15IT322E

Python Programming

(Project)

Name : Shreya Sarkar

Reg. no. : RA1711003010716

Batch : 2

Topic : Color Game using Tkinter

Tkinter (Python GUI)

Python offers multiple options for developing Graphical User Interface (GUI). Out of all the GUI methods, tkinter is most commonly used method. 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. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python.

To create a Tkinter, following steps are to be followed.

- Importing the module – Tkinter
- Create the main window (container)
- Add any number of widgets to the main window
- Apply the event Trigger on the widgets.

There are two main methods to create any Python application using GUI. They are as follows.

- Tk(screenName=None, baseName=None, className='Tk', useTk=1): To create a main window, tkinter offers a method 'Tk(screenName=None, baseName=None, className='Tk', useTk=1)'. To change the name of the window, you can change the className to the desired one.

  The basic code is: m=Tkinter.Tk()

- Mainloop(): There is a method known by the name mainloop() is used when you are ready for the application to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event till the window is not closed.

  The basic code is: m=mainloop()

Tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes' classes.

- pack() method: It organizes the widgets in blocks before placing in the parent widget.
- grid() method: It organizes the widgets in grid (table-like structure) before placing in the parent widget.
- place() method: It organizes the widgets by placing them on specific positions directed by the programmer.

There are a number of widgets which you can put in your tkinter application. Some of them are follows.

- Button: To add a button in your application, this widget is used.
  The general syntax is: w= Button(master, option=value)
- Canvas: It is used to draw pictures and other complex layout like graphics, text and widgets.
  The general syntax is: w= Canvas(master, option=value)
- CheckButton: To select any number of options by displaying a number of options to a user as toggle buttons.
  The general syntax is: w= CheckButton(master, option=value)
- Entry: It is used to input the single line text entry from the user. For multi-line text input, Text widget is used.
  The general syntax is: w= Entry(master, option=value)
- Frame: It acts as a container to hold the widgets. It is used for grouping and organizing the widgets. The general syntax is: w= Frame(master, option=value)
- Label: It refers to the display box where you can put any text or image which can be updated any time as per the code.
  The general syntax is: w= Label(master, option=value)

- Listbox: It offers a list to the user from which the user can accept any number of options.
  The general syntax is: w= Listbox(master, option=value)
- MenuButton: It is a part of top-down menu which stays on the window all the time. Every menubutton has its own functionality.

  The general syntax is: w= MenuButton(master, option=value)

- Menu: It is used to create all kinds of menus used by the application.
  The general syntax is: w= Menu(master, option=value)
- Message: It refers to the multi-line and non-editable text. It works same as that of Label.
  The general syntax is: w= Message(master, option=value)
- RadioButton: It is used to offer multi-choice option to the user. It offers several options to the user and the user has to choose one option.
  The general syntax is: w= RadioButton(master, option=value)
- Scale: It is used to provide a graphical slider that allows selecting any value from that scale. The general syntax is: w= Scale(master, option=value)

- Scrollbar: It refers to the slide controller which will be used to implement listed widgets.
  The general syntax is: w= Scrollbar(master, option=value)
- Text: To edit a multi-line text and format the way it has to be displayed.
  The general syntax is: w= Text(master, option=value)
- TopLevel: This widget is directly controlled by the window manager. It don't need any parent window to work on.The general syntax is: w=TopLevel(master, option=value)
- SpinBox: It is an entry of 'Entry' widget. Here, value can be input by selecting a fixed value of numbers.The general syntax is: w= SpinBox(master, option=value)
- PannedWindow: It is a container widget which is used to handle number of panes arranged in it. The general syntax is: w= PannedWindow(master, option=value)

Color Game using Tkinter

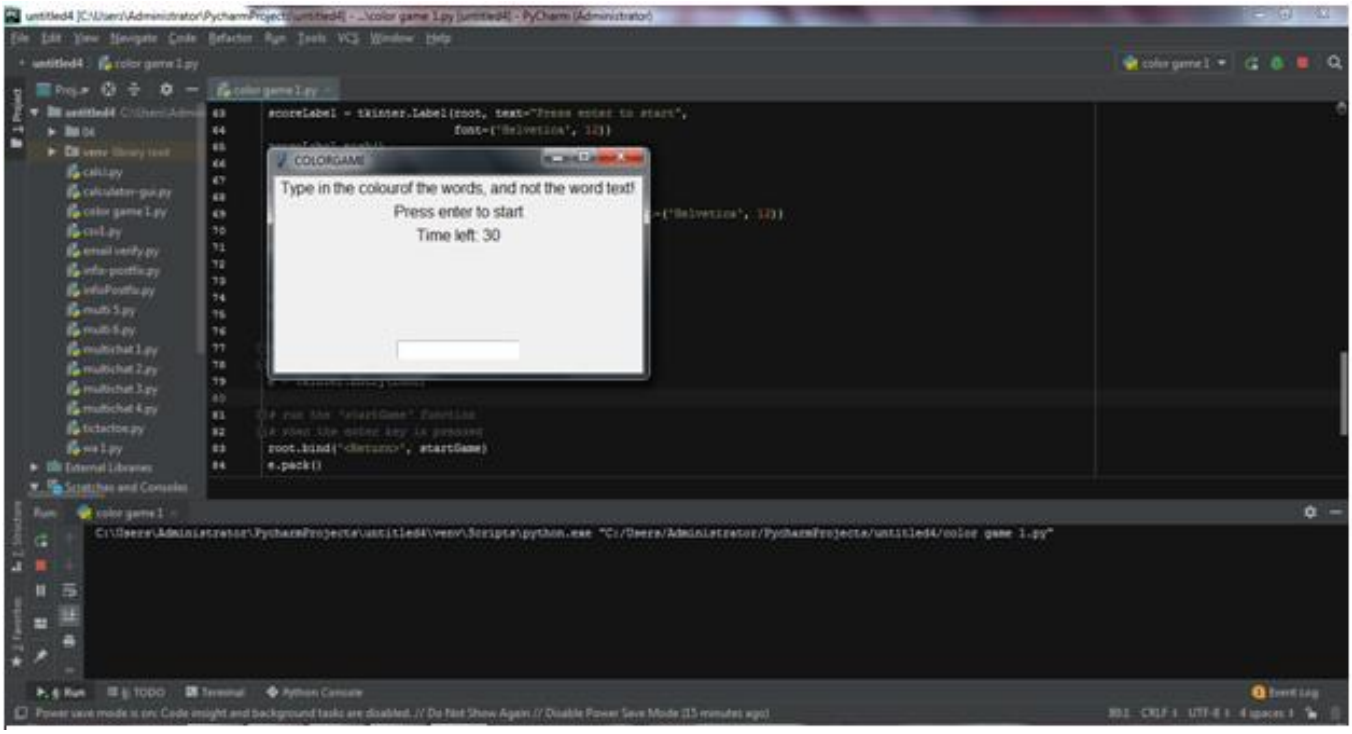
In this game player has to enter color of the word that appears on the screen and hence the score increases by one, the total time to play this game is 30 seconds. Colors used in this game are Red, Blue, Green, Pink, Black, Yellow, Orange, White, Purple and Brown. Interface will display name of different colors in different colors. Player has to identify the color and enter the correct color name to win the game.

Code:

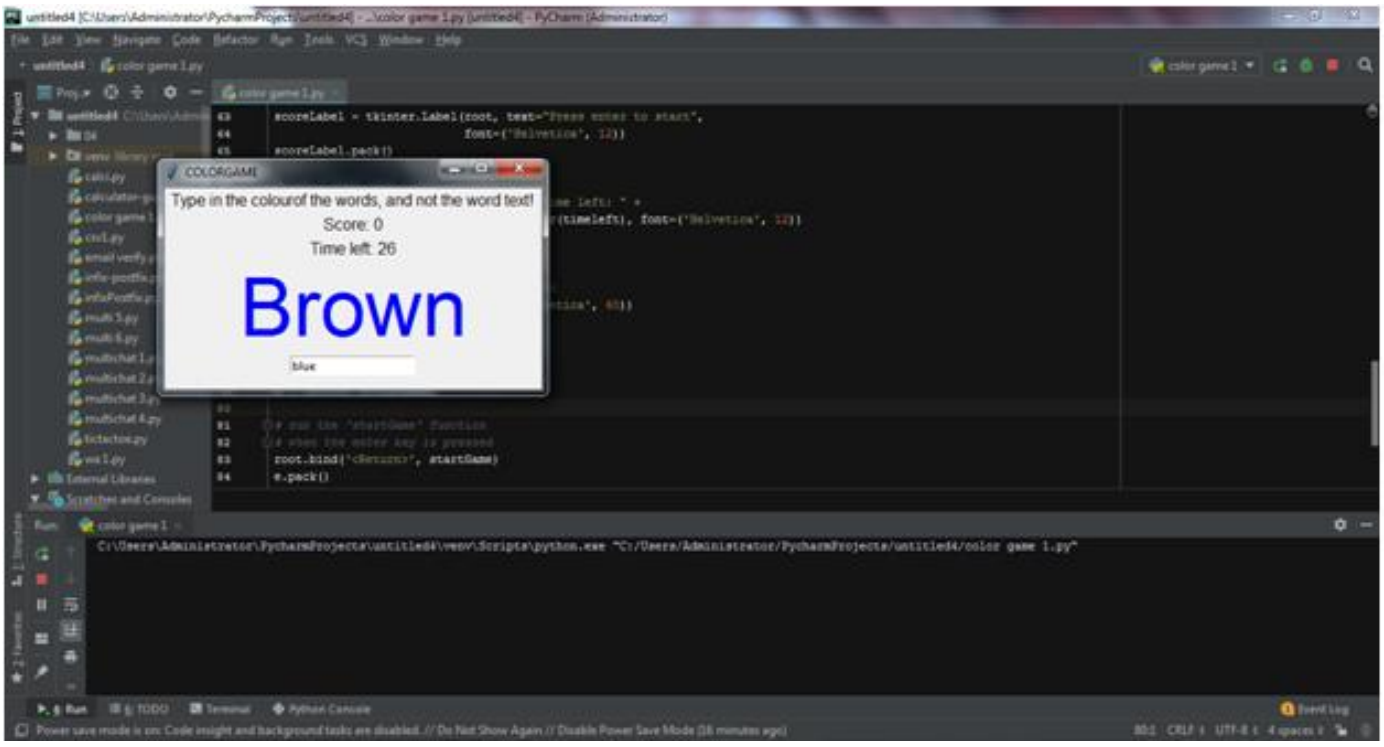
```
import tkinter
import random

#list of possible colour.
colours = ['Red', 'Blue', 'Green', 'Pink', 'Black',
           'Yellow', 'Orange', 'White', 'Purple', 'Brown']
score = 0
timeleft = 30 #the game time left, initially 30 seconds

def startGame(event):
    if timeleft == 30:
        countdown()  #start the timer
    nextColour()

def nextColour():
    global score
    global timeleft

    if timeleft > 0:
        e.focus_set()
        if e.get().lower() == colours[1].lower():
            score += 1
        e.delete(0, tkinter.END) #clear the text entry box
        random.shuffle(colours)
        #change the colour to type, by changing the text _and_ the colour to a random
colour value
        label.config(fg=str(colours[1]), text=str(colours[0]))
        scoreLabel.config(text="Score: " + str(score)) #updating the score

def countdown():
    global timeleft

    # if a game is in play
    if timeleft > 0:
```

```
        # decrement the timer.
        timeleft -= 1

        # update the time left label
        timeLabel.config(text="Time left: "
                               + str(timeleft))

        # run the function again after 1 second.
        timeLabel.after(1000, countdown)

    # Driver Code


# create a GUI window
root = tkinter.Tk()

# set the title
root.title("COLORGAME")

# set the size
root.geometry("375x200")

# add an instructions label
instructions = tkinter.Label(root, text="Type in the colour"
                                        "of the words, and not the word text!",
                             font=('Helvetica', 12))
instructions.pack()

# add a score label
scoreLabel = tkinter.Label(root, text="Press enter to start",
                           font=('Helvetica', 12))
scoreLabel.pack()

# add a time left label
timeLabel = tkinter.Label(root, text="Time left: " +
                                     str(timeleft), font=('Helvetica', 12))

timeLabel.pack()

# add a label for displaying the colours
label = tkinter.Label(root, font=('Helvetica', 60))
label.pack()

# add a text entry box for
# typing in colours
e = tkinter.Entry(root)

# run the 'startGame' function
# when the enter key is pressed
root.bind('<Return>', startGame)
e.pack()

# set focus on the entry box
e.focus_set()

# start the GUI
root.mainloop()
```

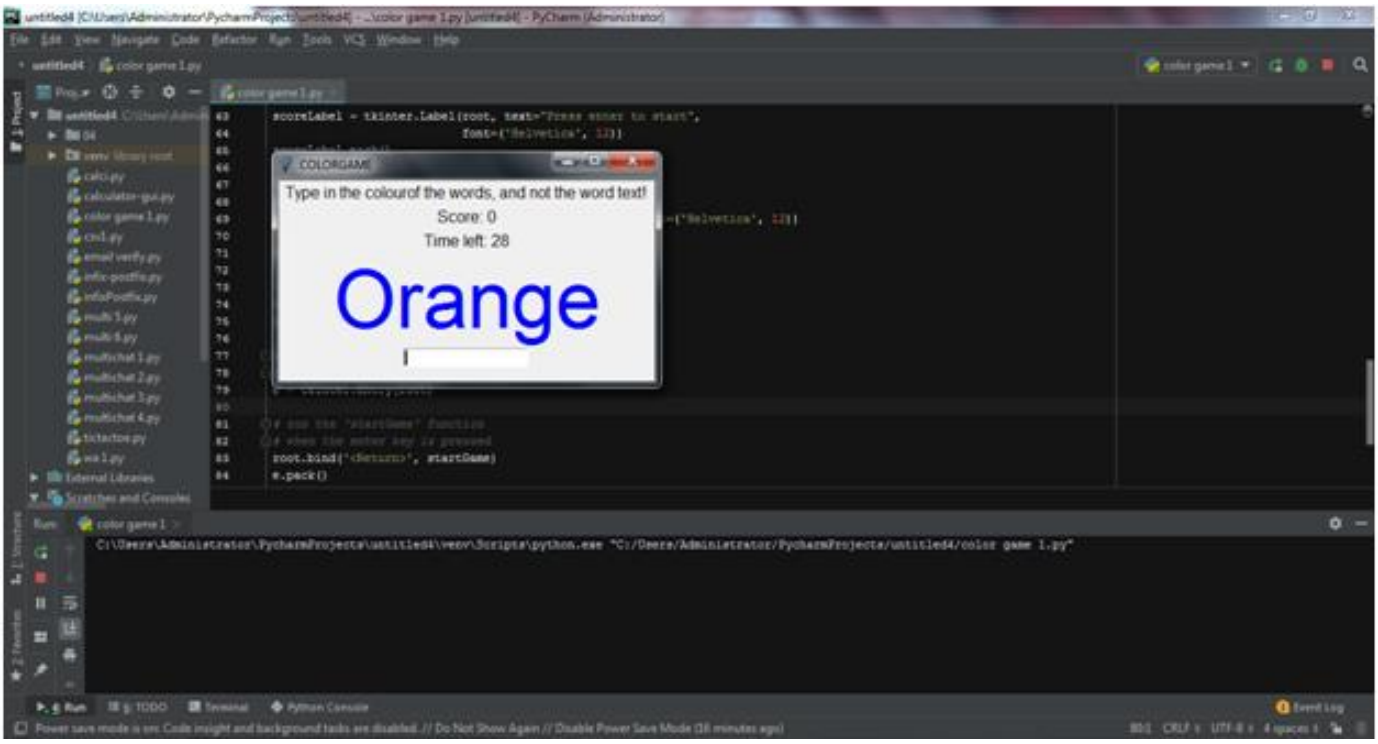
Outputs:



A dialog box appears. Color names with different colors appear after enter is pressed. Enter is pressed to start the game. Since, in the code timeleft is initialized as 30 seconds, the game will end in 30 seconds.



When the game starts, color names start appearing with different colours. If the colour name is entered correctly, 1 point will be gained in score. And if the name which is entered is not correct, then the score is not increased.


COLORGAME
Type in the colourof the words, and not the word text!
Score: 0
Time left: 28
Orange


COLORGAME
Type in the colourof the words, and not the word text!
Score: 11
Time left: 0
Pink

In the example, the word ‘Orange’ appears in Blue color. If Orange is entered in the space provided, no score is gained. To gain points, player has to enter blue in the space provided. The only thing player has to keep in mind is that he has only 30 seconds to play. After 30 seconds the game will end automatically. Once the timeleft is zero second, no color name can be entered in the space provided.