



## Development Diary

*This diary will contain every major meeting, sprint or advancement related to the project Sync or Bite.*

**03/04/2025**

Today we received the wording for the project, we have set up a meeting for next Friday, whose objective will be to plan everything. We've already identified some structures and concurrency methods worth using, we will discuss it next week.

**11/04/2025**

As said, today we organized the whole project, took some decisions in regard to the design and workload distribution. Moving on, *Tunnel* class has been implemented, ensuring that all requirements are followed in a thread safe manner.

**12/04/2025**

Today we have focused ourselves in the *Server.backend*, we implemented various classes, such as *Logger*, *Human*, *Food*, etc. Also, a simple *Main* class was created in *Server.frontend* in order to test basic functionality.

**13/04/2025**

Work continued on both backend and frontend elements. Backend updates included refining the *FoodGenerator*, improving modularity across various methods and implementing zombie attack logic. The frontend received new resources such as icons and a *GuiManager* draft was started. Additionally, the *App* class was created to serve as the launcher.

**14/04/2025**

Significant progress was made integrating client-server communication. Support for pausing functionality was added via a dedicated *PauseManager* and supporting threads. On the frontend, development of the *MapPage* began from scratch and reached a final draft by the end of the day.

**15/04/2025**

The GUI was extensively enhanced. Fonts were configured through a new *FontManager*, and the structure of *MapPage* was reworked to support visual elements like counters and crossing indicators. Functionality was added for showing entity lists across areas such as tunnels and unsafe zones. Furthermore, the "Top 3 most lethal zombies" feature was successfully implemented and general bug fixes applied across multiple components.

**16/04/2025**

Client-server pause synchronization was finalized. *PauseManager* logic was refined and properly integrated with both *MapPage* and *LogPage*. Various frontend bugs were resolved, including display and counting issues in unsafe areas and tunnels. GUI was polished and *App* was renamed to *ServerApp* for clarity. Log visualization was completed and formatting improved.

**17/04/2025**

Work focused on refining the "Top 3 most lethal zombies" feature to ensure accurate and real-time updates on the client side. The data transmission process was restructured to provide a cleaner and more reliable flow between server and frontend display components. Additional adjustments were made to improve the synchronization and responsiveness of the GUI elements involved. Minor polishing included standardizing colour naming across the interface for consistency and readability.

**18/04/2025**

Major improvements were made to the podium visualization system. A progress bar implementation was added to enhance the clarity of player rankings, alongside dynamic counters to reflect real-time changes. The internal logic of the podium component was streamlined for efficiency, reducing redundancy and simplifying updates. Icon synchronization between pause and resume states was also optimized, ensuring visual consistency across all relevant interface sections.

**21/04/2025**

Fixed *ScrollPane* and resizing issues on *MapPage*, enhancing overall layout stability. Visual inconsistencies like zombie label colours were corrected. Minor refactors included removing unused imports and attributes, as well as renaming variables for better clarity.

**22/04/2025**

Refinements to unsafe area logic were made to address stuck entity behaviour. Several method names were updated for clarity, and checks in *PauseManager* and related components were optimized. Connection loss handling was introduced alongside improvements in the client backend.

**23/04/2025**

A major architectural refactor was completed to enforce strict separation between frontend and backend components. Previously, several backend classes were directly invoking frontend methods, leading to circular dependencies and a brittle structure. This tight coupling was removed entirely by transitioning to a listener-based communication model.

The backend now exposes events through observer interfaces, which the frontend subscribes to. This decoupling ensures that the backend operates independently of any GUI logic, improving modularity and testability across the entire codebase. As part of this restructuring, existing direct method calls were replaced with event dispatches and several components were adjusted to support asynchronous updates via listeners.

**24/04/2025**

Efforts focused on refining backend and frontend synchronization, addressing potential concurrency issues. Several redundant or unused components were cleaned up and log behaviour was corrected in refuge zones. Preliminary updates were made to GUI event listeners and visual layouts, particularly in *MapPage* and *ServerApp* components.

**25/04/2025**

The interface was polished with structured GUI changes, improving consistency and layout behaviour. Resizing logic in scrollable panes was revised for smoother user experience. Various backend safety improvements were introduced to prevent unexpected behaviour, and shared variable protection was enhanced across sensitive zones such as tunnels and refuge areas.

**26/04/2025**

Stability updates were prioritized. Exception handling in the frontend was adjusted, redundant checks removed, and several concurrent access points were reviewed and optimized. GUI updates continued with improved resizing behaviour and additional documentation was added across classes to improve maintainability.

**27/04/2025**

Interface structure received notable clean-up and documentation. Colours and layout management were unified, and frontend component organization was improved to align with previous architectural refinements. Minor roaming behaviour adjustments were applied in the zombie movement logic to ensure realistic navigation patterns.

**28/04/2025**

Layout adjustments were made to the *MainClientPage* to better accommodate dynamic components. Overall GUI formatting was revisited to improve spacing and alignment. Streams were properly closed and unused output flows were removed to reinforce memory safety and resource handling.

**29/04/2025**

Explanations and inline comments were revised for clarity across the codebase. Minor GUI component behaviour was polished to ensure expected performance under variable conditions. Logging and user interface feedback were also subtly enhanced.

**30/04/2025**

Client-side GUI received further refinements, with updates to colour structures and zombie roaming logic. The *MapPage* component was enhanced to display newly introduced color-based indicators. Additionally, performance tuning included query refresh rate improvements and replacement of thread-lock mechanisms for better responsiveness.

**01/05/2025**

Various client settings were adjusted, including switching to unused network ports and improving code commentary for clarity. Dispatcher-related documentation was expanded, improving developer understanding of internal communication between frontend elements.

**02/05/2025**

Comments across multiple files were updated for improved accuracy and context. Minor visual and logic corrections were also made, maintaining coherence between documentation and application behaviour.

**03/05/2025**

Icon assets were reviewed and some were removed to avoid redundancy. A colour usage explanation was introduced to assist the user. General polishing concluded a series of GUI refinements, marking a more stable and visually coherent state.

**04/05/2025**

Today we finished the documentation and made some tiny tweaks to the code, such as changing some buttons.

**05/05/2025**

Today we submitted the project.