

P.O.O.

Practica 5

Sockets clientes

Moviendo objetos de un cliente (maquina) a otro cliente (maquina) usando un servidor (VerySimpleChatServer) como intermediario.

Para esa practica dispone del servidor VerySimpleChatServer el cual tiene la capacidad de recibir un objeto de un cliente y enviarlo a los demás clientes.

Si utiliza LeeRed, la clase IncomingReader y la clase Red no olvide implementar la interfaz LeeRed en el cliente.

Ejecute al menos 2 clientes.

1.-Chat 3D (Carpeta estatiProgBarRedSimBot_/hombregatoact2014)

Modificar el programa que cambia el estado de animo del tamagochi en la maquina local y la maquina remota de modo que en lugar de disponer de solo 2 estados de animo se tengan **5 estados de animo** y que por cada estado de animo haya un botón. Cuando se presiona un botón se envía al servidor un objeto que contiene un entero que representa el estado de animo correspondiente al botón presionado.

2.-Cambio de Avatar (estatiProgBarRedSimBot_/Modelos/cartman2)

Modificar el programa que cambia el personaje del tamagochi en la maquina local y la maquina remota de modo que en lugar de disponer de solo 2 personajes se tengan **5 personajes**. .

3.- ChatBot Basico o Nano Alexa (Carpeta estatiProgBarRedSimBot)

Importante si hace esta opción no tiene que modificar el servidor (VerySimpleChatServer) ni escribir uno nuevo.

Modificar el programa que cambia el estado de animo del tamagochi en la maquina local y la maquina remota de modo que: un cliente y otro cliente interactúen del siguiente modo un cliente envía una pregunta y otro cliente le envía la respuesta. Cada cliente puede almacenar al menos 10 preguntas y 10 respuestas predefinidas (se pueden usar 2 arreglos o un HashMap). Si se le pide a otro cliente que cuente un chiste entre 10 chistes disponibles elije uno al azar y lo envia Si se le pide a otro cliente la hora entonces que envíe la hora actual.

En el problema 1 el objeto que se mueve entre maquinas es de la clase Icono dicha clase contiene un entero que es el dato que nos interesa enviar de una maquina a otra.

```
public class Icono implements Serializable {
    int turno;
    public Icono(int turno){ this.turno = turno; }
    public int getTurno(){ return turno; }
}
```

En este caso se debe enviar un objeto que contenga una cadena (mensaje) al servidor y se debe enviar si dicho mensaje es una petición o una respuesta es decir se debe incluir en el objeto enviado el tipo del mensaje. Cuando un mensaje llega el cliente debe ver si se trata de una pregunta o una respuesta. Si lo que llega es una pregunta se envía una respuesta al otro cliente. Si lo que llega es una respuesta se muestra. El servidor no responde las preguntas de los clientes solo actua como intermediario.

Ejemplos de preguntas tipo y respuestas tipo

En que **ciudad** vives? **D.F.**

Cuántos **años** tienes? **20**

En que **escuela** estudias? **ESCOM**

Dame la **hora**: Son las (poner hora actual aqui)

Cuenta un **chiste**: **Habia un perro de goma que cuando se rascaba se borraba.**

Cuenta un **chiste**: **¿Los hacendados hacen dados?**

4.-**Envío de imágenes** (estatiProgBarRedSimBot__/_mancat2015_2maqs/maq1)

Modificar el programa que cambia el estado de animo del tamagochi en la maquina local y la maquina remota de modo que lo que se envíe al servidor sean imágenes (ImageIcon). Esto permitiría que la imagen que se usa para un estado de animo sea distinta de la que se usa en otra maquina.

5.-**Cambio de vistas de manera remota**

Modificar el programa del sistema solar para que si la vista cambia de un planeta a otro en un cliente hacer que la vista también cambie en los demás clientes.

Estrategia para la Practica 5 Sockets clientes

Objetivo general: Mover objetos de una maquina a otra usando un servidor como intermediario.

Para esa practica se dispone del servidor VerySimpleChatServer el cual tiene la capacidad de recibir un objeto de un cliente y enviarlo a los demás clientes. VerySimpleChatServer funciona como los servidores de los chats (por eso lleva ese nombre).

Estrategia para practica 5

Observe que la práctica 5 tiene algunas opciones muy faciles de hacer

Para empezar estudie lo super básico de redes (un Crash Course)Ta

-Tarjeta de Red

-Host

-Paquete (mensaje, dirección de origen[fuente] y direccion de destino)

-Protocolo de Comunicación

-TCP/IP

-Pila de Protocolos

-Modelo OSI de ISO

Capa de Transporte	TCP (Transmission Control Protocol)
Capa de Red	IP (Internet Protocol)
Capa de Enlace	
Capa Fisica	

- Dirección IP ($a_1.a_2.a_3.a_4$)
- Enrutamiento
- Fragmentación del mensaje en paquetes y numeración de dichos paquetes del lado de la máquina local y del lado de la máquina remota el ordenamiento y unión de los paquetes para obtener el mensaje
- Modelo **Cliente-Servidor**
- Servidor (Procesa Peticiones)
- Cliente (Hace peticiones)
- Puerto
- Socket (punto donde se ofrece un servicio)
- Socket cliente
- Ciclo básico de vida de un Socket cliente
- paquete java.net
- paquete java.io
- Flujos de entrada y salida
- Flujos de entrada y salida para enviar y recibir objetos
- Estudie el protocolo DayTime (es un servicio que se ofrece en el puerto 13)

Compile la clase
DayTimeServer.java

y Ejecute
DayTimeServer

y luego Compile la clase

DayTimeClient.java

y Ejecute

DayTimeClient

Nota: puede ejecutar el cliente y el servidor en la misma máquina. El cliente se conectaría al servidor en la IP localhost (**127.0.0.1**).

VerySimpleChatServer envía y recibe objetos por la red. Para ser más preciso VerySimpleChatServer recibe un objeto de un cliente y lo envía a los demás clientes.

Utilice como su servidor a VerySimpleChatServer (el cual se liga al puerto 5000)

Para la parte del cliente estudie

la interfaz LeeRed

las clases Gato, IncomingReader, Red y Tirada

Observe que la interfaz LeeRed procesa lo que llega de la red por tanto debe implementarla en el cliente

Observe que la clase Red tiene un método que permite enviar objetos a la Red

Observe que la clase IncomingReader llama a LeeRed

Observe que lo que se mueve entre máquinas son objetos de tipo Tirada.

Algunas opciones de la práctica 5 tienen carpetas si elige una opción que tenga carpeta vaya a la carpeta y compile las clases que haya ahí y luego ejecute la clase que tenga el método main

En una terminal compile la clase VerySimpleChatServer con

```
javac VerySimpleChatServer.java
```

En una terminal Ejecute el servidor

```
java VerySimpleChatServer
```

En otra terminal compile la clase Tamagochi

```
javac -cp .: j3dcore.jar:j3dutils.jar:vecmath.jar Tamagochi.java
```

y en otra terminal mas ejecute un cliente

```
java -cp .: j3dcore.jar:j3dutils.jar:vecmath.jar Tamagochi
```

y en otra terminal mas ejecute otro cliente

```
java -cp .: j3dcore.jar:j3dutils.jar:vecmath.jar Tamagochi
```

y luego modifique Tamagochi.java de acuerdo a lo que pide la opción elegida.