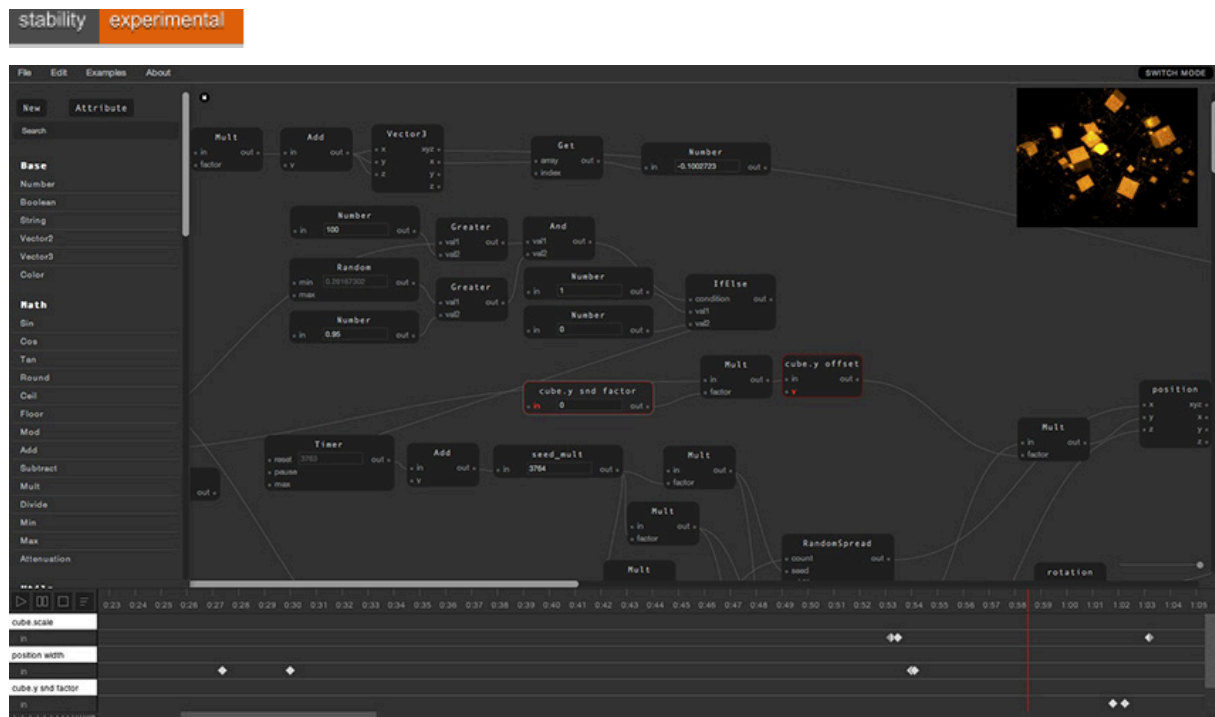


ThreeNodes.js



Experimental

This is an experiment to make something like “vvvv” in javascript, html and webgl.

Live demo: <http://idflood.github.com/ThreeNodes.js/>

Key principles

- modular
- creating a custom “node” must be as easy as possible
- should be possible to switch interface for live performance (selection of buttons/toggle)

Some ideas

- Field array input could have multiple input connections.
- local timelines ?
- automatic layout of node graph
- grouping (multiple nodes in one, possibility to have multiple instances)

-
- ability to take a group, ungroup it, make some changes and regroup it so there are 2 slightly different group definitions (and maybe easier workflow, like ‘make this a new group definition’)
 - some kind of library panel like in flash to manage images/fonts/groups...
 - preloader (+preloader node so the preloader can be customized)
 - maybe add processingjs.org as an alternative to three.js
 - sound nodes (Audiolet for instance)
 - server with possibility to login, save (public/private), browse, fork, +1, ... (a bit like audiotools)
 - when server + grouping are completed, ability to share custom group of nodes (tags, public repository, ...)
 - script nodes (js). Create input/outputs from parsing js vars or simply define a standard way of doing it, 1 return, inputs injected in an array. Be careful when sharing if there is a server with login...
 - glsl nodes (vert/frag)
 - make threenodes.js compatible with other programs (load/export vvvv file (or vvvv.js), meemoo, ...)
 - publish to chrome webstore
 - use most chrome apps features (offline, chromeless renderer, ...)
 - use google drive api (save, load, simultaneous collaboration, ...)
 - more three.js (and other) nodes...
 - native app based on chrome like adobe/brackets with “permissive” settings to easily allow image loading (CORS), ...
 - soundcloud node (or simply a way to put an url in the SoundInput node)
 - flickr image node when they allow CORS, google drive, freesound.org, archive.org, ...
 - possibility to enable/disable plugins so the nodes list is not filled with unnecessary things
 - search in workspace nodes (quickly find a named node, search all ‘time’ inputs for instance, ...)
 - drag and drop image/movies from desktop to threenodes
 - “MIDI” timeline when sound generation is added (tempo, ...)
 - music and sound start/finish/volume/... should be possible to handle in node and in timeline if needed (like after effects)
 - curves in timeline like after effect
 - websocket node with example of server/client in flash/processing/java/scala or other
 - ability to run an instance of a saved workspace from command line (send email alert when some inputs have a certain value for instance)
 - csv, xml, ... node parser (and possibly d3.js renderer, or others)
 - MidiFileInput (and similar) when sound generation is done
 - directly select/move/rotate/scale objects in the renderer like any 3d software (and selecting an object3d (or subclass) node should display a x/y/z axis in renderer window to allow move/rotate/scale)
-

-
- gui button/knob/slider/textfield/bang/toggle/select/... nodes
 - key shortcuts + possibility to assign custom key bindings on toggle buttons, ... ?
 - more flexible webgl renderer (image in node or fixed in corner)
 - procedural texture (layers with multiply/add/color..., canvas image, effects, ...)
 - more UI fun inspired by what other programs do (but stay in 2D and keep performance a priority)
 - make it easy to add a theme (+ add a white version, and make the dark one nicer)
 - record values while playing like it is done in music applications (automation)
 - when sound synthesis is done, add drum machine, sequencer, reverb, ... nodes
 - ability to change layout mode (vertical/horizontal/smaller nodes expanded on hover/...)
 - menu to automatically arrange nodes (spring based solution or other)
 - nice user & dev documentation
 - export sequence of png (or video if possible) like a render in ae (time start/end, fps, resolution, ...)
 - better “flow” highlight (when a node is selected, make the inputs and outputs nodes more visible (maybe optional))
 - display some attribute directly in the node (sliders/textfields/..., add a checkbox in the attribute editor to enable this)
 - display scaled down subchilds of nodes inside a group node. Seeing the global workspace it should be possible to have an insight of the complexity of different group nodes (maybe optional)
 - different color of bullet for each field types (or more general classifications like number, string, vectors and others)
 - when dragging from one field to another, highlight only real possible connections (not all inputs like it is now, but based on field type)
 - ...

Development setup

This will automatically compile coffeescript files to javascript, sass to css and haml to html.

1. install node.js 0.8.x or later (<http://nodejs.org/>)
2. install compass (<http://compass-style.org/install/>)
3. install grunt (<http://gruntjs.com/getting-started#installing-the-cli>)
4. cd in ThreeNodes
5. npm install -d

Build / Deploy

1. cd in ThreeNodes
2. grunt build

Known limitations

- It is recommended to always access ThreeNodes.js from http since chrome has a highly restrictive file access when using file://. (see <http://code.google.com/p/chromium/issues/detail?id=40787>)

Howto create a simple scene

First, create a webglrenderer node. Then add a scene and connect it to the “scene” of the webgl node. On the scene children you would connect a “merge” and to it a mesh. With that there should be a red cube showing in the renderer window.