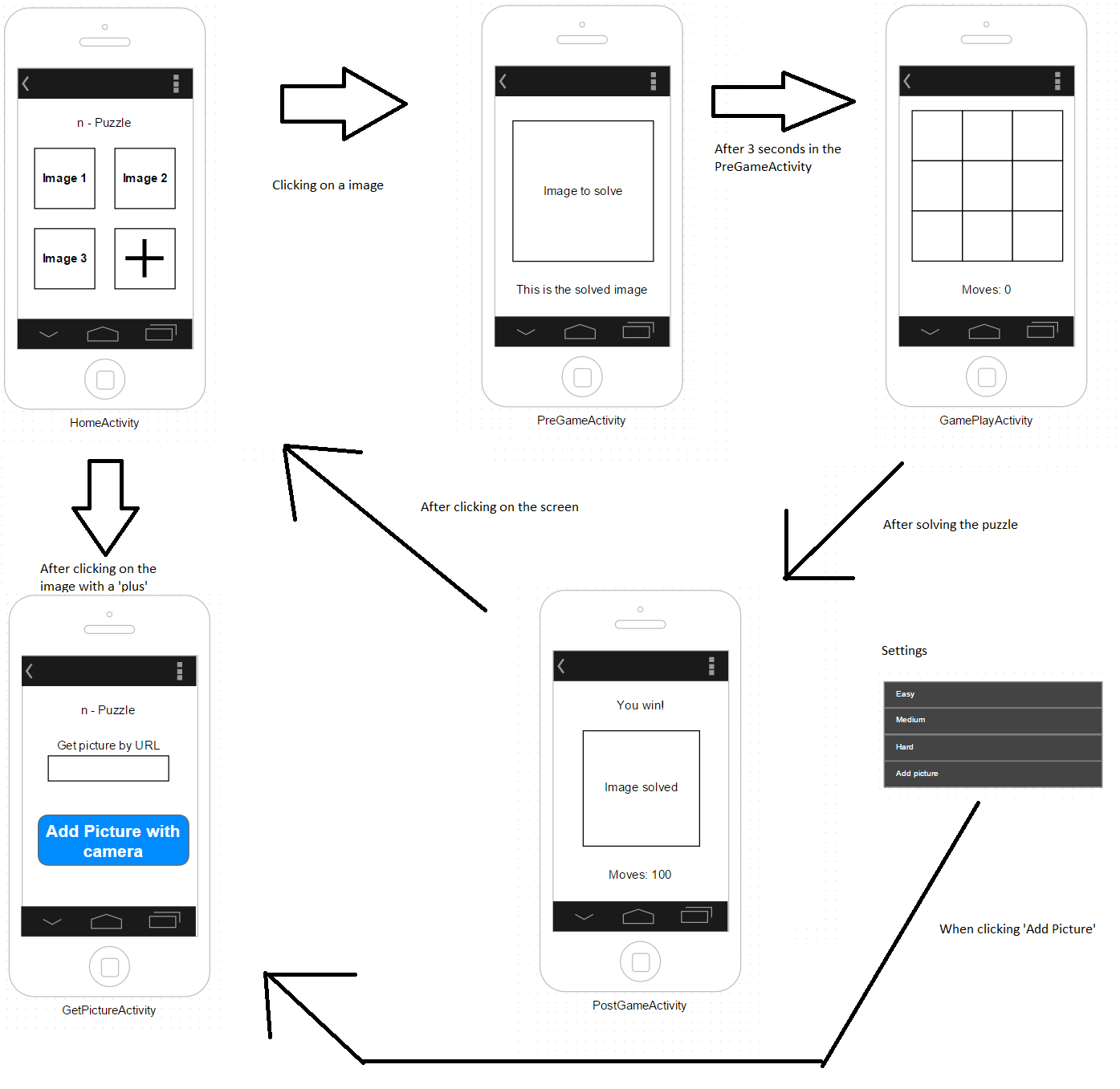
**Design Document**

A game that crops an image into tiles with an empty tile, changes the order and asks the user to solve the game into the correct order again.

**Activities**

**HomeActivity**

Screen with possible images to play the game with and a 4th tile to add your own picture. The images are in a gridview using a custom adapter. Clicking on an image brings you to the PreGameActivity.

**PreGameActivity**

Shows the selected image solved for 3 seconds and then automatically goes go to GameActivity. Load the image using “BitmapFactory.decodeResource(Resources r, int id)” and resize the image using “Bitmap.createScaledBitmap(Bitmap bitmap, int width, int height, boolean filter);”

**GameActivity**

Shuffles the selected picture and makes it possible for the user to solve. When all tiles are in the correct place it brings you to the PostGameActivity.

The picture is cropped using : Bitmap.createBitmap(Bitmap bitmap, intx, int y, int width, int height);

A Tablelayout is used to access the tiles using columns and rows. The empty tile could be a white image and using its id we can locate it every move.

Save the state with onPause() and restore with onResume()

**PostGameActivity**

Shows the selected image solved with the amount of moves done. Clicking on this screen brings you back to the HomeActivity.

**Settings**

Using the SharedPreferences API we store the settings.

**API’s**

**SharedPreferences**

**Bitmap**

Load Image - BitmapFactory.decodeResource(Resources r, int id)

Resize Image - Bitmap.createScaledBitmap(Bitmap bitmap, int width, int height, boolean filter);

Crop Image - Bitmap.createBitmap(Bitmap bitmap, intx, int y, int width, int height);

Get Screen properties - getResources().getDisplayMetrics()

Delete image/Free memory - bitmap.recycle()