TDLOG – session 3
Extensions

Xavier Clerc – xavier.clerc@enpc.fr
Clémentine Fourrier – clementine.fourrier@inria.fr
Michael Herbst – michael.herbst@inria.fr

7 October 2019

Upload your work to educnet on 12 October 2019 at the latest

The objective of this session is to extend the program of the previous session. The students can actually extend their own programs, or the solution for the previous session available on educnet.

1 Extensions

The first step is to enhance the game with the following elements.

1.1 Characters

The game now accepts multiple characters, at most 4, represented by the numbers from 1 to 4. The characters move successively (*i.e.* not simultaneously), with no constraints over the order of moves. The interpretation of the string typed by the user is extended to allow the encoding of the moves for the various characters. The new format for the strings is the repetition of :

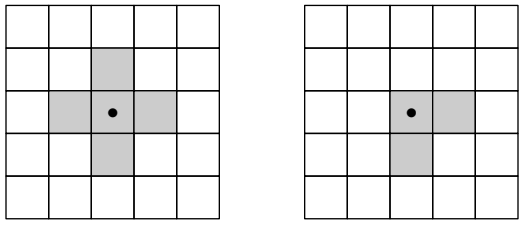
— one integer for the character to move ;
— a possibly empty list of directions, as defined during the previous session.

As a consequence, the string `2>>1^2>` is encoding the following successive moves :

— the character 2 moves towards right twice ;
— the character 1 moves towards top once ;
— the character 2 moves towards right once.

The game is over once any character reaches the door.

FIGURE 1 – Turnstiles : examples.



1.2 Holes

The game now has a new kind of holes, of depth two, the holes defined during session one being of depth one. When a crate is pushed into a hole of depth two, its depth is decreased to one. A hole of depth 2 is represented by the character `0`.

1.3 Turnstiles

The game now has new moveable elements : turnstiles. A turnstile is made of a rotation point in one cell, and from 1 to 4 blocks that can be on the neighbouring cells immediately top, bottom, left, and right. Figure 1 shows two turnstiles, that would be represented as text as follows, *i.e.* with a `%` character for the rotation point and `°` characters for the other blocks.

```
 °
°%°        %°
 °         °
```

Figure 2 shows two examples of turnstile rotations along two columns. On the first line, a character is pushing a block. On the second line, the question marks show which cases should be free for the rotation to be possible. Finally, on the third line, the rotation has been performed and the character has moved to its new position.

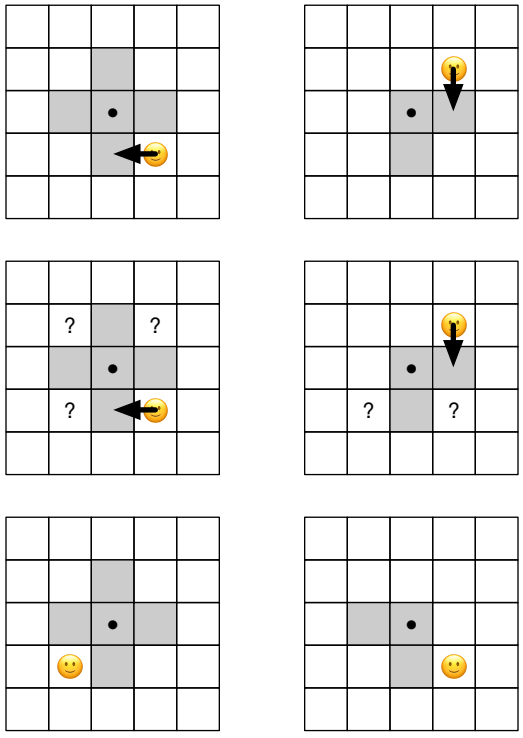
2 (optional) Data loading from file

The step is to modify the program so that the grid is loaded from a text file. The format of the text file is the same as the one used for the display. It is necessary to check that the loaded grid is valid, and to print an error message if it is not.

3 (optional) Pathfinding

The third step is to add the necessary elements to the program in order to compute whether there is a solution, *i.e.* whether there is a sequence of moves for the various characters so that one of them reaches the door. We will be satisfied with a naïve solution, as long as it can

FIGURE 2 – Turnstiles : rotations.



find a path for a small grid (*e.g.* having both width and height equal to 8) in a reasonable amount of time (*e.g.* a few seconds).