

ALGORITHM

D) Login functionality

In this login functionality person can login in game. He enter his name, age and loc.

If he is already login he only have to enter his name. Through this files are created by person name to save history of player.

2) Level Selection:-

As show in digram minesweeper has blocks. In this level selection there are 3 choices professional, semi pro and beginner. The only

1	2	1	0
1	*	1	0
1	1	1	0
0	0	0	0

in them is number of blocks and number of bombs which is written 95 * in diagram. (Number of blocks in diagram is 4×4).

NEW AND LOAD Game:

In this functionality two choices are given to user new and load game. If user enters load game then by the help of login functionality through history of player he can load game through its name. He can also play a new game.

INITIALIZING And Declaration:

In this functionality we declare two arrays. In these arrays one is initialized ~~is one char only~~ and one with another char.

with character zero and the other with another character.

Bomb Placing:

In this functionality the array which is initialized with zero character another character is randomly put. This char is supposed as bombs.

This char is put on limited number in the zero array. The number of bombs in professional, semi pro and beginner were different.

Assigning numbers:-

Basically as stars shown in the blocks there are some blocks number are

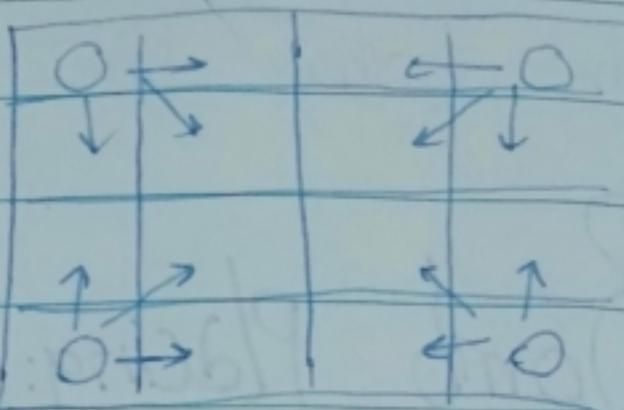
written these shows the number of bombs nearby it. Basically it has

1	1	1	0
1	*	1	0
1	1	1	1
0	0	1	*

3 dimensions. (Corners, along side and in middle)

Corners:-

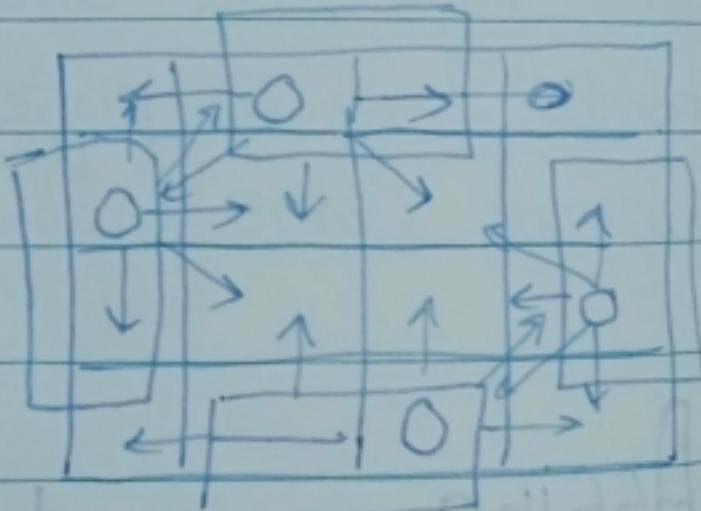
If there is a zero in any corner then these



are three blocks touching it we check these blocks if these blocks have bombs then count the number of bombs and update O of corner with that number.

Along Side:-

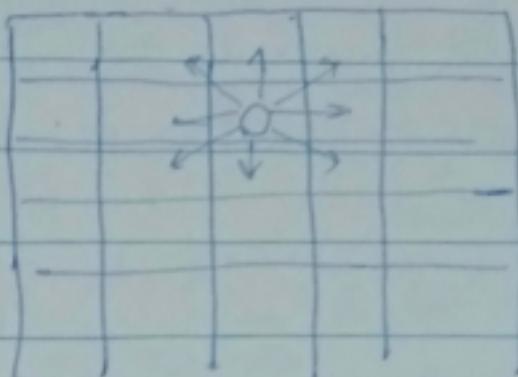
If there is a zero block not in middle and not in a



side then there are five blocks touching them. We check all these blocks and we count if there is or bombs we update O with the number of bomb count.

Middle:-

If there is a 0 not in corners and not along side we check eight blocks touching it. If there are bombs in these numbers of block then count them and update 0 with the count value.



Hiding array of bombs and number:

The array which consists of 0's and numbers and bomb is hided by other array in which we store an another character. The user is seeing the array of only that only one character but behind their is that array which has bombs.

Open box:

In this function the row and column number

array to open is get through user and the (block or) the element of that array is opened means that element of array which has bombs is shown to user.

Score Count:

In this function open of each box is given a score and the total score is counted by taking into consideration how many blocks are open before the game wins or losses.

Score Save:

In this function the score of the person is saved in a Score file so that his performance should be remained saved.

WIN AND Loose functionality:

In this function if the user open a box which has a bomb then user loses the game and game is over. If those block which have bombs are not open and all remaining are opened then the user wins the game.

Sorting in high Score:

The score of the person is sorted in a file of high score. This contains the highest score made by the user.

Saving and Exit functionality:

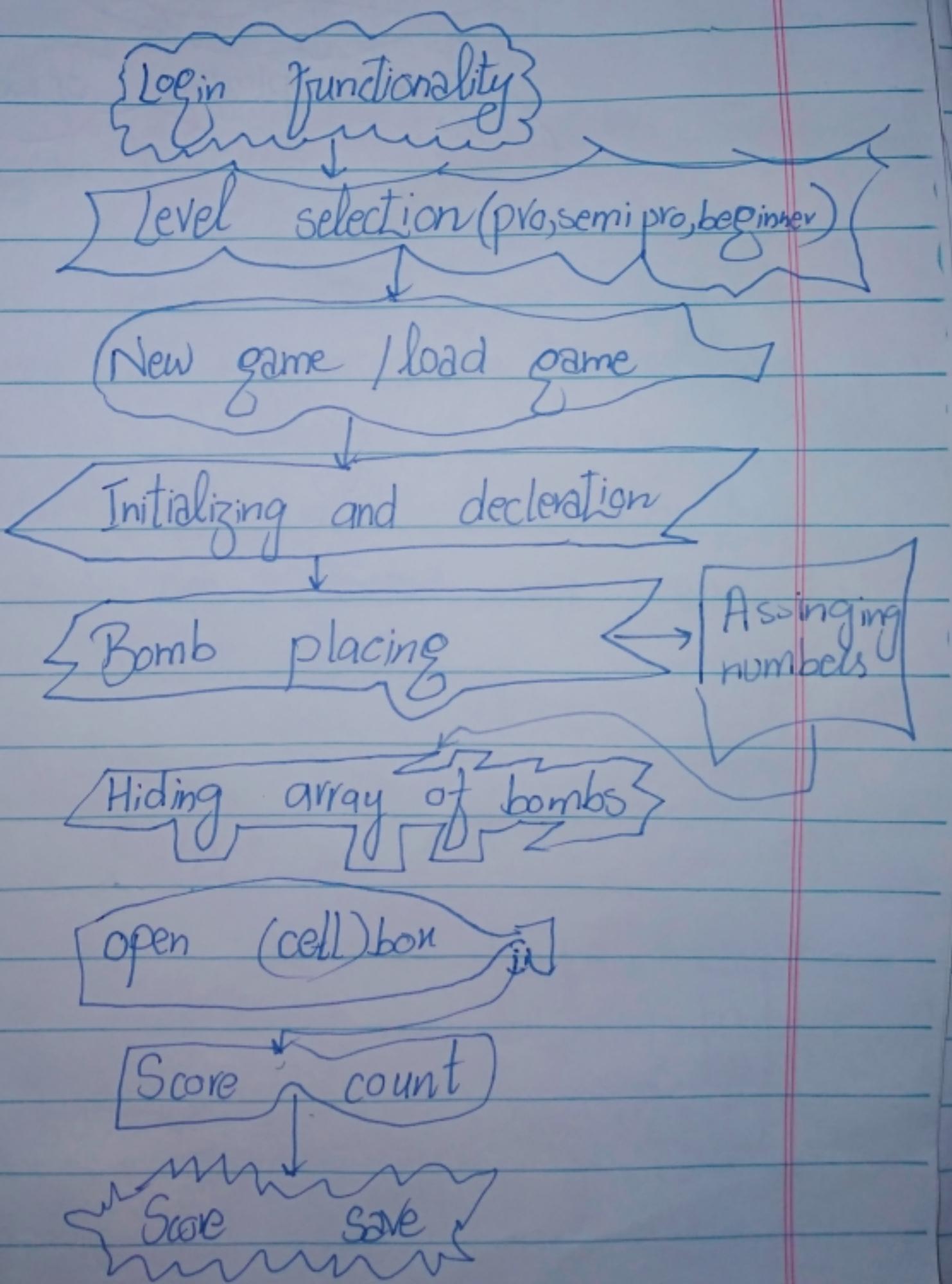
The purpose of

this function is to save the game if the user wants at any point to leave and save his own going name he can give name to that game and store it. In Exit if the user wants to exit he can exit his game at any point.

AGAIN PAYING OPTION:

In this function an option is given if the user wants to again play the game then a new game or loaded game which user wants can play else if not the program stops.

MINESWEEPER



Win and lose functionality Date: _____

Sorting Win high score

Loading (Saving) and
Exit functionality

Again playing option