Client - Server

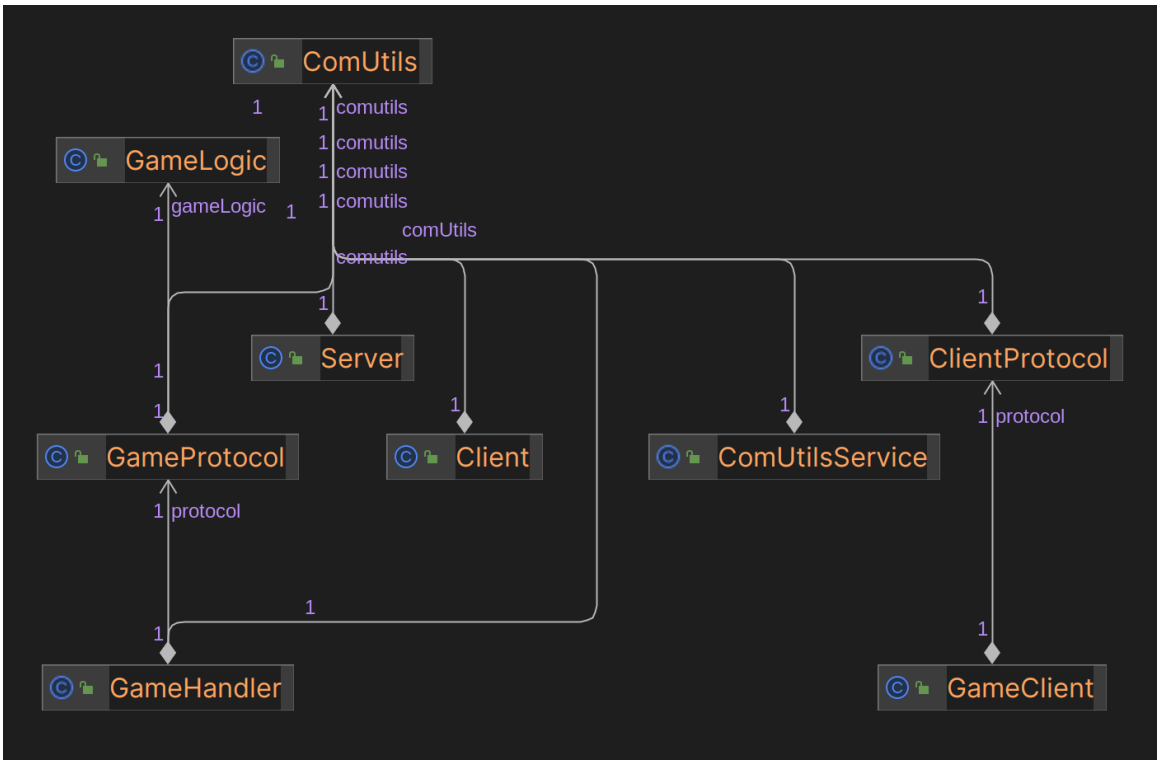
PRÁCTICA 1

Software Distribuït
Eshaan Mittal; NIUB 20620983
Paula Bezares Cano; NIUB 20393015

Desarrollo de la Práctica

Para desarrollar esta aplicación cliente-servidor, trabajamos juntos durante todo el proceso de desarrollo, dividiendo el trabajo por fases. Donde cada fase era una parte del protocolo, empezando con “Hello - Ready”, seguido por “Play - Admit” y así para implementar toda la funcionalidad del juego. Debido a trabajar así, en cada momento los dos sabíamos cómo funcionaba cada implementación, lo que nos facilitaba ayudar al otro.

Diagrama de Clases



Este diagrama de clases muestra la estructura de un juego de tic-tac-toe cliente-servidor. Incluye clases para la lógica del juego (‘GameLogic’, ‘GameHandler’, ‘GameClient’), comunicación (‘ComUtils’, ‘GameProtocol’, ‘ClientProtocol’), y los roles de cliente (‘Client’) y servidor (‘Server’). Las clase clave de comunicación que se comparte entre cliente y servidor es ComUtils, usado por los dos ‘XXXHandlers’ y ‘XXXProtocols’ asegurando que la información se transmita según el protocolo establecido. Cada clase está conectada de manera uno a uno, reflejando una asociación directa entre las responsabilidades de cada componente del sistema.

Ejecución

Para poder compilar el proyecto, hay que ejecutar el comando mvn clean package en el package Práctica 1, y después ejecutar los .jar 's en los propios packages, Server y Client. (El ReadMe está incluido en el fichero Practica 1)

Sesión de Testing

La sesión de testing era muy útil para ayudarnos encontrar pequeños errores que tenía nuestro código debido a casos que no habíamos planteado.

- El error más común que encontramos era al recibir una acción por el lado del servidor. Antes, solo controlamos movimientos desconocidos que no eran entre 0 y 2. Lo que causaba que el código no funcionaba bien en el caso que recibamos una acción cómo “000” o “0 0”. Esto lo hemos solucionado intentando introducir todas las posibilidades que nos ocurrían para poder manejar bien el error.

- También nos salieron otros errores con algunos grupos debido a cambios en el protocolo, algunos flags tenían significados diferentes, lo que se solucionaba rápidamente cambiando los cases del switch.

Nos ayudó mucho esta sesión ya que nos mostró que había casos que no nos había ocurrido que por tanto no teníamos controlado en el código.

Modo 2 Jugadores

El modo 2 jugadores no es compatible con el protocolo que usamos nosotros para este proyecto.

1. Primer Jugador: Nuestro protocolo está implementado para que el cliente siempre juegue primero. Por tanto cuando intentamos crear una versión Client vs Client, los dos mueven primero, que causará errores si van a la misma casilla. Y así para cada jugada que viene después, ya que no hay ningún orden implementado.

2. Aceptar Clientes: A la hora de aceptar clientes, el servidor tendría que gestionar como empareja los clientes y tener una cola de clientes que están esperando para un oponente.

3. Lógica del Juego: También tendríamos que cambiar cómo funciona la lógica del juego por el lado del servidor. Cambiando que ahora solo es un entremedio para dos clientes y lo único que calcula es si alguien ha ganado o si es empate.

Conclusion

Durante esta práctica, hemos aprendido cómo funciona una aplicación básica de cliente-servidor y cómo se comunican. Además, la importancia de un protocolo bien definido que a la hora de crear debe ya tener pensado todas las posibilidades de la aplicación que queremos implementar.