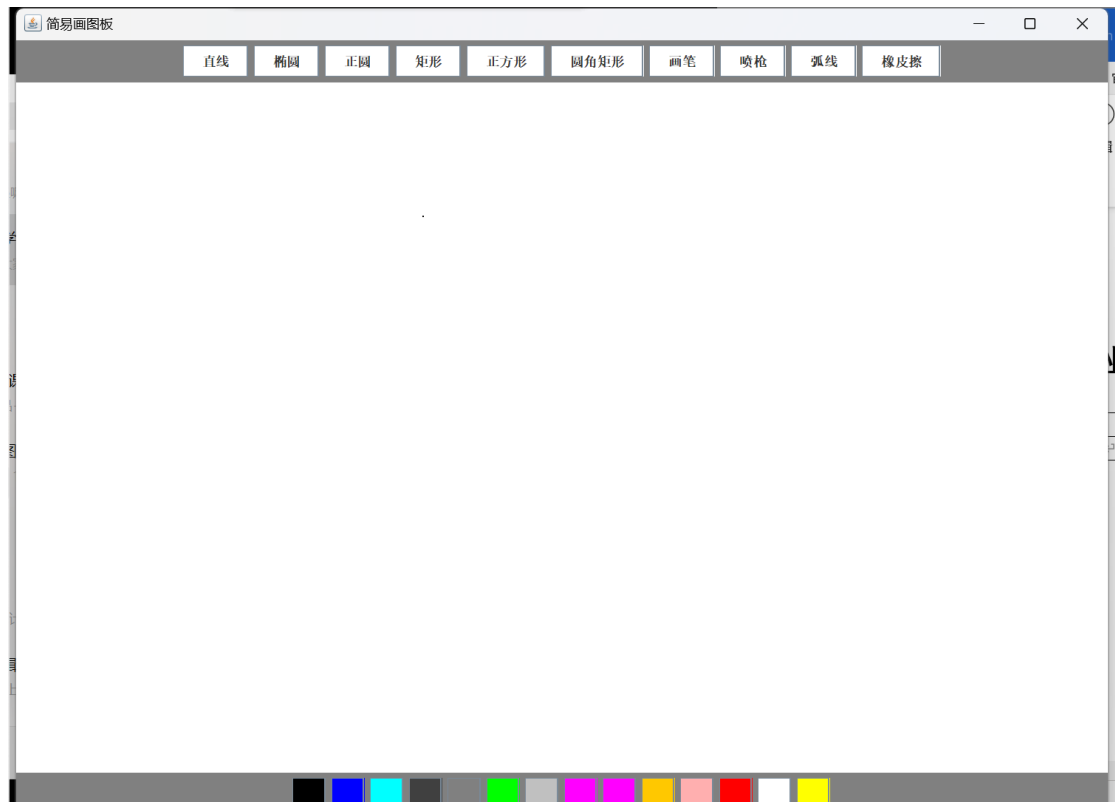
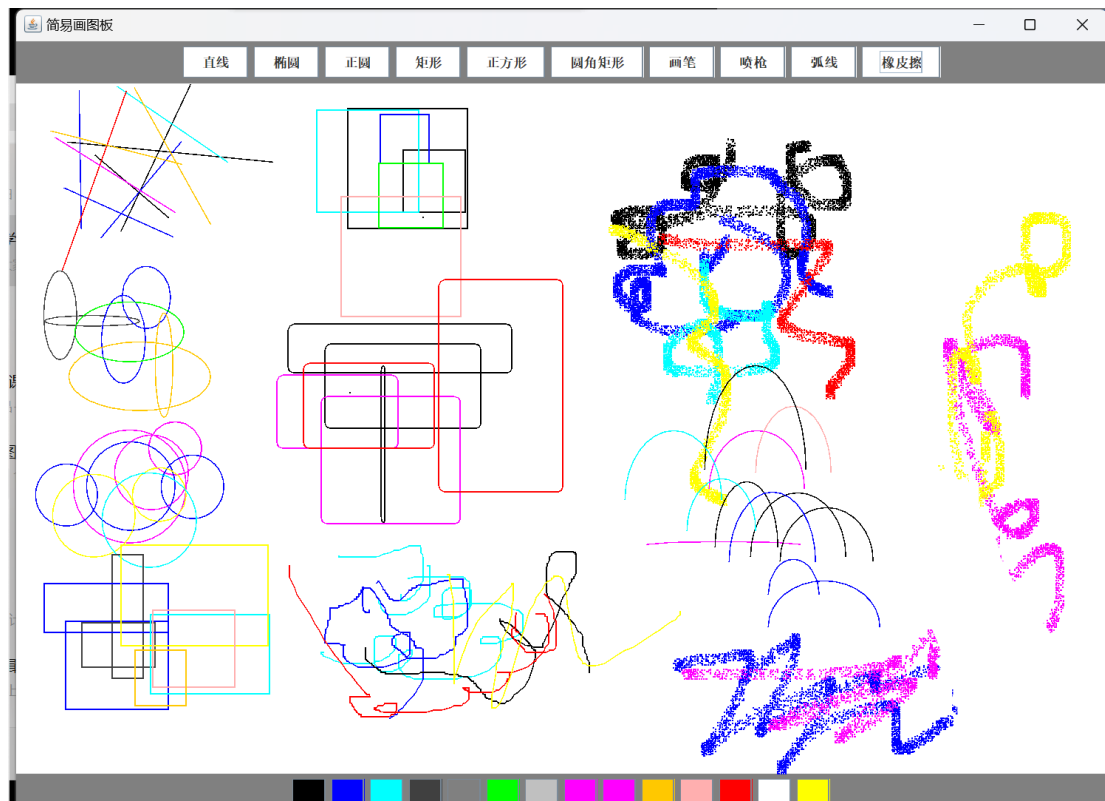


计算机图形学作业

| | | | |
|-----|-----------|-----|-------------|
| 姓名： | 孙涵 | 学号： | 10215102481 |
| 内容： | 计算机图形学大作业 | 日期： | 2022/10/20 |

效果展示





代码部分

1. 构建窗体,按钮,面板

```
1. import javax.swing.*;
2. import java.awt.*;
3. import java.awt.event.KeyEvent;
4. import java.awt.event.KeyListener;
5.
6. public class Prepare {
7.     JFrame jf;
8.     JPanel jp1;
9.     JPanel jp2;
10.    JPanel jp3;
11.    JButton [] drawButton;
12.    JButton [] colorButton;
13.    Graphics g;
14.    ButtonListener buttonListener;
15.    DrawListener drawListener;
16.
17.    public Prepare(JFrame jf, JPanel jp1, JPanel jp2, JPanel jp3, JButton[] dr
awButton, JButton[] colorButton, ButtonListener buttonListener, DrawListener
drawListener) {
```

```
18.         this.jf = jf;
19.         this.jp1 = jp1;
20.         this.jp2 = jp2;
21.         this.jp3 = jp3;
22.         this.drawButton = drawButton;
23.         this.colorButton = colorButton;
24.         this.buttonListener = buttonListener;
25.         this.drawListener = drawListener;
26.
27.     }
28.
29.     public void prepareJf(){
30.         jf.setVisible(true);
31.         jf.setTitle("简易画图板");
32.         jf.setSize(1000,1000);
33.         jf.setLayout(new BorderLayout());
34.         jf.setDefaultCloseOperation(3);
35.         jf.addMouseListener(drawListener);
36.     }
37.
38.     public void preparePanel(){
39.         jp1.setBackground(Color.BLACK);
40.         jp1.setLayout(new FlowLayout(FlowLayout.CENTER));
41.         jp1.setBackground(Color.GRAY);
42.         jf.add(jp1,BorderLayout.NORTH);
43.
44.         jp2.setBackground(Color.WHITE);
45.         jp2.addMouseListener(drawListener);
46.         jp2.addMouseMotionListener(drawListener);
47.         jp2.setVisible(true);
48.         g = jp2.getGraphics();
49.         drawListener.g = g;
50.         jf.add(jp2,BorderLayout.CENTER);
51.
52.         jp3.setLayout(new FlowLayout((FlowLayout.CENTER)));
53.         jp3.setBackground(Color.black);
54.         jp3.setLayout(new FlowLayout(FlowLayout.CENTER));
55.         jp3.setBackground(Color.gray);
56.         jf.add(jp3,BorderLayout.SOUTH);
57.     }
58.
59.     public void prepareButton(){
60.         String [] shape = {"直线", "椭圆", "正圆", "矩形", "正方形", "圆角矩形", "画笔",
            "喷枪", "弧线", "橡皮擦"};
```

```

61.         for(int i = 0; i < shape.length; i++){
62.             drawButton[i] = new JButton(shape[i]);
63.             drawButton[i].setBackground(Color.WHITE);
64.             drawButton[i].addActionListener(buttonListener);
65.             jp1.add(drawButton[i]);
66.         }
67.
68.
69.         Color[] colorList = {Color.BLACK,Color.BLUE,Color.CYAN,Color.DARK_GRA
Y,Color.GRAY,Color.GREEN,Color.LIGHT_GRAY,Color.MAGENTA,Color.magenta,Color.O
RANGE,Color.PINK,Color.RED,Color.WHITE,Color.YELLOW};
70.         for(int i = 0; i < colorList.length; i++){
71.             colorButton[i] = new JButton();
72.             colorButton[i].setBackground(colorList[i]);
73.             colorButton[i].setPreferredSize(new Dimension(30,30));
74.             colorButton[i].addActionListener(buttonListener);
75.             jp3.add(colorButton[i]);
76.         }
77.
78.     }
79.
80.     public void prepareDraw(){
81.         jp2.setVisible(true);
82.         g = jp2.getGraphics();
83.         drawListener.g = g;
84.     }
85.
86.
87. }

```

2.创建监听器(按钮监听器)

```

1. import javax.swing.*;
2. import java.awt.*;
3. import java.awt.event.ActionEvent;
4. import java.awt.event.ActionListener;
5.
6. //获取形状和颜色
7.
8. public class ButtonListener implements ActionListener {
9.     Color color;
10.    String shape;
11.    DrawListener drawListener;
12.
13.    public ButtonListener(DrawListener drawListener) {

```

```

14.         this.drawListener = drawListener;
15.     }
16.
17.     @Override
18.     public void actionPerformed(ActionEvent actionEvent) {
19.         if(actionEvent.getActionCommand().equals("")){
20.             JButton btn = (JButton) actionEvent.getSource();
21.             color = btn.getBackground();
22.             drawListener.setColor(color);
23.             System.out.println(color.toString());
24.         }else {
25.             JButton btn = (JButton) actionEvent.getSource();
26.             shape = btn.getActionCommand();
27.             drawListener.setShape(shape);
28.             System.out.println(shape.toString());
29.         }
30.     }
31.
32.     public Color getColor() {
33.         return color;
34.     }
35.
36.     public String getShape() {
37.         return shape;
38.     }
39.
40.
41. }

```

3.创建监听器(画板监听器)

```

1. import javax.swing.*;
2. import java.awt.*;
3. import java.awt.event.MouseEvent;
4. import java.awt.event.MouseListener;
5. import java.awt.event.MouseMotionListener;
6. import java.util.Random;
7.
8.
9. public class DrawListener extends JFrame implements MouseListener, MouseMotion
    Listener {
10.
11.
12.     Color color = Color.black;

```

```
13.     String shape = "直线";
14.     Graphics g;
15.     JPanel jp2;
16.     int x0,x1,x2,y0,y1,y2,newx1,newy1,newx2,newy2;
17.     boolean flag = true;
18.
19.
20.
21.
22.     @Override
23.     public void mouseClicked(MouseEvent e) {
24.     }
25.
26.
27.     @Override
28.     public void mousePressed(MouseEvent e) {
29.         Graphics2D g2 = (Graphics2D) g;
30.         g2.setStroke(new BasicStroke(1));
31.         System.out.println("mousePressed");
32.         x1 = e.getX();
33.         y1 = e.getY();
34.     }
35.
36.
37.     @Override
38.     public void mouseReleased(MouseEvent e) {
39.         Graphics2D g2 = (Graphics2D) g;
40.         g2.setStroke(new BasicStroke(1));
41.         System.out.println("mouseReleased");
42.         x2 = e.getX();
43.         y2 = e.getY();
44.         int w = x2-x1;
45.         int h = y2-y1;
46.
47.         //绘制图形
48.         if (shape.equals("直线")){
49.             System.out.println("2");
50.             g.drawLine(x1,y1,x2,y2);
51.         }else if (shape.equals("矩形")){
52.             g.drawRect(Math.min(x1,x2),Math.min(y1,y2),Math.abs(w),Math.abs(h)
53.         );
54.         }else if (shape.equals("椭圆")){
55.             g.drawOval(Math.min(x1,x2),Math.min(y1,y2),Math.abs(w),Math.abs(h)
56.         );
57.         }
```

```

55.         }else if(shape.equals("弧线")){
56.             g.drawArc(Math.min(x1,x2), Math.min(y1,y2), Math.abs(x2-
           x1), Math.abs(y2-y1), 0, 180);
57.         }else if (shape.equals("正圆")){
58.             g.drawOval(Math.min(x1,x2),Math.min(y1,y2),Math.abs(w),Math.abs(w)
           );
59.         }else if (shape.equals("正方形")){
60.             g.drawRect(Math.min(x1,x2),Math.min(y1,y2),Math.abs(w),Math.abs(w)
           );
61.         }else if(shape.equals("圆角矩形")){
62.             g.drawRoundRect(Math.min(x1,x2),Math.min(y1,y2),Math.abs(w),Math.a
           bs(h),10,10);
63.         }
64.     }
65.
66.
67.     @Override
68.     public void mouseDragged(MouseEvent e) {
69.         Graphics2D g2 = (Graphics2D) g;
70.         g2.setStroke(new BasicStroke(1));
71.         System.out.println("mouseDragged");
72.         x2 = e.getX();
73.         y2 = e.getY();
74.         if (shape.equals("画笔")) {
75.             g.drawLine(x1, y1, x2, y2);
76.             x1 = x2;
77.             y1 = y2;
78.         }else if(shape.equals("橡皮擦")){
79.             g2.setStroke(new BasicStroke(80));
80.             g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHint
           s.VALUE_ANTIALIAS_ON);
81.             g.setColor(Color.WHITE);
82.             g.drawLine(x1, y1, x2, y2);
83.             x1 = x2;
84.             y1 = y2;
85.         }else if(shape.equals("喷枪")){
86.             for(int k=0;k<20;k++){
87.                 Random i=new Random();
88.                 int a=i.nextInt(8);
89.                 int b=i.nextInt(10);
90.                 g.drawLine(x2+a, y2+b, x2+a, y2+b);
91.             }
92.         }
93.     }

```

```

94.
95.
96.     @Override
97.     public void mouseMoved(MouseEvent mouseEvent) {
98.
99.     }
100.
101.     @Override
102.     public void mouseEntered(MouseEvent mouseEvent) {
103.
104.     }
105.
106.     @Override
107.     public void mouseExited(MouseEvent mouseEvent) {
108.
109.     }
110.
111.     public void setColor(Color color) {
112.         this.color = color;
113.         g.setColor(color);
114.     }
115.
116.     public void setShape(String shape) {
117.         Graphics2D g2 = (Graphics2D) g;
118.         if(this.shape.equals("橡皮擦")){
119.             g.setColor(Color.black);
120.         }else {
121.             this.shape = shape;
122.         }
123.     }
124.
125. }

```

4.打包放进 main 函数

```

1. import javax.swing.*;
2. import java.awt.*;
3. import java.awt.event.KeyEvent;
4. import java.awt.event.KeyListener;
5.
6. public class Prepare {
7.     JFrame jf;
8.     JPanel jp1;
9.     JPanel jp2;
10.    JPanel jp3;

```



```
11.     JButton [] drawButton;
12.     JButton [] colorButton;
13.     Graphics g;
14.     ButtonListener buttonListener;
15.     DrawListener drawListener;
16.
17.     public Prepare(JFrame jf, JPanel jp1, JPanel jp2, JPanel jp3, JButton[] drawButton, JButton[] colorButton, ButtonListener buttonListener, DrawListener drawListener) {
18.         this.jf = jf;
19.         this.jp1 = jp1;
20.         this.jp2 = jp2;
21.         this.jp3 = jp3;
22.         this.drawButton = drawButton;
23.         this.colorButton = colorButton;
24.         this.buttonListener = buttonListener;
25.         this.drawListener = drawListener;
26.
27.     }
28.
29.     public void prepareJf(){
30.         jf.setVisible(true);
31.         jf.setTitle("简易画图板");
32.         jf.setSize(1000,1000);
33.         jf.setLayout(new BorderLayout());
34.         jf.setDefaultCloseOperation(3);
35.         jf.addMouseListener(drawListener);
36.     }
37.
38.     public void preparePanel(){
39.         jp1.setBackground(Color.BLACK);
40.         jp1.setLayout(new FlowLayout(FlowLayout.CENTER));
41.         jp1.setBackground(Color.GRAY);
42.         jf.add(jp1, BorderLayout.NORTH);
43.
44.         jp2.setBackground(Color.WHITE);
45.         jp2.addMouseListener(drawListener);
46.         jp2.addMouseMotionListener(drawListener);
47.         jp2.setVisible(true);
48.         g = jp2.getGraphics();
49.         drawListener.g = g;
50.         jf.add(jp2, BorderLayout.CENTER);
51.
52.         jp3.setLayout(new FlowLayout((FlowLayout.CENTER)));
```

```
53.         jp3.setBackground(Color.black);
54.         jp3.setLayout(new FlowLayout(FlowLayout.CENTER));
55.         jp3.setBackground(Color.gray);
56.         jf.add(jp3, BorderLayout.SOUTH);
57.     }
58.
59.     public void prepareButton(){
60.         String [] shape = {"直线", "椭圆", "正圆", "矩形", "正方形", "圆角矩形", "画笔",
        "喷枪", "弧线", "橡皮擦"};
61.         for(int i = 0; i < shape.length; i++){
62.             drawButton[i] = new JButton(shape[i]);
63.             drawButton[i].setBackground(Color.WHITE);
64.             drawButton[i].addActionListener(buttonListener);
65.             jp1.add(drawButton[i]);
66.         }
67.
68.
69.         Color[] colorList = {Color.BLACK, Color.BLUE, Color.CYAN, Color.DARK_GRAY,
        Color.GRAY, Color.GREEN, Color.LIGHT_GRAY, Color.MAGENTA, Color.magenta, Color.ORANGE, Color.PINK, Color.RED, Color.WHITE, Color.YELLOW};
70.         for(int i = 0; i < colorList.length; i++){
71.             colorButton[i] = new JButton();
72.             colorButton[i].setBackground(colorList[i]);
73.             colorButton[i].setPreferredSize(new Dimension(30, 30));
74.             colorButton[i].addActionListener(buttonListener);
75.             jp3.add(colorButton[i]);
76.         }
77.
78.     }
79.
80.     public void prepareDraw(){
81.         jp2.setVisible(true);
82.         g = jp2.getGraphics();
83.         drawListener.g = g;
84.     }
85.
86.
87. }
```