

Fase 4 WebSockets

Coffee Time!

Juegos en Red URJC

Grupo G



Integrantes:

Pablo Álvarez de Lara Fernández
Daniel Mayoral Fernández-Baíllo
Javier Picado Hijón

Índice

1. Introducción.....	3
2. Inclusión de WebSockets.....	3
3. Diagrama de clases actualizado.....	4
4. Tutorial en vídeo.....	5

1.Introducción

Este documento sirve como memoria de la cuarta fase del trabajo desarrollado para la clase de Juegos en Red. En esta parte de la práctica se pedía implementar un back-end que permita a distintos jugadores conectarse por medio de la red para poder jugar partidas a través de la red.

Para poder implementar esta función, se debía hacer uso de la tecnología WebSockets, por lo que esta memoria cubrirá la inclusión de esta en el proyecto, explicando de manera detallada tanto el protocolo desarrollado sobre WebSockets como la nueva organización de clases del juego.

También se incluye un apartado donde se adjunta un enlace a un vídeo explicativo sobre el juego que muestra los aspectos más relevantes de este como la creación de las partidas o donde se hace uso del protocolo WebSockets.

2.Inclusión de WebSockets

Como se ha mencionado anteriormente, para esta fase era necesaria la utilización del protocolo WebSockets para poder implementar las funcionalidades en línea del juego. Para ello se ha realizado un protocolo por el que se ha modificado parte del juego para poder soportar la comunicación entre los distintos usuarios.

El protocolo de WebSocket maneja diferentes mensajes, que son enviados desde el host hacia los clientes, o desde el cliente hacia el host. Los datos que se envían mediante WebSocket son aquellos que tienen que ir en tiempo real entre ambos jugadores. Estos datos se pueden diferenciar en tres tipos, los que se mandan durante la partida, los que se envían cuando esta finaliza y los que se mandan antes de esta.

Los datos enviados antes de la partida sirven para sincronizar la inicialización de esta para ambos jugadores, enviándose desde el host a cada uno de estos.

Durante la partida, se envía la posición de cada jugador en cada momento, para que el otro pueda ver donde se encuentra su rival. Estos datos son enviados tanto por el host como por el cliente. También se envían los objetos que se crean y si un jugador recoge uno de esos objetos. Lo primero se manda desde el host al cliente, para coordinar que a ambos jugadores se les crea el mismo objeto. Lo segundo se envía desde el host al cliente o al revés, dependiendo de quien recoja el objeto. Esto se hace para eliminar dicho objeto de la pantalla del otro jugador.

Otro dato que se envía durante la partida es el tiempo del cronómetro del host, para que si hay un desfase entre el reloj del cliente y del host, este se corrija y ambos jueguen el mismo tiempo.

Al finalizar la partida se manda la puntuación del cliente al host, para que este compare dicha puntuación con la suya propia y determine quién es el ganador. Tras esto, se envía el ganador desde el host a ambos jugadores, para que estos inicien la escena de victoria a la vez y puedan ver quién ha ganado.

3. Diagrama de clases actualizado

La introducción de estos cambios en el juego, generó por consiguiente algunas modificaciones del diagrama de clases del mismo. Estas modificaciones se pueden resumir en la inclusión de una clase para el manejo de los mensajes recibidos desde la sesión de WebSockets. El propio diseño del juego ha inducido a la creación de una nueva clase para controlar los lobbies, que sirven como salas para conectar a dos personas a una partida.

Además, la inclusión de los lobbies ha derivado en la modificación del chat, pasando este a una nueva escena llamada Lobby. Este cambio se ha realizado para permitir a los jugadores, que estén a la espera de iniciar una partida, comunicarse con otros jugadores e intercambiar los códigos de partidas. Otro cambio que ha propiciado esta nueva clase es la creación de una nueva tabla en la base de datos, donde se almacena el código de la partida, las personas conectadas, las personas listas para empezar la partida, el id de ambos jugadores.

El diagrama de clases de la *Figura 1* muestra cómo se realiza la nueva conexión entre las entidades y los controladores con las escenas en el modelo de API REST.

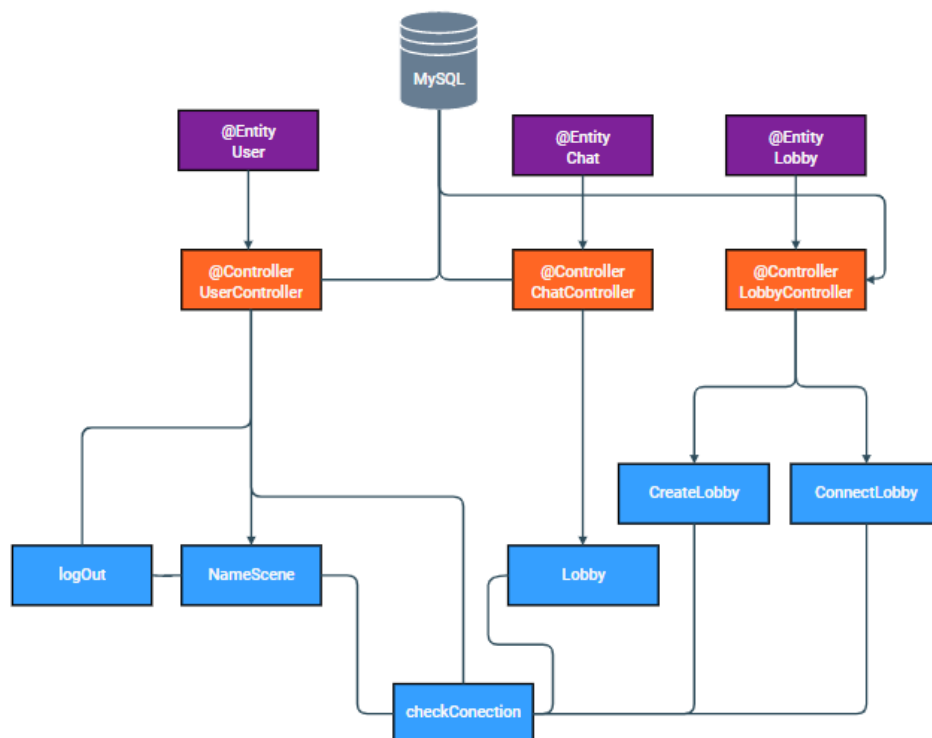


Figura 1. Diagrama Clases Api Rest

El diagrama de clases de WebSocket muestra cómo se relaciona el host con el servidor, el cliente con el servidor y el servidor con ambos (*Figura 2*). La conexión tanto desde el host como del cliente al servidor se realiza mediante la escena PlayOnline. Mientras que cómo manejar los mensajes que le llegan al servidor se realiza mediante la clase ChatHandler.

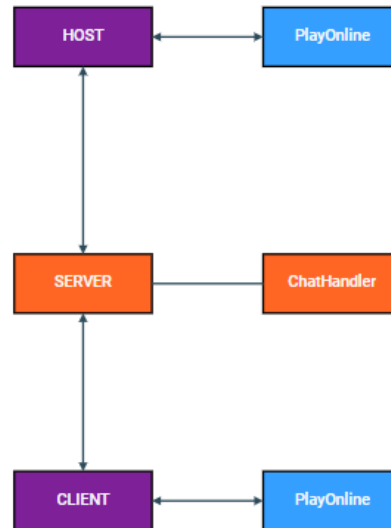


Figura 2. Diagrama Clases WebSocket

4. Tutorial en vídeo

Finalmente, se adjunta un breve vídeo explicativo donde uno de los miembros del equipo de desarrollo muestra los aspectos clave del juego tras las modificaciones realizadas. Estos aspectos son, la creación de una partida online, la unión a una partida ya creada, las partes en las que se utiliza la implementación de una API REST y las partes en las que se implementa el protocolo WebSockets para poder realizar una comunicación entre los distintos usuarios.

Enlace:

https://www.youtube.com/watch?v=phiw4yo-_HA&ab_channel=AraelPicado