# **6 - Project Management Tools**

## 6.1 Project Management

The main method of project management was throughout the use of Trello, this tracked all the sprints within the creation of Timeline Takedown. I followed the Kanban workflow with separate columns for to do, doing, and done I feel like more in this gap can be done.

It included other information than just sprints records, such as links to resources, dates, useful information and more as show in this screenshot. Full Trello board images can be found in Appendix 9.

A screenshot of a computer

AI-generated content may be incorrect.

*Figure 6: Trello Board*

## 6.2 Version Control

GitHub Desktop was used as the version control tool during this project. The desktop client was chosen due to its familiarity, and intuitive user interface which made it quicker and more accessible to manage commits compared to using GitHub through the command line.

Because I committed frequently as I was aiming to log every change made during development to efficiently track progress – using the command line version would have been far more time consuming as it would have required manually typing commands for each commit. With GitHub Desktop, this process was streamlined to just a few clicks.

## 6.3 Meetings

Meetings were used as an important project management tool throughout the development of Timeline Takedown. They provided structured checkpoints where the supervisor could track the overall progress and offer feedback or new recommendations to improve the quality of the project.

These meetings were held every other week in the Smeaton building at the University of Plymouth, typically lasting between 10-15- minutes.

## 6.4 Development Logs

During each sprint, a development log (Devlog) was recorded. These videos served as a project management tool to track progress, document any issues encountered during the sprint, and outline plans and ideas for the upcoming sprint. This approach was highly efficient as the recordings could be reviewed at a later date to quickly recap key developments, challenges, and decisions made throughout the project’s lifecycle. All devlogs can be found at Appendix 3.

## 6.5 Testing Methods

Testing was conducted using three different approaches throughout development. Firstly, self-testing took place during feature implementation. As soon as a new mechanic or functionality was added, I would immediately test it to ensure it worked as intended. This allowed for rapid feedback and adjustments while still in the development phase. For example, when setting the movement speed for enemies, I would quickly playtest to check if the speed felt appropriate. If it did not meet expectations, I would adjust the value and retest until satisfied. The same approach was used when deciding on enemy spawn locations – if the flow of enemies through the map felt unbalanced or unnatural, I could easily reposition the spawn points and test again in real time.

Sprint testing was conducted at the end of each two-week sprint, these sessions focused on the features that had been developed during that sprint. I referred to my Trello board, which listed each completed task, and systematically tested each one. Features would be ticked off the board once they were fully tested and verified. This ensured that no new functionality was forgotten or broken before recording the developer log for that sprint.

Lastly, usability testing was carried out to gather external user feedback. I hosted multiple in person playtesting sessions where volunteers could play the game either at my computer or in the Smeaton building at the University of Plymouth. This method was important as testers often noticed issues I had overlooked. For example, one tester discovered a floating model left above the map that I had completely forgotten about. All usability tests were conducted anonymously, with no personal data collected, in accordance with university guidelines.

In conclusion, testing was a crucial part of the project’s success. It helped uncover bugs, refine gameplay mechanics, and gather real human feedback, which led to important improvements. Without consistent and varied testing methods, many issues would have gone unnoticed, and the overall quality of the game would have been significantly lower.

See Appendix 10 for more information on the testing process, including examples of the testing process such as questions asked, and feedback received.