

```
import java.util.ArrayList;
import acm.program.*;

public class Bogglish extends ConsoleProgram
{
    private static final String ALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

    public void run()
    {
        setFont("*-*-24");
        String[][] board = new String[5][5];
        // version 0.1, 0.2, and 0.5 here
        String letters = ALPHABET;
        for(int r=0; r<board.length; r++){
            for(int c=0; c<board[0].length; c++){
                int index = (int)(Math.random() * letters.length());
                board[r][c] = letters.substring(index, index+1);
                letters = letters.substring(0, index) + letters.substring(index+1);
            }
        }
        printBoard(board);
        while(true){
            String word = readLine("Enter a word: ");
            if(word.length() == 0) break;
            if(checkWord(word, board)) println("Yes, " + word + " is on the board.");
            else println("No, " + word + " is Not on the board.");
        }
        printAllWords(board);
    }

    public void printAllWords(String[][] board){
        ArrayList<String> words = Dictionary.getAllWords();
        for(String word : words){
            if(checkWord(word, board)) println(word);
        }
    }

    public void printBoard(String[][] board){
        for(int r=0; r<board.length; r++){
            for(int c=0; c<board[0].length; c++){
                print(board[r][c]);
                print(" ");
            }
            println();
        }
    }

    /** Returns true if the word is "contained" somewhere in the board,
     *  following our "Bogglish" rules, false otherwise.
     *  @param word the word to check
     *  @param board the Bogglish board
     *  @return true if the word is in the board, false otherwise.
     */
    private boolean checkWord(String word, String[][] board)
    {
        if(word.length() < 3) return false;
    }
}
```

```
for(int i=0; i< word.length()-1; i++){
    String l1 = word.substring(i, i+1);
    String l2 = word.substring(i+1, i+2);

    Position l1pos = getLetterPosition(l1, board);
    Position l2pos = getLetterPosition(l2, board);

    if(l1pos == null || l2pos == null) return false;

    if(Math.abs(l1pos.getRow()-l2pos.getRow()) > 1 || Math.abs(l1pos.getCol() -
l2pos.getCol()) > 1){
        return false;
    }
}
return true;
}

/** Returns the Position of where the letter is in the board
 * Note that this returns a Position, which is a class you wrote.
 * @param letter a String of length 1 representing the letter
 * @param board the Bogglish board
 * @return the Position of the letter, or null if it is not there
 */
private Position getLetterPosition(String letter, String[][] board)
{
    for(int r=0; r<board.length; r++){
        for(int c=0; c<board[0].length; c++){
            if(board[r][c].equalsIgnoreCase(letter)){
                Position p = new Position(r, c);
                return p;
            }
        }
    }
    return null;
}
}
```