

Module	Engineering1 (Eng1) - COM00019
Assessment Title	Assessment 2, Cohort 2
Team	Dragonite (Team 21)
Members	Omar Omar, Rhianna Edwards, Okan Deniz, Craig Smith, Omar Galvao Da Silva, Joel Wallis
Deliverable	Change report

The team took a pragmatic approach to changing the documents. Changes were made in stages to allow for better management and division of labour. For a given document, a team member was assigned to review the document. The initial review focused on the clarity, function and detail level of the content. Small changes were made to sections where the content was not clear. In this stage, any grammatical or structural changes were made. As for the code, the initial review would focus on meeting coding standards and the quality of documentation provided.

The next stage would be to make sure that the document accurately reflects the requirements ordered by the client in the brief and any associated content needed to clarify how it was met, such as architectural diagrams. At this stage, any extensions made to the original requirements were also added to reflect that change. The team member assigned to the document would have to identify potential problems with every change as well as the impact of the change on the project. For example, adding new requirements that require a change to the development cycle. This information would be relayed to the members working on the code or the document who will debate the changes and plan for any applicable amends.

Lastly, the changes would have to be reviewed by the whole team. This final review takes into consideration any information gathered on a change and focuses on incorporating the change in the current development plan. If approved, the change would be implemented and a later review would take place to ensure that the change met its intended purpose.

This cycle allowed the team to be flexible and respond quickly to any changes. However, extra checks were needed to ensure that changes to the code and documentation run smoothly. These involved informing the members in charge of testing of any new feature that will be implemented as well as running a final check on code quality and documentation whenever any changes to the code were finalised.

Changes to requirements

For the requirements, we found that the way the requirements were placed, into tables with colour highlighted sections to signify the importance of each requirement, made it easy to understand. Moreover, the naming conventions used and the content inside each cell were clear and easy to follow. Therefore, the general layout and content available were left intact. Existing requirements met the previous brief provided by the customer and thus were not changed. However, additional requirements were needed to cover points highlighted in the new Assessment 2 brief.

For the tables, the Changes column uses Added, Changed and Removed to describe each unique change.

Please note that the related requirements column is used here to link to other requirements that had to change or be added as a result of the current requirement.

Requirement ID	Changes	Related requirements	Justification
UR_POWER_UPS	Added	FR_UNIQUE_POWER_UPS	To satisfy the requirement related to the power ups in the brief
UR_POWER_UPS_COUNT	Added		To satisfy the requirement related to the power ups in the brief
UR_GAME_DIFFICULTY	Added	FR_GAME_MODES	To satisfy the requirement related to having different levels of difficulty as mentioned in the brief
UR_SAVE	Added	FR_PAUSE_SCREEN	To satisfy the requirement related to game saves as mentioned in the brief
FR_UNIQUE_POWER_UPS	Added	UR_POWER_UPS	An extension to the linked requirement used to provide necessary details.
FR_GAME_MODES	Added	UR_GAME_DIFFICULTY	An extension to the linked requirement used to provide necessary details.
FR_PAUSE_SCREEN	Added	UR_SAVE	An extension to the linked requirement used to provide necessary details.

Changes to Architecture

In terms of Architecture, our team followed the object-oriented design from the previous team's project as this was a convenient way to modify their existing UML diagrams to graphically document the architecture. A reason as to why OOP was chosen in the previous team's project was to create a recognisable methodology for the other team (our team) to follow in Assessment 2 and the ability to indicate where reusability code has been implemented.

This forward-thinking has made OOP very feasible in our case and as a result LucidChart was used to recreate the three charts: Abstract Behavioural View, Abstract Structural View and Concrete Architecture.

Chart 1 - Abstract Behavioural View

This UML activity diagram in the form of a flowchart creating a blueprint for the gameplay process was altered in order to account for the implementation of new features that came together with the new Assessment 2 requirements. The addition of the welcome screen allowing the user to choose a new game or load one of the previous saves as well as the option to pick a difficulty level from easy, medium and hard was now the first screen as opposed to the menu screen in the previous version of the game. The user now had to make these decisions before going on to the decision state of selecting a boat if they had chosen to load up a new game from scratch.

If the user had opted to load a previous game save instead, the flowchart would directly skip to the render boats, obstacles, powerups and progress bar stage as relevant information would have been saved beforehand.

Another new feature was the option to pause the game using ESC allowing the user to either resume, save or exit the game. If the game was to be saved or exited, a decision state of 'Game paused?' would take the user directly to the 'STOP' which is also known as the halting state causing the program to shut down. This decision state checking for game pause was implemented after the rendering of boats, obstacles, powerups and progress bar as pausing is only possible when these things are rendered and consequently, the game is running. On the other hand, opting to resume or not pausing the game at all allows for the program to carry on as usual and go to the state checking 'Last Boat crossed Finish Line?' which is not different to how it was in the previous version.

The final adjustment made to the abstract behavioural view was the check for power-up initialisation. As power-ups were implemented in the same manner as the obstacles, the decision state 'Time for random Obstacle' was adjusted to check for both Obstacles and PowerUps and make the decision to initialise them at the same time.

Chart 2 - Abstract Structural View

The basic class relationship diagram mapping relationships and showing aggregations as well as inheritance between classes was only altered in terms of the implementation of power-ups as this was the only new feature leading to new relationships between classes.

PowerUp was a class that was added to inherit from Obstacle alongside rock, log and goose as this class also extended from obstacle in the code as well as the other listed classes. This was done because power ups have a similar behaviour to obstacles when colliding with a boat. Each individual power-up: Maneuverability, Invincibility, Time Reduction, Repair and Speed Boost had their own classes inheriting PowerUp since they were also classes extending from the PowerUp class in the code.

This allowed for the new inheritance relationships to be clearly and visually mapped.

Chart 3 - Concrete Architecture

The concrete representation of the architecture consisting of a basic UML class diagram during the abstract phase was expanded on further and was designed to fully describe each class as well as the classes that were created more recently with Assessment 2. The new classes now represented in the UML were: WelcomeScreen, Rock, PowerUp, Speed Boost, Repair, Invincibility, Time Reduction and Maneuverability.

To the newly added classes, relevant information regarding variables such as data types were added as well as void functions. Also to the classes that already existed such as Boat, since Power-Up implementations included additional code in boat class, information such as '+ApplyPowerUp():void' were updated in classes created by the previous team too.

These classes were linked using the same format from the previous concrete architecture table and function mappings were labelled for every class relationship.

Relating the concrete architecture to the requirements

In the table relating concrete architecture to requirements, classes were represented alongside their justification & requirement id, changes were made to account for the new classes and the new requirements that were introduced with Assessment 2.

The new classes such as WelcomeScreen, Rock, PowerUp, Speed Boost, Repair, Invincibility, Time Reduction and Maneuverability were added to the table and justified why the corresponding classes existed (e.g. stating that power-up classes such as invincibility and repair are child classes creating unique instances).

All the new classes and old classes were assigned their new relevant requirement ID. For example, the new Welcome Screen met the 'FR_GAME_MODES' requirement whilst the old Boat class now implemented the Power-Ups so it was assigned 'FR_UNIQUE_POWER_UPS' along with its previous requirements.

Changes to Methods and Planning

Our team used an Agile methodology along with the Scrum framework for taking over the other team's project. We chose to follow these methods as it would allow us to easily adapt to the new project and plan our limited time to achieve the revised customer requirements. Not much was changed to this methods section of the document as the previous team also followed the same methodology and framework as we planned to use.

The scrum management tool was updated to Google Drive as this is what we planned to use for creating our own in-house system. This provided many benefits including the ease of access for all team members as the deliverable's repository was also hosted in this same location. Zoom was added to the communications tools section as we used this to set up regular meetings at the end of each scrum sprint. This kept the project running smoothly as the strict deadlines forced work to be completed, reducing the risk of potential backlog. Additional assets for the game needed to be created for the new customer requirements using piskel and photoshop, so these were added to the document also. The implementation tools "Java and LibGDX" were removed from the tool list as these were already used in the project and were stated in the assessment 1 methods and planning document. The testing tools Junit and Mockito were added in the tools section as these were the ideal frameworks that suited our project. We also researched JMockit as an alternative but found that Mockito was suited to our project better. Travis CI was researched and chosen as our continuous integration tool due to the quick setup times which made it ideal for picking up and using quickly in our project.

Descriptions of the flexible meetings held during each sprint were added to the team organization section of the document. Holding these flexible meetings allowed for support and clarification from other team members if a particular task is being found difficult. This ensured that the assigned tasks goals were met by the end of each sprint session, which were then reviewed by the rest of the team.

An additional Gantt chart was created to represent the entire assessment 2 project. This followed the same structure and design as the original assessment 1 chart but included new tasks and milestones to reflect the updated customer requirements. Each of the new tasks were assigned a colour coded priority, along with the start and end dates of each, accounting for the Christmas holidays and revision and exam period. The project evolution description was changed to explain any ongoing modifications that took place, regarding the sprint timeline, for example, extending the sprint periods over the exam period due to lack of time. Our flexible meetings hosted throughout the weeks during sprints allowed us to keep on track so the project would be completed by the deadline.

Changes to Risk assessment and Mitigation

When it came to risks, the general approach to risk management and documentation did not change as it was effective in mitigating risks. No new risks were identified. However, the team found that modifications to the existing risks were needed. These changes included modifying or adding to the current mitigation methods. This was made to better reflect the approach of the current members. The owners of the risks were also changed to reflect which of the team members were currently responsible for them. The table included here will not include the individual owner changes and will focus on other significant changes made.

Please note that in the updated document, modifications are highlighted by a yellow colour.

Risk ID	Changes	Justification
9	Modified - Changed the mitigation approach.	Adding debug periods during development increases the chances of catching bugs.
12	Modified - Added to the mitigation approach.	The previous mitigations were not sufficient
13	Removed	The risk was not realistic as advanced features cannot be implemented without the fundamental features being there
All	Modified - The risk owners were changed.	This was done to reflect the change in the current team members accountable for these risks. This made it easier to track the responsible members when an issue related to a risk needed to be raised.

Document Links

- Requirements Original File:
<https://omar-h-omar.github.io/ENG1-Dragonite-Assessment-2.github.io/docs/deliverables/Req1.pdf>
- Requirements New File:
<https://omar-h-omar.github.io/ENG1-Dragonite-Assessment-2.github.io/docs/deliverables2/Req1.pdf>
- Risks assessment and Mitigation Original File:
<https://omar-h-omar.github.io/ENG1-Dragonite-Assessment-2.github.io/docs/deliverables/Risk1.pdf>
- Risks assessment and Mitigation New File:
<https://omar-h-omar.github.io/ENG1-Dragonite-Assessment-2.github.io/docs/deliverables2/Risk1.pdf>
- Methods and Planning Original File:
<https://omar-h-omar.github.io/ENG1-Dragonite-Assessment-2.github.io/docs/deliverables/Plan1.pdf>
- Methods and Planning New File:
<https://omar-h-omar.github.io/ENG1-Dragonite-Assessment-2.github.io/docs/deliverables2/Plan1.pdf>

- Updated Plan:
<https://omar-h-omar.github.io/ENG1-Dragonite-Assessment-2.github.io/#weekly-snapshots>
- Gantt chart:
<https://omar-h-omar.github.io/ENG1-Dragonite-Assessment-2.github.io/docs/sprint2/gantt%20chart.png>
- Architecture Original File:
<https://omar-h-omar.github.io/ENG1-Dragonite-Assessment-2.github.io/docs/deliverables/Arch1.pdf>
- Architecture New File:
<https://omar-h-omar.github.io/ENG1-Dragonite-Assessment-2.github.io/docs/deliverables2/Arch1.pdf>