

```

% Names: Maya Tuinstra, Maia Judd, UnityIDs: mtuinst, mljudd
% NCSU Emails: mtuinst@ncsu.edu, mljudd@ncsu.edu
% Date: 2/26/20
% Lab Sections: 206 & 211
% Project 2: Minesweeper Game, Spring 2020

clc ; clear ; close('all')
%%
%welcome the player to the game
fprintf('*****\n');
fprintf('Welcome to the Minesweeper Game!\n');
fprintf('*****\n');
%%
playAgain = 1; % while loop and variable = 1 and will restart at
% the end of the game unless the player enters 2 which will end the while
% loop
while playAgain == 1
%the board is a 10x10 grid
sizeBoard = 10;
%there are 10 mines
numMines = 10;
%use genMineCoordinates to generate random coordinates of the mines
mineCoords = genMineCoordinates(sizeBoard,numMines);
%create a 10x10 matrix of zeros to use as the empty board
solBoard = zeros(length(mineCoords));
%place a 9 where all of the mines are
for i = 1:length(mineCoords)
    mineRow = mineCoords(i,1);
    mineCol = mineCoords(i,2);
    solBoard(mineRow,mineCol) = 9;
end
%use the markSquaresAdjToMines function to mark all of the squares on the
%solution board with the number of mines that are near them
solBoard = markSquaresAdjToMines(solBoard);
%create the player board to be a 10x10 array of unturned squares
playerBoard = 11 * ones(length(solBoard),length(solBoard));
%repeat the player's turn while the player has not won using the isItAWin
%function
while isItAWin(playerBoard,numMines) ~= 1
    row = input('Choose row: '); %ask which row the player wants to choose
    col = input('Choose column: '); %ask which column the player wants to choose
    %use validUserSquareInput function to check if the input the user
    %provided is valid
    inputValid = validUserSquareInput(row,col,playerBoard);
    %while input is not correct keep asking for new input
    while inputValid == 0
        disp('Incorrect Input. Try Again');
        %ask for input for rows and columns
        row = input('Choose row: ');
        col = input('Choose column: ');
        %reevaluate if input is valid
        inputValid = validUserSquareInput(row,col,playerBoard);
    end
    %sets user move in game to 1 2 or 3 which determines course of action
    %in the game

```

```

userMove = input('Reveal(1), Flag-as-mine(2) or Unflag(3): ');
%while the user's move is invalid, the program asks the user for a new
%move
while (userMove ~= 1) & (userMove ~= 2) & (userMove ~= 3)
    disp('Incorrect Input. Try Again');
    userMove = input('Reveal(1), Flag-as-mine(2) or Unflag(3): ');
end
%switch statement used to determine move actions and consequences
switch userMove
    %if the user reveals square
    case 1
        %reveals adjacent squares
        playerBoard = revealAdjEmptySquares...
            (playerBoard, solBoard, row,col);
        %if the square has a bomb under it
        if solBoard(row,col) == 9
            % show the entire board
            playerBoard = solBoard;
            % set the square = to 13 which shows an exploded bomb
            playerBoard(row,col) = 13;
            %displays board with one exploded bomb and the rest of the
            %minefeild if player hits a bomb
            displayBoard(playerBoard);
            %print statement tells player they have lost if they hit
            %the mine
            fprintf('*****\n');
            fprintf('A mine has exploded! GAME OVER!\n');
            fprintf('*****\n');
            %break statement breaks the while loop asking for input
            break
        end
        % flags an unturned board square
    case 2
        playerBoard(row,col) = 12;
        %unflags a board square
    case 3
        playerBoard(row,col) = 11;
end
%show board after moves executed
displayBoard(playerBoard);
end
%if the player has won the game shows the entire board and prints a win
%statement
if isItAWin(playerBoard,numMines) == 1
    playerBoard = solBoard;
    displayBoard(playerBoard);
    fprintf('*****\n');
    fprintf('Congratulations! You won the game!\n');
    fprintf('*****\n');
end
% asks player if they would like to play again
playAgain = input('Do you want to play again? Yes(1), No(2): ');
%switch statement determines if another game will be played
switch playAgain
    case 1

```

```
    % game will restart because there is no update to playAgain that
    % would stop the gameloop
    fprintf('*****\n');
    fprintf('        Great! New game beings!\n');
    fprintf('*****\n');
    %if case 2 then the playAgain variable no longer equals 2 and the
    %while loop will end
case 2
    fprintf('OK, thanks for playing! Goodbye!\n');
end
end
```