Implementation Report

The software we inherited was produced to a good standard however in some sections there were sections of redundant and duplicate code. Also, the client requested a change in requirements leading to three functional requirements[1] being added and this meant that the implementation had to be modified to accommodate. Outlined below is a brief overview of the key changes and additions we made to the code and to the GUI. The effects of these changes to the concrete architecture of the software can be found in the Architecture Report[2].

Changes: Second unit type implementation
Files Affected: Sector.java, Map.java and DialogFactory.java
Explanation:
F19 required a new and functionally different type of troop to be implemented. The Sector class was changed such that unitsInSector was replaced by underGradsInSector and postGradsInSector so that both unit types could be represented on sectors and a new troop count icon, (a graduation cap), was added to each sector to represent the new kind of troop. Postgraduate troops grant a defence buff to a sector but cannot be moved once allocated and therefore the conflict resolution behaviour, defined in the Map class, had to be modified to support this. The dialog box for allocating troops was modified to allow the allocation of both troop types.

Changes: Punishment card implementation
Files Affected: Sector.java, Map.java, Player.java, MinigameScreen.java, WidgetFactory.java and DialogFactory.java
Explanation:
F20 and F21 required the player to be able to acquire at least three types of Punishment Cards, which players can obtain and use as a debuff towards enemy sectors. Therefore the minigame was reworded so that the player matched Punishment Cards instead of numbers and they were rewarded with the cards they matched not bonus allocation points. This required the GUI of the MiniGameScreen class to be updated and variables added to the Player class for storing which cards players held. The cards were drawn with NF2 in mind and a set of icons was designed to represent an active card effect on a sector. A punishment card button was added to the top right corner of the HUD which opens a dialog which allows the player to view the cards they hold and select which one to use.

Changes: UiScreen refactoring
Files Affected: UiScreen.java, MenuScreen.java, OptionsScreen.java, GameSetupScreen.java, GameScreen.java and MinigameScreen.java
Explanation:
In the code base we received each of the five screens within the game had there own setup code. This lead to the same pieces of code being written five times in some cases. Therefore a UiScreen class was created that all of the in game screens could inherit from and this class handled the game setup. This change should help with maintaining the program as any changes required to be made to the screen setup now only need to be performed once as opposed to five times over.

Changes: Removed PVC class
Files Affected: PVC.java and GameScreen.java
Explanation:

The mechanism for triggering the minigame was modified such that there was now a fixed probability of it happening each time the player makes a successful attack. This meant that it was no longer necessary to store data about the PVC across the duration of a game and therefore the PVC class was redundant and removed. Instead a method, PVCSpawn(), was added to GameScreen for triggering the minigame.

Changes: Save and Load system updated
Files Affected: GameScreen.java, DialogFactory.java, SaveLoadManager.java, GameState.java and JSONifier.java
Explanation:
The save and load system has been updated so that new data stored, e.g. the Punishment Cards a player owns, is able to be saved and loaded.
The partial implementation of a multiple saves system was removed as it was not required.
A menu button was added to the game HUD which opens a dialog which displays a pause menu, allowing players to save and exit from the game.

Changes: Audio system updated
Files Affected: AudioManager.java, AudioPlayer.java, GameScreen.java, GameSetupScreen.java, Map.java, Phase.java and PhaseReinforcement.java
Explanation:
In the code base we received loading of sound files and playing them was the responsibility of the AudioManager class. Selecting what audio file to play, possibly from a pool of sounds if there were multiple possible choices, was done within the main game logic. This lead to duplicate code for selecting what file to play and also made the game logic harder to follow as it contained large sections of audio code. Therefore the AudioPlayer class was added so that selecting what audio to play could be moved to here helping improve the readability of the main game code and also making it easier to maintain in the case of any audio being added or removed.

Changes: Attack and Movement Phase duplicate behaviour moved to PhaseAttackMove
Files Affected: PhaseAttackMove.Java, PhaseAttack.java and PhaseMovement.java
Explanation:
Both the PhaseAttack and PhaseMovement classes contained logic for creating an arrow between two sectors in the software we inherited. Therefore to reduce duplication this behaviour was moved from these children classes to their parent, PhaseAttackMove.

Changes: GUI Overhaul
Explanation:
The layout and functionality of the GUI was effective and suitable for the game, but the team was not satisfied with its graphics and artistic style. Considering requirements C2 and NF3 the aesthetic was modified with the aim to be more eye catching and engaging to players. The existing artistic style was flat and not attractive, so the GUI was redesigned with the aim of giving more visual depth to the game as well as introducing more colors. All UI components were restyled to be dark and surrounded by bright neon colors and transparency was added to give an appearance of depth.
The game map and dialog box system were also updated accordingly to fit in the new theme. Inspiration for such artistic style was taken from futuristic GUIs present in films such as Terminator, Iron Man and TRON. Screenshots of the updated GUI can be seen in the game manual on the team's website[3].

References

[1] SEPR “Updated Assessment 4 Requirements” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/Assessment4UpdatedRequirements.pdf [Accessed: May. 1 2018]

[2] SEPR “Architecture” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/ConcreteArchitectureUMLDiagram.png [Accessed: May 1 2018]

[3] SEPR “Game Manual” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/Ass4GameManual.pdf [Accessed: May. 1 2018].