Change Report

Cohort 1 - Team 2

Team Members

Camran Qadeer

Kieron Jones

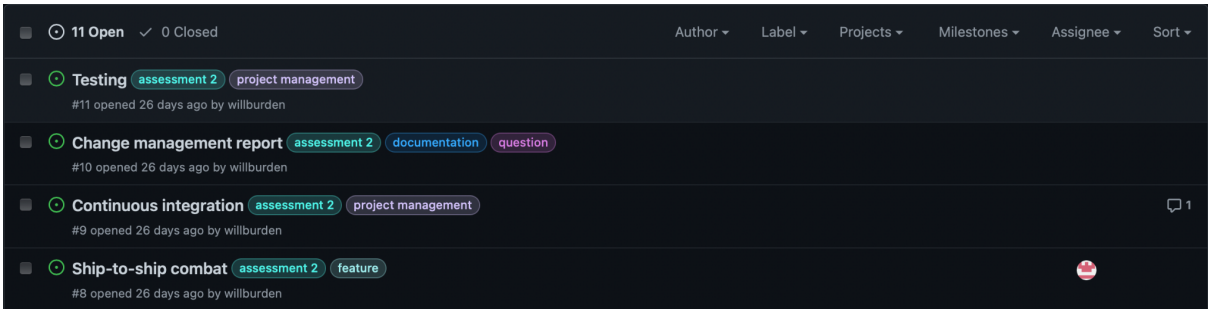
Will Burden

Tabitha Oldfield

Adam Nicholson

2a)

- We needed a change management plan to ensure that the takeover of the previous team's repository was in a controlled environment, and that we could pick out and prioritise the more important changes that needed to be implemented
- The first step was to fork (duplicate) the original repository into our own space so the original was left untouched but could still be referred to whenever necessary
    - We could restore parts of the original if our own repository became unusable, perhaps to a irreversible merge that broke the project
- Using the original team's requirements document, we looked at what features they hadn't implemented and played the game to confirm them
    - Here we also noted any bugs or any other alterations we wanted to make
- To keep track of all the changes to be made, we used the **GitHub Issues** feature as this helped us clearly organise and assign different tasks to people
    - Tasks were sometimes assigned to multiple people, and we could comment on an Issue to update the other team members on the progress being made
    - An 'Issue' is also tagged with the current phase of development it is at, such as 'Completed



- The Issues generally fell into these four categories:
    - Documentation (the original documents were copied and left untouched)
    - Assessment 1 features that still needed to be completed
    - New assessment 2 requirements from the stakeholder
    - Testing
- The main factors* that influenced the making and priorities of these Issues were:
    - The consequences of not making the change:
        - Does it affect the system itself?
        - Was the requirement specifically asked for, or just recommended by the stakeholder?
    - The cost of making the change:
        - Time required
        - Was there a risk of introducing bugs if the change was quite large

*as recommended by *Sommervile's Software Engineering Book [1]*

- The Issues we made were discussed in a meeting where we talked through the old and new requirements and assigned members to each one. Typically, features in the game would be given to one person but something more general like documentation was to be worked on by everyone
- Each Issue would be mentioned again whenever necessary in subsequent meetings in order to keep everyone updated on the progress
- If everyone was happy with a change, this could be reviewed and merged into main, and this was easily done by having everyone work on their own contained branches
- However, once the CI plan was implemented, merges were automated which removed the need to have everyone meet and manually review each Issue

Requirements

- Before starting any implementation, we had a meeting where we went through the original team's requirements in order to **extract those that weren't implemented for assessment one**
    - This includes those that were specifically asked for by the stakeholder, so the condensed versions that *exclude* the already implemented features are in the altered deliverable
- The new assessment 2 requirements have also been added
- There were a lot of requirements that were not necessary to implement, as they were an afterthought of the original team (such as boss battles). We highlighted parts of the table into:
    - **Blue**: Features that do not need to be implemented (the stakeholder did not explicitly ask for), so are **low priority**. The original team has also claimed that some of these were implemented, but we did not find evidence of such
        - A lot of blue requirements were not mentioned in the brief or by the stakeholder during practicals/meetings
    - **Yellow**: Assessment 2 requirements; **high priority**
    - **Red**: Either features that the original team have claimed *were* implemented (based on their implementation document), but aren't apparent in the game, or features from assessment 1 that they did not complete
        - Based on their 'items not implemented' table in their Implementation document, there are a lot of requirements that are missing due to not being in the game
        - One such example is 'UR_FRIENDLY_BUILDING_INTERACT', which is not in the game as there is no apparent sense of interaction between a 'friendly' building
- The tables are different in format, as they are now colour-coded unlike the original
    - This aims to improve the readability

Altered Deliverable:
https://eng1-c1t2.github.io/York-Pirates-2/pdfs/new/Req1_New.pdf

Abstract and Concrete Architecture

a)

Abstract

- The abstract architecture does not need changing from the original because we are only working on the concrete for the new requirements, so we expect no changes for the base framework
    - There is no new significant part of the game framework to be added, other than a few new classes and changes made to existing ones
- By request of the stakeholder, we are not adding any of the testing framework to the architecture

Concrete

- New class diagrams have been created for the new requirements we have implemented, along with removing or altering the existing ones based on our takeover of the code

- The larger diagrams that the original team used took up entire pages, which is why we have **only included the classes that have been changed or added**
    - The entire architecture can still be found here, with the alterations:
    - https://drive.google.com/drive/folders/1Ux-RdH39QXHEAT0aGJmCgRMJGW-bPU9V?usp=sharing

- An issue that we came across during implementation was that the original team had left many unused methods, variables, etc which 'clogged' up a lot of the classes
    - Unfortunately, we only really discovered this after selecting their project for takeover
- Using IntelliJ's built-in problems detector, we were able to comb through each class and remove unused code
    - This was done continuously across development, to try and avoid large conflicts across the entire game
    - For example, there was a class called 'Text' where not a single method was being used in the game, and deleting it confirmed our suspicions that it would have no effect to any aspects of the UI
- The class diagrams have been updated to reflect any deletions that have been made

b)

(Relations To The Requirements)

- This section has been updated to match our newer set of requirements, most notably, the ones for assessment 2 that the stakeholder requested
- Like the original team, we have gone through the important ones and explained how the changes to the class diagrams reflect what has been implemented

Altered Deliverable:

https://eng1-c1t2.github.io/York-Pirates-2/pdfs/new/Arch1_New.pdf

Methods and Planning

a)

- For this deliverable, the way our team operated was marginally different from the original team, but the important points will still be mentioned
- In the deliverable, we have added some points to compare our methods to the original team's
    - For example, we now have to handle a website with GitHub and HTML instead of a 'drag and drop'-style website builder called Notion, which we used in assessment 1

b)

- The original team allocated official roles to people, such as 'Meeting Chair' and 'Secretary'
- This is not something that we did in assessment 1, so we decided to try and follow their method, as well as seeing if we could create other roles

c)

- The original team used a 'task breakdown' table to plan out the entire assessment period
- We felt that this format was a bit messy, especially since it clogged up their website, so we continued with solely using a PlantUML gantt chart instead as this was what worked for us during assessment 1
    - It was a lot easier to do PlantUML code and have it automatically visualised for you compared to making tables in Excel

Altered Deliverable:
https://eng1-c1t2.github.io/York-Pirates-2/pdfs/new/Plan1_New.pdf

Risk Assessment and Mitigation

In the updated deliverable:
Blue Text: Updated text from the original (please note the original team did not order their risks properly (numbers 5, 9, 10, 13 were missing) so we have had to keep them the same for consistency.

Please find below the reasons for any changes made to the risks. Statements about changes made apply to the blue text, such as why we may have changed the level of likelihood or severity. The table is slightly different in style to the original.

R1
'AI interaction' was not clearly explained by the team, but we believe this must be applicable to engaging in combat with enemy ships, which has been implemented, so the likelihood has been moved to low.

R2
The enemy ships are in groups, so there are a lot of cannonballs being fired at once, so we believe the targeting is adequately challenging, especially when considering the chosen difficulty will affect the damage taken.

R3
The only 'decisions' being made are to follow the player and fire at them. The AI is able to get around obstacles in the water fairly well so this risk likelihood is still Low.

R4
During our own development we only came across one issue with physics, but this was due to a minor bug that affected the speed of the cannonball when fired. This was quickly fixed.

R6
We never had an issue with loading times on everyone's machines, and the file size of the executable is not large at all. Any new assets we added were not high resolution.

R7
Nobody experienced any sort of frame drop through development.

R8
There were no apparent slowdowns of the game when it was being rendered.

R11
Even with the addition of obstacles to the tilemap, there were no apparent visual glitches.

Original R12 "AI not advanced as it could be"
Too similar to R2 and R3 so this was removed in favour of those two already implying this statement.

R14
In assessment 1 we found it difficult to meet our own deadlines, so we aimed to be a bit more firm this time on group members when we needed to move onto the next thing.

R15
We did not experience any significantly poor team communication during assessment 1 so this risk is unchanged.

New Risk: R16
Added due to GitHub pages frequently going down for maintenance (or crashing), which sometimes left us unable to edit the website.

New Risk: R17
This risk was added because instantly after takeover, we found that the code was very messy, with an overwhelming amount of classes that really could have been made smaller. As mentioned in the Architecture section, we had to remove a lot of redundant code/methods.

Altered Deliverable:
https://eng1-c1t2.github.io/York-Pirates-2/pdfs/new/Risk1_New.pdf