Lab 00 Report: Mine Counter!

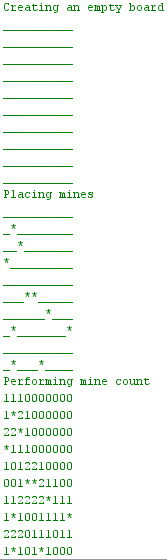
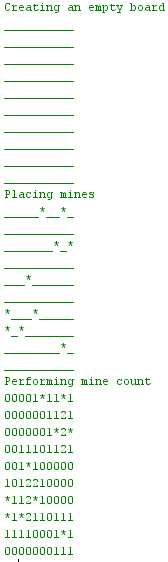
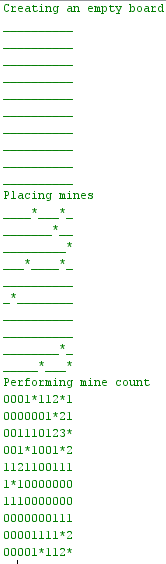
Problem

We had to write a program in which a 10x10 board through a multidimensional array is generated with empty spaces, then selects 10 random spaces on the board and replaces the empty spaces or ‘\_’ with mines ‘\*’ printing it, and finally prints the board for the last time with a mine count ranging from 0-8 based on how many mines are surrounding the place on the board.

Proposed Solution

1. Create a multidimensional array of size 10x10.
2. Print “Creating an empty board”
   1. Use a nested for loop to populate the entire board with the value of 0.
   2. Print the board using a nested for loop.
3. Print “Placing Mines”
   1. Using the random util, inside a for loop that counts up to 10, generate a random x and y position for a mine then set the position to 10 (mine value), and if the position is already a mine decrement by 1.
   2. Print the board using a nested for loop, 0 = ‘\_’ and 10 = ‘\*’.
4. Print “Performing mine count”
   1. Using a nested for loop, check all positions around the current x and y the for loop is at to count the number of mines, adding to a countOfMines variable. These statements must also check that the position it is looking for is inside the 10x10 multidimensional array. Then save the number of surrounding mines at the current place of the array.
   2. Print the board using a nested for loop, 10 = ‘\*’ and for the positions that are not a mine, print the value stored in the array.

Tests and Results

Problems Encountered

Checking the boundaries for the array during the minecount was the hardest portion of this lab. If the program were to check for an imaginary position outside of the multidimensional array, an array out of bounds exception would be thrown. This is because I was not checking for the positions until I recognized this as an issue, and resolved it by making sure the position was in the boundaries of 0 & 10 and the point it was checking was also a mine before adding it to the count.

Conclusions and Discussion

In this lab we revisited multidimensional arrays from 145, and did similar boundary checks such as moving the player in 145 HW 6 ‘Minesweeper’. The way it worked was by creating a multidimensional array with finalize­­d size using variable modifiers, used the random class to generate mines, and many nested for loops to populate or print the board.

While this solution works, one can also solve the lab using enumerations and switch statements to print and manage the board, and also implement extra methods to separate mine generation, and printing the board. Instead of using 0 for ‘\_’ one could’ve used Enum{Empty,Mines,MineCount} and used Empty, etc…

Additional Questions

1. Is it possible to change the size of a standard array in java after it has been constructed?

Arrays are fixed length in java, so it is not possible to change the length of an array in Java after it has been constructed.

1. A multidimensional array can be thought of an array of arrays. In java, does each array of another have to be the same size or does java support ragged arrays?

Ragged arrays are multidimensional arrays that contain arrays at the same dimension with differing lengths, which are supported by java.