1/12/2015

Samuel Williamson

Bournemouth University

Hangman report

Programming Assignment 1

Contents

[Analysis 2](#_Toc408824911)

[Requirement brief 2](#_Toc408824912)

[Design 3](#_Toc408824913)

[User Input/Outputs 3](#_Toc408824914)

[Functions 4](#_Toc408824915)

[Requirement analysis 4](#_Toc408824916)

[Conclusion 5](#_Toc408824917)

# Analysis

## Requirement brief

In this project I was asked to create a game of hangman with several limitations. Limitations included: The code has to be in C++, it must only use Cstrings and can only be structured code. A full list is below.

|  |  |  |
| --- | --- | --- |
| Requirement No. | Requirement | Justification |
| 1. | Must be C++ | On the requirement spec |
| 2. | Must be structured code | On the requirement spec |
| 3. | Must display the gallows | On the requirement spec |
| 4. | The application is required to allow one player to play “Hang Man”. | On the requirement spec |
| 5. | The computer will act as executioner. | On the requirement spec |
| 6. | It should calculate all angles when only given all lengths, or when given one side and two angles. | On the requirement spec |
| 7. | Display the Word /Phrase: Display the correctly guessed letters in their correct positions and also indicate the remaining unknown letters positions. | On the requirement spec |
| 8. | The number of player’s guesses remaining | On the requirement spec |
| 9. | The user should also be able to exit the program from the interface. | On the requirement spec |
| 10. | Theme based hangman. | I will add to make the game better |
| 11. | The man hangs after six incorrect guesses. | On the requirement spec |
| 12. | The software is only to **STRUCTURED** | On the requirement spec |
| 13. | All strings must be **CStrings** | On the requirement spec |

Above, I created a top down view on my project. This not only helps organise the program, but also helps explain the functioning of it.

# Design

## User Input/Outputs

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| User selects theme | The program chooses from the appropriate list. | Word selected from specified theme |
| Users guesses a letter | System checks to see if letter is in the word | True/False |

I created a dataflow diagram in order to help plan out my program. It is very detailed as I used sudo code to create the initial ideas of the program. This flow diagram can be found [here.](code2flow_e2754.pdf)

## Functions

I had several functions in my code in order to make the code more legible and function well. It also allowed me to optimise my code in the way I had less repeated code.

|  |  |  |
| --- | --- | --- |
| ID | Name | Purpose |
| 1. | MainMenu | Is the main menu for the program which the end user will operate. |
| 2. | Category | Allows the end user to select a theme. |
| 3. | Difficulty | Allows for a difficulty to be selected |
| 4. | Instructions | To aid the end user in knowing how to play hangman |
| 5. | Quit | Give the end user the option to quit the game |
| 6. | LoadingScreen | A loading screen to keep the end user distracted while game is being prepared. |

## Requirement analysis

|  |  |  |
| --- | --- | --- |
| Requirement No. | Requirement | Was the requirement met? |
| 1. | Must be C++ | Yes |
| 2. | Must be structured code | Yes |
| 3. | Must display the gallows | Yes |
| 4. | The application is required to allow one player to play “Hang Man”. | Yes |
| 5. | The computer will act as executioner. | Yes |
| 6. | It should calculate all angles when only given all lengths, or when given one side and two angles. | Yes |
| 7. | Display the Word /Phrase: Display the correctly guessed letters in their correct positions and also indicate the remaining unknown letters positions. | Yes |
| 8. | The number of player’s guesses remaining | Yes |
| 9. | The user should also be able to exit the program from the interface. | Yes |
| 10. | Theme based hangman. | Yes |
| 11. | The man hangs after six incorrect guesses. | Yes |
| 12. | The software is only to **STRUCTURED** | Yes |
| 13. | All strings must be **CStrings** | Yes |

# Conclusion

In conclusion, I believe this project went quite well. I included everything I wanted apart from the difficulty function, as I was taking too long to understand how to solve the problem. Next time I shall leave more time in order to get more done. However, the program is fully functioning which means it has gone better than my previous coding assignment.