IMPLEMENTATION REPORT

NP Studios – Team 8

Team Members

Lucy Ivatt, Jordan Spooner, Alasdair Pilmore-Bedford, Matthew Gilmore,
Bruno Davies, Cassandra Lillystone

How Our Code Meets Our Architecture

DicyCat's code was faithful to their concrete architecture; we didn't need to add any code to further implement their existing architecture. However, they did create a **UFO** class which is not defined within their UML diagram. We changed the name of this class to **Alien** to follow object-oriented standards more accurately as you should avoid acronyms as class names. We then added the methods into this class which were required for patrol implementation; it was almost empty due to patrols not being required until now.

Due to other changes/additions that we made, we added to the architecture slightly. For instance, we added the new classes **Minigame**, **Pipe**, **ControlsWindow** and **Button** (https://npstudios.github.io/files/Additions_UML_Diagram.png). We have also added a few additional methods to pre-existing classes. These slight changes to the architecture are shown using our clear commenting standard, as mentioned in the change report.

How our Code Meets Our Requirements

We first began assessment 3 by playtesting and extensively reviewing the previous team's requirements. We then both added and changed some of these (https://npstudios.github.io/files/NewandChangedRequirements.pdf) where we believe the previous team failed to fully define the brief. For any major changes to the existing requirements we inherited from DicyCat, we checked with the customer that he approved of our ideas (**Figure 1**) prior to adding them to the change tracker as we want to ensure that the final product satisfies the customer and their view for the game. Once we believed the requirement list was fully complete we then added the changes required to implement them to our change tracker (https://npstudios.github.io/files/ChangeTracker.pdf) as mentioned in the change report.

Requirement Changes and Additions Justification

UR_INTUITIVE: Originally, known as **UR_TARGET_AUDIENCE**. The original name wasn't suitable in comparison with the description of the requirement. We then extended it so that it requires the game to be easily understood and unambiguous, making the game more inviting for players of all abilities.

UR_PATROLS: New requirement. It's a big feature of the game and was not included in the previous group's requirements. It was specified in the brief that patrols were required for the game after assessment two, which is why it is rated as shall.

SFR_IMPLEMENT_PATROLS: New requirement. This is to complement **UR_PATROLS** and add the functionality for patrols to be implemented.

UR_FORTRESS_ATTACK: New requirement. The user shall be attacked by the fortresses when they are in its range. There was nothing of this nature in the existing requirements. Fortresses should have defensive weapons (as said in brief), so essential for fortresses to attack fire trucks when they are in range.

SFR_FORTRESS_AIM: New requirement to complement **UR_FORTRESS_ATTACK.** Fortress attacks should aim towards their target when firing projectiles.

SFR_FIRETRUCK_STATS: Changed requirement. Originally required the user to choose the type of fire truck at the beginning of the game. We believed that this didn't meet the brief fully, as our interpretation was that all four fire trucks had to be available to choose from and play with during the game, hence changing the requirement **(Figure 1).**

SFR_MINIGAME: After consulting with the client (**Figure 1**), we decided to change the idea of the minigame. We believe our minigame idea had a stronger relationship with the games overall theme and could easily be embedded as it relates to the refilling of the trucks water which occurs during the specific event where the truck returns to the fire station with currentWater less than maxWater.

SFR_REFILL_FIRETRUCK: Originally called **SFR_REFILL_OVER_TIME**, we changed it so that the fire trucks refill instantly as time is already spent playing the minigame to achieve the refill.

SFR_CANCEL_REFILL: We removed this because it was no longer relevant due to the above requirement edit **SFR_REFILL_FIRETRUCK** and we want the user to fully complete the minigame before returning to the game.

UR_FIRETRUCK_MIN_START: Changed requirement so that it meets the brief more precisely.

Implementing Requirements:

Button causes game to crash - FIX

During playtesting we noticed that one of the buttons which appears in the GameOverScreen would cause the game to crash so we fixed this bug.

Memory Leak - FIX

We found that the debug methods, which showed the hitboxes of various objects, caused a memory leak to occur, resulting in the game eventually crashing. As these methods were not fundamental to the final product or needed by us, we commented out the usage of these methods. This fix is important as the game is unplayable if it consistently crashes.

Fortress Stat Variations and Improving over time - TWO CHANGES

To meet the requirement **UR_ET_UNIQUE_SPEC** and the brief, we have given the fortresses different health and damage levels by adding health and damage attributes to the fortress class and modifying the fortress constructor and its calls. To fulfil the requirement **UR_ET_IMPROVEMENT** and the brief, we added a statement which checks the timer and if it is halfway through the game the damage level for each fortress is doubled before the damage is applied. No major new features were implemented as part of this change.

Edge of Map Warping Fix and City Wall Addition - FIX AND ADDITION

While testing the boundaries of the map, we found a bug where the edge of the map warps as the previous team didn't tell OpenGL to clear the previous game frame, so we fixed this using one line of code within the Kroy class. This map warping could make it feel like the truck was moving forever on grass.

Similar to this we decided that the previous teams map border wasn't clearly defined, so we added walls (similar to York castles walls, to keep in consistent with the rubric that the map should have some association to the city of York). These changes relate to the requirement **UR_INTUITIVE**. No new features were implemented, we just updated the tile map.

Removed Second Timer and Change HUD to count down - TWO CHANGES

The original implementation used two attributes to keep track of game time. We removed one of these, increasing both efficiency and consistency through the code. We also modified the HUD timer to count down instead of up. This relates to **UR_INTUITIVE,** making it clear to the user how long they have until the fire station is destroyed. No new major code features were required for these changes.

Stopped Fortress Firing Straight Down - CHANGE

We found that when we first played the game that one of the patterns for the fortress would fire straight down no matter where the player was. We didn't think this was useful and instead made it so that this pattern would fire bullets towards the player instead, requiring the player to dodge and adding more engaging elements to the game. This helps to fulfil **UR_FORTRESS_ATTACK**.

High Score Calculation Change and Saving High Score - CHANGE AND ADDITION

To fulfil the requirement **SNFR_HIGHSCORE** and **UR_HIGHSCORE** we have made a slight change to the code as well as an addition. Originally you would always receive the same score every time you won the game. We thought this could be improved so we added bonus points depending on how much time was remaining before the fire station would be destroyed. We believed this would increase the replay value of the game as you could play again and try to beat your previous time. Additionally, the high score now gets saved to the computer, therefore not being lost if the game is closed. We added this as we thought it was important as users would not like to lose their progress.

Game Size Change and Mini Map Addition - CHANGE AND ADDITION

As part of **UR_INTUITIVE** we have changed the game screen size, and as a result of that the background picture for the main menu. We found when playtesting the game it was difficult to play with such a small game window, especially when we added patrols as you couldn't see them coming to avoid them in time. Because of this we increased the size of the game window and therefore had to change the menu image as it was too small. This involved changing two attribute values in the **Kroy** class and the file for the background texture so no new features. We also implemented a mini map to meet the same requirement. Navigating the game, locating all fortresses and keeping track of fire engines was proving to be difficult and time wasting but adding the mini map fixed these issues. We implemented this by adding another rendering method in **GameScreen**.

Fire Station Stat Changes - CHANGE

We made sure to spawn the fire station in the centre of its green square and set its radius to half the greens squares width. This is because before its hitbox incorrectly overlapped with the green refill/repair square and our change has made this more accurate. As it used to be confusing as to where the truck needed to go to replenish their health/water, which relates to **UR_INTUITIVE**. We also changed the texture of the destroyed fire station, as it was a placeholder before, and we wanted to change it to stay consistent with the aesthetic of the game and be more obvious to the user what had occurred.

Switch Between All Trucks - CHANGE

We believed that the previous group's way of selecting fire trucks didn't fit the brief as you could only use one type of fire truck in each playthrough of the game. We instead had the idea to add all trucks to the game at the beginning and the player can control any of them, assuming they are not destroyed, and switch between them at will. We proposed this to the client and they agreed with our ideas (**Figure 1**) To implement this we added selected (Boolean) variable to the **FireTruck** class (needed so only the selected truck is updated), as well adding an ArrayList of all Fire Trucks to the **GameScreen** and an int variable containing the index of the active truck. Finally, we added input handling code, so the active truck is changed if the user presses 1, 2, 3 or 4. These changes help to meet our altered **UR_FIRETRUCK_MIN_START** requirement. Additionally, due to this change we found the 'trucks remaining' label in the HUD unnecessary as the trucks are already on screen so we changed this to 'fortresses remaining' to help achieve the **UR_INTUITIVE** requirement as it was more obvious to the user how close to their goal they were.

Implemented minigame - ADDITION

We developed a minigame to meet the requirement **UR_MINIGAME**. The minigame is embedded into the main game by being called when a fire truck refills; this ties the minigame thematically into the main game, as the minigame has the user solving a pipe puzzle to transport water, restoring the truck's water. The variables the minigame needs are stored in a **Minigame** class, and an additional **Pipe** class was written to keep in line with OOP principles. Another state was added to the **GameScreen**'s states enum to allow a switch case to simply draw the minigame's stage and switch inputs to it instead of running the main game. The minigame has 3 different puzzles randomly chosen to add variety and avoid monotony.

Implemented patrols - ADDITION

The addition of patrols was needed to fulfil the **UR_PATROLS** requirement and specified in the brief from the client. To complete this we added new methods inside Alien class to move the alien towards its destination, check if it has reached its destination and then set the next waypoint, as well as handling attacks if the active truck is in range.

Controls Screen - ADDITION

We added a button to the main menu screen which takes the user to a controls/instructions screen. This screen outlines all the keyboard hotkeys required to play the game. This addition removes any ambiguity, as **UR_INTUITIVE** requires, as well as providing the instructions asked for by **UR_INSTRUCTIONS** and **SNFR_INSTRUCTIONS**. To add the control screen, it required its own class **ControlsWindow** which sets the layout of the screen and controls the back button.

Final Fortresses - ADDITION

As part of the requirement **UR_ET_MIN_START** and explicitly defined in the client brief we added the final three fortresses. To do this we added 6 new textures, (including both 'alive' and 'dead' versions), to the **GameTextures** class. These new fortresses are the train station, the hospital and central hall. We then instantiated the fortresses in the **GameScreen** class by adding the constructors.

Unfulfilled Requirements

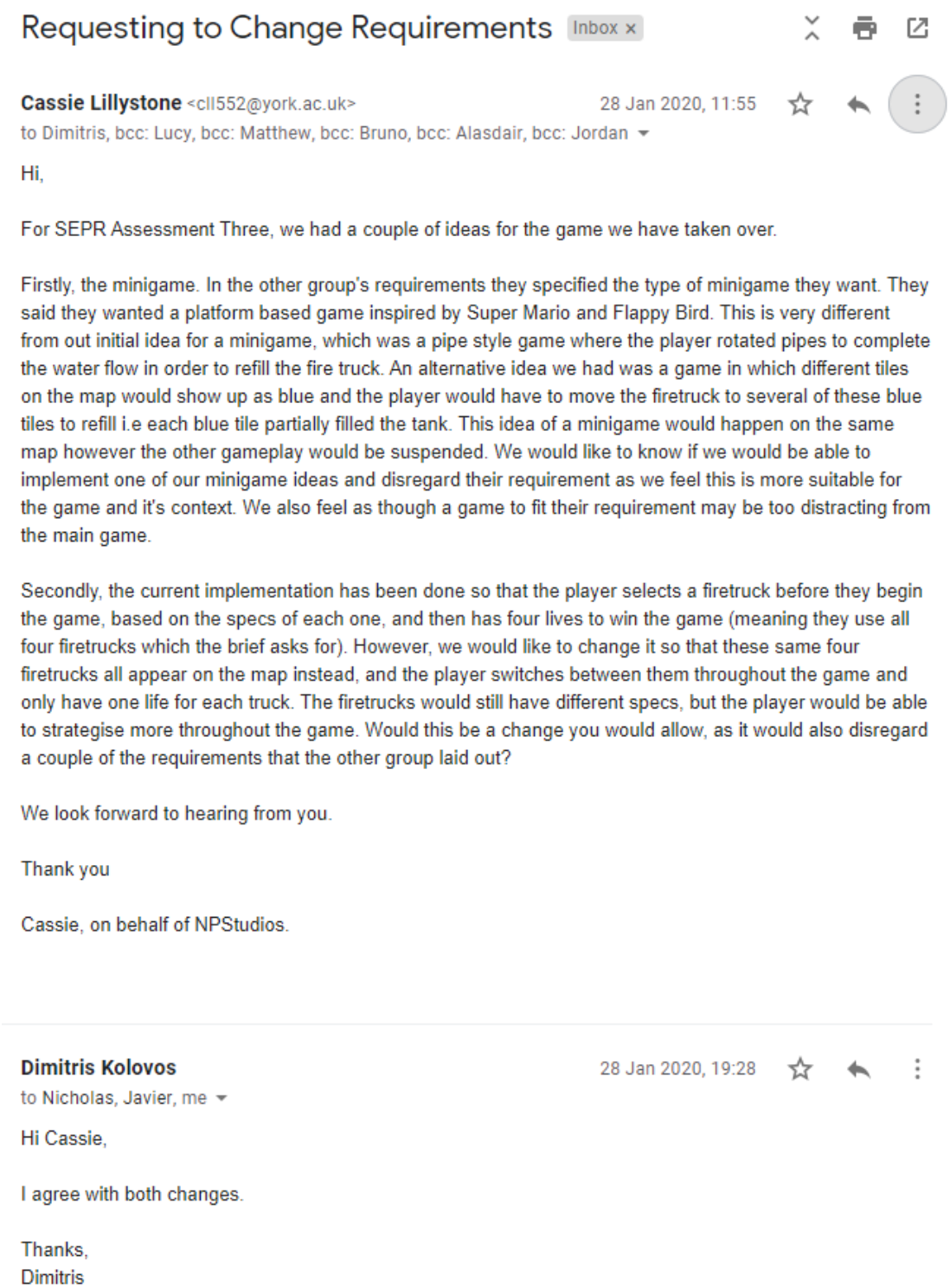
One of the requirements that we didn't implement for this assessment was having different colour schemes (**UR_COLOUR_ACCESSIBILITY**) to enable accessibility. We weren't confident in what accessibility options are standard in modern software, meaning it would require a lot of research to implement this feature accurately. We believe this is not an essential requirement for the implementation of the game but is definitely something which could be included in future development when more time is available.

Secondly, we didn't implement our idea of having to press a key to attack (show in change tracker). Currently, the fire truck attacks anything in range automatically, however we feel that having the user initiate attacks would add more strategy to the game and make it more engaging and fun to play. Users could prevent the trucks water from being wasted on patrols if the user doesn't want to fire back and potentially finish the game quicker than before. This is a possible implementation, again, for future development as we wanted to focus on more important changes in the small time period we had.

Finally, we didn't manage to fulfil the **UR_DIFFICULTY_LEVEL** requirement which is described as 'may', so we as a group decided that it was low priority. This has been thought about due to the fact that the harder fortresses to destroy are placed further away from the fire station on the map, so the user only encounters the easiest ones to begin with and has to avoid less patrols in the process. However, with extra time we would have preferred to have a difficulty mode which would scale the statistics of the aliens and fortresses to make the game easier or harder depending on the users own skill level.

Appendix

Figure 1: Email Requesting to Change Requirements



Requesting to Change Requirements Inbox ×

Cassie Lillystone <cll552@york.ac.uk> 28 Jan 2020, 11:55

to Dimitris, bcc: Lucy, bcc: Matthew, bcc: Bruno, bcc: Alasdair, bcc: Jordan

Hi,

For SEPR Assessment Three, we had a couple of ideas for the game we have taken over.

Firstly, the minigame. In the other group's requirements they specified the type of minigame they want. They said they wanted a platform based game inspired by Super Mario and Flappy Bird. This is very different from out initial idea for a minigame, which was a pipe style game where the player rotated pipes to complete the water flow in order to refill the fire truck. An alternative idea we had was a game in which different tiles on the map would show up as blue and the player would have to move the firetruck to several of these blue tiles to refill i.e each blue tile partially filled the tank. This idea of a minigame would happen on the same map however the other gameplay would be suspended. We would like to know if we would be able to implement one of our minigame ideas and disregard their requirement as we feel this is more suitable for the game and it's context. We also feel as though a game to fit their requirement may be too distracting from the main game.

Secondly, the current implementation has been done so that the player selects a firetruck before they begin the game, based on the specs of each one, and then has four lives to win the game (meaning they use all four firetrucks which the brief asks for). However, we would like to change it so that these same four firetrucks all appear on the map instead, and the player switches between them throughout the game and only have one life for each truck. The firetrucks would still have different specs, but the player would be able to strategise more throughout the game. Would this be a change you would allow, as it would also disregard a couple of the requirements that the other group laid out?

We look forward to hearing from you.

Thank you

Cassie, on behalf of NPStudios.

Dimitris Kolovos 28 Jan 2020, 19:28

to Nicholas, Javier, me

Hi Cassie,

I agree with both changes.

Thanks,
Dimitris