

# 程序设计实训-图片标注作业文档

---

学生：叶宸

学号：2018013400

班级：软82

## 一、实现功能及其使用

---

- 实现功能按大作业要求文档中的顺序给出

### 基本要求

自评各点均符合要求。

各头文件中对类与函数的作用作了注释，在函数的实现中对一些逻辑块做了注释。

采用了OOP的思想，各类封装较良好，关系清晰。

### 功能要求

- 读入
  - 2D图片
    - 支持读入单张图片

使用快捷键Ctrl+O或工具栏中的Open选项选择打开的文件，支持选择一张或多张。
    - 支持读入文件夹中的图片，并可以通过快捷键以及按钮切换文件夹中选中的图片

使用工具栏中的Open Dir打开文件夹，导入文件夹中所有的图片，可以通过点击File List、工具栏中的Next Image(快捷键Tab)、Prev Image(快捷键Shift+Tab)切换图片。
  - 3D图片
    - 支持读入文件夹下的一系列2D图片构成的3D图片

在3D模式下使用OpenDir打开3D图片文件夹即可
  - 检测、分割数据
    - 自动读入图片对应的标注数据

以Open方式打开图片时，会自动搜索图片所在目录及其子目录下以图片名.xml命名的标注信息文件并导入。例如图片为01.png，则会自动搜索01.png.xml为文件名的标注信息文件并读入。

以OpenDir方式打开图片时，会自动搜索打开的文件夹中的所有.xml后缀的文件进行导入。

xml信息支持检测框、分割区域和标签，生成和保存DOM的方式定义在各个类的getElement和createFromElement函数中。

- 手动读入标注数据

可用工具栏中的Import手动选择要导入的xml文件

- 输出

- 能够保存标注的数据

提供两种保存方式：

1. 保存单张图片的标注信息，通过工具栏Save(快捷键Ctrl+S)保存当前编辑图片的标注信息，默认命名为图片名.xml。在Save的时候会自动搜索图片文件夹及其子目录下的图片名.xml文件，如果找到，则会直接覆盖该文件内容。
2. 保存当前打开的所有图片的标注信息，通过工具栏Save All进入。

两种方式均将标注信息存为本程序可识别的xml文件，供后续导入

- 显示

- 支持显示目前选中的图片，3D图片支持显示3个视角下的图片，并以白线标识出当前主视图所在位置
  - 支持显示当前标注的形状，3D图片只支持主视角下的检测标注及分割标注的显示。
  - 支持在Labels中显示不同标注的标签，在Label Type中显示标签的类型及其颜色。不同标签类型有不同颜色，对应标签的方框或区域颜色不同
  - 支持放大图片进行标注，通过鼠标滚轮缩放

- 操作

- 支持检测标注，能通过鼠标操控方框位置

通过工具栏的Rect按钮(快捷键Ctrl+R)绘制方框，Ellipse按钮(快捷键Ctrl+E)绘制椭圆，Curse模式下可拖动任意一个图形。

- 支持分割标注，能通过圆形或正方形画笔绘制分割区域且画笔支持调节粗细，能操控分割区域的位置。不支持绘制完成后的区域的形状调节。

通过工具栏的SquarePen, CirclePen选择正方形及圆形画笔，在工具栏中的滑动条处调节粗细，粗细可选1~80像素

- 3D图片可通过每一层单独分割，绘制完毕每一层的形状后，点击右下角的CreateLabel按钮进行创建

- 支持以鼠标在3D主视图的点击位置来更新另外两个视角看到的切面

- 支持删除某个标注

2D模式下，可通过Curse模式选中图形或者在Labels中选中标签，再通过Delete Label按钮(快捷键Del)删除选中的标签

3D模式下，仅支持通过Labels选中标签，删除方法同上

## 界面要求

- 有菜单栏，包含了文件(File)，编辑(Edit)，以及视图(View)三个菜单

- 文件菜单下包含了所有需求
- 编辑菜单下包含了前进一步(快捷键Ctrl+Shift+Z)、后退一步(快捷键Ctrl+Z)

前进一步和后退一步支持标签的创建、删除以及图片的切换三个事件

- 界面中有状态栏，下部状态栏显示鼠标坐标，右侧状态栏显示标签类型及其名字，标签列表以及文件列表
- 界面中有工具栏

## 加分项

### 加粗为额外功能

- 放大镜，通过工具栏Magnify打开，以一个浮窗形式显示，可以在窗口中调节放大倍率。放大镜窗口不支持编辑。要关闭放大镜，需要再次点击Magnify
- 支持多种格式图片读取，包含jpg、png、gif、bmp、jpeg等
- 支持通过闭合折线分割，通过工具栏Polygon(快捷键Ctrl+P)切换此模式，左键选点，右键结束绘制。
- 支持绘制平滑曲线，通过工具栏Curve(快捷键Ctrl+C)切换，按住左键绘制，松开即结束。
- 支持快捷键
- 支持通过File List中的Line Edit搜索文件，输入文件名后回车。将定位到包含搜索字串的第一个文件
- 支持窗口缩放后自动缩放图片及标注，绘制图形不会超出图片范围，拖动图形不会超出图片范围
- 在绘图窗口中提供十字参照线，并为圆形画笔和正方形画笔提供同等大小的参照线
- 创建标签的对话框会自动记住最后一次创建的标签类型，也支持选择List中的已创建的类型

## 二、模块间逻辑关系

### • Algorithm

- ThreeView类提供了根据2D切片图复原其他视角图片的接口

实现方法为对每层切片取对应行/列的像素进行叠加合成切面图，在合成过程中对每层像素进行一定次数的复制以防效果图过扁

### • Components

- 包含了一系列的窗体组件，继承了Qt自带组件并在类中规定了其样式。用于构建窗口。

其中MyActions定义了用到的Action，MyGraphicsItem对一些QGraphicsItem重写，使它们无法被拖出图片范围。

- 提供了一系列用户交互的信号和槽，例如希望通过鼠标选中某个Label时，LabelsListDock窗体就会发送该Label被选中的信号。信号将被Controller接收并处理。
- 部件类本身只提供交互，负责接收用户操作并将其转发，而不参与例如创建、删除Label的事件处理，信号被转发给Controller后Controller负责所有处理。

### • Controllers

- Shape系列类(Shape, Rect, Ellipse, Polygon, Curve, SquarePen, CirclePen)定义了一系列的形状。

Shape作为虚基类，声明了绘制各类形状接口供PaintScene调用。利用多态方便了PaintScene的形状绘制、方便了各类Shape的统一储存、管理。还声明了将形状信息保存到DOM或从DOM恢复的接口。

各类派生的Shape提供了保存这些形状信息的数据结构（主要是QGraphicsItem）。

提供了一个使得Shape随着窗口的缩放缩放自己的函数（resizeEvent），避免了窗口缩放导致的Item的变形、位移。

- Area类描述了一个Label所占的区域，例如，一个2DLabel的Area即为一张2D图片上的一个Shape（几何图形）；一个3DLabel的Area即为多张2D图片上的Shape，且每张图片有且仅有一个Shape。

Area提供了保存图片及Shape对应关系的数据结构（一个图片名到对应Shape的Map）。

提供了将Area信息保存到DOM或从DOM恢复的接口。

Area负责管理其拥有的Shape的内存。

- AreaController类用于保存绘制中途暂时产生的形状和图片信息，并用这些信息创建Area。

其他类将从AreaController处获取创建完毕的Area。（类似工厂类）。所有PaintScene发送的绘制完成的Shape都经手AreaController处理，避免一些被丢弃的Shape造成内存泄漏。

- Label类包含Label的Type以及Area信息，保存每个Label独一无二的id方便查找，保存Label对应的颜色方便绘制。

提供了获取Label信息的接口。

提供了将Label信息保存到DOM或从DOM恢复的接口。

Label负责管理其拥有的Area的内存。

- LabelController类负责管理程序中的所有Label，其他类只能通过LabelController获取Label信息。统一管理Label的生命周期。

提供了将LabelController信息保存到DOM或从DOM恢复的接口。

提供了各类对Label的查询接口以供其他类访问。

程序中所有的Label都由LabelController创建，并由其销毁，避免了内存管理的混乱。

- FileController类负责管理打开的所有文件，封装了文件打开及关闭的接口，其他类只能通过FileController获取文件信息。

提供了各类对文件的查询接口供其他类访问。

所有文件读写的工作由FileController负责，其他类只能从FileController处获取文件读写后获得的信息。

- PaintScene类负责管理绘图事件，显示图片等。本身不保存任何Label信息。

PaintScene中利用鼠标事件及各类Shape自带的绘图事件完成Shape的绘制，在绘制完成后，Shape会马上被拷贝一份发送给AreaController，同时，原Shape会被马上销毁。

AreaController接受并保存Shape，如果之后选择创建Label，则这些Shape会被Label接管，否则，这些Shape会被AreaController销毁，避免内存泄漏。

- MyCommand定义了一系列可供撤销/重做的命令，继承了QT的QUndoCommand。

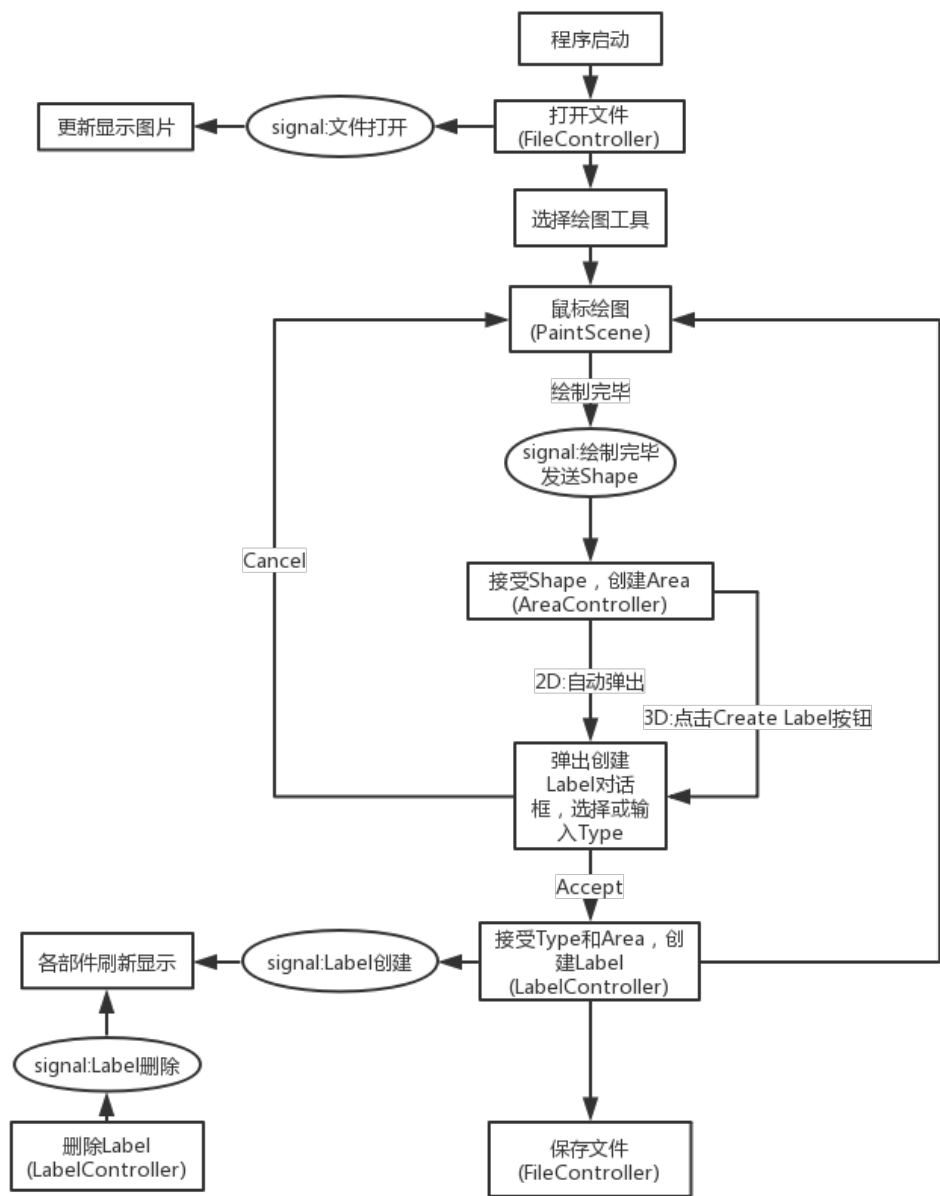
目前支持的重做指令有创建Label、删除Label、切换图片，暂不支持Item的移动的撤销。

在3D中只有点击CreateLabel之后才算创建Label，绘制的临时图形不可撤销。

- **MainWindow**

- 创建、初始化并组合所有Component和Controller。Component和Controller均有2D/3D两份，在模式切换时切换当前的Component/Controller指针。MainWindow中的槽均对当前的Component/Controller指针操作。
- MainWindow不保存任何信息，只提供了交互用的槽函数，并且是所有Component和Controller之间Connect的场所。

### 三、程序运行主要流程



DockWidget和PaintScene在接受用户操作后会发送操作内容给Controller处理。各个Controller在其管理的内容变化时会发出对应的信号，让GraphicsView、PaintScene和各个DockWidget更新显示。

### 四、部分操作说明

主要功能的入口已在第一部分给出

- 画笔工具
  - Curse：左键点击选中标注，拖动移动标注

- Rectangle: 左键点击选择起始点, 左键再次点击选择结束点
- Ellipse: 同Rectangle
- Curve: 按住鼠标绘图, 放开鼠标时结束点会和起始点相连
- Square/Circle Pen: 按住鼠标绘图
- 放大镜
  - 浮窗可改变大小、可拖动, 关闭只能通过再次点击Magnify按钮, 浮窗只显示放大内容, 不可操作

## 五、参考资料

---

- Qt学习之路: <https://www.devbean.net/category/qt-study-road-2/>
- Qt简易画板实现: <https://blog.51cto.com/devbean/244181>
- Qt官方文档: <https://doc.qt.io/>