Wilson Ding - wxd130130

Andrew Zimmerman - alz140030

CS 4365

AI CTF Final Report

wxd130130alz140030Agent.java

One of the interesting phenomena about the maps is that two separate optimal paths (four if you are counting both sides, east side start and west side start) cover all of the provided maps in the game. As a consequence, we were able to define these paths, which became our strategy for approaching this game.

By default, our two agents would identify themselves by their initial spawn location (North vs South), as well as their base location (East vs West). From there, the North agent would begin a pre-defined routine of four moves to determine the given map (whether it was a SimpleMap, EmptyMap, TrapsMap, WallMap, XMap, or the SecretMap). During this time, the South agent would begin moving horizontally towards the enemy base. Within the four turns, the North agent would be able to identify the map, and share this information to the South agent via a static variable. From there, the agents would follow different pre-defined routines for 1. Simple/Empty maps, 2. Traps/Wall/X maps, or 3. Secret map. During these routines, we managed death by checking for the obstacles around the agents, and matching it to the original obstacles at initialization (spawn location). If the obstacles matched, then the agent was assumed to have died, and would restart the pre-defined routine at the very beginning.

In the simple and empty map, we first studied the behavior of the simple agents. The simple agents would move diagonally towards the flag, converge on it, grab it, and then head directly back to their base. Ironically, the simple and empty map seemed like the hardest possible map to actually beat the simple agents in, as in other maps, the enemy agents would often do the work for us, getting themselves stuck in a trap. Our solution was to attempt to kill two agents with one of our own. First, because no diagonal moves are allowed, we determined that there is no true “shortest path” on the simple and empty maps, as they would still require a certain number of moves. As a result, once our north agent determined we were on a simple or empty map, it headed directly in front of its own flag. It would then place a mine in front of its own flag, then go on a suicide run towards the enemy base. It would be a very short time before the first enemy agent collided with it. However, when the second enemy agent behind it moves in to take the flag, it would run into the mine previously laid by the north agent. By doing this, we kill both agents with only one. Meanwhile, the south agent is able to recover the flag, with no one to defend it. On the way back, the south agent would go up one first, then travel all the way back, then move down to deposit the enemy flag, since that kept it out of the line of fire of agents below it, since the enemy agents would be travelling parallel to the base.

For the Traps, Wall, and X maps, we found that the best solution would be to have the South agent continue to traverse horizontally until it was underneath the enemy base, and then go straight up, retrieve the flag, and then retrace its steps to head back to the own base. This creates a “U” shape, hugging the bottom and sides of the map. Meanwhile, the North agent handled the defense, planting mines around the base. The North agent would make sure not to mine the south-side, so that the South agent could return to the base. We did not find these two maps to be very challenging to beat the SimpleAgent, as the SimpleAgent would usually get stuck on these maps and have no chance at winning at all.

For the Secret map, given that it was an unknown environment, we decided to have both the North and South agents revert back to the standard SimpleAgent algorithm to greedily traverse through the map, attempting to retrieve the flag and bring it back to the base. We felt that this was the best option, as we could not make any assumptions on the specifics of the map.

By far the most important heuristic in our solution to this game is the first four moves. By sending the north agent on a “scouting mission” to see if it could determine what map it was on, and all the while sending the south agent to prepare a strike to capture the enemy flag, we were able to optimize the quickest and safest path for our south agent to intercept the flag, while keeping it away from risk of roaming or returning enemy agents. We were further able to optimize the south agent’s safety by using our north agent as a pawn, in some cases, to clear the board of possible enemies.