

浙江大学 实验报告

课程名称: Java应用技术 指导老师: 翁恺 学生姓名: 尹嘉权
实验名称: miniCAD 实验类型: 编程实现 学生学号: 3120000419

一. 实验目的和要求

目的:

使用Java GUI来实现基本的图形绘画, 了解相关的监听控件的使用和JFame、JButton等类库的使用。

要求:

In this project, you should use Java GUI to implement a simple drawing board.

To get full marks, you should implement to:

1. draw line, circle, rectangle, text
2. change the color of a target
3. change the position of a targets
4. change the size of a targets
5. save images as files and load a image file

All processes of changes should be visible.

You should submit a runnable jar file with all your source files

二. 实验内容和原理

内容:

使用Java GUI编程实现绘画板, 需要实现这些功能:

1. draw line, circle, rectangle, text
2. change the color of a target
3. change the position of a targets
4. change the size of a targets
5. save images as files and load a image file

要求所有的操作都是可见的, 最后提交jar文件和源码。

原理:

设计一个基础图形类shape, 里面包含类型、颜色、文字、图像等基本信息;

从JFame extends出miniCAD, implements MouseListener和

MouseListener¹，来捕捉鼠标/触控板上面的动作，当选定需要绘画的图形和颜色之后，捕捉按下鼠标时候的坐标，然后对松开鼠标的时候做监听，如果这时候是非拖拽模式，就计算出需要画出来的图；如果是拖拽模式，则把需要拖拽的内容计算新的坐标。这里面做一个约定，拖拽的target只能是上一次画的 || 拖拽的目标。然后对于移动的target，会有两个操作，第一个是把移动前的目标设置成白色，并加入paint队列；第二个计算新目标，设置为原来的颜色，只是坐标变了。最后repaint的时候就能看到修改的痕迹了。（因为要求所有的操作都是可见的，而又为了区分新的和旧的，只好把旧的需要操作的图形的颜色变成白色，当然，这个白色不会在最后的save image里面显示出来，因为画布本来就是白色的，但是在miniCAD上面是可以看出来的）

对于Menu的处理，在同一个组只能单选，比如说颜色单选，图形单选，因此就要把对应的JRadioButtonMenuItem放到同一个ButtonGroup来控制。颜色默认是黑色，图形默认是直线。

三. 主要仪器设备

MacBook Pro, MacOSX 10.10, 2.6GHz Intel Core i5, 8G RAM
JSE: JavaSE-1.8
JDK: oracle-java8-jdk

四. 操作方法和实验步骤

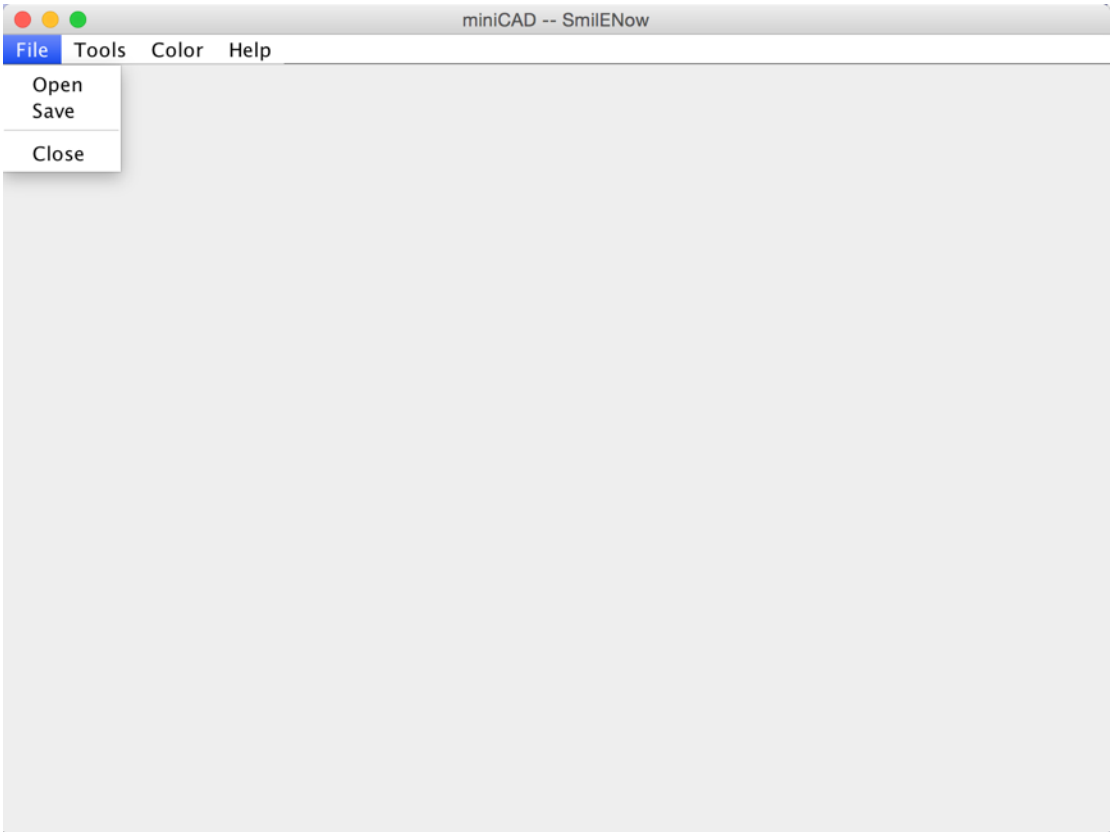
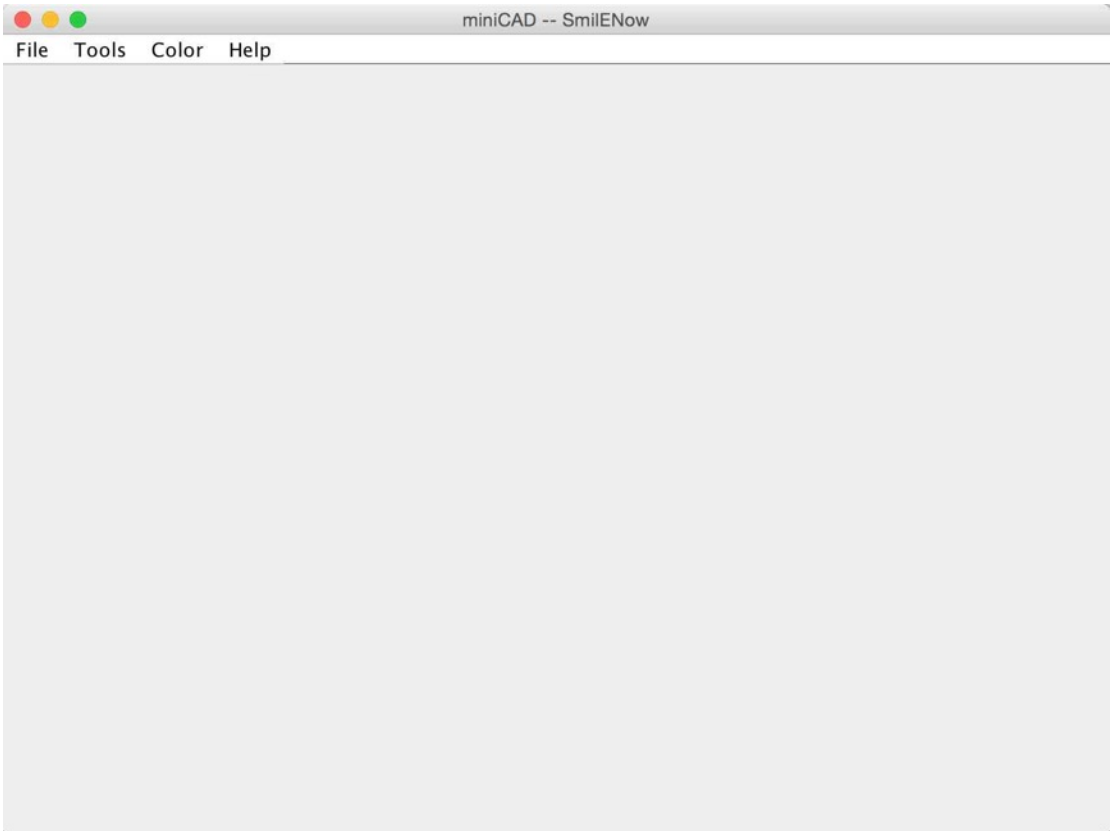
先分析题目和整个程序架构，具体分析可以看第二部分的原理，然后打开Eclipse建立相对应的工程文件，输入代码，进行调试运行之后导出jar文件即可。

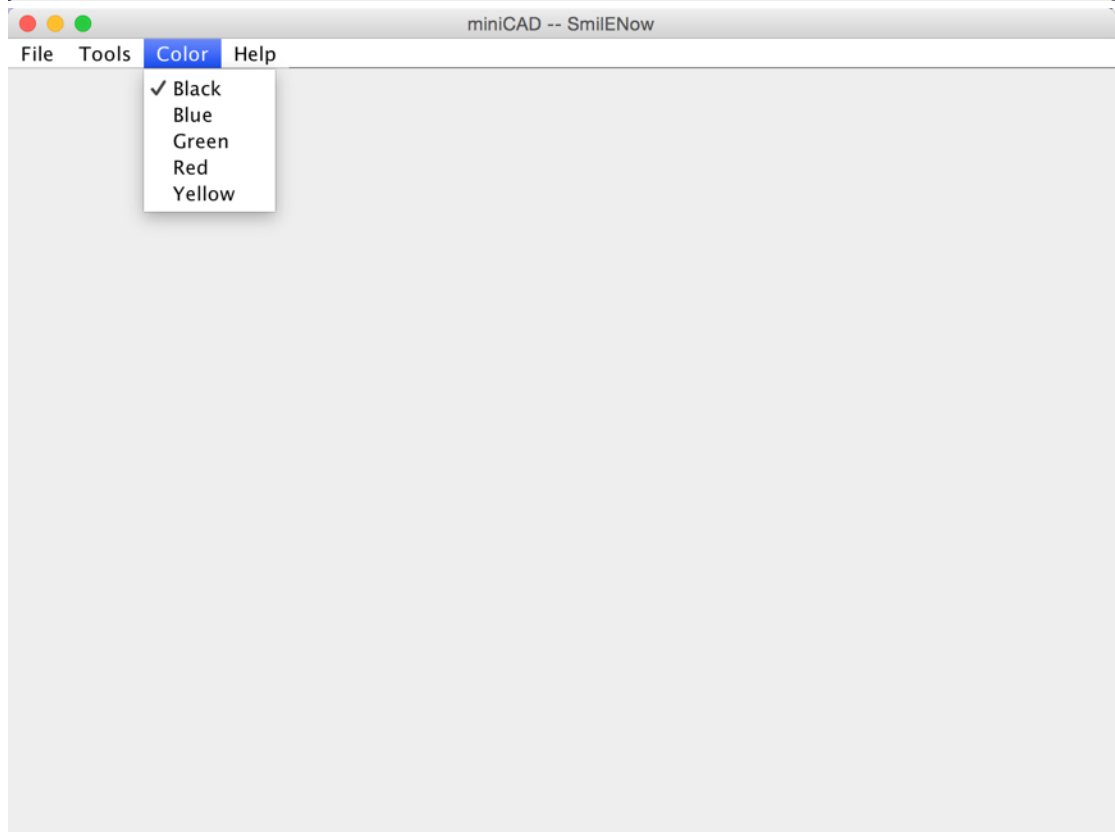
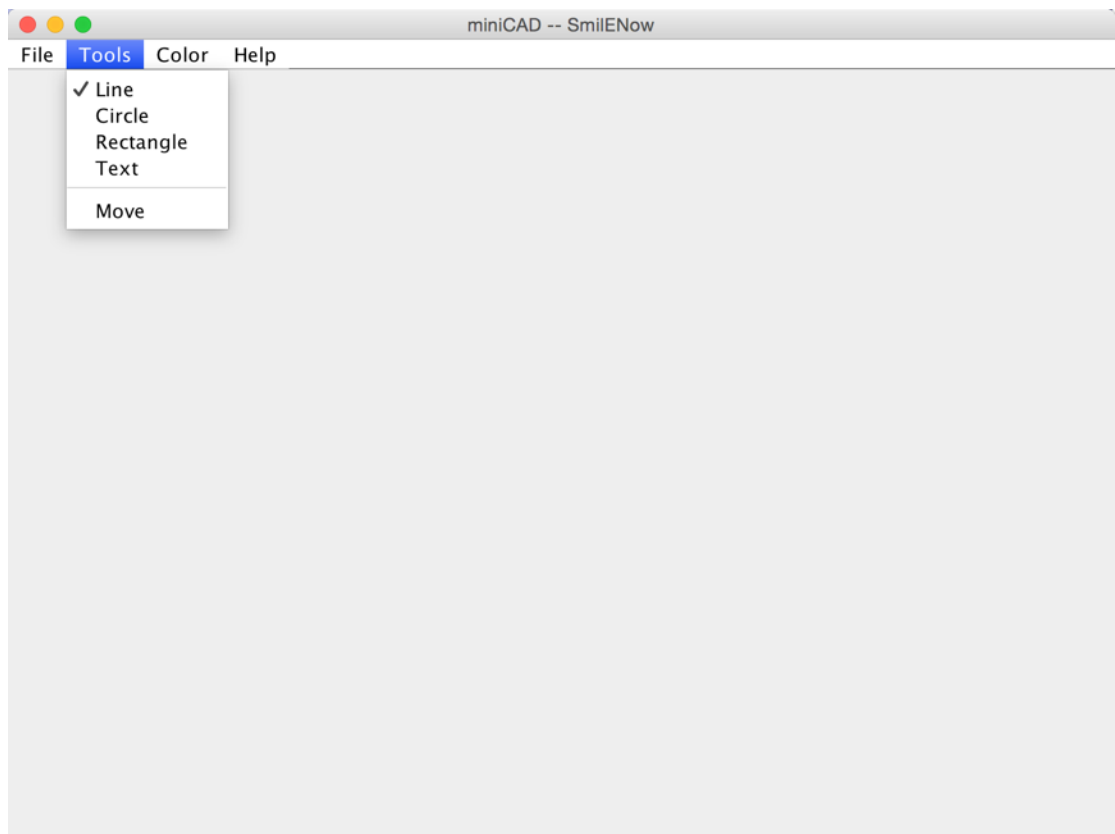
具体的代码可以参见/src文件夹下的各个.java文件。

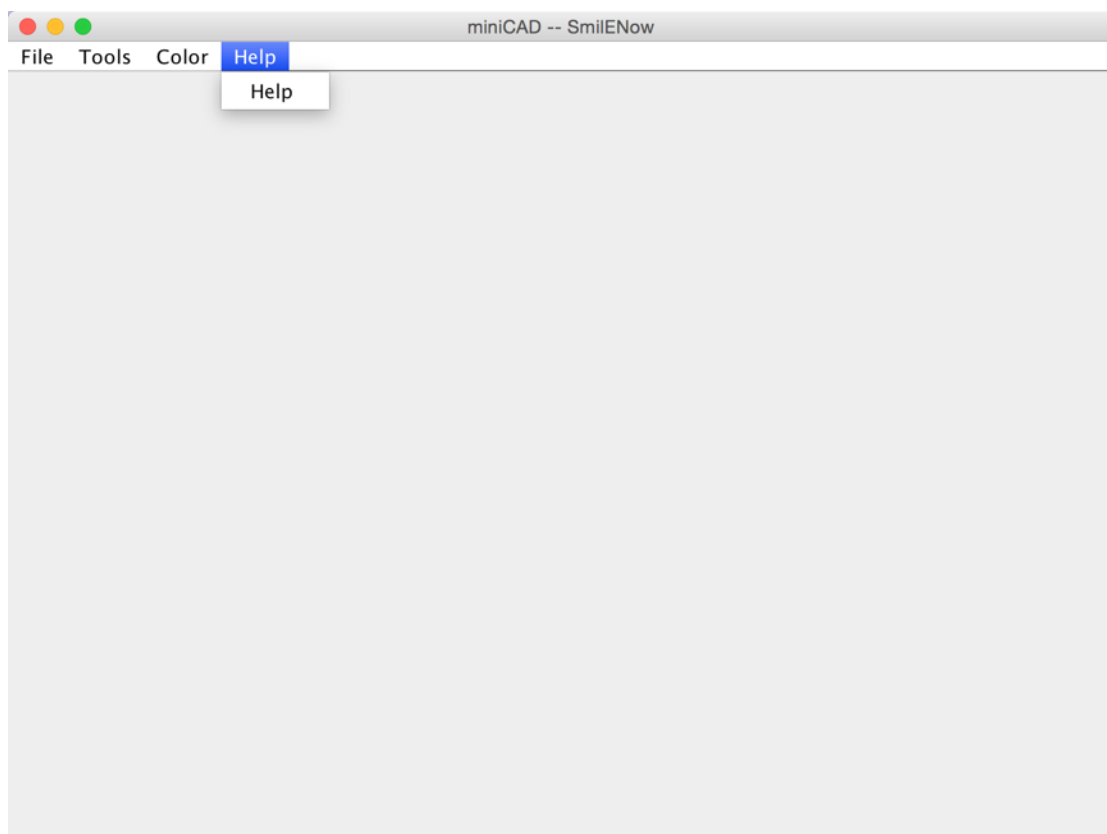
五. 实验数据记录和处理

基本的界面如下：

¹ Reference: <http://www.tuicool.com/articles/ErAF3m>



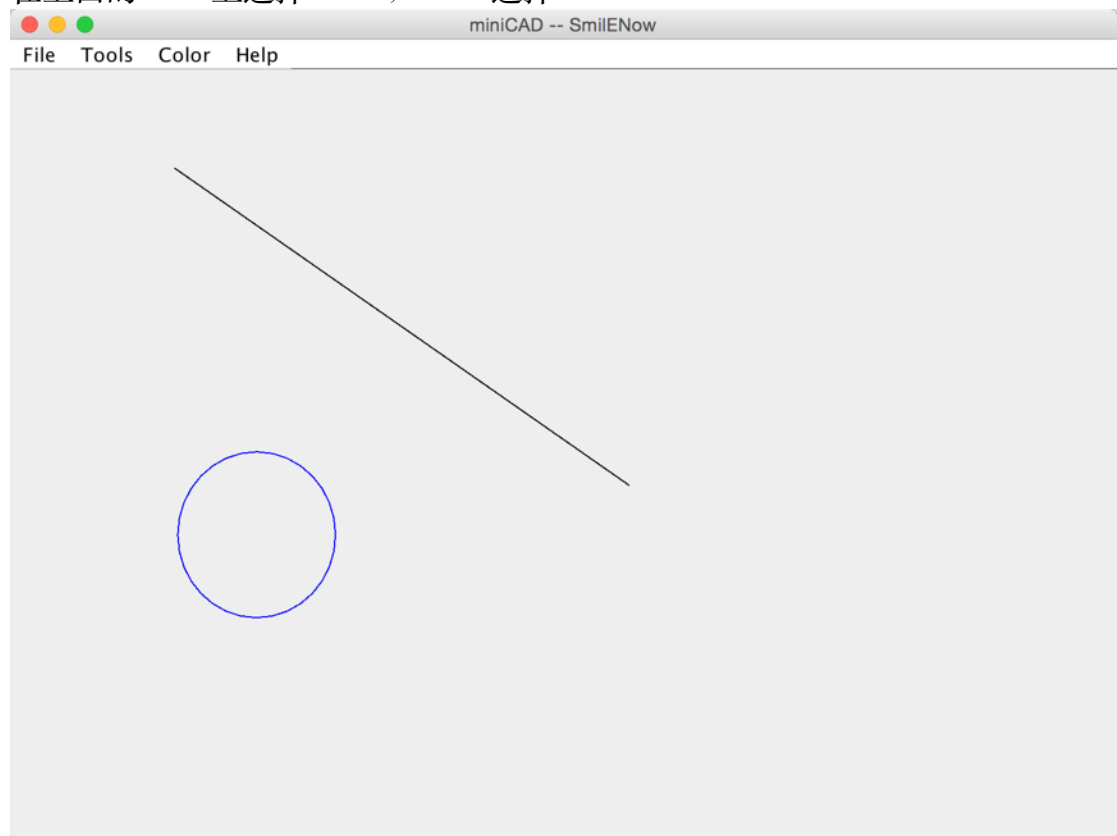




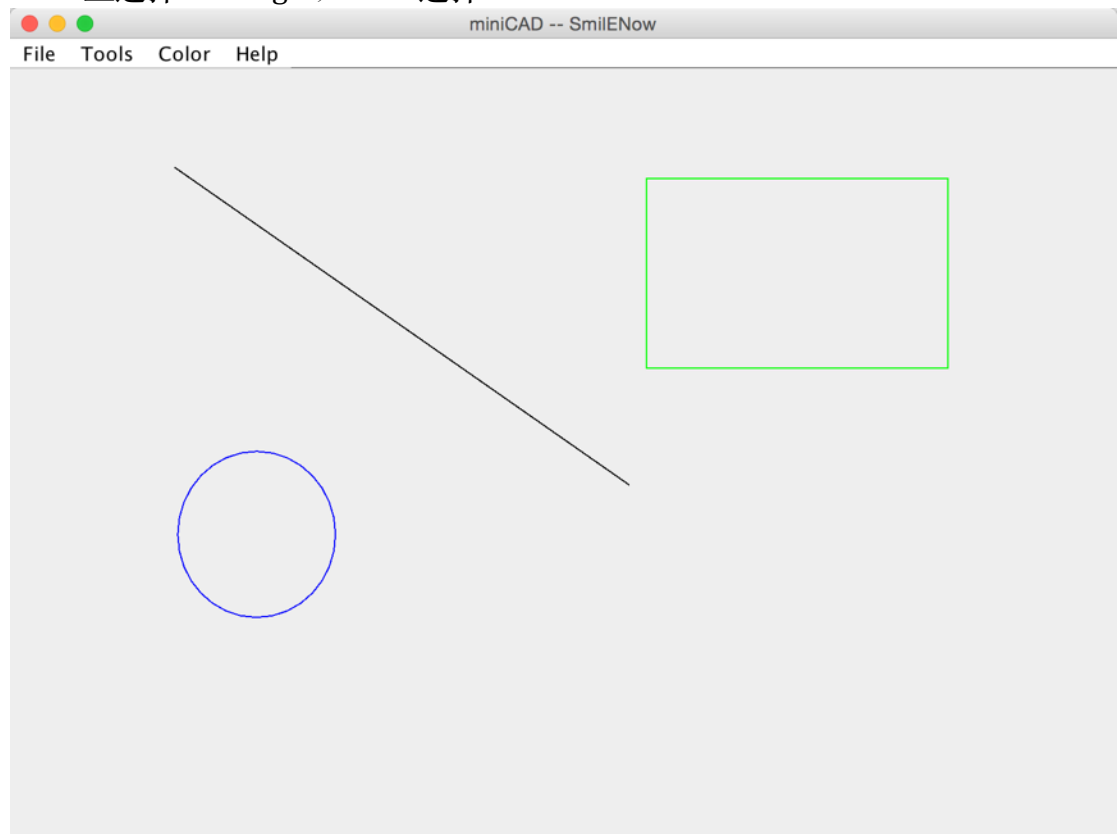
默认是黑色的直线，点击鼠标或者按下触控板，然后移动一段距离之后松手，查看画出来的直线：



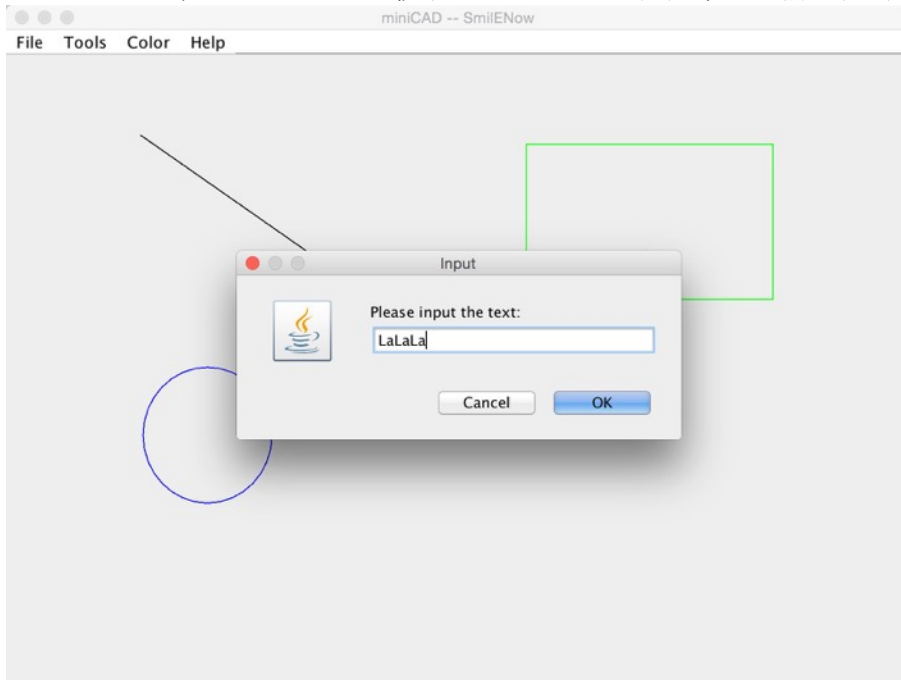
在上面的Tools上选择Circle，Color选择Blue:



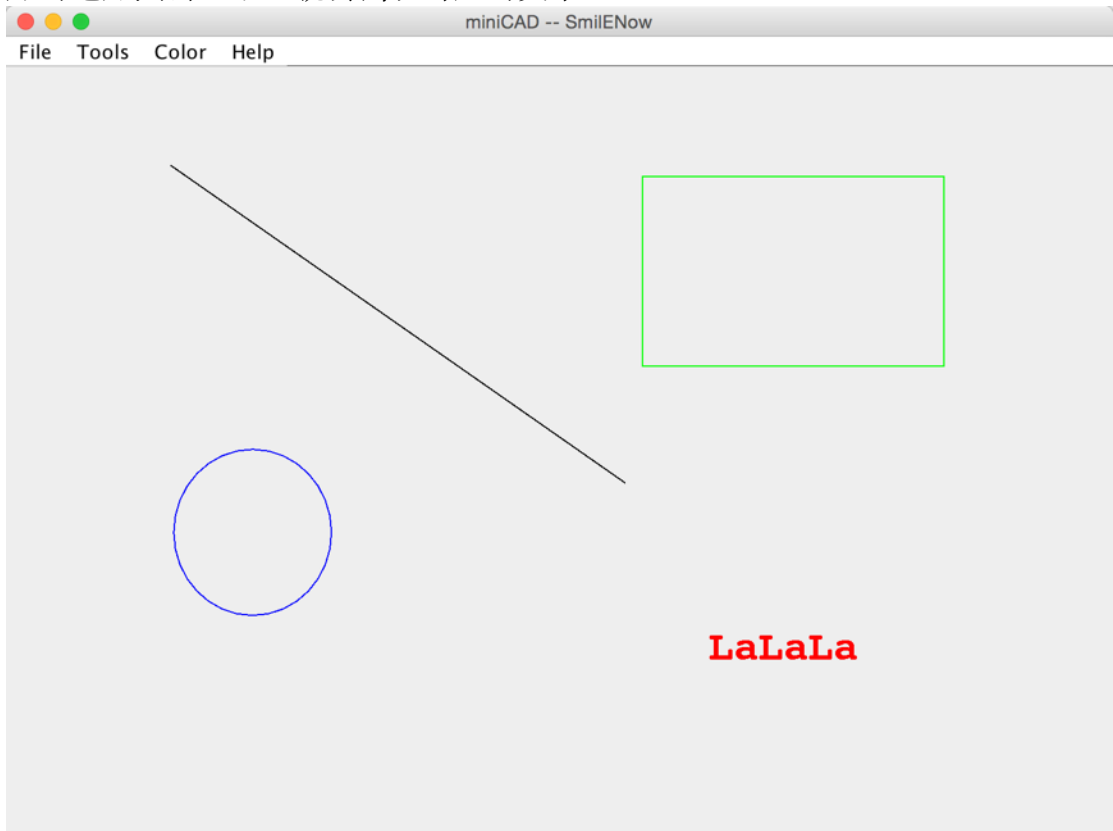
Tools上选择Rectangle，Color选择Green:



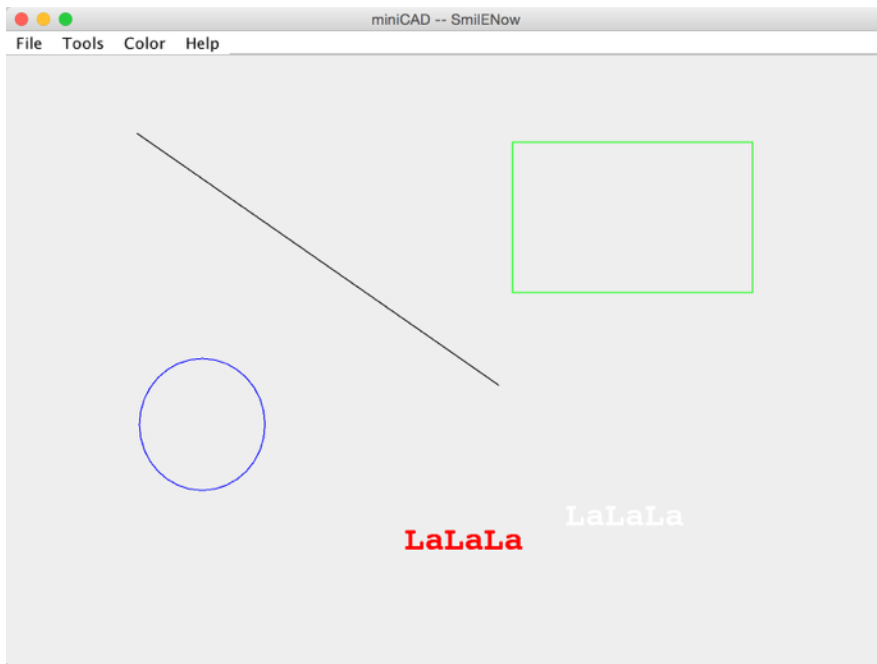
Color选择Red，然后勾选Tools下的Text：注意不能先勾选Tools再选择Color，因为在点击Text的时候设置了监听，会弹出窗口输入文字：



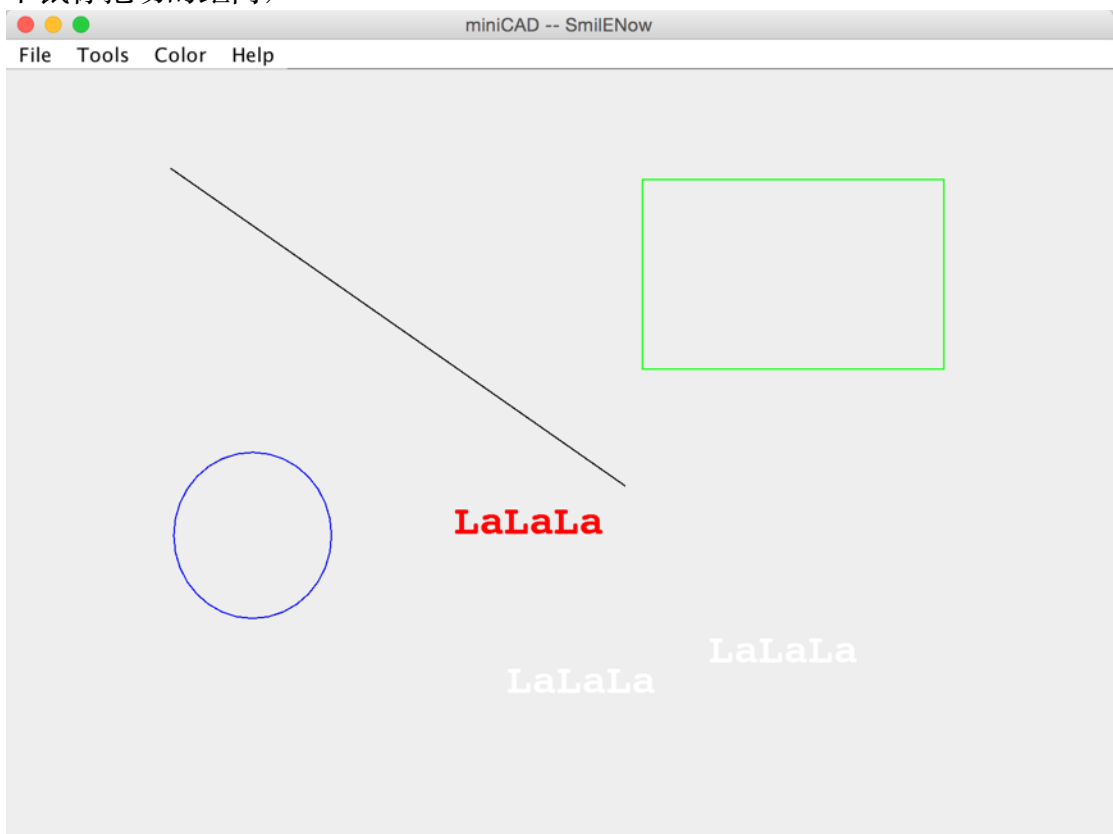
然后回车或者点击OK，这时候输入窗口会关闭，然后移动鼠标，在需要添加的地方点击一下，就会出现对应的文字：



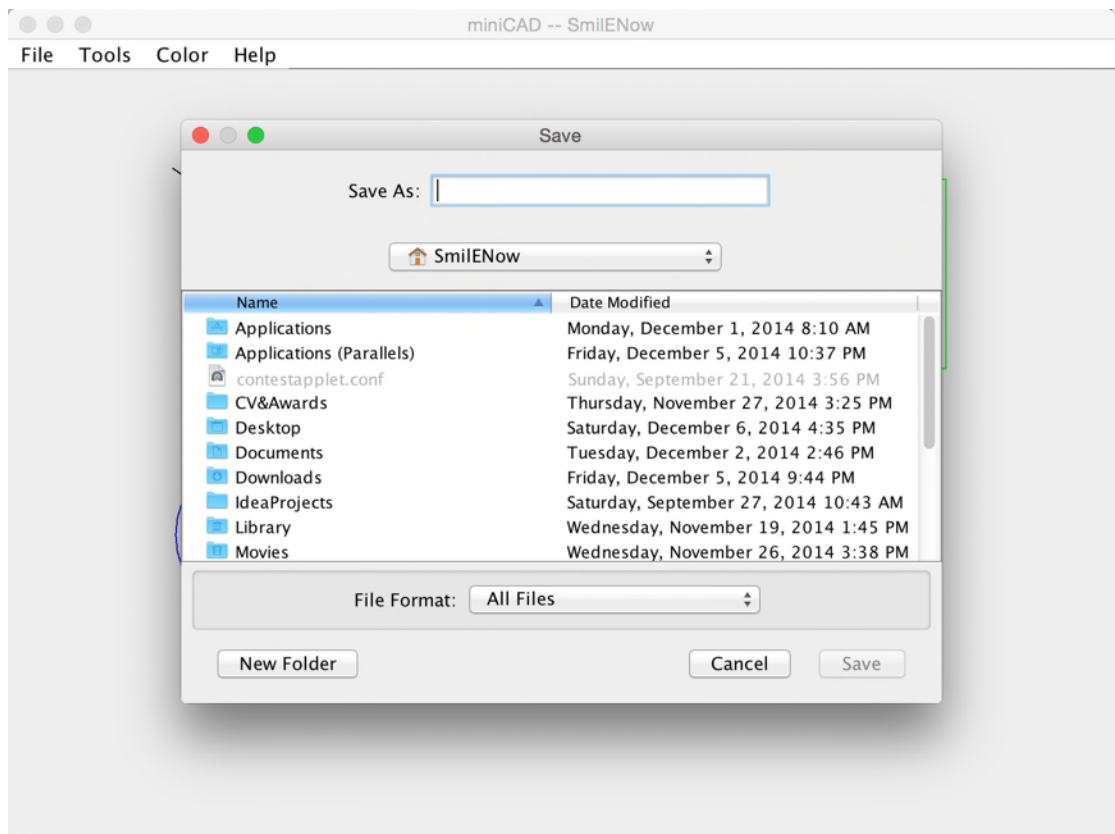
勾选Tools下的Move，由于上次操作是文字，所以这次的Move针对的是LaLaLa的文字来移动：



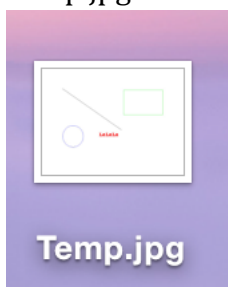
当然移动的时候，鼠标不一定要放在需要移动的目标隔壁或者附近，下面测试在别的空白的地方移动：（因为计算移动的时候是测算点击鼠标和按下鼠标拖动的距离）



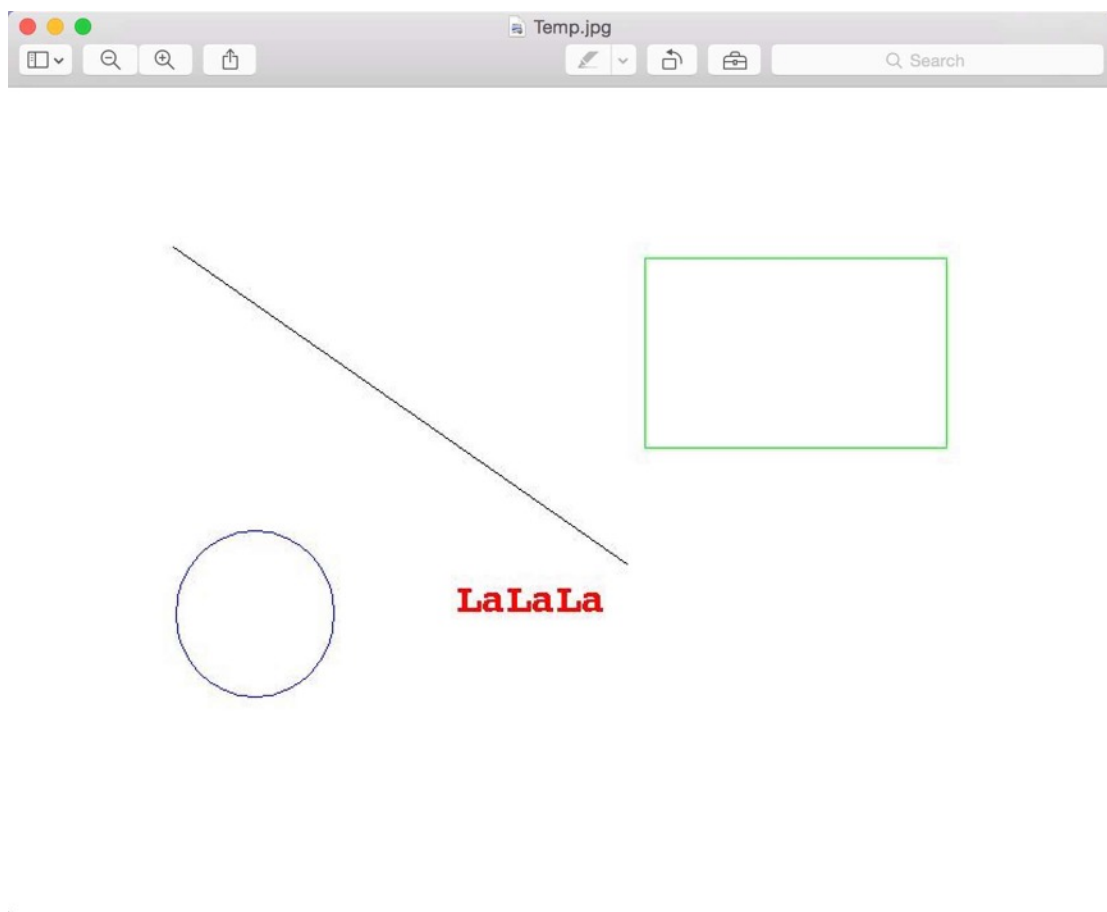
下面选择把这个图片保存下来，点击File->Save:



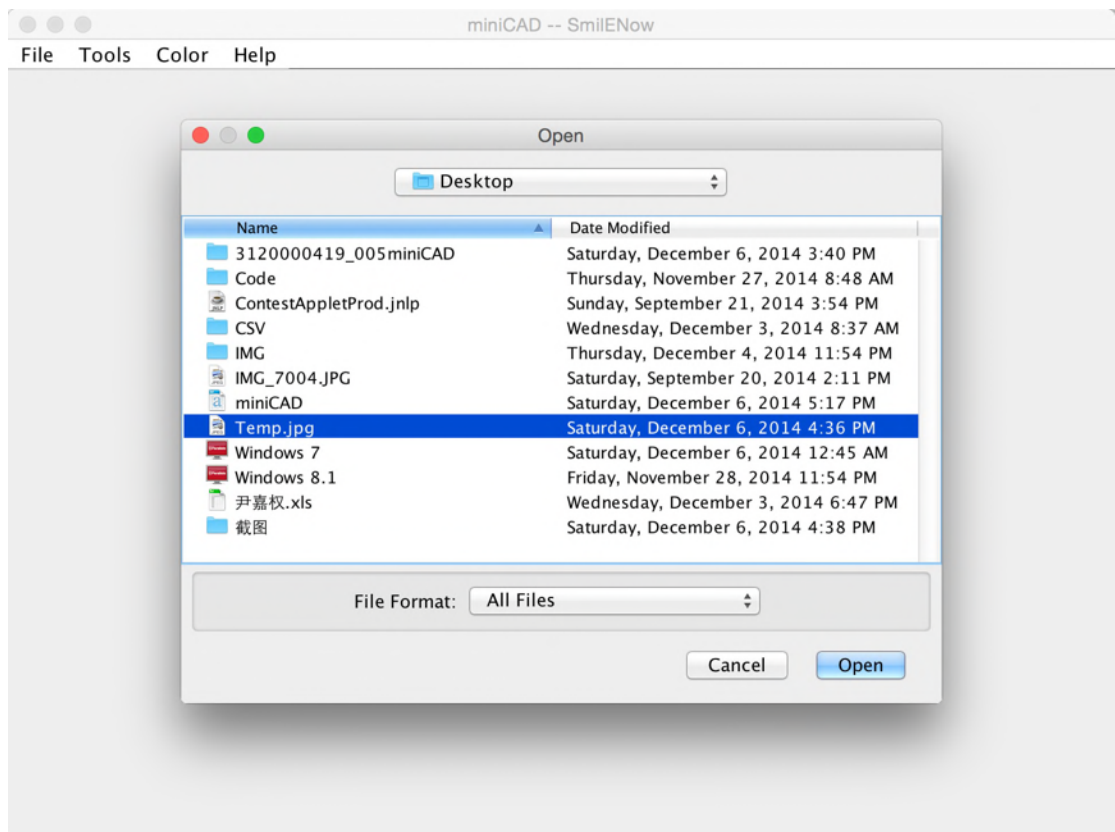
选择路径和输入文件名保存即可。这里我选择Desktop下，保存文件名为Temp.jpg:



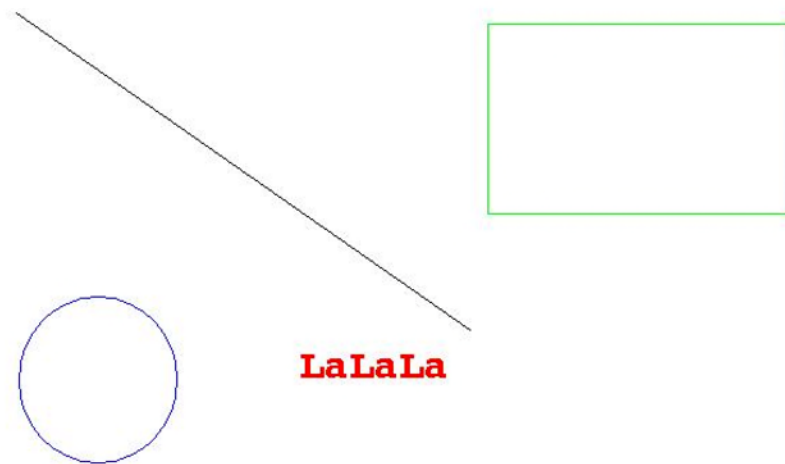
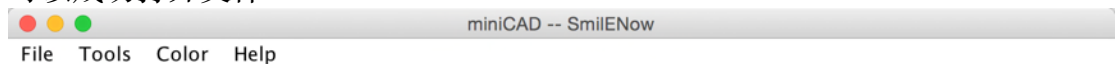
打开文件，发现之前移动的白色痕迹没了，只能看到最后的结果图：



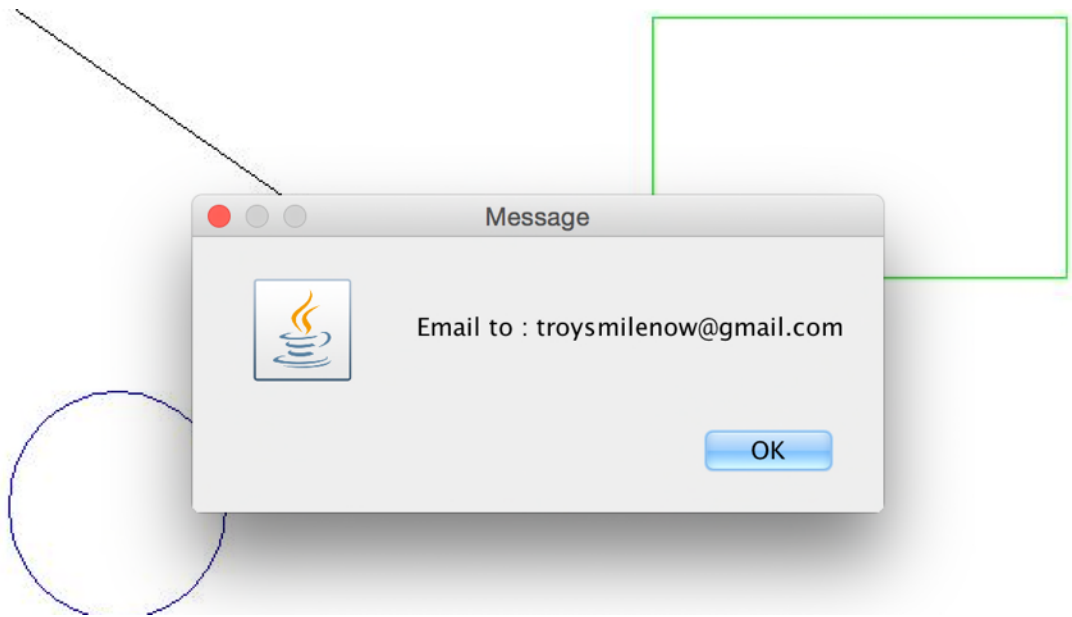
然后我们关闭miniCAD，然后重新打开，之后点击File->Open
路径选择之前我们保存的Temp.jpg



可以成功打开文件！



最后测试一下帮助，会提示Email:



六. 实验结果与分析

实验输出正确。

七. 讨论心得

在这次实验中，让我了解了Java GUI的编程方法，学会了基本的Image读入输出，同时学会了用ButtonGroup来对按钮进行单选处理，为了使程序更加的O O P，对于每个控件被点击都设置了相对应的内部类来actionPerformed，同时，为了能够修改显示内容，学会了通过定义paint来调用repaint函数来重画图像，整体下来还是学会了很多知识的。