The code inherited was reasonably well organised and modularly displayed. Although we did not want to change the structure of the code, we felt it necessary to do so in order to implement the new requirements **F13** and **F14**. Final requirements can be accessed here. All of our changes were made with user experience in mind, therefore will contain the requirement **C3.1** as justification of implementation. This is because, as every change improved the game and made it more playable, that in turn provided a more satisfying experience for the player. Therefore, to avoid repetition, these requirements are not justified in the 'Reasoning' column.

Architecture Changes

ID	Change	Reasoning
Ch1	Added class Typhoon inheriting PhysicsActor, containing a new Texture typhoonTexture and an overriden draw method	Including the concept of a typhoon was a feature required by the client to be implemented with respects to Assessment 4. It inherits from physics actor to allow it to be repositioned so it spawns in a random place as required, and contains a Texture so it's area of effect is visually indicated to the player.
Ch2	Added Typhoon typhoon to SailingScreen	Added utilising the new typhoon class, this object actually implements the new feature within the game. Because the typhoon only occurs within the sailing portion of the game, it is only contained within SailingScreen. Typhoon functionality is contained within SailingScreen due to its relation to player's position on the map and its need to be evaluated as the player moves.
Ch3	Added int upgradeCounter to Department	With crewmembers also affecting ship stats, we needed to add a new integer variable to department to keep track of how many upgrades have been bought and affect upgrade cost to avoid crewmembers increasing the cost of upgrades within departments.

Code changes

Change	**Reasoning**	**Relevant Requirement**
Implemented Typhoon class as a PhysicsActor	New requirement per **Ch1**	F13 C3.1
Added typhoon object to SailingScreen	New requirement per **Ch2**	F13 C3.1
In SailingScreen, updated constructor to include initialization of Typhoon typhoon and binding Typhoon typhoon to a MapObject	Initializes the new typhoon feature within the SailingScreen. Binding it to a MapObject is where the hitbox is defined utilizing the pre-existing tilemap.	F13 C3.1
In SailingScreen, updated constructor to move	After typhoon is bound to its initial set position and hitbox from the tile map, it is then moved to a random position on the map.	F13 C3.1

Typhoon to a random coordinate after binding	This means at game start its position is random, per the requirement. The coordinates are chosen in such a way to avoid clipping outside the map.	
In SailingScreen, updated function update() to include typhoon functional logic	The typhoon is designated as an obstacle that the player should avoid, and as such when the player is within the typhoon it has to do something. Logic to determine player position is within update(), so logic to detect whether the player is within a typhoon and what to do about it is held within update() as well. Effects of typhoon, damages playerShip until they're ¼ of their max hp, slows down the ship's max speed until they're out of typhoon. This was built in the same style as the preexisting code to be as changeable as possible.	F13 C3.1
In SailingScreen, added new variables inTyphoon and typhoonTimer	Utilized in typhoonLogic. inTyphoon is a boolean value signaling that the player is in the typhoon and typhoonTimer is used to determine the time spent in the typhoon for damage calc.	F13 C3.1
In CombatScreen, implementation of crew member system, by checking if a boss is defeated at the end of a fight	Within CombatScreen, there are if statements specifying that depending on which boss is defeated Attack or Defence is increased permanently along with a message stating that they have received a crewmember.	F13 C3.1
In ShipType, CollegeBosses defense were raised across the board to make college warship fights more difficult	Through playtesting, we found the bosses to easy to beat and so we adjusted their stats to force the player to fight non-boss ships to progress through the game.	F8.2 C3.1
In Department, added upgradeCounter variable.	This was added per **Ch3** to allow for crewmembers to not affect the cost of upgrades.	F13 C3.1

GUI Changes

Change	**Reasoning**	**Relevant Requirement**
Implementing HP	As the player was now able to take damage in the sailing mode due to the addition of Typhoons, we had to also implement a health UI. This was done by adding a new variable, healthLabel, to SailingScreen. This added the UI element, structured as (current health / max health), which is continuously updated in the update function. This was done so that the player was informed of their ship's current status and to improve the user experience of the game for the player	**C3.1**