

PAINT APP USING TKINTER



SUBMITTED TO:

BHAVESH

MATHUR

SUBMITTED BY:

PRIYA SINHA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MANGALMAY INSTITUTE OF ENGINEERING AND TECHNOLOGY

GREATER NOIDA-201310,UTTAR PRADESH,INDIA

JANUARY 2021

TABLE OF CONTENTS

S.NO	CONTENTS	PAGE NO.
1.	ABSTRACT	III
2.	INTRODUCTION	III
3.	FEASIBILITY AND STUDY <ul style="list-style-type: none"> • FEASIBILITY STUDY • TECHNICAL FEASIBILITY • ECONOMICAL FEASIBILITY • OPERATIONAL FEASIBILITY • SCHEDULING FEASIBILITY 	IV
4.	DESIGNING <ul style="list-style-type: none"> • ER DIGRAM • USE CASE DIGRAM 	IV-V
5.	IMPLEMENTATION <ul style="list-style-type: none"> • WORKING 	VI-VIII
6.	TESTING	VIII
7.	TECHNOLOGIES AND TOOLS USED: <ul style="list-style-type: none"> • PYTHON 	VIII
8.	SCREEN SHOTS	IX
9.	REFRENCES	IX

PROJECT REPORT

ABSTRACT:

This project report study presents a paint application using tkinter in Python. It is a graphical user interface that allows the user to stimulate painting on the computer by using mouse. The theoretical part of project is reviewed. Conditions for an effective implementation are investigated. This report describes the design implementation and evaluation of a GUI application. Organizational as well as technical work is involved in it. The result of the implementation has been achieved.

INTRODUCTION:

This Mini Project includes “PAINT APPLICATION USING TKINTER” IN Python. A paint program is a software graphics that allows the users to draw or paint bitmapped images on a computer.

Tkinter package is the standard Python interface which is Graphical User Interface (GUI).Python when combined with Tkinter provides a fast and easy way to create a GUI application.

Graphical User Interface is a form of user interface that allow users to interface with electronic devices through graphical icons and visual indicators. The main advantage is -A user can easily navigate to the system without knowing a lot of details. Easy setup and ready to start working are awesome. Most of the software hides the complexity of actions from the users and display only required information is key to good interface. Various example of GUI is Notepad, Calculator etc.

This paint application uses tkinter for creating GUI as it makes easier for the person who does not belongs from the coding background can easily interact with the window. It captures attention and keeps it. In this project various modules are used. Firstly, we import tkinter module from Python and from tkinter we import more modules like, *, Scale, message box, file dialog. Various widgets are also used in it like, Button, LabelFrame, and Canvas. Widgets contain various option which are mainly common in all -bg , fg , font ,bd ,text ,relief ,width. All this together form this application of paint.

FEASIBILITY STUDY:

Our project “PAINT APPLICATION USING TKINTER“ is legally and technically feasible as well as economically justifiable. Feasibility of our project depends upon the pros and cons of our project. The detailed analysis of the practicality of project and whether the project is feasible or not depend upon the feasibility study:

- 1) **Technical Feasibility** —the paint application is technically feasible .The software and the hardware used are easily gets available to us which makes our application feasible and takes less time.
- 2) **Economic Feasibility** —the cost required to make this application is negligible. At all, it serves an independent project assessment and enhances the credibility of the project.
- 3) **Operational Feasibility** — the completion of the paint application project using tkinter meets the required output which makes the project operationally feasible.
- 4) **Scheduling Feasibility** —the project is completed on time and consulted with the guide .The scheduling feasibility of our project is alright.

DESIGNING:

Here presents the designing of the project “THE PAINT APPLICATION “which includes:

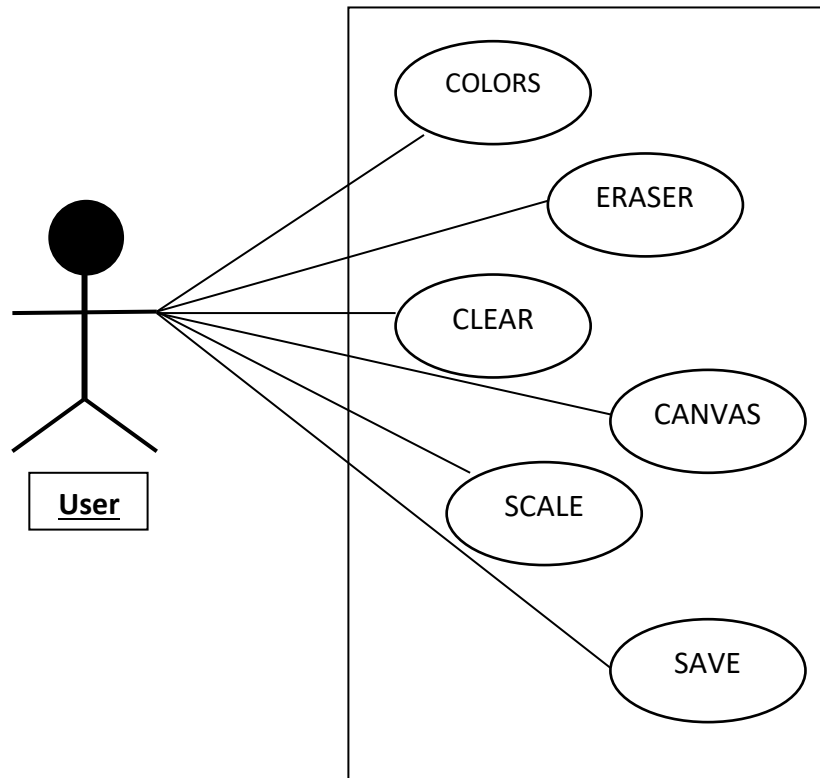
*Entity Relationship Diagram(ER diagram):- represents the connection between the entities present in the program of paint application using tkinter in Python.

*Flowchart:- represents the process and execution steps that are included in the application. Each step is mentioned in the following flowchart.

ER DIAGRAM:

Not applicable in this project “PAINT APPLICATION “ using tkinter in python as it has no database to represent entity and attributes.

Use case diagram:



IMPLEMENTATION:

Here is the implementation of the paint application using tkinter in python:

```
In [1]: import tkinter
from tkinter import *
from tkinter.ttk import Scale
from tkinter import colorchooser,filedialog,messagebox
import PIL.ImageGrab as ImageGrab
class Paint():
    def __init__(self,root):
        self.root=root
        self.root.title("Paint app")
        self.root.configure(background="white")

        self.pen_color="black"
        self.eraser_color="white"

        self.color_frame=LabelFrame(self.root,text="Color",font=('elephant',17),bd=5,relief=RAISED,fg='black',bg="white")
        self.color_frame.place(x=0,y=0,width=80,height=318)
        colors=["#ff0000","#ffa500","#008000","#0000ff","#800080","#9400d3","#835c38","#afeeee","#808080","#ffffff","#ffc0cb",
            "#800000","#8b008b","#00ffff","#800000","#808000","#a0522d","#000000","#9acd32","#ffff00"]

        i=j=0
        for color in colors:
            Button(self.color_frame,bg=color,bd=3,relief=RIDGE,width=3,
                command=lambda col=color:self.select_color(col)).grid(row=i,column=j)
            i+=1
        if i==10:
            i=0
            j+=1
```

Activate Windows
Go to Settings to activate Win

```
        i=j=0
        for color in colors:
            Button(self.color_frame,bg=color,bd=3,relief=RIDGE,width=3,
                command=lambda col=color:self.select_color(col)).grid(row=i,column=j)
            i+=1
        if i==10:
            i=0
            j+=1

        self.eraser_button=Button(self.root,text="ERASER",bd=5,bg="white",command=self.eraser,width=9,relief=RAISED)
        self.eraser_button.place(x=0,y=330)

        self.clear_button=Button(self.root, text = 'CLEAR',bd=5,bg="white",
            command=self.new_canvas,width=9,relief=RAISED)
        self.clear_button.place(x=0,y=360)

        self.save_button=Button(self.root, text = 'SAVE',bd=5,bg="white",command=self.save_paint,width=9,relief=RAISED)
        self.save_button.place(x=0,y=390)

        self.canvas_button=Button(self.root, text = 'CANVAS',bd=5,bg="white",command=self.canvas,width=9,relief=RAISED)
        self.canvas_button.place(x=0,y=420)

        self.pen_size_scale_frame=LabelFrame(self.root,text='Scale',font=('copper black',17,'bold'),bd=5,
            bg='white',relief=RIDGE)
        self.pen_size_scale_frame.place(x=0,y=470,width=80,height=200)
```

Activate Windows
Go to Settings to activate Win

```

self.canvas_button=Button(self.root, text = 'CANVAS',bd=5,bg="white",command=self.canvas,width=9,relief=RAISED)
self.canvas_button.place(x=0,y=420)

self.pen_size_scale_frame=LabelFrame(self.root,text='Scale',font=('copper black',17,'bold'),bd=5,
                                     bg='white',relief=RIDGE)
self.pen_size_scale_frame.place(x=0,y=470,width=80,height=200)

self.pen_size=Scale(self.pen_size_scale_frame,orient=VERTICAL,from_=50,to =0,length=170)
self.pen_size.set(1)
self.pen_size.grid(row=0,column=1,padx=15)

self.canvas=Canvas(self.root,bg='white',bd=10,relief=GROOVE,width=1250,height=670)
self.canvas.place(x=90,y=0)

#bind canvas with mouse
self.canvas.bind("<B1-Motion>",self.paint) #event,function

def paint(self,event):
    x1,y1=(event.x),(event.y)
    x2,y2=(event.x),(event.y)

    self.canvas.create_oval(x1,y1,x2,y2,fill=self.pen_color,outline=self.pen_color,width=self.pen_size.get())

def select_color(self,col):
    self.pen_color=col

def eraser(self):
    self.pen_color=self.eraser_color

```

Activate Windows
Go to Settings to activate

```

self.canvas.create_oval(x1,y1,x2,y2,fill=self.pen_color,outline=self.pen_color,width=self.pen_size.get())

def select_color(self,col):
    self.pen_color=col

def eraser(self):
    self.pen_color=self.eraser_color

def canvas(self):
    color=colorchooser.askcolor()
    self.canvas.configure(background=color[1])
    self.eraser_color=color[1]

def new_canvas(self):
    self.canvas.delete(ALL)

def save_paint(self):
    try:
        filename= filedialog.asksaveasfilename(defaultextension='.png')
        x=self.root.winfo_rootx()+self.canvas.winfo_x()
        y=self.root.winfo_rooty()+self.canvas.winfo_y()
        x1=x+self.canvas.winfo_width()
        y1=y+self.canvas.winfo_height()
        ImageGrab.grab().crop((x,y,x1,y1)).save(filename)
        messagebox.showinfo('paint says','image is saved' +str(filename))
    except:
        print('unable to save')

```

Activate Windows
Go to Settings to activate Windows

```

def canvas(self):
    color=colorchooser.askcolor()
    self.canvas.configure(background=color[1])
    self.eraser_color=color[1]

def new_canvas(self):
    self.canvas.delete(ALL)

def save_paint(self):
    try:
        filename= filedialog.asksaveasfilename(defaultextension='.png')
        x=self.root.winfo_rootx()+self.canvas.winfo_x()
        y=self.root.winfo_rooty()+self.canvas.winfo_y()
        x1=x+self.canvas.winfo_width()
        y1=y+self.canvas.winfo_height()
        ImageGrab.grab().crop((x,y,x1,y1)).save(filename)
        messagebox.showinfo('paint says','image is saved' +str(filename))
    except:
        print('unable to save')

if __name__=="__main__":
    root=Tk()
    p=Paint(root)
    root.mainloop()

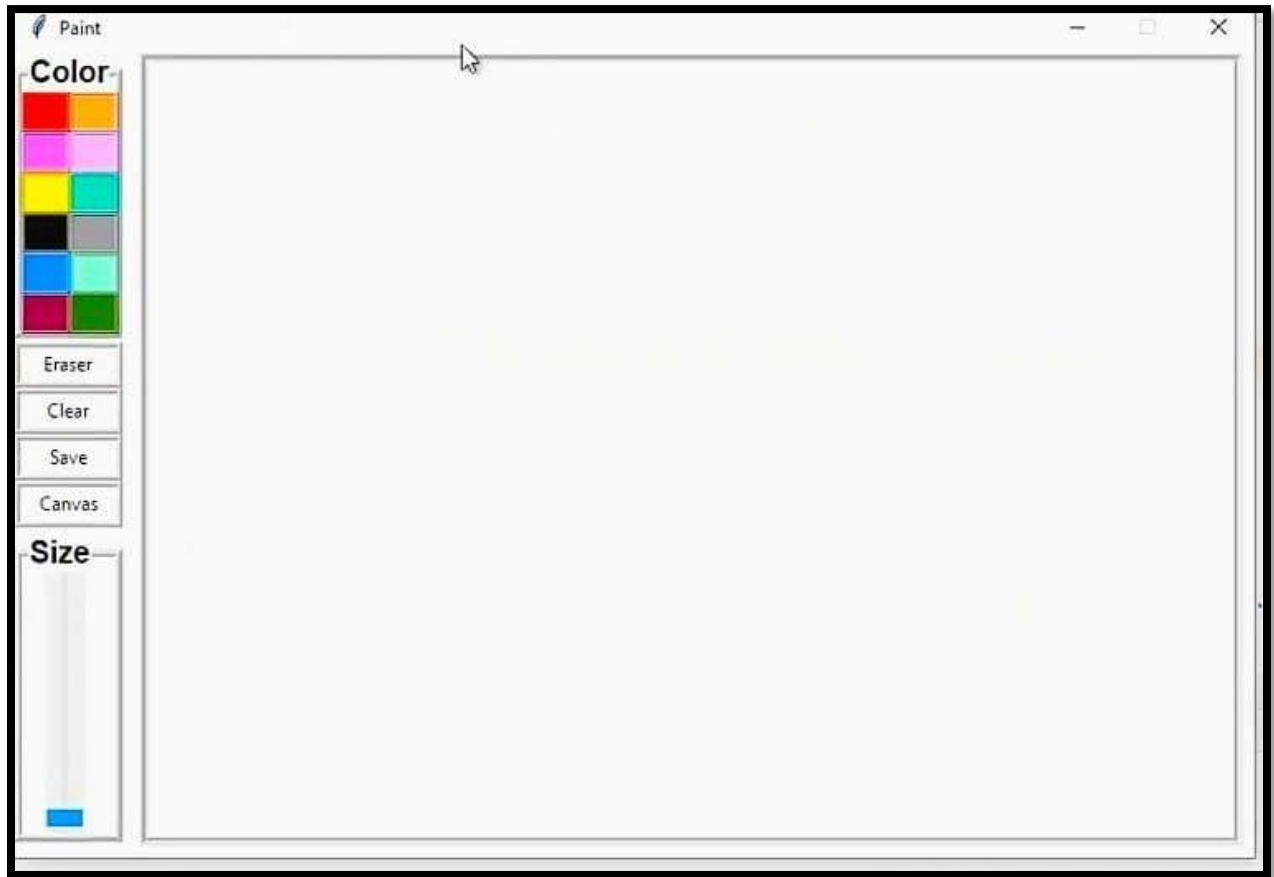
```

Activate Windows
Go to Settings to activate

Testing:

- 1) **Change in color code:** if there is mistake in hash code of color in paint application will give the wrong output as hash codes are pre-defined in the system. If we want to use and apply any color combination in any application we have to use hash codes in the program to display colors.
- 2) **Not mentioning of mainloop function:** if mainloop function is not mentioned in the program of paint application using tkinter in python, the program will run in the background but it will not display output. Logical error will be there.
- 3) **Proper tab space:** if proper tab space is not given in the program, it will show error. Example, after declaring any function the one tab space is given in the next line as it will only receive the statements and conditions in the function.
- 4) **Proper mentioning of grid and place parameter:** the mentioning of grid with parameters row and column should be given with proper calculations and same with place with parameters x axis and y axis. The proper magnitude should be given to the grid and place as if not given the various widgets and color frame will not be placed in the correct location. So be careful while writing dimensions of grid and place geometry managers.

Screenshot:



References:

1. <https://www.youtube.com/watch?v=8-l1KjTj2qw&t=214s>
2. <https://www.youtube.com/watch?v=uW-NLL9dlBs>
3. <https://www.edureka.co/blog/tkinter-tutorial/>
4. <https://www.slideshare.net/EdurekaIN/tkinter-python-tutorial-python-gui-programming-using-tkinter-tutorial-python-training-edureka>
5. https://www.tutorialspoint.com/build_a_paint_program_with_tkinter_and_python/index.asp

