3.3.b | Implementation

Implementation of Architecture and Requirements

Architecture

The changes made by *Un-Two* to *Evil Geese*'s project were as minimal as possible. Changes to scripts occurred in only three areas: Combat, Shop and Dialogue Manager. The only wholly new scripts were for the minigame, as shown in the Architecture Diagram [1]. The MiniGame has been implemented as a separate entity with no callback to other scripts so the new material here has no conflict with old material.

The changes made to the existing code scripts are minimal and can be seen below in *Significant New Features*, they make no changes to the architecture, while only changing some of the functionality within their own scripts. The reason for such little change is that the work done previously by EG set out a lot of inbuilt Unity Methods meaning that the new sections could mostly be created through these methods. This meant that all communication between scripts was handled by the systems already in place written by EG and so a large portion of our work simply extended their implementation.

Requirements

Updated requirements: https://ojp508.github.io/Mambo-Studios/Req3.pdf

Here is the list of requirements uncompleted by EG that are now complete:

- **F2**
- **F4**
- **F9**
- **F10**
- **F13**

These were implemented to complete the set of requirements set by EG and the client. Most of these were fulfilled using the set Unity Methods given by EG, these used their features such as dialogue Manager, CombatManager and the grid layout to create scenes that were for all purposes identical to original scenes created by EG but with different artwork and text and enemies. This meant that not much new code was needed which is good as it decreases the risks of game breaking bugs or overlapping code that could cause error within the project.

Significant New Features

Exp and level tracking

Initially upon starting assessment 3 the requirements to add exp and level for characters had not been implemented. This fulfills requirement F6 for the character to progress and improve. Two new methods were added to the *combatCharacter* class, one which returned the exp and one which returns the level. Each characters exp is stored in a text file which is initially set to zero. After each combat the character used gains exp and therefore may progress in level. The higher level a character is the more damage a character will do, this means that we had to set a damage modifier which interacts with level for elevated damage.

Mini Game

The Mini game was created as a wholly separate feature, activating the game takes you away from the game completely to a new menu, this menu is cloned from the original game menu with only the path changed from each button, this ensures a degree of familiarity with the system. The game itself has very simple controls which are well explained in the menu. Returning takes you back to exactly where you were with no changes to the game state.

There was one small change to existing code to implement this, a new option was added to the dialogue manager for *GoToMiniGame*, this meant a new enum was created, adding a new case that takes you to the *MiniGameMenu*. This was created in the exact same way as the existing three case, implementing the minigame this way kept the game the same in the unity build as like the existing content actions are reached through dialogue.

This implements the Requirement **F13**, this as stated makes no changes to the existing game state, it is first experienced as part of the story, from then onwards it is reached through revisiting chemistry where it is located instead of through the minigame as stated as it was believed to be just as easy to reach through this scene.

New locations

As stated in requirement **F2** ten world locations are available. Five of these were implemented during assessment 3 and other areas were improved. These locations are Central hall, market square, chemistry, biology and locations which link these together. Each location had its art finalized and was then added to the world map.

Shop

The shop is accessible from the town area, this satisfies requirement **F9**, which states that the player must be able to spent assets. The shop is controlled using a series of switch statements to get data about the items which are sold at the shop. The player can buy several items ranging from plastic forks to key cards.

Features not yet implemented

Here we will list the requirements set out by EG that were not implemented by the team and the reasons for the failure to complete them.

- **F11 -** It was decided that a full backstory would take up time that was not needed by the player. The game is created to showcase the University so this was an unnecessary feature that would make meeting the requirement of a quick-play style harder to meet.
- **F12 -** The Demo-mode would be no different from just playing the game normally and ending at the third scene, furthermore with the way EG designed the game, very little happens in the first 3 scenes so a demo mode of these would not fully showcase the game to the best of its ability.
- **UH2 -** Our game uses a colour impaired friendly style, so a new system should not be needed as the original settings should be viewable by all vision types.

References

[1] Architecture Diagram: https://evilgeese.weebly.com/assessment-1.html