

# Trabajo Práctico Final

## Manual de Proyecto

### Grupo 26



<u>Nombre Integrante</u>	<u>Padrón</u>
Máximo Giovanettoni	110303
Ulises Valentín Tripaldi	111919
Bautista Cánepa	111892
Máximo Calderón Vasil	111810

# Introducción

En este informe se detalla la metodología de trabajo del proyecto, la distribución de tareas realizada y la documentación utilizada junto a una reflexión en cuanto a todo el trabajo hecho.

## Desarrollo del proyecto

Todo el proceso realizado se puede dividir en tres etapas distintas: Onboarding, Core development y Post development. Esta diferenciación se ve marcada principalmente por la forma de trabajo, la división de tareas y finalmente el desarrollo realizado. En cuanto a la organización, se realizaron reuniones semanales para hacer actualizaciones en el trabajo y alineamiento de tareas.

### Onboarding

La primera etapa del proyecto fue el Onboarding en el cual el equipo se encargó de analizar el desarrollo, dividir las tareas y hacer las investigaciones necesarias previas al desarrollo. Luego del análisis se dividió el proyecto en 3 partes fundamentales las cuales fueron asignadas a cada uno de los integrantes.

- Lógica de juego: Ulises Valentín Tripaldi y Máximo Giovanettoni
- Desarrollo de Protocolo y Cliente-Servidor: Máximo Calderón Vasil
- UI: Bautista Cánepa

Como parte de la investigación se usó la siguiente documentación para profundizar en las herramientas dadas:

- <https://doc.qt.io/>
- <https://wiki.libsdl.org/SDL2/FrontPage>
- <https://box2d.org/documentation/>

### Core Development

Luego de hacer la investigación se realizó el desarrollo del proyecto como fue previamente dividido. Esta etapa se centró principalmente en el esqueleto del proyecto, desarrollando la base de las clases (desde cero) y posteriormente integrando entre sí los módulos para terminar obteniendo un proyecto con las funcionalidades esenciales.

Durante este proceso se implementaron tanto la lógica del juego como el motor gráfico, hasta alcanzar una interfaz en la que el usuario podía interactuar y controlar un vehículo dentro de un circuito básico. Además, desde el lado del envío de información, para poder realizar esto se tuvieron que desarrollar en su totalidad las clases cliente, servidor y protocolo.

## Post Development

Una vez realizado el esqueleto principal del proyecto, el equipo dejó de tener roles definidos y pasó a agregar las features restantes sobre el desarrollo previamente realizado sin importar las asignaciones previas. Parte de este desarrollo incluyó el sistema de elección de auto y mejoras, NPC, HUD, entre otros. Además, durante este tiempo también se corrigieron errores y refactorizó el código para obtener un resultado más limpio. Finalmente, se hizo la documentación necesaria para la entrega final.

## Reflexiones

A lo largo del proyecto, nuestro principal impedimento fue el uso de nuevas herramientas desconocidas para todos tales como las librerías gráficas, el motor del juego o inclusive el lenguaje mismo y CMAKE. Al momento de la entrega un punto a mejorar sería el modelado del mapa ya que por una cuestión de tiempos el modelado no fue completamente preciso. Otra dificultad fue el manejo de los NPC, para el cual deben tenerse en cuenta y manejar muchas variables en relación a su comportamiento y aleatoriedad. En una posible mejora al respecto se podrían realizar mejores patrones de comportamiento para los mismos. En cuanto al archivo YAML de configuración para las constantes estadísticas, se realizó un desarrollo pero por cuestiones de tiempo nos fue imposible integrarlo con el trabajo.

Finalmente, respecto a la consigna y la cursada de la materia consideramos que sería de utilidad profundizar el uso de las bibliotecas de UI en alguna clase extra. También nos parece de utilidad poder tener alguna clase presencial de consultas generales para poder resolver dudas y conflictos particulares del trabajo. Consideramos que la presencialidad en algunos casos ayuda a una mejor comunicación y resolución de problemas.

## Herramientas utilizadas

- Sistema operativo: Ubuntu 24.04
- IDE: Visual Studio Code
- Control de versiones: Github
- Compilador: GNU Compiler Collection
- Bibliotecas utilizadas: SDL2PP, QT5 y Box2D