2. Change Report [23 marks]:
a) Briefly summarise the processes, tools and conventions your team used to plan, make, keep track of, and review changes to the Assessment 1 deliverables, documentation and code you inherited from the other team. (3 marks, <=1 page)
b) For each of the following items, include a brief explanation and justification of any changes made to the other team's Assessment 1 deliverables (pointing to specific sections of the deliverables that have been added/updated). Include the precise URLs of the original (i.e. the other team's) and the updated versions of the deliverables. The updated deliverables do not need to adhere to the page limits for Assessment 1. If there is no change to report, please state and justify why no change was necessary.
(Maximum 5 marks per item, to a maximum total of 20 marks;

Features Not Yet Implemented

UR_LEADERBOARD - The user should be able to view their past scores via a leaderboard table accessible through the main menu - this should be clear and easy to access
FR_LECTURER - The player should be chased by a lecturer and forced into an unskippable cutscene - the timer should continue to run during this - worsening the player's score.
FR_LOCAL_SCORE - There should be a menu where the user can see a set of saved game scores - only the top ten should be shown along with the details relevant to them
FR_MAZE_EVENT - The player should be able to complete small "side tasks" in the maze in order to earn extra points or increase their "time remaining" during the gameplay - these need not be completed in order to complete the game. (Currently goose food is hidden in maze)
FR_LAKESIDE_EVENT - The player should be able to interact with the area around the lakeside, including the buildings. Completing tasks in each (Currently only lake interaction is when the goose runs into it and the player must follow)
NFR_ACCESSIBILITY - The game should have options for those who are colourblind

https://team-6-eng.github.io/Escape_University_Website/files.html

a) In our team we reviewed the documents of the team we have chosen to identify any clearly unimplemented features that were planned for - such as in the Implementation doc with a list of these. Following from this, we have gone through the deliverables and tried to identify where (if at all) and how well something has been implemented to evaluate if it needs to be changed or added in the first place. This should be cross referenced against the Risk Assessment document to consider how worth it is to implement.

(continue writing this when we've actually made more changes to deliverables ie looking through version control etc)

b)

i) Requirements
https://team-6-eng.github.io/Escape_University_Website/assets/pdfs/Req1.pdf

- UR_LEADERBOARD is being updated so that the player can not only see their own past scores, but also scores of anyone who has played previously. This should make the game more engaging as it allows the user to rank themselves against everyone else who has played.
- FR_RESIZE as a deliverable is likely not being changed - the window should fit the screen of the user - however this has not been implemented so we will be doing this. Currently, the game appears fairly small on the screen and can

take away attention from the game. It is possible we have a windowed and fullscreen such that windowed will appear as it usually does, and full screen will implement the resize deliverable.

- FR_KEYBOARD_CONTROL will be slightly altered to include a shift to sprint feature. This will increase speed minimally (say about x1.2) and is being updated as, upon testing, I feel the game is slightly too slow for the amount the player has to explore. This will also have to update any following-the-player related functionality as otherwise they would no longer be fast enough.

- **Changed version :**
https://github.com/xqg506/Escape_University_Website/blob/3e4e2c90d60c4f700e4b68c265ec596d91d759eb/assets/pdfs/The%20change%20d%20table%20of%20req1.pdf

(ii) Architecture
https://team-6-eng.github.io/Escape_University_Website/assets/pdfs/Arch1.pdf

Changes : No changes made

– Justification : After reviewing the architecture documentation in detail, we determined that the original deliverable already met the objectives and quality expectations of Assessment 1.

Firstly, the report clearly defines the chosen architectural style. The use of a layered architecture combined with object-oriented programming is appropriate for a Java-based game project and is consistently applied throughout the document. The responsibilities of the presentation, domain, and infrastructure layers are clearly explained, and their interactions are logically structured.

Secondly, the connection between the abstract architecture and the concrete implementation using LibGDX is well justified. The report explains how the LibGDX Screen framework influenced architectural decisions, such as distributed input handling and screen-based state management.

In addition, the architectural evolution from early designs to later iterations is clearly documented. The shift away from centralised manager classes towards a more distributed approach is well explained and shows that the architecture was refined based on implementation experience. The lack of major structural changes in later iterations also suggests that the architecture is stable and scalable.

(iii) Method Selection and Planning
https://team-6-eng.github.io/Escape_University_Website/assets/pdfs/Plan1.pdf

Changes : No changes made
After reviewing team6's Assessment 1 deliverables (Plan1.pdf), we evaluated whether any modifications or optimisations were necessary across the required areas, including method selection and planning, development and collaboration tools,

team organisation, and the systematic project plan. Overall, no changes were made to the original deliverables. The justification for this decision is outlined below.

– Justification :
Method Selection and Planning :
No changes were required for this section. The original deliverable provides a clear and coherent justification for adopting an Agile methodology, specifically Scrum, and appropriately adapts it to an academic context through the use of weekly rather than daily sprints. The rationale links the chosen methodology directly to the nature of the project (game development), the small team size, and the limited timeframe. In addition, alternative approaches such as plan-driven development and Extreme Programming (XP) are briefly considered and reasonably dismissed. As the explanation is logically structured, internally consistent, and aligned with the project context, no further improvement was necessary.

Development and Collaboration Tools :
No changes were made to the selection or justification of development and collaboration tools. The original team clearly explains the use of GitHub for version control, Google Drive for collaborative documentation, WhatsApp for ongoing communication, and IntelliJ IDEA as the primary development environment. The choice of LibGDX as the game engine is also justified with reference to community support and available resources. These tools are well matched to the selected Scrum-based workflow and adequately support collaboration, transparency, and incremental development.

Team Organisation :
The team organisation strategy did not require modification. The deliverable demonstrates a well-considered approach to task allocation, where each member is given primary responsibility for specific areas based on their strengths, while still allowing flexibility to contribute to other tasks. This approach is consistent with Agile principles and is appropriate for an academic group project. The communication strategy, combining weekly in-person meetings with continuous online communication, is clearly described and supports effective coordination. Since the organisation model is clear, realistic, and aligned with the project methodology, no changes were necessary.

Systematic Project Plan :
No changes were required to the systematic project plan. The project is clearly broken down into work packages, milestones, and deliverables, and the dependencies between tasks are explained. Importantly, the deliverable documents how the project plan evolved over time through weekly Gantt charts, including adjustments made in response to unforeseen issues. This demonstrates reflective planning and effective use of Agile principles. As the plan is detailed, structured, and responsive to change, it was deemed sufficient without further modification.

(iv) Risk Assessment and Mitigation
https://team-6-eng.github.io/Escape_University_Website/assets/pdfs/Risk1.pdf

Changes : No changes made

– Justification : After reviewing the Risk Assessment and Mitigation document provided by the other team, we did not identify any areas that required modification. Overall, the report is clear, well organised.

The report provides a clear explanation of the team's approach to risk management, including how risks were identified through group discussion and then categorised into relevant areas such as organisation, people, technology, resources, and security. This structure makes the document easy to follow and ensures that the risk registers are well contextualised rather than presented in isolation.

The risks identified are specific to the project and reflect realistic challenges that may arise in a university-based group software engineering project. These include issues related to workload distribution, time constraints, use of unfamiliar tools, version control, and handover of the project to another team. The coverage of risks is comprehensive, and there are no obvious omissions that would require additional risks to be added.

In addition, the risk registers are clearly organised and consistent. Each risk includes a description, an assessment of likelihood and severity, an assigned owner, and a corresponding mitigation or management strategy. The use of unique risk identifiers and the inclusion of a separate management risk register for risks that have already occurred demonstrate an understanding that risk management is an ongoing process throughout the project lifecycle.