SO4 Event Based Scriptable Object Architecture
In this project, you have a lot of different systems that is going to be depended on how the player's health is changing. Since you want to structure the architecture of the project in a nice way from the beginning you decide to make use of an Event-Based Architecture that uses Scriptable Objects.

Exercise
- Create a Scriptable Object named HealthData
- The Scriptable Object should contain two fields
    - CurrentHealth.
    - StartHealth.
    - MaxHealth
- The Scriptable Object should contain different types of logic for changing the health:
    - Add Health to the current amount.
    - Take damage and remove health from the current amount.
    - Reset currentHealth to the value of StartHealth
    - Check if Health gets below Zero.
- The Scriptable Object should contain an event named CurrentHealthChange, which should be invoked every time a change happens.
- Create a Class called UIHealthBar and attached this to a Prefab that can show UI Images.
- This UI class should listen to the CurrentHealthChange event and update the graphics when the event is triggered.
- Create a class for the Player and Enemy, each should have a reference to its own HealthData instances. (ie. one HealthData for the Player and another for the Enemy)
- The class for the Player and Enemy should contain logic such that they can attack each other, but should do so only through the two Scriptable Objects.
- Create a Custom Inspector for the HealthData class
- When clicking a "Trigger Health Change" Button, the CurrentHealthChange event should be triggered.

Optional Exercises:
1. The game designer wants to have a very cool game over screen when the player takes damage and get below zero in health. How would you implement this?
2. The graphics artists want to have different graphics for the UI health bar depending on if it's the Player or the Enemy. Could you use the Scriptable Object for controlling this?
3. Instead of having a "Player" and "Enemy" class, could you implement a solution that only uses one class for both entities?