# Cooperating Intelligent Systems Lab 1
# 15-puzzle

August 28, 2012

## 1 Task

Code your own version of the PuzzleEngine.search method implementing A* search using a heuristic function of your own. Submit your implementation and a brief report stating your main achievements and how well your implementation works.

## 2 Code

Breadth-first search for the 15-puzzle is implemented in the supplied java code in the zip-file. The file contains PuzzleApplet.java, PuzzlePanel.java, PuzzleEngine.java, and PuzzleState.java in the fifteen library, and an applet file fifteenApplet.html. Compile the code by typing:
javac fifteen/PuzzleApplet.java
Run the puzzle by typing:
appletviewer fifteenApplet.html
A description of the goal for the puzzle is supplied in the code. When search is started the only element in the queue (also supplied) is the start state. The function will iteratively work its way through the queue. The queue is updated continuously as states are expanded and new states result. The trick with breadth-first is that new states are put at the end of the queue. To implement depth-first you only need to put new states at the beginning of the queue. Importantly, there is no information supplied regarding any queue ordering preference.

# 3 Grading

Your results are graded as either u,3,4 or 5. These are the requirements for the grades:
3. The assignment has an implementation of A* with a heuristic function (number of displaced tiles).
4. Same as for 3 plus an improvment of the standard implementation (e.g. manhattan distance heuristic).
5. Same as for 4 plus a table showing the improvment on different shuffle levels. The table shows n.o. states for the plain A* implementation v.s. the improved implementation and the depth searched.