Black&White Squares Line Part II - Session 1/1

There is no new course content, nor Java templates, specific to this part of the project. However we will comment in class methods that you will need to implement and use within this part of the project.

Session learning goals:

(Wednesday, April 9th)

- To explain the use of methods **isFullyObserved()** and **gatherPercepts()** from object Problem (formulation template).
- To revise **Hill-Climbing Search** and explain the ON-LINE version of it you are to implement.
- *[Not in class but as part of your homework]* To **code an ON-LINE version of Hill-Climbing** that uses the heuristic you defined in your previous submission.
- *[Not in class but as part of your homework]* Use your implementation of ON-LINE Hill-Climbing to solve the new problem setting you are facing in this second part of the project and explain whether your system produces a satisfactory solution or not.

Homework (5% of final grade):

(Submission deadline: April 15th, 17:00h)

- To answer **questions 1 and 2 of the General Problem Case Description PART 2**. These questions have nothing to do with Hill-Climbing in itself. To answer them you need to revise the second part of Chapter 4.
- To implement methods **isFullyObserved()** and **gatherPercepts()** of your BlWhSquaresProblem object.
- To code **ON-LINE Hill Climbing** as a subclass of the HeuristicSearchMethod class. This object will use the heuristic you defined in your last submission and methods **isFullyObserved()** and **gatherPercepts()** from your BlWhSquaresProblem object.
- **Use your system** with the two XML files available in ALUD2; namely blackwhitesquares1.xml and blackwhitesquaresPartialpercepts1.xml.

  Your algorithm should work as regular Hill-Climbing when the problem environment is fully observable and in an ON-LINE fashion when the problem environment is partially observable. Your algorithm also differs from regular Hill-Climbing in that it will keep the name of each operator applied so that on completing its search, the system can produce the sequence of actions leading to the last state reached by the algorithm.

- *Explain whether your system provides a satisfactory solution to the new problem setting* you are facing in this second part of the project.

- *As part of your system testing you must display the heuristic value of the current state and the heuristic value of its best successor*. This will help you understand the behavior of the algorithm and assess its adequacy as a solving method for the new problem setting.

You root folder & your project must be named BlackWhiteSquares_[IS-XX|SI-XX|IA-XX]

- It should have a **doc** subfolder containing a **.DOC** or **.DOCX** with your answers to the questions above that don't involve coding.

- It should have a **lib** subfolder containing the is-search-algorithms.jar in addition to the is-search-formulation.jar from last time.

- On completing your homework, compress your root folder into a **ZIP** file with the same name (**RAR IS NOT ACCEPTED**) and upload it onto our course in ALUD (maximum file size accepted is 5MB).

- **[ENGLISH GROUP only]: Your code must be commented.**