**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 TERRIS (QT PLATFORM)**

**WEEKLY REPORT**

# Feature List

Our project is the game Terris. We will use QT platform to implement it. Our version is based on the rules of the classic Terris 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. User can choose to start a new game, view high scores, configure the game or Exit.

Your Name

OK Back

**Input Name Screen**

This screen appears to player when he/she starts a new game. It asks player to input his/her name.

Pause

Menu

Score

Level

**Playing Screen**

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

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.

Option 1

Option 2

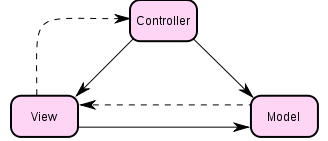
Option 3

OK Back

**View High Scores Screen**

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

# 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 raise 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*);

}

);

}

# Testing

1. Start -> Home screen -><exit> Exit
2. Start -> Home screen -><show high score> Show High Score -><back> Home Screen -><exit> Exit
3. Start -> Home screen -><option> Options -><back> Home Screen-><exit> Exit
4. Start -> Home screen -><new game> Playing -><exit> Exit
5. Start -> Home screen -><option> Option -><back> Home Screen -><new game> Play game -><exit> Exit
6. Start -> Home screen -><show high score> Show High Score -><back> Home Screen -><option> Option -><back> Home Screen -><new game> Playing -><exit> Exit
7. Start -> Home screen -><new game> Playing (lay game until status is game over)
8. Start -> Home screen -><options> Option -><back> Home Screen -><play game> Playing -><pause> Pause -><resume> Resume -> Playing (until status is game over)
9. Start -> Home Screen -><new game> Playing -><back>Home Screen -><new game>Playing -><exit> Exit
10. Start -> Home Screen -><new game>Playing (until status is game over) -> Save Score -><back> Home Screen -><show score> High Score -><back> Home Screen -><exit> Exit
11. Start -> Home Screen -> call phone -> …..
12. Start -> Home Screen -><new game> Playing -> loss battery ->….