



HANGMAN

GAME

RaghavTheCyberdude

©TeraFlik | teraflik.com

Copyright ©2015, TeraFlik

GNU General Public License

Version 3, 29 June 2007

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation version 3 of the License.

CONTENTS

CONTENTS	2
DETAILS	3
Synopsis	3
Features.....	3
Execution	4
REQUIREMENTS AND SETUP	5
System Requirements	5
Support.....	5
SOURCE CODE.....	6
Header Files	6
Classes and Objects	6
Functions	6
Code.....	8
OUTPUT	16
POSSIBILITIES FOR MORE.....	20

DETAILS

Synopses

This project is an interesting game developed in the C++ language, based on the popular classic - Hangman.

User has to guess a secret word in a fixed number of guesses. He will be hanged if he cannot guess the letter in available chances.

With the option to select one level out of 5 available, the gameplay can be customized to fit the user's wish. Help is also included in the game itself and a webpage (teraflik.com/p/hangman.html) has been developed for further assistance.

Development of this game consumed a great deal of work and dedication, but the results has proved to be fruitful.

Features

Some features of the program are:

1. The console output of the program is presented in a neat and clean way.
2. Browsing through the user interface is easy and smooth.
3. User has the choice to select a difficulty level from: Very Easy, Easy, Moderate, Hard, and Extremely Hard.
4. The game is designed so that a word which has been asked, does not come up again, during a runtime.
5. There is a built-in help screen for users who are new to hangman.
6. Program comes bundled up with a big collection of country names.
7. User can easily add his own words and remove existing, due to the use of file handling.
8. The program uses many functions to make the transfer of control easier and hassle free.
9. The code has been written as per ANSI C++ standard, which makes it universally compatible.
10. This program is totally bug free and stable as suggested by results till date.
11. I have created a page on my website teraflik.com to provide support to this game, and also to provide updates.
12. Just visit: teraflik.com/p/hangman.html.

Execution

The game starts with the main menu showing up where user can select his choice by entering the serial number corresponding to his choice.

If user starts the game then all the details about his level are shown, and a hidden word is displayed, asking him to input a guess. This guess is compared to the secret letters, where, if user gets it wrong, he loses a life, or if he guesses correctly, then the process is repeated again until the whole word is guessed.

From the menu, user can change his level and also see how to play the game.

REQUIREMENTS AND SETUP

System Requirements

- OS - Microsoft Windows XP / 7 / 8 / 8.1 / 10
- RAM – 512 MB
- Free Hard Disk Space – 128 MB
- Internet Connection (for support, optional)

Support

Visit <http://www.teraflik.com/p/hangman.html> for full-fledged support and to report any bugs or improvements needed.

Any kind of feedback is important for further development of program.

SOURCE CODE

Header Files

The following header files have been included in the program from the C++ Standard library collection. The functions used are also mentioned.

```
#include <iostream> //for basic input and output
#include <cstdlib> //for exit() and system("cls") functions
#include <ctime> //for seeding srand() for random values
#include <cstring> //for all string functions
#include <fstream> //for handling the file "words.txt"
#include <conio.h> //for getch() function
```

Classes and Objects

A class named "WBank" with the following declaration has been used:

```
class WBank //class to store all the words
{
public:
    char W[MaxL]; //the word
    WBank()
    {
        strcpy_s(W, "null");
    }
};
```

Object of the class WBank used is "Word" declared as follows:

```
WBank Word[MaxW];
```

Functions

Several functions have been used in the program. Their prototypes are given below. Using functions makes the transfer of control easier and the same code doesn't need to be repeated again and again in the program.

```
void ShowMenu() //this is the main menu of the program
void SelectWord() //Word Selection Engine
void Game() //The main code for the execution of game
int LetterFill (char, char) //the function which checks each guess
void ShowHeader() //the function which shows the header on each screen
void HowToPlay() //how to play option
void LoadFile() //the function which loads the text file in memory
void LoadWBank() //to load the read file into our class WBank
void ChangeLevel() //function to change level
void ReturnLevel() //function to display which level is selected
char Input() //function to take the input
```


Code

```

/* This is the source code of the HANGMAN GAME developed by RAGHAV KHANDELWAL
as an open source project and has been posted on his blog: teraflik.com. The
complete source code can be downloaded from there only. The program uses many
functions to make the transfer of control easier and without any hassle. The
concept is very simple as mentioned in the project report.*/

/*The code has been written as per ANSI C++ standard and can be run only on
modern C++ compilers like Orwell Dev C++, or Visual Studio 2010 and later.
This code cannot be compiled on older compilers like Turbo C++.*/

/*The current version of program is totally bug free and totally stable.
But if anyone finds any bug in it please report me immediately at :
admin@teraflik.com or teraflik.com/p/contact.html */

#include<iostream> //The standard library files used
#include<cstdlib>
#include<ctime>
#include<cstring>
#include<fstream>
#include<conio.h>
using namespace std;

void ShowCategory(); //all the function prototypes
void ChangeLevel();
void Game();
void HowToPlay();
void ShowHeader();
void ShowMenu();
void LoadFile();
void LoadWBank();
void ReturnLevel();
void SelectWord();
int LetterFill (char, char []);
char Input();

/*All the global variables*/

const int MaxL = 30; //The max length of a word
const int MaxW = 200; //Max words in the word bank

char FileContent[2000]; //The text file will be read into this
int NoTries = 7; //No of tries depends on the level selected
int NewLineCount = 0; // to count the number of words in text file

```

```

int CurrentW; //to store which word is in process
int UsedW[MaxW]; //to store all the SrNo of used words
int NoUsedW = 0; //to store no of used words

class WBank //class to store all the words
{public:
    char W[MaxL]; //the word
    WBank() //constructor
    {
        strcpy_s(W, "null");
    }
};
WBank Word[MaxW];

int main() //The main function
{
    srand(time(NULL)); //to randomize the random function
    LoadFile();
    LoadWBank();
    ShowMenu();
    _getch();
    return 0;
}

void ShowMenu() //this is the main menu of the program
{
    char cho; //to store choice
    ShowHeader();
    cout<<"Select an Option: ";
    cout<<"\n1. Start Game";
    cout<<"\n2. Change level";
    cout<<"\n3. How to play";
    cout<<"\n4. Exit";
    cout<<endl<<"\t\tChoice: ";
    cho = Input(); //Input function returns input values

    if(cho == '1') Game(); //start game
    else if(cho == '2') ChangeLevel(); //change level
    else if(cho == '3') HowToPlay(); //how to play option
    else if(cho == '4') //exit program
    {
        cout<<"\t\tExited succesfully!";
        _getch();
        exit(0);
    }
    else
    {
        cout<<"\t\tInvalid Input! Try again.";
        _getch();
        ShowMenu();
    }
}

```

```

void SelectWord() //Word Selection Engine
{
    /*generate a random int and compare it to all the previously
    selected words, so that the same word is not put up again*/
    int i=0;
    int n = rand()% NewLineCount;
    for(i=0;i<NoUsedW;i++)
    {
        if(UsedW[i] == n)
        {
            i=0;
            Game();
        }
    }
    UsedW[NoUsedW] = n;
    NoUsedW++;
    CurrentW = n;
}

void Game() //The main code for the execution of game
{
    SelectWord();
    ShowHeader();
    char Unknown[MaxL]; //the secret word is stored in it
    char Letter; //user's guessed letter
    int NoWrongGuess=0; //no of wrong guesses
    int i;
    int length = strlen(Word[CurrentW].W);

    for (i = 0; i < length; i++) // to generate the secret word
    {
        Unknown[i]='*';
        if(Word[CurrentW].W[i] == ' ')
        {
            Unknown[i] = ' ';
        }
    }
    Unknown[i]='\0';

    cout<<"Category: Countries ===== Level: ";ReturnLevel();
    cout<<"== No of Tries: "<<NoTries<<" "<<endl;
    cout<<"~~~~~";
    while (NoWrongGuess < NoTries) //condition for running program
    {
        cout<<endl<<endl<<Unknown;
        cout<<endl<<"\tYour Guess: ";

        while(!( Letter=Input() )) //if user provides invalid input
        {

```

```

        cout<<"\tInvalid input! Try again: ";
    }

    if (LetterFill(Letter, Unknown)==0)//if user provides wrong guess
    {
        cout<<"\tOops! Your guess is wrong!";
        NoWrongGuess++;
    }
    else //else he will find the letter
    {
        cout<<"\tYou found a letter!";
    }

    cout<<" Now You have "<<NoTries-NoWrongGuess<<" tries left.";

    if (strcmp(Word[CurrentW].W, Unknown) == 0) //checking if he guessed
the word
    {
        cout<<endl<<endl<<"Yeah! You got it! The word was:
"<<Word[CurrentW].W;
        cout<<endl<<endl;
        break;
    }
}
if(NoWrongGuess == NoTries) //condition for user to lose
{
    cout<<endl<<endl<<"\nSorry, you lose...you've been hanged.";
    cout<<" The word was: "<<Word[CurrentW].W<<endl<<endl;
}
_getch();
ShowMenu();
return;
}

int LetterFill (char guess, char guessword[]) //the function which checks each
guess
{
    int i;
    int matches=0;
    for (i = 0; Word[CurrentW].W[i]!='\0'; 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 == Word[CurrentW].W[i])
        {
            guessword[i] = guess;

```

```

        matches++;
    }
}
return matches;
}

void ShowHeader() //the function which shows the header on each screen
{
    system("cls");
    cout<<endl<<"-----+";
    cout<<endl<<"|                               Hangman Game by Raghav - XII Science B - GRGA |";
    cout<<endl<<"-----+";
    cout<<endl<<endl;
    return;
}

void HowToPlay() //how to play option
{
    ShowHeader();
    cout<<"What...You've never played hangman! Seriously? Sigh...";
    cout<<"\n\nHere's how to play: ";
    cout<<"\n - Try to solve the riddle before your chances run out.";
    cout<<"\n - Type in the letters you wish to guess. Careful! Wrong letters lose chances.";
    cout<<"\n - You can change the level from the Change Level option.";
    cout<<"\n - For more stuff, visit \"teraflik.com\" (blog by Raghav Khandelwal)";
    cout<<"\n\n\t\t\t\t\t Press any key to go back to main menu...";
    _getch();
    ShowMenu();
}

void LoadFile() //the function which loads the text file in memory
{
    char ch;
    int i=0;
    ifstream fin("Words.txt", ios::in); //open file
    fin.seekg(0);
    if(!fin) //if the file is not found there
    {
        ShowHeader();
        cout<<"\nError 701. Unable to open file!";
        _getch();
        cout<<"\n\n - This program needs the accompanying file \"Words.txt\" to run.";
        cout<<"\n - Please check that the file exists with program and restart again.";
    }
}

```

```

        cout<<"\n - If you can't find the file then create a new text file
with the same name.";
        cout<<"\n - Type the words to be asked in the hangman game in separate
lines.";
        _getch();
        cout<<"\n\n - For further assistance contact me at:
teraflik.com/p/contact.html";
        _getch();
        exit(1); //exit with 1 status
    }
    while(true)
    {
        fin.get(ch);
        if( fin.eof() ) break;
        FileContent[i] = ch;
        i++;
    }
    fin.close();
    //cout<<FileContent; getch();
    return;
}

void LoadWBank() //to load the read file into our class WBank
{
    int j=0, k=0;

    for(int i=0; FileContent[i]!='\0'; i++)
    {
        if(FileContent[i] == '\n')
        {
            NewLineCount++;
            Word[j].W[k] = '\0';
            j++;
            k=0;
            continue;
        }
        Word[j].W[k] = FileContent[i];
        k++;
    }
    return;
}

void ChangeLevel() //function to change level
{
    ShowHeader();
    char cho;
    cout<<"Select a level: ";
    cout<<"\n1. Very Easy";
    cout<<"\n2. Easy";
    cout<<"\n3. Moderate";

```

```

        cout<<"\n4. Hard";
        cout<<"\n5. Extremely Hard";
        cout<<"\n\t\tChoice: ";
        cho = Input();
        if(cho == '1') NoTries = 15;
        else if(cho == '2') NoTries = 10;
        else if(cho == '3') NoTries = 7;
        else if(cho == '4') NoTries = 5;
        else if(cho == '5') NoTries = 3;
        else
        {
            cout<<"\t\tInvalid Input! Try again.";
            _getch();
            ChangeLevel();
        }
        cout<<"\t\tLevel Changed !";
        _getch();
        ShowMenu();
        return;
    }

void ReturnLevel() //function to display which level is selected
{
    if(NoTries == 15)    cout<<"Very Easy =====";
    else if(NoTries == 10)cout<<"Easy =====";
    else if(NoTries == 7) cout<<"Moderate =====";
    else if(NoTries == 5) cout<<"Hard =====";
    else if(NoTries == 3) cout<<"Extremely Hard ";
    else
    {
        ShowHeader(); //if some invalid level is selected by user modification
        cout<<"Error 702. Invalid Level Selected";
        _getch();
        cout<<"\n - Some bug or unwanted reasons have caused conflicts in the
program.";
        cout<<"\n - Please get a proper working program or code from available
sources.";
        cout<<"\n\n - Seek further assistance at: teraflik.com/p/contact.html";
        _getch();
        exit(1);
    }
    return;
}

char Input() //function to take the input.
{
    /*This Input() function makes the input of the program
    bug free. Any invalid or unacceptable input will return value 0*/
    char cho1[40];
    cin>>cho1;

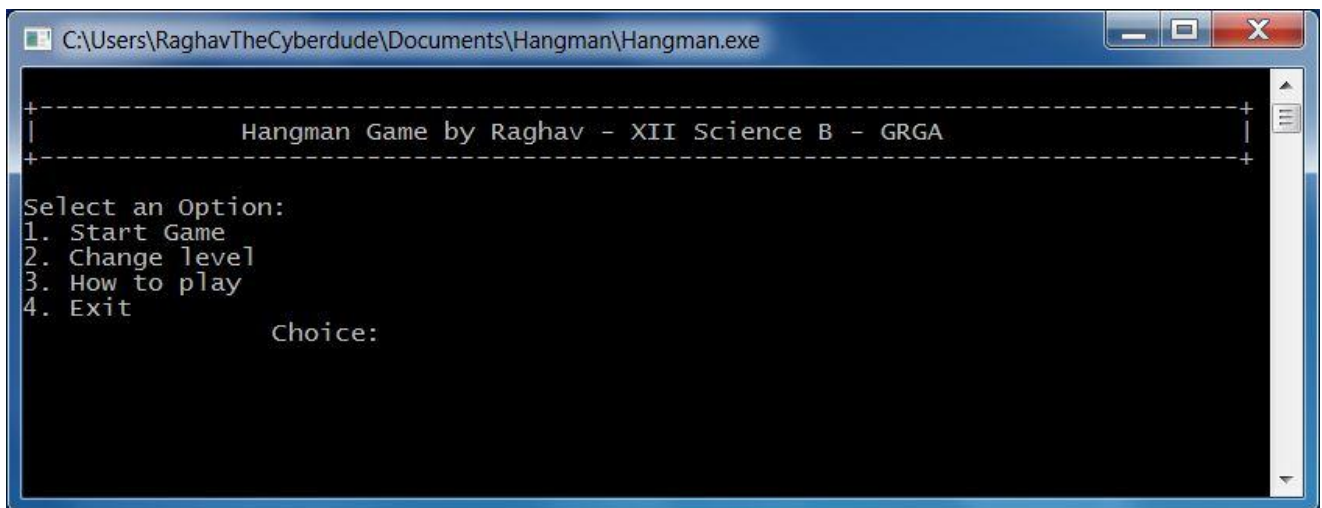
```

```
cin.ignore();

    if(strlen(cho1)>1)
    {
        return '0';
    }
    else
    {
        return cho1[0];
    }
}

/*End of program. For more stuff, visit: teraflik.com */
```


OUTPUT



A screenshot of a Windows command prompt window titled "C:\Users\RaghavTheCyberdude\Documents\Hangman\Hangman.exe". The window displays the title "Hangman Game by Raghav - XII Science B - GRGA" enclosed in a dashed border. Below the title, it prompts the user to "Select an Option:" and lists four choices: "1. Start Game", "2. Change level", "3. How to play", and "4. Exit". The prompt "Choice:" is displayed at the bottom of the menu.

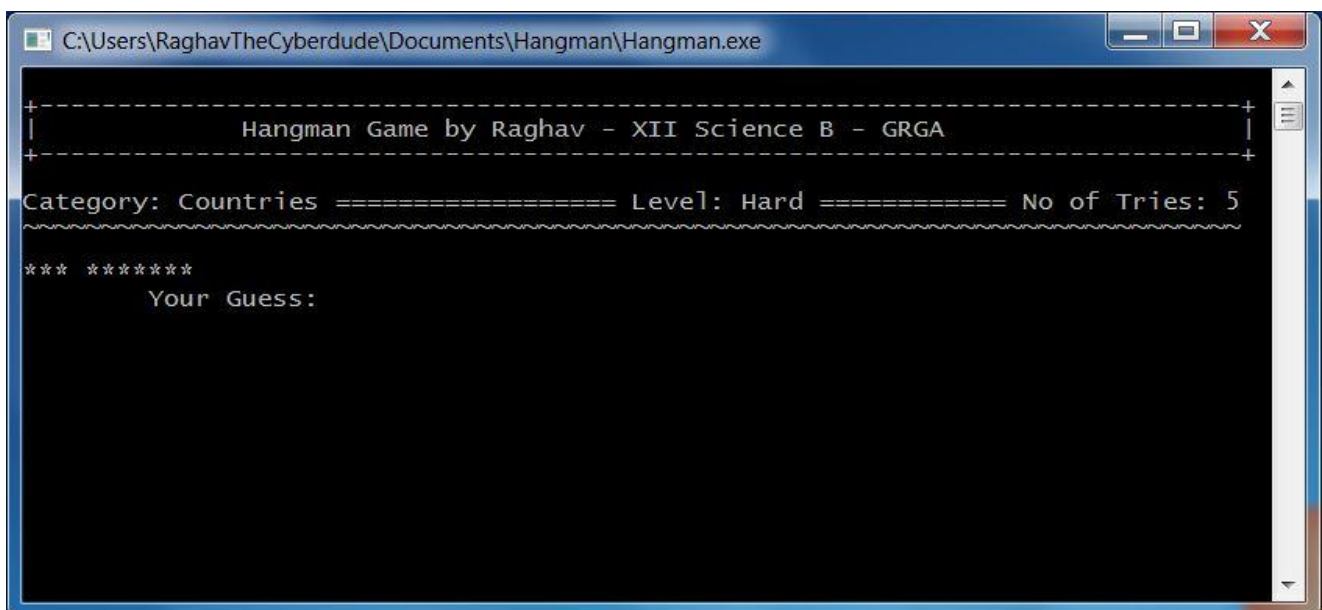
```
C:\Users\RaghavTheCyberdude\Documents\Hangman\Hangman.exe

+-----+
|               Hangman Game by Raghav - XII Science B - GRGA               |
+-----+

Select an Option:
1. Start Game
2. Change level
3. How to play
4. Exit

Choice:
```

Snapshot 1 - The Main Menu



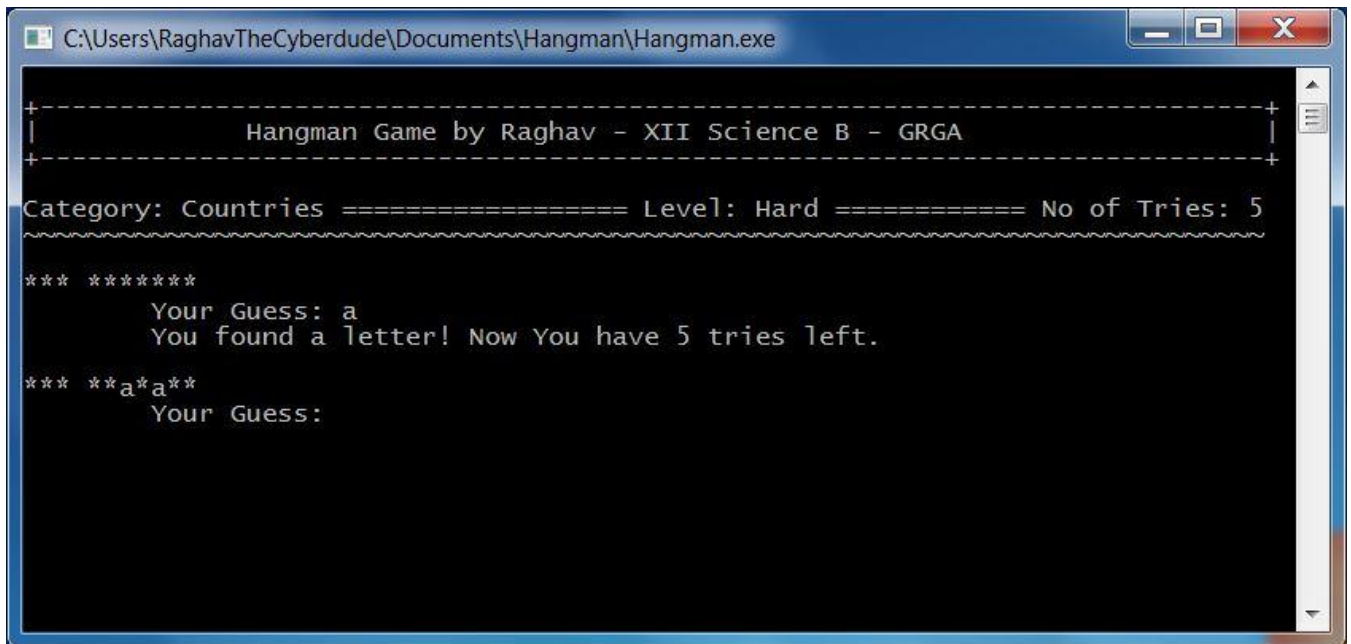
A screenshot of the same Windows command prompt window, now showing the game's starting screen. It displays the category "Category: Countries", the level "Level: Hard", and the number of tries "No of Tries: 5". A dashed line separates this information from the next prompt, which is "Your Guess:" preceded by three asterisks and a series of seven asterisks.

```
C:\Users\RaghavTheCyberdude\Documents\Hangman\Hangman.exe

+-----+
|               Hangman Game by Raghav - XII Science B - GRGA               |
+-----+

Category: Countries ===== Level: Hard ===== No of Tries: 5
~~~~~
*** *****
Your Guess:
```

Snapshot 2 - Starting the Game



A screenshot of a Windows application window titled "C:\Users\RaghavTheCyberdude\Documents\Hangman\Hangman.exe". The window contains a text-based Hangman game interface. At the top, it says "Hangman Game by Raghav - XII Science B - GRGA". Below that, it displays "Category: Countries", "Level: Hard", and "No of Tries: 5". The game state shows a word with one letter revealed: "a". The prompt "Your Guess:" is followed by the user's input "a". A message "You found a letter! Now You have 5 tries left." is displayed. The current state of the word is shown as "a*a".

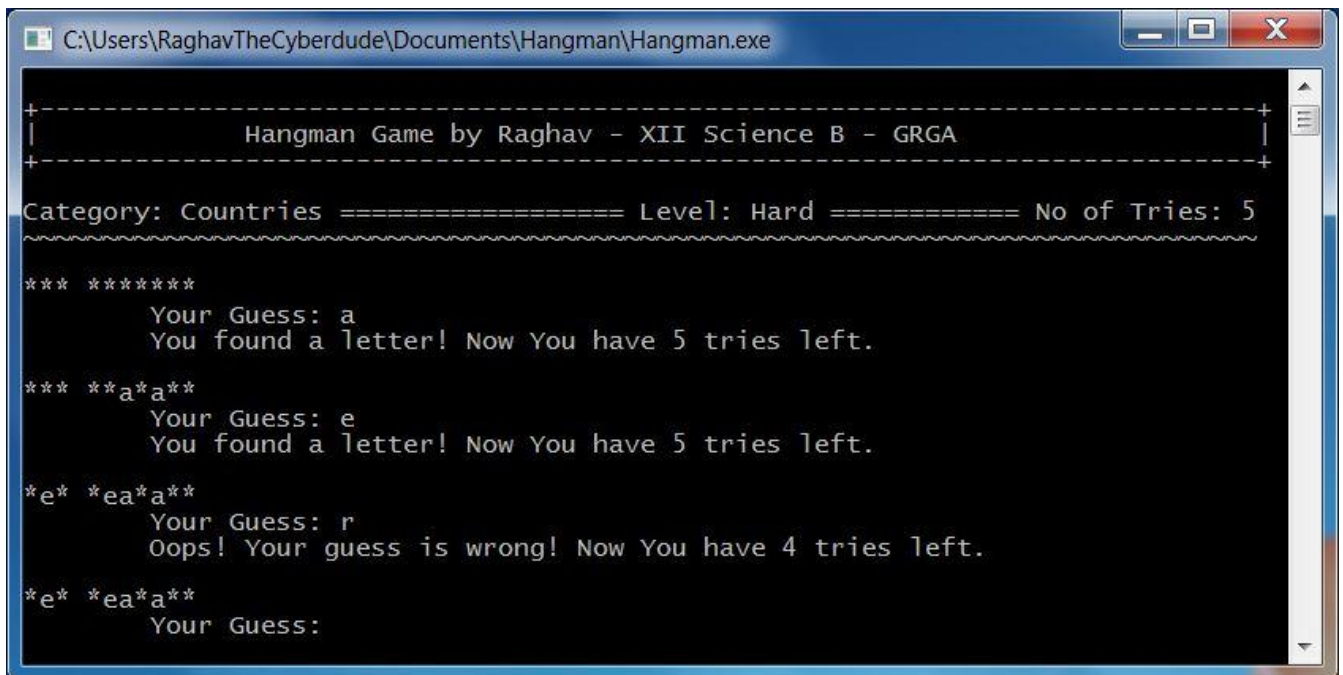
```
+-----+
|               Hangman Game by Raghav - XII Science B - GRGA               |
+-----+

Category: Countries ===== Level: Hard ===== No of Tries: 5
~~~~~

*** *****
    Your Guess: a
    You found a letter! Now You have 5 tries left.

*** **a*a**
    Your Guess:
```

Snapshot 3 - Guessing a letter correctly



A screenshot of the same Hangman game window, showing the next step. The user has guessed "e", which is also correct, as indicated by the message "You found a letter! Now You have 5 tries left." The word state is now "e*a*a". The user then guesses "r", which is incorrect, as indicated by the message "Oops! Your guess is wrong! Now You have 4 tries left." The word state remains "e*a*a".

```
+-----+
|               Hangman Game by Raghav - XII Science B - GRGA               |
+-----+

Category: Countries ===== Level: Hard ===== No of Tries: 5
~~~~~

*** *****
    Your Guess: a
    You found a letter! Now You have 5 tries left.

*** **a*a**
    Your Guess: e
    You found a letter! Now You have 5 tries left.

*e* *ea*a**
    Your Guess: r
    Oops! Your guess is wrong! Now You have 4 tries left.

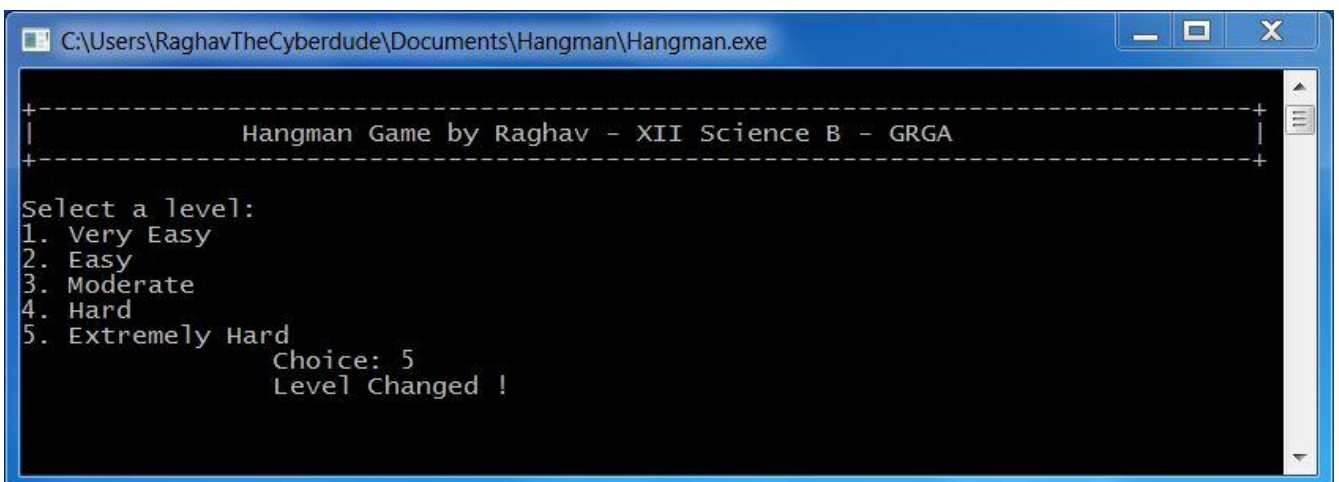
*e* *ea*a**
    Your Guess:
```

Snapshot 4 - Guessing a letter wrong



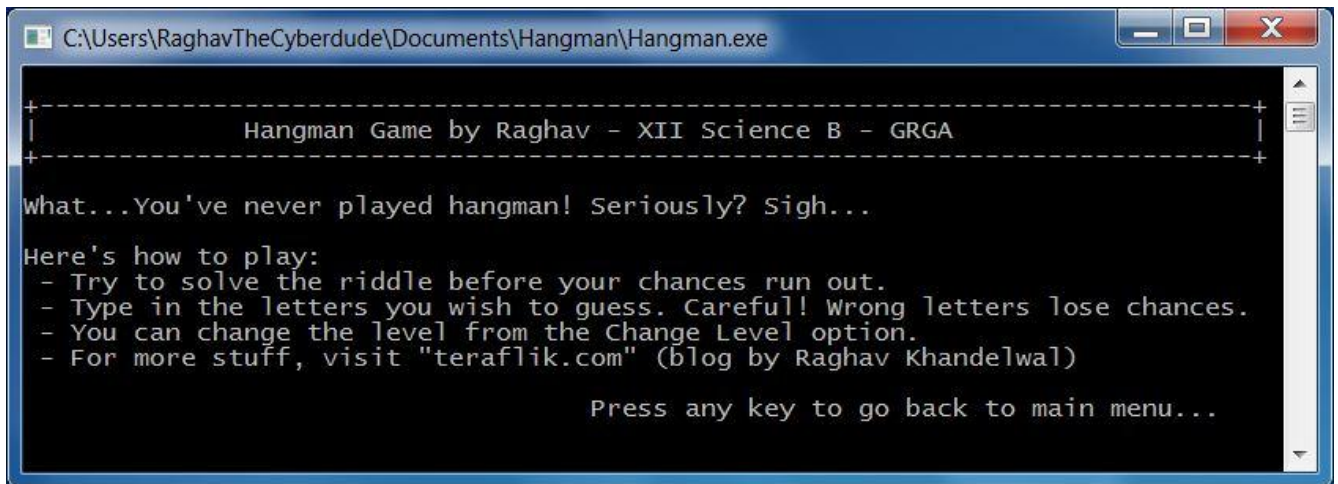
```
C:\Users\RaghavTheCyberdude\Documents\Hangman\Hangman.exe
*e* *ea*a**
  Your Guess: n
  You found a letter! Now You have 4 tries left.
ne* *ea*an*
  Your Guess: d
  You found a letter! Now You have 4 tries left.
ne* *ea*and
  Your Guess: w
  You found a letter! Now You have 4 tries left.
new *ea*and
  Your Guess: z
  You found a letter! Now You have 4 tries left.
new zea*and
  Your Guess: l
  You found a letter! Now You have 4 tries left.
Yeah! You got it! The word was: new zealand
```

Snapshot 5 - Finding the whole word



```
C:\Users\RaghavTheCyberdude\Documents\Hangman\Hangman.exe
+-----+
|                               |
|           Hangman Game by Raghav - XII Science B - GRGA           |
|                               |
+-----+
Select a level:
1. Very Easy
2. Easy
3. Moderate
4. Hard
5. Extremely Hard
      Choice: 5
      Level Changed !
```

Snapshot 6 - Selecting another level



A screenshot of a Windows application window titled "C:\Users\RaghavTheCyberdude\Documents\Hangman\Hangman.exe". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is black with white text. At the top, a title bar is enclosed in a dashed box: "Hangman Game by Raghav - XII Science B - GRGA". Below this, the text reads: "What...You've never played hangman! Seriously? Sigh...". This is followed by "Here's how to play:" and a bulleted list: "- Try to solve the riddle before your chances run out.", "- Type in the letters you wish to guess. Careful! Wrong letters lose chances.", "- You can change the level from the Change Level option.", "- For more stuff, visit 'teraflik.com' (blog by Raghav Khandelwal)". At the bottom, it says "Press any key to go back to main menu...".

```
C:\Users\RaghavTheCyberdude\Documents\Hangman\Hangman.exe

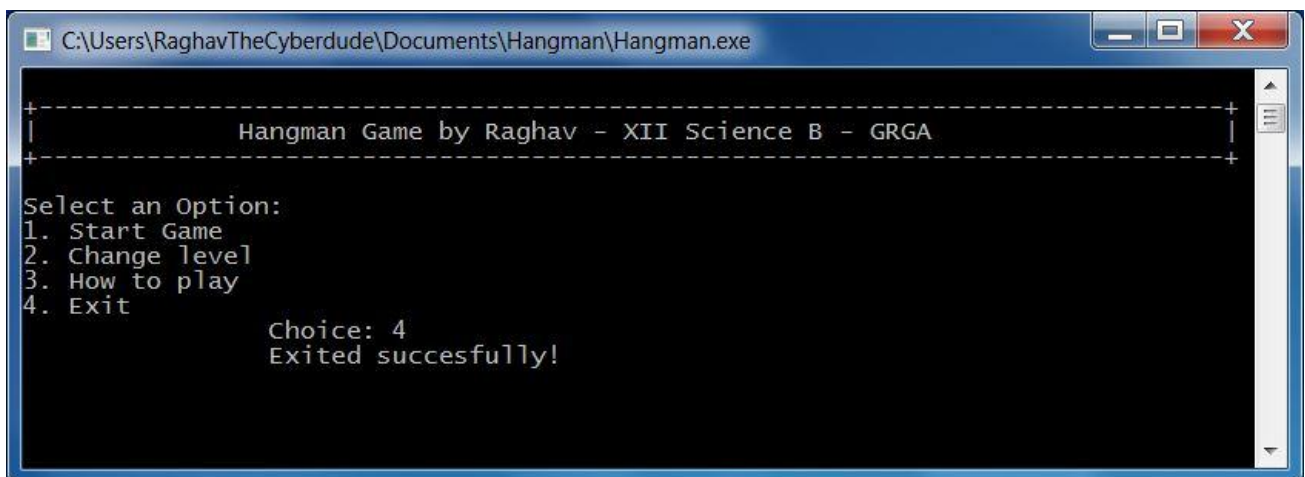
+-----+
|               Hangman Game by Raghav - XII Science B - GRGA               |
+-----+

What...You've never played hangman! Seriously? Sigh...

Here's how to play:
- Try to solve the riddle before your chances run out.
- Type in the letters you wish to guess. Careful! Wrong letters lose chances.
- You can change the level from the Change Level option.
- For more stuff, visit "teraflik.com" (blog by Raghav Khandelwal)

Press any key to go back to main menu...
```

Snapshot 7 - How to play option



A screenshot of the same Windows application window. The title bar and dashed box title are the same. The text now shows a menu: "Select an Option:" followed by a numbered list: "1. Start Game", "2. Change level", "3. How to play", and "4. Exit". Below the list, it says "Choice: 4" and "Exited succesfully!".

```
C:\Users\RaghavTheCyberdude\Documents\Hangman\Hangman.exe

+-----+
|               Hangman Game by Raghav - XII Science B - GRGA               |
+-----+

Select an Option:
1. Start Game
2. Change level
3. How to play
4. Exit

Choice: 4
Exited succesfully!
```

Snapshot 8 - Exiting the game

POSSIBILITIES FOR MORE

Even after putting much efforts to get the game to its current level, there still remain some possibilities to do more. They can include:

- Allowing user to play the game in more categories of words.
- Displaying the characters already used up by the user.
- Using graphics to display the hangman, to make the game more appealing.

I hope to add these features in the game, in its process of further development.