COS214 Project 2021 **Group: Hexamaniacs**

Task 1: Design

Page Number / Location

Link to Github Repository:
https://github.com/trashtaste/HexamaniacsFinal/tree/main/Task

1.1. Functional Requirements

1. A system that simulates the launches associated with SpaceX and Starlink, needs to be designed and implemented.
2. Different variants of “Falcon” rockets (“Falcon 9” and “Falcon Heavy”) need to be created.
3. The “Falcon” rockets must have two stages.
4. Each stage of the “Falcon” rocket must comprise a certain number of “Falcon 9” Cores and “Merlin” Engines or “Vacuum Merlin” Engines.
5. The “Falcon 9” first stage must have a single “Falcon 9” core and 9 “Merlin” engines, and the second stage has one “Vacuum Merlin” engine.
6. The “Falcon Heavy” first stage should have 3 “Falcon 9” cores and 27 “Merlin” engines, and the second stage has one “Vacuum Merlin” engine.
7. In both variants, the first stage gets the payload and second stage almost into orbit, while the second gets the payload fully into orbit.
8. Different variants of “Dragon Spacecraft” rockets (“The Crew Dragon” and “Dragon Spacecraft”) need to be created.
9. “The Crew Dragon” must be able to send both Humans and Cargo to the International Space Station.
10. “Dragon Spacecraft” must be able to send only Cargo to the International Space Station.
11. “Starlink” satellites must be launched into a Low Earth Orbit on “Falcon 9” rockets.
12. “Starlink” satellites must be able to launch in clusters of up to 60 satellites on one “Falcon 9” rocket.
13. Satellites need to be able to communicate with each other by making use of lasers.
14. Satellites need to communicate with users on the ground by making use of radio signals.
15. An interface to setup simulations must be created.
16. The interface must be able to run a simulation in test mode, while building the simulation.
17. The interface must also allow the simulation to be run as if it were a real launch.
18. Simulations in test mode should allow interruptions and adjustments and then continue running afterwards.
19. An actual launch simulation needs to be setup and run.
20. Actual Launch simulations must have the functionality to be stored and run in batches.

1.3 Pattern Selection

A list of the patterns chosen and how they participate in the overall design, based on the identified functional requirements.

1. Composite: At face value we saw that any given rocket is composed of two stages, where each stage is made up of a number of cores and engines. We decided that a Composite pattern was best suited to represent this part-whole hierarchy, in which we may treat each stage of the rocket and the components it is composed of as one.

2. Template: The template method was used to implement the different variations of rockets (Falcon 9 and Falcon Heavy) , engines (Merlin and Vacuum Merlin) and spacecraft (Crew Dragon and Dragon Spacecraft). All concrete classes inherit from their respective abstract classes.

3. Strategy: The Strategy pattern will be used to handle the difference between a test simulation and launch simulation's run methods. An interface for the simulation will be defined and the main will handle which strategy needs to be implemented at run time.

4. Observer: The observer pattern was used for the interaction between the ISS space station and the Rocket that is launched into space. The ISS will attach itself to the rocket that is launched and will update when the rocket changes state (enters a new stage of the launch). The ISS will also monitor the Dragon spacecraft's state as it leaves the Rocket and lands on the space station.

5. Decorator: The Decorator design pattern is used to model the different "payload" objects that can be launched with the Falcon 9 rocket. The types of payloads (concrete decorators) inherit from the shared base decorator (payloadItems) to define common functionality and behaviours. The concrete decorators define different types of payloads: satellites and dragon spacecraft. The dragon spacecraft is also further subdivided into "CrewDragon" which carries both cargo and crew members, and "DragonSpacecraft" which carries only cargo. This aspect of the rocket launch process makes use of the Decorator pattern's ability to attach new behaviours to objects, as needed by the different versions of components.

6. Prototype: The prototype design pattern will be used to model the creation of the Starlink satellites and the Merlin Engines. Since there can be up to 60 satellites per payload. Since all the satellites are almost the same as each other it is better to create each of them by cloning an already created satellite. This is the same for Merlin engines since there can be either 9 or 27 of them for every stage 1 rocket launch. This design pattern gives an interface to create these objects by using a clone method.

7. State: The state design pattern is used to model the different stages of the rockets. Each of the stages will be a different state. This contains the first stage where it

contains Falcon 9 cores and Merlin Engines. Stage 2 contains a Vacuum Merlin Engine and Stage 3 will be after it has detached from the Vacuum Merlin Engine. This also ensures that when it is in a certain stage it can not transition to a invalid state such as going from state 1 to state 3.

8. Factory Method: The factory defines an interface for creating rockets, engines and Dragon spacecraft. This creates a hierarchy of delegation when it comes to the instantiation of objects and ensures that creation of objects is done correctly with the correct parameters. The creator classes have all been defined as abstract with concrete creators that do specialised creation of the particular objects.

9. Command: The Command design pattern is used to model the communication function of the satellites. The Invoker (“Satellite1”) decides which communication method to call, depending on the required functionality. A satellite can communicate with other satellites via a laser message, or to the ground users via a radio message. The Concrete Command participants “radioMessage” and “laserMessage” contain the methods for their specified type of communication, while the Receiver participants handle the messages of their specific type. The command pattern’s ability to make requests stand-alone objects is exploited, in this section.

10. Memento: The memento pattern will be used to save the crucial details involved with launching the Falcon 9 or Heavy Rocket. This is done to ensure that these optimal settings can be restored later when the first stage is landed in the ocean and subsequently reused for other missions. The memento will be linked to the Stage Object and Stage will be able to store and restore its state.