Change report:

a) Formal Approach to Change Management

Documentation

To keep store and manage the changes of our documentation, we decided to use GitHub. Upon starting the project, we read through the brief and added an issue to our project on GitHub 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 such 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.

Below we have gone through each significant change we made to the documentation, listing the main commits showing this change so that the differences can be viewed.

Requirements

Requirements documentation from Assessment 1: https://engteam14.github.io/website2/pdfs/Req1.pdf

8948b74: Removed Unnecessary requirements

view commit

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 estimate of the project completion change significantly.

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

View Commit

Updated the requirements page to include the new requirements that we received in the brief. This is to ensure that the project covers all requirements as the requirement register can be referred back to regularly.

E1a2386 & 396806d: Changes to priority for requirements

View Commit

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
View Commit

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' and occasionally 'dosh'. 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

View Commit

A further inconsistency we noted was the lack of clarity between 'points' and 'XP'. 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

View Commit

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 week long. Due to the fact that in this assesment we have a lot of other projects happening, this felt the most suitable way of ensuring the plan is more fluid and can easily be changed so as to move tasks to later sprints if we fall behind.

bbe4ad3: Added another key point of discussion for weekly meetings

View Commit

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. This was important as it would have potentially made the team focus too much on the implementation side of things when the documentation is equally, if not more, important.

e9c1a4e: Changed Communication and Collaboration Tools

View Commit

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 so that someone can add them to the issues.

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 5on help us when writing our change report.

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

View Commit

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

View Commit

We added a small paragraph where we discussed the testing tool we used (gdx-testing) and why we used it as this was only relevant to assessment 2.

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

View Commit

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)

View Commit

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

Risk Assessment Documentation from Assessment 1: https://engteam14.github.io/website2/pdfs/Risk1.pdf

Note that in this report, unfortunately the risk ID's were mis numbered and this was only realised near the end of the project (see commit d722ff5 and d469cf and a10ea64). 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. As a result of this, the links to risks in the requirements register had to be changed here

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

View Commit

ID	Type	Description	Consequences	Monitoring	Likelihood	Severity	Mitigati
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 interacti

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(R11) 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(R12) 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(R13) 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

View Commit

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

View Commit

added risk R19(R14)

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

View Commit

In this commit we added some information on how we agreed we would create the risk assessment 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 assessment 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 Assessment (Additional requirements)

View Commit

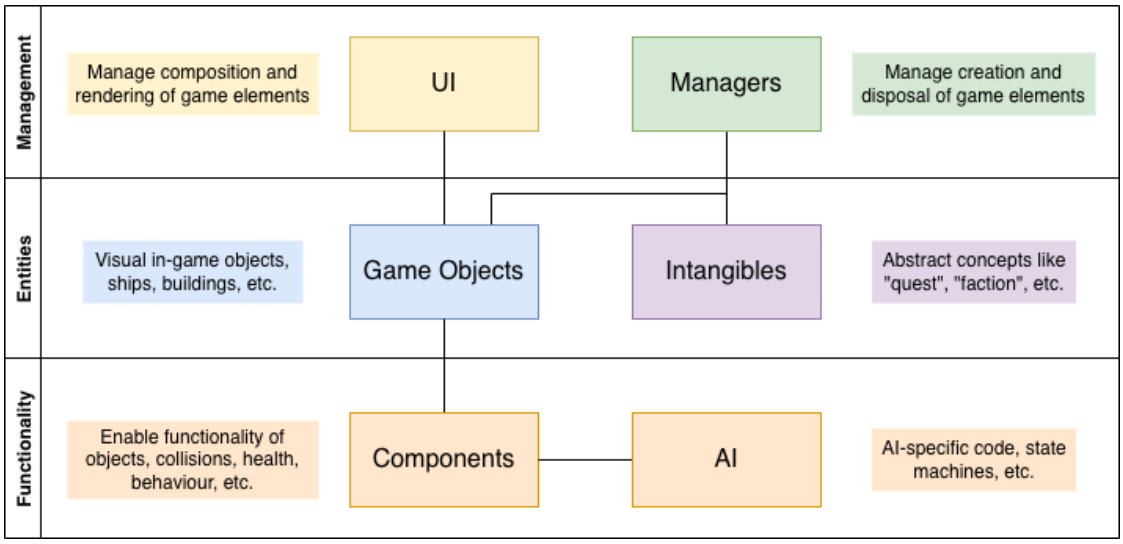
In this commit we added risks R20, R21 and R18 (R15-17), this is due to the fact that we missed out the conflicting opinions and misunderstanding of requirements in our original risk assessment commit and realised that actually these are very much things that happen in software engineering projects. We also decided to add a risk into the register about change management as this is actually an important part of assessment 2.

Architecture

db9690c: Change the Abstract Architecture

View Commit

The previous teams abstract architecture looked like so:



We decided to replace it with a more detailed diagram that shows some key classes and their dependencies while maintaining the clear concepts of the entity-component relationship as well as the relationships

between the Managers and the entities and intangibles and between the UI and entities and AI and components. This is because it gives a clearer idea of the main concepts behind the concrete architecture. It also allowed us to plan out some of the new features required for assessment 2 such as PowerUps and Obstacles

1507ee5: Update the Concrete Architecture

View Commit

The original architecture can be viewed here

Due to the fact that we had to add new classes for the new requirements in the second assessment, the concrete architecture had to be added to. The first change we made was to the the amount of methods and attributes in each class, which has been cut down dramatically. This is due to the space constraints and an attempt at making the diagrams more easily understandable. We kept only the functions and attributes that are the most paramount to the functionality of the class do that it is still clear what the class does but there aren't any getters and setters or functions that simply call on another function. We also removed the `Test` class as it currently isn't used in the codebase so we felt that going forward we can just use the built in functionality of the graphics classes of LibGdx to handle rendering text and fonts.
`Attributes` was also removed as it is actually a private class within the `AINavigation` component so isn't relevant to the wider architecture of the game. Lastly, `NodeHeuristic` class was removed as it isnt currently in use in the code base as AI wasnt a requirement for the assessment. The class was eventually removed from the codebase.

Furthermore we added:

- `SaveManager` into the managers image. This class is for handling the saving and resuming of the game which is a new requirement for Assessment 2 (UR_GAME_SAVE, FR_SAVE_GAME_STATE, FR_LOAD_GAME) and was added to keep this functionality separate for organisation purposes
- `PowerUpAssigned` component which the player has, shown by the new line between them, was added for this assessment too. This is to hold the `powerUp` that the player has, to fulfil FR_POWER_UP. `PowerUp` was also added into the Miscellanious diagram for this assessment as each instance of this would be a new power up.
- The other component that was added is `ObstacleControl` which is used to add damage causing capabilities to the new `Obstacle` entity. We also had the links between `Renderable`, `RigidBody` and `Transform` and the new `Obstacle` class as it makes sense for the obstacles to have these components to allow them to be rendered, register collisions and be placed on the map/move.
- As well as `Obstacle` being added, another class called `Weather` was added to the diagram which inherits `Obstacle` as the weather would have similar functionalities to the obstacles except that the weather moves around the map.
- `PowerUpPickUp` was also added as it is the class for the entity that the player can pick up to give themselves a boost in the game. This is new for assessment 2 - FR_POWER_UP and UR_POWER_UP
- Furthermore, we added the `Building` class into the diagram. This class was already in the code base when we took over the project, but was not detailed in the Concrete architecture. We felt that the colleges having a set of buildings and the `Building` entities being the one to have the `Renderable` and `RigidBody` components meant that it was logical they should be shown on the concrete architecture because otherwise colleges may be mistaken as not having an image to represent them on the screen.

- We also added that the `College` class has the `Pirate` to allow the colleges to be able to shoot as this wasn't previously implemented despite being in the requirements for assessment 2.
- `com.badlogic.gdx.ScreenAdapter` was added because this was an important detail that we misunderstood when taking over the game. The `ScreenAdapter` concept is specific to LibGdx so would only be something you would already know if you had past experience with LibGdx. Along with adding this, we added the `PauseScreen` to help us add the UI functionality to allow the game to pause, restart and be saved.
- Lastly we added the main game class `PirateGame`. This is because it is actually the main class of the game and so is one of the most important.

d267759 & 6d08e8: Replace the justification for architecture

View Commit

View Commit

The previous team had a write up consisting of a page of justification for the abstract and concrete architecture as a whole, and then a page of going through some of the requirements and stating where they were covered by the architecture. unfortunately it is not clear why they chose to only mention the requirements that they did as they didn't cover all of them. Due to this, we felt that the write up didn't cover the "Systematic justification" requirement in the brief.
Instead we replaced this section with a brief explanation on the detail levels of both the abstract and concrete architecture followed by a systematic explanation of each key feature in the game, stating how it is shown in the concrete architecture, some explanation as to the function of the classes, and how it was represented in the abstract architecture. We ensured we mentioned every class that was included in the concrete architecture and their related requirements.