

# Manual de proyecto

## Organización del equipo

En primera instancia como equipo definimos qué tareas resultaban cruciales para lograr cumplir con lo requerido en la primera entrega.

Antes de eso, definimos la estructura del proyecto. Y determinamos tres partes generales: el cliente, el servidor, el editor y la interfaz gráfica, que dentro lleva la renderización de los elementos y la creación de animaciones.

### Semana 1: (7/05 - 14/05)

- Creación del repositorio.
- Organización general del equipo.
- Creación de los archivos main del cliente y servidor.
- Agregado de prototipo básico de físicas.

### Semana 2: (14/05 - 21/05)

- Definición de entidades para el protocolo de comunicación para updates del juego.
- Creación de la estructura básica de clases correspondiente al cliente.
- Agregado de versión básica de la interfaz y lógica del lobby con QT.
- Creación de la versión básica del editor de niveles.
- Implementación básica de la estructura del servidor y game loop.
- Implementación básica del main del cliente.

### Semana 3: (21/05 - 28/05)

- Implementación del loop que renderiza la interfaz.
- Creación el Event Listener, que escucha las teclas y las vincula con los eventos correspondientes.
- Implementación del render.
- Se agregan el Updater y la queue entre el updater y el render.
- Creación de la carpeta commons, que contiene archivos utilizados por ambas partes del proyecto (servidor y cliente) como threads y sockets.
- Incorporación de la librería de SDL.
- Se agregan las animaciones de corriendo y parado.

### Semana 4: (28/05 - 4/06)

- Agregado de manejo temporal de ticks desde el servidor.
- Implementación del módulo del mapa.

- Vinculación de los módulos del cliente, verificamos que se muestre algo por pantalla y el buen funcionamiento del event listener.
- Incorporación del render para el estado del juego.
- Agregado de inicialización de guiloop y render.
- Implementación de la reacción del mapa al movimiento.

#### **Semana 5: (4/06 - 11/06)**

- Corrección de conflictos por endianness.
- Modificación de la clase thread.
- Agregado de pasos a seguir para la instalación al ReadMe.
- Agregado de estructuras comunes entre el servidor y cliente para modulo del Lobby.
- Limpieza del código.
- [Primera entrega] Se mostraba un conejo con animación caminando y respondía a las teclas de movimiento hacia la izquierda y la derecha.
- Actualización protocolo de updates.

#### **Semana 6: (11/06 - 18/06)**

- Implementación del protocolo para módulo Lobby desde el servidor.
- Implementación del Lobby Manager, que permite manejar distintas partidas.
- Implementación de colisiones.
- Implementación de editor de mapas.
- Estructura de archivo de configuración YAML.

#### **Semana 7: (18/06 - 25/06)**

- Implementación de configuración a través de YAML.
- Refactorización de Entity, Game State, Entity Factory y textureManager.
- Implementación de funcionalidades del personaje: gasto de munición, cambio de arma y disparos.
- Creación de Vagrantfile con instalador.
- Implementación de sound manager, con música y efectos de sonido.
- Incorporación de ítems, y otros personajes.
- Respawn de los jugadores y los enemigos.
- Implementación de animaciones restantes.

## Desarrollo por integrante:

**Hernán de San Vicente** se encargó en líneas generales de la lógica del juego del lado del server. Específicamente trabajo en el bucle del juego con sus tiempos limitados para no quemar la CPU, la creación de entidades donde cada una implementa su propio procesamiento, el procesamiento de cada una de las entidades para registro de colisiones con el mapa y con otras entidades, las acciones del jugador  
Y también creó la entidad cliente del lado del server con sus threads de recepción y envío de información

**Mariano Lazzarini** se encargó en líneas generales de las interfaces QT (el lobby, el editor, y la pantalla de puntajes). También creó la estructura general del hilo principal de ejecución del cliente (gui loop), que luego fue revisada y ajustada por Matias.  
Creó el módulo que administra los archivos de mapa, y parte de la estructura de updates para el protocolo.  
Por último, creó el prototipo de física, que luego fue parcialmente usado por las entidades para calcular las colisiones entre ellas, y con el entorno.

**Matias Rea** se encargó en líneas generales del módulo de la interfaz gráfica.  
Específicamente trabajo en: estructura que administra y controla el estado del juego desde la parte del cliente, utilizada en la interfaz gráfica,  
También de la implementación del administrador de texturas, mejorado por Mariano.  
Creación del sistema de renderizado y entidades encargado de mostrar por pantalla el estado del juego, tanto el mapa como las posibles entidades.

**Lara Converso** se encargó en líneas generales del hilo event listener, el protocolo que va alrededor de este. Además trabajó en la implementación de la configuración a través de el archivo yaml, el módulo del Lobby desde el lado del servidor y la estructura de updates de inicialización del servidor.  
También armó el archivo Vagrant y el instalador. E Implementó el Sound Manager utilizado tanto para la música del juego como para los efectos de sonido.

## Herramientas utilizadas

(Entre paréntesis se encuentra el link a la documentación utilizada para la implementación de las herramientas.)

- SDL2pp. ( <https://github.com/libSDL2pp/libSDL2pp-tutorial> )
- QT y QT creator. ( <https://doc.qt.io/> )
- Visual Studio Code.
- LiveShare (VsCode)
- CLion.

- Audacity.
- Pinta
- PlantUml ( <https://plantuml.com/es/guide> )

## Información adicional

### ¿Cuáles fueron los puntos más problemáticos?

- La utilización de sdl wait event.
- Crear todas las entidades en el servidor.
- La implementación de la interfaz gráfica con SDL.
- El manejo de los sonidos, nos trajo problemas por los distintos tipos de archivos.

### ¿Pudieron llegar con todo?

- Los bloques triangulares
- El estado del personaje intoxicado, se encuentra la animación pero el jugador nunca llega a ese estado.
- Acción dash o correr muy rápido, se encuentra la animación pero el jugador no puede realizar la acción.

### Si volvieran a hacer el proyecto, que cambiarían (a nivel código o a nivel organizacional).

En la parte organizacional si bien pudimos dividir bien los sectores y las features, además de utilizar las issues de github como guía, hubiera estado bueno la aplicación de un tablero kanban que nos permita hacer visible el progreso.

### ¿Hay algo que crean que debería darse en Taller?

Administración de archivos, como hacer la parte de cargar la biblioteca

Cómo sincronizar valores entre el cliente y el servidor, si bien implementamos un update general y moldeable a lo que se quiere mandar.