**Set up design document**

**Krijn Hoogendorp 8724970**

**Version 4 (November 23, 2014)**

Goal

Create a two-dimensional puzzle with one empty space into which some numbered tiles can slide horizontally or vertically to occupy. The goal is to arrange the board from smallest to largest, as shown in the example below.

Requirements

1. N-puzzle
2. 3 levels of difficulty: “easy” (n=8), “medium” (n=15), “hard”(n=24)
3. Users must have choice of images
4. Image must appear for 3 seconds
5. During play user is not allowed to hit MENUbutton
6. When changing difficulty a new game should start
7. The game´s state should survive after user quits

Optional

1. Ensure pseudorandom shuffle after start/resest game
2. Allow user to use own image (collect through URL)
3. Implement automatic solver

Implementation

The implementation has 3 main parts: Puzzle layout. Puzzle rules, and State

1. The **puzzle layout** uses a dynamic GUI (as a combination of XML and Java), a ListViews (which will hold various options of the game) and a bitmap class (for resizing an image, breaking up an image, and cropping the image).
2. The base for the **puzzle rules** (gameplay) is that each part of image (the tile) will receive a tag which indicates the unique location (most likely a number). The blank tile will be represented with the tab ´blank´. The tiles will be represented in a Table layout form

The gameplay itself will be in two distinct methods: the swap method and the checkSolution method.

The tiles will be swapped when: (1) the tile is touched setOnClickListener) and is next to an empty tile, provided the following constraints: (a) tile is next to blank tile, (b) tile is not swapped on side of board.

The swapping must take place in the internal representation as well as visually.

The method will be as such:

- find tile tapped on (tapped)

- find empty tile (emtpty)

- if tapped is next to empty >

switch empty and tapped

The checkSolution method is to check whether all tiles are in order (and the game is won). The method might be as such (to be further investigated):

The elements are displayed in a TableLayout form. Tiles can be accessed through (x,y). In the correct solution of a 3 x 3 grid, the tags of tile (2,2) should e.g. be 5

1. The **State** method ensures the possibility of continuation of a game after a quit. The following data should be saved to be able to do this: state of board, number moves made, and difficulty level.

Screens

main

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  | start game |  |  |
|  |  |  |  |  |
|  |  | continue previous game |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Start: screen 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  | Choose level |  |  |
|  |  |  |  |  |
|  |  | easy |  |  |
|  |  |  |  |  |
|  |  | medium |  |  |
|  |  |  |  |  |
|  |  | hard |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Start: screen 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  | Choose picture |  |  |
|  |  |  |  |  |
|  |  | UvA logo |  |  |
|  |  |  |  |  |
|  |  | football |  |  |
|  |  |  |  |  |
|  |  | pretty girls |  |  |
|  |  |  |  |  |
|  |  | import own |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Start: screen 3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  | PICTURE | | |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  | quit |  | reset |  | solve |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Class1 -Activities

mainActivity: onCreate (create First screen), onClick (when choosing an option)

playLevelActivity: onCreate (create screen 2), onClick (When choosing an option )

playPictureActivity: onCreate(create screen 3), onClick (When choosing an option)

Class2 (Gameplay) – Activities

startGameActivity: onCreate, onClick, onQuit (touching quit button), onReset (touching reset button), onSolve (touching solve button)

* This activity hold the game rules (the ´swapping´ and checking whether the solution is correct)

statusGameActivity: onCreate, onClick, onReset, onSolve

* This activity saves the status (state of board, number moves made, and difficulty level)

automaticGameActivity

* This activity is optional and is meant to automatically sove a game