

# **Software Engineering Group Projects JoggleCube Game Requirements Specification**

Author:	C. J. Price
Config Ref:	SE.QA.CSRS
Date:	23rd January 2018
Version:	1.0
Status:	Release

Department of Computer Science  
Aberystwyth University  
Aberystwyth  
Ceredigion  
SY23 3DB  
Copyright © of Aberystwyth University 2018

## CONTENTS

1.	INTRODUCTION .....	3
1.1	Purpose of this Document.....	3
1.2	Scope.....	3
1.3	Objectives .....	3
2.	GENERAL DESCRIPTION .....	4
2.1	Product Perspective.....	4
2.2	Product Functions .....	4
2.3	User Characteristics .....	4
3.	SPECIFIC REQUIREMENTS .....	5
3.1	Functional Requirements .....	5
3.2	External Interface Requirements.....	6
3.3	Performance Requirements .....	6
3.4	Design Constraints .....	6
3.5	Other Requirements .....	6
	Appendix A - Letter Population.....	7
	Appendix B - Adjacency in the cube .....	7

# **1. INTRODUCTION**

## **1.1 Purpose of this Document**

This document describes the requirements for the CS22120 Software Engineering Group Project 2018. It should be read in the context of the Group Project, taking into account the details of the group project assignment and the group project Quality Assurance (QA) Plan [1].

## **1.2 Scope**

This requirements specification describes the functions needed to provide the computer game “JoggleCube”, and the attributes that are expected from the finished product. It also describes the requirements for the process of constructing the system.

## **1.3 Objectives**

The objectives of this document are:

- To describe the background to the CS22120 group project application for 2018 (JoggleCube)
- To provide details of the criteria that the group project product must meet
- To describe the types of interaction with the system which must be supported

## **2. GENERAL DESCRIPTION**

### **2.1 Product Perspective**

JoggleCube is a turn-based computer game for one player. The player is given a 3 by 3 by 3 grid of randomly generated letters, and has to compile a list of words of one or more letters that can be generated by moving around the grid one letter at a time either horizontally, vertically or diagonally, without reusing any of the available letters.

### **2.2 Product Functions**

The product will provide the following features:

- Appropriate generation of random grids
- Enabling the player to indicate words found in the grid
- Deciding whether an indicated word is legal or not
- Keeping a list of legal words that the player has already indicated
- Maintenance of a countdown timer that stops the game on reaching zero
- Scoring of the player's solution to a grid
- Saving of a grid and a score for other players to play against
- Start and finish displays for game
- Maintenance and display of high score table

### **2.3 User Characteristics**

The software will be used by second year Computer Science students. These users are familiar with standard software tools, and with WIMP software. They are, by default, quite lazy, and so the software should provide the indicated features with the fewest possible mouse movements and keystrokes.

## 3. SPECIFIC REQUIREMENTS

### 3.1 Functional Requirements

#### *FR1 Start up conditions*

When the software starts, it will display a start screen for the game, and give the user the choice of starting a new grid, or competing against a past saved grid.

#### *FR2 Generating new grid*

If the user chooses to generate a new grid, then the software will create a new Joggle board, with the following characteristics:

- It will generate a 3 by 3 by 3 grid of letters.
- The letters will be randomly chosen among the full population of letters given in Appendix A.
- Once selected, the letter should not be available for reselection (so no board can have more than 1 Z or 9 A's).

#### *FR3 Loading a past grid*

If the user chooses to compete against a past grid, then they should be given a choice of past grids to compete against. Having selected one, it should be loaded, and the previous scores for other people should be displayed against their names.

#### *FR4 Timing a game*

As soon as a grid is displayed, the game clock should be started, counting down from three minutes. The remaining time should be displayed on the screen. After the three minutes is completed, no further input from the user is accepted, and the score achieved is noted.

#### *FR5 Actions on game completion*

After completion of a JoggleCube game, the user should be able to save the game, start a new game, or load an existing game. If the grid was a new one, and their score was one of the highest ten scores ever achieved, then their name and high score are recorded in the high score table.

#### *FR6 Saving a grid*

If the user chooses to save a new grid for other people to compete against, then they should be prompted for their name, and it should be saved against their score, and they should be prompted for a filename to save the game in. If the grid was originally loaded (i.e. other people have already played it), then they should be prompted for their name, and it should be saved against their score in an ordered table of names and scores (highest score at the top) and the details should be saved in the same filename that they were loaded from.

#### *FR7 Displaying the grid*

The letters in the grid make up a 27 letter cube. This should be displayed to the user as three groups of 9 letters. The user should be able to change the view as if the cube was seen from any of its faces.

#### *FR8 Enabling the user to enter words*

The user should be able to type in words or select them on the grid with the mouse. When they have completed a word (completion needs to be indicated somehow), then it is checked against an English dictionary, and added to a list of words entered. If it is a legal word, then it is given an associated score. If it is incorrect, it is given a score of zero.

#### *FR9 Legal words*

Legal words are composed of a series of letters that can be obtained by starting at any letter on the grid, and repeatedly moving to an adjacent letter on the grid that has not already been traversed, until the word is completed. Adjacency is defined in appendix B. Words of any number of letters greater than zero are allowed. Each word is only counted once, even if it can be composed several different ways from the JoggleCube. Legal words are defined by the dictionary that you use. You should be using a dictionary of at least 50,000 English words.

#### *FR10 Scoring words*

The score for a legal word is the square of the Scrabble score for that word.

Examples:

Word	Scrabble score for word	Game score for word
“MY”	7	49
“TIE”	3	9
“JAWS”	14	196

#### *FR11 Scoring the game*

The total score for the game is calculated by adding up all of scores for the individual words.

### **3.2 External Interface Requirements**

#### *EIR1 Appearance of Interface*

The program should conform to Java cross platform Look and Feel Guidelines. It is recommended that all groups read through the first two chapters of the Guidelines produced by Oracle, (<http://www.oracle.com/technetwork/java/index-136139.html>)

### **3.3 Performance Requirements**

#### *PR1 Response of program to user input*

Any user input should be appropriately reflected on the screen within one second.

#### *PR2 Target computer for system*

All software produced should run correctly on standard Windows PCs within the Department of Computer Science.

### **3.4 Design Constraints**

#### *DC1 Use of JAVA*

It is corporate policy to use Java on all major new developments, and so Java will be used for all coding on this project.

#### *DC2 Reuse of existing software*

Use of existing classes for basic data structures, such as graphs, is encouraged, as is full use of the standard Java libraries. Students should declare in their final report the use of any classes other than these that were not developed on the project. If there is doubt about the eligibility of any classes for use on the project, they should check with the Project Manager. Undocumented use of outside classes will be severely penalised.

### **3.5 Other Requirements**

The project will be developed in line with the group project QA plan, detailed in [1].

## Appendix A - Letter Population

The letter population is given in the following table, along with the Scrabble value for each letter.

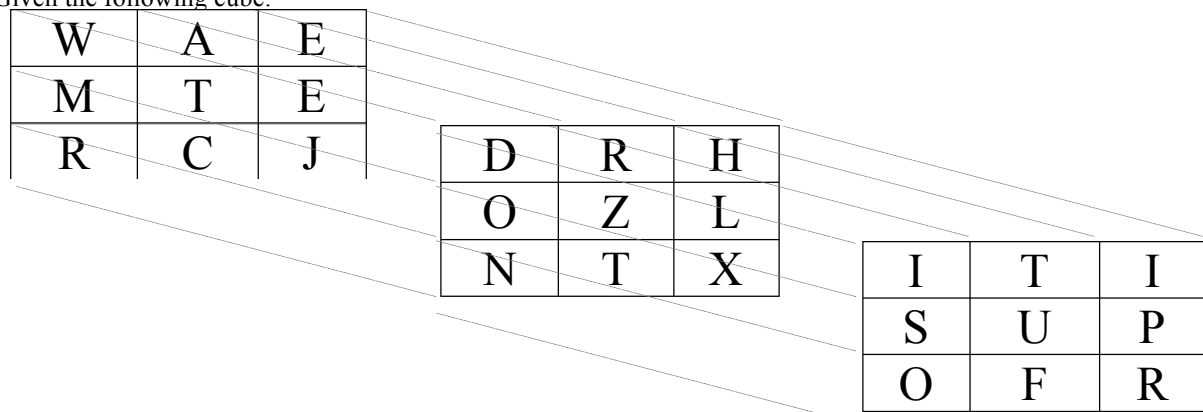
Letter	Number of them	Scrabble value of letter
A	9	1
B	2	3
C	2	3
D	4	2
E	12	1
F	2	4
G	3	2
H	2	4
I	9	1
J	1	8
K	1	5
L	4	1
M	2	3

Letter	Number of them	Scrabble value of letter
N	6	1
O	8	1
P	2	3
Qu	1	8
R	6	1
S	4	1
T	6	1
U	3	1
V	2	4
W	2	4
X	1	8
Y	2	4
Z	1	10

“Qu” will be used as a composite letter as otherwise most Q’s would be pretty useless.

## Appendix B - Adjacency in the cube

Given the following cube:



Examples from the top plane (bottom plane will be the same):

- W is adjacent to A, T, M in the same plane, and D, R, O, Z in the plane below.
- A is adjacent to W, M, T, E, E, in the same plane, and D, R, H, O, Z, L in the plane below.
- T in the centre of the plane is adjacent to all letters in the same plane, and all letters in the plane below.

Examples from the centre plane:

- D is adjacent to R, O, Z in the same plane, W, M, A, T in the plane above, and I, S, T, U in the plane below.
- R is adjacent to D, O, Z, L, H in the same plane, W, A, E, M, T, E in the plane above, and I, T, I, S, U, P in the plane below.
- Z in the very centre is adjacent to everything.

## DOCUMENT HISTORY

Version	Date	Changes made to document	Changed by
0.9	02/01/18	N/A - original draft	CJP
1.0	23/01/18	Minor changes in response to review by NWH	CJP