

# **miniCAD**

2018.12

3160103113 曾显珺

miniCAD 基本功能：

- 绘制矩形，圆形，直线等图形

Shapes.java:

Interface Shapes

```
public abstract class Shapes {
    public Color color;
    public Point p1;
    public Point p2;
    public float fine;
    Shapes(Point p1,Point p2)
    {
        this.p1=new Point(p1.x,p1.y);
        this.p2=new Point(p2.x,p2.y);
        this.fine = 3;
        this.color = Color.BLACK;
    }

    public void setColor(Color c) {
        this.color = c;
    }

    public void setLine(float fine , Graphics g ) {
        Graphics2D g2d=(Graphics2D) g;
        g2d.setStroke(new BasicStroke(fine));
    }
    public abstract void draw(Graphics graphics);
    public abstract boolean isSelected(Point p);
}

圆 Circle.class 选中判定：再圆上或者圆内选中点即选中图案
class Circle extends Shapes //圆
{
    Circle(Point p1,Point p2)
    {
        super(p1, p2);
    }

    public void draw(Graphics g)
    {
        setLine(fine,g);
        g.setColor(color);
        int d=Math.abs(p1.x-p2.x);
        Point[] points=rectify();
        g.drawOval(points[0].x,points[0].y,d,d);
    }
}
```

```

public boolean isSelected(Point p)
{
    Point[] points=rectify();
    return
p.x>points[0].x&&p.x<points[1].x&&p.y>points[0].y&&p.y<points[1].y;
}
}

矩形 Square.class 选中判定：在边框或者矩形内部选中点即选中图案
class Square extends Shapes //矩形
{
    Square(Point p1,Point p2)
    {
        super(p1, p2);
    }

    public void draw(Graphics g)
    {
        setLine(fine,g);
        g.setColor(color);
        Point[] points=rectify();
        g.drawRect(points[0].x,points[0].y,points[1].x-
points[0].x,points[1].y-points[0].y);
    }
}

public boolean isSelected(Point p)
{
    Point[] points=rectify();
    return
p.x>points[0].x&&p.x<points[1].x&&p.y>points[0].y&&p.y<points[1].y;
}
}

线 Line.class
class Line extends Shapes //线条
{
    Line(Point p1,Point p2)
    {
        super(p1, p2);
    }

    public void draw(Graphics g)
    {
        setLine(fine,g);
    }
}

```

```

        g.setColor(color);
        g.drawLine(p1.x,p1.y,p2.x,p2.y);
    }

    public boolean isSelected(Point p)
    {
        if(p1.x>p2.x&&p1.y>p2.y)
        {
            return
(p.x>p2.x&&p.x<(p1.x+p2.x)/2&&p.y>p2.y&&p.y<(p1.y+p2.y)/2)||

(p.x<p1.x&&p.x>(p1.x+p2.x)/2&&p.y<p1.y&&p.y>(p1.y+p2.y)/2);

        }
        else if(p2.x>p1.x&&p2.y>p1.y)
        {
            return
(p.x>p1.x&&p.x<(p1.x+p2.x)/2&&p.y>p1.y&&p.y<(p1.y+p2.y)/2)||

(p.x<p2.x&&p.x>(p1.x+p2.x)/2&&p.y<p2.y&&p.y>(p1.y+p2.y)/2);

        }
        else if(p2.x>p1.x&&p2.y<p1.y)
        {
            return
(p.x>p1.x&&p.x<(p1.x+p2.x)/2&&p.y>p2.y&&p.y<(p1.y+p2.y)/2)||

(p.x<p2.x&&p.x>(p1.x+p2.x)/2&&p.y>p2.y&&p.y<(p1.y+p2.y)/2);

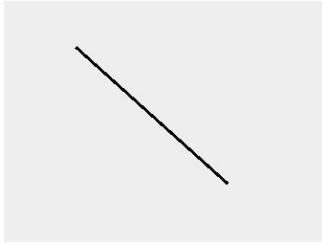
        }
        else
        {
            return
(p.x>p2.x&&p.x<(p1.x+p2.x)/2&&p.y<p2.y&&p.y>(p1.y+p2.y)/2)||

(p.x<p1.x&&p.x>(p1.x+p2.x)/2&&p.y>p1.y&&p.y<(p1.y+p2.y)/2);

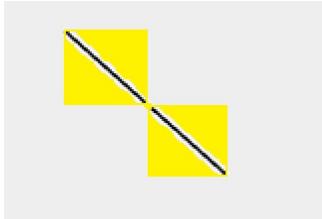
        }
    }
}

```

- 能选中并拖动图案  
Line 的选中判定



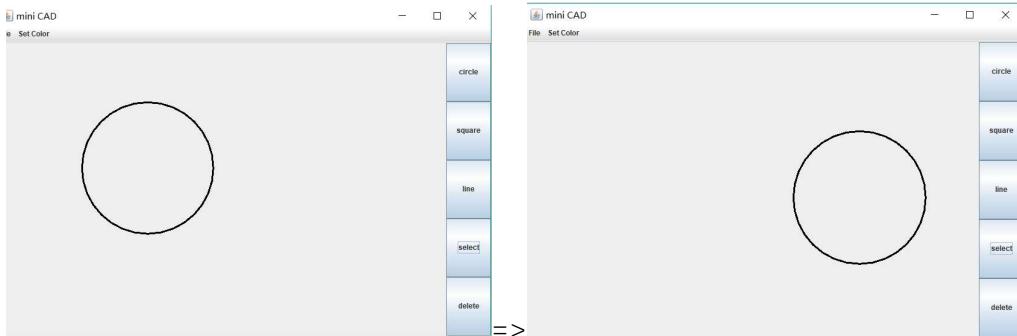
在线条头尾两端构成的矩形范围内，有一半的选中区域



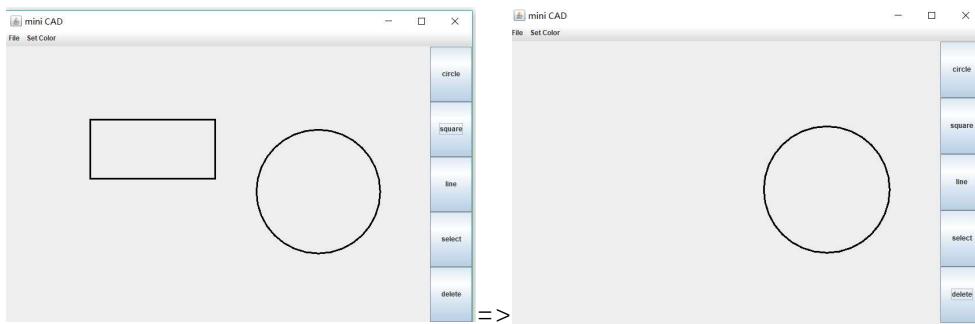
黄色部分为可选中区域

圆和矩形：在外接矩形边缘或者内部都可以选中

选中后可以拖动

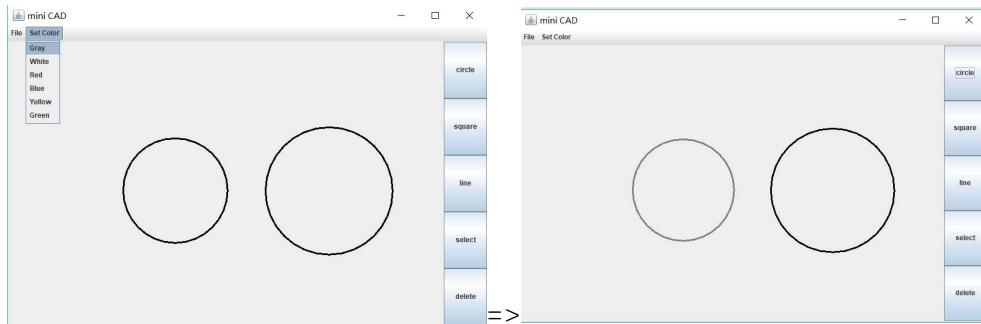


点击 delete 在点击图形区域可以删除该图形



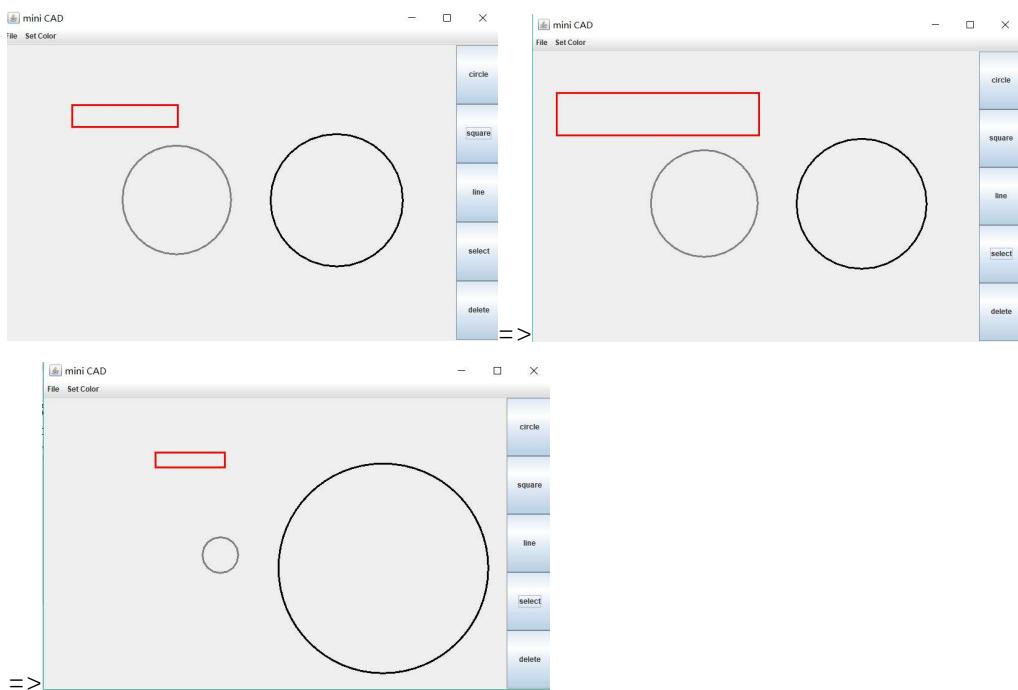
- 能改变图案颜色，大小

改变颜色【只能对新建的图案（最后一个添加的图案）修改颜色】

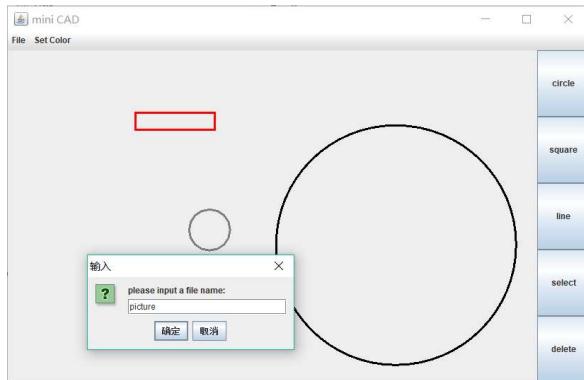


改变大小：可以对选中的图案改变大小（不改变中心位置和粗细）

上滚轮为放大（ $\times 1.25$  倍），下滚轮为缩小( $\times 0.8$  倍)



- 能导入及导出图案文件  
File->Save 导出文件（默认后缀.cad）



File->Open 打开文件  
【在打开时出错，通过修改 Shapes 类为可串行化对象解决】

