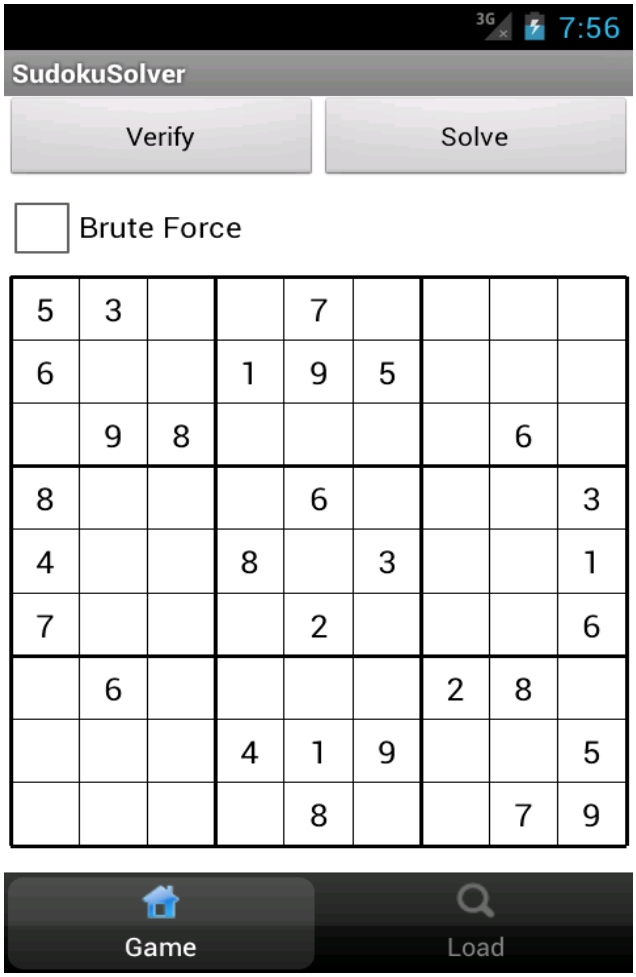
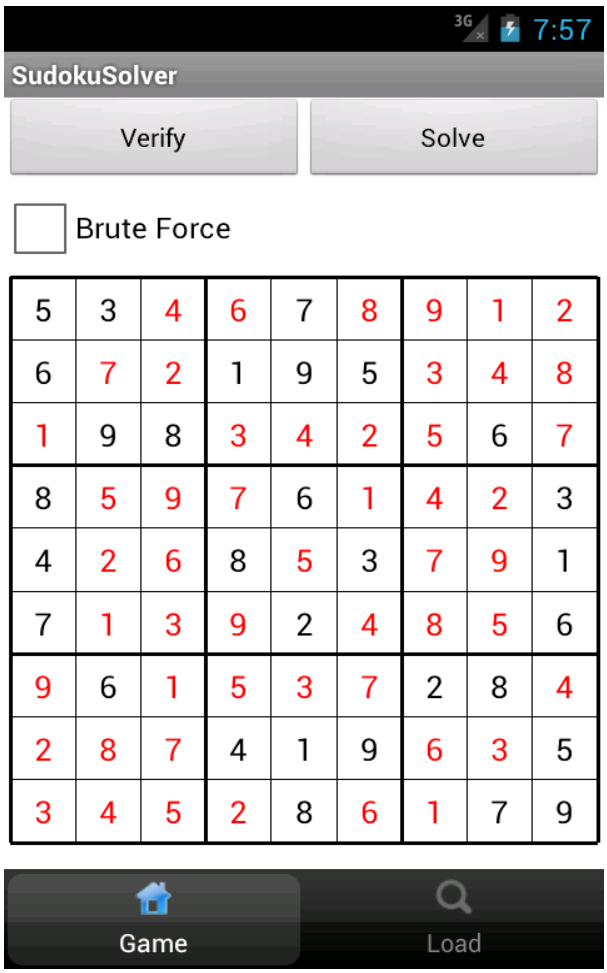
Andrea Bizzotto – Android Sudoku Solver

SudokuSolver has been implemented as a Native Android application. It can be tested on any Android phone and requires an Internet connection.

User interface

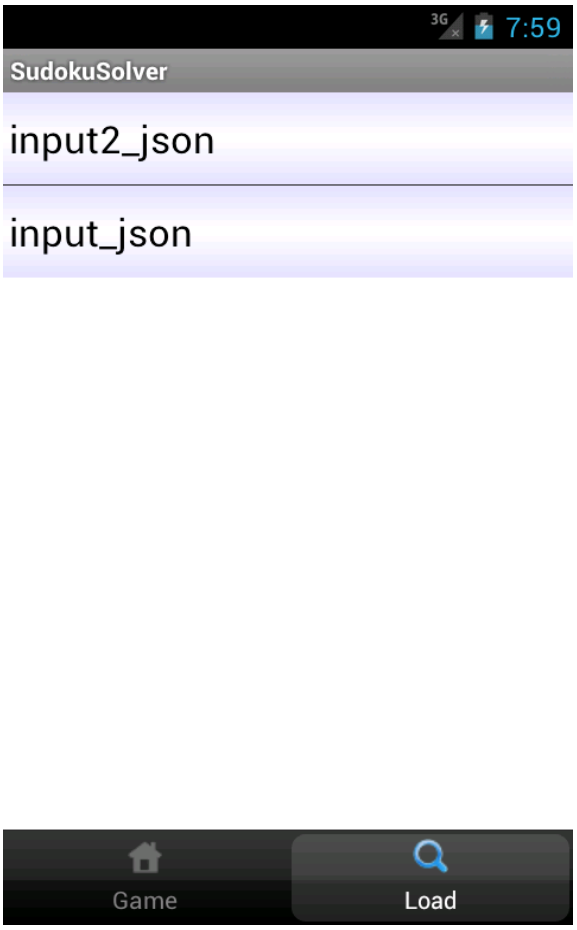
The app is composed by two main activities. The first one is illustrated below and shows a View to show the Sudoku Puzzle as well as some controls to verify and solve it.

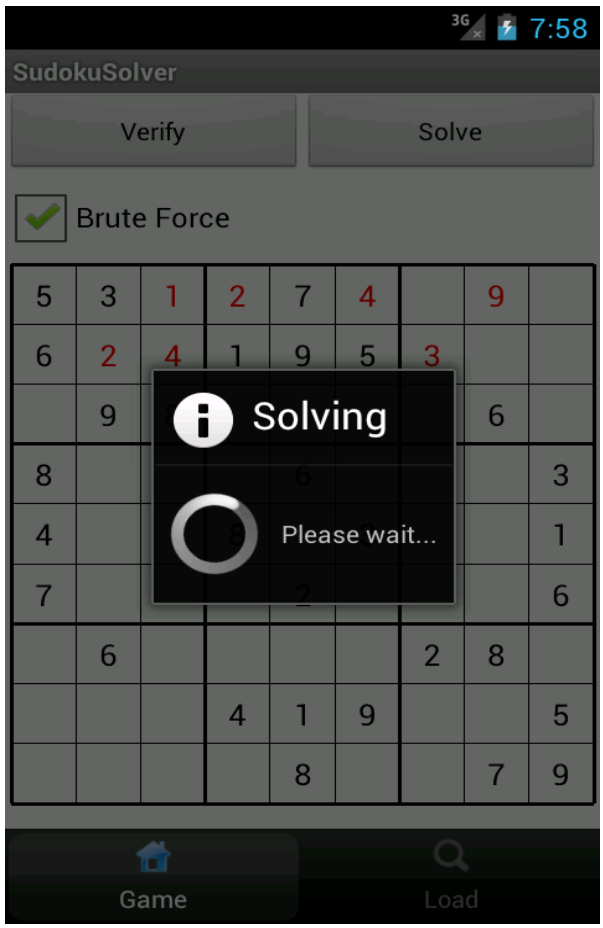




A second view is used to load puzzles from an online server.
Once the view is loaded, a request is sent to a PHP page that scans all *json files in the current folder and composes a global JSON Array of JSON Arrays. This is then sent back to the caller and used to populate a list view[1].

1 NOTE: If no Internet connection is available a message will show telling the user so.





Each element in the list correspond to a valid JSON Array, and can be loaded into the main view by tapping on it.

Sudoku solver

Brute Force

A Brute force Sudoku solver has been implemented and tested on the device. For reference documentation about the implementation please see SudokuSolverBruteForce.java.

It can be observed that as the brute force solver tries all combinations, it doesn't converge in reasonable time and is therefore a not viable method for the assigned task.

Optimised Solver

An optimised solver has been found online (credit: http://www.byteauthor.com/2010/08/sudoku-solver-update/), and imported into the application

code.
This solution converges in a fraction of a second and is therefore the preferred implementation.

How to test different inputs.

As mentioned the app dynamically loads sudoku grids from a web server.

New JSON-formatted arrays can be uploaded to the server by using the following procedure:

ftp musevisions.com

Login: eegeo@musevisions.com

Password: sudokusolver

put file1.json
put file2.json

…

Once the new files have been uploaded, it is possible to load and solve them from the app. Each time a puzzle is solved, it is uploaded again to the server appending a ".solution" suffix to the original file name, and a pop-up message informs the user about this.

Appendix: Error handling

The app expects well formed JSON arrays as specified in the assignment.
Non-well formed arrays will result in silent exceptions being thrown and depending on the type of error the resulting grid may be empty or contain spurious values.