Presentation of the ”GoQ”-application

The content of the project:

Classes:

- GoLBoard: the super class for the boards
- StaticGoLBoard: the static implementation of the GoLBoard, this board has a “zero-row” and a “zero-column” at the borders of the main board that prevents the board from expanding
- DynamicGoLBoard: the dynamic implementation of the GoLBoard, this board expands when “cells” reaches the outer borders of the main board
- Rule: contains rule logic in order to determine the generations
- GameViewer: logic related to drawing the board on a canvas
- ImageConverter: handles the process of converting an image to a byte[][]-representation of the image

Activities:

- MainActivity: the start “screen” where the user inputs information that is further used to create a QR-code
- ShowGameActivity: the “screen” showing when the game is playing, handles inputs in order to adjust size, zoom and the colour of the cells. This “screen” also handles the functionality to take pictures and convert them into a byte[][]-representation using Floyd Steinberg-dithering

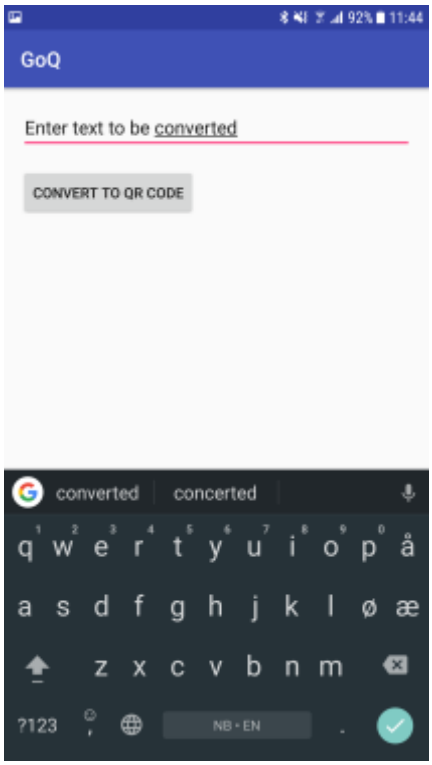
Controls:

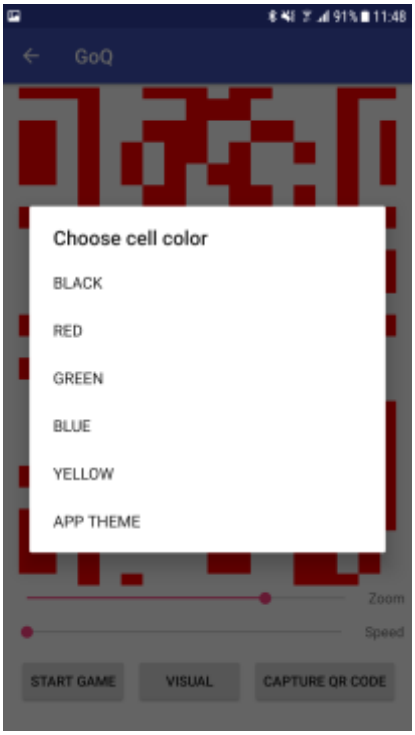
- Convert to QR code-button: a button that receives the input provided in the text field and processes this input in MainActivity
- Enter text to be converted-text field: the text field that receives input from the user
- Zoom-slider: controls the zoom level of the drawing
- Speed-slider: controls the speed of the game
- Start/stop game-button: starts and stops (pauses) the game when pressed
- Visual-button: opens a dialog box that displays the different cell colours to select
- Capture QR code-button: opens a dialog box that lets the user select the used percentage of original dimensions of the image. The percentage could be set to higher values than 10, but (on least) on a Samsung S6 this results in slow animation
- OK-button in Capture QR code dialog box: opens the camera application in order to take a picture

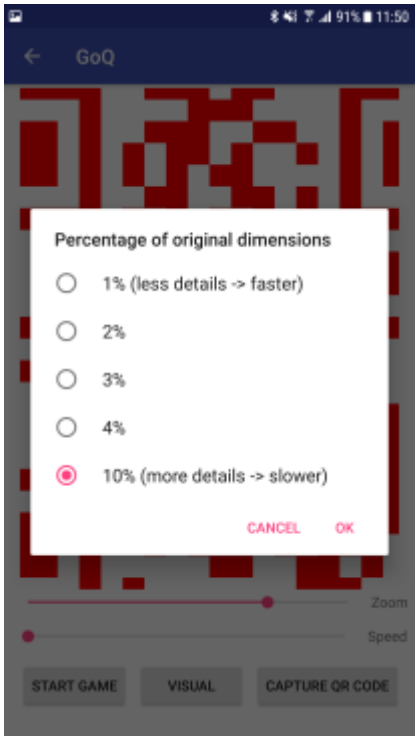
Discussion:
The methods processing the game-data is not optimized and therefore will iterate through every cell of the GoLBoard-object. Optimization of methods related to counting neighbours, updating board containing neighbours and setting the next generation could be optimized in order to increase performance.

The GoLBoard-object could be transferred between the activities in several ways. This is discussed in the JavaDoc of the convertInput-method of MainActivity.java and in the JavaDoc of onCreate-method of ShowGameActivity.java.

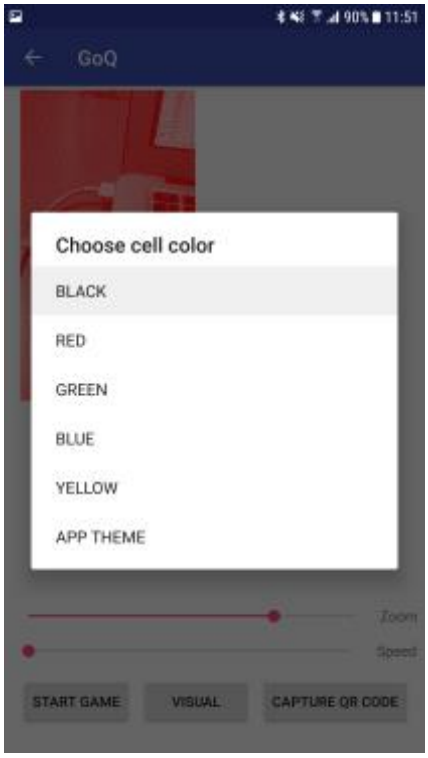
Screenshots:



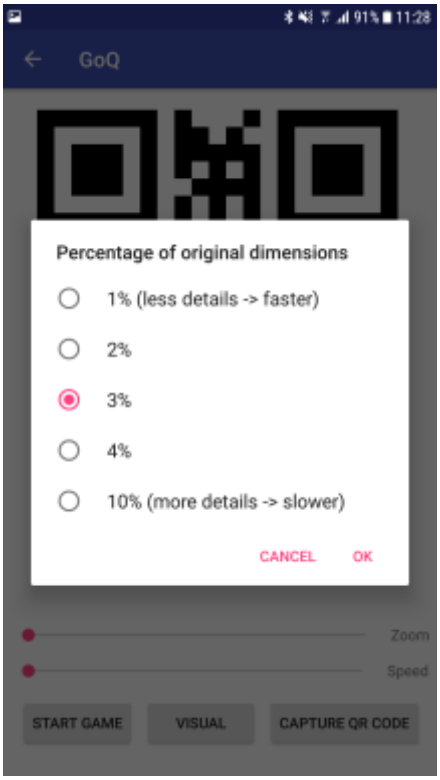





RETRY
OK


GoQ
Choose cell color
BLACK
RED
GREEN
BLUE
YELLOW
APP THEME
Zoom
Speed
START GAME
VISUAL
CAPTURE QR CODE

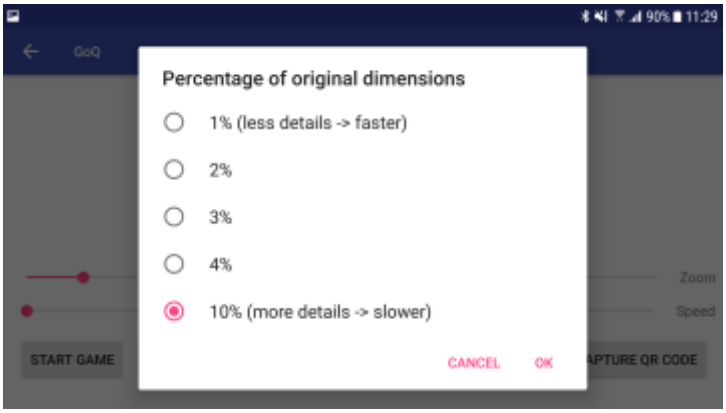

GoQ
Zoom
Speed
START GAME
VISUAL
CAPTURE QR CODE


GoQ
Percentage of original dimensions
1% (less details -> faster)
2%
3%
4%
10% (more details -> slower)
CANCEL
OK
Zoom
Speed
START GAME
VISUAL
CAPTURE QR CODE


RETRY
OK


GoQ
Zoom
Speed
START GAME
VISUAL
CAPTURE QR CODE


11:28
GoQ
Zoom
Speed
STOP GAME
VISUAL
CAPTURE QR CODE


11:29
GoQ
Percentage of original dimensions
1% (less details -> faster)
2%
3%
4%
10% (more details -> slower)
CANCEL
OK
Zoom
Speed
START GAME
CAPTURE QR CODE


11:30
GoQ
Zoom
Speed
START GAME
VISUAL
CAPTURE QR CODE