Project 4 Report

We have implemented a strategy that comprises of the following logics and moves:

1) Split the list of available planets into two categories of conquered and unconquered planets.
2) We iterate over every conquered planet and apply the following logics in the form of methods or direct implementations:
    a. If there is at least one neutral neighbor, send 1 person to all such neighboring neutral planets.
    b. Check if all the neighbors are conquered.
        i. If they are all conquered, get the neighbors who have at least one enemy. Send reinforcements to those neighbors.
        ii. Else, wait till the current population becomes greater than 90% of the maximum size of the planet and then send reinforcements equally to all the neighbors.
    c. Pick the strongest neighboring enemy if it exists, and send half of the current population to the enemy.

We have used the following Data Structures in our strategy implementation:

1) ArrayLists –
    - To store the list of conquered planets in the graph.
    - To store the list of neighboring planets who have enemies attacking on them.
2) Graph – Implemented our own data structure as an adjacency list (MyGraph.java) which contains two data members
    - Vertex: ArrayList which stores planet ids of all the planets in the graph.
    - Edges: Implemented as HashMap which maps Each vertex planet to another Hashmap (maps Adjacent Planets to the edge weight).

    Note: We couldn't implement JGraphT due to some technical difficulties, so we created our own graph structure.
3) HashMap – Other than the use mentioned above in graphs, we've used them to map planetID to IPlanet implementation.

Note: We have also used Function overloading in implementing the above methodology with the following two prototypes.

private void sendReinforcement(IPlanetOperations, Queue<IEvent>, IVisiblePlanet, HashMap<Int, Int>)

private void sendReinforcement(IPlanetOperations, Queue<IEvent>, IVisiblePlanet, List<IVisiblePlanet>)

Known Issues:

1) We are not able to effectively direct our reinforcements towards the frontline allies attacking the enemies. (This could have been implemented using BFS).
2) We are making the planet to wait before sending reinforcement till its population becomes greater than 90% of its maximum population. However, this wait should be dynamically dependant on the planet's habitability.