CHANGE REPORT

GROUP 5 - BITCRUSHED BOB

Maryam Mathews
Joseph Hinde
Jacob Mace
Will Aston
Zathia Jacquesson-Ahmad
Bulganchimeg Munkhjargal
Evan Weston

Processes, Tools, and Conventions

To manage the inherited Assessment 1 deliverables, documentation, and code, our team followed a structured workflow that combined review, tracking, and version control. We began by reviewing all existing work and identifying areas requiring updates or fixes. Change requests were initially captured using our issue reporting system (Jira), where we logged and prioritised tasks based on their impact on project scope. Minor change requests were handled directly, as formal change request forms were deemed unnecessary for our project's scale.

Once a change was approved, team members implemented the updates locally and performed testing to ensure correctness. Changes were then submitted for review via pull requests, allowing peer feedback and ensuring code quality. For significant changes, a more formal review process was conducted before merging. After review and approval, the change request in Jira was closed, and the updated code was integrated into the main branch.

This combination of Jira for issue tracking, Git for version control, and pull requests for review ensured that changes were systematically planned, implemented, and recorded, while maintaining team coordination and project integrity.

Only major, scope-affecting changes are reported below. Minor cosmetic, textual, and low-risk corrections are not repeated here in order to avoid obscuring the significant architectural and requirements-level changes.

Requirements:

Original deliverable URL: https://eng1-c1g11.github.io/assets/Req1.pdf
Updated deliverable URL: https://escape-from-uni.github.io/web/assets/Requirements2.pdf

Summary of changes:

Requirements Report:
- Change: Extended the requirements elicitation section of the report
- Justification: To reflect additional requirements gathered for Assessment 2, providing a clearer explanation of how the game was expanded and refined to fully implement the updated product brief.

UR_LEADERBOARD:
- Change: Added UR_LEADERBOARD as an additional user requirement, describing that the game shall display the top 5 scores.
- Justification: This was added as the extended product brief stipulates that the game must contain "a leaderboard with the name and score of the top 5 scores".

UR_NEGATIVE_EVENTS:
- Change: Changed UR_NEGATIVE_EVENTS to the updated number of required negative events (5).

- Justification: The Brownfield development phase of the assessment requires implementing the remainder of the product brief, and so the requirement was changed to reflect this extension.

UR_POSITIVE_EVENTS:
- Change: Changed UR_POSITIVE_EVENTS to the updated number of required positive events (3).
- Justification: The amount of positive events needed for assessment 2 was increased from 1 to 3.

UR_ACHIEVEMENTS:
- Change: Added UR_ACHIEVEMENTS as an additional user requirement
- Justification: This was an extra requirement only for assessment 2 and was not required by assessment 1 as outlined in the product brief.

Unchanged Components:

Single Statement of Need (SSON)

- Justification: The SSON describing a university-themed maze escape game with a time limit remained unchanged.

Core Gameplay Objective

- Justification: The requirement for the player to escape a university-themed maze within a time limit was retained.

Event-Based Gameplay

- Justification: The use of positive, negative, and hidden events as core gameplay mechanics remained unchanged.

Scoring and Do-Not-Save Constraint

- Justification: The game continued to calculate a final score and did not allow progress or scores to be saved after completion.

These new and modified requirements were used as the baseline for architectural extensions and sprint planning in Assessment 2, providing a controlled mechanism for scope growth.

Architecture:

Original deliverable URL: *https://eng1-c1g11.github.io/assets/Arch1.pdf*
Updated deliverable URL: *https://escape-from-uni.github.io/web/assets/Architecture2.pdf*

Summary of Changes:

Systems and Components:

- Change: Entities, components, and systems were added. Existing components were also updated to allow for better code reuse when extending the current implementation - such as changing GooseComponent to ChaserComponent to be reused for the added negative events.
- Justification: Due to the extended product requirements, as outlined in the updated product brief, additional systems, components, and entities had to be added in order to implement the additional events, achievements, and leaderboard.

CRC Cards:

- Change: First, CRC cards were updated to add missing systems and components that were implemented in the code, but absent from the CRC cards. Then, the additional ECS elements were added.
- Justification: Adding missing systems and components ensured alignment between the design documentation and the existing codebase, reducing inconsistencies. Incorporating the additional ECS elements provides a more complete and up-to-date representation of responsibilities and collaborations, improving clarity for development.

Requirements Met By Systems:

- Change: Updated the table with the new systems and the new requirements.
- Justification: This ensures that the new requirements are being met, and that there are no unnecessary systems (they all are developed to satisfy the requirements).

System Interaction:

- Change: The inherited method of system interaction using messages was maintained, but additional messages were implemented to allow the new systems method of communication.
- Justification: The original method of system interaction was designed to be extendable, making it suitable to support additional communication needs. The introduction of additional messages enables the new systems to communicate effectively while adhering to the established architecture, ensuring backward compatibility and consistency.

Behavioural Diagrams - Activity Diagram:

- Change: Extended to add additional systems.
- Justification: Due to the additional event requirements, new systems had to be implemented, and therefore the activity diagram was changed to now display these extra systems.

Behavioural Diagrams - Sequence Diagram:

- Change: Updated to include additional collision events the player can interact with during the game
- Justification: The inherited game only had one collision event - the coffee collectable with the speed boost - and so we extended this diagram to reflect the additional collision events that we implemented; the surveyor, lecturer, and clock

Structural Diagrams - Class Diagram:

- Change: Updated the diagram to reflect the additional entities, components, and systems
- Justification: Due to the extended requirements, the class diagram was extended to reflect the new elements.

Unchanged Components:

System Updates:

- Justification: We did not change the system update method because it was already generic and extensible. It updates each system in the same way regardless of the game's complexity, so it could be applied to the extended game without modification. The existing design was flexible enough to support the new systems while still meeting performance and consistency requirements.

Asset Management:

- Justification: As outlined in the asset management section of the report, the original team had already changed their method of asset management from the built-in manager provided by libGDX to their own wrapper class AssetLoader. This streamlined the process of adding and managing assets, and therefore didn't need any change when extending the project.

Engine-Level Architectural Class Diagram:

- Justification: This diagram didn't need to be changed because it describes the engine-level ECS and event-driven framework, not the specific gameplay elements. It defines how entities, components, systems, and the messaging system interact in a generic way, and so new gameplay features (such as lecturer or surveyor) are implemented within this existing framework without altering the underlying architecture.

Method Selection and Planning:

Original deliverable URL: https://eng1-c1g11.github.io/assets/Plan1.pdf
Updated deliverable URL: https://escape-from-uni.github.io/web/assets/Plan2.pdf

Summary of changes:

Development Method:

- Change: Updated the section to reflect our Agile and Scrum approach for Assessment 2, with weekly sprints used to control scope and integration of new requirements.
- Justification: While both teams use Agile methodologies, the previous team described their task breakdown in a way specific to Assessment 1. Our plans reflect the two-group structure and iterative workflow used for Assessment 2.

Source Control:

- Change: Removed the discussion comparing GitHub with Sourcehut
- Justification: We did not use Sourcehut during the development of Assessment 2, so the comparison was no longer relevant. Keeping only the tools we actually used makes the section more accurate and avoids unnecessary explanation of platforms outside our workflow

Team Organisation:

- Change: Updated to match our two-group structure (Technical and Non-Technical)
- Justification: Even though the previous team had the same team structure, they had stricter rules, such as assigning one more task to each team member each week. Our team operates more flexibly, with members picking tasks as needed while collaborating and offering support to each other.

Project Plan:

- Change: Revised the project plan to reflect our actual workflow - our weekly sprints.
- Justification: Although our plan was broadly similar to the previous team's sub-group structure, we improved it based on our experience, making it more structured, iterative, and aligned with the requirements and workflow of Assessment 2.

Weekly Plan:

- Change: Completely revised the weekly plan to reflect the workflow and requirements for Assessment 2.
- Justification: The original plan was designed for Assessment 1 and no longer matched our progress, deadliness, or team structure. As the project developed, our priorities and tasks shifted, so a new plan was necessary to accurately represent our updated timeline and approach

Gantt Chart:

- Change: Updated the Gantt chart to match the revised weekly plan while keeping the original design and layout.
- Justification: Since the weekly plan was restructured for Assessment 2, the timeline in the Gantt chart needed to be adjusted accordingly. However, the existing design remained effective for visual clarity, so only the task schedule was modified, not the chart's overall appearance.

Key Milestones:

- Change: Updated to match the deliverables required for Assessment 2.

<u>Unchanged Components:</u>

Development Method

- Justification: Both teams used an Agile approach with weekly meetings and iterative development, which continued to suit the project and supported flexibility throughout Assessment 2.

Source Control:

- Justification: Git and GitHub were retained for version control and collaboration, as they remained effective and appropriate for managing the project.

Collaborative Tools

- Justification: Google Drive and WhatsApp continued to be used for documentation and communication, as they reliably supported team coordination and information sharing.

Risk Assessment and Mitigation:

Original deliverable URL: https://eng1-c1g11.github.io/assets/Risk1.pdf
Updated deliverable URL: https://escape-from-uni.github.io/web/assets/Risk2.pdf

Summary of changes:

Risk register:

- Ownership:
  - Change: Multiple owners of each risk instead of singular owners
  - Justification: In order for risk mitigation to be effective we decided each risk should be owned by multiple people, this way no mitigation falls through the cracks
- Risks covered:
  - Change: we increased the number of risks covered to reflect the new circumstances of assessment 2
  - Justification: The risks included in assessment 1 cover parts of assessment 2, but not all of it so more risks were needed to be seen and mitigated
  - Change: remove some risks that are inapplicable to assessment 2
  - Justification: Some risks in the risk register are only actable upon during assessment 1 so can be removed for more team clarity during assessment 2
- Risk type:
  - Change: Revised type categories to software, people, management and time
  - Justification: The type categories now reflect the types that the team is already familiar with to have a more effective risk assessment
- Status:
  - Change: added a status column to each risk
  - Justification: A column to reflect the status of each risk allows total transparency between the team and their efforts to mitigate the different risks, this makes the risk register and attempts to mitigate the risks far more efficient

Several of the new risks directly relate to the introduction of leaderboard, achievements, and additional ECS systems, ensuring that scope growth was matched by appropriate mitigation.

<u>Unchanged Components:</u>
Risk assessment process:

- Justification: Our process in acquiring the risk assessment was functionally identical to the other team, both held brainstorming sessions to identify risks. Then evaluated on likelihood and impact followed by deciding on mitigation strategies for each risk.

<u>Integrated Change Management Across The Project:</u>

The transition from Assessment 1 to Assessment 2 was managed as a controlled brownfield evolution of an existing software system rather than a restart. All major scope changes originated in the updated Assessment 2 product brief and were captured as new or modified user requirements (UR_LEADERBOARD, UR_ACHIEVEMENTS, UR_POSITIVE_EVENTS, UR_NEGATIVE_EVENTS).

These requirements were then traced forward into the architecture, where new ECS systems, components, and messages were added or refactored (e.g., GooseComponent → ChaserComponent) to implement the new features while preserving existing behaviour. The “Requirements Met By Systems” table in the architecture documentation ensured that every architectural addition was justified by at least one requirement from Requirements2.pdf.

The project plan was re-baselined to reflect the shift from initial development to maintenance and extension of an inherited system, with weekly sprints used to control the introduction of new features and stabilise the code through integration and regression testing.

The risk register was expanded and restructured to reflect the technical and organisational risks introduced by this scope growth, with shared ownership and status tracking ensuring that risks arising from refactoring and feature integration were actively managed throughout the project.

Together, these four deliverables demonstrate a consistent, traceable approach to change management, ensuring that every major extension was planned, implemented, and documented in a controlled and maintainable way.