

Software Requirements **Specification (SRS)**

Ultimate 2048

Final(Stage II) Report

Team Members:-

- 1) Vivek Sonawane (Roll No. 140020030)
- 2) Mahesh Pottulwar (Roll No. 140020035)
- 3) Shubham Kothawade (Roll No. 140020040)

Slot No. 11

Group No. 17

CS Lab TA:-

Sougata Sinha (Roll No. 143050015)

Table of Contents

- 1.0. About Ultimate 2048
- 2.0. Introduction
 - 2.1. Purpose of Document
 - 2.2. Objective
 - 2.3. Gameplay
 - 2.4. References for SRS
 - 2.5. References for Project
- 3.0. Requirements Specification
 - 3.1. Functional Requirements / Modules
 - 3.2. Non-Functional Requirements
- 4.0. Developer-Defined Functions in game
 - 4.1. User Interface
- 5.0. Solution (Product) Overview
 - 5.1. End Product Environment
 - 5.2. Assumptions and Dependencies
- 6.0. Work Allocation
- 7.0. Future Prospects

1.0 About Ultimate 2048

2048 is a popular single player puzzle game developed by Gabriele Cirulli (an Italian web developer and programmer) in March 2014. This sliding block puzzle has more than 10 million downloads on Google Play itself. Since this game runs on open source code, many programmers came up with different varieties of this game like 2048 hex , 2048 in 5*5, 6*6, 7*7, etc. grid.

2.0 Introduction

1.1 Purpose of Document

The purpose Of this document is to present a detailed description of the 'Ultimate 2048' game that we made . The following sections of the document explain the purpose and features of the game, the interface of the game, and the constraints under which it must operate.

1.2 Objective

The main aim of Ultimate 2048 is to create a tile of 2048 using only the the tiles of 2 or 4. This achieved by sliding tiles up, down, left or right direction in 4*4 grid. When whole of the grid is completely filled such that no two same numbers are adjacent to each other, the game is over.

1.3 Gameplay

Ultimate 2048 can be played in 4*4 grid. 2 or 4 numbered tiles are randomly generated in grid at random places. Player has to combine the tiles of same numbers to get sum of both and thus go on. Player can move tiles up, down, left or right using character keys W, S, A and D for moving Up, Down, Left and Right respectively.. The tile thus formed is shifted to the extreme block which is empty. If no new tile is generated, then tiles are shifted as it is in that direction if there is any blank space. After making 2048 numbered tile, player can continue to make the next tiles

1.4 References for SRS

1. [www.wikipedia.com\(http://en.wikipedia.org/wiki/2048_\(video_game\)\)](http://en.wikipedia.org/wiki/2048_(video_game))
2. <https://github.com/gabrielecirulli/2048>
3. <http://cse.iitb.ac.in/~cs101/project.html> - Eyatra.tar.gz

1.5 References for Project

1. www.cplusplus.com
2. <https://github.com/gabrielecirulli/2048>
3. <http://www.cplusplus.com/reference/cstdlib/rand/>
4. <http://www.stackoverflow.com>
5. Let us C++ by Yashwant Kanetkar

3.0 Requirement specifications

3.1 Functional Requirements / Modules

Starting Game

1. Game starts upon running the executable equivalent code of the game.
2. executable (.exe) window appears showing start of the game.
3. Player can start playing the game now.

In Game

1. The game will start with tile 2 or 4 placed at random place in grid.
2. Using keyboard buttons 'w' , 's' , 'a' or 'd' player can move tiles up, down, left and right respectively.(arrow keys will be used in final game)
3. When one of the above button is pressed, tiles in opposite direction move towards that side and occupy blank spaces.

4. After every move, tile of 2 or 4 will be randomly created at any blank space.
5. If all the blocks are filled with tiles such that no two adjacent tiles have same number, then game is over.

3.2 Non-Functional Requirements

1. **General Guidelines:** Priority should be given to performance, adaptability, maintainability, and usability.
2. **Operating Constraints:** This code requires C++ environment compatible with library functions like 'cstdlib' and 'ctime' for random function to work effectively.

4.0 Developer-Defined Functions in game

a] **upward()** :- This function is used to shift the tiles upwards appropriately by checking blank and non-blank spaces in its path.

b] **downwards()** :- This function is used to shift the tiles downwards appropriately by checking blank and non-blank spaces in its path.

c] **left()** :- This function is used to shift the tiles leftwards appropriately by checking blank and non-blank spaces in its path.

d] **right()**:-This function is used to shift the tiles rightwards appropriately by checking blank and non-blank spaces in its path.

e] **assign()**:- This function is used to replace old grid with the new one after above functions are called.

f] **equal()**:- This function is used for checking the termination case of the game.

4.1 User interface

Character keys W, S, A and D are used as interface.

5.0 Solution (Product) Overview

5.1 End-Product Environment

The end product has two active actors. One will be player or user and other is the compiler. Player will be sliding the tiles while compiler will go on generating tiles and place them anywhere in grid. Compiler will also keep on upgrading the grid appropriately.

5.2 Assumption and Dependencies

1. The application will run on Windows XP/Vista/7/Ubuntu and Linux platforms with the C++ Runtime Environment installed.
2. The application does not require an internet connection, other than to download the application from the product website.

6.0 Work Allocation

The work that has been allocated to each developer is as follows:

Vivek Sonawane:-1] Exploring and implementing random functions in programme.

2] Movement of tiles.

3] Appropriate addition of same tiles and upgraded position of all this tiles in the grid.

Mahesh Pottulwar:- 1] Synchronizing user given instructions with C++ programme.

2] Movement of tiles.

Shubham Kothawade:- Creating graphics for the game.

7.0 Future Prospects :-

Ultimate 2048 can be modified and thus new versions can be created which might involve more thinking for their solving. Using graphics in Ultimate 2048 will make the game more interactive with user.

