

Project 1

BugSweeper
(MineSweeper clone)

CSC-5 Winter 2018

Joel Avalos

Introduction and Rules:

BugSweeper is a MineSweeper clone, replacing mines with “bugs” to avoid triggering. A 6x4 grid with 24 “X”s representing spots is shown to the user. The point of the game is to clear each space without finding a bug, labelled as a “B” in the game. When the user has successfully cleared each space without encountering any bugs, then the game has been won. Once you clear a space, a number will display, signifying the number of mines adjacent to the space cleared. This is a clue to aid the user in winning the game.

Summary of Development:

Project size: Approximately 800 lines of code.

Number of Variables: Approximately 80.

I tried very hard to get this game working just like MineSweeper, and I feel as though I can say it works very similarly. Unfortunately, I spent most of my efforts into making the game work well rather than including all the concepts we learned in class. Once I realized this, it was too late, as the game was pretty much completely functional, and any more additions would just alter the playability (as well as muddle an already obtuse block of code even further).

Getting the game to work was difficult. It required lots of thought to get everything working right. Most of the trouble was in displaying the grid while keeping the cleared spots cleared, and the numbers to show after each user input. While I realize I did not include many concepts, and should have read the instructions more carefully, I believe I tried my best and am proud of the result.

Example inputs/outputs

Entering in your choice is comprised of an “A”, “B”, “C”, or “D”, followed by the number (1-6) of the spot you wish to clear. For example, entering in an A1 would clear the top left space on the grid. If that space is not a bug, then a number will display, that number is dependent on how many bugs are nearby. So say the only neighboring space with a bug is A2, then the A1 space will display a “1” after it is cleared. If the space is a bug, a “B” will display and the game will be over!

Psuedocode:

Initialize

Display start menu

If the user starts the game, display the grid and wait for an input.

Else if the user wishes to see rules, display rules.

Else, quit the program.

If the user input has an "A", clear the numbered space in row A.

Else if the user input has a "B", clear the numbered space in row B.

Else if the user input has a "C", clear the numbered space in row C.

Else if the user input has a "D", clear the numbered space in row D.

If the space is a bug, output a "B" and end the game.

Else, keep asking the user for input until a bug is triggered, or all spaces except bugs have been cleared, in which case the game is won. Return to start menu.

Flowchart: see project Folder