Project 1

<Hangman Game>

Version 1

Date: 7/19/15

Due Date: 7/21/15

Course Code: 46090

Professor's Name: Dr. Mark Lehr

Programmer/Author: Abdul Hakim Abbas

Course Name: CSC5 Introduction to programming in C++

Index:

Introduction:

Title: Hangman Game

The version I have made has no draw up of the hanging man, and all other objects that are included in the known version of it.

This game is played by two people.

One player knows the secret word.

Also he is the one whom does the fill in of the guessed characters of the second player into the empty spaces denoted by underscores

And they win when the second player ends up using their maximum possible characters.

And the second player wins when they guess the word correctly.

And in the game the amount of guesses are at least the equal to the size of the secret word that has to be guessed.

This version of the game does not have display the hanging man.

It displays the amount of incorrect guesses, and it shows the letters that have been input correctly.

And there is possibly many other versions of this game.

Summary:

Project Size: 256 Lines

The System Libraries that are used:

1. iostream
2. cstdlib
3. ctime

4. string

The Global Variable that has been used for this program is MAX_TRIES.

The Functions that have been used are:

1. Letter Fill

2. Main

This game uses an array to hold the secret words, it has around 145 words which are hard set into an array.

I tried to use the concepts that we have learned so far in this Project to make a functional hang man game in which the version I have programed was the one that does not display a hanging stick figure man.

I have used an array to hold the words, and there are many other possible ways to implement that part of the program.\

This program may have missed using some of the seven constructs that should be found in it.

And this code works well, and I don't know what may occur due the modification.

I hope to implement more than that in the next project which would be the modification of this program in such a way that it will try to hold all the needed constructs and any other improvement.

But I think it is best to have the deep modifications on the next version of this project.

It took me a couple of days to create this program.

It definitely was not that hard to code up this game.

And I have seen harder things to code up such as the Midterm's Problem #1.

And that is partly due to that I have learned many programming languages they are: C#, JavaScript, CSS, HTML5, PHP, Python, Ruby, J Query, Java, SQL, also I learned how to use certain API’s like the Mandrill API, and there might be more that I did not remember to mention here.

And all of these I have learned form different places, and at different times.

And some of them I still am learning, also some of them I have completed, and others I have stopped using.

But they have diffidently improved my understanding, and overall it will result in me becoming a more well rounded programmer, and individual.

And now thanks to this course I can now add C++ to my list of programming languages that I know about, and that I hope will expand my understanding of the needed skills for my major in Computer Science.

I have used both Gaddis and Savitch books, and I have searched the web for tips on the certain issues that I faced while coding this Project.

I am not sure if there is a better way to implement code for this Project.

But this is was surely was a beneficial process in that it gave me more experience with what I may face in my future career path, and to what I could possibly face in the progression through my higher studies at College and University level of education.

I hope to learn what I needed to succeed, also I want to learn as much as possible.

So that I can know how to face certain issues correctly.

Description:

The main point of this program is that it is a game in which there are two players in which one will guess a secret word know to the other player.

This program uses an array to hold the secret words which the computer will choose from that list to be guessed by the user.

It has those words fully written in lower case, so the user must enter lower case letters to be able to play and they get to keep on inputting characters until they reach the maximum limit of possible entries.

In which they may end up guessing the secret word which would mean they have won the game, and then the program displays a message which congratulates them for winning the game.

Or they reach the maximum amount of entries and they do not end up guessing the secret word, which will result in that the program will display a message saying You have been hanged.

Which means the user has not won the game, and it also displays the secret word that was picked randomly.

Documentation:

File Reference of Main:

```
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <string>
```

Functions:

```
int letterFill (char, string, string &)
int main (int argv, char *argc[])
```

Variables:

```
const int MAX_TRIES =20
```

Program Code:

```
/*
 * File:   main.cpp
 * Author: AbdulHakim Abbas
 * Purpose: Is the First Project which is the Game of Hangman
 * Created on July 15, 2015, 3:48 PM
 */

//System Libraries that is used for this program are found in the hash include library
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <string>
using namespace std;
//User libraries are declared here
//Global Constants are declared here
const int MAX_TRIES=20;
//Function Prototypes are declared here
int letterFill (char, string, string&);
//Execution begins here!
int main(int argv,char *argc[]){
//Declare the variables needed for this program
string name;
char letter;
int num_of_wrong_guesses=0;
string word;
//Make an array that holds the words of the game that are chosen at random
string words[] =
{
"course",
"mode",
"plans",
"process",
"disposal",
"battery",
"midterm",
"office",
"notes",
"laptops",
"machines",
"center",
"butter",
"milk",
"water",
"culture",
"passport",
"passed",
"failed",
"equator",
"paper",
"approach",
"final",
"projects",
"college",
"university",
"code",
"factory",
"method",
"source",
"supply",
"nectar",
"ambiguity",
"model",
"clarity",
"stereotypes",
"marketing",
"comply",
"three",
"vacation",
"information",
"printer",
"application",
"hardware",
"software",
"objects",
"router",
"technology",
"design",
"mobile",
"systems",
```

```
"security",
"networks",
"java",
"manners",
"computer",
"compilers"
"pillars",
"noble",
"society",
"data",
"file",
"complete",
"rules",
"guide",
"lines",
"edition",
"first",
"volume",
"area",
"poetry",
"organs",
"internal",
"external",
"elements",
"elementary",
"education",
"power",
"charts",
"statements",
"principles",
"last",
"green",
"blue",
"cyan",
"red",
"black",
"white",
"gray",
"pink",
"cards",
"paper",
"chalk",
"walls",
"wallet",
"painter",
"boards",
"experience",
"assisting",
"revolting",
"hacking",
"backing",
"engaging",
"forensics",
"general",
"platforms",
"fragmentation"
"minimal",
"cascading",
"style",
"height",
"width",
"ground",
"shortcut",
"parameter",
"disks",
"extract",
"compression",
"yellow",
"extensible",
"sockets",
"array",
"framework",
"header",
"footer",
"animation",
"extract",
"intersect",
"acquisition",
"espionage",
"enterprise",
"remote",
"physical",
"regional",
"abdominal",
"laceration",
"tension",
```

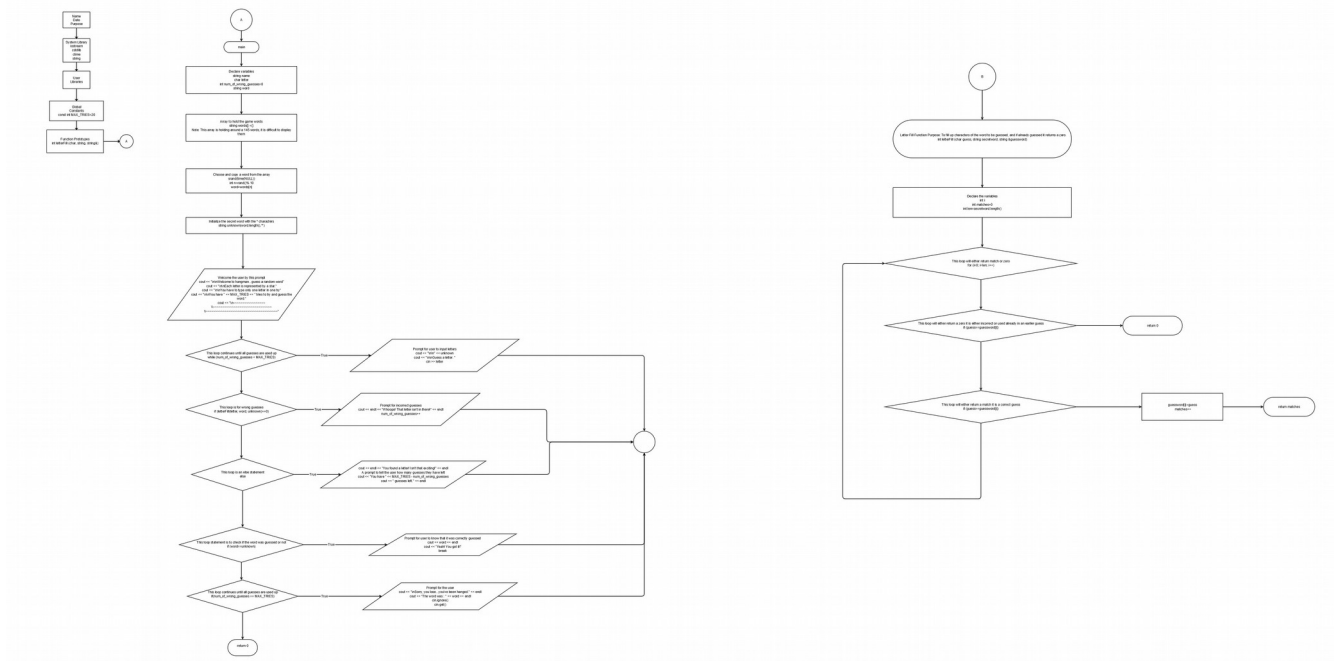
```
"assault",
"ligaments",
"tennis",
"classic",
"modern",
"emergency",
"wilderness",
"hammer"
};

//Choose and copy a word from array of words randomly
srand(time(NULL));
int n=rand()% 10;
word=words[n];

//Initialize the secret word with the * characters
string unknown(word.length(),'*');

//Welcome the user by displaying a prompt

cout << "\n\nWelcome to hangman...guess a random word";
cout << "\n\nEach letter is represented by a star.";
cout << "\n\nYou have to type only one letter in one try";
cout << "\n\nYou have " << MAX_TRIES << " tries to try and guess the word.";
cout << "\n~~~~~~~~~~~~~~~~~\\~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\\~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

//This Loop is done until the all guesses are used up
while (num_of_wrong_guesses < MAX_TRIES)
{
cout << "\n\n" << unknown;
cout << "\n\nGuess a letter: ";
cin >> letter;
//Fill secret word with letter if the guess is correct.
//Otherwise increment the number of wrong guesses.
if (letterFill(letter, word, unknown)==0)
{
cout << endl << "Whoops! That letter isn't in there!" << endl;
num_of_wrong_guesses++;
}
else
{
cout << endl << "You found a letter! Isn't that exciting!" << endl;
}
//Tell user how many guesses has they have left
cout << "You have " << MAX_TRIES - num_of_wrong_guesses;
cout << " guesses left." << endl;
//Check if user has guessed the word
if (word==unknown)
{
cout << word << endl;
cout << "Yeah! You got it!";
break;
}

}
if(num_of_wrong_guesses == MAX_TRIES)
{
cout << "\nSorry, you lose...you've been hanged." << endl;
cout << "The word was : " << word << endl;
}
cin.ignore();
cin.get();
return 0;
}

/*******************************************************************
 * Letter Fill Function                                            *
 *******************************************************************
 * Purpose: To fill up characters of the word to be guessed, and if
 * already guessed it returns a zero.
 */
int letterFill (char guess, string secretword, string &guessword)
{
int i;
int matches=0;
int len=secretword.length();
for (i=0; i<len; i++)
{
// Did we already match this letter in a previous guess?
if (guess==guessword[i])
return 0;

// Is the guess in the secret word?
if (guess==secretword[i])
{
guessword[i]=guess;
```

```
matches++;
}
}
return matches;
}
```

Pseudo Code:

```
File:   main.cpp
Author: AbdulHakim Abbas
Purpose: Is the First Project which is the Game of Hangman
Created on July 15, 2015, 3:48 PM

System Libraries that is used for this program are found in the hash include library.
User libraries are declared here.
Global Constants are declared here.
Function Prototypes are declared here.
Execution begins here.
Declare the variables for this program.
Make an array that holds the words of the game that are chosen at random.

Choose and copy a word from array of words randomly

Initialize the secret word with the * characters.

Welcome the user by displFloaying a prompt.

This Loop is done until the all guesses are used up.

Fill secret word with letter if the guess is correct.
Otherwise increment the number of wrong guesses.

Tell user how many guesses has they have left.


Letter Fill Function
Purpose: To fill up characters of the word to be guessed, and if already guessed it returns a zero.

Did we already match this letter in a previous guess?

Is the guess in the secret word?
```

Flowchart:



Reference:

I used the following as reference materials they are either Websites, Forums, Blogs, Wikipedia, and also I used our books Gaddis and Savitch. The sources I have used were useful to allow me to complete this project quicker than I expected.

1. wikipedia.com
2. edparrish.net
3. eclipse.org
4. google.com
5. stackoverflow.com
6. ocw.mit.edu
7. printactivities.com
8. lifehacker.com
9. chegg.com
10. cs.duke.edu
11. diffcultcoding.wordpress.com
12. dreamincode.net
13. cplusplus.com
14. forums.codeguru.com
15. stackexchange.com
16. mesacc.edu
17. en.wikibooks.com

There might be more sites and forums included in this that I may have missed to put into this list here.

The books I used were our books which are Gaddis and Savitch.

I might have used some other books but to a much lesser extent than our own course books.

The course books were heavily used in this project, in ways to make it function in a better way.