Hangman Game

Angeline Indahsi

2001585335

**Section 1**

**Description of Project**

Programming is one of the important things in the modern world. With programming, we can make a program that can make people’s life easier and develop something new for other people. We can also make something that can entertain people with programming, such as games. As a beginner in programming, I learned how to do the basics of programming first in order to be able to make something new and useful in the future. And the way to do that is to try and apply the things that I had learned during class by doing the final project. For my final project, the program that I made as my final project is a basic hangman game, in which the players try to guess the word by inputting each letter. The players were given six chances before they lose the game, in which the remaining chances will be told using a text. The theme for the game is the name of animals, in which the players need to guess the name of an animal.

The way the program works is the same as the usual hangman game. When the program starts, there is an instruction for the players to type any key, such as an ‘a’ and then press enter to start the game. Then the game will start with an underscore, to show how many letters in the word that the player needs to guess. Once the player has put a letter and it’s correct, the underscore will be replaced with the correct word. If the word has the same letters, then there will be two underscores that will be replaced with the correct letter. But if the player guesses the wrong word, then the number of chances will be reduced to one, and the figure of the hanged man will be added. And if the player has guessed the wrong word six times, and the hanged man figure is complete, then the game is over. And if the player has guessed the word correctly, the game is over and there will be a congratulations message to the player.

**Section 2**

**2A. Design/Plan:**

The graph below shows how the flow of the game looks like:

As I have stated in the description in section 1, once the program has run, there will be an instruction in which the player needs to press any key then press enter to start the game. Once the game starts, unknown words represent by underscores ( \_ ) will be shown, and the player will enter a letter to start guessing. Just like the standard hangman game, there will be six chances, and the chances will be reduced if the player guessed the wrong letter. If there are no chances left, the player loses the game, and the program ended. And if the player guesses all of the letters correctly, the player wins and the program will end.

The chart below shows the functions that I am planning to use to make the hangman game:

This hierarchy chart is made a week before I started making the code for the game. This is based on what I have learned during introduction to programming class. The random generator is used to randomize the words that the player needs to guess in the game. If then is used to determine the letters entered by the player, for example if the letter entered is correct, then there will be a message that says that the player is correct and if the letter entered is incorrect, then there will be a message that says that the letter is wrong.

**2B. Explanation of the Functions:**

Functions:

* int letterFill (char, string, string&)

This function is used to determine whether the words that were filled is the same or not. And if in the word there’s the same letter, then the player does not need to guess the same word again.

* int main()

This function is the most important function where it is required for the program to run. For my hangman game program, this function is where all of my programs will work.

**Section 3**

**3A. What I’ve learned during the process**

During the process of making the hangman game for around six weeks, there are so many things that I had learned. What I’ve learned during the process of making this program is:

1. I get more understanding of the way each of the function works, instead of only learning it in class, reading the book, or asking friends, and apply it into my project.
2. I got to manage my time wisely and plan the things I will use inside of my program.
3. Understanding the proper way to use loop and how to stop the loop from looping until the end.
4. Understand how to randomize the words for the players to guess, and how using “rand” is an important way to randomize the numbers.
5. Be able to add picture of the hanged man figure by using cout, instead of only using text to determine how many chances the players have left.

**3B. Problems met and how I overcome it**

During the process of completing my program, there are many problems that I met, and these problems made the program unable to run smoothly like I expected. When I met these problems, I had to find a way to overcome it so that my program can work well. The problems that I overcome during the whole program making process is:

1. Getting confused on the functions that I need to put first, especially when I got confused on putting the if then first and the loop and vice versa. I overcame this problem by searching on the internet on which function should I use first then try to apply it on my code to see if it works or not.
2. I got really confused with the functions that I declared, because I thought that I need many functions to operate the hangman game. But I overcame it by looking at the functions carefully, and delete the ones that I don’t use so that I can know which are the declared functions that I used.
3. Selecting the words for the player to guess. I overcame it by deciding a theme for the words that the players need to guess, and I decided to use the name of animals as the words that the players need to guess so that the players will know what kind of word they have to guess without having to use hints for each of the words.
4. Placing the words that need to be guessed on either a separate file (in this case, a .txt file), or putting it inside the same .cpp file to make it easier for me to manage. I overcame this by deciding that maybe putting the words on the same .cpp file is easier for me to manage instead of separating it on a different file. I tried how to place the words that I will use on the same .cpp file, and finally decided to use arrays to keep the words that the player needs to guess.
5. The int letterFill function does not work when I used it inside the loop to determine if the letter that the player inputted is correct or not. I overcame this problem by searching on the internet on how they applied their int letterFill function that had been declared before the main function.
6. The unknown word is always the same when I keep restarting the game. I overcame this problem by putting #include<cstdlib> and #include <ctime> by searching the solution on the internet.

**Section 4**

**Coding**

#include <iostream>

#include <string>

#include <cstdlib>

#include <ctime> //Using csdtdlib and ctime so that the words will always be randomized

using namespace std;

int letterFill (char wordGuess, string unknownWord, string &guessword); // The function that will be used after the main function

int main()

{

string start;

const int numberOfTries = 6; // The maximum number of tries

char letter;

int wrongLetter;

int remainingTries;

string guessWord;

//The words that the player needs to guess was inside the array

string wordArr[23] =

{

"alpaca",

"butterfly",

"crocodile",

"crow",

"gazelle",

"hamster",

"lizard",

"hippopotamus",

"kangaroo",

"leopard",

"mongoose",

"tiger",

"lion",

"cheetah",

"panther",

"mouse",

"armadillo",

"rhinoceros",

"giraffe",

"sloth",

"otter",

"hyena"

};

//To randomize the words that the player needs to guess

srand(time(0));

int n=rand()% 23;

guessWord=wordArr[n];

//The letters will be replaced with \_

string unknownWord(guessWord.length(),'\_');

cout << " HANGMAN " << endl;

cout << " By: Angeline Indahsi " << endl;

cout << "===================================================" << endl;

cout << endl;

cout << "Welcome to Hangman! Now try to guess the name of an animal!" << endl;

cout << "You have 6 chances to guess the word. If you have reached 0 chances, you lose!" <<endl;

cout << "Press any key then press enter to start the game!" << endl;

cin >> start;

cout << "Guess the name of the animal by entering the letter! Good luck & have fun!" << endl;

while (wrongLetter < numberOfTries) //Loop until the player win or lose

{

cout << unknownWord << endl; //The unknown word represented with underscores will be shown

cout << "Enter a letter: ";

cin >> letter; //So that the player can input a letter

if (letterFill(letter, guessWord, unknownWord)==0) //If the letter guessed is wrong

{

cout << endl << "Sorry, wrong letter!" << endl;

wrongLetter++;

}

else

{

cout << endl << "Yay you found a letter! Keep it up!" << endl;

}

// Remind the user of the remaining chances left before losing

remainingTries = numberOfTries - wrongLetter;

cout << "You have " << remainingTries << " guesses left." << endl;

if (guessWord==unknownWord) //If the player had guessed all of the word right

{

cout << guessWord << endl;

cout << "Congratulations, you won!" << endl;

cout << "Thank you for playing hangman!" << endl;

}

}

if(wrongLetter == numberOfTries) //If there's no chances left

{

cout << "You have been hanged! You lose..." << endl;

cout << "The word was : " << guessWord << endl;

cout << "Thank you for playing hangman!" << endl;

}

cin.get();

cin.ignore();

return 0;

}

//This function is to determine whether the letter that the player had input is the same or not

int letterFill (char wordGuess, string unknownWord, string &guessword)

{

int i;

int matched=0;

int length=unknownWord.length();

for (i = 0; i< length; i++)

{

if (wordGuess == guessword[i])

return 0;

if (wordGuess == unknownWord[i])

{

guessword[i] = wordGuess;

matched++;

}

}

return matched;

}

**Resource**

(2014). C++ Hangman Game. Retrieved from <http://stackoverflow.com/questions/22616201/c-hangman-game>

Ripityrap (2015). C++ Hangman. Retrieved from <http://code.runnable.com/VZxNEjMvnqYR0Oim/c%2B%2B-hangman>