public class Hangman {

private final String[] wordBank = {"Faceless", "Abyssal", "Magus", "Stroke",// The word bank the game is going to use

"Bounty", "Siren", "Templar", "Oracle", "Phantom",

"Enchantress", "Assassin", "Wyvern"} ;

private String[] letters = {"", "", "", "", "", "", "", "", // An array to store the users guessed letters

"", "", "", "", "", "", "", "",

"", "", "", "", "", "", "", "",} ;

private final Random rng = new Random() ; // The games random number generator

private int rdm = rng.nextInt(wordBank.length) ; // The random number generated to use to pick a word from the bank

private String wordPicked = wordBank[rdm] ; // The word the user is going to guess

private int wordlen = wordPicked.length() ; // Length of selected word

private String[] chopped = new String[wordlen] ; // An array for the word selected

private String[] mask = Arrays.copyOf(chopped, chopped.length) ; // An array for masking the selected word

private String masked ; // The string for the masked word

private int guesses = 8 ; // The number of guesses remaining

private int letCntr = 0 ; // A counter that counts how many letters the user has guessed

public Hangman() {

for(int i = 0 ; i < wordlen ; i++) {

chopped[i] = wordPicked.substring(i,i+1) ;

}

for(int i = 0 ; i < wordlen ; i++) {

mask[i] = "\*" ;

}

masked = String.join("", mask) ;

}

/\*\*

\* Shows if word contains the given input

\* @param input the letter guessed by the user

\* @return masked progress of word

\*/

public String revealWord(String input) {

for(int i = 0; i < wordlen ; i++) {

if(chopped[i].toLowerCase().equals(input.toLowerCase())) {

mask[i] = chopped[i] ;

}

}

masked = String.join("", mask) ;

return masked ;

}

/\*\*

\* Checks if word contains the given input

\* @param input the letter guessed by the user

\* @return returns a flag value for users input

\*/

public int checkGuess(String input) {

int flag = 0 ;

for(int i = 0 ; i < wordlen ; i++) {

if(chopped[i].toLowerCase().equals(input.toLowerCase())) {

flag++ ;

}

}

return flag ;

}

/\*\*

\* Reduces users remaining guesses if word does contain given input

\* @param input the letter guessed by the user

\*/

public void flagGuess(String input) {

int flag = 0 ;

for(int i = 0 ; i < wordlen ; i++) {

if(chopped[i].toLowerCase().equals(input.toLowerCase())) {

flag++ ;

}

}

if(flag < 1){

guesses-- ;

}

}

/\*\*

\* Adds the letter guessed by the user to letters Array

\* @param input the letter guessed by the user

\*/

public void addLetter(String input) {

letters[letCntr] = input ;

letCntr++ ;

}

/\*\*

\* Shows the letters guessed by the user

\* @return a string of all guessed letters by the user

\*/

public String showGuesses() {

String letGuess = String.join("", letters) ;

return letGuess ;

}

/\*\*

\* Removes remaining "\*" in the masked word

\* @return the unmasked word

\*/

public String unMasked() {

return masked = wordPicked ;

}

/\*\*

\* Checks if the word still contains "\*"

\* @return boolean value if word is still masked

\*/

public boolean hasMask(){

return masked.contains("\*") ;

}

/\*\*

\* Shows the word the user has to guess

\* @return string value of the word the user has to guess

\*/

public String getWord() {

return wordPicked ;

}

/\*\*

\* Shows the number of guesses remaining

\* @return int value of the remaining guess of the user

\*/

public int getGuesses() {

return guesses ;

}

/\*\*

\* Shows the masked word the user has to guess

\* @return string value of the masked word the user has to guess

\*/

public String showProgress() {

return masked ;

}

}

public class Tester {

/\*\*

\* A test class for the Hangman file

\*/

public static void main(String[] args) {

boolean status = true ;

Scanner scan = new Scanner(System.in) ;

do{

Hangman hangman = new Hangman() ; // Selects a word from the word bank, masks them and

// Display the masked word to the user for them to guess

System.out.println("Welcome to Hangman!\nWhere my codes are made "

+ "up and the points don't matter\n\nPlease guess "

+ "the following word: " + hangman.showProgress() +"\nYou"

+ " have 8 guesses left.") ;

while(hangman.hasMask()){

String input = scan.nextLine() ;

if(input.length() > 1) { // If user inputs more than one letter the program

System.out.println("Please enter only one letter") ; // Program will ask user to submit only one letter

input = scan.nextLine() ;

}

if(hangman.checkGuess(input) == 0) { // If the selected word does not contain users input

if(!hangman.showGuesses().contains(input)) { // Program will flag guess and see if the user has

hangman.addLetter(input) ; // Already guessed that letter. If user has not guessed

hangman.flagGuess(input) ; // That letter, the program will reduce the remaining

System.out.println("The letter does not exist in this " // Number of guesses and add it to the array of letters

+ "word") ; // The user has guessed

System.out.println("Letters you've guessed: " +

hangman.showGuesses()) ;

if(hangman.getGuesses() > 0) {

System.out.println("You have " + hangman.getGuesses()

+ " guesses left.") ;

}

System.out.println("\n" + hangman.showProgress()) ;

}

else if(hangman.showGuesses().contains(input)) {

System.out.println("You've already guessed this "

+ "letter.") ;

System.out.println(hangman.showGuesses()) ;

System.out.println("Letters you've guessed: "

+ hangman.showGuesses()) ;

System.out.println("\n" + hangman.revealWord(input)) ;

}

if(hangman.getGuesses() == 0) { // If user has no guesses remaining, the program will

hangman.unMasked() ; // Reveal the word and end the game

System.out.println("You have run out of guesses.\n"

+ "The word was \"" + hangman.getWord() +"\".\n"

+ "Better luck next time!") ;

}

}

else if(hangman.checkGuess(input) > 0) { // If the selected word does contain users input the

if(!hangman.showGuesses().contains(input)) { // Program will see if the user has already guessed the

hangman.addLetter(input) ; // Letter. If user has not guessed that letter, the

System.out.println("Great! You've guessed " + // Program will unmasked that letter from the word and

hangman.checkGuess(input) + " letters.") ; // Add the word the the array of letters the user has

System.out.println("Letters you've guessed: " // Guessed

+ hangman.showGuesses()) ;

System.out.println("\n"+ hangman.revealWord(input)) ;

}

else if(hangman.showGuesses().contains(input)) {

System.out.println("You've already guessed this "

+ "letter") ;

System.out.println("\n" + hangman.revealWord(input)) ;

}

if(!hangman.hasMask() && hangman.getGuesses() > 0) { // If the user has remaining guesses and has unmasked the

System.out.println("\nCongratulations! You won!"); // Word. The program will congratulate the user and end

} // The game

}

else{

System.out.println("Please enter a letter.");

}

}

System.out.println("Do you want to play again?\nEnter:\n0 - " // If the game has ended the program will as the user if

+ "YES\n1 - NO") ; // They want to play another game

int playAgain = scan.nextInt() ;

if(playAgain == 1) {

status = false ;

}

}while(status) ;

System.out.println("Thank you for playing!");

}

}