Software Requirements Specification

For

TetrisBlast

Version 0.1

Prepared by

Group Name: .DUEL

|  |  |  |
| --- | --- | --- |
| LIM PEI YONG | 36732 | mike.limpeiyong@gmail.com |
| WILLIAM NGU WEE HONG | 39322 | wnwh6717@gmail.com |
| ABDUL RAHIM BIN MOHAMAD | 35282 | chongpitt@gmail.com |
| LIM CHONG FATT | 36712 | shakion14@gmail.com |
| HOE YOU TIEN | 36249 | youtien3062@gmail.com |

|  |  |
| --- | --- |
| Instructor: | Mdm Nurfauza Jali |
| Course: | TMP3413 Software Engineering Lab |
| Lab Section: | *Wednesday 9am – 11am* |

|  |  |
| --- | --- |
| Date: | 12 Nov 2014 |
|  |  |

Contents

Revisions ii

1 Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definition 1

2 Overall Description 2-3

2.1 Game Perspective 2

2.2 Game Functionality 2

2.3 Operating Environment 3

2.4 Design and Implementation Constraints 3

2.5 Assumptions and Dependencies 3

3 Specific Requirements 4-6

3.1 External Interface Requirements 4

3.2 Functional Requirements 5

3.3 Behaviour Requirements 6

4 Other Non-functional Requirements 7

4.1 Performance Requirements 7

Appendix A – Data Dictionary 8

Appendix B - Group Log 9-10

# Introduction

This project to develop a game called “TetrisBlast”. This game is more or less like the classic Tetris. A document called Software Requirements Specification (SRS) is very important no matter it is about an application or a system. Requirements of TetrisBlast are described in detail in SRS like user interface, playing mode, high scores and so on.

## Purpose

The purpose of this SRS is to provide a detailed description of the requirements for TetrisBlast. Description of the requirements is needed for the development of the game and the further improvements.

This document will cover the interface of the game, how the game run, and the constraints under which it must run and how the game will react to external stimuli. The document will also cover hardware, software, and various other technical dependencies.

## Scope

This document describes the software requirements for the TetrisBlast. TetrisBlast must be able to run like a classic Tetris.

It must have the features like high scores and the difficulty mode. High scores is a must to let player plays repeatedly to beat the high score. Difficulty mode is to let player not just stick at easy mode or normal mode, but also could try the more challenge mode like hard mode.

## Definitions

LCD – Liquid-crystal display

OS – Operating system

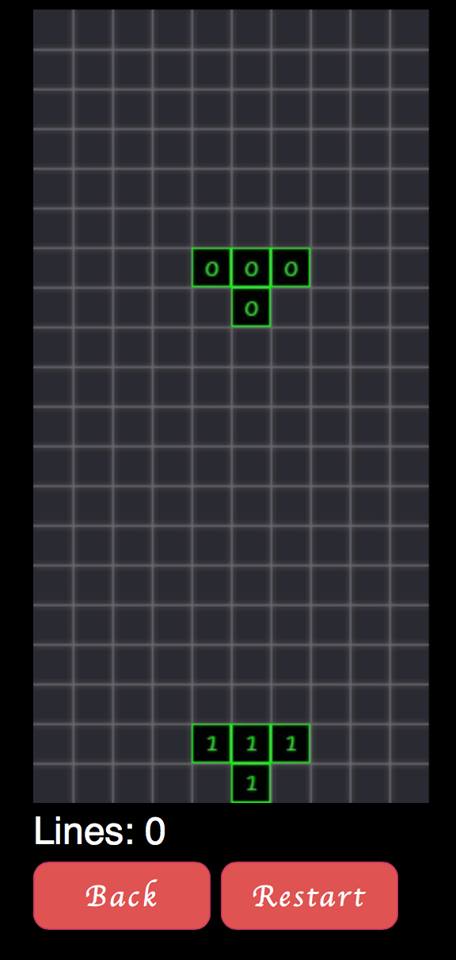
SRS – Software requirement specification

TCP/IP – Transmission control protocol/internet protocol

# Overall Description

## Game Perspective

The original of this game are Tetris. After modified the game, a brand new game is born called TetrisBlast. In this game, player can choose their favourite mode, either in normal mode or hard mode. After entered the game, player can play the game using with Tetris method but when the game started, the blocks will drop from the top.



## 

## 2.2 Game Functionality

* It can choose game mode.
* It can view high score
* It can guide player how to play the game.

## Operating Environment

The TetrisBlasts game will operates by using Web apps. Besides that, all the smartphones that compatible can access for this game. The game can get from gitHub and it is an online store. A built-in speaker of devices should be included and internet connection will required for multiplayer mode. Diagram 1 has shown the major components in the system and their connections.

## 

## *Diagram 1*

## Design and Implementation Constraints

This program is developed using intel XDK and it is only available to run in smartphone and pc. The language for developed the program is only English language. Other than that, size of the program will be fix in range for few dozen megabytes to avoid too heavy for smartphone. An installer is needed to install the program.

## Assumptions and Dependencies

Assumption

* Game system is compatible with smart phone
* Game has no serious mistake.
* Users understand English.
* Devices must have touch screen function.

Dependencies

* It does not need software platform to run.
* It does not need external devices to control such as keyboard and mouse.

# Specific Requirements

## External Interface Requirements

### User Interfaces

In Figure 1, it shows the main menu interface that having three user options which is:

* New Game
* How To Play
* View High Scores

In Figure 2, it shows when user chooses the “Normal game” and “hard game”option. The game will show player on the stat of the game such as difficulty and effects.

In Figure 3, it shows that player capable see the instruction.

In Figure 4, it shows that player can view the high score.

## Functional Requirements

|  |
| --- |
| **DUEL\_Activity** |
| - DIALOG\_GAME\_OVER:int  - DIALOG\_PAUSE:int  - GAME\_STATUS\_LOSS:int  - GAME\_STATUS\_PROGRESS:int  - GAME\_STATUS\_WIN:int  - ICICLE\_KEY:String  - MAX\_LINES\_TO\_SEND:int  - MSG\_END\_GAME:int  - MSG\_LINES\_CLEARED:int  - MSG\_NEXT\_PIC:int  - MSG\_PAUSE:int  - MSG\_UNPAUSE:int  - MSG\_UPDATE:int  - TAG:String  - linesToSend:int |
| + increaseLinesToSend():void  + increaseScore():void  + setNextPic():void  + onBackPressed():void  + onCreate():void  + onPause():void  + onResume():void  + onStop():void |

|  |
| --- |
| **MainMenu** |
| - GAME\_MODE:String  - HAVE\_ACTIVE\_PROFILE:String  - MODE\_COOP:int  - MODE\_MULTY:int  - MODE\_SINGLE:int  - MODE\_UNDEFINED:int  - MODE\_VS:int  - PREF\_TAG:String  - PROFILE\_ID:String  - TAG:String |
| + onCreate ():void  + onCreateOptionsMenu ():void  + onOptionsItemSelected ():void  + onStart ():void |

|  |
| --- |
| **Setting** |
| - OPT\_MUSIC\_ON:String  - OPT\_PATTERN:String  - OPT\_SENS\_LIST:String  - OPT\_SOUND\_ON:String |
| + getMusicEn():boolean  + getSoundEn ():boolean  + onCreate ():void  + onStop():void |

|  |
| --- |
| **NewGameActivity** |
| - DIFFICULTY:String  - HOST:String  - SHADOW:String  - TAG:String  - WHATCH\_DOG:int  - WHATCH\_DOG\_TIMEOUT:long |
| + saveActivitySettings ():void  + showStatus ():void  + onCreate ():void  + onProgressChanged ():void  + onStart ():void  + onStartTrackingTouch ():void  + onStopTrackingTouch ():void |

|  |
| --- |
| **TetrinoMap** |
| - MAP\_X\_SIZE:int  - MAP\_Y\_SIZE:int |
| + createRandomLine ():void  + removeLine ():void  + addLines ():void  + copyFrom ():void  + getMap():int  + getMapValue ():void  + lineCheckAndClear ():int  + restMap():void |

|  |
| --- |
| **OTetrino** |
| - BLOCK\_TYPE:int |
| + initTetrino ():void |

|  |
| --- |
| **Tetrino** |
| - SIZE:int  - ghostEnabled:boolean  - gMap:int  - ghostPosPoint  - posPoint  - sMap:int |
| + initTetrino ():void  + copyTetrinoMap ():void  + drop ():void  + getGhostXPos ():void  + getGhostYPos ():void  + getSize ():void  + getXPos ():void  + getYPos ():void |

|  |
| --- |
| **STetrino** |
| - BLOCK\_TYPE:int |
| + initTetrino ():void |

|  |
| --- |
| **ITetrino** |
| - BLOCK\_TYPE:int  - I\_SIZE:int |
| + ITetrino ():void  + initTetrino ():void  + getSize ():void  + initGhost ():void  + getYPos ():void |

|  |
| --- |
| **LTetrino** |
| - BLOCK\_TYPE:int |
| + initTetrino ():void |

|  |
| --- |
| **TTetrino** |
| - BLOCK\_TYPE:int |
| + initTetrino ():void |

|  |
| --- |
| **JTetrino** |
| - BLOCK\_TYPE:int |
| + initTetrino ():void |

## Behaviour Requirements

### Use Case View

1.

Start a new game

Play hard mode

Learn how to play

Player

Play normal mode

View high scores

# Other Non-functional Requirements

## Performance Requirements

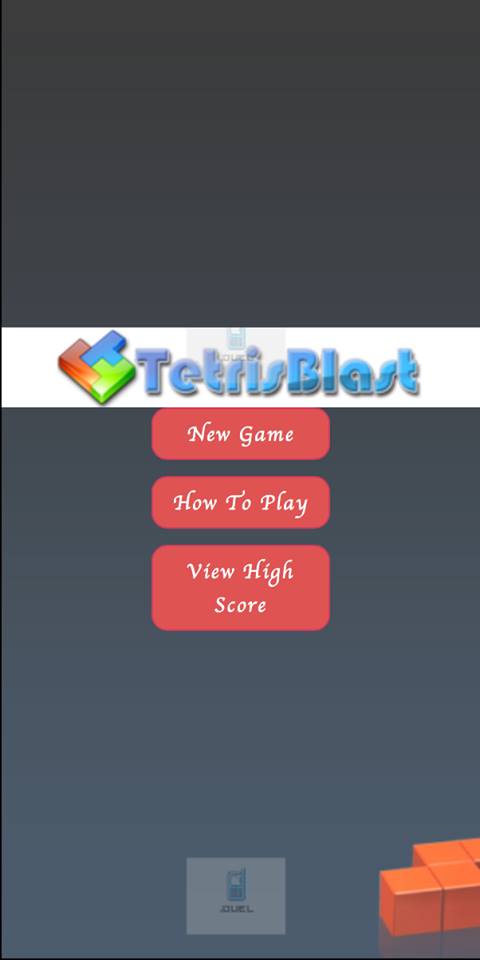
* The actual game should always be inverted vertical side of the phone. Even if the auto-rotation button is turned on. The game should be built with its interfaces that does not change according to phone’s rotation. Even the game play screen will be built upside down to suit it to the original tetris game. If the game were to change according to user’s phone auto-rotation, a horizontal inversion will make the game play screen look flat and hard to play with.
* Each of the number stated in the blocks need to be as clear as possible. Mind all the other things. This game is based on that number to win, so the number need to be visible by the users. If the number is not clear enough, then this game would be the same as the classic tetris but easier to lose.
* The game should load in a second. If the loading time become too long, user should be prompt whether to close the game or wait for it to load. Since the game is just a simple game and does not require the internet to load, then it should load within a second.
* There should never be a buffer while the game is played. Since the game itself is just a simple game, did not use other resource to load such as the internet, the game should be smooth enough when played. It is hard for a user to both deal with math and alongside the buffering. So, the game should not be buffering while being played by making it as simple as it can.

Appendix A – Data Dictionary

Table 1: Tetris functions

|  |  |
| --- | --- |
| **Attributes** | **Methods** |
| -ROWS | +onReady() |
| -COLS | +getInput() |
| -SIZE | +onImagesLoaded() |
| -canvas | +initGame() |
| -ctx | +update() |
| -blockImg | +copyData() |
| -bgImg | +checkLines() |
| -gameOverImg | +zeroRow() |
| -curTetrino | +drawBoard() |
| -gameData | +drawTetrino() |
| -imgLoader | +checkMove() |
| -prevTime | +addEventListener() |
| -curTime | +BulkImageLoader() |
| -isGameOver | +addImage() |
| -lineSpan | +loadImages() |
| -curLines | +getElementById() |
| -touchX | +getContext() |
| -touchY | +preventDefault() |
| -touchId | +getImageAtIndex() |
| -difY | +getRandomTetrino() |
| -touchEndX | +toString() |
| -touchEndY | +requestAnimationFrame() |
| -touch | +copyData() |
| -difX | +clearRect() |
| -r | +drawImage() |
| -c |  |
| -xpos |  |
| -ypos |  |
| -state |  |
| -lineFound |  |
| -fullRow |  |
| -drawX |  |
| -drawY |  |
| -result |  |
| -newX |  |
| -newY |  |

Appendix B - Group Log



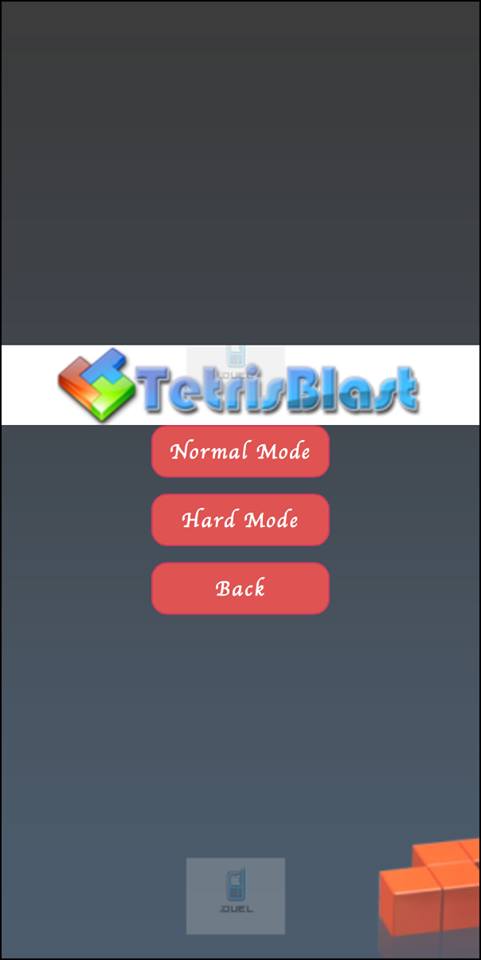


Figure 1

Figure 2

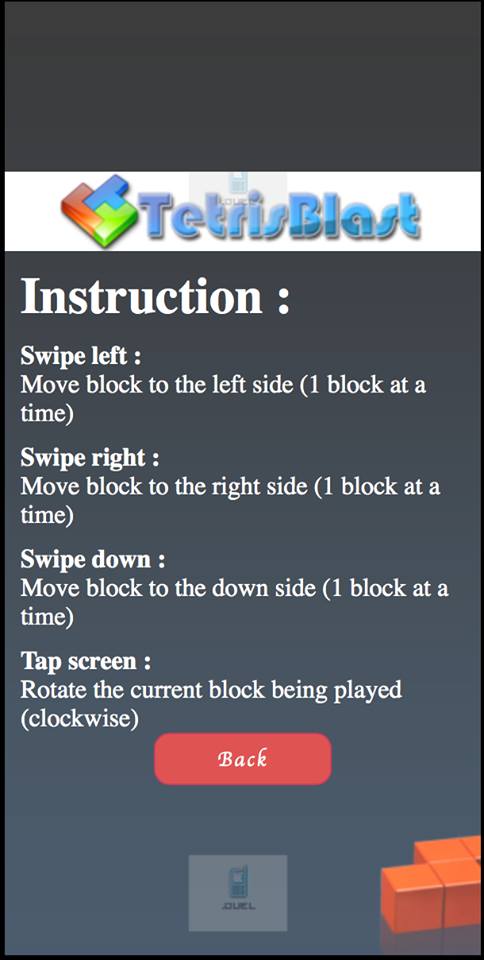
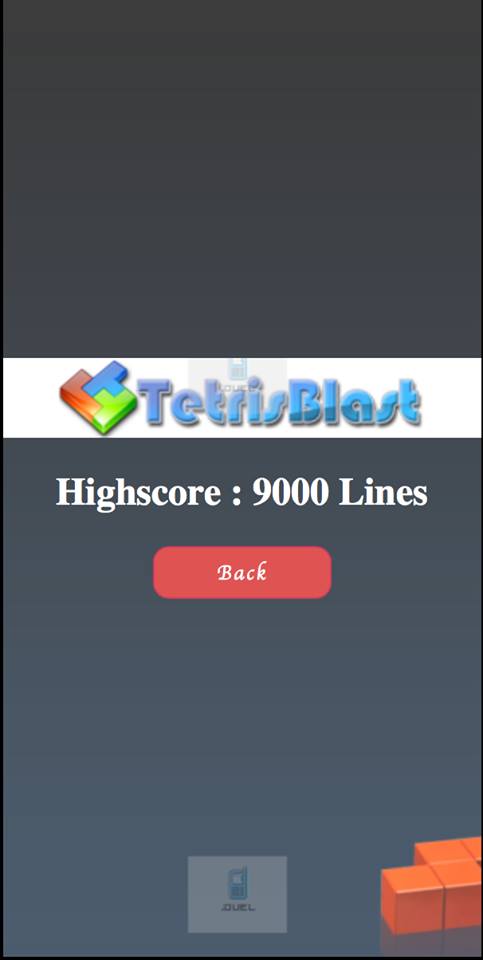


Figure 6

Figure 5

Figure 4

Figure 5

Figure 3

Figure 4

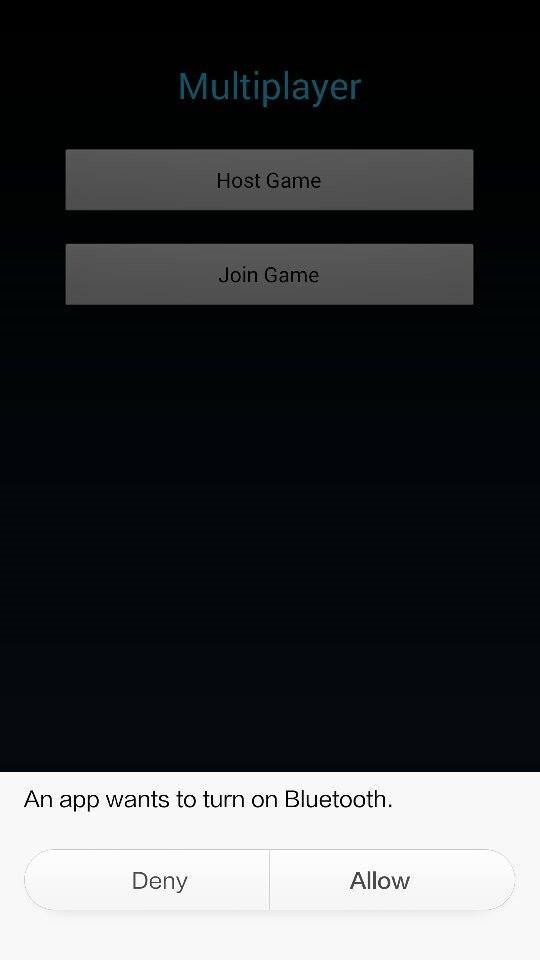


Figure 6

Figure 5