CIS 227 Assignment 5

Assignment Details

Modify the Assignment 4 to include Set and/or Multiset containers

Provide an option for the user to enter their player name and use it in player output

Add a point system for your hangman game. Points per letter, points for a win.

Team Roles

Lead Programmer – Christian McQueen

UX/UI Programmer – Fernando Alvarez

Functional Programmer – Robert Duffing

Program – 70

UX/UI – 35

Function - 35

Documentation – 30

Total Possible Points – 100

**Version 0.0.0**

| REVISION HISTORY | | | |
| --- | --- | --- | --- |
| DATE | VERSION | DESCRIPTION | AUTHOR |
| 02-10-2022 | 1.0.0 | Select a word from the array and put it into a container. Give the user the option to select words from the file, or the set you created last week. Read the word list from an external file by creating a separate read file class. Prompt the user to guess the letters in the word. Provide feedback if the user is correct or not. Finally, exit the program upon user request. | Christian McQueen  Fernando Alvarez  Robert Duffing |
| 02-16-2022 | 1.1.0 | Modify the Assignment 3/Week 4 Assignment to include a Deque, and/or a List. Limit the number of attempts to guess the word. Provide additional feedback to the user, such as guessed incorrect letters, you win/you lose messages, etc. Give the user a hint upon request. | Christian McQueen  Fernando Alvarez  Robert Duffing |
| 02-19-2022 | 1.2.0 | Modify Assignment 4 to include Set and/or Multiset containers. Provide an option for the user to enter their player name and use it in player output. Adda point system to your hangman game. Points per letter, points for a win. | Christian McQueen  Fernando Alvarez  Robert Duffing |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# INTRODUCTION

## PURPOSE

Identify and describe scope of product whose technical specifications are being documented and describe desired outcome.

The scope of this product is to provide the user with a hangman game that allows a saved username and a points system to monitor progression.

## DOCUMENT CONVENTIONS

Describe any naming or structural conventions employed throughout document and how they benefit reader.

The names of variables and structures are in place to closely reflect what they do. For instance, the map siMapPlayers is used to log usernames (once it gets to that point).

## REFERENCES

List any referenced document names or links.

None.

# DESCRIPTION

## FEATURES

List main features with brief description.

Main() function – Calls the other functions and provides the direction of the program. Calls for the user to input yes or no for exiting the program.

mainProgram() function – The heavy lifting function that contains the game. Takes user input for guesses and provides direction based on correct/incorrect input.

## USER OVERVIEW

Define groups and describe user characteristics.

Groups: The programmer and the user or player(s).

User characteristics: Use of Visual Studio/Visual Studio Code or the Command Line to access the game via files. No math is required.

## ASSUMPTIONS / DEPENDENCIES

Detail all assumed factors (not known facts) that could potentially impact technical specifications set forth. Include external factors.

There is no data validation for input outside of letters contained in the word. The user could input any character and use up a guess as a result, even if this was not the intent. External factors could include the player’s computer not successfully running the program given multiple avenues of access.

# SYSTEM FEATURES

## SYSTEM FEATURE 1

|  |  |
| --- | --- |
| **DESCRIPTION AND PRIORITY** | Main() function – The main function is used to call the mainProgram() function that contains the game. Main() is used to provide general beginning/end inquiries to start and end the program given user input. I had said before that this function is the first priority, but in actuality it is second because the game function is the make or break of the entire program. |
| **STIMULUS / RESPONSE SEQUENCES** | The map is stored in main to track the usernames of player(s) participating in the game. Here it prompts hello and asks for a username. Input is gathered and then mainProgram() is called to run the game. Once the end of this function is reached, main() asks if the user would like to restart or end the program given user input. |
| **FUNCTIONAL REQUIREMENTS** | Requires mainProgram() to run the game. Provides start/end statements and collects necessary input from the user. |

## SYSTEM FEATURE 2

|  |  |
| --- | --- |
| **DESCRIPTION AND PRIORITY** | mainProgram() function – The mainProgram() function is the function containing the game. This function is the highest priority as without it, there is no game. |
| **STIMULUS / RESPONSE SEQUENCES** | The function starts by reading in a list of words from ‘myWordList.txt’ that contain the pre-populated words for the game. There are numerous variables declared at the top of the function to aid in storing user input and other sequences. User input is gathered for which word the player would like to take a guess at. If the user is correct, they are given one point and the letter replaces the asterisk-covered letter in the word. This is repeated unless there is an incorrect guess and until the player wins. The player starts with three guesses, with each incorrect guess decrementing the guesses variable. Once it hits zero, game over. |
| **FUNCTIONAL REQUIREMENTS** | Need user input for which word the player wants to guess at, and guesses. Multiple if/else statements are used to here to direct the player through their input choices. |

## SYSTEM FEATURE 3

|  |  |
| --- | --- |
| **DESCRIPTION AND PRIORITY** | N/A |
| **STIMULUS / RESPONSE SEQUENCES** | N/A |
| **FUNCTIONAL REQUIREMENTS** | N/A |

# REQUIREMENTS OF EXTERNAL INTERFACE

## USER INTERFACES

Describe product / user interface characteristics, including standards, style guides, constraints, functionality, and sample screens if applicable.

Product/UI characteristics include prompts saying hello, asking for a username, repeating the username, asking for which word the player wants to guess at, and guess input for the game. Prompts are used depending on the condition of the guess (true/false), and necessary prompts are issued like, “You win!” or “You lose!”. User input is needed finally to either restart or end the program/game.

# APPENDICES

## APPENDIX A: GLOSSARY OF TERMS

Define all terms and unique acronyms employed throughout document and specific to project.

Map<string, int> siMapPlayers – Used to store usernames of players playing the game.

Map<string, int>::iterator siIterator – Iterator for the siMapPlayers map.

String username – Variable for storing the username input which gets stored in the map.

Int points – Points variable that increments by one every correct guess.

## APPENDIX B: ANALYSIS DOCUMENTATION

List file / document names / provided links to all diagrams, models, additional findings pertinent to technical specification development.

Assignment5.cpp

Assignment5Header.h

myWordList.txt

## APPENDIX C: ISSUES

List all unresolved issues, TBDs, pending decisions, findings required, conflicts, etc.

| ISSUES | | |
| --- | --- | --- |
| ID | DESCRIPTION | PARTY RESPONSIBLE |
| 1 | I am still unable to determine a good method of implementing the hint feature starting after the first incorrect guess. | Christian McQueen |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |