Tank AI Analysis and Strategy

Gilman Thomas

Shawnee State University

For the task of creating a tank AI for the tank battle, I have created my tank in a way that it prioritizes attacking the closest tank within line of sight. The second task the tank performs is just simple bounds checking to adjust its movement. I chose to prioritize shooting calculations over dodging calculations due to the limited time frame of operation given. The tank first creates a class object and adds each tank to a map inorder to handle code organization and lookup speed. This also allows me to keep track of which direction a tank is going, easily calculate the direction to fire in, and the distance to the other tank. On each turn my tank first updates the position of all other tanks if still alive and returns a list of alive tank names and distances back, it then checks to see if it is able to fire before performing any calculations. If the tank can fire, it passes the list of tank names and distances to a function which sorts them from closest to furthest away, then performs a line of sight check, checking the fire angle and position of tanks against every barrier's edges. This is done for each tank until the closest tank in sight is found, in which it returns the angle to fire at. If an angle to fire at has been found, the tank changes the default action to fire and changes the action angle to the fire angle, and proceeds to return the command. Otherwise, the tank proceeds to check if its move angle will collide with any of the barriers or the sides of the world in which it will then update the move angle or occasionally pick a random angle.

In testing my tank, I have tried parallelizing the fire angle to barrier checks to see if this would improve performance, and through observation and how few tanks are on the board in a match, the parallelization decreased performance and caused the tank to timeout if too many checks are performed. I could probably tweak this in a way that would make it perform better, but from my observations so far, using the main thread to perform all the tasks appears to be faster. Prioritizing firing at the closest tank allows my tank a much higher chance of hitting the other tank, and taking the time to shield when mostly moving around in a somewhat random way seems like it would put my tank at more of a disadvantage than it would benefit it, unless i implemented code that would allow my tank to take cover. Implementing a take cover move does not guarantee my tanks safety, as the move may also jeopardize my tank as it could put it in the line of sight of other tanks. By attacking all other closer tanks first; this gives my tank a good opportunity to get a lot of points quickly as long as it stays alive long enough. Some possible improvements that could be made to my tank are better aiming with firing ahead of the other tank as long as the tank is visible or will be visible. Simple bullet checking, to possibly allow for dodging of bullets that will hit the spot moving to if the bullet is less than or equal to 168 units away from that spot ( bullet travels 8 ticks, edge of tank is 20 away from center and tank moves 1 tick), or move in time if the bullet is 8 ticks away from next position. These are some possible improvements that can be made to allow my tank to better perform against the other tanks, as long as the order of operations is optimized and can perform these checks in the 3milisecond time frame given. Given the operation of my tank, it should perform moderately ok being able to move around the map and possibly dodge bullets and throw other tanks off with somewhat random movement, as long as the majority of the bullets hit, the other tanks don't have time to dodge them, and the tank is placed next to a bunch of other tanks.