**A PROJECT REPORT**

**ON**

**HANGMAN GAME**

Submitted in partial fulfilment of the requirement for the 3rd semester

**Bachelor of Technology**

**By**

**Aman Singh**

**20021728**



**DEPARTMENT OF COMPUTER SCIENCE &** **ENGINEERING**

**GRAPHIC ERA DEEMED UNIVERSITY**

**DEHRADUN**

**2021-2022**

Project title:

HANGMAN

Acknowledgement

Here by I am submitting the project report on **“HANGMAN project”** as per the scheme of Graphic Era Deemed University, Dehradun.

I would like to express our sincere gratitude to **Dr. Devesh Pratap Singh,** Head of Dept. of Computer Science, for providing a congenial environment to work in and carry out our project.

I consider it mine cardinal duty to express the deepest sense of gratitude to Mr. **Deepak Uniyal**, our mini project guide and Application for the invaluable guidance extended at every stage and in every possible way.

Finally, I am very much thankful to all the faculty members of the Department of Computer Science and Technology, friends and our parents for their constant encouragement, support and help throughout the period of project conduction.

Aman Singh

Roll no. 2016620

Motivation

Since, the burden of academic studies and the need to perform well in my college, I took a toll on my extra learning hours, I decided to opt for this project thereby indirectly making it a part of my syllabus and hence it led me to know how to efficiently work on large amount of data. I would also like to add that this project would have been harder to make if it wasn’t for the immense help and support of my mentor Mr. Deepak Uniyal, who was always ready to clear any doubts I had. He was always just a text or a call away. I would like to thank him from the bottom of my heart.

**Problem Statement-**

Given the attached 50,000 words dictionary, write a program that can play hangman by choosing a letter based on the current state of the board. A board would initially have a series of blanks, each representing one letter in the word. As letters are guessed, all spaces in the word that match the letter should be replaced with that letter. Any letters that have no matches are put on a list of missed letters. Once the list of missed letters reaches 6, the game is lost. Write a C++ program that takes the word as a command-line argument and shows the board after each step as well as ultimate outcome. The guesses should come from the program, not from user input.

**The code must pass all these constraints:**

1.Number of words tested: 50,000

2.Number of words guessed correctly. Minimum 40,000

3. Correct Guesses (%): Minimum 80.0%

4.Time to run: less than 60 seconds

**Problem Analysis-**

We have to create two programs that run simultaneously one program should be of the hangman that asks for a letter and checks if that letter is in the word if it is then it should ask another letter and so on if the letter guessed isn’t in the word it should decrease the life until and then again ask for the letter it should do so until the word is guessed or the life which was initially 5 eventually goes down to zero.

The second program function is to supply letters to hangman program. It supplies letter on some logic.

**Programming concepts used in making of HANGMAN project –**

Basic Knowledge of:

1. C++
2. file handling
3. Linked list
4. 2D array
5. Dynamic arrays

Methodology and Logic

We created two tables (2D array) using create\_tables() function the first table has words in increasing order of the word length and the second table consist of information on first table about each and every word length i.e where is the start of the word having a particular word length (starting row of particular word length) and the ending row of particular word length it also contains information about the most used letter ,second most used letter, third most used letter, fourth most used letter, fifth most used letter. Here, the most used series comes from a very basic logic that we maintain a record of the frequency of the letter that comes after we traverse through the table 1 from starting row to end row for a particular word length, that has all the words arranged in ascending order of their word length. the hangman program asks for a letter which is either provided by the table 2 or the used series linked list created after we have correctly guessed a single letter or more than a single letter whenever a letter is guessed correctly there is a temporary table created that contains word of that particular pattern and then again, we create a most used series based on that temporary table

The table1 contains all 50,000 words in the file in ascending order of their word length

The table 2 contains start which represents the starting of the row of a corresponding word length

Suppose the word is A d m i n s t r a t i o n

Word length -13

From table 2 the program would take the values from the 0th and 1st column of the 12th row, and then fetch all the data from table 1 from the value of 0th column of the table 2 to 1st column of table 2 and then it will take the most used letter as a guess suppose the word was ‘z’ then it would deduct a life and then again ask for a word then it would take 2nd most used word suppose it is ‘a’ the program will get new pattern as the letter was present in the word

The new pattern will be a\_ \_ \_ \_ \_ \_ \_a\_ \_ \_ \_

Then it will create a temporary table that will contain words following this pattern and it will create a most used series list for that temporary table and then the most used letter in that series would act as the next guess for the program if the letter is found, we get a new pattern and thus a new temporary table and a corresponding most used letter list, if the letter is not in the word then it will take the next most used letter from the series.

This process will occur until the life runs out or the word is guessed correctly.

CONCLUSION

We have successfully created Hangman project and We have learnt about various concepts of C++ such as file handling, linked lists, Dynamic Arrays and 2D arrays. We have also learnt about various technique for efficiently creating tables(2D arrays) for managing data in a file. The accuracy for the program is 98%.our program takes approximately 25 seconds to run.

**Accuracy=(correctly guessed word/total words)x100**

BIBLIOGRAPHY

1. <https://www.youtube.com/watch?v=IRBxR56mvQg&t=1s>
2. <https://www.cplusplus.com/>
3. <https://github.com/deepakuniyaliit/mini-project-ideas>