

Tres en raya en red

David Blanco Fuentes
Ildefonso de la Cruz Romero
Miguel Ángel Pérez García
Oscar Delgado Miranda

8 de junio de 2015

1. Introducción

Este proyecto consiste en el juego del 3 en raya de manera distribuida(cliente/servidor). Su utilidad principal es proporcionar una distracción lúdica a los usuarios.

En el programa en conjunto, hemos hecho uso de Zmq para las conexiones entre servidor y distintos clientes, junto con la implementación del juego 3 en raya. Principalmente la finalidad del trabajo ha sido trabajar sobre las distintas conexiones y para probarlo hemos implementado el juego.

El servidor actuará como una sala de espera, al cual se conectarán distintos clientes(jugadores) en busca de una partida. Una vez que el servidor tiene a más de un jugador esperando para jugar les pasa a estos sus IP para que comiencen una nueva conexión entre ambos clientes, donde uno actuará como server y otro como cliente.

2. Requisitos previos

1. Estar conectado a una red local
2. Tener las librerías de zmq instaladas (sudo apt-get install python-zmq)

3. Cómo se ejecuta

Debemos de tener en cuenta si vamos usar el juego de manera local para probar su funcionamiento o en una red para hacer uso de este programa y jugar por diversión algunas partidas.

En caso de que se quiera hacer una prueba en un mismo PC hay que tener en cuenta lo siguiente: Primero ejecutamos el servidor (python Server.py), el cuál nos imprimirá la IP y los puerto que está usando. Segundo ejecutamos en un terminal diferente cada cliente(pythn cliente.py), este nos pedirá la IP del servidor, que ya nos había mostrado este, y la IP que usará este cliente para jugar. Es necesario que cada cliente tenga una IP distinta, uno de ellos puede tener la IP que este usando el PC, la misma que usa el servidor, y el otro cliente debe de usar la IP de localhost(127.0.0.1)

Si solo se quiere jugar con otro jugador dentro de una misma red, se decide un PC de dicha red para ejecutar el servidor. Y en los clientes solo ponemos la IP del servidor y la del equipo en el que estemos.

4. Qué nos proporciona Zmq y el funcionamiento del programa

Zmq está destinado a aplicaciones distribuidas asíncronas, así que fue lo primero que necesitábamos para trabajar.

Una biblioteca que importáramos para hacer uso del socket para la transferencia de mensajes, ya sea utilizando UDP o TCP.

Cuando se conectan al servidor(Puerto 5000) los distintos clientes, se establece la conexión mediante REQ(cliente-server), REP(server-cliente). En este momento se incluyen en la cola de espera almacenando su Ip. El servidor actúa de esta forma como una sala de espera para conectar a los jugadores.

Cuando hay más de un cliente se empieza a preparar la partida y el servidor(Publisher) publica mediante PUB a todos los clientes(jugadores) que están suscritos a él por el puerto 6000, los dos jugadores que pueden empezar la partida. Dado que la cola es de tipo FIFO, serán los dos primeros jugadores de la lista. Después de esto se sacan de la lista de espera.

Es el momento de que el jugador1 reciba la IP del jugador2 y al revés, desconectándose del servidor y conectándose entre ellos. A cada uno se le otorgará un perfil(a uno el de servidor con puerto de escucha 7000 y al otro el de cliente). Cuando finalice la partida, jugador1 y jugador2 pueden volver a la sala de espera si así lo eligen.

-En el fichero server.py: Tenemos una cola de clientes. En el momento que la cola contiene dos jugadores, enlaza una partida con los dos primeros jugadores. El servidor sigue escuchando en su puerto nuevas peticiones de clientes

-En el fichero cliente.py: Contiene las funciones de actuar como servidor o cliente, una vez se haya creado la partida. El servidor(el que contiene la cola) es independiente ya a esta conexión. Junto a estas funciones está la de conectar con el servidor principal(cola) y la función que contiene el juego y que será utilizada por cada jugador. El paso de mensajes entre Servidor-jugador y jugador-jugador se realiza mediante RE/REQ.