ANAGRAM GAME

A Project Work Report

Submitted in the partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Submitted by:

ARSHIYA SARMAI

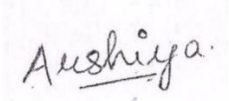
19BCS6060

Under the Supervision of:

PROF. SOMSIRSA CHATTERJEE

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI-140413, PUNJAB

December, 2020



Arshiya Sarmai
Name and signature of student(s)

Name and signature of Supervisor

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

PROJECT COMPLETION CERTIFICATE

All in one converter

This is to certify that the Arshiya Sarmai has successfully completed the project work titled **"**Anagram Game**"** ***Submitted in the partial fulfilment for the award of the degree of*** **BACHELOR OF ENGINEERING IN Artificial intelligence and Machine learning. This project is the record of authentic work carried out during the academic year 2020-2021**

Project Guide

Date: 12 November 2020

DECLARATION

I the undersigned solemnly declare that the project report is based on my own work carried out during the course of our study under the supervision of PROF. SOMSIRSA CHATTERJEE. I assert the statements made and conclusions drawn are an outcome of my work. I further certify that the work contained in the report is original and has been done by me under the general supervision of my supervisor.

II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

III. We have followed the guidelines provided by the university in writing the report.

IV. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Name: Arshiya Sarmai
19BCS6060

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Chandigarh University for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.
I would like to express my gratitude towards my parents and my department for their kind co-operation and encouragement which help me in completion of this project.

THANKS AGAIN TO ALL WHO HELPED

Content **Page number**

Chapter 1: Introduction to project

'The Anagram Game'. An Anagram of a word is another word obtained by shuffling its letters. For example, the words God and dog are anagrams of each other. This is what the anagram game is all about. The basic idea of this project is to read words pick a random word from this dictionary, shuffle it and ask the user to guess the correct word from its anagram. The user is asked to enter his name and start the game. Then, the random words from the list will be given. The user is given one attempt to guess the correct word. After each incorrect attempt or if the user is not able to guess the correct word then the user is provided with the correct word and the game moves on to the next word. After all the five words, the score will be given in the last.

This Anagram game is in Python. Module Used – Tkinter() -It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task. Also, the design of this system is pretty simple so that the user won't get any difficulties while working on it.

Chapter 2: Project Requirements

Hardware:

The minimum hardware specifications for building up this project will be as follows:

>> Processor: Core 2 Duo – 2.0 GHz

>> Random Access Memory – 1 GB

>> Keyboard or Mouse

Software:

The software required in order to build the system is as follows:

>> Documentation: WPS Office

>> Diagram Generation: WPS Office

>> Coding: IDLE(Python 3.8)

>> Operating System : Windows, Mac, Linux

Chapter 3: Implementation Details

Algorithm:

1. Start.
2. Import tkinter.
3. Make a list of all the words.
4. Make an entry box for name.
5. Shuffle all the words and store it in a list.
6. Use random method and pick five words to play.
7. Take input for the attempts from the user.
8. Print the correct words after each attempt.
9. Display the total score after the game.
10. Exit

Code :

```
from tkinter import *
root = Tk()
import random
import os
wordlst =
['good','self','book','basket','mouse','laptop','television','phone','mobile','chair','guitar'
,'hide','towel','door','football','switch','mirror','bed','sofa','teddy','rubber','bulb','shirt','
pant','shoes','socks','child','mother','father','clean','best','photo','frame','time','pass','g
eneral','search','type','join','leave','hand','raise','screen','share','class','clock','box','priz
e','bottle','flower','pen','save','project','game','call','message','stress']
global lst
lst=[]
lst= wordlst
rlst=[]
rword=[]
for i in lst:
  rs=''.join(random.sample(i , len(i)))
  rlst.append(rs)
rwind = []
for i in range(0,5):
  rwind.append(random.randint(0,len(rlst)))
global a
a=[]

for i in range(0,5):
  a.append(int(rwind[i]))
def gamestart():
    global wordgame
```

```
    wordgame=[]
    for i in range(0,5):
        b= int(rwind[i])
        wordgame.append(rlst[b])
root.geometry("300x300")
ename = Entry(root,borderwidth='10')
ename.get()
ename.pack(pady="20")
ename.insert(0, "enter your name here")
def namee():
    global userr
    userr=ename.get()
    ename.delete(0,'end')
    global gamewindow
    gamewindow = Toplevel(root)
    gamewindow.title("THe game begins")
    gamewindow.geometry("600x700")
    gamestart()
    global ewordbutton1
    ewordbutton1 = Button(gamewindow, command=ewordget1 , width='30',
text='next')
    global ewordbutton2
    ewordbutton2 = Button(gamewindow, command=ewordget2 , width='30',
text='next')
    global ewordbutton3
    ewordbutton3 = Button(gamewindow, command=ewordget3 , width='30',
text='next')
    global ewordbutton4
    ewordbutton4 = Button(gamewindow, command=ewordget4 , width='30',
text='next')
    global ewordbutton5
```

```
    ewordbutton5 = Button(gamewindow, command=ewordget5 , width='30',
text='finish')
    word1()
ebutton=Button(root , text="start the game" ,command=namee)
ebutton.pack()
def word1():
    word1lab = Label(gamewindow , text = wordgame[0])
    word1lab.pack(pady='5')
    global eword1
    eword1 = Entry(gamewindow, width='40')
    eword1.get()
    eword1.pack(pady='5')
    ewordbutton1.pack()
def word2():
    word2lab = Label(gamewindow, text=wordgame[1])
    word2lab.pack(pady='5')
    global eword2
    eword2 = Entry(gamewindow, width='40')
    eword2.get()
    eword2.pack(pady='5')
    ewordbutton2.pack()
def word3():
    word3lab = Label(gamewindow, text=wordgame[2])
    word3lab.pack(pady='5')
    global eword3
    eword3 = Entry(gamewindow, width='40')
    eword3.get()
    eword3.pack(pady='5')
    ewordbutton3.pack()
def word4():
    word4lab = Label(gamewindow, text=wordgame[3])
```

```
    word4lab.pack(pady='5')
    global eword4
    eword4 = Entry(gamewindow, width='40')
    eword4.get()
    eword4.pack(pady='5')
    ewordbutton4.pack()
def word5():
    word5lab = Label(gamewindow , text = wordgame[4])
    word5lab.pack(pady='5')
    global eword5
    eword5 = Entry(gamewindow, width='40')
    eword5.get()
    eword5.pack(pady='5')
    ewordbutton5.pack()
def ewordget1():
    global answer
    answer=[]
    temp1=a[0]
    answer.append(eword1.get())
    global count
    count = 0
    if(answer[0]==lst[temp1]):
        count+=1
        word1label = Label(gamewindow , text = "correct answer")
        word1label.pack()
    else:
        word2label = Label(gamewindow , text = "not the correct answer \n the
correct word is " + lst[temp1])
        word2label.pack()
    word2()
def ewordget2():
```

```
    global count
    temp1=a[1]
    a2 = eword2.get()
    if(a2==lst[temp1]):
        count+=1
        word3label = Label(gamewindow , text = "correct answer")
        word3label.pack()
    else:
        word4label = Label(gamewindow , text = "not the correct answer \n the
correct word is " + lst[temp1])
        word4label.pack()
    word3()
def ewordget3():
    global count
    temp1=a[2]
    a3 = eword3.get()
    if(a3==lst[temp1]):
        count+=1
        word1label = Label(gamewindow , text = "correct answer")
        word1label.pack()
    else:
        word2label = Label(gamewindow , text = "not the correct answer \n the
correct word is " + lst[temp1])
        word2label.pack()
    word4()
def ewordget4():
    global count
    temp1=a[3]
    a4 = eword4.get()
    if(a4==lst[temp1]):
        count+=1
```

```
        word1label = Label(gamewindow , text = "correct answer")
        word1label.pack()
    else:
        word2label = Label(gamewindow , text = "not the correct answer \n the
correct word is " + lst[temp1])
        word2label.pack()
    word5()
def ewordget5():
    global count
    temp1=a[4]
    a5 = eword5.get()
    if(a5==lst[temp1]):
        count+=1
        word1label = Label(gamewindow , text = "correct answer")
        word1label.pack()
    else:
        word2label = Label(gamewindow , text = "not the correct answer \n the
correct word is " + lst[temp1])
        word2label.pack()
    resultlabel = Label(gamewindow , text = "you finished the game, your score is "
+ str(count))
    resultlabel.pack()
root.mainloop()
```

Chapter 4: Output Analysis

Fig 1: The first box to enter name and start the game

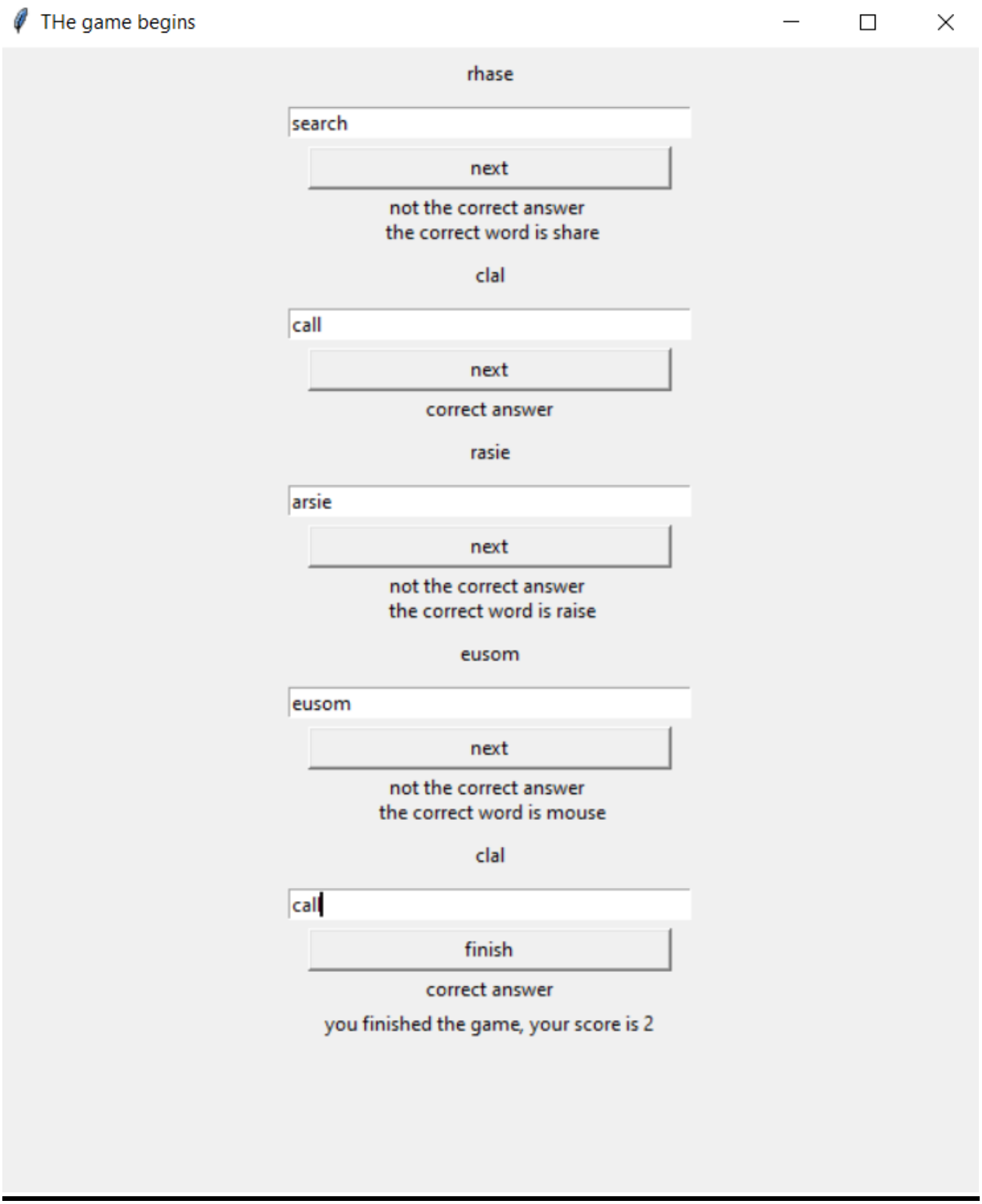


Fig 2: The game and the final score