How to prepare input files for Klotski Solver?

1. Open your favorite Klotski program and choose a Level

Here I have taken *Baltic Sea* from freeware *Bricks game* – see Fig.1.

2. Number of all pieces – identical pieces should have got consecutive numbers

Here we have two groups of identical pieces: 7 smallest squares (numbers from *1* to *7*) and 2 horizontal bars (numbers from *8* to *9*). Pieces with numbers *10* and *13* aren't identical because only *10* is a „master" piece – Fig. 2.

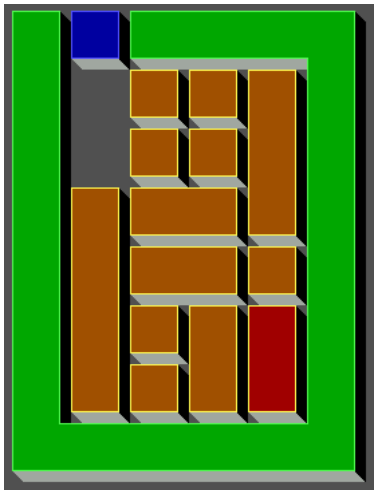

Fig. 1. Original screen shot.

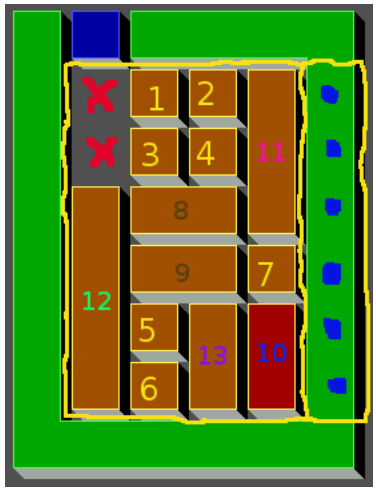

Fig. 2. Numbered pieces and special squares.

3. Open your favorite **plain** text editor and create new text file

It **can** be Notepad, Kedit, Geany...
It **can't** be Word from MS Office, Writer from OpenOffice, ... unless you explicit save files with type *.txt* (plain text).

4. Write size of the board

Our board has at minimum 4 columns and 6 rows, because we can skip borders. We have to (due to the construction of Klotski Solver v1.7) add one additional column behind last our column – see blue dots at Fig. 2. So finally our board has 5 columns and 6 rows and we write these informations at first line our input file – see *input file* at the end of this manual.

5. Write a starting board

Next we have to tell the program how does our board look. So we simply write numbers of pieces in every line. We write zeros for blank squares. So in our example we have to write 6 lines with 5 numbers in each one (overall: 30 numbers) – vide *input file*.

6. Write an ending board and constraints

Next we have to write what does we mean as a “resolved level”. Here we only want to have a piece number *10* at special position (marked with red crosses at Fig. 2).

Additionally we have to specify which squares are not accessible (we mark them with number *-1*) and optionally which squares are accessible but at finish position they should be empty (we write: *-2*) or which squares are accessible only for pieces listed in ending board (we write: *-3*). In our example we have only one column which is not accessible for all pieces so we mark these squares with *-1* – see *input file*.

All remained squares we mark with *0*.

7. Write groups of pieces

At last we have to write amount of identical pieces (it can be *0*, but we have to write this) and specify which they are. So here we have 2 groups: pieces with number from 1 to 7 and from 8 to 9. Now we see why identical pieces have to have got consecutive numbers – we write only first and last number for all groups.

8. Save an input file

That's all!

Input file:

```
5 6

 0  1  2 11  0
 0  3  4 11  0
12  8  8 11  0
12  9  9  7  0
12  5 13 10  0
12  6 13 10  0

10  0  0  0 -1
10  0  0  0 -1
 0  0  0  0 -1
 0  0  0  0 -1
 0  0  0  0 -1
 0  0  0  0 -1

2

1 7
8 9
```