DaisyHead

Ben Taylor

2013

Browser-enabled network music

Starting *DaisyHead*

1) On a computer with Node.JS installed (nodejs.org), unzip contents of daisyhead.zip.
2) Open the file js/manager.js and change line 4 to include your current local IP address.
3) In your command-line console, navigate to the daisyhead folder and execute:

    node daisyserver.js

DaisyHead will now be accessible from your browser or any browser on your local network, at the address: your ip, port 8080. (E.g. 123.4.5.6:8080)

Performing *DaisyHead*

DaisyHead is for 3 - ? performers, all logged into *DaisyHead* through web browsers on individual machines.

DaisyHead makes music by a pulse that travels through the network to the browser of different performers. When it hits each player, it plays a short sample through that player's local amplifier. The central performative parameter of each player is choosing who s/he will send the pulse to after it hits them. **Through coincidence or intention, two, three, and four-person loops of connections often occur in *DaisyHead*, creating isolated patterns that create a momentary musical structure** before dissolving when one player breaks the loop.

Controls

Each player has control over the duration of their note and volume of their note by moving their mouse in a labelled 2D grid.
T starts drone
A and D change your target left/right (who you will send the pulse to after it hits you).
Shift changes the portion of the sample that you will play when the pulse hits you.
Space sends your current buffer position to your target (no sound is made).

Starting the Pulse

One player starts the piece by pressing the "Start Pulse" button.

Under the hood of *DaisyHead*

DaisyHead leverages Node.JS and Socket.IO for zippy websocketing and networking. Web Audio API is the engine for sample playback, synthesis, and effects. Nexus UI provides parts of the user-interface.