Change Report

Change management formal approach

The first step of Change Management, as outlined in the IEEE 828 Standard for Configuration Management in System and Software Engineering[1], is identifying the configuration items of the project.

During the first meeting of Assessment 3 we carefully reviewed the project brief to understand which elements had to be maintained and reworked. A table of the identified CIs is available on the team website[2].

After having defined all the CIs we thoroughly analysed the product made by the team we chose and discussed in a stand up meeting the changes we would like to make.

The discussion allowed us to have an extensive list of ideas which were then assessed against the project requirements[3, 4] first, and then against the CIs to determine their impact on the whole product. A table of all proposed changes is available on the team website [5].

Once a change was approved, an issue on JIRA was created so that its implementation could be started.

In order to keep track of all the changes to the code we used GitHub. When implementing a change we created a new branch with the JIRA ticket as its name so that we could track which changes were being implemented on which branch. Then once the change was fully implemented it was merged into the master branch. When merging the changes with the master branch we had to ensure that the new behaviours did not cause any negative side effects which could potentially break other sections of the code. So to ensure this was not the case we tested our program continuously as we developed it.

Generally this strategy allowed us to produce correctly functioning code and avoid any collateral damage, however we had a particular issue with the unit tests which could not be automated because of structural changes to the code architecture, more information about this issue can be found in the Implementation Report[6].

Since we are a small team we followed an informal approach to manage the project documentation. In particular, we decided to progressively update each document as we developed the software. In order to do this, as we felt there were necessary changes to be made for each document, we created a JIRA ticket describing what needed to be changed and assigned it to a team member to complete. The documents were written using Google Docs as it allowed the entire team to review changes and collaborate on writing them.

Following our first meeting we reached out to team Lazer Dolphin Games to discuss their implementation and project requirements, this was done so that we could ensure that we fully understood their software and vision for continuing its development.

Additionally, their opinions enabled us to review and truly understand the scope of our proposed changes.

Explanation and Justification of Changes to GUI Report

The original GUI report mainly described what elements composed the interface, but did not cover what approach was followed to produce that specific UI design and artstyle. Additionally it did not explain the interaction of the user with the system. Because of these reasons and we also elicited new requirements regarding the UI, the graphical user interface report has been completely rewritten[7].

The major changes of the interface aimed at improving the user interaction, which we considered was poor, as the game felt really unintuitive and not friendly to the user.

A full explanation behind the changes made is available in the Updated GUI Report, but the key changes are:

- Added a Main Menu to allow players to choose if to start a new game or load one
- Added a Dialog System, to communicate information to the player and obtain input
- Restructured the in game HUD to be clearer
- Changed perk labels to be more intuitive for new players
- Added End Turn button
- Added Actions label to display remaining player's actions
- Added menu button to pause game; save game and return to main menu
- Created an interface for the minigame

The only component of the UI that has not been modified is the game map.

The key rationale between these changes was to ensure that the game was made to be far more intuitive for new players. To do this we made sure to use standard UI components, as well as terminology and icons common to the strategic game genre, that users are likely to be familiar with in order to minimize how much a player has to learn in order to play the game. All interaction with the game can in fact be done by just using the left click.

Even though there was no information on what influenced the original artstyle, we really appreciated the “Doodled” aesthetics and decided to develop that theme and shape all new UI components to look as if they could be sketched on paper.

Explanation and Justification of Changes to Testing Report

Good quality testing is an important feature to repeat for this assessment so before starting any work we traced the previous teams tests[8, 9, 10, 11, 12] to ensure that the quality so far was up to standard. The style and amount of unit testing the previous team carried out was sufficient enough for us to carry on in a similar format. However due to the previous teams style of architecture we had to modify/remove a few tests they had created and, after continuing implementation on the game very obscure errors appeared in our unity test runner where all tests would fail or wouldn't run. We spent a few weeks trying to resolve the issue however we could not find a solution, therefore we tried many different ways to get as many unit tests running as we could managing successfully to get 16 unit tests pass. This data can be seen in the updated unit test document [13] and within the unity test runner.

We did decide however that the previous teams black box testing was not very thorough and the format of their statistics was not easily traceable. In addition we added many black box tests in this assessment as we introduced a lot of new GUI elements into the game. Any GUI tests we carried out came under black box testing. We decided to reformat the black box tests into one concise document as opposed to two seperate ones as done previously. We took a methodical approach of testing all the core elements within the game taking relevant screenshots for reference to how a specific feature of the game should be working. These GUI elements could only be tested from a users perspective which led to us developing a cause and effect based table where we would test if something had happened what would be the effect and is it working correctly. Data on such tests can be found in our updated black box tests document [14].

During this assessment we have also implemented the game fully according to the deliverable. Hence why adding rigorous play testing was needed as opposed to the previous assessment. In order to do this we made the client play the game initially to make sure the game was up to standard. We also play tested as team members to find obscure bugs which led to a much cleaner final executable of the game.

Further explanation about the procedure followed can be found in the Updated Testing Report available on the team website[15].

Explanation and Justification of Changes to Methods and Plans

The previous team used the Scrum Agile framework which was definitely suitable for this kind of project and was also in line with our team's methods of work. We were already familiar with such framework because we have been using it throughout the first parts of the project, and therefore felt confident continuing to use it.
In addition to that, keeping the same framework allowed us to maintain our team roles as per the previous assessments. This was considered important because we really felt we had found the most suitable positions for each team member.

We considered the software they had already chosen as valid and therefore the systems used for File Sharing, Communication and Version Control are the same as the ones described in their original Method selection and planning document[16].
Furthermore, the tools had been already employed by our team so using them was not an issue.
The only tool we decided to change was Trello in regards to task management. Our alternative was JIRA, because whilst they have similar functionality, it is tailored towards agile team software development and because we had already set up all our projects on it.

The previous team did not specify what programs they used for writing the code and for producing all the game assets. Therefore we decided to use Visual Studio as the preferred IDE; and Procreate and Photoshop as graphic editors. Further explanation on such choices can be found on the team website in the Updated Methods and Plan document[17].

The planning for Assessment 4 was modified relatively little since, as mentioned above, the previous team used the Scrum framework which aligns well with how our team wanted to work.

The main changes made to the planning are: the addition of two main tasks regarding the presentation of the software produced during Assessment 3 and the rescheduling of the main task regarding the architecture report, because we prefer to finalize the architecture of the software before starting its implementation.

Since the original Gantt Chart was made using Google Sheets, we decided to recreate it using ProjectLibre because it allowed us to have a much clearer and more detailed visual representation of the project plan, highlighting dependencies and the critical path.
The updated Gantt Chart is available on the team website[18] and the explanation of the format used to create it is also in the Updated Methods and Plan document referenced above.

Explanation and Justification of Changes to Risk Assessment

The approach for identifying and classifying risks was considered valid and therefore has not been modified, the same applies to the Risk Monitoring strategy, which has been useful to periodically review our risk analysis and identify new risks.

Even though the original Risk Table was quite comprehensive, there were no risks pertinent to the take over of a new project, something which did cause issues, especially early on. In particular we had some troubles learning the new project environment and tools, as well as understanding some parts of the code. We therefore added a number of new risks to outline this.

We also changed the system used to assign risks to owners. The original document used a system of assigning each risk to an individual group member, noted by using their initials. Due to the nature of our group, we found it more beneficial to introduce a new system of assigning risks to small groups of people: Group leader, Team leader, Developer and Client Interface.

The Extended Risk Assessment is available on the team website[19].

References

[1] “IEEE Std 828-2012 (Revision of IEEE Std 828-2005) - IEEE Standard for Configuration Management in Systems and Software Engineering” [Online]. Available: https://standards.ieee.org/findstds/standard/828-2012.html [Accessed: Feb. 18 2018].

[2] SEPR “Configuration Items” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/ConfigurationItems.pdf [Accessed: Feb. 18 2018].

[3] SEPR “Requirements Document” Lazer Dolphin Games [Online]. Available: https://sepr-team-margaret.github.io/content/Req1U2.pdf [Accessed: Feb. 14, 2018].

[4] SEPR “Extended Requirements Elicitation” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/ExtendedRequirementsElicitation.pdf [Accessed: Feb. 14 2018].

[5] SEPR “Proposed Changes” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/ProposedChanges.pdf [Accessed: Feb. 18 2018].

[6] SEPR “Implementation Report” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/Impl3.pdf [Accessed: Feb. 18 2018].

[7] SEPR “Updated GUI Report” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/UpdatedGUIReport.pdf [Accessed: Feb. 14 2018].

[8] SEPR “Testing Report” Lazer Dolphin Games [Online]. Available: https://sepr-team-margaret.github.io/content/Test2.pdf [Accessed: Feb. 19, 2018].

[9] SEPR “Unit tests results” Lazer Dolphin Games [Online]. Available: https://sepr-team-margaret.github.io/content/UnitT2.pdf [Accessed: Feb. 19, 2018].

[10] SEPR “GUI testing document” Lazer Dolphin Games [Online]. Available: https://sepr-team-margaret.github.io/content/GUIT2.pdf [Accessed: Feb. 19, 2018].

[11] SEPR “GUI test results” Lazer Dolphin Games [Online]. Available: https://sepr-team-margaret.github.io/content/GUITR2.pdf [Accessed: Feb. 19, 2018].

[12] SEPR “Test design document” Lazer Dolphin Games [Online]. Available: https://sepr-team-margaret.github.io/content/TestD2.pdf [Accessed: Feb. 19, 2018].

[13] SEPR “Updated Unit Testing” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/UpdatedUnitTesting.pdf [Accessed: Feb. 18 2018].

[14] SEPR “Updated Black Box Testing” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/UpdatedBlackBoxTesting.pdf [Accessed: Feb. 18 2018].

[15] SEPR “Updated Testing Report” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/UpdatedTestingReport.pdf [Accessed: Feb. 19 2018].

[16] SEPR “Method selection and planning document" Lazer Dolphin Games [Online]. Available: https://sepr-team-margaret.github.io/content/Plan1U2.pdf

[17] SEPR “Updated Methods and plans” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/UpdatedMethodsAndPlan.pdf [Accessed: Feb. 18 2018].

[18] SEPR “Updated Assessment 4 Gantt Chart” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/UpdatedGanttChartAssessment4.png [Accessed: Feb. 18 2018].

[19] SEPR “Extended Risk Assessment” Risky Developments [Online]. Available: http://www.riskydevelopments.co.uk/documents/ExtendedRiskAssessment.pdf [Accessed: Feb. 18 2018].