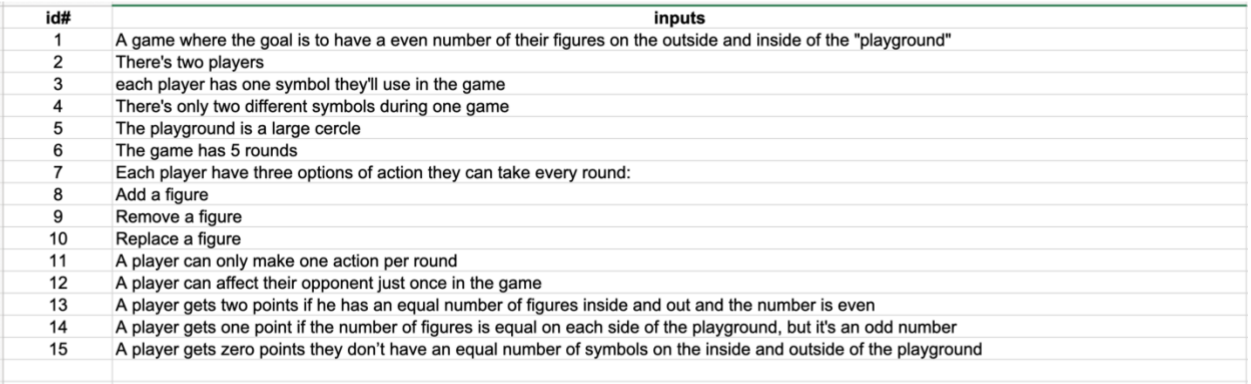
I improved the prompt injections by adding line cuts instead of giving the AI one whole paragraph of instructions so he can get a clear idea of each instructions

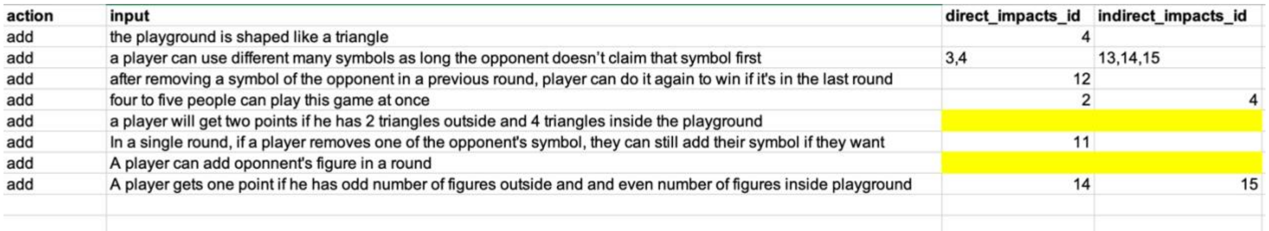
Added the rules logic in the rest of the prompts so it takes in consideration the rules all the time and not just when adding something on the canvas, which improved Dynotate's response a lot, it's a lot more accurate now.

To create a benchmark of the same format as the Mars game for the other project, I started writing rules like like "a cat should be next to a box" similar to what we have been doing to test the rule logic quickly.. But then I thought I should just use Arabi's game rules like we planned to do and wrote the rules down in an excel sheet this way:



id#	inputs
1	A game where the goal is to have a even number of their figures on the outside and inside of the "playground"
2	There's two players
3	each player has one symbol they'll use in the game
4	There's only two different symbols during one game
5	The playground is a large cercle
6	The game has 5 rounds
7	Each player have three options of action they can take every round:
8	Add a figure
9	Remove a figure
10	Replace a figure
11	A player can only make one action per round
12	A player can affect their opponent just once in the game
13	A player gets two points if he has an equal number of figures inside and out and the number is even
14	A player gets one point if the number of figures is equal on each side of the playground, but it's an odd number
15	A player gets zero points they don't have an equal number of symbols on the inside and outside of the playground

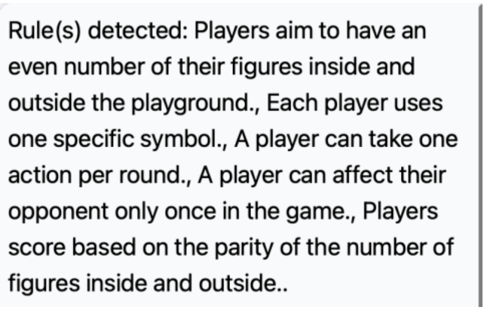
And in another sheet, I wrote a list of inputs that either directly or indirectly contradicts/impacts the initial rules. Also added some examples of inputs that should not go against the game rules(see yellow highlighted rows) to test the AI in various ways.



action	input	direct_impacts_id	indirect_impacts_id
add	the playground is shaped like a triangle	4	
add	a player can use different many symbols as long the opponent doesn't claim that symbol first	3,4	13,14,15
add	after removing a symbol of the opponent in a previous round, player can do it again to win if it's in the last round	12	
add	four to five people can play this game at once	2	4
add	a player will get two points if he has 2 triangles outside and 4 triangles inside the playground	🟨	🟨
add	In a single round, if a player removes one of the opponent's symbol, they can still add their symbol if they want	11	
add	A player can add oponnent's figure in a round	🟨	🟨
add	A player gets one point if he has odd number of figures outside and and even number of figures inside playground	14	15

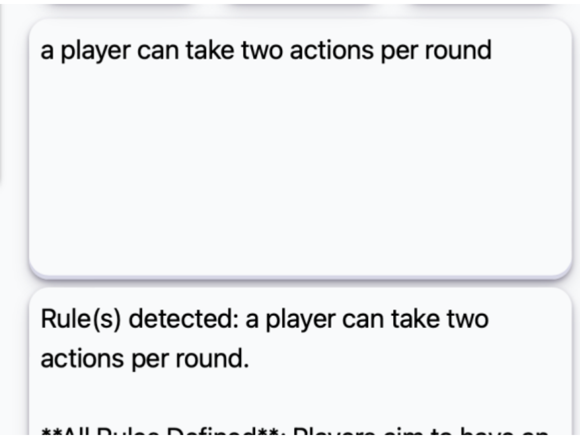
Dynotate test?

First, in one input I told the AI that I'm going to explain a game and gave it the rules of the game as it is in the first excel sheet but added "\n" to make it easier for the AI to understand. This was the output:



Rule(s) detected: Players aim to have an even number of their figures inside and outside the playground., Each player uses one specific symbol., A player can take one action per round., A player can affect their opponent only once in the game., Players score based on the parity of the number of figures inside and outside..

It seems to be adding rules in general in a good way but not storing the details, for example, yes the player can take one action per round but it didn't store the specific actions they can take.



a player can take two actions per round

Rule(s) detected: a player can take two actions per round.

All Rules Defined: Players aim to have an

Dynotate is still limited to the DEFINE, ADD, RULE logics, so I shouldn't expect it to output in the terminal or in the app that I'm contradicting my initial rules.. I guess this should be improved next..?