

# 基础图形绘制软件的实现

191810140沈炫辰

## 一、引言

本实验主要利用设计模式完成一个基础图形绘制软件，因为涉及到图形界面，所以运用Qt图形库来完成实验，下面列出了实验的环境：

图形界面：Qt

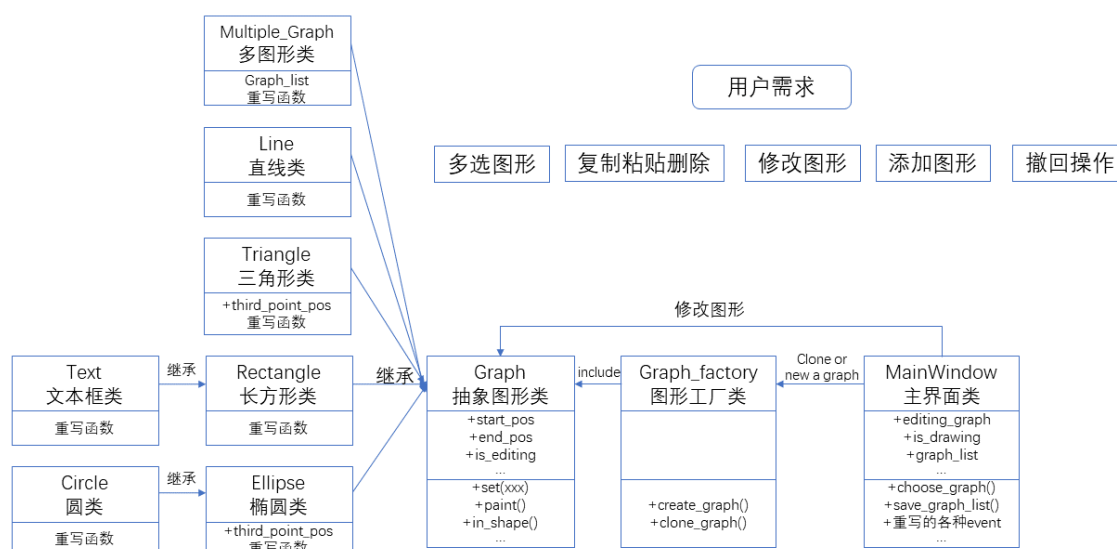
编译器：Qt6.0.4MINGW

IDE:Qt Creator5.0.1

系统：windows10

语言：C++

## 二、对目标系统的分析和设计



对于多种图形的需求，用一个虚基类（抽象类）Graphic类来作为父类，各种图形类重写父类中的各个函数。对于实现MainWindow，为了创建对象更为方便，更加简洁，运用简单工厂模式，将图形对象放到一个工厂类中生成。

在MainWindow储存的图形序列中，有单个图形，也有由多个图形组合而成的组合图形，运用组合模式，可以让编写MainWindow对于同一的图形进行处理（如复制粘贴删除）时，不需要考虑其到底是组合图形还是单个图形，都可以用统一的接口来完成。

为了实现撤销功能，每一次对图形进行操作前，都将原来的状态保存，采用备忘录模式，可以轻松恢复到之前的状态，缺点是占用很大的内存空间。

多个图形组合而成的图形仅仅在选中和编辑图形时才会出现，而同时只能有一个在编辑的图形，所以利用单例模式，可以避免多个图形组合而成的对象被创建多次，违反了编写者的本意。

## 三、实现方案

### 1.功能大体实现方案

没有使用qt现成的QGraphics系列库，而是单纯通过重写MainWindow的鼠标和键盘响应函数来实现程序。

对于画面上图形的维持使用了qt封装的QList结构。

创建图形时，采取拉取的方式拉出图形，鼠标先左键点击确定图形的起始点，然后通过按住左键拉动，最后放开决定结束点（重写鼠标事件）。大部分图形只需要两个点就可以确定形状，椭圆则需要三个点，创建时，先使其成为一个圆，然后在后面的调整中可以成为椭圆。

如果图形后创建，则会居于更上面，在鼠标点击时，从新到老判断鼠标是否点击到了某个图形，即根据坐标算法计算鼠标是否在图形内，对于图形的边角会有一些的模糊，让鼠标更容易点中图形，如果图形较小，鼠标点击效果可能会因为各个部分的重叠而变差。

选中图形后，可以进行多选，这个时候创建多图形实例，并且设定为正在编辑的图形。单个图形可以拖拽移动或者拖拽焦点来缩放，两种图形都可以复制粘贴剪切删除（通过重写键盘事件）。

文字则继承自长方形，并且有初始文字。当选中文本框时，可以在上方的工具栏中修改文字内容。文字大小自适应文本框大小（通过qt信号槽实现）。

在不处于选中图形、画图等时刻时，可以按下ctrl+s截图，利用qt的截图功能(grab)实现。

### 2.设计模式实现方案

·工厂模式：创建了一个工厂类，接受MainWindow类发出的指令造出不同的对象

·组合模式：既有单个图形的类又有包含多个图形的多图形类，而在大部分具体操作中，它们都被视为一种类，来实现多态和代码复用。

·备忘录模式：每次进行操作时，先把操作前的图形情况复制一遍誊写在备忘录上，如果要撤销就从栈式的备忘录中不断取出之前的记录。备忘录模式虽然实现简单，但是缺点是占用内存空间巨大。

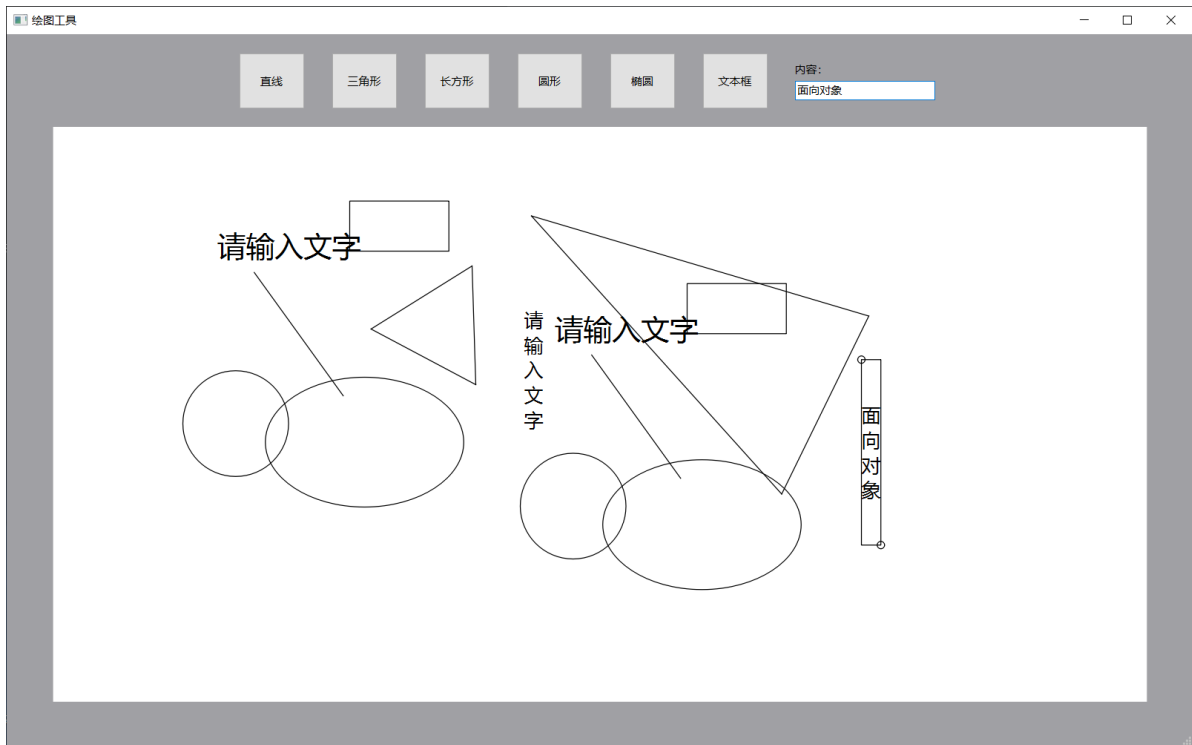
·单例模式：多图形类在同一时间只能出现一次，所以使用单例模式，隐藏其构造函数，每次需要的时候从static函数取出其单例，避免创造多个实例引发不必要的问题。

## 四、实现功能介绍、界面展示

### 1.基础功能

·图形的绘制，实现了直线、三角形、正方形、圆形、椭圆五种形状，通过点击菜单栏对应的选项，然后在画布上拉取得到图形

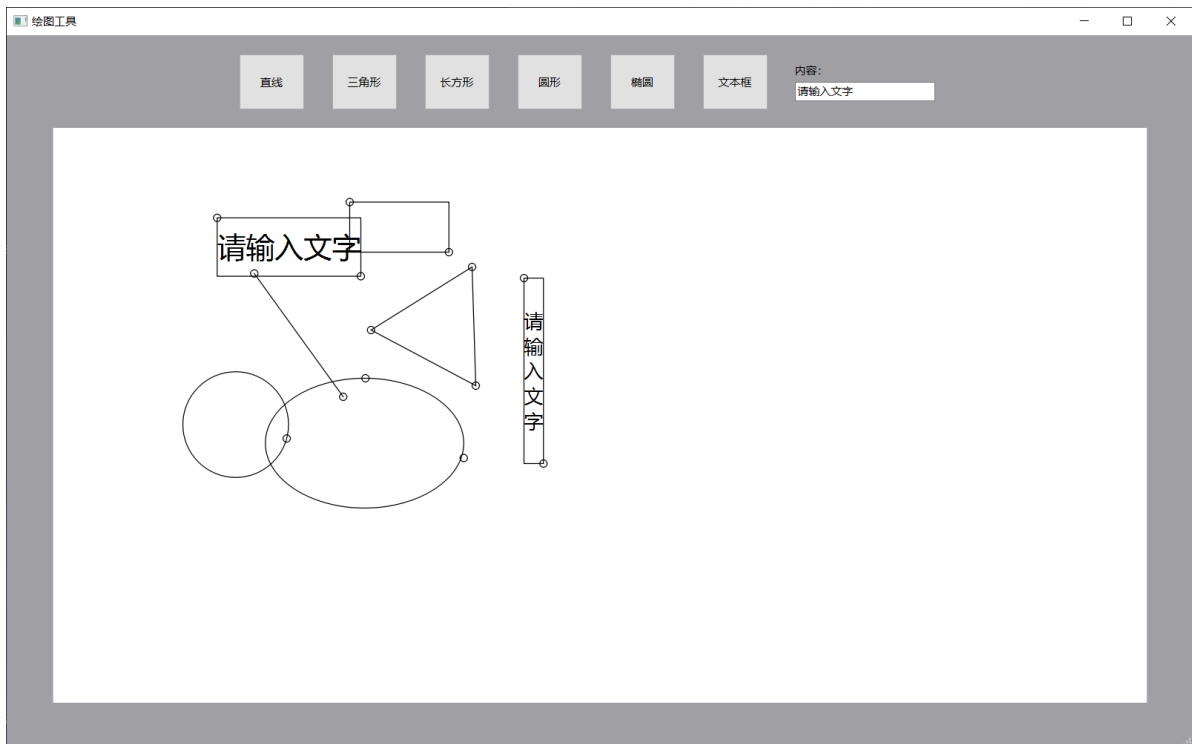
·文本框（文字描述）的绘制，实现了文本框的绘制，绘制完成点击文本框后，可以在上方工具栏修改其内容，文本的字体大小自动适应文本框的大小



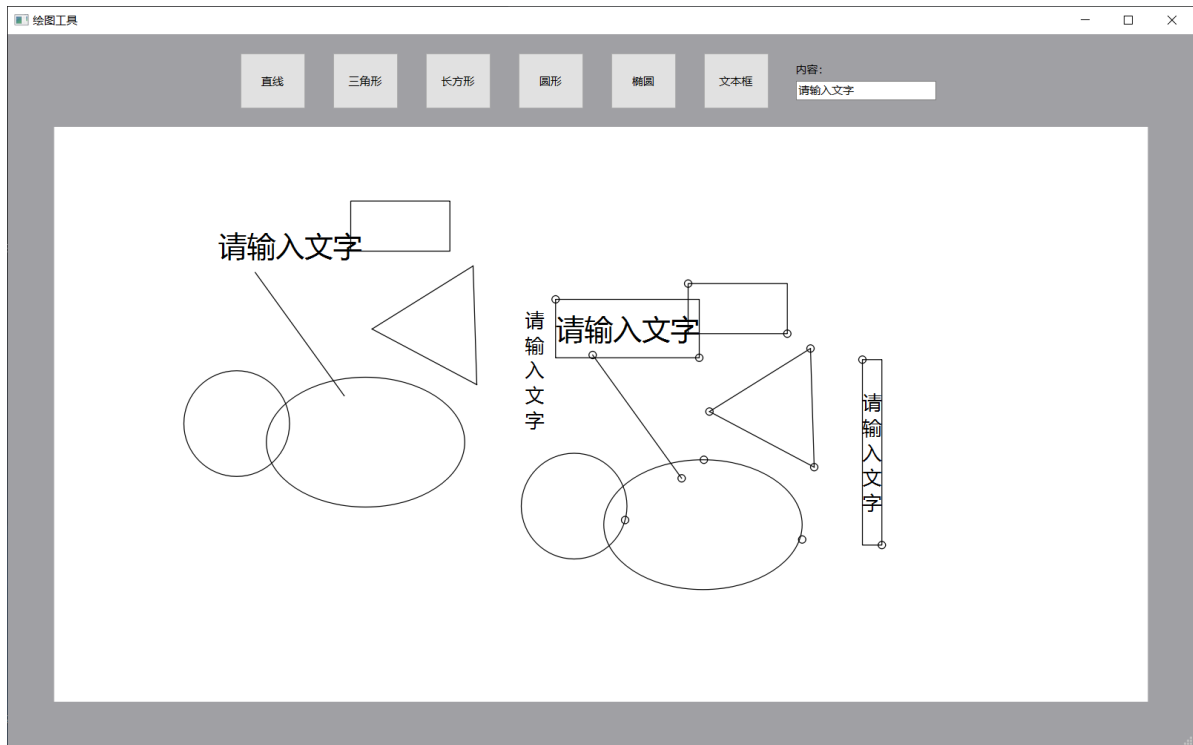
### 文字的修改

·完成了对单个图形点击的处理，如果点击了空白区域，则会去除之前的点击效果。点击后，被选中的图形上会出现一些圆圈，这时候可以对图形进行复制粘贴剪切删除和移动，图形上的圆圈为图形的基点，决定了图形的形状

·如果点击了一个图形之后按下ctrl点击另外的图形（注意点击图形只会选中最顶层的图形），则会多选，多选同样支持复制粘贴剪切删除和移动



### 图形的多选



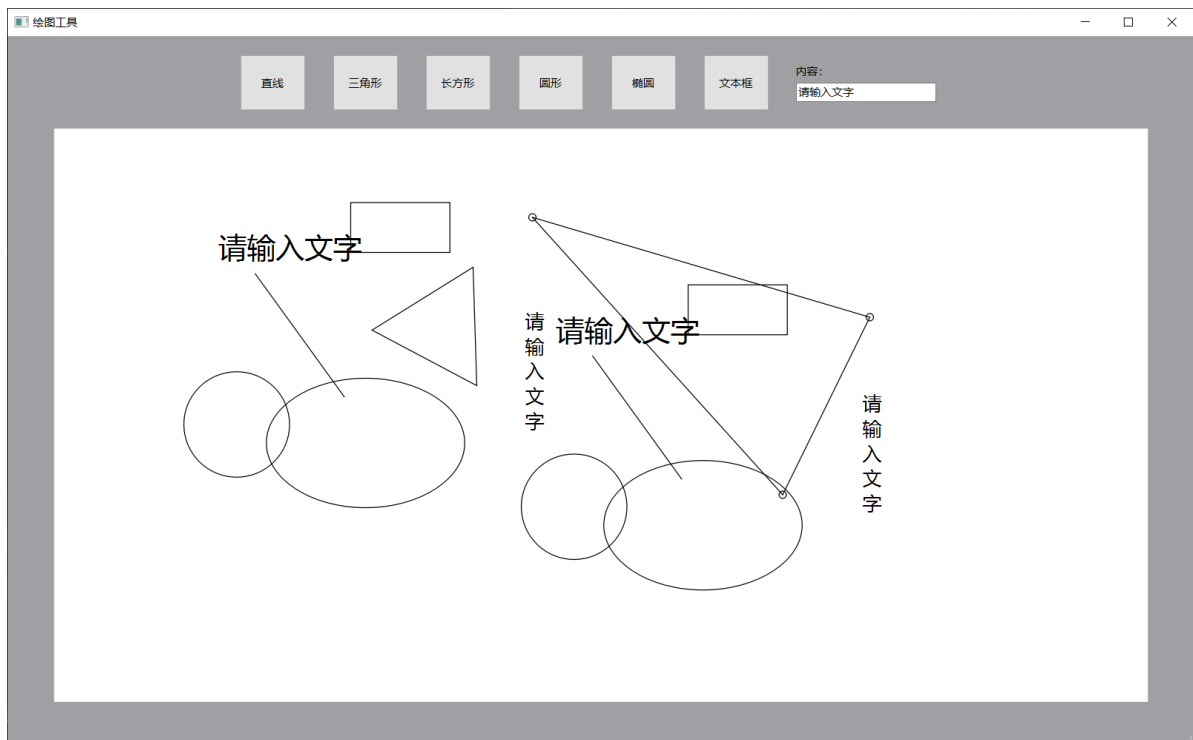
多选图形的复制粘贴

·允许多步撤销，按ctrl+z

## 2.扩展功能

·多步撤销

·支持拖拽单个图形的基准点（即点击图形后出现的空心圆点）来改变图形的大小形状，拖拽图形其他区域则会移动图形



图形的变形

·支持对完成的作品进行截图（但是没有实现以一定的数据格式存储再读取）

·支持ctrl+u置顶图形；ctrl+l置底图形

## 五、小结

- 1.由于时间仓促，没想到最后会那么忙，扩展功能实现的不多，很可惜
- 2.因为在搭建好程序的底层架构之后才发现qt有QGraphics系列接口，但是已经来不及更换框架，于是相当于自己造了轮子，这样也有好处，如果用了qt的scene和QGraphics，基本上自己不用写什么代码了，不过下次一定广泛阅读文档，看看有没有能够直接拿来用的库