Cohort 3 - Group 22: The Wafflers

Eng1 Assessment 1

Change Report

Team Members:

Tunahan Sisman
Yousef Omar
Merrill Davis
Tom Comrie
Rohan Sandhu
Cameron Pounder
Daniel Redshaw

Change Management, Processes and Tools

This section highlights the methods, tools, and conventions the team has used to manage and track the Assessment 1 deliverables and documentation inherited from the previous team's project. To achieve the best outcomes, every change that was made when updating the new code and documentation was reported so they could be kept track of. To plan for this project, the team conducted weekly meetings to discuss task assignments. This ensured that all team members had equal contribution to tasks and were aware of their responsibilities and deadlines.

Documentation Management

Google Docs was used to edit and update the documents due to its collaboration features and version history. Each document has a designated reviewer, whose role is to leave comments to suggest changes to areas of the document that they feel need clarification or extending.

Code Management

For managing changes to the codebase, Github was used as our central version control system. This choice was because of Github's simple branching and commit features, which allowed team members to work on different features synchronously. Pull requests were utilised to ensure no code was added without thorough checks in order to prevent the inclusion of bugs into the main code base. As well as this, updated methods, classes, and other changes made to the inherited code base have been tracked and commented within the code base.

REQUIREMENTS

Original Requirements Document:	Updated Requirements Document:
https://uoycs.github.io/HeslingtonHustleAss2Website/A1/docs/Req1.pdf	https://uoycs.github.io/HeslingtonHustleAss2Website/A1U/docs/Req1.pdf

After evaluating the system and identifying necessary enhancements, the main improvements identified for the requirements of the system were to improve the clarity, accuracy, and alignment of the game design with user expectations and technical feasibility. Also, the updated list of requirements from the client was added into the requirements document.

Change 1:
Revised How requirements were elicited

Previous Description: From our customer meeting and looking at the product brief, we categorised the requirements
Revised Description: The requirements were elicited and negotiated using the product brief… during the development of the product. (full paragraph at beginning of doc)
Justification: The original explanation of how requirements were elicited was very brief and lacked detail on requirement elicitation. Additional requirements were added during the period of development and the requirement elicitation statements should reflect this.
Impact: This better explains how requirements were elicited to prospective customers and promotes more effective elicitation if followed for any future requirements by the team.

Change 2:
Added User Requirement for Multi-OS Support

Additional User Requirement: UR_Multi_OS
Description: The game shall operate seamlessly across multiple operating systems, including Windows, Linux, and MacOS.
Justification: The need for explicit multi-OS support was identified to ensure accessibility on various student devices.
Impact: Ensuring cross-platform compatibility increases the game's accessibility and usability, aligning with the inclusivity goals of the project.

Change 3:
Added the new Requirements requested by the client

User Requirements: UR_Leaderboard, UR_Streaks, UR_New_Areas
Functional Requirements: FR_Leaderboard, FR_Leaderboard_Input, FR_Streaks, FR_Streak_Achievements, FR_New_Areas, FR_Map_Location_Display, FR_Interaction_Points
Justification: These new requirements were added to accommodate the features specified in the new brief.

Impact: Adding these requirements ensures that the game meets the expanded scope and functionality outlined in the updated project brief provided by the client.

Change 4:
Revised Functional Requirement for Game Session Duration and Multi-OS Compatibility

Original Functional Requirement: FR_Game_Duration_and_Compatibility
Previous Description: The game should have a reasonable level of difficulty.
Revised Description: The game session lasts exactly ten minutes, providing a quick and enjoyable experience, compatible with Windows, Linux, and MacOS.
Justification: Clarification was needed to specify the exact game duration and explicitly list the supported operating systems.
Impact: This precision in game duration and system compatibility ensures a consistent and fair user experience across different platforms.

Change 5:
Reclassification of Misclassified Functional Requirements

Issue: Several requirements such as FR_Difficulty, FR_Simplicity, and FR_Background_Display were initially misclassified as functional.
Correction: These have been reclassified as non-functional requirements focusing on game quality attributes like ease of use and visual appeal.
Justification: This reclassification aligns with standard practices in requirement specification, ensuring that functional requirements strictly define what the system should do.
Impact: Clear categorization aids in better system design and testing, ensuring that each aspect of the game is developed and evaluated correctly.

Conclusion
The changes made to the requirements documentation address identified issues and ensure that the game meets both the customer’s expectations and technical standards. These refinements enhance the document's clarity, usability, and traceability, which are crucial for the successful development and deployment of the game.

METHOD SELECTION AND PLANNING

Original Method/Planning Document:	Updated Method/Planning Document:
https://uoycs.github.io/HeslingtonHustleAss2Website/A1/docs/Plan1.pdf	https://uoycs.github.io/HeslingtonHustleAss2Website/A1U/docs/Plan1.pdf

When considering the team's methodology and planning, alternative engineering methodologies and platforms were considered and compared to help justify our choices of IDE and game engine. Following discussions of how work was done, focus was placed on role designations and team coordination, exploring how these changes could be reflected in the Gantt chart.

Change 1:
Added explanation to choice of GitHub as a version control platform, as opposed to others

Justification:
The selection of GitHub as a version control platform was strategic, based on team proficiency. The team wouldn't have to waste valuable time learning a new tool, instead opting to use that time mastering the game engine. The branching capabilities of GitHub and ease of integration allow the team to focus on separate features in parallel, resolving conflicts that could arise from simultaneous code implementation. Collaboration is at the core of the project, and so the attributes of the chosen version control platform must make this easier.

Change 2:
Added justification for selected IDE

Justification:
The team's IDE of choice was IntelliJ, given its built in version control compatibility and assistive code features. The IDE should be easy to use, providing a suitable interface for code implementation. Other IDEs were considered, but decided against due to compatibility issues with LibGDX, our chosen Java game library.

Change 3:
Change details of Discord communication

Justification:
The team used a different set of channels within our Discord server to organise communication and ensure that laser focus is kept on the project tasks at hand. Given the conveniently customisable nature of a Discord server, team communication can remain efficient and organised since the Discord server can be tailored to the team's needs.

Change 4:
Specify role designations

Justification:

Assigning specific responsibilities to team members prevents overlap in work being completed and allows each member to focus on a task relating to their strengths, maximising efficiency. With each member being assigned a role, it became clearer as to what work was being done by who, as well as abstracting the entire project into more manageable tasks.

Change 5:
Further justify adherence to methodology and final coordination.

Justification:
The structured approach taken by the team is reinforced by adherence to the Scrum methodology. Given the importance of a cohesive project management narrative, and the adaptability of team members, the benefits of regular communication and collaboration kept the team up to date with every area of the project, allowing for shifts in workloads if a task was taking longer than expected to maximise the throughput of each part of the project. Our thorough quality control process is highlighted by the comprehensive final meeting to ensure the project was ready for evaluation.

Change 6:
Added Gantt charts for weeks proceeding the team taking over the project and a new corresponding work breakdown

Justification:
In order to stick to the structure put forward by the previous team, new Gantt charts were made to allow the team to better understand the workflow on a week by week basis. A new work breakdown was also made to abstract the new parts of the system, and better understand what each new deliverable required.

RISK MANAGEMENT

Original Risk Management Document:	Updated Risk Management Document:
https://uoycs.github.io/HeslingtonHustleAss2Website/A1/docs/Risk1.pdf	https://uoycs.github.io/HeslingtonHustleAss2Website/A1U/docs/Risk1.pdf

Changes to the risk report were made with the intent to clarify and further justify the risk management strategies utilised during the project.

Change 1:
Added two more classification options in the “Risk Type” field of the risk register: TECHNOLOGY and BUSINESS

Justification:
Further separation of risks into more specific classes enables the team to split focus and tend to risks with more precise awareness. Where the severity scale places a sense of priority on risks, the added classes provide valuable insight into their root cause. While some risks are more likely to occur than others, preparing for all of them ensures that there are no delays to project progress. The ability to tackle risks quickly and directly from the root is massively valuable.

Change 2:
Added further justification for the design of the risk register

Justification:
The design of the risk register is motivated by the need for ease of access to information and simple comprehension. Throughout the duration of the project, regular references need to be made to the register to ensure that risks may be prevented before they pose any issues to the project’s development. The primary aim with the project is to ensure all requirements and quality standards are met, achieved with the aid of a systematic and disciplined approach to risk management that promotes prevention before mitigation.

Change 3:
Added description of different mitigation strategies, including team member dependencies, team communication, documentation, and redundancy analysis

Justification:
Given the collaborative work of multiple team members on different aspects of the game, highlighting potential dependencies enables the proactive mitigation of risks via resource reallocation and timeline adjustment. Effective communication ensures aligned team goals and expectations, relating to the production timeline and quality. Emphasising communication strategies in the risk report encourages prompt and smooth response to production issues. Documentation captures product information on a timeline basis, which directly helps the team keep on top of deadlines, whilst also sharing all information relevant to development with all stakeholders. The risk of information loss, which could prove detrimental to the development lifecycle, is also reduced. Redundancy strategies work to

achieve all of the above goals, protecting the integrity of documentation and allowing team members to switch between development roles with ease. A flexible team with a variety of mitigation strategies will be able to stay on track regardless of any issues encountered.

Change 4:
Updated the risk register: adjusted “Owner” field to reflect new team members and removed R20 field

Justification:
With taking over the other team’s project, it’s important to reassign risk ownership to ensure that these risks continue to be monitored. The R20 field was irrelevant; the game has no age rating therefore no requirement to follow related regulations.

ARCHITECTURE

Original Architecture Document:	Updated Architecture Document:
https://uoycs.github.io/HeslingtonHustleAss2Website/A1/docs/Arch1.pdf	https://uoycs.github.io/HeslingtonHustleAss2Website/A1U/docs/Arch1.pdf

The primary focus when making changes to the architecture is to provide clearer relationships between the functionalities. The changes to architecture were only minimal since the classes that were already in place were worked off of, with most of the previous diagrams and information still being factual for the continuation of the project.

Change 1:
Added more detailed information on the diagrams. Added explanation to the arrows and dashed lines on the scenarios table, and explanation on the classes in the diagram

Justification:
The classes in the diagrams are pretty straightforward, however some need further explanation. Classes such as structures don't need explaining, but core, main and settings do. A clear explanation will allow stakeholders and developers to make changes and understand the architectures. The explanation of the arrows and dashed line will give a better understanding by adding a graphical notation so the reader can not misunderstand the diagrams.

Change 2:
Updated the explanation of entities: time and energy

Justification:
In the old architecture energy and time were both described as “a core mechanic of the game given to us by the specification”. This is very vague and does not go into detail of what they are for. Time is used to show how long the user has spent playing the game. Energy can be lost when studying or conducting activities, and can be gained when eating or sleeping.

Change 3:
Updated the scenario tables.The second diagram “Interact with building” is renamed to “playing the game”

Justification:
The “player opens game” table will look very different as the user now has the option to start or go to settings and make changes to the game. The name change of the second diagram is because the term “Interact with building” is very specific as the user can do more than that throughout the game.