

Andrew Stephenson
Tanton Jeppson
CS470 – Grid Filter Lab Report

The most obvious thing to state that we learned is not only how the grid filter application actually works in AI, but also how simple it was to actually program (thanks to the “head start” we were given with pseudo code). After frobbing the specific values used for a bit, the agents were actually able to detect the obstacles and free space pretty early on in our implementation of this lab. The difficult part for us was finding an efficient way keep track of what parts of the world had been explored and where to explore next (especially if there were obstacles on the map that the agents were particularly prone to getting stuck on). As we brainstormed up some ideas on how to handle this issue we ended up settling for a mix that was time based (especially to avoid getting stuck) and had a few options for the agent to explore next.

To explore the world our tanks were assigned the nearest unexplored region of the map. We broke the map into 20x20 sections to avoid the tanks needing to explore every single pixel of the map. If the tank had explored a section and was found in that same section again 10 iterations later we assumed he was stuck there and would use a random spot from the grid as his next destination to get him out of that area. Once he was unstuck the nearest unexplored part of the map was then used again until the whole grid had been explored.

As far as using different sensor parameters the biggest difference for our implementation was the noise of the graph. The range was not overly affected. The noise could be overcome, but it took more time. As far as moving objects we can only speculate that this would have been affected too due to the noise. With the higher time required to with more noise the object would have moved more and the tank’s belief of that region of the map would be skewed/inaccurate.

In a Capture the Flag game we actually feel like our implementation would do pretty well. The tanks are always moving which helps to avoid being shot. They also have some aspects of the algorithm that aren’t necessarily predictable (like if a tank gets stuck and is then assigned a random region of the map to then explore). One thing we might change for a real game of Capture the Flag is the threshold of acceptance for what qualifies as “explored.” After seeing how well our implementation was able to recognize some of the obstacles we would be able to lower the threshold to save computation time. This would allow the map to be explored quicker, although technically a little unsure of where the obstacles truly are at.