实验报告：GraphicsLibrary

董奕柳

2024 年 12 月 23 日

1 项目概述

这是一个简单的图形函数库，用于绘制基本的几何图形和文本显示，包含以下功能：

- 绘制直线段
- 绘制圆弧
- 绘制椭圆弧
- 多边形区域填充
- 显示名字

该项目采用 C++ 编写，并使用 CMake 构建工具。

2 功能设计

2.1 绘制直线段

直线段的绘制基于 Bresenham 算法，其核心思想是通过整数计算近似实现直线段的绘制，避免使用浮点运算，从而提高效率。算法的数学公式及伪代码如下：

2.1.1 数学公式

假设起始点为 (x_1, y_1)，终点为 (x_2, y_2)，直线段的斜率为 $k = \frac{\Delta y}{\Delta x}$，其中：

$$\Delta x = x_2 - x_1, \quad \Delta y = y_2 - y_1$$

Bresenham 算法通过计算误差项 e 来决定当前像素的位置更新：

$$e = 2\Delta y - \Delta x$$

根据误差更新规则：- 若 $e > 0$，则表示需要调整纵坐标 y：

$$e = e - 2\Delta x$$

- 否则，只调整横坐标 x：

$$e = e + 2\Delta y$$

2.1.2 算法描述

Algorithm 1 Bresenham 绘制直线段算法

Require: 起始点 (x_1, y_1)，终点 (x_2, y_2)

Ensure: 绘制从 (x_1, y_1) 到 (x_2, y_2) 的直线段

计算 $\Delta x = |x_2 - x_1|$, $\Delta y = |y_2 - y_1|$
初始化误差项 $e = 2\Delta y - \Delta x$
设置步长 $y_{\text{step}} = 1$ 若 $y_2 > y_1$，否则 $y_{\text{step}} = -1$
初始化 $y = y_1$
for $x = x_1$ to x_2 **do**
    绘制点 (x, y)
    if $e > 0$ **then**
        $y \leftarrow y + y_{\text{step}}$
        $e \leftarrow e - 2\Delta x$
    end if
    $e \leftarrow e + 2\Delta y$
end for

2.2 绘制椭圆弧

椭圆弧的绘制基于参数方程，其核心思想是使用角度增量逐点计算椭圆弧上的像素点。特别地，当椭圆的长轴和短轴相等时，椭圆弧就退化为圆弧。

2.2.1 数学公式

椭圆弧的参数方程为：

$$(x, y) = (cx + a\cos\theta, cy + b\sin\theta), \quad \theta \in [\text{start}, \text{end}]$$

其中：- (cx, cy) 为椭圆的中心坐标；- a 为椭圆的长轴半径；- b 为椭圆的短轴半径；- θ 为椭圆弧的角度。

当 $a = b$ 时，椭圆弧退化为圆弧，其参数方程变为：

$$(x, y) = (cx + r\cos\theta, cy + r\sin\theta), \quad \theta \in [\text{start}, \text{end}]$$

其中 $r = a = b$ 为圆的半径。

2.2.2 算法描述

Algorithm 2 椭圆弧绘制算法

Require: 椭圆中心 (cx, cy)，长轴 a，短轴 b，起始角度 start，终止角度 end

Ensure: 绘制从起始角度到终止角度的椭圆弧

设置步长 step 用于角度增量
for $\theta =$ start to end step step **do**
    $x \leftarrow cx + a\cos\theta$
    $y \leftarrow cy + b\sin\theta$
    绘制点 (x, y)
end for

2.3 多边形填充

多边形填充基于扫描线算法，其核心思想是逐行扫描像素并填充多边形内部区域。

2.3.1 数学公式

设多边形的顶点集合为 $(v_1, v_2, \ldots, v_n)$，对于扫描线 $y = k$，交点的 x-坐标可以通过多边形的边方程计算：

$$x = x_1 + \frac{(k - y_1)(x_2 - x_1)}{y_2 - y_1}, \quad y_1 \leq k < y_2$$

2.3.2 算法描述

Algorithm 3 扫描线多边形填充算法

Require: 多边形顶点集合 $(v_1, v_2, \ldots, v_n)$

Ensure: 填充多边形内部

计算多边形的最小 y-坐标 $y_{\min}$ 和最大 y-坐标 $y_{\max}$
for $y = y_{\min}$ to $y_{\max}$ **do**
    找到扫描线与多边形边的交点集合
    按 x-坐标对交点排序
    for 每对交点 $(x_{\text{left}}, x_{\text{right}})$ **do**
        填充从 x_{left} 到 x_{right} 之间的像素
    end for
end for

3 结果展示

图 1 展示了通过本图形库绘制的不同图形及其效果，包括以下几个部分：

- **红色直线段**：从点 $(50, 50)$ 到点 $(400, 50)$，在窗口的顶部水平排列。

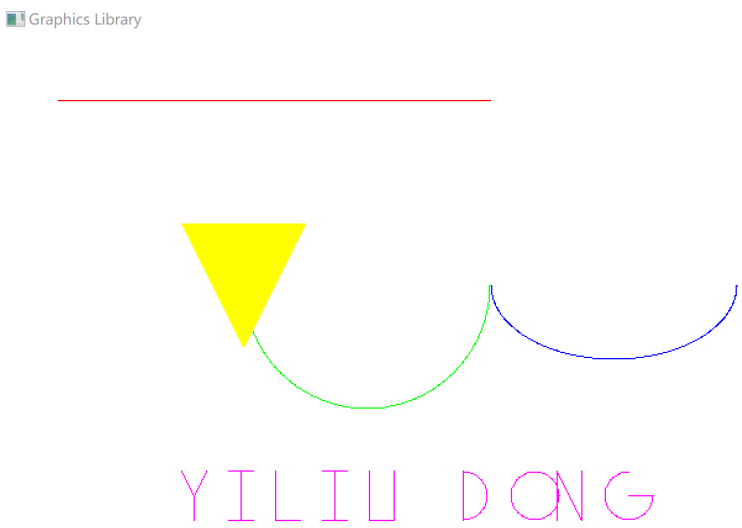

图 1: 通过图形库绘制的图形效果展示

- **绿色半圆弧**：圆心位于 $(300, 200)$，半径为 100，起始角度为 0 度，终止角度为 180 度，形成一个半圆。
- **蓝色椭圆弧**：圆心位于 $(500, 200)$，长轴为 100，短轴为 60，起始角度为 0 度，终止角度为 180 度，形成一个椭圆弧。
- **黄色三角形**：顶点分别位于 $(150, 150)$、$(250, 150)$ 和 $(200, 250)$，并进行了填充。
- **紫色文本**：在位置 $(150, 350)$ 绘制的 “YILIU DONG” 文本，展示了如何在图形界面中插入文本。

4 总结

通过本图形库，我们能够方便地绘制直线段、圆弧、椭圆弧和多边形填充，并实现文本显示功能。