

## void setup(Links)

- <https://constellationsss.hotglue.me/?livecoding>
- <https://www.visor.live/> (based on processing)
- --> <https://tidalcycles.org/Welcome/fr> ça a l'air assez complexe à installer... je vais demander des conseils à Benjamin
- Installing the following packages:
- tidalcycles
- By installing you accept licenses for the packages.
- Progress: Downloading git 2.31.1... 100%
- Progress: Downloading git.install 2.31.1... 2%
- <https://supercollider.github.io/> - je suis en train de faire n'importe quoi avec supercollider ^^

void share(11h10!)

<https://foxdot.org/> This site is the home of the Live Coding environment, **FoxDot** - a Python-based language and editor for making music. (besoin de supercollider ou autre pour la sortie de son)

Collaborative livecoding : <https://toplap.org/troop-a-collaborative-editor-for-live-coding/>

:::

Ce qu'on pensait stable et fixe est en fait mou et malléable. Série de curseurs invisibles qui montrent les structures de la machine au sens "the machiine" au sens the matrixx. Les bases de la peinture en fait :

**couleurs - opacité - superposition - taille** - ah non sculpture en fait - **déformation - marge - décalage** - c'est de l'édition - j'ai compris, oui oui, c'est de la musique donc - **rythme - rythme d'apparition - variation - pattern - mixage** - cinéma et performance - **espace-temps**. On parle de "bibliothèques de fonctions".

En fait, les petits bugs dans la machine c'étaient nous, plein de bugs ça fait un monde.

On fait déjà du livecoding :)

DISPLAY :

- écran visible, vidéoprojection (partage d'écran ?)
- est-ce que ça induit forcément un aspect performatif ? Est-ce que c'est la perf de la "machine", ou bien il faut voir quelqu'un agir sur le code ?

MUSIC :

Pas nécessaire de comprendre à fond le code et le logiciel, tant que la musique "sonne bien" - ça peut valoir pour l'image aussi, non ? Alice parlait du glitch.

"JUST GO WITH IT"!!!!

[<https://www.legacyrussell.com/GLITCHFEMINISM>] (nan mais le CV hahahaha énorme la compet NYC)

[https://fr.wikipedia.org/wiki/Athanasius\\_Kircher#Acoustique,\\_musique\\_et\\_harmonie](https://fr.wikipedia.org/wiki/Athanasius_Kircher#Acoustique,_musique_et_harmonie) //

<https://larkfall.wordpress.com/2014/06/06/kircher-schotts-computer-music-of-the-baroque/> en lisant ça je me dis que voir la machine (ou le programme) en live c'est hyper important - y'a un truc opaque dans la compréhension du programme, et en même temps pas de secrets car on voit la "machine" et l'interaction à l'oeuvre

Poésie pédagogique ?

"To me this is using computers on a deep level for what they are – language machines." <3 <3 <3

on peut directement intervenir dans le code à gauche de <https://netnet.studio/?gh=nickmontfort/the-bug/main> et voir entendre ce que ça produit

mais j'ai l'impression qu'il faut un compte github pour vraiment garder ton travail de mix - c'est faisable, voir facile

à quoi serve l'autorisation du micro et de la webcam ? ça influence le code ? sur différents site je donne des autorisations huuu...? (c'est dans hydra, ne pas dans the bug - on peut travailler l'image webcam dans hydra)

*"the existence of this GitHub repository should not indicate to anyone  
that we are maintaining this AT ALL! feel free to fork, of course..."*

HYDRA - C'est un synthé en fait !

<https://hydra.ojack.xyz/> (il y a tout une communauté avec des meetings régulier)

en haut à droite, on a un bouton random sketch pour visualiser les propositions, pas mal de femmes! <3

Super HYDRA ! oui c'est top

J'aime beaucoup comment certaines personnes rangent leur code - d'autres non

HYDRA utilise des oscillateurs - comme le son  $\text{osc}(x.x.x)$  ça tourne en python? c'est du java / un genre de java simplifié / ou alors python basé sur du java ? je connais pas du tout comment le python fonctionne mais  
Features:

- Written in javascript and compatible with other javascript libraries

*The API is inspired by analog modular synthesis, in which multiple visual sources (oscillators, cameras, application windows, other connected windows) can be transformed, modulated, and composited via combining sequences of functions.*

**Source** - noise - voronoi - osc - shape - gradient - src- solid- - - -

**Geometry** - rotate - scale - pixelate - repeat - repeatX - repeatY - kaleid - scrollX - scrollY

**Color** - posterize - shift - invert - contrast - brightness - luma - thresh - color - saturate - hue - colorama -

**Blend** - add - layer - blend - mult - diff - mask

**Modulate** - modulateRepeat modulateRepeatX - modulateRepeatY - modulateKaleid - modulateScrollX - modulateScrollY - modulate - modulateScale - modulatePixelate - modulateRotate - modulateHue -

Collaborative livecoding : <https://toplap.org/troop-a-collaborative-editor-for-live-coding/>

<https://editor.p5js.org/KatieChen/sketches/H1Baj-ZYf>

je comprends pas comment faire? (error)

Ah il faut effacer le code dedans qui était déjà là qui a été fait par quelqu'un et visiblement qui ne fonctionne pas (je pense que c'est fait exprès, cadeau mais pas complet :-° )

Ou transformer la dernière ligne (28) qui est incomplète en :

endShape(CL);

- au lieu de
- endShape(CL (vous vous y connaissez :)
- je fais du html/css/ un peu de PHP, java et C#! Mais pas possible d'improviser - j'ai besoin d'une bibliothèque, de regarder des fonctions, etc (d'ailleurs en fait c'est endShape(); tout court ça marche oui
- mon désir initial résiste : voir si on peut imiter <https://www.youtube.com/watch?v=fN9qLt0IPC0> oui c'est vraiment hyper simple et beau ! enfin, je veux dire simple dans le dispositif, parce que son code... omg oui je pense qu'il y a beaucoup de prep elle a déjà fait en amont ses fonctions  
elements\_of\_chance

- elle maitrise ses outils graphiques (c'est son métier j'imagine, réflexe de webmaitresse inspecteur web)  
ah wiwi trop magnifique