

Informática II

Práctica 2: MasterMind en red. Versión 1

GSyC

Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación

Noviembre de 2022

1. Introducción

Partiendo de la base de la práctica **MasterMind Unchained** desarrollada en esta asignatura, vamos a comenzar el desarrollo del juego en modo cliente-servidor.

Para ello será necesario portar la aplicación local y el funcionamiento del juego a un paradigma cliente-servidor donde el servidor ejecute toda la lógica de juego y el cliente sea el encargado de recoger los mensajes que envíe el usuario y mostrar la información relativa al juego.

Esta será la primera iteración del juego con lo que es primordial que tengáis todo el código bien estructurado y comentado para futuras ampliaciones de la funcionalidad.

2. Requisitos Generales

Los requisitos que debe cumplir la práctica son los siguientes:

- La comunicación entre el cliente y el servidor se hará a través de un canal seguro que no se pierda ningún paquete enviado.
- El servidor debe ser capaz de ejecutar múltiples partidas de un sólo jugador.
- Deberá utilizar el paradigma de programación orientada a objetos siempre que se pueda.
- El desarrollo de la práctica no tendrá una única solución sino que cada solución desarrollada puede ser válida.

3. Requisitos específicos del cliente

3.1. Argumentos

Al ejecutar el cliente se podrá pasar los siguientes argumentos:

- **-n** Este argumento irá acompañado de un nombre o nick, que representa el nombre del jugador. El programa también deberá permitir recibir como argumento `--name` (Ejemplo: `--name=jesus`). El nick puede tener cualquier longitud y puede contener números.
- **-i** Este argumento irá acompañado de la ip dónde se encuentra el servidor. El programa también deberá permitir recibir como argumento `--ip` (Ejemplo: `--ip=127.0.0.3`). Si no se proporciona el argumento, el valor por defecto será 127.0.0.1.
- **-p** Este argumento irá acompañado del puerto dónde se encuentra el servidor. El programa también deberá permitir recibir como argumento `--port` (Ejemplo: `--port=6500`). Si no se proporciona este argumento, el valor por defecto será 1234.

Es importante que se compruebe que puerto sea un número entero. También hay que tener en cuenta que si no se proporciona un nombre para el jugador, el programa deberá finalizar. Los argumentos relativos a la ip y el puerto son opcionales ya que hay definidos parámetros por defecto.

3.2. Lógica de juego del cliente

El cliente deberá ser ligero en los diferentes cálculos que va a ejecutar, de forma que no deberá realizar ninguna opción y su única función es enviar las combinaciones que el usuario introduzca por teclado y recibir las órdenes que le dicte el servidor conforme a cada uno de los movimientos realizados.

De esta forma, el principal uso del cliente será el tratamiento de cadenas de caracteres. En el apartado 5 se mostrarán los mensajes del protocolo que deberá ser capaz de interpretar el cliente.

4. Requisitos específicos del servidor

4.1. Argumentos

- **-i** Este argumento irá acompañado de la ip de lanzamiento del servidor. El programa también deberá permitir recibir como argumento `--ip` (`--ip=127.0.0.3`). Si no se proporciona el argumento, el valor por defecto será 127.0.0.1.
- **-p** Este argumento irá acompañado del puerto dónde se encuentra el servidor. El programa también deberá permitir recibir como argumento `--port` (Ejemplo: `--port=6500`). Si no se proporciona este argumento, el valor por defecto será 1234.

El servidor siempre se lanzará en la ip 127.0.0.1. Dentro de la red de la Universidad podéis probar a iniciar el servidor en la ip 0.0.0.0 e intentar conectarlos desde otros ordenadores.

4.2. Fases del juego

El juego será en todo momento controlado por el servidor, y para su correcto funcionamiento tendremos diferentes fases donde se desarrollarán las diferentes configuraciones de la partida.

- **Fase inicialización:** En esta fase se define el número de rondas que se van a jugar antes de la finalización de la partida. El cliente deberá enviar al servidor el número de rondas. En este caso, el servidor deberá responder positiva o negativamente dependiendo de si es un valor correcto o no.
- **Fase de configuración de la palabra secreta:** Posteriormente, el cliente deberá enviarle al servidor la palabra secreta que permita comenzar el juego y que el jugador deberá encontrar. Se puede usar una palabra reservada para denotar al servidor que genere aleatoriamente una combinación secreta. Aunque se puede usar cualquier palabra, siempre que se notifique en los mensajes de control, se puede usar la palabra **nocombiCode**.
- **Fase juego:** En esta fase se repetirá el modo de funcionamiento del juego en local, se pedirá una combinación en el cliente y será enviada al servidor que validará si es correcta o no. Esto se repetirá hasta que el jugador se quede sin turnos o que el jugador gane. Por cada turno que se juegue el servidor deberá enviar de vuelta al cliente información de los acierto así como de los semiaciertos que ha tenido el jugador en esa ronda.

5. Mensajes del protocolo

Dado que la comunicación entre el cliente y el servidor será a través del intercambio de cadenas de caracteres convertidas a información binaria, hemos de definir los mensajes de control

5.1. Fase de inicialización:

Mensaje de control de inicialización deberá enviar una cabecera del tipo de mensaje de inicialización así como un mensaje de si se ha producido la inicialización correctamente o errónea.

| Tipo | Mensaje |
|------|---------|
|------|---------|

Donde:

- **Tipo** será el tipo de mensaje de inicialización de la partida.
- **mensaje** será información sobre si ha sido correcta la inicialización o no.

5.2. Fase de juego:

Durante el juego se intercambiarán ciertos mensajes en donde el servidor enviará información vital al cliente para mostrar al jugador, o el cliente deberá enviarle información al servidor sobre la tirada actual.

■ Mensaje: Turno

Este mensaje se ejecutará en cada turno que se juegue

| | | | |
|------|-------------|----------|--------------|
| Tipo | combinacion | aciertos | semiaciertos |
|------|-------------|----------|--------------|

Donde:

- **Tipo** será el tipo de mensaje del turno.
- **combinación** Es la combinación del turno que ha recibido el servidor.
- **aciertos** el número de aciertos que se ha tenido.
- **semiaciertos** el número de semiaciertos del turno.

■ Mensaje: Error

Este mensaje se ejecutará en cada turno que se juegue

| | |
|------|--------|
| Tipo | motivo |
|------|--------|

Donde:

- **Tipo** será el tipo de mensaje de error.
- **motivo** Muestra un motivo por el que se ha producido el error. En ese caso el turno no avanza.

■ Mensaje: Fin de la partida

Este mensaje se ejecutará en cada turno que se juegue

| | |
|------|--------|
| Tipo | motivo |
|------|--------|

Donde:

- **Tipo** será el tipo de mensaje de fin de partida.
- **motivo** Muestra un mensaje al jugador diciendo que no ha conseguido ganar el juego en el número de turnos que se definió al inicio del juego.

■ Mensaje: Ganador

Este mensaje se ejecutará en cada turno que se juegue

| | |
|------|--------|
| Tipo | motivo |
|------|--------|

Donde:

- **Tipo** será el tipo de mensaje de ganador.
- **motivo** Muestra una felicitación al jugador y el número de turnos que ha usado para ganar.

■ Mensaje: Combinación

Este mensaje se ejecutará en cada turno que se juegue

| | |
|------|-------------|
| Tipo | combinación |
|------|-------------|

Donde:

- **Tipo** será el tipo de mensaje de la combinación enviada del cliente al servidor.
- **combinación** contiene la combinación de fichas que ha introducido el jugador por teclado y que se envía al servidor.

6. Condiciones de funcionamiento

La práctica en su conjunto deberá funcionar conforme a las especificaciones marcadas, deberá gestionar diferentes excepciones que gestionen los posibles errores que se produzcan en la aplicación. Los mensajes son orientativos y pueden ser modificados, incrementar su número o cambiarlos sin ningún problema.

7. Entrega

La entrega de esta práctica se hará a través del aula virtual, con límite: **01 de diciembre de 2022 a las 08:00h.**