FENCES (for the Hub, 2014)
Matt Ingalls

A “port” of a few passages from acoustic pieces i have written for sfSound - it’s basically a Wolff-inspired, “hot potato” piece. There should be lots of silence and quasi-unison chords so it might be kind of dramatic for the audience - and hopefully enough “manual” playing to make it fun for us. (In this spec, i use “manual” to mean something *you* do during the performance and “automatic” as something your software does during the performance.)

Sounds
You need to produce 2 kinds of sounds:

+ “Posts”
    Short and percussive sounds that should be pitched when responding to ***note*** messages,
      but may be pitched or unpitched when responding to ***response*** and ***stop*** messages.
    Generally loud, but dynamic variations are OK.
    Will need to be triggered manually and automatically.

+ “Slats”
    Long sounds that can be pitched and/or unpitched, used when responding to ***response*** messages.
    They need to be able to sustain indefinitely, until signaled to be turned off.
    Generally soft, but dynamic variations are OK.
    Will always be triggered manually (but will be turned off automatically).
    At the beginning of the piece, the sustained sounds should be relatively static,
      but as the piece progresses you can (should?) start adding more dynamic elements
      (modulation, pitch contour, ornamentation, dramatic swells, etc).

Spatial Placement
Each player should be panned in a shared sound system so that their localization in the sound field corresponds to their position on stage.
(if separate speakers for each player are available, all the better!)

Messaging
For the “1.0” version, you will need to send and receive these events.

+ Send
    note

+ Receive (addressed to you **OR** /hub)
    respond
    note
    stop

/hub(or /hubster) <from_hubster> respond
(for v1.0, <from_hubster> will always be matt)

In your software, make sure there is a way for you to VISUALLY monitor the ***respond*** message (a max “button” object or something?).

Then design your algorithm to automatically react to this message:

    IF you are currently sustaining a “slat” note:
        - Turn off the “slat”.
        - The software can optionally play a simultaneous short “post” note here (essentially “cutting off” the slat)
    ELSE (you aren’t currently sustaining a “slat”):
        - Turn on your button/light/etc to display that this message has arrived.

When you do NOTICE your button/light/signal go on, MANUALLY trigger either a “post” or “slat” to play.
Try to respond as fast as you can! The idea here is to create interesting rhythmic patterns based on reaction times.

/<hubster> <from_hubster> note <fMaxDelayInSeconds> <fMinDelayFac> <fCps> <fTransposeFac>

The response to this message is all automatic. Have your software react:

+ CALCULATE the “minimum delay value”:
    minDelay = fMinDelayFac * fMaxDelayInSeconds

+ PICK a delay value (in seconds) that is in the range of your calculated minimum value and fMaxDelayInSeconds:
    fMyDelay = (minDelay <—> fMaxDelayInSeconds)

+ DELAY for fMyDelay seconds

+ PLAY a short “post” note at frequency fCps

+ CALCULATE - the next person’s pitch value:
    fNewCps = fCps * fTransposeFac

+ if (40 > fNewCps < 10,000), then SEND a new ***note*** message WHERE:

    hubster - is another player
    from_hubster - is you
    fMaxDelayInSeconds - is the “fMyDelay” value you used, above
    fMinDelayFac - is the same ***fMinDelayFac*** originally passed to you
    fCps - is the “fNewCps” value you just calculated, above
    fTransposeFac - is the same ***fTransposeFac*** originally passed to you

/hub matt stop
Automatically respond:
+ turn off all currently playing sounds
+ kill any “delaying” notes (without playing them)
+ simultaneously (and immediately), play a short note

Version 2.0 added features
+ /hubster matt sendmode <iMode>

values for iMode:
0 = no instantiation allowed
1 = you are now allowed to send “note” messages manually
2 = you are now allowed to send “respond” messages manually
3 = you are now allowed to send “note” and “respond” messages manually

+ allow ***note*** messages to be sent to /hub ??