

Informe Final - Proyecto Battleship

Resumen

El proyecto **Battleship** es una implementación de Batalla Naval multijugador en C++ con arquitectura cliente-servidor. Permite a los usuarios registrarse, iniciar sesión, configurar tableros y jugar partidas 1v1 en red. El sistema soporta servidores en Windows y Linux, con una interfaz de consola y comunicación TCP usando un protocolo personalizado (BSS/1.0).

Objetivos

- Desarrollar un juego funcional de Batalla Naval en red.
- Implementar autenticación y emparejamiento de jugadores.
- Soportar múltiples plataformas (Windows y Linux).
- Proveer una interfaz de usuario clara y un sistema de logs robusto.

Metodología

1. **Diseño:**
 - Arquitectura cliente-servidor con TCP.
 - Protocolo BSS/1.0 para comunicación.
 - Clases modulares (Board, GameManager, Network, UI).
2. **Implementación:**
 - Cliente: Windows, con Winsock.
 - Servidor: Windows (Winsock) y Linux (POSIX).
 - Uso de hilos y mutex para concurrencia.
3. **Pruebas:**
 - Pruebas unitarias de colocación de barcos y disparos.
 - Pruebas de integración para autenticación y partidas.
 - Simulación de desconexiones y errores de red.

Resultados

- **Funcionalidad:**
 - Autenticación segura con almacenamiento en usuarios.txt.
 - Configuración de tableros manual y aleatoria.
 - Partidas 1v1 con turnos alternados y notificaciones en tiempo real.
 - Interfaz de consola con colores para barcos, aciertos y fallos.

- **Rendimiento:**
 - Soporta múltiples clientes.
 - Latencia baja en comunicación cliente-servidor.
- **Compatibilidad:**
 - Cliente funcional en Windows.
 - Servidor funcional en Windows y Linux.
- **Logs:** registro detallado de eventos en log.log para depuración.

Dificultades Encontradas

- **Sincronización:** problemas iniciales con acceso concurrente a datos, resueltos con mutex.
- **Desconexiones:** manejo de clientes desconectados durante partidas, implementado con sockets no bloqueantes y limpieza de recursos.
- **Compatibilidad:** adaptación del servidor a Linux requirió cambios en manejo de sockets y señales.

Conclusiones

El proyecto cumple con los objetivos establecidos, ofreciendo un juego funcional y robusto. La arquitectura modular facilita futuras mejoras, como agregar persistencia de partidas o una interfaz gráfica. La compatibilidad multiplataforma asegura flexibilidad en el despliegue.

Recomendaciones

- **Mejoras:**
 - Agregar reintentos automáticos en conexiones fallidas.
 - Implementar un sistema de reconexión para partidas interrumpidas.
 - Añadir una interfaz gráfica (ej. usando SDL o Qt).
- **Optimizaciones:**
 - Reducir el uso de hilos mediante un modelo basado en eventos.
 - Comprimir datos serializados para tableros.
- **Pruebas:**
 - Realizar pruebas de carga con múltiples clientes simultáneos.
 - Validar seguridad del protocolo contra inyecciones.