Final Project Report GDG, 2CC

Vellore Institute of Technology

By, K.N.Sri Dharmasastha 16BCE0956

Contents

1. Abstract ………………………………………………………………..…. 3

2. Introduction …………………………………………………………..….. 3

3. Methodology …………………………………………………………..….. 4

4. Block Diagram ………………………………………………………….... 5

5. Execution ……………………………………………………………….… 6

6. Difficulties ………………………………………………………………... 8

7. Conclusion ………………………………………………………………... 8

8. References …………………………………………………………....…… 9

Abstract

Create a GUI game in which whenever game starts , the game gives 10 randomly selected anime names and jumbled , the user has to predict the correct name of the anime and accordingly how many he guesses correct he gets 1 point. Remember the jumbled word should be different every time and also jumbled in a different way.

Introduction

Jumble is a word puzzle with a clue, a drawing illustrating the clue, and a set of words, each of which is “jumbled” by scrambling its letters. A solver reconstructs the words, and then arranges letters at marked positions in the words to spell the answer phrase to the clue. The clue and illustration always provide hints about the answer phrase. The answer phrase frequently uses a homophone or pun. But the exact word is taken as the correct answer and not he homophone or pun of the given jumbled word.

I use a python toolkit PyQt to create the GUI. You can use other toolkits for creating the GUI such as Tkinter, wxPython etc. PyQt is the easiest among all the toolkits I’ve tried working with.

Methodology

The first thing I did was create a list of anime titles. Then created a function which will randomly select a title from the list made before. Then I jumbled the word using **jumble()** in-build function. I stored this in a new variable.

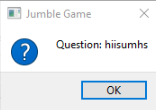
I, then passed this value to the PyQt function which is messagebox that pops up a window when called. This window will display the question that is the jumbled word.

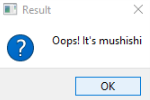
Once you click okay, a new window opens up with an input box where the player types their guess. This value will be passed to logic again and in the logic, the string taken from the player, will be compared to the original anime title string.

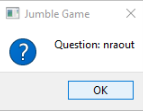
If the string matches the original title, then I increase the score by 1 and display the word correct.

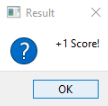
If the string does not match, the score stays the same and the correct answer is displayed.

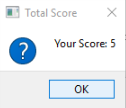
Execution











Difficulties

Learning PyQt was easy. Learning and creating the logic for this problem statement is easy. But I had lot of difficulty attaching the logic to and its inputs and outputs to the windows and displaying functions of PyQt. I couldn’t figure how to loop them and where and how to pass the values created and collected among the logic and the interface functions.

Conclusion

This project helped me a lot to learn about python and better use the toolkit made for GUI creation. I used PyQt which I could understand the easiest among all the other toolkits I tried. This was a fun experience to experiment with.

References

<https://www.tutorialspoint.com/pyqt/>

<https://pythonprogramming.net/basic-gui-pyqt-tutorial/>

<https://www.tutorialspoint.com/python/>

<https://www.w3schools.com/python/>