

# Manual de proyecto

## Área de trabajo de los participantes

Los integrantes del equipo son Juan Manuel Pascual Osorio, Maximiliano Otero y Martin Juan Cwikla.

Las tareas en el principio del proyecto nos las dividimos de la siguiente manera:

- Juan Manuel se encargó de las visuales del juego utilizando la librería SDL, lo que incluyo tareas como organizar el flujo del render loop, detectar la información era necesaria para el cliente y procesarla al recibirla, manejar los distintos estados de nivel según el elegido y organizar las entidades del juego.
- Maxi en un principio se encargó de las bases de la comunicación entre cliente servidor y fue el principal responsable de organizar los threads y la correcta administración de los recursos. Luego se encargó de realizar el menú del juego con la librería QT, apporto también en algunas tareas pendientes como la cámara, su sistema de prioridad y finalmente el sistema de turnos.
- Martin se encargo de toda la parte asociada al engine de físicas el cual incluía tareas como, calcular trayectorias, detectar colisiones y reaccionar a ellas, adaptar el engine a los distintos tipos de armas, corregir los casos en donde la precisión del modelo pudiera llegar a interrumpir con la jugabilidad, entre otras.

## Forma de trabajo

La primera semana el foco estuvo en leer y aprender documentación de las librerías que teníamos que implementar para el trabajo, como SDL, Box2d.

La segunda semana, partimos de la base de la base que teníamos del segundo trabajo practico y la modificamos según las especificaciones del TP en jornadas de peer programming.

En la tercera semana, ya con una base, fuimos capaces de distribuirnos las tareas asociadas a las bibliotecas que habíamos investigado. Empezaron a aparecer las primeras animaciones y reacciones del engine ante el presionado de botones.

En la cuarta semana, nos centramos en implementar la trayectoria y lógica de colisiones de la primera arma desde el lado del servidor, manejando las físicas, su comunicación con el cliente, y desde el lado del cliente, ocupándonos de su renderización. Al mismo tiempo, comenzamos con la implementación del menú del juego utilizando la biblioteca QT. Finalmente, realizamos los primeros pasos para implementar el sistema de turnos y la incorporación de la lógica de la cámara.

Durante la quinta semana de desarrollo, se implementaron las físicas de diversas entidades del juego, como la granada verde, la granada roja, el mortero, la granada santa, el ataque aéreo, la teletransportación y el bate. También se agregaron múltiples worms por cada jugador para enriquecer la dinámica del juego. En paralelo, se realizó una refactorización del lado del servidor para mejorar la eficiencia del código. Se introdujo un sistema de daño por caída y se aplicó el viento a las entidades del juego, agregando elementos realistas y dinámicos. Además, se estableció una comunicación eficiente de los estados necesarios para la renderización de las armas, y se implementaron las físicas y renderización de las cajas. Se realizaron correcciones en la movilidad de los worms y se añadió un menú de trucos para ofrecer opciones adicionales a los jugadores.

En la sexta semana, se trabajó en la animación de muerte y se corrigieron problemas en la implementación de algunas armas para garantizar su funcionamiento correcto. Se completó la renderización de todas las armas, y se solucionaron problemas relacionados con la cámara del juego. Se crearon nuevos mapas con puntos de spawn estratégicos para los worms. Se implementó una animación de victoria-derrota para mejorar la conclusión de cada partida, y se realizaron ajustes en la movilidad de los worms. Se añadieron efectos sonoros para mejorar la experiencia auditiva, y se llevaron a cabo pruebas del protocolo. Además, se implementó la asignación de nombres y equipos para los worms, y se creó un instalador para facilitar la distribución y el acceso al juego. Finalmente, se realizaron importantes avances para mejorar la experiencia del juego. Se trabajó en la animación de la fuerza del viento, brindando a los jugadores una representación visual dinámica de este elemento en constante cambio. Además, se perfeccionaron las texturas del agua, añadiendo un toque visual más realista y envolvente al entorno del juego y se agregó una interfaz que indica las armas disponibles y la cantidad de balas de cada una de ellas.

Finalmente, en la séptima semana, nos encargamos de documentar todo nuestro trabajo y arreglar algunos problemas restantes.

## IDEs y documentación

Los IDEs que utilizamos fueron Clion y Visual Studio Code, y como linter utilizamos Cppcheck.

La documentación principal que utilizamos para trabajar y aprender acerca de Box2d fue:

- <https://box2d.org/documentation/index.html>
- <https://www.iforce2d.net/b2dtut/>
- <https://discord.gg/NKYgCBP>

En el caso de SDL, la principal documentación fue:

- <https://wiki.libsdl.org/SDL2/FrontPage>
- <https://github.com/libSDL2pp/libSDL2pp-tutorial>
- <https://lazyfoo.net/tutorials/SDL/>
- <https://github.com/Taller-de-Programacion/clases/tree/master/bibliotecas-gui/sdlpp>

## Puntos más problemáticos

Los puntos más problemáticos fueron, del lado del cliente, la animación de las herramientas, como la teletransportación y el ataque aéreo, ya que precisaba de concatenar dos animaciones en simultaneo. Del lado del servidor, el sistema de turnos dado que requería de un análisis profundo por la cantidad de casos bordes presentes, y el las multiples colisiones de las entidades del juego, ya que es una acción que sucede con mucha frecuencia en el juego y las diferentes entidades tienen reacciones distintas según la situación.

A pesar de los desafíos previos mencionados, logramos cumplir con todas las especificaciones de la consigna. Sin embargo, persisten algunos problemas menores que afectan la jugabilidad. Un aspecto destacado es la gestión de la colisión de los worms: implementamos la lógica para evitar colisiones entre ellos, pero la esencia de la consigna apunta a permitir la colisión, con la condición de que el jugador que no está en su turno no reaccione ante la misma, permaneciendo en su lugar.

Asimismo, enfrentamos ciertas dificultades con la mecánica del daño por caída. Aunque hemos perfeccionado esta característica en la medida de lo posible, aún existen casos donde la aplicación del daño puede no ser óptima.

Finalmente, en cuanto a la renderización de las vigas, se ha detectado un pequeño problema. Actualmente, están desplazadas unos píxeles a la derecha de su posición correcta.

## Conclusión

Siendo nuestro primer proyecto de esta envergadura, como equipo experimentamos un desempeño sólido a lo largo de todo el trabajo. La capacidad de organización fue un punto fuerte, logrando una distribución eficiente de tareas y una colaboración efectiva cuando fue necesario trabajar conjuntamente. En términos organizativos, no cambiaríamos nada; sin embargo, en retrospectiva, reconocemos que podríamos haber tomado decisiones más efectivas en cuanto a la documentación del código.

Una mejora potencial podría haber sido adoptar la práctica de documentar el código de manera incremental a medida que avanzábamos en el proyecto. En lugar de posponer la documentación hasta el final, hubiera sido beneficioso registrar decisiones clave, explicaciones y comentarios durante el desarrollo. Esto no solo habría facilitado el entendimiento del código por parte de otros miembros del equipo, sino que también habría simplificado la tarea de revisar y corregir secciones específicas. Aprendiendo de esta experiencia, consideraremos la documentación continua como una práctica valiosa en proyectos futuros, evitando así la acumulación de esta tarea al final del desarrollo.