

Laboratory Assignment

Subtask 1



LABORATORY GROUP: A1 – 07

COMPONENTS:

AMANDA SÁNCHEZ GARCÍA

FERNANDO VELASCO ALBA

GITHUB REPOSITORY: A1-07

11/10/2017

## INDEX

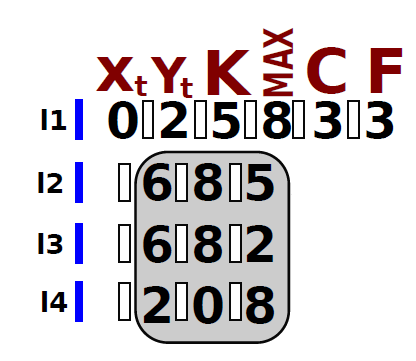
1. A BRIEF RESUME OF THE PROBLEM
2. OUR IMPLEMENTATION OF THE PROBLEM

### A BRIEF RESUME OF THE PROBLEM

The main goal of this laboratory assignment consists of defining, designing and developing an agent program to find the sequence of actions to be taken by a tractor to ensure that all the sand in a field is evenly distributed on the ground. So, all the boxes will have an equal amount of sand K.

The first things to be done are:

* Implement an internal representation of the field.
* Create a field.
* Reading and writing a field from/to a file.
* Generate all possible actions from a field with the tractor in the (Xy, Yt) box.
* Get a new field after applying an action to a given one.

It is necessary to take into account the format of the file where the provided information is going to be:

### OUR IMPLEMENTATION OF THE PROBLEM

### The programming language that we are going to use is Java. We have decided to use it because it is the programming language that we know best, also Java is a very complete language so we will have available all the data structures we are going to need.

### The field is going to be represented as a bidimensional array which boundaries are defined through the file. The file defines the position (x,y) of the tractor in the field, the desired quantity of sand in each square (k), the maximum sand that can be placed in a square, and the number of columns and rows of the field.

### We have implemented five classes:

### **Tractor**. Where the tractor is defined, with the current sand it’s carrying, and its position in the field (row, column).

### **Field.** Definition of the field where the tractor is in, providing the number of rows and columns, the desired amount of sand for each square and the maximum sand permitted per square.

### **FileHandler.** This class is used to manage the file, that is, to have the possibility to read it and write on it. It checks that the file is read correctly, taking into account the format, throwing the corresponding errors.

### **InputExceptions.** Class to check that the format of the file is the correct one (checking that there are only positive integers, the correct use of blank spaces or other error).

### **MainClass.** Main class of the program. Here the field is created and the file is read and written. It also has a method providing the possible movements of the tractor through the field, considering in which position it’s at the moment.