Throughout the semester we have been developing an AI ship to traverse through SpaceWars, this time the game took a turn making it a team based game, capture the flag. With that we had to implement a multi agent cooperative system between our ships. To accomplish such goals we had to split responsibilities between the ships. Since we were in the competitive ladder we focused on an offensive strategy while trying to build our army of ships. To achieve that we started the game with 3 ships and through the game hopefully ending with 6, where the goals of the ship entailed as such:

* The first ship created will only go for the flag
* The second ship will defend the flag area
* The third ship will gather resources
* The fourth ship will harass enemy players by shooting down the nearest enemy at a time
* The fifth ship will gather resources
* The sixth ship will defend the flag area with number two

When delegating responsibilities for ships we thought it was optimal to start with a ship doing one of each task given to it, where any additional ship will further aid our cause. The reason the sixth ship is a defender instead of a gatherer in the competitive ladder is because there will only be one flag active for the enemy team to take at a time as well as the harasser already inhibiting the other team from capturing the flag by playing the entire field. This allows us to maximize resource gain with the fifth ship while having a solid defensive structure in place.

While ships are being built, it is also determined that if a base can be built that only the flag carrying ship will be able to buy said base. This is done because the flag carrier is already travelling the optimal path to the enemy’s flag, and when a base gets dropped in its path it will inherently shorten the path and optimize point gain. The key to our strategy was to defend our own flag while giving our flag carrier ship the shortest path it could take to get to the flag.

In the early stages of the project we had put just one ship up on the ladder to see how it did by just going for the flag, as it turns out that little ship singlehandedly won the ladder most days. Because of those results found on the early ladders, it was then determined that the flag carrier did not need any additional defense in place, as the potential ship defending it could be put to better use elsewhere.

The way we implemented our multi-agent system was by keeping a list of current active ships and movement targets to our Actions class to sort which ship will do what. During the whole process of each game the responsibilities for each ship will not change even when the ship dies because the UUID for each ship stays static the entire time. Once the lists are fed into the Actions class, the ship that needs to have an action determined will go through a switch depending on which ship created it will be, it will then follow the logic behind whatever role was delegated to it at creation.

Implementing actions the way described above is done to optimize the agent in a way that there is no need for it to think dynamically in that if there is a closer option it will change its responsibility. The reasoning behind keeping things static is mostly done from findings in the early projects, changing directions in this environment is a costly operations and keeping momentum keeps fuel costs low and points high.

There were many more things we implemented in this project from previous findings as well. The traversal method we used was a combination of our A\* mechanics but done differently. When we implemented A\* into SpaceWars originally we went with a roadmap method in that each mineable asteroid or beacon was a node on our dynamic traversal graph. In capture the flag that was an unreasonable request because at one time there is only 6 mineable asteroids/beacons to obtain at a time. We had to create a new graph which contains 64 nodes that we could traverse the map with ease and allowed us to minimize collisions between other ships.

Along with our traversal method it was found in our Genetic Algorithm a relatively optimal max velocity to travel at in which we used here. Using that max velocity allowed us to minimize collisions with asteroids because of momentum shifts, especially while collecting the flag.

Finally, since we had chosen the competitive ladder the entire semester we had retained the same logic through the year to determine the highest valued target as well as the optimal time to shoot. When we had created those algorithms it had worked out very well for us and allowed us to win many nights in the ladder.