

MINI CAD

3160101557 商钰坤

项目要求

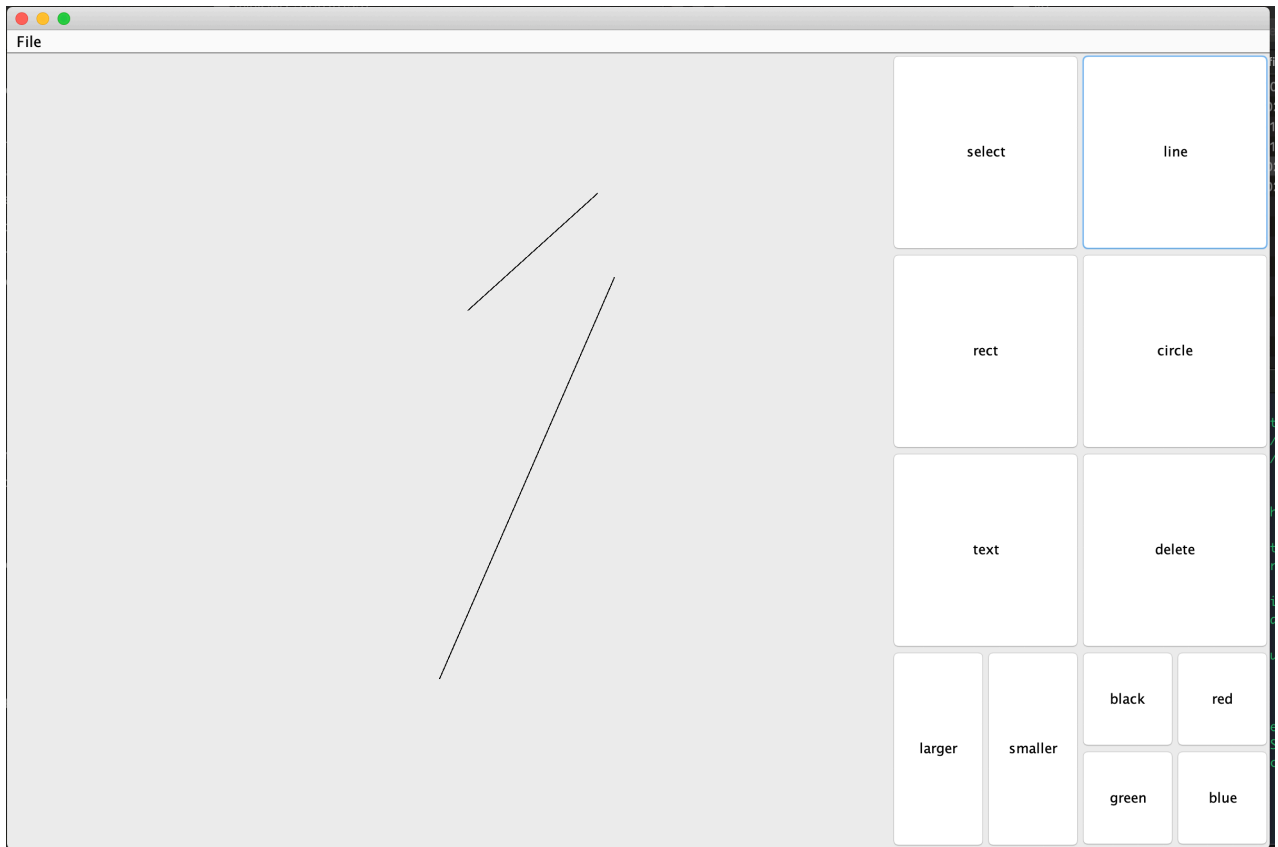
做一个简单的绘图工具，以CAD的方式操作，能放置直线、矩形、圆和文字，能选中图形，修改参数，如颜色等，能拖动图形和调整大小，可以保存和恢复。功能请参考视频演示

实现功能

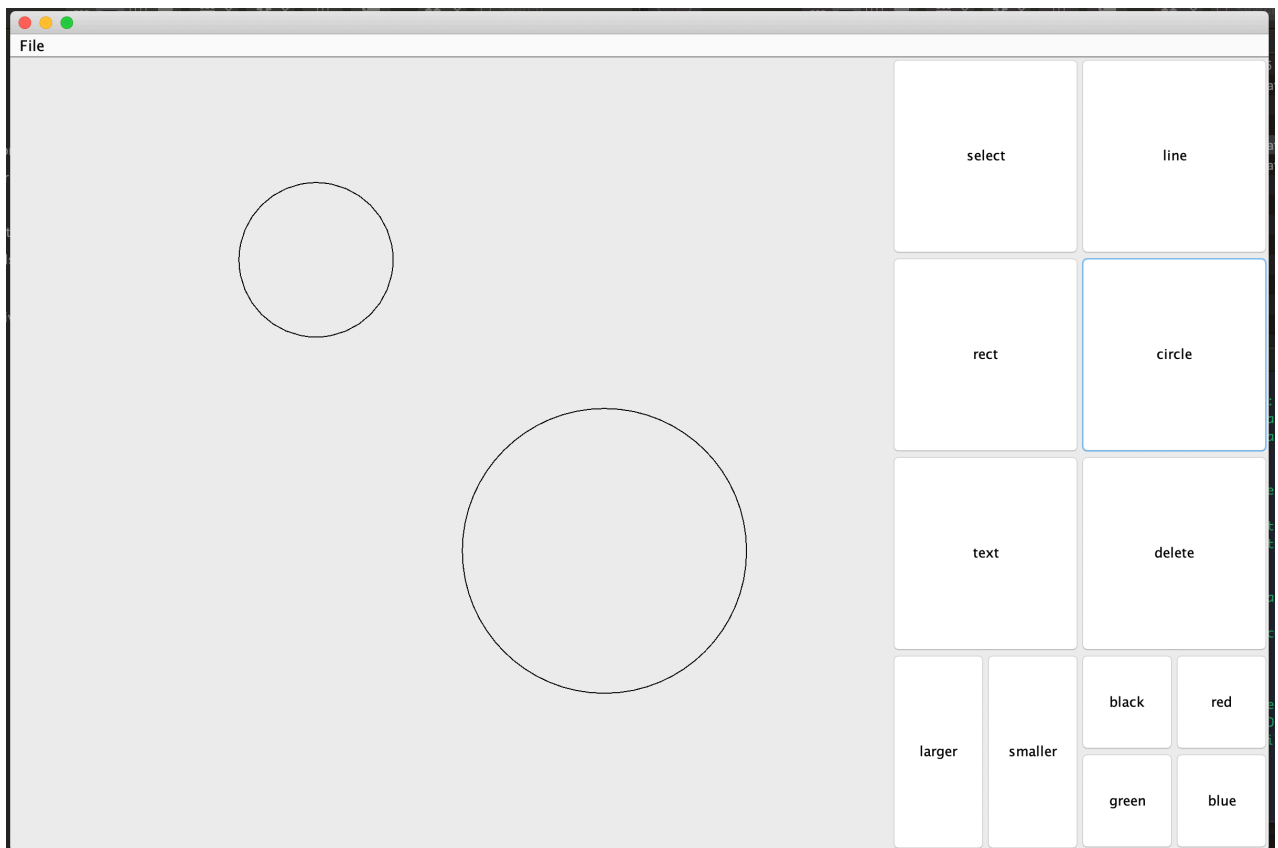
- 画直线
- 画圆
- 画矩形
- 改变大小：（直线、圆、矩形）
- 改变颜色：（直线、圆、矩形、文字）
- 输入文字：输入后用鼠标点击拖动后得到，文字可以改变大小
- 选中后删除
- 保存和恢复

功能演示

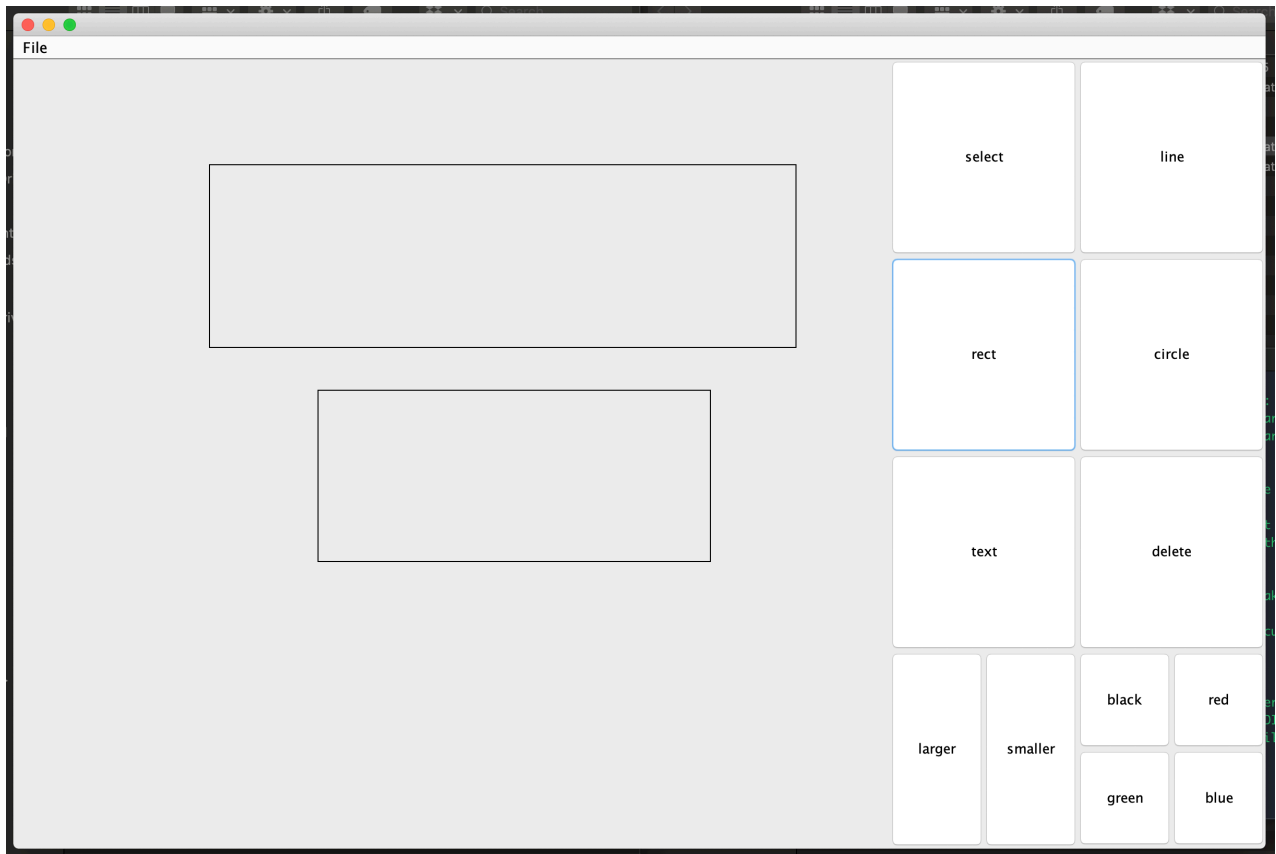
- 画直线：



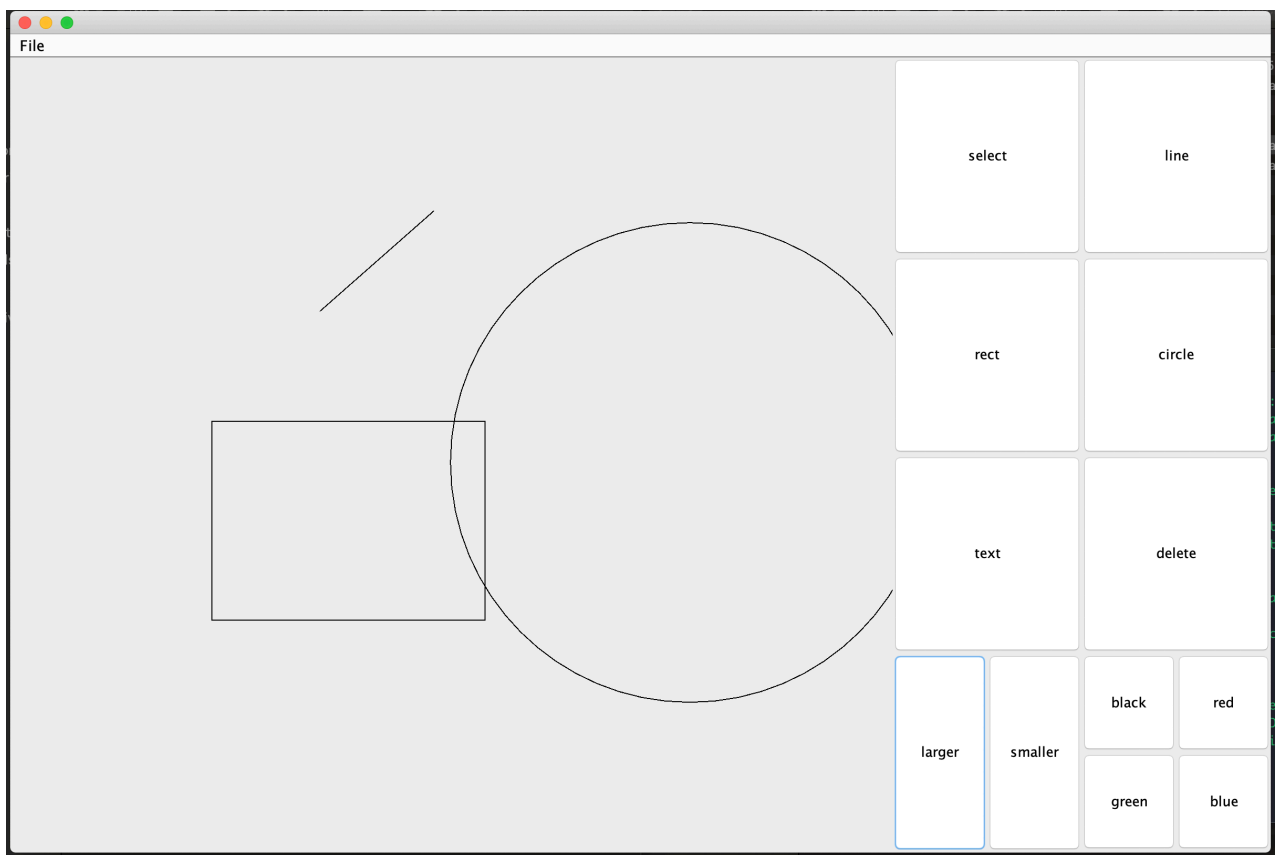
- 画圆



- 画矩形

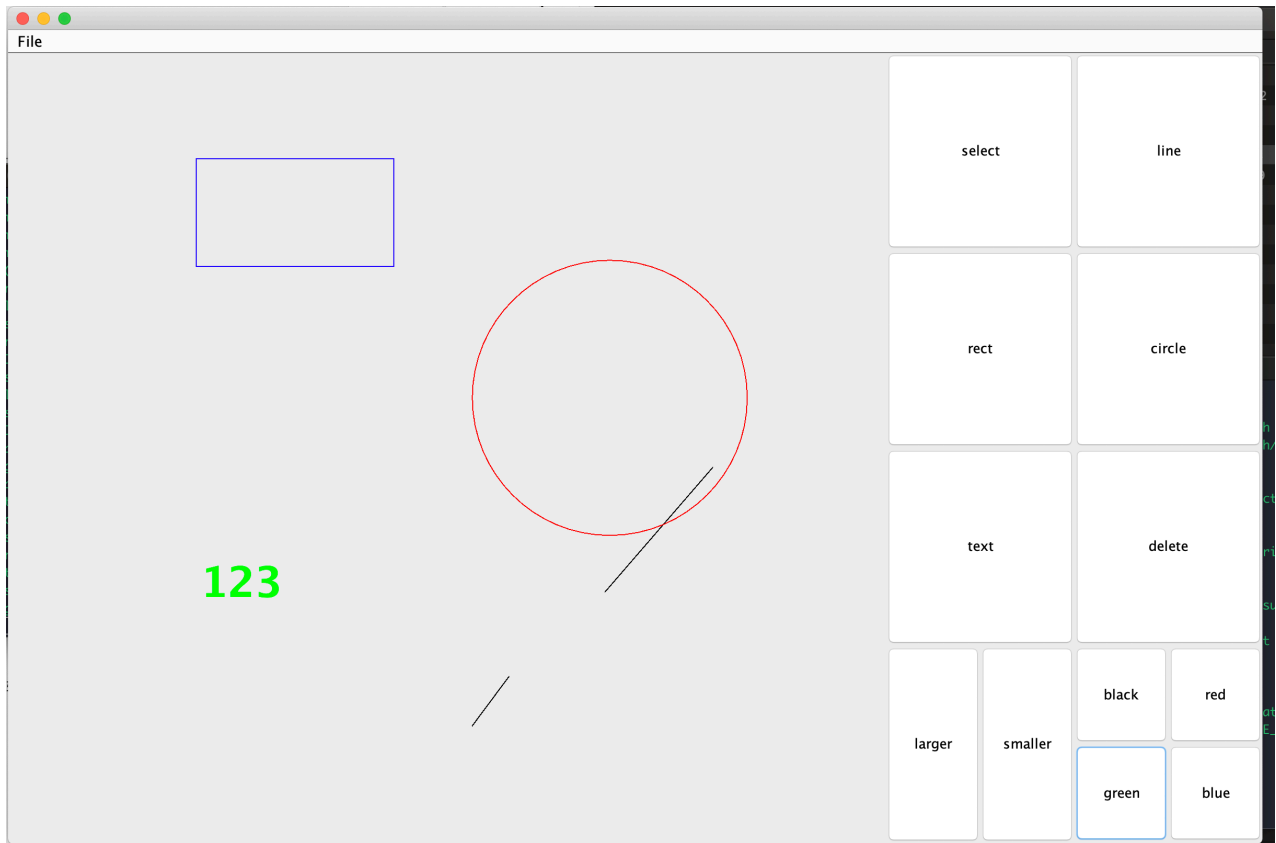


- 改变大小：（直线、圆、矩形）

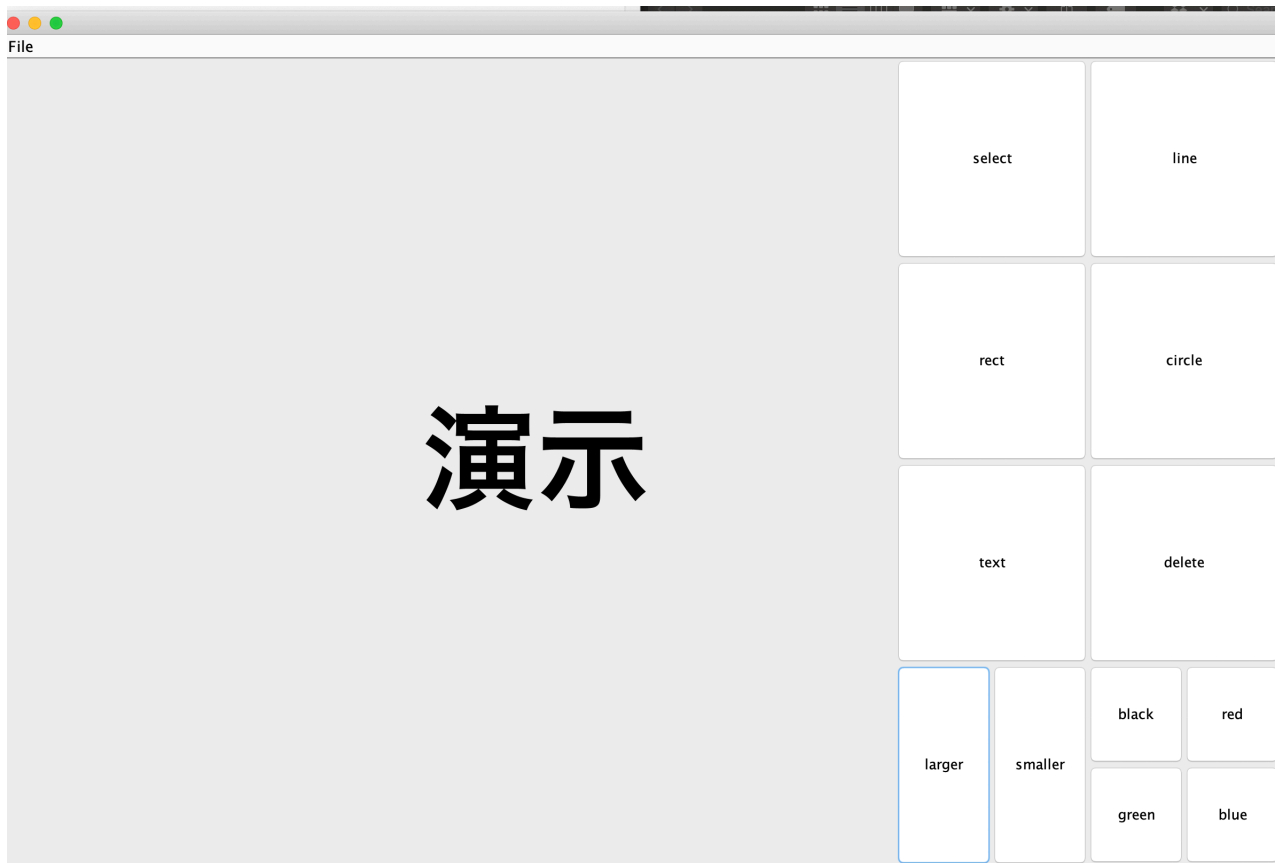


按下“larger”或者“smaller”实现

- 改变颜色：（直线、圆、矩形、文字）



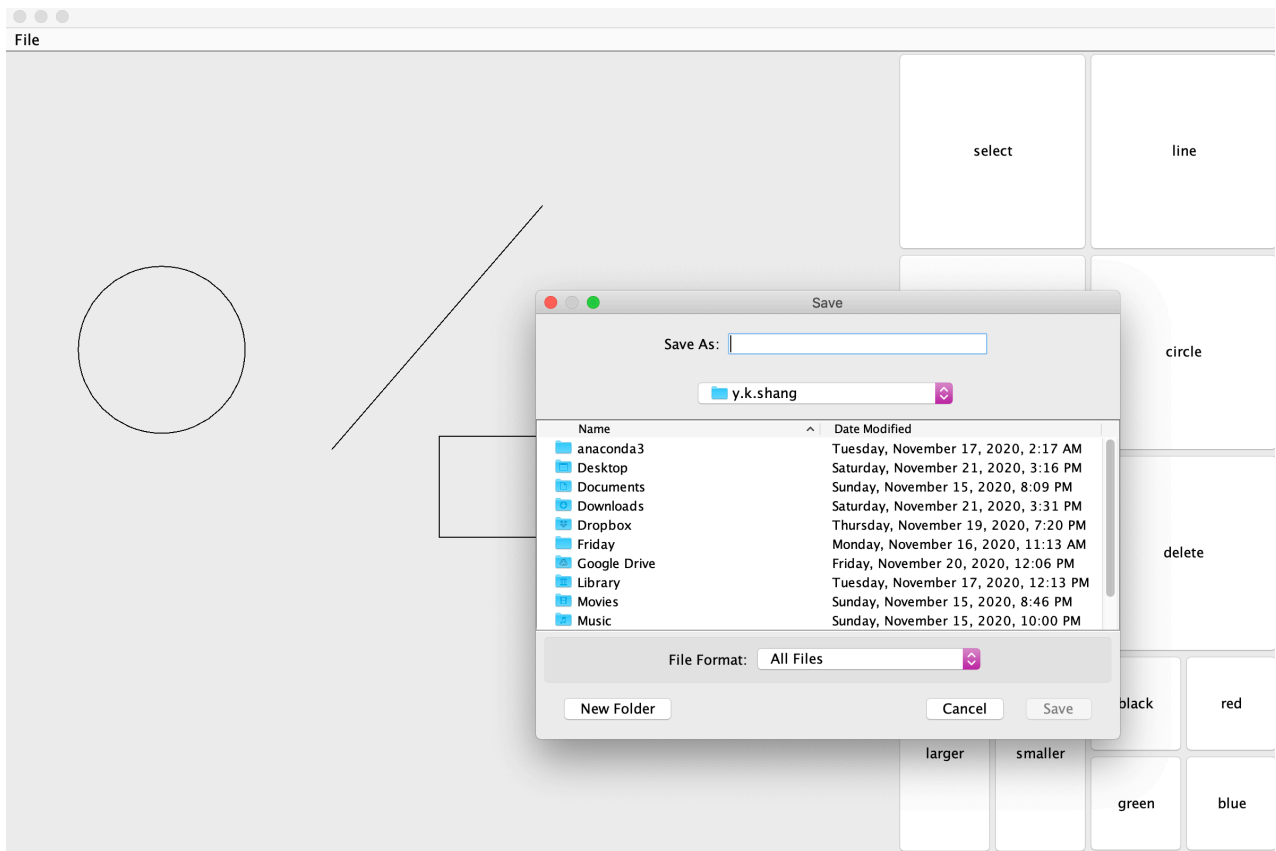
- 输入文字：输入后用鼠标点击拖动后得到，文字可以改变大小



- 选中后删除

选中图形，按下delete删除图形。

- 保存和加载



保存：可以保存到指定地址。

加载：从已有文件加载。

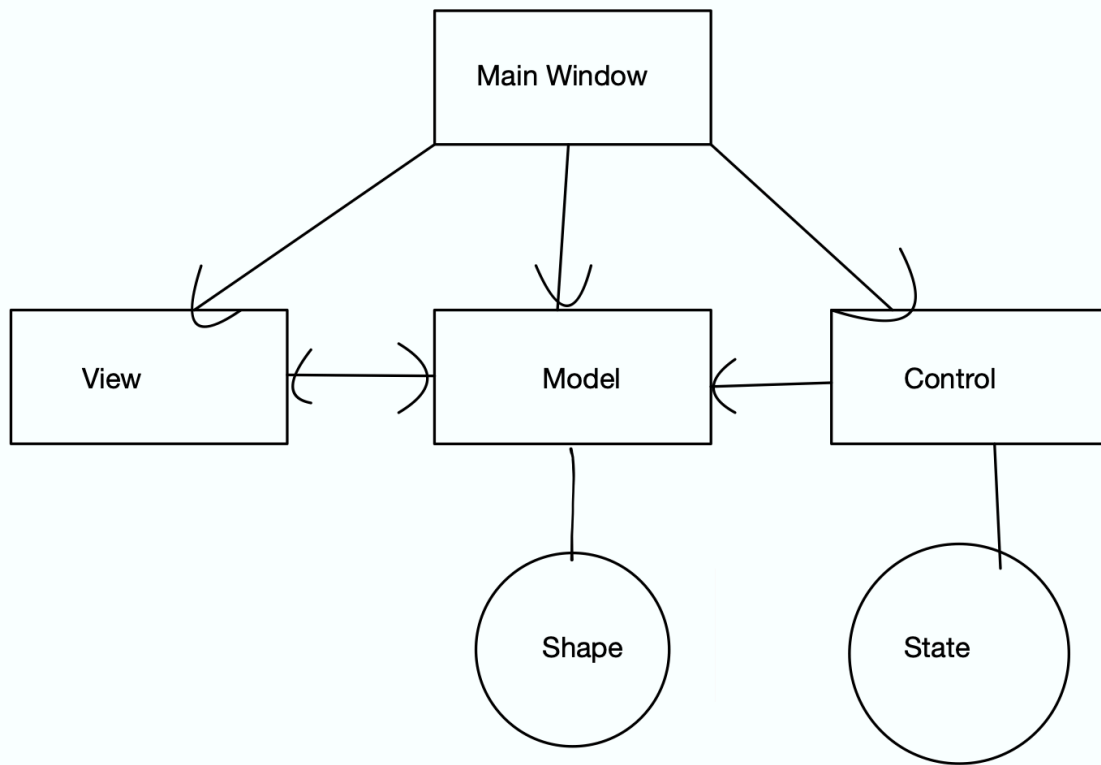
设计思路

MVC模型：

- View，Model和Control之间的关系：View和Model可以相互传递消息，Control把消息传递给Model。View和Control之间不传递消息。
- View得到Model传递的shape_list，遍历容器中的元素，绘制图形。
- Model存储了各种图形及其实现方法。
- Control 监控鼠标事件，将用户做出的改变传递给Model。同时Control负责管理状态机，相同的输入在不同的状态下有不同的输出。
- 一般消息传递的流程：control接收来自用户的操作消息，维护state，再将消息传递给model，model类更新shape_list，消息传递给View，view根据来自model的shape_list，重新绘制。

设计模式介绍

本项目MVC模型如下图所示：



核心代码介绍

- Main.java文件

设计Main.Window。其主要功能是从JFrame中继承一个Mainwindow类，在里面设计各Jbutton和Jpanel来对窗口进行布局。例如建立“select”按钮：

```
JButton select_btn = new JButton("select");
select_btn.addActionListener(control.new StateButtonListener());
opetaion_panel.add(select_btn);
```

- View.java文件

从JPanel中继承View类，View类的布局被放置在Mainwindow的Center位置。接受来自Model的消息，

```
public void modified() {
    repaint(); //调用paintComponent 去绘制
}
```

并重新绘制图形：

```
//重新绘制
@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    shape_list = model.get_shape_list();
    for (Shape shape : shape_list) {
        shape.render((Graphics2D) g);
    }
}
```

- Model.java文件

通过modified函数通知View有内容被修改, Model 相关代码如下:

```
public class Model {
    ArrayList<Shape> shape_list = new ArrayList<>();
    View view;
    Control control;

    public ArrayList<Shape> get_shape_list() {
        return shape_list;
    }
    public void set_view(View view) {
        this.view = view;
    }
    public void modified() {
        view.modified(); // tell view
    }
    public void set_shape_list(ArrayList<Shape> shape_list) {
        this.shape_list = shape_list;
    }
}
```

同时该文件中设计了Shape类, 从Shape类继承出Line, Rectangle等等, 在类内部实现各类图形和文字及其相关操作。

- Control.java文件

对各类ActionListener接口进行实现。有内容被修改会告知model类, 例如:

```
public class DeleteListener implements ActionListener {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        state.delete();  
        System.out.println("Delete Button");  
    }  
}
```

同时在该文件中设计了State类，用以维护状态机。鼠标选择不同的按钮以及鼠标不同的操作，都是不同的状态。