# 4. Method Selection and Planning

Team 10 | YorKorsairs

Abdullah
Tom Burniston
Omer Gelli
Adam Kirby-Stewart
Sam Mabbott
Benjamin Stevenson

## A. Software Engineering Methods

Our chosen software engineering method is a merger between plan-based and agile, with a Gantt chart that also acts as project guide. This seemed most appropriate as we anticipated requirement adjustments to manage furthermore, this plan management system will allow the team to create a timescale and focus on project completion.

Additionally to this the implementation, working with a predominantly spiral lifecycle with elements of the waterfall model will minimise risks such as going beyond time limits. Superficially, the linear waterfall style will be beneficial as it sets a rigid and easy structure however the spiral allows for new iterations to be introduced resultantly our project will cope with the requirements changing as elicited in customer meetings etc. and improvements the team may make.

Moreover, XP programming in an agile method will produce the game portion of the project with requirements met and all aspects of the code being checked and debugged effectively and within a reasonable time. Additionally the XP style is suited to our team as the development is conducted in a small collaborative group and appropriate tests can be carried out.

We organised frequent team meetings, on average twice a week. These meetings served two key purposes:

- Providing a platform to discuss and agree upon key development decisions, most often but not limited to game design and other general decisions affecting the entire project.

- Allowing each team member to communicate their progress on their assigned tasks, and for new tasks to be assigned when necessary.

Preallocating tasks to particular people, means we are planning ahead and more than that, it allows each individual to pace themselves towards a set date yet our plan does allow for flexibility. Utilising each member's skill set whilst keeping the workload even is important and by creating our plan this way equality can be achieved, much more easily than could occur. The frequent team meetings in which we discussed task progression, allowed us to keep each other accountable for our productivity and with open and honest discussions the project will be kept on track and produce the best outcome.

## Tools Used

### Communication and Collaboration

- For our team meetings we used Zoom. This was an intuitive choice as it is a platform that we all have lots of experience with and therefore avoided an unnecessary learning curve.

- We created a Discord server for general communications throughout the project. This was crucial in allowing team members to ask clarifying questions

without having to wait for the next meeting, avoiding any unnecessary obstructions to task progression.

- Google Drives allows synchronous, asynchronous, distributed and in-person meetings thus providing the ultimate flexibility and the opportunity for producing the best project possible.

- We created an Excel spreadsheet in order to organise and keep track of our tasks throughout the project in the form of a Gantt chart. Alongside the team meetings, this was crucial in keeping us organised by providing a visual representation of tasks that were completed, in progress and yet to begin.

- GitHub is being employed for the joint implementation of the project. The java project will collaboratively and constantly be available to team members. So GitHub was an obvious and effective choice in this regard. This is extremely beneficial as the use of this collaborative tool means we can keep different parts of the project separate but still use the same system meaning they are well organised, effectively managed, easily accessible and navigable by all team members.

**Website**

- We used GitHub pages to develop our website. This meant that our resulting site would be simple to implement and refactor for our game and assessment deliverables.

**Architecture**

- Draw.io was used in order to create the abstract architecture diagram, as this relatively easy to use tool seemed more appropriate for this simpler diagram

- PlantUML in addition to adobe photoshop was used in order to create the concrete architecture diagram;

    - PlantUML was used at first in order to create representations of classes, categories of classes and the relationships within each category

    - Adobe photoshop was later used in order to add the inter-category relationships

**Implementation**

- We chose to use IntelliJ as our IDE. This was due to both the ease of use offered by the tool, along with the fact that team members had previous experience with the tool: it felt like the ideal choice to avoid unnecessary and time-consuming learning curves.
- We utilised the libGDX game development framework during the implementation of our game. Similarly, to our choice of IDE, this was largely influenced by the previous experience of the team. We were also aware that LibGDX was a popular choice amongst other teams and therefore our use of this framework may make it easier for another team to take over and expand our code.

## Alternatives considered

- We considered using other IDEs such as Eclipse for software implementation, however as mentioned previously we chose to use IntelliJ due to team members having previous experience with this IDE

- Google drive was an obvious choice as it provided an intuitive way to store our documentation in a freely accessible and collaborative way. It has 15 GB storage capacity which will be more than enough for our project (a clear winner when compared to others e.g. dropbox). Google Drive syncs across multiple devices and operating systems so team members can always collaborate no matter the device.

- In GitHub for example, version control is extremely useful. This aspect means tracking changes is easy and identifiable, which is helpful when retrospectively viewing work (meaning asynchronous collaboration doesn't cause redundancy and work can be clearly documented, removing confusion). Also, in certain circumstances, we may need to revert to a previous version, by using this software this can be done extremely easily which ultimately saves a lot of time. Other platforms were available (Bitbucket, TaraVault etc.) but GitHub is now the most widely used software by developers so, is familiar to the team furthermore, because it is a trusted cloud service provider with security the software element of our project is secure.

- GitHub Pages was compared against others (including Vercel etc.) and the comparison concluded that it would be the most suitable. Some benefits for example are that the website can be changed by team members easily, commits are made with version control, instant version rollback as well as having all other key features in similar tools allowing for effective updation and maintenance of the website.

- We also considered game engines other than libGDX in order to enable software development. Unity was considered, however we were discouraged from this choice by factors such as Unity's reputation for largely outdated/incomplete documentation and the fact that many useful features are behind a paywall.We also considered using the Unreal game engine, but quickly decided against this as it seemed inappropriate for developing what is a relatively small game and would cause the resulting game to be unnecessarily bloated.

## B. Team Organisation

The team approach to organisation is very methodical and focuses on the acceleration of output without loss of work integrity or quality. Explaining this further gives an in-depth view of the structure: first, the steady start begins, starting attentively was prudent as it lets the team figure out and gather all the information required before commencing. Because of this our team has not produced rushed or erroneous work.

As the project continues work throughput increases at pace as all members know the tasks they have been assigned and are fully informed. This method allows adequate time for the whole team to review each other's work again adding to quality control and agreement. It is also important to expand on how this method works for the members of the team as well, in this regard members can more easily discuss work, prevent disagreements and be flexible in their work process (as time can be reallocated, see Gantt charts for effectual demonstrations of this).

There will be allocated 2-hour (term-timed) weekly sessions which will make up a large proportion of review and discussion time but in addition to this, there are also Discord sessions (ranging between 30-90 minutes) to provide a supplementary period to complete and add additional content to the project.

The structure of our group means that not all members need to be present in meetings so their time can be used more productively and sessions become less cumbersome and more streamlined.

Having the right balance of people for a specific section is important and the manager/managers of that section can get work done with the non-invasive oversight and outside perspective of at least one other member which engenders a work driven atmosphere but with the security of peer review from at least one other person both during and post writing.

The hierarchy of the group is flexible and is based on a democratic and fair structure. This way of working may lead to less strong and decisive decision making however, for our team it is and will continue to be effective and due to this way of working a mix of ideas and perspectives are introduced to the project and everyone is as is far as possible pleased with outcomes that are made as a result of a whole group decision. Also, there are managers for different sections and a singular leader might not be able to make specific decisions for each part of the project so, having these managers make important decisions is crucial to the good progress of the project.

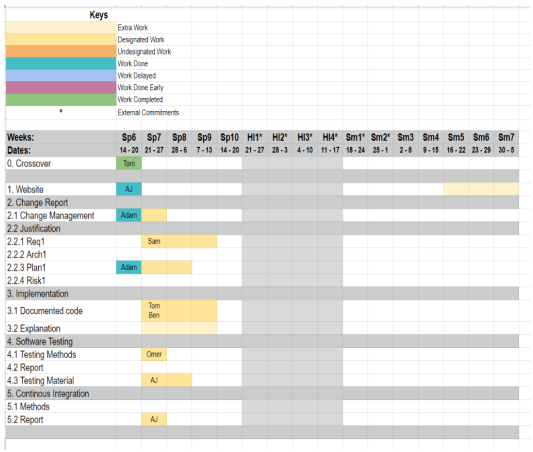
## C. Systematic Planning

---

Our first step in planning this assessment was to create a task breakdown into sections which can be monitored and have the respective team member's progress recorded.

Knowing if a project can meet objectives on time is essential so having a systematic plan is required. As a group we will draw up a plan based on what should be delivered, when it should be delivered, and who will work on the development of the project deliverables. A plan-based approach requires a stable view of the development processes to accomplish this simple but effective planning method, a combination of written records and visual spreadsheets will be created which will adhere to the Agile methodology but also include the property of visibility.

Key tasks overview (Sp-Spring, Hl-Holiday, Sm-Summer):




(E.g. Sp7 Gantt chart).

Key is as below:

Extra work - Additional work for extra improvement.
Designated work - Future work timetabled.
Undesignated work - Timetabled without designation.
Work in Progress - Work started and will continue.
Delayed Work - Due to dependency/commitments
Early Start - Indicates work started before scheduled.
Completed Work - Finished section.
External Commitments (*) - Limited time/Restrictions etc. due to external factors.

As for priority, the current plan is equality, simply to achieve the best quality all parts are required, of course, mark distribution will play a part especially if a component is running behind.

We have set an ambitious and firm timetable but this is to ensure that the project continues to proceed at speed however, this allows for unexpected and unforeseen events because it has flexibility — of course, strict and ultimate deadlines are still enforced to complete the module on time.