CSCE 1030: Homework Assignment 3
Due: 11:59 PM on Sunday, March 29, 2020

PROGRAM DESCRIPTION:

In this assignment, you will be creating an ESCAPE the LAVA game in C++. In this game, the user will escape the dungeon of a 10x5 array. For example, the array could be initialized like the following:

	0	1	2	3	4
0	X	X	E	X	X
1	X	E	X	E	X
2	X	R	E	X	X
3	X	X	X	R	X
4	R	X	R	X	X
5	X	X	R	X	X
6	X	X	X	R	X
7	X	R	X	R	X
8	R	R	X	X	X
9	X	X	X	R	X

In this case, X represents an empty slot. The E represents an Exit, and the R represents a Rock. The user will also have a limited number of guesses, and if they do not find an Exit before they run out of guesses they lose the game.

Each turn, the bottom layer of the dungeon will be flooded with LAVA, and all Letters will be replaced by 'L'. Should the player select a 'L', they will lose one additional guess!

To make the game a bit more challenging, your program will actually maintain two, 10x5 arrays. One array, the viewable array, will display which displays user choices and will be initialized to all X's. This array will need to be updated as the user inputs guesses. The other array, the hidden array, will actually contain the information about where the objects are located (e.g. Rocks, Exit, Lava and Empty space). The example array above represents a possible hidden array.

Game Rules:

- Finding X: Lose a turn. Lava rises by one row starting from the bottom.
- Finding R: Lose a turn, but Lava does not rise.
- Finding L: Lose two turns, Lava rises.
- Finding E: Win!

PROGRAM REQUIREMENTS:

- As with all homework programs in this course, your program's output will initially display the department and course number, your name, your EUID, and your email address. This output should be performed through a function you declare and define.
- You will need to declare and define a function to initialize the value of all slots in both arrays to X initially, and then assign the Rocks and Exits to random locations in the hidden array.
    - It should have two, two-dimensional `char` arrays and an integer as its parameters.
    - You should initialize exactly 10 Rocks and 5 Exits.
    - It should not return a value.
- You will need to declare and define a function to output the `char` value of every slot in an array in a grid format, along with the numeric headers. See the sample output for an example.
    - It should have a constant two-dimensional `char` array and an integer as its parameters
    - It should not return a value.
- You will need to declare and define a function to update the `char` value of every slot in both the hidden and viewable arrays, this should update the bottom row with lava. See the sample output for an example.
    - It should have a constant two-dimensional `char` array , an integer, and a reference integer as its parameters
    - It should replace any existing character value within the bottom row of each array with the char 'L'.
    - It should only replace non-lava rows.
    - It should replace the row directly above the existing lava row if lava already exists.
    - You can maintain the currently level of the lava through an integer value. You can pass the value to this function by reference in the parameters
    - It should not return a value.

- You will need to declare and define a function to process a user's guesses. A user will guess an X and Y coordinate positions to attempt to find an 'E' the Exit. After each guess, you should inform the user what symbol was at the guessed position or if it was empty take the according actions.
    - The function should have two, two-dimensional `char` arrays ,an integer and two reference integers as its parameters.
    - It should return a `bool`.
    - The function should prompt the user for one set of x and y coordinates within the bounds of the array.
    - If the user guesses a position that is outside of the game area, then the user has forfeited their attempt, and should be informed of this. The function should return false.
    - If the guess discovered an X, they lose a turn and the Lava rises.

        - **If the guess discovered a Rock, 'R', they lose a turn, but the Lava does not rise!**
        - If the guess is the Lava, 'L', they LOSE an EXTRA turn.
        - If the guess discovered an Exit, 'E', they win!
        - If the Number of Guesses reaches 0 without finding the Exit, Inform the user that they have LOST and display the resulting array.
- Inside your main function you will need to
    - Create two 10x5 `char` arrays, one hidden array to store the positions of all of the random letters, and one visible array to display the locations that have been discovered.
    - Create a constant integer `ROW`, initialized to 10, which represents the rows of the arrays.
    - Create a constant integer `COLUMN`, initialized to 5, which represents the columns of the arrays.
    - Initialize both char arrays using the appropriate function.
    - Using an appropriate loop, you should allow the user 10 attempts to escape the dungeon.
        - Each attempted guess should use the appropriate function
        - After each successful or failed find or guess, the visible array should be output to the user via the appropriate function.
        - As soon as the user succeeds in finding an Exit, they win the game.
        - If the user fails to escape in 10 tries, they lose the game.
    - At the end of the game, the user should be informed whether they won or lost, and the hidden board should be displayed via the appropriate function.
- Your code should be well documented in terms of comments. For example, good comments in general consist of a header (with your name, course section, date, and brief description), comments for each variable, and commented blocks of code. This means, that in addition to the program printing your information to the terminal, it will also appear in the code in the comments as well.
- Your program source code should be named "**euidHW3.cpp**", without the quotes. where euid should be replaced by your EUID.
- Your program will be graded based largely on whether it works correctly on the CSE machines (e.g., cse01, cse02, …, cse06), so you should make sure that your program compiles and runs on a CSE machine.
- This programming assignment is designed to help you practice your coding on a larger project with various pieces of functionality. While the coding should be primarily your sole work, any help you receive from outside sources (TAs, Instructors, Other Individuals, Online Sources) should be specifically cited in a comment block at the beginning of your code. If your code is determined to be too similar to a cited outside source you will receive a 0 for the assignment. If your code is determined to be too similar to an uncited outside source, this will be interpreted as plagiarism and will result in a grade of "F" for the course, along with a report filed into the Academic Integrity Database.

DESIGN (ALGORITHM):

On a piece of paper (or word processor), write down the algorithm, or sequence of steps, that you will use to solve the problem. You may think of this as a "recipe" for someone else to follow. Continue to refine your "recipe" until it is clear and deterministically solves the problem. Be sure to include the steps for prompting for input, performing calculations, and displaying output. You should attempt to solve the problem by hand first (using a calculator as needed) to work out what the answer should be for a few sets of inputs.

Type these steps and calculations into a document (i.e., Word, text, or PDF) that will be submitted along with your source code. Note that if you do any work by hand, images (such as pictures) may be used, but they must be clear and easily readable. This document shall contain both the algorithm and any supporting hand-calculations you used in verifying your results.

SAMPLE OUTPUT:

Here is a sample output to help you write and test your code. The item in bold is the information entered by the user.

```
$ ./a.out
+---------------------------------------------+
|       Computer Science and Engineering      |
|        CSCE 1030 - Computer Science I        |
|   Student Name     EUID    euid@my.unt.edu   |
+---------------------------------------------+
{HIDDEN BOARD}
  0 1 2 3 4
0 X E R X X
1 X E R X X
2 X X R X X
3 X E E X X
4 X X R R X
5 R R X X X
6 X X X X X
7 R R X X X
8 X R X X X
9 X X X X X
```

(FOR testing and demonstrations purposes, the hidden board is revealed ABOVE. Do not reveal it in the submission!)

```
(VIEWABLE BOARD)
  0 1 2 3 4
0 X X X X X
1 X X X X X
2 X X X X X
3 X X X X X
4 X X X X X
5 X X X X X
```

```
6 X X X X X
7 X X X X X
8 X X X X X
9 X X X X X
10 Guesses left.
Please enter an x and y coordinate:9 4
You found an empty tile! The Lava rises!
  0 1 2 3 4
0 X X X X X
1 X X X X X
2 X X X X X
3 X X X X X
4 X X X X X
5 X X X X X
6 X X X X X
7 X X X X X
8 X X X X X
9 L L L L L

9 Guesses left.
Please enter an x and y coordinate:10 10
Your position is outside the bounds of the game!
  0 1 2 3 4
0 X X X X X
1 X X X X X
2 X X X X X
3 X X X X X
4 X X X X X
5 X X X X X
6 X X X X X
7 X X X X X
8 X X X X X
9 L L L L L

8 Guesses left.
Please enter an x and y coordinate:9 4
You found the LAVA! You lose an extra guess!
  0 1 2 3 4
0 X X X X X
1 X X X X X
2 X X X X X
3 X X X X X
4 X X X X X
5 X X X X X
```

```
6 X X X X X
7 X X X X X
8 L L L L L
9 L L L L L
6 Guesses left.
Please enter an x and y coordinate:0 0
You found an empty tile! The Lava rises!
  0 1 2 3 4
0 X X X X X
1 X X X X X
2 X X X X X
3 X X X X X
4 X X X X X
5 X X X X X
6 X X X X X
7 L L L L L
8 L L L L L
9 L L L L L

5 Guesses left.
Please enter an x and y coordinate: 5 0
You found a Rock, ouch! Please try again.
  0 1 2 3 4
0 X X X X X
1 X X X X X
2 X X X X X
3 X X X X X
4 X X X X X
5 R X X X X
6 X X X X X
7 L L L L L
8 L L L L L
9 L L L L L

4 Guesses left.
Please enter an x and y coordinate:0 1
You found the Exit! GOOD JOB! YOU WON!
  0 1 2 3 4
0 X E R X X
1 X E R X X
2 X X R X X
3 X E E X X
4 X X R R X
5 R R X X X
```

```
6 X X X X X
7 L L L L L
8 L L L L L
9 L L L L L
```

TESTING:

Test your program to check that it operates as desired with a variety of inputs. Then, compare the answers your code gives with the ones you get from hand calculations.

SUBMISSION:

- Your program will be graded based largely upon whether it works correctly on the CSE machines, so you should make sure your program compiles and runs on the CSE machines.
- Your program will also be graded based upon your program style. This means that you should use comments (as directed), meaningful variable names, and a consistent indentation style as recommended in the textbook and in class.
- We will be using an electronic homework submission on Canvas to make sure that all students hand their programming projects on time. You will submit both (1) the program source code file and (2) the algorithm design document to the **Homework 3** dropbox on Canvas by the due date and time.
- Note that the dates on your electronic submission will be used to verify that you met the due date and time above. All homework up to 24 hours late will receive a 50% grade penalty. Later submissions will receive zero credit, so hand in your best effort on the due date.
- As a safety precaution, do not edit your program (using vim or nano) after you have submitted your program where you might accidentally re-save the program, causing the timestamp on your file to be later than the due date. If you want to look (or work on it) after submitting, make a copy of your submission and work off of that copy. Should there be any issues with your submission, this timestamp on your code on the CSE machines will be used to validate when the program was completed.