**Group No. 21**

**Members**:

1. Đoàn Ngọc Bảo
2. Lê Trảo Việt Cường
3. Huỳnh Chí Hiếu
4. Dương Tử Huy
5. Đinh Kim Ngân

**Project**: **GAME TETRIS (QT PLATFORM)**

**WEEKLY REPORT**

# Feature List

Our project is the game Tetris. We will use QT platform to implement it. Our version is based on the rules of the classic Tetris that we all know. Specifically, these following features are likely to have:

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** |
| REQ\_01 | Start a new game | High |
| REQ\_02 | Allow to choose the difficulty level for the game. There are 3 difficulty levels: Easy, Normal and Hard. | Medium |
| REQ\_03 | Allow user to choose whether to save his/her score. | High |
| REQ\_04 | Show the list of high scores that are saved before. | Medium |
| REQ\_05 | Player can pause/resume the current game. | High |
| REQ\_06 | Allow user to configure the game. E.g. Set sound on/off | Medium |

# Mock-up Design

**Home Screen**

New Game

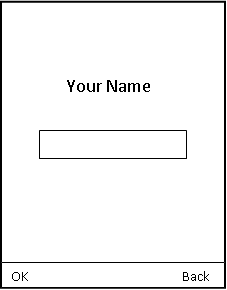
High Scores

Options

Exit

This screen shows the options when user starts the game. This screen includes 4 buttons. They are:

* *New game* button: go to playing status
* *High scores* button: view top score
* *Options* button: configure game before start game
* *Exit* button: exit game



**Input Name Screen**

This screen appears to player when he/she reaches the game over status. It asks player to input his/her name.

Player can either ignore by click *Back* button or click *OK* button to store player’s score with *your name field*.

**Playing Screen**

Pause

Back

Score

Level

This is the most important screen. Player plays game here!

The top of screen shows 3 parts which are:

* The left-most: show next figure
* The middle: show *level* of player (*easy*, *medium* or *hard*)
* The right-most: show *score* player has got

The center screen is an area which shows status game

The bottom of screen includes the controllers. They are:

* *Pause/Resume* button: pause/resume game
* *Back* button: back to *Home Screen*
* *4 Direction* buttons: use to control the figure block: go left, go right, go fast down, rotate figure

Craig 1000000

Huy 500

Bao 500

Ngan 500

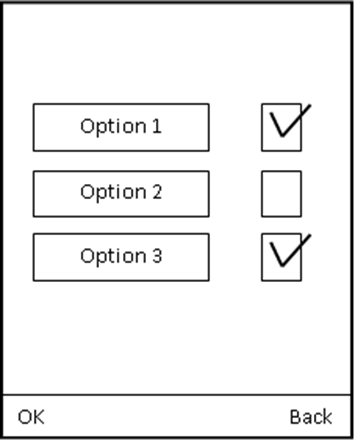
OK Back Back Back

Hieu 500

**View High Scores Screen**

This screen shows the list of high scores. The table has two columns: the first column is player name and the second column is player’s score

Player can click *OK* button to come back the *Home Screen*



**Configuration Screen**

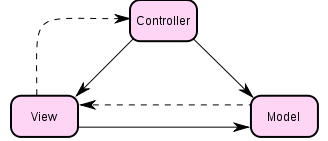
This screen shows the list of options that player can configure for his/her convenience.

If player click *OK*: all configurations are applied

If player click *Back*: all configurations are ignored

After player clicks *OK/Back* button, player will come back the *Home Screen*

# Program design



## View

QML files implement the screens of game and accept the events from user.

## Controller

Controller handles the events from view and choose the appropriate operation in model to process them.

Controller in our game is Tetris.js file.

## Model

Model manipulates the logic of game.

**Example:**

View raises a click event.

onStatusChanged: {

if(*status* == 2){

*Utils*.showHighScore(1)

}

}

Button {

id: *btnShowScore*

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.top: *btnStart*.bottom

text: "Show Scores"

onClicked:{

*btnShowScoreClick*();

}

}

function *showHighScore*(level){

var *db* = *openDatabaseSync*("TetrisScores", "1.0", "Local Tetris High Scores",100);

*db*.transaction(

function(tx) {

*tx*.executeSql('CREATE TABLE IF NOT EXISTS Scores(name TEXT, score NUMBER, level NUMBER)');

var *rs* = *tx*.executeSql('SELECT \* FROM Scores WHERE level = "level" ORDER BY score desc LIMIT 10');

var *r* = "\nHIGH SCORES for this grid size\n\n"

for(var *i* = 0; *i* < *rs*.rows.length; *i*++){

*r* += *rs*.rows.item(*i*).name +' got '

+ *rs*.rows.item(*i*).score + ' points in '

+ *rs*.rows.item(*i*).level + ' \n';

}

*dialog*.show(*r*);

}

);

}

# Functionalities

## Future-based

* Timer handler: use for handle event that is going down of figure
* Key handler: use for handle key event to play game
* Data manager: use SQLite which stores high score information
* Single-touch and single language (English)

## Algorithm-based

# Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Owner** | **Date** | **Content** | **Result** | **Status** | **Comment** |
| PRO\_01 | Group 21 | 10/06 | -Start game  -At *home screen*, press *exit* button to exit game | Game is exited | Passed |  |
| PRO\_02 | Group 21 | 10/06 | -Start game  -At *home screen*, press *high score* button to display top ten high score  -After viewing top ten score, press *back* button to come back *home screen*  -In *home screen*, press *exit* to quit game | Game is exited | passed |  |
| PRO\_03 | Group 21 | 10/06 | -Start game  -At *home screen*, click *option* button  -After that, click *back* to come *home screen* and click *exit* to quit game | Game is quitted | passed |  |
| PRO\_04 | Group 21 | 10/06 | -Start game  -At *home screen*, click *new game* button  -After that, click *exit* to quit game | Game is quitted | passed |  |
| PRO\_05 | Group 21 | 10/06 | -Start game to go *home screen*  -Click *option* button to configure game  -After configuration finished, click *back* to come back *home screen* and click *new game* button.  -In playing status, press *exit* to quit | Game starts with default configurations.  Game is exited | Passed |  |
| PRO\_06 | Group 21 | 10/06 | -Start game to go *home screen*  -Enter view *high score* status, after that click *back* to go *home screen*  -At *home screen*, click *option* to enter *configuring status*.  After configuration finished, click *back*  -click *new game* to start play game | Game starts with the previous configurations. | Passed |  |
| PRO\_07 | Group 21 | 10/06 | -Start game to go *home screen*  -Click *new game* to enter playing status  -Play game until status is *game over* | Game over | Passed |  |
| PRO\_08 | Group 21 | 10/06 | -Start game to go *home screen*  -Enter configure mode by clicking *option* button, after that click *ok* to go *home screen*  -Click *new game* to enter playing status.  -In playing mode, click *pause* button to pause game, after that, click *resume* to resume game.  -Play game until status is game over | -Game starts with the effect of new configurations  -Game over | Passed |  |
| PRO\_09 | Group 21 | 10/06 | -Start game to go *home screen*  -Click *new game* to enter playing mode  -After some minutes, click *back* button to come back *home screen*  -At *home screen*, click *new game* button to enter playing mode  -After that, click *exit* button | -After new game button is clicked for the second time, new playing mode is created.  -After exit, game is exited | Passed |  |
| PRO\_10 | Group 21 | 10/06 | -Start game to go *home screen*  -Click *new game* button and play game until status is game over  -After save score with name that is entered from player, click *back* to go *home screen*  -Press *high score* to view high score  After viewing, click *back* to *home screen* and click *exit* to quit | -After click high score, player will see his/her name in high score list if he/she is in top ten players who play the best  -Game is exited | Passed |  |
| PRO\_11 | Group 21 | 10/06 | -Start game to go *home screen*  -Click *Option* to go configure mode  -Change current configure and click *OK*  -Click *new game* to enter the playing mode  -Play game until status is game over | -Game starts with new configurations | Passed |  |
| SYS\_01 | Group 21 | 10/06 | -Start game to go *home screen*  -At *home screen*, player has *phone call* | Application is exited | --- |  |
| SYS\_02 | Group 21 | 10/06 | -Start game to go *home screen*  -Click *new game* to enter playing mode  -Click *end phone call* | Application is exited | --- |  |
| SYS\_03 | Group 21 | 10/06 | -Start game to go *home screen*  -Click *new game* to enter playing mode  -In playing mode, *device loss battery* | Application is exited and the game state is lost | --- |  |

**Note**: Tetris game which is used for testing is the version 1.1. The best device is Nokia N9. Some another devices done with testing are Nokia N900, Nokia N97 and Nokia N8