Implementation Report

Throughout this document we have used our new requirements [1] and testing [2] to justify various parts of implementation.

New Features

Feature	**Justification**	**Relation to Requirements and Architecture**	**Comments**
NonZombies added to the game. Some spawn every wave. They become zombies when they die.	New requirements for assessment 4 said a cure item needed to be added to the game which turns zombies into nonZombies. There were no nonZombies apart from the player in the game already.	Our new requirement F12 says there must be nonZombies in the game and when their health is 0 they must become zombies. We added this without changing the architecture. Instead we added new types to the Type enum in zombie. This allows quick changing to or from a zombie. We had to add new information to the config file to spawn the nonZombies.	Improved zombie avoiding AI may help. Majority of changes were in the zombie class.
Cure added to the game. Cures zombies in a certain area when the player picks it up.	New requirements for assessment 4 said a cure item needed to be added to the game.	A new class was added powerUpCure. This has been implemented in the same way as other power ups to fit the architecture. F12.2 says there must be a cure item to cure zombies in a certain area.	Size of cure area may need tweaking. Most changes in PowerUPCure.
Scoring added to the game. Score is represented by the number of survivors left. Survivors is displayed on the menu screen.	Gives players a reason to save the nonZombies. We required this in the new requirements we made.	F12.3 in our requirements said that we must add a scoring system based on the nonZombies. The architecture hasn't significantly changed to add this.	Could add information to the end of level screen saying how many people were saved. This required changes to Level and SelectLevelScreen.
Player becomes a zombie when their health reaches zero if they not a zombie. If player picks up a cure as the zombie they stop being a zombie.	Fits with nonZombies becoming zombies and was part of the new assessment 4 requirements.	F11 of the requirements we added states that the player must become a zombie when they die. This hasn't significantly changed the architecture. Zombie types for the player are added to the PlayerType enum.	Could provide player with more information when they become a zombie. Changes within Player class.
Music added to	Makes the game	This hasn't changed the	Could stop less

each level.	more tense and keeps users interest.	architecture. This wasn't required.	abruptly

Changes

Feature	Justification	Relation to Requirements and Architecture	Comments
Minigame tweaks. Prevented constantly clicking from stopping reloading. Allowed player to miss zombies. Scaled difficulty.	Constant reloading could prevent player progression and make the minigame harder to understand. Without scaling the difficulty the minigame would be endless.	F2 requires a progressive change in difficulty, difficulty scaling incorporates this in to the minigame. The other features make the minigame flow better and make it easier to learn fitting N1. This hasn't changed the architecture.	MiniGame and MiniZombie classes changed.
Zombie movement and attack tweaks. Speed wasn't changing and boss1 wouldn't do any damage.	No boss damage made the game to easy and no change in zombie speeds meant some of the zombie types didn't actually change.	F4.1 requires different zombies to have different stats. Being able to change speed allows this. F9 requires the zombies to be able to damage to the player changing boss1 allowed this.	May affect game difficulty. Changes in Zombie and Constants classes.
Power up disposal fixed	Powerup used to be disposed before it was removed from the map. This showed a black box on the screen briefly.	This improves feedback to the player fitting requirement E2. This hasn't changed the architecture.	Changes in Level class.
Zombie attack feedback and rebalancing.	Player could attack through walls before. To stop this we reduced player range, but this made the game too difficult.	This improves feedback to the player fitting requirement E2. This hasn't changed the architecture. Also, F2 requires the game to get progressively more difficult.	May still need some tweaking. Changes is Zombie and Constants classes.
Corrected resizing of screens	Reducing the screen size would cause buttons and text to overlap and become impossible for the user to interact with correctly.	This hasn't changed the architecture and was not explicitly required.	Changes in Menu, SelectLevel and Story screens.

Bibliography

[1] Geese Lightning, ‘Updated Requirements’ [Online], May 2019, Available: https://lloydbanner.github.io/SEPR-Team-7/Req4.pdf
[2] Geese Lightning, ‘Evaluation and Testing Report’ [Online], May 2019, Available: https://lloydbanner.github.io/SEPR-Team-7/ET4.pdf