

Videojuego Snake con Sockets

1.- Diseño del protocolo:

1.1.- Tipos de mensajes:

Servidor → Cliente (S → C)

Cliente → Servidor (C → S)

1.2.- Mensajes de inicialización:

El servidor envía un mensaje al cliente apenas se conecta al servidor, o cuando otro cliente se conecta al servidor. El mensaje consta de un ID_CLIENTE único por cada cliente de tamaño 3 chars, una PIEZA que es 1 caracter único por cada cliente, un SIZE_NICKNAME de tamaño 3 chars, y un NICKNAME de tamaño variable.

S → C : Mensaje de inicialización de otro cliente: 'U' ID_CLIENTE 'I' PIEZA SIZE_NICKNAME NICKNAME

S → C : Mensaje de inicialización de mi cliente: 'U' ID_CLIENTE 'I' PIEZA SIZE_NICKNAME NICKNAME

1.2.- Mensajes de movimiento:

Mensajes de movimiento para el juego "Snake", se envían estos mensajes cuando se cumplen las siguientes condiciones: Se han conectado jugadores iguales al número de jugadores que se define al inicio; Se presiona el botón arriba, abajo, izquierda, derecha; El juego no ha terminado.

C → S : Movimiento hacia arriba: 'M' 'U'

C → S : Movimiento hacia abajo: 'M' 'D'

C → S : Movimiento hacia la izquierda: 'M' 'L'

C → S : Movimiento hacia la derecha: 'M' 'R'

S → C : Movimiento hacia arriba de jugador con id: 'U' ID_CLIENTE 'M' 'U'

S → C : Movimiento hacia abajo de jugador con id: 'U' ID_CLIENTE 'M' 'D'

S → C : Movimiento hacia la izquierda de jugador con id: 'U' ID_CLIENTE 'M' 'L'

S → C : Movimiento hacia la derecha de jugador con id: 'U' ID_CLIENTE 'M' 'R'

1.3.- Mensajes de chat:

Mensajes cuando el usuario escribe en la entrada: 'C' NICKNAME MENSAJE

Y después se presiona ENTER. Donde MENSAJE es un mensaje de tamaño variable.

C → S : Mensaje a cliente con id ID_CLIENTE: 'C' ID_CLIENTE SIZE_MSJ MENSAJE

S → C : Mensaje recibido de cliente con id ID_CLIENTE: 'U' ID_CLIENTE 'C' SIZE_MSJ MENSAJE

1.4.- Mensajes para cambiar nickname:

Mensajes para cambiar nickname:

C → S : Mensaje para cambiar mi nick a NICKNAME: **'R' SIZE_NICKNAME NICKNAME**

S → C : Mensaje recibido de cliente con id ID_CLIENTE que cambio su nick a NICKNAME: **'U' ID_CLIENTE 'R' SIZE_NICKNAME NICKNAME**

1.5.- Mensajes de ganador/perdedor:

Un jugador pierde cuando choca contra su snake o contra otras snakes. Y gana el último snake restante. El servidor detecta esto y envía un mensaje a todos los clientes notificando quién ha perdido o ganado.

S → C : Mensaje de cliente con id ID_CLIENTE que perdio: **'U' ID_CLIENTE 'L'**

S → C : Mensaje de cliente con id ID_CLIENTE que gano: **'U' ID_CLIENTE 'W'**

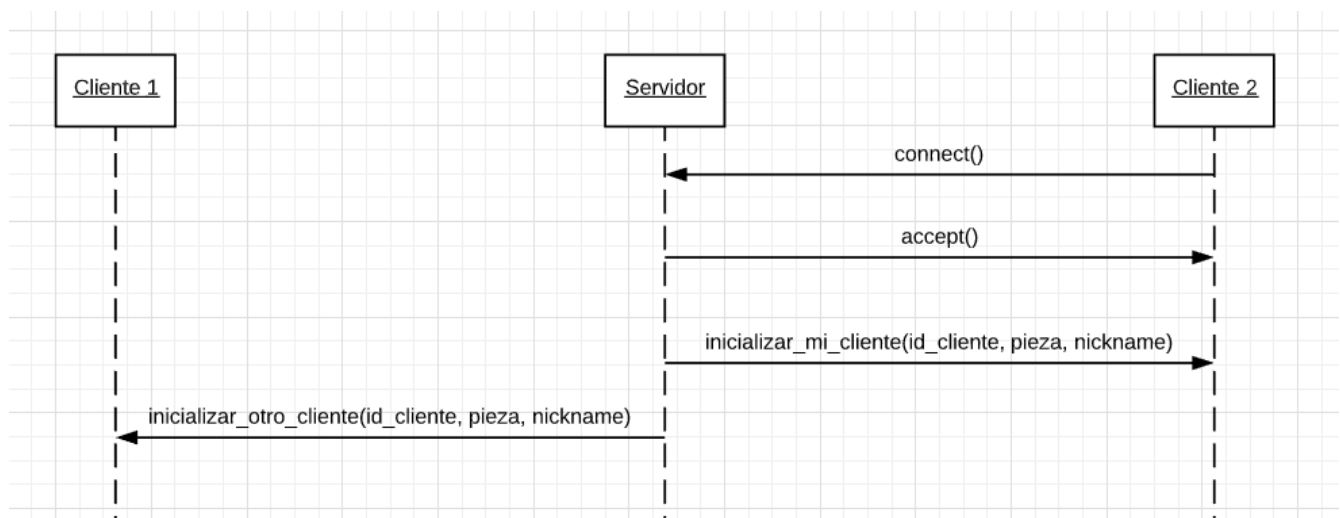
2.- Estructura de datos:

La estructura utilizada es un array dinámico de chars, que puede crecer de tamaño dependiendo de si se insertan datos. En el lenguajes C++ esto se implemeta con un: `vector<char> buffer_data`;

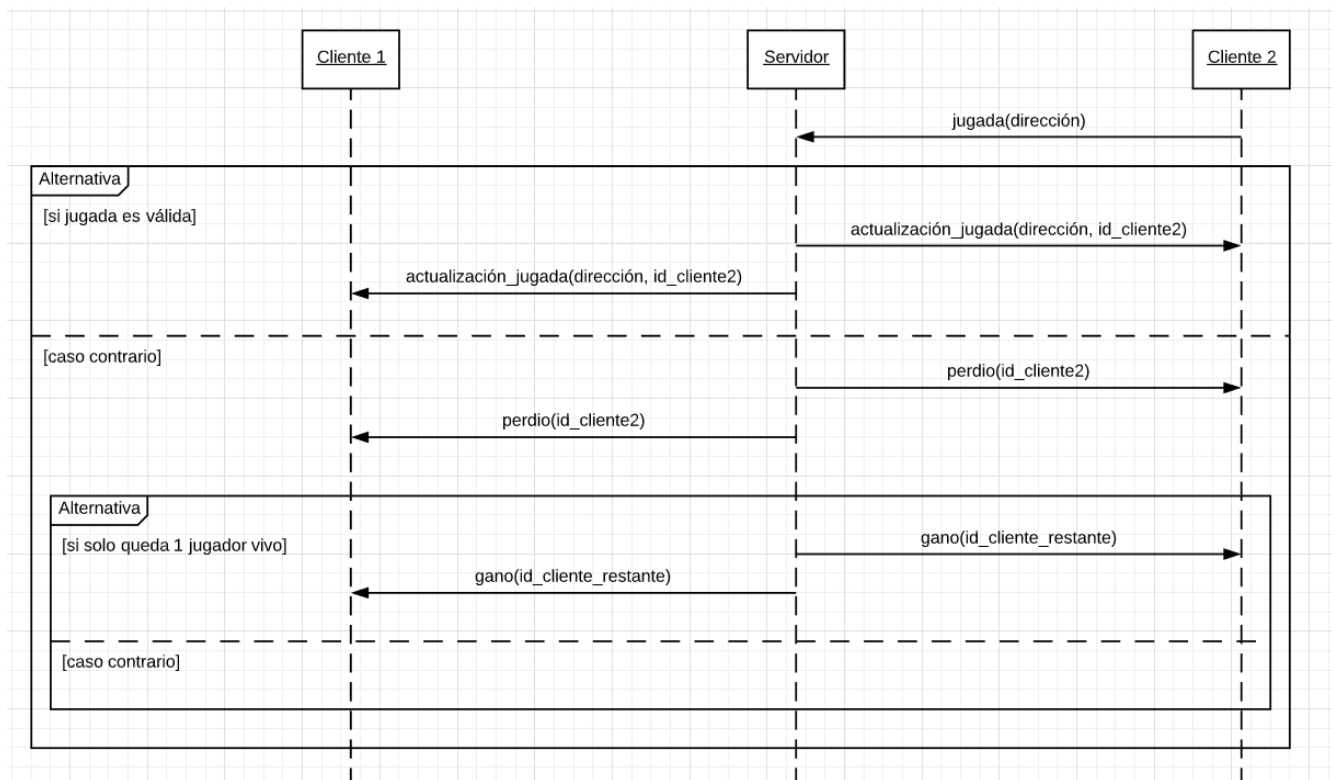
Para llenar este vector de chars "buffer_data" se llama a una función que recibe como entrada un: `vector<pair<char*,int> > buffers_sizes`; que esta lleno de punteros a chars que son arreglos, y los enteros son los tamaños de los buffers. Luego cada buffer dentro de "buffers_sizes" se utiliza para llenar el buffer final "buffer_data".

3.- Diagrama de secuencia:

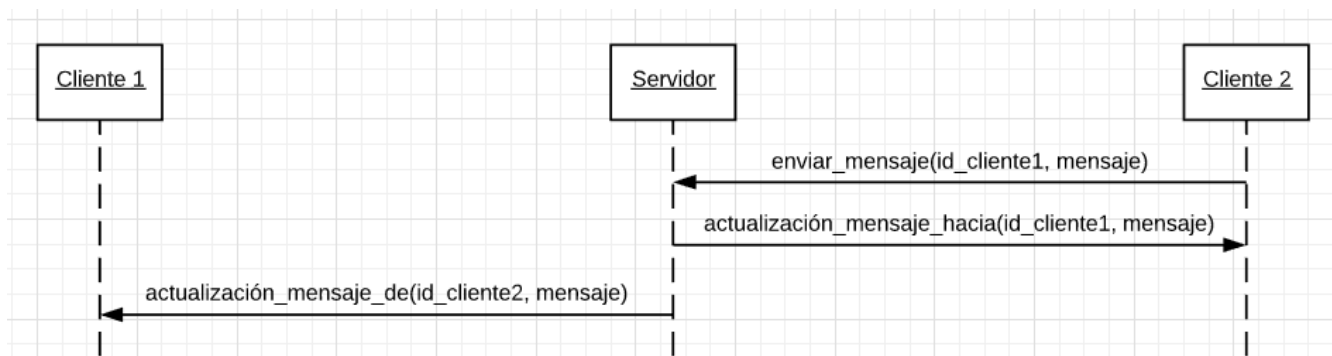
3.1.- Diagrama para inicializar cliente:



3.2.- Diagrama para enviar una jugada y ver si ganó/perdió alguien:



3.3.- Diagrama para enviar mensaje de chat hacia 1 destinatario:



3.4.- Diagrama para cambiar nickname:

