Assignment #5

Eric Rouse; Individual Programming 53

Understanding the Problem

Write 3 classes, 2 of which are games (hangman and tic tac toe) and only contain the methods and attributes necessary for playing those games. The third class is the game player class which calls the appropriate methods from the game classes to make a playable 2 person tic tac toe and hangman.

Devising a Plan/Design

Here is the pseudocode that I came up with.

MAIN

1. Prompt the user:
   `Would you like to play [h]angman or [t]ic tac toe or [q]uit?:`
2. If they want hangman go to the hangman method (below)
3. If they want tic tac toe go to the tic tac toe method (below)
4. If they select quit, exit the program

HANGMAN

1. Determine if the default names are OK.
2. Once names are selected allow player1 to set a word for player 2 to guess.
3. Allow player2 to guess until the either get it right or run out of guesses.
4. Display the blanks for each unguessed letter, the right letter in the right spot for each guessed letter and how many guesses are left between each turn.
5. If player 2 guesses it before they run out of guesses, they win!

TIC TAC TOE

1. Determine if the default names are OK.
2. Once names are selected allow player1 to make their mark on the board by selecting first the row, then the column where they want to put it (0,0 is upper left, 2,2 is lower right).
3. Check if there is a winner.
    a. If winner, print message, quit tic tac toe
    b. Else, keep going
4. Check if the board is full.
    a. If full, print message, quit tic tac toe

    b. Else, keep going
5. Do the same for player 2
6. Check if there is a winner.
    a. If winner, print message, quit tic tac toe
    b. Else, keep going
7. Check if the board is full.
    a. If board is full, print message, quit tic tac toe
    b. Else, keep going
8. Repeat until winner/full board

Error Handling:

TIC TAC TOE

- Player chooses a position that is not on the board, i.e. row 5, column 2.
- Player chooses a position that is already occupied.
- Players do not enter an 'X' or an 'O' or if allowing player 2 to choose, player 2 doesn't select player 1's choice.

HANG MAN

- Player chooses same name as other payer
- Player enters a null word

Looking Back/Self-Reflection

OK, so now I understand that the "this" operator is required for private variables inside a Class. Encapsulation is pretty powerful, I really like how it keeps the program separated into manageable chunks.

It was difficult to make sure the output was correct, there are so many different paths the code can take. There must be some way to automate this…

Also, this time I was rushed to get code written, so I didn't do any psuedocode beforehand. That was dumb. I didn't realize how much that helps organize your mind. It really does help to take the time and think through the code before writing it.