Introduction to AI

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

Not 3D

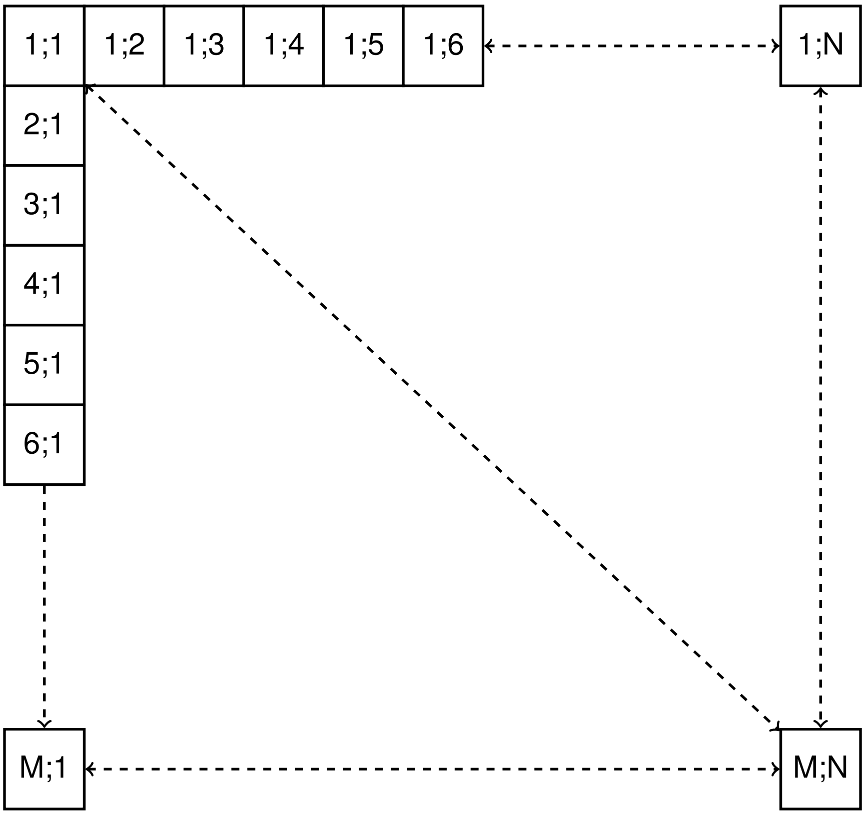
Samy Aittahar

Date: *26th October 2017*

1 Pratical Information

- Individual project
- Deadline: November 30
- Deliverable 1: A Python implementation of the `Agent` class, as defined in the `agent.py` template.
  - ⚠If the proposed agent action violates the game constraints, the agent misses the turn.
  - ⚠The execution time of the `move(grid)` method will be limited.
- Deliverable 2: A report of 3 pages (max.) which describes your work. Appendices with some statistics accepted.
- Upload your deliverables (pattern : *name*_tictactoe.{ pdf|py }) on the Montefiore submission platform.

2 $N \times M$ Tic Tac Toe Grid



This is a classical tic tac toe grid. Notice that 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 will go for 1 and 2 (0 is empty).

The rules[1] 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) with the same symbol. Players can use the fact that two alignments can share (at most) one symbol.

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.

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

[1] For 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