Method Selection and Planning

Team 24

Joe Cambridge
Joss Davis
Emily Dennison
Louise Evenden
Amber Hemsley
Josh McWilliam
Lianyu Zhao

Software Engineering Methods

The software development framework we used was an agile framework that meant that different stages to be completed simultaneously by other members in the team.

- We chose this over waterfall development framework because waterfall software development framework works by completing the stages one at a time which was impractical for this project.
- We used agile instead because this project is split up into different parts, like we had half the team focusing on implementation while the other half focused on the documentation. This meant it would not be practical to do one stage at a time as part of the group would've been not doing anything until the other half is finished with their task.
- Agile also consists of considerable documentation, which means minimal team member knowledge is lost if a member was to become unavailable for whatever reason.

The Java framework we decided to use was libGDX.

- libGDX was the most:
    - Mature
    - Widely Used
    - Extensible
- It consists of mainly regular java, which the team was already familiar with.

For our IDEs we considered a few options but ultimately we decided to use visual studio code. Visual studio code because:

- Everyone in the team used this IDE and was confident using it.
- Its live share feature was incredibly convenient for collaborating together during meetings even whilst not physically together.

Tiled was used as a map creator for libGDX as it is the most supported map editor so this was chosen due to the fact it had more tutorials and was a simpler tool then any competing product.

We then decided that we needed a version control system to upload code and enable every member of the team working on implementation to be capable of developing the code simultaneously. We decided on GitHub for these reasons:

- Easy design
- Its high market share means that the team could have experience with real world tools required in software development.
- We could sign up for free (as we are students) to GitHub premium. This allowed us to have web based support in case of any issues we encountered.
- Crucial features like being able to have protected branches and having required reviewers. This reduced our risk of:
    - R_SUS: Data loss due to malicious actors or mistakes.
- This reduces the chance of us losing changes made by others or introducing merge conflicts as now it needs two group members to review and commit these changes.
- Tracking of who and when team members create changes to their work, which allows for a simple way to track which members have contributed individual parts of the code.

As a team it was decided at the beginning of the project to create a project schedule, with tasks set for each member of the team and deadlines. In order to support this we decided to use a gantt chart software system. This helped us organise and keep track of any task progress from meeting to meeting.

We decided on using PlantUML to create the gantt chart as this was an open source gantt software that was capable of creating the gantt charts that we needed. We also used GitHub tasks to list off tasks that needed to be done and then the gantt chart was used to keep track of when tasks were completed successfully and who had completed them.

We agreed to try and have several meetings regularly, because of this we needed a way to communicate outside of practicals. After some discussion we agreed on discord.

- Discord allows for:
  - Different channels to be used for different topics and whatsapp would only allow for one group chat.
  - Sharing of desktop screens allows us to have much more productive calls with significantly more collaboration.
  - Voice calls for meetings .
  - Messaging for organising meetings and tasks both to individuals and the group as a whole.
- Overall we decided on Discord due to these reasons and since everyone already had a discord account, so the team thought it was the best option.

Using discord we were able to arrange regular meetings - towards the beginning this was around twice a week and closer to the deadline this became multiple times a week (up to everyday) - in person as much as possible, and if someone was unable to attend in person sometimes they joined on Discord call instead. During these meetings we had several things we needed to do each week:

- Discuss everyone's progress on any tasks they had been assigned.
- Adjust the plan based on whether these tasks had been completed.
- Set goals for the next meeting and assign any new tasks.

We would do most of the work together during the meetings but some of our work was done independently before or after them.

In order to complete our documentation we decided it would be best to use Google Docs for several reasons:

- Sharing documents was easier
- Live collaboration was simpler
- Our team had more experience with google docs

So whilst it is less capable than word it has significantly better collaboration capabilities. This made it fit with our engineering methods very well which ultimately made us choose this platform.

Team Organisation

Originally, we decided that it was best to have a "leader" so that there was someone ensuring that tasks were completed on time and effectively. We decided to split the team up into a documentation team and an implementation team. We did this so that both parts were being completed simultaneously, also people had different strengths such as some people being more confident with documenting and others preferring to do the programming. The people in the implementation team were Amber, Josh and Joss then the documentation team was Louise, Joe and Alan. Emily helped out where needed, she started on the documentation team but the implementation team needed help so she moved to their team, she also created the website.

Due to the fact that we had a "leader" and two groups we didn't feel the need to allocate any other specific roles meaning that everyone could easily flex into the roles like quality control, diagram creator, etc. The two subgroups communicated well as everyone would listen to each other as we had roundtables where each member would describe what they had been doing so everyone would be able to jump in and understand if needed.

We felt it was suitable to split the group into two teams so:

- More focused work can be done on individual strengths and to increase productivity.
- Each team member works on multiple sections of the documentation, rather than every section being the responsibility of one person.
- Different members would complete a base idea for what the section would look like, with other people then assisting by filling out details and ensuring accuracy.

This method also increases the bus factor, because we would then have multiple people working on each section at once so progression would be as consistent as possible. To maintain a high bus factor Emily checked over a couple of the documents so that if all of the documentation team were ill or couldn't attend then she would know about the documentation in case it needed doing or editing for any reason.

We considered different systems for organising the project, such as everyone working on the programming and then everyone working on the documentation afterwards. However, we felt this was not a good plan because some of our team members were not confident with coding and others were not confident with documenting. Also, it would mean that everything would be hard to document if not done simultaneously with the implementation. Therefore, we felt that our plan to split up and work simultaneously was a much better plan.

Systematic Plan

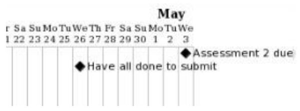
To make our initial plan we had a meeting with the entire team where we discussed what needed to be done and gave base dates for when we wanted each task to be done by. We met around twice a week at first and as the deadline got closer these meetings became more frequent. All our meetings were noted in "minutes" documents which kept track of when the meeting was, who was there and a brief overview of what was discussed during the meeting and what will ideally need to be done by the next meeting.

This was our plan for the project. This consisted of:

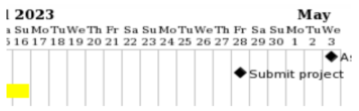
- Our tasks for completing during the practicals and outside of meetings.
- We had different parts of the chart based on the components of assessment 2:
  - Updated website
    - This keeps track of any changes to the website.
  - Change report
    - This is keeping track of any changes to any documents or the code.
  - Implementation
    - This is going to keep track of who's completing which parts of the coding.
  - Software testing report
  - Continuous integration report

Shown on this gantt chart are all the tasks that anyone completed over the time of assessment 2. It includes details about what the task was, who completed it and how long it took/when it was completed.

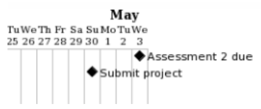
Mostly our gantt chart didn't change as we found it easy to stick to, when people choose their deadlines but we did update deadlines when needed, for example we wanted to have the project completed a week in advance but we updated this as we felt needed a couple extra days to complete the project to the best that it could be.



→



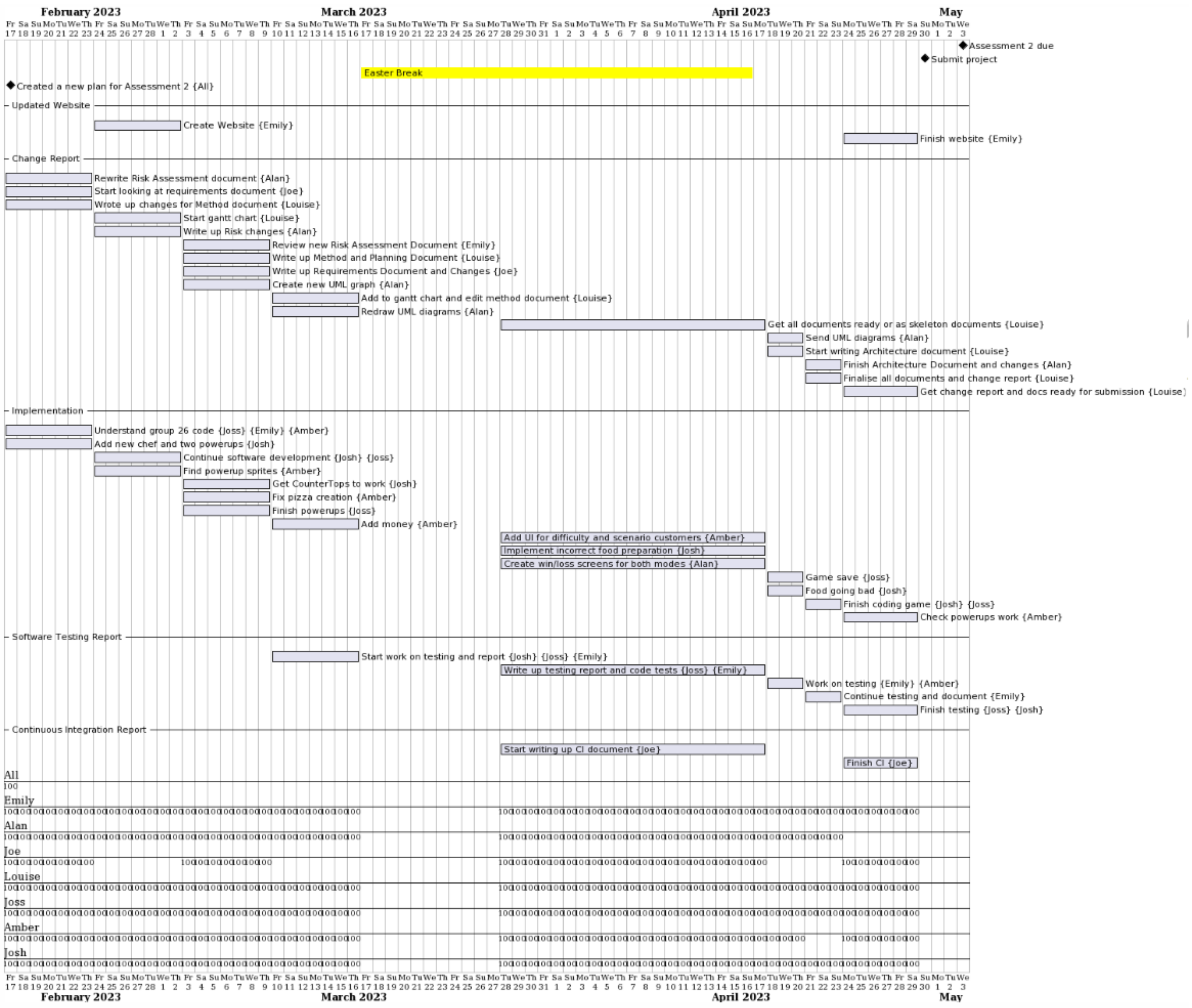
→



We didn’t give out tasks for Easter break for the first part as most people were unavailable to really work on the project properly as everyone was at home and busy that first week or so. We were unable to arrange a meeting that everyone could attend so we found a time when the highest number of people were free at once to have a meeting. During the meeting the members who attended gave out tasks and wrote them in the minutes then informed the other members of the tasks they would have to complete. This worked well as it gave people time for a small break before then continuing with the project and having tasks to complete.

Once we came back from Easter break, we all started working on the project more to complete it at the best ability and fastest time possible. We arranged extra weekly meetings to ensure that everything was getting done and make sure that nothing was forgotten or not completed over the break.

This was our final gantt chart, showing our final project deadlines and was updated regularly with tasks needed to be completed, the dates that people completed them and who was designated to finish those tasks.


February 2023
March 2023
April 2023
May
Assessment 2 due
Submit project
Easter Break
Created a new plan for Assessment 2 {All}
Updated Website
Create Website {Emily}
Finish website {Emily}
Change Report
Rewrite Risk Assessment document {Alan}
Start looking at requirements document {Joe}
Wrote up changes for Method document {Louise}
Start gantt chart {Louise}
Write up Risk changes {Alan}
Review new Risk Assessment Document {Emily}
Write up Method and Planning Document {Louise}
Write up Requirements Document and Changes {Joe}
Create new UML graph {Alan}
Add to gantt chart and edit method document {Louise}
Redraw UML diagrams {Alan}
Get all documents ready or as skeleton documents {Louise}
Send UML diagrams {Alan}
Start writing Architecture document {Louise}
Finish Architecture Document and changes {Alan}
Finalise all documents and change report {Louise}
Get change report and docs ready for submission {Louise}
Implementation
Understand group 26 code {Joss} {Emily} {Amber}
Add new chef and two powerups {Josh}
Continue software development {Josh} {Joss}
Find powerup sprites {Amber}
Get CounterTops to work {Josh}
Fix pizza creation {Amber}
Finish powerups {Joss}
Add money {Amber}
Add UI for difficulty and scenario customers {Amber}
Implement incorrect food preparation {Josh}
Create win/loss screens for both modes {Alan}
Game save {Joss}
Food going bad {Josh}
Finish coding game {Josh} {Joss}
Check powerups work {Amber}
Software Testing Report
Start work on testing and report {Josh} {Joss} {Emily}
Write up testing report and code tests {Joss} {Emily}
Work on testing {Emily} {Amber}
Continue testing and document {Emily}
Finish testing {Joss} {Josh}
Continuous Integration Report
Start writing up CI document {Joe}
Finish CI {Joe}
All
Emily
Alan
Joe
Louise
Joss
Amber
Josh