

# Manual de Proyecto Wolfenstein 3D

[7542] Taller de Programación I  
Segundo cuatrimestre de 2020

Alumno :	BIANCARDI, Julián
Padrón:	103945
Alumno :	CZOP, Santiago
Padrón:	104057
Alumno:	GIARDINA, Fernando
Padrón:	103732
Alumno :	STENGHELE, Juan Francisco
Padrón:	104000

# Índice

<b>1. Integrantes - División de Tareas</b>	<b>2</b>
<b>2. Evolución del proyecto</b>	<b>2</b>
<b>3. Inconvenientes encontrados</b>	<b>2</b>
3.1. Tiempo . . . . .	2
3.2. Conocimientos de diseño . . . . .	2
<b>4. Análisis de puntos pendientes</b>	<b>3</b>
4.1. Bots LUA: Disparos . . . . .	3
4.2. Creación del Instalador . . . . .	3
<b>5. Herramientas</b>	<b>3</b>
5.1. QtDesigner . . . . .	3
5.2. Git . . . . .	3
5.3. CLion . . . . .	3
5.4. VSCode . . . . .	3
<b>6. Conclusiones</b>	<b>3</b>

## 1. Integrantes - División de Tareas

- Biancardi, Julián : Editor completo, pantalla de login y de partidas (launcher). Leaderboard. Testing. Armado de documentación necesaria. Carga y reproducción de animaciones y de algunos sonidos. Implementación del HUD.
- Czop, Santiago : Ecuación de rayos y visualización. Motor 3D completo mediante el uso de Ray Casting. Comunicación cliente-servidor: creación de "packets". Puertas y pasadizos Secretos en el cliente. Carga de mapas.
- Giardina, Fernando : Bosquejo del modelado del servidor, creación de archivo de configuraciones YAML, lógica inicial del ataque, integración con LUA: módulo de IA, reproducción de sonidos, testing, armado de documentación.
- Stenghele, Juan Francisco : Lógica de movimientos y colisiones de los personajes. Lógicas de las partidas en el servidor y parte en el cliente. Clock de la partida y eventos por tiempo. Funcionamiento de los misiles. Puertas y pasadizos Secretos en servidor. Armado de documentación y testing.

## 2. Evolución del proyecto

- Semana 1: Bosquejo del modelado del servidor. Ecuación de rayos. Análisis del enunciado y de posibles stories.
- Semana 2: Ray-Casting y visualización de elementos. Archivo de configuraciones YAML.
- Semana 3: Lógica inicial del ataque. Lógica de movimientos. Finalización del motor 3d: Ray-casting.
- Semana 4: Conexión cliente-servidor. Lógica de las partidas.
- Semana 5 (+ receso): Módulo de IA (LUA). Editor completo.
- Semana 6: Reproducción de sonidos. Lógica de los misiles. Clock del juego.
- Semana 7: Puertas. Pasadizos Secretos. Resolución de bugs. Testing, armado de la documentación.

## 3. Inconvenientes encontrados

### 3.1. Tiempo

Consideramos que el tiempo fue un inconveniente bastante grande, propio de lo que sucede en gran cantidad de proyectos informáticos. Si bien pudimos trabajar durante parte del receso, de todas formas encontramos dificultades para cumplir plazos de algunas tareas, lo que generó trabajos más intensos durante los últimos días previos al deadline.

### 3.2. Conocimientos de diseño

Otro problema que tuvimos fue que, al no tener suficientes conocimientos en la rama de diseño, nos faltó diseñar ciertos Sprites, que no encontramos en internet, como el sprite del perro quieto, entre otros.

## 4. Análisis de puntos pendientes

### 4.1. Bots LUA: Disparos

Por razones de tiempo, no se pudo hacer más de un tipo de bot, en nuestro caso, lo que éste hace, es buscar al jugador más cercano e intentar acuchillarlo. En el futuro, podría ser implementado sin mayores dificultades la posibilidad de que también pueda disparar con algún tipo de arma, y no solo acuchillar. Nos hubiera encantado poder hacerlo, si bien el enunciado no requería más de uno.

### 4.2. Creación del Instalador

Otro punto que nos hubiera gustado realizar, es el instalador, que por razones análogas a lo anterior (tiempo), no pudimos, y que, si bien no lo pedía el enunciado nos hubiera encantado que fuera más semi-automático el proceso de instalación del juego.

## 5. Herramientas

### 5.1. QtDesigner

Esta herramienta nos permitió diseñar la interfaz gráfica del juego, como ser la pantalla de login, y la pantalla en la sala de espera.

### 5.2. Git

El proyecto fue realizado de manera colaborativa por los cuatro integrantes. Para simplicidad, se utilizó la plataforma GitHub con algunas de las herramientas que éstas nos provee, con el fin de lograr fluidez en el trabajo conjunto.

### 5.3. CLion

La mitad del equipo usó CLion, debido a que nos provee herramientas para facilitar el trabajo, como ser la integración con Github, un análisis de sintaxis de tipo estático, nos permite saber los métodos que tiene cierta entidad, y además nos provee herramientas simples, como ser la búsqueda de determinados strings a lo largo de todo el proyecto.

### 5.4. VSCode

Similar a lo que sucede con CLion, VSCode nos brinda herramientas de caracter similar. La elección de un IDE u otro está determinada meramente por una cuestión de gusto personal, ya que los dos funcionan muy bien en su labor, y son ampliamente utilizados en el mercado.

## 6. Conclusiones

Como conclusión nos gustaría afirmar que la experiencia fue muy positiva, debido a que pudimos desarrollar un trabajo a mediano plazo y en equipo de forma satisfactoria, a pesar de que tuvimos dificultades obvias del trabajo, ya que parte de nosotros nunca había desarrollado un trabajo de esta envergadura. Aprendimos también, que es muy importante pensar en el problema, en la solución, y en las posibles complicaciones de cada implementación, antes de realizarla, debido a que cada línea de código se traduce en tiempo, y si posteriormente vemos que no era viable la solución por no haberla planeado correctamente, estamos perdiendo una de las cosas más valiosas: tiempo. Esto nos sucedió en algunas partes del trabajo: perder tiempo que hubiéramos necesitado al final para estar más tranquilos.