

作业一 MiniDraw-实验报告

姓名：孙浩然 id：09

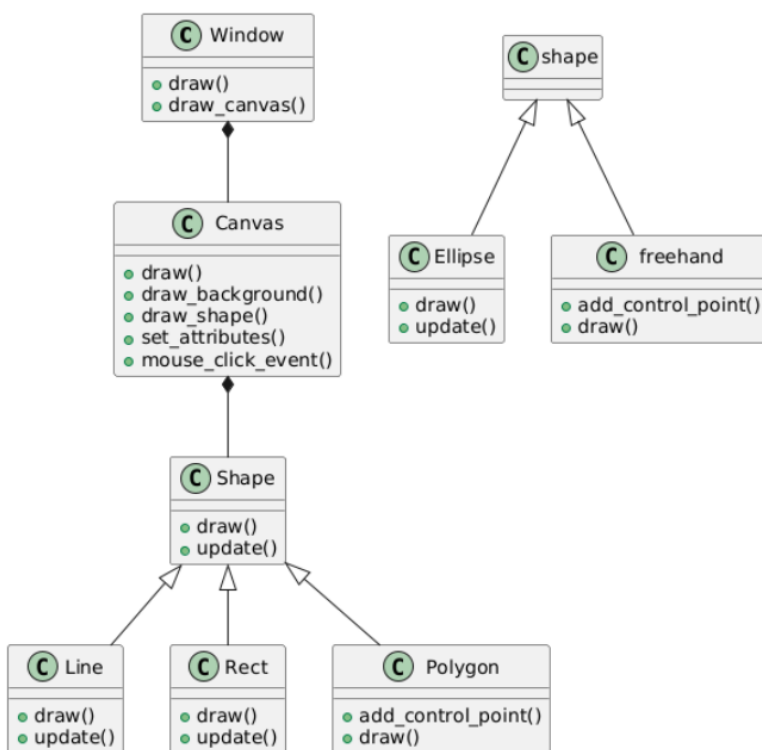
2025 年 3 月 2 日

运行环境：cmake+vscode+vs studio 2022

1 算法原理

程序主要有三个部分：MiniDraw 控制窗口，Canvas 管理并调用图形来进行绘制和保存，Shape 具体的图形实现。MiniDraw 作为一个窗口类，继承自 windows 父类，在程序运行时调用 `w.run()` 方法，进行界面的初始化，随后每一帧调用 `render()` 方法，实现 `draw canvas`，进而实现 `canvas` 类中的 `draw()` 方法。Canvas 也是一个子类，包含绘图的具体方法。由于 `canvas` 类只在第一次时创建依次，所以这个类的 `private` 变量是在每一帧的 `draw()` 方法调用中共享的，所以可以看作鼠标的操作点击，拖动等是在屏幕上实时交互的。以实现 `polygon` 举例，鼠标左击，就会激活调用 `canvas` 中的方法，改变 `private` 成员变量，最后实现 `draw()`，进行图像的绘制。`shape` 类是父类，对应的具体图片是子类，子类继承了父类的函数，对于父类中的虚函数，子类进行不同的编写，以实现在绘图时实现不同的 `draw()` 方法，此为继承和多态。

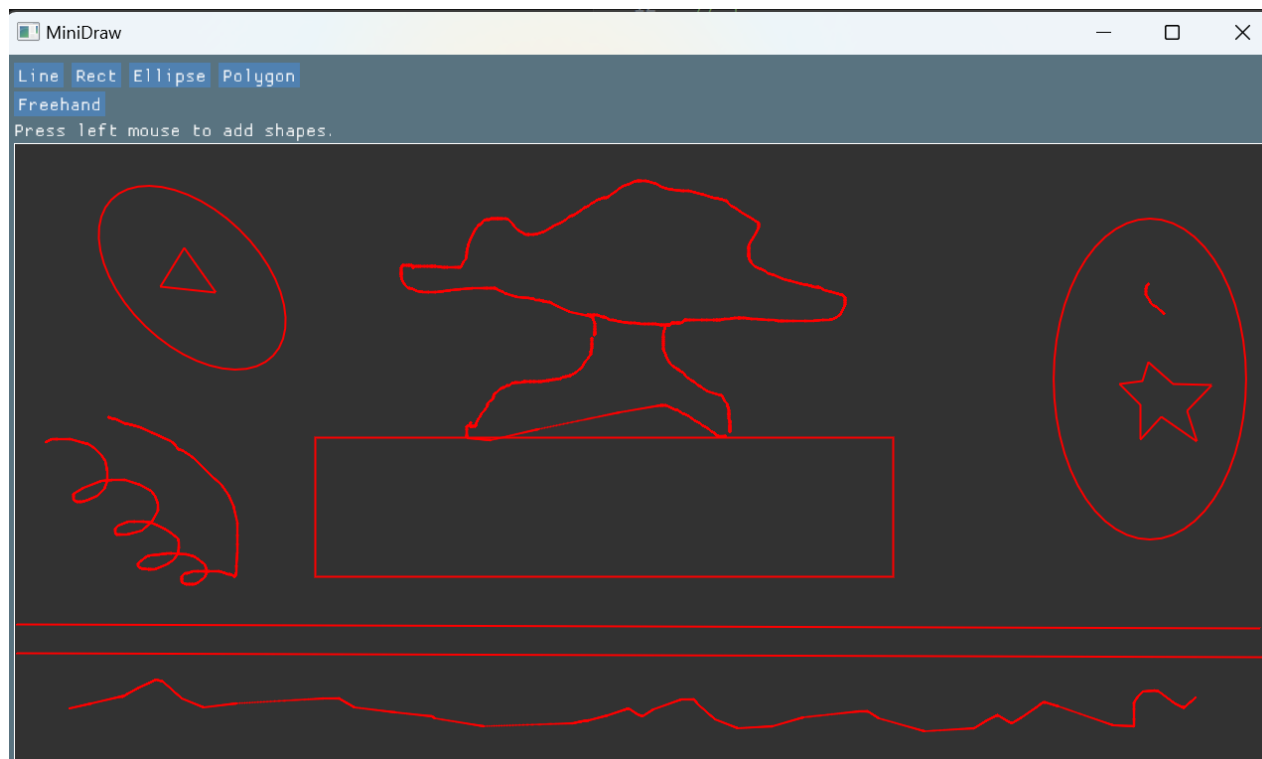
2 程序实现



3 程序运行方式

使用 CMake 配置后，点击要绘制的图形，除多边形外，点击左键开始绘制，绘制过程实时更新，再次点击左键，停止绘制。多边形则是左键添加控制点，右键结束。自由作画则是点击左键，开始绘画，移动光标对应做出绘画，再次点击左键，停止绘画。

4 关键结果



对于椭圆，第一次点击确定长轴的一个端点，第二次点击确定长轴的第二个端点，预设短轴是长轴的 0.6。

对于多边形，将其分为控制点之间的连接，第一个连第二个，第二个连第三个，依次下去直到最后一个连第一个。

对于 freehand，采取鼠标位置为控制点，对于其中的每三个点使用二次贝塞尔曲线进行拟合，再对时间进行采样。对于每个采样点，为保证粗细与参数一致，以给定参数为直径进行画圆。

5 总结

1. 在绘制多边形时，一开始编译成功但运行不出来，发现 canvas 类中 current shape 类型是 std::shared_ptr<Shape>，在具体实现时 current shape = std::make_shared<Polygon>(start point.x,start point.y)，无法调用 add point 函数。

原因：在 shape 类中，该函数前未加 virtue，并非是虚函数，因此该指针实际上是 shape 类型，会调用 shape 中的 add point() 方法，是空实现。

解决方法：1. 加上 virtue。2. auto polygon = std::dynamic_pointer_cast<Polygon>(current shape) 这样就会优先调用 polygon 类中的 add 方法。

2. 在绘制多边形时，左击添加控制点，如果不右击，直接切换类别。画好的图形会直接消失。

原因：由于没有点击右键，**current shape** 未被存入 **shape list**，也未被清除。但切换到其他类型时，会调用 **set line** 函数，令 **draw staus = false**，导致当前 **current** 未被绘制。

3.freehand 时，鼠标移动过快导致画出的线离散。

解决方法：减小采样的时间间隔。