



TALLER DE PROGRAMACIÓN I  
(75.42) CURSO VEIGA

# Informe del proyecto - Duck Game

2do Cuatrimestre 2024

Ascencio Felipe Santino	Zielonka Axel
110675	110310

## 1. Introducción

Este informe presenta el trabajo desarrollado por cada integrante del proyecto. Se describen las herramientas utilizadas y se incluyen reflexiones sobre los desafíos enfrentados, el alcance final y recomendaciones para futuros desarrollos en este entorno.

## 2. Listado de Tareas Realizadas

### 2.1. Ascencio Felipe Santino

**Responsable del área del cliente y las pruebas**

#### Semana 1 (15-10-2024 / 22-10-2024)

Instalación y configuración de dependencias 'SDL' en el repositorio.

Diseño de la lógica de la interfaz gráfica para el usuario.

#### Semana 2 (22-10-2024 / 29-10-2024)

Configuración de procedimientos de compilación y ejecución (mediante 'CMake').

Implementación de la clase 'Parseador'.

Parseo de todos los sprites del juego.

Creación/obtención de imágenes, mapas, fuente de texto, sonidos y música del juego.

#### Semana 3 (29-10-2024 / 5-11-2024)

Implementación de la clase 'Controlador', 'Dibujador' y 'Handler'.

Implementación de los 'Threads' enviador y receptor del cliente.

Implementación del dibujo del pato.

Implementación de la animación del pato.

#### Semana 4 (5-11-2024 / 12-11-2024)

Implementación del dibujo del mapa.

Implementación del dibujo del tablero de puntuaciones.

Añadido de música de fondo.

Añadido de sonidos (disparos y 'quack').

Refactoring general de todo el código.

#### Semana 5 (12-11-2024 / 19-11-2024)

Reestructuración de carpetas del repositorio.

Lógica del dibujo de disparos.

Lógica del dibujo de la pantalla de victoria/derrota.

Creación de scripts y documentación.

Arreglo de bugs.

Preparación para la primera 'Release'.

### **Semana 6 (19-11-2024 / 26-11-2024)**

- Corrección de bugs visuales.
- Redacción de documentación.
- Testing general del juego.
- Realización de diagramas técnicos (UML).

### **Semana 7 (26-11-2024 / 03-12-2024)**

- Implementación de los 'GTest' (Tests Unitarios de comunicación).
- Corrección de los últimos bugs visuales.
- Emprolijado final de código mediante a 'Clang-format', 'CppCheck' y 'CppLint'.
- Preparativos finales para la última Release.

## **2.2. Zielonka Axel**

### **Responsable del área del servidor y los protocolos**

### **Semana 1 (15-10-2024 / 22-10-2024)**

- Estructurado inicial del repositorio.
- Idear el sistema de clases a seguir, herramientas a utilizar y hooks del repositorio.
- Diseño de la clase 'Pato' y Clase 'Arma'.

### **Semana 2 (22-10-2024 / 29-10-2024)**

- Configuraciones de 'CPPLINT', 'clang-format' y 'pre-commit'.
- Implementación de métodos de la clase 'Pato' y 'Mapa'.

### **Semana 3 (29-10-2024 / 5-11-2024)**

- Implementación de los 'Threads' enviador, recibidor y aceptador del servidor.
- Implementación del 'Gameloop'.

### **Semana 4 (5-11-2024 / 12-11-2024)**

- Diseño e implementación de los protocolos de comunicación 'servidor-cliente' (desde el lado del servidor, y el lado del cliente).
- Implementación de la clase 'Arma'.
- Refactoring general de todo el código.

### **Semana 5 (12-11-2024 / 19-11-2024)**

- Implementación de la clase 'AK47', 'Shotgun', 'Magnum'.
- Implementación de condición de 'victoria/derrota'.
- Arreglo de bugs funcionales.
- Creación de la documentación.
- Preparación para la primera 'Release'.

### Semana 6 (19-11-2024 / 26-11-2024)

Implementación de la 'Pew Pew Laser' y el 'Sniper'.  
Implementación de la lógica de rondas.  
Implementación de los 'spawn-places'.  
Implementación del 'loot' de las cajas.  
Implementación final de los 'Cheats'.  
Finalización de protocolos.  
Realización de diagramas técnicos (UML).

### Semana 7 (26-11-2024 / 03-12-2024)

Implementación de la lógica de múltiples partidas.  
Solución de 'Leaks' de memoria.  
Corrección de los últimos bugs jugables.  
Preparativos finales para la última Release.

## 3. Herramientas Utilizadas

El proyecto ha sido desarrollado en C++ bajo el estándar POSIX. Las herramientas y tecnologías utilizadas incluyen:

- **Sistema Operativo:** Distribuciones *Ubuntu 24.04* o *Xubuntu 24.04*.
- **Compilador:** GNU Compiler Collection (GCC) v2.
- **Lenguaje de Programación:** C++17.
- **Entorno de Desarrollo Integrado (IDE):** Visual Studio Code.
- **Bibliotecas Gráficas:** SDL2 y SDL2pp.
- **Framework de Pruebas:** GTest.
- **Herramienta de Chequeo de Memoria:** Valgrind.
- **Formato de Configuración:** YAML (versión 0.8).
- **Scripts de Automatización:** Shell Scripting.
- **Control de Versiones:** GitHub.

## 4. Reflexiones

### 4.1. Puntos difíciles en el desarrollo

Uno de los principales desafíos fue la configuración de las dependencias de SDL2, que consumió un tiempo considerable durante la primera semana del proyecto. Además, la creación de mapas mediante Tiled y la búsqueda de música y efectos de sonido fueron tareas complejas, aunque nos permitieron adquirir nuevas habilidades que resultaron fundamentales para el desarrollo del juego.

## 4.2. Alcance del Proyecto

Dado que el desarrollo del juego fue realizado por dos integrantes en lugar de los cuatro planeados originalmente, se establecieron ciertos ajustes en el alcance del proyecto. Las principales características que quedaron fuera del desarrollo actual son las siguientes:

- Implementación de un lobby.
- Desarrollo de un editor de niveles.
- Ampliación de la variedad de armas.

Implementar estas funcionalidades en futuras versiones agregaría valor y mejoraría considerablemente la experiencia del usuario final.

## 4.3. Recomendaciones para Futuros Cuatrimestres

A pesar de nuestra experiencia previa en la mayoría de las tecnologías utilizadas, algunas herramientas eran completamente nuevas para nosotros. Por ello, consideramos que sería valioso ofrecer clases adicionales sobre los siguientes temas en futuros cuatrimestres:

- **SDL2:** Aunque se brindó una introducción básica y algunos tutoriales, una capacitación más profunda en esta biblioteca nos habría facilitado el desarrollo de funcionalidades gráficas y de sonido.
- **GTest:** Una clase específica sobre el uso de GTest sería de gran ayuda para familiarizar a los estudiantes con la estructura y los conceptos de esta herramienta de pruebas antes de implementarla en el proyecto.
- **Scripts de Shell:** Aunque aprendimos su uso de manera autodidacta sin mayores inconvenientes, una clase que explore las aplicaciones de los scripts de Shell a mayor escala permitiría conocer su verdadero potencial y eficiencia en entornos de desarrollo profesional.

## 5. Conclusión

Estamos muy satisfechos con el juego desarrollado. A pesar de las limitaciones de equipo, logramos obtener un resultado sólido y gratificante, que además nos permitió aprender significativamente sobre áreas en las que teníamos poca o nula experiencia previa.

Agradecemos de manera especial a Martín Di Paola, quien fue nuestro asesor y nos brindó un apoyo fundamental a lo largo del proyecto, ayudándonos a superar varios desafíos y a mejorar nuestro desarrollo.

Ascencio Felipe Santino y Zielonka Axel