



UNIVERSIDAD NACIONAL DE LA MATANZA

Departamento de Ingeniería e Investigaciones
Tecnológicas

SISTEMAS OPERATIVOS

Jefe de Cátedra: Fabio E. Rivalta

Jefe de TP: Fernando Boettner

Docentes: L. Catalano, M. Volker, F. Piubel, D. Barbieri

TRABAJO PRACTICO N° 4 MANUAL DE USUARIO

<i>Alumno</i>	<i>DNI</i>
Gavidia Hernán	36.376.771
Martín Gonzalo	36.170.285
Miranda Cristian	35.349.257

INDICE

1 - ARQUITECTURA	3
1.1. ESTRUCTURA DEL PROYECTO	3
1.1.1. DKK	3
1.1.2. DKKClient	3
1.1.3. DKKMatchServer	3
1.1.4. DKKMonitor	3
1.1.5. DKKTournamentServer	3
1.2. EJECUCION	4
1.2.1. DKKTournamentServer	4
1.2.2. DKKMatchServer	4
1.2.3. DKKMonitor	4
1.2.4. DKKClient	4
1.3. MONITOR	4
1.4. LOGS	4
1.5. COMUNICACION ENTRE CLIENTES	5
1.5.1. MOVIMIENTOS	5
1.5.2. GANADOR DE LA PARTIDA	5
1.5.3. RESCATE PAULINE	5
2 - MANUAL DE USO	6
2.1. INSTALACION Y CONFIGURACION	6
2.1.1. INSTALAR DEPENDENCIAS	6
2.1.2. COMPILACION	6
2.1.3. CONFIGURACION	6
2.2. EJECUCION	7
2.3. MONITOREO	7

1- ARQUITECTURA

1.1. ESTRUCTURA DEL PROYECTO

El proyecto se divide en 5 módulos bien definidos detallados a continuación:



DKK



DKKClient



DKKMatchServer



DKKMonitor



DKKTournamentServer

1.1.1. DKK

Módulo común a los otros cuatro que contiene los recursos utilizados por todo el sistema. Aquí se encuentran funciones útiles, estructuras compartidas y constantes.

1.1.2. DKKClient

Maneja gráficos y toda la dinámica del juego para un solo cliente. Contiene los recursos utilizados para mostrar gráficos tales como imágenes y fuentes. Tiene como dependencias las librerías de SDL.

1.1.3. DKKMatchServer

Este módulo compila un binario que será ejecutado por cada partida entre dos clientes. Es el encargado de comunicar al cliente con el servidor de torneos y viceversa.

Utiliza un hilo por cada cliente para mantener una comunicación ininterrumpida con ambos.

1.1.4. DKKMonitor

Monitor del sistema. Contiene la lógica de monitoreo y liberación de recursos tomados tanto por el servidor de torneo y partidas, como para el cliente (dependiendo de donde esté siendo ejecutado).

1.1.5. DKKTournamentServer

Servidor del torneo. Habrá una sola instancia en ejecución. Maneja la inscripción de los clientes y la ejecución de los servidores de partida.

Utiliza un hilo por cada partida generada, sincronizados mediante semáforos.

1.2. EJECUCION

1.2.1. DKKTournamentServer

No recibe ningún parámetro. Lee los necesarios de su archivo de configuración.

1.2.1. DKKMatchServer

Es ejecutado por el servidor de torneos cada dos clientes. Recibe el ID de partida y la cantidad de partidas totales para poder acceder a la memoria compartida.

1.2.1. DKKMonitor

Siempre será ejecutado y monitoreado por el servidor de torneos, si un cliente corre sobre la misma PC donde se ejecutó este último, no ejecutará el servicio monitor ya que el server lo inició primero. De lo contrario será iniciado y monitoreado por el cliente.

1.2.1. DKKClient

No recibe ningún parámetro. Lee los necesarios de su archivo de configuración.

1.3. MONITOR

El servicio monitor funciona de la misma manera tanto para el servidor de torneo como para el cliente. A partir de ahora llamaremos al servidor de torneos y al cliente "monitoreado".

El monitor se comunica con el monitoreado a través de sockets de forma sincrónica enviando y recibiendo un mensaje liviano (un carácter). Posee un sistema de "retry" que prueba el envío/recepción del mensaje y falla al tercer intento procediendo a liberar los recursos correspondientes. El *timeout* es de 20 segundos por intento.

El monitoreado utiliza un hilo dedicado a la comunicación con el servicio monitor.

En caso de que el servicio monitor finalice, será reiniciado solamente por el servicio que lo ejecuto en primer instancia.

Para el caso de los servidores de partida el monitor controla que el o los procesos correspondientes a la o las partidas sigan en ejecución. No utiliza en este caso sockets ni nada por el estilo. Y ante la caída de uno de los procesos partida el monitor procederá a finalizar todas las partidas en curso, como así también el servidor de torneos (criterio seleccionado por el grupo).

1.4. LOGS

Cada módulo que produce un binario ejecutable (todos excepto por DKK) genera tres archivos de log.

<modulo>.log: Archivo de log principal.

<modulo>Error.log: Solo errores.

<modulo>Loop.log: Solo mensajes cíclicos

1.5. COMUNICACIÓN ENTRE CLIENTES

Todo mensaje intercambiado entre los clientes de una misma partida pasan primero por el servidor de partidas. Pueden intercambiar lo siguiente:

1.5.1. MOVIMIENTOS

Cada cliente comunica a través de su socket con el servidor de partida el movimiento del personaje con el siguiente formato:

"{move}:0-1=posicionX|posicionY_velocidadX|velocidadY "

Ejemplo:

{move}:0=010|200_005|007 → Mario se encuentra en el mapa en la posición (10;200) y se mueve a una velocidad de 5 en el eje X y de 7 en el eje y

1.5.2. GANADOR DE LA PARTIDA

Comunica el ganador de la partida actual mediante un mensaje que contiene la etiqueta {winner}. Si además fuese el ganador del torneo también le llegara el mensaje con la etiqueta {champion}.

1.5.3. RESCATE A PAULINE

Comunica cada vez que un jugador rescata a Pauline, le llegara al servidor de partida el mensaje con la etiqueta {rescue}.

2- MANUAL DE USO

2.1. INSTALACION Y CONFIGURACION

2.1.1. INSTALAR DEPENDENCIAS

- a. `sudo apt-get install libsdl1.2-dev libsdl1.2debian`
- b. `sudo apt-get install libsdl-image1.2-dev libsdl-mixer1.2-dev libsdl-ttf2.0-dev`

2.1.2. COMPILACION

- a. Acceder a la carpeta TP4 desde un terminal.
- b. Dentro de ella ejecutar el comando "make".
- c. Corroborar que no haya errores de compilación en pantalla.
- d. Luego de la compilación se generará la carpeta "STAGING", dentro de ella 2 carpetas adicionales "SERVER" y "CLIENT".
 - i. Dentro de la carpeta SERVER encontraremos:
 1. Binario del Servidor de Torneos **DKKTournamentServer**.
 2. Binario del Servidor de Partidas **DKKMatchServer**.
 3. Binario del Monitor **DKKMonitor**.
 4. Archivo de Configuración **server.conf**.
 - ii. Dentro de la carpeta CLIENT encontraremos:
 1. Binario del Cliente **DKKClient**.
 2. Archivo de Configuración **client.conf**.

2.1.3. CONFIGURACION

Procederemos a chequear los parámetros de configuración, tanto del servidor como del cliente:

- i. Para configurar el servidor accedemos al archivo **STAGING/SERVER/server.conf**
 3. IP del servidor de torneo (**server.host**)
 4. Puerto en el que atiende el servidor de torneo (**server.port**).
 5. Timeout de inscripción de clientes (**connection.timeout**).
 6. Dificultad del juego (**game.level**).
 7. Cantidad de rescates para ganar la partida (**rescue.amounts**).
 8. Ruta al binario del proceso Monitor (**monitor.path**).
 9. Puerto en el cual se comunica con el monitor (**monitor.port**).
 10. Ruta al binario del Servidor de Partida (**match.server.path**).
- ii. Para configurar el cliente accedemos al archivo **STAGING/CLIENT/client.conf**.
 11. IP del servidor de torneo (**server.host**).
 12. Puerto del servidor de torneo (**server.port**).
 13. Ruta al binario del proceso Monitor (**monitor.path**).
 14. Puerto en el cual se comunica con el monitor (**monitor.port**).
 15. Teclas para los movimientos.
 - a. Arriba (**key.up**).
 - b. Abajo (**key.down**).
 - c. Izquierda (**key.left**).
 - d. Derecha (**key.right**).
 - e. Saltar (**key.jump**).
 - f. Salir (**key.quit**).

2.2. EJECUCION

- 1) Para ejecutar tanto el servidor de torneo como el cliente debemos hacerlo desde un terminal.
- 2) Para ejecutar el servidor de torneo debemos realizar lo siguiente:
 - a. Nos dirigimos al siguiente directorio **STAGING/SERVER/**
 - b. Ejecutamos el binario correspondiente al servidor de torneos **DKKTournamentServer**
- 3) Para ejecutar el cliente debemos realizar lo siguiente:
 - a. Nos dirigimos al siguiente directorio **STAGING/CLIENT/**
 - b. Ejecutamos el binario correspondiente al cliente **DKKClient**
 - c. Nos pedirá que ingresemos el nombre del jugador, y a continuación esperará la respuesta del servidor de torneo.
 - d. Una vez que genere nuestra partida se nos abrirá la ventana correspondiente y podremos jugar.
 - e. Si ganamos partidas debemos aguardar a que se genere la siguiente con el ganador que corresponda, y de ser la final nos lo informara en pantalla.

2.3. MONITOREO

- 1) Por cada binario ejecutado se generaran 2 o 3 archivos de log (dependiendo de si se producen errores), los cuales se generaran en la carpeta donde se encuentre cada binario:
 - a. Los LOGS del Servidor de torneo, Servidor de Partida, Monitor (si lo ejecuta el servidor), se encontraran en **STAGING/SERVER/**
 - b. Los LOGS del Cliente, Monitor (si lo ejecuta el cliente), se encontraran en **STAGING/CLIENT**
- 2) Podrá entonces monitorear el estado de los procesos desde dichos archivos de log.
- 3) Una buena práctica podría ser ejecutar un **tail -f <archivo>.log** en terminales aparte para ver todos los eventos en línea.