Computer Programming Fundamentals

CS 152
Professor: Leah Buechley
TAs: Melody Horn, Noah Garcia, Andrew Geyko, Juan Ormaza
Time: MWF 10:00-10:50am
https://handandmachine.cs.unm.edu/classes/CS152_Fall2021/

ASSIGNMENT 4 MOSTLY GRADED

MIDTERM GRADES BY FRIDAY

questions?

OPEN UP PROJECT FROM LAST CLASS

ENTIRE PROGRAM

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseMotionListener;

public class MouseInput extends JPanel implements KeyListener, MouseMotionListener {
    int width;
    int height;
    char keyPressed;
    int keyCode;
    int mouseX, mouseY;
    Graphics g;

    MouseInput(int w, int h) {
        width = w;
        height = h;
        System.out.println("mouseX is: " +mouseX);
        System.out.println("mouseY is: " +mouseY);
        Dimension d = new Dimension(width, height);
        setPreferredSize(d);
        addKeyListener(this);
        addMouseMotionListener(this);
        setFocusable(true);
        requestFocusInWindow();
        setVisible(true);
        g = getGraphics();
    }

    public static void main(String[] args) {
        MouseInput panel = new MouseInput(500,500);
        MyFrame f = new MyFrame(panel);
        //panel.animate(60);
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        //do your drawing here
        setBackground(Color.WHITE);
    }

    void delay(int time) {
        try{ Thread.sleep(time); }
        catch (Exception exc){}
    }

    void animate (int framerate) {
        int delay = 1000/framerate;
        while(true) {
            repaint();
            delay(delay);
        }
    }

    @Override
    public void keyTyped(KeyEvent e) {
        keyPressed = e.getKeyChar();
        System.out.println(keyPressed);
        if (keyPressed==' ')
            repaint();
    }

    @Override
    public void keyPressed(KeyEvent e) {
        keyCode = e.getKeyCode();
    }

    @Override
    public void keyReleased(KeyEvent e) {}

    @Override
    public void mouseMoved(MouseEvent e) {
        mouseX = e.getX();
        mouseY = e.getY();
        System.out.println("The mouse position is: " +mouseX +", " +mouseY);
        g = getGraphics();
        if (mouseX < width/2)
            g.setColor(Color.ORANGE);
        else
            g.setColor(Color.MAGENTA);
        int size = 50;
        g.fillOval(mouseX-size/2, mouseY-size/2, size, size);
    }

    @Override
    public void mouseDragged(MouseEvent e) {

    }
}

class MyFrame {
    private JFrame j;
    private JPanel panel;

    MyFrame (JPanel p) {
        panel = p;
        j = new JFrame();
        j.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        j.add(panel);
        j.pack();
        j.setVisible(true);
    }
}
```

MouseInput class

MyFrame class

YOU CAN DEFINE MORE THAN ONE CLASS IN A FILE

ONLY ONE CAN BE PUBLIC
THE ONE WITH SAME NAME AS FILE

TODAY: MOUSE INTERACTION CONT.

LAST CLASS: MouseMotionListener
TODAY: MouseListener

ADD MouseListener

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseMotionListener;
import java.awt.event.MouseListener;

public class MouseInput extends JPanel implements KeyListener, MouseMotionListener, MouseListener {
```

tells compiler you'll be using mouse input
you'll be "listening" for mouse clicks

ADD MouseListener TO OUR PANEL

```
MouseInput(int w, int h) {
    width = w;
    height = h;
    Dimension d = new Dimension(width, height);
    setPreferredSize(d);
    addKeyListener(this);
    addMouseMotionListener(this);
    addMouseListener(this);
    setFocusable(true);
    requestFocusInWindow();
    setVisible(true);
    System.out.println("The initial value of mouseX is:" + mouseX);
    System.out.println("The initial value of mouseY is:" + mouseY);
}
```

tells program to listen for mouse clicks in our panel/window

ADD 5 METHODS

```
@Override
public void mousePressed(MouseEvent e) {
}
```

code that will run when mouse button is pressed

```
@Override
public void mouseClicked(MouseEvent e) {
}
```

code that will run when the mouse is clicked and released

```
@Override
public void mouseReleased(MouseEvent e) {
}
```

code that will run when mouse button is released

```
@Override
public void mouseEntered(MouseEvent e) {
}
```

code that will run when mouse enters the window

```
@Override
public void mouseExited(MouseEvent e) {
}
```

code that will run when mouse exits the window

note: I will not expect you to remember these on any exam or quiz

COMPILE & RUN TO CHECK CODE

REMOVE PRINT STATEMENT FROM mouseMoved()

```
@Override
public void mouseMoved(MouseEvent e) {
    mouseX = e.getX();
    mouseY = e.getY();
    //System.out.println("The mouse position is: " +mouseX +", " +mouseY);
    g = getGraphics();
    if (mouseX < width/2)
        g.setColor(Color.ORANGE);
    else
        g.setColor(Color.MAGENTA);
    int size = 50;
    g.fillOval(mouseX-size/2, mouseY-size/2, size, size);
}
```

ADD PRINT STATEMENTS TO THESE METHODS

```
@Override
public void mousePressed(MouseEvent e) {
    System.out.println("Mouse pressed.");
}

@Override
public void mouseClicked(MouseEvent e) {
    System.out.println("Mouse clicked.");
}

@Override
public void mouseReleased(MouseEvent e) {
    System.out.println("Mouse released.");
}
```

COMPILE & RUN
CLICK MOUSE SLOWLY

NOTE ORDER OF EVENTS

`Mouse pressed.`	code that runs when mouse button is pressed
`Mouse released.`	code that runs when mouse button is released
`Mouse clicked.`	code that runs when mouse button is pressed and released. *runs when the click is finished.*

questions?

USING THE MOUSE TO CONTROL GRAPHICS

EDIT THE mousePressed METHOD

```
@Override
public void mousePressed(MouseEvent e) {
    System.out.println("Mouse pressed.");
    g = getGraphics();
    g.setColor(Color.DARK_GRAY);
    int size = 100;
    g.fillRect(mouseX-size/2, mouseY-size/2, size, size);
}
```

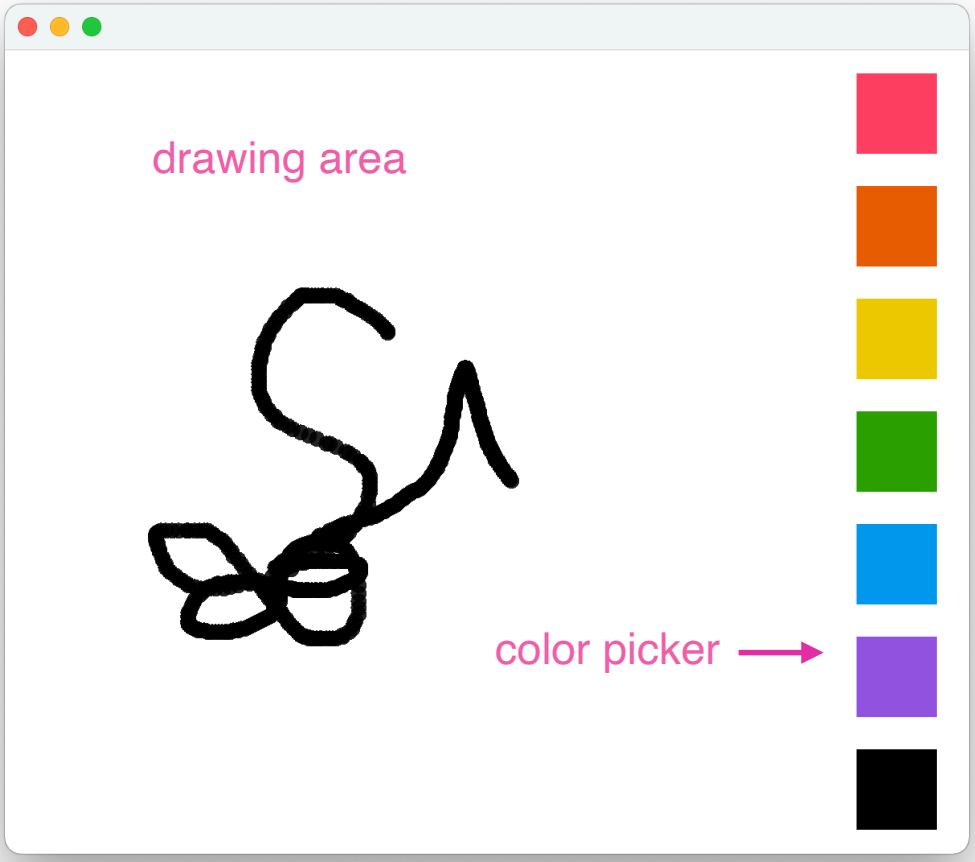
what does this code do?

COMPILE & RUN

LET’S MAKE A DRAWING PROGRAM

LET’S MAKE A DRAWING PROGRAM

A SIMPLE DRAWING PROGRAM



- click on a color square to change drawing color
- press spacebar to clear drawing (but not color picker)
- draw only when mouse is dragged
- don’t draw on color picker

THINGS WE NEED TO DO

- Draw color picker squares along edge
- Implement color picking
  - create variable to store current color information
  - determine which square is being clicked on
  - change color when a different square is clicked
- Implement drawing when dragged
  - draw when mouse is dragged
  - don’t draw on color picker
- Implement screen clearing
  - clear screen when space bar is pressed
  - don’t clear color picker

WHERE TO START?

- Let’s start with a smaller version of the challenge
- Draw 1 color picker square along edge
- Implement program for that color only

ADD A VARIABLE FOR COLOR

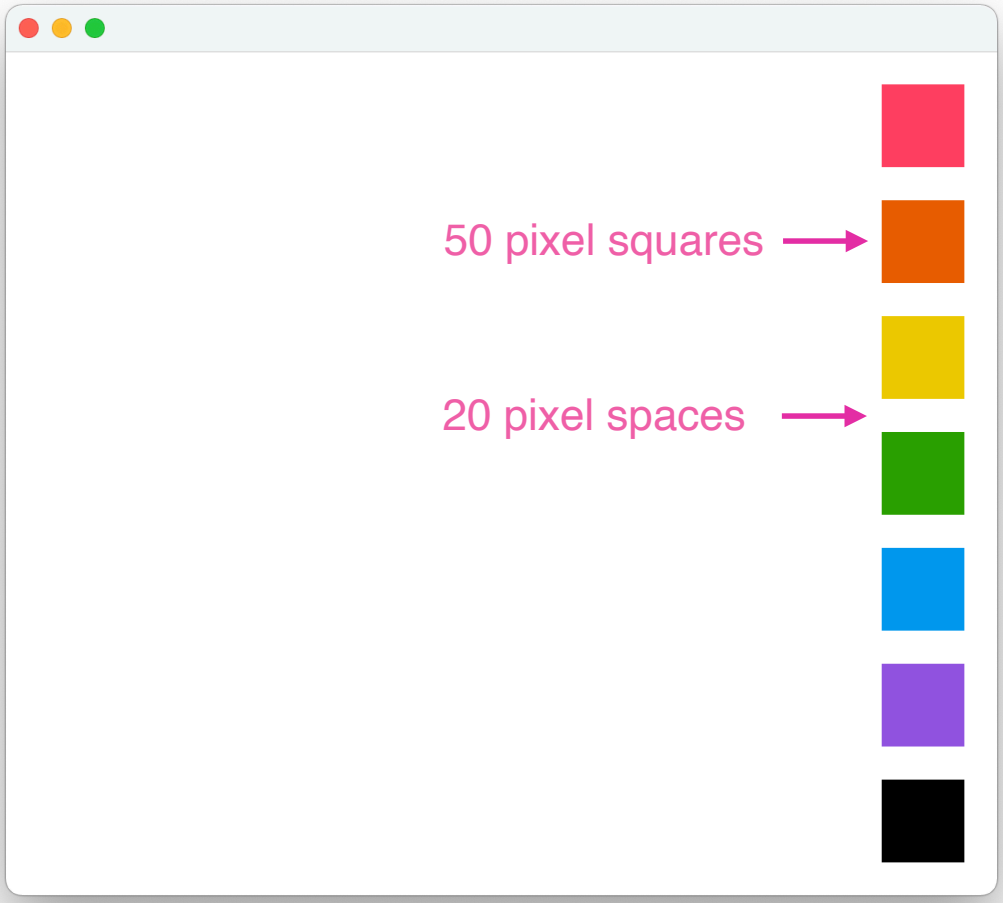
```
public class MouseInput extends JPanel implements KeyListener, MouseMotionListener, Mouse Listener {
    int width;
    int height;
    char keyPressed;
    int keyCode;
    int mouseX, mouseY;
    Graphics g;
    Color drawingColor;
```

will store the color we are drawing with

INITIALIZE IN CONSTRUCTOR

```
MouseInput(int w, int h) {
    width = w;
    height = h;
    drawingColor = Color.BLACK;
    Dimension d = new Dimension(width, height);
    setPreferredSize(d);
    addKeyListener(this);
    addMouseMotionListener(this);
    addMouseListener(this);
    setFocusable(true);
    requestFocusInWindow();
    setVisible(true);
    System.out.println("The initial value of mouseX is:" + mouseX);
    System.out.println("The initial value of mouseY is:" + mouseY);
}
```

THINK ABOUT SPACING



- 7 colors
- size: 50 pixels
- spacing between:
- 20 pixels
- spacing from edge:
  20 pixels

ADD A VARIABLES FOR PICKER

```
public class MouseInputPractice extends JPanel implements KeyListener, MouseMotionListener,
MouseListener {
    int width;
    int height;
    char keyPressed;
    int keyCode;
    int mouseX, mouseY;
    Graphics g;
    Color drawingColor;
    //color picker variables
    final int squareSize = 50;
    final int spacing = 20;
```

information about the color picker

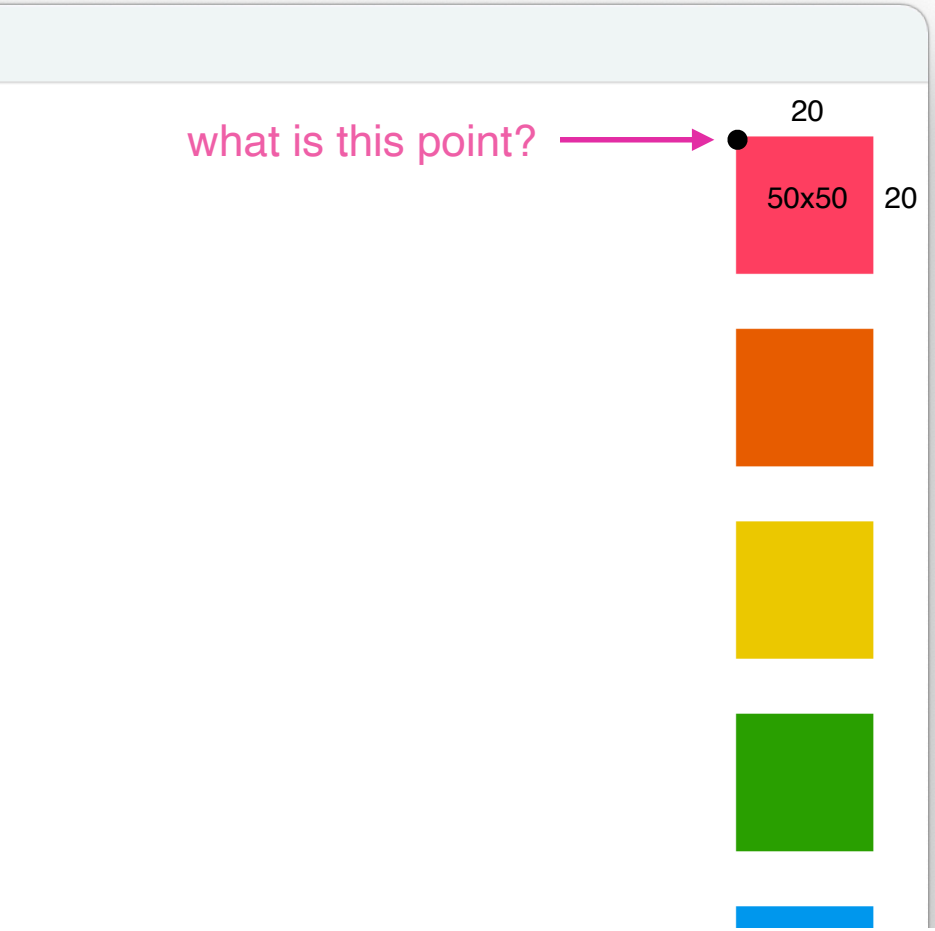
JAVA KEYWORD FINAL

```
//color picker variables
final int squareSize = 50;
final int spacing = 20;
```

- used to define constants
- values that do not change in your program

questions?

DRAWING THE FIRST SQUARE



x = width - 50 - 20

y = 20

x = width - squareSize - spacing

y = spacing

WHERE IN OUR CODE SHOULD WE DRAW COLOR PICKER?

IN paintComponent()

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    setBackground(Color.WHITE);
    //draw color picker
}
```

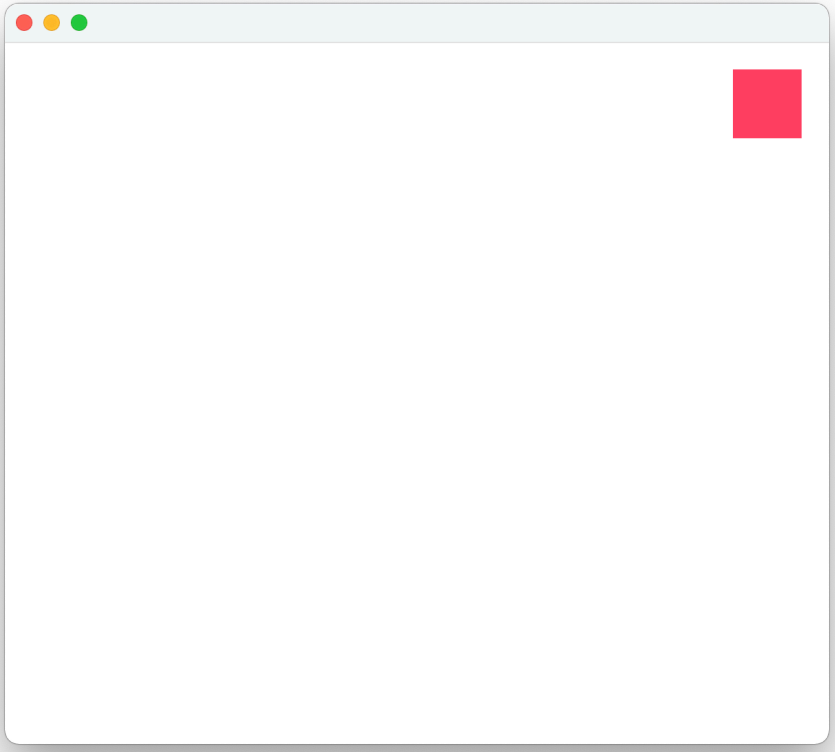
IN paintComponent()

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    setBackground(Color.WHITE);
    //color picker variables
    int x,y;
    x = width-squareSize-spacing;
    y = spacing;
}
```

IN paintComponent()

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    setBackground(Color.WHITE);
    //color picker variables
    int x,y;
    x = width-squareSize-spacing;
    y = spacing;
    Color color = new Color(238, 82, 101);
    g.setColor(color);
    g.fillRect(x, y, squareSize,squareSize);
}
```

COMPILE & RUN



questions?

LET’S GET THIS SQUARE TO WORK

THINGS WE NEED TO DO

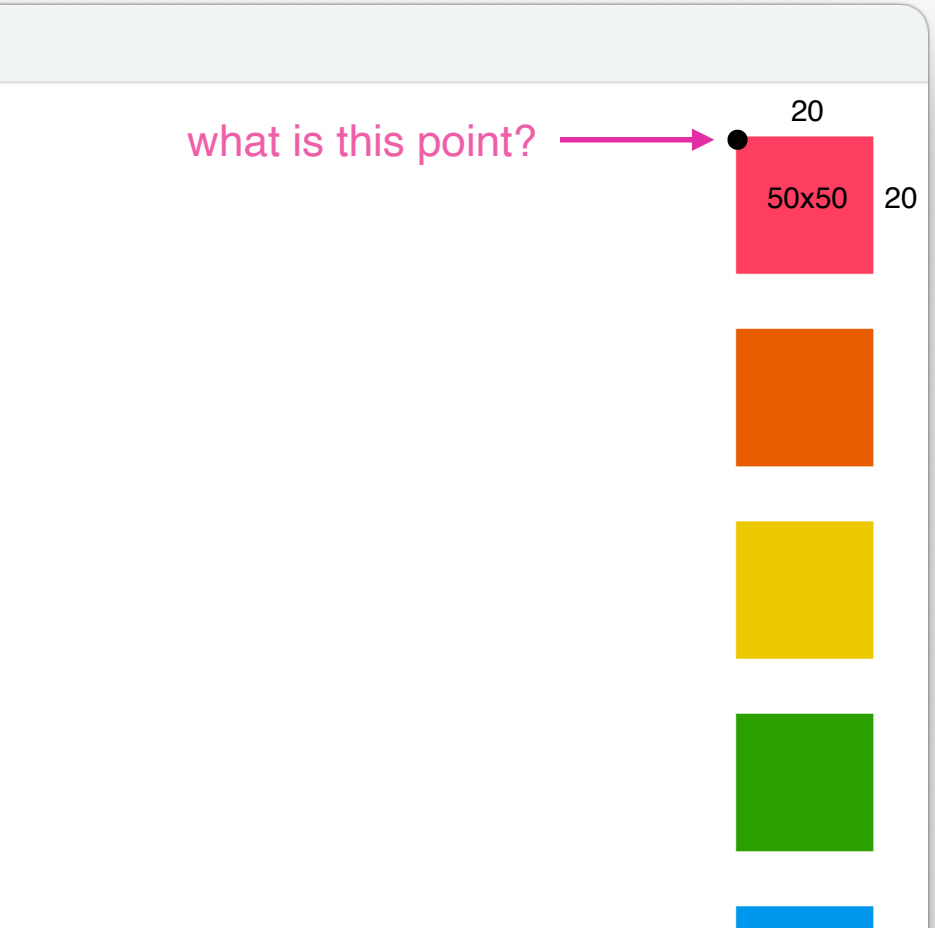
- ~~Draw 1 color picker square along edge~~
- Implement color picking
  - ~~create variable to store current color information~~
  - determine which square is being clicked on
  - change color when a different square is clicked
- Implement drawing when dragged
  - draw when mouse is dragged
  - don’t draw on color picker
- Implement screen clearing
  - clear screen when space bar is pressed
  - don’t clear color picker

WHERE IN OUR CODE SHOULD WE CHECK WHICH SQUARE IS BEING CLICKED ON?

IN mousePressed()

```
public void mousePressed(MouseEvent e) {
    System.out.println("Mouse pressed.");
    g = getGraphics();
    //determine which square is being clicked on
}
```

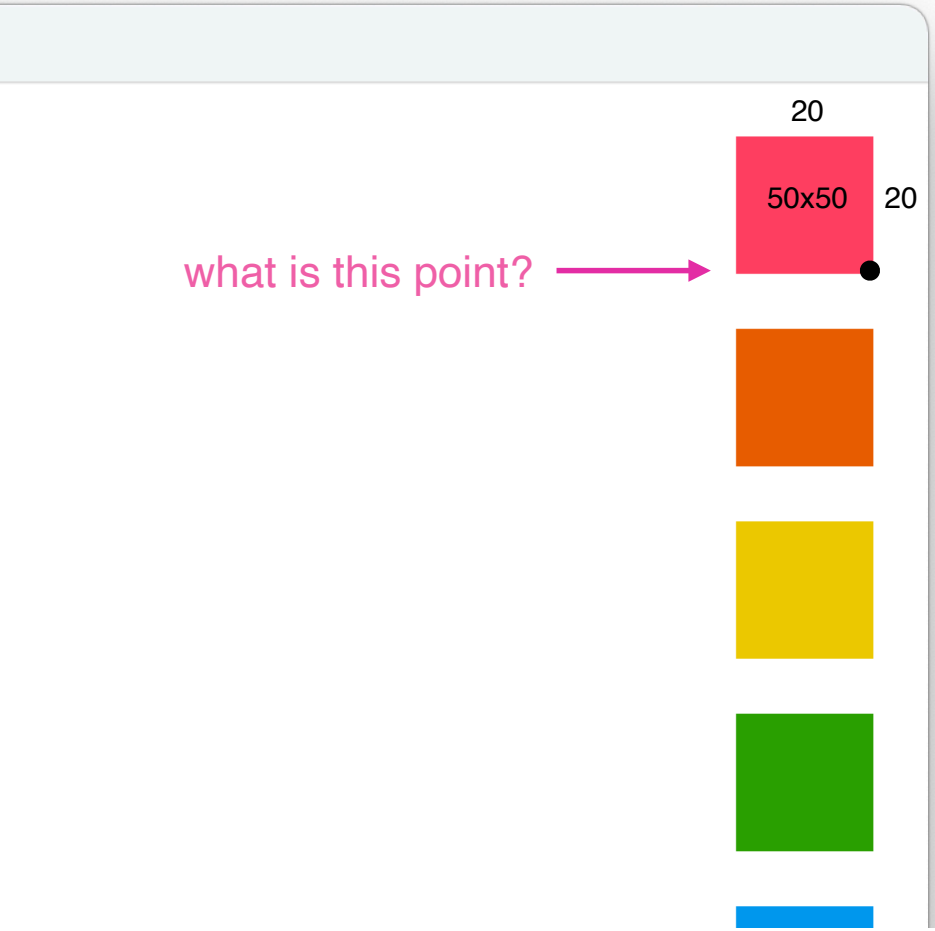
BACK TO POSITIONS



minX = width - squareSize - spacing

minY = spacing

BACK TO POSITIONS



maxX = width - spacing

maxY = spacing + squareSize

IN mousePressed()

```
public void mousePressed(MouseEvent e) {
    mouseX=e.getX();
    mouseY=e.getY();
    System.out.println("Mouse pressed.");
    g = getGraphics();
    int minX,maxX;
    int minY,maxY;
    minX = width-spacing-squareSize;
    maxX = width-spacing;
    minY = spacing;
    maxY = spacing+squareSize;
}
```

IN mousePressed()

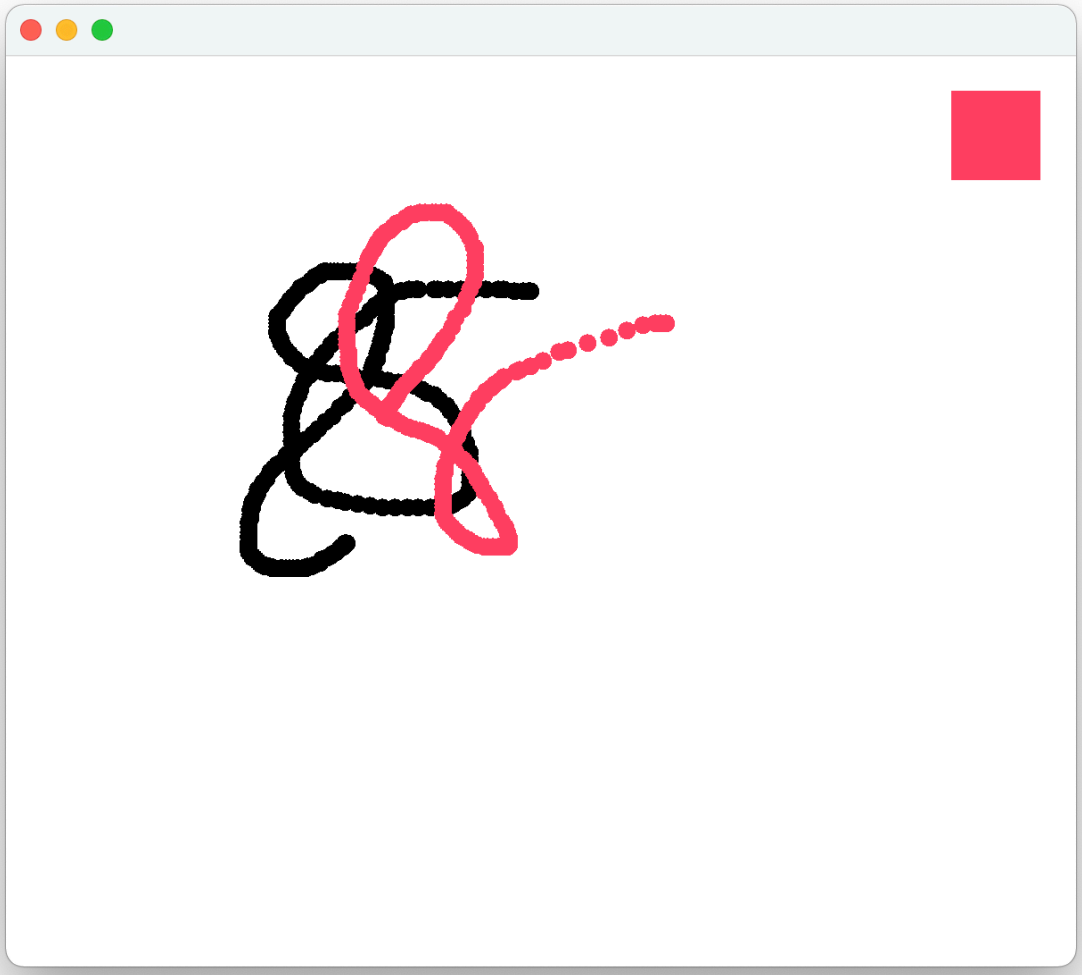
```
public void mousePressed(MouseEvent e) {
    mouseX=e.getX();
    mouseY=e.getY();
    System.out.println("Mouse pressed.");
    g = getGraphics();
    int minX,maxX;
    int minY,maxY;
    minX = width-spacing-squareSize;
    maxX = width-spacing;
    minY = spacing;
    maxY = spacing+squareSize;
    //determine which square is being clicked on
    if (mouseX>minX && mouseX<maxX && mouseY>minY && mouseY<maxY)
        drawingColor = new Color(238, 82, 101);
}
```

questions?

WHERE IN OUR CODE SHOULD WE DO DRAWING?

IN mouseDragged()

```
public void mouseDragged(MouseEvent e) {
    mouseX = e.getX();
    mouseY = e.getY();
    g = getGraphics();
    g.setColor(drawingColor);
    int size = 10;
    g.fillOval(mouseX-size/2,mouseY-size/2,size,size);
}
```



questions?

Thank you!

CS 152
Professor: Leah Buechley
TAs: Melody Horn, Noah Garcia, Andrew Geyko, Juan Ormaza
Time: MWF 10:00-10:50am
https://handandmachine.cs.unm.edu/classes/CS152_Fall2021/