Requirements specification

The goal is to develop a fully working 2048 clone. Here it will be explained what is defined under "fully working", together with other requirements taken from the requirements analysis:

All of the requirements have each been given a unique identifier followed by a number:

- **R** for game rules;
- **I** for interface requirements;
- **NF** for non-functional requirements;
- **O** for other functional requirements.

Functional Requirements:

Game rules:

Grid

- **R01:** The game is played on a 4x4 grid.
- **R02:** Every cell in the grid can only hold a single tile, because of this every tile has a unique position on the grid.
- **R03:** Every tile on the board has a value that is zero or a power of two.
- **R04:** Tiles are moved within the 4x4 grid and cannot be moved outside of this grid.
- **R05:** Initially, the grid should contain only two tiles with either a value of two or four.
- **R06:** The chance of a new tile appearing with the value two is 90%, in contrast to a chance of 10% on a value of four.

Tile movement

- **R07:** Moves can go into the upper, lower, left and right directions.
- **R08:** Moves are only valid when adjacent squares in the direction of the movement have the same value, or are empty.
- **R09:** A move affects every tile on the grid pushing all of the tiles in the direction pressed.
- **R10:** Making a move will send the tiles in that direction on the grid as far as possible.
- **R11:** Since every cell in the grid can only hold one tile, the tiles that are moved to the same spot collide.
- **R12:** The player can move tiles around on the grid using the four arrow keys on the keyboard.
- **R13:** After every valid move a new tile should appear, with a value of either two or four.

Tile collisions

- **R14:** When two tiles of the same value collide with each other they will merge together into a tile that has the added value of both.
- **R15:** On the contrary, when a tile collides with a tile of a different value it will not merge and will not move on or beyond that tile.
- **R16:** Tiles that have already merged once in the current move, cannot merge again until the next turn.
- **R17:** Merging two tiles adds the value of the merged tile to the score.
- **R18:** Merging two tiles also sets the highest tile ever reached to the new tile if it is higher than the current highest tile.

Winning and losing the game.

- **R19:** The game is won when two tiles of the value 1024 merge into a single 2048 tile.
- **R20:** When no more moves are possible before the game is won, the game is lost.
- **R21:** After winning the game, the user can decide to keep playing or to restart the game.
- **R22:** If the player decides to continue after winning, he will proceed with the current grid.
- **R23:** While continuing, the player is able to get tiles with a value higher than 2048.
- **R24:** While continuing, the player can "endlessly" increase his score until he runs out of legal moves.
- **R25:** While continuing, when the player runs out of legal moves, the player loses the game.

User Interface:

- **I01:** The interface should display a board to the user with a 4x4 grid on it.
- **I02:** The 4x4 grid should display the tiles it holds.
- **I03:** There should be an indication of the current score.
- **I04:** There should be an indication of the highscore.
- **I05:** There should be an indication of the highest tile ever reached.
- **I06:** A replay button should be present that gives the player the possibility of restarting the game.
- **I07:** The tiles should show the value they hold.
- **I08:** Tiles of different values should have a distinct colour compared to tiles of other values.
- **I09:** When the game is won, a dialog should pop up telling the player he won and asking whether or not to continue.
- **I10:** When the game is lost, a dialog should pop up telling the player he lost, asking him if he wants to restart.
- **I11:** When a tile spawns, it should do so with a "jojo" animation.
- **I12:** When two tiles merge, there should be a bounce animation.
- **I13:** When tiles move, they should do so with a sliding animation.

Other functional requirements:

- **O01:** After closing a game, upon restart, the game should start where it left of before being closed.
- **O02:** The highscore should be saved.
- **O03:** The highest tile ever reached should be saved.

Non-functional Requirements:

- **NF01:** The game should run on all major operating systems: Linux, Windows and Mac OS.
- **NF02:** The use of Java 1.8.
- **NF03:** The use of Maven to make sure the software builds and runs in a clean environment.
- **NF04:** The use of Git as a tool for revision control.
- **NF05:** The use of DevHub to make use of continuous integration.
- **NF06:** A working version before the deadline on saturday the 14th of september 23:55 CEST.