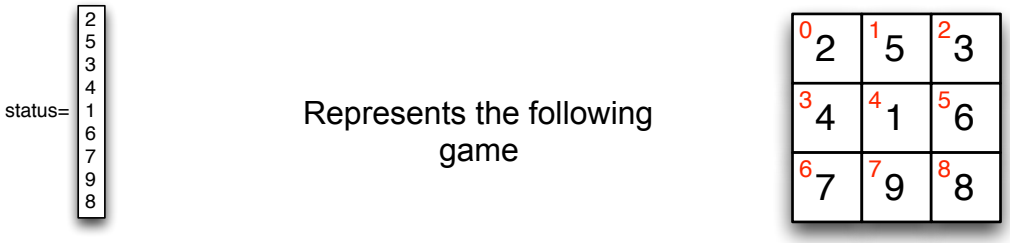
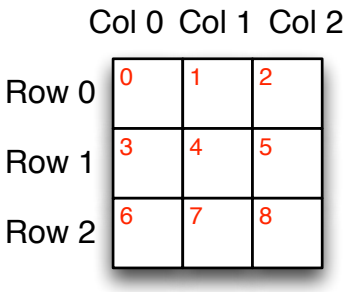
ITIS/ITCS 4180/5180 Mobile Application Development
Homework 3 Help Sheet

- Create a separate class to manage the game logic. For example, GameLogic class could manage all the game related logic.
- Use an integer array (inside the GameLogic class) to manage the game status. The array stores the game state, for example status[0] is the number stores in the upper left block of the game. The below figure shows and example array, note that the empty block is represented by number "9".



- Note that you always need to maintain where is the empty block, this is important because the empty block is the main block that moves and from its index we can derive the game logic such as the possible moves.
- You should be able to easily identify the row and column number of a block at a specific index using simple integer division and modulo computations.



To identify the row #, divide the block index by 3 using integer division. For example, if index/3=0 then row 0.

To identify the column #, compute the block index modulo 3. For example, if index%3=2 then column 2.

- In the GameLogic class create methods that enable you to move blocks into the empty space. Note that there are 4 possible movements up, down, right, left. These moves are not always possible depending on the position of the empty block.
- Note that:
  - No block can move up into the empty block if the empty block is in row 2.
  - No block can move down into the empty block if the empty block is in row 0.
  - No block can move left into the empty block if the empty block is in column 2.
  - No block can move right into the empty block if the empty block is in column 0.
- Note that, when creating the move methods you need to check if the game can perform this move. It is a good idea to return a boolean value indicating if the move was possible or no. For example move_right() method should first check if a block can move right into the empty block using the above logic (empty block not in column 0).
  - If Yes: then swap empty block with the block to its left (index-1). Return true.
  - If No: Return false.

- Similarly moving up and down is checked based on the row number and the swap is performed by subtracting or adding 3 from the index of the empty block respectively.
- In the GameLogic class you can create a method that swaps two entries in the game status array.
- To generate the initial game state, you should start from the goal board with status array {1,2,3,4,5,6,7,8,9} and then randomly permute the empty block using the legal moves. This can simply be done by flipping a 4 sided coin and if it is side 0 then call the move up method, side 1 call the move down method and so on. With respect to the game the move method simply move something into the empty block if it is permissible. Repeat the process 50 times and this will result into a randomly generated status array.
- You should create a simple method that is able to translate a given status array into the required user interface by placing the blocks based on the status array. In the main activity it is a good idea to have a static array which maintains the ImageView ids and another array to maintain the photo ids. Please refer to the Mapping game for more help related to this.