

《计算机图形学》系统使用说明书

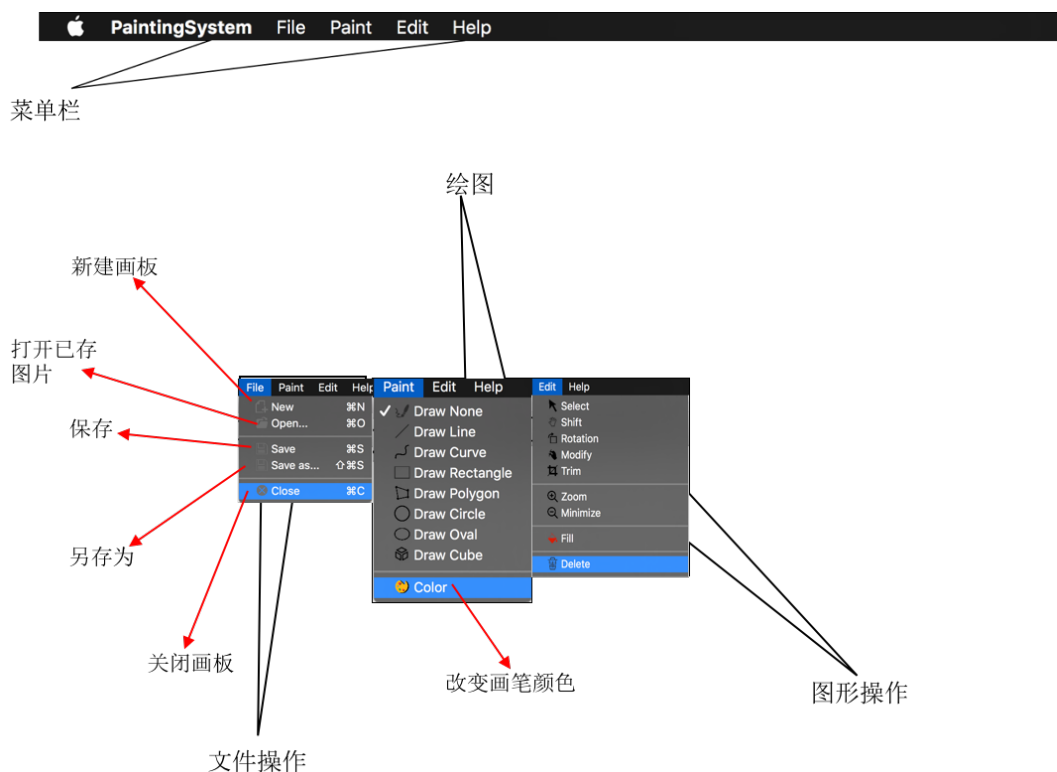
王亚楠 141220107 wyn1996@hotmail.com

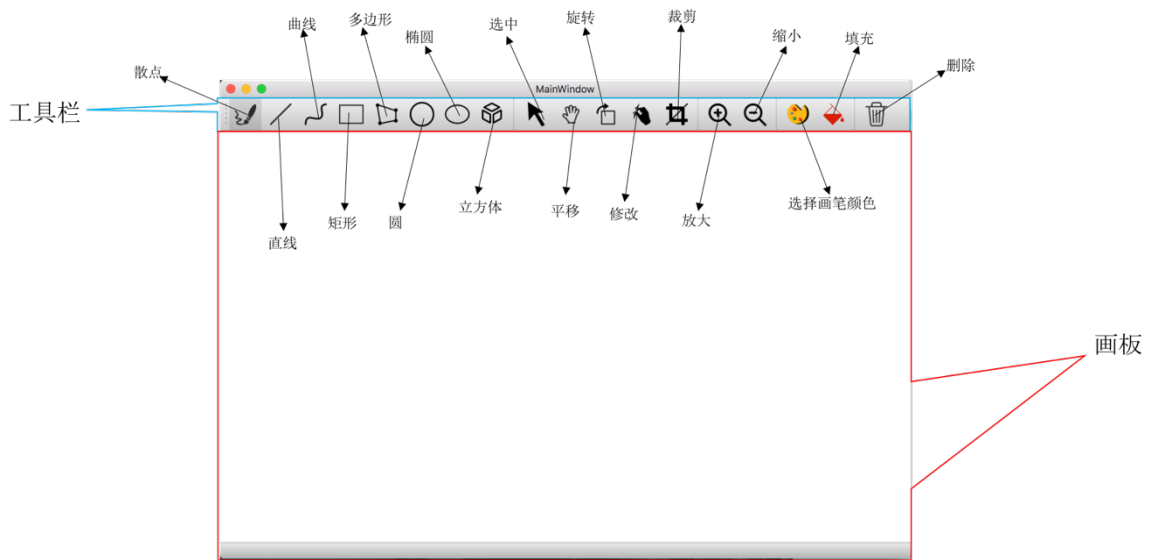
(南京大学 计算机科学与技术系, 南京 210023)

摘 要: 基于 Qt 完成了整体系统的设计工作, 实现了系统的界面设计和框架设计。确定并完成了直线、圆、椭圆和多边形、曲线、立方体这四种图形的绘画算法。同时实现了直线、圆、椭圆、多边形、曲线的平移, 直线、多边形、曲线、立方体的旋转, 直线、圆、椭圆、多边形、曲线的修改和缩放, 直线和多边形的裁剪, 圆、椭圆、多边形的填充。此外, 还增加了绘图时画笔颜色的选择。

1 系统简介

PaintSystem 系统是根据计算机图形学原理, 在 Qt 平台框架上开发的、允许进行图形绘制和图形操作的系统软件。下图是该系统的操作界面, 从上到下分为 3 个部分, 分别是最上部的菜单栏 (由于系统原因菜单栏被放在了电脑屏幕的最上部)、工具栏以及画板。





工具栏和菜单栏的功能基本上是相同的，工具栏的存在是为了更加方便用户的操作，不用去菜单栏中再寻找所需要的操作按钮，直接通过点击工具栏中的对应按钮便能进行操作。

工具栏中的按钮有几种不同的类型，也用分隔符加以区分，从左到右分别是绘图、操作以及改色和填充。当选中某一个按钮时，或出现阴影表示已选中。下面按照按钮出现顺序依次介绍。

新建：重新创建一张画板，同时允许用户对画板的大小、颜色进行设置。同时，会提示用户是否对旧画板的操作进行保存。

打开：打开之前保存的画板，打开后的画板无法在对其中对象进行操作。

保存：点击后弹出对话框，选择保存路径，将当前画板中的内容保存为 **png** 格式到指定位置。

另存为：将当前画板的内容以 **png** 格式保存到另一指定位置。

关闭：关闭当前画板，同时会提示是否对当前画板的操作进行保存。

散点：进入相应的图形绘制模式，通过鼠标操作进行散点的绘制。

直线：先选择绘制直线的算法（**DDA** 或 **Bresenham**），然后进入相应的图形绘制模式，通过鼠标操作进行直线的绘制。

曲线：先选择绘制曲线的算法（**B 样条**或 **Bezier**），然后进入相应的图形绘制模式，通过鼠标操作进行曲线的绘制。

矩形：进入相应的图形绘制模式，通过鼠标操作进行矩形的绘制。

多边形：进入相应的图形绘制模式，通过鼠标操作进行多边形的绘制。

圆：进入相应的图形绘制模式，通过鼠标操作进行圆的绘制。

椭圆：进入相应的图形绘制模式，通过鼠标操作进行椭圆的绘制。

立方体：在另一个新窗口中显示一个立方体。

选中：选中一个图形。

平移：通过鼠标操作对当前选中的图形进行平移操作。

旋转：通过鼠标操作对当前选中的图形进行旋转操作。

修改：通过鼠标操作对当前选中的图形进行修改操作。

裁剪：绘制一个矩形的裁剪窗口对选中图形进行裁剪操作。

放大：通过点击对当前选中的图形进行放大操作。

缩小：通过点击对当前选中的图形进行缩小操作。

选择画笔颜色：通过点击跳出颜色选择框进行画笔颜色的选择操作。

填充：通过点击跳出颜色选择框进行填充颜色的选择操作。

删除：删除选中的图形。

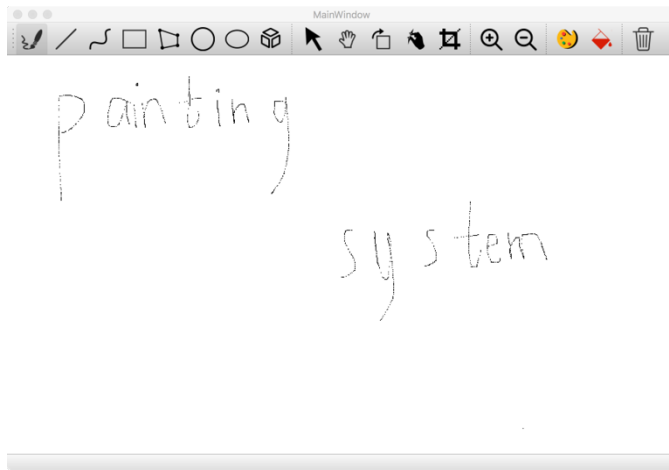
2 功能介绍

2.1 图形绘制

PaintSystem 系统提供图形绘制功能，图形绘制是它所有功能的前提和基础，系统能够绘制的图形包括：散点、直线（线段）、矩形、圆、椭圆、多边形、曲线和三维六面体。其中直线、矩形、多边形、圆、椭圆都是动态画图，即鼠标左键按下时记录起始点，鼠标移动过程中会不断显示当前图形的形状，当鼠标左键放开时该图形的绘制操作才会结束。而曲线是通过点击来设置控制点实现绘制的，三维六面体是系统默认绘制好的。

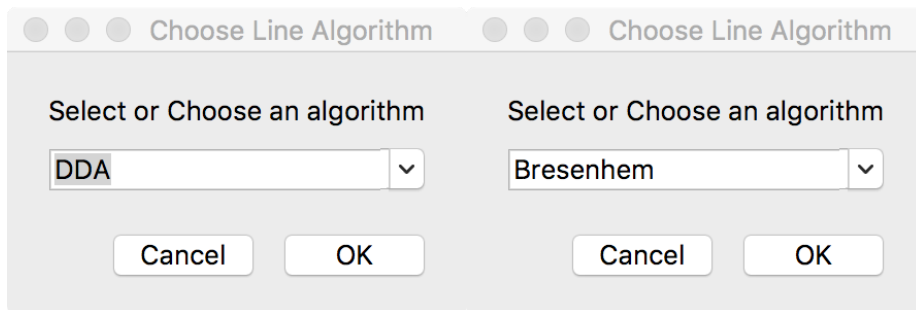
下面是各种图形的绘制效果图：

2.1.1 散点

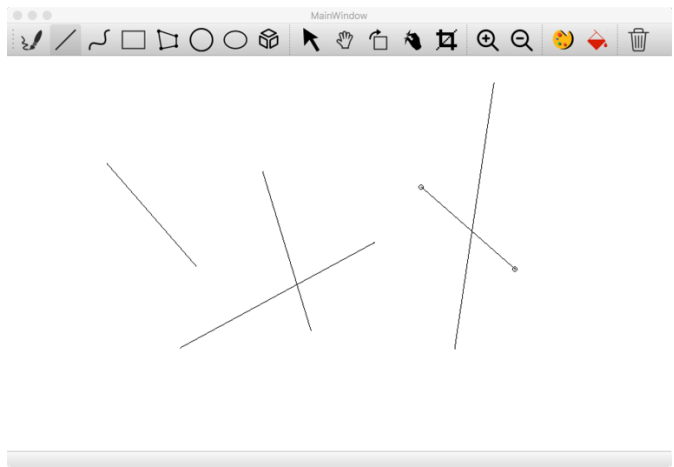


2.1.2 直线

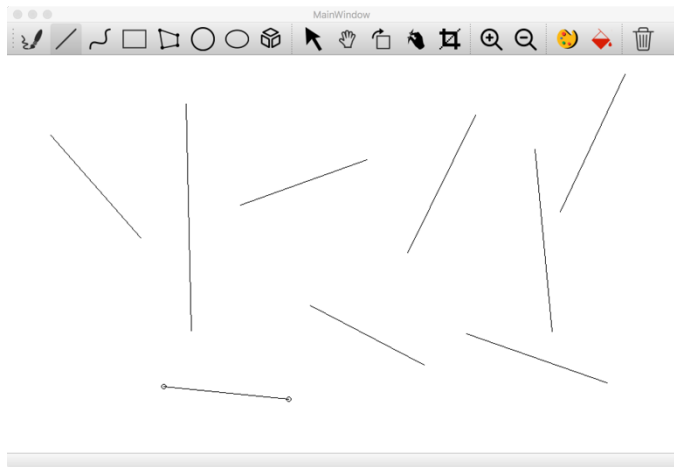
选择算法窗口



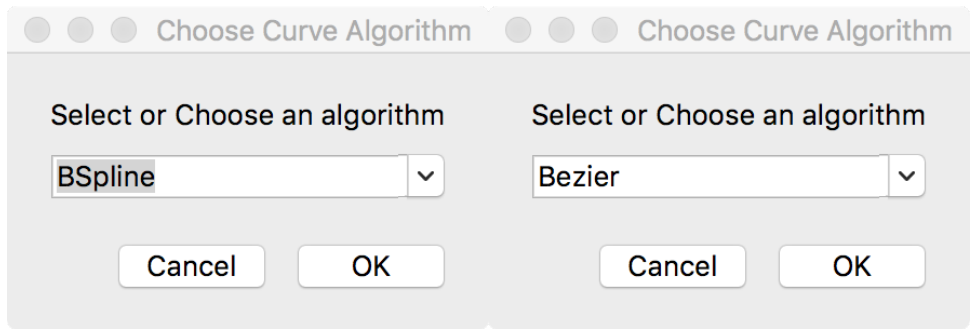
DDA 算法展示图：



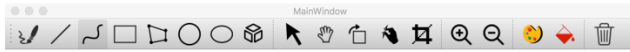
Bresenham 算法展示图:



2.1.3 曲线
选择算法窗口



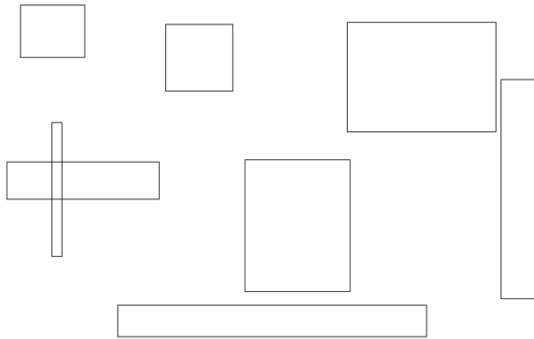
B 样条曲线展示图:



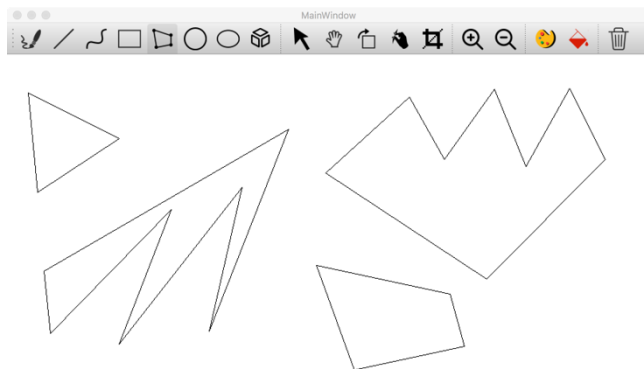
Bezier 曲线展示图:



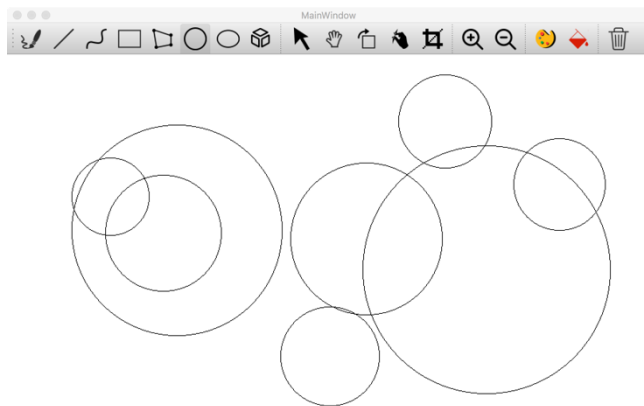
2.1.4 矩形



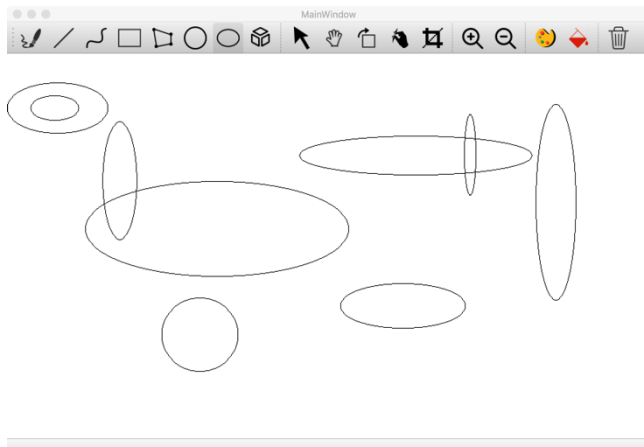
2.1.5 多边形



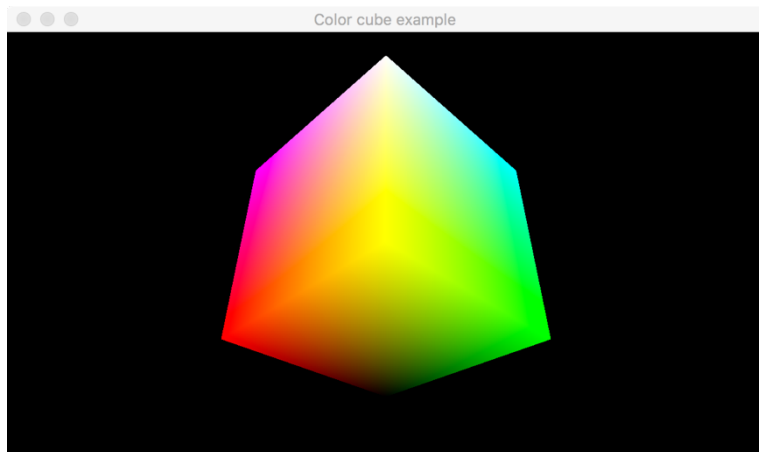
2.1.6 圆



2.1.7 椭圆



2.1.8 三维六面体



2.2 图形变换

图形变换功能主要针对二维图形的变换，三维六面体只有旋转变换。

图形变换功能主要包括：平移、旋转、缩放。但是由于某些图形的特殊性以及开发平台的局限性，一些变换操作难以实现或者实现没有意义，所以各类图形变化支持的图形又有所不同，如下列所示：

平移：支持所有图形。

缩放：支持所有图形。

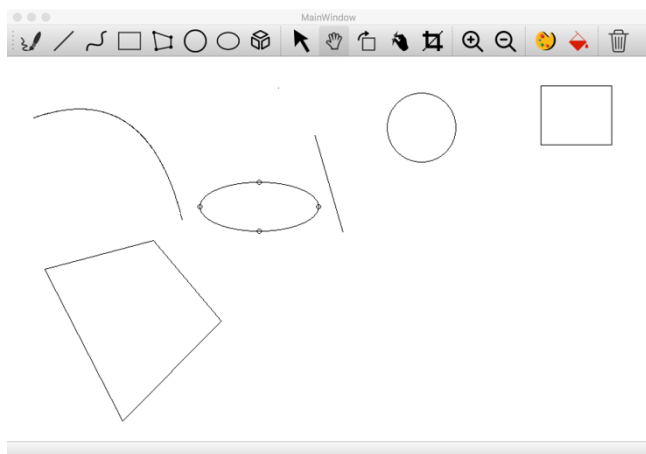
旋转：支持直线、曲线、矩形、多边形。

简单来说，系统中未实现圆和椭圆的旋转，其余功能均全部实现。首先，对于圆而言，旋转操作没有实际意义，旋转后的圆与原图无异；对于椭圆，因为其绘图算法使用的是中点椭圆算法，绘制椭圆只用到两个控制点而不是图形上所有点，算法中绘制的关键是一个水平垂直的 XY 轴，而旋转后标准坐标轴也发生了旋转，再次绘制就需要更加复杂的操作，所以也将其忽略掉了。

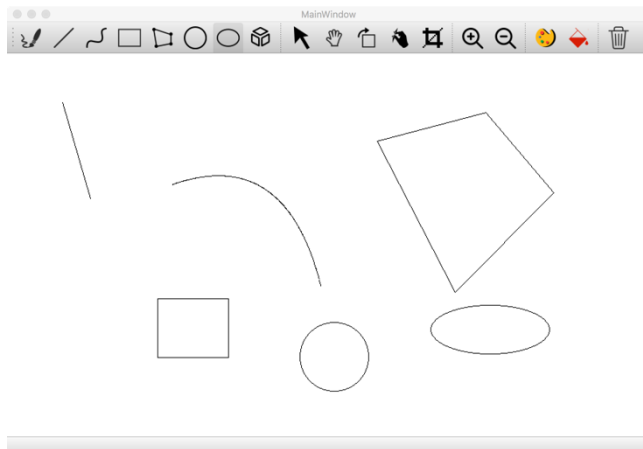
下图是图形变换的效果图。

2.2.1 平移

原图：

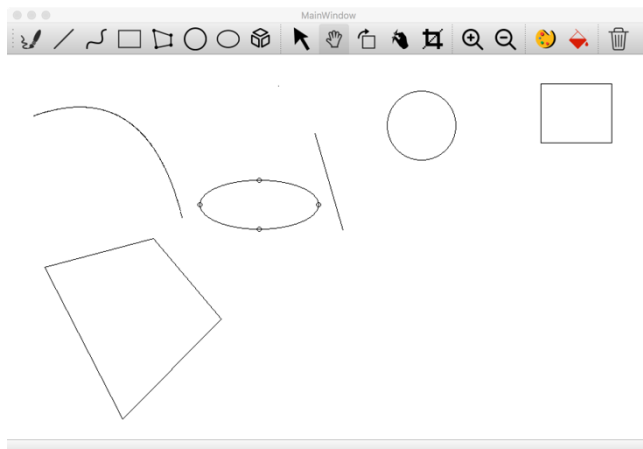


平移后的图：

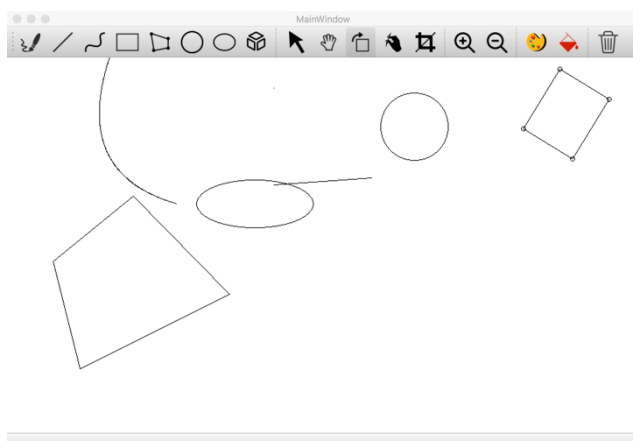


2.2.2 旋转

原图：

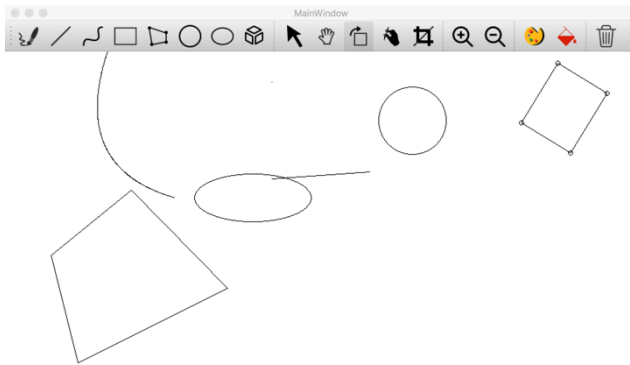


旋转后的图：

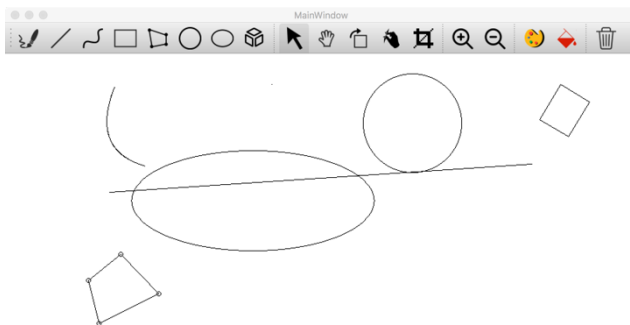


2.2.3 缩放

原图：



缩放后的图:



2.3 图形编辑

图形编辑功能是指在一个图形绘制好了以后,允许再对其进行操作,即修改图形中的控制点来间接的改变图形的形状以达到用户的标准。图形编辑功能支持所有的二维图形。具体来说,编辑功能是指:

修改直线的起始点和终点的位置;

修改矩形四个顶点的位置来改变矩形的形状;

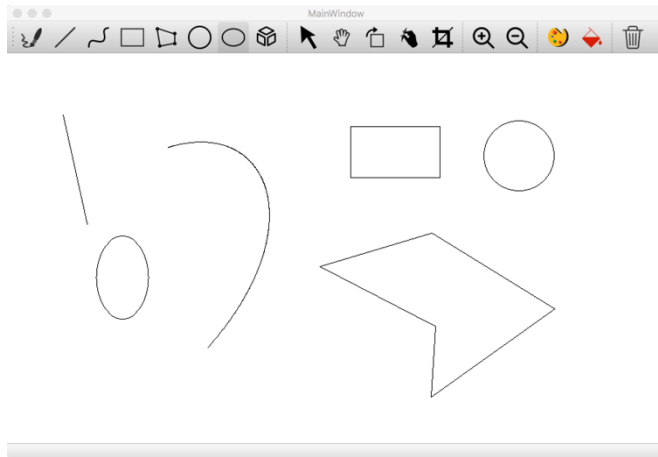
改变圆的半径;

改变椭圆的长半轴或短半轴;

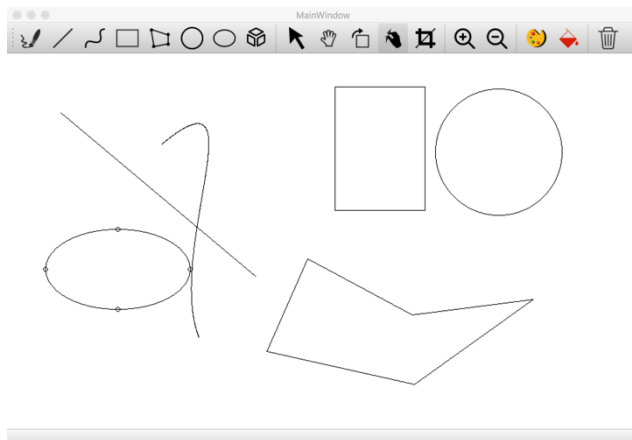
编辑多边形或者曲线的某个控制点来改变多边形或曲线的形状。

图片编辑过程也是一个动态的过程。

原图:



编辑后的图：



2.4 裁剪

利用裁剪窗口裁剪出用户需要的部分，系统仅支持对直线、矩形和多边形的裁剪。裁剪只对于选中图形进行，忽略其他为被选中但在裁剪框中的图形。

原图：



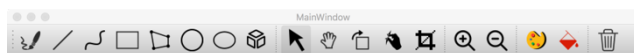
裁剪后的图:



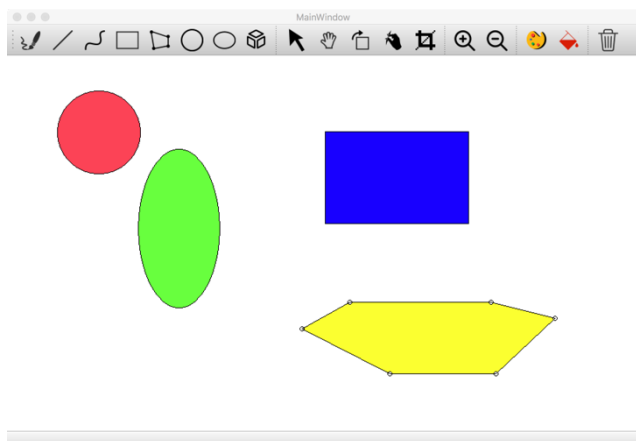
2.5 填充

填充功能是指将封闭图形的内部区域填充上指定的颜色。能够进行填充的图形有：矩形，圆，椭圆和多边形。填充后的图形不影响正常的图形变换。**填充之前需选中被填充图形。**

原图:



填充后的图:

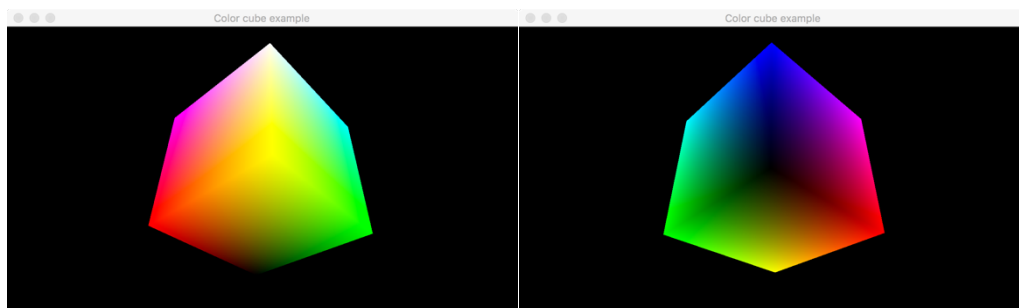


2.6 三维六面体

PaintingSystem 系统支持三维六面的显示、旋转，并且能够消隐。三维六面体会出现在另一个新的窗口中。

原图：

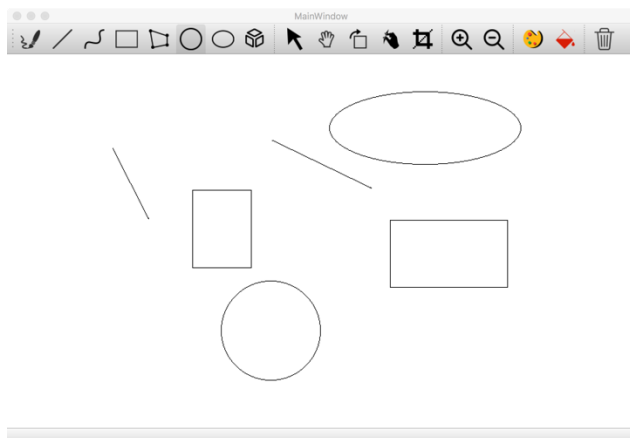
旋转后的图：



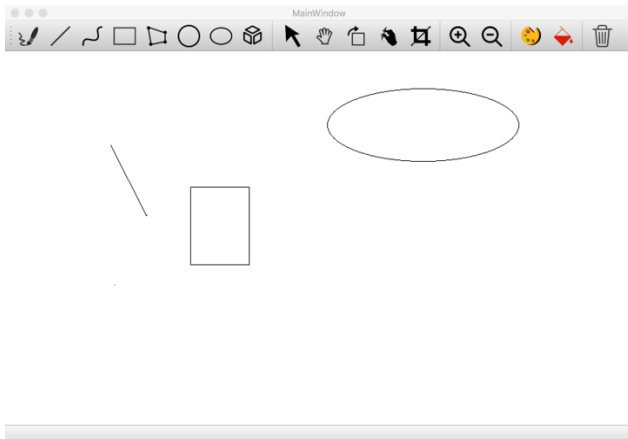
2.7 图形删除

删除所选中的图形。

原图：

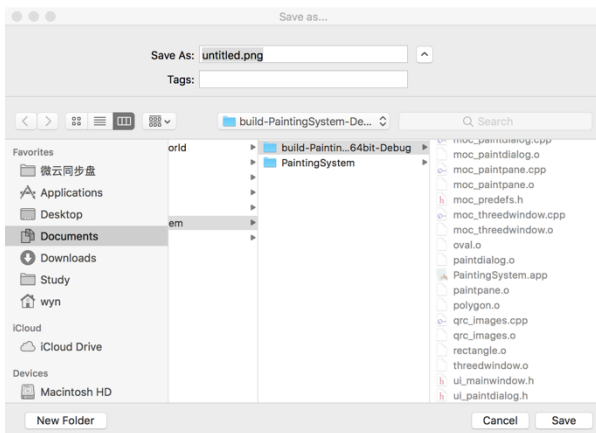


删除后的图：



2.8 数据存储

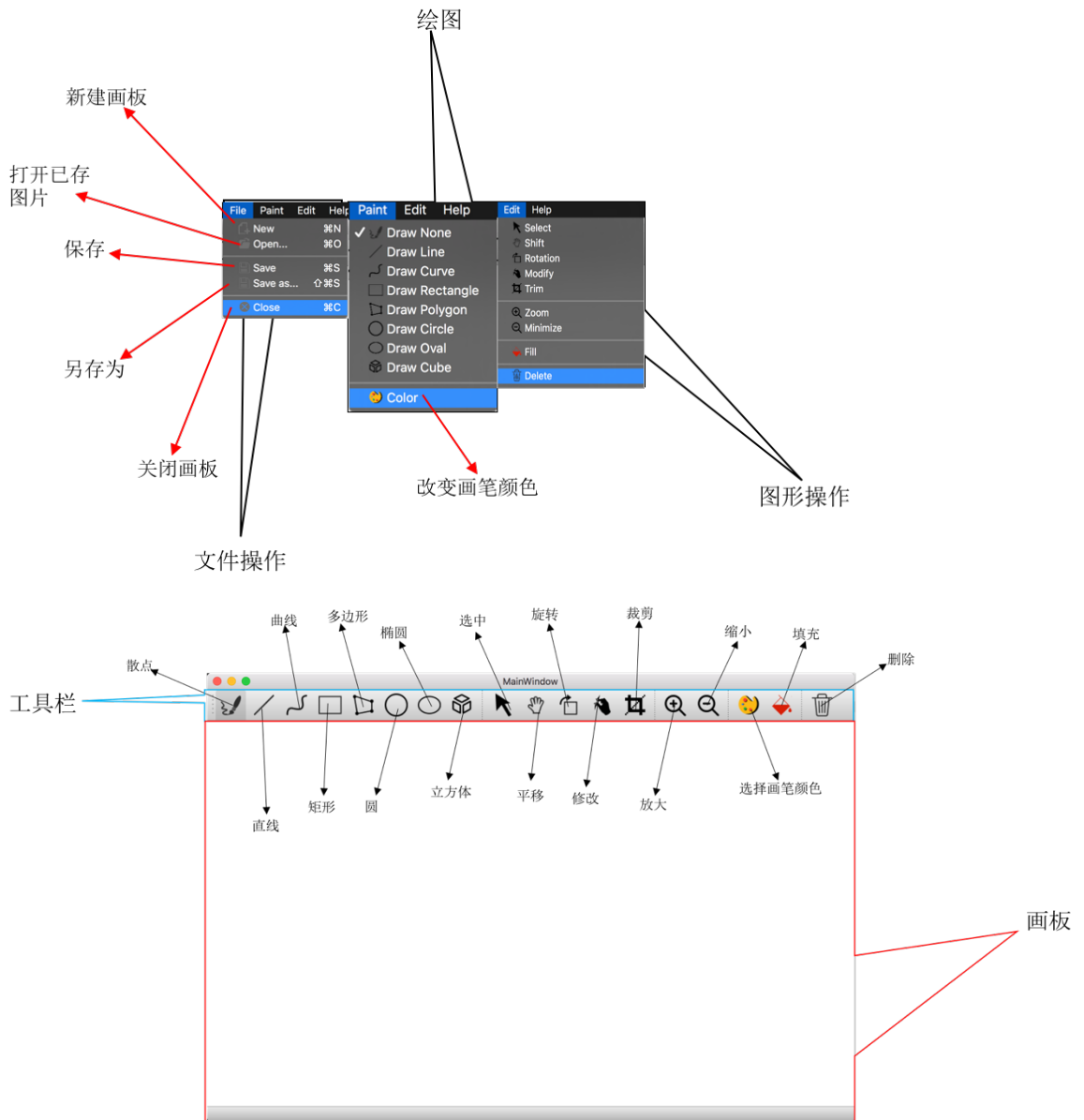
点击保存按钮可以进行数据保存，将当前画板上的图形保存为 png 格式的图片，用户可以指定存储路径和文件名。



3 操作说明

3.1 工具栏和菜单栏





工具栏和菜单栏的功能基本上是相同的，工具栏的存在是为了更加方便用户的操作，不用去菜单栏中再寻找所需要的操作按钮，直接通过点击工具栏中的对应按钮便能进行操作。

工具栏中的按钮有几种不同的类型，也用分隔符加以区分，从左到右分别是绘图、操作以及改色和填充。当选中某一个按钮时，或出现阴影表示已选中。下面按照按钮出现顺序依次介绍。

新建：重新创建一张画板，同时允许用户对画板的大小、颜色进行设置。同时，会提示用户是否对旧画板的操作进行保存。

打开：打开之前保存的画板，打开后的画板无法在对其中对象进行操作。

保存：点击后弹出对话框，选择保存路径，将当前画板中的内容保存为 **png** 格式到指定位置。

另存为：将当前画板的内容以 **png** 格式保存到另一指定位置。

关闭：关闭当前画板，同时会提示是否对当前画板的操作进行保存。

散点：进入相应的图形绘制模式，通过鼠标操作进行散点的绘制。

直线：先选择直线算法（DDA 或 Bresenham），然后进入相应的图形绘制模式，通过鼠标操作进行直线的绘制。

曲线：先选择曲线算法（B 样条或 Bezier），然后进入相应的图形绘制模式，通过鼠标操作进行曲线的绘制。

矩形：进入相应的图形绘制模式，通过鼠标操作进行矩形的绘制。

多边形：进入相应的图形绘制模式，通过鼠标操作进行多边形的绘制。

圆：进入相应的图形绘制模式，通过鼠标操作进行圆的绘制。

椭圆：进入相应的图形绘制模式，通过鼠标操作进行椭圆的绘制。

立方体：在另一个新窗口中显示一个立方体。

选中：选中一个图形。

平移：通过鼠标操作对当前选中的图形进行平移操作。

旋转：通过鼠标操作对当前选中的图形进行旋转操作。

修改：通过鼠标操作对当前选中的图形进行修改操作。

裁剪：绘制一个矩形的裁剪窗口对选中图形进行裁剪操作。

放大：通过点击对当前选中的图形进行放大操作。

缩小：通过点击对当前选中的图形进行缩小操作。

选择画笔颜色：通过点击跳出颜色选择框进行画笔颜色的选择操作。

填充：通过点击跳出颜色选择框进行填充颜色的选择操作。

删除：删除选中的图形。

3.2 快捷键说明

系统设置有一系列的快捷键，在熟悉使用流程后能够更方便用户的操作。

Ctrl+N：新建画板

Ctrl+O：打开图片

Ctrl+S：保存画板


Ctrl+C：关闭画板

3.3 双控制点的动态绘图

通过双控制点进行绘制的图形有 **直线**、**矩形**、**圆** 和 **椭圆**。所谓的动态绘图是指绘制过程中会动态地显示所绘制图形的当前形状，以供用户参考。动态绘图时是通过鼠标配合完成的，即鼠标左键按下记录起始点，鼠标移动过程中不断显示当前图形的形状，当鼠标左键放开后该图形的绘制操作结束。

3.4 多控制点的绘制


通过多控制点进行绘制的图形主要是 **曲线**。下面主要介绍其操作细节。

用户通过选中曲线绘制按钮 ，选择所需要的绘制算法，然后通过鼠标单击设置控制点，系统会根据已经创建的控制点不断绘制曲线，最后通过双击结束曲线绘制。其中 **B 样条是三次 B 样条，所以至少需要绘制 4 个点**

3.5 多过程绘制


通过多过程绘制的图形主要是 **多边形**。所谓多过程绘制是指需要不断绘制多边形的边直至构成完整的多边形。首先左键按下，然后移动鼠标放开左键，完成第一条边的绘制，然后在上一个点处按下左键，移动鼠标，放开左键，完成第二条边的绘制，重复上述步骤，直至最后一条边，将最后一条边的终点放到多边形的第一条边的起点处便自动完成整个多边形的绘制。


3.6 选择



在工具栏上点击，便进入选择模式，在画板上点击你所要选择的图形，被选中的图形会显示其控制点。


3.7 变换及编辑的操作说明

在进行变换和编辑之前需要首先选择操作图形，只有在选中操作图形后方可进行变换和编辑操作。


平移：点击，然后左键单击被选中图形且不放开，移动鼠标便可实现平移。

旋转：点击，然后左键单击被选中图形且不放开，移动鼠标便可实现旋转。


缩放：点击便可实现被选中图形的放大；点击便可实现被选中图形的缩小。

编辑：当图形被选中时，图形会显示其控制点，点击，左键单击被选中图形的控制点且不放开，移动鼠标便能完成对控制点的移动进而进行图形的编辑。（圆显示的控制点为其圆心，而编辑需要左键单击其边界才可实现圆的编辑，而非圆心）


3.8 裁剪

裁剪前需先选中被裁减图形，然后点击进入裁剪模式；然后绘制裁剪窗口（过程类似于绘制矩形），此时的裁剪窗口为红色的矩形窗口；当抬起鼠标左键时裁剪窗口完全确定便开始进行裁剪操作。裁剪后只会保存被选中图形在裁剪框中的部分。（裁剪操作只能对直线和多边形进行操作，其他图形不会被裁剪）裁剪操作后的图形不影响其原来的图形变换以及裁剪、填充等操作。

3.9 填充

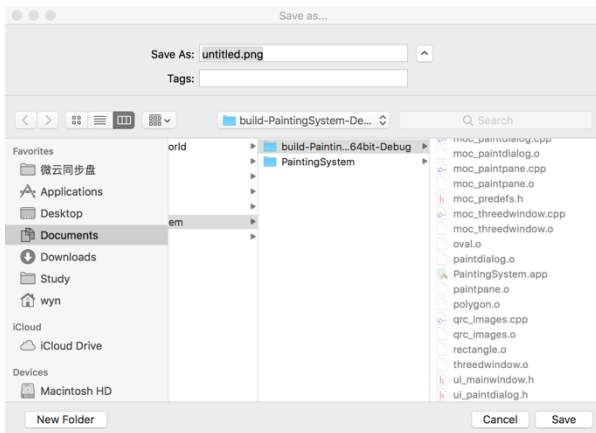
填充前需先选中被填充的图形，然后点击进入填充模式，首先会跳出颜色选择框进行填充色的选择，然后鼠标点击当前选中图形的内部区域时就会对其进行填充。填充后再次点击则会撤销之前的填充。对于已被填充的图形的再次填充，点击图形的内部区域图形会先复原，即不无填充色，然后再次点击图形的内部区域，图形会被填充上新的填充色。因为采用了种子泛滥填充算法，如果对填充物体进行编辑并且使得种子点位于图形外可能会触发填满整个画布的情形。如果对填充物体进行裁剪，裁剪后的物体仍然填充原来的颜色。其中对于凹多边形，填充后不应该进行裁剪操作，否则可能使得种子点位于图形外。

3.10 删除

选中所要删除的对象，然后点击，选中图形便会删除掉。

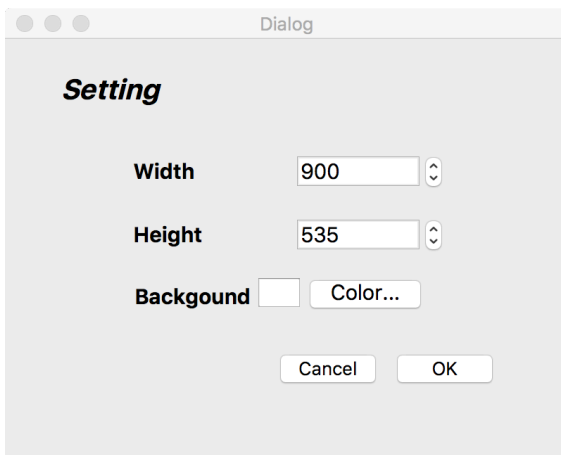
3.11 保存

点击保存按钮，或 Ctrl+S，弹出保存对话框，选择图片保存的路径，修改文件名，单击保存按钮即可。



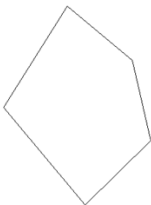
3.12 新建

点击 New 或者 Ctrl+N，弹出新建画板对话框，可以设定画板的大小以及背景色。

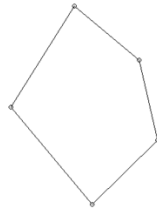


4 简单演示

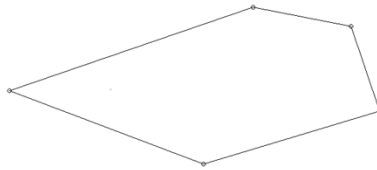
首先，绘制一个五边形。



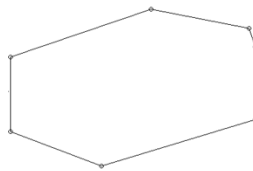
对多边形进行平移。



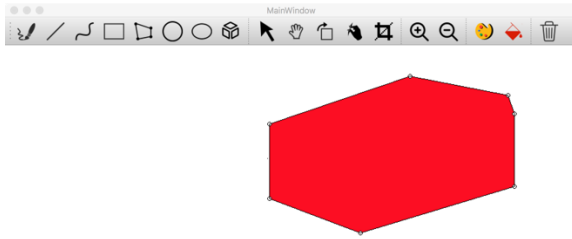
对多边形进行修改。



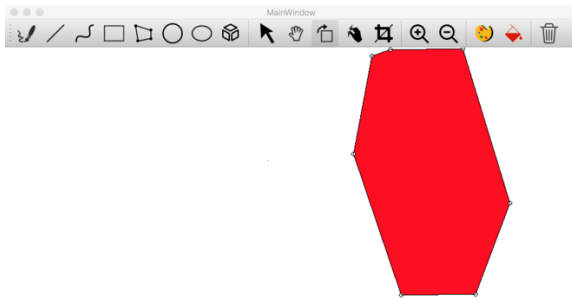
对多边形进行裁剪。



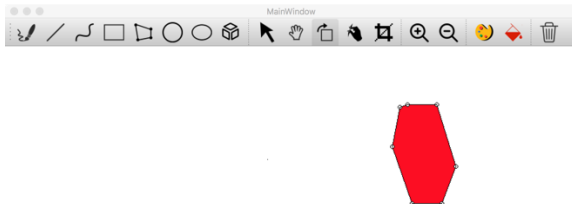
对多边形进行填充。



对填充后的多边形进行旋转。

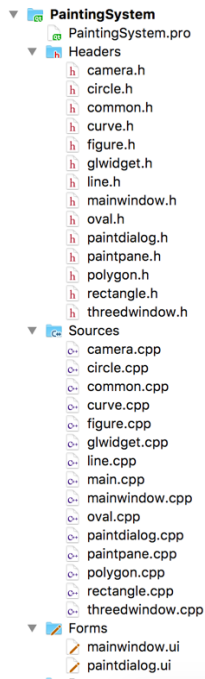


对填充后的多边形进行缩放。



5 代码说明

下面为系统的代码框架，现在作简要说明。



首先, PaintingSystem 系统是在 Clang 7.0 下进行开发的, 所有的交互操作均是通过 Qt 的机制实现的。

其次, PaintingSystem 系统中支持的所有图形均继承于同一个父类 Shape。在 figure.h 中定义了多个虚函数允许子类不同程度上的继承, 这些虚函数主要有:

```
virtual int distance(QPoint cur) {return cur.x();} // Compute the distance of mouse
and shape to judge whether be chosen
virtual QPoint getCenter() {QPoint p; return p;} // Get the center point
virtual void move(int dx, int dy) { int a=0; a += dx + dy;} // Shift
virtual void rotate(double angle) { double a=0; a += angle;} // Rotate
virtual void scale(QPoint start, QPoint end) {int a=0; a += start.x() + end.x();} //
Modify
virtual void zoom(ZoomMode mode) { int a; a = mode;} // Zoom in and Zoom out
virtual QList<Shape*> trim(QList<QPoint>) { QList<Shape*> res; return res; } // Trim
virtual QList<QPoint> getPoints() { QList<QPoint> l; return l;} // Return all points
of this shape
virtual void reset() {} // Reset some parameters
```

在子类中, 如果具体实现了这些虚函数, 则代表该类图形支持该类操作, 否则代表不支持。这些函数的调用均通过父类指针的动态绑定进行, 所以在外部调用时代码显得很简洁。

致谢

非常感谢孙老师和各位助教对实验的设计, 也非常感谢孙老师课上对知识点的详细讲解。通过这个实验, 从实践上对理论知识进行了巩固和加强, 其中也学会很多新的知识, 同时也增加了对 Qt 的理解和操作。