

Manual de Proyecto

Integrantes

- 109186 - Esteban Ariel Brandán
- 106930 - Ignacio Viau
- 108230 - Lucas Ezequiel Villarrubia

División de tareas

Esteban	Ignacio	Lucas
<ul style="list-style-type: none">• Implementación de armas y proyectiles en cliente y servidor• Control de sistema de físicas de proyectiles• Correcta visualización de armas con sprites correspondientes en el cliente con zoom ya implementado• Cambios de estado de personajes e interacciones con armas• Adaptación de protocolo para el ingreso de comandos de usuario en cliente y servidor para interacciones con armas	<ul style="list-style-type: none">• Setup de librerías SDL para el renderizado de la demo y manejo de runtime errors en Renderer• Renderizado de personajes, texturas en runtime• Renderizado de mapa en cliente gráfico: zoom-in y zoom-out de cámara que sigue a patos en juego• Utilización de mapas en cliente y servidor en distintas clases• Movimientos de personajes y control de sistema de físicas primero en cliente• Construcción de mapas y entidades dentro de mapas: inserción por medio de archivos .yaml• Instalador shell• Sonidos	<ul style="list-style-type: none">• Modelado de esquema de clases e hilos utilizados en servidor y cliente• Modelado de protocolo de comunicación principal en clases y structs• Adaptación de servidor con estado de juego y personajes según estado inicial• Manejo de input de usuario por teclado y envío de información de cliente a servidor• Construcción de lobby de juego y manejo de eventos de lobby.• Motor de colisiones de personajes con elementos del terreno de juego• Modelo de multijugador y partidas en servidor• Regulación de performance parcial y rediseño de renderer

		<ul style="list-style-type: none"> • Extracción de archivos yaml para inicialización en servidor • Manejo de estados en servidor y cliente sin contar estados correspondientes a las interacciones con armas. • Frame dropping en cliente • Tests y documentación
--	--	---

Items faltantes

1. Cajas, interacciones: colisiones, destrucción, recompensas.
2. Armas: Armas láseres y rebote. Dispersión de proyectiles, retroceso de pato al disparar.
3. Instalador: se instalan las dependencias y se mueven los binarios correctamente pero la ejecución es errónea.
4. Mapa ciudad: no funciona.
5. Cliente no renderiza efectos de casco y armadura (los patos la llevan por defecto).
6. Aplicación de linters para commit (uso de pre-commit)

Recursos utilizados

Para este trabajo práctico se utilizaron las librerías de las clases Socket, Resolver, ResolverError, LibError, Queue, y Thread; y el esquema de directorios y archivos CMakeLists.txt provistos en tres repositorios de la cátedra:

- [hands-on-sockets](#)
- [hands-on-threads](#)
- [template](#)

Entre otras tecnologías, se utilizaron los IDEs de CLion y Visual Studio Code, así como el Editor de Texto para programar el código presentado.

Se utilizó el repositorio template provisto por la cátedra para el setup del proyecto con *cmake* y *makefile*.

En el caso de la visualización de la demo, se utilizaron las librerías *image*, *mixer* y *tff* de SDL2, y en el código, clases de su versión SDL2pp.

Para el lobby de la demo, se utilizó la librería QT en su versión 6 y su personalización fue esquematizada al principio con QtCreator.

