浙江大学

本科实验报告

课程名称：JAVA应用技术

姓名：卢佳盈

学院：计算机科学与技术学院

专业：计算机科学与技术专业

学号：3180103570

指导老师：翁恺

2020年11月21日

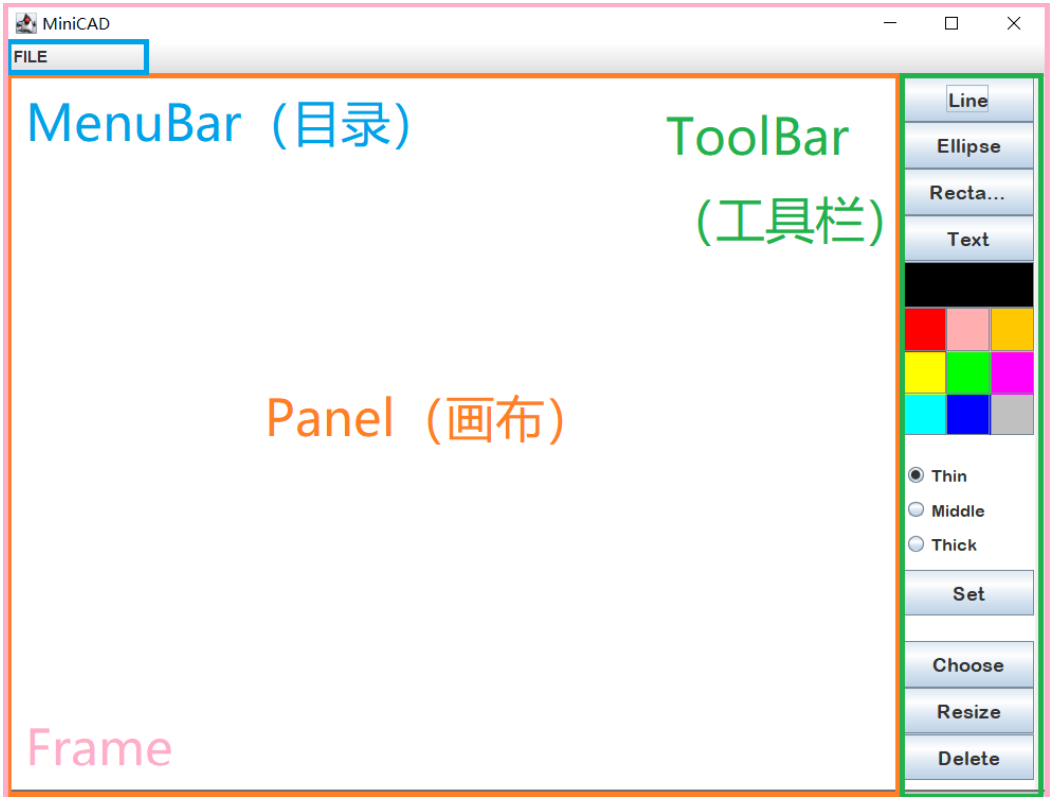
mini CAD

一、实验要求

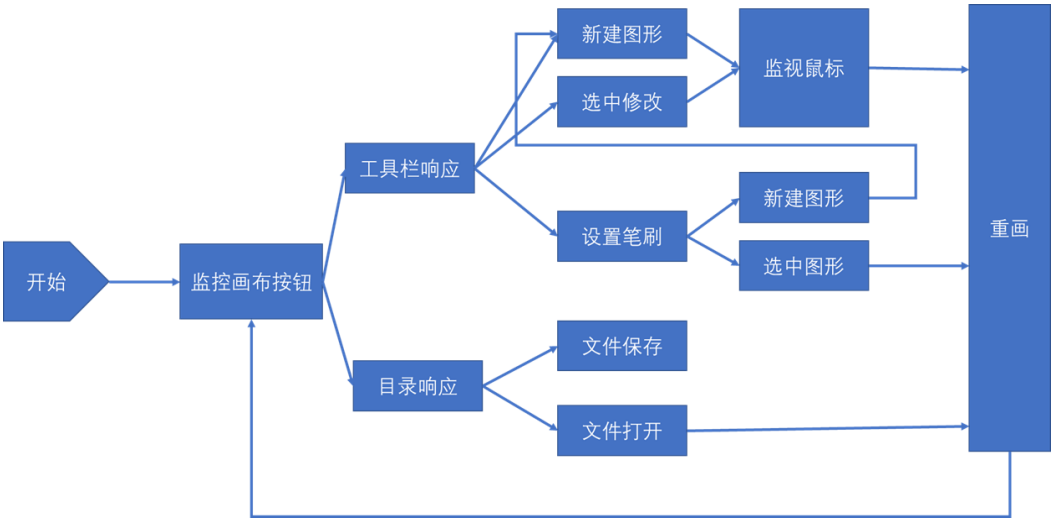
完成一个简单的绘图工具，以CAD的方式操作，能放置直线、矩形、圆和文字，能选中图形，修改参数，如颜色等，能拖动图形和调整大小，可以保存和恢复。

二、实验设计

2.1 UI设计



2.2 业务逻辑



2.3 类功能与类间逻辑

Starter

- 开始程序，新建一个窗口Frame

MyFrame

- 新建画布Panel，工具栏Toolbar，目录Menubar

MyPanel

- 监听画布上的鼠标响应（click,press,release,drag），并将该信号传递给Controller控制器
- 在画布上实时绘制图形

MyTool

- 包括绘画工具（line、ellipse、rectangle、text），编辑工具（choose、resize、delete），笔刷工具（color、width）
- 监听按钮元件，并调用画布的Controller中的对应状态函数

MyMenubar

- 监听save和open两个menuBarItem，执行文件的打开与保存

Controller

- 对鼠标的四种状态函数进行重写，实现绘画、编辑、笔刷的不同功能
- 调用Shape中的对应形状函数，进行paint、repaint与move

Shape

- 将对四种图形在画布上的绘制信息传回给Panel
- 保存图形对应的笔刷参数

##

三、重点代码分析

3.1 图形的绘制

> 以绘制一个ellipse为例

MyPanel.java:对于画布上鼠标状态的监听，同时实现对画布上所有图形的重新绘制

```
public class MyPanel extends JPanel {
    ArrayList<Shape> shapes=new ArrayList<>();

    Shape tmpShape;
    Shape chooseShape;
    int initX,initY;
    int tmpX,tmpY;
    Color brushColor=Color.black;
    float brushWidth=1.0f;
    Controller tmpController=new nopeCtrl();
    public MyPanel() {
        setBackground(Color.WHITE);
        tmpShape=null;
        chooseShape=null;

        addMouseListener(new MouseAdapter() {
            @Override
            public void mousePressed(MouseEvent iEvent) {
                tmpController.mousePress(MyPanel.this, iEvent);
            }
            @Override
            public void mouseReleased(MouseEvent iEvent) {
                tmpController.mouseRelease(MyPanel.this, iEvent);
```

```
            }
            @Override
            public void mouseClicked(MouseEvent iEvent) {
                tmpController.mouseClick(MyPanel.this, iEvent);
            }

        });
        addMouseMotionListener(new MouseMotionAdapter() {
            @Override
            public void mouseDragged(MouseEvent iEvent) {
                tmpController.mouseMove(MyPanel.this, iEvent);
            }
        });
    }

    @Override
    public void paint(Graphics iGraphics) {..}
}
```

MyTool.java:对ui中的按键进行监听

```
public class MyTool extends JPanel{
    JButton btnEllipse = new JButton("Ellipse");
    public MyTool(MyPanel panel){
        btnEllipse.addActionListener(new ActionListener() {
            //画圆
            @Override
            public void actionPerformed(ActionEvent e) {
                MyTool.this.panel.tmpController=new ellipseCtrl();
            }
        });
    }
}
```

Controller.java:重写当前形状为椭圆时，鼠标不同状态所对应的Panel参数

```
public abstract class Controller{
    abstract void mousePress(MyPanel panel,MouseEvent event);
    abstract void mouseRelease(MyPanel panel,MouseEvent event);
    abstract void mouseClick(MyPanel panel,MouseEvent event);
    abstract void mouseMove(MyPanel panel,MouseEvent event);
}

class ellipseCtrl extends Controller{
    @Override
    public void mousePress(MyPanel panel,MouseEvent event) {
        panel.tmpShape=new Ellipse();
        panel.initX=event.getX();
        panel.initY=event.getY();
        panel.tmpShape.setColor(panel.brushColor);
        panel.tmpShape.setWidth(panel.brushWidth);
        panel.tmpShape.initShape(panel.initX, panel.initY);

    }
    @Override
    public void mouseRelease(MyPanel panel,MouseEvent event) {

if(panel.tmpShape.x1==0||panel.tmpShape.x2==0||panel.tmpShape.y1==0||panel.
tmpShape.y2==0) {
            panel.tmpShape=null;
        }
        else {
            panel.shapes.add(panel.tmpShape);
```

```
            panel.tmpShape=null;
        }
    }
    @Override
    public void mouseClick(MyPanel panel,MouseEvent event) {

    }
    @Override
    public void mouseMove(MyPanel panel,MouseEvent event) {
        panel.repaint();
        panel.tmpX=event.getX();
        panel.tmpY=event.getY();
        panel.repaint();
    }
}

```

Shape.java:设置笔刷参数，在graphics上新建/修改目标图形

```
public abstract class Shape implements Serializable{
    int x1,y1,x2,y2;
    float brushWidth;
    Color brushColor;
    public void setColor(Color targetColor) {
        this.brushColor=targetColor;
    }
    public void setWidth(Float targetWidthFloat) {
        this.brushWidth=targetWidthFloat;
    }
    abstract void draw(int tx2,int ty2,Graphics G);
    abstract void draw(Graphics G);
    abstract void choose(Graphics G);
    abstract boolean isChosen(int choseX,int choseY);

    public void initShape(int tx1,int ty1) {
        this.x1=tx1;
        this.y1=ty1;
    }
}
class Ellipse extends Shape{
    @Override
    public void draw(Graphics G) {
        Graphics2D iGraphics2d=(Graphics2D)G;
        Stroke iStroke=new
BasicStroke(brushWidth,BasicStroke.CAP_BUTT,BasicStroke.JOIN_ROUND,3.5f,new
float[] {1,0,},0f);
        iGraphics2d.setStroke(iStroke);
        iGraphics2d.setColor(brushColor);
        iGraphics2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
        iGraphics2d.drawOval(Math.min(x1, x2), Math.min(y1,
y2),Math.abs(x1-x2),Math.abs(y1-y2));
    }

    @Override
    public void draw(int tx2,int ty2,Graphics G) {
        this.x2=tx2;
        this.y2=ty2;
        Graphics2D iGraphics2d=(Graphics2D)G;
        Stroke iStroke=new
BasicStroke(brushWidth,BasicStroke.CAP_BUTT,BasicStroke.JOIN_ROUND,3.5f,new
float[] {1,0,},0f);
        iGraphics2d.setStroke(iStroke);
        iGraphics2d.setColor(brushColor);
```

```
        iGraphics2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
        iGraphics2d.drawOval(Math.min(x1, x2), Math.min(y1,
y2),Math.abs(x1-x2),Math.abs(y1-y2));
    }
    @Override
    void choose(Graphics G) {
        Graphics2D iGraphics2d=(Graphics2D)G;
        Stroke iStroke=new
BasicStroke(brushWidth*2,BasicStroke.CAP_BUTT,BasicStroke.JOIN_ROUND,0.5f,n
ew float[] {15,10,},0f);
        iGraphics2d.setStroke(iStroke);
        iGraphics2d.setColor(Color.red);
        iGraphics2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
        iGraphics2d.drawRect(Math.min(x1, x2), Math.min(y1, y2),
Math.abs(x1-x2), Math.abs(y1-y2));
    }
    @Override
    boolean isChosen(int choseX,int choseY) {
        boolean result;
        if(choseX<=Math.max(x1, x2)&&choseX>=Math.min(x1,
x2)&&choseY<=Math.max(y1, y2)&&choseY>=Math.min(y1, y2)) {
            return true;
        }
        else {
            return false;
        }
    }
}
```

3.2 图形的选中

采用了如下算法判断目前鼠标的click点是否对应图形、对应的是哪个图形（Controller.java）

1. 对画布上的形状列表进行遍历
2. 如果判断鼠标点在该形状的范围内，则将目标形状设置为该形状
3. 当两个或两个以上的形状范围都包含了鼠标点，则选中最近添加的形状

```
Shape targetShape=null;
for(Shape iShape:panel.shapes) {
    if(iShape.isChosen(event.getX(), event.getY())) {
        targetShape=iShape;
    }
}
if(targetShape!=null) {
    panel.chooseShape=targetShape;
    hasShape=true;
}else {
    panel.chooseShape=null;
}
```

3.3 文件的操作

打开文件：将目标文件读入画布的shape列表，并重绘画布

```
openBtn.addActionListener(new ActionListener() {
    //打开
    @Override
    public void actionPerformed(ActionEvent e) {
        JFileChooser openChooser=new JFileChooser("D://");
        int getValue=openChooser.showOpenDialog(openChooser);
```

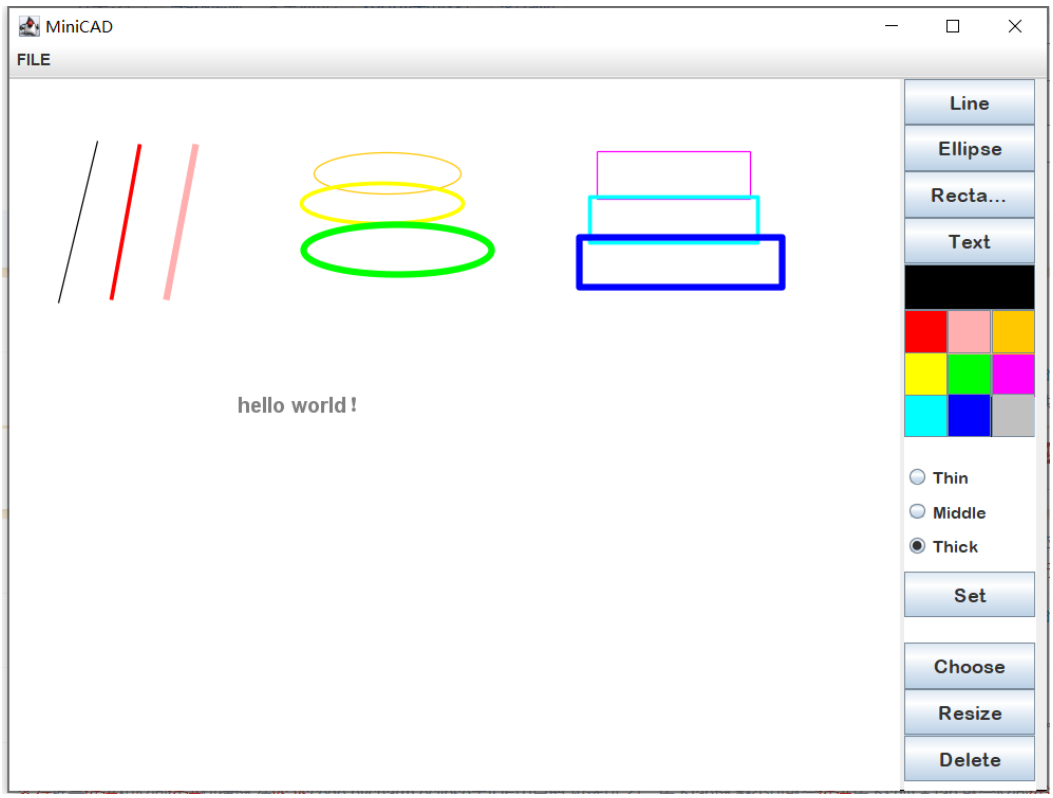
```
        String fileNameString="";
        if(getValue==JFileChooser.APPROVE_OPTION) {
            fileNameString =
openChooser.getSelectedFile().getAbsolutePath();
        }
        if(!fileNameString.isEmpty()) {
            try {
                ObjectInputStream inputShapeStream=new
ObjectInputStream(new FileInputStream(fileNameString));
                panel.shapes=
(ArrayList<Shape>)inputShapeStream.readObject();
                panel.repaint();
            }
            catch (Exception error) {
                error.printStackTrace();
            }
        }
    }
});
```

保存操作：将画布中shape列表的所有元素以二进制形式写入目标文件

```
saveBtn.addActionListener(new ActionListener() {
    //保存
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        JFileChooser saveChooser=new JFileChooser("D://");
        saveChooser.showSaveDialog(null);
        File file=saveChooser.getSelectedFile();
        String fileNameString=file.getPath();
        if(!fileNameString.isEmpty()) {
            try {
                ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(fileNameString));
                out.writeObject(panel.shapes);
                out.close();
            }
            catch (Exception error) {
                error.printStackTrace();
            }
        }
    }
});
```

四、实验结果

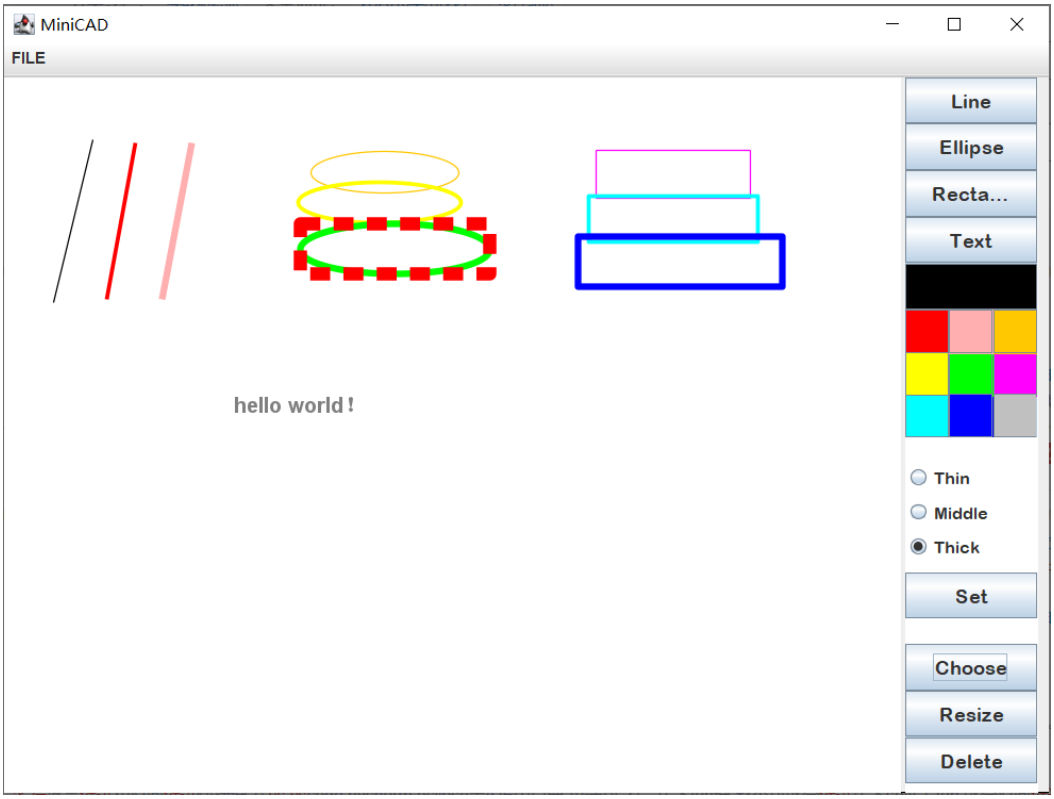
4.1 图形新建



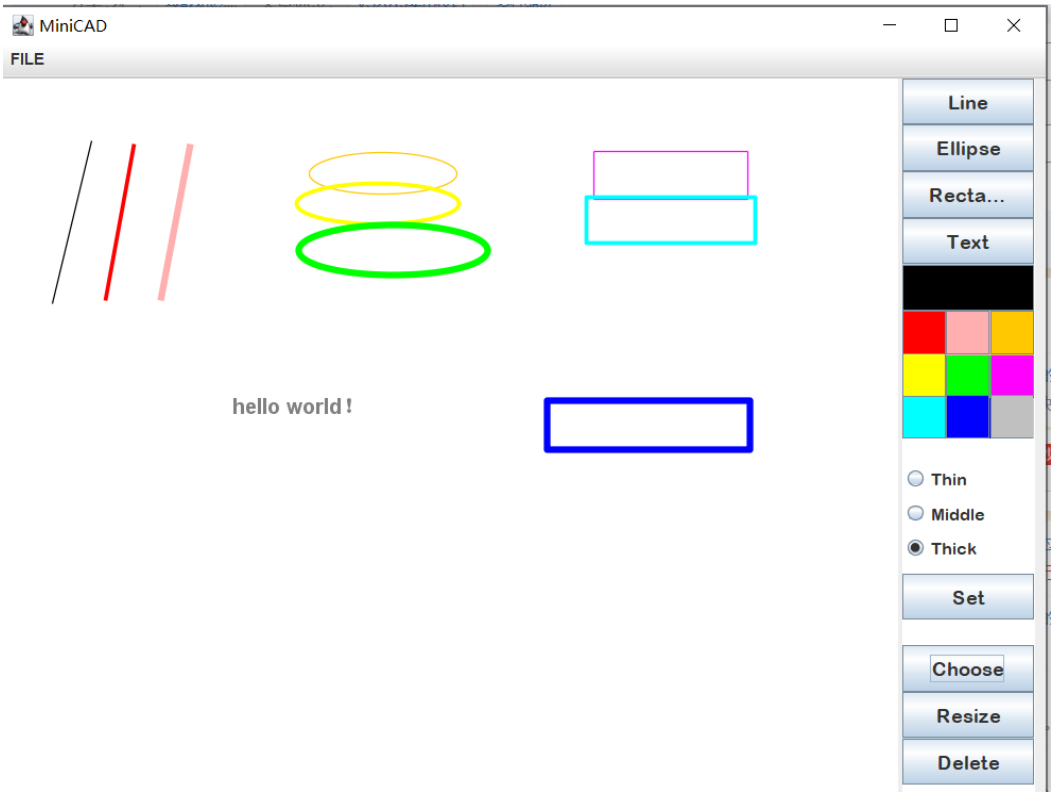
##

4.2 选中、移动、删除

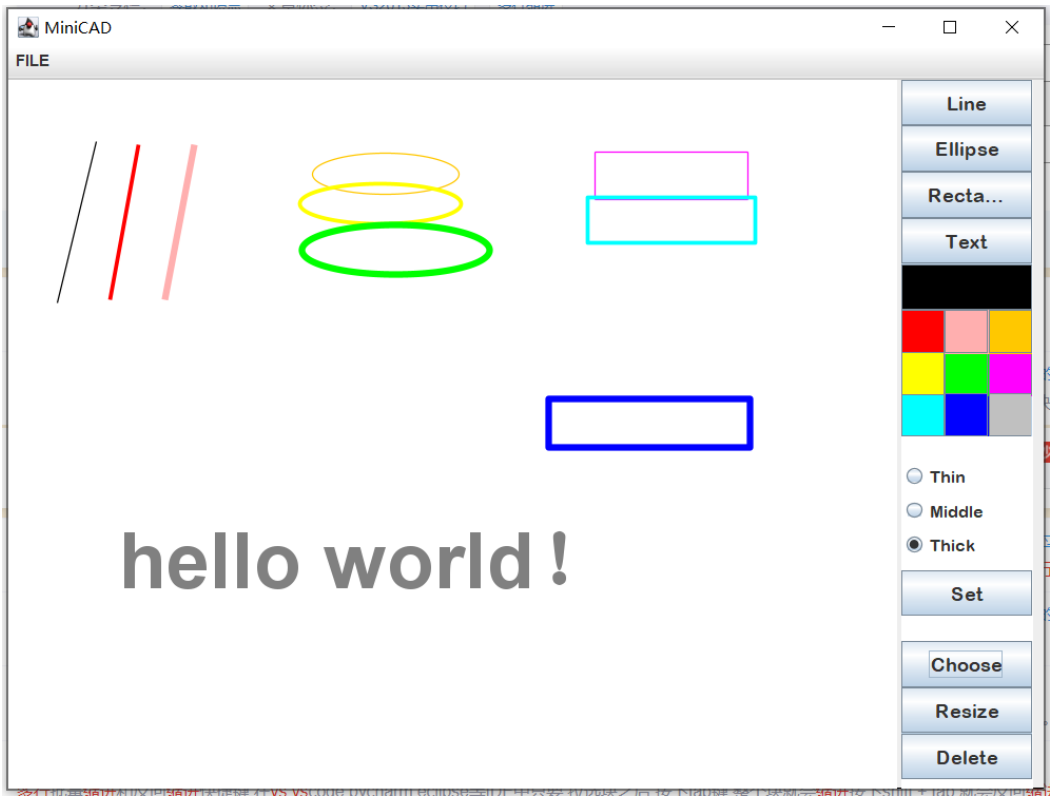
选中一个椭圆：点击choose后，选择图形



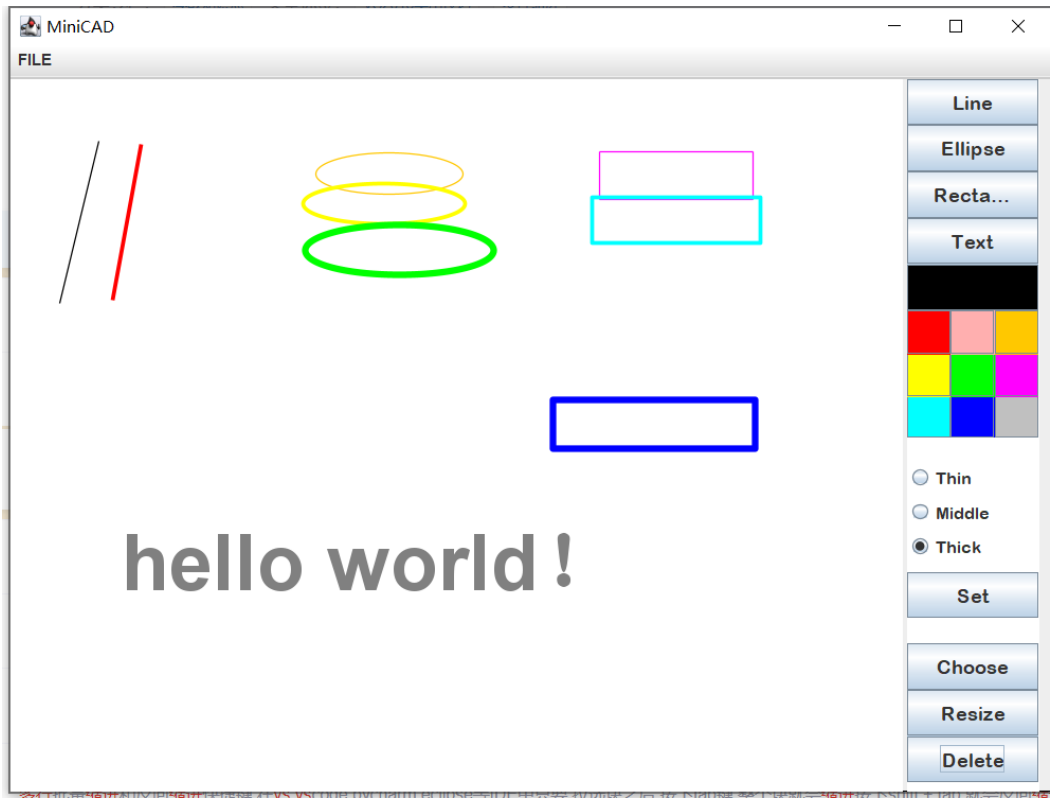
移动一个矩形：点击choose后进行移动



调整文本大小：点击resize后选择一个图形，进行大小重绘

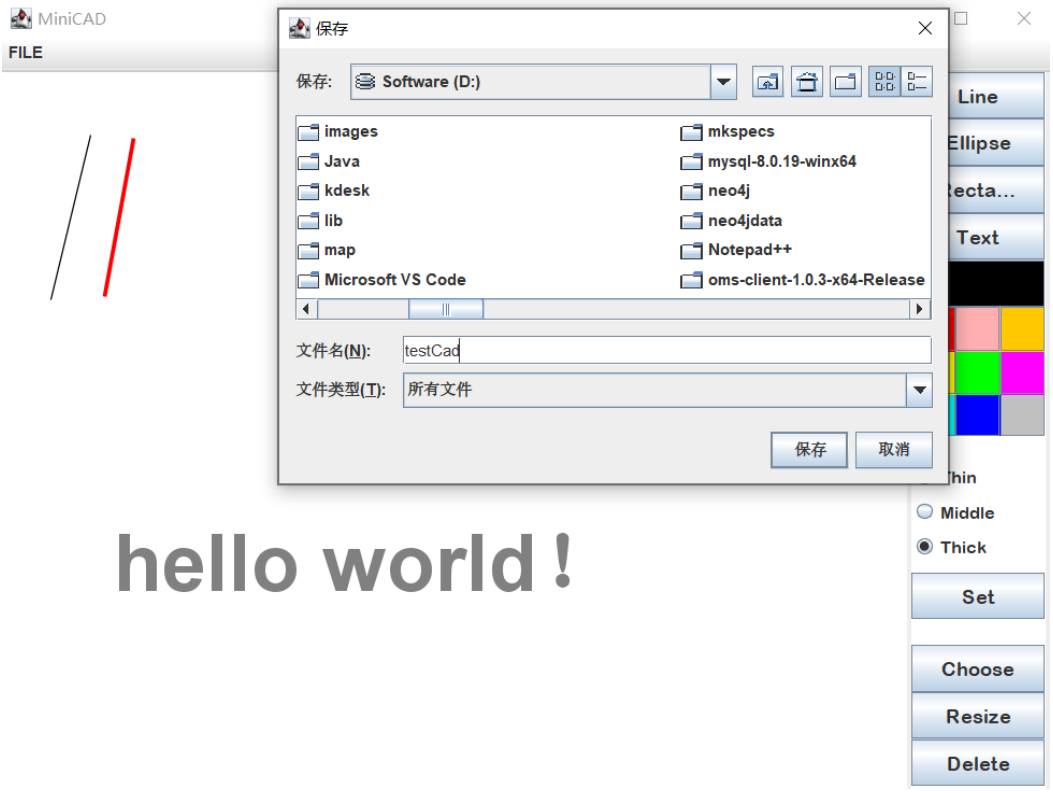


删除一条直线：点击choose后，选中想删除的图像，再点击delete

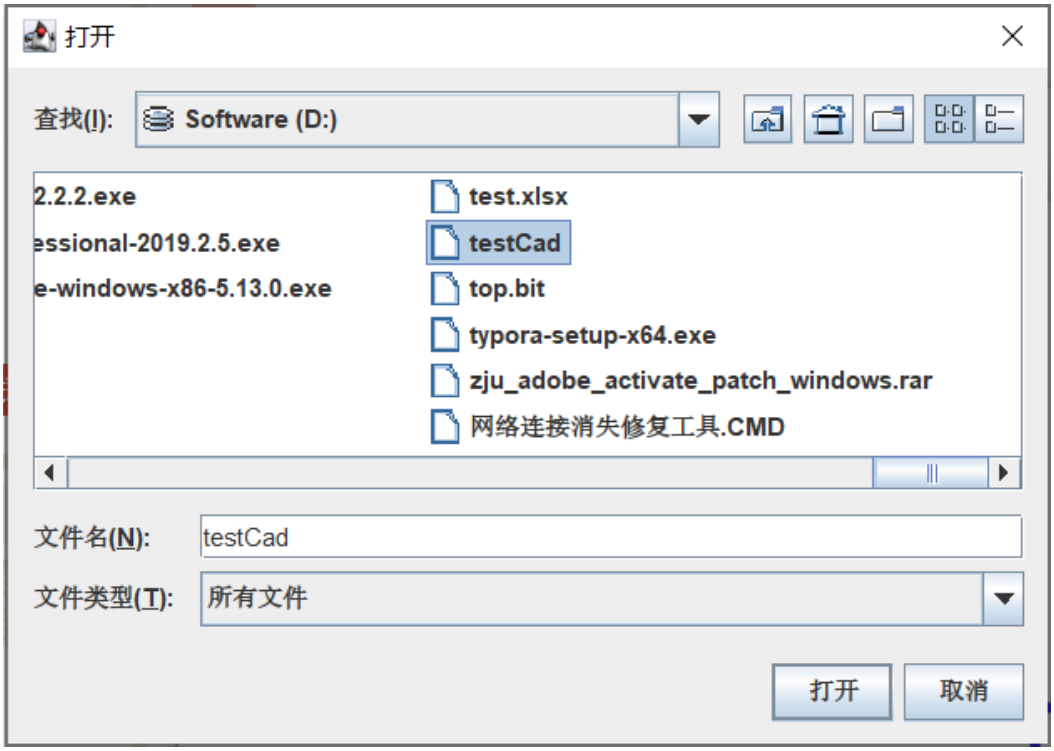


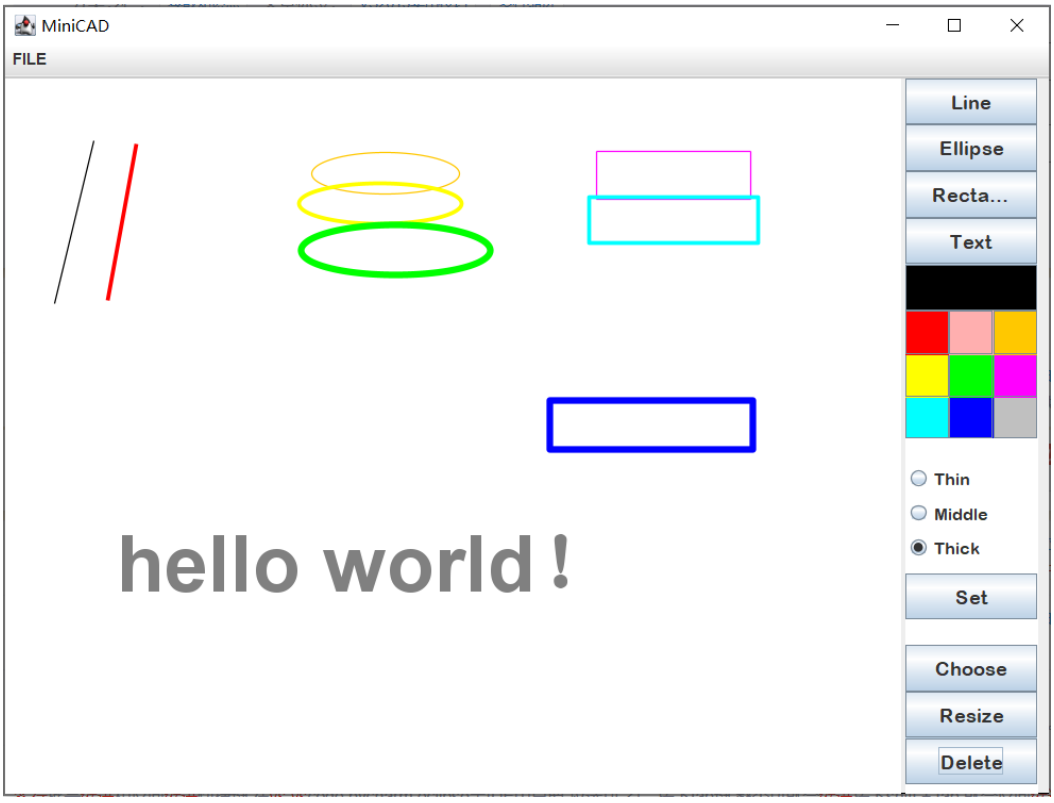
4.3 文件保存、打开

文件保存：



文件重新打开：


打开
查找(I):
Software (D:)
2.2.2.exe
essional-2019.2.5.exe
e-windows-x86-5.13.0.exe
test.xlsx
testCad
top.bit
typora-setup-x64.exe
zju_adobe_activate_patch_windows.rar
网络连接消失修复工具.CMD
文件名(N):
testCad
文件类型(T):
所有文件
打开
取消


MiniCAD
FILE
Line
Ellipse
Recta...
Text
Thin
Middle
Thick
Set
Choose
Resize
Delete
hello world !

##