

UtVRPN

Version 393, © PublicVR

(unedited first draft)

(UtVRPN will be replaced very soon with a complete rewrite, VrpnUT.)

Introduction

UtVRPN is a modification to the game [Unreal Tournament 2004](#) (UT2004) which allows a remote (controller) process to move the viewpoint of a single player via a UDP connection. UtVRPN is packaged as a “mutator”, a standardized package used in UT2004 for modifications to the game. This mutator is freely available to the public (see below).

UT2004 is a multiplayer networked game where users connected via a standard LAN or the Internet can share a single virtual world. One way to do that is for one player to “host” everyone else by running a “Listen Server” instance of the game on their computer. They will have their own player on their own machine, and their machine will referee communication with all the others. This gives the hosting player certain administrator privileges over the game, but once play starts s/he is generally coequal with the other players.

When active, UtVRPN modifies the Listen Server mode by opening (for read only) a UDP port on the local machine. This allows an external process, either on the machine or elsewhere on the network, to write packet to the UDP port. The modified listen server picks up those packets and uses the instructions there to move the player’s avatar on that machine. For example, when UtVRPN reads a packet which tells it to turn the local player’s avatar 30° to the left, the local user will see their entire view turn 30°. Other players in the game see that player’s avatar turned 30° to the left. While UtVRPN is active, press the "J" key to toggle between normal UT2004 navigation and UDP driven movement.

Compatibility

UtVRPN works with all versions of CaveUT 2.5, is only guaranteed to work with UT2004 if you run the [3369 code patch](#), on the local machine. It will upgrade the UT2004 installation to the final version (number 3369) of the [Unreal Engine 2.5](#). The UtVRPN code is short simple and changes the game in only one place. Because of this, a direct clash with other mutators and other modifications to the game are unlikely. Incompatibilities may result from the way in which the game engine interprets UtVRPN’s command to move. It is not quite the same as a top-level command from the user via some game controller or the keyboard. We have not extensively tested this and would welcome anyone else’s efforts to do so.



Figure 1: UtVRPN is used in this cave to drive the user's viewpoint through the scene.

UtVRPN was primarily developed for use with [CaveUT](#), another mutator which allows UT2004 to produce a single contiguous view across multiple screens. The screens can be in any orientation to each other, usually arranged as an enclosure around the user, similar to the [CAVE](#). UtVRPN also works well other software that projects the image into a dome display such as [DomeUT](#).

UtVRPN is intended to be the very beginning of a "client" (information receiver) for the open-source [Reality Peripherals Network](#) (VRPN). VRPN handles a range of VR interface devices, such as HMDs, tracking gloves, treadmills, and much more. It allows the client to communicate with all these devices via a generic format. UtVRPN's UDP packets are in a VRPN compatible format.

Installation

First, download `UtVRPN-v393.zip`, and unpack the `UtVRPN-v393` folder onto your desktop or into some convenient directory. In the following instructions, all pathnames to files begin at the `UtVRPN-v393` folder as their root. So, `Documentation/UtVRPN.pdf` refers to a PDF file in the `documentation` subfolder under the `UtVRPN-v393` general folder.

If you are using UtVRPN **without** CaveUT copy the contents of the `Distribution` folder into the root folder for your UT2004 installation, which is `C:\UT2004` by default, but you can name it anything you want. Allow it to copy files into the "System" subfolder. If you are using UtVRPN **with** CaveUT, Then don't copy the `System/CaveUT.ini` file. Instead, copy and paste `[UtVRPN.VRPNPawn]`, `[UtVRPN.UDPVRPN]`, and `[UtVRPN.VRPNmut]` sections from the `CaveUT.ini` file in the UtVRPN distribution into `UT2004/System/CaveUT.ini`. (This silliness will go away when UtVRPN has its own *.ini file.) You can now load and use UtVRPN as a standard UT2004 mutator.

Usage

To activate UtVRPN, start a listen server with the CaveUT and UtVRPN mutators active. Connect the spectator clients to the server in the normal way. To move the client using UtVRPN, simply send one or more UDP packets to the port (on the server machine) UtVRPN is listening to. Press the "J" key to toggle between normal UT2004 navigation and UDP driven movement.

The UDP Packet is a space delimited string of 5 floating-point numbers in this format:

```
Linear_Velocity(m/s) Rotational_Velocity(deg/s) Head_delta-X(cm) Head_delta-Y(cm) Head_delta-Z(cm)
```

Linear Movement and Rotational Movement continue at the specified velocity until the next packet is read. Linear Velocity is positive when moving forward and negative when moving in reverse. A value of zero stops the movement. Rotational Velocity is positive when rotating clockwise (right) and negative when rotating counter-clockwise (left). Again, a zero value stops rotation.

The Head delta* parameters are currently unused. They are included for VRPN compatibility. All parameters for UtVRPN are contained in the configuration file UT2004/System/CaveUT.ini, in two sections: (Soon, we will move these parameters into their own file, UtVRPN.ini.)

```
[UtVRPN.VRPNPawn]
bTreadmill=true           # Activates UtVRPN when true.
VelocityStep=50.000000    # param unused and will be removed.
bControlRotation=false    # Activates UDP control of angular
                           # motion when true

[UtVRPN.UDPVRPN]
VRPNPort=50000           # Specifies the port UtVRPN listens
                           # to for new packets

[UtVRPN.VRPNmut]         # These parameters define the
BaseEyeHeight=38.00      # collision volume around the
CollisionRadius=25.00     # user's viewpoint.
CollisionHeight=44.00
```

The last set of parameters defines the collision volume around the user. UtVRPN pushes the user viewpoint in the direction specified by each packet. However, world collision detection is still in effect; that plus the collision volume allows terrain following.

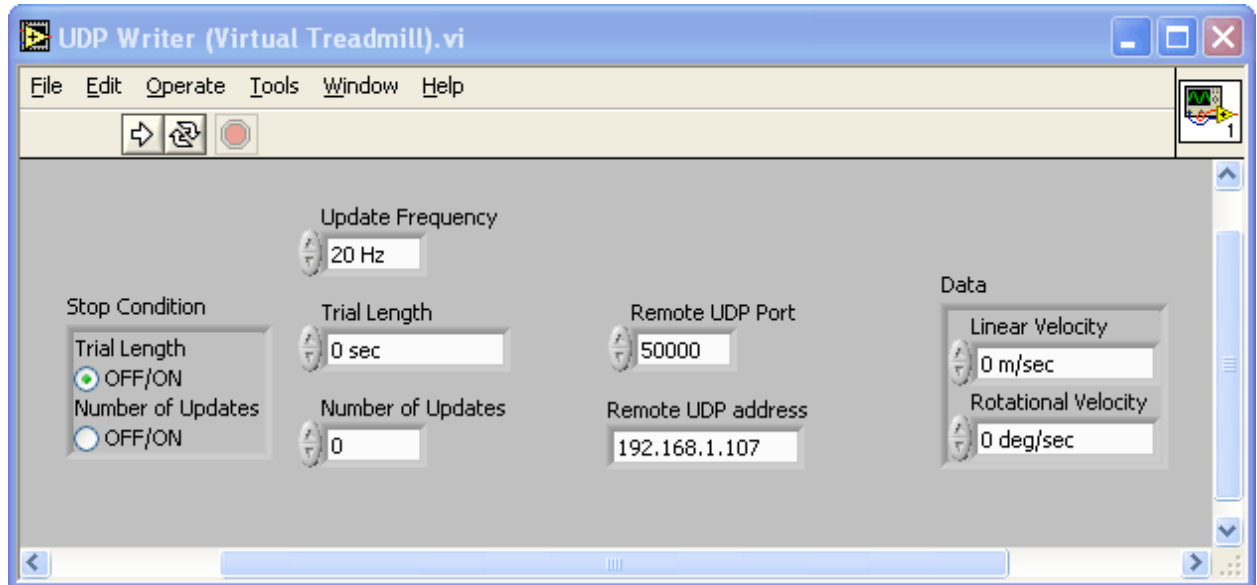
Please direct any questions you have to jeff@planetjeff.net

Test Packet Writer

The Labview-PortWriterUDP folder contains a simple testing program that will write UtVRPN format packets to particular port on a particular machine. Is written in LabVIEW, a visual scripting language usually intended control devices in a research laboratory. To run the port writer, you must first install the LabVIEW engine, which is free for distribution with LabVIEW programs. On your PC, execute “Labview-PortWriterUDP\Installer\Volume\setup.exe” and follow the instructions. Once that is done, you can run the UDP Writer by executing

```
“Labview-PortWriterUDP\UDP Writer.exe”
```

You will see the following dialogue open your desktop, with the default values shown.



The “Remote UDP address” is simply the IP address of the machine where you are Unreal Server (with UtVRPN active) is running. If you’re running the report writer and server on the same machine local machine, you can enter its general IP address or 127.0.0.1. The port writer to it will send packets to the “Remote UDP Port” specified on the unreal server’s machine. We chose port 50000 for no particular reason, except that it does not interfere with UT2004 in any way that we have noticed. You should make sure that you don’t have any other software on the machine listening at that port (unlikely), or whatever port you specified.

The “Update Frequency” parameter governs how many packets per second report writer will send. That the default value, 20Hz, it will send 20 per second. These packets will not be different from each other – they will merely repeat the parameters entered in the other boxes. This is actually useful for testing network speed and response times for the UT2004/UtVRPN/CaveUT installation.

“Trial Length” limits how long report writer will keep sending packets, while the “Number of Updates” parameter limits the total number of packets sent. However, the limitations will only take effect if the “Stop Condition” for it is active (on). For example, when the “Trial Length” toggle under “Stop Condition” is checked, the Port writer will stop sending packets when the length of the trial is reached. If it is not checked, the port writer will ignore the time limit.

The linear velocity parameter sets a continuous forward or backward motion for the user’s viewpoint, (Z-axis) at the velocity specified in meters per second. (We’ve assigned 52 unreal units per meter.) If the value is positive, the user moves forward. If the value is negative, the user moves backward. The value entered will be written to z-axis parameter of all packets sent from the UDP writer the next time it is run. Rotational Velocity refers to rotation around the world’s Y-Axis, with

a positive value rotating the user's viewpoint to the right and a negative value rotating the user's viewpoint to the left.

To activate the UDP Writer, click on the white arrow button in the upper left-hand corner. It will turn grey, and the button with the red circle will go from grey to full color. If you want to halt execution of the Port writer before it is done, press the red circle. Unfortunately the Port writer does not always terminate correctly. You may have to close its window completely and restart it for a new test.

Bugs and Fixes:

We plan to give UtVRPN its own independent configuration file, UtVRPN.ini

Credits:

[Jonathan Hagewood](#) at [Psyonix Inc.](#) (Programming, primary author)

[Jeffrey Jacobson](#) at [PublicVR](#) (Specifications, Testing, Documentation)

[James Cook](#) at the [Medical Virtual Reality Center](#) (UDP Writer, Testing)

[Patrick Sparto](#) at the [Medical Virtual Reality Center](#) (Specifications)

[Medical Virtual Reality Center](#) (Funding)

Download:

<http://publicvr.org/UtVRPN/UtVRPN.zip>

End User License Agreement:

Copyright (c) 2005 PublicVR. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

This product employs the software, UtVRPN(tm)© PublicVR, free to the public at "www.publicvr.org".

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. UtVRPN is a trademark of PublicVR. Accordingly, products derived from this software may not be called "UtVRPN", nor may "UtVRPN" appear in their name, without prior written permission from PublicVR.
5. All versions of UtVRPN v198 are the property of PublicVR. However, UtVRPN is a derivative work based on a Unreal Tournament 2004 by Epic Games. As such, it is subject to the End User License Agreement distributed with the installation package for Unreal Tournament 2004.

This software is provided "as is" and any expressed or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall PublicVR or their contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.