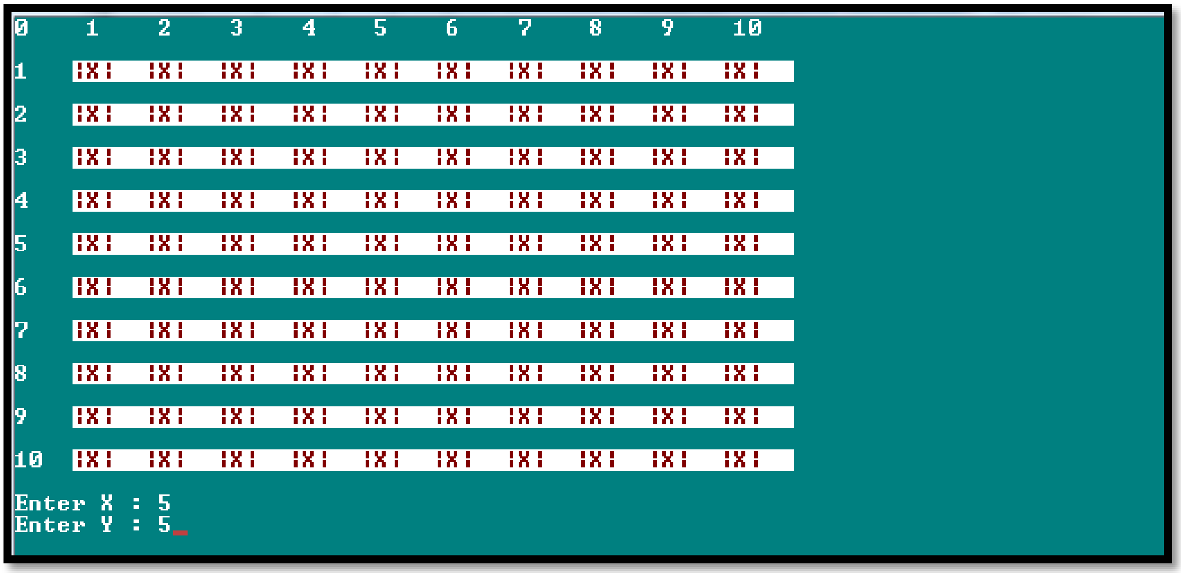
Each option will direct you to another page to complete that action and after that it returns you back to the home page.

➢ New Game



```
                         ---Welcome To---
                         Minesweeper Game
              Credits to: Ahmed Hussien & Bassam Aiman
                            HAVE FUN! ;)


Please Choose An Option To Continue:-


-New Game (N)
-Load Game (L)
-Scoreboard (S)
-Sounds (On "O", Off "F")
-Help (H)
-Exit (E)

Please enter the desired action: n

Enter the Height : 10

Enter the Width : 10_
```

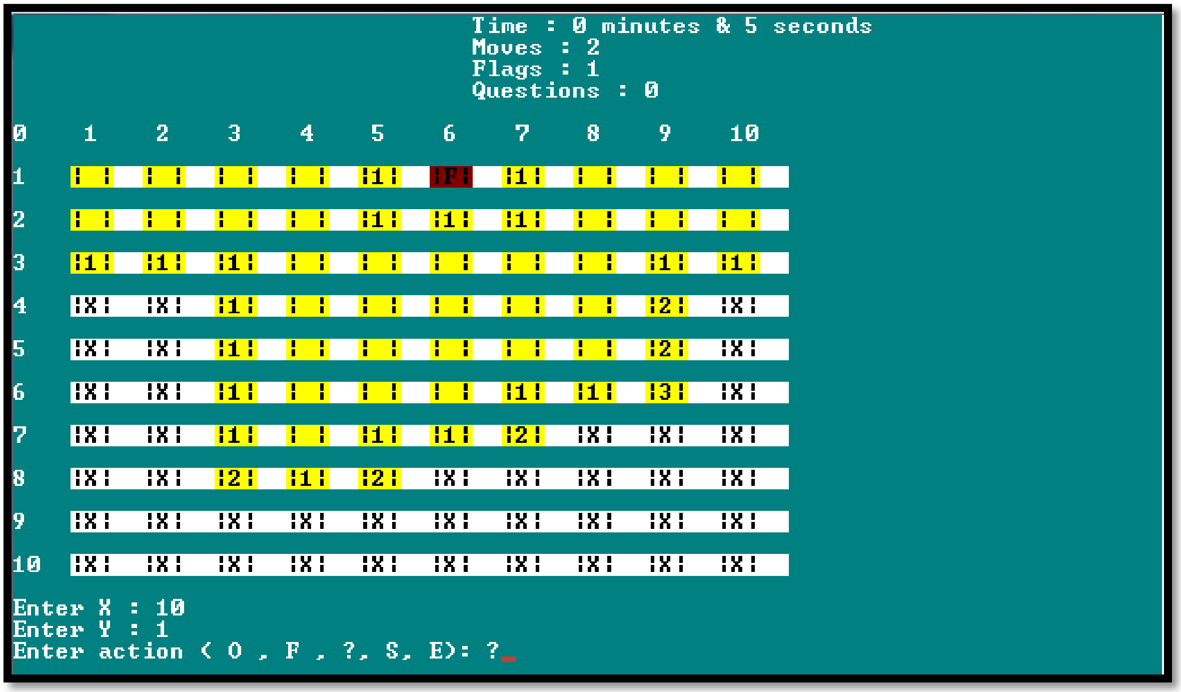
Once you enter (N or n), it will ask you to enter a horizontal and vertical size for the game grid, note that maximum size is 100 for each axis and the minimum is 2 for each axis.



```
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|

Enter X : 5
Enter Y : 5_
```

After that the program will generate your grid with the specified size (including some animation), asking you to enter the coordinates of the first cell you want to open.

Note that the first row and the first column is numbered to make it easy for the user to input the coordinates of the desired cell.



```
                                Time : 0 minutes & 5 seconds
                                Moves : 2
                                Flags : 1
                                Questions : 0

0    1    2    3    4    5    6    7    8    9    10

1   ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦1¦  ¦F¦  ¦1¦  ¦ ¦  ¦ ¦  ¦ ¦

2   ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦1¦  ¦1¦  ¦1¦  ¦ ¦  ¦ ¦  ¦ ¦

3   ¦1¦  ¦1¦  ¦1¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦1¦  ¦1¦

4   ¦X¦  ¦X¦  ¦1¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦2¦  ¦X¦

5   ¦X¦  ¦X¦  ¦1¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦2¦  ¦X¦

6   ¦X¦  ¦X¦  ¦1¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦1¦  ¦1¦  ¦3¦  ¦X¦

7   ¦X¦  ¦X¦  ¦1¦  ¦ ¦  ¦1¦  ¦1¦  ¦2¦  ¦X¦  ¦X¦  ¦X¦

8   ¦X¦  ¦X¦  ¦2¦  ¦1¦  ¦2¦  ¦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¦

Enter X : 10
Enter Y : 1
Enter action ( O , F , ?, S, E): ?_
```

After opening the first cell, such same 3 inputs must be made at each time

1. “Enter X:” – To enter the x-coordinate.
2. “Enter Y:” – To enter the y-coordinate.
3. “Enter action” – To enter the desired action.

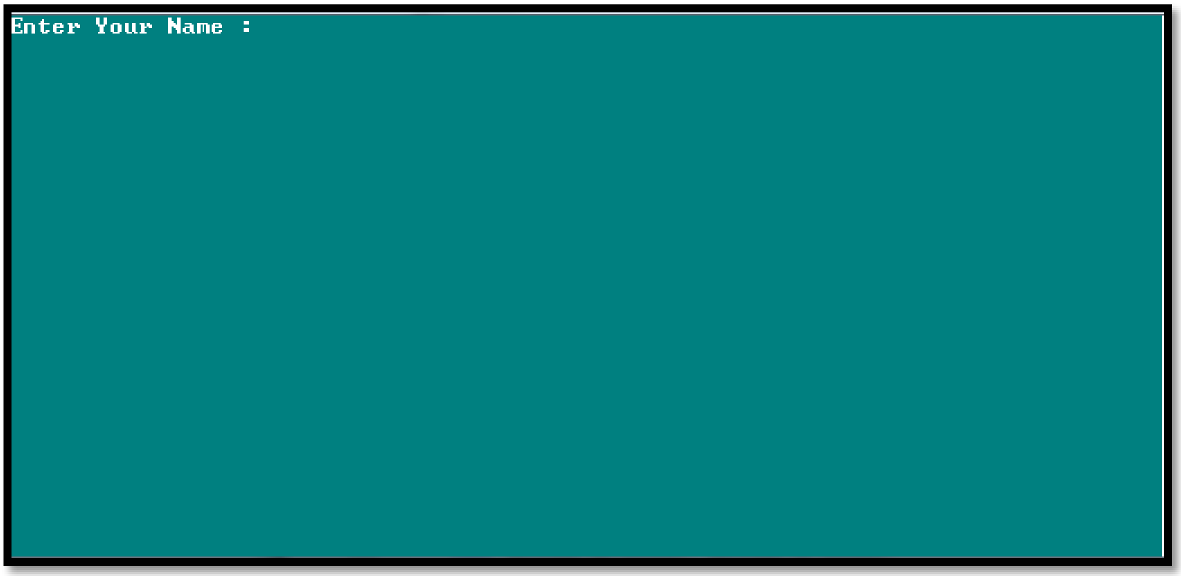
There are 5 different actions that can be done while playing:

1. “O”: To open an unopened cell, or to open an opened cell containing a number equal to the surrounding number of mines.
2. “F”: To put a flag on the specified cell.
3. “?”: To put a question mark over the specified cell, treated as the flag but it’s just for the user.
4. “S”: To save the game to continue later.
5. “E”: To exit the game.

Note that each opened/numbered cell is colored with a different color from the opened ones, also the flagged and questioned cells are colored differently.

Also one of the provided functions is that the game detects when the user is idle for more than 60 seconds, it writes a message each 1 minute displaying the total idle time away from the game.

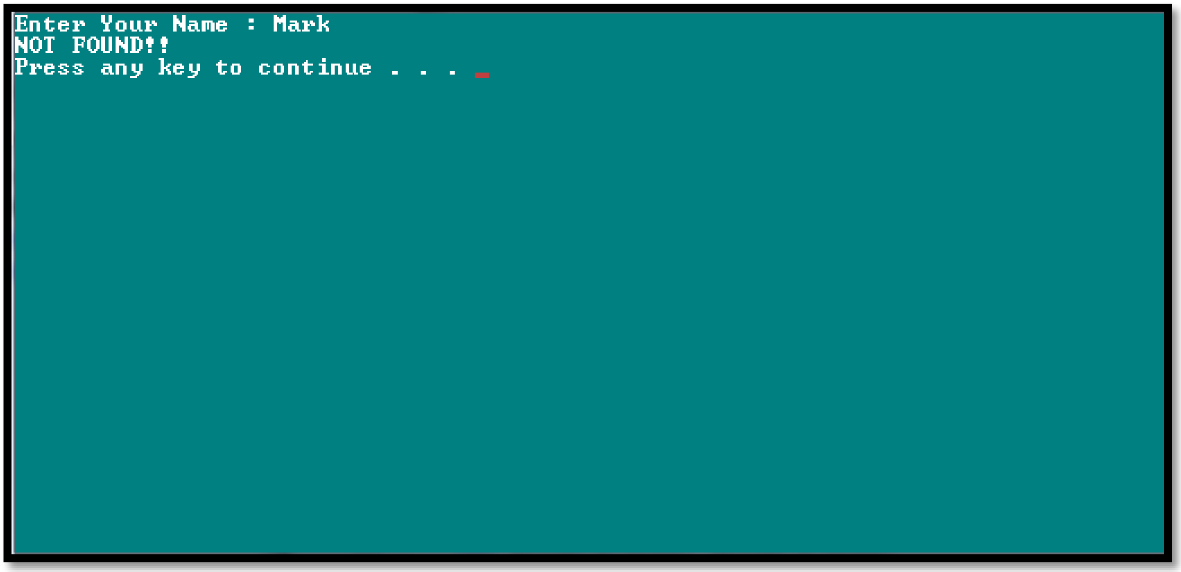
➢ Load Game



```
Enter Your Name :
```

Simple command, just type the name of the game you saved and it will load automatically.

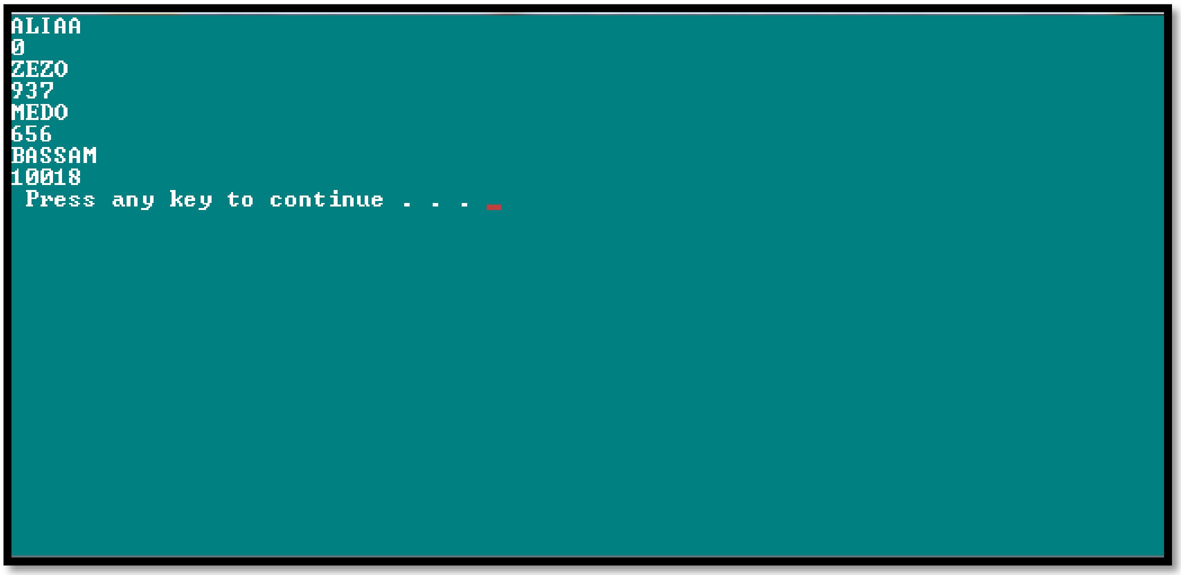
It will load the total time, number of moves, flags, questions …



```
Enter Your Name : Mark
NOT FOUND!!
Press any key to continue . . .
```

When the written name is not found it displays an error message then will redirect you to the home screen.

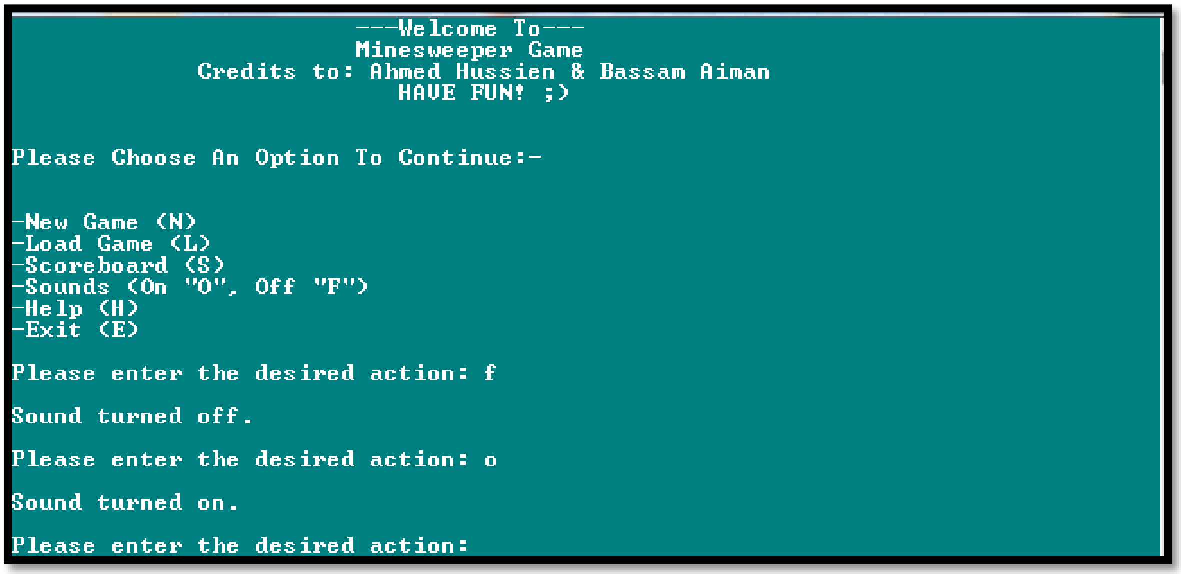
➢ Scoreboard



```
ALIAA
0
ZEZO
937
MEDO
656
BASSAM
10018
 Press any key to continue . . .
```

This screen shows the scoreboard of the players who won the game.

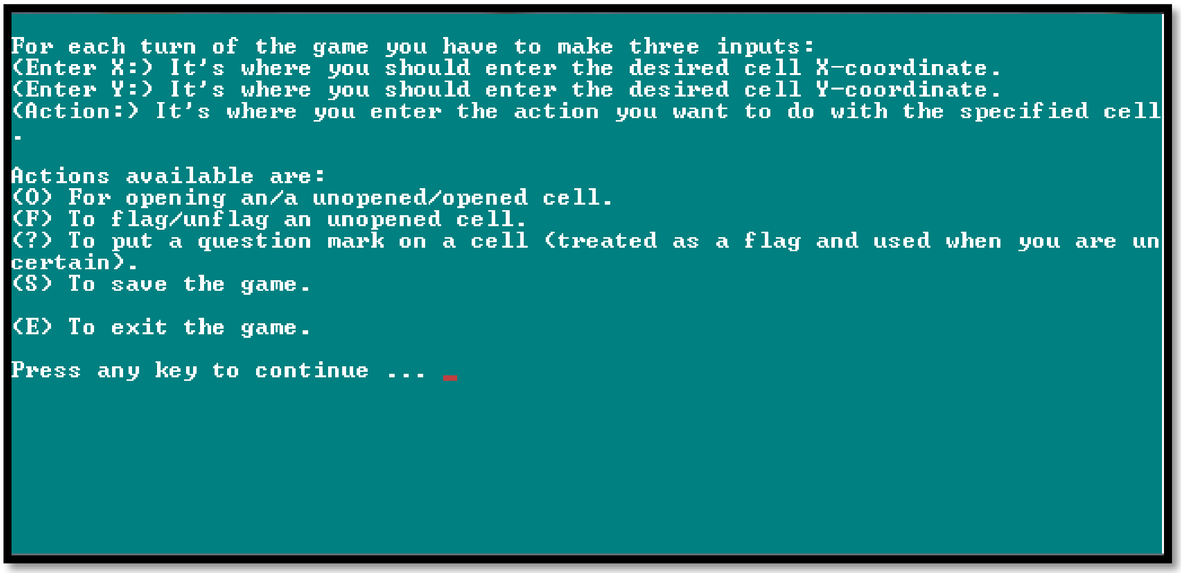
➢ Sounds



```
                        ---Welcome To---
                        Minesweeper Game
             Credits to: Ahmed Hussien & Bassam Aiman
                           HAVE FUN! ;)


Please Choose An Option To Continue:-


-New Game (N)
-Load Game (L)
-Scoreboard (S)
-Sounds (On "O", Off "F")
-Help (H)
-Exit (E)

Please enter the desired action: f

Sound turned off.

Please enter the desired action: o

Sound turned on.

Please enter the desired action:
```

This option when enabled (enabled by default), it makes sounds when a player win or lose a game, this can be turned off by pressing “F” or turned on by pressing “O”.

➢ Help



```
For each turn of the game you have to make three inputs:
(Enter X:) It's where you should enter the desired cell X-coordinate.
(Enter Y:) It's where you should enter the desired cell Y-coordinate.
(Action:) It's where you enter the action you want to do with the specified cell
.

Actions available are:
(O) For opening an/a unopened/opened cell.
(F) To flag/unflag an unopened cell.
(?) To put a question mark on a cell (treated as a flag and used when you are un
certain).
(S) To save the game.

(E) To exit the game.

Press any key to continue ...
```

This screen provides the user with the basic tips and commands that will help him to play the game.

➢ Exit

The user can exit the game from the home screen by pressing "E".