Change Report

Group Name: Group 11

Group Number: 11

Group Members:

Zubair Shaik, Dom Decicco, Damian Myszko, Yousif Zuhair, Elijah Jones, Charlie Somerscales, Tawfig Osman

Processes, Tools and Conventions Used

TOOLS:

In order to plan how we were going to approach assessment 2 we used PlantUML. PlantUML makes it very easy to create diagrams we could use to lay out everything that needed to be done and make effective use of our limited resources. In the first assessment we used PlantUML and so did team 1, so we believed the best course of action was to use it again.

To communicate throughout the project we used Discord. Team 1 had used whatsapp, but during the first assessment we found discord to be very good and as the communication software used has no effect on the deliverables, we saw no need to change software from what we already knew.

For the documentation we decided to use Google drive/docs, as we can very easily convert the pdfs from team 1's website into docs we can edit on google docs. Team 1 had used Google docs in the first assessment and so had we so it seemed like the most obvious choice. The software allows for collaboration and easy sharing on google drive and access to version history to keep track of any changes made to the original documents.

To make and monitor changes to team 1's code we used Github. Github was our first choice as it is the most widely used version control software available and both team 1 and ourselves had used it in the first assessment.

Requirements

Original Requirements, Updated Requirements

The requirements document has been updated to address key feedback points and enhance clarity, completeness, and structure. The changes reflect a more detailed description of the system, improved presentation of requirements, and the addition of new functional and nonfunctional requirements to align with project goals and stakeholder expectations. All changes are highlighted in yellow.

Changes Made:

1. Introduction and System Description:
   1.1. Added to the introductory section to fully satisfy the SRS (Software Requirements Specification) approach that was partially used on the first deliverable
   1.2. Detailed system description including purpose, audience, user needs, business requirements, and key features:
      1.2.1. The game simulates university life and is playable on desktops and laptops.
      1.2.2. Users navigate a virtual campus map, interact with locations, and complete activities for points.
      1.2.3. Business objectives include an engaging user experience, campus map representation, study-focused gameplay, limited interaction resources (time and energy), and compliance with licensing and asset use regulations.
      1.2.4. User needs are addressed with an easy-to-understand interface, realistic university settings, and character customization.
2. Detailed Explanation of Requirements Presentation
   2.1. Added a detailed overview of how requirements are presented, emphasising the use of tables to categorise and communicate functional, non-functional, and user requirements.
3. Functional Requirements (FR) Updates:
   3.1. Updated the description of character interaction (FR_CHARACTER_INTERACTION) to accurately reflect user interaction with game objects and locations.
   3.2. Introduced new functional requirements:
      3.2.1. FR_MINIGAMES: Incorporate mini-games within the gameplay experience.
      3.2.2. FR_STREAKS: Implement streak-based achievements to reward consistent user engagement.
      3.2.3. FR_LEADERBOARD: Include a leaderboard feature to track and display user scores.
4. Non-Functional Requirements (NFR) and Constraint Updates:

4.1. Removed constraints (TECH_CONSTRAINT_CODE_LANG, TECH_CONSTRAINT_JAVA_VER).

4.2. Added new non-functional requirements and constraints:

4.2.1. LICENSING_CONSTRAINT: Adhering to licensing requirements for third-party assets and libraries.

4.2.2. NFR_IMAGES: Ensuring that the software complies with licensing agreements for third-party libraries, frameworks, or assets used in the development.

4.2.3. NFR_QUICK_START: Ensure quick start-up and response times for the game.

4.2.4. NFR_AGE_COMPATIBILITY: Address compatibility of game with different ages

4.2.5. NFR_RESPONSE_TIME: Specify response time criteria for user interactions.

4.2.6. NFR_GRAPHICS: Define graphical quality and rendering standards.

5. Requirement Refinements:

5.1. Renamed and refined existing requirements:

5.1.1. FR_ACTIONS -> FR_ACTIVITY

5.1.2. FR_ACTIONS_INIDICATIONS -> FR_ACTIVITY_INDICATOR

5.1.3. FR_ACTION_TYPES -> FR_ACTIVITY_TYPES

5.1.4. FR_ACTIVITY_INSUFFICIENT -> FR_ACTIVITY_INSUFFICIENT

The revised requirements document ensures alignment with project objectives and stakeholder expectations. Enhanced clarity and completeness support effective development, future enhancements, and validation of the software system. Addition of new requirements and constraints enriches the functionality and quality attributes of the game. Improved presentation facilitates understanding, traceability, and communication of system requirements across project stakeholders.

Architecture

Original Architecture, Updated Architecture

The architecture of the project has stayed relatively unchanged except for the introduction of a few classes which support the new minigame and leaderboard features we implemented. All of the new features were designed with the same OOP architecture in mind.

Changes made:

1. New Classes introduced:
    1.1. “PlayerTracker” - used for detecting activity and transition tiles.
    1.2. “MapManager” - ensures maps are loaded and fit to screen correctly.
    1.3. “FoodNinja” - houses the minigame code.
    1.4. “AcademicWeapon” - houses the minigame code.
    1.5. “BasketBall” - houses the minigame code.
    1.6. Minigame components:
        1.6.1. “Bullet” - controls bullet physics.
        1.6.2. “Mouse” - controls how the mouse interacts with the minigame.
        1.6.3. “Obstacle” - controls interactions between the individual minigame components as well as the mouse e.g. bullet destroying a target.
        1.6.4. “ObstacleSpawner” - generates obstacles in minigames.
        1.6.5. “MinigamePlayer” - controls collisions, shooting and movement of minigame components.
    1.7. “LeaderboardManager” - handles the saving and reading of leaderboard files.
    1.8. “LeaderboardEntry” - prepares a new entry into the leaderboard file.
    1.9. “LeaderboardScreen” - displays the leaderboard.
    1.10. “PauseScreen” - the screen that is shown when the game is paused.
2. The class diagram was simplified and new classes were added to the diagram.
3. We changed the definition of our architectural style to OOP. Even though we did not change the style, we thought it more closely resembled OOP, as opposed to ECS.
4. Provided justification for the use of PlantUML in producing class and sequence diagrams.

Method Selection and Planning

Original Method Selection and Planning, Updated Method Selection and Planning
The method selection and planning document has been updated to be more in line with the methods we agreed upon with respect to our feedback.The structure and clarity of the document overall were strong, but areas such as identifying leadership roles within the group have been developed to fulfil the requirements of the documentation. Also the formatting was inconsistent at times. Finally the weekly plan was updated as we progressed through the second assessment.

Changes made:

1. Methods - Team 1 decided upon using a combination of agile and waterfall methodologies, but we have decided to use an agile methodology: scrum. We followed this methodology during the first assessment and found it to be very effective and flexible.
    1.1. Updated agile paragraph to support as we have experience with it.
    1.2. Updated hybrid to illustrate it was team 1's idea.
    1.3. Added a final decision paragraph.

2. Tools - Added some notes to discuss alternatives to what was chosen and small changes to match the tools we used for our project when this differs from what was already written
    2.1. Added VSCode to the development tools.
    2.2. Communication tool changed from whatsapp to discord.
    2.3. Deleted Jira paragraph as we didn't use that cause we didn't think it was necessary.
    2.4. Edited PLantUML, GitHub and Google drive/doc paragraph to include a bit more info as to why we chose to use them.
    2.5. Added references to all tools used so anyone can easily go look at their websites

3. Organisation - Modified to match our group organisation
    3.1. Added information about group leaders
    3.2. Changed the weekly meetings to reflect the fact our meetings were virtual

4. Systematic plan - The structure of the systematic plan is great, as it clearly shows the key tasks and their dependencies, priorities and start-end dates; so for the most part we are going to keep it as is and continue to add to it as we move through the project. That being said the diagrams on the document are unclear so we are going to draw up new plans and keep them separate on the website so they don't have to be rescaled to fit the format of the document.
    4.1. Added Assessment 1 and Assessment 2 titles to differentiate between the 2.
    4.2. Added weeks 1-4 for Assessment 2.

5. General formatting - Formatting changes to increase clarity
    5.1. Changed section labels from "4A, 4B, 4C" to headings for additional clarity.
    5.2. Made sure formatting was consistent throughout.

Risk Assessment and Mitigation

Original Risk Assessment and Mitigation, Updated Risk Assessment and Mitigation
The risk assessment as laid out by team 1 is well formatted and the process to create the document is well documented. Due to this we won't be changing the format of the document or the outline of the process. Also the collection of risks and their features is fairly extensive and covers the risks we could imagine may affect our project. The main changes were updating the risks that either don't apply to us or we have deemed them to be worth a different severity/risk and adding our own risks that have been left out.

Changes made:

1. Names - Changed the names of the risk owners to match the members of our time responsible for ensuring the risk is monitored.
2. Current risks:
   - 2.1. R1 - Changed likelihood to medium and talked about maybe ensuring regular meetings with clients to discuss game state.
   - 2.2. R3 - Changed likelihood to low as it is very easy to find non copyrighted music and added to mitigation to make it clearer.
   - 2.3. R4 - Changed likelihood to low because it is fairly simple to add and implement assets and in the case of problems there are a lot of tutorials.
   - 2.4. R6 - Changed to medium severity as ui elements being miss-scaled could cause a user to miss something important
   - 2.5. R7 - Deleted as we are all on windows so this risk is redundant. Updated the names R8, R9 and R10 to R7, R8, R9 respectively so there is no gap where R7 should be.
3. Added R10 and R11 as I feel they are appropriate risks to consider but haven't been mentioned by team 1.