After implementing the moving target for part two of the project, we let the search make use of this extra information by simply scanning the grid and improving the probability of finding the target in the two cells that are of Type1xType2 that the moving target function reports back. Specifically after we move the target and get the report of the Type1xType2 back, we have a function called update that will scan the grid and count the number of matches the scan finds and then change the probability of those cells to 1.0/#ofMatches. That also has to do with our belief state, because it will help observe other parts of the grid that the search didn't get to yet and also allows for the search to narrow down where the target may be, especially after a number of failed searches. The probabilities of the cells that could possibly contain the target goes up so a general part of the grid will have a raised probability. This also can be explained by the general application of Bayes Theorem. After every try to search for the target, we can update the probabilities based on what gets reported back and if the target moves maybe 50 times, again, a general focused part of the gridworld will have a higher probability then the rest of the grid.

	Rule 1:	Rule 2:	MovingTarget:
Counts:	4143	12984	2164
	1975	6444	2330
	6610	310	3103
	37798	18053	2646
	2144	253	4219
	11726	2102	5152
	6470	7324	7048
	9536	134260	1643
	571	29236	144
	5986	39580	573
Average count:	8695.9	25054.6	2902.2

As you can see, compared to Rules 1 & 2, the MovingTarget one is generally more tight in terms of its count while the other two search methods seem almost random. When we use the rules to search for the target, it just looks at the probabilities of whether it may be what our belief state considers to have the highest probability of containing the target or finding the target. However, with the MovingTarget rule use, we do not search to update our belief state, we search based on

our belief state. What I mean by that is that in the explanation above I explained how the new rule changes how we view the belief state, so it actually narrows the cells to search down, while the other two rules simply look and hope that the highest probability has the target.
Type1xType2 > Rule1 > Rule2.
If we are comparing question 4 between the Rule 1 and Rule 2 and the MovingTarget rule, even then the MovingTarget rule is going to be faster in the average case. The extra information of what Type1xType2 cells contain the target allows for the “agent” to update it’s belief state if the first search is a failure and then it can then travel to the closest Type1xType2 matches and then update it’s belief state if it fails there. Again, as said before, after a couple of failed searches there will be a general part of the grid that is “focused” with a higher probability then the rest of the grid. The “agent” can prioritize that area of cells and use less “actions” on randomly traveling to high probability cells that may hold or contain the target. Especially with a moving target, the probabilities in our belief state of what cell may hold a target is always going to be a state behind or off, considering the target moves one cell randomly after each failed search.