

The first knowledge representation is mostly egocentric, focused on finding the nearest objects to the given ship. Its method `pickNearestEnemyShip` finds the closest enemy ship by comparing shortest distance between the `Toroidal2DPhysics` space. The method `findNearestBase` returns the nearest base in the same way. `PickNearestBeacon` functions similarly to return the beacon closest to the ship. Already implementing some decision making, `pickHighestValueFreeAsteroid` determines the best asteroid by comparing the shortest distance and the value of the asteroid in a linear combination. This knowledge representation was a good choice because it allows easy access to the most important aspects of the state. The ship mostly needs to know about which objects it can reach quickest.

The heuristic used for the one ship is contained in `getAsteroidCollectorAction`. Because the highest priority is not dying, the heuristic first checks if the energy is below a certain value (2000). If it is, the next goal should be to get more energy, so the ship moves to the nearest beacon. Before collecting more resources from asteroids, we need to check that the ship is not full. Otherwise, there would be no point to mine more asteroids. So if the resources on board the ship are greater than 500, the ship moves to the nearest base to drop off those resources. If the ship is not dying and has room for resources, then it moves to the nearest asteroid to mine it.

Our second knowledge representation incorporates the nearest beacon, enemy, and base provided in the aggressive heuristic, but adds the following abstractions: number of neighbors within a radius of an object, whether a resource is probably attainable, mineable and non-mineable sets of asteroids, benefit-cost ratio of resources, and highest valued resource using the benefit-cost ratio. The mineable and non-mineable sets of asteroids are useful when determining obstacles to maneuver around versus obstacles to collide with. The number of neighbors near an object, among other things, allows the agent to determine whether a resource is attainable; for example, energy nodes are destroyed by missiles and collected by players, so if a node has too many missiles or players within a radius, it is likely the node will not be available by the time we reach it. Thus, it is good to anticipate when navigating to a resource, burning precious fuel, probably won't be beneficial.

The heuristic that makes use of our second knowledge representation is also adapted from aggressive heuristic with some key differences. Rather than only seeking energy nodes when the ship is low on fuel, the heuristic also considers the ship's bases as refueling stations. Decision between seeking a base or energy node is decided by the benefit-cost ratio of energy available to the distance needed to travel. Additionally, energy and resource nodes considered for collection are judged on attainability. Too many ships or missiles surrounding a node or too long of a distance might deter the agent from making an action to said node; instead, it will cause the agent to wait for a more fruitful opportunity.

We chose the cooperative option because it allows for more clear cut decision making. You do not have to consider when to shoot or how to avoid the missiles of others. This creates a more deterministic search.