Homework 1 MiniDraw

华南理工大学 曾亚军

一　实验目的

- 写一个画图小工具 MiniDraw，要求画直线 (Line)，椭圆 (Ellipse)，矩形 (Rectangle)，多边形 (Polygon)，自由绘制 (Freehand) 等图形元素 (图元)。
- 学习类的继承和多态。
- 入门 Qt。
- 实现 MiniDraw 更多的功能。

二　功能描述

实现的画图小工具 MinDraw 具有较为整洁的界面如下：

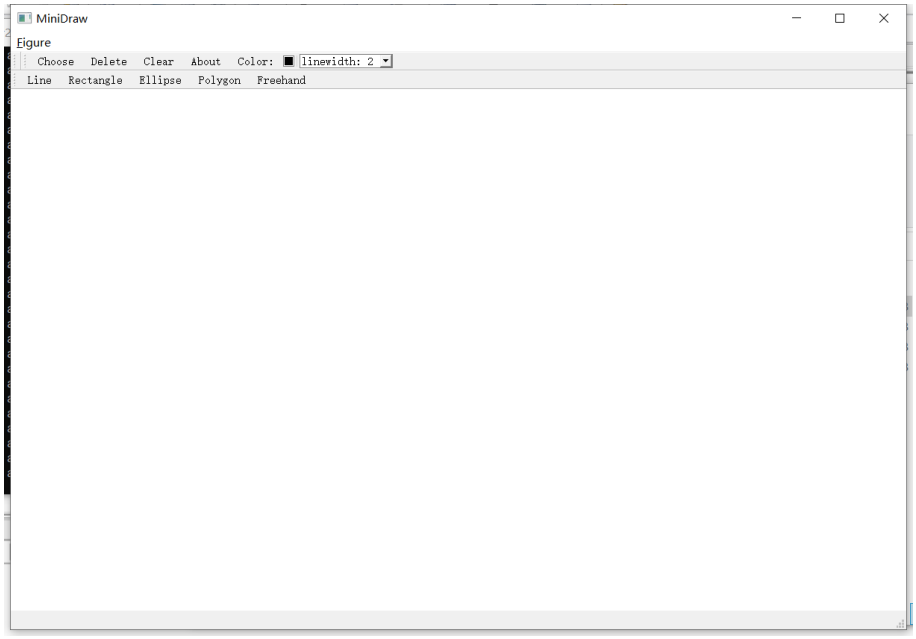

图 1: 界面

在本作业的 MiniDraw 工具中，实现了较多功能：

- 直线 (Line)，椭圆 (Ellipse)，矩形 (Rectangle)，多边形 (Polygon)，自由划线 (Freehand) 等图形元素 (图元) 的绘制。

- 图元的即时绘制。在绘制各种图元过程中（即鼠标不释放并移动的过程中），图元即时地给予反馈，不会等到鼠标释放才显示图形。
- 线段的粗细和颜色等风格的选择。
- 通过派生 CRectangle 类产生选择框 CSelectRect，实现图元的选择和移动。
- 鼠标左键单击瞬间弹出该点坐标，释放或移动时消失。
- 撤销，清空等功能的实现。

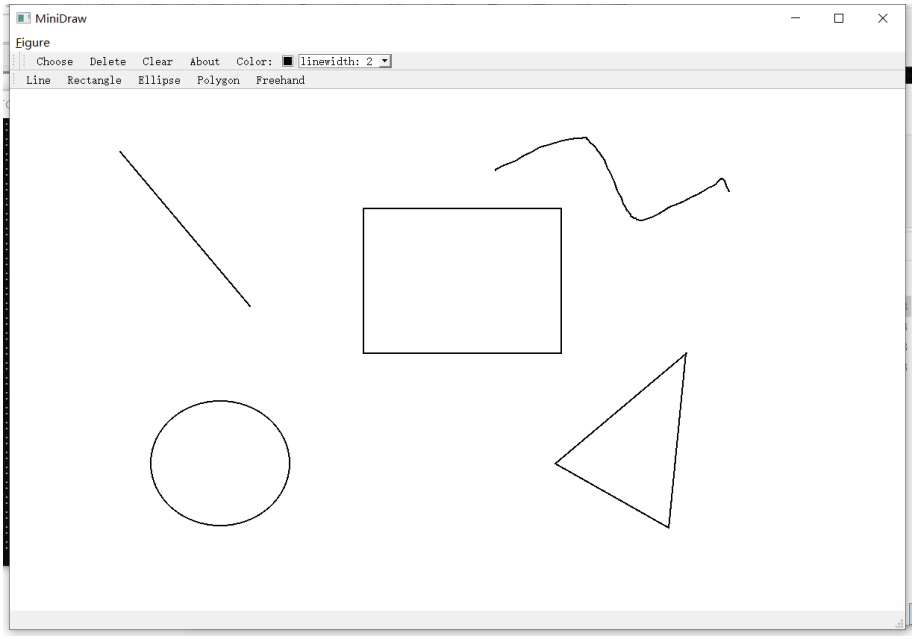

图 2: 图元的实现

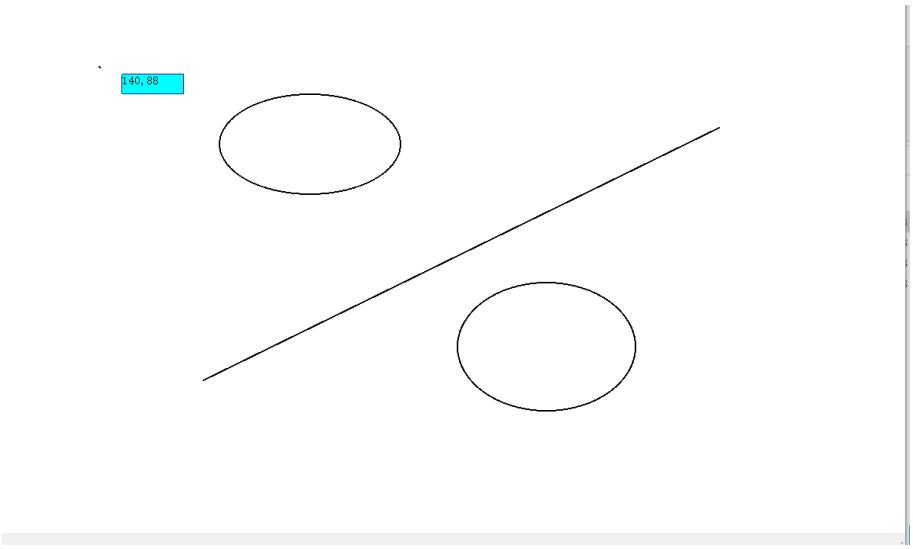

图 3: 单击弹出坐标

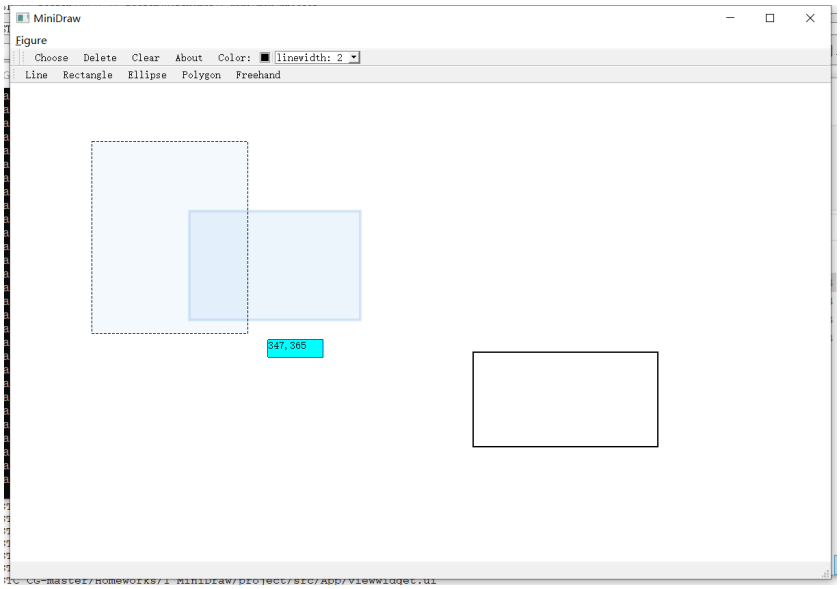

图 4: 选择模式和与选择框相交的图元

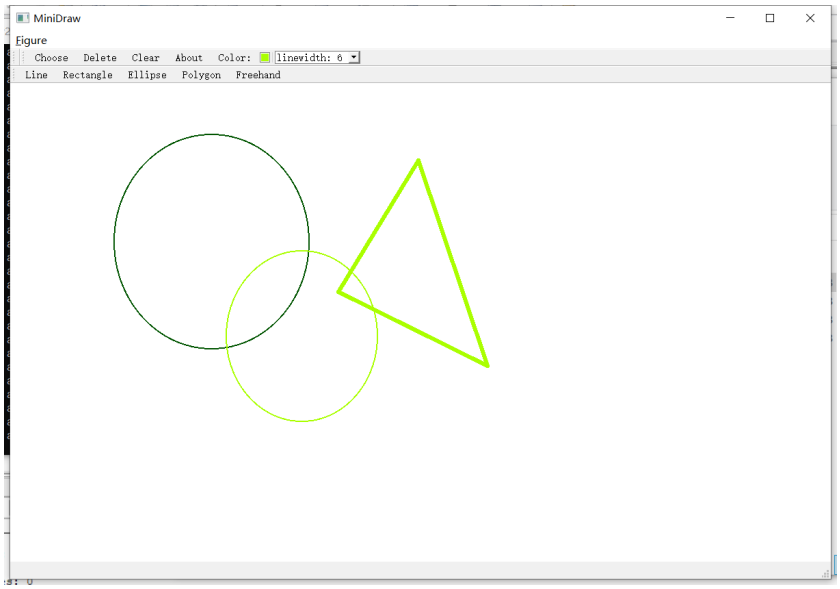

图 5: 颜色和线段粗细选择

三　类设计

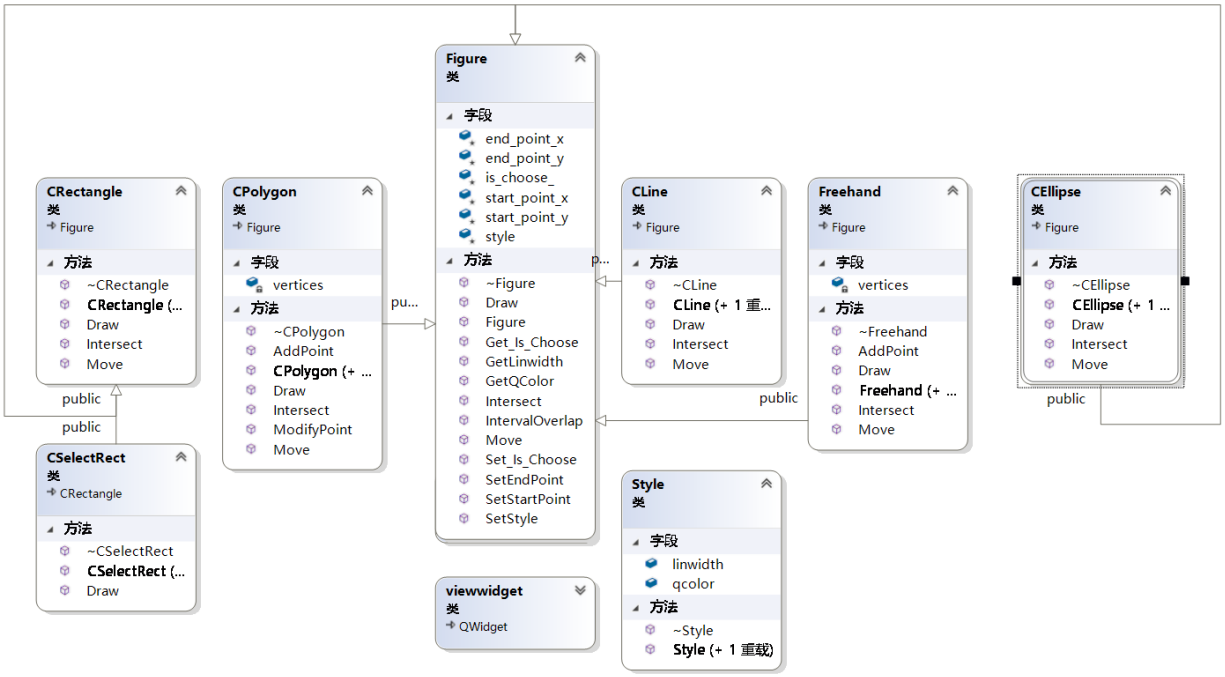

图 6: 类图

基本图元 Figure 包括了很多公用的接口和可继承的属性。而其他基本图元如直线 (Line)，椭圆 (Ellipse)，矩形 (Rectangle)，多边形 (Polygon)，自由划线 (Freehand) 等都继承自 Figure 基类（公有继承）。其主要区别在于 draw 函数和 intersect 函数，即图元在如何绘制和如何如选择框相交上表现行为的差异。

选择框 CSelectRect 继承自基本图元矩形，并在 draw 函数上有了不同表现。

四　实现难点

由于本次作业是我第一次自己实现 Qt 程序，对很多 Qt 的程序命令，包括绘图工具都不是很熟悉。但 PPT 有比较详细的教程，因而基本的 Qt 程序可以很容易理解和实现。算法过程中遇到的几个难点主要在于：

1. 图元的即时绘制：

产生基本的图元都比较简单，但后来发现，自己撰写的程序不能即时地出现绘制结果。如绘制矩形的时候，会等到鼠标释放之后，才开始绘制。后来我在开始绘制的时候，就将直接新建图元，并加入 current_figure 中，在绘制过程中，不断修改终点。

2. 多边形的鼠标事件和存储结构：

在这些图元里面，区别最大的是多边形和自由绘制工具。自由绘制工具本质上是点的集合，因而我用 $vector < QPoint > \ vertices$ 可以很好地实现自由绘制。其鼠标事件也只是单纯的左击开始释放结束。多边形我这里同样采取 $vector < QPoint > \ vertices$ 存储点

的信息。但在绘制过程中，由于多边形每次左击之后仍然保持绘制状态，而且鼠标移动的过程中，程序需要跟着绘制（此时鼠标是释放状态），增添了鼠标事件的判断难度。

3. 相交和选择事件：

选择框我通过派生 CRectangle 类产生选择框 CSelectRect，并修改风格（颜色，粗细）很容易实现。但困难的是判断相交。在程序在我细致地考虑了选择框和各种图元的相关情况和怎么判断，这对于椭圆、多边形而言更加困难。

五　注意事项

- 用 new 新建的图元对象一定要用 delete 删除。
- 由于绘制过程中，已存在的图元保存在 figure_array 中，而被选择工具选择的图元同时还保存在 choose_figure_array 中。这个时候特别注意，当删除其中任何一者的某个图元，另外一者所存储的对应图元一定要清空，否则会在该 array 中占具位置，且为迷失指针，不指向 NULL。
- 绘制不同图元的时候，用的时同一个 Qpainter，因而我们需要把图元的风格（颜色和线条粗细）作为图元的内部信息，否则就会造成绘制的时候统一变色或者改变粗细。
- 由于多边形、自由绘制工具、选择工具大大增添了程序鼠标事件的各种选择，因此要细心注意什么状态下鼠标触发什么事件，并多做测试。

六　收获与感悟

本次作业其实是我第一次撰写 Qt 类的应用。之前编写 windows 窗体应用或者手机 APP 时基本是使用 WPF 和 Java 进行开发。这次尝试一方面让我了解和学习了一个构架窗体应用的新方法，且 Qt 非常适合几何处理和绘制；另一方面，我对几何图形的数据结构，如何绘制有了更清晰的认知。

程序的时间花了不少是时间，一方面是在不断地 Debug，之前的编程习惯让我很容易在一些不起眼的地方犯错，但这样也恰好锻炼了我的 Debug 能力和水平；另一方面，是在不断优化这个画图小程序，先是加入了线条粗细和颜色等风格设置，再加入了清空画板的功能，最后进一步开发了选择模式，判断选择框与图元是否相加，并对选择图元进行移动和删除等操作。最终得到一个差不多的“成品”，心中有一定的成就感。