

PROYECTO DE REDES EN VIDEOJUEGOS

1. Introducción.

Para diseñar un juego en Red suele haber dos esquemas: uno de **servidor** y otro de **cliente P2P**.

En el modelo de **servidor**, el juego realmente se ejecuta en un servidor y la consola o el cliente se conecta al servidor e interacciona con otros clientes a través del juego que realmente se ejecuta en gran parte en el servidor.

En el modelo cliente P2P, es posible que haya un servidor, pero solo registra información de los posibles clientes, nos conectamos al servidor, recibimos la información de red del cliente con el que queremos jugar y a partir de entonces nos conectamos directamente al otro cliente y el juego se está ejecutando en los dos clientes con interacción directa entre ellos a través de la red. En este caso nosotros omitimos el servidor y los clientes se conectan directamente entre si.

La diferencia fundamental está en el flujo de información a través de la red. En el modelo servidor el flujo de información va del cliente al servidor y este (el servidor) que es realmente el que está ejecutando “el mundo del juego” lo distribuye a otro cliente o clientes con el que estamos jugando y viceversa.

En el modelo P2P el cliente recibe la información de red como la dirección del otro cliente y a partir de ahí el intercambio de la información es de forma directa entre los dos clientes sin mediación del servidor, “el mundo del juego” está en cada uno de los clientes.

2. Objetivo del Proyecto.

Desarrollar dos versiones de un juego sencillo en red utilizando como herramienta de programación Java y utilizando directamente las primitivas de comunicación de sockets en java. Se programará el juego “a pelo” sin necesidad de ningún motor ni herramienta que facilite el trabajo. Lógicamente no hace falta que el juego tenga gráficos extraordinarios ni nada por el estilo. Lo que queremos es que se practique con los dos esquemas de juego en red descritos en clase pero haciéndolo directamente con las primitivas de comunicación de red del lenguaje de programación Java.

3. Descripción del Proyecto un poco más Detallada.

En el juego habrá que desarrollar dos versiones una el “juego usando el modelo cliente-servidor (un programa cliente y un programa servidor)” y otra “el juego usando el modelo P2P (un programa cliente)”.

El caso del modelo cliente P2P los clientes actúan tanto de cliente como de servidor. Por ello, en el programa cliente, a cada cliente se le pasa la dirección y puerto del proceso al que conectarse (sería el otro cliente haciendo de servidor y esperando conexión). Una vez conectados los clientes, se inicia el juego. Cada cliente hará simultáneamente de cliente y servidor desde el punto de vista de redes de computadores y cada proceso ejecutará exactamente el mismo código (es un solo programa cliente que se ejecuta dos veces).

En el esquema de servidor el juego está realmente ejecutándose en el servidor, los clientes se conectan a la dirección IP y puerto del servidor, y a partir de ahí solo manda sus propias jugadas, y recibe la información desde el servidor del estado del juego mediante algún tipo de representación (en texto gráfica) en pantalla. Pero es el servidor el que realmente ejecuta el juego y lleva el control del juego siendo él quien “decide” quien ha ganado. En el caso de que

como consecuencia de una jugada de un cliente gane, se le comunicará a los dos extremos por parte del servidor. Por el contrario, en el esquema P2P dado que la interacción es directa, estos análisis los tienen que hacer los dos clientes por su lado.

4. Trabajo a desarrollar y requisitos.

Habrá que programar en java los dos esquemas de juego “cliente-servidor” y P2P.

En el caso de esquema de Servidor (modelo cliente-servidor), se programará un cliente y un servidor. Para probarlo ejecutareis el servidor y después dos clientes para que se conecten y compitan entre sí.

Una vez finalizado, se comenzará la programación del juego en el esquema P2P. Aquí solo hay que realizar un programa “EL CLIENTE” que actuará tanto de cliente como de servidor. Para probarlo ejecutareis el programa, primer cliente (se quedar esperando que se conecte un cliente) y volveréis a ejecutar el programa, segundo cliente (se conecta al primero y espera que el primero se conecte a el (a su servidor). Una vez conectados juega uno contra el otro.

REQUISITO IMPRESCINDIBLE: Los programas .java debe de poder compilarse sin errores en cualquier entorno y sistema operativo, debes probar que compilan en Windows y Linux antes de entregarlo. Utiliza para escribirlos un editor de texto plano que no te introduzca ningún tipo de basura que origine la no compilación y no utilices acentos, ñ, ni caracteres extraños en los comentarios.

En el Aula Virtual se habilitará una tarea para la entrega de vuestros programas donde se indicará el plazo de entrega.

Primero habrá un plazo para el esquema de Cliente-Servidor y posteriormente otro para el esquema de P2P.

Se deben entregar los programas .java y un documento de texto o manual que debe contener, una portada con nombre del juego y el nombre de los componentes del grupo y otras hojas en las que se explique en que consiste vuestro juego (descripción similar a la de este boletín), como ejecutarlo (si se pasan parámetros decir cuales), cual es la forma de jugar y cuando y como acaba la partida. También cualquier otro detalle que facilite la comprensión del uso del juego.

Para cada tarea se os suministrará un algoritmo para facilitar la programación de cada uno de los esquemas.

5. Juegos que se pueden implementar.

Mastermind:

El juego consiste en adivinar una combinación de 4 fichas de color, hay que adivinar los colores y el orden de los colores.

Cada jugador tendrá 15 tiradas. Habrá que almacenar las tiradas en quince variables.

Hay cuatro posibles fichas de colores: R, Y, A, M (rojo, amarillo yellow, azul y marón). Una jugada típica del jugador que introducirá por teclado será: RYYM.

Las posibles configuraciones iniciales de fichas que el jugador tiene que acertar estará acumulada y predefinida por el programa. Puedes almacenar 5 o más configuraciones y que aleatoriamente el programa asigne una a cada jugador.

Ante una jugada de un jugador, el juego tendrá que contestar por ejemplo N2B2 (2 negros y 2 blancos), es decir, hay dos fichas en su sitio y del color adecuado y hay dos fichas del color adecuado, pero mal colocadas.

Después de cada jugada el juego debe imprimir las jugadas y resultados anteriores. Por ejemplo:

⇒ -si por ejemplo hay que adivinar ARYA

Te toca ¿dame una jugada?

---RYRA

Resultado

RYRA N1B2

Te toca ¿dame una jugada?

---ARMA

Resultado

RYRA N1B2

ARMA N3B0

Te toca ¿dame una jugada?

.....

GANASTE.... o si el otro acaba antes PERDISTE

El juego termina cuando se consiguen cuatro negros N4 o cuando se acaban las 15 tiradas.

Gana el jugador que antes consiga los 4 negros y se empate si los dos consiguen las 15 tiradas sin acertar la combinación.

Se debe indicar por pantalla a cada jugador si gana o pierde.

En el caso del modelo servidor. El servidor escogerá aleatoriamente una combinación inicial de fichas para cada jugador de las que tiene almacenada. El servidor le pide al primer cliente su jugada. Analiza la jugada e imprime los resultados de sus jugadas al cliente. Después le pide su

jugada al otro cliente y realiza la misma operación. Va alternando entre los clientes hasta que uno acierte (4N) GANA, o los dos lleguen a 15 jugadas, EMPATE. En ese caso se informará a los clientes que se ha acabado el juego y si ganan, pierden o empatan.

En el caso del modelo cliente. Cada cliente tiene almacenados 5 o más configuraciones iniciales y escoge una aleatoriamente para dársela a adivinar al otro cliente. Uno de los dos empieza y manda una jugada el otro cliente, la recibe, la analiza e informa del resultado al contrincante (le imprime lo necesario). El turno pasa al siguiente que hace una jugada y el proceso continua. Los dos clientes deben de analizar cuando reciben una jugada si han perdido porque el otro haya adivinado o han empatado porque los dos han alcanzado las 15 jugadas, en cuyo caso comunican a ambos el resultado y se acaba el juego.

Parejas:

El juego tiene almacenados inicialmente 5 o más tableros (matrices 4x4) en las que hay 8 parejas de letras repartidas aleatoriamente por el tablero, las letras serían ABCDEFGH.

Por ejemplo:

A	E	B	F
C	D	C	G
A	H	G	B
D	H	E	F

El programa asignará aleatoriamente un tablero de juego a cada jugador. Inicialmente mostrará a todos los jugadores **los tableros de todos** de la siguiente forma:

- un mensaje “tablero del jugador x” seguido de su tablero con todo a X.

Después da el turno a un jugador (con un mensaje a todos) que realiza lo siguiente:

- Se le pide a ese jugador su jugada y da su jugada indicando dos números (fila, columna). Por ejemplo (1,1).
- El programa con un mensaje contesta cual es la letra oculta y pide la otra jugada. Contesta: A
- El cliente da otra jugada indicando dos números (fila, columna). Por ejemplo (3,1).
- El programa contesta la letra oculta. Ahora la A.
- Si la letra de la primera jugada coincide con la de la segunda jugada, el programa BORRA LA PANTALLA y a continuación imprime: “acertaste” y saca el tablero de 4x4 de ese jugador mostrando las letras acertadas Y SIGUE jugando el mismo jugador.
- Si las letras no coinciden, el programa imprime: “fallaste” y a continuación BORRA LA PANTALLA, mostrará a todos los jugadores **los tableros actualizados de todos**, da el turno a un jugador con un mensaje a todos y repetimos el proceso.

Esto se repite hasta que uno de los jugadores haya destapado todas las letras. El juego imprimirá entonces a todos GANA EL JUGADOR X.

EJEMPLO DEL juego

Tablero inicial:

Tablero de jugador 1

X X X X

X X X X

X X X X

X X X X

Tablero de jugador 2

X X X X

X X X X

X X X X

X X X X

Turno jugador 1

Dame tu primera jugada: 1,1

Es una A

Dame tu segunda jugada: 3,1

Es una A

“acertaste”

-----aquí se debe borrar la pantalla y escribir-----

Turno y tablero de jugador 1

A X X X

X X X X

A X X X

X X X X

Dame tu primera jugada: 1,2

Es una E

Dame tu segunda jugada: 2,1

Es una C

“fallaste”

-----aquí se debe borrar la pantalla y escribir-----

Tablero de jugador 1

A X X X

X X X X

A X X X

X X X X

Tablero de jugador 2

X X X X

X X X X

X X X X

X X X X

Turno jugador 2

Dame.....

.....etc.

ES IMPRESCINDIBLE BORRAR LA PANTALLA DESPUES DE QUE SE HAGA LA COMPARACION DE LAS DOS JUGADAS YA QUE SE TRATA DE UN JUEGO DE MEMORIA Y NO DEBE ESTAR LA INFORMACION TODO EL RATO EN PANTALLA.

En el caso del modelo servidor. El servidor debe tener guardados los tableros iniciales y escoger aleatoriamente uno para cada cliente. Tras las jugadas el servidor debe conservar el estado más actual (aciertos) de cada uno de los tableros de juego de los jugadores. Ganará el jugador que destape antes su tablero acertando todas las parejas de letras.

En el caso del modelo cliente. Cada cliente debe tener guardados los tableros iniciales y escoger aleatoriamente uno para que juegue el otro cliente. Uno de los dos empieza y manda las jugadas, el otro las recibe las compara y si ha acertado seguirá jugado el oponente y si falla cambia el turno de jugador. Así, un cliente tiene el turno para jugar mientras acierta (pareja correcta). Sí un cliente llega a la última jugada y acierta, el oponente (que es el que la analiza) informa a todos de que jugador gana y quien pierde. En este modelo proceso que este haciendo las comprobaciones es el que debe actualizar los tableros a todos cuando el otro jugador falla.

Adivina el refrán:

El juego tiene almacenados refranes en las filas de una matriz de 23 columnas. Cada fila tiene como elementos letras o el carácter guión “-” (que representa un espacio o separador entre palabras).

El juego consiste en ir diciendo letras hasta adivinar el refrán completo.

Inicialmente tendrás dos matrices de tamaño (5x23), una REFRANES y otra SOLUCION en la que se han sustituido las letras por “*”. Usa las siguientes:

Matriz REFRANES:

E	L	-	Q	U	E	-	E	S	P	E	R	A	-	D	E	S	E	S	P	E	R	A
E	L	-	Q	U	E	-	N	O	-	L	L	O	R	A	-	N	O	-	M	A	M	A
D	E	-	T	A	L	-	P	A	L	O	-	T	A	L	-	A	S	T	I	L	L	A
E	L	-	S	A	B	E	R	-	N	O	-	O	C	U	P	A	-	L	U	G	A	R
D	O	N	D	E	-	L	A	S	-	D	A	N	-	L	A	S	-	T	O	M	A	N

Matriz SOLUCION:

*	*	-	*	*	*	-	*	*	*	*	*	*	-	*	*	*	*	*	*	*	*	*
*	*	-	*	*	*	-	*	*	-	*	*	*	*	*	-	*	*	-	*	*	*	*
*	*	-	*	*	*	-	*	*	*	*	-	*	*	*	-	*	*	*	*	*	*	*
*	*	-	*	*	*	*	*	-	*	*	-	*	*	*	*	*	-	*	*	*	*	*
*	*	*	*	*	-	*	*	*	-	*	*	*	-	*	*	*	-	*	*	*	*	*

El juego escogerá para cada jugador una fila (entre la 1 a la 5) y se guardará dicha fila de ambas matrices en dos vectores (1X23). Cada jugador tendrá dos vectores, el que contiene el refrán a adivinar y el que presenta como tablero de juego que es donde se irán actualizando los aciertos.

El mecanismo del juego será el siguiente. Comienza el JUGADOR1

- Se mostrará a todos los jugadores los vectores_solucion de todos, de la forma: un mensaje "refrán del jugador x" seguido de su **vector_solución** . Después un mensaje: "Turno del jugador x" y se cambia el turno a ese jugador.
- Se le solicita al jugador una letra. La letra se comprobará en cada una de las columnas del **vector_refran** y se copiarán los aciertos en el **vector_solución** de ese jugador.
- Si ha habido aciertos o no se ha acabado se volverá a mostrar al mismo jugador su **vector_solución** actualizado con los aciertos y se pedirá otra letra. Al otro jugador se le imprimirá: "sigue el turno del jugador x".
- Si se ha acabado (significa que el número total de aciertos del jugador es igual a 23 menos el número de caracteres "-" que hay en el vector) se acabará el juego imprimiendo a dicho jugador GANASTE y al otro HAS PERDIDO.
- Si no se ha acabado y no hay aciertos se repetirá el proceso.

Ganará el juego el primero con complete su refrán.

En el caso del modelo servidor. El servidor debe tener guardadas las matrices iniciales y escoger aleatoriamente un número de fila para cada cliente guardando esas filas en vectores. El servidor le pide al primer cliente su jugada. Analiza el resultado, imprime lo necesario al jugador. Si ha acertado continua con el turno y si ha fallado el turno pasa al otro jugador. El juego acaba cuando un jugador acierta el refrán y el servidor informar quien gana y quien pierde.

En el caso del modelo cliente. Cada cliente debe tener guardados los tableros iniciales y escoge aleatoriamente un número de fila seleccionando el refrán que debe adivinar el otro cliente. Un cliente controla el juego del otro y viceversa. Uno de los dos empieza y manda una jugada el otro cliente, el otro la recibe y la analiza dice el resultado al oponente y si ha acertado seguirá jugado el oponente y si falla cambia el turno de jugador. Así, un cliente tiene el turno para jugar mientras acierta. Si un cliente acierta la última jugada, el oponente (que es el que la analiza) informa a todos de que cliente gana y quien pierde.