Minh Nguyen

Project 1

1/19/2014

Number guessing game

1: My understanding of the problem for this program to work is that I need to be able to write a loop that successfully allows a person to choose a number, and a second player to be able to guess that number, while at the same time the program must calculate how many guesses remain, and how close they are to the number at every guess.

2. Design for program.

Set integers and variables

*Open a while loop to run entire program with a bool statement to control for replay later

Ask for input from player 1 to choose a number in range 1-20

if (number is in range)

> go to loop to start game

else (number out of range)

>go to loop to prompt player 1 to choose number again

while (game start loop)

create loop to run 3 times with a variable like (tries) and have it (tries<=2) and count it down after each loop

if else-if else statements here for number too low, number too high, and correct number

if number too low, have cout to tell player 2 too low, and do a ++lowerlimit number to show player 2 that their guess was too low and needs to guess again in new range with this updated ++lowerlimit number

else-if number too high, similar to number too low, except something like ++upperlimit

else number guessed correct, set tries=3 so it leaves loop

make a simple if/else math equation that takes players guesses and choose which one is the closest guess. Something similar to ... if guess number < actual number, then offby = ++ actual - real number.

then design a simple y/n question to set the bool statement to a false statement if answer is y so game restarts, or no and to exit entire loop of game.

3. Source file included as .cpp file.

4. I tested my code by playing the game, and tested every possible outcomes. Such as putting in the wrong number ranges, winning immediately, winning after 2 guess, 3 guesses, and losing. I tested the play again feature and it works for both yes and no answers. A no answer exits the game with return 0; while a yes answer repeats the loop of the game. Each loop is entirely a new game, and does not overlap with the previous game. Game outputs correct number that was closest guess and how off you were, if you did not win the game.

5. The techniques I needed for this program was working with while loops and if-else statements. And nesting 2-3 while loops together and keeping track of their positions and account for all brackets needed. Most of my problem that occured was syntax error, usually from not having double = marks in the while loop arguments (ie: while (tries ==8). This caused more trouble than I'd like to admit but, it taught me a valuable lesson in making sure your syntax is correct and saving yourself a splitting headache trying to debug your