Informe de Aportes y Desarrollo del Proyecto

Manuel Pato Ibáñez	110640
Lucas Nicolás Pagani	110777
Federico Zanor	112097
Joaquín Schapira	112459

3 de Diciembre de 2025

Índice

1. Aportes de cada integrante

1.1. Manuel Pato Ibáñez

Manuel trabajó principalmente en el **mapeo de colisiones de los mapas** y en el **editor**. Realizó el **manual del usuario**, diseñó los **tests**, hizo los **PDFs** de documentación en **LaTeX** y tuvo pequeños aportes en el apartado visual del **lobby**.

Herramientas y documentación utilizadas:

- Utilizó en todo momento **Visual Studio Code**.
- Para la parte del mapeo de colisiones usó un **script propio en Python** disponible en:
  `https://github.com/Manupato/Script-Mapeo`
- El desarrollo en Qt6 se basó en la documentación de **Qt Group** (`doc.qt.io`), particularmente en su módulo de **widgets para C++**:
  `https://doc.qt.io/qt-6/qtwidgets-module.html`

1.2. Lucas Pagani

Lucas Pagani se dedicó de lleno al **servidor**. En constante comunicación con Manuel, quien generaba la matriz de colisiones y aportó numerosas ideas creativas sobre cómo representar distintos elementos en la matriz (sensores, checkpoints, paredes de diferentes niveles, etc.).

Por otro lado, diariamente mantenía comunicación con Federico sobre los **nuevos features del servidor** que requerían implementación en el cliente, indicando qué líneas descomentar cuando implemente su contraparte y el protocolo correspondiente.

Herramientas y documentación utilizadas:

- Visual Studio Code
- Cppcheck, clang-format, cpplint

- **box2d**:
  `https://box2d.org/documentation/index.html`
  `https://www.youtube.com/playlist?list=PLRqwX-V7Uu6Zy4FyZtCHsZc_K0BrXzxfE`
- **yaml-cpp**:
  `https://github.com/jbeder/yaml-cpp`
- **nlohmann_json**:
  `https://github.com/nlohmann/json`

1.3. Federico Zanor

Federico Zanor trabajó exclusivamente en el **cliente**, enfocándose en la implementación del protocolo de comunicación Cliente-Servidor, manteniendo una comunicación constante con Lucas.

También implementó la lógica del juego en el cliente utilizando SDL2pp, incluyendo gestión de eventos, renderización y manejo de inputs.

Herramientas y documentación utilizadas:

- Visual Studio Code
- **SDL2pp**:
  `https://github.com/libSDL2pp/libSDL2pp-tutorial`
- **Assets**:
  `https://pixabay.com/sound-effects`
  `https://ar.pinterest.com/`

1.4. Joaquín Schapira

Joaquín Schapira se encargó principalmente de toda la parte relacionada con la **lobby del juego**, desarrollada en Qt, y su integración con el protocolo del cliente.

Su trabajo abarcó la lobby, la pre-lobby y las distintas interacciones entre ambas. Además, implementó la música y efectos de sonido del juego.

Finalmente, realizó las supresiones de Valgrind y preparó el video promocional del proyecto.

Documentación utilizada:

- Documentación oficial de Qt6, especialmente el módulo de widgets para C++: `https://doc.qt.io/qt-6/qtwidgets-module.html`

---

2. Desarrollo del proyecto

La primera semana se centró en **charlar, realizar llamadas y entender la consigna**, cómo organizarnos y dividir las tareas. La primera reunión con el corrector fue un día después de recibir la consigna y allí consultamos sobre un posible **plan de acción**. Finalmente se optó por asignar una persona al **editor**, otra al **cliente**, otra al **servidor**, y la cuarta como **comodín y mediador**.

Tras esa primera semana dedicada al estudio de las librerías necesarias, durante la segunda semana comenzamos con el **código**, avanzando desde lo más bajo nivel (mover un auto, colisiones básicas) hasta lo más alto nivel (partidas múltiples, carreras simultáneas).

La coordinación entre **cliente-servidor** y entre **editor-servidor** se hacía principalmente por mensaje, dado el protocolo utilizado. Las comunicaciones fueron siempre rápidas y claras. Cuando una parte completaba algo, dejaba comentado en el código qué descomentar cuando la otra parte lo implementara.

Siempre intentamos tener la rama main funcionando, por lo que cuando empezabamos a desarrollar cosas que iban a tardar más de unas horas, lo haciamos en ramas separadas para mantener esa decisión de trabajo. Las ultimas semanas donde los features agregados solamente sumaban a una base que andaba (no existian commits que rompian lo ya implementado), empezamos a subir directamente todo a main.

3. Resultados y estado final

Se implementaron **todos los features solicitados**. Nos hubiese gustado tener tiempo para **embellecer más el juego y código**, con mas animaciones y detalles mas minusculos que podian llegar a sumar.

Todos los errores detectados fueron corregidos y el resultado final es un juego con **muy pocos bugs**. Actualmente solo se observan:

1. Un auto muy grande puede quedar atorado en un puente muy estrecho (comportamiento físico esperado).
2. No es un bug, ya que directamente no lo permitimos, pero no se puede poner la salida de autos en puentes.

Si rehiciéramos el proyecto, sabríamos qué problemas atacar desde el comienzo. Algunas soluciones iniciales parecían correctas pero luego dejaban de serlo al incorporar nuevos features. Un ejemplo fue el primer intento de **elevación de autos al pasar por puentes**, que resultó no solo incorrecto, sino que más complejo que la versión final.

4. Dificultades encontradas

- Representar vehículos pasando **por debajo de puentes u objetos**.
- Implementar correctamente la **aceleración** de los automóviles.
- El **mapeo de colisiones**, especialmente al incorporar sensores de altura.
- Manejar **múltiples partidas simultáneas** en un mismo servidor.
- La **manejabilidad de los autos** fue uno de los aspectos más desafiantes de balancear. Nos llevó varias iteraciones encontrar el punto justo entre que el manejo sea demasiado sencillo (lo que volvía al juego aburrido) o excesivamente difícil (resultando frustrante para el jugador). Actualmente llegamos a un punto medio donde al principio es un poco dificil pero jugando un par de carreras, ya se acostumbra el usuario al movimiento, siendo los autos más pesados mucho más fáciles de manejar y en el otro extremo, los autos más rápidos tienen un manejo significativamente más sensible y pueden volverse difíciles de dominar.

5. Sugerencias

Creemos que todo lo dictado en la materia fue acorde y **necesario** para poder realizar el trabajo práctico.

Aunque alguna clase básica de box2d hubiera servido mucho, tambien se entiende que la gracia del trabajo es la investigación propia por lo que no tenemos ninguna sugerencia.