CPSC1420 Programming and Problem Solving I

Fall 2020

Homework 4

Due: 6:00 pm, Friday, Nov. 6

A popular card game (with regular cards or special ones) is to lay out a tableau of a number of cards placed face down. There are two copies of each card on the table. By turning over no more than two cards at a time, the point is to find the matching cards and remove both cards from play, turning over as few cards as possible.

In C++, using arrays and functions, create a memory game:

1. Read in the file with card locations.
2. Display the field – an * (asterisk) indicates a card that can be turned over, an _ (underscore) indicates a card that has been removed from play.
3. Allow the user to choose a card (using row and column labels), and turn over that card. Then allow them to choose a second card, turn over that card and check to see if it is a match. If it is a match, remove both cards from play. If not, start the timer. When the timer is done, turn both cards back.
4. The game is won when all cards have been removed from the field.
5. Count the number of turns and give the user their score (i.e. the number of turns) at the end. Offer to play the game again.

Notes:

- The field for this game will always consist of 12 cards – 6 sets. The layout is three rows and four columns. You must use a 2-dimensional array to store the field.
- There are two input files available (posted on Canvas, named as “p4data1.dat” and “p4data2.dat”). The game randomly chooses which file to use. The input files each contain two copies of the capital letters A … F in random order. Read in the file and place the characters in the 2-dimensional array row by row. You can also make your own file for testing.
- Do not allow the user to choose a card that is already removed from play.
- Turning over two cards counts as a turn and one point on their score. A player cannot “go back” on a turn - they must turn over two cards each time.
- When two cards do not match, the exact “wait” time is your decision (i.e. you can decide how long the program should pause).
- When your program starts execution, it needs to display a welcome message that describes the program to user in a short, but clear message. Blank lines appear before and after the welcome message to help user read the screen. In addition, goodbye message should be displayed to wrap up the program before exiting. Again, blank lines appear before and after the goodbye message to help user read the screen.

Please see below for the information for the “wait” function.

```
#include <ctime>
```

```
void wait(int seconds); //prototype of the wait function
```

```
//pauses the program for seconds
//IN: seconds
void wait(int seconds)
{
  clock_t endwait;
  endwait = clock() + seconds * CLOCKS_PER_SEC;

  while (clock() < endwait)
  {
  }
}
```

A sample output is provided on Canvas, as shown in hw4_sampleoutput.txt. In the sample output, the row numbers and the column numbers are from user input. In addition, in the place where it shows “pausing for 5 seconds”, the program’s execution should pause for 5 seconds (you can change the exact “wait” time from 5 seconds to pausing another amount of time).

Grading

The assignment will be graded in accordance with the “Labs and Programming Assignment Expectation” handout and the rubric posted on Canvas. Failure to adhere to the guideline could result in losing points.

Submitting your Program

Your program must be stored in a ***single*** file called ‘`hw4.cpp`’. Use Canvas to submit your program.

Failure to submit your program before the due date and time will result in a zero. **Programs that fail to compile will result in a zero.** You can submit your program multiple times - only the last submission before the due date and time will be graded.