```java
import java.io.*;
import java.util.*;

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

/**
 * NNEED TO HAVE HANGMAN PICS
 * This Hangman Game was create to be the final challenge once the player
 * has traversed through the game map and has made it through all the smaller
 * challenges. The Hangman Gui works by doing most of the work in the actionListener
 * as the action listner is a continuous loop itself depending on the number of clicks.
 * The way it works (AS OF NOW AND SUBJECT TO UPDATE)is that we have given the user the freedom
 * to continue inputing things even though it is determined that they have won or lost. There
 * had to be several changes seeing as how the gui reacts differently with its use of
 * textfield/listeners instead of scanners
 * */
public class THangmanGui{

  //variables that are not data types
  private String phrase;
  private final static int MAX_ERRORS = 7;
  private int errors;
  private String gueDisplay;
  private Boolean wonLost;

  //objects for gui
  private JPanel test;
  private JFrame frame;
  ////////above are temps
  private JTextField letterInput;
  private ImageIcon room;
  private JLabel img;
  private JLabel instructionsTxt;
  private JLabel gameTxt;
  private JLabel question ;
  private JLabel currentGuesses;
  private JLabel errorsLeft;
  private JLabel incorrectInput;

  //data that needs to be accessed by the actionListener
  private LinkedList<String> uniqueLetters;
  private LinkedList<String> guesses;
  private LinkedList<String> correctGuesses;



  public THangmanGui(String p){ //will take in the frame and the panel

    phrase = p.toLowerCase();
    errors = MAX_ERRORS;
    gueDisplay = displayDisguise(guesses);
    wonLost = false;

    //temporary----------------------------------------
    test = new JPanel();
    frame = new JFrame();

    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setPreferredSize(new Dimension(870,700));

    frame.setTitle("Adventure Time");
    //frame.setResizable(false);
    //temporary------------------------------------

    letterInput = new JTextField(20);
    room = new ImageIcon(hangmanGraphics(MAX_ERRORS));
    JLabel img = new JLabel(room);
    instructionsTxt = new JLabel("");
    gameTxt = new JLabel("");
    question = new JLabel("");
    currentGuesses = new JLabel("");
    errorsLeft = new JLabel("");
    incorrectInput = new JLabel("");

    uniqueLetters = makeUniqueLetters();
    guesses = new LinkedList<String>();
    correctGuesses = new LinkedList<String>();

  }

  /**
   * @return string of a jpg picture
   * */
  public String hangmanGraphics(int w){
    String s = "hangmanPic" + w + ".jpg";
    return s;
  }


  /**
   * @return LinkedList holding the letters(no duplicates)
   * of the phrase
   * */
  public LinkedList<String> makeUniqueLetters(){
    LinkedList<String> uqLetters = new LinkedList<String>();

    String temp = phrase.replace(" " ,"");//makes sure no spaces are considered a character that needs to be guesses
    int i = 0;
    while (temp.length() > 0 ){

      String ch = Character.toString(temp.charAt(0));//gets the first character always and turns it into a string
      uqLetters.add(ch);

      temp = temp.replace(ch ,"");//gets rid of all characters that equal the first character preventing duplicates
      i++;
    }
    return uqLetters;
  }


  /**
   * @return String that disguises the letters of the phrase and shows
   * the letters that the user has correctly guessed
   * */
  public String displayDisguise(LinkedList<String> foundLetters){
    String s = "";

    System.out.println("Entering");

    for (int i = 0; i < phrase.length(); i++){

      String charact = Character.toString(phrase.charAt(i));


      if (foundLetters != null){
        if (foundLetters.contains(charact)){
          s += charact;
        }else if (phrase.charAt(i) == ' '){
          s += "&nbsp;&nbsp;";
        }else {
          s += " _ ";
        }
      }else if (phrase.charAt(i) == ' '){
        s += "&nbsp;&nbsp;";
      }else {
        s += " _ ";
      }

    }
    return s;
  }



  /**
   * @return boolean SUBJECT TO CHANGE
   * AS THE GUI CHANGES EVERYTHING
   * */
  public boolean playHangman(){



    //test.add(img);


    instructionsTxt.setText("<html>You are almost there! All you have to do now is win this game by guessing the"+
                       " correct letters in the phrase and without getting more than 7 letters wrong."+
                       " Your boo's life hangs by a thread. Hurry" +
                       " up and save him!</html>");
    test.add(instructionsTxt);


    gameTxt.setText("<html><font size = '5' face ='Courier New'>"+ gueDisplay +"</font></html>");
    test.add(gameTxt);


    question.setText("What is your guess?");
    test.add(question);

    currentGuesses.setText("<html>Guesses So Far : "+ guesses+"</html>");
    test.add(currentGuesses);

    errorsLeft.setText("<html>Errors Left : "+ errors +"</html>");
    test.add(errorsLeft);


    test.add(incorrectInput);


    frame.add(test);
    frame.setVisible(true);

    eventBox eveB = new eventBox();
    letterInput.addActionListener(eveB);
    test.add(letterInput);




    return wonLost;
  }

  /**
   * ActionListener where most of the game goes on
   * Game continues on as the user continues
   * and user has the liberty to stop or not at any time
   *
   * */
  public class eventBox implements ActionListener{
    public void actionPerformed(ActionEvent eveB) {

      System.out.println("enters the action");
      String input = letterInput.getText().toLowerCase();


      if(guesses.contains(input) || !Character.isLetter(input.charAt(0)) || input.
length() > 1 || input.charAt(0) == ' ' ){
        incorrectInput.setText("<html>*** Invalid input. Make sure to check that
the input is"+
                               "one letter and has not been used before.</html>");
      }else{
        incorrectInput.setText("");
        guesses.add(input);

        if(uniqueLetters.contains(input)){
          correctGuesses.add(input);
          gueDisplay = displayDisguise(correctGuesses);
        }else{
          errors--;
        }

        //img.setIcon(new ImageIcon(hangmanGraphics(0)));
        gameTxt.setText("<html><font size = '5' face ='Courier New'>"+ gueDisplay
+"</font></html>");
        currentGuesses.setText("<html>Guesses So Far : "+ guesses+"</html>");
        errorsLeft.setText("<html>Errors Left : "+ errors +"</html>");

      }

         if(gueDisplay.equals(phrase)){
           wonLost = true;
           //change icon to a winner thing??

           //i dont think this one is needed???------------------------
         }else if(errors == 0){
           //have a you lose sorrry text?
           System.out.println("you lose");

         }

         frame.getContentPane().validate();
         frame.getContentPane().repaint();

         }

  }


  public static void main(String[] args){
   // System.out.println(Character.isLetter(""));
    //Character.isLetter(gue.charAt(0))

    THangmanGui x = new THangmanGui("Hello World");

    System.out.println(x.playHangman());
  }
}
```