



Python Programming Lab Report

On

DISPLAYING GIFS IN TKINTER

Using Python

Submitted

In partial fulfillment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS & COMMUNICATION ENGINEERING

By

U.Pooja

21311A04L7

Under the Supervision of

E. Lavanya

Assistant professor, E.C.E Department



Department Of Electronics & Communication Engineering

Sreenidhi institute of Science & Technology

(Autonomous)

Yamnapet, Ghatkesar, R.R district, Hyderabad – 501301(Affiliated to JNT University
Hyderabad, Hyderabad and Approved by AICTE - New Delhi)



SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY

Yamnapet, Ghatkesar, Hyderabad - 501301

CERTIFICATE

Date: 12/01/2024

This is to certify that the **Python Programming Lab Report** entitled “**DISPLAYING GIFS IN TKINTER**” being submitted by **U.Pooja-21311A04L7** in partial fulfillment for the award of **Bachelor of Technology** degree in **Electronics & Communications Engineering** to Sreenidhi Institute of Science and Technology, Yamnapet, Ghatkesar [Telangana], is a report of review work carried out by him during academic year **2023-2024** under our guidance and supervision.

E. Lavanya
Assistant Professor
ECE Department

Dr. S. P. V. Subba Rao
Head of Department
ECE Department

ACKNOWLEDGEMENT

I hereby declare that the work described in the Python Programming Lab Report, entitled “**DISPLAYING GIFS IN TKINTER using Python**” which is being submitted by us in partial fulfillment for the award of **Bachelor of Technology** in the Dept. of **Electronics & Communication Engineering, Sreenidhi Institute Of Science & Technology** affiliated to Jawaharlal Nehru Technological University Hyderabad, Kukatpally, Hyderabad (Telangana) is the work on our own effort and has not been submitted elsewhere.

We are very thankful to **E. Lavanya**, Assistant professor, ECE Department., Sreenidhi Institute of Science and Technology, Ghatkesar for providing an initiative to this technical seminar and giving valuable timely suggestions over the work.

We convey our sincere thanks to **Dr. S.P.V. Subba Rao**, Head of the Department (ECE), Sreenidhi Institute of Science and Technology, Ghatkesar, for his kind cooperation in the completion of this work.

We even convey our sincere thanks to Director and Principal, Sreenidhi Institute of Science and Technology, Ghatkesar for their kind cooperation in the completion of the technical seminar.

Finally, we extend our sense of gratitude to all our friends, teaching and nonteaching faculty, who directly or indirectly helped us in this endeavor.

Place: Hyderabad

Signature

Name of the Candidate: U.Pooja

Roll No: **21311A04L7**

CONTENTS

1. Abstract
2. Introduction
3. Source Code
4. Output
5. Conclusion

ABSTRACT

Tkinter, derived from "Tk interface," is Python's standard GUI (Graphical User Interface) library. It wraps the efficient Tk GUI toolkit, which has been a significant part of the programming world since its inception in the early 1990s. Over the years, Tkinter has evolved alongside Python, growing from a simple tool for creating small graphical interfaces to a fully-fledged library capable of crafting complex and attractive applications.

GIF stands for Graphics Interchange Format. It was developed in mid-1987. Once the use of GIF was considered unprofessional and not of much use. Today it is widely used in creating a lot of moving images in the world of graphic content creation. This is because they quickly catch the eye and are easily sharable. Unlike the JPEG format, the gif format does not lose its quality. It uses lossless compression to preserve the image quality and keep it intact.

If you want to see an animated gif in tkinter use this script `import tkinter as tk from PIL import Image, ImageTk from itertools import count, cycle class "ImageLabel(tk.Label):" A Label that displays images, and plays them if they are gifs :im: A PIL Image instance or a string filename. Adding animation to your GUIs can make them more attractive and engaging for the users, and with Tkinter and PIL, it's a simple process to achieve.`

Assistant Professor

E.LAVANYA

Name of the student:

U.POOJA

21311A04L7

INTRODUCTION

Gif can be described as just a collection of frames(images) that are shown in a sequence, in a loop, thus creating an animation effect. Tkinter is a Python library for creating graphical user interfaces (GUIs). Tkinter supports displaying GIF images, but not the animation. To show a GIF animation in Tkinter, you need to use some additional techniques. Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. It is a thin object-oriented layer on top of Tcl/Tk.

Tkinter is not the only GuiProgramming toolkit for Python. It is however the most commonly used one. CameronLaird calls the yearly decision to keep TkInter "one of the minor traditions of the Python world." Graphical User Interface(GUI) is a form of user interface which allows users to interact with computers through visual indicators using items such as icons, menus, windows, etc. It has advantages over the Command Line Interface(CLI) where users interact with computers by writing commands using keyboard only and whose usage is more difficult than GUI. Tkinter is the inbuilt python module that is used to create GUI applications. It is one of the most commonly used modules for creating GUI applications in Python as it is simple and easy to work with. You don't need to worry about the installation of the Tkinter module separately as it comes with Python already. It gives an object-oriented interface to the Tk GUI toolkit. Widgets in Tkinter are the elements of GUI application which provides various controls (such as Labels, Buttons, ComboBoxes, CheckBoxes, MenuBars, RadioButtons and many more) to users to interact with the application. Fundamental structure of tkinter program.

Creating windows and dialog boxes: Tkinter can be used to create windows and dialog boxes that allow users to interact with your program. These can be used to display information, gather input, or present options to the user. Building a GUI for a desktop application: Tkinter can be used to create the interface for a desktop application, including buttons, menus, and other interactive elements.

Adding a GUI to a command-line program: Tkinter can be used to add a GUI to a command-line program, making it easier for users to interact with the program and input arguments.

Creating custom widgets: Tkinter includes a variety of built-in widgets, such as buttons, labels, and text boxes, but it also allows you to create your own custom widgets.

Key Features:

Import required packages:

Here, we will use tkinter for GUI and pillow module for opening an image.

```
import tkinter as tk  
  
from PIL import Image
```

Create the main application window and run the event loop:

```
root = tk.Tk()  
  
root.title("Displaing Gif")  
  
  
  
root.mainloop()
```

Add labels and buttons:

```
gif_label = tk.Label(root, image="")  
  
gif_label.pack()  
  
start = tk.Button(root, text="Start", command = lambda:  
animation(current_frame = 0))  
  
start.pack()  
  
stop = tk.Button(root, text="Stop", commnad = stop_animation)
```

```
stop.pack()
```

In `gif_label` gif will be displayed. And we have a `start` and `stop` button to control the gif.

Store gif's path in the variable and open it using the pillow module:

```
file = "gif_file.gif"

info = Image.open(file)
```

Get the number of frames in the Gif and then create photoimage object for each frame and store them in a list:

```
frames = info.n_frames # number of frames

photoimage_objects = []

for i in range(frames):

    obj = tk.PhotoImage(file = file, format = f'gif -index {i}')

    photoimage_objects.append(obj)
```

format parameter in the photoimage class is required to read the frames of the gif.

Now since we have all the frames as photoimage objects, we can display them in the image label.

Create functions: animation and stop animation to start and stop the gif respectively:


```
def animation(current_frame=0):  
  
    global loop  
  
    image = photoimage_objects[current_frame]  
  
    gif_label.configure(image = image)  
  
    current_frame = current_frame + 1  
  
    if current_frame == frames:  
        current_frame = 0 # reset the current_frame to 0 when end is reached  
  
    loop = root.after(50, lambda: animation(current_frame))
```

The above animation function takes `current_frame` as an argument, which indicates the frame number that is going to be displayed.

So to display gif we have to start showing frames from the 0th index.

And to continuously update the frame we have to keep calling the function in a loop so we used `root.after()` which calls the function after every 50ms.

Create stop animation function:

```
def stop_animation():  
    root.after_cancel(loop)
```

The above animation function takes `current_frame` as an argument, which indicates the frame number that is going to be displayed.

So to display gif we have to start showing frames from the 0th index.

And to continuously update the frame we have to keep calling the function in a loop so we used `root.after()` which calls the function after every 50ms.

SOURCE CODE

```
import tkinter as tk  
from PIL import Image  
  
root = tk.Tk()  
file="name_of_gif.gif"  
  
info = Image.open(file)  
  
frames = info.n_frames # gives total number of frames that gif contains  
  
#creating list of PhotoImage objects for each frames  
im = [tk.PhotoImage(file=file,format=f'gif -index {i}') for i in range(frames)]  
  
count = 0  
anim = None  
def animation(count):  
    global anim  
    im2 = im[count]  
  
    gif_label.configure(image=im2)  
    count += 1
```

```
if count == frames:
    count = 0
anim = root.after(50,lambda :animation(count))

def stop_animation():
    root.after_cancel(anim)

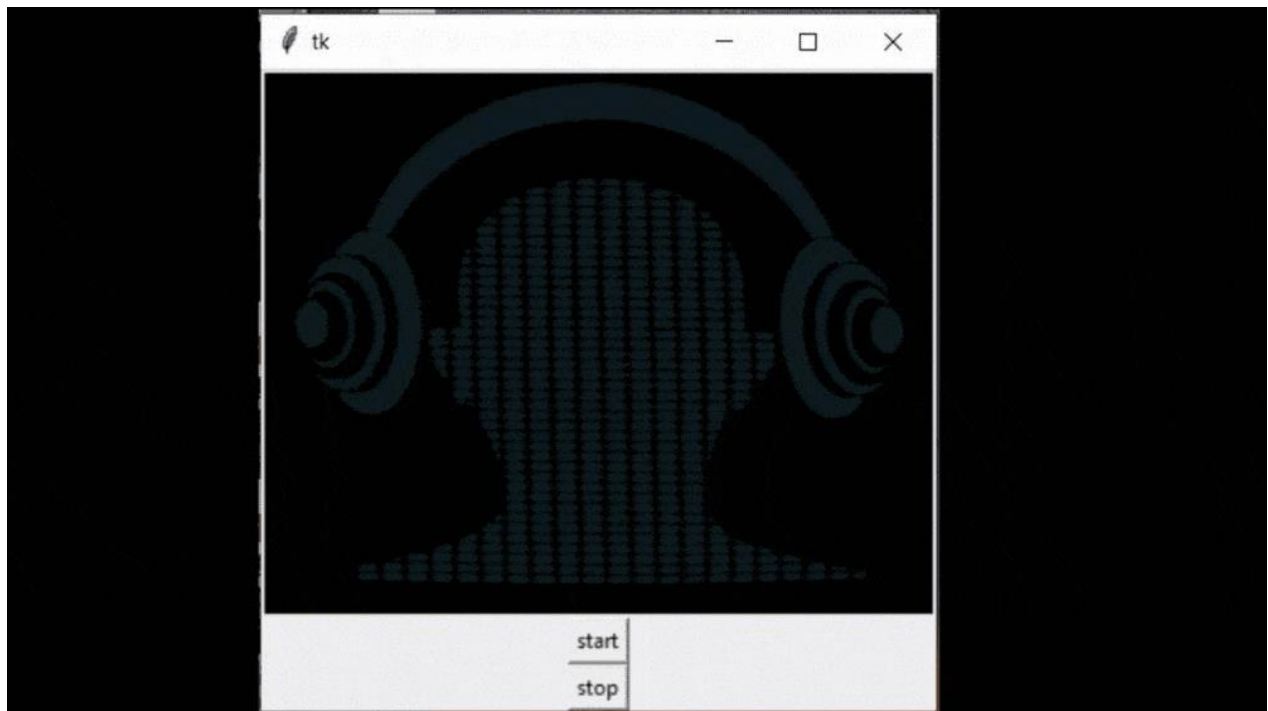
gif_label = tk.Label(root,image="")
gif_label.pack()

start = tk.Button(root,text="start",command=lambda :animation(count))
start.pack()

stop = tk.Button(root,text="stop",command=stop_animation)
stop.pack()

root.mainloop()
```

OUTPUT



CONCLUSION

In this article, we have shown you how to easily display and play a GIF file in Python Tkinter GUIs. Adding animation to your GUIs can make them more attractive and engaging for the users, and with Tkinter and PIL, it's a simple process to achieve. With Tkinter's built-in support for GIF format and its ability to manipulate widgets and graphics, creating animated applications in Python has never been easier. Whether you're creating a simple game or adding some visual flair to your user interface, Tkinter's animation capabilities are a powerful tool for any Python developer.