project[7]: Functions, Arrays, Pointers
Due Wednesday, 11/28/2017, 11:59 am

Project Goals

The goals of this project are to explore the use of functions, 2D arrays and pointers.

Important Notes:

1. **Comments:** Header comments are required on all files, for each function, and recommended throughout the rest of the program. Guidelines for header comments and function comments were discussed in lecture or online (google c comment style or go to https://www.cs.swarthmore.edu/~newhall/unixhelp/c_codestyle.html). Points will be deducted if no header/function comments are included.

Problem 1

For this project you will write a program that implements the game of Tic-Tac-Toe where you can play against the computer (player 1 will be the user, player 2 will be the computer). Your program should go through the following steps:

1. Generate an empty Tic-Tac-Toe table (3x3 array)
2. Run a loop as long as none of the players have placed three in a row (a player has won) and the table is not full (stalemate). This loop should:
    a. Display the current layout of the table
        i. blank squares are displayed as underscores
        ii. O for player 1's moves (user)
        iii. X for player 2's moves (computer)
        iv. put spaces between the squares
    b. If it is player 1's turn (the user): ask the user to enter his/her selection (the location in the table where the X or O should be placed (row, col)). Check that player 1 has entered a legal option (an empty cell); if the selection is on a cell that is already occupied, the program should ask the user to enter the option again.
    c. If it is player 2's turn (the computer): randomly generate a move (row, col) in the table that is not currently occupied (if the computer selects an occupied cell, then skip it and generate a new move). Print out the selection of player 2 (the computer generated option).
        i. **Hint:** to generate the random move, generate two random integers (one for the row and one for the column), each between 0 and 2.
    d. Update the table with the provided option
3. If the above loop ends because one of the players has placed three in a row, the program should print a winning message of that player (see below). Else, it should print exactly the following : “Game over, no player wins.”

The program should function as follows (items underlined are to be entered by the user):

```
This program plays the game of tic-tac-toe
The current state of the game is:
_ _ _
_ _ _
_ _ _

Player 1 enter your selection [row,col]: 1,2

The current state of the game is:

_ O _
_ _ _
_ _ _

Player 2 has entered [row,col]: 2,2

The current state of the game is:
_ O _
_ X _
_ _ _

…

Player 1 enter your selection [row,col]: 1,3
The current state of the game is:

O O O
X X _
O _ X

Congratulations, Player 1 wins!
```

Constraints:

1. Do not use global variables
2. Your program should implement and use the following functions:

- clear_table: this function should take as parameter a 3x3 array and clear it out for the beginning of the game. You are free to chose the type of the array and how you store the values for empty, O or X.
- generate_player2_move: this function should take as parameter the 3x3 array, and pointers to two integers row and col, in which the function will return the valid play position for player 2. This function should make use of the check_legal_option function below to see that the randomly generated move is valid or not.
- check_table_full: this function takes as parameter the table array and returns true or false depending on whether all the cells are occupied or not
- check_three_in_a_row: this function takes as parameter the table array and returns 0 if no player has three in a row (on rows, columns, or diagonals) or the ID of the player (1 or 2) who has three in a row (three Xs represent player 1, three Os represent player 2)

- display_table: this function takes as input the table array and prints out the current status as shown above (the function should print an undescore '_'for an empty cell.
- check_legal_option: this function takes as input the table array and a possible move (row, column where the X or O should go) and returns true or false depending on whether the option is valid or not
- update_table: this function takes as input the table array and the move (row, column where the X or O should go) currently entered by the user and updates the table with the latest entered move.

Your program should be saved in a file called tictactoe.c.

Challenge (10 extra credit points). Make your program run in a loop. This means, that at the end of the game, it should ask the user if he/she wants to play again. If yes, then the table should be reset and a new game started. Your challenge program should be saved in a file called tictactoe_c.c.

Grading Rubric

Grading will be done for each problem as follows:

Correctly-named file	5%
Header comment	2%
Program compiles	5%
Correctly-reading data from terminal	28%
Correct use of functions	30%
Correct result printed	30%

Submission details

To submit your project, you will have to save your project files to nomachine:

- create a directory called "project7"
- save your *.c files in that directory
- TO Submit:
  > cd project7
  > submit

The submission script copies all files in the current directory to our directory. You may submit as many times as you like before the deadline, we only keep the last submission.

Academic Honesty

Academic dishonesty is against university as well as the system community standards. Academic dishonesty includes, but is not limited to, the following:

Plagiarism: defined as submitting the language, ideas, thoughts or work of another as one's own; or assisting in the act of plagiarism by allowing one's work to be used in this fashion.

Cheating: defined as (1) obtaining or providing unauthorized information during an examination through verbal, visual or unauthorized use of books, notes, text and other materials; (2) obtaining or providing information concerning all or part of an examination prior to that examination; (3) taking an examination for another student, or arranging for another person to take an exam in one's place; (4) altering or changing test answers after submittal for grading, grades after grades have been awarded, or other academic records once these are official.

Cheating, plagiarism or otherwise obtaining grades under false pretenses" constitute academic dishonesty according to the code of this university. Academic dishonesty will not be tolerated and penalties can include canceling a student's enrollment without a grade, giving an F for the course, or for the assignment. For more details, see the University of Nevada, Reno General Catalog.