
Introduction to AI

Project : $N \times M \times K$ Tic Tac Toe

Not 3D

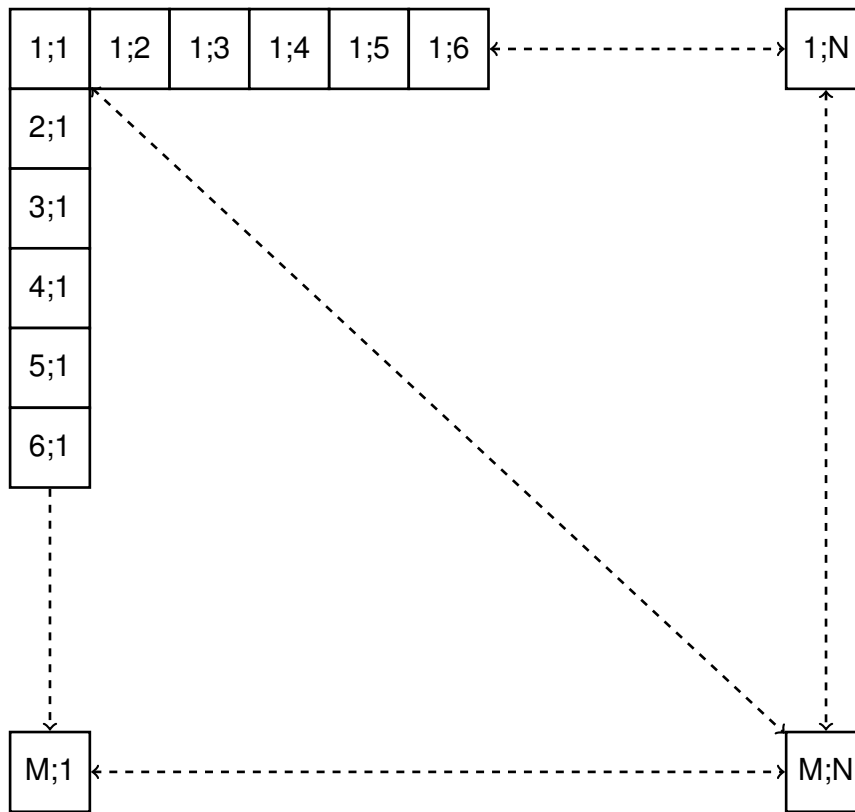
Samy Aittahar

DATE: *3rd November 2017*

1 PRATICAL INFORMATION

- Individual project.
- Deadline: November 23.
- Deliverable 1: A Python implementation of the `Agent` class, as defined in the `agent.py` template.
 - \triangle If the proposed agent action violates the game constraints, the agent misses its turn.
 - \triangle The execution the `move(grid)` method is subject to a time limit.
- Deliverable 2: A report of 3 pages (max.) which describes your work. Appendices with some statistics accepted.
- Upload your deliverables (`agent.py` and `report.pdf`) on the Montefiore submission platform.

2 $N \times M$ TIC TAC TOE GRID



This is a classical tic tac toe grid. The grid can be really large, and not necessarily square.

Usually, we denote 'X' and 'O' for each player token, but in this project we denote them as 1 and 2 (0 is empty).

The rules¹ remain the same, except the following. Instead of filling entire diagonals/rows/columns with the same symbol, players have to build diagonal/row/column alignments of size k (denoted as a k -alignment). The score of a player is the number of diagonal/row/column alignments of size k , given that two alignments can share (at most) 1 cell.

The goal of the game is to have more points than the opponent until it is not possible for any of the players to build more k -alignments. The game is made more challenging with a time budget of 1 minute. Plus, if a player hit an alignment, he also grabs the next turn.

Below are some common scoring examples with a 10×10 grid and $k = 5$.

¹For a quick reminder : <https://en.wikipedia.org/wiki/Tic-tac-toe>

				X					
				X					
				X					
				X					
				X					
			X						
		X							
	X								
X					X	X	X	X	X

Figure 2.1: Here, the score for the player X is 3

				X					
				X					
				X					
				X					
				X					
				X					
				X					
				X					
X	X	X	X	X	X	X	X	X	

Figure 2.2: Here, the score for the player *X* is 4