Change Report

Group Number: 10

Team Name: Decassociation

Group Member Names:
Mohammad Abdullah
Tom Broadbent
Poppy Fynes
Owen Lister
Michael Marples
Lucy Walsh

From the very start of Assessment 2, in taking over Team 3's project, we needed to track any changes that we made to Team 3's Assessment 1 deliverables - their documentation and their code. To do this, we created and used a Change Management document (can be found at:_____ website link_____) which stores many entries of changes we made to the Assessment 1 deliverables. Each entry has the format of:

Change #:

- **Change requester(s):**
- **Change worker(s):**
- **Start date:**
- **End date:**
- **Document / deliverable affected:**
    - **Impacted classes:**
- **Change:**
- **Significance:**
- **Comments (optional):**

This format for entries allows multiple related changes to be stored together and the key details of the change(s) are noted - who made the change, why the change was necessary, what the actual change was etc. The dates associated with each change also enabled tracking of any issues that arose (mainly with the code to see where it may have been broken etc.). Another significant aspect to note is that the 'Change requester(s)' part of an entry isn't really a request as such; due to the project being quite small-scale, there wasn't a need for a change to be requested and approved because the whole team would already know about the change(s) that needed to be made and we would discuss any issues if they arose in the game code. Issues with the documentation never arose - obviously the documentation cannot be 'broken' or 'riddled with bugs' as the game code can.

To plan changes we needed to make to deliverables, we just used Gantt Charts (can be found at:_____ website link_____), as they worked for Assessment 1 and it seemed reasonable to stick with the same format. If we had used real change requests that had to be approved, changes would have been sort of planned more, but it wasn't practical. One by one, we worked through the Assessment 1 deliverables, checking if they were suitable and actually fitted the brief and any changes were made accordingly (and are documented in the Change Management document).

In addition to the Change Management document, GitHub was instrumental in checking recent changes to the game code. It's quicker and easier to see even the smallest changes to the code through the project's GitHub commits. The Change Management entries detail which classes have been changed in the code but they don't detail specific lines that were added/removed/changed which is quite an important detail when developing the code.

Changes to Deliverables:

I. *Requirements*

II. *Architecture*

- Link to Team 3's architecture original document: https://decassociation.github.io/project_eng1_team3/Arch1.pdf
- Updated version of the architecture document:

  _____

- Changes:
  - Added a justification of use of PlantUML because Team 3's feedback commented on the absence of this justification and PlantUML is a significant part of the architecture design.
  - Added a new behavioural diagram to reflect the game with our assessment 2 changes.
  - Updated the interaction sequence diagram and the gamescreen state diagram based on Team 3's feedback from assessment 1, so that the diagrams make sense, are correct and have a point to them.
  - Removed the serving state diagram because it seemed almost pointless. The diagram just displayed the fundamental idea of serving customers in the game and the new behavioural diagram that we have added covers this a bit as well.
  - Removed the state diagram displaying user input results because, similar to the serving state diagram, it was so high-level that it didn't really have any value or point; it showed the basic idea of what games do - take player input with a keyboard and/or mouse and change the game accordingly.
  - Added new class diagrams of each of the packages; it's split up into packages to be more readable unlike the original class diagram. The original class diagram was very large and unreadable, without a working link to the original. The original was also missing lots of fields/attributes and functions which have now been included.
  - Links have been provided for all diagrams so that they can be viewed larger (making them actually readable).
  - Added a small section at the end just stating our processes for developing the architecture for assessment 2.

III. *Method selection and planning*

IV. *Risk assessment and mitigation*

- Link to Team 3's risk assessment and mitigation original document: https://decassociation.github.io/project_eng1_team3/Req1.pdf
- Updated version of risk assessment and mitigation document:

  _____

- Changes:
  - Extra details were added to the risk register section on the first page, to explain the risk register format in greater detail than the original so it's crystal clear what each column details and why it is included.

- The ‘Implementation’ risk type used in the ‘Type’ column of the risk register has been removed and the ‘Product’ type now encompasses the previous ‘Implementation’ type. This was done because both types were very similar and it wasn’t very clear what distinguished risks of each type.
- Risks have been assigned to our team members instead of Team 3’s. This is of course a very necessary and significant change for monitoring and mitigating the identified risks.
- Some of the original risks have been amended or removed, and a few extra have been added to the original risk register table. Generally the changes to existing risks were just slight adjustments to the likelihood and severity ratings. However, some risks were removed because they seemed very insignificant - not really worthy of being counted as risks. Changes made:
    - Amended R1 description & severity; R2 likelihood & severity; R3 changed from local repository corruption to full repository corruption; R4 severity and mitigation; R5 severity and mitigation; R6 severity and mitigation; removed original R7; removed original R8, removed original R9, removed original R10, removed original R12; removed original R14; removed original R15; amended original R16 description and mitigation. *THE NUMBERS NO LONGER ALL MATCH UP TO THE ORIGINALS*.
    - For example, the original R10 - ‘The requirements change’. This is not a risk. The requirements will not change unless they are misunderstood, which is not a risk, that’s an issue with the requirements and the team. The requirements only changed Assessment 2. The description of it, ‘Leave the description on every function implemented which will make the re-edit faster.’ didn’t really make sense either; I’m not sure what functions they were referring to.
    - One of the new risks relates to the actual gameplay as there were previously no risks about the user experience/gameplay and these are actually quite important even though they aren’t exactly technical concerns that could go wrong.
- Some new risks have been added in a separate risk register below the original. This table is specifically made up of risks from our Assessment 1 risk register that are still relevant and Team 3 did not consider. A few of them have also been changed to reflect Assessment 2 work such as testing.
- A ‘Risk Reviews’ section has been added to the bottom of the document to monitor the risks by reviewing them every couple of meetings and checking if any have become a problem.