Broken down:

- SpaceX:
    - Command center
    - Communicates with everything
- Spacecraft:
    - Dragon Crew:
        - Send and return humans from space station
        - Sends cargo as well
    - Dragon:
        - Sends only cargo to space station
- Engines:
    - Merline
    - Vacuum Merline
- Startlink:
    - Family of satellites
    - Cluster of satellites = 60 satellites
    - Laser communication between satellites
    - Radio communication to and from satellites
- Rockets:
    - Falcon 9
    - Falcon Heavy

Assembly:

- Rocket:
    - Falcon 9
        - 9 Merlin Engines
        - 1 Vacuum Merline Engine
    - Falcon Heavy
        - 3 Falcon 9 cores
        - 27 Merline Engines
        - 1 Vacuum Merline Engine
- Both have two stages:
    - Stage 1:
        - Merline engines used to get in low orbit
    - Stage 2:
        - Vacuum engines used to get into higher orbit
- Carry a Spacecraft:
    - Dragon Crew:
        - Send and return humans from space station
        - Sends cargo as well
    - Dragon:
        - Sends only cargo to space station
- Carry a single cluster of satellites

How it will run

- Rocket/s need to be assembled with either spacecraft with or without satellites
    - If there are satellites, then that rocket will belong to Starlink as well
- After rockets are assembled, they will enter the Static Fire test
    - This is where we will have to generate requirements for the rocket to satisfy in order to pass the test
    - If it doesn't, have to issue a fix (engineer) to fix the part, run the test again and it MUST pass. (So we can false a fault and fix)
- Start simulation:
    - During simulation, at any time, something can go wrong
    - The Rocket launches
    - Stage 1, the rocket is going from ground to low orbit
        - Using the Merline Engines ONLY
    - Stage 2, the rocket enters high orbit
        - Using the Vacuum Engines
    - If a rocket has satellites, it will have to deploy them and communicate with them.
        - Satellite deployment:
            - Cluster detaches
            - Satellites spread out
            - Satellites are either in position or out position
    - Deploy the spacecraft:
        - Successful deployment
    - Spacecraft scenario:
        - Deploys from the rocket
        - Successfully travels to the space station
        - Successfully locks onto space station
        - Unloads cargo and crew
        - If Crew Dragon:
            - Come back to rocket
        - Else
            - Stay at space station
    - Craft reenters the Earth atmosphere
    - Lands in ocean
        - Successfully
        - Unsuccessfully
    - Simulation a success!

Can store successful sims with memento, so launches can use these

Static Fire Requirements

Must satisfy all to pass test

- Has fuel for rockets to start
- Must be either under or at max capacity
- If satellites
    - Make sure there are 60
- Communication works
    - Command center sends message to rocket
    - Rocket Forwards message to Spacecraft
    - If all messages are sent and signals are sent back then communication work
- Does it turn on and off?

Determine interrupts during simulation

- We have categories that can cause interrupts:
    - Sudden loss of Fuel
    - Mechanical
        - Engine failure
        - Components falling apart
    - Unsuccessful deployment of spacecraft/satellites
    - Spacecraft delivery unsuccessful
- Then choose a random number:
    - If random number falls below 10
    - Randomly insert it into one of the categories
    - That category will print the appropriate response causing the simulation unsuccessful
- Example Scenario:
    - Generate random number 0 - 100
    - If above 10:
        - No interrupt
    - Else
        - Generate category number 0 - 3
        - Signal the correct category failure to end simulation
- We randomly put in a interrupt opportunity while running the simulation

Design Patterns to use

- Template:
    - Can be used as the Static Fire test
    - No rocket can be launched without taking the test
    - Have checks:
        - Rocket has fuel
        - Spacecraft
        - …
    - Return a result, Pass or Fail
- Memento Pattern:
    - Used for simulation reset and continue
    - If simulation fails, reset it and continue the simulation until success
    - Or restart prompting changes? Cuz it will probs fail again if its the same
    - Store states of sims that have succeeded, so that actual launches use these states
- Composite Pattern:
    - Creating the cluster of satellites
- Iterator:
    - Iterate through the satellites
    - *Can count that all 60 are present*
- AbstractFactory:
    - Can be used to in the creation of satellites or rockets
- Decorator:(Done in composite pattern)
    - Add satellites (optional)
    - Add spacecraft (which spacecraft)
- Adapter:
    - Can be used for the rocket to deploy the spacecraft
- Command:
    - Used to ignite
    - Accelerate
    - Decelerate the rocket
    - As well as the spacecraft
- State:
    - Determine the stages of the rockets
    - Stage 1 or 2
- Observer:
    - Oversee the cluster of satellites being deployed
- Mediator:
    - Can act as a communication guru from SpaceX overseeing the entirety of it all
    - SpaceX tells rocket what to do
    - Rocket tells spacecraft what to do
- Chain of responsibility:
    - Can be used over mediator

    - If spaceX says something, the chain can determine if the message is intended for the rocket or the spacecraft or the satellites
-

TASK 1
1.1)
Main- able to simulate rocket launches, static fire test, store in batches, for real launch
Each rocket has own engines/requirements/components, pass all respective stages