Group Project

Game Title:

Minesweeper

Course:

Cis17B

Class#

47697

Created by:

James, Rashad, Sage

1 Introduction

The game of minesweeper is simple on the surface; however, there are key rules that are implemented into the game in order to win. The player must clear the 10 by 10 grid without hitting any mines. Meaning the player must make educated guesses as to where the mine is, and what tile they should reveal. The numbers that get revealed on the board as the players progress will indicate the number of mines in the area. The player must use those indications to then attempt to flag tiles that may hold a mine. The clear condition is to open all the tiles with exceptions of the mines on the grid.

The new updated version of minesweeper now allows players to select difficulties, resulting in grid size and bomb count being much large. As well as a custom mode that allows players to make any size grid with any number of bombs. Once a game is completed the player can now add their name, score, and time to the minesweeper database.

2 How to play

For our version of minesweeper, the user clicks on the tile to reveal the space. The user has the option to place flags using a right click on that selected tile. Another condition is that the users first input to the board cannot be a bomb, meaning the first move cannot cause a game over. Any move after that will be subject to the board that was generated.

3 Development Summary

Lines of code: 711 lines (without including html forms)

Comments: 90 lines of comments

Total amount: 735 lines

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Variable Name | Description | Location |
| Int | Boardsize | Size of the board being used. | Board {} |
| Int | numsquares | number of squares within the board | Board {} |
| boolean | First click | Used to start game without getting game over. | Board {} |
| boolean | Losegame | Flag used to indicate if user has gotten game over. | Board {} |
|  | Array[] | Array that sets the whole board with its value | Board {} |
| Boolean | Wongame | Flag to check if the game was won | Board{} |
| Int | Time, row, column, numbombs | Time is for the timer to start at 0.  Row/Column set up the size of the board.  numBombs set the number of bombs on the board | Board{} |

|  |  |
| --- | --- |
| Functions | Descriptions |
| fire | Now Passes over row, column, and event over to Onfire in Controller. |
| Generate Grid | Creates and places bombs within the board. |
| inilGrid | Makes numbers on the board. |
| filltable | Fills each square within the board with a number, blank, or bomb. |
| startopen | Checks in range and opens 1 square, used within the open function to open multiple squares. |
| open | Opens a 3 X 3 when you select a square. |
| floodfill | Opens multiple squares when a 0/blank square is selected, depending on adjacent 0’s. |
| firstpick | This function works with first click; it shows a grid with no bombs on it for the first click only. |
| nobomb | This function is used within firstpick, it helps generate the board 1st with no bombs on it. |
| checkwin | The final function that counts the board to determine if all non-mine squares have been opened, if they have then the win condition is passed. |
| onFire(row,column,event) | Takes the row, column and event from Fire Function to create the board and run the game. |
| startTimer() | This is the Timer function that sets global timer for the board, used to display time and set cookie for ending time. |
| init(ur,uc,ub) | This is the handlerInit Function that takes the url from Menu.html and sets the row, column, and number of bombs |

Concepts for Final:

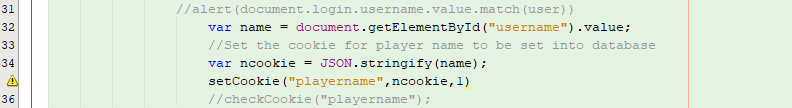
2) Objects - Javascript/PHP - Serialization

Passing info with Javascript Objects to PHP Objects with Cookies

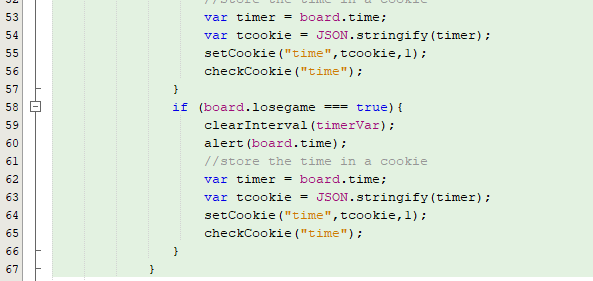
4) Databases SQL - Identify Entities, Xref, and Enum Tables

7) Cookies - Sessions - Securing Pages

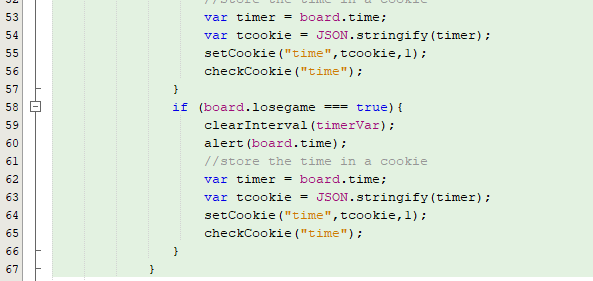
Login.html stores a cookie for the username:



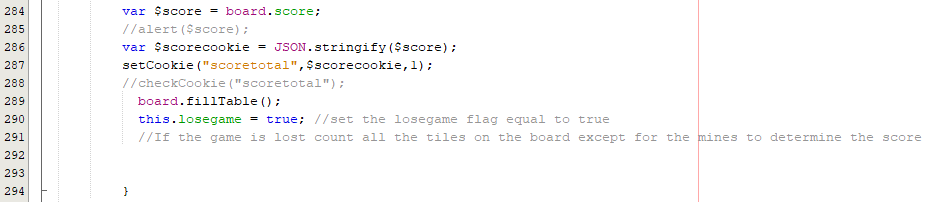
MinesweeperPreset.html stores a cookie for the time:

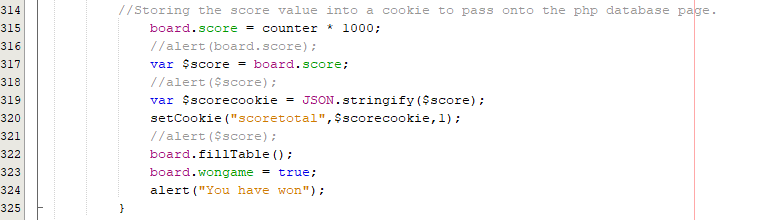


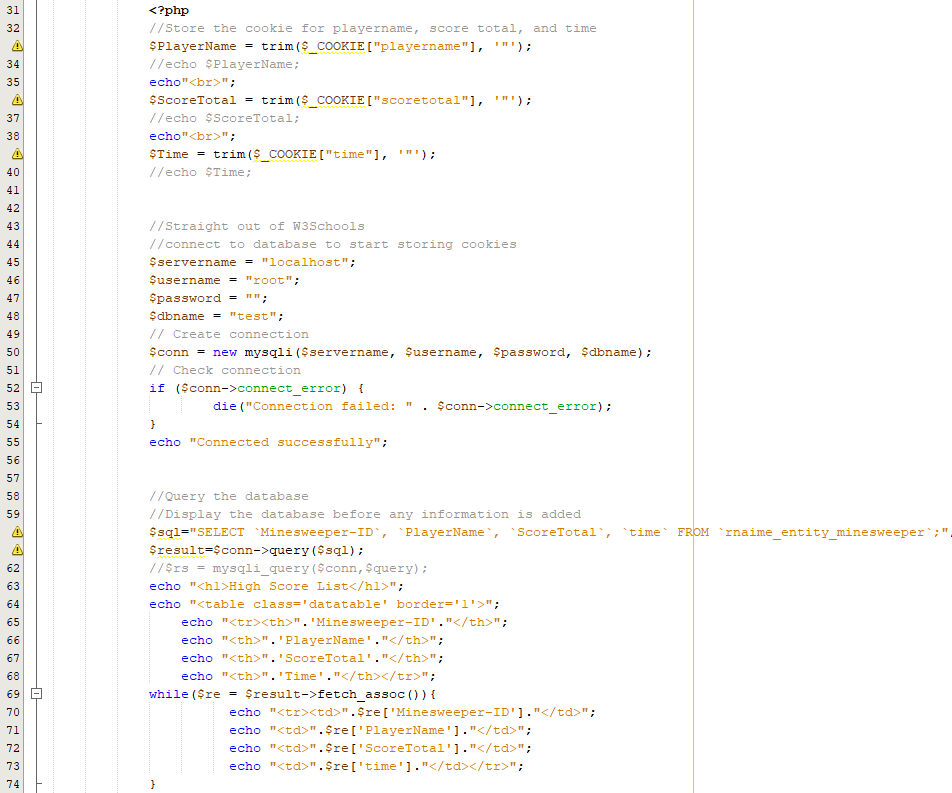
MinesweeperCustomTable.html also stores a cookie for time. These pages are dependent on the difficulty selected.

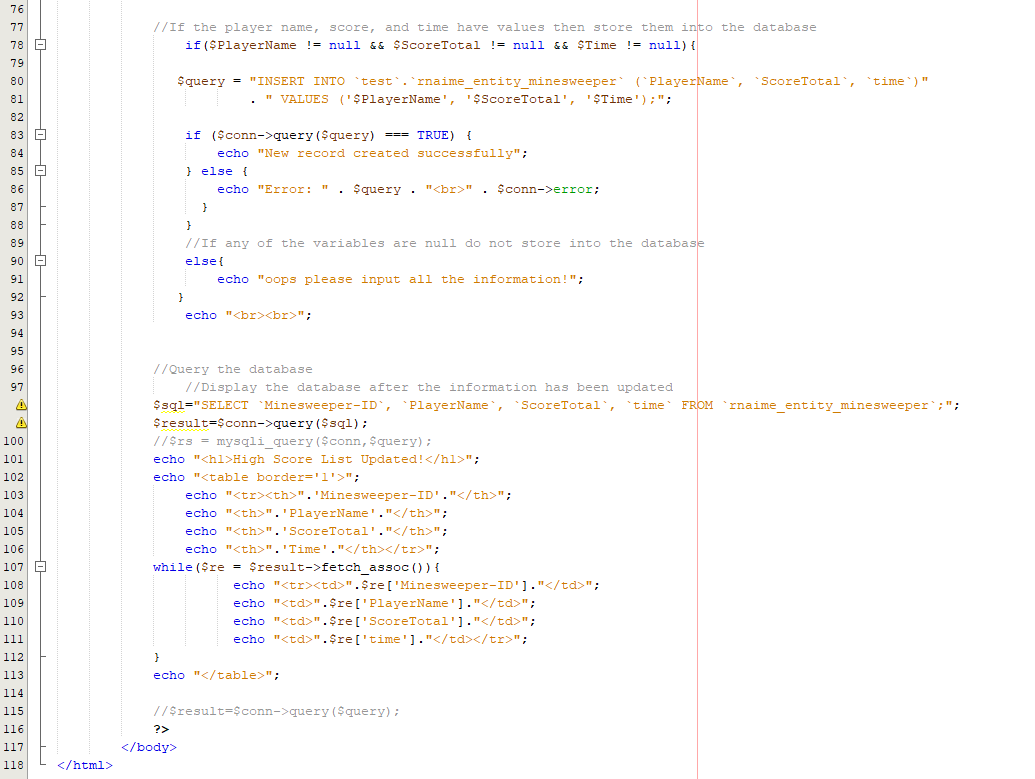


Board.js gamecheck(), wingamecheck()





MinesweeperDatabase.php takes all the cookies from the previous pages and stores them into the database.



5) Form Validation - Reqular Expressions

Login.html uses regular expressions to match the username and password and proceed to the next page.

1) MVC - How you delineated your objects

Minesweeper is broken up by, Board.js, Controller.js, and HandlerInit.js:

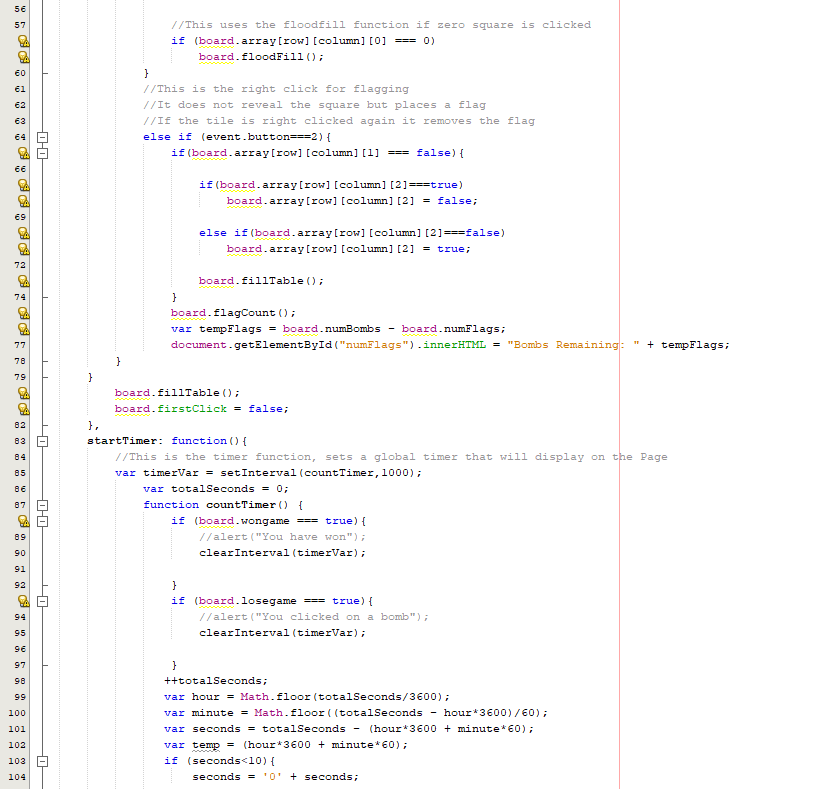
Board.js:

A screenshot of a computer

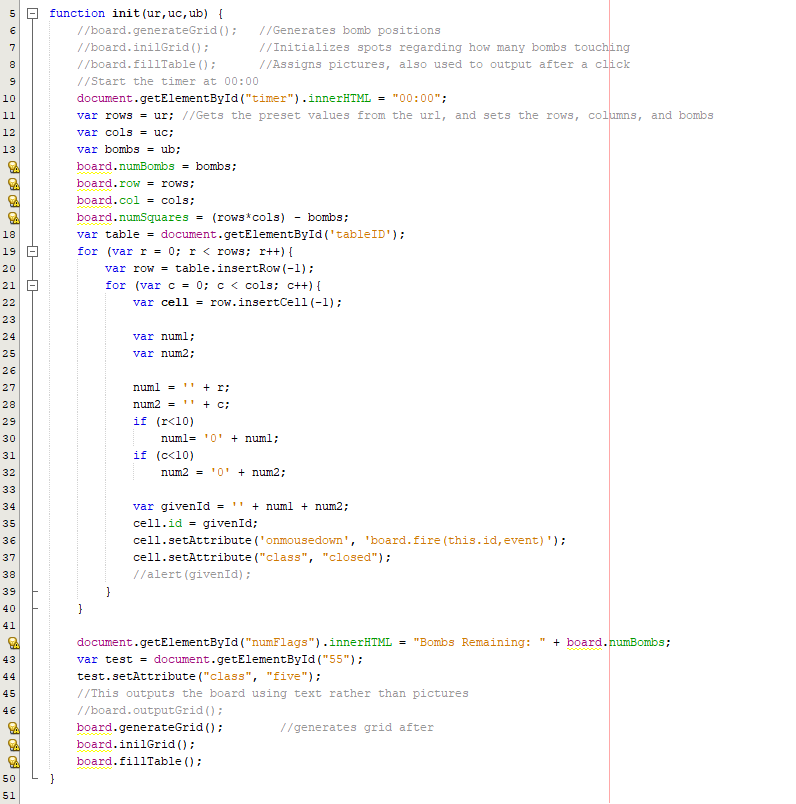
Description automatically generated

Controller.js:





HandlerInit.js:



A close up of text on a white surface

Description automatically generatedFlowchart:

A close up of text on a white surface

Description automatically generated

A close up of a map

Description automatically generated

A close up of text on a white surface

Description automatically generated

A close up of text on a white surface

Description automatically generated

A picture containing text

Description automatically generated

A close up of a map

Description automatically generated

A picture containing text, wall, indoor, building

Description automatically generated

A close up of a map

Description automatically generated

A picture containing text

Description automatically generated

A close up of a map

Description automatically generated

A picture containing indoor, wall, shoji

Description automatically generated

A close up of text on a white background

Description automatically generated

A close up of a map

Description automatically generated

A close up of text on a white background

Description automatically generated

A close up of a map

Description automatically generated

A close up of text on a white background

Description automatically generated

A close up of a piece of paper

Description automatically generated

A picture containing text

Description automatically generated

A picture containing text, sky, indoor

Description automatically generated

A close up of a map

Description automatically generated

A picture containing athletic game, shoji, sport

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

Psuedo Code:

*Board.js*

*Set Board size, number of empty or number squares.*

*Create Array of any large number sized grid.*

***Fire:***

*Row and Column pass the id of the table, converts them integers and then passes Row and column into function onfire from controller.*

***onFire:***

*if the user has not made the first click*

*then the first click is true, grid is generated without bombs*

*It is then initialized, and the table initial table is filled*

*Else if first click has been made then*

*if flag is false and if reveal is false then*

*Then reveal is set to true to later allow the user to click the square*

*Next check to see if lose condition has been fulfilled*

*The function floodfill is then called to open all 0 squares adjacent to click*

*Update and fill the board*

*Else if reveal has been set to true*

*Open function is called to allow the user to open squares on the grid*

*Floodfill is called*

*Then the loss check is called, and the table is filled. After win check is called*

*Else if flag is true*

*Set reveal to false, and place flag over the button*

*Once checks have been complete fill table one last time*

*And set firstclick equal to false to stop initial grid from being created again*

***GenerateGrid:***

*Create bomb variable*

*The bomb variable is indicated by a -1 on the board.*

*While number of bombs is less than 15*

*Randomize up to 15 bombs on the board, and stop any duplicates on the same square.*

***InilGrid:***

*If square is not a bomb*

*When clicked it will open a 3X3 around the square selected*

***FillTable:***

*Creates the 10X10 Grid that matches the table in Minesweeper.html*

*Set location using x and y coordinates.*

*Set cell variable to location*

*If reveal is true*

*If the coordinate is equal to 0*

*Set blank square*

*Else if the coordinate is equal to -1*

*Set bomb*

*Else if the coordinate is equal to 1*

*Set one*

*Else if the coordinate is equal to 2*

*Set two*

*Else if the coordinate is equal to 3*

*Set three*

*Else if the coordinate is equal to 4*

*Set four*

*Else if the coordinate is equal to 5*

*Set five*

*Else if the coordinate is equal to 6*

*Set six*

*Else if the coordinate is equal to 7*

*Set seven*

*Else if the coordinate is equal to 8*

*Set eight*

*Else if the coordinate is equal to 9*

*Set nine*

*Else if reveal is false*

*Set closed*

*Else if flag is set to true*

*Set flag*

***StartOpen:***

*If input is within range, then reveal square*

***Open:***

*Calls start open function for every coordinate of the grid so that they can be revealed*

***Floodfill:***

*If 0 square is clicked, all 0 squares touching it must be opened*

*Calls the open function to reveal the tile and check adjacent squares*

***noBomb:***

*when a click is made, it cannot be a bomb*

***FirstPick:***

*Calls nobomb function for every coordinate of the grid so that they can be revealed, and this is the first grid before the initial click, after this function is not called again until a reset*

***gameCheck:***

*Creates a clickbomb flag to indicate whether bomb was clicked*

*Loops through the table*

*If bomb is revealed, then*

*Clickbomb flag is true*

*If bomb is clicked set the cookie for the score which is number of open squares times a thousand.*

*If clickbomb is true*

*Output you have hit a mine, and set losegame to true*

*Then fills the table*

***checkWin:***

*Creates a counter and loops through the table*

*Adds to the counter for every square revealed that is not a bomb*

*If counter is equal to all available non-bomb spaces*

*Then output winning message “You have won”*

*If the player has won, set the score which is number of open squares times a thousand to a cookie to be placed into the database.*

***StartTimer****:*

*Sets a timer for minesweeper when the game is started. Will continue to increment until the game is lost or won.*

*Once the game is complete the timer will stop and set a cookie to be passed over and stored into the database.*

A screenshot of a computer

Description automatically generatedProgram: Board. js

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generatedHandlersInit.js

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generatedController.js

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

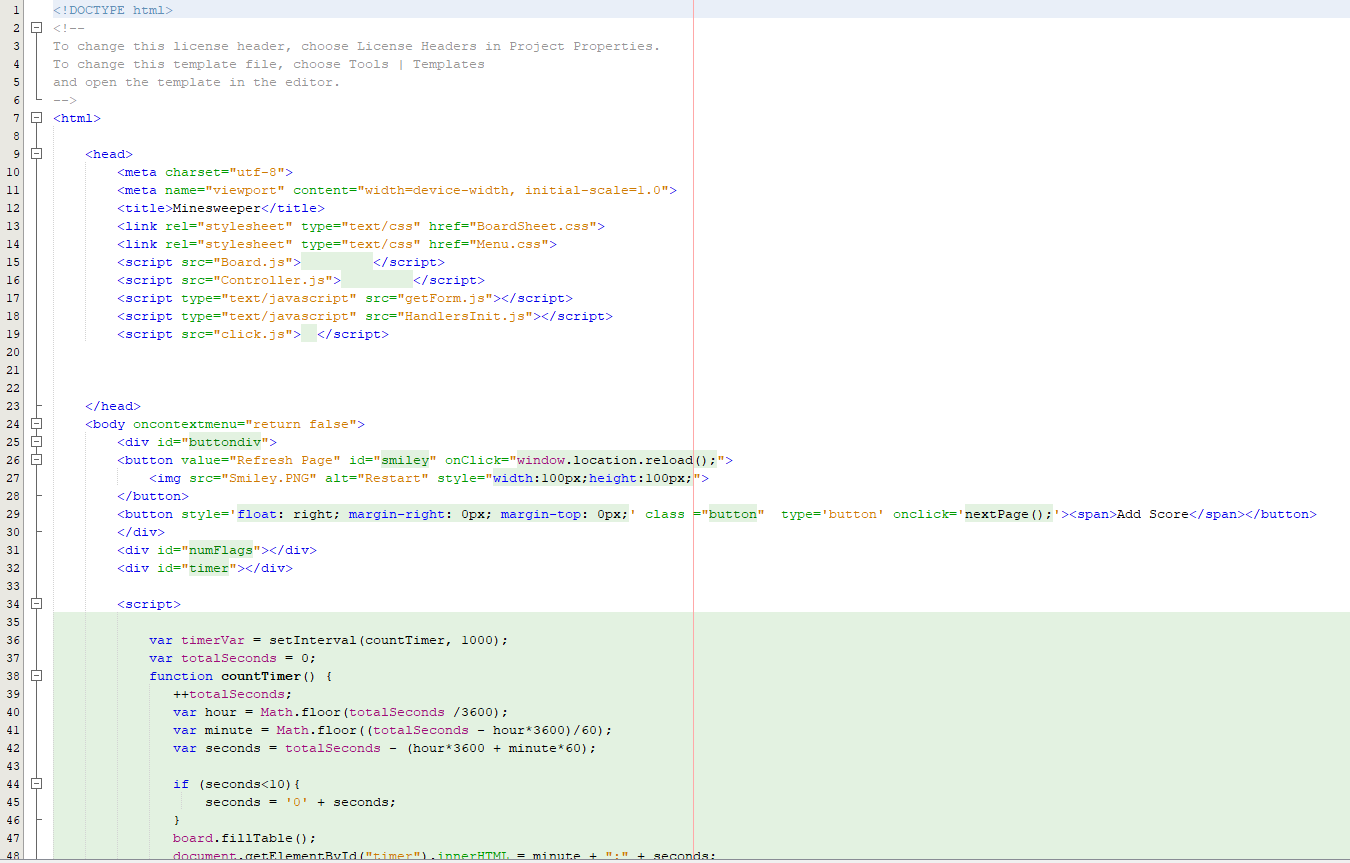
Description automatically generatedLogin.html

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generatedMenu.html

MinesweeperPreset.html

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generatedMinesweeperCustomTable.html

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generatedMinesweeperCustomInput.html

A screenshot of a social media post

Description automatically generated