Workshop

Computational Management of Data 25/26

Silvio Peroni

Digital Humanities Advanced Research Centre (/DH.arc)
Department of Classical Philology and Italian Studies, University of Bologna, Bologna, Italy
silvio.peroni@unibo.it – https://www.unibo.it/sitoweb/silvio.peroni/en

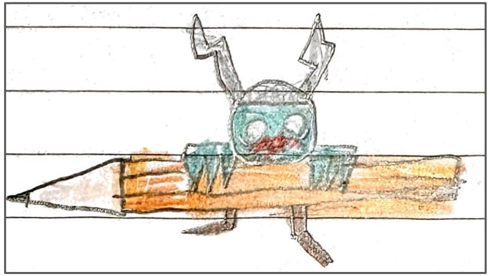
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
DEPARTMENT OF CLASSICAL PHILOLOGY
AND ITALIAN STUDIES

Run In The Arena (RITA)

Picture “Arena” by bartb_pt, https://www.flickr.com/photos/bartb_pt/2289459378/, CC-BY-NC-SA

Plot: premise

What happened in the previous episodes: Myntrakor, also known as Who Must Not Be Thought or the Eminence Behind the Pencil, created the Tower Labyrinth, a complex system made of 100 squared mazes of different dimensions placed one upon the other designed to custody the Book of Indefinite Pages, containing all the possible books ever written in just one portable volume.



Myntrakor hired Urg, a despicable man, as a guarantor of the Tower Labyrinth. Urg implemented a mechanism that allows the walls to move while someone is trying to reach the exit, making it impossible for thieves to enter the Tower Labyrinth, steal the Book of Indefinite Pages, and go out from all the mazes alive.

However, the Interplanetary Guild of Thieves, humans with very bad habits after all, created a special cannon that can be quickly teleported into a maze and can destroy the walls of each maze of the Tower Labyrinth, to guarantee a safe exit to the thieves entered in the Tower Labyrinth.

Plot: the new challenge

While escaping from the Tower Labyrinth thanks to the cannon, the thieves were able to kidnap the troll Mok, the Lord of the Pies, one of the finest collaborators of Myntrakor.

After being judged guilty of all the despicable actions of his Master, Mok was condemned to fight in 100 arenas of different dimensions against the most dangerous living beings in the entire universe: humans! To make this revenge even more bloody, each arena is run for a specific fixed duration and if Mok is able to survive in that timeframe, he can move to the next arena.

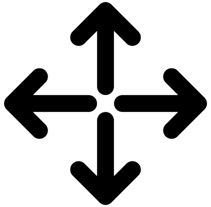
Myntrakor, aware of this fact, ran to the arenas to help his troll. However, it could not enter without being imprisoned by humans and, thus, it did all it could to provide intelligence data for supporting Mok in running away from the humans by suggesting, using telepathy, the position of cells and walls of the arena, the starting position where Mok is, and the updated position of the enemies.

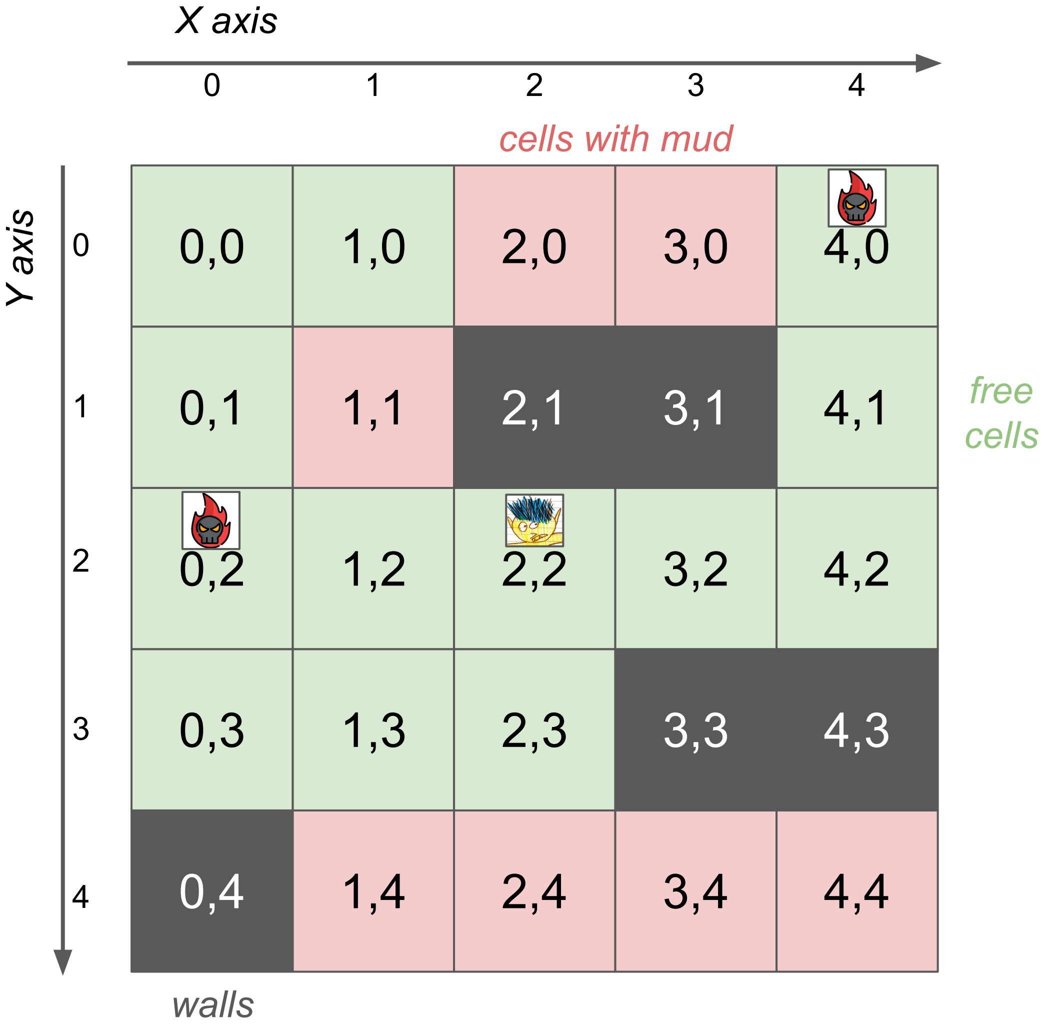
Structure of an arena

Mok

and the enemies

can move only in the following directions





	0	1	2	3	4
0	0,0	1,0	2,0	3,0	4,0
1	0,1	1,1	2,1	3,1	4,1
2	0,2	1,2	2,2	3,2	4,2
3	0,3	1,3	2,3	3,3	4,3
4	0,4	1,4	2,4	3,4	4,4

Rules

1. Each room is a square initially filled with an arbitrary number of free cells, cells with mud and walls
2. Mok is initially positioned in a precise cell, the coordinates of that cell are listed in his diary, and he has to survive for a number of turns (i.e. edge of the arena * 3) in each arena
3. Mok can perform from zero to three steps in each turn, and each step should be in one of the four possible directions, but cannot perform any additional step in a turn if he moves to a cell with mug
4. The enemies move after Mok in a turn, and they can perform from zero to one step, having the goal to move closely to Mok in every turn
5. Mok dies if he is placed in or cross a cell occupied by an enemy
6. Mok is saved by Myntrakor if he is still alive after the given number of turns
7. Mok cannot cheat – and he is caught to cheat if:
    a. he moves more than three steps per turn
    b. he does not stop moving if it is in a cell with mug
    c. he moves either outside the arena, on a wall, or has a brain crash (i.e. Python raise an error)

Function to implement

```
def do_move(arena, enemies, diary)
```

It takes in input:

- **`arena`**, a list of lists of dictionaries representing the rows of the arena and the cells in each row, with the related type of cell (free, wall, mud)
- **`enemies`**, a list of tuples of X/Y coordinates identifying the current position of the enemies
- **`diary`**, a dictionary storing, at the beginning, the starting position of Mok

It returns a list of tuples, each representing a X/Y coordinate, of the steps Mok does in a turn starting from the current position occupied – e.g. **`[(2,3), (3,4)]`**

Example of an arena

```
[                   The main list defining the arena
    [                   The first row (i.e. a list) of the arena
        {   'x': 0,                 X coordinate of the first cell of the first row
            'y': 0,                 Y coordinate of the first cell of the first row
            'type': 'free' },       type of cell (‘free’, ‘wall’, or ‘mud’)
        {   'x': 1,                 X coordinate of the second cell of the first row
            'y': 0,                 Y coordinate of the second cell of the first row
            'type': 'wall' }, ...   type of cell (‘free’, ‘wall’, or ‘mud’)
    ],
    [ … ], …        The second row (i.e. a list) of the arena
]
```

How it works

The file responsible for showing how to run the game is **`run.py`**

The initial part of the file **`run.py`** contains an exemplar 5x5 arena with free cells, walls, and cells with mud, followed by the starting position for Mok, the enemies, and Mok's diary (with listed the starting position)

The function **`do_move`** is included in the file **`group.py`** and will be called in the body of the for loop in **`run.py`** – please note that the code in **`run.py`** does not check if the move is valid or not

To run your implementation of **`do_move`** you can run

```
python run.py
```

(The initial implementation of **`do_move`** will return None, which is an invalid input (it should be a list of tuples defining X/Y coordinates)

Points to the groups

1 point will be assigned to all groups that developed a function that allowed their Mok **to avoid cheating**

1 point will be assigned to all groups that developed a function that allowed their Mok to be saved in **at least 30** different arenas

1 point will be assigned to all groups that developed a function that allowed their thief to be saved in **at least 90** different arenas

Material and submission

All the files are available at
https://github.com/comp-data/2025-2026/tree/main/docs/workshop

At the end, each group must send an email to silvio.peroni@unibo.it where are specified:

- The name of the group
- The members of the group
- The file containing the implementation of `do_move` (rename it from `.py` to `.txt` before attaching it to the email)

The full run script, which you can use to test your implementation of Mok, will be made available before Christmas 2025, while the final results of each group will be published on 19 January 2026

End

Workshop

Computational Management of Data 25/26

Silvio Peroni

Digital Humanities Advanced Research Centre (/DH.arc)
Department of Classical Philology and Italian Studies, University of Bologna, Bologna, Italy
silvio.peroni@unibo.it – @essepuntato – https://www.unibo.it/sitoweb/silvio.peroni/en

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
DEPARTMENT OF CLASSICAL PHILOLOGY
AND ITALIAN STUDIES