基于画点函数的简易图形库实现

学号-2014211280-谢非

2017年6月14日

目录

1	实验目的	2
2	实验要求	2
3	实验环境	2
4	实验原理	3
5	实验总结	7

1 实验目的 2

1 实验目的

- 1. 学习并应用实现基本的直线和弧线生成算法
- 2. 应用基本图形库进行图案绘画,熟悉绘制原理

2 实验要求

- 1. 基于画点函数,应用直线及弧线生成算法实现一个简易的图形库
- 2. 运用自己实现的图形库打印自己的名字

3 实验环境

实验在 python2.7 环境下进行,运用了 python 的一个图形库 PIL 的画点函数来实现整个图形库

4 实验原理

应用 Bresenham 直线的伪代码如下: function line (x0, x1, y0, y1)int deltax := x1 - x0int deltay := y1 - y0real error := 0real deltaerr := deltay / deltax // 假設deltax != 0 (非垂直線), // 注意: 需保留除法運算結果的小數部份 int y := y0for x from x0 to x1 plot(x,y)error := error + deltaerrif abs (error) 0.5 then y := y + 1error := error - 1.0实验的绘制直线的代码如下: # Bresenham draw line algorithm def DrawLine(allList, x1, y1, x2, y2): dx = x2 - x1dy = y2 - y1error = 0derror = float(dy) / float(dx) xList = []yList = [] $y = \min(y1, y2)$ $x = \min(x1, x2)$ $\max X = \max(x1, x2)$ maxY = max(y1, y2)counter = 0while x < max X and y < max Y: xList.append(x)counter +=1error += derror if abs(error) >= 0.5: error = 1.0x += 0.1

pie = (float(y2)-float(y1))/counter;

```
for i in range(counter):
    yList.append(y+i*pie)

for i in range(counter):
    allList.append(xList[i])
    allList.append(yList[i])
```

其中 allList 是一个 x1,y1,x2,y2,x3,y3 ... 存储的列表,最终可调用 draw.point(allList) 来绘制所有的点。

实验过程中效果图如下:

```
### DrawCircle ## DrawThief(allList) ## DrawThief(allList
```

图 1: 用三条直线画出三角形

```
# DrawMatrix
DrawMatrix(allList, 40, 40, 50, 70)

#clearDraw(allList)

# # DrawCircle
# DrawCircle(allList, 90, 90, 30);

# clearDraw(allList)

# # Draw author name
# DrawThief(allList)

draw.point(allList, fill=125)
im.show()
```

图 2: 画矩形

```
# DrawCircle
DrawCircle(allList,90,90,30);

# clearDraw(allList)

# # Draw author name
# DrawThief(allList)

draw.point(allList, fill=125)
im.show()
```

图 3: 画圆

```
#DrawCircle(allList,90,90,30);

# clearDraw(allList)

# Draw author name
DrawThief(allList)

draw.point(allList, fill=125)
im.show()

# DrawLine(4,5,8,9)
# xps = [0, 7, 8, 9, 2]
# yps = [0, 5, 7, 3, 1]
# plt.scatter(xps, yps)
# plt.show()
```

图 4: 打印自己的名字

5 实验总结 7

5 实验总结

本次实验过程中,我对实际的基本的直线以及弧线算法有了更深入的认识和理解,期望之后可以用更加高效简洁的算法来实现这个基本的图形库。