Reto 2

Actualización de diseño e implementación del Triqui Distribuido

11/09/2012

UNIVERSIDAD EAFIT

Luisa Fernanda Querubín Osorio

Jorman Andrés Bustos Gómez

# Antecedentes del Reto2:

Durante el desarrollo de las clases y como material de apoyo en Eafit Interactiva, se ha entregado dos (2) versiones del triqui. La versión 1, es un Triqui Standalone y la Version 2 es un Triqui Simple distribuido cliente/servidor bajo Sockets TCP.

La versión 1 obviamente, permite jugar un triqui al tiempo y los 2 jugadores deben estar en la misma máquina. La interface de usuario es Modo Consola.



Figura 1: Diagrama de clases versión standalone

La versión 2, implementa un servidor de triqui, el cual si bien permite varios juegos remotos, NO los permite al tiempo, solo atiende un juego al tiempo. Del lado del cliente, el triqui permite que los 2 jugadores estén en la misma interface y máquina, es decir, los jugadores tienen que estar en el mismo sitio.



Figura 2. Diagrama de clases versión Sockets

Más detalles del diseño e implementación de estas 2 versiones, se encuentran en este documento y el material publicado en Eafit Interactiva.

# Requerimientos del Reto2:

A continuación se presenta los requerimientos del reto 2, los cuales se sintetizan en los siguientes 3:

1. Migrar el Servidor que atiende a los diferentes clientes a un modo concurrente, es decir, que permita atender varios juegos simultáneamente varias sesiones de juego remotas.
2. La implementación del TriquiSocket en la versión 2, permite comunicar cliente y servidor a través de 2 primitivas básicas “send” y “receive”, lo cual implica que entre las partes se defina un protocolo de comunicación, para la versión 3 del triqui, se requiere que el Cliente SIMULE invocación remota de métodos y objetos distribuidos. Lo que se requiere, es que tomen la versión Standalone, y simulen que la GUI o interfaz de usuario (TriquiPlayerRMISimulation), acceda de forma transparente a un objeto Simular a TriquiGame, pero que realmente no tiene la implementación sino que es un proxy hacia el servidor. Tomar como base el ejemplo de la calculadora y el chat rmi que se comenzaron a diseñar e implementar en clase.
3. Realizar una modificación al diseño e implementación que permita a los 2 jugadores, ESTAR EN DOS CLIENTES O MAQUINAS DISTRIBUIDAS. Es decir, ya el TriquiPlayerSocket2 o TriquiPlayerRMISimulation2 solo conecta a un Jugador. Esto implica, que se deben adicionar funcionalidades al servidor para que permita GESTIONAR la conexión y los juegos entre los JUGADORES distribuidos. Este requerimiento es nuevo, y no existía en las versiones 1 y 2 del Triqui.

La implementación de los tres (3) requerimientos anteriores pueden ser abordados de forma integral en una sola solución, pero también se pueden abordar de forma independiente.

# Desarrollo

# Explicar como se realizo el punto y agregar el cambio al diagrama de clases versión sockets

# Explicar como se realizo el punto y agregar el cambio al diagrama de clases versión sockets

# Explicar como se realizo el punto y especificar cambios en el diseño agregar el cambio al diagrama de clases versión sockets