NotYourAveragePatcho APCS pd8

Faiza Huda, Gabriel Thompson, Jacob Ng Fall 2022

Final Project Proposal
M1N3SW33P3R

Overview

We plan to make minesweeper on the terminal by implementing java code. This project will include a board that can be interacted with by means of a coordinate system and a time based score. We'll be using inspiration from the Google version of Minesweeper.

Tools We Might Use

While nothing is final, and our vision so far is just a work in progress, we predict that we'll be using the following topic/tool:

- 2d array - to display the minefield and to keep track of the bombs
- Insertion Sort - to keep track of the scores in one specific sitting and update to present the highest score
- Boolean values - to indicate if the tile has a mine
- ANSI escape code - to color code the numbers to make gameplay easier
- currentTimeMillis- to keep track of time/score
- Scanner - to take in user input

In order to implement the idea of minesweeper on the terminal, we will create two 2d arrays. One will display the actual field that the user interacts with and the other will keep track of all the mines and numbers which are used to indicate how many mines are near. These mines will be randomly generated to be 15% of the grid size and will be represented by boolean values where 'true' indicates a mine.

The empty 2d array will be printed out every time the end user interacts with it. In order to select a tile, there will also be coordinates horizontally: A-Z[1] and vertically: 1-10[1]. This array will keep track of the viewed and flagged tiles. We will also keep track of the grid size, number of mines, number of flagged tiles, and the score time.

Every time the user opens a tile, a new board will be printed wherein the tile that was previously selected will now have a value of how many mines

[1]These coordinates are subject to change due to grid sizing

are nearby. This value will have been already stored by the second, hidden array that at the start used a for loop to count the nearby mines. The number will also be a certain color which we will achieve through ANSI escape code. At the end of the game, the user's score will be calculated using currentTimeMillis and stored in a score array. If it is the highest value, a message along the lines "New High Score" will be displayed. If not, then both the highest score and the score from the game will be printed with the option of the user to play again.

Gameplay

1. At the start of the game, you are presented with a grid characters representing tiles on your screen, and told to choose a starting row and column to select
2. In each following "turn":
    a. An updated screen is displayed showing the current status of your board (tiles cleared, tiles with numbers, tiles marked as mines). The row and column marks are displayed on the side of the board
    b. The player is told to choose a row and column to select. The player can determine which is which by the marks on the side of the board.
    c. The player is then asked what to do with the tile (either flag, open, or cancel move). If the move is canceled, go back to step 2a[2]
    d. If the tile selected was a mine, and the player has selected to **open** said tile, end the game and inform the player they lost.
    e. If, alternatively, the player has cleared all of tiles *except* for every mine, go to step 3
    f. If neither 2d nor 2e, take one of the following actions:
        i. If the player select to "flag" the tile, mark the tile as flagged
        ii. If the player selected to "open" the tile, perform a recursive flood fill of the tile and surrounding tiles, until the flood fill reaches a mine
        iii. If the player canceled their move, go back to step 2a[2].

UML Diagram of one class we already confirmed to have

Board
[-] int width **[-] int height**

[2] It's best to go back to 2a, rather than 2b, because the prompts may clog up the screen, and make it harder to make a selection. It's best to just re-display to board.


[-] int minesCount
[-] boolean [][] mines
[-] boolean [][] viewed
[+] Board (int,int)
[+] void generateMines()
[+] void displayBoard()
[+]