

Java 程序设计实验报告

Project2

朱理真 3190101094

计算机科学与技术

2021-11-13

目的和要求

实现一个简单的绘图工具，以CAD的方式操作。

1. 支持的绘制对象有矩形，圆形，文字和直线
2. 而且可以修改它们的一些参数比如颜色。
3. 可以拖动图形和调整大小
4. 可以保存和恢复

设计结构

1. 主窗口CADFrame

设计了一个窗体结构类CADFrame，继承了JFrame类。

采用BordLayout布局，由菜单栏，工作区和工具栏组成。

其中，菜单栏由subMenuBar实现，工作区由subToolBar实现，工作区则由subDrawPanel实现。

如何实现菜单栏、工具栏与工作区之间消息的传递？

subToolBar提供方法getXXAction(包括open,save,clear,delete,copy,paste,cut,choose color等对应的Action),返回匿名Action类，而subMenuBar和subToolBar通过add方法加入这些Action事件侦听器，由此实现消息的传递。

2. 工作区subDrawPanel

subDrawPanel实现对背景板的显示，消息的传递以及作为图形的容器，参与对图形的管理，并响应上层的业务需求（工具栏，菜单栏）。

背景板方面，通过修改颜色参数显示不同的背景；

图形的管理方面，支持添加图形和删除图形的功能，这些功能是在JFrame提供的add和removeComponent方法的基础上实现的。图形只在subDrawPanel范围内显示，且其上的任何改动都会反映到subDrawPanel上来。

图形的布局方式不采用任何LayOut布局（setLayout(null)）。图形自行通过setBounds方法，改变其在subDrawPanel中的大小和位置。

向对象传递消息

对每一个容器内的图形对象，subDrawPanel都给它们增加了一个鼠标侦听类，用于选中图形，每当某个图形被鼠标点击，subDrawPanel就会记录下当前的选中的对象，同时使其进入focus，用于后续操作在其上的应用。

如何响应各种业务

◦ 创建图形

接口方法是getCreateAction.返回一个匿名类Action对象。在此对象的ActionPerformed方法中，subDrawPanel会add一个鼠标侦听类(包括拖动)ElementCreate,它接收鼠标响应，对创建图形的过程做出相应的显示。在该类中，需要调用subDrawPanel的add方法使新创建的图形对象加入到容器中，但是为避免过多的耦合，ElementCreate对象不会直接调用该方法，而是提供一个接收Action的方法，由此实现ElementCreate对subDrawPanel的反馈。图形创建完毕后，ElementCreate会自行从鼠标侦听列表中退出。待下次再有图形创建时再加入。

◦ 对象的保存

类似于创建图形，图像的保存需求也在一个匿名类Action对象的ActionPerformed方法的定义中处理。下面业务也是如此，不再赘述。首先调用Java库中的JFileChooser类创建一个文件选择窗口，得到文件名后，便调用saveDrawElements进行处理。saveDrawElements的流程主要是先对subDrawPanel的components加锁，然后获取components的list,首先写入components的个数，接着通过ObjectOutputStream类的writeObject方法一个一个将component序列化并写入文件中。

◦ 对象的恢复

保存与恢复类似，都是选择到一个文件后进行文件处理。在将当前容器内所有组件进行清除后，打开文件，首先读出components的个数,接着将component依次读出，但我们注意到每个component对象内部的事件监听对象是不能被序列化的，因此要对事件侦听对象进行恢复，因此调用了DrawElement类的bilnit方法。所有组件都加入后，再重新绘制界面。

◦ 对象的颜色更改

该功能只限制在指定的对象上而非全部对象，因此如前所述，选择正处于focus状态的对象，并打开JColorChooser窗口选择颜色，将颜色赋予该对象后重新绘制界面。

◦ 对象的移除

与更改颜色一样，选择focused对象，将其remove后重画即可。

◦ 对象的复制、剪切

subDrawPanel中需要增设一个成员对象，以记录当前在剪贴板的对象。一旦一个图形对象被选中复制/剪切，就会调用其clone方法，得到一个新的对象放在剪贴板上。

◦ 对象的粘贴

粘贴时，不会直接将剪贴板上的对象add到容器中，而是add它的clone()返回的对象。

3. 图形对象DrawElement

DrawElement的是一个抽象类，其有矩形、圆、直线、文字四种子类。所有这些子类都有一些共同的属性和操作。可以设置长宽、坐标、颜色以及画笔大小。同时每个DrawElement都有4个Corner组件，分布在图形所在矩形的四角，用于拖动图形。Corner类继承JPanel,是DrawElement的内部类，表现为一个固定大小的白色小矩形，可以通过它间接地访问DrawElement。

实现一些共有的操作

- 图形创建

如上所述,该操作涉及到鼠标侦听类ElementCreate,但是由于此时图形对象还未创建,因此该鼠标侦听对象是应用于工作区而非对象本身的。在工作区上按下鼠标,便会记住该按下的点。如果后续进行了拖动操作,就会根据拖动出的大小创建一个新的对象(通过DrawElement的静态方法createElement创建),并通知Action事件侦听函数,将其加入到工作区容器中并重绘。之后所有连续的拖动都会不断调整图形的大小。在这个状态下,如果发生鼠标松开事件,ElementCreate对象就会自己把自己从工作区的鼠标事件侦听对象列表中删除。

- 图形拖动

图形拖动的实现则较为简单,只需让DrawElement加入侦听对象ElementMove,后续的拖动操作都会使其改变图形在工作区中的相对位置并通过setBounds方法反馈到工作区。

- 改变形状

改变形状的鼠标侦听类ElementResize并不直接加入到图形对象中,而是加入到4个Corner对象中。通过在Corner上面进行拖动,改变其在图形对象中的相对位置以及图形对象本身的长宽,从而实现图形对象大小形状的调整。

- 图形选择

如前所述,每个图形对象被加入容器时,都会被加入一个鼠标侦听类,当鼠标按在这个图形上时,就会触发选中操作,图形被设置为focus状态,可以响应键盘事件。基于图形类的Focus事件侦听类的实现,进入focus时,图形对象的Corner会被设为可见,同时图形外围增加了虚线边框。由于各种原因失去focus时,图形对象的Corner被视为不可见,同时虚线边框消失。

子类特有的属性和操作

- 绘制方式

每个子类都有不同绘制方式。矩形默认是一个填充的长方形,圆则是一个填充的椭圆。文字则是显示一段文字,默认是“Write Something...”,直线则是显示为矩形的一个对角线。

- 设置文字和字体

对于文字对象,选中后键入“Enter”,就可以设置文字和字体。该功能通过一个键盘事件侦听类实现。

- 创建时的朝向

对于直线对象,它既可以显示为主对角线,也可以是副对象线,但是其中的一个端点,一定要是创建对象时按下的那个点,基于此,直线对象在创建的过程中会有一个可改变朝向的特性。但在创建完毕后,朝向就不可改变了。

使用说明

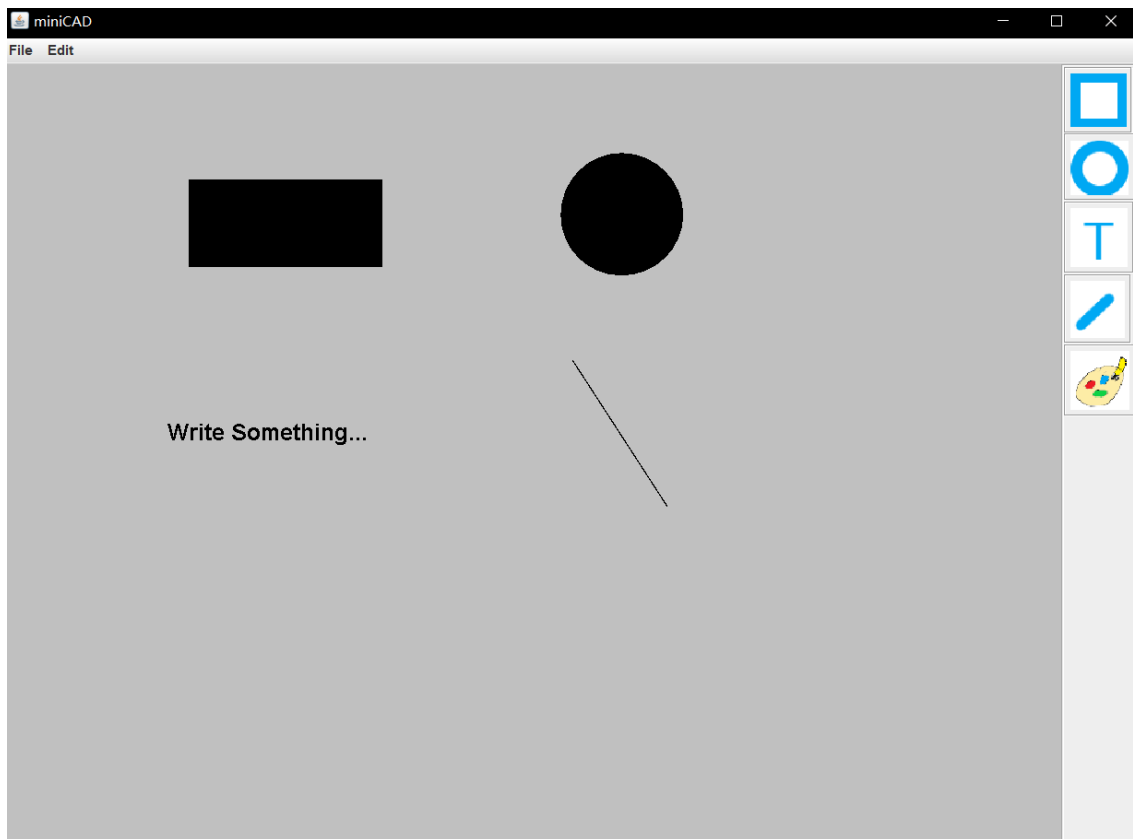
- 界面框架



可以看到，界面分为菜单栏，工具栏，和灰色的工作区，其中菜单栏有两个子菜单 `File` 和 `Edit`，工具栏有5个图标按钮，其作用分别是创建矩形、圆形、文字、直线和更改指定对象的颜色。

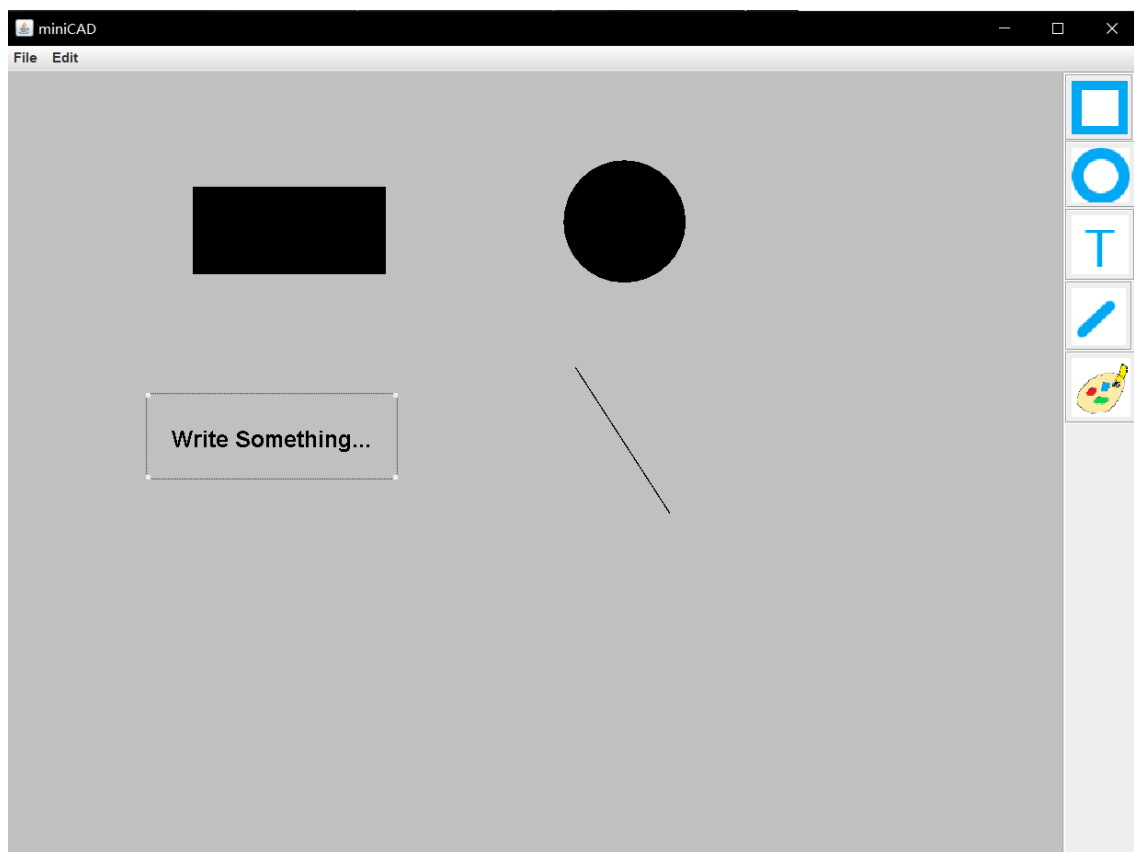
- 创建图形

选中对应的按钮，拖动即可，如下图所示。

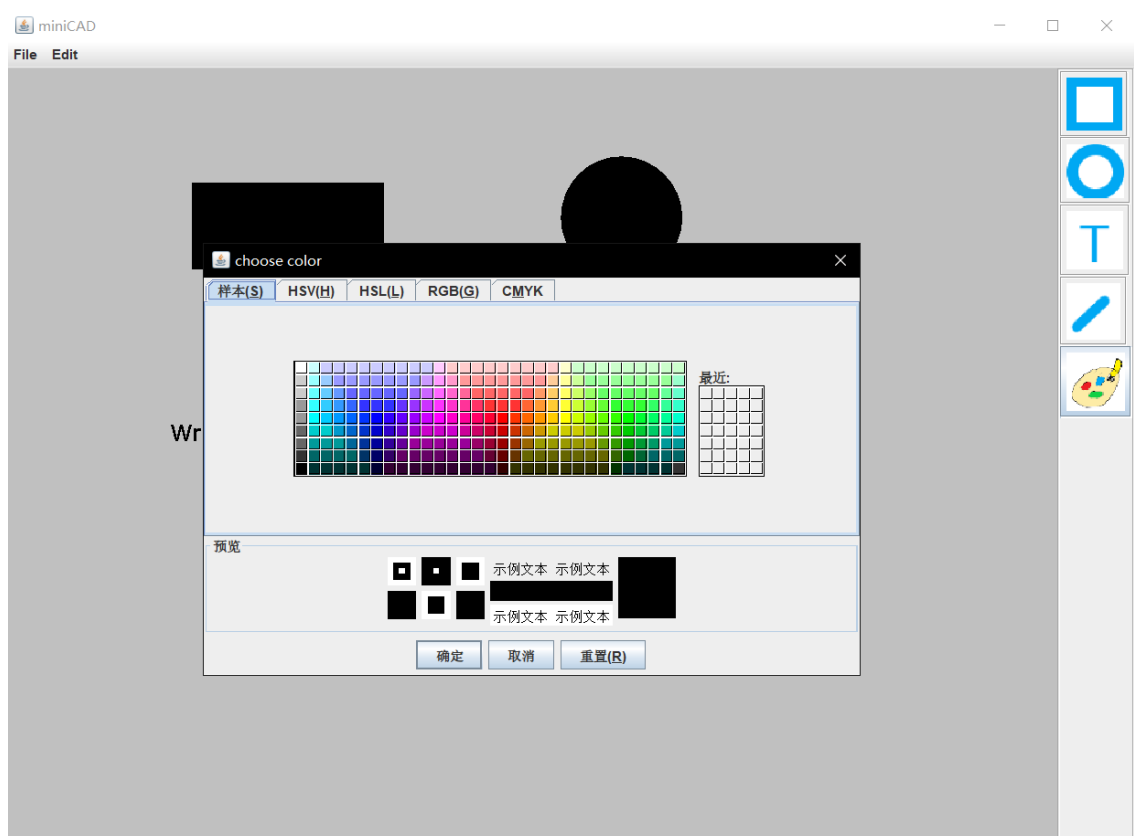


- 改变颜色

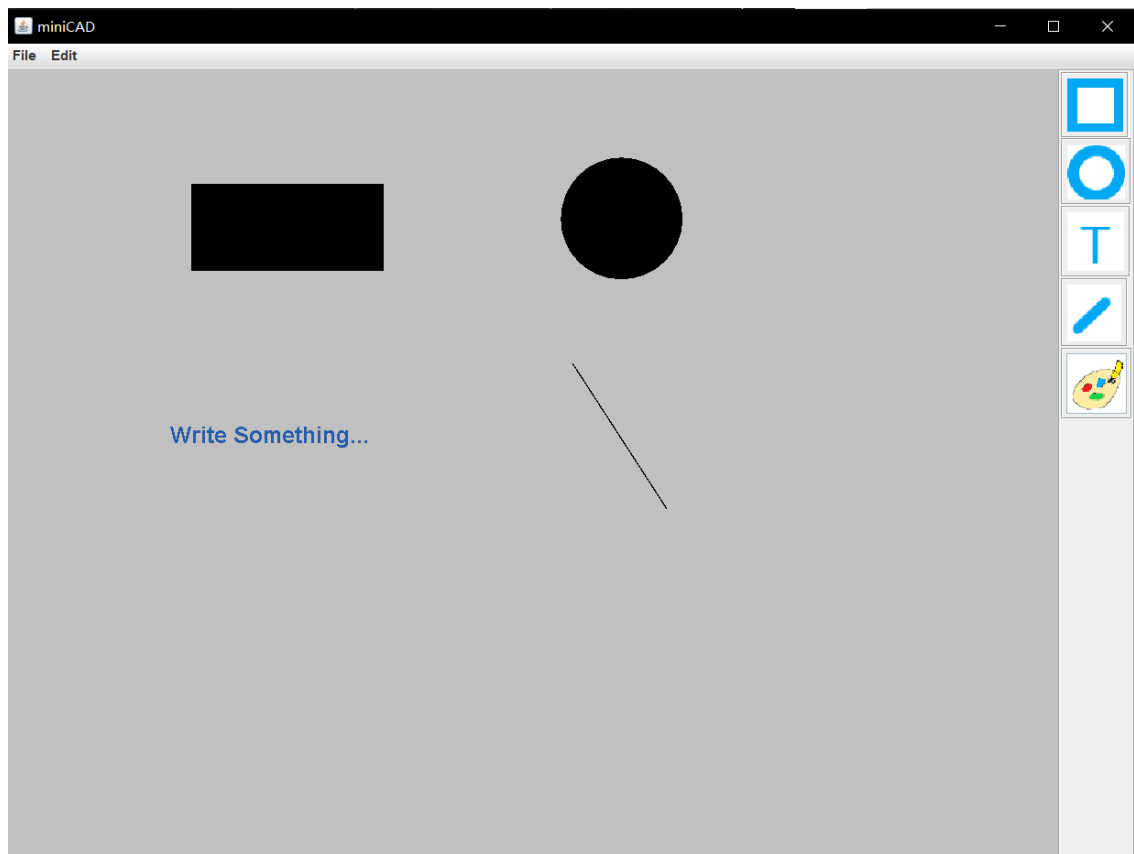
选中想要改变颜色的对象，然后点击工具栏的调色板按钮，即可设置。



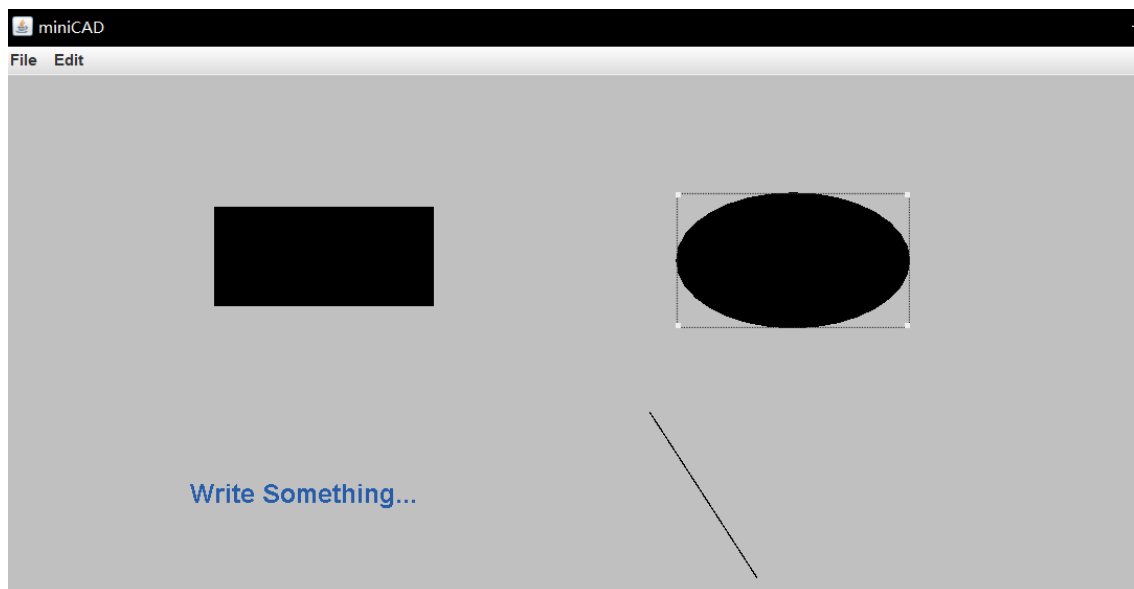
选中文字图形



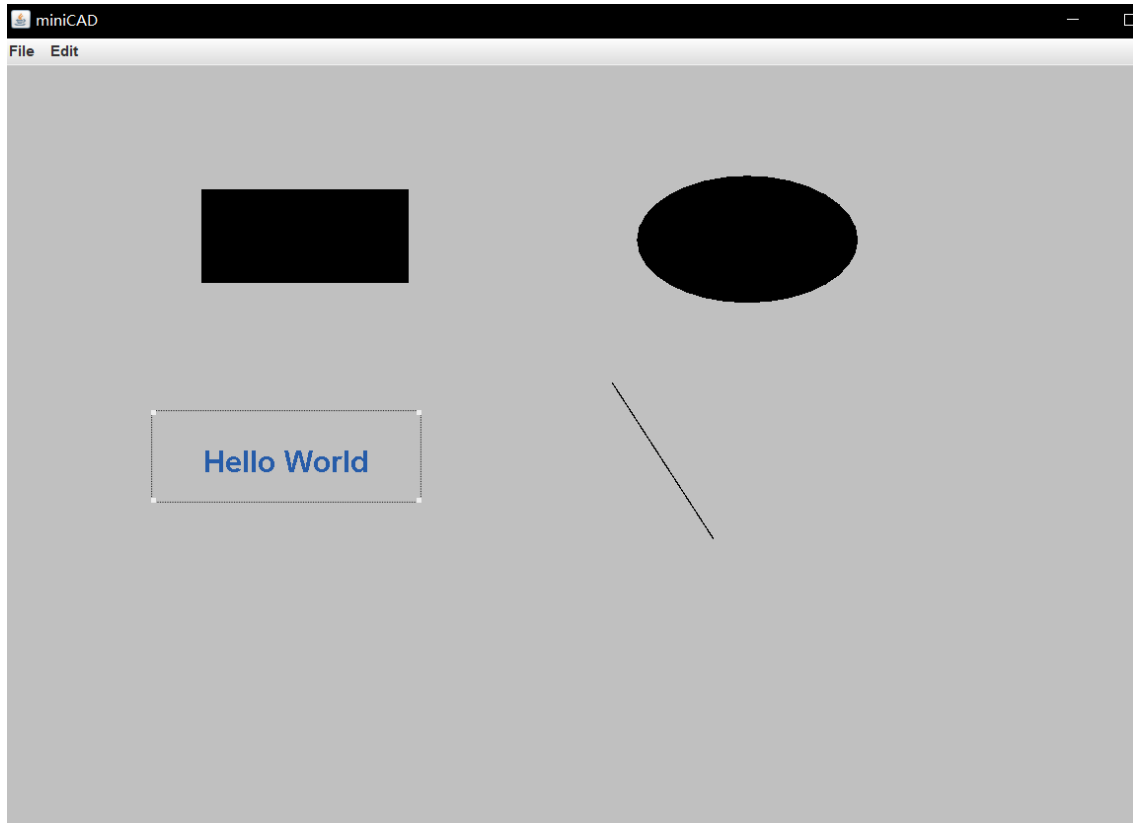
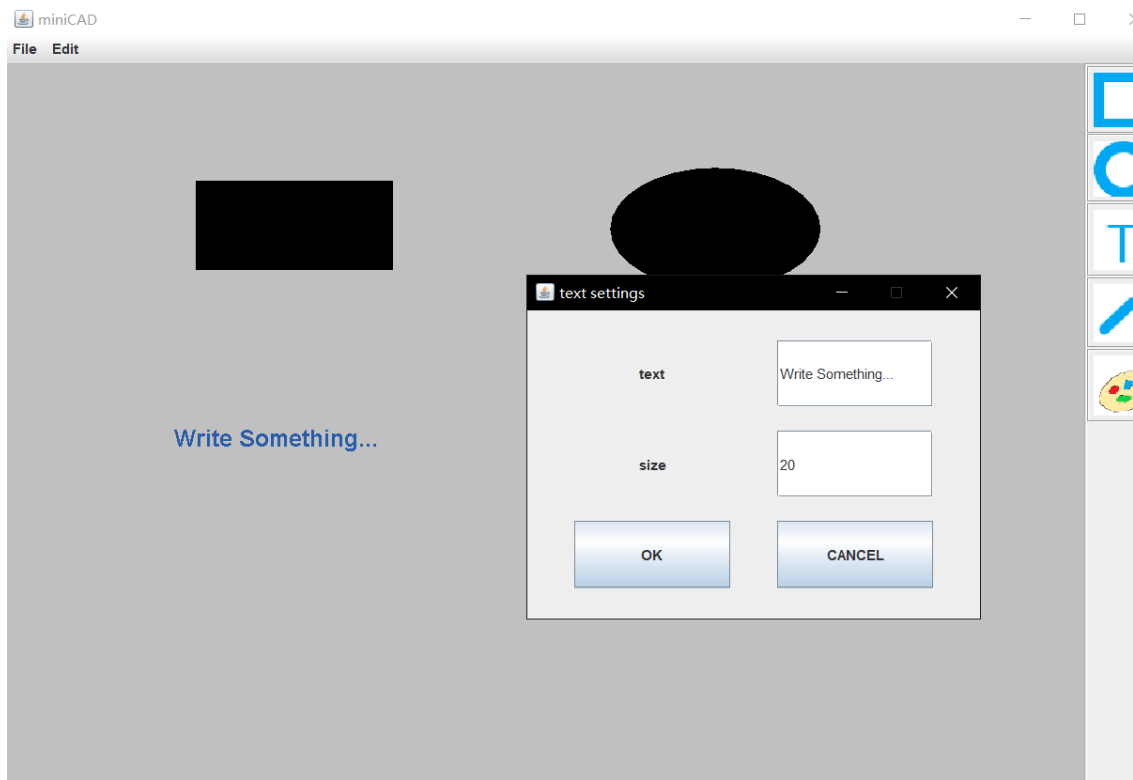
选择颜色



- 图形的拖动
鼠标拖动即可
- 图形的形变
拖动图形边缘的白色小矩形即可

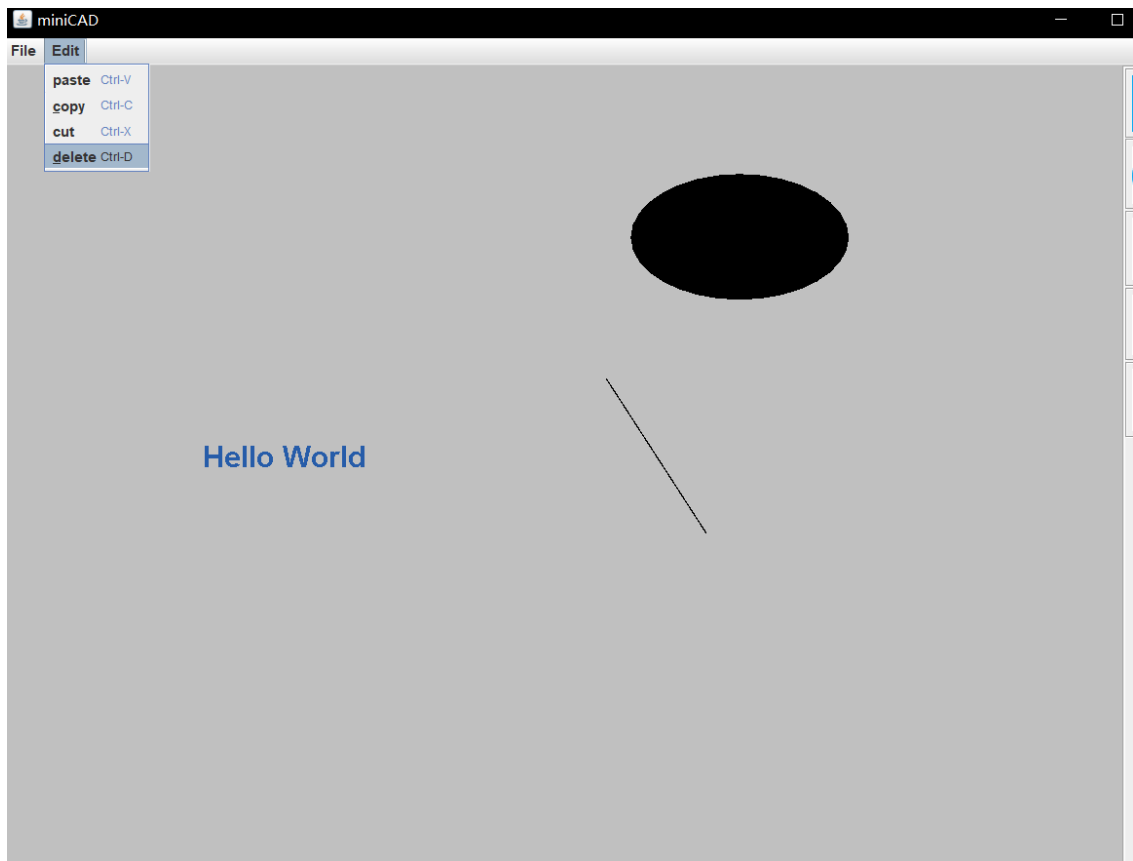


- 更改文字与字体大小
选中文字图像，键入 `Enter` 即可

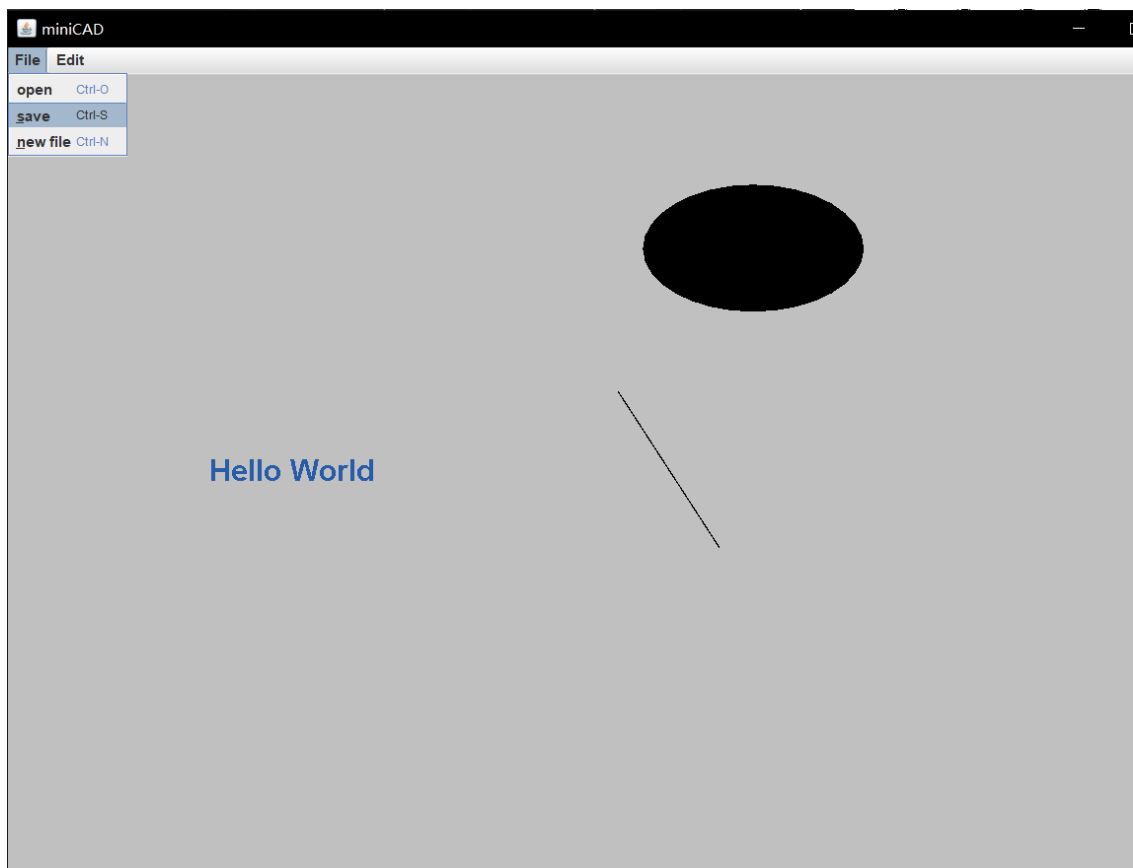


- 图形的删除

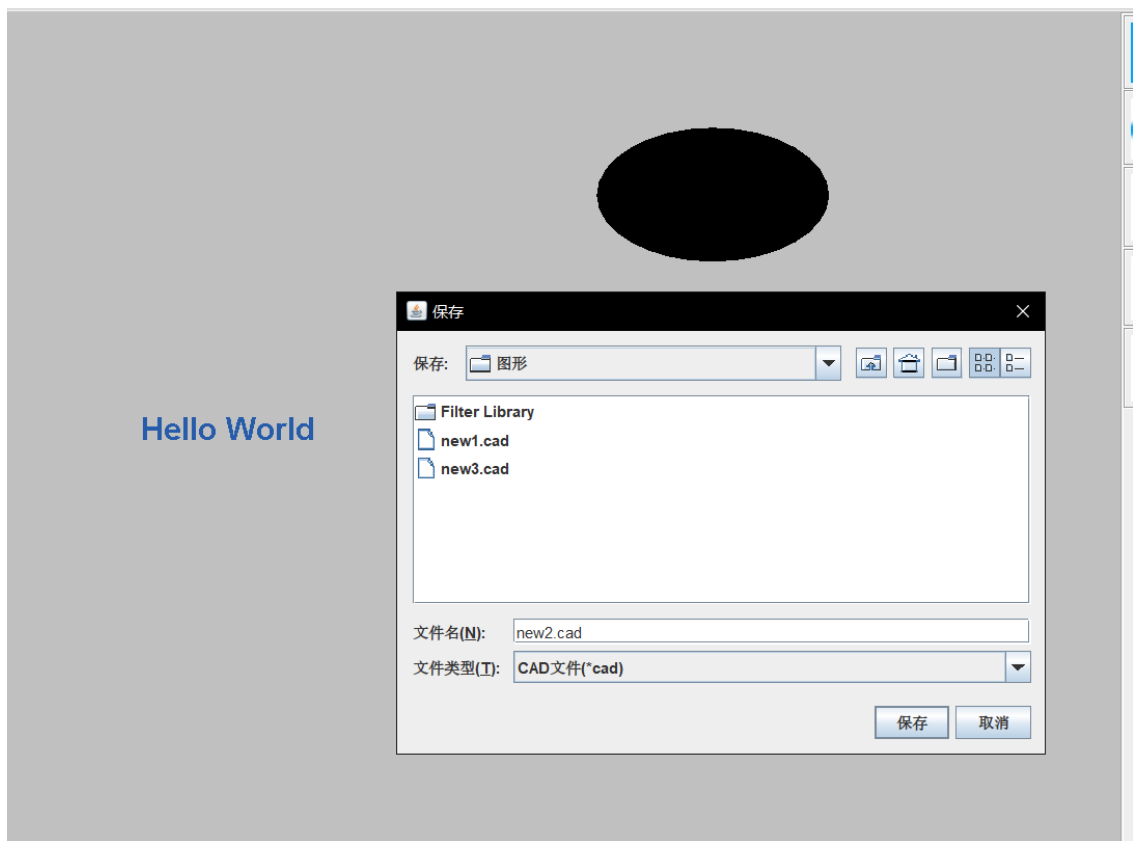
选中图形后，在菜单栏中选择 `Edit->Delete` (快捷键 `Ctrl-D`)



- 图形的复制、剪切、粘贴
选择图形，通过菜单栏的按钮或其快捷键进行处理
- 图形文件的保存

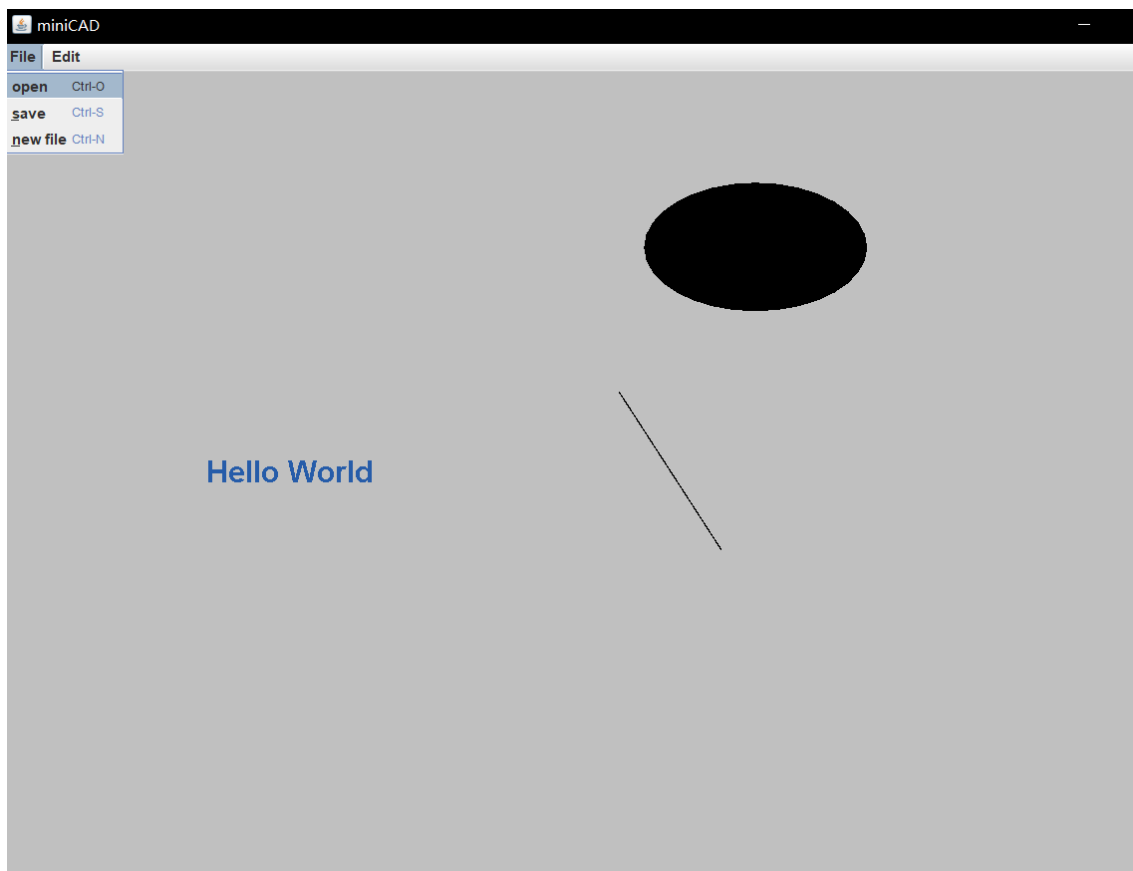


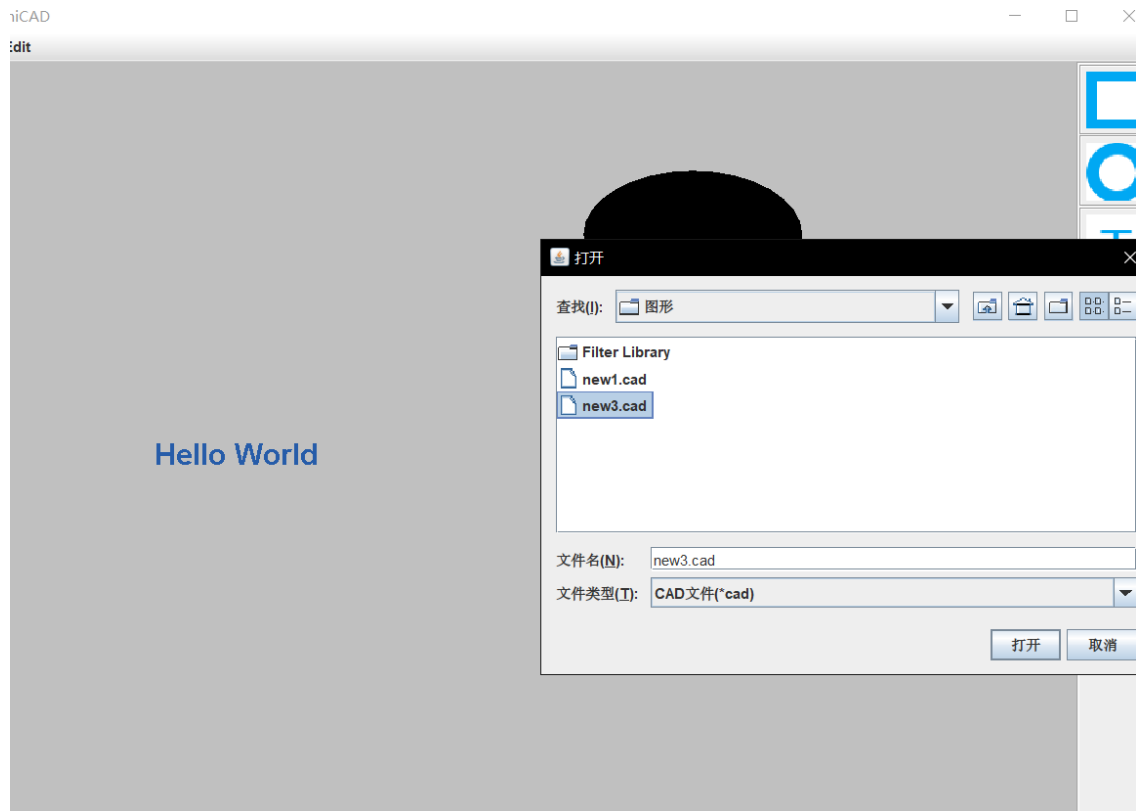
点击之后，弹出文件选择界面



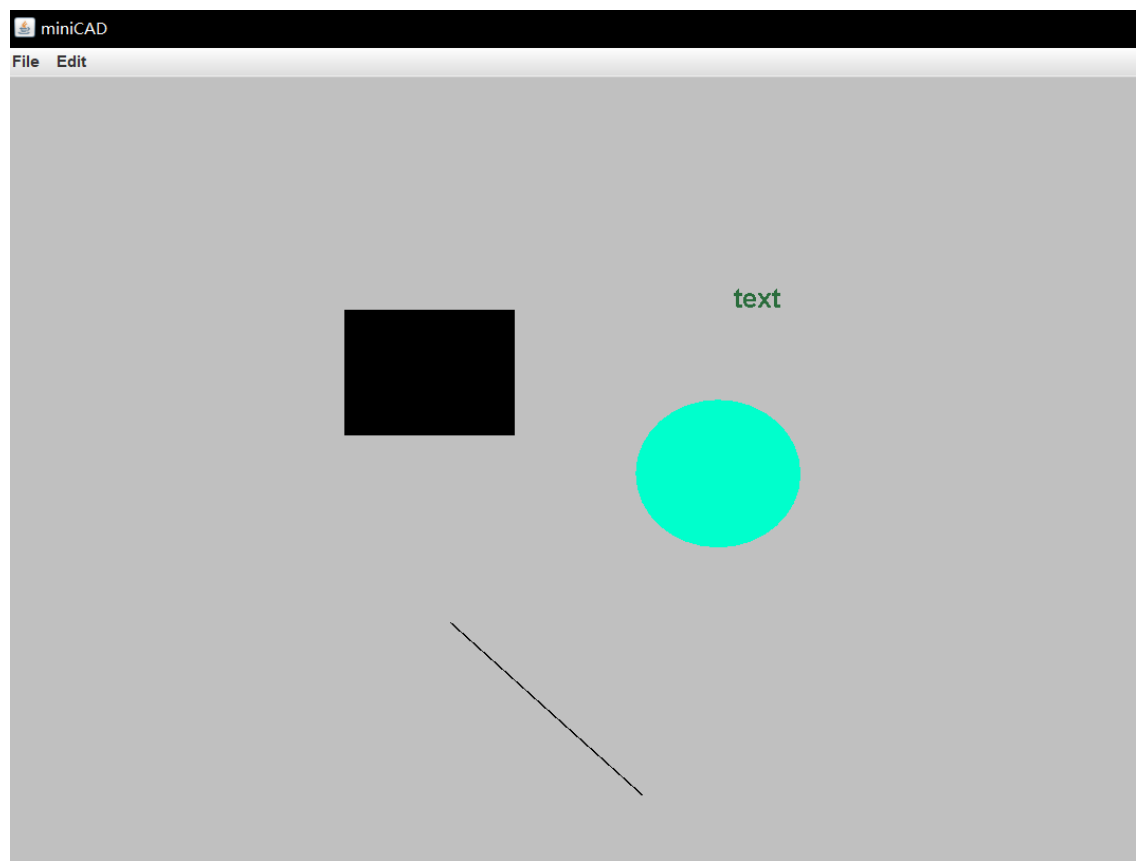
即可进行保存

- 图形文件的打开





选择文件后，即覆盖现有图形，重新加载文件中的图形。



问题与总结

- 待解决的问题
 - 直线的朝向问题

基于现有的显示结构，直线只能是矩形的对角线，因此难以将其水平或竖直放置。修复该问题可能要对直线的结构做出一些调整。
 - 图形直接的层次问题

操作中发现，新创建的图形总是可以覆盖比它更早创建的图形，无论是否被选中。它们的层次应该和它们在容器中的列表的位置有关。通过移除老的图形，再重新加入列表，可能就能解决这个问题。

- 待增加的功能

- 可以选择是否填充图形

目前默认都会填充图形，也可以选择不填充

- 可以选择画笔粗细
- 可以设置一个全局的前景色
- 可以设置背景色
- 增加未保存文件的提示功能

- 遇到但已解决的问题

- 按下按钮、弹出窗口时，被选中图形的图形总是会被污染，选中点击数次后又恢复正常。

解决：失去和获得focus时，选择全局重绘而非局部重绘

- 创建对象时无显示

原因：开始工作区加了一个和工作区规模一样大的白板组件，创建的对象都被覆盖在白板下无法显示，经过大量调试才找到原因。

解决：移除白板

- 总结

Project2有一定的工程量和复杂度，对我来说是一个挑战，也是一个有趣的作业。在实践的过程中，我更加熟悉了swing和awt库的各种常用类的使用，同时也学习了有关设计模式、消息机制的理念，为日后我参与编写前端项目（如果有）打下了一定的基础。总之，这次Project很有意义。