PROJECT REPORT

Name – Arunima Garg

Registration number – 17BCE0844

Question:-

Create a GUI game in which whenever game starts, the game gives 10 randomly selected TV show names and jumbled, the user has to predict the correct name of the TV show 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. Hint – use random library in python.

Solution:-

ABSTRACT

Graphical User Interface (GUI) denotes a desktop application which helps users to interact with the computer. The libraries of python that are used for creating GUI are Kivy, Python Qt, wxPython and Tkinter.

In this project, the library used is PyQt. Out of all the python libraries, PyQt is most commonly used. Functions and events are also used. Creating a PyQt involves importing the module PyQt, creating the main window (container), adding any number of widgets to the main window and applying the event Trigger on the widgets.

INTRODUCTION

Python is used in GUI apps. PyQt is used to build GUI applications. The following application allows the player to play the game as many times as he wants. In a single game, he is given a set of any random ten questions in which each and every question will be the name of a TV show and will be jumbled each time the user plays a new game. At each correct guess he will be awarded one point. The scores are also being displayed.

METHODOLOGY

Concepts being used-

Creating the GUI involves importing the module PyQt, creating the main window (container), adding any number of widgets to the main window and applying the event Trigger on the widgets.

Code being implemented-

```
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'gui.ui'
#
# Created by: PyQt5 UI code generator 5.5.1
#
# WARNING! All changes made in this file will be lost!

from PyQt5 import QtCore, QtGui, QtWidgets
import sys
import random
import time
localScore = 0
cnt = 0
org = []
jum = []

class Ui_Game(object):
        def __init__(self,parent=None):
                        super(Ui_Game,self).__init__()
                        global org, jum
                        i=0
                        a = ['sherlock', 'threshold', 'fame', 'friends', 'office', 'flash', 'Lost',
'bigbang', 'dexter', 'suits', 'madmen',
'breathe','atlanta','StrangerThings','haunting','homecoming','maniac','westworld','queereye','Bi
```

```
gMouth','pose','AmericanVandal','JessicaJones','arrow','counterpart','WalkingDead','hatim','po
kemon','ninja','doremon']

                while(i<10):
                    pick = random.choice(a)
                    if pick not in org:
                        org.append(pick)
                        i+=1
                for i in range(len(org)):
                    l_word = random.sample(org[i],len(org[i]))
                    jumbled = ''.join(l_word)
                    jum.append(jumbled)


    def setupUi(self, Game):
        Game.setObjectName("Game")
        Game.resize(565, 287)
        self.Heading = QtWidgets.QLabel(Game)
        self.Heading.setGeometry(QtCore.QRect(190, 20, 181, 51))
        self.Heading.setStyleSheet("font: 75 14pt \"Ubuntu\";")
        self.Heading.setObjectName("Heading")
        self.Score = QtWidgets.QLabel(Game)
        self.Score.setGeometry(QtCore.QRect(220, 190, 131, 17))
        self.Score.setObjectName("Score")
        self.label_4 = QtWidgets.QLabel(Game)
        self.label_4.setGeometry(QtCore.QRect(20, 110, 111, 71))
        self.label_4.setObjectName("label_4")
        self.label_5 = QtWidgets.QLabel(Game)
        self.label_5.setGeometry(QtCore.QRect(20, 90, 111, 17))
```

```
        self.label_5.setObjectName("label_5")
        self.jum_word = QtWidgets.QLabel(Game)
        self.jum_word.setGeometry(QtCore.QRect(160, 90, 151, 17))
        self.jum_word.setObjectName("jum_word")
        self.submit = QtWidgets.QPushButton(Game)
        self.submit.setGeometry(QtCore.QRect(400, 130, 81, 27))
        self.submit.setObjectName("submit")
        self.plainTextEdit = QtWidgets.QPlainTextEdit(Game)
        self.plainTextEdit.setGeometry(QtCore.QRect(160, 130, 221, 31))
        self.plainTextEdit.setObjectName("plainTextEdit")

        self.retranslateUi(Game)
        QtCore.QMetaObject.connectSlotsByName(Game)

        self.retranslateUi(Game)
        QtCore.QMetaObject.connectSlotsByName(Game)

        self.submit.clicked.connect(self.myfunc)

def retranslateUi(self, Game):
        global jum
        _translate = QtCore.QCoreApplication.translate
        Game.setWindowTitle(_translate("Game", "Game"))
        self.Heading.setText(_translate("Game", "Jumble Word Game"))
        self.Score.setText(_translate("Game", ""))
        self.label_4.setText(_translate("Game", "Original Word : "))
        self.label_5.setText(_translate("Game", "Jumbled Word :"))
```

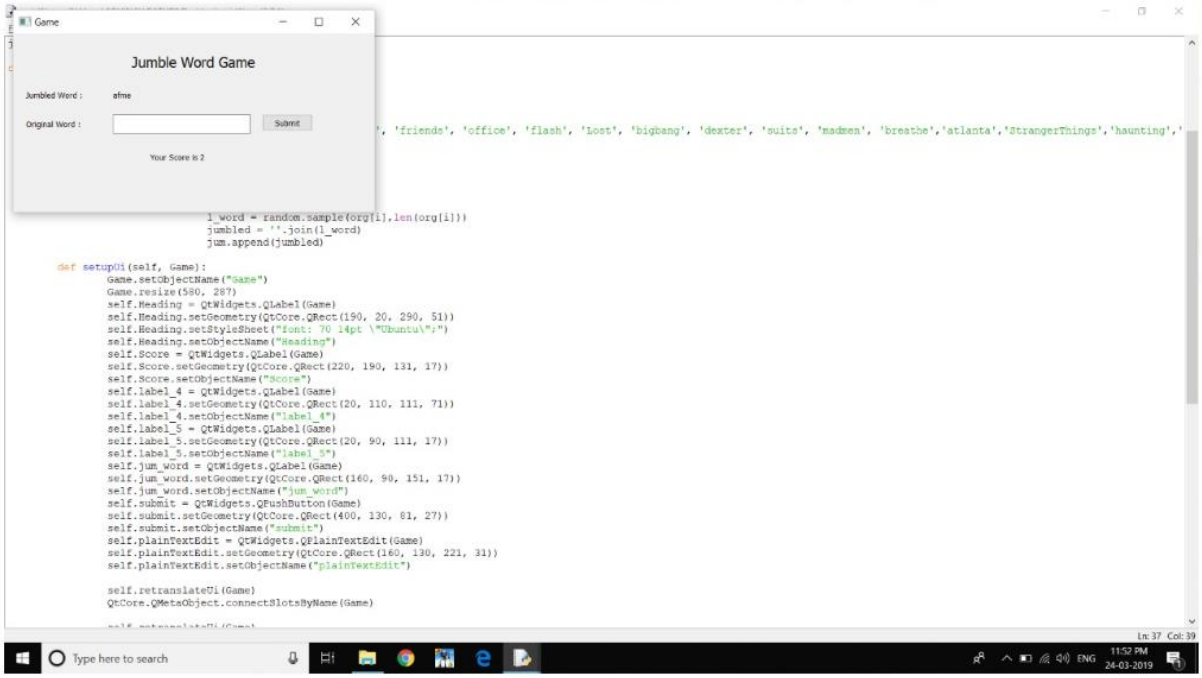
```
            self.jum_word.setText(_translate("Game", "Null"))
            self.submit.setText(_translate("Game", "Submit"))
            self.jum_word.setText(jum[0])

    def myfunc(self):
            global cnt, jum, org, localScore
            data = self.plainTextEdit.toPlainText()
            if(data==org[cnt]):
                    localScore+=1
            cnt+=1
            if(cnt!=10):
                    self.jum_word.setText(jum[cnt])
                    self.plainTextEdit.setPlainText("")
            else:
                    self.Score.setText("Your Score is " + str(localScore))
                    #time.sleep(5)
                    #sys.exit()


app = QtWidgets.QApplication(sys.argv)
MainWindow = QtWidgets.QMainWindow()
ui = Ui_Game()
ui.setupUi(MainWindow)
MainWindow.show()
sys.exit(app.exec_())
```

RESULT

The program has executed successfully.



Difficulties faced –

- It was difficult to use the PyQt module.

CONCLUSION

The PyQt framework was used to implement the GUI. This project allows a player to guess the correct names of TV shows which are displayed after jumbling their letters. On each correct guess they are awarded one point. The score is displayed to the user after they answer ten questions. They can play it as many times as they want.

REFERENCES

- https://geeksforgeeks.com
- https://www.tutorialspoint.com/python/python_gui_programming.htm
- https://www.youtube.com/