Hundir la flota Sockets

Contenido

1 Enunciado

Se debe implementar un hundir la flota usando sockets y la arquitectura cliente-servidor.

Al servidor podrán conectarse múltiples clientes.

El cliente solo se conectará a un único servidor.

Cuando el cliente se conecte al servidor deberá enviarle su usuario y contraseña, en caso de que no exista el servidor le informará y cerrara la conexión.

Un mismo usuario solo puede estar conectado una vez al servidor, se debe tratar esta situación.

Un usuario conectado tendrá varias opciones:

1. Empezar nueva partida con otro usuario (verá una lista de los clientes a elegir)
2. Ver partidas ya terminadas y poder reproducirlas paso a paso.
3. Ver resto de partidas sin terminar en la que está y disparar en las que es su turno.
4. Ver resto de partidas sin terminar y renunciar en cualquier momento, dando la victoria por abandono a su contrincante.

El servidor se encarga de la seguridad y lógica de negocio, independientemente de que se tenga que implementar toda o parte de ella también en los clientes. Nótese que puede haber temas de sincronismo en la aplicación.

Ante cada cambio en la situación de la partida, es decir, cada disparo de cada uno de los jugadores, el servidor será el encargado de persistir el estado de la partida, de esta forma, en caso de caída del servidor o de los clientes se podría recuperar sin problema el estado.

El servidor solo tiene que enviar la información mínima y necesaria a los clientes para poder jugar la partida. Ejemplo: en ningún caso el servidor enviará a un usuario la información de los barcos del oponente.

En caso de que los dos jugadores de una partida estén conectados a la vez y uno de ellos haga un movimiento, este le debe llegar inmediatamente al otro jugador.

Gráfico

1. Lista de partidas sin terminar en las que el contrincante SÍ está conectado.
2. Lista de partidas sin terminar en las que el contrincante NO está conectado y es el turno del usuario hacer movimiento.

3. Lista de partidas sin terminar en las que el contrincante NO está conectado y es el turno del contrincante hacer movimiento.

Para el punto 1, el usuario verá la lista de partidas de color verde cuando sea su turno y de color rojo cuando sea el turno del contrincante.

Para el punto 2, si el usuario dispara esta partida ya no debería estar en esta lista.

Para el punto 2 y 3, si en cualquier momento uno de los jugadores de esas partidas se conectan el cliente será informado y se sacará las partidas de esta lista para ponerlas en la lista 1 avisando al jugador de ello con un mensaje por pantalla.

Ocurre lo mismo para el punto 1, si alguno de los jugadores de esas partidas se desconecta se avisará al cliente para que lo pase a la lista 2 o 3.

Se deberán poder colocar los barcos de forma gráfica.

Se deberá poder disparar en una casilla de forma gráfica.

2 Calificación

A la hora de calificar la práctica se podrá realizar preguntas al alumno, desde explicaciones de código fuente, razones, etc. Hasta cambios de código fuente (cambiar funcionalidades, nuevas aplicaciones), etc. Tales preguntas tendrán repercusión en la nota de cada apartado siguiente y dependiendo de los fallos podría llevar a un 0 y a suspender la práctica. Pues ciertos fallos implican que la práctica no es realmente del alumno (por copia o recibir demasiada "ayuda") o no ha aprendido los CE que se califican.

Los puntos de cada apartado son "hasta X punto/s".

Código fuente (hasta 7,25):

1. Empezar nueva partida con otro usuario (verá una lista de los clientes a elegir) **(1 punto)**
2. Ver partidas ya terminadas y poder reproducirlas paso a paso. **(1 punto)**
3. Ver resto de partidas sin terminar en la que está y disparar en las que es su turno. **(1 punto)**
4. Ver resto de partidas sin terminar y renunciar en cualquier momento, dando la victoria por abandono a su contrincante. **(1 punto)**
5. Usuario conectado una sola vez **(1 punto)**
6. Persistir la información en cada paso **(1 punto)**
7. Recibir disparo del otro jugador en tiempo real (**1 punto**)
8. Información mínima necesaria en cada cliente (**1,25 puntos**)

Documentación (hasta 1,75):

En un documento de texto se debe entregar: **(1 punto)**

9. Razón de porque has elegido usar el tipo de socket que usas en el código fuente.
10. Que ventajas y desventajas tendría usar una conexión directa entre clientes en vez de usar la actual arquitectura cliente-servidor

Adicionalmente:

11. Documentación de cómo funciona el programa
12. Mensajes intercambiados entre cliente y servidor junto con un diagrama de estados **(0,75 puntos)**

3 Entrega

Se debe entregar todo el código fuente, un proyecto para cliente y otro para servidor. A su vez se debe entregar dos .jar, uno por el cliente y otro por el servidor.

Si se usa basa de datos (MySQL) el Script inicial con datos, si se usa otro tipo de persistencia los ficheros correspondientes.

Documento .docx o .pdf