

Proyecto

X-Kating

Grupo

WASTED HORCHATA

Documento de diseño técnico del motor de red

Hito: 1

Fecha entrega: 22-12-2017

Versión: 2

Componentes:

- Luis González Aracil
- Laura Hernández Rielo
- Adrián Francés Lillo
- Pablo López Iborra
- Alexei Jilinskiy

1.Introducción

A continuación se detallan los detalles técnicos del motor de red que va a utilizar nuestro proyecto. Como no se ha desarrollado hasta el momento, se trata únicamente del diseño técnico que se va a seguir, sin entrar en detalles concretos de la implementación.

Al igual que en el otro documento de red, se asume que estamos trabajando sobre la tecnología RakNet, la cual proporciona las funcionalidades básicas para la implementación del motor.

2. Diseño técnico del servidor

El servidor en red se inicializará a partir de un documento XML que inicializará todas las variables del servidor, tales como el puerto de escucha, el número de sockets y la cantidad de jugadores simultáneos. Tendrá dos partes claramente diferenciadas, la **sala de lobby** y el **servidor de juego**.

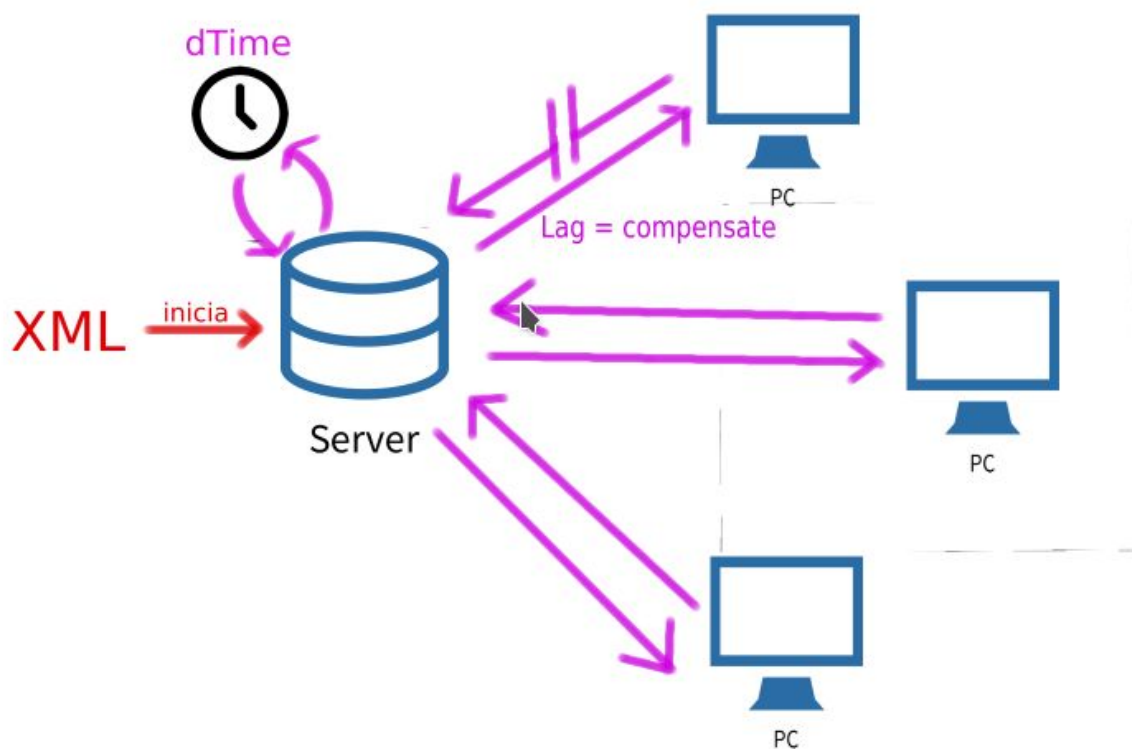
Después de la inicialización, la aplicación del servidor abrirá un puerto de escucha y quedará a la espera hasta que reciba alguna conexión entrante. Cuando lo haga, establecerá la comunicación con dicha conexión y, en caso de cumplir todos los requisitos, le dará acceso a la **sala de lobby**, creandola en el proceso si no estaba creada.

Desde el momento que haya al menos un jugador y, por lo tanto, la sala esté creada, el servidor quedará tanto a la espera de nuevas peticiones en el socket como atendiendo a las acciones del jugador o jugadores en el lobby.

Una vez los jugadores en el lobby decidan iniciar la partida, se cerrará el lobby y comenzará el estado de juego. En primer lugar, se iniciará una pantalla de carga, momento en el cual tanto el servidor como cada uno de los clientes cargarán todos los elementos necesarios del juego, y una vez todos hayan realizado dicha carga, se iniciará el juego simultáneamente para todos.

Desde ese momento, la comunicación consistirá en el siguiente bucle:

- **El servidor se ocupará de los updates**, ya que su versión es la verificada, **y determinará los eventos ocurridos**.
- **Recopilará y retransmitirá estos updates** a todos los jugadores.
- De forma asíncrona durante este tiempo, irá **almacenando los distintos inputs que reciba de los jugadores**, los cuales utilizará para procesar el siguiente update.
- En caso de no tener suficientes inputs de algún jugador, **el sistema de compensación de lag** actuará, generando la información necesaria.
- Finalmente, antes de la siguiente iteración, el sistema **comprobará si lleva demasiado tiempo sin recibir información de algún jugador** en concreto, y si se da el caso, lo echará de la partida.



3. Diseño técnico del cliente

Por el lado del cliente, la gestión de la conexión en red vendrá dada por el NetworkManager, el cual también leerá los datos necesarios para la conexión de un XML. El juego de cada uno de los clientes contará con:

- Varios **NetworkComponents**, asociados a cada uno de los GameObjects del juego que necesiten recibir inputs de la red, los cuales se comunicarán directamente con el manager para leer los datos que reciban del servidor remoto. En ciertos casos, este componente sustituirá a otro u otros para ocupar su lugar. Por ejemplo, en el caso de los otros jugadores, sustituirá a su componentes de IA, ya que lógicamente se tratará de otro jugador en red.
- **El NetworkManager**. Aparte de la comunicación directa tanto con el servidor remoto como con sus componentes asociados, se encargará de obtener y controlar las estadísticas de la conexión, de modo que si ésta cayera se lo notificaría al jugador por pantalla. Además, el propio NetworkManager lanzará ciertos eventos, como por ejemplo las colisiones, y se encargará de aplicar las correcciones necesarias para que la versión del cliente sea igual a la del servidor.

El proceso de comunicación del cliente con el servidor será el mismo que hemos definido en el apartado anterior, solo que a la inversa. Ya que la velocidad de comunicación será distinta para cada cliente dependiendo de la conexión, estos intentarán transmitir la información siempre lo más rápido que sea posible, con el objetivo de minimizar el ping de cada cliente.