Asignatura Redes
INFO
Draft:230901

C. Lazo
UACh
Septiembre 2023

TAREA de REDES
Desarrollo de un Juego de Batalla Naval en Línea utilizando el protocolo de transporte UDP, en un modelo cliente servidor

Introducción:

El juego de batalla naval, también conocido como *battleship*, es un juego de mesa estratégico para dos jugadores que simula una batalla naval entre flotas enemigas. El objetivo principal del juego es hundir todos los barcos del oponente antes de que él hunda los tuyos. El juego se juega tradicionalmente en papel y lápiz, pero también se ha adaptado a juegos de mesa electrónicos y versiones en línea multijugador como el que vamos a desarrollar en esta tarea.

Aquí están los elementos clave de un juego de batalla naval:

1. Tablero de Juego: El tablero es una cuadrícula rectangular compuesta por casillas, generalmente de 20x20. Cada jugador tiene su propio tablero, uno para ubicar sus barcos y otro para registrar los ataques del oponente.

2. Barcos: Cada jugador tiene una flota de barcos que varían en tamaño y forma. Los barcos para este juego serán un Submarino (3 casillas), un Destructor (2 casillas) y un Barco Patrulla (1 casillas). El objetivo es colocar los barcos en el tablero de manera estratégica sin que el oponente conozca su ubicación.

3. Ataques: Los jugadores se turnan para realizar ataques al tablero del o los oponentes indicando una coordenada (x,y). Si el ataque alcanza uno o más barcos enemigos, se considera un "impacto". El objetivo es hundir completamente los barcos enemigos.

4. Ganar el Juego: El juego continúa hasta que uno de los jugadores haya hundido todos los barcos del o los oponentes. El jugador que lo logra primero es el ganador.

5. Estrategia: El juego de batalla naval requiere de estrategia y deducción. Los jugadores deben adivinar la ubicación de los barcos enemigos en función de los ataques anteriores y, al mismo tiempo, proteger sus propios barcos. También es importante seguir patrones lógicos para maximizar las posibilidades de encontrar y hundir los barcos enemigos.

El juego de batalla naval se ha convertido en un juego clásico y es disfrutado por personas de todas las edades en todo el mundo. Además de su versión de juego de mesa, existen numerosas adaptaciones digitales que permiten a los jugadores enfrentarse en línea, contra una inteligencia artificial o un bot aleatorio.

Funcionamiento del Juego en Línea:

1. Servidor Central: Se establecería un servidor central que actúa como el "anfitrión" del juego. Este servidor es responsable de coordinar y gestionar todas las partidas en curso, así como de administrar la comunicación entre jugadores.

2. Partidas Multijugador: Los jugadores se conectan al servidor central para unirse a partidas multijugador. Pueden crear sus propias partidas o unirse a partidas existentes.

3. Comunicación UDP: Se utiliza el protocolo UDP para la comunicación entre jugadores durante la partida. UDP es ideal para juegos en línea debido a su velocidad y bajo costo en comparación con TCP que puede introducir retrasos.

4. Tablero y Barcos: Cada partida multijugador tiene su propio tablero de juego y una flota de barcos para cada jugador. Los jugadores pueden colocar estratégicamente sus barcos en su tablero antes de que comience la partida.

5. Ataques en Tiempo Real: Los jugadores pueden realizar ataques en tiempo real contra los tableros de sus oponentes enviando coordenadas de ataque mediante mensajes UDP. Por ejemplo, un jugador podría enviar un mensaje UDP al servidor con la coordenada del ataque (por ejemplo, "x,y").

6. Sincronización y Respuestas: El servidor central coordina la recepción de ataques y actualiza el estado de los tableros de los jugadores en tiempo real. Los jugadores reciben respuestas sobre el resultado de sus ataques y ven los resultados en sus propios tableros.

7. Ganar el Juego: El juego continúa hasta que un jugador haya hundido todos los barcos de su oponente. El servidor central verifica y anuncia al ganador de la partida.

8. Conexión Persistente: Aunque UDP no es una conexión persistente como TCP, se podría implementar una capa de gestión de conexión en el juego para mantener a los jugadores conectados y sincronizados durante la partida.

9. Interfaz de Usuario: La interfaz de usuario en el cliente mostraría el tablero del jugador, el estado de la partida y la comunicación en tiempo real con otros jugadores.

10. Lógica del Juego en el Servidor: El servidor central sería responsable de implementar la lógica del juego, validar las acciones de los jugadores y garantizar la sincronización adecuada entre los tableros y jugadores.

Descripción de la tarea:

En esta tarea, se les solicita diseñar e implementar soluciones de la capa de aplicación (cliente y servidor) para un juego de batalla naval en el que un servidor gestionará múltiples partidas con jugadores. Esta tarea tiene como objetivo que comprendan cómo se establece la comunicación entre:

1. Un servidor y múltiples clientes en un entorno de juego, para probar la solución desarrollada se aconseja que se provea de un bot (puede ser probabilístico) que permita jugar directamente con el servidor en modo monousuario.
2. Un servidor y otro servidor para intercambiar información en modo multiplayer.

Enunciado del Protocolo:

Este protocolo describe la comunicación entre un servidor y uno o varios clientes del juego de batalla naval. El juego involucra la colocación estratégica de barcos en un tablero y el intercambio de ataques entre jugadores hasta que se cumplan las condiciones de victoria o finalización. A continuación, se detallan los aspectos clave del protocolo:

1. Conexión Inicial:
   - El cliente se conectará al servidor proporcionando la dirección IP y el puerto del servidor.
   - El servidor aceptará la conexión entrante y confirmará la conexión con el cliente.

2. Inicio de Partida:
   - El cliente solicitará al servidor iniciar una partida.
   - El servidor confirmará la solicitud y permitirá el inicio de la partida.

3. Flujo del Juego:
   - El juego se desarrollará mediante un ciclo de turnos alternos entre los jugadores.
   - Los jugadores realizarán acciones como ataques, que serán enviadas al servidor para su validación.
   - El servidor validará las acciones de los jugadores, actualizará el estado del juego (incluyendo la ubicación de los barcos y los resultados de los ataques) y comunicará estas actualizaciones a los clientes.
   - El ciclo de juego continuará hasta que se cumplan las condiciones de victoria o finalización.

4. Finalización de Partida:
   - Cuando se cumplan las condiciones de victoria o finalización, el cliente notificará al servidor.
   - El servidor notificará a todos los clientes los resultados de la partida.

5. Desconexión:
   - El cliente podrá desconectarse en cualquier momento.
   - El servidor confirmará la desconexión y ajustará el estado del juego si es necesario.

Propuesta del Diagrama de Clases:

Para facilitar la comprensión de la estructura del juego, se proporciona un diagrama de clases que representa las clases y sus relaciones:

Clase: Jugador
Atributos:
- nombre: String
Métodos:
- realizarAccion(coordenada: Coordenada): void

Clase: Barco
Atributos:
- tipo: String
- estado: String
Métodos:
- recibirAtaque(): void
- hundir(): void

Clase: Tablero
Atributos:
- casillas: Matriz de Casilla
Métodos:
- colocarBarco(barco: Barco, coordenadas: Lista de Coordenadas): boolean
- realizarAtaque(coordenada: Coordenada): boolean

Clase: Servidor
Atributos:
- jugadoresConectados: Lista de Jugadores
Métodos:
- iniciarPartida(): void
- finalizarPartida(): void

Clase: Cliente
Atributos:
- nombre: String
- servidor: Servidor
Métodos:
- conectarAServidor(servidor: Servidor): void
- desconectar(): void
- jugarTurno(): void

El diagrama de clases incluye las siguientes clases principales:

- `Jugador`: Representa a un jugador y sus acciones en el juego.
- `Barco`: Modela las características y estados de los barcos.
- `Tablero`: Gestiona el tablero de juego y las interacciones relacionadas con la ubicación de barcos y ataques.
- `Servidor`: Administra la lógica del juego, incluyendo el ciclo de turnos y la gestión de múltiples partidas.
- `Cliente`: Facilita la comunicación entre el jugador y el servidor.

Diagramas de Secuencia Cliente-Servidor:

A continuación, se presentan diagramas de secuencia que ilustran la interacción entre el cliente y el servidor en diferentes situaciones clave:

Secuencia: Funcionamiento del Servidor

```
Servidor -> Jugador: Esperando conexiones entrantes

loop hasta que se reciba una conexión
    Jugador -> Servidor: Solicitud de conexión
    Servidor -> Jugador: Confirmación de conexión
    Jugador -> Servidor: Inicio de partida
    Servidor -> Jugador: Confirmación de inicio de partida
    loop mientras el juego está en curso
        Jugador -> Servidor: Acción del jugador (por ejemplo, ataque)
        Servidor -> Jugador: Validación de la acción del jugador
        Servidor -> Jugador: Actualización del estado del juego
        Jugador -> Servidor: Fin de turno o nueva acción
    end loop
    Jugador -> Servidor: Finalización de la partida
    Servidor -> Jugador: Notificación de resultados de la partida
    Jugador -> Servidor: Desconexión
    Servidor -> Jugador: Confirmación de desconexión
end loop
```

1. Secuencia: Inicio de Partida

```
Cliente -> Servidor: Conectar al servidor
Servidor -> Cliente: Confirmación de conexión
Cliente -> Servidor: Solicitar inicio de partida
Servidor -> Cliente: Confirmación de inicio de partida
loop mientras el juego está en curso
    Cliente -> Servidor: Realizar un ataque en la coordenada (x, y)
    Servidor -> Cliente: Validar el ataque
    Servidor -> Cliente: Actualizar el estado del juego
end loop
Cliente -> Servidor: Finalizar partida
Servidor -> Cliente: Notificar resultados de la partida
Cliente -> Servidor: Desconectar
Servidor -> Cliente: Confirmación de desconexión
```

2. Secuencia: Funcionamiento del Servidor

```
Servidor -> Jugador: Esperando conexiones entrantes

loop hasta que se reciba una conexión
    Jugador -> Servidor: Solicitud de conexión
    Servidor -> Jugador: Confirmación de conexión
    Jugador -> Servidor: Inicio de partida
    Servidor -> Jugador: Confirmación de inicio de partida
    loop mientras el juego está en curso
        Jugador -> Servidor: Acción del jugador (por ejemplo, ataque)
```

```
        Servidor -> Jugador: Validación de la acción del jugador
        Servidor -> Jugador: Actualización del estado del juego
        Jugador -> Servidor: Fin de turno o nueva acción
    end loop
    Jugador -> Servidor: Finalización de la partida
    Servidor -> Jugador: Notificación de resultados de la partida
    Jugador -> Servidor: Desconexión
    Servidor -> Jugador: Confirmación de desconexión
end loop
```

Funcionamiento del Servidor: El servidor realizará las siguientes funciones:

- Esperará conexiones entrantes de los clientes.
- Administra múltiples partidas simultáneas, asignando jugadores a partidas disponibles.
- Implementa la lógica del juego, incluyendo la validación de acciones de los jugadores y el seguimiento del estado del juego.
- Proporciona respuestas a las acciones de los jugadores y notifica a los clientes sobre eventos relevantes en el juego.
- Maneja la desconexión de los clientes y garantiza la integridad del juego.

3. Secuencia: Funcionamiento del Cliente en un Juego de Batalla Naval

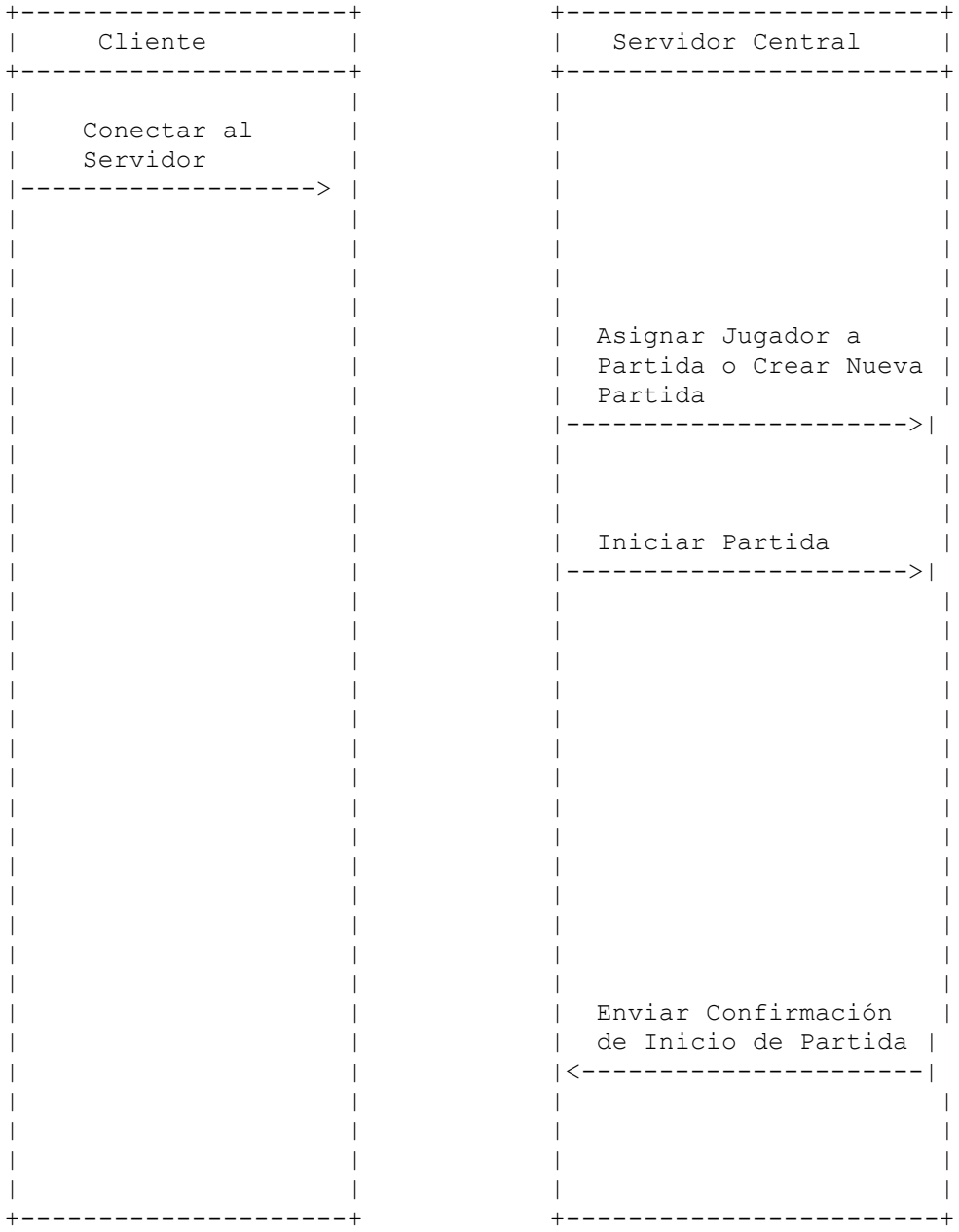
```
Cliente -> Servidor: Conectar al servidor
Servidor -> Cliente: Confirmación de conexión
Cliente -> Servidor: Solicitar inicio de partida
Servidor -> Cliente: Confirmación de inicio de partida
loop mientras el juego está en curso
    Cliente -> Servidor: Realizar una acción (por ejemplo, un ataque)
    Servidor -> Cliente: Respuesta del servidor (validación de la acción y actualización del estado del juego)
    Servidor -> Cliente: Estado actualizado del juego
    Cliente -> Servidor: Fin de turno o nueva acción
end loop
Cliente -> Servidor: Finalizar partida
Servidor -> Cliente: Notificación de resultados de la partida
Cliente -> Servidor: Desconectar
Servidor -> Cliente: Confirmación de desconexión
```

Funcionamiento del Cliente: El cliente desempeñará las siguientes funciones:

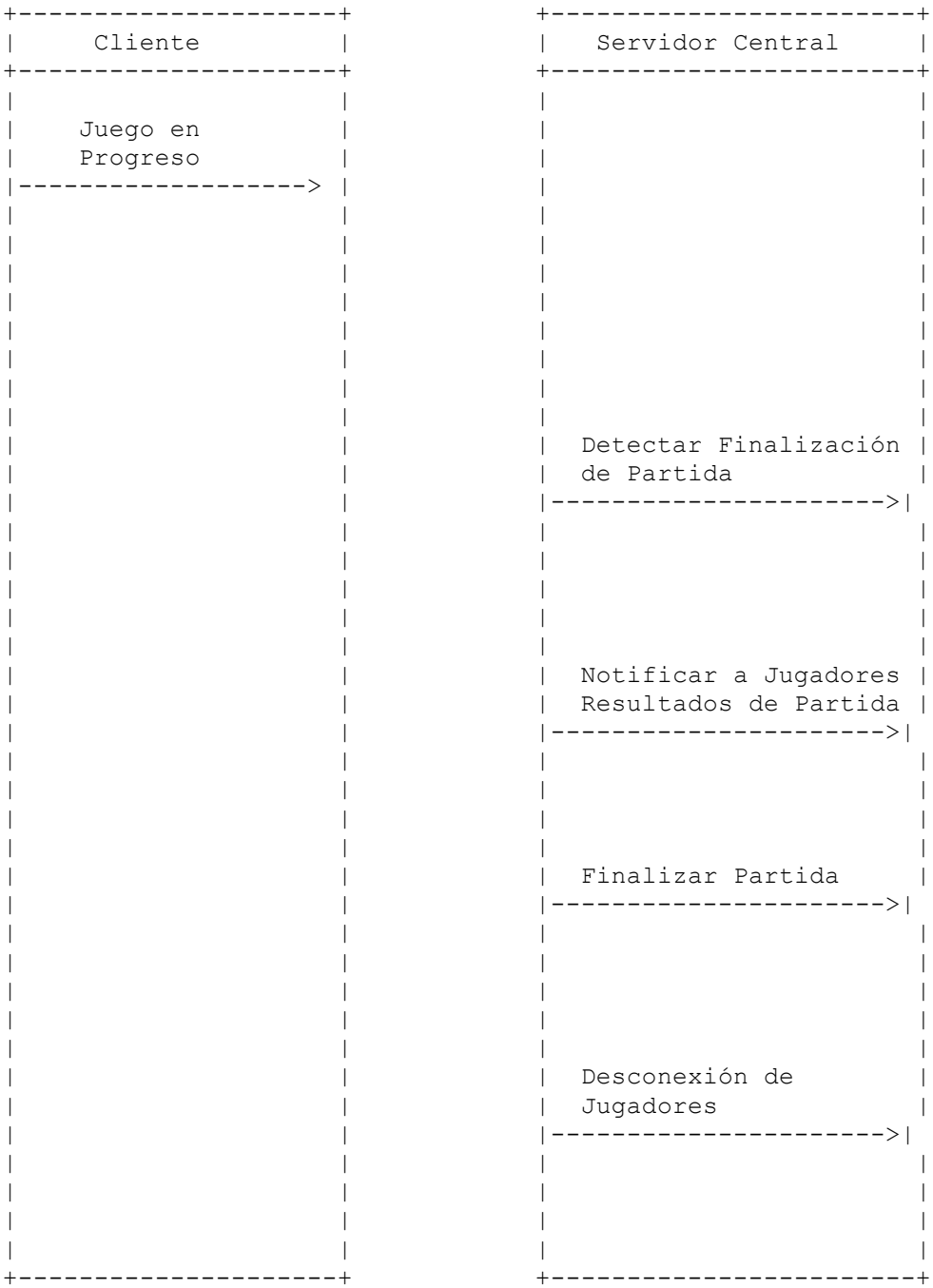
- Se conectará al servidor proporcionando la dirección IP y el puerto del servidor.
- Solicitará al servidor iniciar una partida.
- Participará en el juego realizando acciones, como ataques.
- Enviará acciones al servidor para su validación.
- Recibirá respuestas del servidor, incluyendo actualizaciones del estado del juego y resultados de acciones.
- Notificará al servidor cuando se cumplan las condiciones de victoria o finalización.
- Podrá desconectarse en cualquier momento.

4. Este diagrama de secuencia representa el proceso de inicio de juego desde la perspectiva del cliente y el servidor central. El cliente se conecta al servidor, y el servidor asigna al cliente a una partida existente o crea una nueva partida si no hay partidas disponibles. Luego, el servidor inicia la partida y envía una confirmación al cliente para que comience a jugar.

Ten en cuenta que este diagrama es una simplificación y podría haber interacciones adicionales dependiendo de la implementación específica del juego.



5. diagrama de secuencia que muestra cómo podría ser el proceso de finalización de partida en un juego de batalla naval en línea usando el protocolo UDP. Este diagrama representa la interacción entre un cliente y un servidor centra



Bibliografía

https://es.wikipedia.org/wiki/Batalla_naval_(juego)

http://es.battleship-game.org/

Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980, <https://www.rfc-editor.org/info/std6>.

Saldana, J; Suznjevic, M. , Tutorial: Traffic of Online Games https://datatracker.ietf.org/meeting/87/materials/slides-87-tsvarea-1