BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT

Yelahanka, Bangalore – 560 064

Department of Computer Science & Engineering

Synopsis for the Project work

"Anagrams Using Python Programming"

Submitted By:

1. Abhishek B (1BY15CS005)

2. Mohammed Daaniyaal (1BY15CS051)

Under the Guidance of

Mrs. Mari Kirthima A

2017-2018

ABSTRACT :

This project is based on Anagrams i.e a simple game involving strings where the user will be shown a set of 5 or 6 letters. The task is the user should form a meaningful word with the available letters. Based on the length of the correct word the user has formed, he will be awarded so many points. Such games are also available on game portals such as Miniclip and Zapak.com. We will be trying to provide a better user interface for the users of the game along with some added features such as leader board.

INTRODUCTION :

The project demonstrates the working of a simple 2-dimensional game. It is incorporated from the online game "Anagram Magic". It is implemented as a web application containing a front end and a back end. It would be an intellectual game where the user's power of word formation is tested. The main focus of this game is to provide the user with an appealing user interface which would give the user a realistic game feeling. The backbone of this project is strings. The user will be provided a string containing random characters and he has to form a meaningful word out of them. Based on the length of the correct word formed, the user will be awarded points. Additional features such as leader board will be available and also features like real time competition with an other user online can be added later.

MOTIVATION :

We got the idea of this project after going through a lot of python projects on various sources such as github, youtube, stack overflow. We found the idea of this game interesting and easy to implement. We mainly drew inspiration from the online game “Anagram Magic” and wanted to build a game similar to that with enhanced capabilities. This project can be implemented in python using frameworks such as Flask and internet development applications such as pyjs.

EXISTING SYSTEM :

There are several anagram games found online. The best known up till now is Anagram Magic from miniclip.

LIMITATIONS OF EXISTING SYSTEM :

A lot of games present do not offer real time competition with other players online or friends through social media. This system has that privilege and also enables players to be ranked according to their scores on a virtual leader board.

PROPOSED SYSTEM :

The whole system will be implemented using pyjs and Flask. pyjs is a Rich Internet Application (RIA) Development Platform for both Web and Desktop. With pyjs you can write your JavaScript-powered web applications entirely in Python. It also contains a Python-to-JavaScript compiler, an AJAX framework and a Widget Set API. pyjs Desktop allows the exact same Python web application source code to be executed as a standalone desktop application (running under Python) instead of being stuck in a Web browser. Flask, a web framework in python will be used at the back end for storing the words and users online.

Real time competition of users will be provided with the concept of a web forum. At any time only 2 users will be allowed to compete each other until they finish the game. The virtual leader board will be stored at the back end in a database.

SYSTEM REQUIREMENTS SPECIFICATIONS :

HARDWARE REQUIREMENTS -

- 1 GB free hard disk space
- 2 GB RAM
- Intel core i3 or higher
- 1 GB graphics card

- **SOFTWARE REQUIREMENTS -**
- Windows 7 or higher or Unix operating system
- Flash Player
- Web browser – Google Chrome / Mozilla Firefox / Internet Explorer

METHEDOLOGY :

There are different ways of implementing this project—

1. Checking off—

Our first solution to the anagram problem will check to see that each character in the first string actually occurs in the second. If it is possible to "checkoff" each character, then the two strings must be anagrams. Checking off a character will be accomplished by replacing it with the special Python value None. However, since strings in Python are immutable, the first step in the process will be to convert the second string to a list. Each character from the first string can be checked against the characters in the list and if found, checked off by replacement.

2. Sort and Compare—

Another solution to the anagram problem will make use of the fact that even though `s1` and `s2` are different, they are anagrams only if they consist of exactly the same characters. So, if we begin by sorting each string alphabetically, from a to z, we will end up with the same string if the original two strings are anagrams.

3. Count and compare—

Our final solution to the anagram problem takes advantage of the fact that any two anagrams will have the same number of a's, the same number of b's, the same number of c's, and so on. In order to decide whether two strings are anagrams, we will first count the number of times each character occurs. Since there are 26 possible characters, we can use a list of 26 counters, one for each possible character. Each time we see a particular character, we will increment the counter at that position. In the end, if the two lists of counters are identical, the strings must be anagrams.

REFERENCES :

- https://stackoverflow.com/
- www.sourceforge.net
- https://www.youtube.com/
- https://www.wikipedia.org/
- The Python Workbook – Ben Stephenson