TNet 3: Example Scene Overviews

The TNet example scenes are designed to help you understand the concepts behind using the networking library while keeping the code as simple as possible. They can all be run directly, however for multiple users they should be run via the menu options after connecting to a server in the Menu scene.

Menu

Menu is used to access most of the other example scenes. In it you can start a server, a server which doesn't use ports, and/or connect to a remote ip address. Once connected you will be presented with buttons to access the other example scenes.

RFCs

Remote Function Calls (RFC) are regular functions which can be called over the network. Using the Send command one client can send information to all other clients in the same scene. This example displays a series of boxes, which when clicked pass a colour value over the network which each client then applies to its local boxes.

Object Creation

The Object Creation example instantiates cubes over the network for all connected clients. It uses a Remote Creation Call (RCC) function to not only instantiate the cube with a position and rotation but with other parameters as well.

Frequent Packets

This example demonstrates one client taking control of a scene's object using TCP packets and updating information quickly over UDP while it is being dragged around the screen. It uses multiple RFC's in unison with MonoBehaviour events to achieve its result.

Chat

The Chat example demonstrates how to implement a chat system in multiplayer games. It uses RFC's to pass what is written by one user to all other clients connected to the same channel.

Chat also operates similar to an in game console by reading and writing custom server data using the /get and /set commands. Any time one of the server values is set, all other clients will be automatically notified of it.

Movement

The car in this example demonstrates how to update the position of a game character without flooding the network with packets. The user's keyboard/joystick input is synced frequently enabling each client can apply the same values to generate the new position of the vehicle. The vehicle's rigidbody is synchronized less frequently, letting each client run a local physics-based simulation for a smoother experience that gets corrected only periodically.

Multiple Channels

This example emulates a scaled down version of a large world and shows how it's divided up into separate network channels. When the car drives from one quadrant to another, the channel that it is a member of changes. This reduces the amount of traffic any one particular client sees while still giving the appearance of a large open world.

Example AutoJoin

The AutoJoin example connects to the specified IP address and port on startup. If your game has a central server or if you want to automatically connect to some central server on startup for testing, the AutoJoin example is a good place to see how that's done.

Example AutoSync

AutoSync is a script included with TNet to periodically sync a simple object over the network. You can customize which inspector fields to update and adjust how often they synchronize without writing any code.