Kolby, Nischal, Quinn

1. Introduction

The Test Plan includes the use of Black box testing in order to test all the system requirements of our application which includes testing of the start game, validity of letters chosen, validity of words, scoring of the words based on their length and the UI aspects of the game such as the submit button and other functional requirements that can be traced back to the Requirement Analysis Document.

2. System Overview

The overview of our system consists of an easy to use single player boggle game digitally which implements Swing framework for GUI consisting of a 4x4 matrix and viable dice letters provided to us. The players can select letters which is then highlighted and checked for validity and when the player submits the word, the word is checked for validity against a local dictionary file which is then used to calculate the score for that given word. The process can be repeated for 3 minutes after the game has been started.

3. Features to be Tested

Testing Requirement	System Requirement	Short Description
1	F1	Player starts a game
2	F2, N3	Players clicks on the board/ button and receives feedback
3	F3	Player crosses their path while creating a word
4	F3	Player enters an invalid word
5	F3	Player enters a valid word
6	F4	Player tries to submit a word after the timer has run out
7	F5, N2	Player enters a 2 or 1 letter word and is awarded 0 points
8	F5, N2	Player enters a 3 or 4 letter word and is awarded 1 point
9	F5, N2	Player enters a 5 letter word and is awarded 2 points
10	F5, N2	Player enters a 6 letter word and is awarded 3 points
11	F5, N2	Player enters a 7 letter word and is awarded 5 points
12	F5, N2	Player enters an 8 or more letter word and is awarded 11 points
13	F6	Player submits a word that has already been submitted
14	F6	Player submits a word that has not already been submitted

15	F7	Player clicks start game
16	N1	Have a new person play the game and time how long it takes to understand and play the game
17	N5	Run Boggle by Team 12 on Windows
18	N5	Run Boggle by Team 12 on Mac OS
19	F9	Player chooses a language
20	F8	Displays correct dice when choosing a language
21	F8	Selects the correct dictionary file for the language selected
22	F8	Awards correct amount of points for words in German

4. Test Environment

- To run the tests you need to have a computer that has Java installed.
- 1080p screen is recommended but not required (makes the game look better)

5. Test Cases

Test Case 1

Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 5, 8, 14

Initial conditions: n/a

Input: a valid 4 letter word

Output: the 4 letter word appearing in the list on the left and points going up by 1

Test Case 2

Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 5, 9, 14

Initial conditions: n/a

Input: a valid 5 letter word

Output: the 5 letter word appearing in the list on the left and points going up by 2

Test Case 3

Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 5, 10, 14, 18

Initial conditions: n/a

Input: a valid 6 letter word

Output: the 6 letter word appearing in the list on the left and points going up by 3

Test Case 4

Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 5, 11, 14, 17

Initial conditions: n/a

Input: a valid 7 letter word

Output: the 7 letter word appearing in the list on the left and points going up by 5

Test Case 5

Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 5, 12, 14, 15

Initial conditions: n/a

Input: a valid 8 or more letter word

Output: the 8 or more letter word appearing in the list on the left and points going up by 11

Test Case 6

Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 5, 7

Initial conditions: n/a

Input: a valid 2 or less lettered word

Output: clears the highlighting of the board and does not display the word in the list on the left and also does not add points to the score

Test Case 7

Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 3
Initial conditions: n/a

Input: a valid 4 letter word that crosses its path

Output: clears the highlighting of the board and does not display the word in the list on the left and also does not add points to the score

Test Case 8

Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 4
Initial conditions: n/a

Input: an invalid 4 letter word

Output: clears the highlighting of the board and does not display the word in the list on the left and also does not add points to the score

Test Case 9

Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 6

Initial conditions: n/a

Input: a valid 3 letter word after the timer has expired

Output: does nothing

Test Case 10

Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 13

Initial conditions: n/a

Input: a valid 4 letter word that has been submitted already

Output: clears the highlighting of the board and does not display the word in the list on the left again and also does not add points to the score

Test Case 11
Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 16

Initial conditions: player does not know how to play the game

Input: player learns how to play the game within a minute and inputs valid words more than 2 letters

Output: the entered valid word appears in the list on the left and points going up by the given rules

Test Case 12
Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 19, 22

Initial conditions: Player selects the German option

Input: a valid 5 letter german word that has not been entered yet

Output: the entered valid word appears in the list on the left and points going up by the given rules

Test Case 13
Component under test: Boggle by Team 12

Feature(s) to be tested: 1, 2, 19, 20, 21, 22

Initial conditions: Player selects the German option

Input: a valid 3 letter german word that has not been entered yet

Output: the dice show the correct german dice and the entered valid word appears in the list on the left and points going up by the given rules