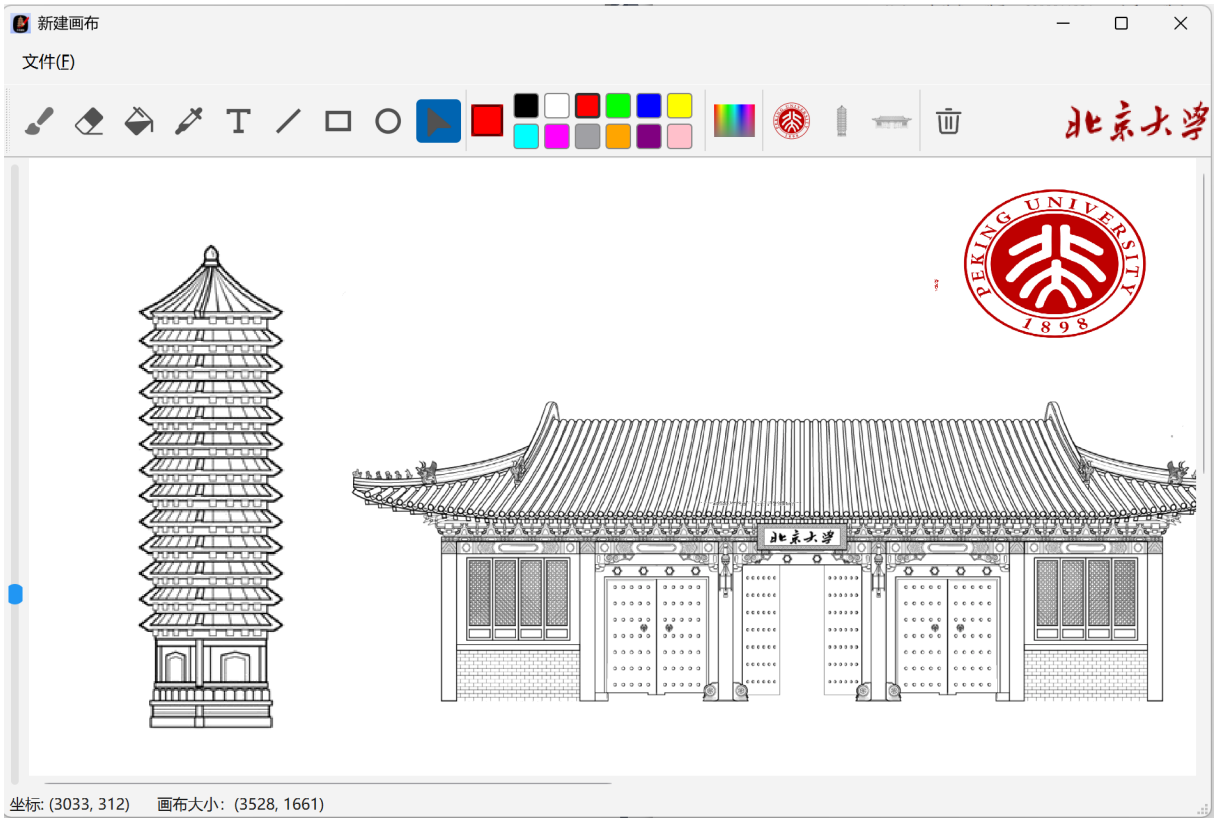
程序设计实习大作业报告

姓名 朱骏豪 学号 2200011084 院系 工学院
姓名 沈嘉琛 学号 2400017739 院系 元培学院
姓名 眭铭涛 学号 2400011018 院系 工学院



目录

1 程序功能介绍

我们组使用 Qt 设计了一款画图软件，支持画笔自由绘画、橡皮擦擦除、颜色填充、调整画笔颜色和粗细等基本功能，同时内置了许多基本形状，包括直线、圆、矩形等，还有具有北大特色的众多贴图，包括校徽、博雅塔、西门等，方便了学校相关的海报等的设计。在界面交互方面，我们实现了画布的缩放与拖动，便于调整绘画位置，同时我们将所有形状与贴图设为可选中的，方便在绘画过程中对画面中的形状进行移动和删除。汇总功能如图所示。

文件操纵

- 打开文件
- 新建画布
- 另存为

绘画操作

1. 工具

- 画笔
  - 颜色修改
  - 线条修改
- 橡皮擦
- 放大镜
- 填充
- 取色器
- 文本框
- 选择（移动+删除）

2. 形状

- 直线
- 矩形
- 圆

3. 特色形状

- 北大校徽
- 博雅塔
- 西门

图 1: 功能展示

2 程序模块与类设计

我们的程序共分成了六个模块，其中有三个界面类模块，一个工具栏模块，一个绘画工具模块和一个形状模块。下面依次介绍一下各个文件模块的类设计细节。

2.1 StartWidget 类

startwidget 类主要定义了有关开始界面的画面显示与交互操作。

图 2: 开始界面

从画面中可以看到，我们除了设计页面以外，主要是为该界面提供了两个按钮，分别与新建画布和打开文件的槽函数连接起来，整个页面的设置比较简单。在开始界面的设计上比较花时间，比如背景的选择，“未名画板”艺术字的生成等等。当前我们选定的背景由大模型生成，和我们项目的名字和功能都比较贴切。

值得一提的是，无论是开始界面的界面设计还是主页面的页面设计，我们都选择用代码去实现，而并没有选择使用 Qt 自带的设计页面。原因有两个，第一个原因是 Qt 的设计页面并不是特别好操作，对于控件的属性和样式修改，也没有代码这么直接。第二个原因是除去开始界面设计外，我们对界面控件设计的需求比较少，我们绝大部分的笔墨聚焦在了绘画逻辑和交互的设计上。

2.2 DrawingWidget 类

这个类中声明的对象是我们的画布，主要定义并实现了有关画布的各项属性以及具体事件。其相关变量和函数主要分成了两类，第一类与显示有关，包括图层对象、视口参数和滚轮对象等等，让我们能实现对绘画内容的正确显示，并且实现放大镜功能，便于观察和修改局部细节；第二类与绘画有关，包括画笔对象、鼠标坐标和形状列表等等，正确地得到当前的操作工具，并通过定义的相关辅助变量来实现鼠标的各种事件，从而实现画面的正确修改。

2.2.1 主要成员变量和函数

i. drawingImage,backgroundImage,shapeImage,originalImage：画布的具体内容，其中 drawingImage 为绘画层，backgroundImage 为背景层，shapeImage 为形状层，originalImage 为三层合并后的原始图像。

ii. viewportX,viewportY,viewportWidth,viewportHeight,scaleFactor： 视口相关参数，即画布缩放和拖动时的显示范围，通过视口操作使得缩放时无需改变原始图像的大小。

iii. convertToOriginalCoordinates.adjustViewport： 视口调整相关函数，在缩放的画布中计算鼠标在原始图像中的坐标并得出视口大小，从而使得显示范围正常。

iv. mousePressEvent,mouseMoveEvent,mouseReleaseEvent： 鼠标相关事件，主要是向 DrawingTools 类中的 DrawingEvent 函数传递当前的鼠标坐标以及画布状态，以便进行绘画操作。

v. wheelEvent,handelHorizontalScroll,handelVerticalScroll：滚轮相关事件，在缩放的图片中滚动滚轮时上下滑动图片，同时调整右侧和下侧滑动条的位置，也可以通过滑动条调整显示范围。

vi. getBackgroundImage，getDrawingImage：得到画布对象的 public 函数，用于其他类对象对其做修改。

2.2.2 实现主要难点

DrawingWidget 类作为画布对象的基础实现，其与其他多个模块均有联系，它需要加入主页面的布局，需要和我们的绘图工具相关联，需要与我们的工具栏做交互，因此也迭代更新了许多版本。在这里，我们的图层从一开始的单绘画层修改成了最终三层：绘画层、背景层与形状层。其中背景层的存在是用于适配打开文件操作的，将我们的文件作为我们的背景来实现。形状层则与我们后续将要提到的形状操作息息相关。

我们在绘画工具中有着放大镜功能的规划，但放大镜功能与其将其说成一种绘画功能，更多的是一种视图操作，后续我们将其放在这里。就算不实现放大镜功能，对于绘图软件而言，界面大小的改变和画布大小的调整也是我们必须要解决的一个问题，这也是上述众多有关视口参数和函数存在的原因。

我们在 adjustViewport 函数的实现过程中需要计算当前的图像坐标和原始图像的坐标变换，形成正确的图像，并且在当前放大镜模式下作出的修改，也必须反映在原图像中。Qt 有自带的图片放大功能，但缩小放大后，图片就会因插值精度不足而变得模糊。我们最终通过保存原始 1：1 大小的图像来实现这样操作，确保过程中的分辨率不会有所损失。

2.3 DrawingTools 类

DrawingTools 类中声明了我们的绘画工具，主要定义和实现了与具体绘图相关的操作。其主要的内容也可以分为两类，第一类是各种绘画工具的具体实现，第二类是一些额外的绘画修改，比如修改我们的线条粗细和颜色。

2.3.1 主要成员变量和函数

i. color,pen,size,mode： 这些静态成员变量主要定义了画笔的属性，包括颜色、粗细、样式、类型等。在这里，我们通过 mode 变量来实现不同的绘画工具，比如，当 mode 为 0 时是我们的

画笔，当 mode 为 1 时是我们的橡皮擦。这里我们并没有通过派生的方式来声明不同的绘画工具，不然每次调整绘画工具就生成一个绘画工具对象，这是不合理的。

ii. setColor,setPen,setSize,setMode：为静态成员变量提供外部接口便于修改。

iii. DrawingEvent：最关键的绘图函数，重载为是否需要更改形状图层两个版本。定义了不同工具的具体绘画操作。

iv. ShapeDrawing：若为形状绘制，则通过多态调用 shape 类中对应形状的绘制函数进行绘制。

v. scanLineFill：颜色填充对应函数。通过扫描线方法进行 DFS，每次判断选中的像素颜色是否与当前区域颜色相同，若相同则进行填充。

vi. ColorPicker：取色器对应函数，得到不同图层当前鼠标坐标上的颜色，对比得到当前画布上的颜色。

vii. text,textPosition,textSize,setText,setTextSize：文本对应函数，通过当前的线条粗细和颜色可调整字体的大小和颜色。

2.3.2 不同工具实现

1. 画笔: 对应 mode 0，自由绘画功能。通过不断更新鼠标位置并在相邻两位置间连线实现相应效果。

2. 橡皮擦：对应 mode 1，擦除绘画内容功能。使用透明画笔对绘画图层进行擦除。

3. 填充：对应 mode 5，对封闭区域进行颜色填充。通过 scanLineFill 算法函数得到封闭区域，从而进行颜色填充。

4. 选择：对应 mode 6，选择不同的形状对象从而进行移动，删除等功能。对比鼠标坐标和各个形状区域的相交情况得到当前的形状对象，并实现选择下的操作。

5. 取色器：对应 mode 7，得到当前位置下的颜色。直接查找得到当前画布下鼠标坐标的颜色，同时会记录上一个绘画工具并自动返回。

6. 文本：对应 mode 8，插入文本功能。目标位置插入文本框实现。

7. 删除：形状删除操作，只在选择形状模式下可以进行。将形状从我们的形状列表中删去，重绘各个形状。

2.3.3 实现主要难点

DrawingWidget 类和 DrawingTools 类是我们实现过程中最复杂也 debug 最多的两个模块，一部分原因在于我们想要实现的操作的矛盾性。其实观察不同的绘图软件可以发现：形状的选择功能和绘画内容的随意擦除功能往往不可兼得。因为“被擦除的形状”应该如何判定是一个比较复杂的问题。这也是我们在后续单独对形状添加删除操作的原因，让形状通过删除操作擦除，且不认可形状的橡皮擦操作，防止形状在擦除和选择操作上的矛盾，但这只是一个保护策略，比较遗憾的是，我们并没有想到很好的办法去解决这个矛盾。

2.4 Shapes 类

Shapes 类中主要定义和实现了所有形状的基本属性，包括是否选中，画笔样式，边框矩形等。Rectangle,Ellipse,Line 均派生自该基类。在这些派生类中分别根据对应的形状实现了 draw,contains, move,changeSelectedWidget 等方法。另外，对于北大特色贴图，我们也把它们看作一种特殊的贴图在该类中派生。

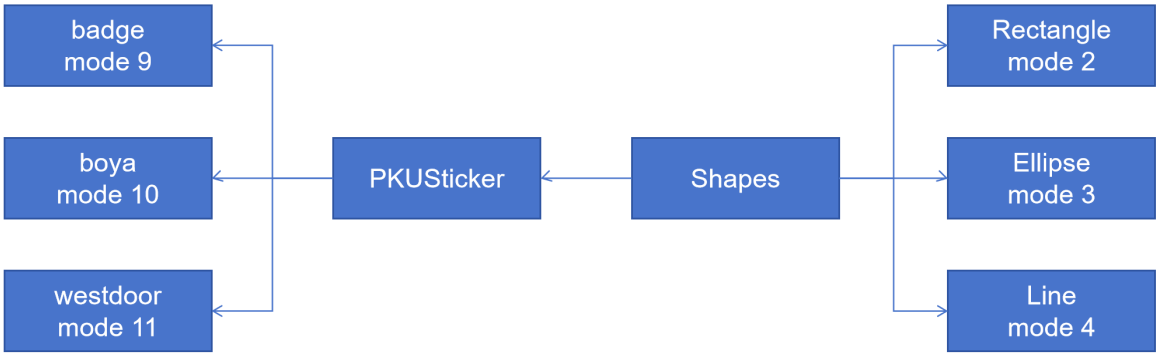


图 3: 形状类派生图

其中，PKUSticker 由 Shapes 派生，而不同的贴图通过传入的图片不同来区分。

2.4.1 主要成员变量和函数

i. draw：通过 Qt 内置的形状绘制函数在边框矩形内进行绘制。

ii. contains：判断鼠标坐标是否在形状范围内，从而确定形状是否被选中。

iii. move：根据鼠标移动方向和距离，改变边框矩形坐标并进行重绘从而实现形状的拖动。

iv. changeSelectedWidget：根据是否被选中改变选中状态，从而判断是否显示选中提示框。

2.5 DrawingToolBar 类

DrawingToolBar 类定义了有关工具栏的相关动作和样式。

图 4: 工具栏样式

如图所示，工具栏左侧为各种工具和形状，中间为颜色选择面板，左边为当前颜色，右边可点入颜色盘自调颜色，往右是北大特色工具，最后是形状删除工具

2.5.1 主要成员变量和函数

i. toolGroup,colorAction,deleteAction：toolGroup 为工具组，包含了各种工具的 Action，确保每个时刻只有一个工具会被选择。colorAction 和 deleteAction 分别为颜色动作和删除动作。

ii. setupTools：最主要的函数，建立了整个工具栏的整体布局。

iii. createColorPalette,colorChanged,updateCurrentColorIndicator：颜色修改相关函数，确保画面颜色的正确显示和颜色修改的正确实现。

2.6 MainWindow 类

MainWindow 类定义了主页面中的整体布局，实现了文件打开、文件另存为和新建画布等文件操作。

2.6.1 主要成员变量和函数

i. newActionSlot：新建画布槽函数，当用户点击画布菜单项时，调用 drawwidget->clear() 方法清空绘图区域, 并更新窗口标题。

ii. openActionSlot：打开文件槽函数，当用户点击打开文件菜单项时，打开文件选择对话框，选择图像文件并将其设置成画板的背景图像。通过 QFileDialog 库中的 getopenFileName 方法实现选择要打开的图像文件。firstTimeOpen 标记是否是第一次打开文件，默认为 false，若为 true 则调用 setupDrawingPage 方法设置绘图页面。如果用户选择的图像文件加载成功，则调用 drawwidget->setBackgroundImage 方法将其设置为背景图像，更新绘图区域，显示状态栏和菜单栏，并调整窗口大小。ScaleFactor 和 scaledSize 通过计算实现加载图片自适应。

iii. saveActionSlot：保存文件槽函数，当用户点击保存文件菜单项时，打开文件保存对话框，将绘图区域的背景图像和绘图图像合并并保存为一个图像文件。通过 QFileDialog 库中 getSaveFileName() 方法实现打开文件保存对话框，让用户选择保存文件的路径和文件名。如果 drawwidget 存在，创建一个新的 QImage 对象 combinedImage，并将其填充为透明色，使用 QPainter 将背景图像和绘图图像绘制到 combinedImage 上，再将 combinedImage 保存到用户选择的文件中。

3 项目分工

- **朱骏豪**：主页面的基本设计，开始界面实现，北大特色形状加入，取色器功能实现，演示视频制作，项目 debug。
- **沈嘉琛**：DrawingTool 类的基础创建，Shapes 类的基础创建，画笔、橡皮擦、填充、选择、放大镜功能实现，演示视频制作，项目 debug。
- **眭铭涛**：基础的文件操作，文本功能实现，项目 debug。

4 项目总结

- **朱骏豪**：

  作为一名信双选手，在大三这个学期才选上了程序设计实习，这次大作业的感受也比较多吧。从工作时间上讲，这次大作业确实自己花的时间比较多的一次大作业，从刚接触 Qt，不断学习 Qt 的使用方法到项目的规划安排，从界面的基本设计到演示视频的制作，和组员们一起一步步完成了这个大作业，即使最后没有进入路演，从中学到的丰富知识也已经受益颇多了。

  作为队长，本次大作业自己感受比较多的还是时间的规划吧。比起许多作业不断地赶 ddl，本次大作业，我们从一开始就规划好了想要实现地功能，并在后续每周讨论一次，分配好本周大家需要去完成地任务，在过程中大家也积极讨论遇到的各种 bug，主动去添加和完善项目的功能。从团队协作上来讲，我会给我们团队打满分。

  在项目的设计实现上，我们确实还有很多可以去完善的地方，即使大作业结束了，我们仍会试着在我们的 github 库不断更新完善我们的项目，试着去实现一个真正的“未名画板”。

- **沈嘉琛**：

  在这次大作业中, 我了解了 Qt 的基本使用与功能, 加深了我对面向对象程序设计的理解. 在程序 debug 过程中, 我掌握了一些项目创作中的方法与技巧, 收获了丰富的经验. 同时由于这是我上大学以来第一个小组大作业, 在完成大作业的过程中我与其他组员积极交流内容设计与代码实现方面的种种问题, 在交流中我的代码水平与思维得到了提高, 锻炼了我的合作能力, 并且掌握了利用 Git 管理项目版本的方法。

  同时, 大作业在路演中落选也让我意识到我们组的大作业虽然完成度较高, 在创新与创意方面仍有所欠缺, 缺少吸引人的亮点. 因此在今后的学习过程中我要有意识地培养相关方面的能力与素养。

- **眭铭涛**：

  这次 Qt 大作业是我上大学以来第一个小组合作实现的程序设计项目，在完成项目的过程中我熟悉了 Qt 这一 C++ 语言程序开发框架的基本使用方法和功能，有效地训练了课程上学到的面向对象程序设计方法，也加深了对 OOP 的理解，更重要的是收获了小组成员共同努力为某一目标而奋斗的集体主义和团结协作精神。Debug 的过程固然是辛苦的，但这也丰富了我的实践经验，在和小组成员的交流探讨中我成功拓宽了自己的视野，提高了编程思维的深度和广度，同时，也正是这次 Qt 大作业促使我学习 git 的基本操作，养成了良好的代码管理习惯。

  虽然最后我们小组的项目没能成功入选路演，没能画上一个完美的句号，但这些许遗憾也促使我们反思项目的不足之处，让我们意识到创新创意方面的重要性，为我们未来的学习之路铺砖添瓦。