

UnityOSC + TouchOSC Integration

TouchOSC is an application for iOS and Android which facilitates remote control and receives feedback from any software or hardware that implements the OSC or MIDI protocols such as Max/MSP, Pure Data, Ableton Live to name a few.

Getting Started:

1. Download and install TouchOSC application for iPad or iPhone (<http://hexler.net/software/touchosc>)
2. Download and install the appropriate version of TouchOSC Bridge on your desktop. This will act as an interface in the communication between your desktop and the TouchOSC application on the iPad/iPhone.
3. Follow the steps specified in the *"Getting Started: MIDI with TouchOSC Bridge"* (<http://hexler.net/docs/touchosc-getting-started-midi>) on the TouchOSC site to setup TouchOSC Bridge as well as the app and configure it to work. (both your system and iPad/iPhone needs to be connected to the same Wi-Fi network).
4. Make sure the Layout selected in the options menu of the TouchOSC app on iPad/iPhone is "Simple".
5. To test the connection, go to "touchosc_integration/Assets" folder inside the "tests" folder in the UnityOSC repository and copy paste all its contents inside the main "Assets" folder in the repository's root.
(Note: If you have already copy-pasted contents of the main "test" folder before, then doing Step 5 will overwrite the oscControl and oscHandler C# scripts)
6. Double-click on the "TestScene" to open it inside the Unity3D project
7. Hit Play.
8. If you were able to configure it correctly, then the leftmost fader should be able to control the size.
9. This snippet of code from oscControl.cs controls the scale of the Cube.

```
//converts the values into MIDI to scale the cube
float tempVal = float.Parse (item.Value.packets [lastPacketIndex].Data [0].ToString ());
cube.transform.localScale = new Vector3 (tempVal, tempVal, tempVal);
```
10. In addition, the oscControl.cs will randomly send messages from your system to the TouchOSC app which will control all the middle, yellow-coloured faders on their own.

```
if (UnityEngine.Random.value < 0.01f) {
    randVal = UnityEngine.Random.Range (0f, 0.7f);
    OSCHandler.Instance.SendMessageToClient ("TouchOSC Bridge", "/1/fader1", randVal);
    OSCHandler.Instance.SendMessageToClient ("TouchOSC Bridge", "/1/fader2", randVal);
    OSCHandler.Instance.SendMessageToClient ("TouchOSC Bridge", "/1/fader3", randVal);
    OSCHandler.Instance.SendMessageToClient ("TouchOSC Bridge", "/1/fader4", randVal);
}
```

For questions, assistance or further information contact me at lightnarcissus@gmail.com

Follow me on Github (<http://www.github.com/lightnarcissus>) or Twitter (<https://twitter.com/lightnarcissus>)