

浙江大学实验报告

课程名称： Java应用技术 实验类型： 综合型

实验项目名称： MiniCAD

学生姓名： 管嘉瑞 专业： 计算机科学与技术 学号： 3200102557

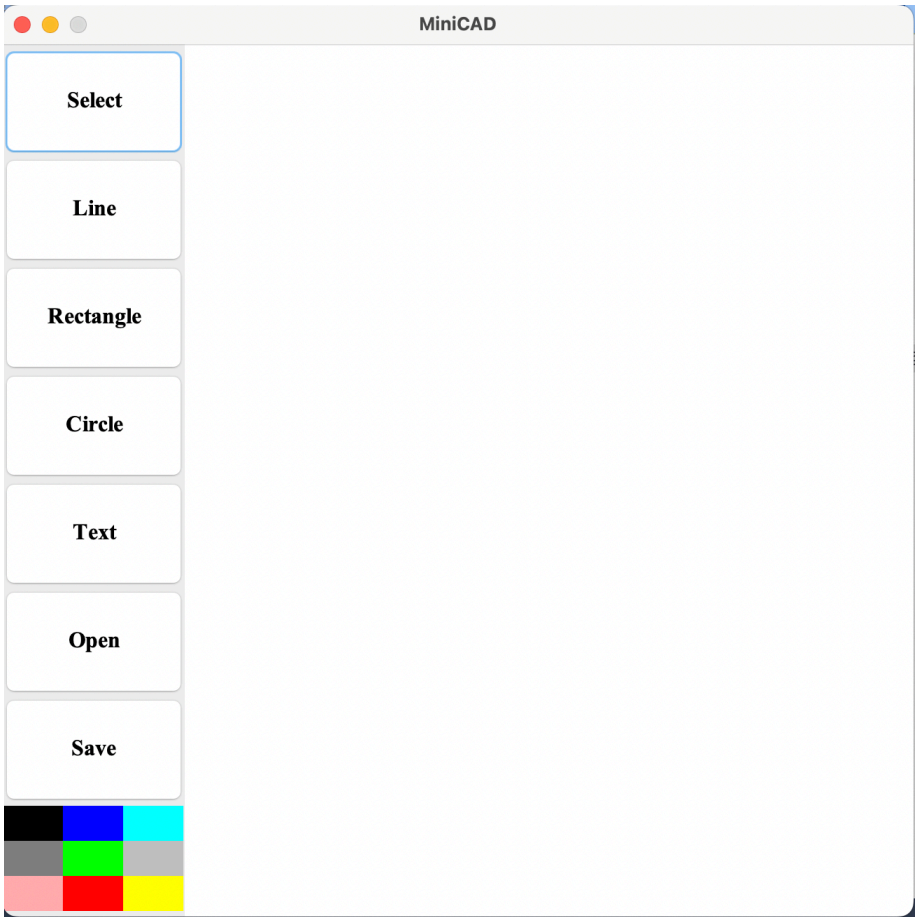
电子邮件地址： 3200102557@zju.edu.cn 手机： 13588084334

实验日期： 2022 年 11 月 23 日

一、功能需求

本实验要求用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。

本实验设计的MiniCAD满足了以上全部需求，界面友好、容易上手。



二、环境配置

- Java版本：

openjdk 18.0.2.1 2022-08-18

OpenJDK Runtime Environment (build 18.0.2.1+1-1)

OpenJDK 64-Bit Server VM (build 18.0.2.1+1-1, mixed mode, sharing)

- Linux/MacOS平台运行指令:

在根目录下

```
1 | $ java -jar MiniCAD.jar
```

若需要重新编译运行，可运行脚本文件

```
1 | $ ./run.sh
```

此脚本文件将重新编译、打包jar文件、运行。

- 其他请参考ReadMe.md

三、实验内容

- 模块整体设计

除了Main类外，程序设计了五个类：工具栏(ToolBar)、画板(MyPanel)、图形(Item)、图形管理器(ItemManager)、文件管理器(FileManager)。其中工具栏和画板与用户直接交互，图形管理器维护了画板上的图形链表，并提供了选择图形的接口。图形类有四个子类。文件管理器负责cad文件的保存、读取工作。

- 工具栏模块

工具栏位于界面左侧，包含七个按钮和一个3x3的颜料版。七个按钮设置了鼠标点击的响应函数。

工具栏类中维护了成员变量btnStatus，记录的是当前选中的按钮的编号，按钮响应函数会修改这个值。btnStatus决定了画板上操作的行为。注意工具栏类中创建了画板对象，可以通过画板的setStatus函数修改画板的状态控制变量。

```
1 | class ButtonListener implements ActionListener {
2 |     private int status;
3 |
4 |     public ButtonListener(int status) {
5 |         this.status = status;
6 |     }
7 |
8 |     //修改选中按钮的文字颜色
9 |     @Override
10 |    public void actionPerformed(ActionEvent e) {
11 |        jbtn[btnStatus].setForeground(Color.BLACK);
12 |        btnStatus = status;
13 |        jbtn[btnStatus].setForeground(Color.RED);
14 |        cadPanel.setStatus(btnStatus);
15 |    }
16 | }
```

- 画板模块

画板的功能是根据工具栏选中的按钮，对鼠标、键盘行为进行响应，如绘制图形、修改图形属性等。

注意工具栏状态大体上分为选中图形状态、绘制图形状态、空状态三种，要根据不同的状态对鼠标行为进行不同响应。

画板与图形管理模块也有交互，表现为画板可以从ItemManager中获取当前被选中的图形、绘制过程中需要预览的图形，也可以通过其对图形进行修改、删除。另外还需要借助ItemManager的选择函数判定选中逻辑。

- 图形与图形管理模块

图形类Item有直线(Line)、矩形(Rectangle)、圆(Circle)、文本(Text)四个子类。需要实现绘制、设置颜色、更改大小、更改粗细等方法。

图形管理器ItemManager负责图形链表的维护、预览图形维护、选择函数doSelect的实现。

```
1  public static void doSelect(int x, int y)
2  {
3      boolean flag = false;
4      for (int i = 0; i < itemList.size(); i++) {
5          Item e = itemList.get(i);
6          e.isSelected = false;
7          if (flag == false) {
8              if (e instanceof Line) {
9                  if (lineDist((Line) e, x, y) <= selectDelta) {
10                     selectedIndex = i;
11                     e.isSelected = true;
12                     flag = true;
13                 }
14             } else if (e instanceof Rectangle) {
15                 if (rectDist((Rectangle) e, x, y) <= selectDelta) {
16                     selectedIndex = i;
17                     e.isSelected = true;
18                     flag = true;
19                 }
20             } else if (e instanceof Text) {
21                 if (textDist((Text) e, x, y) <= selectDelta) {
22                     selectedIndex = i;
23                     e.isSelected = true;
24                     flag = true;
25                 }
26             } else {
27                 if (ovalDist((Circle) e, x, y) <= selectDelta) {
28                     selectedIndex = i;
29                     e.isSelected = true;
30                     flag = true;
31                 }
32             }
33         }
34     }
```

```
35     if(!flag)
36         selectedIndex = -1;
37 }
```

其中维护的selectedIndex表示的当前选中的图形的下表，doSelect中可以进行设置。doSelect接受点坐标(x, y), 然后选择图形中接近该点、可以被选中的图形，其中需要跟不同图形计算距离，不同图形逻辑不同。

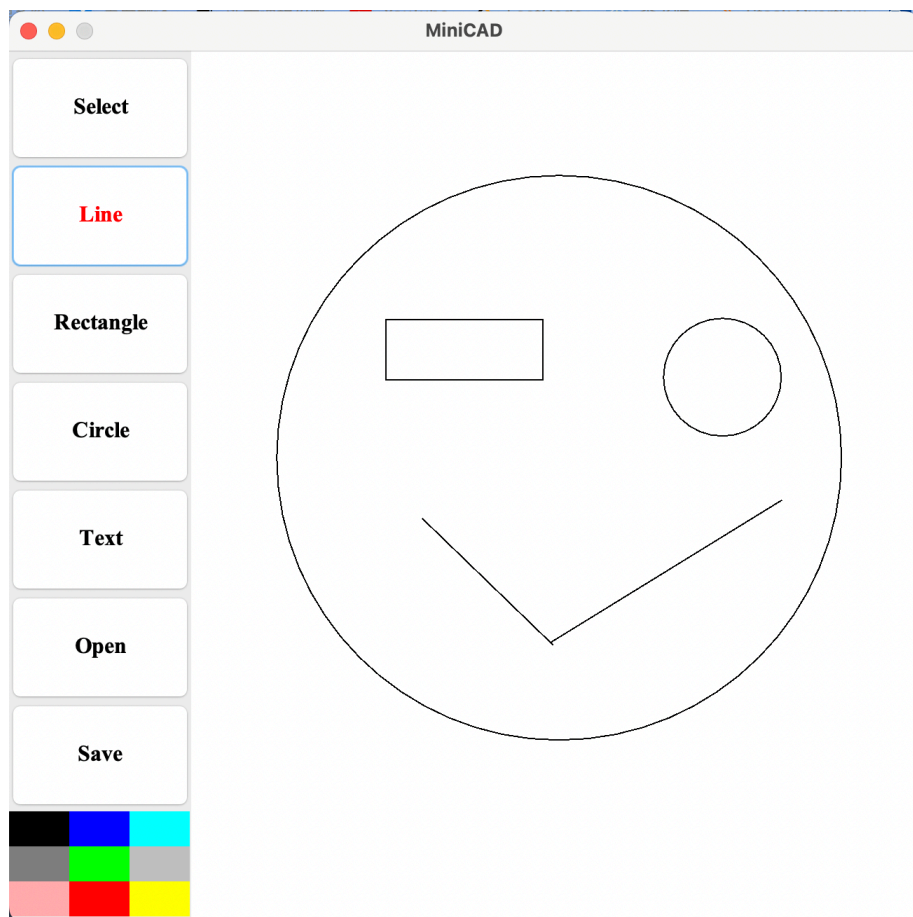
- 文件管理模块

FileManager负责文件的保存读取，文件以文本形式存储，记录了图形的属性。

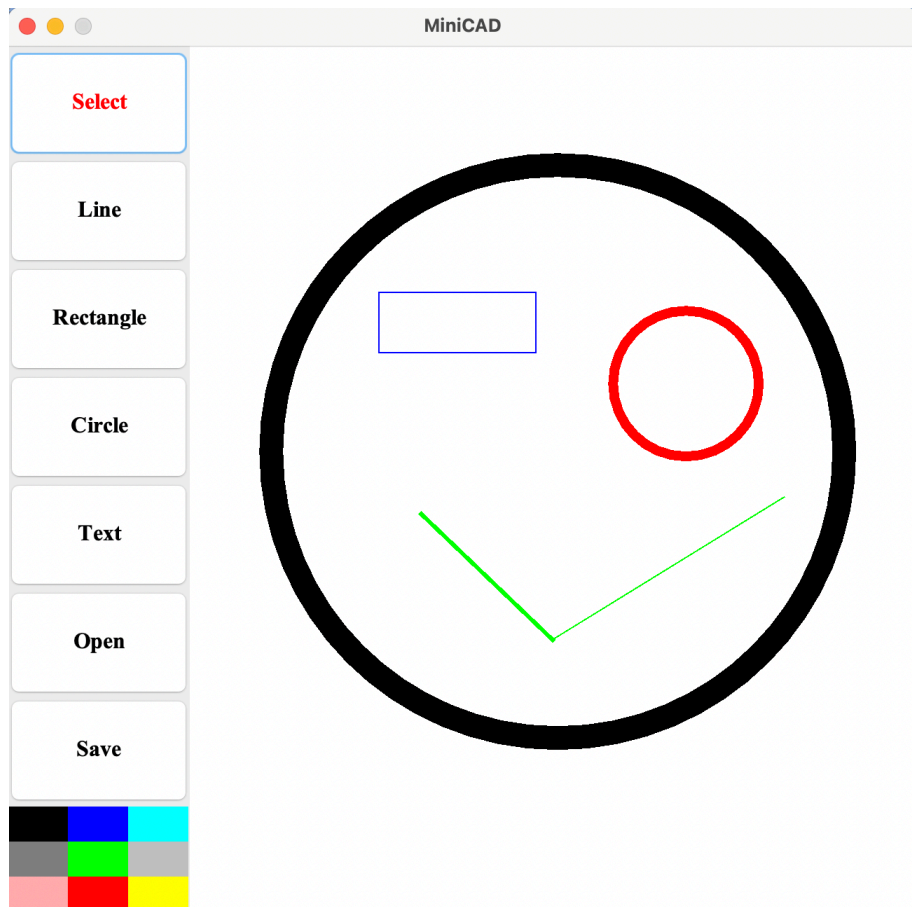
在实现中要特别注意异常捕捉处理。

四、实验结果和分析

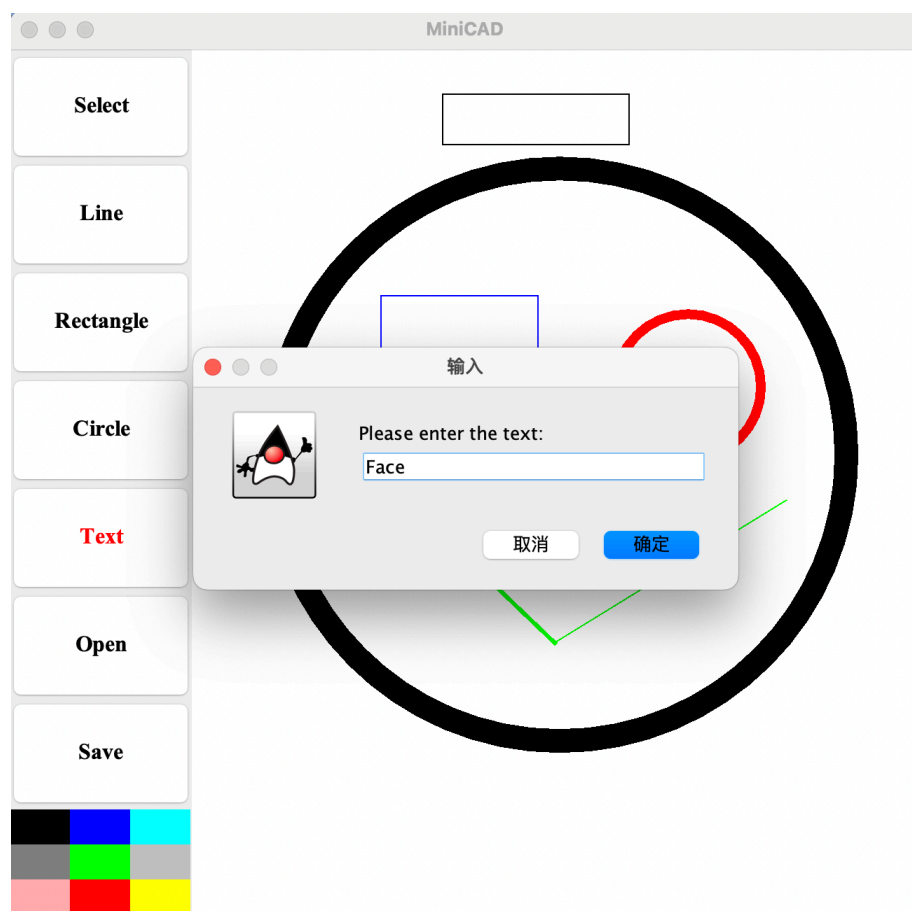
- 打开MiniCAD并绘制一些基本图形：

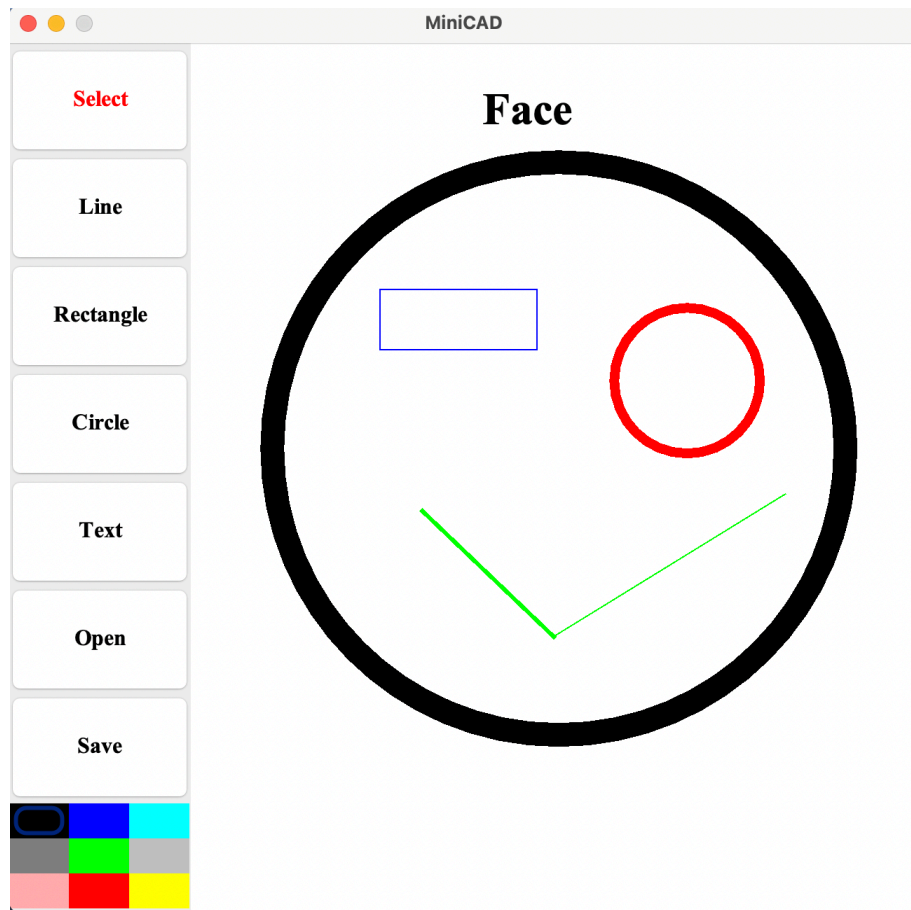


- 选中图形并改变图形属性（大小、粗细、颜色、位置等）

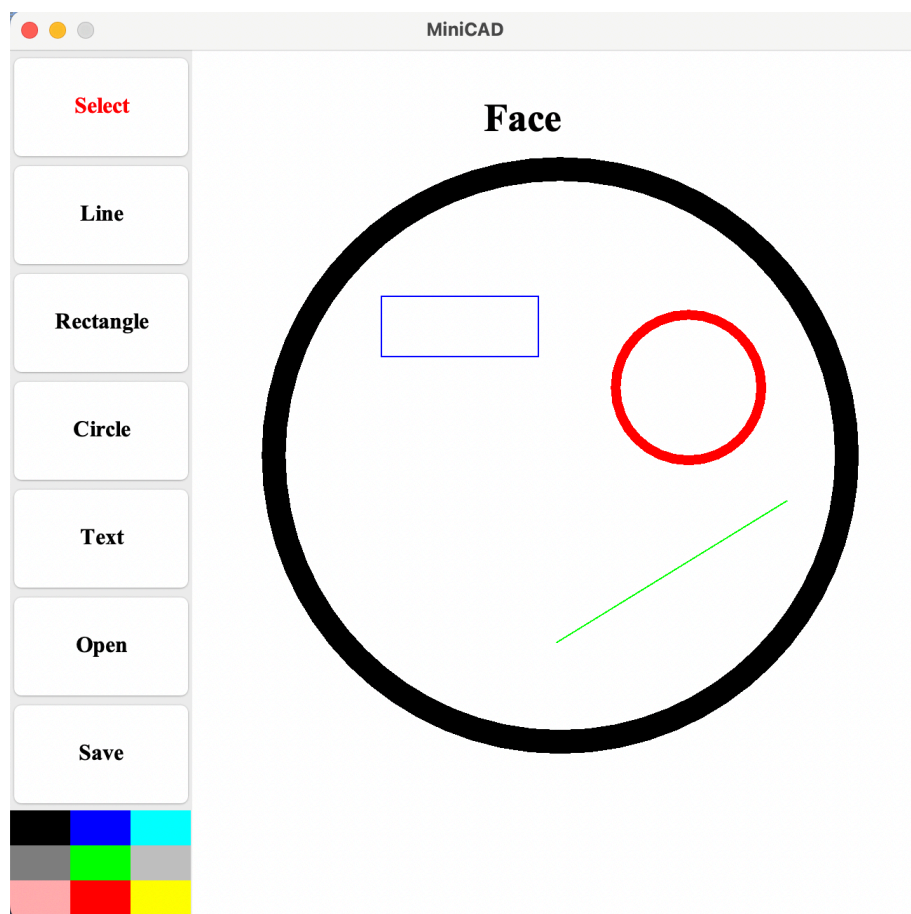


- 添加文本并改变文本属性

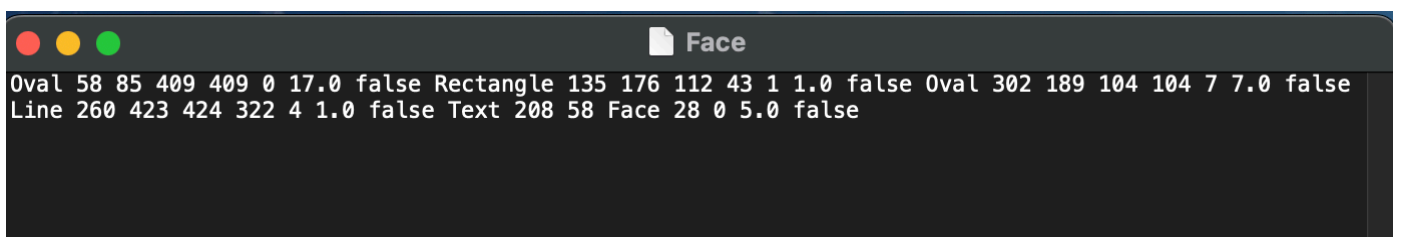
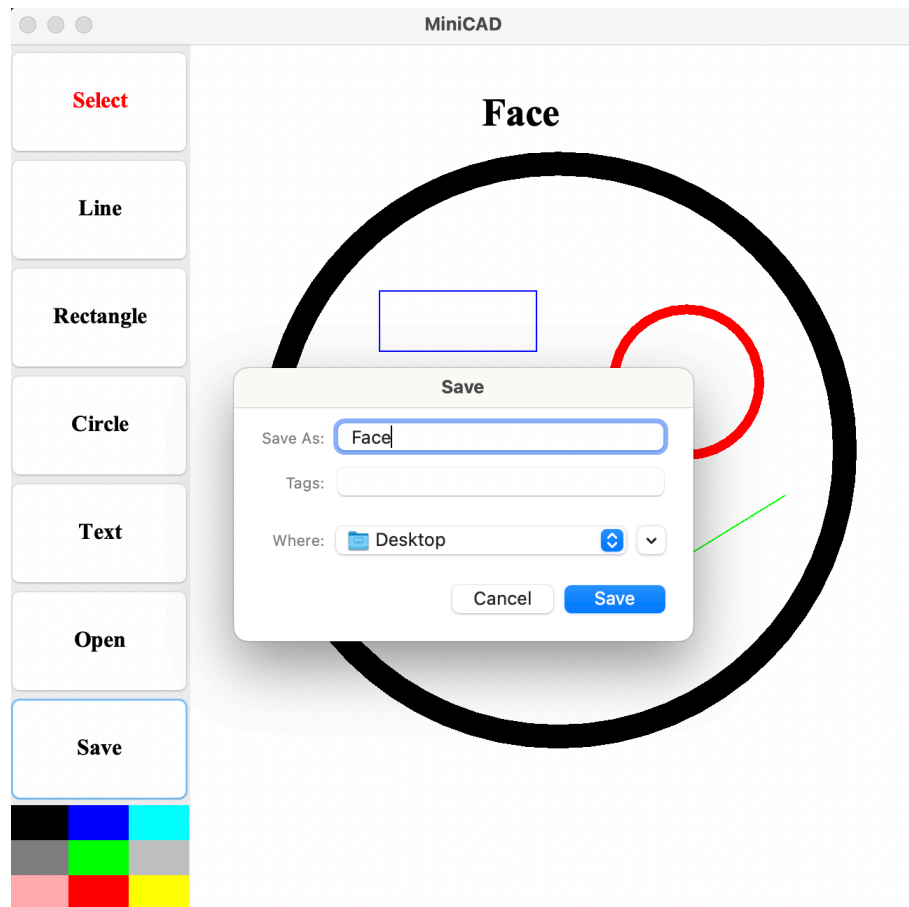




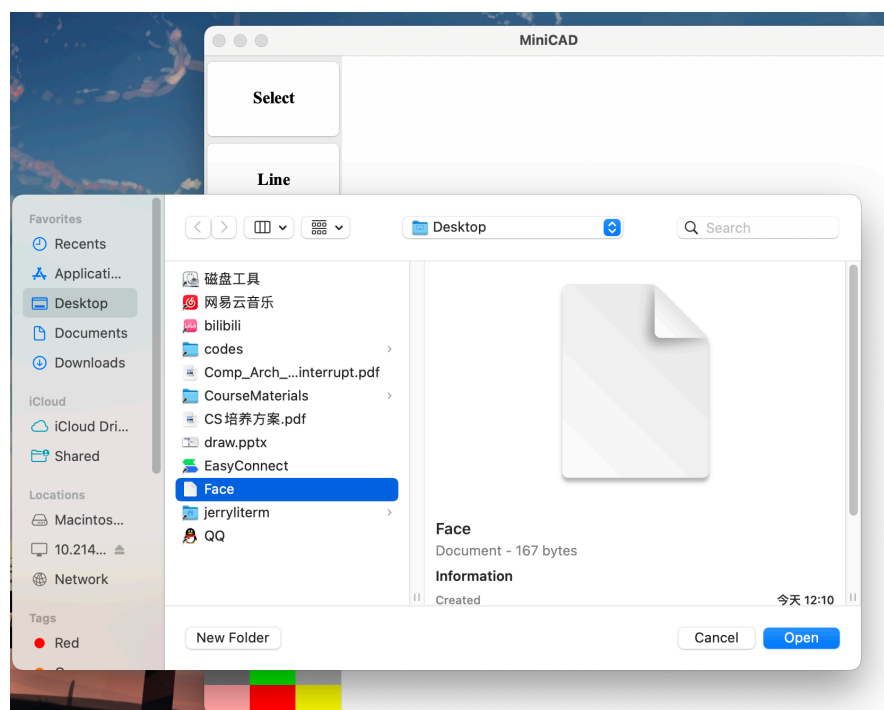
- 删除图形（选中图形并按D或删除）

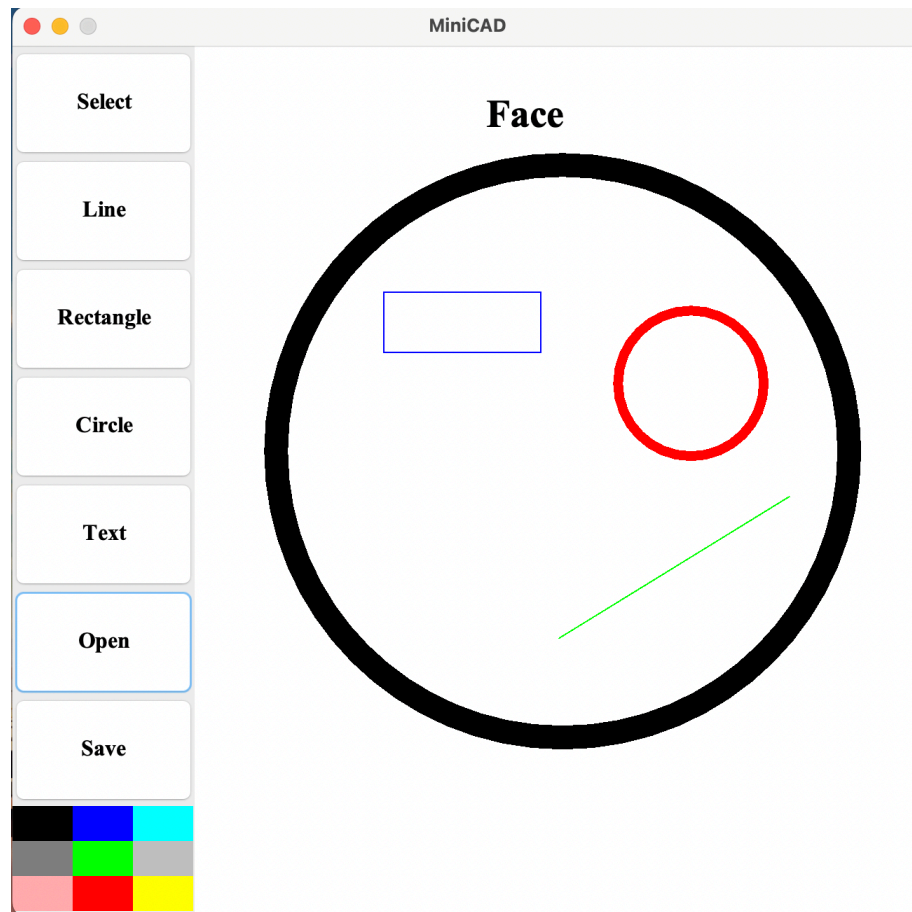


- 保存文件(文件保存到桌面上，可以查看文件内容)



- 重新打开MiniCAD软件，读取文件





文件被成功读取，并且可以在上面继续编辑。

五、实验心得

本次实验让我熟悉了Java到awt和swing，对Java对GUI编程有了更深的理解。

另外就是对Event Model更加熟悉了，之前C语言课程、数据库课程分别用c库、Qt的回调函数设计了类似的用户交互程序。Java实现起来更为清晰、抽象地更为完整，困难少了很多，这也可能是Java作为完全面向对象语言的优点吧。

实验中的难点主要是鼠标位置与图形绘制的逻辑、文件保存读取机制的设计，经过搜索资料、阅读代码和自己反复思考后克服了困难。