# Project 1

## Yahtzee

CIS 17A
Juan Castellon
717/18

**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: ~630 lines of code

This project was a little different in terms of actually getting to work compared to my attempts in CSC5. For starters, using structures made the project both a lot easier to handle and create, but at the same time made using pointers and dynamically allocating memory a pain. I wasn't sure how to pass dynamically allocated arrays of structures (a mouthful) to a function, but after some tinkering, I found out that it was just a matter of passing in the pointer to the array. I utilized to structures, one for players to keep track of all the things

**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

--Function Prototypes End--

Set random number seed
Set precision and decimal point

--Declare Variables--

Array sizing constants:
Dice variable of size 5

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
Boolean variable for

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,player 2,etc

Turn do-while loop

Determining the winner and then outputting to a file

Close output file

Deleted allocated memory

Exit main

--Winner Function--
Sort the player structures based on the one with the highest amount of points
Output all the players,their placements, and how many points they got
--End of Winner Function--

--First Player Function--
Declare some variables for totals
Roll for all players
Add up their roles to their respective index variables within the structure
Sort the player structures based on the one with the highest amount of points
--End of First Player Function--

--Turn Function--
Declare and initialize some variables
Ask the user if they want to reroll
If they answer yes, while loop until conditions are not met
Output categories for scoring
Prompt user to choose
Validate input
Use scoring function
Output score and return it
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--

--File Open Function--
Tests file to see if it opens okay
Returns false if it does
Returns true if it does not
--End of File Open Function--

Proof of running on my machine