McMaster University

Software Project Management

SFWR ENG 3XA3

Requirements Document

Authors:
Mohammad Naveed **1332196**
Josh Voskamp **1319352**
Stephan Arulthasan **1308004**

December 3, 2015

Contents

List of Tables **3**

List of Figures **4**

Revision History **5**

1 Purpose of the Project **6**
1.1 What is the problem being solved? 6
1.2 Why is this an Important Problem? 6
1.3 Context of the problem 6

2 Stakeholders **6**
2.1 Client 6
2.2 Customer 7
2.3 Other Stakeholders 7

3 Users of the Product **7**
3.1 Hands-On Users of the Product 7
3.2 Priorities assigned to Users 7
3.3 User Participation 7
3.4 Maintainence Users and Testers 7

4 Project Constraints **7**
4.1 Mandated Constraints 7
4.2 Off the Shelf Software 8
4.3 Anticipated Workplace Enviroment 8
4.4 Schedule Constraints 8

5 Functional Requirements **8**
5.1 Scope of the Project 8
5.2 Context of the Work 9
5.3 Product Uses Case 10
5.4 Input and Output 11
5.5 Functional and Data Requirements 11

6 Non-Functional Requirements **13**
6.1 Look & Feel Requirements . . . 13
6.2 Usability Requirements . . . 13
6.3 Performance Requirements . . . 13
6.4 Operational Requirements . . . 14
6.5 Maintainability and Support Requirements . . . 15
6.6 Security Requirements . . . 15
6.7 Cultural Requirements . . . 15
6.8 Compliance Requirements . . . 15

7 Project Issues **15**
7.1 Open Issues . . . 15
7.2 Off-the-Shelf Solutions . . . 16
7.3 New Problems . . . 16
7.3.1 Effects on the Current Environment . . . 16
7.3.2 Effects on the Installed Systems . . . 16
7.3.3 Potential User Problems . . . 16
7.3.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product . . . 16
7.3.5 Follow-Up Problems . . . 16
7.4 Costs . . . 16
7.5 User Documentation and Training . . . 17
7.6 Tasks . . . 17
7.7 Risks . . . 17
7.8 Waiting Room . . . 17
7.9 Ideas for Solutions . . . 18

List of Tables

1 Revision History 5
2 Input and Output Table 11

List of Figures

1 Context Diagram 9
2 Activity Diagram for Move 10

Revision History

Revision Number	Revision Date	Description	Author
0	Oct 5 2015	Created Document	Mohammad Naveed
0	Oct 5 2015	Added Off the Shelf Software	Josh Voskamp
0	Oct 7 2015	Added Functional Requirements	Stephan Arulthasan
0	Oct 7 2015	Improve Formatting	Josh Voskamp
0	Oct 9 2015	Improved Functional Requirements	Stephan Arulthasan
0	Oct 9 2015	Added Non-Functional Requirements	Mohammad Naveed
0	Oct 9 2015	Create Project Issues	Josh Voskamp
0	Oct 9 2015	Add to Project Issues	Mohammad Naveed
0	Oct 9 2015	Complete Project Issues	Stephan Arulthasan
0	Oct 9 2015	Finalize Requirements Documentation	Josh Voskamp
1	Dec 3 2015	Add Client, Risks, Context Diagram, Input and Output, Activity Diagram	Josh Voskamp

Table 1: Revision History

1 Purpose of the Project

1.1 What is the problem being solved?

It is commonly known that technology, although serving countless purposes, is a large source of distraction to many people. More specifically, online applications, although highly entertaining and addicting, are very limited in cognitive stimulation. This is highly problematic as we are enabling a culture of absent minded technological engagement.

1.2 Why is this an Important Problem?

According to Jane McGonigal, a well known and world renowned game designer; we spend 3 billion hours a week playing video games. That is a lot of time that many people argue could be spent better, and that is what 2048 aims to accomplish. More and more people are playing video games everyday and 2048 is a fun and challenging game that tests the users' mathematical as well as their spatial intelligence. This allows 2048 to be fun, yet still be brain enhancing. Since the target audience for this game is so large, we can take advantage of this by providing users an option to spend their gaming time in a way thats beneficial mentally while still being entertained.

1.3 Context of the problem

Everyone experiences idle time in their day; this could be waiting for an appointment, a class, a bus or for friends. This game is intended to appeal to everyone looking for a more entertaining way to spend their idle time. The complexity of the game is meant to provide a challenge so that the user does not feel like they are wasting time, but using their time constructively. The game will be playable on all of the three major operating systems, OSX, Windows, and Linux with possible future expansion to mobile devices.

2 Stakeholders

2.1 Client

- Dr. Spencer Smith

- Peng Li

2.2 Customer

- Gamer

2.3 Other Stakeholders

- Developers and Testers

3 Users of the Product

3.1 Hands-On Users of the Product

Any person with any computer running the Java Runtime Environment can use our game to relieve stress and build on their mathematical and spatial intelligence.

3.2 Priorities assigned to Users

Primary Users: Windows, Mac OSX, and LINUX users
Secondary Users: Developers, testers and supervisors

3.3 User Participation

- User just has to play the game to participate

3.4 Maintainence Users and Testers

- Developers/Testers

4 Project Constraints

4.1 Mandated Constraints

Description: The product shall operate using Windows, Mac OSX and LINUX

Rationale: Users do not want to change operating systems just for the game and it will be more convenient for them.
Description: Product should be easy to use for people over the age of 10
Rationale: The product does require a certain level of mathematical and spatial intelligence which is why users under 10 years of age may find it difficult to use.

4.2 Off the Shelf Software

The open source project that is being improved can be found at:
`https://github.com/bulenkov/2048`
This implementation of 2048 is based off the original 2048 game created by Gabriele Cirulli which was also based off a small clone of 1024. https://github.com/gabrielecirulli/2048.

4.3 Anticipated Workplace Enviroment

This application is expected to be used at home, in the workplace, at school, and in the public.

4.4 Schedule Constraints

Requirements Document Revision 0	October 9
Proof of Concept Demonstration	October 26-28
Final Demonstration	November 30 - December 4

5 Functional Requirements

5.1 Scope of the Project

Everyone experiences idle time in their day; this could be waiting for an appointment, a class, a bus or for friends. That is a lot of time that many people argue could be spent better, and that is what 2048 aims to accomplish. 2048 is a game in which the user uses the arrow keys to combine alike tiles in an attempt to reach the "2048" tile.

5.2 Context of the Work

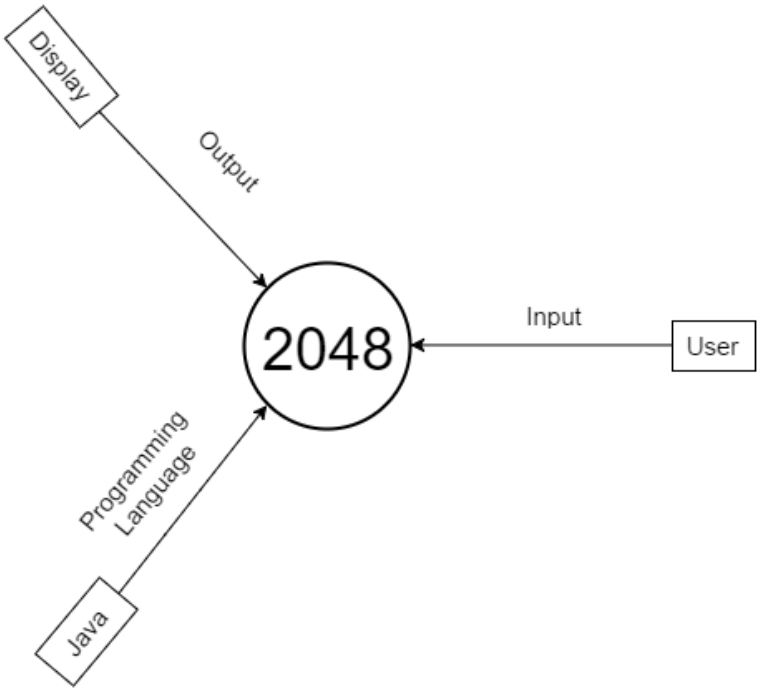

Figure 1: Context Diagram

5.3 Product Uses Case

Product Use Case Name: Move
Trigger: User Access
Preconditions: None
Interested Stakeholders: Client and Gamer

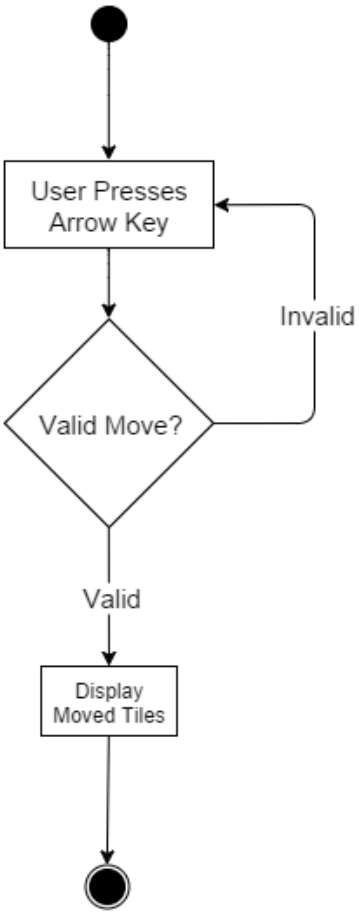

Figure 2: Activity Diagram for Move

5.4 Input and Output

Event Name	Input and Output	Summary
Move Left	Left Arrowkey(in)	Tiles Shift Left
Move Right	Right Arrowkey(in)	Tiles Shift Right
Move Up	Up Arrowkey(in)	Tiles Shift Up
Move Down	Down Arrowkey(in)	Tiles Shift Down
Restart	ESC(in)	Game Restarts

Table 2: Input and Output Table

5.5 Functional and Data Requirements

Requirement #:1

Description: The user must be able to start a game.
Rational: In order for the player to play the game, it has to initialize first.
Fit Criterion: User successively starts game so they can play.

Requirement #:2

Description: The user must be able to restart the game.
Rational: The user may want to restart if they made a bad move or have already won or lost the game.
Fit Criterion: Users successfully restarts the game.

Requirement #:3

Description: The user must be able to exit the game.
Rational: The user may want to exit the session at any time.
Fit Criterion: User successfully exits game.

Requirement #:4

Description: The user must be able to make valid moves.
Rational: In order to play and win the game, the user must be able to move the tiles around.
Fit Criterion: User successfully makes a valid move within the gameboard, where the tiles join correctly and score counter increases by the value created.

Requirement #:5

Description: The user must be able to win the game.
Rational: The player is working towards making the 2048 tile.
Fit Criterion: User successfully makes the 2048 tile and the game ends.

Requirement #:6

Description: The user must be able to lose the game.
Rational: It is possible to make a bad sequence of moves in the game so that all the tiles are filled and no more moves can be created.
Fit Criterion: The user fills up all the tile spaces and the game ends.

Requirement #:7

Description: The product must display the game score.
Rational: The user will be able to rank themselves against other players and try to beat their own high score.
Fit Criterion: Product successfully displays score based on moves made by user.

Requirement #:8

Description: The product must notify the user if they win or lose.
Rational: The user must be able to tell if they successfully win the game. Also, the tiles on the screen may be filled up but that doesn't necessarily mean they lost because there can be moves available.

Fit Criterion: The game displays a win notification when the user reaches the 2048 tile and a lose notification when the user fills up all the tile spaces with no moves left.

6 Non-Functional Requirements

6.1 Look & Feel Requirements

- Appearance Requirements
  - Board size must be 4x4
  - The interface must be intuitive

6.2 Usability Requirements

- Ease of use Requirements
  - Product shall be easy to use for anyone who is older than the age of 10
- Personalization and Internationalization Requirements
  - N/A
- Ease of Learning of Requirements
  - The product shall be easy to learn for anyone who is older than the age of 10

6.3 Performance Requirements

- Speed requirements
  - The game must start in under 2 seconds
  - Any interaction with the user and the product during gameplay must have a maximum response time of 2 seconds

- Precision Requirements
  - All values on the tiles must be integers
  - The value of the smallest tile must be 2
  - The value of the largest tile must be 2048
  - The value on all tiles must be a multiple of 2
- Reliability & Availability Requirements
  - The product shall be available for use 24 hours a day for 365 days of the year
- Safety Critical Requirements
  - This game must not hurt anyone
- Capacity Requirements
  - The product shall cater to 1 user on the machine

6.4 Operational Requirements

- Expected Physical Environment
  - The product is to be used by gamers at home, in the workplace, at school, and in the public
- Expected Technological Environment
  - The product shall be available on LINUX, Mac OSX, and Windows operating systems
  - The environment must have the Java Runtime Environment (JRE)
- Partner Applications
  - N/A

6.5 Maintainability and Support Requirements

- Maintenance Requirements
  - Make it easy to fix any potential bugs in the code
- Supportability Requirements
  - The product will not be supported, however the source code will be available for examination or enhancement.
- Adaptability Requirements
  - The product is expected to run on LINUX, Mac OSX, and Windows operating systems

6.6 Security Requirements

- N/A

6.7 Cultural Requirements

- The product shall not use icons that could be considered offensive in any of our market countries.

6.8 Compliance Requirements

- N/A

7 Project Issues

7.1 Open Issues

- Creating a GUI compatible with various screen resolutions
- Animation for the sliding of the tiles
- Add a FAQ option

7.2 Off-the-Shelf Solutions

- `https://github.com/bulenkov/2048` The Open-source Java project being improved.
- `https://github.com/gabrielecirulli/2048` The Original Open-source project for the game 2048, It was originally implemented in JavaScript.

7.3 New Problems

7.3.1 Effects on the Current Environment

N/A

7.3.2 Effects on the Installed Systems

N/A

7.3.3 Potential User Problems

N/A

7.3.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

N/A

7.3.5 Follow-Up Problems

N/A

7.4 Costs

There are no direct monetary costs associated with this project, but about half a year of development time will be required.

7.5 User Documentation and Training

User's will provided with information on program use via a FAQ option which, when selected, will open a dialog box detailing general functionality of the program. Beyond the help document, user?s familiar with casual computer use should require no further training.

7.6 Tasks

- Revise requirements document.
- Create a test plan.
- Demonstrate a proof of concept.
- Draw up design documents.
- Revision 0 project demonstration.
- Create a user guide for the project.
- Write up a test report
- Final revision project demonstration.
- Write final revisions to documentation.

7.7 Risks

- Team members not doing their assigned tasks
- Team members not following deadlines set by the team, or client
- Project not compiling for the Professor or TA

7.8 Waiting Room

There are no plans to introduce new releases of this product. If that were to change, new features would include an animated GUI and an online high score list.

7.9 Ideas for Solutions

- Use Java Swing and AWT for GUI development