

What's new in Yaw VR Emulator v.0.2

- SimTools support
- Automatic saving of previous settings

Yaw VR Emulator v. 0.2

Description

Yaw VR Emulator emulates the functionality of Yaw VR Motion Simulator, supporting you to develop softwares targeting the real simulator, without the need of an actual device.

Features:

- Just like the real device, Yaw VR Emulator is discoverable on local network
- Allows games to connect to it
- Follows TCP and UDP commands, sends responses, shows them on the screen
- Shows current state (Available, Connected, In game)
- Shows a 3d simulator model from 4 different directions
- Simulates motion
- Shows rotation limits
- **SimTools support**

Usage

Be sure that the device on which the emulator runs is connected to the same network as the device you are connecting to the emulator.

If you have difficulties in communication between emulator and your game:

- Check if your network / router allows sending TCP and UDP messages to the chosen ports.
- Set your firewall settings accordingly

After starting the emulator, in the left bottom corner of the screen you can find current TCP and UDP ports and IP address of the emulator. The default ports are 50010 for UDP and 50020 for TCP. You can modify port numbers by changing the value of their fields.

How to control the emulator

- Using with direct commands (authenticated):
 - You can find emulators connected to your local network by sending a DEVICE_DISCOVERY UDP broadcast (on IP 255.255.255.255 and emulators' UDP port), and the available emulators will send a DEVICE_DISCOVERY_RESPONSE command to the sender device, with their id, name, TCP port and status.
You are also able to directly target your emulator without this discovery phase.
 - Then you need to connect to the emulator's TCP server with your TCP client.
 - Sending CHECK_IN (TCP) command, you can reserve the emulator for your gaming session.
(CHECK_IN_ANS is sent back as response)
 - Send START command to tell the emulator that it should follow rotation values sent with UDP messages.
(Same command is sent back as response)
 - Send rotation values with SET_POSITION (UDP) commands, in this phase the emulator will rotate its 3d simulator model according to the given euler angles, and will send its current positions back with ACTUAL_POSITION (UDP) commands.
 - Sending STOP command tells the emulator not to follow rotation commands until the next START command (Same command is sent back as response)
 - Sending EXIT command sets the emulator to be available for other games (Same command is sent back as response)
- **Using SimTools :**

- In the bottom-left corner of the window, check the Using SimTools checkbox
- Enter the output bit range, which you have set in the Interface Settings of SimTools

TCP commands

To be able to send tcp commands, the game has to connect to the emulator's TCP server.

Every tcp command starts with a one byte long identifier, followed by bytes representing command parameters if there are any. Floating point and Integer number parameters are sent on 4 bytes, string parameters converted to byte array with ASCII character encoding.

You have to send number types in big endian byte encoding. YawVR Unity game package and the emulator handles the conversion to little endian format when runned on little-endian systems.

If the emulator accepts a TCP command, the default response is that it sends back the message to the game, if there is no other informative response to it.

TCP commands sent only from the game to the emulator:

- **CHECK_IN**
 - Command id: 0x30 (1 byte)
 - Parameters: int UDPListeningPort (4 bytes), string gameName,
 - If the emulator is available, accepts connection from the game and becomes reserved
 - CHECK_IN_ANS response is sent back to the sender

TCP commands between the emulator and the game (both direction):

- **START**
 - Command id: 0xA1 (1 byte)
 - After receiving START command, the emulator follows rotation orders from the game
 - (Same command is sent back to the sender as response)
- **STOP**
 - Command id: 0xA2 (1 byte)
 - After receiving STOP command, the emulator stops accepting rotation orders from the game
 - (Same command is sent back to the sender as response)
- **EXIT**
 - Command id: 0xA3 (1 byte)
 - After receiving EXIT command, the emulator disconnects from the game and becomes available
 - (Same command is sent back to the sender as response)
- **SET_TILT_LIMITS**
 - Command id: 0x40 (1 byte)
 - Parameters in order: int pitchForwardLimit (4 bytes), int pitchBackwardLimit (4 bytes), int rollLimit (4 bytes)
 - (Same command with actually accepted values is sent back to the sender as response)
- **SET_YAW_LIMIT**
 - Command id: 0x70 (1 byte)
 - Parameter: int yawLimit (4 byte)
 - (Same command with actually accepted value is sent back to the sender as response)

TCP commands sent only by emulator to game:

- **CHECK_IN_ANS**
 - Command id: 0x31 (1 byte)
 - Parameters: string answer
 - If the device is available, the answer is: "AVAILABLE"
 - If the device is in use by another game, the answer is:
"RESERVED" + ";" + /name of the game using the simulator/ + ";" + / IP address from where the simulator is used /
- **ERROR**
 - Command id: 0xA5 (1 byte)
 - Parameters: string description

UDP commands

UDP commands are ASCII encoded strings.

UDP messages sent from the game to the emulator:

- **DEVICE_DISCOVERY**
 - Message: "YAW_CALLING"
 - Used as broadcast message, when an emulator receives this command, sends DEVICE_DISCOVERY_RESPONSE to the sender
- **SET_POSITION**
 - Yaw, pitch and roll euler angles in the following format:
 - Example: "Y[000.00]P[359.99]R[342.54]"
 - Values are from 000.00 - 359.99 (use 000.00 instead of 360.00)
 - (no response command)

UDP messages sent from the emulator to the game:

- **DEVICE_DISCOVERY_RESPONSE**
 - If the emulator is available:
"YAWDEVICE;" + /Simulator ID/ + ";" + /Simulator name/ + ";" + /TCP server port/ + ";AVAILABLE"
 - If the emulator is reserved:
"YAWDEVICE;" + /Simulator ID/ + ";" + /Simulator name/ + ";" + /TCP server port/ + ";RESERVED"
- **ACTUAL_POSITION**
 - Yaw, pitch and roll euler angles in the following format:
 - example: "Y[000.00]P[359.99]R[342.54]"
 - Values are from 000.00 - 359.99 (000.00 instead of 360.00)