

**GTU Department of Computer
Engineering CSE 241/501 – Fall 2019
Winter Homework Report**

**Akif KARTAL
171044098**

1. SYSTEM REQUIREMENTS

Description of the Project: Developing an Android Application that will allow users to play the N-Puzzle app. The application should allow users to select different board sizes from 3x3 to 9x9. Note that the board does not have to be square.

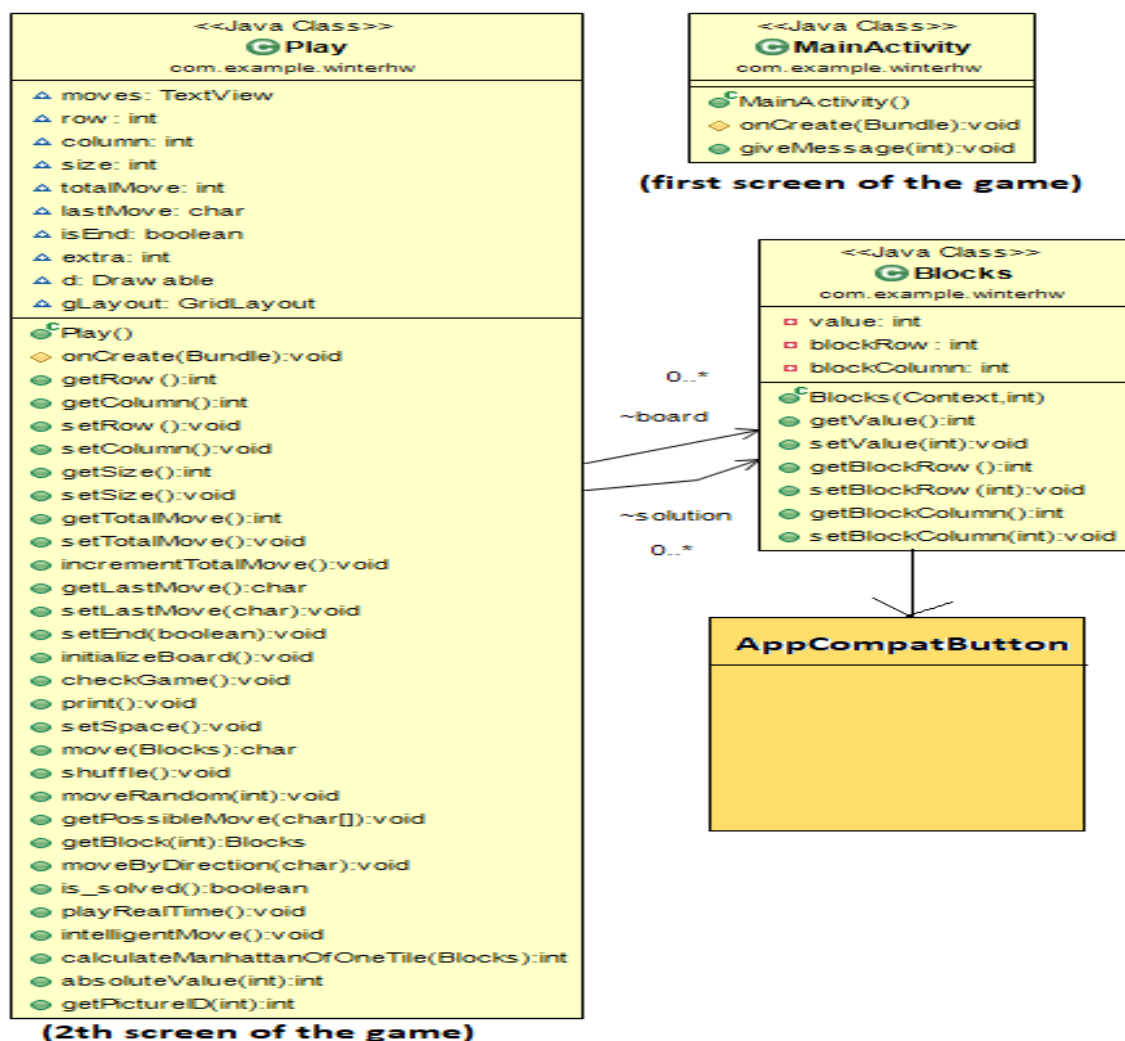
The Application should only create initial board configurations that are solvable. (There is no limit on the number of moves required to solve the puzzle.) The Application should show the number of moves performed during the game. The application should present a hint button that will help users a step toward the solution.

There will be 1 user to play game at a time.

Development Environment

In order to develop the project I used **Android Studio IDE**. Also to test my my program I create a virtual device called **Nexus 5 API 29**. Lastly I used Java Programming Language.

2. CLASS DIAGRAM



3. PROBLEM SOLUTION APPROACH

First I started with Designing entry screen of the game after that I created a second screen to play game. In this screen I used **Buttons** to represent each block on the puzzle therefore to create buttons by using code I wrote a **Block** Class which extends from **AppCompatActivity** Class so that each object of Block class can act like a Button . Lastly in order to put Buttons on the screen properly I used **GridLayout** and I added each button to the grid layout. Thus I can use **OnClickListener** method of buttons to control each button such as swapping two buttons etc.

4. SOME IMPORTANT NOTES


I put directly my **android studio project file** in the zip. You can check and run my project by using this file which name is "WinterHW" and **test cases are below**.

Directory for source codes : WinterHW\app\src\main\java\com\example\winterhw

Also to move Blocks just click or touch them.

Lastly , I didn't use **machine(deep) learning techniques** for real intelligence instead I used classical **manhattan distance** algorithm.

5. RUNNING AND RESULTS

Description	Test Result
Entry Screen of the Game	

Play Screen of
the Game with
3x3 Puzzle



Play Screen of
the Game with
5x5 Puzzle



Play Screen of
the Game with
9x9 Puzzle



Checking
whether Puzzle
solved or not.



**Checking Row
and Column
Number valid or
not .**

