Method selection and planning

Our team is employing the Agile Development methodology for our project. It is a flexible and iterative approach that allows us to incrementally work on our project in sprints and regroup to reflect on our work and make changes rapidly if needed. It has been reported that nearly 65% of software design projects use some Agile or iterative component, with projects that adopt these techniques obtaining higher rates of success [1, p.9]. These figures were attained by comparing Agile methodologies with traditional methods, which focused heavily on extensive planning [1, p.4]. The team deemed this approach unsuitable due to the timescale of our project and the necessity to produce functional prototypes at required intervals. The main features we are using in our development process are:

- **Continuous Collaboration:** Scheduling bi-weekly team meetings ensures that the whole team is aware of their responsibilities and provides the opportunity for members to voice concerns or seek advice from peers on a regular basis. Moreover, it allows the team to identify key milestones to achieve for the following week whilst sharing current progress.
- **Iterative Development:** The flexibility afforded by sprints within an Agile methodology allows the project to be broken down into smaller functional components that are easier to develop and manage, before combining them at the end of development. This approach is complemented by the use of version control systems like git and GitHub that allow individuals to work on separate components of the project simultaneously without the overhead of making drastic changes to a single file. By iteratively making small non-critical changes to software through GitHub, it minimises debugging requirements and lends itself to more efficient software development.
- **Adaptability:** The agile methodology encourages changes in requirements even during later stages of the development cycle. It would be naive to assume that issues and mistakes from individuals or the collective won't arise during development, however the methodology employed focuses on how we respond to these issues. The suitability of this methodology for our project is primarily categorised by the frequency we can communicate with each other, thus negating the effects that mistakes can cause and instead allowing us to refocus and adapt to these changes.
- **Customer-Centric:** Due to the ability to easily change our requirements, we are able to more easily accommodate the specifics of what our client needs. We can comfortably keep in contact and be able to deliver a product as close to their requirements as possible. Furthermore, by utilising our iterative design approach effectively, we will be able to produce functioning prototypes that can be tested by the client whilst verifying that these iterations conform to the requirements outlined in the product specification.

We have used GitHub and Google Drive as the main tools to support our projects development and collaboration needs. The primary benefits and suitabilities of these systems are discussed in detail below.

GitHub serves as our version control system, it plays an important role in our Agile methodology, namely an iterative design approach. Some of the key features that make it a good fit for our project are:

- **Version control:** GitHub provides an intuitive interface for version control, allowing team members to observe multiple development branches.It also maintains a log of all code changes across the duration of the project, so in the event our main branch needs to be reverted to an earlier iteration, GitHub has measures in place to achieve this.
- **Issue Tracking:** GitHub's issue tracking system will allow us to manage bugs, tasks and any new features that may prove necessary in latter stages of development. This feature also allows members to be assigned to certain issues, providing clarity for team members regarding their responsibilities within the project. In addition, individuals can request help for certain issues, improving the collaboration between members and prompting a more thorough debugging process.
- **Collaborative Documentation:** Github's features such as the README enables the team to document the project's technical details alongside its processes and architecture in one centralised location. This is ideal for us to pass our project onto another team as it will help them to understand the base of our code and to be able to continue it in assessment two without issue.
- **Pull requests and Code Reviews:** Using GitHubs branching system, members are able to create changes to the source code on their local machine without affecting the main workflow. Once an individual is satisfied that their changes are functional, they can create a pull request and submit them for approval. Our project will operate on a 2-review basis, meaning any pull requests must have at least 2 approving reviews before they can be merged to main. These measures make GitHub an extremely suitable source control system for our project as it provides an appropriate level of protection for our main branch.
- **CI/CD**: We use GitHub Actions as our CI/CD tool, to automatically integrate different modules of our game, compile and test them in a cloud environment.

Github aligns perfectly with our use of the Agile methodology. It allows for us to make use of iterative development with its branches and merging function, which supports frequent updates after sprints. It also facilitates code reviews and feedback which are essential for a small team with varying experience in software development.

An alternative for for GitHub is **GitLab,** which offers similar features to GitHub but has a more seamless CI/CD integration as it fulfils all of the fundamentals of CI/CD in one environment [2].It also includes some features useful when using an Agile Methodology such as time tracking, which allows an individual to see how long they have spent on a certain task which could assist with project alignment and meeting deadlines. The reasoning behind adopting GitHub as our version control system is due to the extensive documentation available and the fact that some team members have prior experience with this system. By utilising a system that is familiar, we aim to massively reduce the setback of learning new software and can instead devote efforts to the actual implementation of our project.

Google drive acts as a shared repository for all of our project-related documents, it provides a host of features that will benefit our projects lifecycle:

- **Project Planning Documents**: Our project planning documents such as meeting minutes from sprints and our requirements elicitation will be readily accessible for all members of the team to utilise. This ensures all members are familiar with expectations for the project and also allows documentation to be revised remotely by multiple members.
- **Collaborative Editing:** Google Drive allows real-time collaboration as multiple people can be working on the same document at the same time, this is useful in the development process as contributions can be reviewed by different team members, similar to that of GitHub. A key feature of Google Drive is the ability to comment on passages of text to leave feedback, aiding remote collaboration outside of timetabled meetings. Additionally, Google Drive keeps a log of all document contributions and changes which supports the dynamic nature of Agile methodologies, as team members can query each other directly to modify textual elements without the confusion and delay of determining who has contributed and where.
- **Non-Text based assets:** Google Drive can be easily used to store parts of our project that are graphical assets, including product designs, UI mockups and UML diagrams. This gives team members the opportunity to review UML remotely, thus providing another level of security and approval for the deliverables related to our project.

Google drive complements our workflow by providing a central location for all of our team's documentation. Its real-time editing capabilities help them to work together effectively. This allows quick access to important information to all team members which is of utmost importance during the short sprints completed on a weekly basis.

Confluence is a good alternative to google drive for document storage, it offers integration for systems like Jira which can be used for task management and tracking, whilst also providing document storage functionality. The reasoning behind the selection of Google Drive mirrors the reasoning of selecting GitHub, as all members are familiar with its functions and therefore can approach the documentation deliverables with a high degree of confidence.

Our team adopted a structured approach to our organisation, focused on voluntary role assignment, regular communication, risk mitigation and client goals. Key behaviours and decisions to achieve this are listed below:

- **Role Assignment:** Each team member was assigned roles on a voluntary basis to support their strengths and confidence with different tasks. The clear definitions provided in our initial role allocations give the team structure and establish a collective understanding of what is required in the project and from who. For a larger project with more tasks and intricacies, having more flexible roles may be necessary, but for a small project with a short turnaround time having specific, structured roles will improve the workload completed to achieve iterative functional prototypes and strong documentation.
- **Backup for Each Role:** Alongside our assigned roles, we allocated understudies to each role to avoid any major setbacks in the face of illness, as outlined in our risk assessment. This will ensure that no role is unfulfilled and will assist with our review system for documentation, as multiple team members will be familiar with each component of the project.

- **Regular Meetings:** To supplement our agile development methodology we chose to have two meetings a week, whilst using applications such as Discord and WhatsApp to communicate outside of timetabled meetings. This allows us to assess the current workloads of each member, address any problems we encounter and monitor the overall progress. These meetings will also allow us to learn from each other and utilise alternative views to construct a more holistic and representative final deliverable. It is imperative that all voices are heard and considered fairly and it will be these conflicting discussions that will ultimately determine the success and direction of our project. The more opportunity we have to communicate, the more coherent and structured our project will be.
- **Client Meetings:** We have had a meeting with our client to ensure that our ideas aligned with their expectations and goals. The requirements elicited during this meeting will act as our main point of reference for any changes and features within the project.
- **Meeting Diary:** Our sprints have been recorded in a meeting diary, highlighting developments between sprints and what is expected of each member by the following sprint. This record will continue to provide a timeline of our project and will be referenced throughout our project's lifecycle to assess our progress in relation to our initial Gantt chart outlining the expected timeline of our work.

The Gantt chart used to plan our project visually organises tasks, deadlines and dependencies over the five-week project timeline. This allows us to see all of the key phases: design, implementation and testing, laid out with a clear start and end date for every task, as well as showing us key dependencies in our project. An example of this is us needing to understand our implementation techniques before starting to code the project. This also helped us to allocate resources effectively, set milestones and keep track of our progress. Enabling everyone to stay updated on current and upcoming tasks, as well as any shifts in the timeline due to evolving project needs or unforeseen challenges. An image of the chart can be found **here.**

We also used a meeting table to keep track of tasks by documenting current tasks, upcoming tasks that are needed to be completed. This worked alongside our agile design methodology to make sure that each week everyone knew their priorities and understands the next steps for the project timeline. We broke these tasks down into these three columns so we could visualise our current progress as well as preparing for future deliverables, allowing us to adjust workloads if needed or reassign tasks if someone requires support or priorities shift. Helping us keep track of our project's milestones. Here is a link to our **weekly meeting document.**

As shown in our Gantt chart our project is composed of two main parts, this being the design phase and implementation phase. In order to manage the adaptive demands of an Agile project, it was imperative that we outlined a comprehensive set of risks and elicited descriptive requirements prior to starting development. This project composition was agreed upon unanimously by the team prior to the creation of the Gantt chart and will complement our choice of agile methodology, as the research and and outlining of key dependencies in the project act as an anchor for reference even if other parts of the project adapt over time.

Reflection:

As we approach the deadline for the project, we felt it would be appropriate to consider the ways in which our plan changed over the course of development.

Our choice of Agile methodology proved itself invaluable and presented the team with all the benefits we had desired when starting the project. The decision to hold bi-weekly sprints gave us the opportunity to learn from one another and develop skills in an unfamiliar environment. During our implementation phase of design, we encountered a number of bugs and issues when merging branches into main using GitHub, however the frequency with which we were meeting provided us with the opportunity to cross reference each other's work and debug the repository as a collective. Were we to have chosen a more traditional method with less contact hours [1, p.4], these issues would have certainly been harder to manage. Our iterative approach contained in this methodology was hugely beneficial as we were able to deliver consecutive working prototypes which used a bottom up approach, first creating the UI and map screens before layering the settings screen, title screen and building functionalities further into the development lifecycle. Again, were we to have selected an alternative methodology such as Waterfall, the rigidity would not have allowed us to adapt to changing requirements, resulting in a final product that might not have met the requirements.

Our roles throughout the project remained static which we attribute to the voluntary nature of our role allocation. By allowing team members to utilise their strength in a meaningful way, we experienced positive morale and interaction throughout the duration of the project. The understudy system adopted by the group to mitigate the effects of illness worked seamlessly, with allocated members taking responsibility for the workload in these events.

Our choice of GitHub for version control and Google Docs for housing documentation worked successfully to achieve the requirements for the project. GitHub's code review system proved its main benefit as team members utilised this feature to provide constructive feedback to each other and ensure any changes made to the main branch were validated by at least two other members. Furthermore, we were able to set up GitHub actions to automatically review our codestyle to ensure we conformed to the Google style guidelines - a task that would have been incredibly time consuming were it done retrospectively at the end of our implementation.

Overall, we found the requirements for the project to be manageable and achievable, largely due to the small scale and non-critical nature of the assets we were working with. There was limited variance in requirements over the duration of the project as we had documented them descriptively prior to starting implementation. As a team we attribute the majority of our successes within the project to our choice of Agile methodology and will look to continue the subsequent stage of work using many of the techniques we adopted for our first deliverable.

References

[1] Pedro Serrador, Jeffrey K. Pinto, *“Does Agile work? — A quantitative analysis of agile project success”*, International Journal of Project Management, March 2015.

[2] *“What is CI/CD”,* Gitlab.com, https://about.gitlab.com/topics/ci-cd/#why-git-lab-ci-cd (accessed Nov 7th, 2024)