Design Document
Erin Keith
Minesweeper Game

Data

Field – 2D array of int (8 x 8)
Uncovered – 2D array of chars (8 x 8)
File pointers field, scores
Won – bool
Continue choice – char
Player Name – string
Player Wins – int
Saved Player Names – array of strings
Saved Scores – array of ints

Game Play

Player views scores. Field is loaded and data is reset. Player views board and provides move. Board is updated. If the player has not lost (by revealing an unmarked bomb or revealing an incorrectly marked cell), this continues until all bombs have been properly marked and all other cells have been uncovered, at which point the player wins. When a game ends, they player can choose to play again. If they do not, their name and score is added to the player scores.

Functions

main()
Data: Arrays for field and player moves. Variables for won, wins, and play again.
Functionality: Load the field from the file. Display scores. In a loop, clear player moves, play the game, update wins, prompt to play again. After the player is done playing, save their score.

loadField()
Input Parameters: File pointer, int rows, int columns, int field 2D array
Returned Output: none
Functionality: Load values from file into field 2D array.

clearMoves()
Input Parameters: int rows, int columns, char moves 2D array
Returned Output: none
Functionality: Set all elements in the moves array to hidden.

playGame()
Input Parameters: int rows, int columns, int field 2D array, char moves 2D array
Returned Output: boolean
Functionality: Display cells. Repeat getting the user's move, updating the user moves, displaying cells, and checking to see if the user lost or won until the player has lost or won. Return whether the game was won.

displayCells()
Input Parameters: int rows, int columns, int field 2D array, char moves 2D array
Returned Output: none
Functionality: Display a line. Display column numbers. For each row, display the row number, then decide what to display for each cell in the row. If the cell is uncovered, display either a blank, a bomb, or a number depending on the field. If the user has marked the cell, display a "flag". Otherwise, the cell remains hidden. Display another line.

getMove()
Input Parameters: int* row, int* column
Returned Output: char
Functionality: Continue to prompt the player for a cell location until they have entered a valid location. Continue to prompt the player for a move type until they have entered a valid move type. Update the row and column locations to work with the arrays. Return the move type.

updateMoves()
Input Parameters: int rows, int columns, char moves 2D array, char move type, int row location, int column location
Returned Output: void
Functionality: If the player marked a cell as a bomb, update that cell to marked. If the player uncovered a cell, update that cell to uncovered. If the player checked a cell, mark the 8 surrounding cells in all directions (if those cells are "in bounds") as uncovered.

uncover()
Input Parameters: char moves 2D array, int row location, int column location
Returned Output: void
Functionality: If the current cell is hidden, mark it as uncovered.

uncoverBomb()
Input Parameters: int rows, int columns, int field 2D array, char moves 2D array
Returned Output: bool
Functionality: Check each element of the field and player moves for an uncovered bomb. Return true if found, otherwise return false.

checkBomb()
Input Parameters: int rows, int columns, int field 2D array, char moves 2D array, int row location, int column location
Returned Output: bool
Functionality: Start at the player's chosen cell. Check each of the 8 surrounding cells in all directions (if those cells are "in bounds"). If a surrounding cell is "a hit", return true, otherwise return false.

isHit()
Input Parameters: int columns, int field 2D array, char moves 2D array, int row location, int column location
Returned Output: bool
Functionality: If the player has marked a cell which does not contain a bomb, return true. If the player has not marked a cell which does contain bomb, return true. Otherwise, return false.

gameWon()
Input Parameters: int rows, int columns, int field 2D array, char moves 2D array
Returned Output: bool
Functionality: If each of the players marks cover a bomb and all other cells are unhidden, return true. Otherwise, return false.