

## Assignment 5: Events and Interaction

Assignment due 9 March 2009

### 0. Reading

- a. Concurrency in ChuckK

<http://chuck.cs.princeton.edu/doc/language/spork.html>

- b. Events

<http://chuck.cs.princeton.edu/doc/language/event.html>

- d. OSC

Glance at section in Events documentation above

Run OSC examples at <http://chuck.cs.princeton.edu/doc/examples/>

Download OSC handout at

[http://www.cs.princeton.edu/~fiebrink/314/2009/osc\\_handout.pdf](http://www.cs.princeton.edu/~fiebrink/314/2009/osc_handout.pdf)

Optionally read OSC section in ChuckK manual for more info

Optionally check out the code for Clix (both client and server files)

- e. Laptop Interaction & SMELT

Check out example code at <http://smelt.cs.princeton.edu>

If you're interested in joysticks/game controllers, see

[http://wiki.cs.princeton.edu/index.php/PLOrk\\_fall2008/BasicControllerCode](http://wiki.cs.princeton.edu/index.php/PLOrk_fall2008/BasicControllerCode)

**For all parts of this assignment, you have the option of reusing your drum machine code from the last two assignments, or building something new from scratch.**

### 1. Event-based synchronization

Store all code for this question in one subdirectory that lives inside your assignment 5 directory.

Modify your drum machine or create a new mini-piece (self contained, non-trivial, musical code that might not be “polished” or long enough for a full performance) to use event-based synchronization instead of the on-the-fly synchronization that you’ve been using. For example, you might have a “time-keeper” shred that broadcasts an event every beat (or sub-beat). Or you might have some other solution.

Make it interesting and expressive. Include a brief written explanation of why event-based concurrency is an appropriate solution for the code you’ve written.

### 2. OSC communication

Copy your solution for question 1 into a new directory. Instead of broadcasting events, use OSC messages to communicate between (and synchronize) shreds.

For this question, I highly recommend starting by copying from the OSC examples at <http://chuck.cs.princeton.edu/doc/examples/> and paying attention to the OSC handout online. Minor mistakes, like naming your OSC messages slightly differently in the sender and receiver code, or passing slightly different argument lists in the sender and receiver, will result in the code not working.

Comment on the pros and cons of using on-the-fly, event-based, and OSC-based synchronization.

### 3. Real-time interaction

Store the code for this question in a 3<sup>rd</sup> subdirectory of your assignment directory.

Create a mini-piece that you can interactively perform in real-time using the keyboard, mouse, motion sensor, microphone, joystick, and/or other devices. (Ask us if you want to borrow a joystick, game controller, etc.) You can reuse your code from Questions 1-2 or write something new.

Include a brief written explanation of how to play your piece (e.g., if it's a joystick, say which joystick you've used, and briefly describe what each axis/button does).

Make it interesting and expressive! And be prepared to demo.

#### **What to hand in:**

- A zip file containing a directory for each of questions 1, 2, and 3.
- Your written work for questions 1, 2, and 3.