C:\Documents and Settings\Brian Cullinan\My Documents\My Homework\CS 477\Project 2\src\project2\Main.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package project2;

/**
 *
 * @author Brian Cullinan
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        new Application();
    }

}
```

C:\Documents and Settings\Brian Cullinan\My Documents\My Homework\CS 477\Project 2\src\project2\Application.java

```
package project2;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */



import javax.swing.*;
import java.awt.*;
import javax.swing.event.*;
import java.awt.event.*;

/**
 *
 * @author Brian Cullinan
 */
public class Application extends JFrame {

    public Application()
    {
        super();
        setTitle("Easy Paint");
        setSize(640, 480);
        setExtendedState(JFrame.MAXIMIZED_BOTH);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        new Window(this);
    }
}
```

C:\Documents and Settings\Brian Cullinan\My Documents\My Homework\CS 477\Project 2\src\project2 \Picture.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package project2;

import javax.swing.*;
import java.awt.*;
import javax.swing.event.*;
import java.awt.event.*;
import java.util.*;
import java.io.*;
import java.awt.image.*;
import javax.imageio.*;
/**
 *
 * @author Brian Cullinan
 */

public class Picture extends JInternalFrame implements MouseListener, MouseMotionListener {

    protected int current_count = 0;

    Window window;
    int last_x = 0;
    int last_y = 0;

    BufferedImage red;
    BufferedImage orange;
    BufferedImage yellow;
    BufferedImage green;
    BufferedImage blue;
    BufferedImage purple;


    Vector<PaintPoint> points = new Vector<PaintPoint>();

    public Picture(Window window)
    {
        super();
        this.window = window;

        setBounds(0,
               0,
               window.getWidth() - window.getInsets().left - window.getInsets().right,
               window.getHeight() - window.getInsets().top - window.getInsets().bottom);

        // show painting surface
        addMouseListener(this);
        addMouseMotionListener(this);
        window.add(this);
        setVisible(true);
        try {
            setMaximum(true);
        } catch (Exception ex){}
             try {
                red = ImageIO.read(project2.Tools.class.getResource("red.gif"));
                orange = ImageIO.read(project2.Tools.class.getResource("orange.gif"));
                yellow = ImageIO.read(project2.Tools.class.getResource("yellow.gif"));
                green = ImageIO.read(project2.Tools.class.getResource("green.gif"));
                blue = ImageIO.read(project2.Tools.class.getResource("blue.gif"));
                purple = ImageIO.read(project2.Tools.class.getResource("purple.gif"));
            } catch (Exception ex) {
            }
    }

        @Override
        public void paint(Graphics g) {
            //super.paint(g);
            for(int i = 0; i < points.size(); i++)
            {
                PaintPoint tmp = points.get(i);
                //g.fillRect(i, i, i, i)
                if(tmp.shape == PaintPoint.Shape.ERASER)
                {
                    g.setColor(Color.WHITE);
                    g.fillRect(tmp.position.x-(tmp.size.width/2), tmp.position.y-(tmp.size.height/
                }
                else
                {
                    if(tmp.shape == PaintPoint.Shape.CIRCLE)
                    {
                        g.setColor(tmp.color);
                        g.fillOval(tmp.position.x-(tmp.size.width/2), tmp.position.y-(tmp.size.hei
                    }
                    else
                    {
                        if(tmp.shape == PaintPoint.Shape.SQUARE)
                        {
                            g.setColor(tmp.color);
                            g.fillRect(tmp.position.x-(tmp.size.width/2), tmp.position.y-(tmp.size
                        }
                        else
                        {
                            if(tmp.shape == PaintPoint.Shape.HAND)
                            {
                                // TODO: draw hand picture here
                                     try {
                               if(tmp.color == Color.RED)
                                {
                                        g.drawImage(red, tmp.position.x-(tmp.size.width/2), tmp.po
                                }
                               if(tmp.color == Color.ORANGE)
                                {
                                        g.drawImage(orange, tmp.position.x-(tmp.size.width/2), tmp
                                }
                               if(tmp.color == Color.YELLOW)
                                {
                                        g.drawImage(yellow, tmp.position.x-(tmp.size.width/2), tmp
                                }
                               if(tmp.color == Color.GREEN)
                                {
                                        g.drawImage(green, tmp.position.x-(tmp.size.width/2), tmp.
                                }
                               if(tmp.color == Color.BLUE)
                                {
                                        g.drawImage(blue, tmp.position.x-(tmp.size.width/2), tmp.p
                                }
                               if(tmp.color == Color.magenta)
                                {
                                        g.drawImage(purple, tmp.position.x-(tmp.size.width/2), tmp
                                }
                                    } catch (Exception ex) {

                                    }
                            }
                        }
                    }
                }
            }
            // paint tool
            if(window.tools.selected_button == window.tools.eraser)
            {
                g.drawImage(window.tools.eraser.background_image, last_x, last_y, 50, 50, null);
            }
            if(window.tools.selected_button == window.tools.pen)
            {
                g.drawImage(window.tools.pen.background_image, last_x, last_y, 50, 50, null);
            }
            if(window.tools.selected_button == window.tools.stamp)
            {
                g.drawImage(window.tools.stamp.background_image, last_x, last_y, 50, 50, null);
            }
        }

        public void mouseClicked(MouseEvent e)
        {
        }
        public void mouseEntered(MouseEvent e)
        {

        }
        public void mouseExited(MouseEvent e)
        {

        }
        public void mousePressed(MouseEvent e)
        {
            current_count = 0;
        }
        public void mouseReleased(MouseEvent e)
        {
            window.tools.undo_stack.push(new Integer(current_count));
            window.tools.undo.setEnabled(true);
            window.tools.redo_stack.clear();
            window.tools.redo.setEnabled(false);
            window.tools.tmp_points.clear();
        }
        public void mouseDragged(MouseEvent e)
        {
            last_x = e.getX();
            last_y = e.getY();
            current_count++;
            points.add(new PaintPoint(new Point(e.getX(), e.getY()), window.colors.selected, windo
            repaint();
            e.consume();
        }
        public void mouseMoved(MouseEvent e)
        {
            last_x = e.getX();
            last_y = e.getY();
            this.repaint();
        }

    }
```

C:\Documents and Settings\Brian Cullinan\My Documents\My Homework\CS 477\Project 2\src\project2 \Colors.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package project2;
```

```java
import javax.swing.*;
import java.awt.*;
import javax.swing.event.*;
import java.awt.event.*;
import javax.imageio.*;
import java.io.*;
import java.net.*;
/**
 *
 * @author Brian Cullinan
 */
public class Colors extends JInternalFrame implements ActionListener {

    Window window;

    HighlightButton red;
    HighlightButton orange;
    HighlightButton yellow;
    HighlightButton green;
    HighlightButton blue;
    HighlightButton purple;

    Color selected = Color.RED;
    HighlightButton selected_button;

    public Colors(Window window)
    {
        super();
        this.window = window;

        setBounds(0,
               0,
               90,
               window.getHeight() - window.getInsets().top - window.getInsets().bottom);
        setLayout(null);
        setLayer(9999);
        setTitle("Colors");

        // set up color buttons
        red = new HighlightButton("Red");
        red.setBackground(Color.RED);
        red.setBounds(0, 0, 80, 80);
        red.addActionListener(this);
        add(red);
        selected_button = red;
        selected_button.highlighted = true;

        orange = new HighlightButton("Orange");
        orange.setBackground(Color.ORANGE);
        orange.setBounds(0, 80, 80, 80);
        orange.addActionListener(this);
        add(orange);

        yellow = new HighlightButton("Yellow");
        yellow.setBackground(Color.YELLOW);
        yellow.setBounds(0, 160, 80, 80);
        yellow.addActionListener(this);
        add(yellow);

        green = new HighlightButton("Green");
        green.setBackground(Color.GREEN);
        green.setBounds(0, 240, 80, 80);
        green.addActionListener(this);
        add(green);

        blue = new HighlightButton("Blue");
        blue.setBackground(Color.BLUE);
        blue.setBounds(0, 320, 80, 80);
        blue.addActionListener(this);
        add(blue);

        purple = new HighlightButton("Purple");
        purple.setBackground(Color.magenta);
        purple.setBounds(0, 400, 80, 80);
        purple.addActionListener(this);
        add(purple);

        // show toolbar
        window.add(this);
        setVisible(true);
        //getLayeredPane().getComponent(1).setFont(new Font("Lucida",Font.PLAIN,48));
        //getLayeredPane().getComponent(1).getHeight();
    }

    public void actionPerformed(ActionEvent e) {
        if(selected_button != null)
        {
            selected_button.highlighted = false;
            selected_button.repaint();
        }
        selected_button = (HighlightButton)e.getSource();
        selected_button.highlighted = true;
        selected = selected_button.getBackground();
        try {
        if(e.getSource() == red)
            window.tools.stamp.background_image = ImageIO.read(project2.Tools.class.getResourc
        if(e.getSource() == orange)
            window.tools.stamp.background_image = ImageIO.read(project2.Tools.class.getResourc
        if(e.getSource() == yellow)
            window.tools.stamp.background_image = ImageIO.read(project2.Tools.class.getResourc
        if(e.getSource() == green)
            window.tools.stamp.background_image = ImageIO.read(project2.Tools.class.getResourc
        if(e.getSource() == blue)
            window.tools.stamp.background_image = ImageIO.read(project2.Tools.class.getResourc
        if(e.getSource() == purple)
            window.tools.stamp.background_image = ImageIO.read(project2.Tools.class.getResourc
        window.tools.stamp.repaint();
        } catch (Exception ex) {

        }
    }
    public void paint(Graphics g) {
        super.paint(g);
    }
}
```

C:\Documents and Settings\Brian Cullinan\My Documents\My Homework\CS 477\Project 2\src\project2 \Tools.java

```
import javax.imageio.*;
import java.io.*;
import java.net.*;
```

C:\Documents and Settings\Brian Cullinan\My Documents\My Homework\CS 477\Project 2\src\project2 \Window.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package project2;
```

```java
import javax.swing.*;
import java.awt.*;
import javax.swing.event.*;
import java.awt.event.*;

/**
 *
 * @author Brian Cullinan
 */
public class Window extends JDesktopPane implements ComponentListener {

    Application application;

    Colors colors;
    Tools tools;
    Picture picture;

    public Window(Application application)
    {
        super();

        this.application = application;
        application.setContentPane(this);
        addComponentListener(this);

        // show the window
        application.setVisible(true);

        // set up toolbars
        this.colors = new Colors(this);
        this.tools = new Tools(this);

        // set up the paintable area
        this.picture = new Picture(this);
    }

    public void componentHidden(ComponentEvent e)
    {

    }
    public void componentMoved(ComponentEvent e)
    {

    }
    public void componentResized(ComponentEvent e)
    {
        if(colors != null)
        colors.setBounds(0,
               0,
               90,
               getHeight() - getInsets().top - getInsets().bottom);
        if(picture != null)
        picture.setBounds(90,
               0,
               getWidth() - getInsets().left - getInsets().right,
               getHeight() - getInsets().top - getInsets().bottom);
        if(tools != null)
        tools.setBounds(getWidth() - getInsets().left - getInsets().right - 90,
               0,
               90,
               getHeight() - getInsets().top - getInsets().bottom);
    }
    public void componentShown(ComponentEvent e)
    {

    }
}
```

C:\Documents and Settings\Brian Cullinan\My Documents\My Homework\CS 477\Project 2\src\project2\PaintPoint.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package project2;

import java.awt.*;
/**
 *
 * @author Brian Cullinan
 */
public class PaintPoint
{
    public enum Shape { CIRCLE, SQUARE, HAND, ERASER }
    Shape shape;
    Point position;
    Dimension size = new Dimension(20, 20);
    Color color;

    public PaintPoint(Point position, Color color, Shape shape)
    {
        this.position = position;
        this.color = color;
        this.shape = shape;
    }
}
```

C:\Documents and Settings\Brian Cullinan\My Documents\My Homework\CS 477\Project 2\src\project2\HighlightButton.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package project2;

import javax.swing.*;
import java.awt.*;
import javax.swing.event.*;
import java.awt.event.*;
/**
 *
 * @author Brian Cullinan
 */
public class HighlightButton extends JButton {

    public boolean highlighted = false;
    public Image background_image;

    public HighlightButton(String title)
    {
        super(title);
    }

    public void paint(Graphics g) {
        super.paint(g);
        if(background_image != null)
        {
            g.drawImage(background_image, 10, 10, 60, 60, null);
        }
        if(highlighted)
        {
            float HSBvalues[] = new float[3];
            Color.RGBtoHSB(this.getBackground().getRed(), this.getBackground().getGreen(), thi

            HSBvalues[0] += .5;
            if(HSBvalues[0] > 1.0)
                HSBvalues[0] -= 1.0;

            Color tmpcolor;
            if(HSBvalues[1] == 0)
            {
                tmpcolor = Color.YELLOW;
            }
            else
            {
                tmpcolor = Color.getHSBColor(HSBvalues[0], HSBvalues[1], HSBvalues[2]);
            }
            double tmpred = (this.getBackground().getRed() - tmpcolor.getRed()) * .05;
            double tmpgreen = (this.getBackground().getGreen() - tmpcolor.getGreen()) * .05;
            double tmpblue = (this.getBackground().getBlue() - tmpcolor.getBlue()) * .05;

            //if(tmpcolor.getGreen() > this.getBackground().getGreen()) tmpgreen = tmpcolor.ge

            int red = tmpcolor.getRed();
            int green = tmpcolor.getGreen();
            int blue = tmpcolor.getBlue();

            for(int i = 0; i < 20; i++)
            {
                red += tmpred;
                green += tmpgreen;
                blue += tmpblue;

                if(red > 255) red = 255;
                if(red < 0) red = 0;
                if(green > 255) green = 255;
                if(green < 0) green = 0;
                if(blue > 255) blue = 255;
                if(blue < 0) blue = 0;

                g.setColor(new Color(red, green, blue));
                g.drawRect(i, i, 80 - i*2, 80 - i*2);
            }
        }
    }


}
```

C:\Documents and Settings\Brian Cullinan\My Documents\My Homework\CS 477\Project 2\src\project2 \FileDialog.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package project2;

import javax.swing.*;
import java.awt.*;
```

```java
import javax.swing.event.*;
import java.awt.event.*;
import java.util.*;
import java.io.*;
/**
 *
 * @author Brian Cullinan
 */
public class FileDialog extends JDialog implements DocumentListener, ActionListener {

    Window window;

    JComponent file_name;

    JComboBox folder_path;

    JButton save;
    JButton load;

    public static final String[] keys = {"1234567890", "QWERTYUIOP", "ASDFGHJKL", "ZXCVBNM"};

    public class KeyInput implements ActionListener
    {
        public void actionPerformed(ActionEvent e) {
            if(((JTextField)file_name).getText().equals("Enter File Name"))
                ((JTextField)file_name).setText("");
            JButton key = (JButton)e.getSource();
            ((JTextField)file_name).setText(((JTextField)file_name).getText() + key.getText()
        }
    }

    public FileDialog(Window window, Boolean is_load)
    {
        super();
        this.window = window;

        // place in middle of window
        setBounds((window.getWidth() - window.getInsets().left - window.getInsets().right) / 
               (window.getHeight() - window.getInsets().top - window.getInsets().bottom) / 2,
               450,
               300);
        this.setLayout(new BorderLayout());
        this.setResizable(true);
        //this.setClosable(true);

        // make panel for filepath and folder
        JPanel file = new JPanel();
        file.setLayout(new GridLayout(2, 1));
        this.add(file, BorderLayout.NORTH);

        JPanel file_path = new JPanel();
        file_path.setLayout(new BorderLayout());
        file.add(file_path);

        JLabel file_str = new JLabel("Filepath:");
        file_path.add(file_str, BorderLayout.WEST);

        if(is_load == false)
        {
            file_name = new JTextField("Enter File Name");
            ((JTextField)file_name).getDocument().addDocumentListener(this);
            file_path.add(file_name, BorderLayout.CENTER);

            save = new JButton("Save");
            save.addActionListener(this);
            file_path.add(save, BorderLayout.EAST);
        }
        else
        {
            File directory = new File(System.getProperty("user.home"));
            File[] files = directory.listFiles();
            int count = 0;
            for(int i = 0; i < files.length; i++)
            {
                String filename = files[i].getName();
                if(filename.length() > 10 && filename.substring(filename.length()-10, filename
                {
                    count++;
                }
            }
            String[] files_str = new String[count];
            count = 0;
            for(int i = 0; i < files.length; i++)
            {
                String filename = files[i].getName();
                if(filename.length() > 10 && filename.substring(filename.length()-10, filename
                {
                    files_str[count] = filename.substring(0, filename.length()-10);
                    count++;
                }
            }
            file_name = new JComboBox(files_str);
            file_path.add(file_name, BorderLayout.CENTER);

            load = new JButton("Load");
            load.addActionListener(this);
            file_path.add(load, BorderLayout.EAST);
        }

        JPanel folder = new JPanel();
        folder.setLayout(new BorderLayout());
        file.add(folder);
        JLabel folder_str = new JLabel("Folder:");
        folder.add(folder_str, BorderLayout.WEST);

        String[] paths = {System.getProperty("user.home")};
        folder_path = new JComboBox(paths);
        folder.add(folder_path, BorderLayout.CENTER);

        // make panel for keys
        JPanel key_panel = new JPanel();
        key_panel.setLayout(new GridLayout(FileDialog.keys.length, 1));
        this.add(key_panel, BorderLayout.CENTER);

        KeyInput key_listener = new KeyInput();

        // set up keyboard
        for(int i = 0; i < FileDialog.keys.length; i++)
        {
            JPanel row = new JPanel();
            row.setLayout(new GridLayout(1, FileDialog.keys[i].length()));
            key_panel.add(row);
            for(int j = 0; j < FileDialog.keys[i].length(); j++)
            {
                JButton key = new JButton(""+FileDialog.keys[i].charAt(j));
                key.addActionListener(key_listener);
                row.add(key);
            }
        }
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == save)
        {
            try{
                FileOutputStream file = new FileOutputStream(folder_path.getSelectedItem().toS
                for(int i = 0; i < window.picture.points.size(); i++)
                {
                    file.write(window.picture.points.get(i).color.getRed());
                    file.write(window.picture.points.get(i).color.getGreen());
                    file.write(window.picture.points.get(i).color.getBlue());
                    file.write(window.picture.points.get(i).position.x >> 8);
                    file.write(window.picture.points.get(i).position.x);
                    file.write(window.picture.points.get(i).position.y >> 8);
                    file.write(window.picture.points.get(i).position.y);
                    file.write(window.picture.points.get(i).size.width);
                    file.write(window.picture.points.get(i).size.height);
                    switch(window.picture.points.get(i).shape)
                    {
                        case CIRCLE:
                            file.write(0);
                            break;
                        case SQUARE:
                            file.write(1);
                            break;
                        case HAND:
                            file.write(2);
                            break;
                        case ERASER:
                            file.write(3);
                            break;
                    }
                }
                file.close();
            } catch (Exception ex)
            {
                JOptionPane.showMessageDialog(window,
                "There was a problem saving the file!",
                "Save Error",
                JOptionPane.ERROR_MESSAGE);
            }
            this.setVisible(false);
        }
        else
        {
            if(e.getSource() == load)
            {
                try{
                    window.picture.points.clear();
                    FileInputStream file = new FileInputStream(folder_path.getSelectedItem().t

                    int buffer = 0;
                    while((buffer = file.read()) != -1)
                    {
                        int red = buffer;
                        int green = file.read();
                        int blue = file.read();
                        Color color = new Color(red, green, blue);
                        int x = file.read() << 8;
                        x += file.read();
                        int y = file.read() << 8;
                        y += file.read();
                        int width = file.read();
                        int height = file.read();
                        int type = file.read();
                        PaintPoint.Shape shape = PaintPoint.Shape.CIRCLE;
                        switch(type)
                        {
                            case 0:
                                shape = PaintPoint.Shape.CIRCLE;
                                break;
                            case 1:
                                shape = PaintPoint.Shape.SQUARE;
                                break;
                            case 2:
                                shape = PaintPoint.Shape.HAND;
                                break;
                            case 3:
                                shape = PaintPoint.Shape.ERASER;
                                break;
                        }
                        PaintPoint point = new PaintPoint(new Point(x, y), color, shape);
                        window.picture.points.add(point);
                    }
                } catch (Exception ex)
                {
                    JOptionPane.showMessageDialog(window,
                    "There was a problem loading the file!",
                    "Load Error",
                    JOptionPane.ERROR_MESSAGE);
                }
                window.picture.repaint();
                this.setVisible(false);
            }
        }
    }

    public void changedUpdate(DocumentEvent e) {
    }
    public void removeUpdate(DocumentEvent e) {
    }
    public void insertUpdate(DocumentEvent e) {
        if(((JTextField)file_name).getText().length() > 15 && ((JTextField)file_name).getText(
        {
            EventQueue.invokeLater(new Runnable()
            {public void run() {
            ((JTextField)file_name).setText(((JTextField)file_name).getText().substring(15));
            }
            });
        }
    }

}
```