```java
/**
 * Hangman.java
 * for Adventure Time
 * Created by: Jessenia Aguilar-Hernandez
 *
 * Mimics a game of hangman. This class is used in the final room of the game map.
 * Beating this minigame allows the user to save the don in distress and win the game.
 * Constructor reads in a number of phrases from a text file and randomly chooses one
 * for this game.
 */
import java.io.*;
import java.util.*;

public class Hangman{
  //instance variable
  private String phrase;

  //takes in a file of hangman phrases
  //chooses random phrase each time the constructor is called
  public Hangman(String fileIn) {
    phrase = "";

    try {
      Scanner sc = new Scanner(new File(fileIn));
      String[] phrases = new String[10];
      int count = 0;
      while(sc.hasNextLine()) {
        phrases[count++] = sc.nextLine().toLowerCase().trim();
      }

      int ran = (int)(Math.random()*count);
      phrase = phrases[ran];
      sc.close();
    } catch (IOException ex) {
      System.out.println("error in reading hangman phrases file");
    }
  }

  /**
   * can be removed if we have graphics take the place but meanwhile it will show these little figures
   * @return Returns a graphic according to how many mistakes the player has made
   * */
  public String hangmanGraphics(int w){
    String[] hG = {
      "\n"+
      "           \n" +
      "┼────────|\n"+
      "|  \n"+
      "|  \n"+
      "|  \n"+
      "|  \n" ,

      "\n"+"_________\n"+
      "|      |\n" +
      "|      0 \n"+
      "|  \n"+
      "|\n"+
      "|\n" ,

      "\n"+"__________   \n"+
      "|      |\n" +
      "|      0\n"+
      "|     /\n"+
      "|\n"+
      "|\n",

      "\n"+"__________ \n" +
      "|      |\n"+
      "|      0\n"+
      "|     /|\n"+
      "|\n"+
      "|\n",

      "\n"+"__________\n"+
      "|      |\n"+
      "|      0\n"+
      "|     /|\\ \n" +
      "|  \n"+
      "|  \n" ,

      "\n"+"__________\n"+
      "|      |\n"+
      "|      0\n"+
      "|     /|\\ \n"+
      "|     / \n"+
      "|\n" ,

      "\n"+"__________\n"+
      "|      |\n"+
      "|      0\n"+
      "|     /|\\ \n"+
      "|     / \\ \n"+
      "| \n"+
      "GAME OVER!\n"



    };
    return (hG[w]);
  }


  /**
   * @return Returns a new Linked List of each unique letter in the phrase
   * */
  public LinkedList<String> makeUniqueLetters(){

    LinkedList<String> uqLetters = new LinkedList<String>();


    //makes sure no spaces are considered a character that needs to be guesses
    String temp = phrase.replace(" " ,"");

    int i = 0;


    while (temp.length() > 0 ){

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

      uqLetters.add(ch);

      //gets rid of all characters that equal the first character
      //so that there will not be duplicates
      temp = temp.replace(ch ,"");

      i++;
    }
    return uqLetters;
  }

  /**
   * @return Returns a string that again is for aesthetic purposes that can be
changed in gui
   * string represents the letters that have been guessed correctly and letters
   * that need to be guessed still
   * */
  public String displayDisguise(LinkedList<String> foundLetters){
    String s = "";


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

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

      if (foundLetters.contains(charact)){
        s += charact;

      }else if (phrase.charAt(i) == ' '){
        s += " ";
      }else{
        s += "_";
      }
    }

    return s;
  }


  /**
   *
   * @return Returns the boolean so if at the end of the game if the person did not
pass
   * they can not enter the final room and will return true if they have and so the
   * will be able to finally enter
   * */
  public boolean playHangman(){

    int i  = 0;

    boolean wonLost = false;
    LinkedList<String> uniqueLetters = makeUniqueLetters();
    LinkedList<String> guesses = new LinkedList<String>();
    LinkedList<String> correctGuesses = new LinkedList<String>();
    String gueDisplay = displayDisguise(guesses);

    System.out.println("You are almost there! All you have to do now is win this
game by guessing the"+
                       " correct letters in the phrase"+
                       " Your boo's life hangs on a thread. Hurry" +
                       " up and save him!");


    //prints out the aesthetics
    System.out.println(hangmanGraphics(i));
    System.out.println(gueDisplay);

    //7 is the number of failed guesses they are allowed to have
    while (i < 7){

      System.out.println("What is your guess? ");
      Scanner sc = new Scanner(System.in);
      String gue = sc.next();

      //stops user from inputing an already guesses letter, other character or more than one character
      if (guesses.contains(gue.toLowerCase()) || !Character.isLetter(gue.charAt(0)) || gue.length() > 1){
        System.out.println("*** Invalid input. Make sure to check that the input is one letter and has not been used before.");
        continue;
      }else{
        gue = gue.toLowerCase();
        guesses.add(gue);

        //checks if the inputed char is in the phrase and updates aesthetics
        if(uniqueLetters.contains(gue)){
          correctGuesses.add(gue);
          gueDisplay = displayDisguise(correctGuesses);
        }else{
          i++;
        }
        //prints updated aesthetics
        System.out.println(hangmanGraphics(i));
        System.out.println(gueDisplay);
        System.out.println("Guesses :"+ guesses);
      }


      if(gueDisplay.equals(phrase)){
        wonLost = true;
        sc.close();
        break;
        //i dont think this one is needed???------------------------
      }else if(i == 6){
        sc.close();
        break;

      }

    }

    return wonLost;
  }


  public static void main(String[] args){
    Hangman ugh = new Hangman("hello world");
    /*
     ugh.makeUniqueLetters();

     ugh.displayDisguise();

     ugh.addGuess("a");
     ugh.addGuess("b");
     ugh.addGuess("c");
     ugh.addGuess("d");

     ugh.displayDisguise();
     ugh.addGuess("d");
     ugh.addGuess("o");
     ugh.addGuess("l");
     System.out.println("\nSecond time\n");
     ugh.displayDisguise();
     **/
    //hangmanGraphics();
    System.out.println(ugh.playHangman());
  }
}
```