Bermuda Digital Entertainment (Team 7)

Method Selection and Planning

CJ Donoghue, Joseph Sisson, Sebastian Sobczyk, David Ademola, Prajwal Binnamangala, Aymeric Goransson

a) Outline and justification of the team's software engineering methods

- We took an agile programming approach using SCRUM with one-week sprints, identifying deliverables and goals in a weekly stand-up meeting and tracking progress with a Gantt chart.
- At the start of the project, we waited until we had a discussion with our client before we then started to determine our approach and distribution of work.
    - This is because we wanted to have a good picture of what we needed to do at the start instead of jumping in blind.
    - Therefore, most of the coding and documentation was completed after the Christmas break.
- Using a SCRUM agile approach was chosen over alternatives such as a plan-driven or soft approach. This is because:
    - We have clear objectives for the project already set out and can further define requirements with the client.
    - We needed to prioritise creating working code as soon as possible over wasting too much time pre-planning.
- During our weekly stand-up meeting, we reviewed the previous sprint and planned out the next one:
    - The discussion leader goes over each open task, and the team member responsible for each task shows their work.
    - The team discusses whether the task is complete, whether any possible changes may be required, and what next steps lead on from this.
    - The Gantt chart is updated to reflect what tasks were completed and what new ones should be added in for the next sprint.
    - These weekly meetings were held online on Wednesday afternoons, in addition to the in-person practical sessions on Fridays where demonstrations and discussions that were better facilitated face to face occurred. The plans for Friday were discussed in advance on Wednesday.
- We decided to delegate the implementation to a small group (Joseph and Sebastian) to prevent difficulties due to varying skill levels and understanding, and to avoid merge conflicts from too many features trying to be added at the same time.
    - However, the rest of the team was kept up to date on the state of the code during the weekly meetings, and the structure of the code was agreed with everyone.
    - We ensured everyone had equal input and understanding of how the implementation worked, so we could effectively write the documentation.

Tools used to support the project or the team working

- GitHub
    - All of the team have access to a GitHub repository where we can all access the files for the game and the website.
    - It allows for the whole team to access a central, updated copy of the files and have a clear record of changes being made to the code.
    - Many of us had prior experience with Git, and we decided it was simpler than alternatives such as SVN.
- Visual Studio Code
    - All of us had experience with this IDE and it was simple to set up for Java.
- LibGDX

    - Our research showed this is a popular game development library that was used in famous games such as Minecraft.
    - Due to its prevalence, it is easy to find resources online such as tutorials and UI skins, and it has a relatively low entry barrier, so we chose it over alternatives such as LITI Engine and LWJGL.
    - Using pure Java would be messy as the basic graphics class needs to redraw the whole screen when something changes. Using a game engine would be more restrictive on our coding style.
    - It had clear documentation which we then referenced during the development process.
- Google Drive
    - Stores all documents, notes, and files on the Cloud, ensuring we all see the same version, have record of edit history and can collaborate on the same document at the same time.
    - Also mitigates the issue of losing files due to hardware damage.
    - Every member already has a university-issued Google account so there were no setup overheads.
    - This was especially helpful for developing questions to ask our client, reviewing the documentation each member wrote and the Gantt chart.
- Google Calendar
    - A shared calendar was made so all members would be notified about when team meetings are and receive reminder notifications 10 minutes before.
- Discord
    - A Discord server was made for the team, with:
        - Text channels corresponding to different areas of the project to
            - facilitate discussion when we are working
            - jot down quick notes
            - enable asynchronous working as queries can be sent, and the relevant parties can respond when they become available.
        - Voice channels for virtual team meetings.
        - Delegation of tasks within sprint goals outside of meetings was made easy by Discord's functionality to "@ ping" a specific team member, which sends them a notification with the relevant message.
- Resources and tools for developing the game's interface and graphics:
    - OpenGameArt.org (online free game asset resource)
    - Itch.io (game community which contains a game asset store)
    - Pixel-Me (online tool that converts pictures to pixel art)
    - Piskel (online pixel sprite creation tool)

b) Approach to team organisation

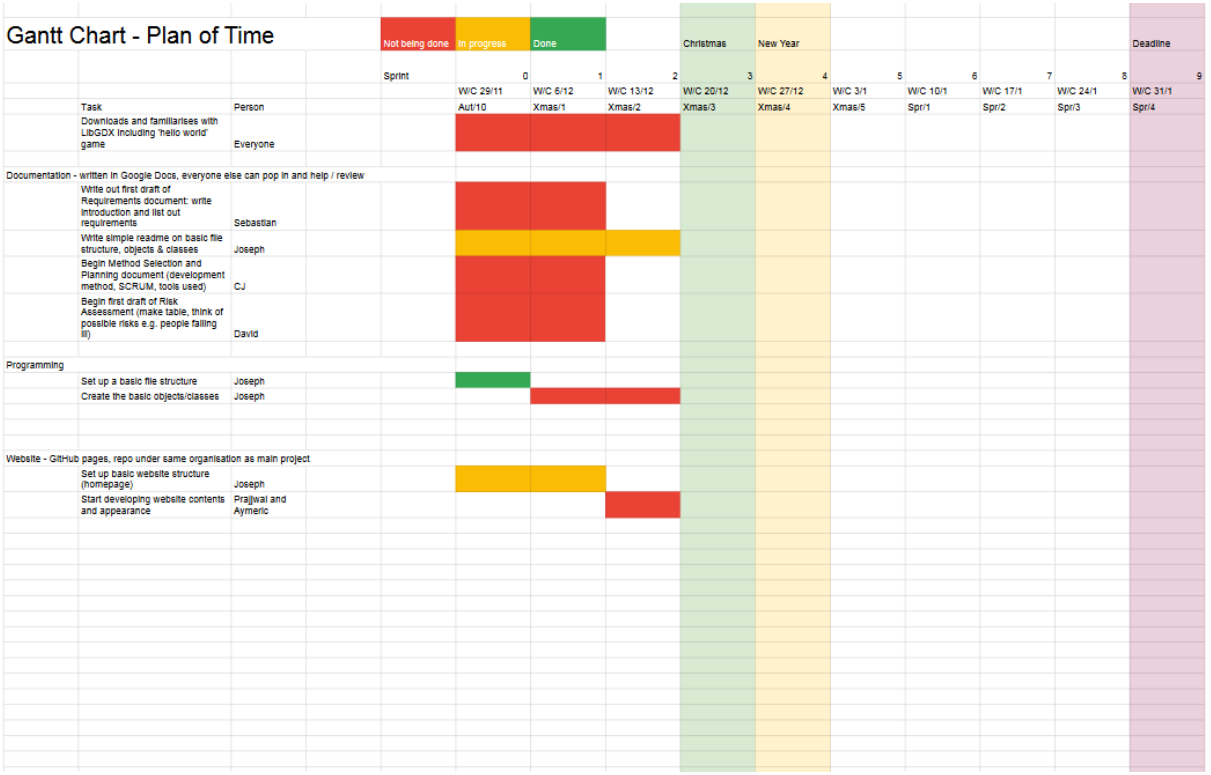
Team member	Role name	Role details
CJ	Chair/Leader	Plans when and where meetings are and make sure all topics are covered. Books rooms for meetings. Zoom host.
Aymeric	Secretary	Minuting/Recording meetings and decisions made in meetings. Writes brief summary of meetings.
Sebastian	Report Editor	Checks the format of reports. Ensures that reports and documents are organized properly
Joseph	Software Editor	Checks the consistency of code. Ensures that the code compiles and complies with general standards.
David	Progress	Ensures that the team is keeping on schedule.
Prajwal	Customer Support	Organises meetings with the customer. Makes sure that the project meets the requirements set out by the assessment guidelines

- Initially, each one of us was assigned a Team Role, as follows:

- As we worked on the project, we adapted our working style:
    - In some cases, sticking to the roles was helpful, for example CJ organised and lead meetings, ensuring everything was covered, and Joseph maintained the code's consistency and standard.
    - However overall it was more beneficial to assign ourselves tasks based on availability, and for the schedule to be maintained by us all collectively through Gantt chart updates during our meetings.
- Whilst Joseph and Sebastian were in charge of the implementation, and Prajjwal was responsible for the website, the rest of the team each selected a report to be in charge of.

c) Plan

Start:



Final: