**Project 2**

**HANGMAN**

**Author:**

**Dennise Toscanini**

**CSC-5**

**Introduction**

**HANGMAN**

Hangman is a guessing game. The computer will read in a set amount of words from a file then randomly choose one to display. The user then guesses letters one by one, if a letter guessed is correct you continue to guess, if the letter guessed in incorrect a strike is incurred and counted. The user is able to guess incorrectly a maximum of six times. Once six incorrect guesses have been made, the game is lost. If you happen to guess the word while not incurring six wrong guesses you win!

**Summary**

**Project size:**  288 lines of code

**Number of Libraries:** 5

**Number of Variables:** 19

**Comment Lines:** 46

**Functions:** 2

**One Dimensional Array:** 1

**Two Dimensional Arrays:** 6

**Variable Summary**

**int** variables: 3

**const int** variables: 1

**char** variables: 10

**bool** variables: 2

**ifstream** variables: 1

**string:** 2

**Variable List**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Name** | **Description** | **Declared on Line Number** |
| int | number | Used to randomly pick word | 33 |
| int | num\_try | Counter for number of wrong guesses | 33 |
| int | count | Counter for words in text file | 34 |
| const int | maxtry | User gets only six wrong guesses | 55 |
| char | again | Used to allow user to play game numerous times | 26 |
| char | blank | Shows user how many letter are in a word by outputting a ‘\_’ | 53 |
| char | space | Puts a space in between each blank | 53 |
| char | guess | User’s guess | 53 |
| bool | gotlett | Declares how many characters are in theWord | 58 |
| bool | found | Set to always be false unless user guesses a correct letter | 58 |
| Ifstream | in\_stream | Allows program to read in from file | 30 |
| string | theWord(words[number] | Array of characters in word to be guessed | 52 |
| string array | words[6] | Array of words to be guessed | 32 |
| char array | guy1[3][3] | Two dimensional array that will output hangman head | 141 |
| char array | guy2[3][3] | Two dimensional array that will output hangman head and body | 145 |
| char array | guy3[3][3] | Two dimensional array that will output hangman head, body, and right arm | 149 |
| char array | guy4[3][3] | Two dimensional array that will output hangman head, body, right arm, and right leg | 153 |
| char array | guy5[3][3] | Two dimensional array that will output hangman head, body, right arm, right leg, and left arm | 157 |
| char array | guy6[3][3] | Two dimensional array that will output hangman head, body, right arm, right leg, left arm, and left leg | 161 |

**Function List**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Return Type** | **Line Number** |
| Gueswrd | Returns false if all letters have not been guessed  Returns true if all letters have been guessed | bool | 129 |
| Hangman | Contains a switch statement of hangman character arrays to be output when user guesses a wrong letter | void | 139 |

**Pseudo code**

DO{

Open text document “hangman.txt”

IF{

“hangman.txt” fails

EXIT

}

Clear text marker back to beginning of document

WHILE{

count number of lines in text document

}

FOR{

the length of word randomly chosen from file

Output blank spaces

}

DO{

Prompt user to guess a letter

FOR{

Correctly guessed letter return true

}

IF{

Returns false

Increment number of incorrect guesses

}

FOR {

Correctly guessed letter replace blank space with correct letter

}

ELSE{

Output all letters that have not been correctly guessed with blank spaces

}

CALL FUNCTION hangman to output proper case from switch statement to show hangman’s body

}WHILE

The user has not guessed the entire word and user has not exceeded maximum number of tries

IF{

User guesses all of the letters in the word correctly user WINS!

}

ELSE{

User has not guessed all the letters correctly and has exceeded maximum number of tries the user WINS!

}WHILE

User declares whether they would like to play again or not