Stellaris Combat Simulation Phase 2 Goal

echelon

June 2021

General Idea

- Basic draft of *Enemy* class and *Combat* class. In this phase, the enemy does not fight back but is initialized with specific shield/armor/hull values. In addition, in this phase, a I/O system to export and import fleet details into/from local json file needs to be implemented.
- The goal of this phase is to test the time it takes for fleets with different setup to destroy an enemy dummy target such that we could test what setup could optimize the fire power for a fleet.

Specific TBD

I/O system

- export(fleet: Fleet) $\rightarrow$ local json file: encode fleet details and dump into a local json file.
- import(fleet_number: str) $\rightarrow$ fleet: Fleet: load from a local json file and decode into a Fleet object.
- existing_fleets() $\rightarrow$: read the local json file and return a list of fleet_number of existing fleets in the json file.
- check_existing(fleet_number: str) $\rightarrow$ boolean: check if the local json file contains a fleet with fleet_number same as the input. The purpose of this is to avoid export fleet with same reference into the local json file such that all fleet stored in the file have different fleet_number.

Enemy Class

- enemy_number: str: reference of enemy.
- spec: dict(): a dictionary of undamaged value of shield/armor/hull.
- shield: float: current shield value.

- armor: float: current armor value.
- hull: float: current hull value.
- has_shield: boolean.
- has_armor: boolean.
- is_destroyed: boolean: current status of the enemy.
- receive_damage(type: str, damage: float) → None: type is shield/armor/hull specifies what value should be decremented. damage is how much should be decremented. In this process, shield/armor/hull values should be updated according to inputs and has_shield/has_armor/is_destroyed should be updated according to the previous 3 values.

Dummy Combat Class

- fleet: Fleet: the fleet that deals damage
- dummy: Enemy: the enemy that receive damage
- duration: int: the duration of the combat
- has_end: boolean: current status of the combat
- __init__(fleet: Fleet, dummy: Enemy) → None: initialize the combat.
- start() → combat_result: str: start the combat and terminates when one candidate runs out of hull value. Every tick, the combat update weapons of vessels, fire weapons if possible, update the remaining specs of dummy enemy and update its is_destroyed boolean along with has_end boolean. When terminated, return a brief details of the combat including the duration and remaining specs of the dummy enemy.