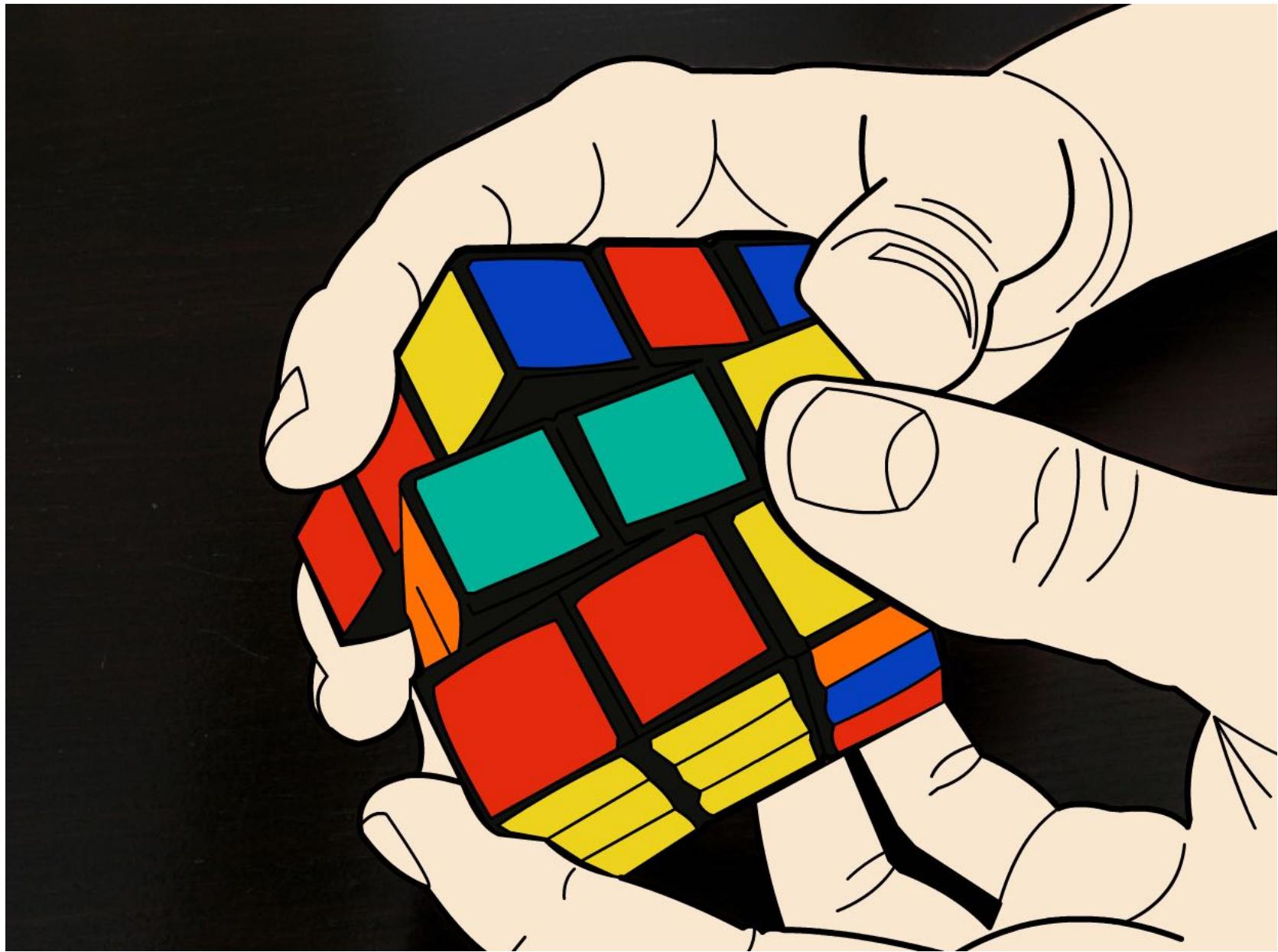


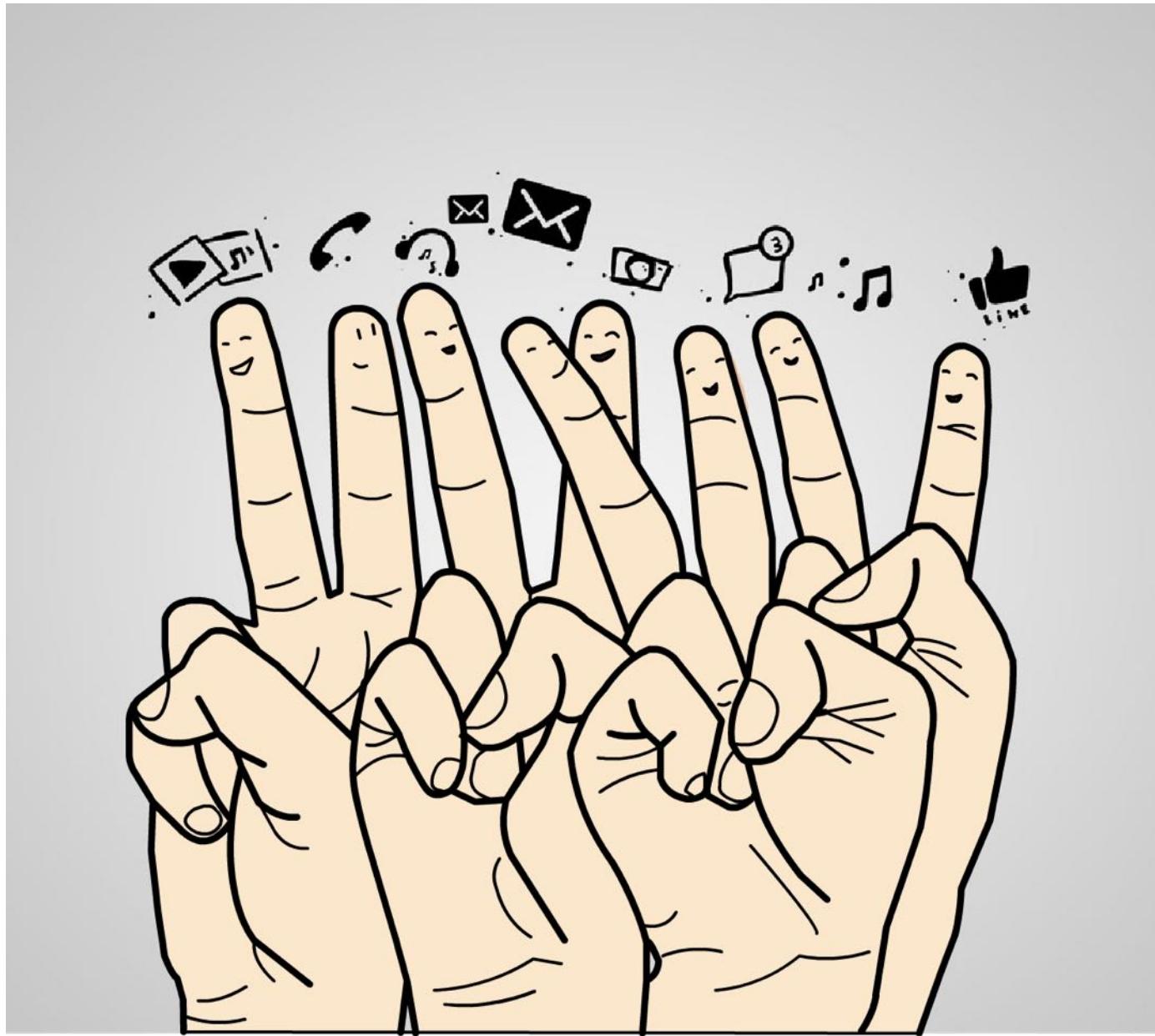
# **Interaction Design & Virtual Reality**

**Liwei chan 詹力韋**  
Assistant Prof.

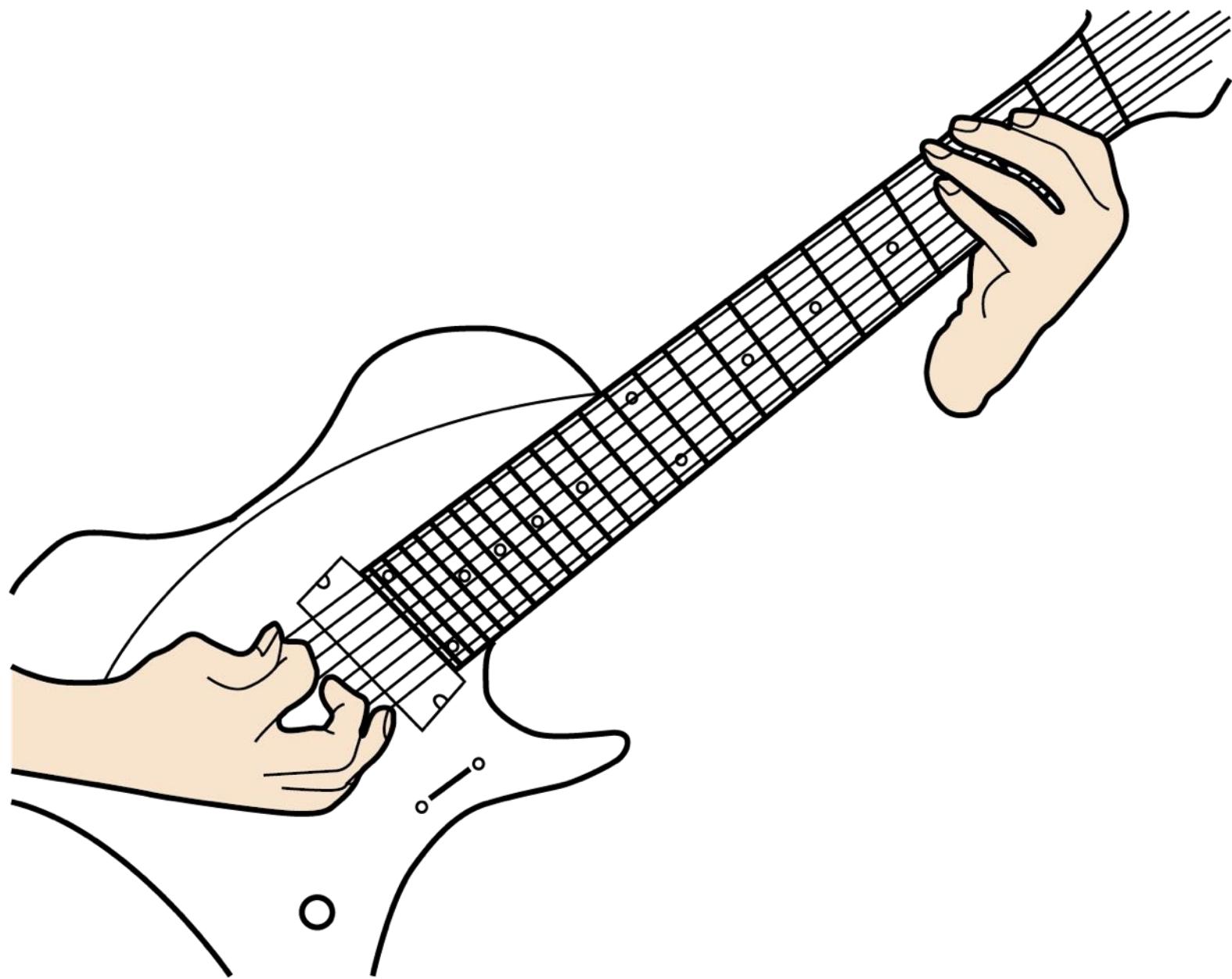
**2016.11.08**





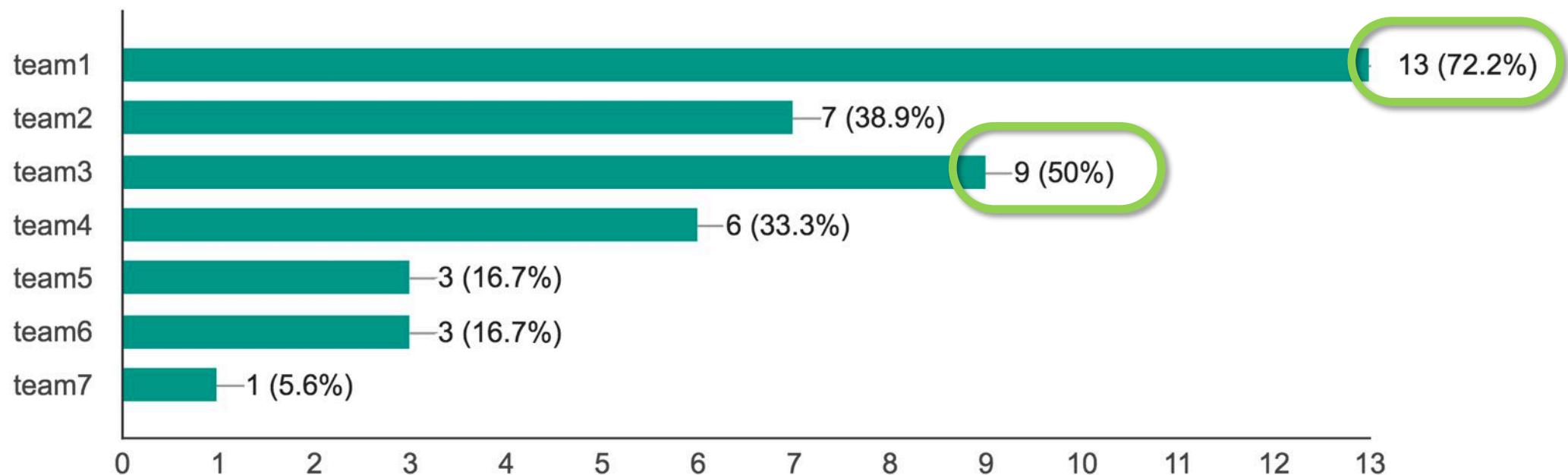


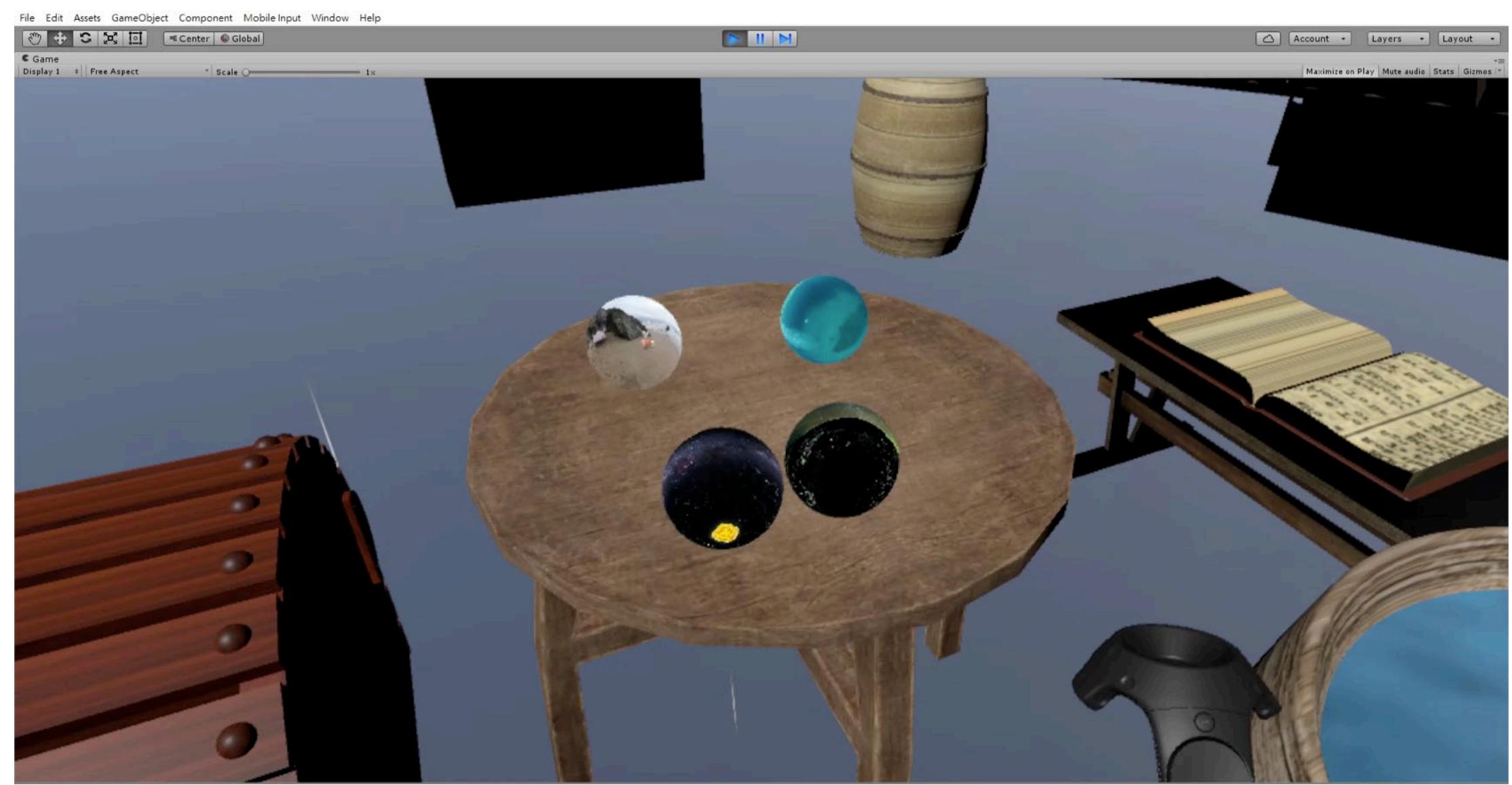






(18 responses)







# **osc (open sound control)**

simple yet powerful protocol for real-time  
control among networked computers

# Management Problem - Solution



get directly down

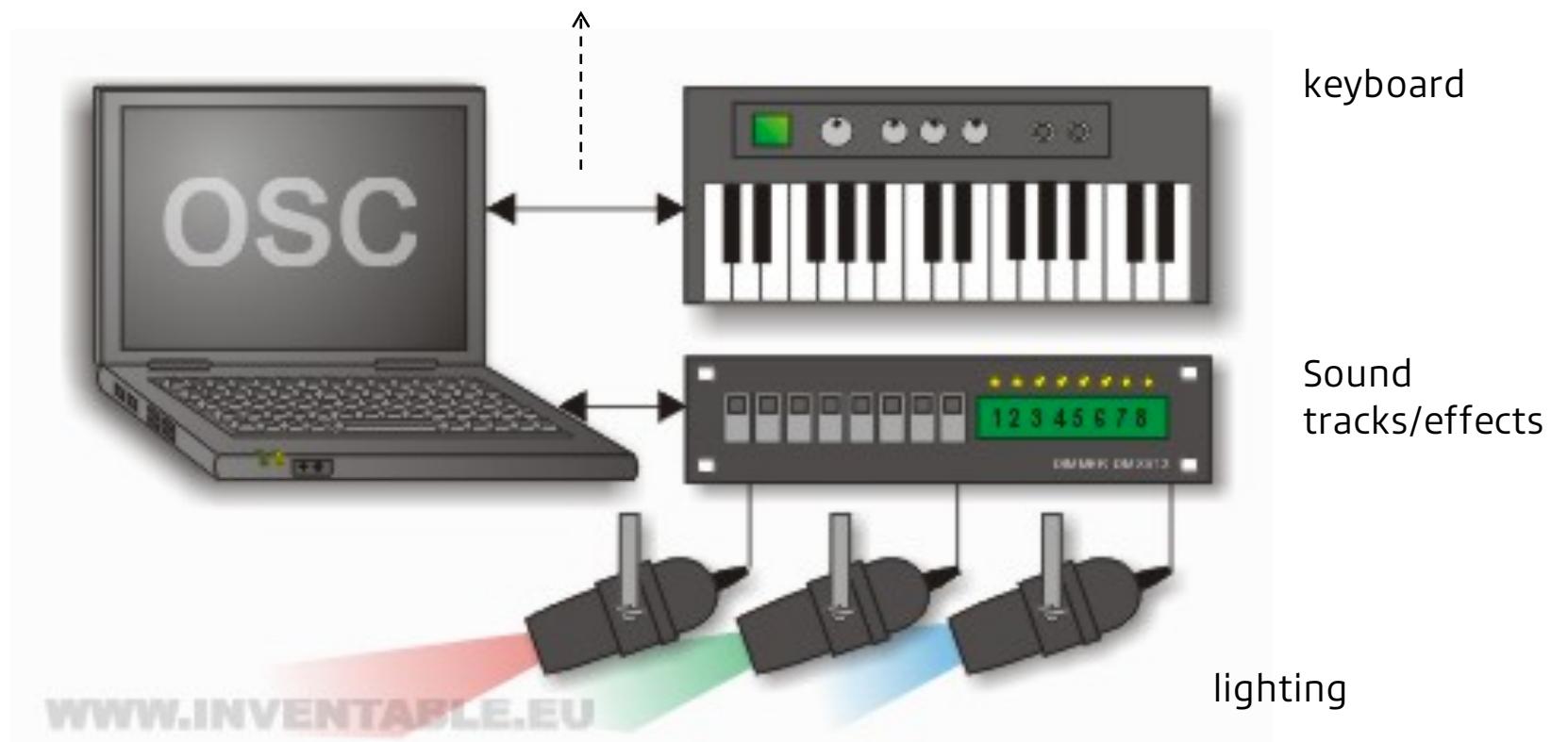


Maurits Burgers



Ivo Riemann

they all wired together,  
but each of them running on different  
protocols/designs/algorithms



migration of OSC to  
PCs communicating over network

# Reactable

an electronic musical instrument.

Developed at **2007**.

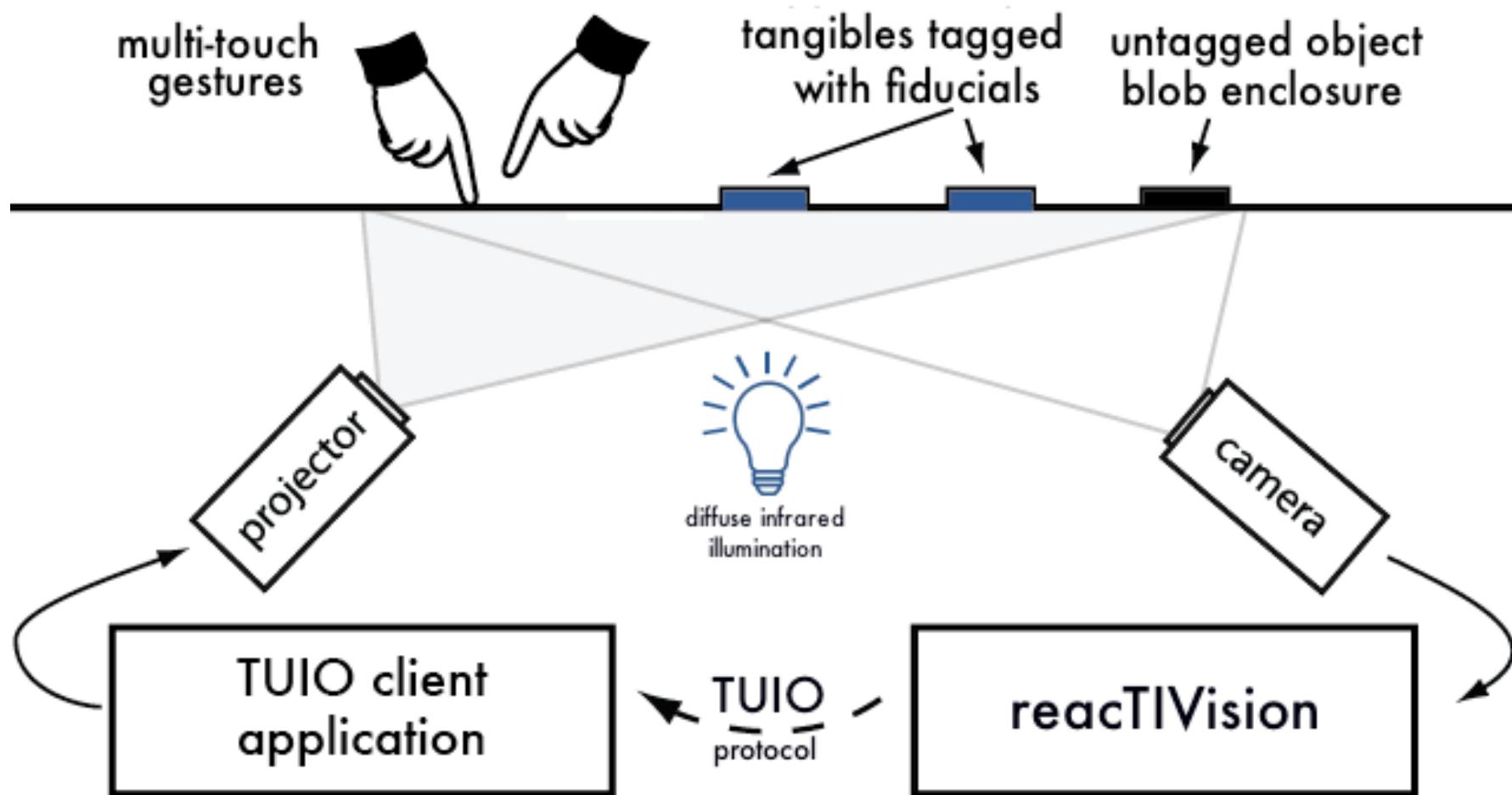
Have been presented at Ars Electronica,  
NIME and SIGGRAPH



# TUIO

the initial publication of the TUIO protocol specification  
to the public domain as part of the [Reactable](#) synthesizer





## Source Image



## Tracked Image



### Source Properties

- CAMERA SETTINGS (V)
- FLIP VERTICAL (J)
- FLIP HORIZONTAL (H)

### GPU Properties

- GPU MODE (G)

### Communication

- SEND TUOI OSC (T)
- SEND TUOI TOP | FOR FLASH (F)
- SEND HEIGHT & WIDTH

### Calibration

- ENTER CALIBRATION (C)

### Files

- SAVE SETTINGS (S)

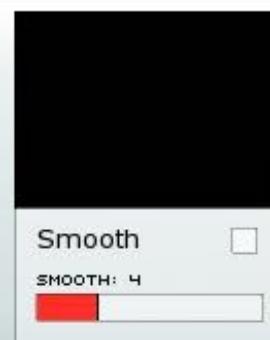
Calc. Time [ms]: 12

Camera [Res]: 640 x 640

Camera [fps]: 24

Press spacebar to toggle fast mode

| ~ Itbeta.nuigroup.com



# features of OSC

- Open-ended, dynamic, URL-style symbolic naming scheme
- **Symbolic** and high-resolution **numeric** argument data
- Pattern matching language to specify multiple recipients of a single message
- High resolution time tags
- "Bundles" of messages whose effects must occur **simultaneously**
- Query system to dynamically find out the capabilities of an OSC server and get documentation

# many open-source implementations

<a href="#">MouseTrap</a>	<a href="#">Python - Simple OSC</a>	<a href="#">Liblo: Lightweight OSC API</a>	<a href="#">Gluon</a>
<a href="#">KWLIVE Realizer</a>	<a href="#">Python - scosc</a>	<a href="#">Csound</a>	<a href="#">Ethersense</a>
<a href="#">KWLIVE hid2osc</a>	<a href="#">pure data</a>	<a href="#">Matlab OSC</a>	<a href="#">CNMAT Connectivity Pro</a>
<a href="#">ROB - Remote over Bluetooth</a>	<a href="#">Net::OpenSoundControl Perl Modules</a>	<a href="#">Open Sound Control for PHP</a>	<a href="#">Smart Controller</a>
<a href="#">oscjoy</a>	<a href="#">OpenSoundControl for MaxMSP</a>	<a href="#">SwingOSC Java Server</a>	<a href="#">Matrix3 Audio Show Co</a>
<a href="#">MXWendler Realtime Compositing</a>	<a href="#">Ruby OSC implementation</a>	<a href="#">cl-osc</a>	<a href="#">The WiSe Box - Wireless</a>
<a href="#">DiABlu</a>	<a href="#">Java OSC</a>	<a href="#">NetUtil OSC Library for Java</a>	<a href="#">La Kitchen Hardware</a>
<a href="#">Ventuz Realtime 3D Rendering (.net)</a>	<a href="#">Delphi/FreePascal OSC</a>	<a href="#">DSSI ("Disposable Soft Synth Interface")</a>	<a href="#">Monome</a>
<a href="#">OSCulator</a>	<a href="#">J# OSC (.NET 2.0)</a>	<a href="#">Squeak OSC</a>	<a href="#">Make Controller Kit</a>
<a href="#">Inventec, a C/C++/Python OSC sequencer from the UPIC</a>	<a href="#">oscP5</a>	<a href="#">oscpack</a>	<a href="#">BluePD</a>
<a href="#">Damp</a>	<a href="#">Bespoke OSC (.NET 3.5)</a>	<a href="#">OSCgroups</a>	<a href="#">micro-OSC</a>
<a href="#">Nyquist</a>	<a href="#">Erlang OSC Application</a>	<a href="#">WOscLib</a>	<a href="#">oosc - open open sound</a>
<a href="#">a steins - sketch for a RT modular video software</a>	<a href="#">ruby-osc</a>	<a href="#">OSC-Kit</a>	<a href="#">Bowsense</a>
<a href="#">Braun</a>	<a href="#">osc-ruby</a>	<a href="#">Chuck =&gt; OSC</a>	<a href="#">Arduino, OSC, iPhone a</a>
<a href="#">A lib for making OSC messages for SuperCollider</a>	<a href="#">OSC API for SuperCollider</a>	<a href="#">Jamoma</a>	<a href="#">Milkymist One: open ha</a>
<a href="#">Macromedia Flash</a>	<a href="#">TOSC + OSC port based on Python</a>	<a href="#">OSC.net (V1.4.1)</a>	<a href="#">Arduino, Processing, OS</a>
<a href="#">q3osc</a>	<a href="#">liboscparse: liblo for embedded systems</a>	<a href="#">LiveAPI for Ableton Live</a>	<a href="#">The Missing Link OSC/N</a>
<a href="#">TouchOSC</a>	<a href="#">osc-clj - Clojure OSC Library</a>	<a href="#">Oscal Scripting Language</a>	<a href="#">Serpent</a>
<a href="#">mrrmr</a>	<a href="#">OscPkt</a>	<a href="#">Lily</a>	<a href="#">ofxOsc</a>
<a href="#">vvvv</a>	<a href="#">Open Sound World</a>	<a href="#">Python - Simple OSC</a>	<a href="#">SynOSCopy "SYN" Nam</a>
<a href="#">TJShow (show controller software)</a>	<a href="#">devosc: a unix device for reading OSC messages</a>	<a href="#">Python - scosc</a>	
<a href="#">OscVstBridge</a>	<a href="#">Mirage</a>	<a href="#">pure data</a>	
<a href="#">OSCRemote</a>	<a href="#">MadJACK - MPEG Audio Deck</a>	<a href="#">Net::OpenSoundControl Perl Modules</a>	



<http://hexler.net/software/touchosc>

its free



so for,  
its interesting...  
5 years ago.



# OSC for Arduino

<https://github.com/CNMAT/OSC>

CNMAT / OSC

Watch 55 Star 194 Fork 60

Code Issues 24 Pull requests 3 Projects 0 Wiki Pulse Graphs

OSC: Arduino and Teensy implementation of OSC encoding <http://cnmat.berkeley.edu/oscuino>

169 commits 1 branch 1 release 11 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

adrianfreed	Latest teensy fixes	Latest commit 543a1cF on Jul 17
Applications	Latest teensy fixes	4 months ago
examples	Merge branch 'master' of https://github.com/CNMAT/OSC	a year ago
test	Suite of basic validation tests	a year ago
.gitignore	Removed SPI stream stuff	3 years ago
API.md	fix bad function name in API doc	a year ago
LICENSE	created a markdown version of the README	a year ago
OSCBoards copy.h	Latest teensy fixes	4 months ago
OSCBoards.h	Latest teensy fixes	4 months ago
OSCBundle.cpp	bundle methods return '*this' to enable chaining	a year ago
OSCBundle.h	bundle methods return '*this' to enable chaining	a year ago

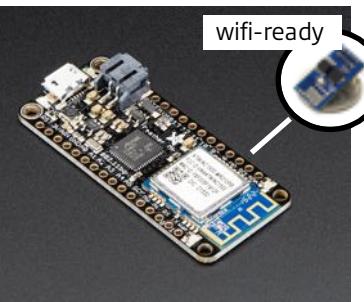
# OSC for Arduino

---

This is an Arduino and Teensy library implementation of the [OSC](#) (Open Sound Control) encoding. It was developed primarily by Yotam Mann and Adrian Freed at CNMAT where OSC was invented. It benefits from contributions from John MacCallum, Matt Wright, Jeff Lubow and Andy Schmeder and many beta testers.

## Features:

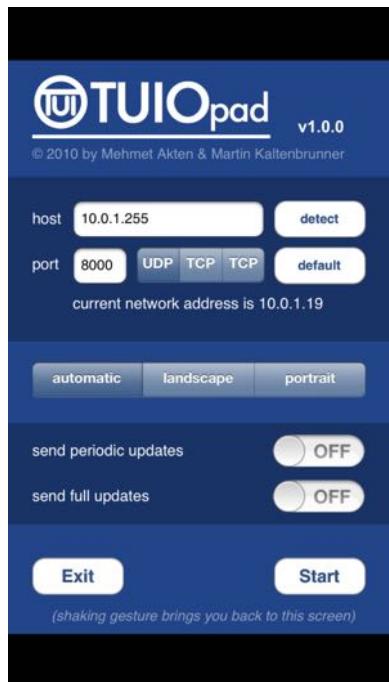
- Supports the four basic OSC data types (32-bit integers, 32-bit floats, strings, and blobs - arbitrary length byte sequences)
- Supports the optional 64-bit timetag data type and Booleans
- Address pattern matching
- Dynamic memory allocation
- Sends and receives OSC packets over transport layers that implements the Arduino Stream Class such as [Serial](#) and [Ethernet UDP](#) and [wifi-UDP](#) for sure.



**let them all  
talk in one  
language: OSC**



for iOS or Android



download **TUIOpad**  
from app store.

download **UniOSC** package  
from unity asset store.



# UniOSC

Scripting/Input - Output  
monoflow

★★★★★ (134)

\$30

Buy now



Requires Unity 4.6.3 or higher.

UniOSC is a tool to easily create applications which can be remote controlled by hardware/software that uses the OSC protocol for communication. It streamlines your workflow to get an OSC enabled system with a few clicks. Monitor your dataflow for better debugging. Create your own tools to remote control the editor.

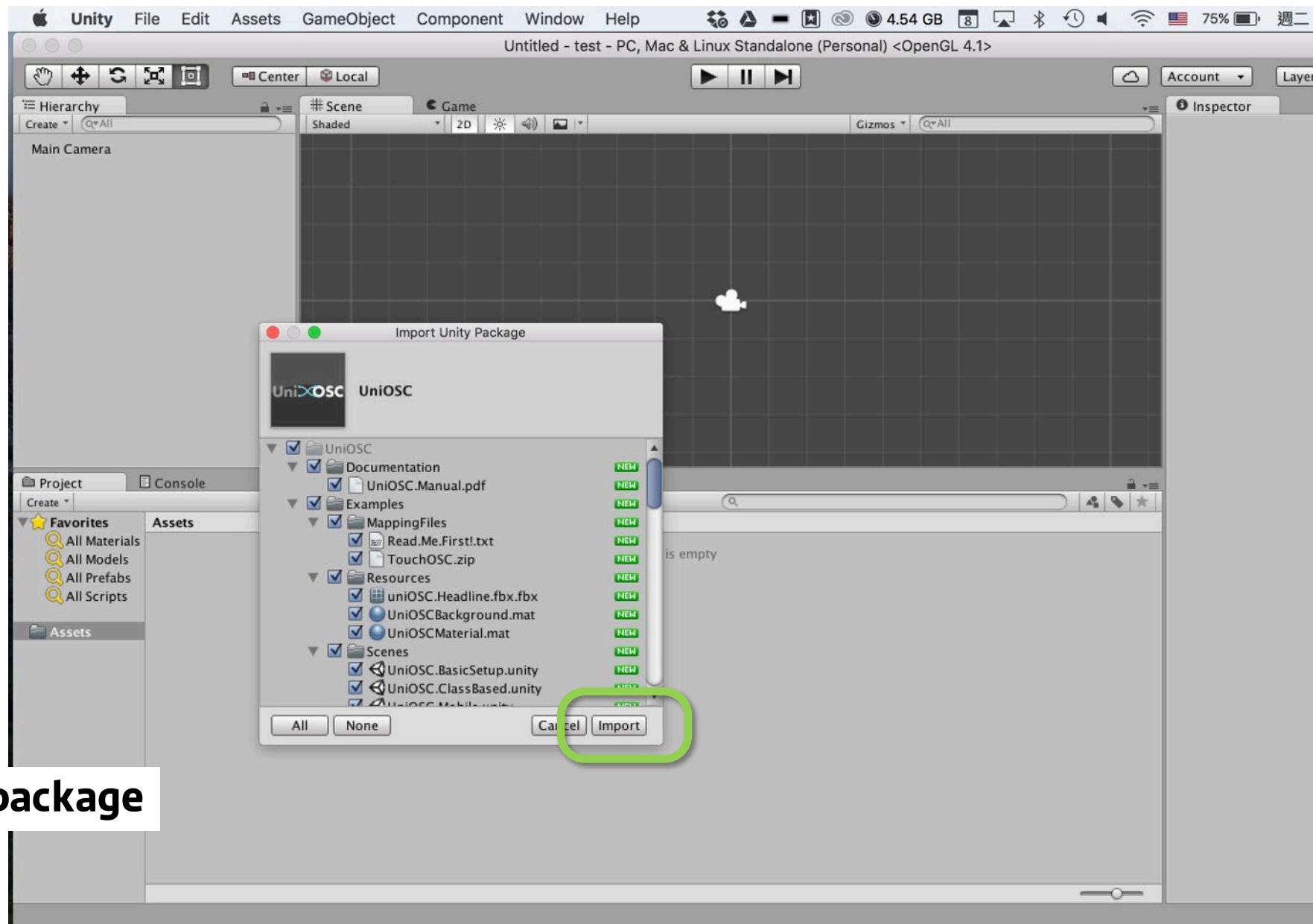
# UniOSC

<https://www.assetstore.unity3d.com/en/#!/content/17658>

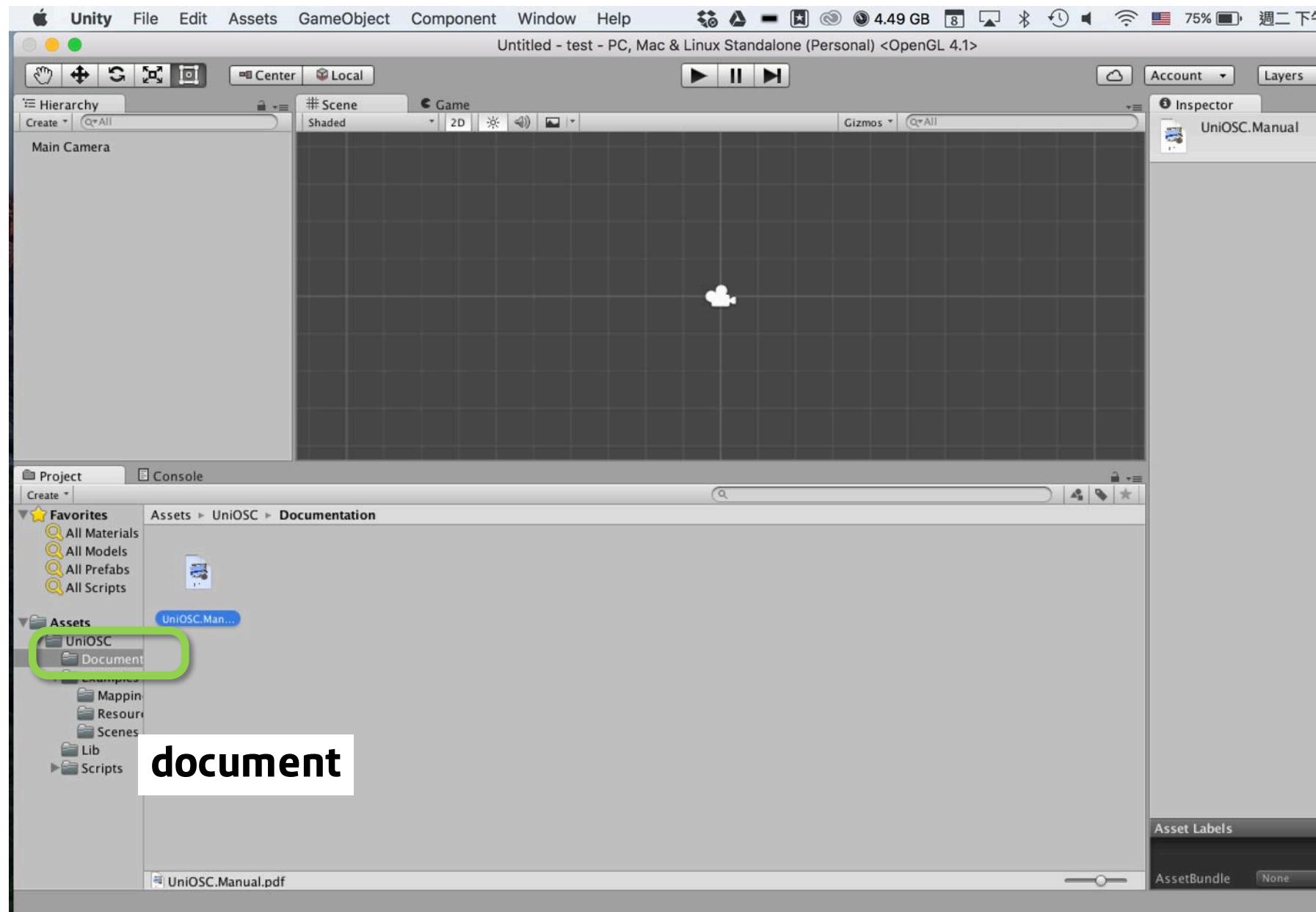
Some utility libraries are really useful to speed up implementation.  
We bought them for this course.

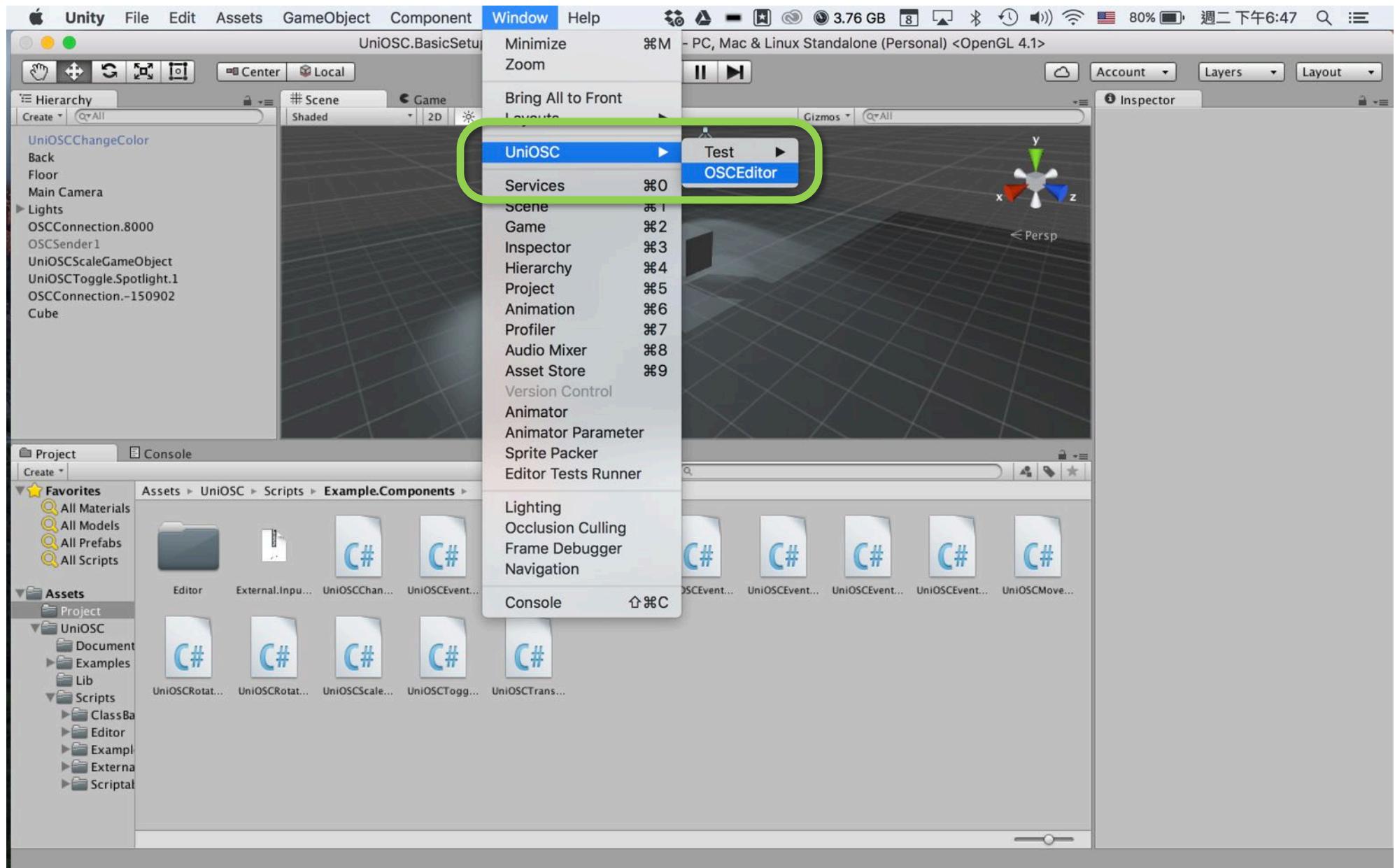
- Access the packages here →
- Use them only for this course.

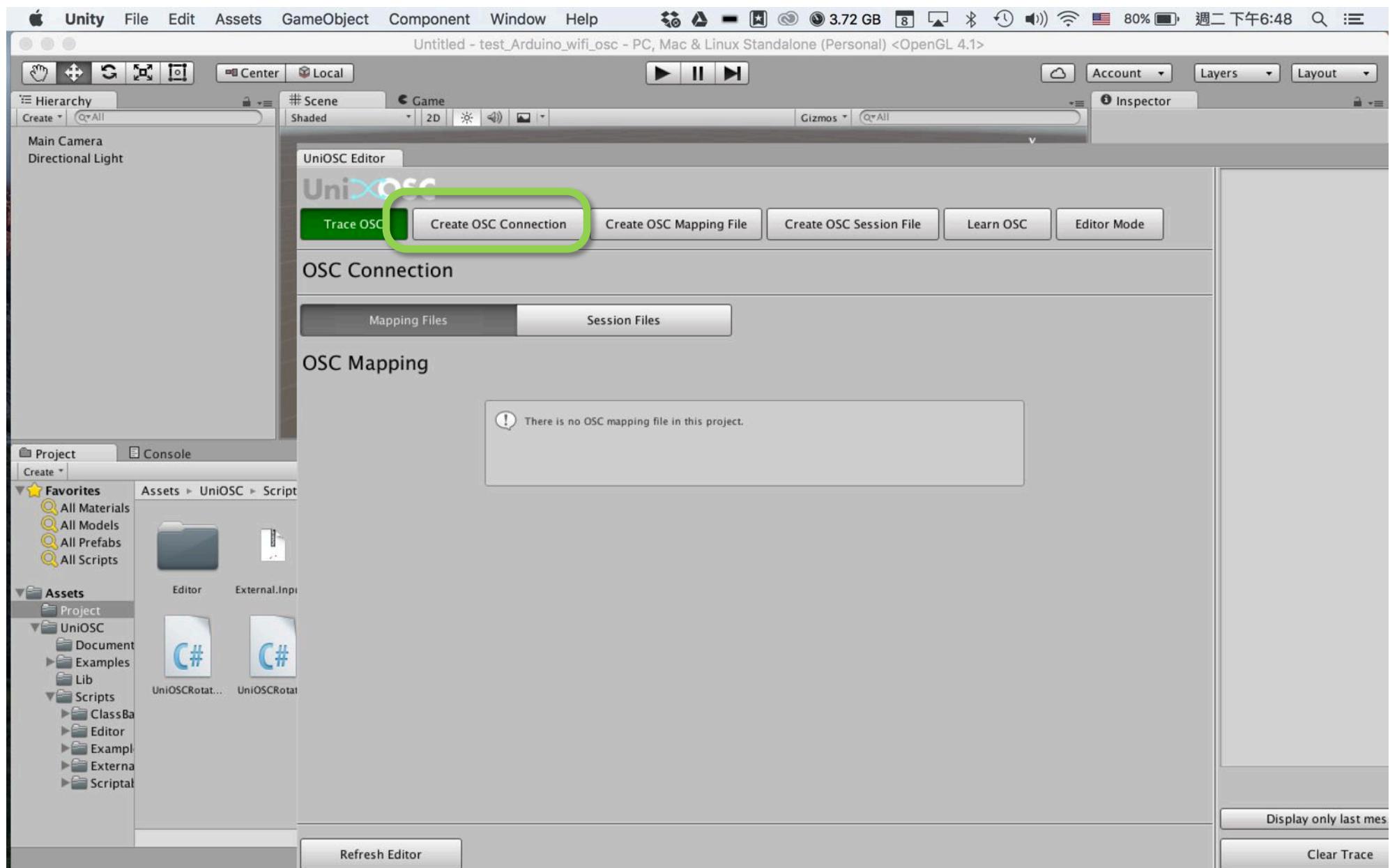
<https://www.dropbox.com/sh/26was0knbenkpda/AAASAyumj1wAlfZ4PUcztpeja?dl=0>

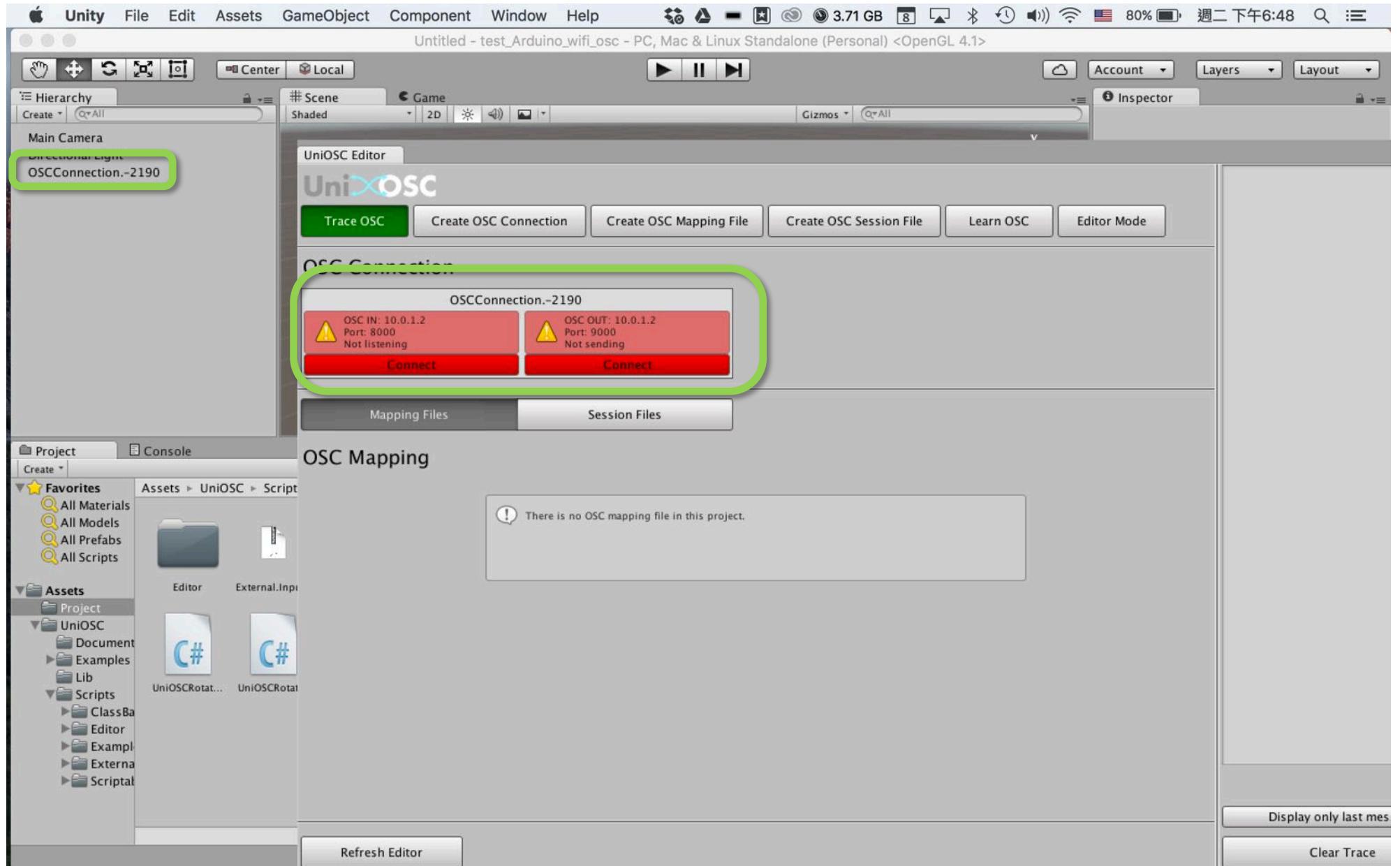


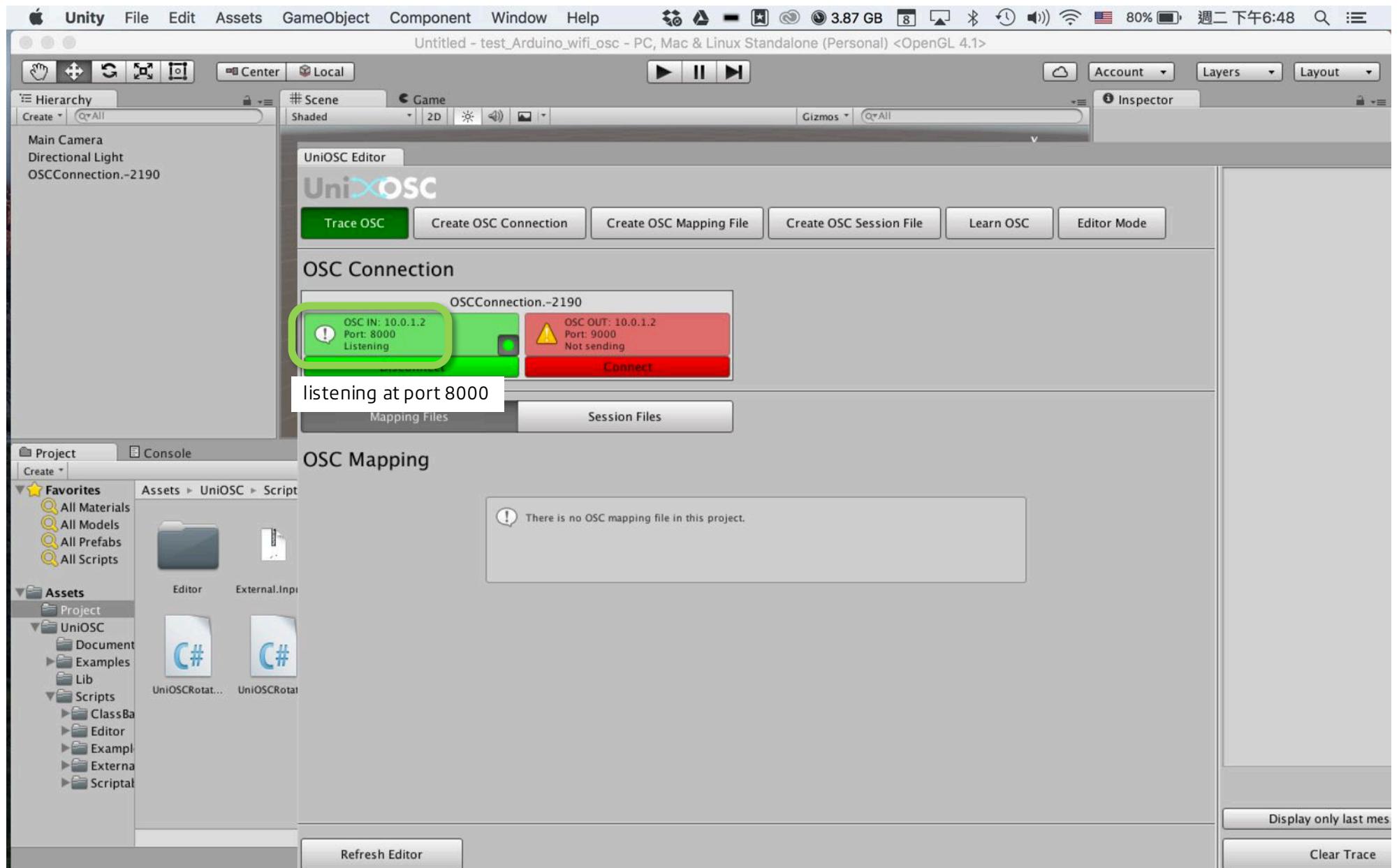
import UniOSC package

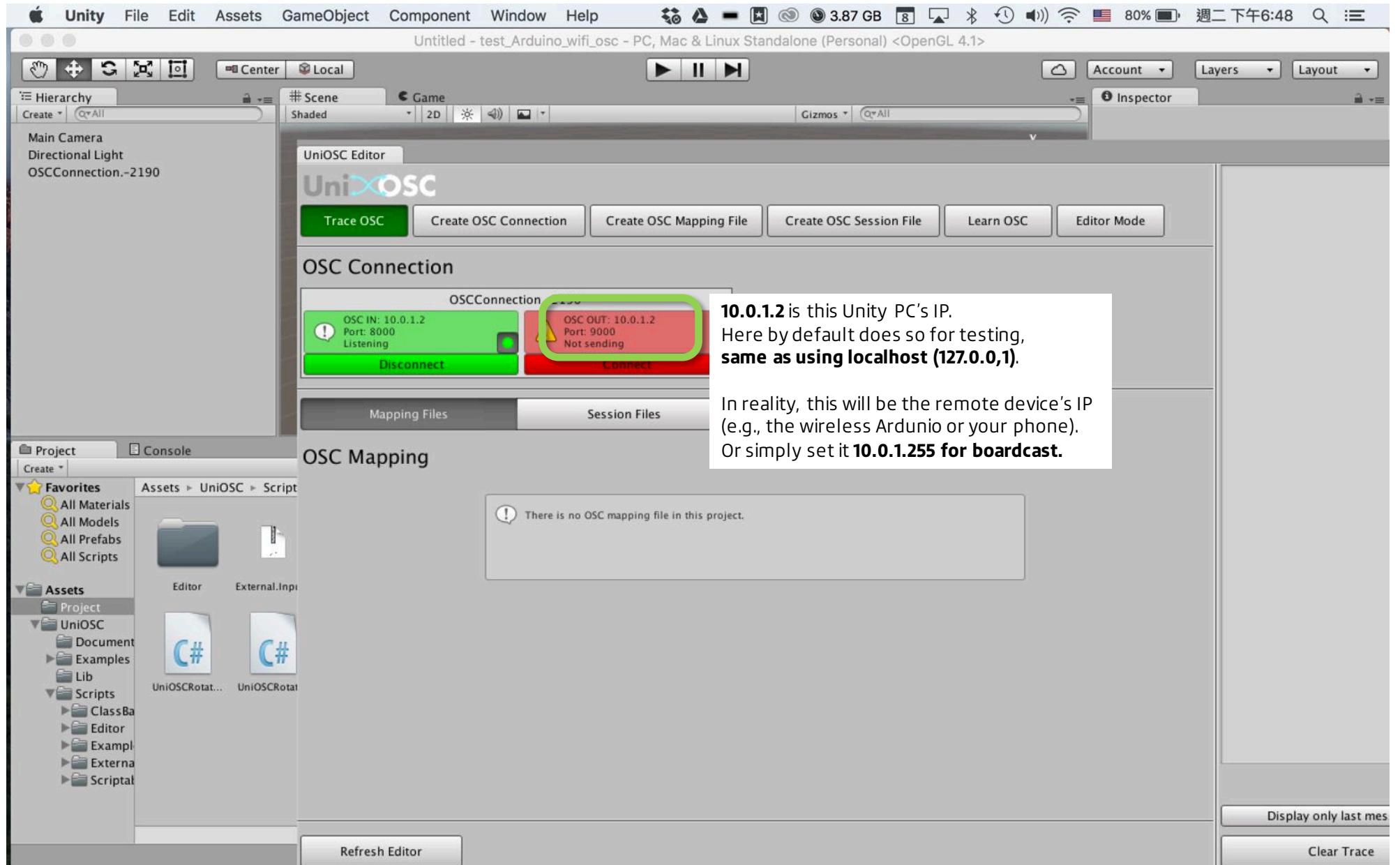


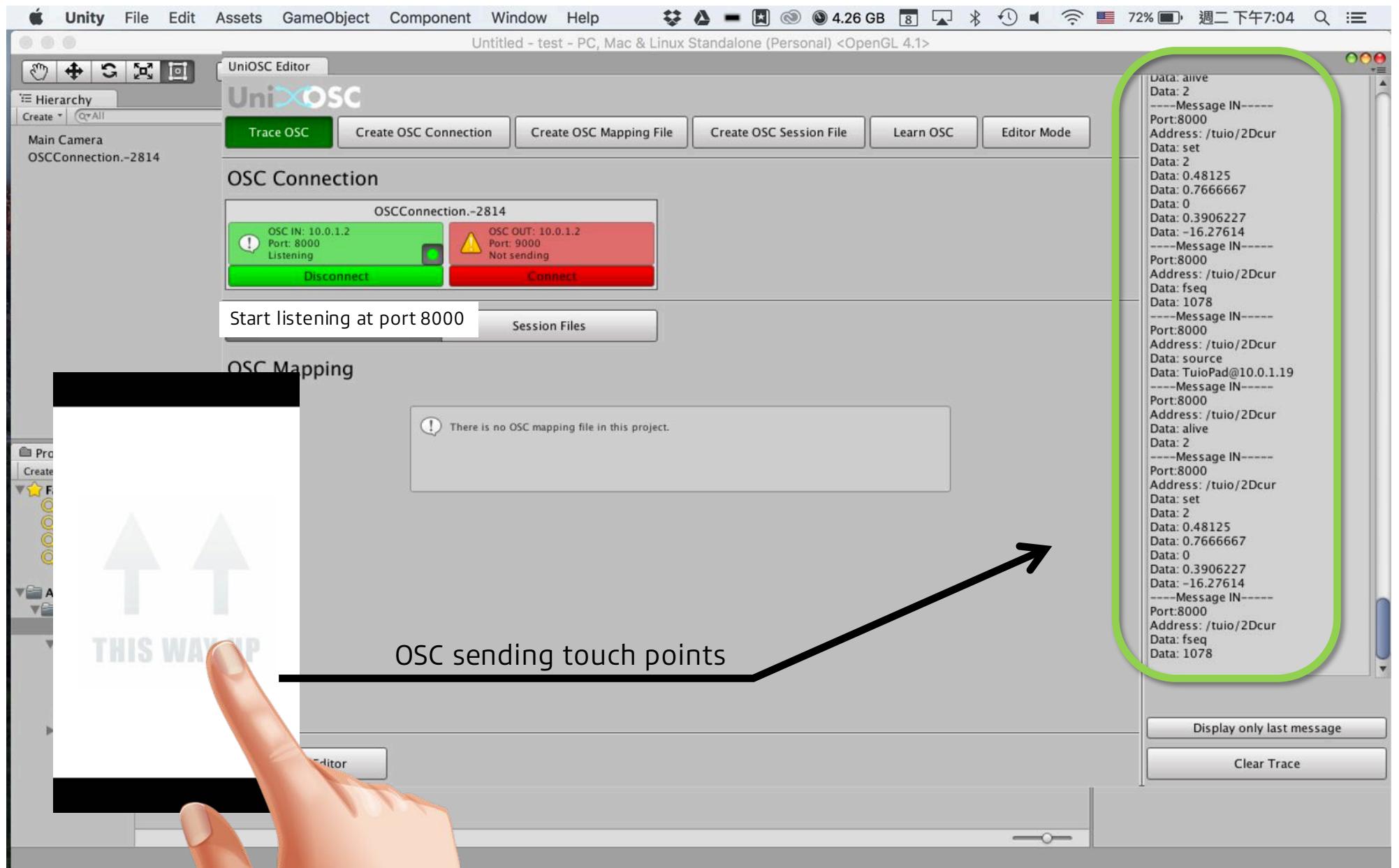


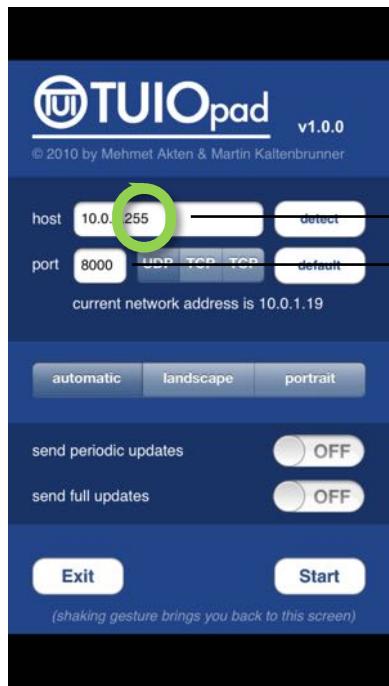












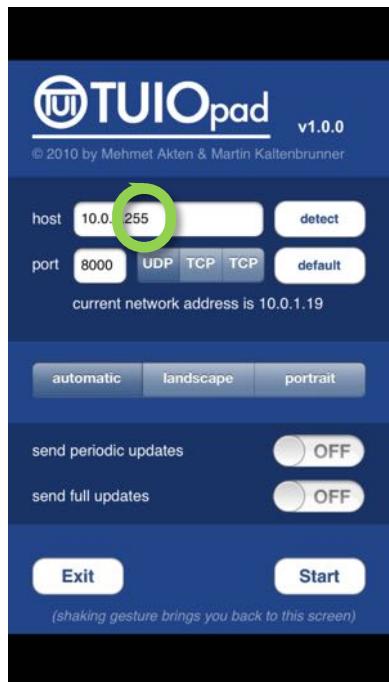
IP: 10.0.1.19

broadcasting

Sending at port 8000

IP: 10.0.1.2  
listening at port **8000**





IP: 10.0.1.19  
**sending** (broadcasting) at  
port **8000**

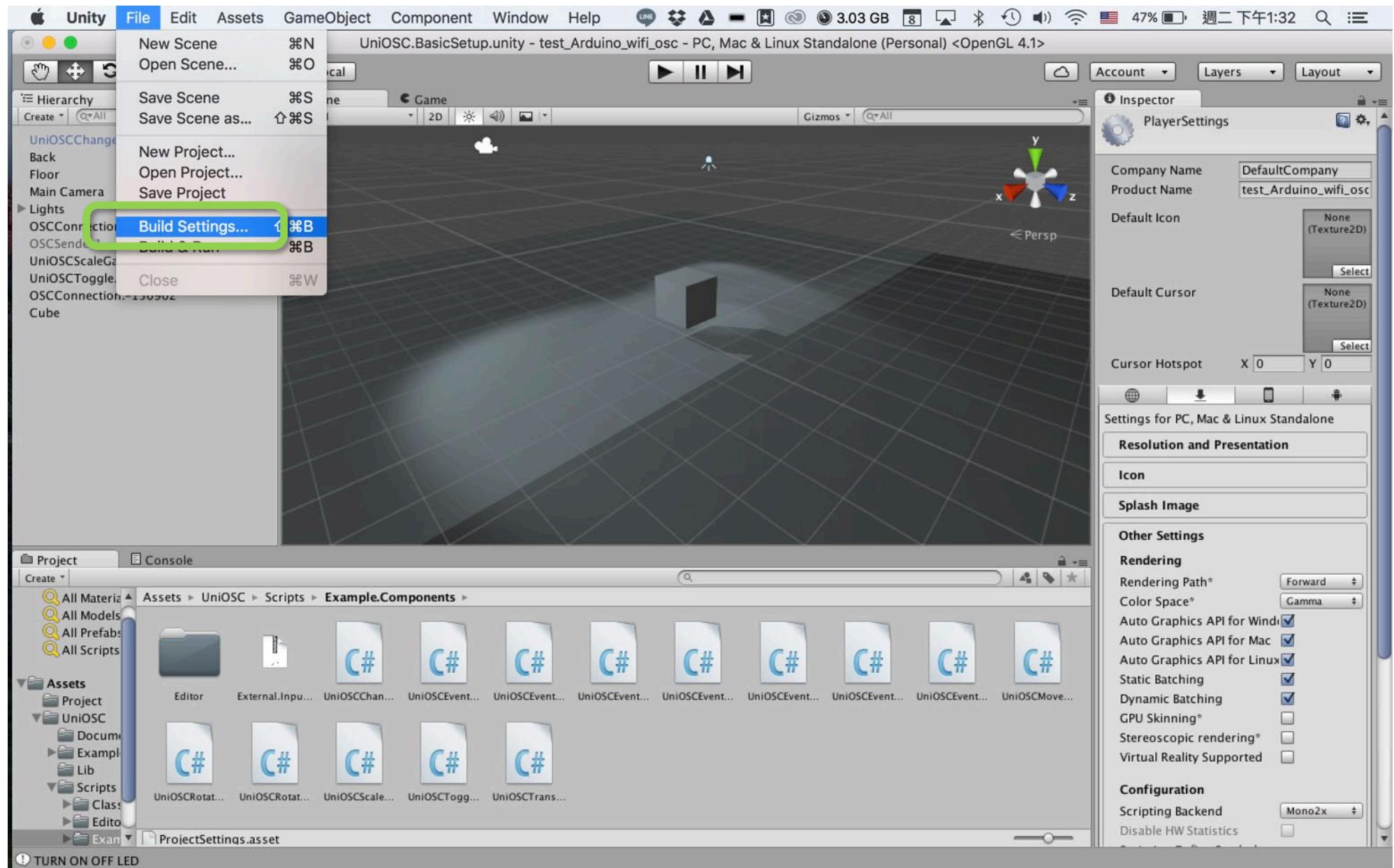
**Sending at 10.0.1.255**  
**means broadcasting the data**

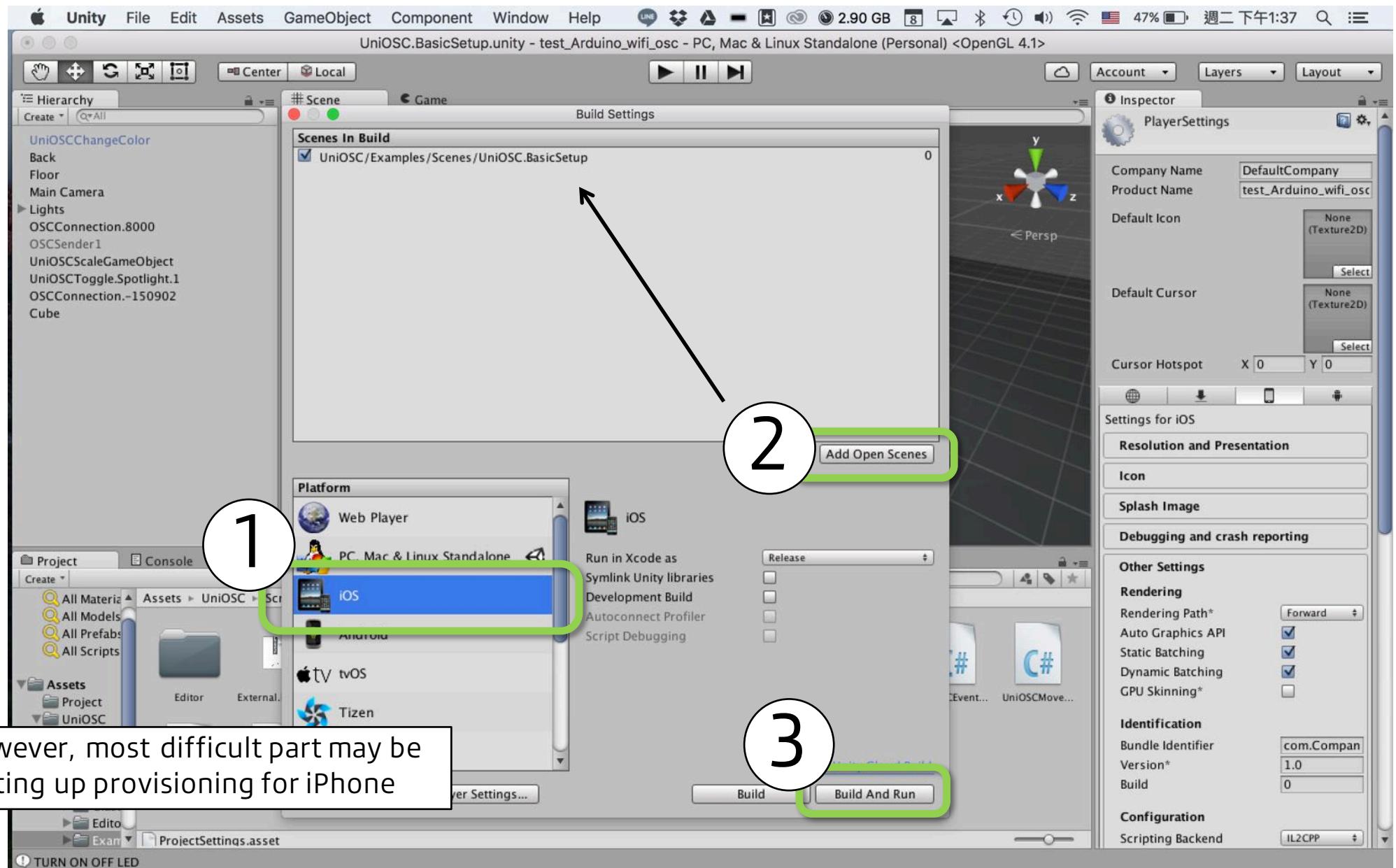
IP: 10.0.1.2  
listening at port **8000**



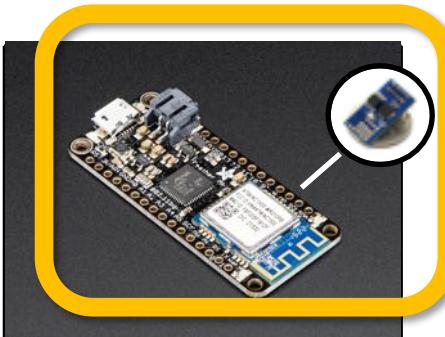
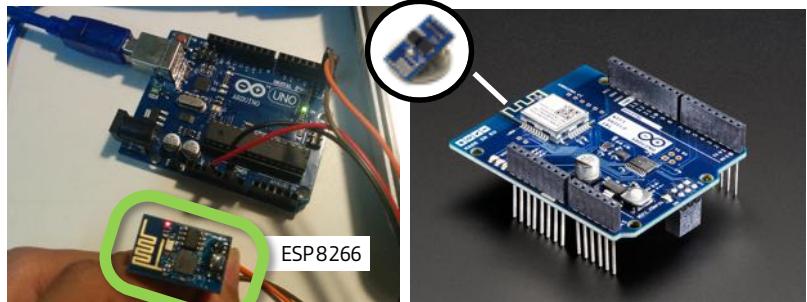
# Implementation using UniOSC

- Check **document**, and read **source code** of same scene
- Implement your own OSC sender/receiver in Unity following **UniOSCEventDispatcherImplementation.cs**
- For Phone version,  
while building you need to ...
  1. config **Build Settings..**
  2. choose **iOS** or **Android**
  3. **Build and Run**





for Arduino



- install **adafruit board**,
- install **WiFi 101 and OSC** libraries

<https://learn.adafruit.com/adafruit-feather-m0-wifi-atwinc1500?view=all>

download **UniOSC** package  
from unity asset store.

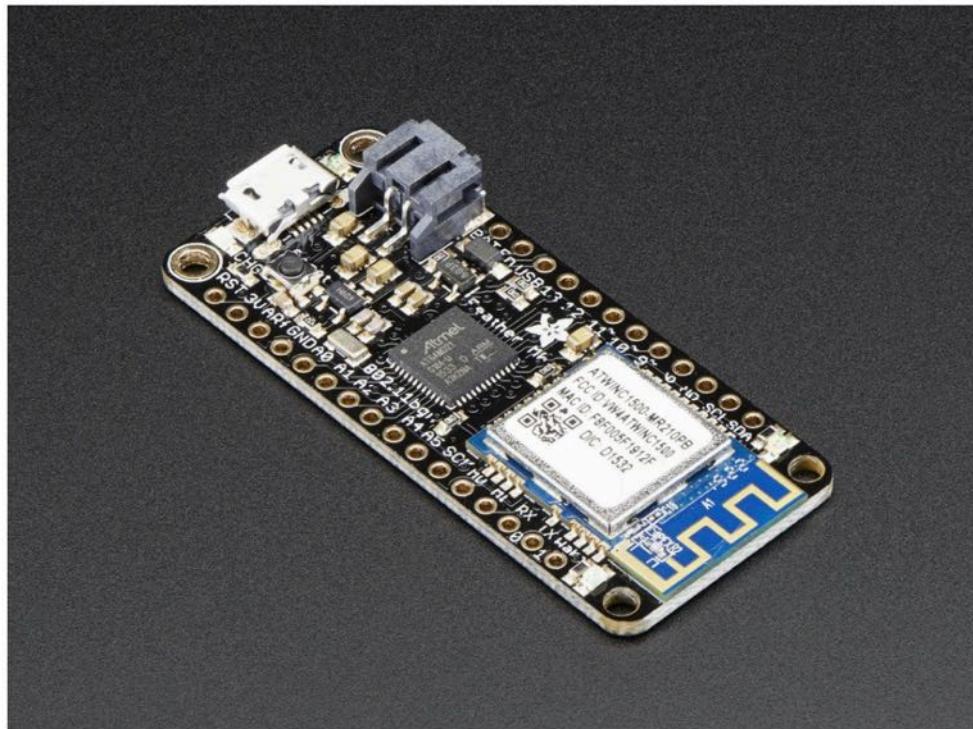


# Overview

by lady ada

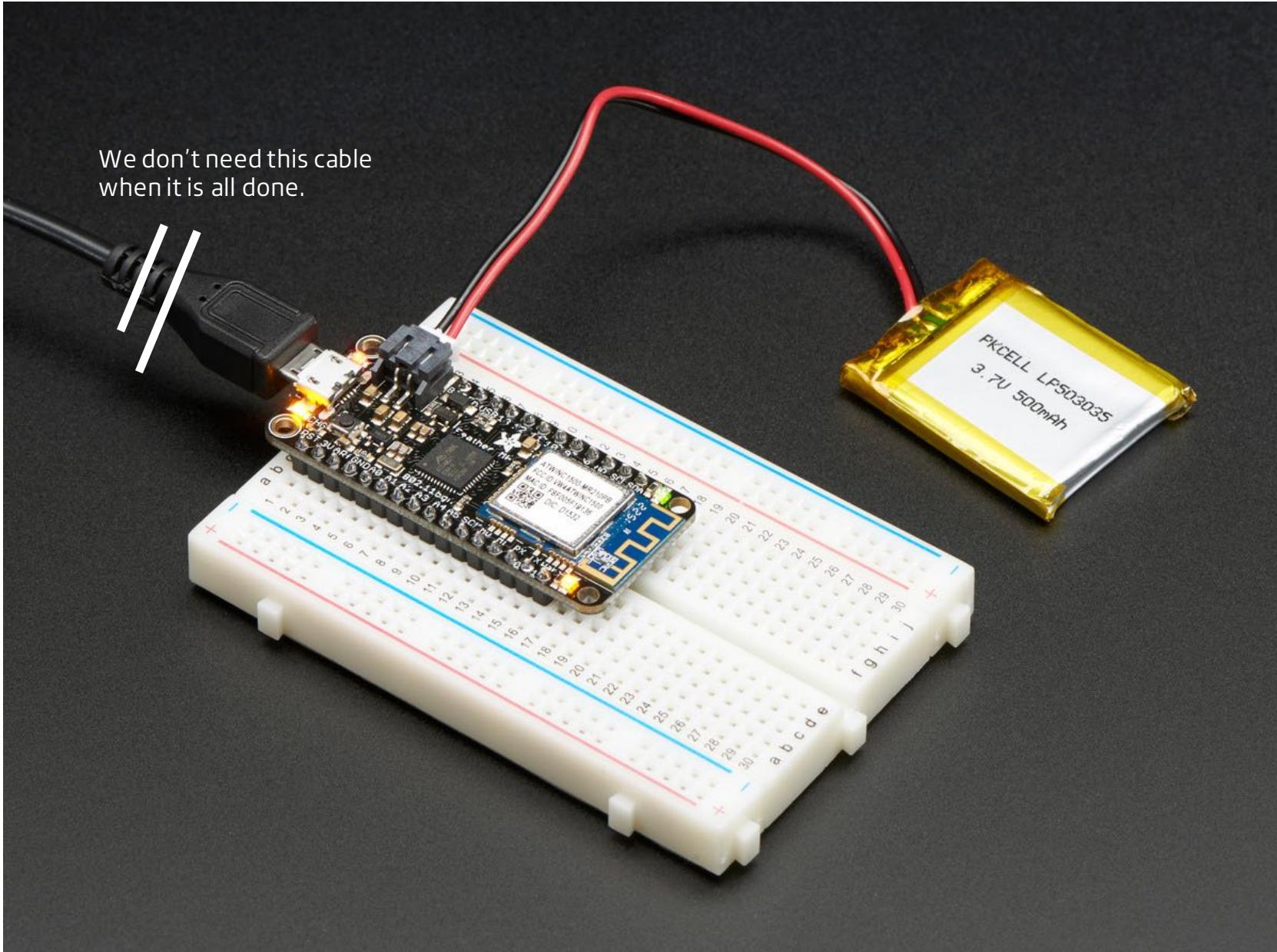
Feather is the new development board from Adafruit, and like its namesake it is thin, light, and lets you fly! We designed Feather to be a new standard for portable microcontroller cores.

This is the **Adafruit Feather M0 WiFi w/ATWINC1500** - our take on an 'all-in-one' Arduino-compatible + high speed, reliable WiFi with built in USB and battery charging. Its an Adafruit Feather M0 [with a WiFi module](#), ready to rock! [We have other boards in the Feather family, check'em out here.](#)



<https://learn.adafruit.com/adafruit-feather-m0-wifi-atwinc1500?view=all>

We don't need this cable  
when it is all done.



# WiFi101

<https://github.com/arduino-libraries/WiFi101>

The screenshot shows the GitHub repository page for WiFi101. At the top, there are statistics: 229 commits, 1 branch, 8 releases, and 9 contributors. Below this is a navigation bar with links for Code, Issues (17), Pull requests (4), Projects (0), Wiki, Pulse, and Graphs. A dropdown menu for the master branch is open. The main area displays a list of recent commits, with the most recent one being a fix for WiFi::ping return value on success by sandeepmistry. Other commits include updates to examples, src, .travis.yml, CHANGELOG, README.adoc, keywords.txt, and library.properties.

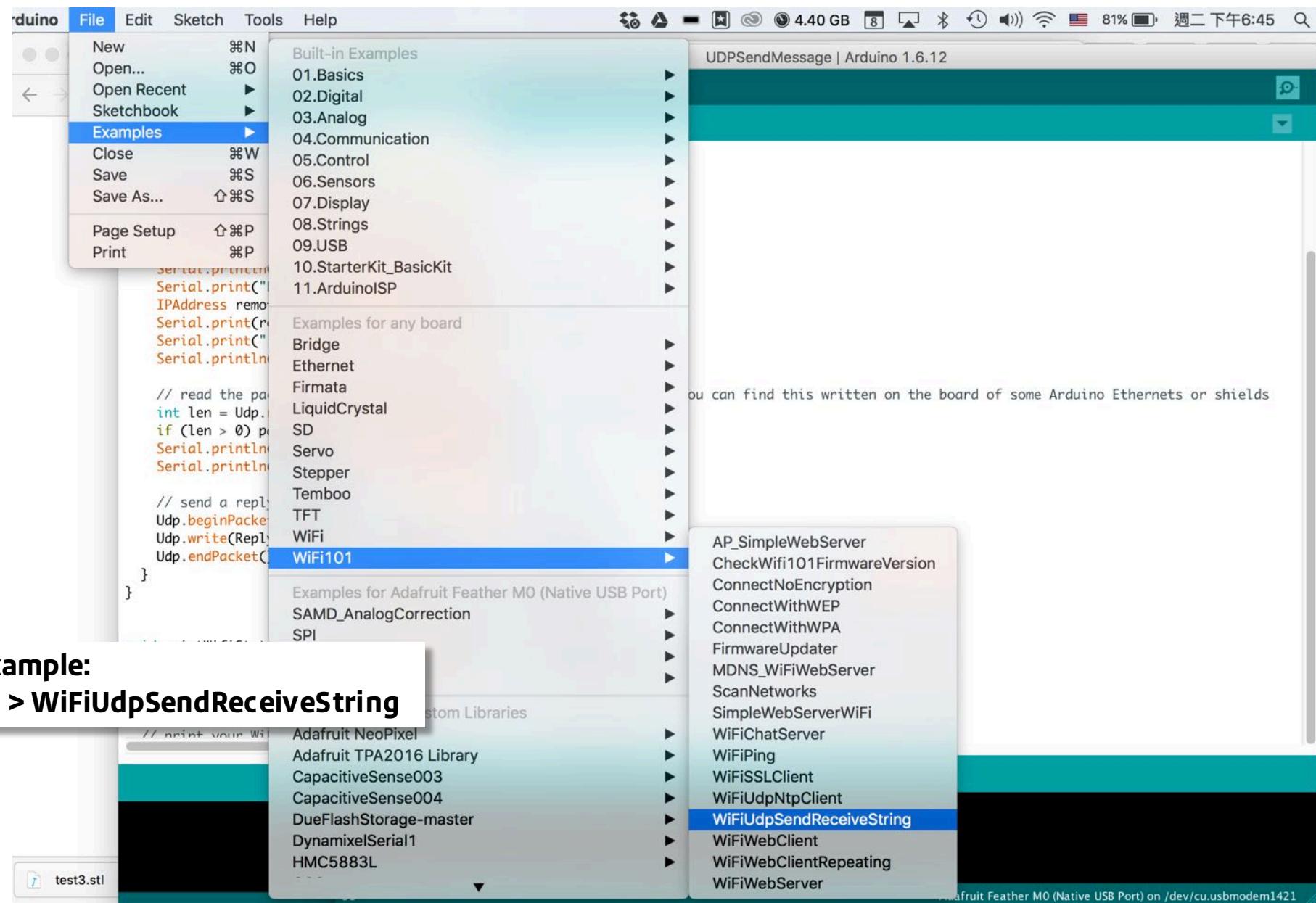
Author	Commit Message	Date
sandeepmistry	Correct WiFi::ping return value on success	Latest commit 274c701 11 days ago
examples	Add comment to example on how to override the local IP address	2 months ago
src	Correct WiFi::ping return value on success	11 days ago
.travis.yml	Use IDE 1.6.12	a month ago
CHANGELOG	Updated changelog	2 months ago
README.adoc	Changed wifi to WiFi in README	2 months ago
keywords.txt	Add getTime to retrieve system time from WINC	2 months ago
library.properties	Release v0.10.0	2 months ago

# OSC for Aruduino

<https://github.com/CNMAT/OSC>

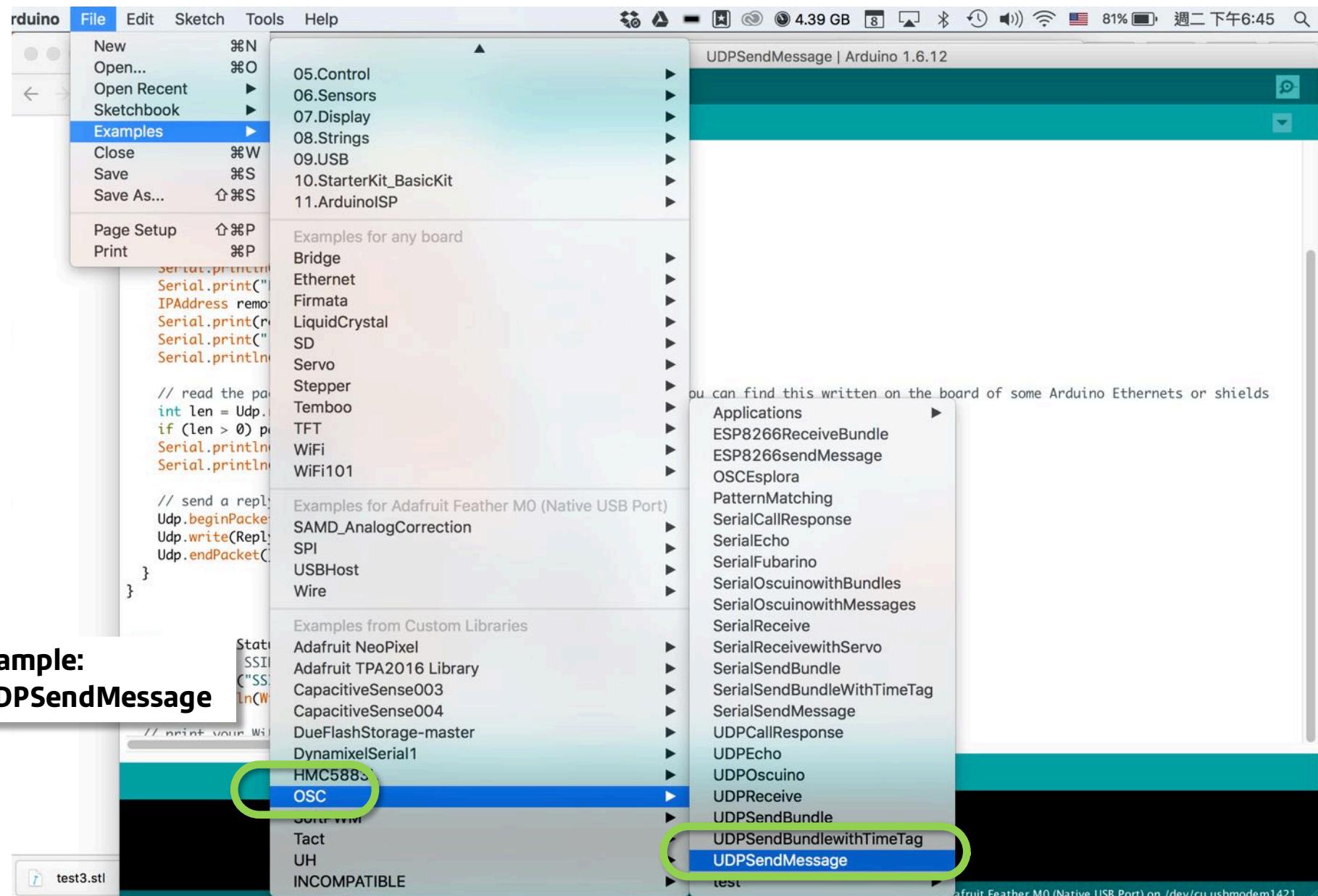
The screenshot shows the GitHub repository page for OSC for Aruduino. At the top, there are statistics: 169 commits, 1 branch, 1 release, and 11 contributors. Below this is a navigation bar with links for Code, Issues (24), Pull requests (3), Projects (0), Wiki, Pulse, and Graphs. A dropdown menu for the master branch is open. The main area displays a list of recent commits, with the most recent one being a fix for teensy fixes by adrianfreed. Other commits include updates to Applications, examples, test, .gitignore, API.md, LICENSE, OSCBoards copy.h, OSCBoards.h, OSCBundle.cpp, and OSCBundle.h.

Author	Commit Message	Date
adrianfreed	Latest teensy fixes	Latest commit 543a1cf on Jul 17
Applications	Latest teensy fixes	4 months ago
examples	Merge branch 'master' of https://github.com/CNMAT/OSC	a year ago
test	Suite of basic validation tests	a year ago
.gitignore	Removed SPI stream stuff	3 years ago
API.md	fix bad function name in API doc	a year ago
LICENSE	created a markdown version of the README	a year ago
OSCBoards copy.h	Latest teensy fixes	4 months ago
OSCBoards.h	Latest teensy fixes	4 months ago
OSCBundle.cpp	bundle methods return `*this` to enable chaining	a year ago
OSCBundle.h	bundle methods return `*this` to enable chaining	a year ago

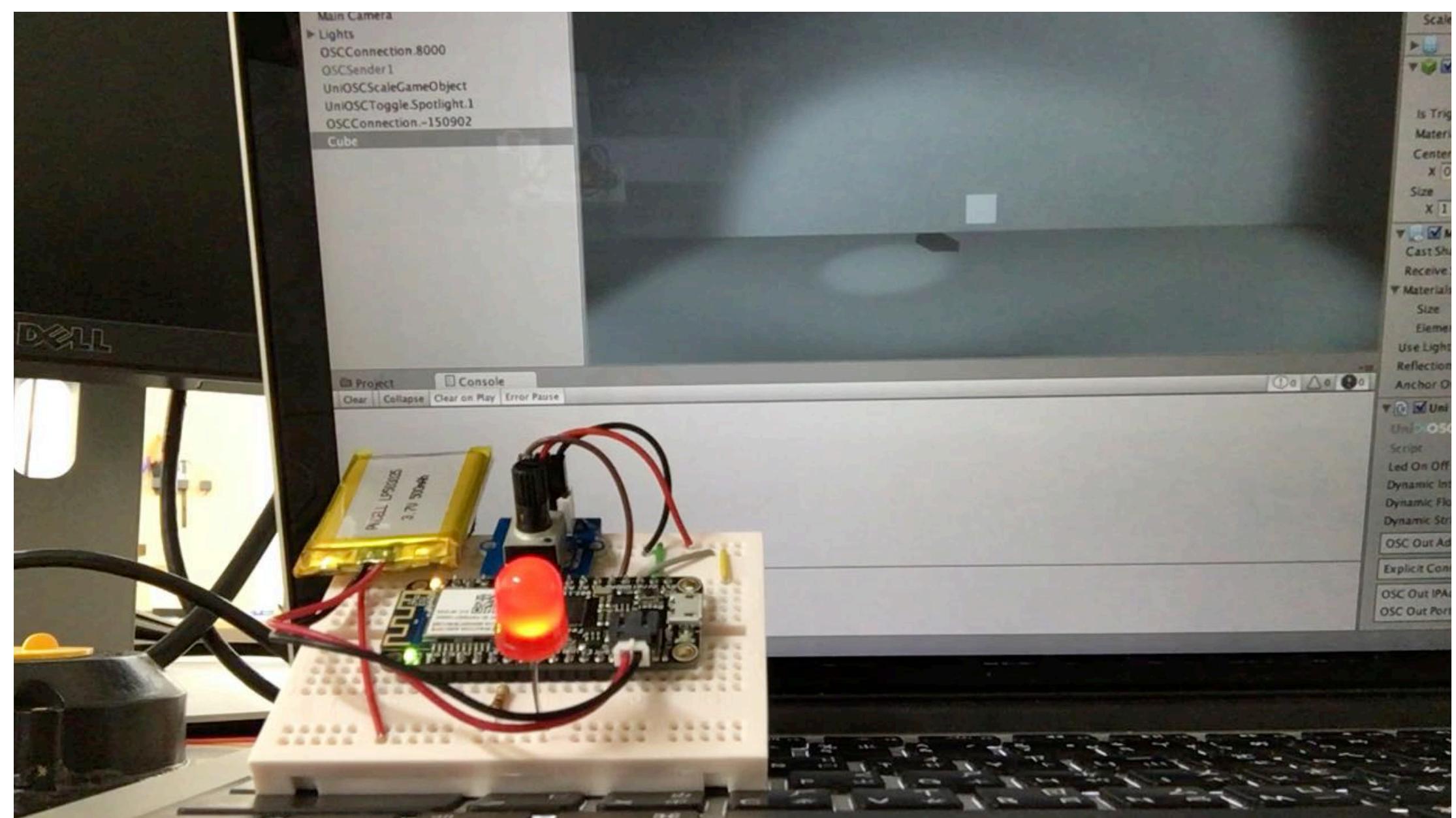


## Code example:

### WiFi101 > WiFiUdpSendReceiveString



## Code example: OSC > UDPSendMessage



**OSC**

**WiFi101**

**Adafrute Wifi Arduino**

**TuioPad (OSC)  
for testing**

**MobilePhone**

**UniOSC  
Unity  
MobilePhone**

**UniOSC  
Unity  
PC**