

# QT 大作业设计报告

组名：魑魑队 组长：于米多 2300012962 组员：陈子睿 2300012966 罗宇泽 2300013128

## 一、分工

1. 查找资料：于米多、陈子睿、罗宇泽
2. 方案设计、功能设计：于米多、陈子睿、罗宇泽
3. 代码实现：
  - Ctrl 键多选功能：于米多
  - “绘制”按钮功能：于米多
  - Alt 键缩放功能：陈子睿
  - 删除功能：罗宇泽
4. 报告撰写：于米多，陈子睿，罗宇泽

## 二、函数与类的部分介绍

**class CustomItem : public QGraphicsRectItem**

//构建一个图形类，继承矩形类，QGraphicsScene 管理 QGraphicsItem（单击/选择/移动/缩放/删除）

**class CustomScene : public QGraphicsScene**

//自定义一个画板类，继承 QGraphicsScene

**CustomItem::CustomItem(QGraphicsItem\*parent):QGraphicsRectItem(parent)**

// CustomItem 初始化

**void CustomItem::mousePressEvent(QGraphicsSceneMouseEvent \*event)**

//左键（选中功能）ctrl 键（多选）功能实现

**void CustomItem::mouseMoveEvent(QGraphicsSceneMouseEvent \*event)**

//Alt 键（缩放）功能实现

**void CustomItem::mouseReleaseEvent(QGraphicsSceneMouseEvent \*event)**

//Alt 键（缩放）功能实现

**int CustomItem::type() const**

//图形类别

**void CustomScene::mousePressEvent(QGraphicsSceneMouseEvent \*event)**

//左键（构建图形）功能实现

**void CustomScene::mouseMoveEvent(QGraphicsSceneMouseEvent \*event)**

//检测画板的移动

**void CustomScene::keyPressEvent(QKeyEvent \*event)**

//backspace（删除）功能实现

**void CustomScene::setrect()**

//与 main 函数中的“绘制”按钮链接，保证点击“绘制”按钮后才能左键绘制图形

### 三、功能介绍与实现示例

1. 左键单击绘制按钮之后，在画板上单击左键可以构建一个矩形，刚绘制出的矩形大小是固定的

（不点击绘制按钮则不能绘制图形）

绘制

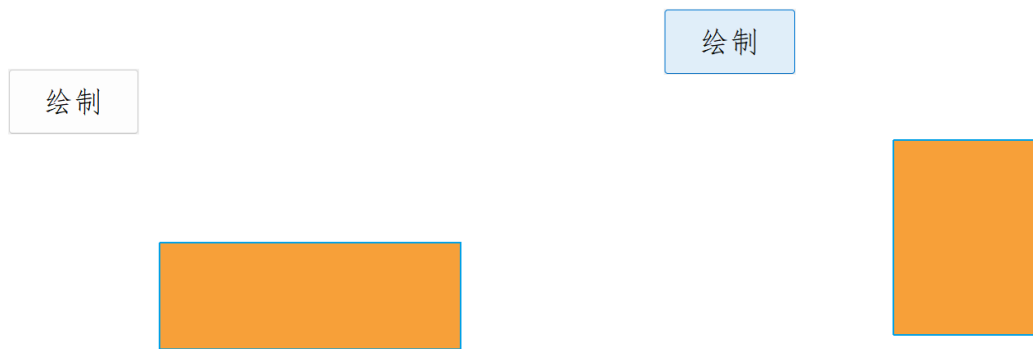


2. 左键单击矩形可以选中矩形，实线边框变为虚线边框

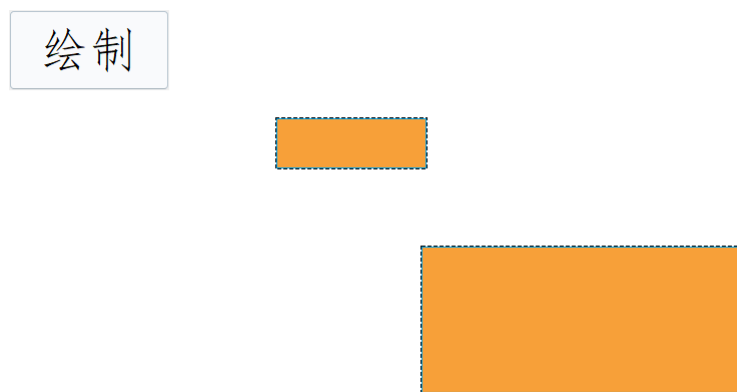
绘制



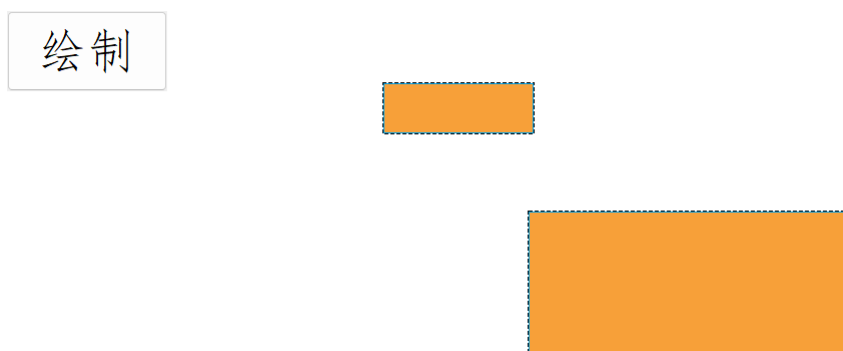
3. 长按 alt 键，同时左键按住选中的矩形的右下角区域可以进行拖动，改变矩形大小和形状，过程中保证矩形的中心点不变，并以鼠标所在位置作为顶点



4. 长按 ctrl 键，同时左键单击多个矩形可以实现同时选中



5. 单个选中或者同时选中的矩形可以通过 backspace 键进行删除，也可以通过左键单击选中的矩形的内部区域进行拖动。（同时选中的矩形会一起删除或者拖动）



6. 选中矩形后，单击背景空白区域可以取消选中

绘制



#### 四、初心与进展

我们组最初的想法是制作一个基于可以构造各种形状以及生成可扩展的思维导图的编辑类软件。这个设计想法与本学期学习的面向对象编程内容是高度相关的，如果把每一个形状视为对象，再将派生的类似树的概念运用于思维导图的扩展，整个程序的设计似乎十分清晰且明确。但事与愿违，在将设计落到实处的过程中，我们遗憾地发现，尽管 Qt 提供了实现诸多功能的工具，比如使用 `QGraphicsView` 以及 `QGraphicsItem` 来实现可拖动的具体形状，但当把它们合并在一起的时候，这些工具互相存在掣肘，难以荟萃一堂。

我们试图跨越从设计方案到具体实现的鸿沟，因此花费了时间而并无所得。在极其有限的时间内，我们选择了以画图的方式实现初稿，即通过鼠标事件的具体坐标与已经存在的形状坐标数据进行比较，进而更改存储的形状描述数据，再重新绘画到画面上。在这样简陋的条件下，初稿实现拖拽功能的时候会导致画面卡顿，同时在功能可扩展性上也不够强。

最终版本我们选择 `QGraphicsView` 和 `QGraphicsScene` 为实现手段，在此基础上，形状拖动和缩放的功能得以顺利实现。