

## Trabajo Práctico: Fase 2 - Red

### Objetivo

Desarrollar el soporte de red para el sistema.

### Aplicación

Se debe desarrollar una aplicación, la cual podrá ser lanzada a través de CLI (interfaz de línea de comando) en modo cliente o en modo servidor. La aplicación tomará los parámetros de configuración inicial a través de un archivo de configuración.

### El Servidor

El servidor deberá proveer una partida multijugador de hasta 4 participantes como máximo. El valor de jugadores admitidos será configurable desde el archivo Json. La partida iniciará una vez que se ha alcanzado ese número de jugadores conectados. No es necesario que el servidor tenga una interfaz gráfica, basta con lanzarlo por consola.

Cuando un jugador se desconecta voluntariamente ó por falla técnica, su personaje pasa a ser controlado por su compañero de equipo, si este jugador se reconecta deberá continuar desde la pantalla actual.

Si un jugador se quiere conectar al servidor y se supera la cantidad máxima de jugadores para la partida, el servidor le informa al cliente que la partida ya está llena y el jugador lo visualiza en pantalla.

### El Cliente

Al iniciar la aplicación cliente, el usuario deberá poder conectarse al servidor, luego de ser aceptada la conexión queda en espera a que se conecten todos los jugadores. Una vez conectados todos los jugadores se pasa a la pantalla de selección de personaje. Finalizada la selección de los personajes se pasa al inicio de la pelea en el campo de batalla.

### Consideraciones especiales

Todas las conexiones y desconexiones deberán ser identificadas tanto por el cliente como por el servidor y ser logueadas.

Los datos de puerto y dirección IP deberán ser ingresados al lanzar las aplicaciones por línea de comando.

### Configuración

El archivo de configuración será un archivo de texto plano en formato Json, el cual contendrá los parámetros de configuración inicial de la aplicación. En caso de no encontrar el archivo, se debe leer un archivo de configuración por defecto.

A continuación se da un ejemplo de archivo de configuración de tipo Json, al que pueden agregarse la cantidad de parámetros que se consideren necesarios, como así también darle la estructura que consideren adecuada.

```
{
  "client":{
    "log":"ERROR",
    "battlefield":
    [
      {"background":
        {
          "filepath":"file11.png",
          "zindex":3
        }
      },
      {"background":
        {
          "filepath":"file2.png",
          "zindex":2
        }
      },
      {"background":
        {
          "filepath":"file1.png",
          "zindex":1
        }
      }
    ],
    "characters":
    [
      {"name":"ryu"
        "filepath":"file55.png",
        "height":45,
        "width":20,
        "zindex":99
      },
      {"name":"spiderman"
        "filepath":"file56.png",
        "height":45,
        "width":20,
        "zindex":99
      }
    ]
  },
  "window":{
    "height":800,
    "width":600
  }
},
"server":{
  "players":4,
  "log":"INFO"
}
}
```

## **Restricciones**

- La implementación deberá estar hecha en C/C++.
- Para la lectura y escritura de archivos JSON debe utilizarse una biblioteca.
- Para el manejo de gráficos se deberá usar la biblioteca SDL.
- Todo el código debe ser desarrollado íntegramente por cada grupo. No se permite la reutilización de código de cuatrimestres anteriores o de otras materias. Ante cualquier duda se deberá consultar con los docentes. La reutilización de código sin consulta previa será condición suficiente para la desaprobación de la materia.

## **Fechas**

(Ver cronograma en el Campus)