CLASS:

```
1   package notes12;
2   import javax.swing.*;
3   import java.awt.*;
4   import java.awt.event.*;
5
6   public class MyFrame extends JFrame implements ActionListener{
7       final int FRAME_WIDTH=300;
8       final int FRAME_HEIGHT=150;
9       JLabel heading;
10      JLabel spacing;
11      JTextField billField;
12      JButton button1;
13      JButton button2;
14
15      public MyFrame(){
16          super("My Frame");
17          heading=new JLabel("Tip Calculator");
18          billField=new JTextField(12);
19          button1=new JButton("15%");
20          button2=new JButton("20%");
21          spacing=new JLabel("     ");
22
23          setSize(FRAME_WIDTH,FRAME_HEIGHT);
24          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25
26
            button1.addActionListener(this);
            button2.addActionListener(this);
            billField.addActionListener(this);
30
31          heading.setFont(new Font("Arial",Font.BOLD,16));
32          button1.setFont(new Font("Arial",Font.BOLD,16));
33          billField.setFont(new Font("Arial",Font.BOLD,16));
34
35          setLayout(new FlowLayout());
36          add(heading);
37          add(billField);
```

```
    private Color colors2[][]=new Color[ROW][COL2];
    private Color colors3[][]=new Color[ROW][COL3];
    //color
    private Color color1=Color.BLACK;
    private Color color2=Color.WHITE;
    //JComboBox for difficulty of game
    private String[] choice={"Selection","Easy","Normal","Hard"};
    private JComboBox combo=new JComboBox(choice);
    //JTextArea for how to play game
    private JTextArea text=new JTextArea("Easy: tap left and right to move\n"
            + "Normal: tap left, up and right to move\nHard: tap left, up, down"
            + " and right to mome");
    //count the black tiles
    private int counter=0;
    //the selection of difficulty
    private int hard=0;
    //mark black tiles in roll to help construct the game
    private int black;
    //choice of difficultys
    private boolean choose1;
    private boolean choose2;
    private boolean choose3;
    //check if the player touches the black tile
    private boolean isblack=false;

    public game(){
        super("Piano Tile");

        setSize(300,600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        setLayout(c);
        //add menu
        setJMenuBar(bar);
        bar.add(file);
        file.add(home);
        file.add(exit);
```

```
75          //panel of home
76          panel.setLayout(new FlowLayout());
77          panel.add(label1);
78          panel.add(combo);
79          panel.add(begin);
80          panel.add(text);
81          add(panel,"panel");
82          //panel of easy
83          panel1.setLayout(new BorderLayout());
84          panel1.add(label21,BorderLayout.NORTH);
85          panel1.add(panel1C,BorderLayout.CENTER);
86          add(panel1,"panel1");
87          //panel of normal
88          panel2.setLayout(new BorderLayout());
89          panel2.add(label22,BorderLayout.NORTH);
90          panel2.add(panel2C,BorderLayout.CENTER);
91          add(panel2,"panel2");
92          //panel of hard
93          panel3.setLayout(new BorderLayout());
94          panel3.add(label23,BorderLayout.NORTH);
95          panel3.add(panel3C,BorderLayout.CENTER);
96          add(panel3,"panel3");
97          //panel of result
98          panel4.add(label3);
99          add(panel4,"panel4");
100         //construct easy
101         for(int i=0;i<ROW;i++){
102             //random one in a row will be black
103             black=(int)(Math.random()*2);
104             for(int j=0;j<COL1;j++){
105                 panels1[i][j]=new JPanel();
106                 panel1C.add(panels1[i][j]);
107                 if(j==black){
108                     panels1[i][j].setBackground(color1);
109                     colors1[i][j]=color1;
110                 }
111                 else{
```

```
                    panels1[i][j].setBackground(color2);
                    colors1[i][j]=color2;
                }
            }
        }
        //construct normal
        for(int i=0;i<ROW;i++){
            black=(int)(Math.random()*3);
            for(int j=0;j<COL2;j++){
                panels2[i][j]=new JPanel();
                panel2C.add(panels2[i][j]);
                if(j==black){
                    panels2[i][j].setBackground(color1);
                    colors2[i][j]=color1;
                }
                else{
                    panels2[i][j].setBackground(color2);
                    colors2[i][j]=color2;
                }
            }
        }
        //construct hard
        for(int i=0;i<ROW;i++){
            black=(int)(Math.random()*4);
            for(int j=0;j<COL3;j++){
                panels3[i][j]=new JPanel();
                panel3C.add(panels3[i][j]);
                if(j==black){
                    panels3[i][j].setBackground(color1);
                    colors3[i][j]=color1;
                }
                else{
                    panels3[i][j].setBackground(color2);
                    colors3[i][j]=color2;
                }
            }
        }
```

```
149             //addlistners
                home.addActionListener(this);
                exit.addActionListener(this);
                begin.addActionListener(this);
                combo.addItemListener(this);
                addKeyListener(this);
155         }
156
157         @Override
            public void actionPerformed(ActionEvent ae) {
159             Object Source=ae.getSource();
160             Container con=getContentPane();
161             //come back to home from menu and clear counter
162             if(Source==home){
163                 c.show(con,"panel");
164                 counter=0;
165             }
166             //exit from home
167             else if(Source==exit){
168                 System.exit(0);
169             }
170             //jump to game panels from home
171             else if(Source==begin){
                    if(hard==1){
173                     c.show(con,"panel1");
174                     choose1=true;
175                     choose2=false;
176                     choose3=false;
177                 }
178                 else if(hard==2){
179                     c.show(con,"panel2");
180                     choose1=false;
181                     choose2=true;
182                     choose3=false;
183                 }
184                 else if(hard==3){
185                     c.show(con,"panel3");
186                     choose1=false;
```

```
187                     choose2=false;
188                     choose3=true;
189                 }
190             }
191         }
192
193         @Override
194         public void itemStateChanged(ItemEvent ie) {
195             Object source=ie.getSource();
196             int select=ie.getStateChange();
197             //get the difficulty from JComboBox
198             if(source==combo){
199                 if(select==ItemEvent.SELECTED)
200                     hard=combo.getSelectedIndex();
201             }
202         }
203
204         @Override
205         public void keyTyped(KeyEvent ke) {
206
207         }
208
209         @Override
210         public void keyPressed(KeyEvent ke) {
211             int keyCode=ke.getKeyCode();
212             //easy mode to check if player touches black tile
213             if(choose1){
214                 //if taping left and the color of left is black, then game continues
215                 if(keyCode==KeyEvent.VK_LEFT){
216                     if(colors1[4][0]==color1){
217                         isblack=true;
218                         counter++;
219                     }
220                     //the tile is not black, game over
221                     else{
222                         c.show(getContentPane(),"panel4");
223                         label3.setText("Your record is:"+counter);
```

```
224                         label3.setFont(new Font("Arial",Font.BOLD,20));
225                     }
226                 }
227                 //same logic
228                 else if(keyCode==KeyEvent.VK_RIGHT){
229                     if(colors1[4][1]==color1){
230                         isblack=true;
231                         counter++;
232                     }
233                     else{
234                         c.show(getContentPane(),"panel4");
235                         label3.setText("Your record is:"+counter);
236                         label3.setFont(new Font("Arial",Font.BOLD,20));
237                     }
238                 }
239             }
240             //normal, same logic as esay
241             else if(choose2){
                    if(keyCode==KeyEvent.VK_LEFT){
243                     if(colors2[4][0]==color1){
244                         isblack=true;
245                         counter++;
246                     }
247                     else{
248                         c.show(getContentPane(),"panel4");
249                         label3.setText("Your record is:"+counter);
250                         label3.setFont(new Font("Arial",Font.BOLD,20));
251                     }
252                 }
253                 else if(keyCode==KeyEvent.VK_UP){
254                     if(colors2[4][1]==color1){
255                         isblack=true;
256                         counter++;
257                     }
258                     else{
259                         c.show(getContentPane(),"panel4");
260                         label3.setText("Your record is:"+counter);
261                         label3.setFont(new Font("Arial",Font.BOLD,20));
```

```
262                     }
263                 }
264                 else if(keyCode==KeyEvent.VK_RIGHT){
265                     if(colors2[4][2]==color1){
266                         isblack=true;
267                         counter++;
268                     }
269                     else{
270                         c.show(getContentPane(),"panel4");
271                         label3.setText("Your record is:"+counter);
272                         label3.setFont(new Font("Arial",Font.BOLD,20));
273                     }
274                 }
275             }
276             //hard, same logic as easy
277             else if(choose3){
                    if(keyCode==KeyEvent.VK_LEFT){
279                     if(colors3[4][0]==color1){
280                         isblack=true;
281                         counter++;
282                     }
283                     else{
284                         c.show(getContentPane(),"panel4");
285                         label3.setText("Your record is:"+counter);
286                         label3.setFont(new Font("Arial",Font.BOLD,20));
287                     }
288                 }
289                 else if(keyCode==KeyEvent.VK_UP){
290                     if(colors3[4][1]==color1){
291                         isblack=true;
292                         counter++;
293                     }
294                     else{
295                         c.show(getContentPane(),"panel4");
296                         label3.setText("Your record is:"+counter);
297                         label3.setFont(new Font("Arial",Font.BOLD,20));
298                     }
```

```
299                 }
300                 else if(keyCode==KeyEvent.VK_DOWN){
301                     if(colors3[4][2]==color1){
302                         isblack=true;
303                         counter++;
304                     }
305                     else{
306                         c.show(getContentPane(),"panel4");
307                         label3.setText("Your record is:"+counter);
308                         label3.setFont(new Font("Arial",Font.BOLD,20));
309                     }
310                 }
311                 else if(keyCode==KeyEvent.VK_RIGHT){
312                     if(colors3[4][3]==color1){
313                         isblack=true;
314                         counter++;
315                     }
316                     else{
317                         c.show(getContentPane(),"panel4");
318                         label3.setText("Your record is:"+counter);
319                         label3.setFont(new Font("Arial",Font.BOLD,20));
320                     }
321                 }
322             }
323         }
324 
325         @Override
326         public void keyReleased(KeyEvent ke) {
327             //easy
328             if(choose1){
329                 //check if the player moves right
330                 if(isblack){
331                     //all the tiles moving down one row, and create a new row on the
332                     //top, delete bottom row
333                     for(int i=ROW-1;i>=0;i--){
334                         black=(int)(Math.random()*2);
335                         for(int j=0;j<COL1;j++){
```

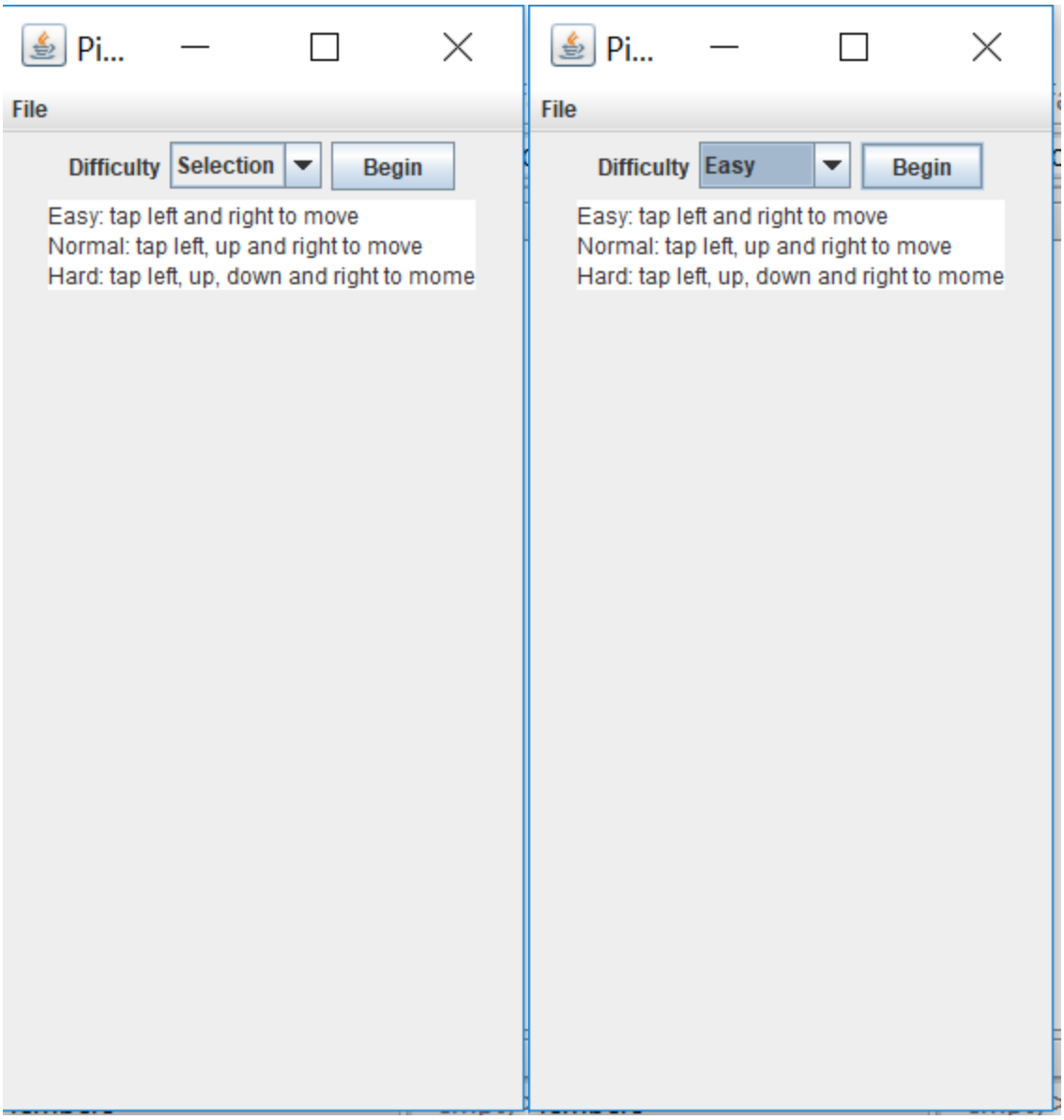
```
                    if(i==0){
                        if(j==black){
                            panels1[i][j].setBackground(color1);
                            colors1[i][j]=color1;
                        }
                        else{
                            panels1[i][j].setBackground(color2);
                            colors1[i][j]=color2;
                        }
                    }
                    else{
                        panels1[i][j].setBackground(colors1[i-1][j]);
                        colors1[i][j]=colors1[i-1][j];
                    }
                }
            }
            isblack=false;
        }
    }
    //normal, same logic as easy
    else if(choose2){
        if(isblack){
            for(int i=ROW-1;i>=0;i--){
                black=(int)(Math.random()*3);
                for(int j=0;j<COL2;j++){
                    if(i==0){
                        if(j==black){
                            panels2[i][j].setBackground(color1);
                            colors2[i][j]=color1;
                        }
                        else{
                            panels2[i][j].setBackground(color2);
                            colors2[i][j]=color2;
                        }
                    }
                    else{
                        panels2[i][j].setBackground(colors2[i-1][j]);
```

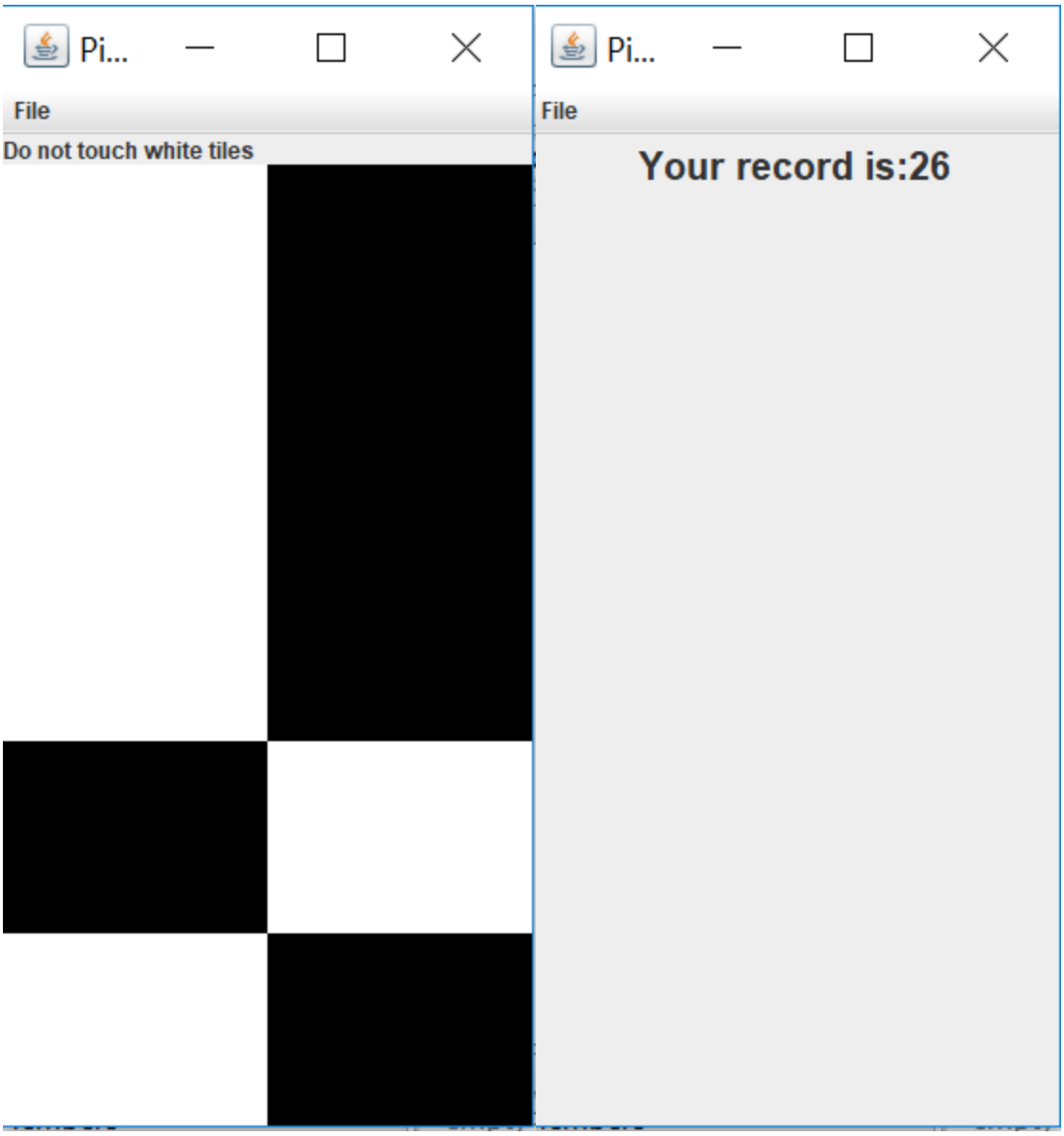
```
                    colors2[i][j]=colors2[i-1][j];
                }
            }
        }
        isblack=false;
    }
}
//hard, same logic as easy
else if(choose3){
    if(isblack){
        for(int i=ROW-1;i>=0;i--){
            black=(int)(Math.random()*4);
            for(int j=0;j<COL3;j++){
                if(i==0){
                    if(j==black){
                        panels3[i][j].setBackground(color1);
                        colors3[i][j]=color1;
                    }
                    else{
                        panels3[i][j].setBackground(color2);
                        colors3[i][j]=color2;
                    }
                }
                else{
                    panels3[i][j].setBackground(colors3[i-1][j]);
                    colors3[i][j]=colors3[i-1][j];
                }
            }
        }
        isblack=false;
    }
}
    }
}

```

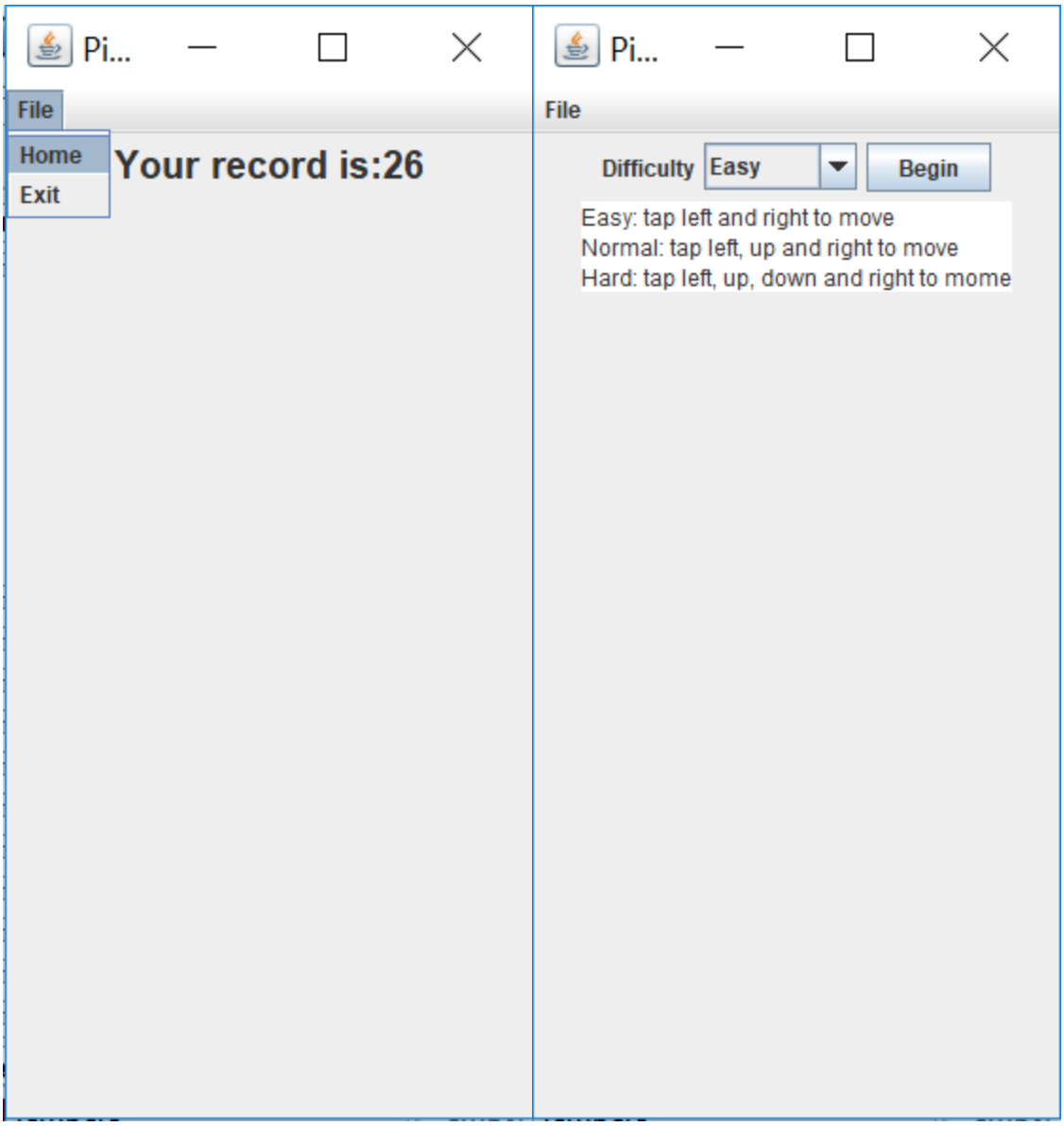
MAIN:

```java
package pkgfinal;

/**Final-Piano Tiles
 * Date:11/27/2018
 * Name:Yifei Feng
 * Using GUI to create Piano Tiles
 */

public class Final {

    public static void main(String[] args) {
        game g=new game();
        g.setVisible(true);
        //making keylistener work
        g.setFocusable(true);
        g.requestFocusInWindow();
    }

}
```

OUTPUT:


Pi...
File
Difficulty
Selection
Begin
Easy: tap left and right to move
Normal: tap left, up and right to move
Hard: tap left, up, down and right to mome
Pi...
File
Difficulty
Easy
Begin
Easy: tap left and right to move
Normal: tap left, up and right to move
Hard: tap left, up, down and right to mome


Pi...
File
Do not touch white tiles
Pi...
File
Your record is:26


Pi...
File
Home
Exit
Your record is:26
Pi...
File
Difficulty
Easy
Begin
Easy: tap left and right to move
Normal: tap left, up and right to move
Hard: tap left, up, down and right to mome