**Table of Contents**

[**Overview of Project** 2](#_Toc406145075)

[**Project Scope** 2](#_Toc406145076)

[**Product Functionality** 2](#_Toc406145077)

[**Use case diagram** 3](#_Toc406145078)

[**Sequence diagram** 3](#_Toc406145079)

[**Activity diagram** 4](#_Toc406145080)

[**Screen shots of the project interface with description** 4](#_Toc406145081)

[**Conclusion** 7](#_Toc406145082)

[**References** 8](#_Toc406145083)

[**Appendix A** 9](#_Toc406145084)

**List of Table**

[Table 1. Attributes and methods 9](#_Toc406144949)

[Table 2.Button descriptions 10](#_Toc406144950)

**List of Figures**

[Figure 1. Use case diagram 3](#_Toc406140831)

[Figure 2. Sequence diagram 4](#_Toc406140832)

[Figure 3. Activity diagram 4](#_Toc406140833)

[Figure 4.First user interface 5](#_Toc406140834)

[Figure 5.Second user interface 5](#_Toc406140835)

[Figure 6.Third user interface 6](#_Toc406140836)

[Figure 7.Fourth user interface 6](#_Toc406140837)

[Figure 8.Fifth user interface 7](#_Toc406140838)

[Figure 9.Tetris brain 7](#_Toc406140839)

# **Overview of Project**

This project is for developing a game called “TetrisBlast”. TetrisBlast game is a mobile application where players have to try arranged five different shapes of block in a way that exactly filled up all the spaces in the graphical interface shown to players (Over, 2010). Basically TetrisBlast game contains of three components which are user interface, functionality, and game control. All the interfaces inside the game were provided by the three components.

# **Project Scope**

This documents describe the software requirements for the TetrisBlast game. TetrisBlast game is a traditional game which need some planning strategy (Humphrey, 2000). In the TetrisBlast game, player can choose different mode such as normal mode and hard mode. To guide the player, there is an instruction to guide player how to play. There is one more function which called ‘View highest score’ which can let player know their high score.

# **Product Functionality**

Functionalities of TetrisBlast game have been show to player through different interfaces. Players can choose to start a new game with either normal or hard gaming mode. Inside the game, different Tetris blocks will dropped one by one with constant speed and players have to control the blocks by moving or changing their position to fix into a row and then the row will be dissolved automatically (Sankaran et al., 2010). The game will automatic recorded player’s scores every time they dissolved a row of blocks and renew the highest score when they break the record. Besides these functions, TetrisBlast game also enables players to view about the rule of how to play the game and the highest score.

# **Use case diagram**



Figure 1. Use case diagram

# **Sequence diagram**

Figure 2. Sequence diagram

# **Activity diagram**

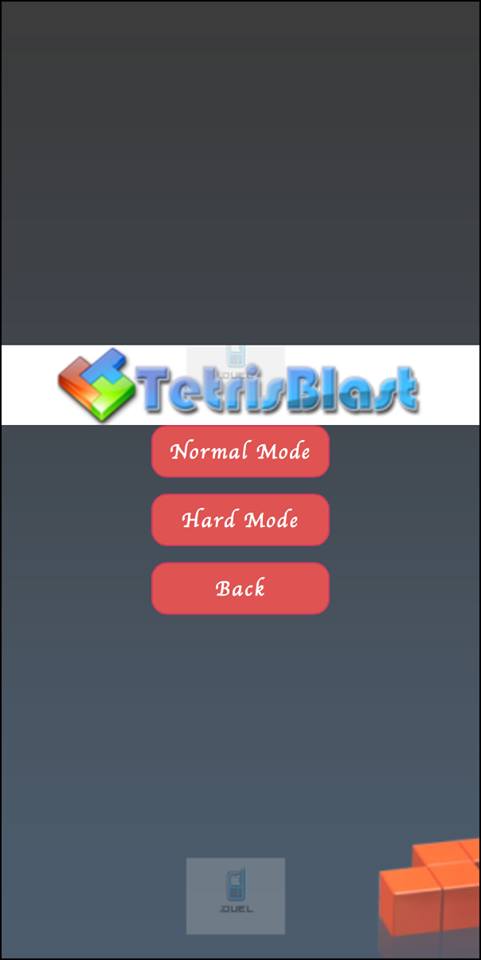


Figure 3. Activity diagram

# **Screen shots of the project interface with description**

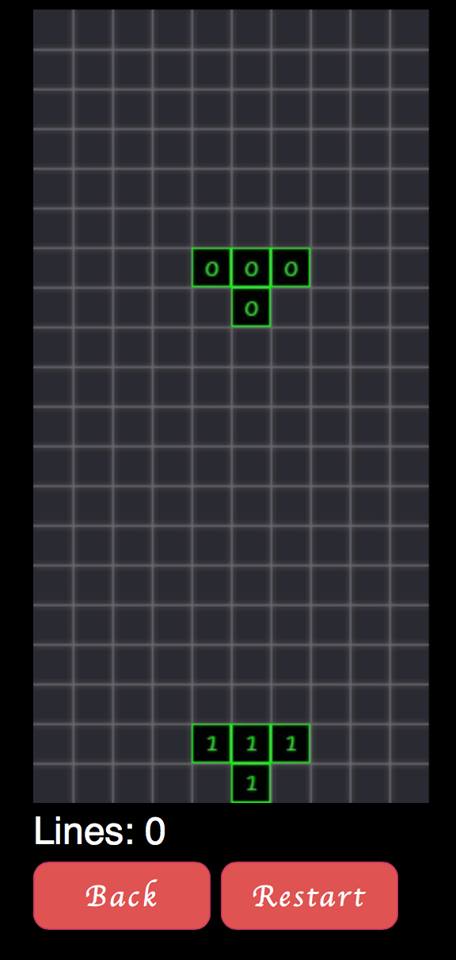
This screen shot show the first interface of our game. There is three buttons on this interface which are ‘New Game’ button, ‘How to play’ button and ‘View High Score’ button. Each button can access to another interface if user click on it.

Figure 4.First user interface



Game will be directed to this interface when user click ‘New Game’ button. In this interface, there are 3 more buttons in this interface which are ‘Normal Mode’ button, ‘Hard Mode’ button and ‘Back’ button. Player can choose different level of game difficulty based on what player choose. Player can back to previous interface by selecting ‘Back button’.

Figure 5.Second user interface



After player select either normal mode or hard mode, player will be directed into this interface. At the bottom of this interface, there is a ‘Lines’ which is used to record the result of player. Below ‘Lines’ there are two buttons which are ‘Back’ button and ‘Restart’ button. The function of ‘Back’ button is to redirect player back to first interface and ‘Restart’ button is to restart the whole game.

Figure 6.Third user interface

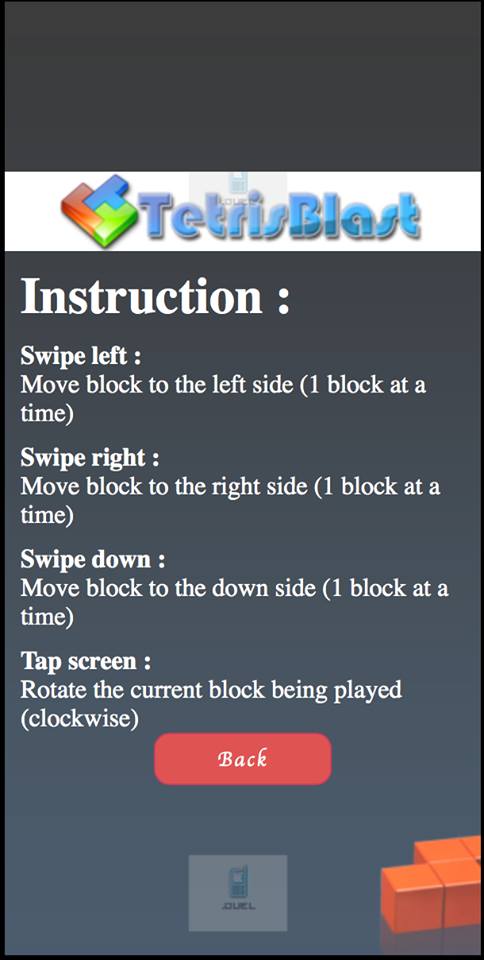
If player click ‘How to play’ button, player will be directed into instruction pages which to teach player how to play the game. At the bottom of the interface, there is a ‘Back’ button which will direct player to first interface.

Figure 7.Fourth user interface

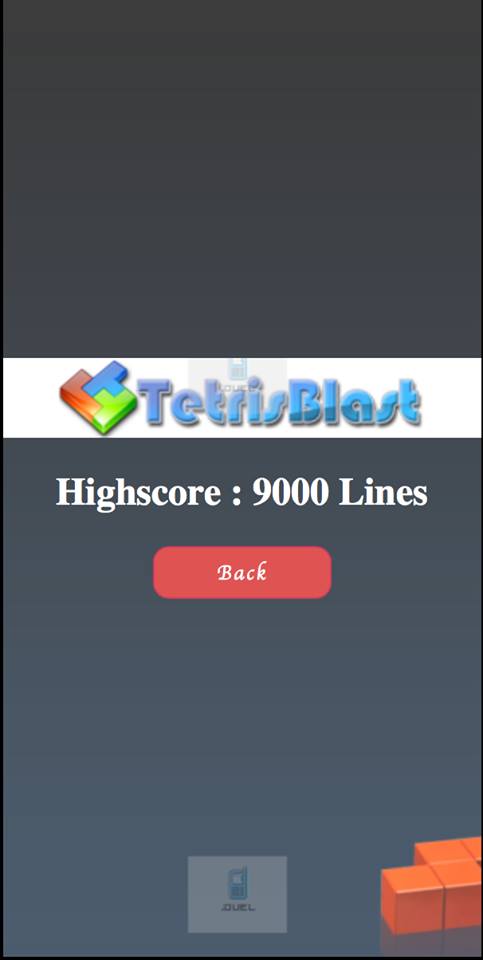
Player will see this interface if player click on ‘View High Score’ button. User can also back to the first interface by clicking the ‘Back’ button which been set below the Highscore.

Figure 8.Fifth user interface

# **Conclusion**

In this project, we have named our team .DUEL. The aim of .DUEL is to develop a game called TetrisBlast. At first, the name of the game was 3-seven Tetris. 3-seven Tetris was a game that combined by Dr. Mario and Tetris. But, we have changed the game concept to standard Tetris. Standard Tetris means the standard playing rules of Tetris. We have used HTML 5, JavaScript, CSS, and node.js to build up the game. The tool we have used is called Intel XDK. We used the tool to design, debug, and build the game.

In our TetrisBlast, we let the players to have two modes to play, which are normal mode and hard mode. Players can choose to play normal mode in the beginning. After they learnt the skill, they could try the hard mode. We have added in features like view high scores. Players can always refer to the high scores to strike for higher score. We have also added how to play feature. That is a guideline for the players who do not know about TetrisBlast playing rules at all.

TetrisBlast is just look like a classic Tetris. Normally playing Tetris is good for the player’s brain (SEI Partner, 2013). This game is not only let the players feel fun to play it, it is also make the players to keep on using their brain. So, this is not just a game, it is also can be said as a healthy game.

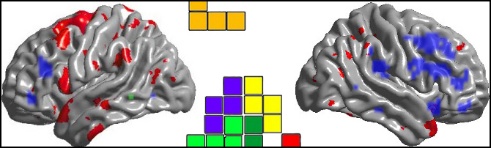


Figure 9.Tetris brain

# **References**

Humphrey, W. S. (2000). *The team software processSM (TSPSM)* (p. 51).

Over, J. (2010). *Introduction to the team software process* (p. 142).

Sankaran, A., Samsonyuk, A., Attar, M., Parham, M., Zayikina, O., Rifai, O. J., … Hassan, R. (2010). *Software design document, testing, and deployment and configuration management* (p. 136).

SEI Partner. (2013). *TSP and PSP methodology* (p. 7).

# **Appendix A**

|  |  |
| --- | --- |
| **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 |  |

Table 1. Attributes and methods

**User Manual**

TetrisBlast is just like a classic Tetris game. What players have to do is just arrange each block that dropped from the top of the playing field. Players have to arrange them so that a line that filled up with blocks is formed and it is dissolved immediately. Its playing rules is just the same as classic Tetris. But their user interface is a bit different. The user interface of TetrisBlast is shown below:

|  |  |
| --- | --- |
| **Buttons** | **Descriptions** |
| New game | Players start a new game. |
| How to Play | Players learn how to play the TetrisBlast. |
| View highest score | Players view their previous scores. |
| Normal mode | This mode is the easiest mode for the players. |
| Hard mode | This mode is the hardest mode for the players. |

Table 2.Button descriptions