Method Selection and Planning

Group 19, “Piazza Pitstop Crew”

By: Noah Forryan, Lewis Morgan, Naufal Tun Thamanian, Tom Owen, Dan Manby, José Fernandes

a)

We decided to use the game engine LibGDX for several reasons such as it allows us to write game code, and it has a great number of features for the development of the game like graphic rendering, physics simulation and much more. It also integrates well with the chosen map editor (Tiled Map Editor) which makes incorporation of external resources into the game easier and faster. It's known for its performance optimisation that allows the game to run smoothly and efficiently. It's an excellent choice for developing Java-based games. We consider a few other game engines, LWJGL and Slick2D. LWJGL has good support and an active community, but a very steep learning curve so was more suitable for more experienced programmers. LibGDX also has these advantages and more, so we chose that instead. Slick2D had the benefit of simplicity and a focused 2D development experience, but its development has not been as active as LibGDX and therefore we decided to choose LibGDX.

For easy communication, our team decided to use Discord given that it has a comprehensive set of features that helped maintain organisation within the team and facilitated communication amongst the team members. We also use WhatsApp due to its convenience and ease of communication.

We decided to use a Gantt chart and a Kanban board together. The Kanban board is very flexible and allows us to make any necessary changes in requirements and priorities quickly. The 'In Review' section of the Kanban board is a crucial part of the workflow because it allows us to make improvements and gather feedback on the completed work. This ensures that the completed work meets the quality standards of all the team members. Only then can the task be moved to the completed column. We used the Gantt chart to visualise the project timeline and task dependencies. This was useful to see the tasks' start and finish dates. We used both to have the flexibility and the important review section of the Kanban board and the clear timeline of the Gantt chart.

The project is organised into a Kanban workflow, facilitating visual management of tasks across different stages of development. These stages are "To Do", "In Progress", "In Review", "Completed" and a "Backlog" for future tasks. This system guarantees a flexible approach to task management and task prioritisation.

In the “To Do” section, we organise the tasks that haven’t been started, assigning each team member a different task or in some cases two members are assigned to one singular task to accelerate the completion of the task and ensure that if one was unable to complete the task, the other could.

In the “In Progress” section, we have the tasks that have been initialised by the team members but still need to be completed. These are the tasks that are currently being implemented.

In the “In Review” section there are tasks that have been completed but are awaiting a review by another team member. If any team member has any feedback to give regarding tasks done by other team members, this can be brought up in one of the in-person meetings or even on Discord/WhatsApp.

The “Backlog” contains tasks that are not yet scheduled for immediate development. This includes tasks that may arise from feedback or additional features considered for future releases.

The project is progressing through various critical phases, from initial planning and design to implementation and review.

This workflow illustrates a well-organised approach, allowing the team to systematically progress through development stages while maintaining flexibility to adjust priorities and address feedback.

For the design of the map, we decided to use 'Tiled Map Editor' because it offers several advantages over other popular map editors that we looked into. It's easy use and comprehension which allows developers of all skill levels to create maps rapidly and efficiently, it also has a wide range of customization options permitting us to create the map following our specific requirements. It supports multiple layers for organising all the map elements (background layers, object layers, tile layers, etc) which allows us to enhance the details of the map. Finally, it integrates with our chosen game engine (LibGDX) without any need for additional conversion.

b)

To facilitate good team organisation we arranged weekly meetings on top of the required practicals we attended. Within these meetings there were scrums where each team member stood up and spoke about their work since the last meeting and what they are currently working on. We took meeting notes detailing our progress so we wouldn't forget the work that each group member was doing. For the team, this was beneficial as it held all members accountable for their workload, so if someone was underperforming the group would find out and be able to rectify the situation which has certainly increased the group's effort. This was appropriate since we had a large project with a tight deadline, so we needed continuous effort from all members.

When we wrote the risk assessment at the beginning of the project, we decided to always have a bus factor of greater than one. This meant that if there were any circumstances where members of the group couldn't complete their tasks, there would be another group member to take over their workload. We split this into the development, risks or requirements, and assets. With six members, there were two people for each of these sections. This benefited the team as we were able to collaboratively work on things which made the quality of the work better, and the project as a whole since it ran smoothly with less risk.

After writing an initial risk assessment as a group, at every meeting, twice a week, we reviewed the risk assessment and updated any changes. We did this for each meeting with the whole group which allowed everyone to contribute, allowing all the potential hazards to be discussed. This made us more organised because the whole group was aware of these risks.

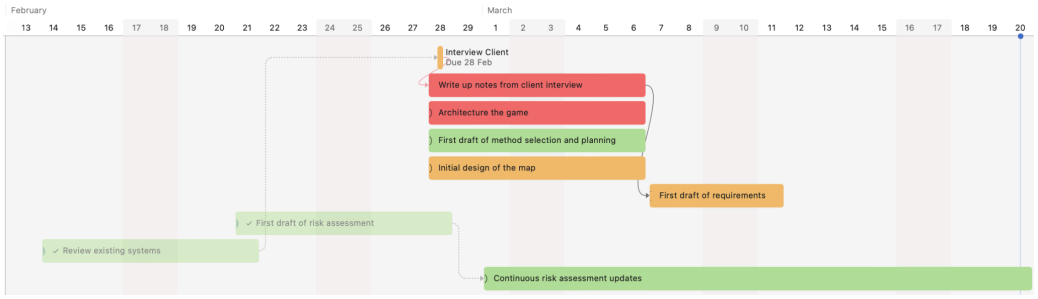
Using Agile Development, in both the development and write-up stages, we were able to be flexible. In the write-ups, we went through iterative stages of writing, then a review from another team member, and a retrospective discussion with comments about the section with both of the involved members. These stages are optionally repeated, until the task is completed.

In development, we used a similar approach where each team member who was either coding or designing the map would write a bit of code and then show it to another member of the group to review it, this member would then give feedback and say if anything needed improvement. Similarly to the write-ups, these stages could optionally be repeated, until the task was completed.

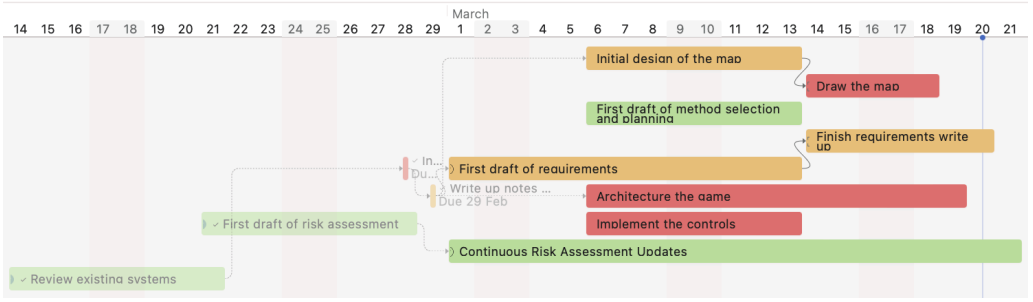
In conclusion, during our team's weekly meetings, we discussed our progress and what we were working on, which helped us stay on track with everything that needed to be done for the project and allowed us to collaborate more effectively. By reviewing what we did, we were able to correct and improve our work. The flexibility of agile development and our organised approach to meetings ensured that our project ran smoothly and without any major issues.

c)

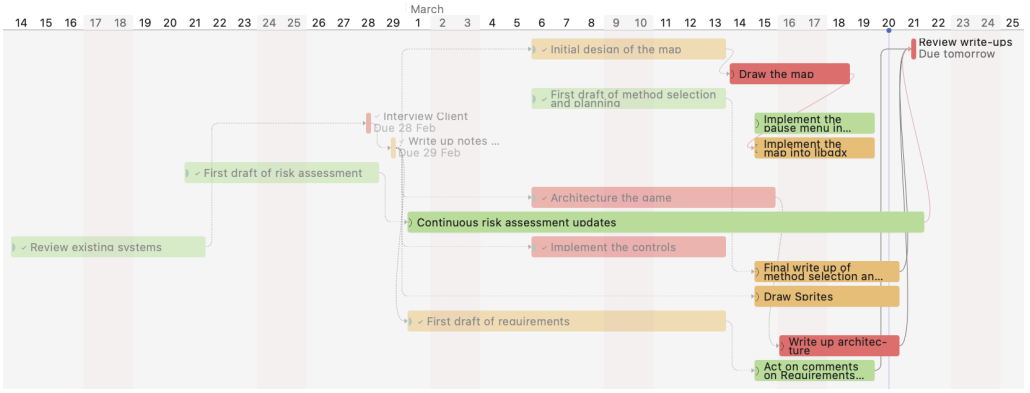
28th February



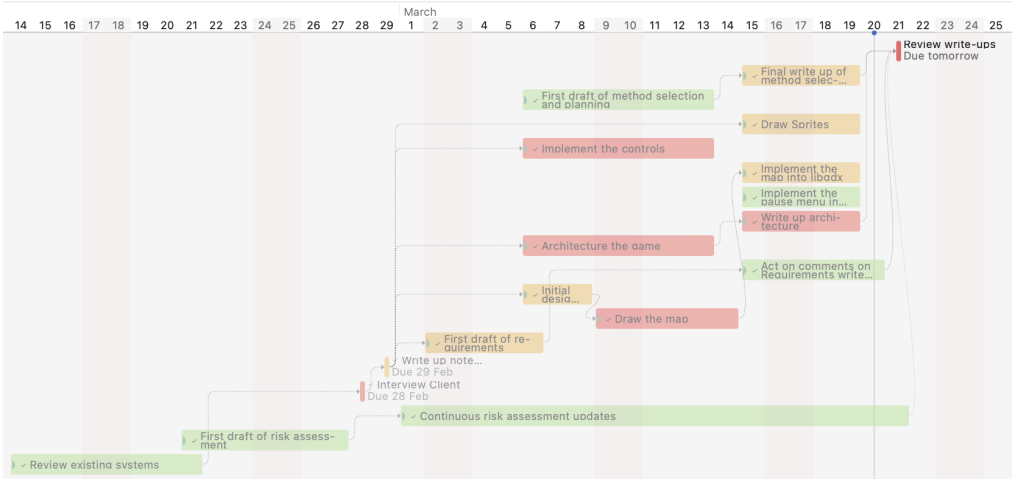
6th March



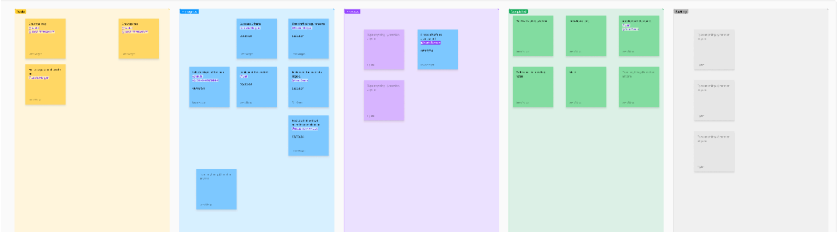
15th March



19th March

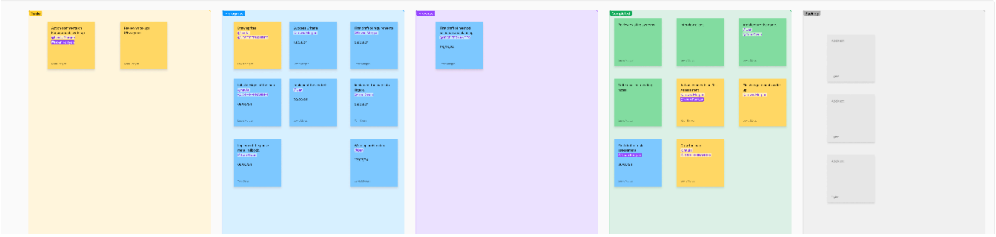


6th March Kanban



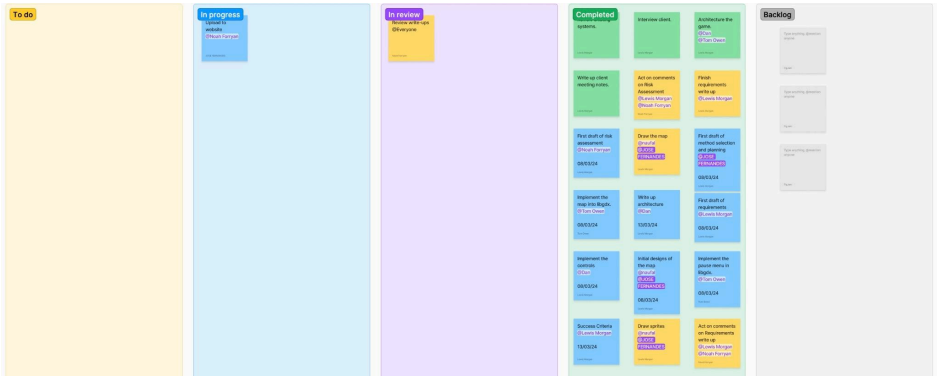
https://ibb.co/VBDv1ym (for a full view of the image click this link)

15th March Kanban



https://ibb.co/SXzqbHL (for a full view of the image click this link)

19th March Kanban



https://ibb.co/gtG3ngn (for a full view of the image click this link)

Throughout the project, we updated the Kanban board and Gantt chart during our meetings to show the work we have been doing or will do. This meant we had clear goals and knew which areas each team member should focus on, allowing us to evenly distribute the workload between members throughout the project.
There were a few reasons for the Gantt chart, and why the project schedule changed. Firstly, we initially overestimated how much work the whole project would take. As you can see in the earlier Gantt charts, there were overly optimistic plans for the completion of the project. For example, the first draft of the requirements' time frame changed, as one member realised that a different approach was needed, and needed to start again. Secondly, we realised we needed to do in-depth reviews of each other's work, increasing the quality of the work. These delays were not impactful as we had already given ourselves an extra day before the deadline, giving us some extra time in case of issues.