

Java Project - MiniCAD

姓名：王晨雨

学号：3200102324

Chapter 1 背景

1.1 问题介绍

用Java的awt和swing做一个简单的绘图工具，以CAD的方式操作，能放置直线、矩形、圆和文字，能选中图形，修改参数，如颜色等，能拖动图形和调整大小，可以保存和恢复（文件操作）。

1.2 实验环境

- IDEA：IntelliJ IDEA 2022.2.3
- Java JDK版本

```
C:\Users\17260>java --version
openjdk 11.0.12 2021-07-20
OpenJDK Runtime Environment Microsoft-25199 (build 11.0.12+7)
OpenJDK 64-Bit Server VM Microsoft-25199 (build 11.0.12+7, mixed mode)
```

1.3 Linux环境下如何编译

更改到src目录下，命令行输入 `javac @sourcefile.list` 编译java文件

编译成功后输入 `jar -cvf MiniCAD.jar *.class` 打包成 `jar` 文件

打包成功后输入 `java -Dfile.encoding=utf-8 -jar MiniCAD.jar` 运行



```
MINGW64:/d/Course/2022_2023FW/Java/project/project2/miniCAD/src
17260@LAPTOP-45N5IQPH MINGW64 /d/Course/2022_2023FW/Java/project/project2/miniCAD/src
$ java -Dfile.encoding=utf-8 -jar MiniCAD.jar
```

Tips：建议跳过自己编译和打包的步骤，就用我打包好的jar文件，或者在IDEA等软件中直接编译运行，因为我打包的时候报了错然后自己通过网上的方法解决了，有点麻烦，我会在Chapter4中说明

Chapter 2 实现

这个程序主要包含下面三个部分：主面板，交互，以及文件操作

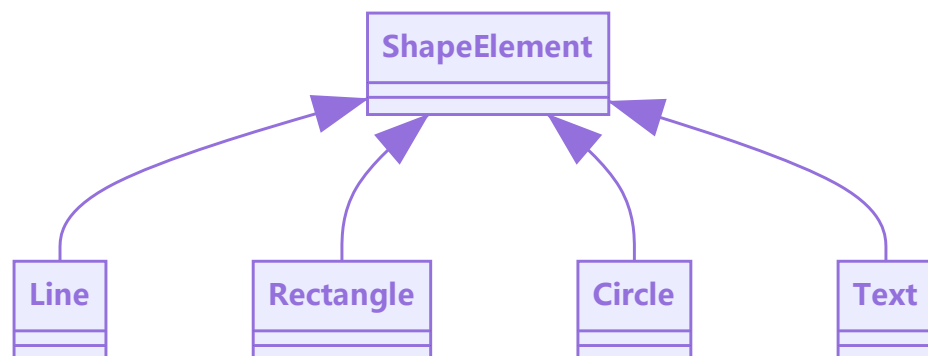
具体实现细节可以参照代码

- 主面板

相关的 `class`： `ButtonPanel`， `Initialization`， `MainPanel`

- `Initialization`

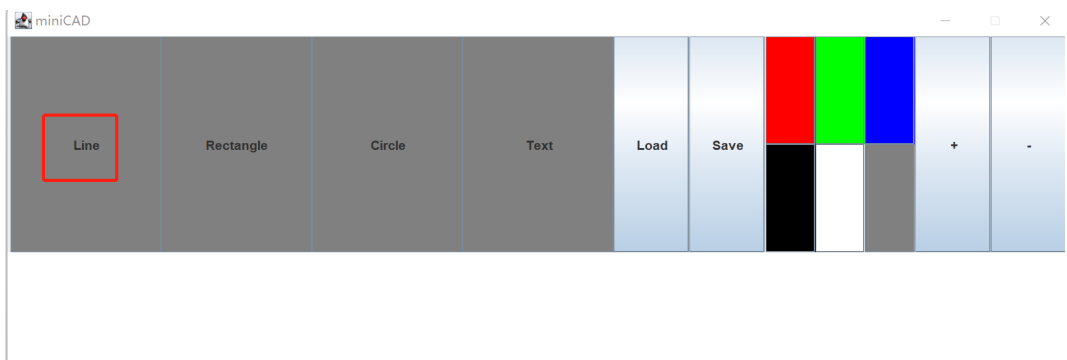
- 新建一个miniCAD窗口
 - 在上面放置许多图形模块 `ButtonPanel`
 - `ButtonPanel`
 - 添加操作面板 `Mainpanel`
 - 添加操作按钮 `JButton` 以及各自的 `ActionListener`
 - `MainPanel`
 - 白色主面板
 - 添加 `MouseMotionListener` , `MouseListener`
- 交互
- 相关的 `class` : `MainPanel` `ButtonPanel` `ShapeControl` `select` 以及 `ShapeElement`
- `MainPanel`
 - `MouseMotionListener`
 - 捕捉 `mouseDragged` 动作
 - 实现图形的拖动
 - `MouseListener`
 - 捕捉 `mouseClicked`
 - 实现选中图像
 - `mousePressed` 和 `mouseReleased`
 - 实现新增图像 `addShape()` 函数, 在 `shapeControl` 中调用新建图像的方法
 - 同时在屏幕上回显
 - `ButtonPanel`
 - 实现了按钮的 `ActionListener` 函数, 允许用户修改图像参数
 - `ShapeControl`
 - 管理界面上已有的图形
 - `public static List<ShapeElement> list = new ArrayList<>();`
 - `list` 即为所有图形 `ShapeElement` 的list
 - 提供了新增图形的方法 (`MainPanel` 中会对它进行调用) 以及移除所有图形的方法
 - `select`
 - 判断用户是否选中了一个图形
 - 选中一个图形并且实现屏幕回显
 - 被 `MainPanel` 中的 `mouseClicked` 函数调用
 - `ShapeElement`

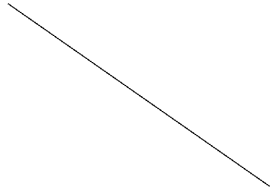
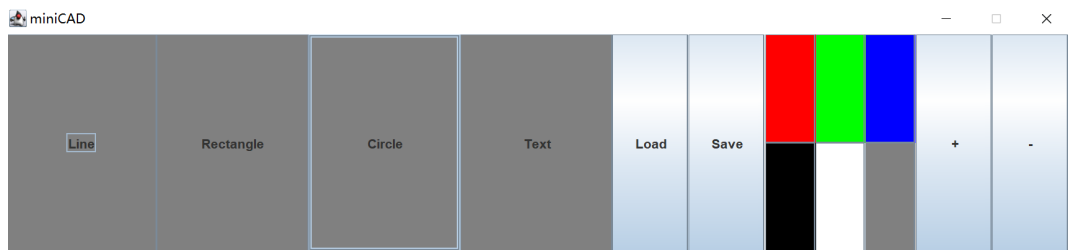


- 提供了可以添加的图形的所有类以及它们的参数（位置，颜色，是否被选中等）
 - **Line**
 - 直线的起始坐标
 - **Rectangle**
 - 矩形的左上角坐标
 - 长宽
 - **Circle**
 - 圆的左上角坐标
 - 半径（ $a==b$ 时，可以把a和b看作半径）
 - **Text**
 - 位置坐标
 - 文本内容
- 文件
 - 相关的 **class** : **FileOperation**
 - **FileOperation**
 - 点击 **ButtonPanel** 的 **Load** 和 **Save** 时候调用
 - 实现将文件写入和读出的方法
 - 本质上就是把 **ShapeControl** 中的 **list** 写入文件再读出

Chapter 3 测试

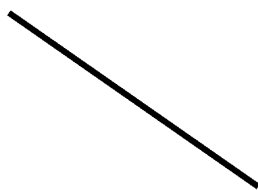
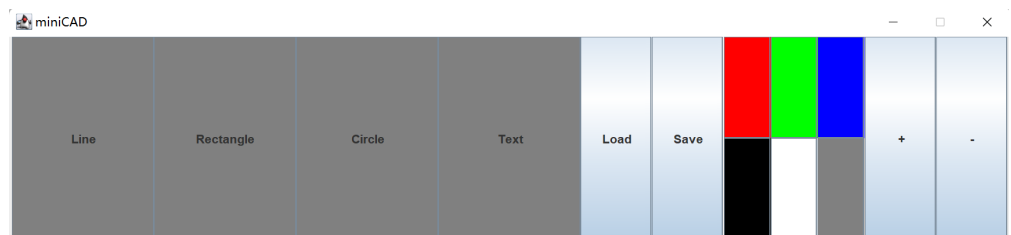
- 画一条直线
 - 选中左上角的Line，在白色界面上拖动鼠标从直线起始段拖到末端即可画出直线



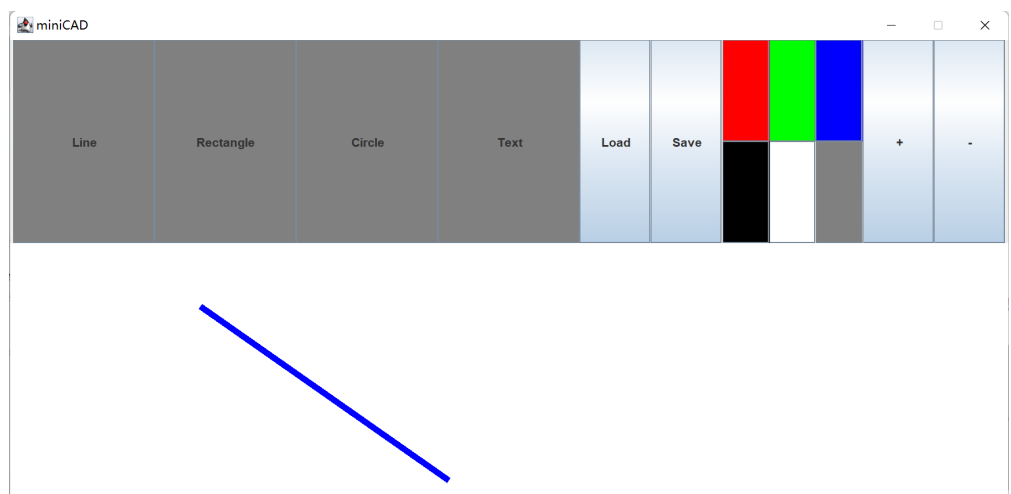


。更改颜色

- 选中直线（鼠标点击直线），它就会变粗显示选中（选中后，鼠标单击旁边的空白处即可取消选中）



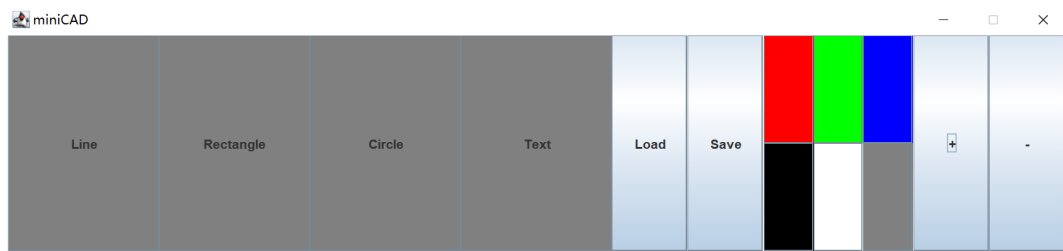
- 在选中状态下点击右上角那几个颜色，就可以改变直线的颜色



- 调整大小

在选中状态下，点击右上角的 + 增大图像， - 减小图像

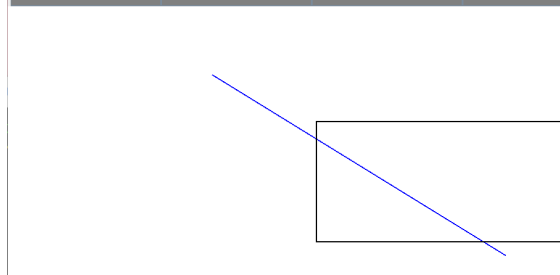
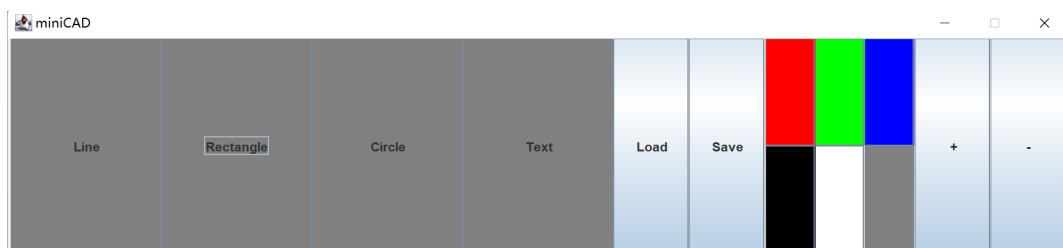
此时直线增长



- 移动位置（同样在选中的状态下拖动，见下文的矩形）

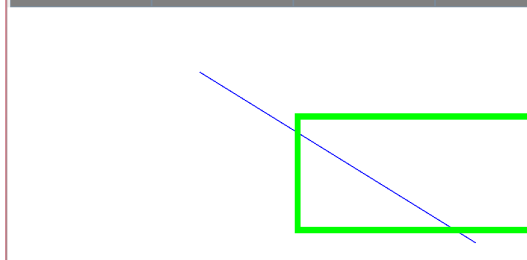
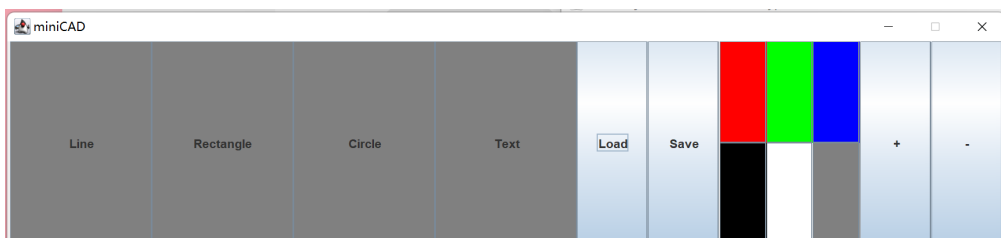
- 画一个矩形

- 选中上侧的Rectangle，同画直线的方法：



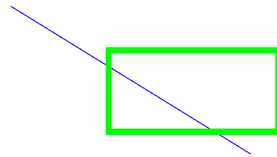
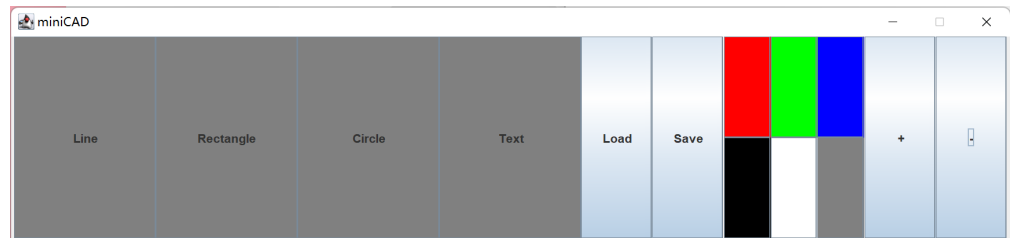
- 更改颜色

- 与line一样的方法，点击矩形内部选中矩形，即可更改大小和颜色

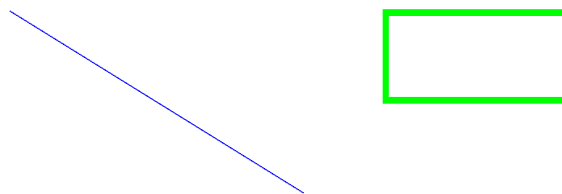


- 调整大小

■ 变小

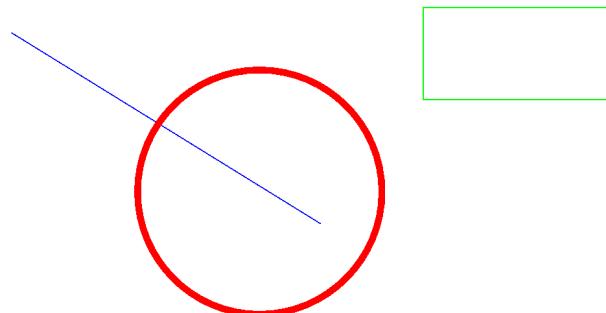
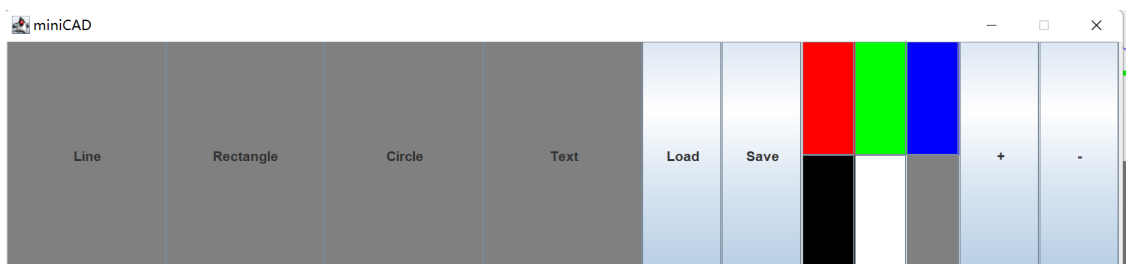


○ 移动位置



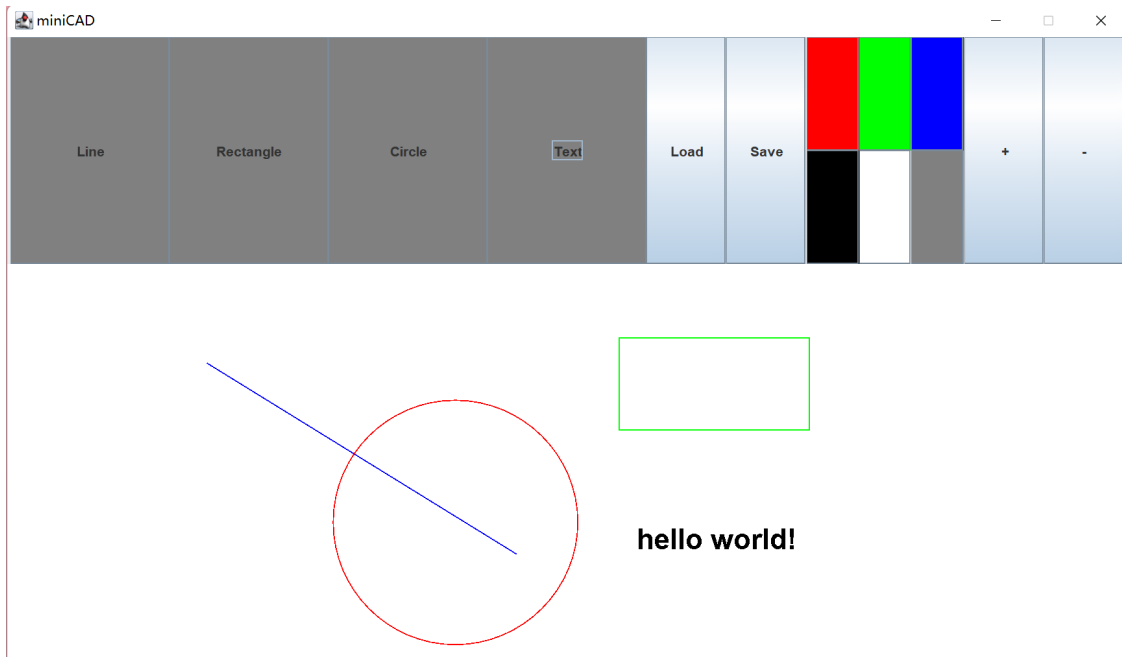
• 画一个圆形

所有操作和直线和矩形一样

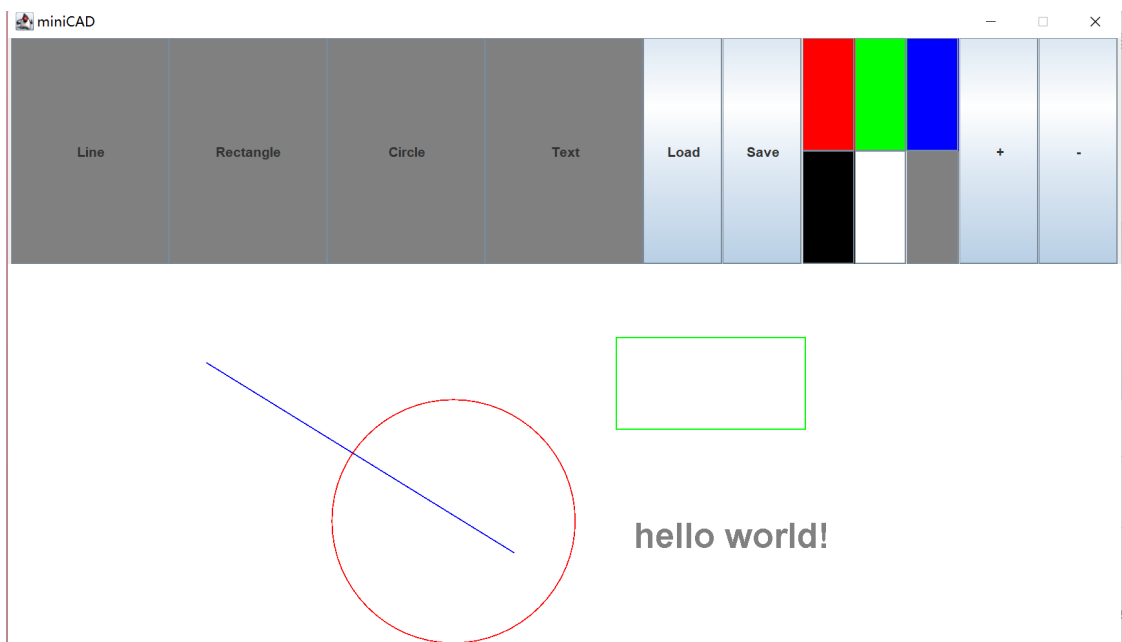


• 输入文本内容

选中Text，在屏幕上拖动你想放置文本框的位置，在弹出的对话框中输入你想输入的内容

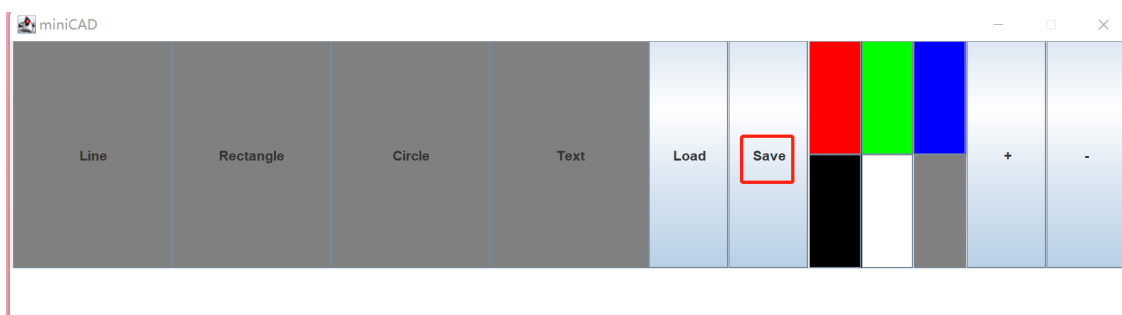


选中文本框后，字体会变大，表明已经选中，此时你可以和图型一样更改大小和颜色。

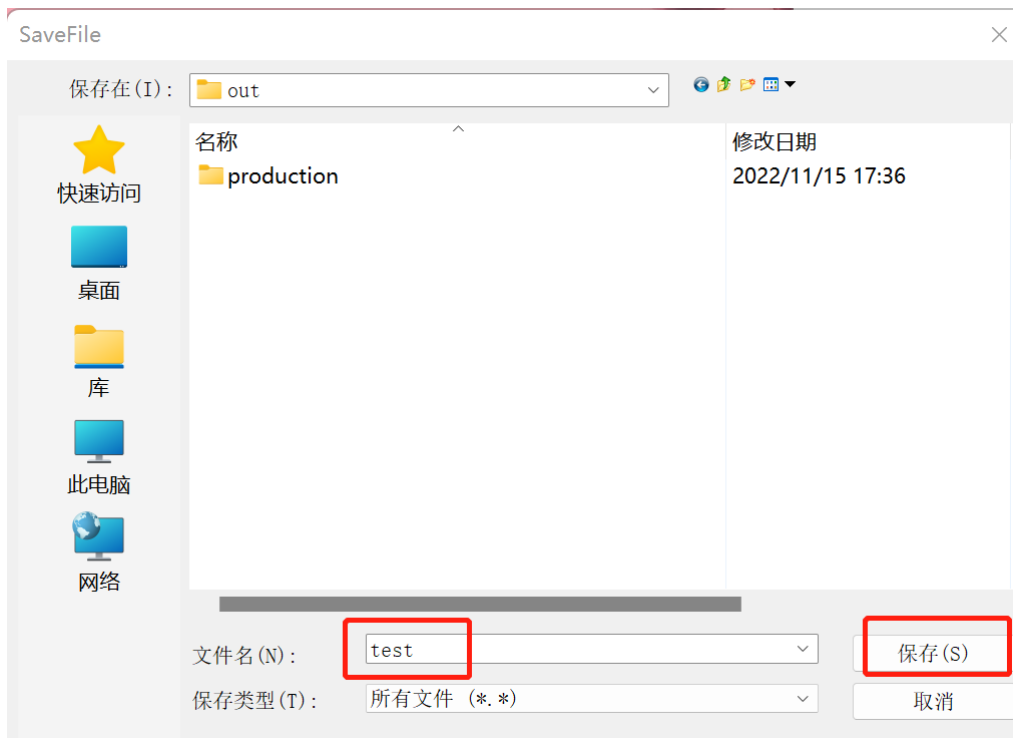


- 保存

选择上册的Save按钮

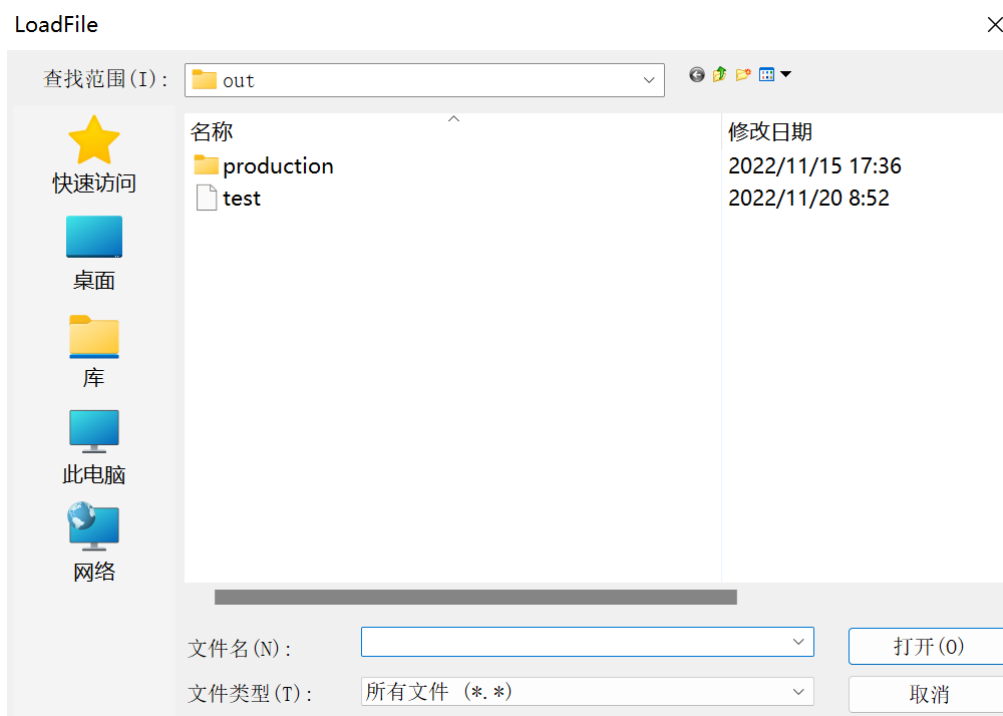


在弹出的会话框中选择保存路径后选择保存即可

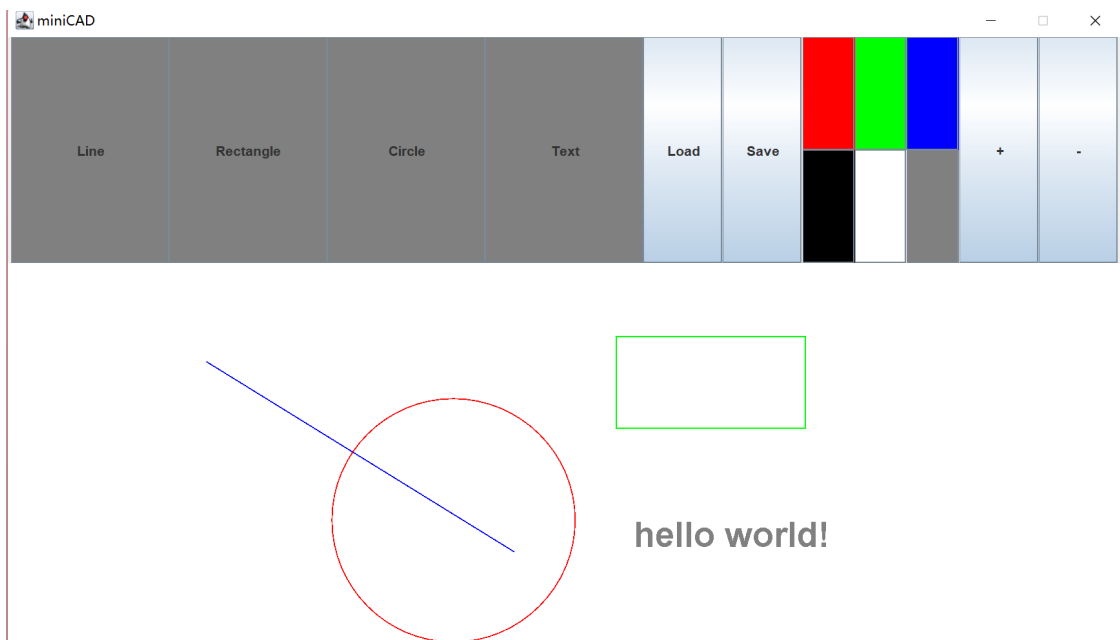


- 恢复

打开一个空的miniCAD程序，单击load



选择test点击打开，此时即可出现刚刚保存的文件



此时仍能根据上述方法修改参数和大小。

Chapter 4 问题与总结

- 编译的问题

我一开始写代码都是用的idea上直接编译运行调试的，最后我在linux平台上打包生成jar包，后来发现了许多问题

- linux平台上用 `javac @sourcefile.list` 编译时，会报错（注释是中文然后导致了乱码），应该用utf-8编码，然后我索性把注释改成了英文，不过应该有更好的解决方法
 - 在打包jar包时，我选择了 `-Dfile.encoding=utf-8` 参数，这样命令行上的信息也不会出乱码
 - 打包jar包报错：jar中没有主清单属性。后来我上网查询解决办法，根据这篇博客：https://blog.csdn.net/qq_43649223/article/details/108766040，在我的jar包下的 `MANIFEST.MF` 文件中末尾添加 `Main-Class:Initialization` 保存后重新打包即可
- 有的时候操作太快，可能会把 `mouseclick` 识别为 `mousedrag`