Spike: Spike_11
Title: Tactical Analysis

Author: Daniel Newman, 6449921

Goals / deliverables:

Develop a system that utilises tactical analysis to compare against random behaviour and exploit predictability from other systems.

Technologies, Tools, and Resources used:

Python 3.6.4
Python compatible IDE

Tasks undertaken:
In order to test the tactical analysis of our bot, we had to determine a set of rules we wanted it to follow to aid in determining its behaviour. In order to fulfil the rules we wanted to implement, we needed to gather some information about our world state and create a representative model

```
class TA1(object):
    def update(self, gameinfo):

        #Keep a list of my planets and non-my planets positions
        if gameinfo.my_fleets:
            return

        planet_details = {}
        for planet_id, planet in gameinfo.planets.items():
        # Find a list of the planet with the a short distance away from my planets
            planet_details[planet_id] = {
                'ID': planet_id,
                'reference': planet,
                'owner': 'me' if planet_id in gameinfo.my_planets else 'enemy' if planet_id in gameinfo.enemy_planets else 'neutral',
                'fleet_size': planet.num_ships,
                'ships_required_to_take': 0 if planet_id in gameinfo.my_planets and planet.num_ships > 500
                else 100 if planet_id in gameinfo.my_planets and planet.num_ships < 500 else planet.num_ships,
            }
        for enemy_fleet in gameinfo.enemy_fleets.values():
            planet_details[enemy_fleet.dest.id]['ships_required_to_take'] += enemy_fleet.num_ships




        source_planet_details = list(filter(lambda planet: planet['owner'] is 'me', planet_details.values()))
        target_planet_details = list(filter(lambda planet: planet['owner'] is not 'me', planet_details.values()))

        for i in range(0,source_planet_details.__len__()):
            if i < target_planet_details.__len__():
                gameinfo.planet_order(source_planet_details[i]['reference'], target_planet_details[i]['reference'], target_planet_details[i]['ships_required_to_take'])
            else:
                return
```

Our rules defined that if a planet we held had a significant number of ships (> 100) it was allowed to send out ships to other planets. This ensured that planets wouldn't leave themselves too exposed. This gave us a list of available planets to send ships from, we then gathered another list which contained all the planets which didn't belong to us and each player owned planet was assigned a target to send their ships to. Each player owned planet only sent the required number of ships to take the planet at that moment.

What we found out:
The Tactical analysis behavior allowed us to create more complex and interesting behaviour that would provide more dynamic responses to a players actions. In this case, the rules we implemented in the game led to behaviour where the bot doing tactical analysis would rapidly spread out its forces, taking new planets, which in turn gave the bot a larger pool of planets to use after each new conquest. This gave a distinct advantage over bots that didn't utilise tactical analysis as you can see from the data.

Rounds	TA Bot	Non-TA Bot
1 (204 turns)	Win (90% planets claimed)	Lose
2 (422 turns)	Lose	Win (100% planets)
3 (237 turns	Win (100% planets claimed)	Lose
4 (290 turns)	Win (100% planets claimed)	Lose

Over multiple rounds we found that the TA bot had a significant advantage in most maps. From the data we can illustrate that the TA bot captured all planets within 200-300 turns, almost half the time it took the Non-TA bot to capture all planets, with a round total of 422 turns to take the round.

However the TA bot was not with problems, the TA bot didn't take into account the growth rate of the enemy planet when sending its forces, so it was often sending enough to take the planet at the time, but once the ships reached it was no longer enough to take the planet. Also as the TA bots planet pool grew and in turn the non-TA bots shrunk, the TA bot didn't reinforce other fleets that were being sent to the remaining non-TA bots planet pool, since one planet was assigned one target, once one list ran out the rest of the planets we're told to simply do nothing. This could be improved in future iterations.