Project 2

Yahtzee

CSC5
Juan Castellon
6/5/18

Introduction

Title: Yahtzee

Yahtzee is point-based game played with 6 six-sided dice. The game is played by the players initially rolling the dice to see who goes first based on the highest rolling player. Players then take turns rolling the dice to see who can get the highest rolls and then choosing how to score themselves based on the rolls.

The point system is as follows:

Score 1s: Add up all 1s and add them to your total points.
Score 2s: Add up all 2s and add them to your total points.
Score 3s: Add up all 3s and add them to your total points.
Score 4s: Add up all 4s and add them to your total points.
Score 5s: Add up all 5s and add them to your total points.
Score 6s: Add up all 6s and add them to your total points.
Score 3 of a Kind: Multiply 3 of the same roll together.
Score 4 of a Kind: Multiply 4 of the same roll together.
Score Full House: If 3 dice are of one value and 2 are of another value, gain 25 points.
Score Short Straight: Gain 30 points if you roll 3 dice sequentially.
Score Long Straight: Gain 40 points if you roll 4 dice sequentially.
Score Yahtzee: Gain 100 points if all dice are same value. 50 points for each additional Yahtzee.
Score Chance: Add up all dice together.

The game lasts 20 turns and the player with the highest number of points is the winner.

Summary:

Project Size: ~450 lines of code

I mainly tackled this game by making a few supplementary functions to take care of smalls tasks, like rolling the dice, adding up the rolls, determining the first players, the sorts and implemented them in my two big functions: the scoring function and the turn function. The smaller functions are self-explanatory, they do simple things like roll or add up rolls or sort, but the bigger functions make use of the smaller ones. For example, the turn function basically runs the turn system of the game, asking the user how they want to score. The score function takes their choices and applies it to switch statement where it chooses the correct scoring category and adds points accordingly. The functions most certainly made testing and editing the function easy, but I still feel like I could have split my two big function up some more, but at the same time I feel like splitting my code into too many functions would have made the code unwieldy and just as difficult to follow, so I feel like I had the right balance.

Pseudo Code:

--Function Prototypes--

Prototypes for starting the game:
Player total function for calculating the total roll of a player
First player function for seeing which player goes first

Essentials for playing the game:
Turn function for playing a turn
Dice roll function

Scoring
Scoring function that returns points

Sorting functions:
Bubble sort
Selection sort

--Function Prototypes End--

Set random number seed

Set precision and decimal point

--Declare Variables--

Array sizing constants:
Dice variable of size 5

Player names:
Names of players
Empty player 1 and 2 strings that get assigned names based on the outcome of the first roll

Boolean variables:
Boolean variable for determining who goes first

Game related variables:
Decision to play the game or not
Game lasts 20 turns, starts at 1
Array for holding the dice
Total score for that game
How many Yahtzees scored in a game for players
Separate variable to differentiate sort funcs
Total points and average points per turn

--End of Variable Declaring--

Initialize some variables that need to be initialized:
Output file
Name of file
Outputs wins and losses of players

Opening files

Introduction to game

Input decision to play or not
Input validation regarded decision

Exit if no, continue if yes

Use cin.ignore to skip to next line

Input user names

Roll to determine who goes first

Determine which person is player 1

Turn do-while loop

Determining the winner and then outputting to a file

Close output file

Exit main

--First Player Function--
Declare some variables for totals
1st person's roll
2nd person's roll
Return boolean
--End of First Player Function--

--Player Total Function--
Declare some variables for totals
Roll the dice
Add the dice up
Return the total
--End of the Player Total Function--

--Dice Roll Function--
For loop to cycle through the 1D array
Input random number from 1-6
Another For Loop to Cycle through the 1D array
Output the array's contents
--End of Dice Roll Function--

--Turn Function--
Declare and initialize some variables

Output categories for scoring
Prompt user to choose
Validate input
Use scoring function
Output score and return it
--End of Turn Function--

--Scoring Function--
Declare and initialize some variables
Switch statement for all of the scoring possibilities:
Score 1s: Add up all 1s and add them to your total points.
Score 2s: Add up all 2s and add them to your total points.
Score 3s: Add up all 3s and add them to your total points.
Score 4s: Add up all 4s and add them to your total points.
Score 5s: Add up all 5s and add them to your total points.
Score 6s: Add up all 6s and add them to your total points.
Score 3 of a Kind: Multiply 3 of the same roll together.
Score 4 of a Kind: Multiply 4 of the same roll together.
Score Full House: If 3 dice are of one value and 2 are of another value, gain 25 points.
Score Short Straight: Gain 30 points if you roll 3 dice sequentially.
Score Long Straight: Gain 40 points if you roll 4 dice sequentially.
Score Yahtzee: Gain 100 points if all dice are same value. 50 points for each additional Yahtzee.
Score Chance: Add up all dice together.
Output score and return it
--End of Scoring Function--

--Bubble Sort Function--
Just a bubble sort
--End of Bubble Sort Function--

--Selection Sort Function--
Just a selection sort
--End of Selection Sort Function

(I'll send the proof of it running on my machine through email along with the rest of everything since inserting an image through google docs makes it blurry)