Sea Of Geese
Implementation Report

Code Modifications

Crew Members: For the new requirement A1.4, we created a new class CrewSelectionScreen. As a card selection screen had already been implemented, we based our new class off their old design, which made it much easier to implement and test. Our idea for crew members would be three different effects that would give the user bonuses for each member that they had. Our three final crew members are: Master Gunner (which reduces the cost of drawing cards), Quartermaster (which increases the amount of gold you get from enemies) and Carpenter (which heals you after each fight).
To be able to store the number of each crew type the player had, we added an array into the Player class called crew. When the user selected which type of crew member they wanted to have, the corresponding entry in the array is incremented, which also enables the bonuses not be unique (having four carpenters will heal you for four times the amount of just one). This was chosen as it not only allows the user to gain more of the bonuses that they prefer having, but it also reduces the amount of design work we would have to do (we only have to create 3 crew members instead of having enough so that each is unique)
Justification: Having a seperate screen for crew selection enables us to seperate it from the rest of the code, making it easy to modify and read. We added the crew array to Player so that we could keep it specific for them and easily accessible where required. Each crew member gives a bonus so that we can meet requirement 1.6, and we felt like having these bonuses would make it so it is clear to the user what its use is. Requirement A1.5 specifies that the crew member should be added to the ship after achieving an objective, and we felt like defeating each boss and moving to the next level is a very obvious objective to the user and would be the best way to get the crew members.

Obstacles: For the new requirement A1.7, we implemented natural obstacles into the node map that was previously generated. We created a new class called Obstacle which defines the obstacle. Another new class, ObstacleManager imports the Obstacle json file which defines each different type of Obstacle, including the damage that occurs and its name and other attributes. To allow some variation in obstacles, we defined three obstacles: Scurvy (which has a chance to kill a crew member if you have one), Typhoon (which has a chance to make you lose 10 health) and Whirlpool (which has a chance to make you lose 5 health). This fulfills requirement A1.9 [1]. The NodeUtil class which was previously created and is where the map that the user navigates is created. Within NodeUtil is a variable called obstacleNodeChance which defines the chance for each node to be changed into an obstacle node. We use another class ObstacleNode to allow screen switching to the obstacle screen.
Justification: The Obstacle system was set up so that obstacles can be both avoided and endured. Avoided is easy, because it's on a NodeMap and it appears randomly because of ObstacleNode, meeting requirement A1.8 and A1.9. Should you go down the path with one, you must endure them. We have a separate ObstacleScreen for them which handles the logical calculations, meeting the enduring requirement. We felt it was necessary to keep this all handled within the screen, rather than having a completely separate class in this instance to handle it so that further development is performed easily. Obstacle just handles the encountered Node's JSON attributes, thus all ObstacleScreen needs to do is compare the attributes using simple getters for operations. ObstacleManager just handles loading in the JSON files into Obstacle objects and making sure each Obstacle has its own effect.

Minor Modifications:

As will be explained in more detail in the GUI section, we modified the HUD code to include messages to the user for their health being modified from crew, obstacles and encounters as well as showing to the user the number of each crew member that they had. To do this, we extended the messageHUD and statsHUD classes. We also implemented a points system as per requirement A2.7 and B1.7 shown in the final requirements here: [1]. This point system allows points to be gained in various ways and is no longer treated as gold being the same as points.

GUI Changes

For our key GUI changes, we had to implement a different node to the map which is used as an obstacle node. This has the design of a question mark to show that it is random, and is similar looking to the encounter node design so that the player could get surprised by the obstacle. When the player encounters an obstacle node, it switches to a new screen which is based off the encounter screen. Similarly, we had to create a new screen for the crew selection which was likewise based off the previously created card selection screen. During testing, we found that it was difficult for the player to know how many of each crew they hard as well as whether they had gained or lost health, due to the crew or obstacles/encounters respectively.

To alleviate this problem, we added an additional row to the main map HUD which previously showed the player their points, gold and health. This new row shows the names of the three crew members followed by the number of each that the player has. We also added new messages for the player's health changing due to encounters, obstacles and nodes.
We felt that this makes it much this information much more easily available to the player, which should make the game more enjoyable to play.

Extra Features

During development for Assessment 4, we went through the requirement specification multiple times and decided that the previous assessment's work had met all of their requirements. Due to this, and the fact that we all felt that the game was completed and thoroughly enjoyable to play, we didn't include any extra features as we felt that would take away from the core experience that had previously been developed.

References:

[1] Requirement Specification
https://davidjnorman.github.io/SEPR/assessment_4/Req4.pdf