Yuanhao Ruan ruanx054

My strategy is: first divide the visible planets to conquered and unconquered List. During the process of classifying planets, use two Hashmaps to store the information of neutral planets and their IDs and store the information of opponent planets and their ID.

When it's my turn, start from all the conquered planets and do a breadfirst traversal to determine the planets I want to send people to if they are visible to me. The planets were added to the priority queue according to numbers of their neighbors. If they had more neighbors, they would have a higher priority. If the target planet is neutral planet, first I check if I can defense my planets, I will send 1 people to this neutral planet(by getNumber methods to determine how many people to send). If the target planet is opponent planet, I will check if I can defense, then check if can attack that planet and I will send the people to that planet (by getNumber methods to determine how many people to send). If the target planet is my planet, check if can defense first, then send most people to that planet to supply that planet.

Data structures used: List, dictionary(HashMap) and queue(Priority queue).

Used list because it is easier to get, retrieve and traverse elements in it.

Used dictionary because it can store multiple information at the same time, for example, I used planet ID as key and IPlanet as value.

Queue was used used during the process of traversal. Queue is a first in first out data structure. When the planets were traversed, they were stored in priority queue and could be taken out from queue according to the priority.