**Project 2**

**<Word Jumble Game>**

**CIS-5 40137**

**Name: Samuel Hernandez**

**Date: 02/10/19**

**Introduction**

Title: Word Jumble

This is a word guessing game.

You are given a word based on the difficulty you choose, (easy, medium or hard), and you have to guess what it is.

The letters of the word will be randomly scrambled around.

If needed, you can request a hint on what the word is.

The game counts each turn it takes you to guess the word, (counting hints as a turn), and ends when the player correctly guesses the secret word.

The words become increasingly more complicated, both in spelling and definition, depending on the difficulty the player has chosen.

**Summary**

Project size: 305 lines of code

The number of variables: 17

The number of methods: 6

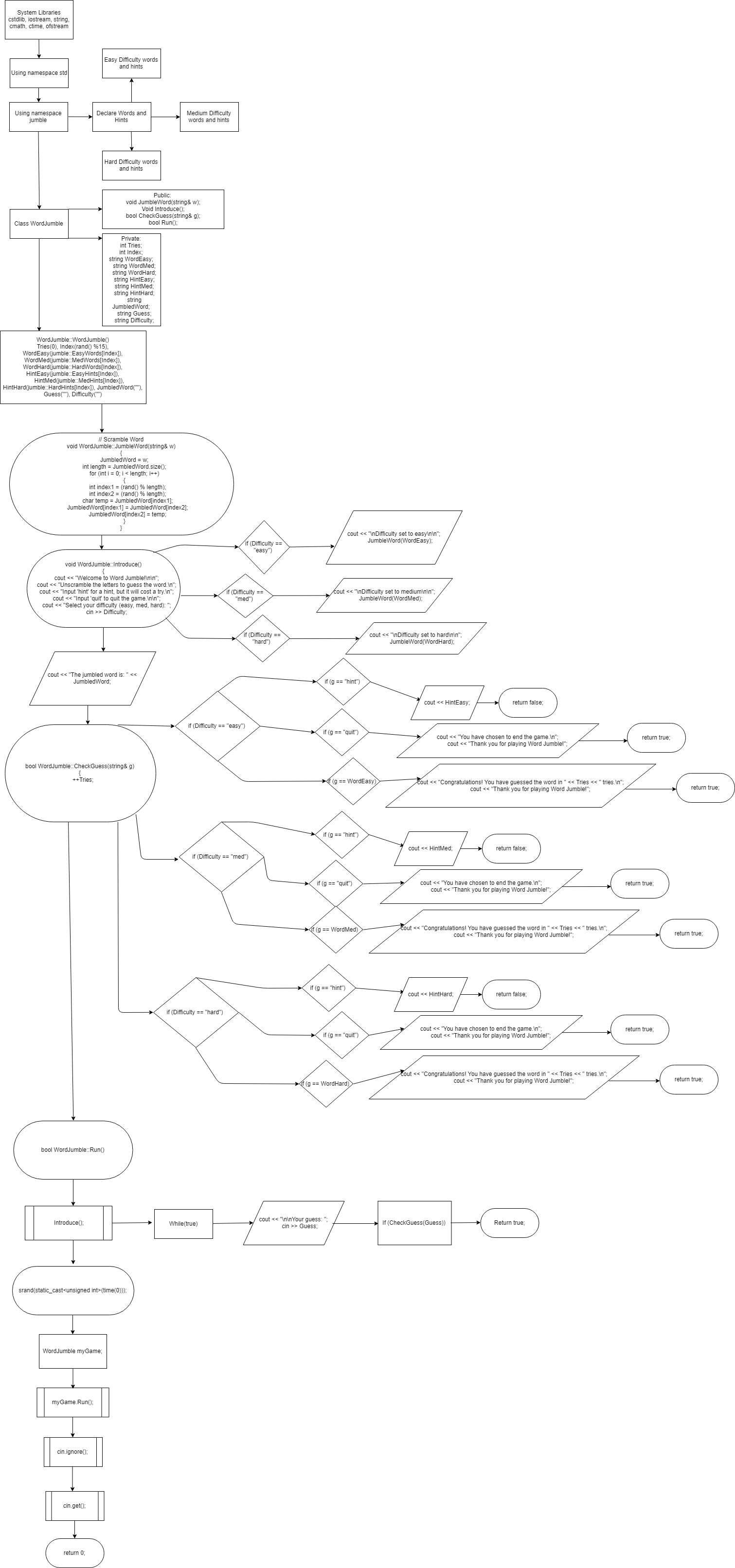
I used many of the methods we have learned throughout this course inside of this program, along with a few others we have not covered very much, such as classes. I felt the need to use concepts such as these to make the game a bit more complex and work a lot smoother than it otherwise would without the help of object-oriented programming methods. To do this I had to rely a lot on stack overflow for help, along with a few videos whenever I got really stuck on either trying to get the difficulty working right, or just an error code I could not figure out.

There are 3 versions of the game, each one with new stuff added. From the first to the second version I added the ability for the player to request a hint if they cannot figure out the word. This meant added a specific hint for each individual word. From the second to final versions of the game I added a total of 15 words to each difficulty, resulting in the game having now 45 different words to play from, each with a hint.

**Description**

The main point of how this program is run is based on first what choice of difficulty is chosen, then will randomly choose a word from that list. The program continues to keep track of each turn the player takes.

**FlowChart**



**Pseudo Code**

*Int Main()*

srand(static\_cast<unsigned int>(time(0))) *// Generate Random number*

*Set MyGame*

*MyGame.Run()*

*Introduce();*

*Tell the player rules*

*Give Jumbled word*

*While (true):*

*Take player guess*

*If correct guess*

*Return true;*

*If wrong guess*

*Return false;*

*If Hint*

*Output hint*

*If Quit*

*End Game*

*Cin.ignore() // Ignore inputs after game has run*

*Cin.get()*

*Return 0;*

**Reference**

* 1. Textbook
  2. StackOverflow.com
  3. YouTube.com

**Program**

/\*

\* Word Jumble Game

\*/

#include <cstdlib>

#include <iostream>

#include <string>

#include <cmath>

#include <ctime>

using namespace std;

namespace jumble

{

// List of Easy Difficulty Words

const string EasyWords[15] =

{

"programming", "computer", "software", "hardware",

"semicolon", "coffee", "yogurt", "bread", "music",

"balloon", "mammal", "water", "correct", "snowflake",

"number"

};

// Medium Difficulty Words

const string MedWords[15] =

{

"chocolate", "asthma", "island", "receipt", "february",

"knead", "colonel", "pronunciation", "recommend",

"deductible", "division", "octagon", "grapefruit",

"nauseous", "gallon"

};

// Hard Difficulty Words

const string HardWords[15] =

{

"hydrogen", "electrolytes","hemoglobin","thermodynamics",

"extraterrestrial","conscience","handkerchief", "rhythm",

"pharaoh", "illusion", "chemist", "analytical", "extract",

"absorption", "bazaar"

};

// Easy Hints

const string EasyHints[15] =

{

"This is what software engineers do.",

"This is made of a bunch of components in a case.",

"This is a technical term for a program.",

"These are the components that make up your computer.",

"A type of punctuation mark.",

"A caffeinated drink, popular in the morning.",

"Bacterial fermentation of milk.",

"Baked flour and water.",

"Art form made from sound, rhythm, and pitch.",

"Flexible bag inflated with gas, such as helium.",

"Animals constituting the class Mammalia.",

"Transparent, tasteless, odorless, nearly colorless substance.",

"Free from error.",

"A flake of snow.",

"A mathematical object used to count, measure, and also label."

};

// Medium Hints

const string MedHints[15] =

{

"Roasted and ground cacao seeds.",

"Condition which a person's airways become inflamed.",

"Sub-continental land that is surrounded by water.",

"Action of receiving something or the fact of its being received.",

"Second month of the year in the Gregorian calendar.",

"Work into dough or paste with the hands.",

"Army officer of high rank.",

"The way in which a word in pronounced.",

"Put forwards with approval as being suitable for a particular purpose or role.",

"Able to be deducted.",

"Action of separating something into parts.",

"Eight-sided polygon.",

"Relatively large sour to semi-sweet, somewhat bitter fruit.",

"Affected with nausea.",

"Equal to 3.79 liters."

};

// Hard Hints

const string HardHints[15] =

{

"Atomic number 1.",

"Produces electrically conducting solution when dissolved in a polar solvent.",

"Iron-Containing oxygen-transport metalloprotein in red blood cells.",

"Branch of Physics that has to do with heat.",

"Of or from outside of earth or its atmosphere.",

"Inner feeling or voice viewed as acting guide.",

"Square of thin fabric or paper which can be carried in the pocket or handbag.",

"Repeated pattern of movement or sound.",

"Monarchs of ancient Egypt.",

"Distortion of the senses.",

"An expert in chemistry.",

"Relating to our using analysis or logical reasoning.",

"Remove or take out, especially by effort or force.",

"Process or action by which one thing absorbs or is absorbed by another.",

"Market in a Middle Eastern country."

};

}

class WordJumble

{

public:

WordJumble();

void JumbleWord(string& w);

void Introduce();

bool CheckGuess(string& g);

bool Run();

private:

int Tries;

int Index;

string WordEasy;

string WordMed;

string WordHard;

string HintEasy;

string HintMed;

string HintHard;

string JumbledWord;

string Guess;

string Difficulty;

};

WordJumble::WordJumble()

: Tries(0), Index(rand() %15), WordEasy(jumble::EasyWords[Index]), WordMed(jumble::MedWords[Index]), WordHard(jumble::HardWords[Index]), HintEasy(jumble::EasyHints[Index]),

HintMed(jumble::MedHints[Index]), HintHard(jumble::HardHints[Index]), JumbledWord(""), Guess(""), Difficulty("")

{}

// Scramble Word

void WordJumble::JumbleWord(string& w)

{

JumbledWord = w;

int length = JumbledWord.size();

for (int i = 0; i < length; i++)

{

int index1 = (rand() % length);

int index2 = (rand() % length);

char temp = JumbledWord[index1];

JumbledWord[index1] = JumbledWord[index2];

JumbledWord[index2] = temp;

}

}

// Introduce Game

void WordJumble::Introduce()

{

cout << "Welcome to Word Jumble!\n\n";

cout << "Unscramble the letters to guess the word.\n";

cout << "Input 'hint' for a hint, but it will cost a try.\n";

cout << "Input 'quit' to quit the game.\n\n";

cout << "Select your difficulty (easy, med, hard): ";

cin >> Difficulty;

if (Difficulty == "easy")

{

cout << "\nDifficulty set to easy\n\n";

JumbleWord(WordEasy);

}

else if (Difficulty == "med")

{

cout << "\nDifficulty set to medium\n\n";

JumbleWord(WordMed);

}

else if (Difficulty == "hard")

{

cout << "\nDifficulty set to hard\n\n";

JumbleWord(WordHard);

}

cout << "The jumbled word is: " << JumbledWord;

}

// Verify if Guess is correct

bool WordJumble::CheckGuess(string& g)

{

++Tries;

// Easy Difficulty

if (Difficulty == "easy"){

if (g == "hint")

{

cout << HintEasy;

return false;

}

else if (g == "quit")

{

cout << "You have chosen to end the game.\n";

cout << "Thank you for playing Word Jumble!";

return true;

}

else if (g == WordEasy)

{

cout << "Congratulations! You have guessed the word in " << Tries << " tries.\n";

cout << "Thank you for playing Word Jumble!";

return true;

}

else

{

cout << "Sorry, that is the incorrect guess.";

return false;

}

}

// Medium Difficulty

else if (Difficulty == "med")

{

if (g == "hint")

{

cout << HintMed;

return false;

}

else if (g == "quit")

{

cout << "You have chosen to end the game.\n";

cout << "Thank you for playing Word Jumble!";

return true;

}

else if (g == WordMed)

{

cout << "Congratulations! You have guessed the word in " << Tries << " tries.\n";

cout << "Thank you for playing Word Jumble!";

return true;

}

else

{

cout << "Sorry, that is the incorrect guess.";

return false;

}

}

// Hard Difficulty

else if (Difficulty == "hard")

{

if (g == "hint")

{

cout << HintHard;

return false;

}

else if (g == "quit")

{

cout << "You have chosen to end the game.\n";

cout << "Thank you for playing Word Jumble!";

return true;

}

else if (g == WordHard)

{

cout << "Congratulations! You have guessed the word in " << Tries << " tries.\n";

cout << "Thank you for playing Word Jumble!";

return true;

}

else

{

cout << "Sorry, that is the incorrect guess.";

return false;

}

}

}

bool WordJumble::Run()

{

Introduce();

while(true)

{

cout << "\n\nYour guess: ";

cin >> Guess;

if (CheckGuess(Guess))

{

return true;

}

}

}

int main(int argc, char\*\* argv)

{

srand(static\_cast<unsigned int>(time(0)));

WordJumble myGame;

myGame.Run();

cin.ignore();

cin.get();

return 0;

}