

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：2	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

利用操作系统 API 实现基本直线绘制功能。

二、实验内容与实验步骤

完善上一次实验内容设置的基本绘制逻辑

存储每次鼠标移动的信息

恢复上一次鼠标移动所画的区域

实现 DDA 直线算法

实现 Bresenham 直线算法

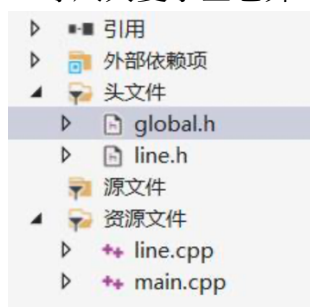
三、实验环境

Windows 10

Visual studio 2019

四、实验过程与分析

1. 导入大夏学堂老师给的API文件



2. 在line.cpp文件补充两个直线算法DAA和Bresenham算法
把直线起始终止点坐标存入vector<Point> result

DAA算法实现：

```
std::vector<Point> draw_line_DDA(Point p_src, Point p_dst) {

    std::vector<Point> result;
    float x0 = p_src.x, y0 = p_src.y;
    float x1 = p_dst.x, y1 = p_dst.y;
    float dx = x1 - x0;
    float dy = y1 - y0;
    int steps = abs(y1 - y0);
    if (fabs(dx)>fabs(dy))
        steps = abs(x1 - x0);
    //初始化坐标
    float x = x0;
    float y = y0;
    //增量
    float xinc = dx / steps;
    float yinc = dy / steps;
    Point po;
    po.x = round(x); po.y = round(y);
    result.push_back(po);
    for (int i = 0; i<steps; ++i)
    {
        x += xinc;
        y += yinc;

        po.x = round(x); po.y = round(y);
        result.push_back(po);
    }
    return result;
}
```

Bresenham

```
std::vector<Point> draw_line_bresenham(Point p_src, Point p_dst) {

    std::vector<Point> result;
    int x0 = p_src.x, y0 = p_src.y, x1 = p_dst.x, y1 = p_dst.y;
    int dx = abs(x1 - x0), sx = x0 < x1 ? 1 : -1;
    int dy = abs(y1 - y0), sy = y0 < y1 ? 1 : -1;
    int err = (dx > dy ? dx : -dy) / 2, e2;
    for (;;) {
        Point P;
        P.x = x0;
        P.y = y0;
        if (x0 == x1 && y0 == y1) break;
        e2 = err;
        if (e2 > -dx) { err -= dy; x0 += sx; }
        if (e2 < dy) { err += dx; y0 += sy; }
    }

    return result;
}
```

五、实验结果总结

本来在课下因为自己的电脑系统的原因用python中的tkinter库实现的画直线，但是发现没有办法像C++一样可以根据鼠标拖动来实现任意位置的任意直线，所以在课上根据老师提供的Api把两个算法的代码改成C++后，添加到了line.cpp中，通过C++实现

(附Python代码)

```
line.py
#直线
import tkinter as tk
root=tk.Tk()
root.geometry('320x240')
b1=tk.Canvas(root)
x0,y0,x1,y1=map(int,input().split())
print(x0,y0,x1,y1)
b1.create_line(x0,y0,x1,y1,width=5)
b1.pack()
root.mainloop()
```