

“Wordle”

Dominique McDevitt, BS in Computer Engineering, 2022
Dr. Joseph T. Wunderlich Ph.D.

OPPORTUNITY

This project allows for an opportunity to attempt to recreate a game that is currently very popular in a unique way.

DESIGN SPECIFICATIONS

The JFrame interface has a set of five (5) radio buttons used to choose between the different game-types. Once one of those buttons is clicked, the interface changes, adding a “Guess” button underneath the radio buttons and a section to inform the player of how many guesses they have remaining. Underneath that are two (2) text areas: the topmost one is used to collect the player’s input while the bottom one displays previous guesses. Between the text areas is a line that displays the code for the game (Figure 1), which is meant to help the player get to the correct word. At the bottom of the interface, there is a “New Game” button, which is disabled until the player finishes their current game.

Code: \$ = correct, ? = wrong spot, - = wrong

Figure 1: Code displayed on interface

TECHNICAL DESIGN

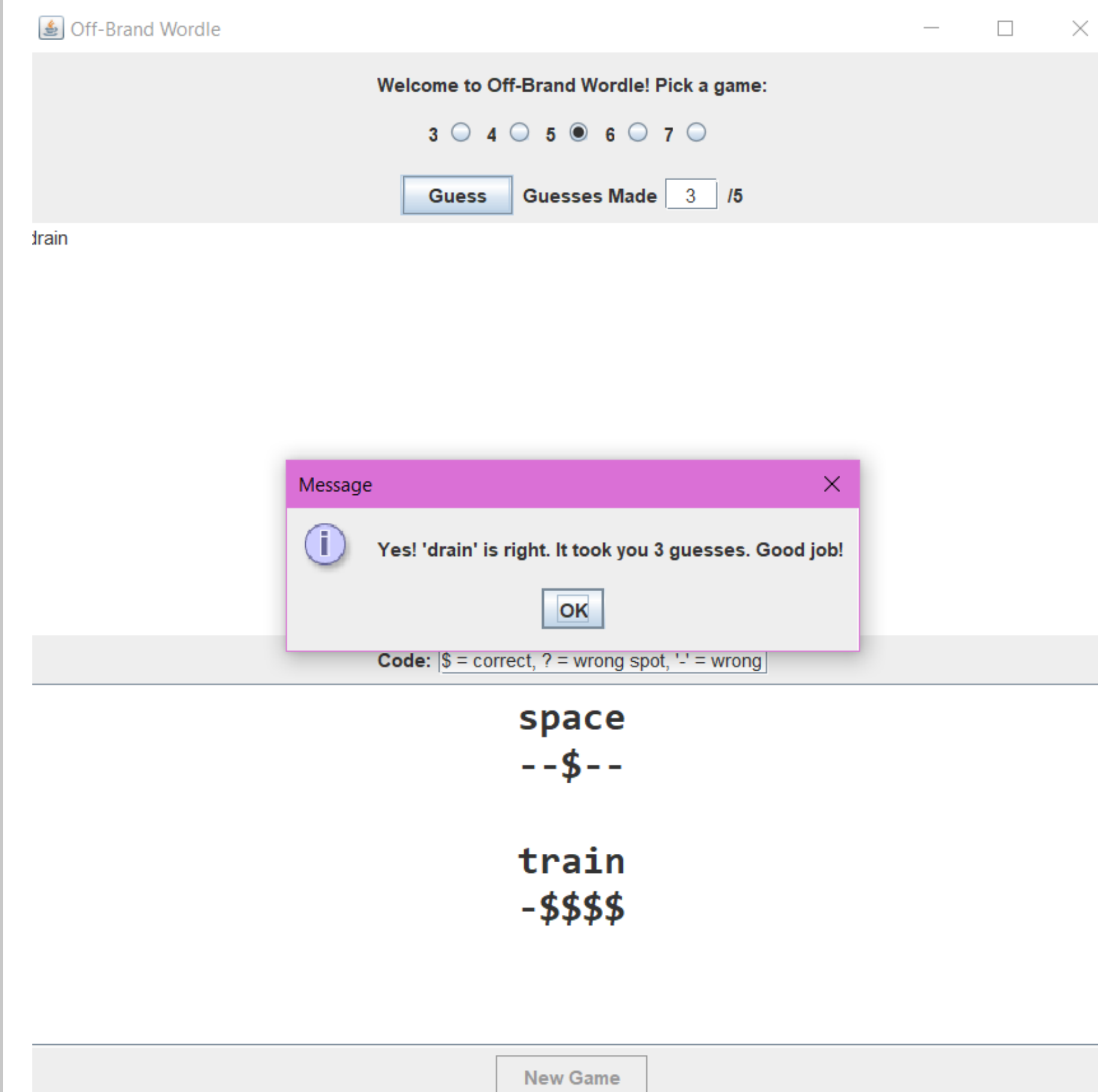


Figure 4: Winning a five-letter game

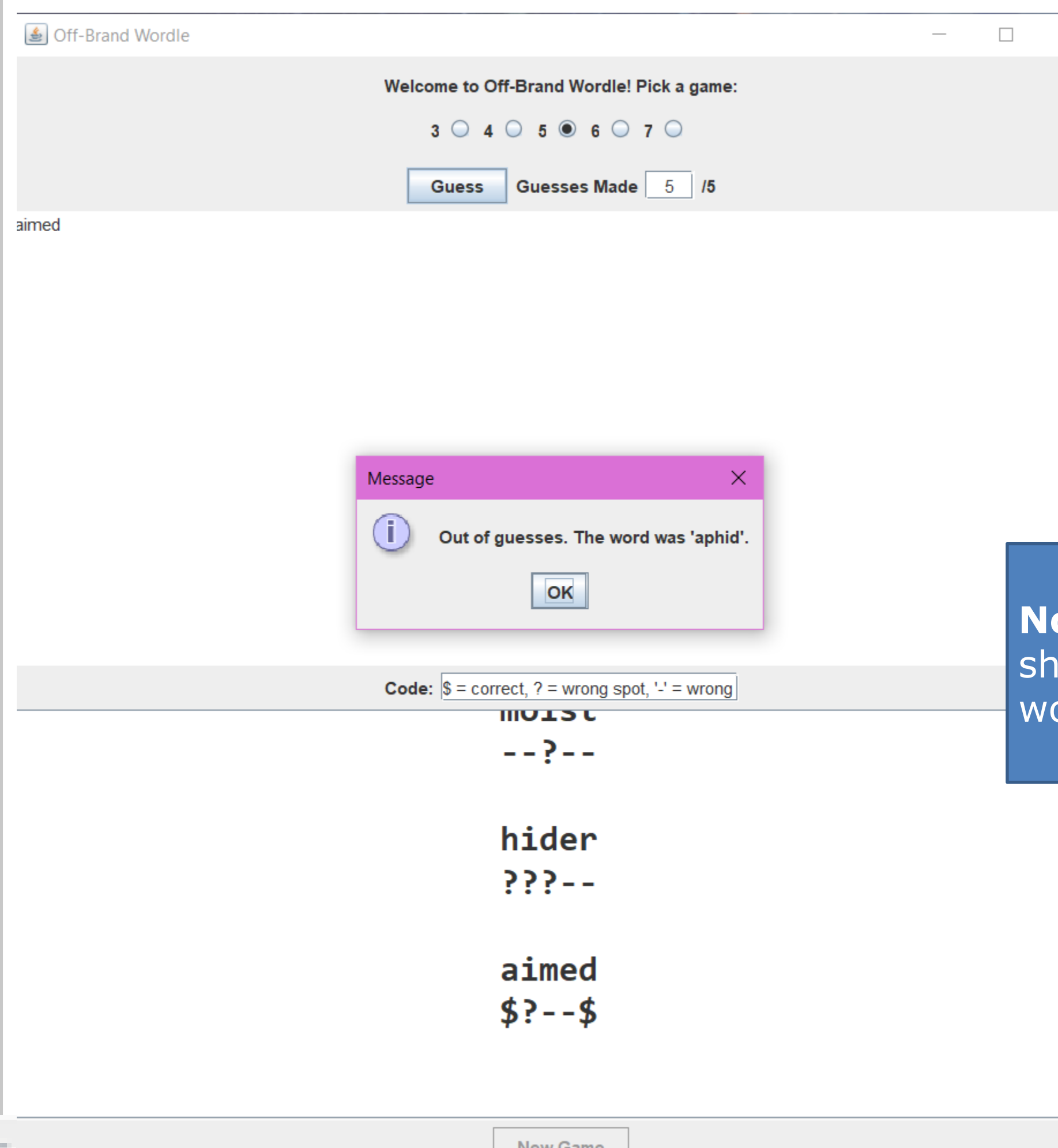


Figure 5: Losing a five-letter game

CODE

There are two (2) “dictionaries” utilized in this game: one for analyzing each of the player’s guesses and one for choosing a random answer that the player is trying to guess. The guess dictionary is made from a long list of uncommon English words [1] while the answer dictionary is a shorter list of more common words [2]. To make the code easier to read, these word lists are read in before the player even chooses which variation they would like to play. The player is given a choice between five (5) game-types to guess three-, four-, five-, six-, and seven-letter words. After the player has specified which game they’re playing, a word of that length is picked from the answer list randomly. Then the player inputs their guess which is checked against the guess list to ensure it is a real word that matches the length of the chosen answer (Figure 2). The guess is then checked against the answer for correct letters and is output along with a code to tell the player how accurate their guess was (Figure 3). The process continues until either the player guesses the word or runs out of guesses. Depending on the game-type chosen determines how many guesses are allowed. The five-letter game, for example, allows five (5) guesses while the three-letter game only allows three (3).

```
1031 public boolean checkWord(ArrayList<String> words, String w, int wordlength) {
1032     boolean done=false;
1033     boolean result=false;
1034     int begin,end, mid;
1035
1036     if(w.length()!=wordlength) { //immediately return if the word length isn't right
1037         return false;
1038     }
1039
1040     begin=0;
1041     end=words.size()-1;
1042     while(!done) {
1043         mid=(begin+end)/2; //split the list in half and look at the word in the middle
1044         if(w.compareTo(words.get(mid))<0) { //if the guess looks like it comes before the middle, continue searching the beginning half of the list
1045             end=mid;
1046         }
1047         else {
1048             begin=mid; //if the guess looks like it comes after the middle, continue searching the end half of the list
1049         }
1050         if(w.compareTo(words.get(begin))==0 || w.compareTo(words.get(end))==0) { //if the guess is found in the list, it's valid
1051             done=true;
1052             result=true;
1053         }
1054         else if(begin==end-1) { //if the guess isn't found in the list, it's invalid
1055             done=true;
1056             result=false;
1057         }
1058     }
1059     return result;
1060 }
1061
```

Figure 2: checkWord() function

Note: Only the five-letter game is shown for clarity. The other game-types work the same way.

```
1062 public String answerCode(String guess, String answer) throws BadLocationException {
1063     String code="";
1064     String answer2=answer;
1065     for(int i=0;i<answer.length();i++) { //create blank answer code
1066         code+="-";
1067     }
1068     for(int i=0;i<guess.length();i++) { //go through guess letter by letter
1069         if(guess.charAt(i)==answer.charAt(i)) {
1070             //if current guess letter is the same as the current answer letter -> correct letter in correct spot
1071             code=insertChar(code,'$',i);
1072             answer2=insertChar(answer2,'X',i); //replace with random filler so not to double check the same letter
1073         }
1074     }
1075     for(int i=0;i<guess.length();i++) { //go through guess letter by letter
1076         for(int j=0;j<answer.length();j++) { //go through answer letter by letter
1077             if(guess.charAt(i)==answer2.charAt(j)) {
1078                 //if current guess letter is the same as different answer letter -> correct letter in wrong spot
1079                 code=insertChar(code,'?',j);
1080                 answer2=insertChar(answer2,'X',j); //replace with random filler so not to double check the same letter
1081             }
1082         }
1083     }
1084     return code;
1085 }
1086
1087 public String insertChar(String str, char ch, int position) {
1088     return str.substring(0,position)+ch+str.substring(position+1,str.length());
1089 }
1090
```

Figure 3: answerCode() function

TESTING

There are still a few improvements that could be made to the program. For example, after finishing a game the “New Game” button is enabled and is meant to clear the previous game and start a fresh one of the same type unless the player chooses a different variation via the radio buttons. While the “New Game” button clears the screen, there is a slight glitch wherein the “Guess” button temporarily disappears and the “Guesses Made” section appears out of order. When the player begins guessing again, a few pop-ups will occur, suggesting errors that shouldn’t occur. Similar glitches occur when switching to the other game-types, i.e., when switching from the three-letter game to the five-letter game, the “Guesses Made” section will continue to read as “0/3” instead of “0/5”.

IMPACT
This project is meant to create a fun, relaxing experience through an interpretation of the popular word-guessing game, Wordle.

REFERENCES

- [1] Dwyl, “Dwyl/English-words: A text file containing 479k English words for all your dictionary/word-based projects e.g: Auto-completion / autosuggestion,” *GitHub*. [Online]. Available: <https://github.com/dwyl/english-words/>. [Accessed: 20-Feb-2022].
- [2] “WordList,” *Wordlist-Wolfram Language Documentation*, 2015. [Online]. Available: <https://reference.wolfram.com/language/ref/WordList.html>. [Accessed: 28-Mar-2022].