# “SWING” JAVA CODING STANDARDS FOR RPS

The following coding technique is common among swing programs.

**Public Class PDGameApp extends JFrame implements ActionListener,ListSelectionListener {**

(then code all instance variables/pointers of the GUI interface that will be accessed by various local methods of this class during execution of game/program)

//instance variables ..some have pointers filled in,some will be filled in later

private final JButton startGameBtn = new JButton("Start New Game"); //several buttons needed

private final DefaultListModel<String> model = new DefaultListModel<String>(); //these 4 lines are for upper left list of games

private final JList<String> finishedGamesList = new JList<String>(model);

private final JScrollPane gamesListPane = new JScrollPane(gamesList, JScrollPane.VERTICAL\_SCROLLBAR\_AS\_NEEDED,

JScrollPane.HORIZONTAL\_SCROLLBAR\_NEVER);

private JTextArea roundResultsTA = new JTextArea(15, 35); //these 2 lines are for lower right output area showing a round played

private JScrollPane outputAreaPaneScroll = new JScrollPane(roundResultsTA);

private JTextField rockTextF = new JTextField();

and a bunch of “panels” and buttons ;

private RPSGame currentgameRPS =null; //this gets filled in when we start a game on start button click

**public static void main**{ //start execution here

createAndShowGUI();}

**public static void createAndShowGUI()** {

RPSGameApp frame1 = new RPSGameApp(); //new up 🡪 this class, & call constructor, --due to extends, it is a frame

frame1.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

frame1.pack();

frame1.setVisible(true);

…}

**RPSGameApp()** { //constructor where the GUI is built and added to frame, lots of code here

super(“Rock Paper Scissors”); //call frame class constructor,recommended by java

setLayout(new BorderLayout); //set layout for frame, no reference is needed here since we are in a non-static method

//create and add west panel

JPanel WestPanel= new JPanel(new BorderLayout); //then set color, borders on panel1

add(WestPanel, BorderLayout.WEST);

//create JList.., add to a scroll pane, then add it to north region of west panel

finishedGamesList = new JList<string> (listmodel) … …..

WestPanel.add(new JScrollPane(finishedGamesList), BorderLayout.NORTH);

finishedGamesList.addListSelectionListener(this);

//sample code to add a label and text field to bottom left panel, assume panel 2 is lower left panel

panel2.add(new JLabel(" ")); //how to add a blank label to a panel

panel2.add(new JLabel(“Rock”); //this label will not change, can new it up here in constructor

panel2.add(rockTextF); //text field is defined above because it will change over time with other methods

+more coded to add all panels, and also hook up all buttons with listeners example: startGameBtn.addActionListener(this);}

//user has clicked some button and this event occurs,determine which button was clicked and take action

**public void actionPerformed(Actionevent e)**

if e.getsource()==startBtn

if (currentgameRPS not null) //this means we are finishing a game that was just completed

add to hashmap gamestat and addelement to jlist in upper left corner

show results in bottom left text areas

else

//we are starting first game

currentGame RPS= new RPSGame();

enable buttons for user

endif

else if e.getsource() = radiorRockBtn

usergesture=1

else if….btn = GO

roundResultsTA.append (string) //string is the string you get back after you call play round

else if etc

//event occurs when user clicks on a game date in upper left jlist

**Public void valueChanged(ListSelectionEvent e)** {

If(!finishedGamesList.isSelectionEmpty(){

String key= (String)finishedGamesList.getselectedValue() ; //get the date key that the user has chosen in listbox

then use key to find the entry in hashtab…..

rockTextF.setText(Integer.toString(gameStat.getUserGestureCount(1))); //how to put text in the text field for rock

rockTextF.setFont( new Font("SansSerif", Font.BOLD, 18)); //gives you large bold font

ENDIF

**SWING GUI LAYOUT FOR RPS**

Create the JFRAME as usual and shown in the sample code

Add a west panel to the jframe, the west panel is border layout

In the west panel, add the gameslist pane to the north border

In the west panel, add a stats panel to the south border

In the statspanel , add a count panel to the north and make it a 3x4 grid layout and add labels and textfields

In the statspanel, add a results panel to the south and make it 2x4 for ties and winner and add labels,etc

Add an east panel to the jframe, the east panel is border layout

In the east panel, add a control panel which is grid 2x 1, and add it to the north area

In the control panel add a start panel which is flow and put a start button there.

Then add a an inputChoice panel which is flow and has the radio buttons and go button on it

Add to the south border the outputTextArea scroll pane which will show the results of each round

**Creating custom colors**

Color c1 = new Color(211, 211, 55); //higher numbers means lighter colors

myPanel.setBackground(c1) //sets the color