

**CT30A9300 Qt Code Camp**  
**8-puzzle game**  
**REPORT**  
**GRP4**

Jarkko Pulli, 0303117, jarkko.pulli@lut.fi  
Lauri Kyttälä, 0328231, lauri.kyttala@lut.fi  
Petteri Pekonen, 0259122, jouni.pekonen@lut.fi

## Contents

1. Introduction.....	2
2. Description about the idea.....	2
2.1. Features .....	3
2.2. Optional features .....	4
3. What is the problem .....	4
4. Non-technical description .....	4
5. Technical description .....	5
6. Involved technologies .....	6
7. Conclusion .....	6

## 1. Introduction

The idea of the 8-puzzle game is to be an easily understandable and playable game from children to adults. The game has three difficulty levels and the player can use own image or the default number puzzle as the board. The game has a high score list so players can follow up their improvement.

## 2. Description about the idea

8-puzzle is a game where the player tries to get the 8 pieces into certain order on the board where there are 9 slots. The board consist of an image which is splitted into 9 pieces, one of which is left out. The upper left corner is left blank (see. Figure 1).



Figure 1.

In our game we will measure the time used until the puzzle is solved. Using time as a criteria the game creates high score list for top 3 players.

Game can be played using touch screen or using device movement controlling for peace movements.

Game difficulty can be selected between easy to hard. Difficulty is based on mixing pieces. Easy game can be solved about moving 3 pieces and hard can be solved moving pieces 50 times.

Game can get images from phones memory and create a puzzle based on that image. Qt Platform supports a variety of image formats beginning from BMP, GIF and JPG to PNG. The image file Game is solved when the player sees a similar image than the original one (see Figure 1).



Figure 2.

## 2.1. Features

The following table (see Table 1.) shows the implemented and non implemented features and their priority order.

Feature	Priority	Implemented
Board	1	Yes
Piece movement	1	Yes
Game end	1	Yes
Exit	1	Yes
New game	2	Yes
Time/Move	2	Yes/No
Instructions	3	Yes
High scores	3	Yes
Device movements	3	Yes
Changeable image in puzzle	3	Yes
Picture from the Web according the GPS location	3	No

Table 1.

## 2.2. Optional features

Two player's game. Connect two phones and two players. Idea is to play against human player and the one who solves the puzzle first is the winner. Main idea is that on the phone screen can be seen two similar game boards. Player can move pieces only on his own board. Player can see opponents board and piece movements and both players have the same starting puzzle.

## 3. What is the problem

The problem was to make the game as user friendly (easy to use and readable) as possible while maintaining all the necessary functions. The actual functions buttons <new game>, <high scores>, <load image>, <instructions>, <exit> and <difficulty level> plus the actual puzzles should react precisely to the users actions. The other demand was to make the acceleration sensor input react smoothly to user's commands meaning also the sliding of a puzzle piece to another slot.

## 4. Non-technical description

Implementation based on features, priorities and dividing tasks. The idea was first to implement the most critical features like board, piece movement, game end and exit. After that there is a puzzle that can be played.

In the second phase we implemented new game, time counting and loading an image plus splitting it to 9 puzzles.

In the last phase we implemented instructions, high score listing (for 3 top scores), and device movement control system – taking input from the touch screen or from acceleration sensors. We did not have time recourses to implement the desired feature to get an image from the web. We think that counting of moves is useless for a player and counting time is enough.

On the first day evening we had a plan and a vision how to start the actual project. On Tuesday morning we made some practical decisions how to start the actual implementing. We divided coding tasks and documentation among the group members according to their competencies. When the basic coding tasks were done we started the integration.

## 5. Technical description

Technical description about the architecture and class diagram (see Figure 3).

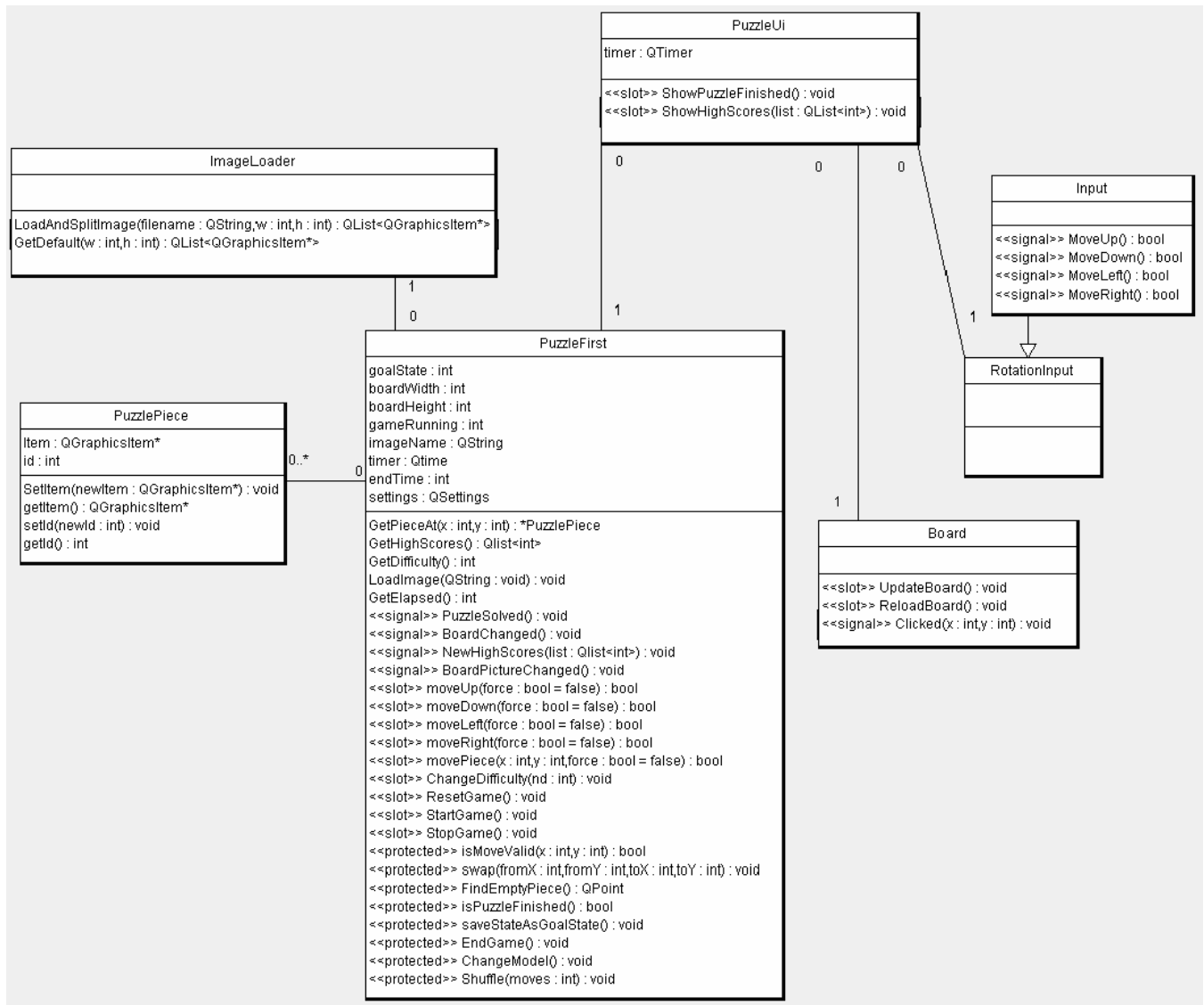


Figure 3.

The user interface consists of two classes `PuzzleUI` and `Board`. `PuzzleUI` handles layouts, buttons and messages to user. `Board` handles drawing the game board and sends signal when board item has been clicked. `PuzzleUI` also has an instance of `RotationInput` class which sends signal when phone has been rotated over 15 degrees.

`PuzzleFirst` class handles board model and all game logic. This means that it checks if move is valid and moves pieces in board model accordingly and sends signal when board has changed. It also sends signals when puzzle image has been changed and when puzzle has been solved.

`ImageLoader` class handles loading and splitting image into 9 pieces. If it fails to load image it will give a default image.

## 6. Involved technologies

We used Qt modules core, gui and mobility in our application. We used gui to make game's user interface and mobility to use rotation sensor.

## 7. Conclusion

Qt is a cross-platform application development framework using e.g. widgets in deployment of gui programs.

Qt contains lot of functions which can be used but it was somehow quite hard to find what you were looking for. When we found appropriate function and got the idea how it works it was quite easy to add new features into our application.

Signals and slots was quite easy to understand and use.