



# IASD 2022/23 Project Assignment #1: ‘Roll the Ball’ Slide Puzzle

Luis Custódio and Rodrigo Ventura  
*Instituto Superior Técnico, University of Lisbon*

(Version 1.0, September 19, 2022)

## Introduction

Free the Ball, or Roll the Ball, is a classic tile puzzle where the goal is to move the sliding tiles to unblock a path for the ball to roll to the exit (Figure 1).

The puzzle is similar to the 8-puzzle in the sense that i) it is played in a squared grid of  $N$  cells; ii) it has empty cells to where the movable tiles can be slid; iii) it starts in an initial configuration. It is different from the 8-puzzle since i) it does not have a final configuration but a goal to be achieved; ii) it may have more than one empty cell; iii) some tiles can be movable (can slide) others cannot.

This year’s IASD project consists in developing a program using Python to solve ‘Roll the Ball’ puzzles. The project will be divided in three parts/assignments, each one with a deliverable. This first assignment aims at reading a puzzle from an input file and determining if in the initial configuration the goal is achieved (i.e. the ball can roll to the exit).

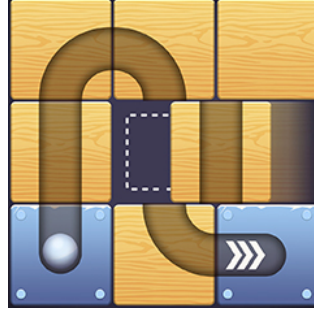


Figure 1: ‘Free the Ball’ Puzzle

## 1 Problem Statement and Solution

The puzzle is represented by a grid of  $N$  by  $N$  cells where some (most of) the cells are occupied by tiles and some (few) are empty. For instance, in Figure 2, there is a 4 by 4 puzzle, where the black squares represent empty cells (4 of them) and the rest of the cells are occupied by different tiles.



Figure 2: An example of a ‘Free the Ball’ puzzle

All puzzles should have two special tiles: i) one that establishes where the ball starts, and ii) another one that defines where the ball should exit. In Figure 2, the former is the tile in the top-right corner and the latter is the left most tile in the 3rd line. Both of these tiles are colored blue because they are not movable.

For the purpose of this project all puzzles are squared (defined by parameter  $N$ ) and numbered from top to bottom (lines) and from left to right (columns). So the puzzle in Figure 2 is numbered like presented in Figure 3. The cells are represented by the pair  $(row, column)$ . The top-left corner cell is always cell  $(0, 0)$  and the bottom-right cell is always cell  $(N - 1, N - 1)$ .

The solution for puzzle in Figure 2 is presented in Figure 4. It involves three moves: (i) slide tile  $(2, 1)$  to empty cell  $(1, 1)$ , (ii) slide tile  $(2, 2)$  to empty cell  $(2, 1)$ , and (iii) slide tile

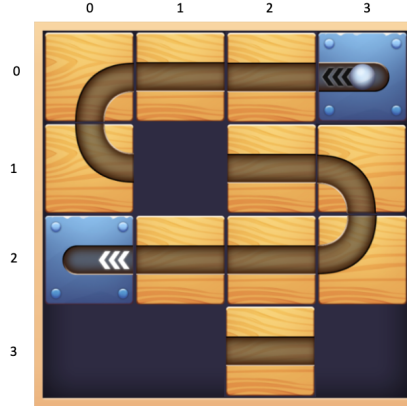


Figure 3: The numbering of a puzzle

(3, 2) to empty cell (2, 2).

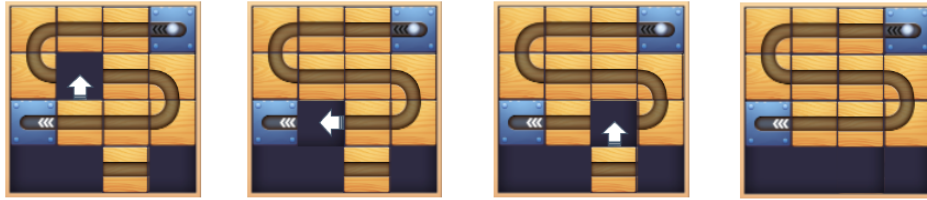


Figure 4: The solution for the puzzle

## 1.1 Tiles

A ‘Roll the Ball’ puzzle may involve tiles from the set of tiles presented in Figure 5. The names used for the tiles mean that the ball can roll in both directions. For instance, in tile ‘right-down’ the ball can roll from right to down, or from down to right. Of course, for the tile ‘no-passage’ the ball cannot pass.

The yellow-wood pattern for a tile means that the tile is movable, i.e. it can be slid into an empty cell. If any of these tiles appears instead with a blue-steel pattern it means that the tile is not movable. For instance, in Figure 6, tile (0, 1) is a **not movable** ‘right-left’ tile, whereas tile (2, 0) is a ‘right-down’, also not movable, tile.

As said before each puzzle has two mandatory, blue-steel pattern, tiles: i) one with the initial position of the ball, and ii) another with the goal position of the ball (see Figure 7 for the sets of possible initial and goal tiles).

## 2 Objective

The goal of the project is to develop a program using Python (version 3) to search for a solution, if any, for a ‘Roll the Ball’ puzzle. The project will be divided in three parts, and

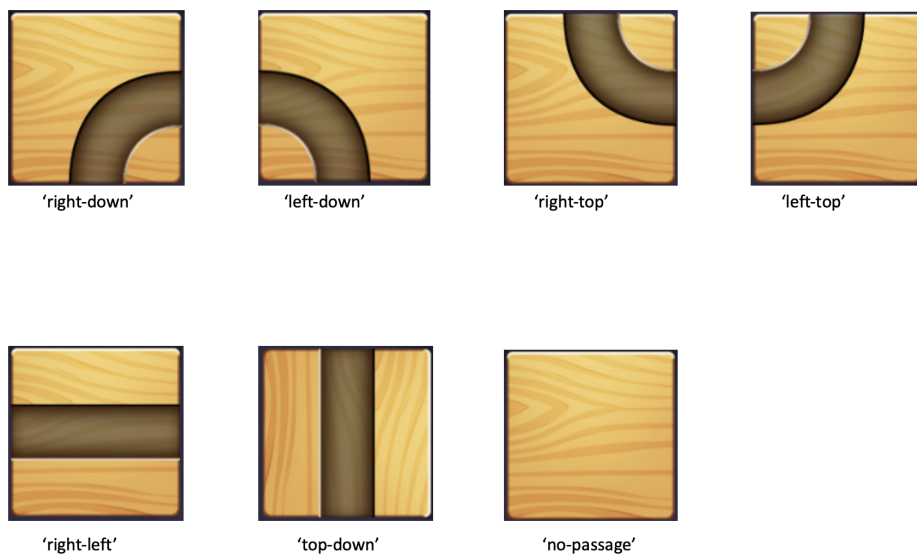


Figure 5: The set of movable tiles



Figure 6: Example with blue (not movable) tiles

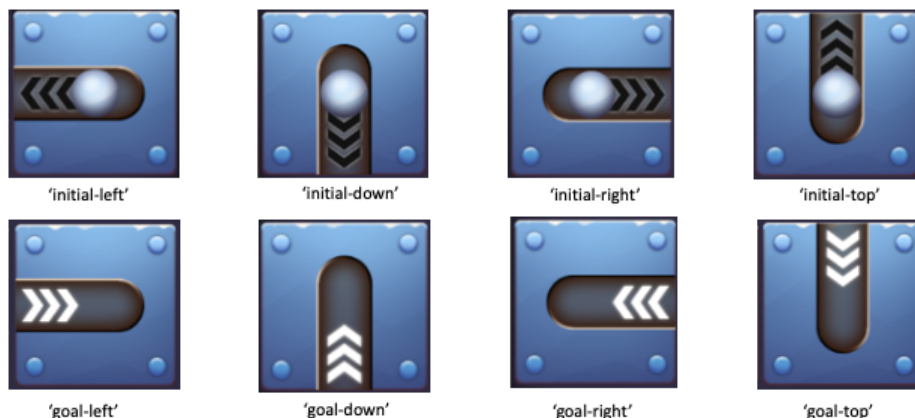


Figure 7: Set of possible initial and goal tiles

three deliverables. This document defines the first assignment and deliverable.

Notice that puzzles may have a solution or not; a puzzle may have more than one solution, and the size of the puzzles vary.

### 3 First Assignment

The first assignment is to develop a Python program capable of reading a puzzle (through an input file whose format will be defined in the next section) and asserting whether the configuration read from the file allows the ball to roll from the initial position to the goal position.

### 4 Input format (dat file)

The input files containing the initial configuration of a puzzle are of type `.dat`. Each contains a single puzzle configuration and is structured as follows:

- The input file may start with one or more comment lines. If a line starts with a `#` is considered a comment and be ignored;
- the first line, after any comment lines, has a single integer number, defining the parameter  $N$ , the size of the puzzle;
- then the following  $N$  lines in the file defines the initial configuration;
- each of these  $N$  lines has  $N$  strings separated by a single space;
- the file ends with a blank line.

The set of strings that may appear in the definition of a puzzle is the union of the following sets:

- A. {“initial-left”, “initial-right”, “initial-top”, “initial-down”}
- B. {“goal-left”, “goal-right”, “goal-top”, “goal-down”}
- C. {“right-left-not”, “top-down-not”, “right-top-not”,  
“right-down-not”, “left-top-not”, “left-down-not”, “no-passage-not”}
- D. {“right-left”, “top-down”, “right-top”, “right-down”, “left-top”, “left-down”, “no-passage”}
- E. {“empty-cell”}

The set  $C$  includes the tiles that are not movable, whereas the tiles in set  $D$  are movable.

## 4.1 Input file: examples

In this section three files are presented as examples of input files for this assignment. The first example is for the puzzle presented in Figure 2:

```
# this is a first comment line
# this is a second comment
4
right-down right-left right-left initial-left
right-top empty-cell right-left left-down
goal-right right-left right-left left-top
empty-cell empty-cell right-left empty-cell
```

The second example is for the puzzle presented in Figure 6:

```
4
right-down right-left-not right-left initial-left
empty-cell right-top left-down empty-cell
right-down-not right-left right-left left-top
goal-top right-left right-left empty-cell
```

The final example is for the following configuration presented in Figure 8:

```
# this is also a comment line
4
right-down right-left right-left initial-left
right-top right-left right-left left-down
goal-right right-left right-left left-top
empty-cell empty-cell empty-cell empty-cell
```



Figure 8: Another example

## 5 Deliverable

The Python program to be developed must read an input file and answer with (return) **1** if the ball can roll from the initial position to the goal position, and **0** if cannot.

The Python program to be delivered should be called `solution.py` and include (at least) a class with name `RTBProblem` containing (at least) the following methods (see Annex A for the class template):

`load(fh)` loads a puzzle (initial configuration) from a file object *fh*

`isSolution()` checks if the initial configuration of the puzzle is already a solution, i.e., the ball can roll from the initial position to the goal position. It returns 1 if the puzzle is a solution, 0 otherwise.

For instance, for puzzles in Figures 2 and 6 the program should return 0, whereas for the puzzle in Figure 8 should return 1.

## 6 Evaluation

The deliverable for this assignment is shown through DEEC Moodle, with the submission of a single python file, called `solution.py`, implementing the modules mentioned above. Instructions for this platform are available on the course webpage. Finally, the grade is computed in the following way:

- 50% from the public tests;
- 50% from the private tests; and
- -15% from the code structure, quality and readability.

Deadline: **7-October-2022**. Projects submitted after the deadline will not be considered for evaluation.

### Closing Remarks on Ethics:

- Any kind of sharing code outside your group is considered plagiarism;
- Developing your code in any open software development tool is considered sharing code;
- You can use GitHub. Make sure you have private projects and remove them afterward;
- If you get caught in any plagiarism, either by copying the code/ideas or sharing them with others, you will not be graded; and
- The scripts and other supporting materials produced by the instructors cannot be made public!

## A Class Template

```
import search

class RTBProblem(search.Problem):

    def __init__(self):
        """Method that instantiate your class.
        You can change the content of this.
        self.initial is where the initial state of
        the puzzle should be saved."""
        self.initial = None

    def load(self, fh):
        """loads a RTB puzzle from the file object fh.
        You may initialize self.initial here."""
        pass

    def isSolution(self):
        """returns 1 if the loaded puzzle is a solution,
        0 otherwise."""
        pass
```