

## 华东师范大学计算机科学与技术学院实验报告

实验课程：计算机图形学	年级：2018级	实验成绩：
实验名称：图形API基本操作	姓名：李泽浩	
实验编号：1	学号： 10185102142	实验日期：2021-3-9
指导教师：王长波、李洋	组号：	实验时间：13:00-14:30

## 一、实验目的

熟悉 Windows 操作系统下图形 API 的基本功能。

## 二、实验内容与实验步骤

1. 创建 VisualStudio 项目并配置
2. 创建第一个窗体
3. 定义窗体过程(Windows Procedure)
4. 在窗体中使用SetPixel
5. 实现画笔工具

## 三、实验环境

由于课上没有及时完成，回到宿舍后发现这次实验只能在Windows系统上完成，经老师助教同意后，改用python实现。

OS: macOS

IDE: Pycharm

主要函数库: tkinter

## 四、实验过程与分析

### 1. 调用tkinter库

```
from tkinter import *  
from tkinter import messagebox  
from tkinter.colorchooser import askcolor
```

### 2. 初始化类

```
def __init__(self, master=None):
    """初始化方法"""
    super().__init__(master) # 调用父类的初始化方法
    self.x = 0
    self.y = 0
    self.fgcolor = 'yellow'
    self.lastdraw = 0
    self.start_flag = False
    self.master = master
    self.pack()
    self.createWidget()
```

### 3. 创建绘图区域及按钮（画笔、清空、颜色）

```
def createWidget(self):
    """创建画图区域"""
    self.drawpad = Canvas(self, width=win_width, height=win_height, bg=bgcolor)
    self.drawpad.pack()
    # 创建按钮
    self.btn_pen = Button(self, name='pen', text='画笔', font=('黑体', 30, 'bold'))
    self.btn_pen.pack(side='left', padx=25)
    self.btn_clear = Button(self, name='clear', text='清屏', font=('黑体', 30, 'bold'))
    self.btn_clear.pack(side='left', padx=25)
    self.btn_color = Button(self, name='color', text='颜色', font=('黑体', 30, 'bold'))
    self.btn_color.pack(side='right', padx=25)
    # 绑定事件
    self.btn_pen.bind('<Button-1>', self.eventManager) # 点击按钮事件
    self.btn_clear.bind('<Button-1>', self.eventManager) # 点击按钮事件
    self.btn_color.bind('<Button-1>', self.eventManager) # 点击按钮事件
    self.master.bind('<KeyPress-r>', self.hotKey) # 绑定快捷键
    self.master.bind('<KeyPress-g>', self.hotKey) # 绑定快捷键
    self.master.bind('<KeyPress-b>', self.hotKey) # 绑定快捷键
    self.master.bind('<KeyPress-y>', self.hotKey) # 绑定快捷键
    self.drawpad.bind('<ButtonRelease-1>', self.stopDraw) # 左键释放按钮
```

并设置了切换颜色的快捷键

#### 4. 将按钮绑定事件

```
def eventManager(self, event):
    name = event.widget.wininfo_name()
    print(name)
    self.start_flag = True
    if name == 'pen':
        self.drawpad.bind('<B1-Motion>', self.mypen)
    elif name == 'clear':
        self.drawpad.delete('all')
    elif name == 'color':
        c = askcolor(color=self.fgcolor, title='请选择颜色')
        print(c) # c的值 ((128.5, 255.99609375, 0.0), '#80ff00')
        self.fgcolor = c[1]
```

选择手绘:

```
def mypen(self, event):
    self.startDraw(event)
    print('self.x=', self.x, ',self.y=', self.y)
    self.drawpad.create_line(self.x, self.y, event.x, event.y, fill=self.fgcolor)
    self.x = event.x
    self.y = event.y
```

开始绘画:

```
def startDraw(self, event):
    self.drawpad.delete(self.lastdraw)
    if self.start_flag:
        self.start_flag = False
        self.x = event.x
        self.y = event.y
```

结束绘画:

```
def stopDraw(self, event):  
    self.start_flag = True  
    self.lastdraw = 0
```

切换颜色的热键:

```
def hotKey(self, event):  
    c = event.char  
    if c == 'r':  
        self.fgcolor = 'red'  
    elif c == 'g':  
        self.fgcolor = 'green'  
    elif c == 'b':  
        self.fgcolor = 'blue'  
    elif c == 'y':  
        self.fgcolor = 'yellow'
```

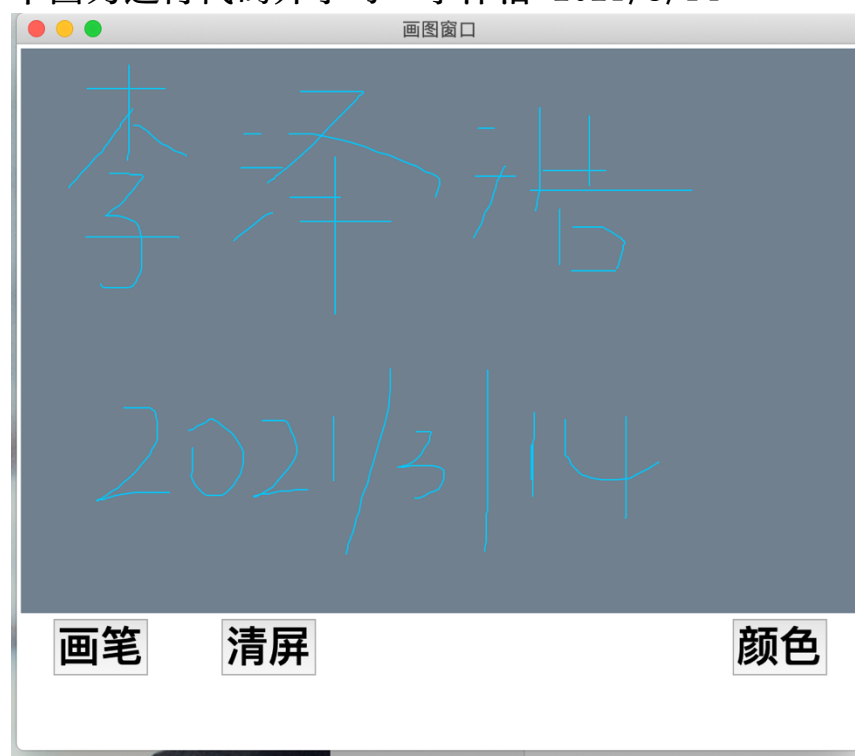
5. 主函数:

```
if __name__ == '__main__':  
    root = Tk()  
    root.title('画图窗口')  
    root.geometry('600x500+200+200')  
    app = Application(master=root)  
    root.mainloop()
```

### 五、实验结果总结

运行上述代码，出现绘图框后，先选择画笔选项（如需更改颜色点击“颜色即可选择”或使用热键），按住鼠标拖动即可完成绘画功能

下图为运行代码并手写“李泽浩 2021/3/14”



### 六、附录（参考资料）