Project 2

<Hangman Game>

Version 2

Date: 7/23/15

Due Date: 7/31/15

Course Code: 46090

Professor's Name: Dr. Mark Lehr

Programmer/Author: Abdul Hakim Abbas

Course Name: CSC5 Introduction to programming in C++

Index:

Index:

Introduction:

Title: Hangman Game

The version I have made has a 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:

This game uses an array to hold the secret words, it has around 20 words which are hard set into an array.

It also has 280 words in a file which can be used if the user picks the related choice, is due to that I used a vector to hold those words.

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 does display a hanging stick figure man.

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

This program may lack the player name, the word which they had to guess in their turn, and so on to be put into a file.

I did not include this as it made errors occur, and it seemed to be too time consuming for we have a limited amount of time.

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

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 in this version of the project.

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

It definitely was harder to code up this version of the game, than the previous version.

And also I know there is definitely harder things to code up and solve than this which are found outside in the real world, which I and many others will face in our future work.

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:

System Libraries:

```
#include <iostream>

#include <cstdlib>

#include <ctime>

#include <string>

#include <fstream>

#include <vector>
```

Functions:

```
int letterFill(char, string, string&)

void printman(unsigned int stage)

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

Variables:

```
unsigned int stage=0;

int MAX_TRY=9,numWrGs=0,choice,choice2,nmWrGs2=0,MX_TRY2=9;

char letter,letter2;

string word;

string pNm;
```

Program Code:

```
/*

 * File:   main.cpp

 * Author: AbdulHakim Abbas

 * Purpose: Project 2

 * Created on July 23, 2015, 7:05 PM

 */

//System Libraries

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <string>

#include <fstream>

#include <vector>

using namespace std;

//User libraries

//Global Constants

//Function Prototypes

int letterFill (char, string, string&);

void printman(unsigned int stage);

//Execution begins here!

int main(int argv,char *argc[]){

//Declare the variables needed for this program

unsigned int stage=0;

int MAX_TRY=9,numWrGs=0,choice,choice2,nmWrGs2=0,MX_TRY2=9;

char letter,letter2;

string word;

string pNm;

//Make an array that holds words
```

```
string words[] =

{

"trash",

"cash",

"rat",

"bat",

"tag",

"tap",

"rap",

"pack",

"belt",

"heat",

"argue",

"faith",

"cat",

"whole",

"rash",

"heap",

"reap",

"king",

"rage",

"target"

};



//Choose and copy a word from the 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~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

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 9 tries to try and guess the word.";

cout << "\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";



//Prompt the user for them to input their name

cout << "\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

cout<<"\n\nDear player enter your name please!";

cin >> pNm;

cout << "\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";



//Prompt user to pick if they want to play or not!

cout<<"\nDo you want to play a game?"<<endl;

cout<<"Press 1 for Yes, and 2 for No"<<endl;

cout<<" "<<endl;

cout<<"Please input your choice here: ";

cin>>choice;




    //If Choice Is Yes, Ask For Easy Or Hard Mode

    if(choice==1){

        cout<<"\n "<<endl;

        cout<<"\nOkay! Now that we know you want to play, what path would you like to pick?";
```

```
cout<<"\nThe First Path or the Second Path? ";
cout<<"\nPlease press 4 for The First Path ";
cout<<"\nand 5 for The Second Path."<<endl;
cout<<" "<<endl;
cout<<"\nPlease enter your choice here: ";
cin>>choice2;

//If First, Start First Mode
if(choice2==4){
    cout<<"You choose the First Path. Get ready";

    //Randomize The Words In The Game
    srand(time(NULL));
    int n=rand()%10;
    word=words[n];

    //Initialize Secret Word With *
    string unknown(word.length(),'*');

    //Loop Until The Guesses Are Used Up
    while(numWrGs<MAX_TRY){
        printman(stage);

        cout<<" "<<endl;
        cout<<" "<<unknown<<endl;
        cout<<"Please choose a letter: ";
        cin>>letter;

         //Fill The Secret Word With A Letter, Otherwise
```

```
    //Increment Number Of Wrong Guesses
    if(letterFill(letter,word,unknown)==0){
        cout<<"You got the ";
        cout<<"wrong letter!"<<endl;
        numWrGs++;
        ++stage;
    }else cout<<"Amazing you found a letter! ";
          cout<<" "<<endl;

    //Tell The User How Many Guesses They Have Left
    cout<<" You have "<<MAX_TRY-numWrGs<<" guesses left."<<endl;

    //Check To See If User Guesses Word
    if(word==unknown){
        cout<<" "<<endl;
        cout<<word<<endl;
        cout<<" "<<endl;
        cout<<"You guessed the right word."<<endl;
        break;
    }


}

//Output If User Loses
if(numWrGs==MAX_TRY){
    printman(stage);
    cout<<" "<<endl;
    cout<<"The word was : "<<word<<"."<<endl;
```

```
 }


//INITIALIZE Second MODE!!!!! You Go Bro!
}else if(choice2==5){

    //Welcome User To Second Mode!!!
    cout<<"I see you've chosen the Second Path. ";
    //Declare Variables
    ifstream myfile;
    vector<string> words2;
    string word2;
    //Open File
    myfile.open("words.dat",ios::in);

    if(!myfile){
        cout<<"Sorry! No file found!"<<endl;
        return 0;
    }

    while(myfile>>word2){
        words2.push_back(word2);
    }

    //Randomize The Words In The Game
    srand(time(NULL));
    int s=rand()%10;
    word2=words2[s];

    //Close The File
```

```
myfile.close();


//Initialize Secret Word With *
string unknwn2(word2.length(),'*');


//Loop Until The Guesses Are Used Up
while(nmWrGs2<MX_TRY2){
    printman(stage);


    cout<<" "<<endl;
    cout<<" "<<unknwn2<<endl;
    cout<<"Please choose a letter: ";
    cin>>letter2;


 //Fill The Secret Word With A Letter, Otherwise
 //Increment Number Of Wrong Guesses
 if(letterFill(letter2,word2,unknwn2)==0){
     cout<<"Wrong letter! Try again!"<<endl;
     nmWrGs2++;
     ++stage;
 }else cout<<"You found a letter! I knew you could do this!";
    cout<<" "<<endl;


 //Tell The User How Many Guesses They Have Left
 cout<<" You have "<<MX_TRY2-nmWrGs2<<" guesses left."<<endl;


 //Check To See If User Guesses Word
 if(word2==unknwn2){
     cout<<" "<<endl;
```

```
                cout<<word2<<endl;
                cout<<" "<<endl;
                cout<<"How astounding! Amazing! ";
                break;
            }
        }

        //Output If User Loses
        if(nmWrGs2==MX_TRY2){
            printman(stage);
            cout<<" "<<endl;
            cout<<"It seems you didn't have what it takes to win."<<endl;
            cout<<" "<<endl;
            cout<<"The word was : "<<word2<<"."<<endl;
            cout<<" "<<endl;
        }

    }

}

//If Choice Is No, End The Game
else if(choice==2){
    cout<<"Fine, don't play this game!"<<endl;
    return 0;
}

//If User Picks A Letter, End The Game
else {
```

```
        cout<<"You choose the wrong input! This means I must exit! ";
    }

    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;

}


/*****************************************************************************************

 * Print Man Function                                                                    *

 *****************************************************************************************

 * Purpose: To use stages to print out the stick figure man due to the wrong guesses.

 */


void printman(unsigned int stage){


        // |/---

        // |  o

        // | /|\

        // | / \

        // |____


        switch(stage)

        {

             case 0:

                cout << "\n\n\n\n\n";

                break;

            case 1:

                cout << "\n\n\n\n_____";

                break;

            case 2:

                cout << "|\n|\n|\n|\n|____";

                break;
```

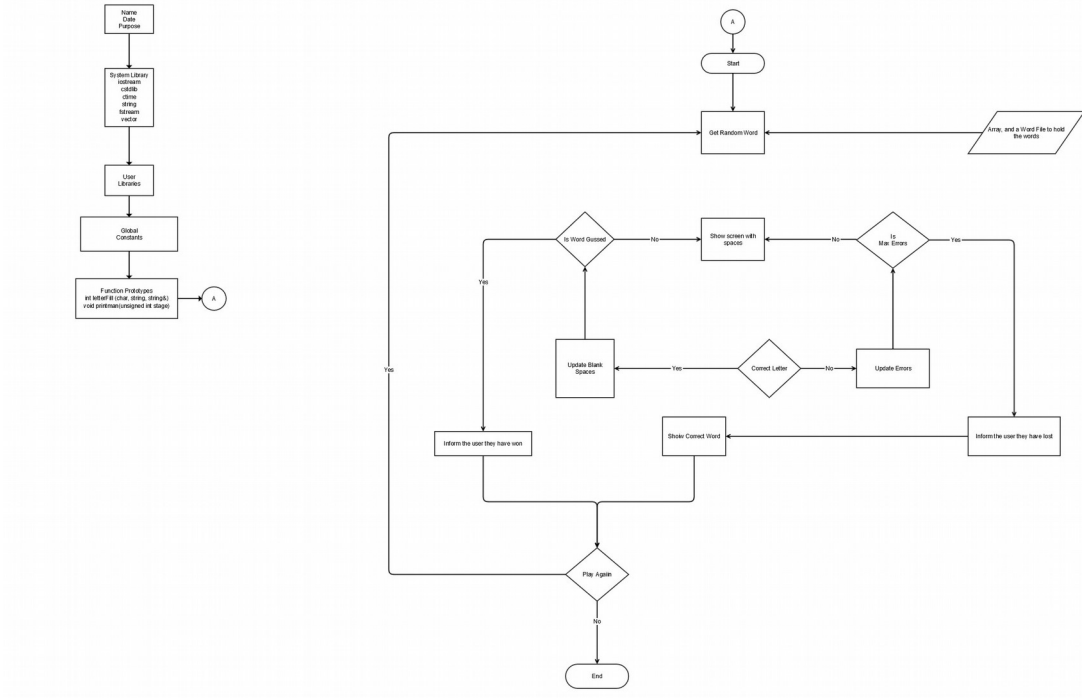
```
            case 3:
                cout << "|/---\n|\n|\n|\n|____";
                break;
            case 4:
                cout << "|/---\n|  o\n|\n|\n|____";
                break;
            case 5:
                cout << "|/---\n|  o\n|  |\n|\n|____";
                break;
            case 6:
                cout << "|/---\n|  o\n| /|\n|\n|____";
                break;
            case 7:
                cout << "|/---\n|  o\n| /|\\\n|\n|____";
                break;
            case 8:
                cout << "|/---\n|  o\n| /|\\\n| /\n|____";
                break;
            case 9:
                cout << "|/---\n|  o\n| /|\\\n| / \\\n|____";
                break;
            default:
               cout << "|/---\n|  o\n| /|\\\n| / \\\n|____";
               break;
        }
    return;
}
```

Pseudo Code:

```
/* File:   main.cpp

* Author: AbdulHakim Abbas

* Purpose: Project 2

* Created on July 23, 2015, 7:05 PM

*/

System Libraries

User libraries

Global Constants

Function Prototypes

Execution begins here!

Declare the variables needed for this program

Make an array that holds words

Choose and copy a word from the array of words randomly

Welcome the user by displaying a prompt

Prompt the user for them to input their name

Prompt user to pick if they want to play or not!

If Choice Is Yes, Ask For First Or Second Mode

If First, Start First Mode

 Randomize The Words In The Game

 Initialize Secret Word With *

 Loop Until The Guesses Are Used Up

  Fill The Secret Word With A Letter, Otherwise

  Increment Number Of Wrong Guesses

 Tell The User How Many Guesses They Have Left

 Check To See If User Guesses Word

 Output If User Loses

INITIALIZE Second MODE!!!!! You Go Bro!

 Welcome User To Second Mode!!!
```

```
 Randomize The Words In The Game

 Close The File

 Initialize Secret Word With *

  Loop Until The Guesses Are Used Up

  Fill The Secret Word With A Letter, Otherwise

  Increment Number Of Wrong Guesses

 Tell The User How Many Guesses They Have Left

 Check To See If User Guesses Word

 Output If User Loses

If Choice Is No, End The Game

If User Picks A Letter, End The Game

/*****************************************************************

 * 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?

/*************************************************************************************

 * Print Man Function                                                                *

 *************************************************************************************

 * Purpose: To use stages to print out the stick figure man due to the wrong guesses.

 */
```

Flowchart:


Name
Date
Purpose
System Library
iostream
cstdlib
ctime
string
fstream
vector
User
Libraries
Global
Constants
Function Prototypes
int letterFill (char, string, string&)
void printman(unsigned int stage)
A
A
Start
Get Random Word
Array, and a Word File to hold the words
Is Word Gussed
No
Show screen with spaces
No
Is Max Errors
Yes
Yes
Update Blank Spaces
Yes
Correct Letter
No
Update Errors
Yes
Inform the user they have won
Show Correct Word
Inform the user they have lost
Play Again
No
End

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
18. github.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.