

Manual de Proyecto

1. Integrantes y Roles

- Maydana Gonzalez Lionel Gabriel -106512 - lmaydana@fi.uba.ar

Encargado de la lógica del servidor y de la unión con el protocolo. También sirvió como soporte de ayuda para otros ámbitos del proyecto.

- Torraca Lautaro - 108813 - ltorraca@fi.uba.ar

Encargado del protocolo de comunicación y menu (Qt) del juego. Desarrollo en conjunto con Lionel Maydana en partes del protocolo.

- Tomas Gonzalo Ruiz Quispe - 106904 - truiz@fi.uba.ar

Encargado de la lógica del cliente, renderizado del juego, texturas y sonidos.

2. Organización del Trabajo

Desde el inicio del proyecto, cada integrante eligió una parte específica del sistema para desarrollar, en función de sus intereses y habilidades. Esta división inicial incluyó componentes como la interfaz gráfica, el protocolo de comunicación, la lógica del juego, el manejo de red y la sincronización.

A lo largo del desarrollo, mantuvimos una dinámica de comunicación constante mediante reuniones diarias, donde cada uno compartía su progreso, dificultades y próximos pasos. Estas instancias nos permitieron colaborar activamente, ayudarnos mutuamente y resolver problemas de forma colectiva, favoreciendo una integración más fluida entre las distintas partes del sistema.

3. Herramientas utilizadas

- IDEs: CLion, nvim, VSCode en casos particulares
- Linters o herramientas de formato: clang_format, ast_grep
- Herramientas externas: GitHub, Discord, PhotoShop, PixelArt Maker (para los mapas), ChatGPT para facilitar la búsqueda de documentación de librerías.

4. Recomendaciones para el Taller

CMake: con explicado en las clases no fue suficiente para entender el funcionamiento básico de CMake y como configurarlo para proyectos con múltiples directorios que utilizan diferentes librerías y dependencias.

6. Dificultades encontradas

Uno de los principales desafíos técnicos fue la configuración inicial del entorno con CMake, especialmente al integrar múltiples módulos y bibliotecas externas como SDL y Qt. Tuvimos que resolver varios errores relacionados con rutas, dependencias y compatibilidad entre plataformas.

Además, la organización del trabajo se vio condicionada por la alta carga académica de otras materias. Fue complejo coordinar horarios y avanzar de forma sostenida, ya que muchos integrantes contaban con exámenes y entregas en paralelo. Esto nos obligó a ajustar plazos, reorganizar tareas sobre la marcha y priorizar aquellas funcionalidades esenciales para asegurar una versión funcional del proyecto dentro de los tiempos establecidos.

No llegamos a implementar todas las mejoras que teníamos planificadas. Algunas funcionalidades, como la mejora de la sala de espera (Waiting Room), el refinamiento del sistema de movimiento del jugador y la incorporación de animaciones adicionales, quedaron pendientes o implementadas de forma básica. Aun así, logramos entregar una versión jugable y estable del juego, cumpliendo con los objetivos principales.

Durante las pruebas finales detectamos algunos errores funcionales que no pudieron ser resueltos a tiempo:

- El botón de salida en la sala de espera (Waiting Room) presenta un comportamiento incorrecto: al presionarlo, cierra el menú pero no regresa al menú principal, como sería lo esperado.
- En el proceso de creación de partidas, si el jugador cierra la ventana durante la creación de una sala, se produce una inconsistencia en el estado del servidor. Esto genera una partida “fantasma” que en realidad no existe, pero que sigue apareciendo en la lista de salas, permitiendo que otros jugadores intenten unirse y provocando errores críticos en el juego.

7. Reflexión final

Desde el punto de vista del código, creemos que la estructura del cliente podría estar mejor separada en clases, lo que facilitaría su mantenimiento y escalabilidad. En el lado del servidor, observamos GameMonitor y GameMap pueden mejoradas. Además, dentro del menú del juego, consideramos que la clase WaitingRoom podría tener una implementación más limpia y desacoplada del resto de la lógica.

También identificamos oportunidades para mejorar algunas prácticas de programación, tanto en el cliente como en el servidor, aplicando principios de diseño más claros, una mejor separación de responsabilidades y un uso más consistente de patrones de diseño, lo cual permitiría reducir la complejidad del código y facilitar futuras extensiones.

A nivel grupal, creemos que tuvimos un buen desempeño, manteniendo una comunicación fluida y colaborando activamente en la resolución de problemas. La distribución de tareas fue adecuada y, si bien hubo contratiempos, incluyendo la baja de un integrante, logramos coordinar esfuerzos para alcanzar una versión funcional y estable del proyecto.