Change report:

a) Formal Approach to Change Management

Documentation

Github

Upon starting the project, we read through the breif and added an issue to our project specifying each section of the documentation that needs to be looked at/changed. For each section of documentation (i.e. Risk Assessment, Requirements etc.) we created an independent branch. We did this to allow each change to the core documentation to be approved by one or more other member of the team before being pulled to main. This means that mistakes in grammar or content is more likely to be spotted if multiple team members read through changes.

Furthermore, this allowed us to create a template for pull requests which allowed the person making the changes to specify which issue number is resolved, a description of the changes made and the reasons for making those changes. The ability to specify the issue number relevant to made changes proved to be extremely useful as it meant not only were the specified issues automatically resolved but it also allowed us to easily keep track of ongoing issues. The unique numbers assigned to each change made also allowed for easy referencing to specific changes in our documentation.

Where possible, a single team member did all changes for a single deliverable and then went on to complete the relevant section of the change report so that the information provided in the change report was as accurate as possible.

Implementation

For change management for our implementation, we added each change that needed to be made along with each new feature needed as issues in our repository linked to our project board. When making changes, we wrapped each block of change in comments like `//start of change for assessment 2` and `//end of change for assessment 2` and single line changes just having the comment `//change for assessment 2'`. `We also added` //added for assessment 2 `to the JavaDoc for each new method. This means that to view all changes in our codebase, one can simply search for` assessment 2`. Similar to documentation, our code repository is set up so that pull requests to main must be reviewed by another team member before being allowed to merge. Again this means that mistakes and issues sich as forgetting to comment code are more likely to be spotted. Further more the Continuous Integration running tests on Pull Requests helps with change management as it ensure that the changes made havent effected other parts of the game.

Requirements

8948b74: Removed Unnecessary requirements

https://github.com/engteam14/documentation2/pull/66#issue-1166239474

We felt that some of the requirements the team had previously implemented did not appear in any brief, weren't asked for by the customer or were just deemed to be redundant. For example the requirements 'FR_BOSS_UNLOCK_TRACKING' and 'FR_BOSS_SPAWN' involved the use of a boss that was not mentioned in the brief nor customer interviews and therefore had no justification to appear in requirements. This will change R9 as the estimation of the scope of the project would be significantly increased if these requirements were left in therefore making the time esimate of the prject completion change significantly.

ef9f295: Added requirements relevant to the second stage of the assessment

https://github.com/engteam14/documentation2/pull/68#issue-1166289794

Updated the requirements page to include the new requirements that we received in the brief.

9845cef: Fixed Formatting

Fixed formatting mistakes made when editing the markdown table for requirements.

E1a2386 & 396806d: Changes to priority for requirements

https://github.com/engteam14/documentation2/pull/73#issue-1169705599

We felt that the use of 'shall' and 'May' were redundant measures of a requirement's priority and didn't help management of requirements. For this reason, we changed the priorities of the requirements to high/medium/low which we feel is a clearer measure for anyone reading and helps justify why we went in specific directions with our implementation.

6ec20bc: Changed word 'money' to 'plunder' to match wording in the code

We noticed some inconsistencies in wording between the documentation and implementation sides of this project. Most notably, in the documentation section the word 'money' is consistently used while in the code this same thing is referred to as 'plunder'. Not only did we feel that 'plunder' fit the theme but it was also the word used in the brief which led us to change the documentation in favour of keeping consistency.

9917596: Cleared up points/XP confusion

https://github.com/engteam14/documentation2/pull/77#issue-1173441720

A further inconsistency we noted was the lack of clarity between 'points' and 'XP' which was pointed out in the feedback as an issue. The team felt it was best to merge the two so we ended up merging the two and using 'XP' for both.

Method Selection and PLanning

05e5339: Changed Software Engineering method

https://github.com/engteam14/documentation2/pull/67#issue-1166273468

We decided to change the engineering method from Plan-based to Agile as we divided the team into two parts, Implementation and Documentation, so that we could work on all parts of the assessments simultaneously. We also discussed about using the Scrum framework instead of Plan based and had sprints that were 1 weeklong.

bbe4ad3: Added another key point of discussion for weekly meetings

https://github.com/engteam14/documentation2/pull/67#issue-1166273468

Instead of just discussing about game design as the purpose of our weekly meetings, we changed it and added "documentation" as well, as it was necessary to be up to date with both aspects of the assessment as they go hand-in-hand with one another.

e9c1a4e: Changed Communication and Collaboration Tools

https://github.com/engteam14/documentation2/pull/69#issue-1166296504

We decided to change the Team Meeting tool from Zoom to Discord and in-person meetings as in-person meetings allowed us to separate into smaller teams and work together while allowing us to keep real-time progress of the project and Discord calls were more preferred as compared to zoom as it avoided the hassle of having a Meeting ID and Password to join a meeting and the screensharing capability provided on discord was much better than that provided by zoom, according to the groups point of view.

In addition to having a discord server for general project discussions, we decided to create a "To-Do List" channel, where a team member would post notes regarding what was discussed in the meeting and what needed to be done by the next scheduled meeting.

After some discussions, we changed our task progression tool from Trello to GitHub Issues as this was something new that we thought of trying and also the fact that using GitHub issues, we could keep track of all the commits and why something was changed from the previous project, which would later on help us when writing our change report.

299e9c9: Changed tools used to create Architectures

https://github.com/engteam14/documentation2/pull/69#issue-1166296504

Instead of using draw.io for Abstract architecture and plantuml for Concrete architecture, we used plantuml for both as well as for our Gantt Charts, as we had used this tool for the first assessment and found it relatively simple to use.

25d9e44: Added an explanation for our teams' approach towards Quality Control

https://github.com/engteam14/documentation2/pull/75#issue-1172569118

We added a paragraph explaining how we implemented a Quality Control Process to ensure that our work was of high-quality, and this also helped with mitigating the risk of a team member being unable to complete their tasks due to any justifiable reason, as others would know where to pick-up from.

28638b3: Added a Testing tool

https://github.com/engteam14/documentation2/pull/78#issue-1196040441

We added a small paragraph where we discussed the testing tool we used (gdx-testing) and why we used it.

44fbf5d: Updated Team Management and Changed how Tasks were assigned

https://github.com/engteam14/documentation2/pull/72#issue-1168460153

We combined all the roles discussed by the previous team into one role (Team leader) as they seemed largely interdependent and we added two additional roles, Implementation leader and Documentation leader, where 2 additional members were assigned to each role after a group discussion.

We also changed how tasks were assigned (assigning tasks as the project progressed) to how our team seemed fit, which was by calculating the marks of each section required and evenly distributing tasks to team members based on the calculated marks, ensuring everyone had more/less a similar amount of contribution towards the project and nothing will be left/forgotten till the last minute.

ff9ba36: Changed all of Part C (Systematic Project Plan)

https://github.com/engteam14/documentation2/pull/75#issue-1172569118

Instead of continuing with the previous teams' approach of having a task breakdown table followed by an initial Gantt chart showing a theoretical schedule for when each task should be completed, we decided to design an initial roadmap of all the major tasks that needed to be done (as a Gantt chart) which also showed a theoretical schedule of when each of these tasks needed to be completed. We then justified how we went about creating this chart and why we scheduled the tasks the way we did. We then decided to update this initial Gantt chart every week based on task progression/completion in that week, depicting the evolution of our plan throughout the project. We then added the final Gantt chart to the document so that it would be easy to compare the changes made with regards to the initial plan and all the intermediate Gantt charts can be found on our website.

Furthermore, instead of creating snapshots by condensing the roadmap to keep track of our progress and using a colour-coding scheme to highlight tasks that were due, we decided to use a colour-coding scheme to assign tasks to individuals/teams in the form of a Gantt chart which can be seen here.

Risk Assessment

Note that in this report, unfortunately the risk ID's were missnumbered this was only realised near the end of the project (see commit d722ff5 and d469cf). As a result of this, the risk register numbers in the commits do not line up with the final report. Therefore we have added the final versions register ID's in brackets. The register ID's not in brackets are the ones that refer to the commit in that section.

dd7a86a: Added team specific risks and removed non-applicable risks and changed risk owners

https://github.com/engteam14/documentation2/pull/82#issue-1204388497

ID	Type	Description	Consequences	Monitoring	Likelihood	Severity	Mitigatic
R12	Product	AI not being as advanced as it could be	The AI is either too good or bad. Making the gameplay worse for the user.	not currently happening	M	L	Fake AI via scripted interactic

was removed because we have removed the A* algorithm and simplified the AI for the game, furthermore an advanced and genuine AI system is not a requirement for the project.

Updated the risk owners to be members of our team rather than the previous team to ensure all risks are watched by members of the current team.

Added risk R16(R12) as one of our team members has a child which means they are more likely to become unavailable in the project. This is especially important to list as a risk as the team member is team leader.

Added risk R17(R13) of GitHub going down because the previous team included only codebase related risks but risks of lost work are an important factor to take into account in a software engineering Project

Added R18(R14) as a member of our team is spending 4 weeks in Dubai and may be unavailable in the evenings due to the time difference. This is useful to add as it reminds us to schedule meetings at a time which also suits him. Saud is the leader for documentation so it is important that he is able to attend as many meetings as possible

434dc7a: Changed mitigation for R8

https://github.com/engteam14/documentation2/pull/82#issue-1204388497

The previous team listed the mitigation for R8(R9) (Rendering during movement may stutter/lag/flicker) as 'Cry in a pillow, curse the gods, switch code to Unity' which the team decided was unsuitable for a formal risk assesment and replaced with 'Consistent manual testing to spot graphical glitches'. We felt this is more appropriate for not only the writing style in a formal report but also an actual mitigation that can be reasonably carried out.

838cb5b: Add merging issue risk

https://github.com/engteam14/documentation2/pull/82#issue-1204388497

added risk R19(R15)

ID	Type	Description	Consequences	Monitoring	Likelihood	Severity	Mitigatic
R19	People	Merging issues causing delays	If people make mistakes causing major bugs in the product, the project may be delayed	Consistent risk	M	M	Use continuo integratic to test the product between pull requests so issues can be spotted quickly

this is because our team are not that experienced with using github and merging pull requests so there is more of a risk of errors happening. Also the mitigation of continuous integration is specific to assessment 2.

5535a7: Update Introduction + github outage

In this commit we added some information on how we agreed we would create the risk assesment using the steps from Ian Sommervilles book. Despite not being the original team who created the original risks, we still followed these steps when adding new risks to the risk assesment and we decided that it is important to detail how you discuss and analyse the risks to show that the process is thorough and involves more than just thinking of them on the spot.

We also added some information on what it means to have a risk owner and why it is useful.

Furthermore, in this commit we added some information on times that github has gone down in the previous weeks as we were monitoring this risk and discussing it in our discord channel and felt it is important to show that the risk was monitored and provide proof of github going down in the form of a statement from GitHub themselves.

7e721bb & cb173e4 : Update Risk Assesment (Additional requirements)

https://github.com/engteam14/documentation2/pull/94#issue-1220341590

In this commit we added risks R20, R21 and R18 (R16-18), this is due to the fact that we missed out the conflicting opinions and misunderstanding of requirements in our original risk assesment commit and realised that actually these are very much things that happen in software engineering projects. We also decided to add a risk into the register about change management as this is actually an important part of assesment 2.