
Damien Baïs

10/06/1984 à Sallanches (dept. 74), France

—

damien.bais@gmail.com

+33 6 27 25 11 45

—

—

Pratique professionnelle / artistique

What's Wind Drawing?

Installation électronique
2010

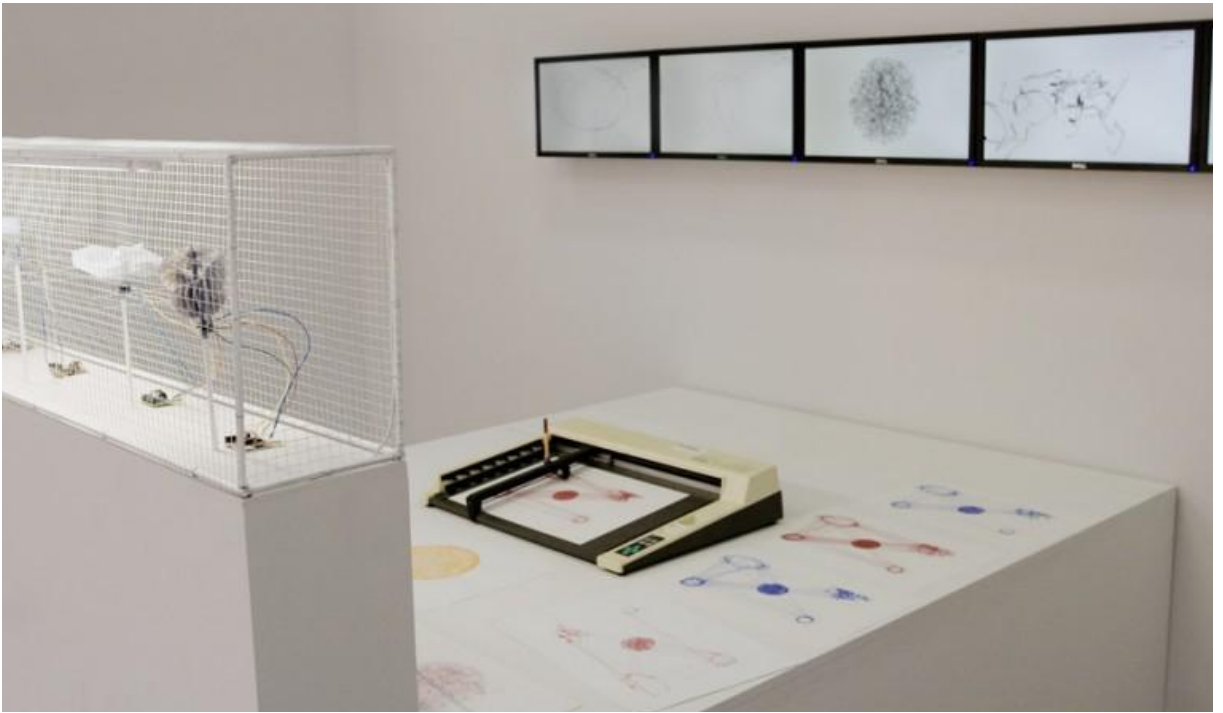
Sur un pupitre, plusieurs hélices nous invitent à souffler en direction d'une table traçante. Seul ou à plusieurs, le dispositif transcrit et récompense simultanément nos efforts en créant d'étranges dessins. Sur les murs, des écrans analysent en temps réel l'activité de cette chaîne de création. Des algorithmes ouvrent et ferment des portes, chaque paquet de données circule d'un écran à l'autre aussi fluide qu'un courant d'air. La machine semble observer et maîtriser les acteurs de ce jeu innocent. What's Wind Drawing? est la première activation d'un projet de création autour du détournement d'algorithmes d'intelligence artificielle.

Expositions :

- Galerie Ars Longa – Paris, Exposition Nouveaux Paramètres, novembre 2011
- Biennale design de Saint-Etienne – Exposition N-1, novembre 2010



← Photo de
l'exposition à la
biennale Design
2010



↑ Détail du dispositif complet



↓ Photo de presse



999999 seconds

Installation électronique in-situ
2012

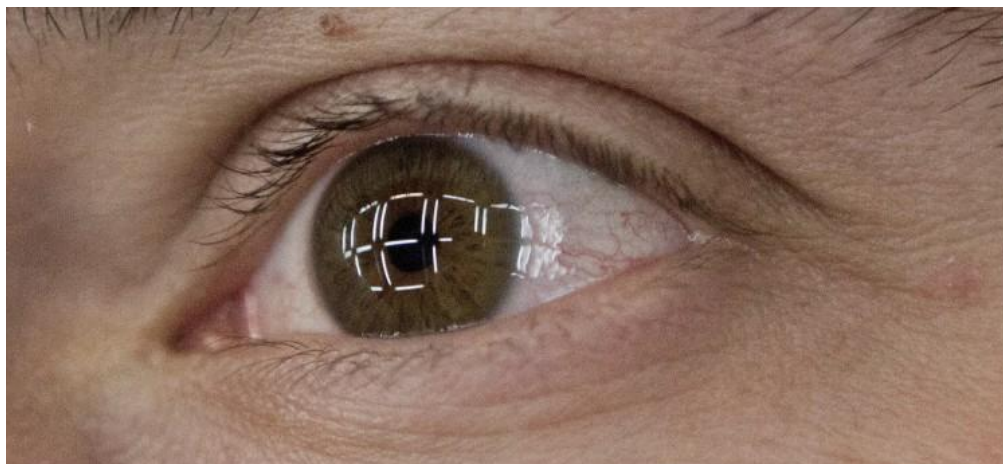
Cet automne 2012, une question se fait de plus en plus évidente, au point que la passer sous silence est devenue assourdissant. Tout autour du monde des images nous arrivent, nous inquiètent et s'archivent dans notre mémoire en pesant, petit à petit, sur nos consciences. Nos voisins nous regardent du coin de l'oeil, avec l'intention, toujours avortée, de briser le tabou. Les médias auscultent, jusque dans les moindres détails, les effets mais ne remontent à aucun moment vers la cause du péril. L'Homme tourne dans tous les sens tel un animal sans tête. Ici, nos vieux se laissent partir, presque soulagés de ne pas avoir à l'affronter. Là, nos enfants vont à l'école, cartable sur le dos, chargés d'un poids, comme un léger malaise, pas encore identifiable pour la plupart. Les plus intelligents d'entre eux, néanmoins, le perçoivent distinctement, peut-être plus que leurs parents, car ce sentiment est débarrassé d'illusion. Cet automne 2012, il ne reste plus beaucoup de temps, quelques milliers, quelques centaines de minutes, pour se poser la bonne question. Chacun en est capable mais tout le monde n'y parviendra pas à temps. Il semblait donc urgent pour .CORP de construire un outil universel pour évaluer le temps qu'il reste. Grâce aux moyens mis à disposition par l'Assaut de la Menuiserie, la prise de conscience a un horizon. Et quand le compte-à-rebours atteindra zéro, le 21 décembre 2012, on ne pourra pas dire qu'on ne l'a pas vu venir.

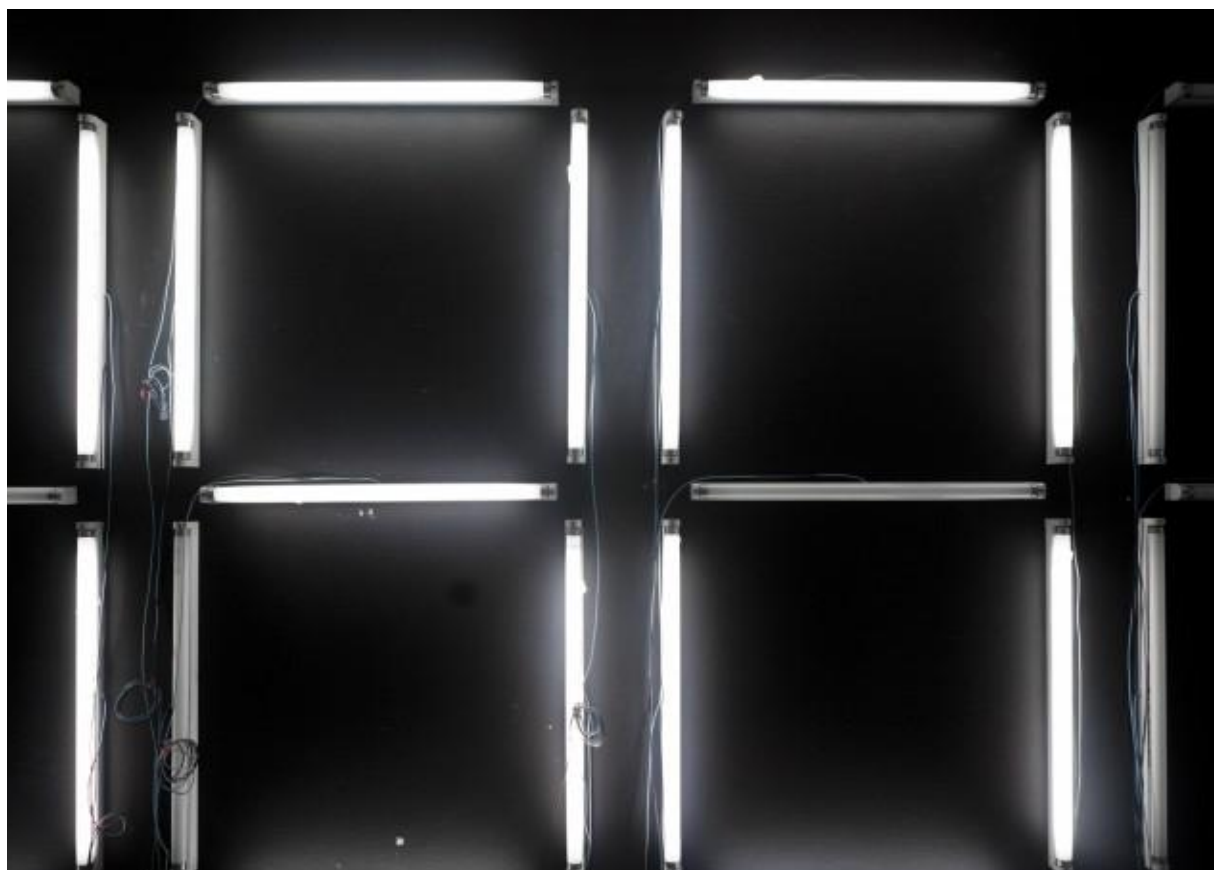
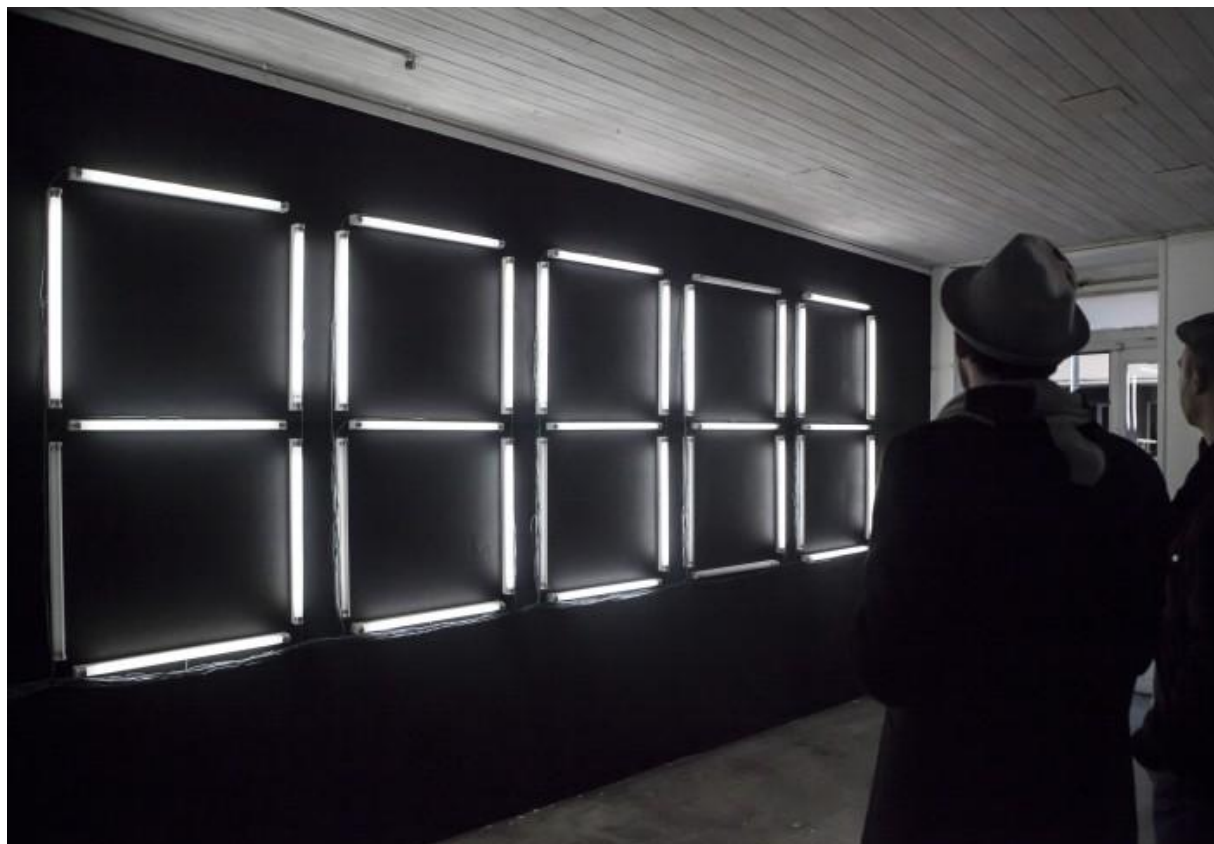
→ Exposition de l'installation à "l'assaut de la menuiserie"

De septembre à octobre 2012, l'Assaut de la Menuiserie, lieu d'art contemporain à Saint-Étienne a mis à disposition son espace pour exposer quatre pièces spécialement conçues par .CORP pour l'occasion. L'exposition Si Dieu existe, il doit bien y avoir une interface de contrôle présente ainsi TradeWar!, Chro[no]matologie, Windows lick et 99999.

Avec la participation de Pierrick Faure et Martin Guillaumie

↓ Photo du dossier de presse





Window Licker

—
Installation électronique sur vitrine
2012
—

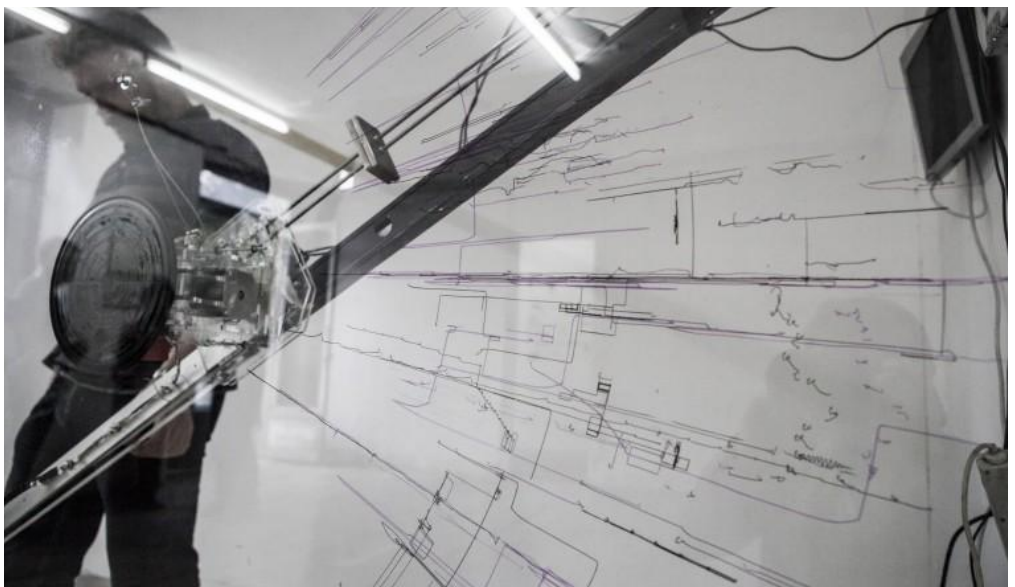
Depuis les années 90, chaque génération pense que la prochaine résoudra tous ses problèmes grâce aux machines et aux robots qu'elle sera capable d'inventer. Nous sommes probablement celle qui atteindra cet objectif. Cette affirmation que la presse a passé sous silence, pour des raisons basement capitalistes que nous n'énumérerons pas ici afin d'assurer une promotion convenable de l'exposition dans les journaux, cette affirmation, disions-nous, volontairement péremptoire, n'est pas divulguée, aujourd'hui, à la légère.

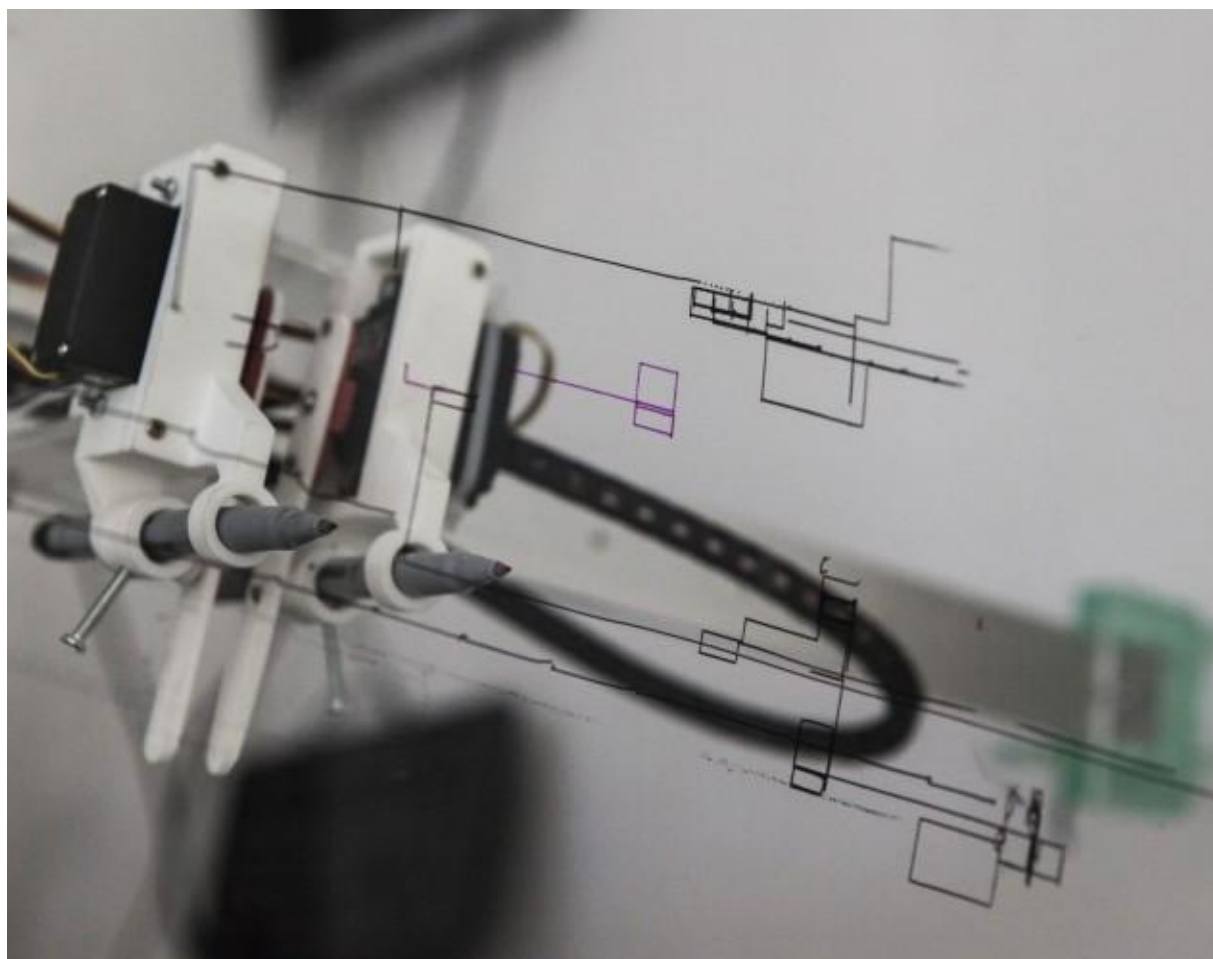
→ Pièce
imprimée en 3d
pour le
positionnement
du stylo

.CORP croit que le grand public est en mesure de ne pas céder à la panique ou à la folie désinhibitrice qu'une telle annonce peut provoquer chez les personnes les plus faibles. Oui, sans mégalomanie, nous pouvons avancer aujourd'hui que nous sommes la génération élue. Nul besoin de génie, nous en avons peu parmi nous, nul besoin de divinité, les machines vont bientôt être créées pour accéder à nos moindres désirs. Le processus a déjà commencé, dans les sous-sols, les garages et les caves, dans les ateliers clandestins, notre génération programme des robots capables de répondre à telle ou telle tâche ingrate que Madame ne supportait plus d'accomplir. Nous n'avons ici qu'un exemple parmi des milliers d'autres, mais il n'est pas négligeable. Il n'existait pas de machine qui lèche les vitrines au stylo, il existe aujourd'hui Windows Licker.

Cette pièce a aussi été présentée au festival Accès(s), à Pau.
Avec la participation de Pierrick Faure et Jacques-Daniel Pillon
Réalisé avec le soutien : DICRéAM (CNC) – Galerie Ars Longa, Paris

→ Compositions
générée par le
robot durant le
temps de
l'exposition





† Détail du dispositif
sur une vitre à
l'assaut de la
menuiserie

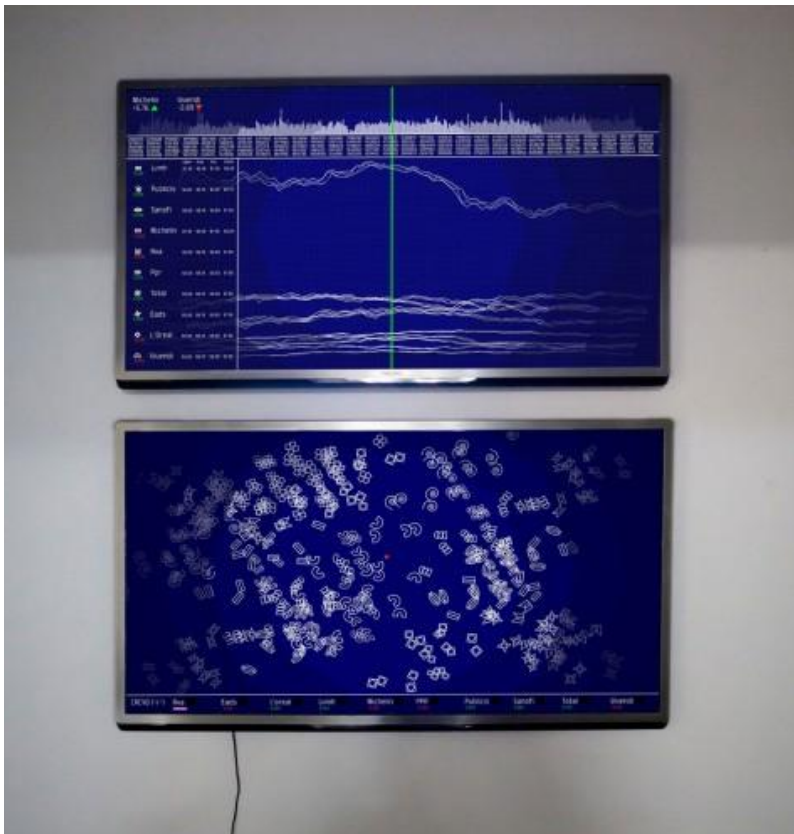
Jeux vidéo basé sur l'économie du CAC 40
2012

Il est des hommes que l'on croise dans la rue, tôt le matin mais souvent tard le soir, habillés de noir, cravatés, parfois ridiculement, qui marchent d'un pas sûr dans les rues des grandes métropoles. Ils ont, systématiquement, une mallette, qu'ils cramponnent comme un soutien, si bien que leur enlever les ferait basculer. Nous les croisons ainsi, bien souvent seuls, sans savoir vraiment où ils se dirigent. Certaines de nos connaissances disent en avoir comme voisins, parfois même comme amis, mais il nous est permis d'en douter. Leur mode de vie nous inquiète, et à juste raison. Ces hommes-là ont leur propre langage, qui n'est pas un langage oral mais composé de chiffres, de courbes, de flèches qui montent, de flèches qui descendent, de la couleur verte et de la couleur rouge. Ces éléments sont exploités de telle manière qu'ils composent un code qu'eux seuls sont en mesure d'appréhender. Leurs chiffres défilent à toute allure sur les écrans noirs qu'ils possèdent en double, voire triple, exemplaires, ce qui nous donne une idée de la vitesse à laquelle ils sont capables d'intégrer les informations, pire encore, ils paraissent réagir en temps réel à ces données. Il est donc nécessaire, aujourd'hui, de questionner l'influence qu'ils ont sur l'instabilité de notre monde, mais surtout d'envisager le pouvoir que nous-même pouvons exercer sur cette masse informe qu'ils représentent. L'ennemi est insidieux, entraînons-nous !

Le simulateur mis à disposition par nos services s'appuie sur les résultats, des dix dernières années, de dix entités, qui sont au centre des intérêts de ces hommes-là, et qu'ils nomment CAC40. Des signes les représentant se multiplient ou se divisent en fonction de la montée ou de la descente de leurs courbes. Il s'agit alors de savoir si l'on veut qu'ils occupent tout notre territoire.

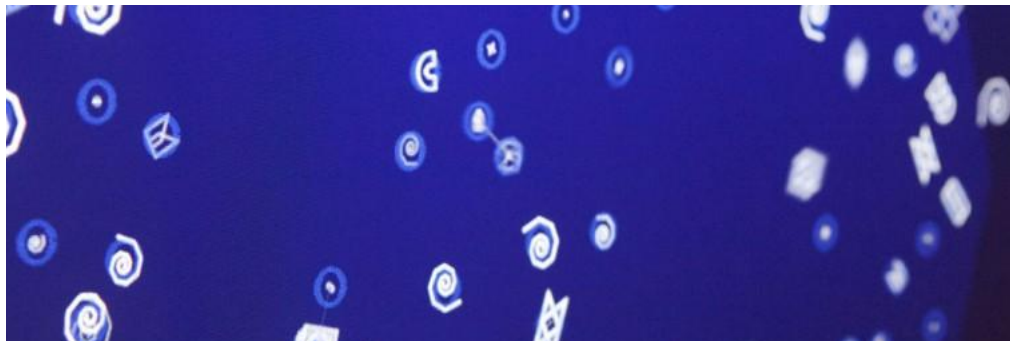
Design sonore : FullPitch

Réalisé avec le soutien : Fonds [SCAN], Région Rhône-Alpes



← Dispositif :
sur l'écran du
haut, les courbes
économiques du
CAC40, sur celui
du bas le jeu.

↓ Détail du jeux
vidéo et des
symboles
représentants les
entreprises



← Vu de
l'exposition à
l'assaut de la
menuiserie

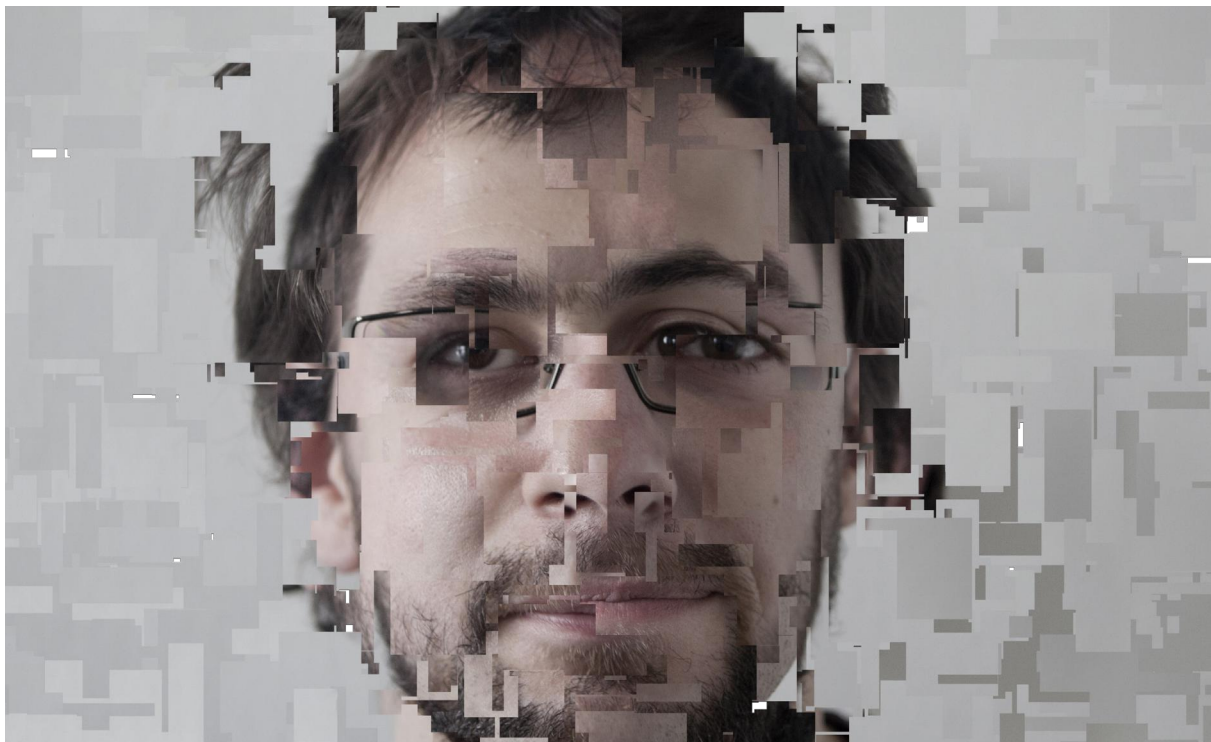
Programme génératif photographique 2013

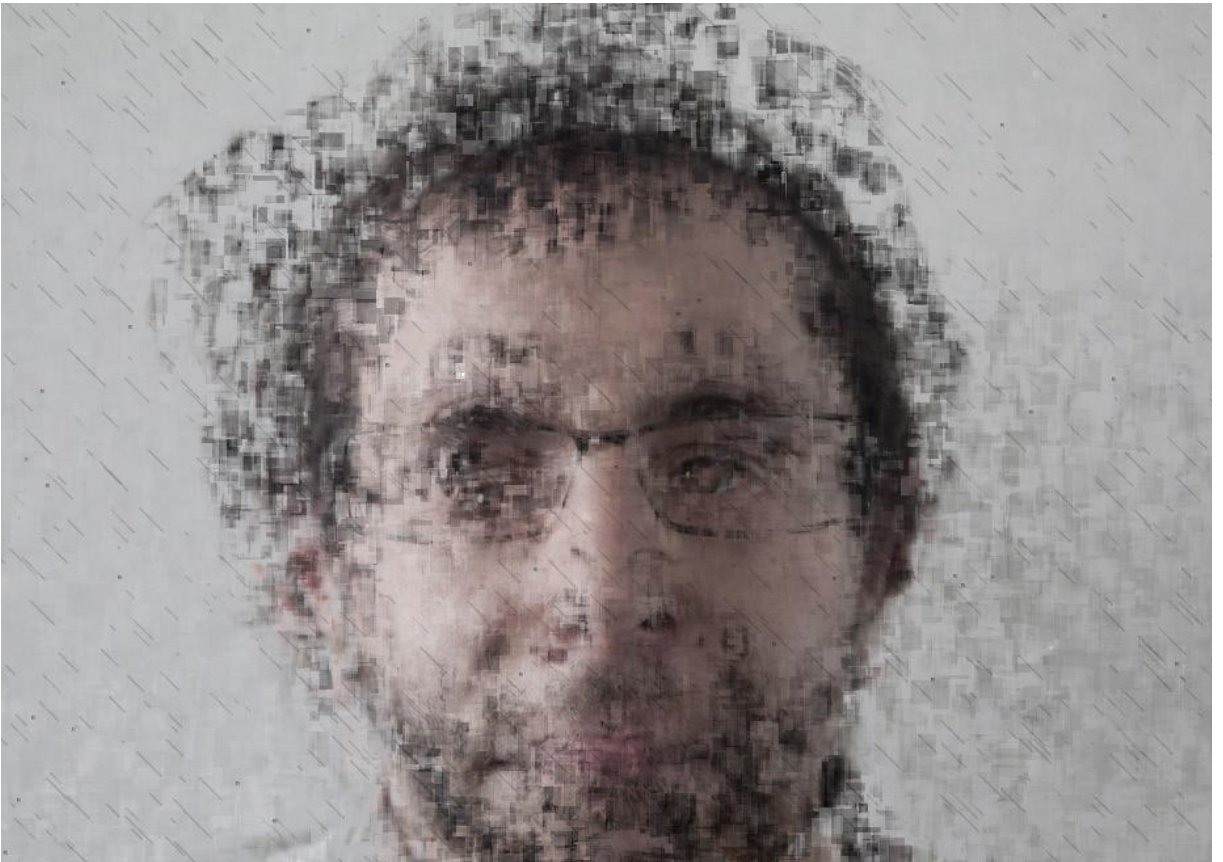
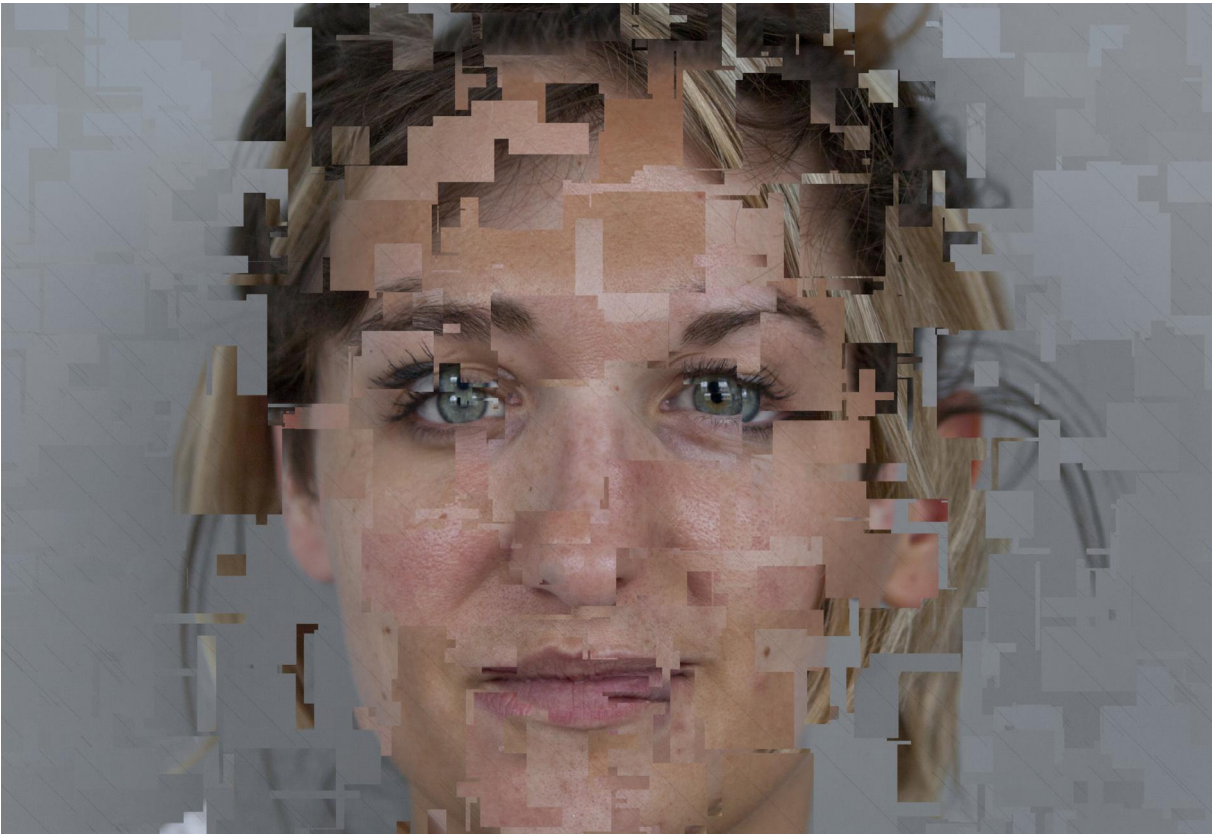
Ce visuel, généré avec un logiciel programmé par .CORP pour la biennale de Saint-Etienne 2013, évoque l'entreprise du designer qui tente de comprendre le plus justement possible les besoins et les émotions d'un individu pour proposer des solutions à un ensemble d'utilisateurs. Le designer s'adresse à la multitude à partir du particulier. La question à l'origine du projet est : si la machine à empathie du roman de Philip K. Dick Les Androïdes rêvent-ils de moutons électriques, mise en images par Ridley Scott dans le film Blade Runner existait, à quoi ressemblerait l'image de sa mémoire vive ? Le visage humain est la source première d'empathie, ce par quoi transparaît la plupart des émotions. Cette machine a gardé en mémoire une base de données de visages pour les comparer et les analyser. Un logiciel spécialement conçu pour la Biennale 2013 permet de produire des visages reconstitués partir de multiples morceaux de portraits.

Expositions :

- Biennale Design 2013

↓↘→ Trois
compositions
générées par le
programme





Citizen Saint-Etienne

Site internet pseudo réseau-social
2009

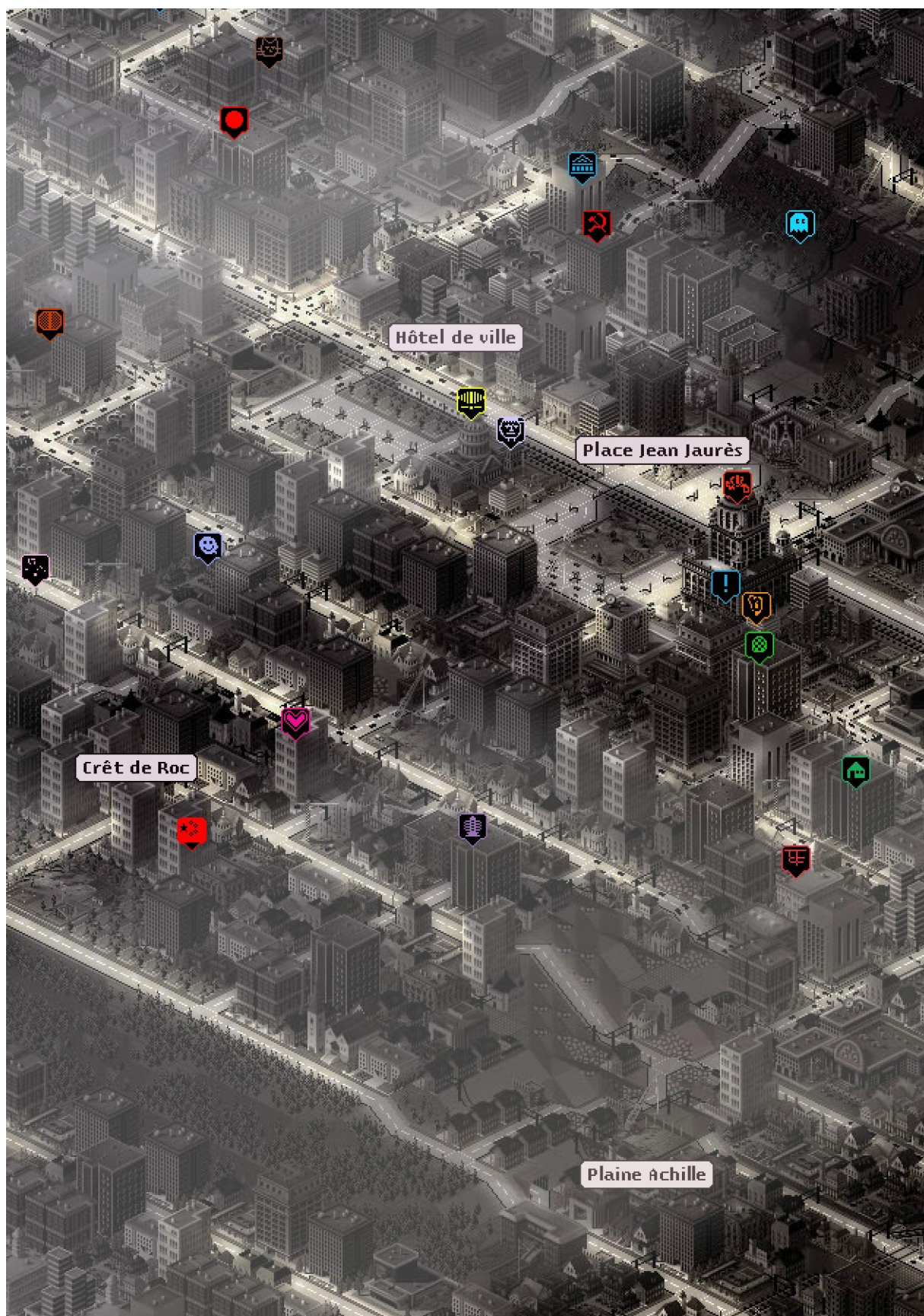
Carte de Saint-Étienne en 3D à partir des
architectures de Sim-City 2000 comprenant
plusieurs niveaux d'interaction avec l'internaute.
Chaque visiteur peut de manière anonyme laisser
à la fois une icône et un message de façon pérenne.

→ Capture
d'écran du site
web Citizen

www.citizen.corp-lab.com

↓ Carte générale
de Saint-Étienne



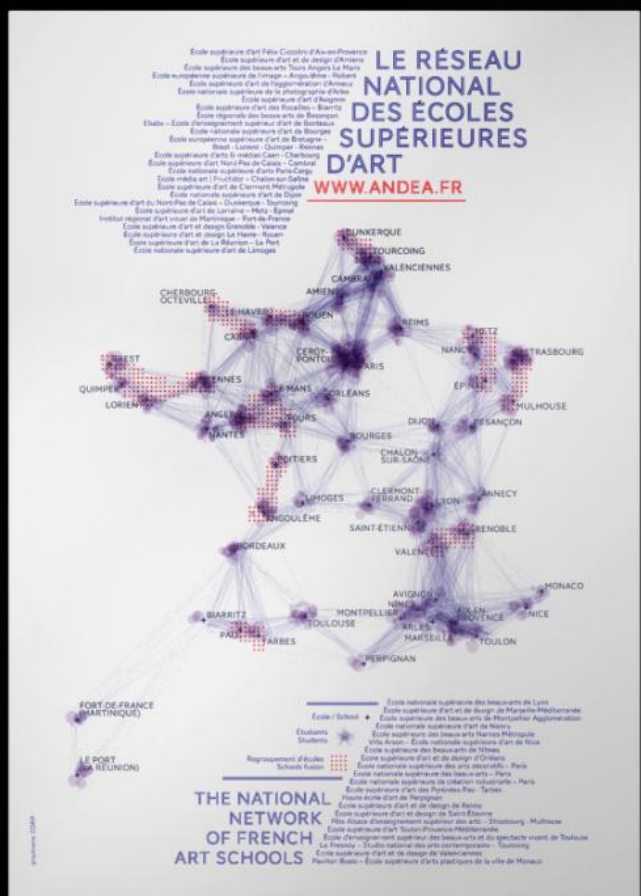
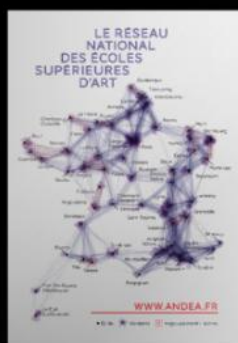


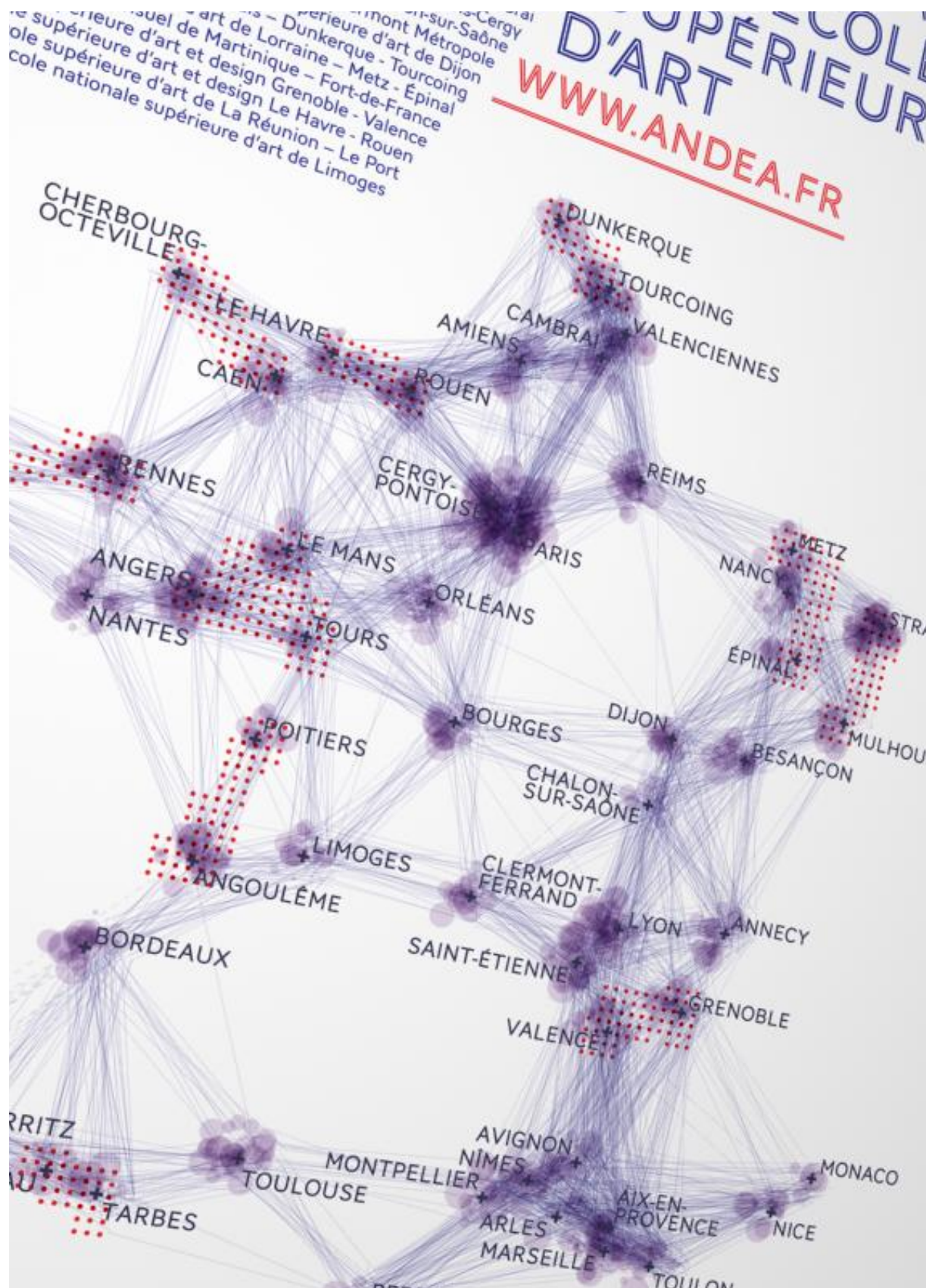
Réseau des écoles d'art en France

Graphisme et visualisation de donnée
2010

L'ANDEA (Association nationale des directeurs d'écoles d'art) a commandé à .CORP une cartographie du réseau des écoles d'art en France pour Art Press 2 qui a ensuite été déclinée sur flyers. La carte a été générée avec l'outil de programmation Processing.

→ Carte générale du réseau des écoles d'art sous diverses forme de publication





† Détail de la
carte générale du
réseau des écoles
d'art

Nouveaux Paramètres

Graphisme génératif / Communication
2015

Création de l'affiche/programme de l'exposition collective Nouveaux paramètres à la galerie Ars Longa, de novembre 2011 à avril 2012. Le visuel de l'affiche est généré avec Processing à partir d'une collection de motifs tramés.

↓ Détail du système de tramage

→ Affiche générale de l'exposition



GALERIE
ASTORIA
EXPOSITION

HELVETICA
PARADISE

NOV 2012
MAY 2013

Ars Longa, Paris

Identité graphique générative
2015

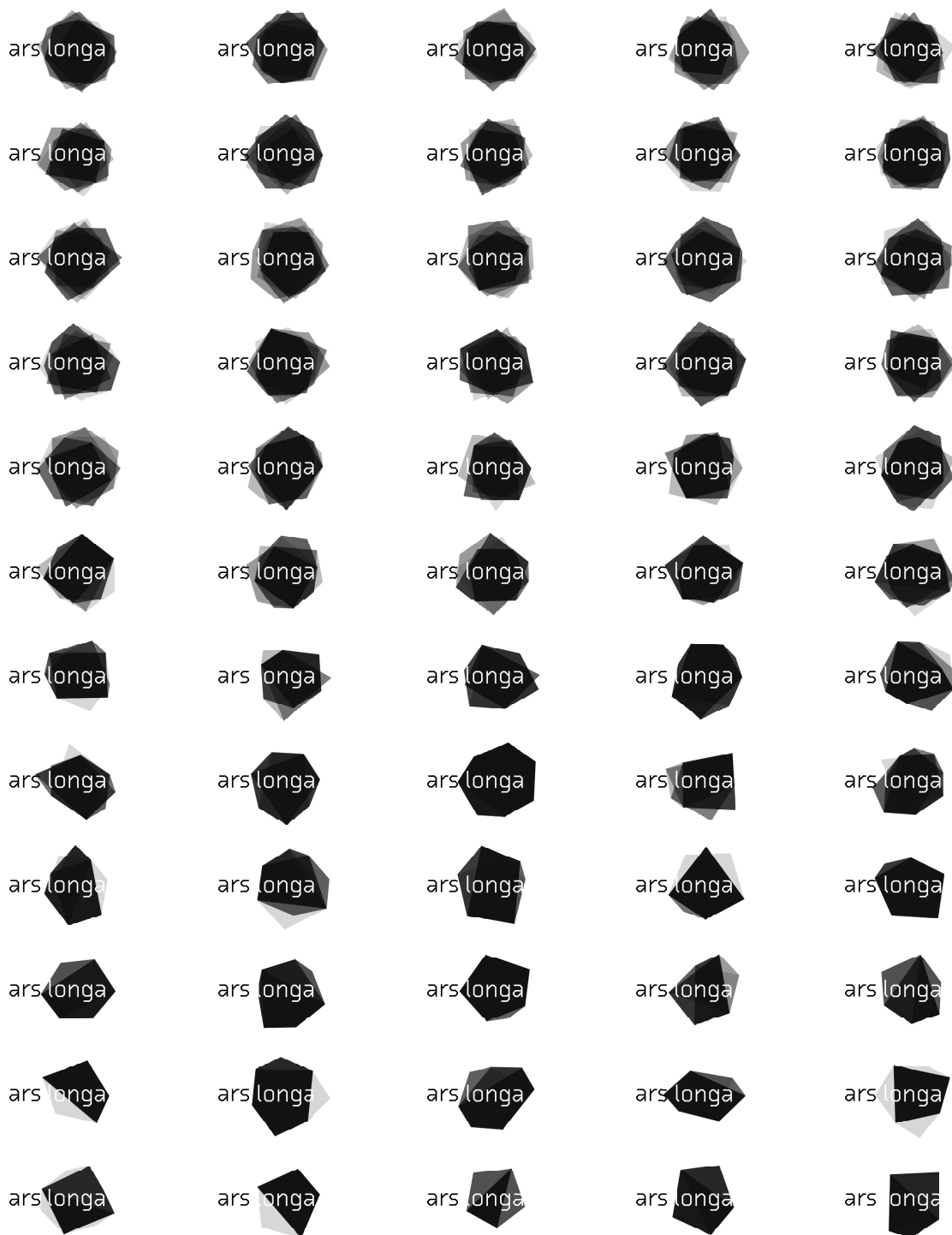
Création d'une identité visuelle dynamique et modulaire pour Ars Longa, structure de production, de diffusion et de médiation de projet artistique.



← Exemple de
logo généré avec
le programme

→ Planche de
variation du logo



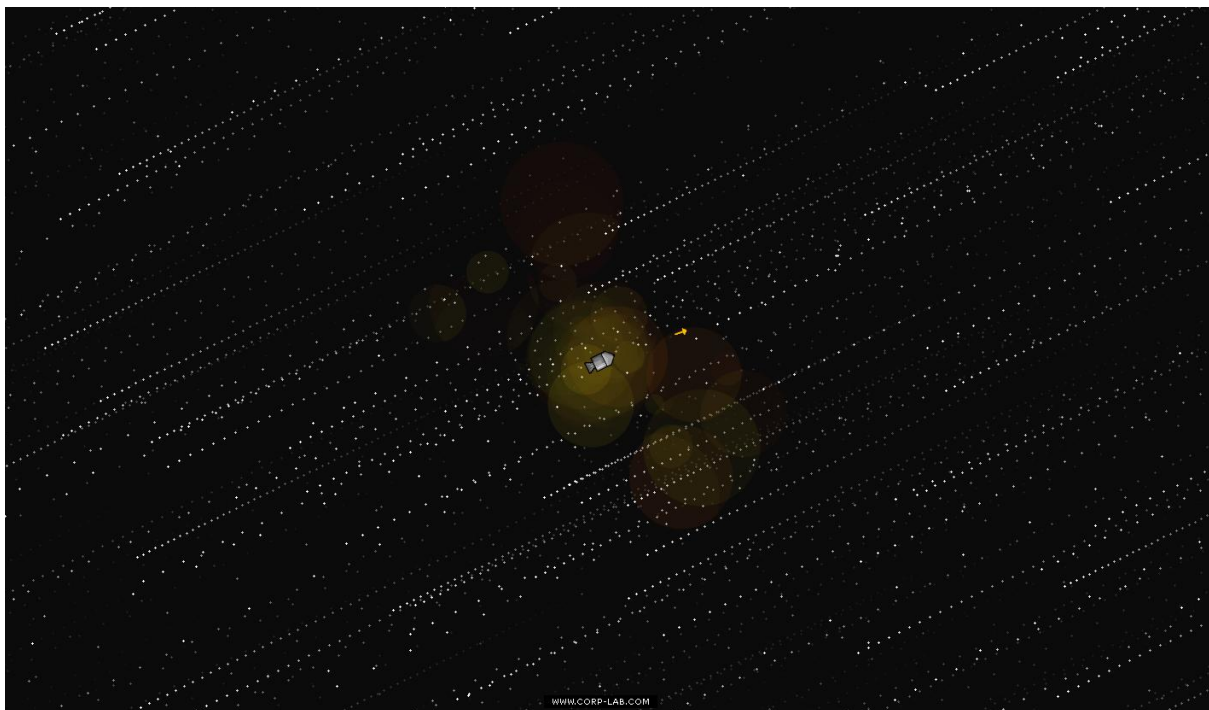


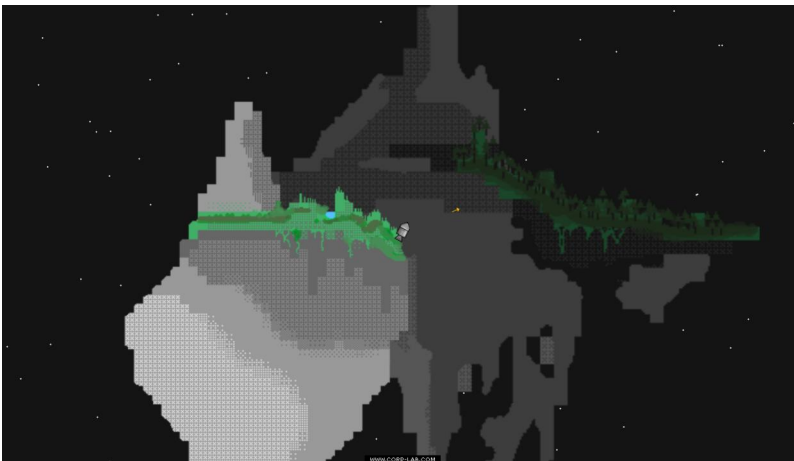
Happy New Year 2015

Site web / Jeu vidéo
2015

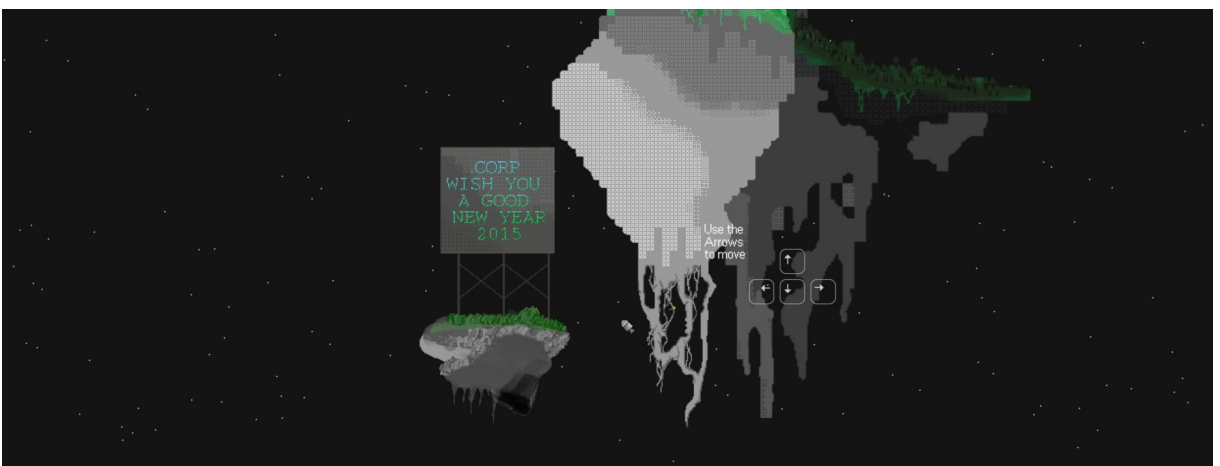
"Happy New Year" est une série de pages présentant la bonne année à nos partenaires. En 2015, un mini-jeu vidéo a été mis en route à cette occasion. Ce genre de projet est toujours l'occasion d'expérimenter des fonctionnalités inédites des navigateurs ou de nouvelles bibliothèques de code, par exemple dans ce projet l'avènement de P5.js.

↓ Capture écran
du site internet





↑←↓ Capture
écran du site
internet



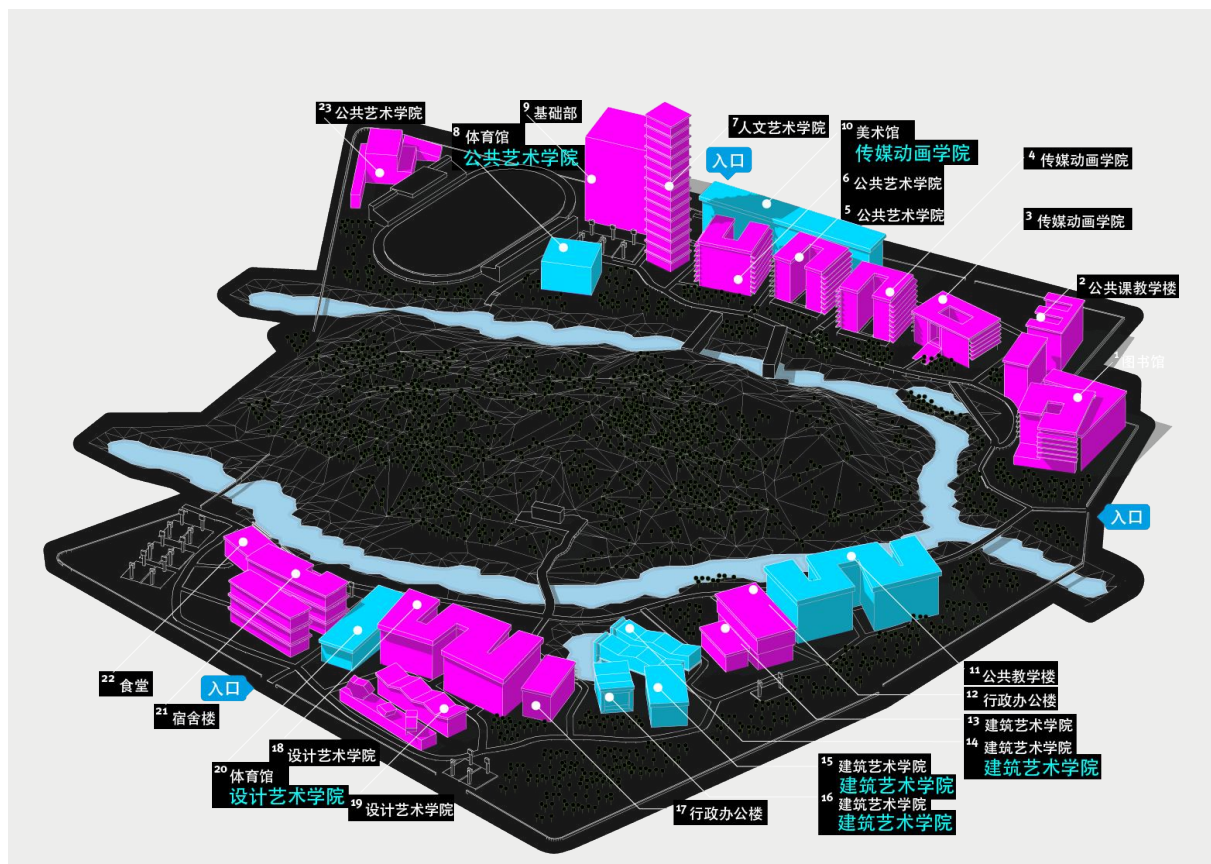
China Exhibition Graduate School, China

Site web et identité graphique
2016

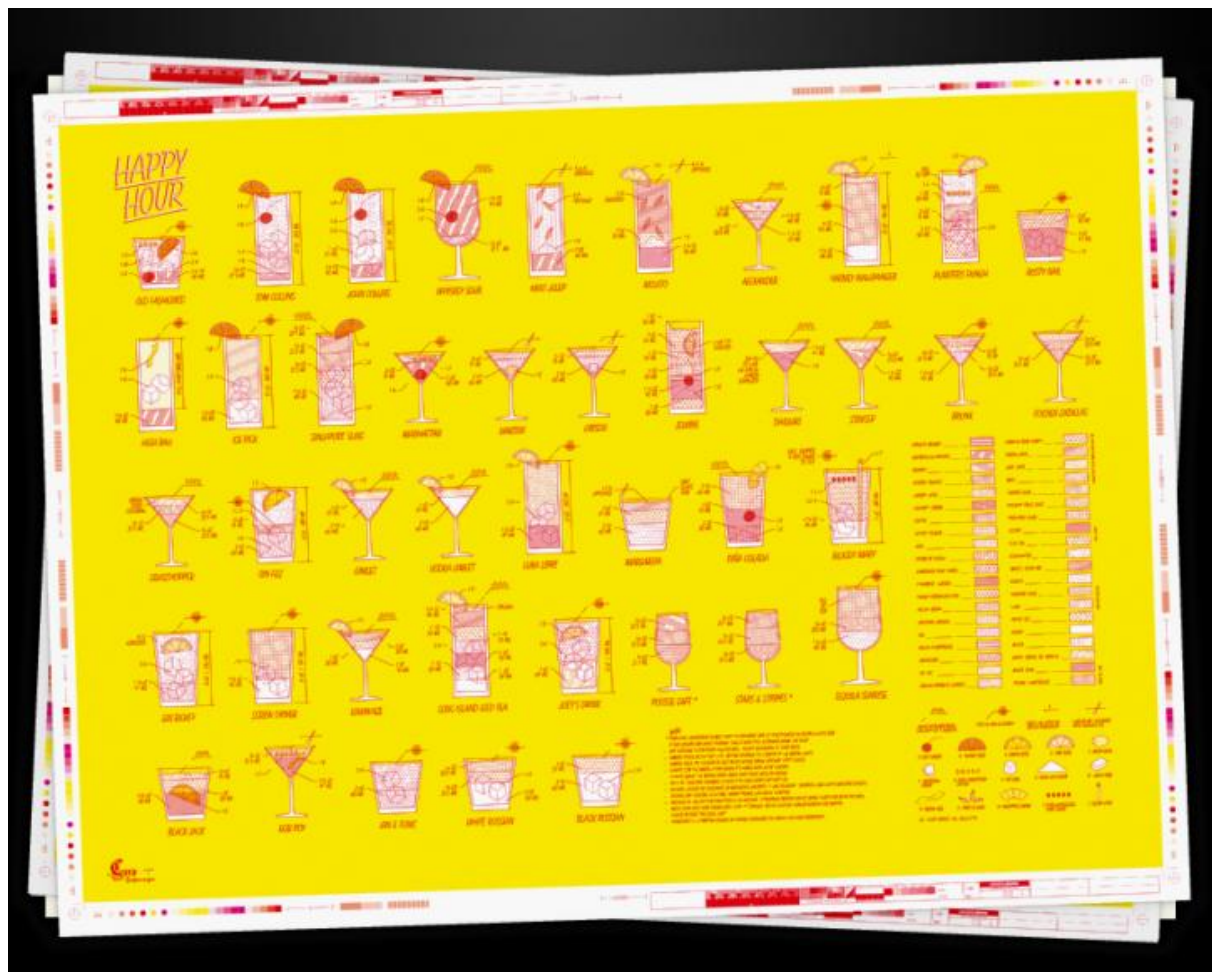
Identité visuelle et site Web de l'exposition CAA
Graduation Exhibition 2012 présentant les travaux
des étudiants sortants de l'Académie des beaux
arts de Chine à Hangzhou.

— \ \ Site
internet créé pour
l'évènement

↓ Détail de la
carte



↓ Tirage offset de la carte "Poster cocktail"





1. TAPE ASSEMBLY WITH BANGLED PEE & REMOVE

2. STRIP ON AND ASSEMBLY

3. GLUE ON LAYER ONE TOP OF BANGLED PEE

4. STRIP ON AND ASSEMBLY WITH BANGLED PEE

5. RED CIRCLE

6. CHOPPING SLIT

7. CHOPPING SLIT

8. CHOPPING SLIT

9. CHOPPING SLIT

10. PINK CIRCLE

11. PINK CIRCLE

12. PINK CIRCLE

13. PINK CIRCLE

14. PINK CIRCLE

15. PINK CIRCLE

16. PINK CIRCLE

17. PINK CIRCLE

18. PINK CIRCLE

19. PINK CIRCLE

20. PINK CIRCLE

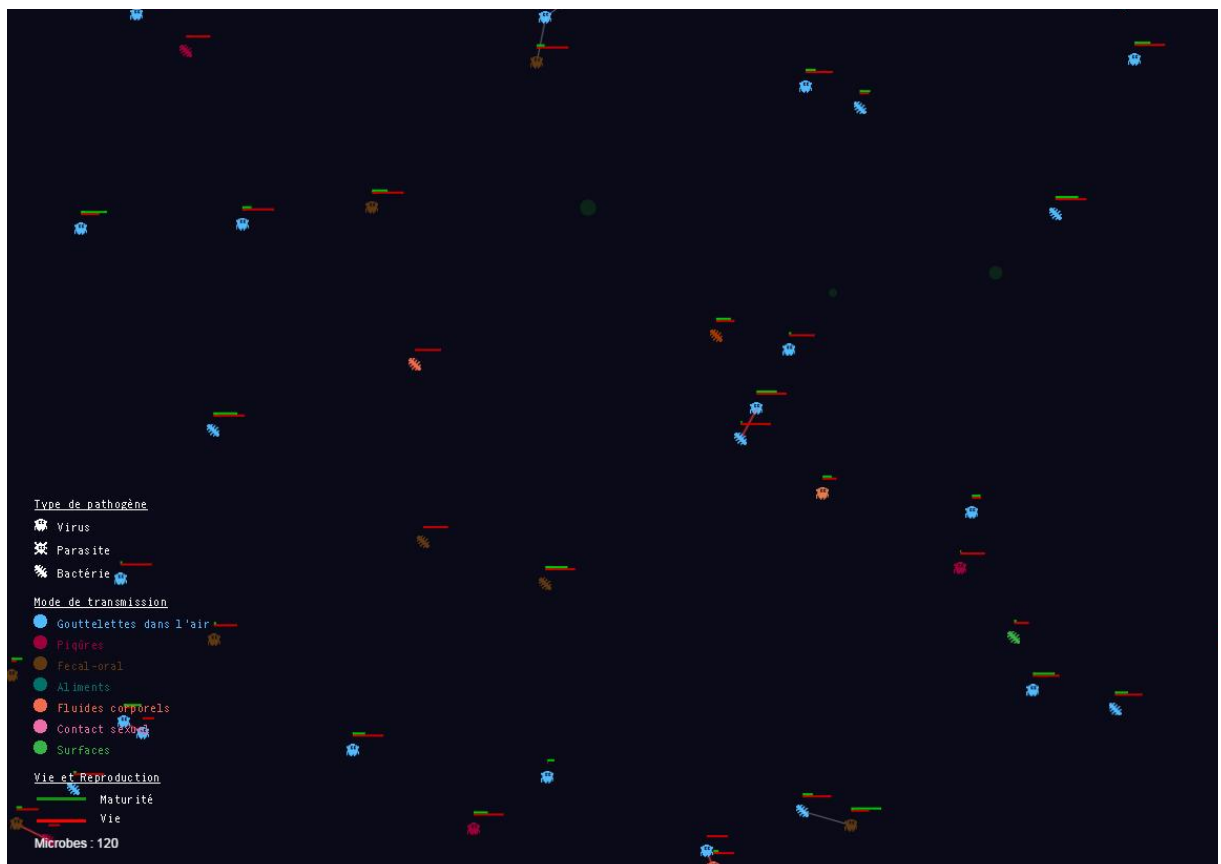
Pest Wars

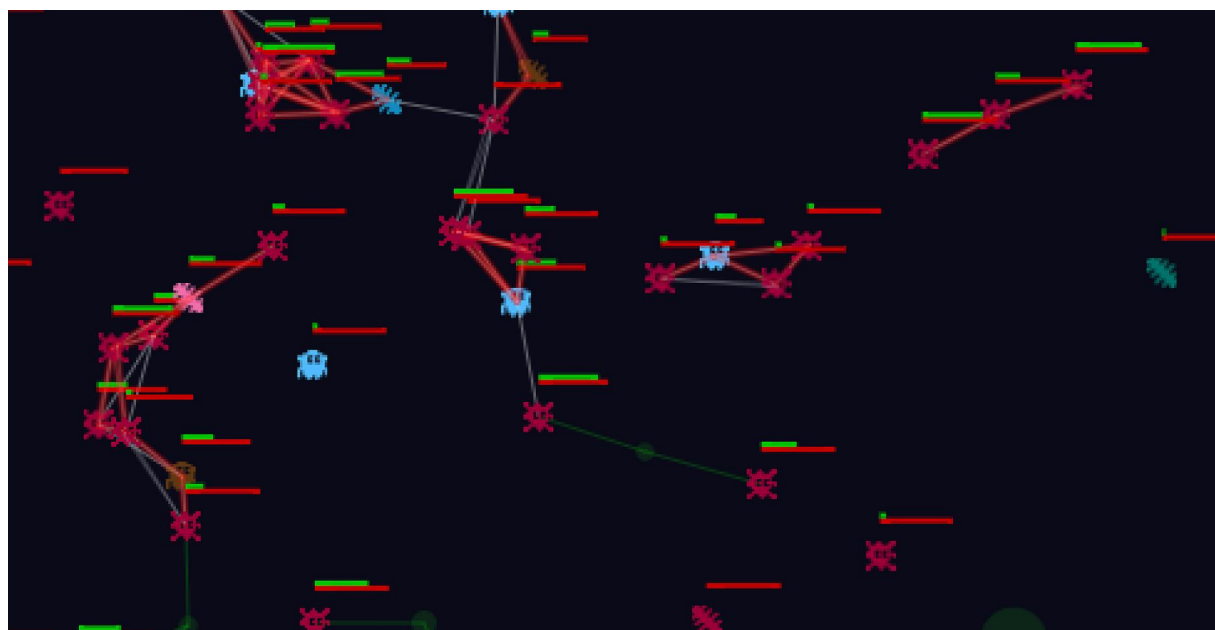
Performance numérique / Conférence
2013

Représentation/simulation des 40 maladies les plus mortelles dans le monde dans une "sand box".
Programme codé en deux heures l'occasion d'une invitation à Haute école Albert Jacquard (Namur).

↓ Capture écran
complète de
"Pest war"

→ \ Différents
états du jeu vidéo





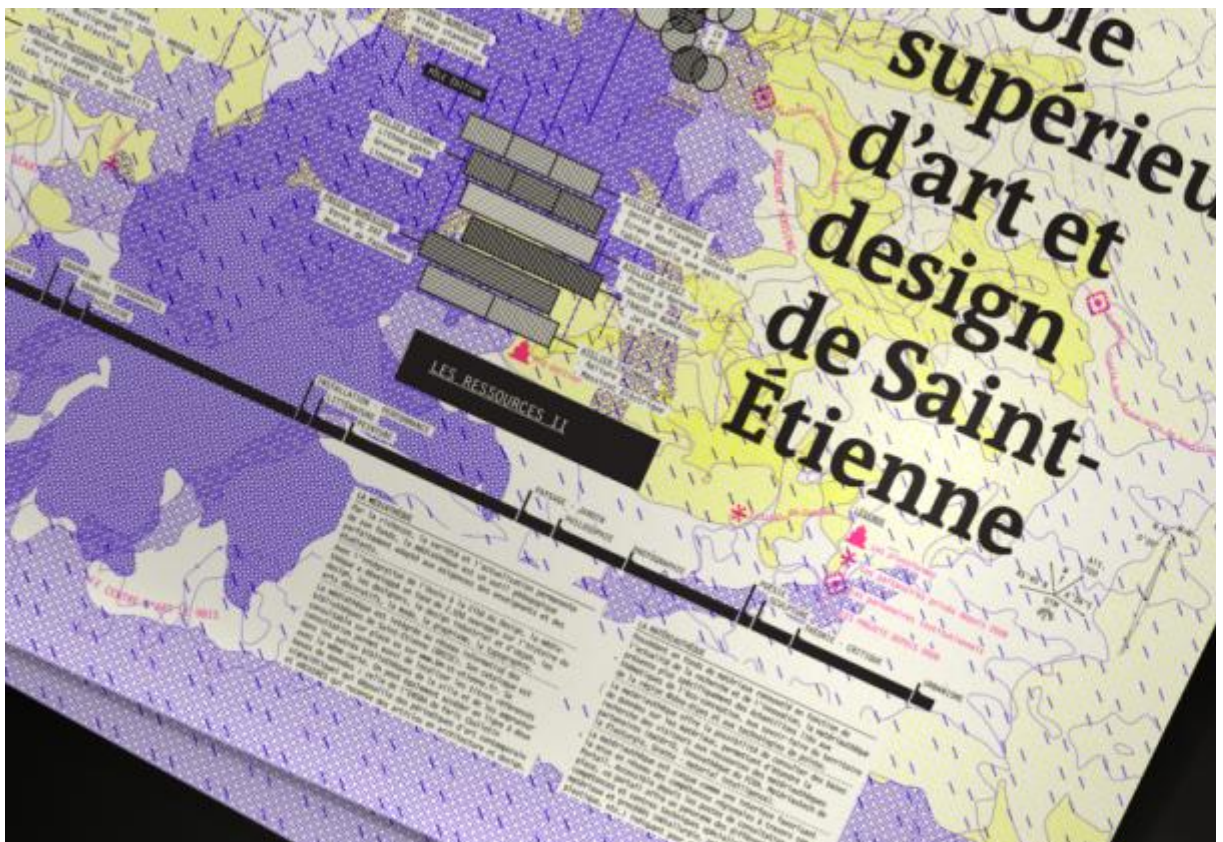
Cartographie d'une école d'art

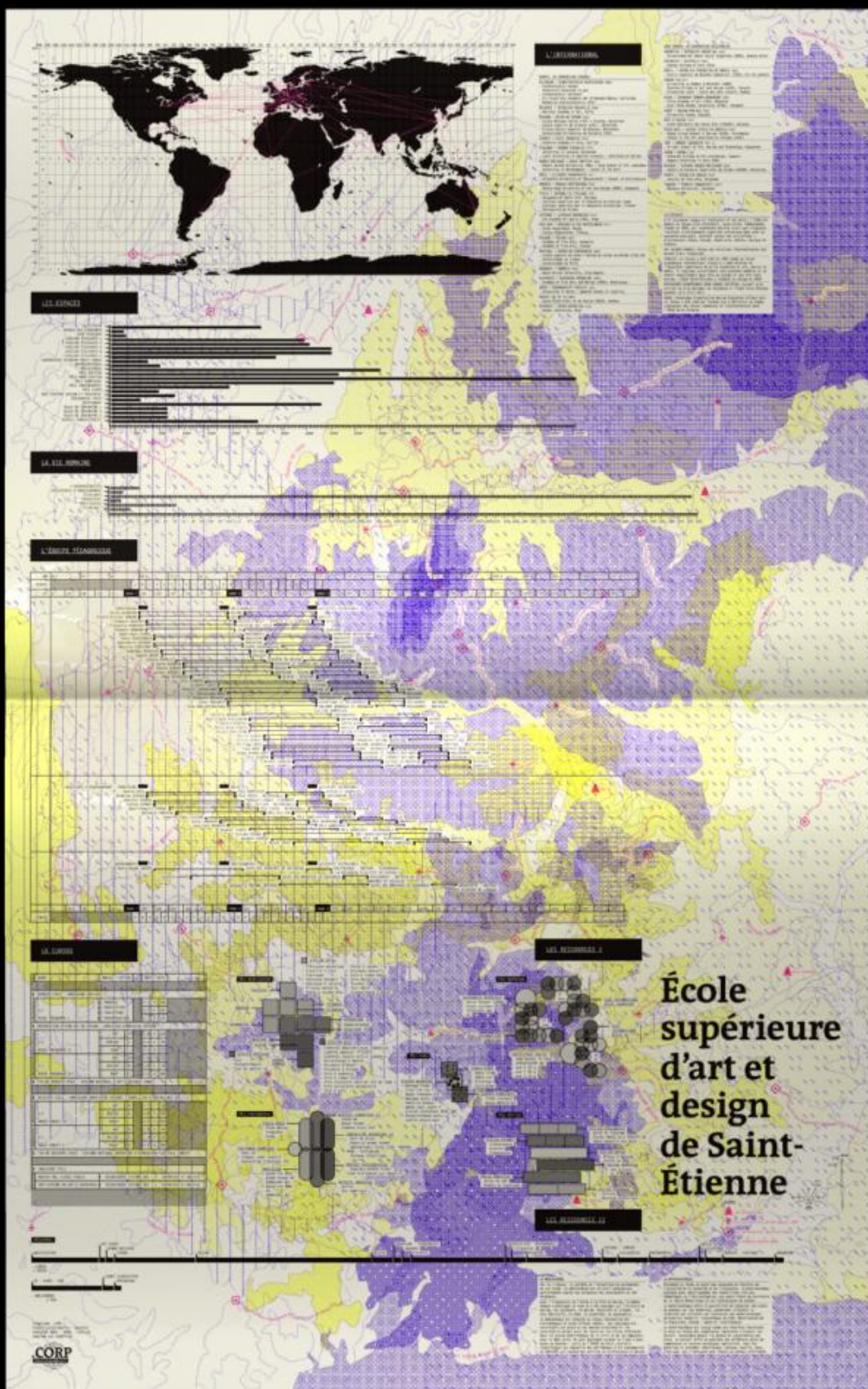
Graphisme / Communication
2012

Représentation cartographique de l'École
supérieure d'art et design de Saint-Étienne à partir
d'une analyse de données.

→ Affiche
représentant tout
les ressources de
l'école

↓ Détail de
l'affiche





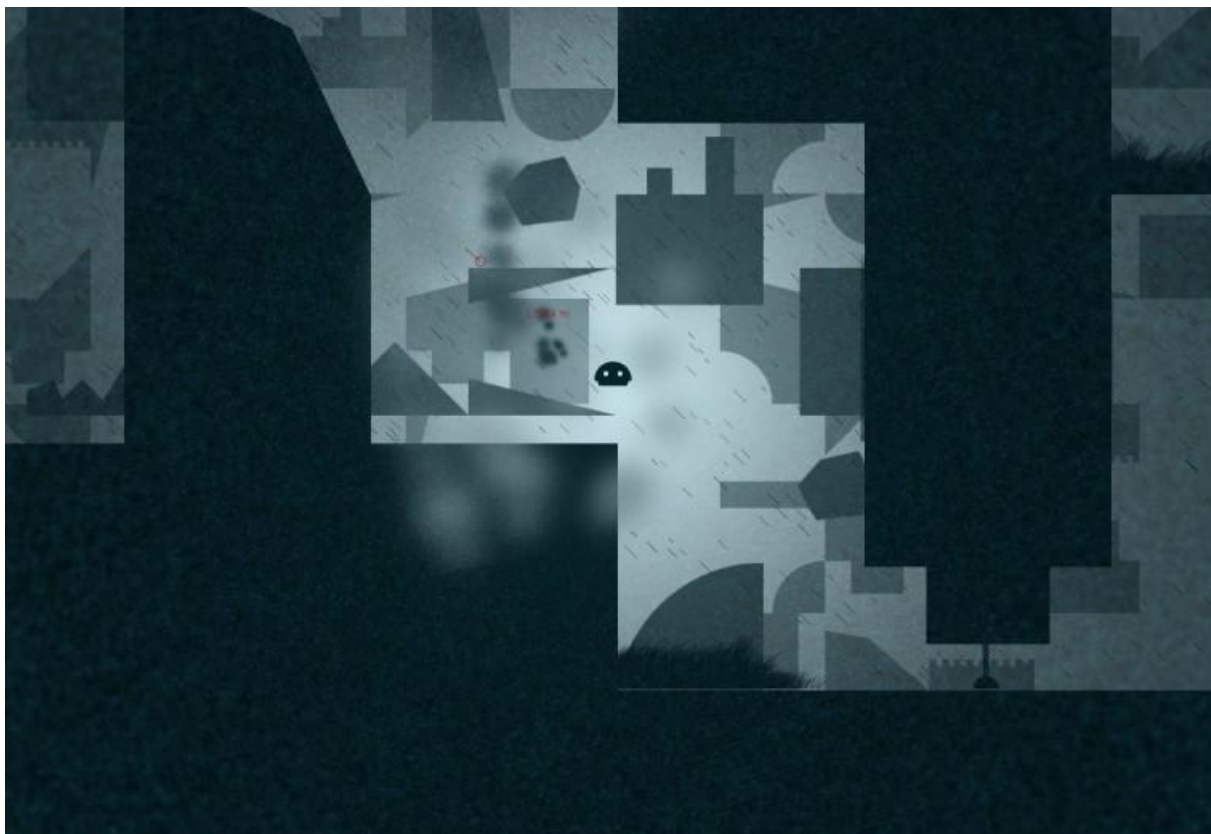
Hide & Seek

Jeu vidéo / Installation
2012

Présenté dans un mobilier qui s'inspire des bornes d'arcade et développé avec Processing, Hide & Seek est un jeu vidéo pour deux joueurs basé sur le principe du cache-cache. Les décors sont générés aléatoirement par un algorithme. Hide & Seek a été présenté à la biennale design de Saint-Étienne.

↓ Capture d'écran
du jeu





↑ Capture d'écran
du jeu



← Jeu sous forme
de borne d'arcade
pour une
exposition

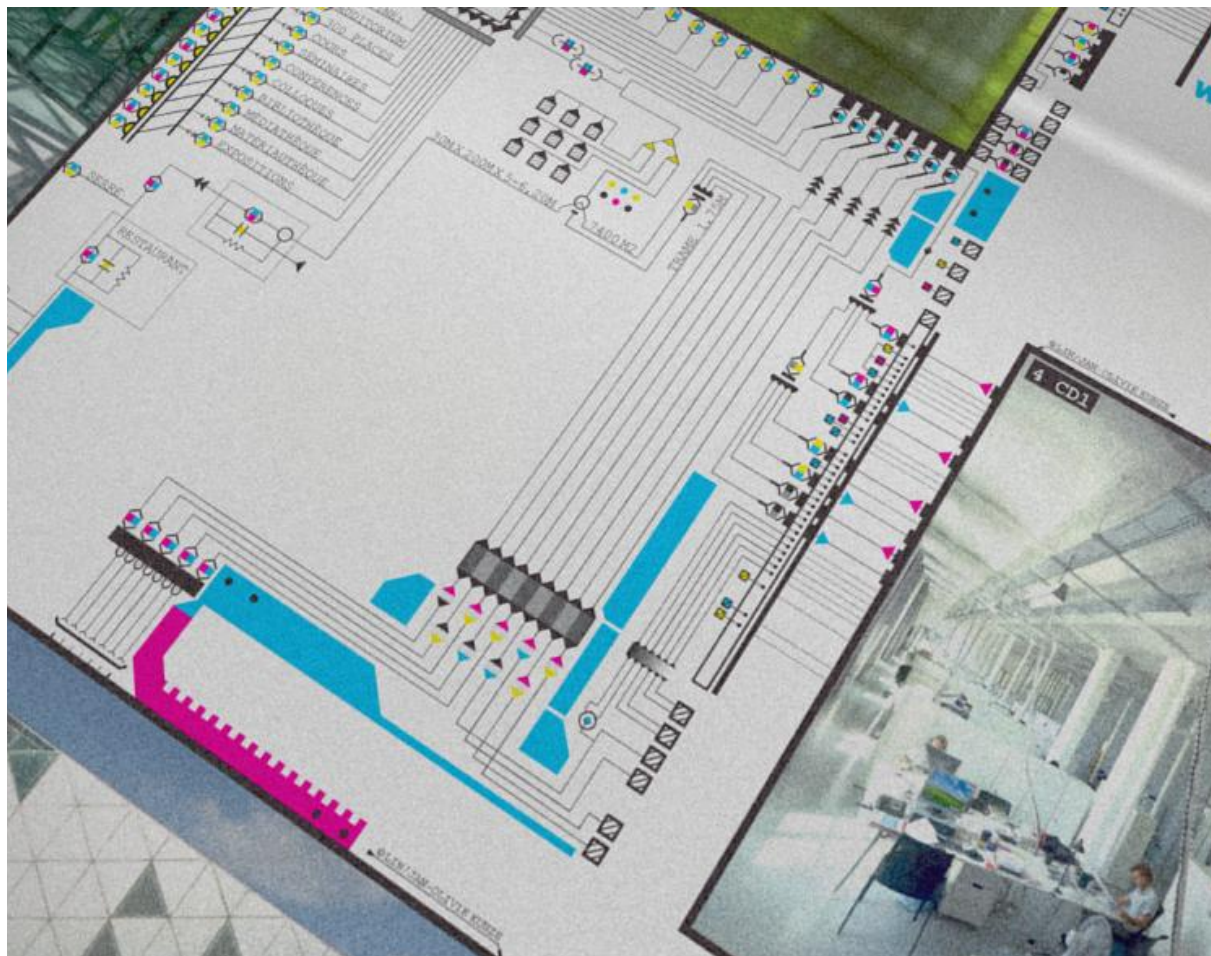
La Tribune/ Le progrès

Communication / Expérimentation graphique
2010

Le quotidien régional La Tribune – Le Progrès a offert une carte blanche à .CORP à l'occasion de l'inauguration de la Cité du design de Saint-Etienne. Dans sa proposition .CORP a intégré le placement habituel des colonnes de textes et d'images en perturbant ce qui les relie. Les flux d'énergie entre les différentes plateformes dessinent le réseau de la Cité.

→ Photographie
de la "Une"
publiée

↓ Détail de la
"Une"



CNAP

Rapport d'activité

DataVisualisation / Publication
2010

Le Centre National des Arts Plastiques a com-mandé à .CORP son rapport d'activité annuel.

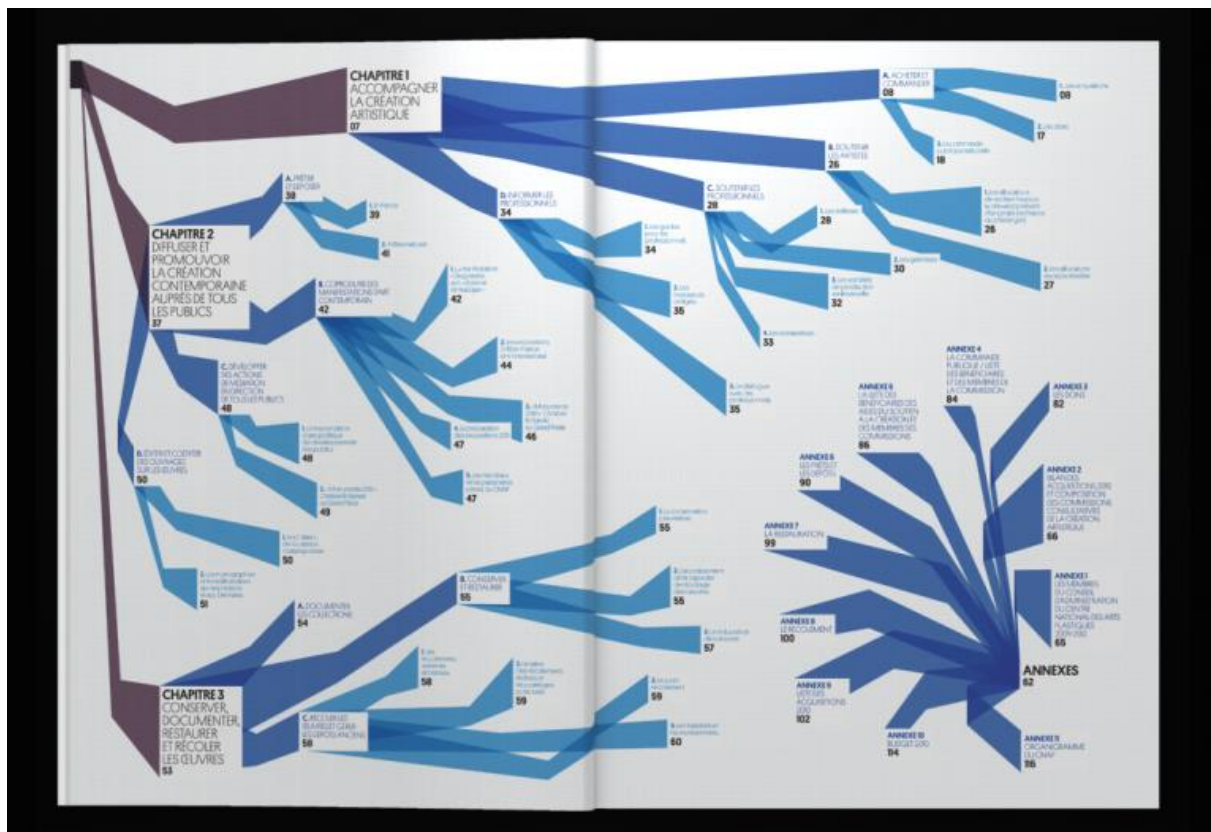
Le langage de programmation Processing a permis de construire les squelettes des diagrammes en améliorant la spatialisation des éléments et en affectant des propriétés physi-ques aux entités géométriques.

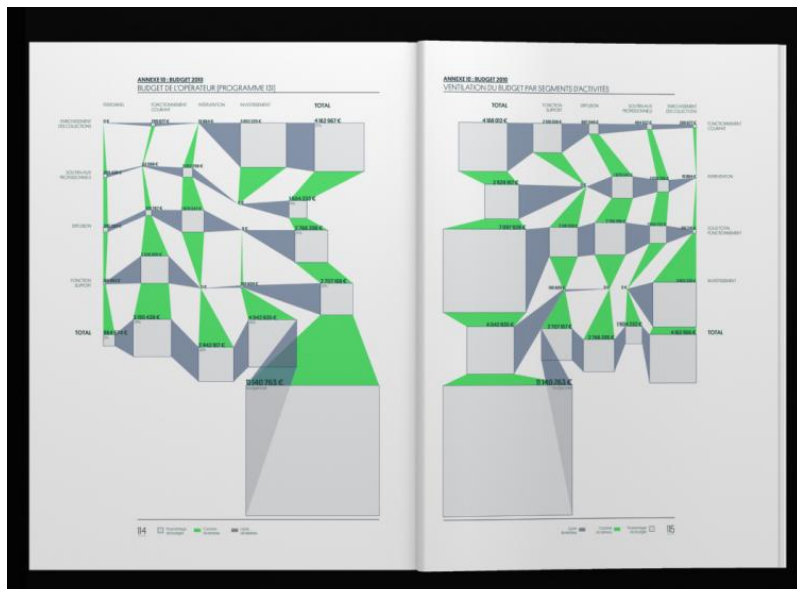
La matière ajoutée aux squelettes a été limitée : 2 couleurs, des épaisseurs de tracés, des trames de points.

La grille de la partie texte et images du rapport garde l'aspect géométrique des diagrammes à travers des éléments discrets mais répétitifs : cercles, rectangles, losanges, tracés horizontaux.

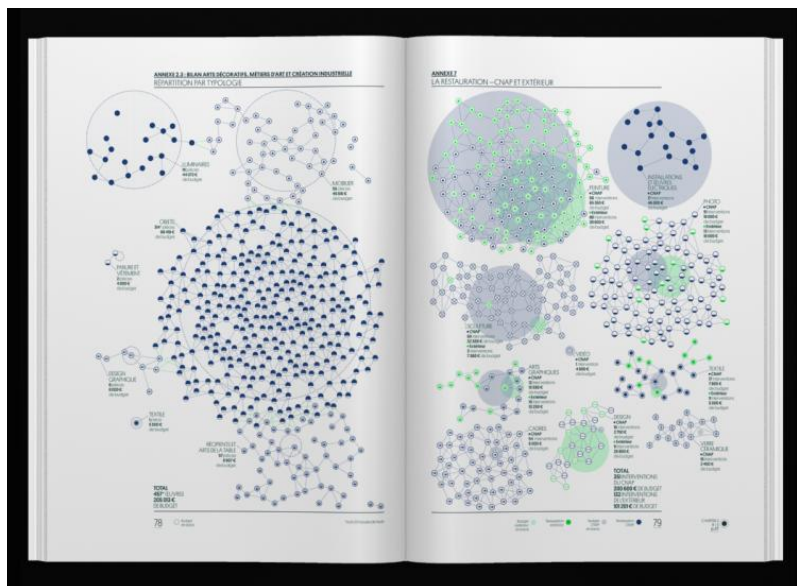
Le rapport est composé avec les typographies Tiempos de Kris Sowersby et Platform de Berton Hasebe.

↓ Sommaire
généralisé par la
programmation





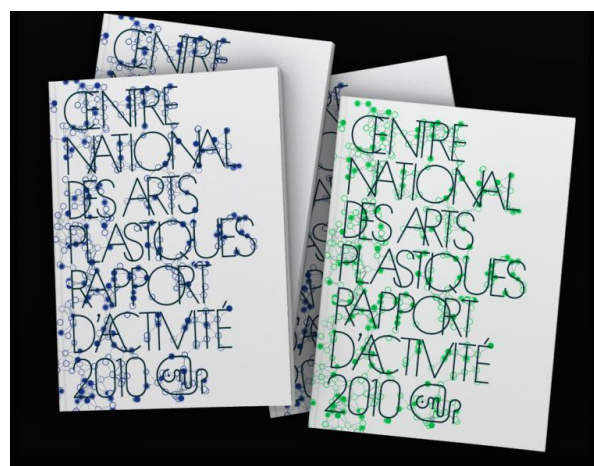
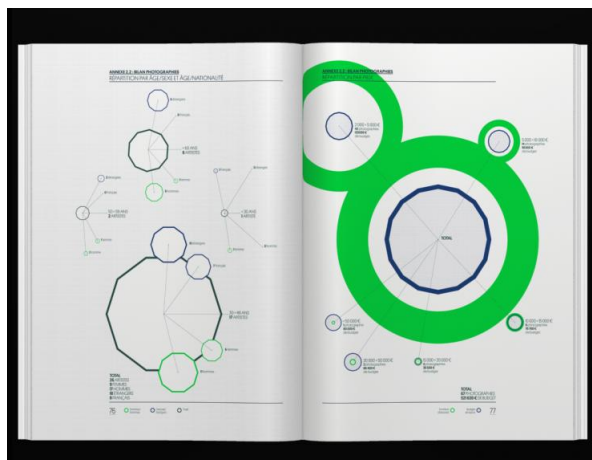
← Page présentant la répartition des budgets



← Page présentant la répartition des achats par typologie

↓ Page de couverture

↓ Bilan des aides photographiques



En une formule lapidaire

Installation / Graphisme
2010

Le Musée gallo-romain de Saint-Romain-en-Gal propose un face à face inédit entre objets archéologiques et objets design, .CORP réalise une pièce numérique destinée à être exposée entre une borne milliaire du 1er siècle et une stèle funéraire du 2e siècle. Par une suite de formes brèves cette pièce interroge ses relations et différences aux œuvres antiques qui l'entourent et cherche à établir des connexions entre les pratiques graphiques contemporaines et antiques.

→ ↗ Vue de l'exposition

→ Détail de la borne kilométrique





Dispositif outil / Electronique / Programme
2010

Avant-propos :

Le projet Freehand Tracer est issu d'une recherche menée par Damien Baïs avec le collectif .CORP. Il bénéficie d'un partenariat avec le random(LAB) de l'École supérieure d'art et design de Saint-Étienne. Une première version de Freehand Tracer a été montée durant l'été 2014 lors d'une résidence de .CORP à l'Assaut de la Menuiserie, lieu d'art contemporain de Saint-Étienne.

Le projet est également soutenu par L'association Carton Plein qui lui met un atelier à disposition depuis septembre 2014.

Une seconde étape est maintenant nécessaire pour finaliser la conception mécanique et logicielle du projet avant de pouvoir l'exposer et l'utiliser de manière optimale. Plusieurs expositions sont prévues notamment lors la biennale design Saint-Étienne 2015 et à l'Assaut de la Menuiserie.

Contexte :

Freehand Tracer s'inscrit dans le mouvement actuel de création, par les designers, de leurs propres outils. La recherche nécessaire à la réalisation de ce dispositif interactif multimédia passe par l'expérimentation et le mélange de différentes techniques et technologies (impression 3D, Arduino, programmation Processing, etc.)

Projet :

Freehand Tracer est un dispositif de dessin et d'écriture pouvant réagir à des données en temps réel. Cette "table traçante" est d'abord un outil ouvert et modulaire adapté pour tracer à partir de différentes sources de données.

Par sa capacité à tracer avec des plumes ou des pinceaux et simuler ainsi le dessin à la main, la machine rejoue le graphein original : "tracer des signes pour écrire et dessiner".

Comme la main de l'homme, le bras mécanique s'adapte en temps réel aux données qui lui sont transmises et peut être paramétré pour peindre un paysage comme pour dessiner des visualisations de données économiques.

Caractéristiques techniques et technologiques :

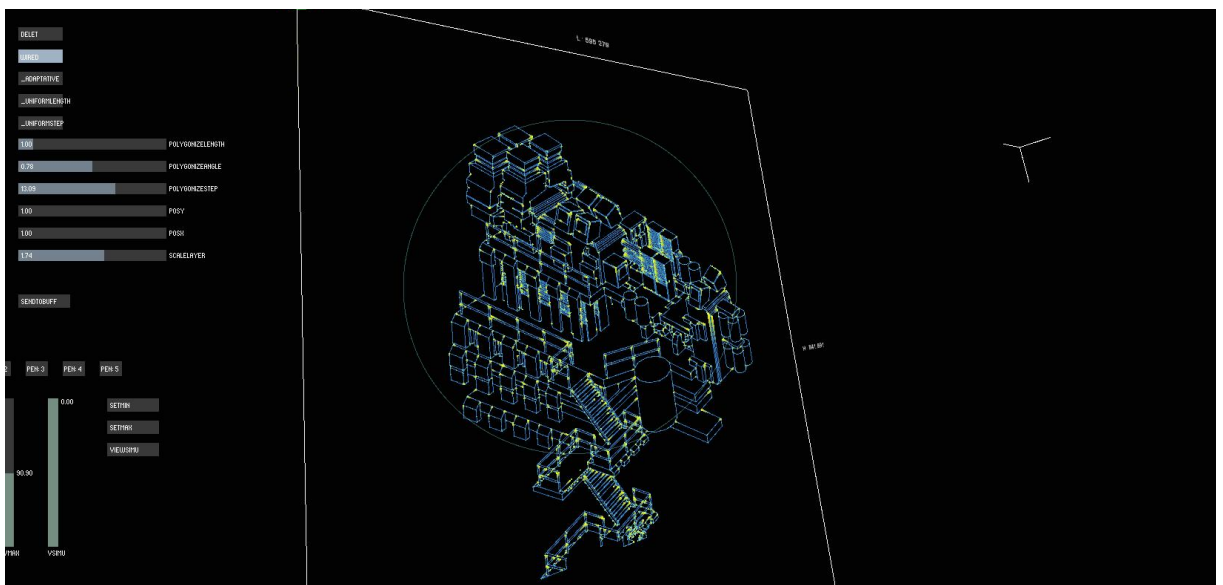
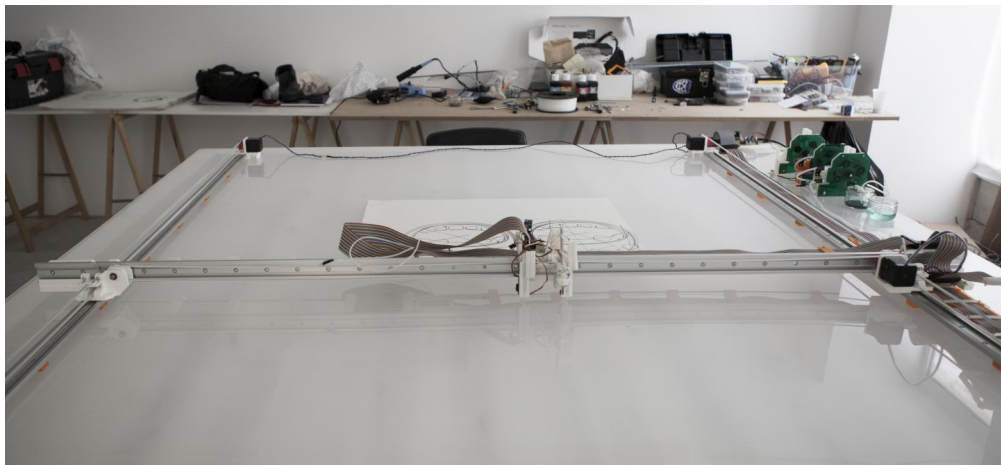
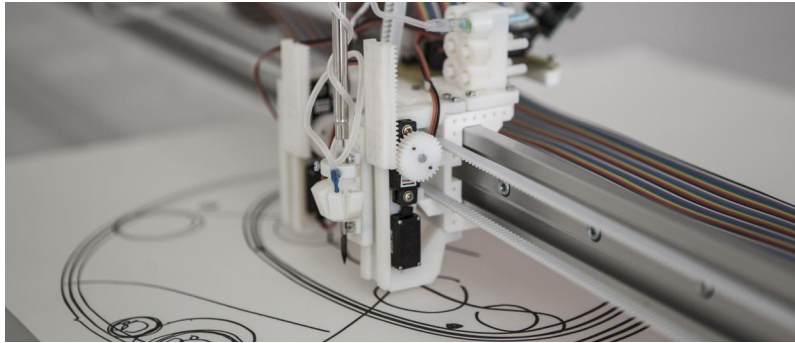
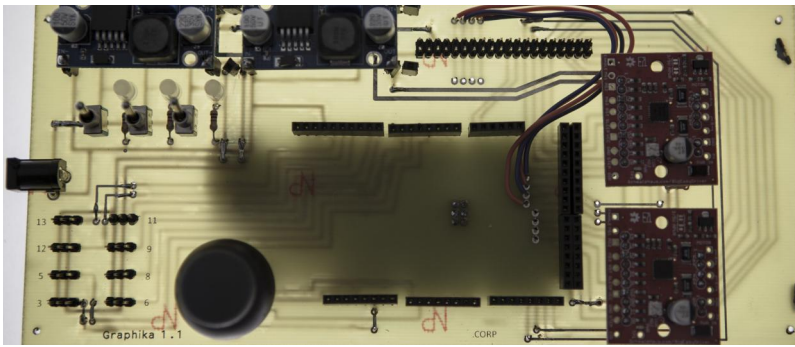
Freehand Tracer fonctionne sur le principe d'un bras mécanique traçant sur une table rétro-éclairée de 180 cm de côté pour une surface d'impression de 160 x 160 cm.

Le bras motorisé (moteurs pas à pas + servomoteur) est muni de 6 têtes d'impression, permettant une utilisation modulaire de la machine. Les têtes d'impression permettent la fixation d'une grande diversité de stylos, crayons ou pinceaux.

Des fluides peuvent également être injectés dans les pinceaux (5 fluides possibles par tête d'impression) grâce à un système de pompes.

Mise à part les rails, la majorité des pièces composant Freehand Tracer ont été imprimées en 3D, et la carte d'assemblage est également faite à la main.

Le logiciel permettant de paramétrer le dessin est développé avec Processing.



ATtiny Multicodeur

Dispositif outil / Electronique / Programme
2016

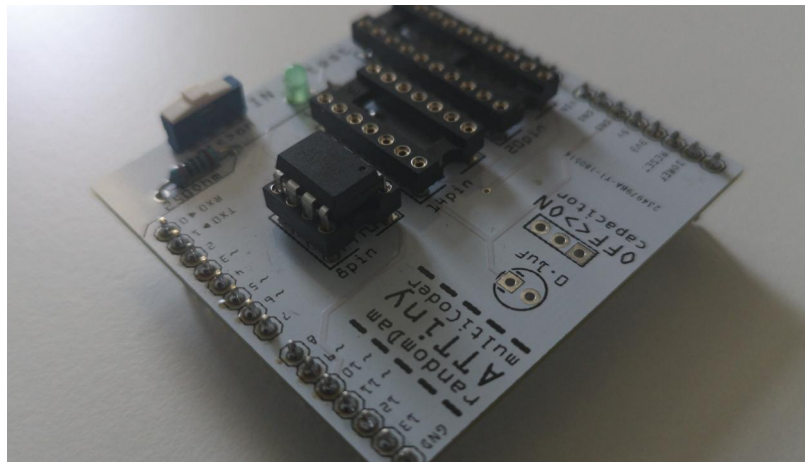
ATtiny multicodeur est un projet de carte opensource.

Il est une alternative intéressante pour exécuter une programme Arduino simple: il est petit, bon marché et relativement facile à utiliser. Il a cependant quelques limitations par rapport au ATmega328P que l'on trouve sur un Arduino Uno.

- Il a moins de broches et vous ne pourrez pas y connecter beaucoup de composants.
- Il y a moins de mémoire flash (4KB ou 8KB à place de 32KB), ce qui implique que votre programme ne peut pas être aussi grand.
- Il y a moins de RAM (256 ou 512 octets/bytes à la place de 2KB), ce qui vous empêche de stoker autant de donnée que sur un Uno.

→ Premier prototype de la carte électronique

↓ Carte électronique avec un ATtiny85



Installation interactive
2017

Nord Ouest est une interface basée sur les principes d'un jeu vidéo multijoueur pour produire une pièce musicale collective. Les joueurs sont invités à télécharger sur leur smartphone l'application Nord-Ouest qui sert de manette au dispositif interactif. Cette création originale de .CORP (développement) et Gaël Moissonnier (musique) est présentée pour la première fois dans le cadre de la Nuit des musées 2016, produite pour la Cité du design de Saint-Etienne.

↓ Performance au
musée de la mine



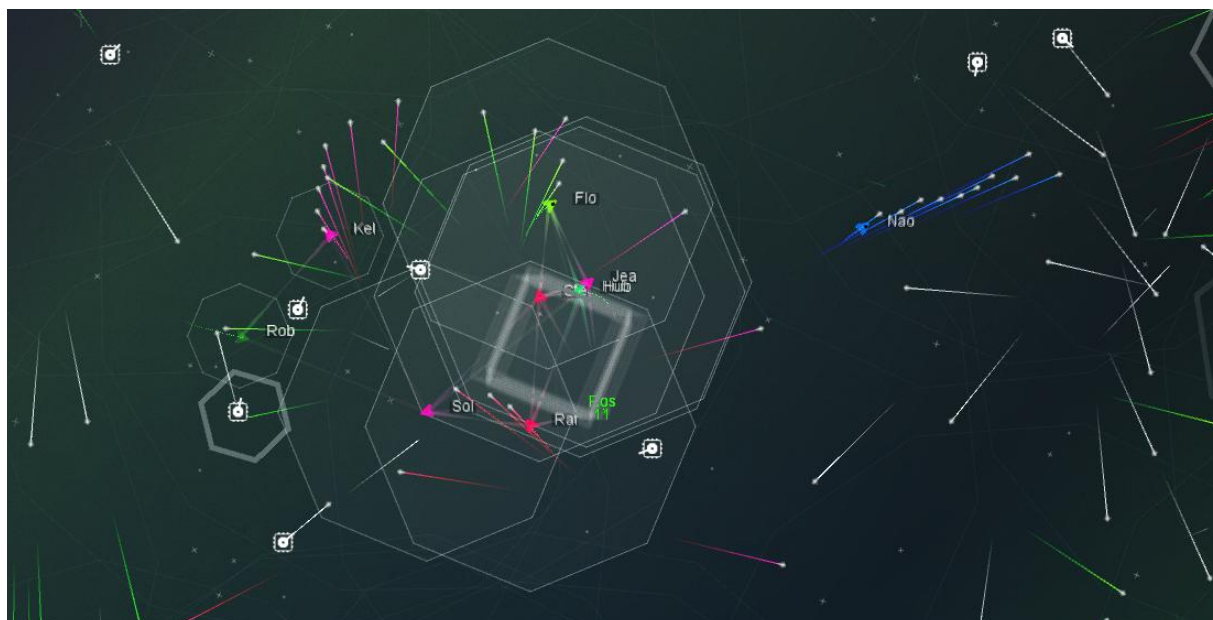


← Performance au musée de la mine



← Détail du jeu vidéo

↓ Capture d'écran



Coco's souvenirs

Publication générative
2019

BLOB, davantage qu'un point d'étape sur les travaux en cours, donne l'occasion aux chercheuses et chercheurs de l'UR numérique de se retrouver autour d'une thématique ou d'un sujet d'intérêt dans une publication qui s' imagine résolument évolutive. L'intention n'est pas de produire une somme de textes ou de visuels "de référence" mais plutôt de montrer une pensée en développement, un moment dans un parcours de recherche.

↓ Page avec
l'occurrence
"Plane"

↘ Page avec
l'occurrence "Men"



↓ Page avec
l'occurrence "Men"



↓ Page avec
l'occurrence "Dog"



Peter Campus App.

Application / base de données
2019

Ce titre d'anarchivage destiné à faire mieux connaître l'œuvre d'un célèbre pionnier de l'art vidéo a été publié à l'occasion de l'exposition peter campus video ergo sum au Jeu de Paume (février-mai 2017).

La publication consiste en trois volumes dont la conception et la réalisation graphique ont été confié à Jérémie Nuel : le catalogue de l'exposition avec la collaboration du Jeu de Paume, la reproduction du premier catalogue de l'exposition de Peter Campus à l'Everson Museum of Art de Syracuse (NY) depuis longtemps épuisé, et un volume anarchivage 7 qui rassemble les écrits de l'artiste et comprend onze mots-clefs destinés à l'application de réalité augmentée. Ces notions donnent accès à des regroupements thématiques d'œuvres de toutes périodes et catégories qui s'affichent dans une matrice 3D qui a été conçu par Peter Campus en collaboration avec David-Olivier Lartigaud.

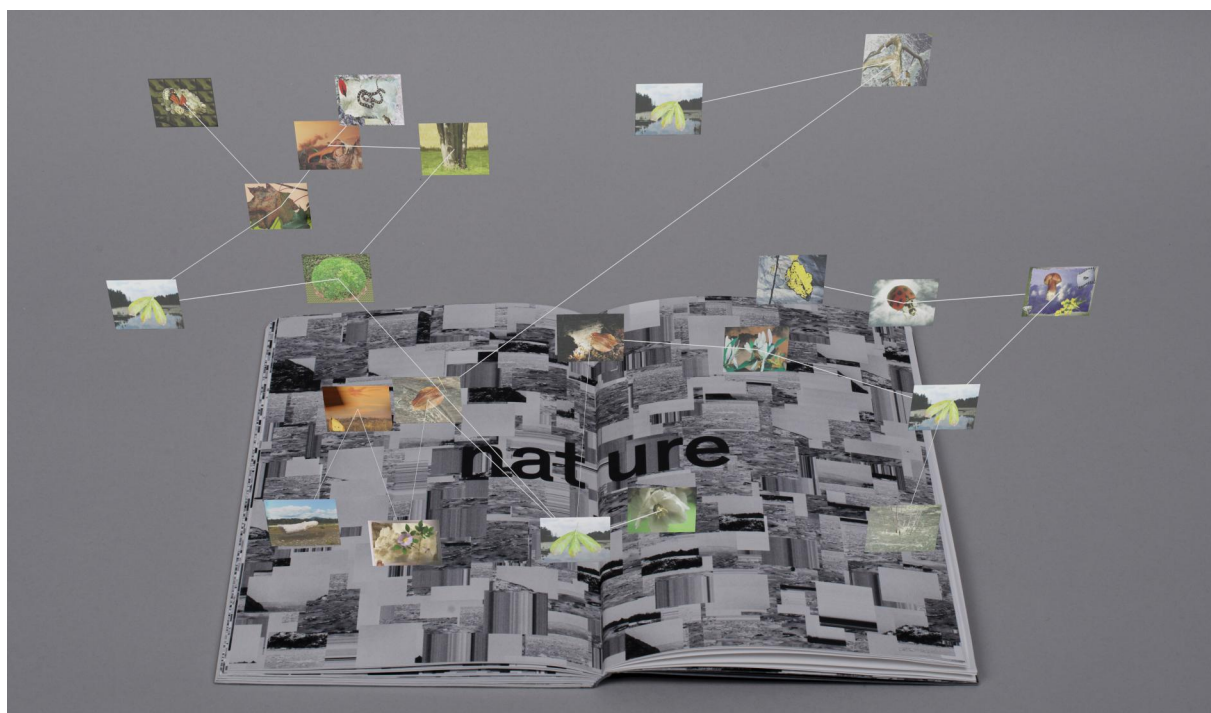
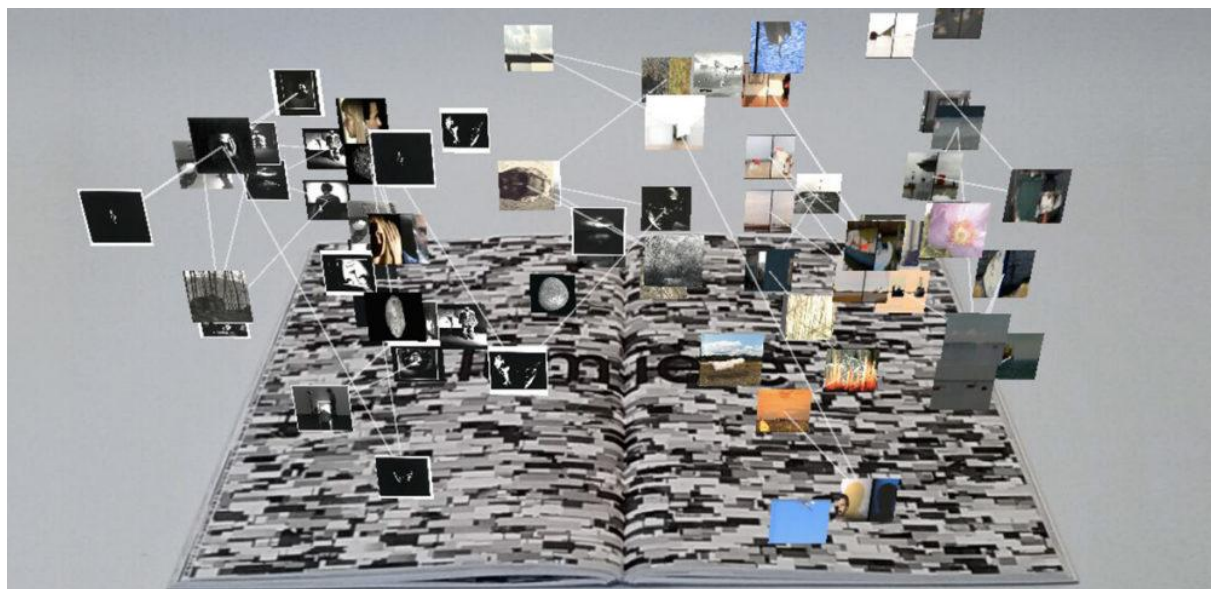
Damien Bais a créé une base de données "dynamique" avec un programme spécialement écrit pour ce projet. Cela signifie qu'il est toujours possible de la modifier, d'ajouter des données ou d'en corriger. C'est un aspect important pour un artiste qui continue à produire des œuvres. L'application de réalité augmentée et la base de données sont consultables sur tablettes et smartphones Android ou iOS. Le téléchargement de l'application sur Google Play ou App Store est gratuit.

→ \ Capture
d'écran de
l'application

À ce jour la base de données comprend : 362 œuvres accessibles directement par décades et par catégories à partir de :
270 images : photographies réalisées par l'artiste, dessins par ordinateur, photographies des installations, dessins préparatoires et
171 vidéos : vidéos réalisées par l'artiste des années 1970 à aujourd'hui ou des vidéos documentant les installations.



← Motif de reconnaissance de positionnement pour l'application



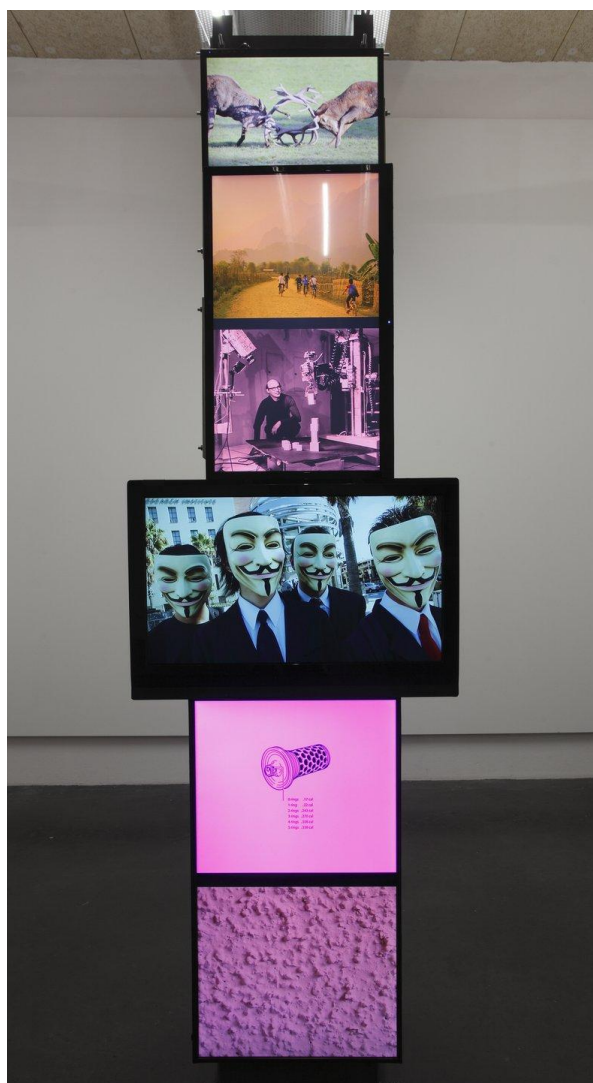
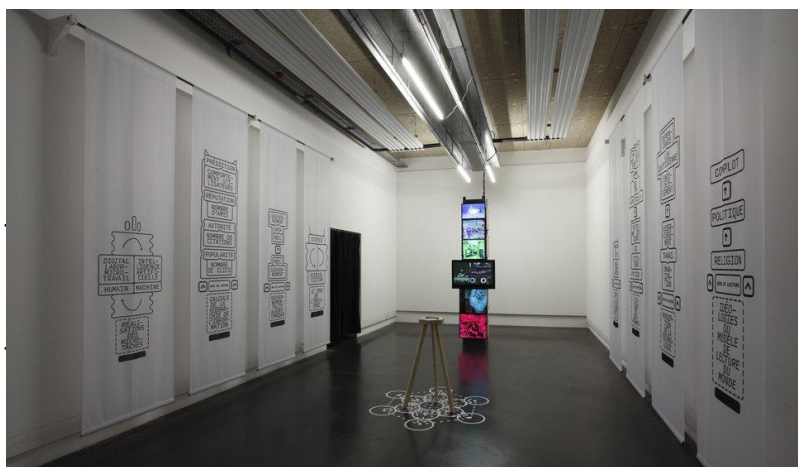
Une relation intime et tout à fait spéciale...

Installation interactive
2019

L'intelligence artificielle nous projette-t-elle dans un futur plus ou moins lointain ou est-elle toute proche de nous : dans notre environnement, dans notre poche, dans notre comportement ? Est-elle familière ou étrangère ? Rassurante ou menaçante ? Naturelle ou culturelle ? L'intelligence artificielle existe-t-elle ? Est-elle une nouvelle mythologie contemporaine, ni tout à fait divine, ni purement mécanique, que l'homme se serait construit ? D'autre part, pour préciser la notion de totémisme, et selon Wikipedia, ce nom provient du mot "totem", lui-même emprunté à "ototeman", un terme de la langue Ojibwé parlée autour des grands lacs de l'Amérique du Nord par un groupe Amérindien. Le mot totémisme revient à un Anglais, John Long, qui l'utilisa en 1791 pour désigner un esprit bienveillant qui protège les hommes. Un groupe d'hommes est ainsi sous la protection d'un totem. Le totémisme est un concept anthropologique qui désigne un mode d'organisation social et religieux, clanique ou tribal, fondé sur le principe du totem. Le point le plus important est la définition du totem comme apparemment ou amitié entre une espèce naturelle et un groupe humain, en même temps que la définition du totémisme comme organisation sociale du clan. James George Frazer, l'auteur du Rameau d'Or en propose la définition suivante en 1887 : Un totem est une classe d'objets matériels que le sauvage considère avec un respect superstitieux et environnemental, croyant qu'il existe entre lui et chacun des membres de la classe une relation intime et tout à fait spéciale.

→ → Salle
d'exposition du
totem au "Bel
ordinaire"

Exposition :
Bel Ordinaire - Pau - Décembre 2018 à Février 2019



↑ Compositions
générés par le
totem

OSC-Env

Logiciel / Environnement réseau
2019

Osc-Env est un logiciel destiné à capturer les événements réseaux. Pour les installations numériques complexes, il est commun de faire communiquer plusieurs machines entre elles. OSC-Env capture tous les messages dans le temps ainsi que leurs contenues, leurs natures et leurs destinations. À la manière d'un enregistreur audio, il peut ensuite retranscrire et rediffuser un environnement réseau afin de le recréer à un "instant T". Ce développement, "en bêta", pourrait être publié en open source.

↗→\ Captures
d'écran de
l'application

```
FrameRate : 61
Port : 8000
Vitesse input : 2634 m/second
```

[illegible]

```
FrameRate : 60
Port : 8000
Vitesse input : 2718 m/second
```

STARTRECORD

RECT in PAUSE

F

FILE
.THT
TEST1.THT
TEST2.THT
TEST3.THT
TEST4.THT
TEST5.THT
TEST6.THT
_CHIME_ONLY.THT
_CURVE_ONLY.THT
_RELATION.THT
_RELATION_WITHCHIME.THT
_TV_ONLY.THT
_INFERRATION2.THT

```
session/_RELATION.txt | play | loop : true
buffer : 197 | count : 5372
vit : 172 m/second
```

```
session/_CURVE_only.txt | play | loop : true
buffer : 197 | count : 3193
vit : 134 m/second
```

```
session/_TV_only.txt | play | loop : true
buffer : 199 | count : 753
vit : 38 m/second
```

```
session/test3.txt | play | loop : true
buffer : 194 | count : 10395
vit : 510 m/second
```

```
session/test.txt | play | loop : true
buffer : 197 | count : 2949
vit : 151 m/second
```

```
session/test6.txt | play | loop : true
buffer : 185 | count : 29705
vit : 1585 m/second
```

```
FrameRate : 60
Port : 8000
Vitesse input : 199 m/second
```

19652	/192.168.1.1.96:46999	/bg/TV5.1	int > 1
19652	/192.168.1.1.96:46999	/f/req/TV5.f	f float > 15.375122
19652	/192.168.1.1.96:46999	/f/req/TV3.f	f float > 1.901272
19652	/192.168.1.1.96:46999	/c/TV1.c	int > 1
19652	/192.168.1.1.96:46999	/bg/TV0.1	int > 1
19652	/192.168.1.1.96:46999	/f/req/TV5.f	f float > 15.290283
19634	/192.168.1.1.96:46999	/bg/TV0.1	int > 1
19634	/192.168.1.1.96:46999	/f/req/TV3.f	f float > 21.4375
19634	/192.168.1.1.96:46999	/bg/TV1.1	int > 1
19618	/192.168.1.1.96:46999	/c/TV1.c	int > 1
19618	/192.168.1.1.96:46999	/c/TV1.c	int > 1
19618	/192.168.1.1.96:46999	/f/req/TV4.f	f float > 12.539185
19618	/192.168.1.1.96:46999	/f/req/TV2.f	f float > 6.146484
19618	/192.168.1.1.96:46999	/f/req/TV0.f	f float > 1.5307007
19617	/192.168.1.1.96:46999	/c/TV1.c	int > 1
19617	/192.168.1.1.96:46999	/c/TV0.1	int > 1
19617	/192.168.1.1.96:46999	/f/req/TV2.f	f float > 6.092651
19601	/192.168.1.1.96:46999	/c/TV2.c	int > 1
19601	/192.168.1.1.96:46999	/c/TV1.c	int > 1
19601	/192.168.1.1.96:46999	/c/TV1.1	int > 1
19584	/192.168.1.1.96:46999	/bg/TV1.1	int > 1
19584	/192.168.1.1.96:46999	/bg/TV0.1	int > 1
19584	/192.168.1.1.96:46999	/f/req/TV5.f	f float > 12.465398
19584	/192.168.1.1.96:46999	/f/req/TV0.f	f float > 1.4837036
19584	/192.168.1.1.96:46999	/c/TV0.1	int > 1
19568	/192.168.1.1.96:46999	/f/req/TV5.f	f float > 15.205745
19568	/192.168.1.1.96:46999	/f/req/TV0.f	f float > 12.363905

WMotor

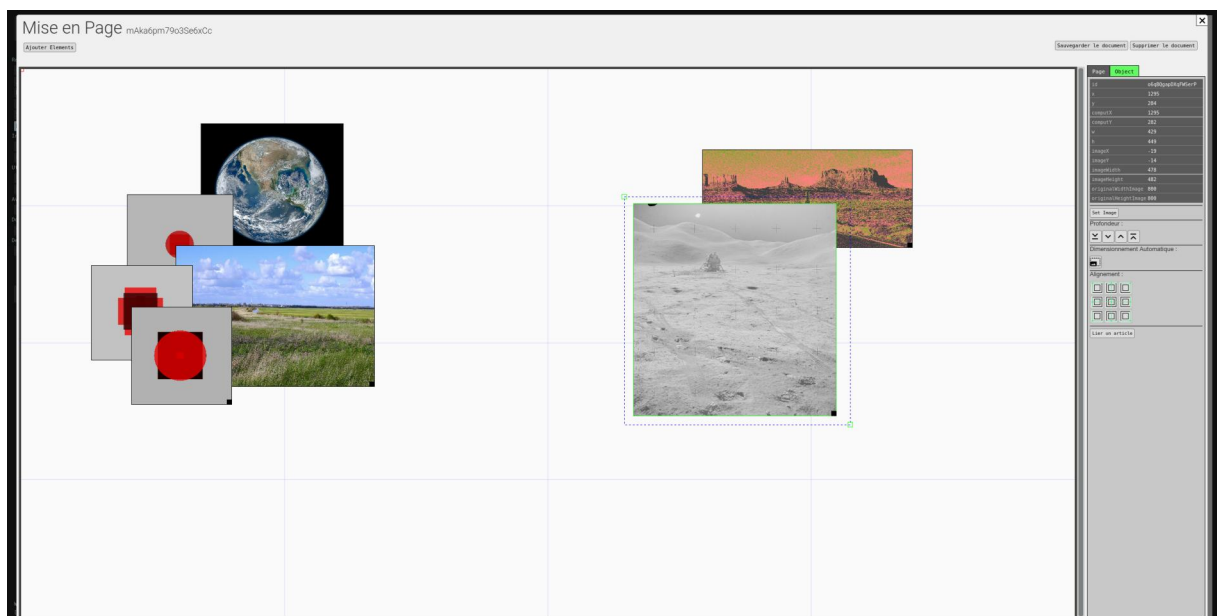
Site Internet Multiple / Framework Web
2019 > 2021

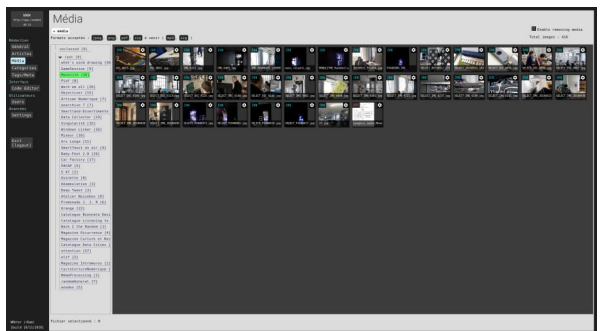
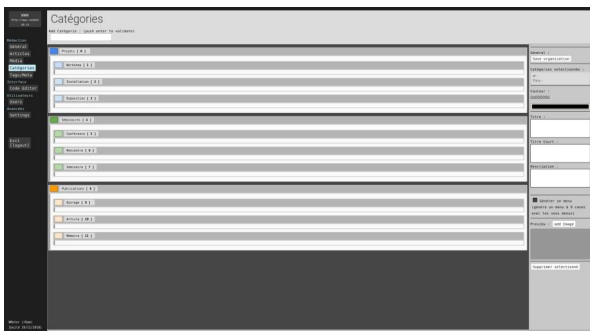
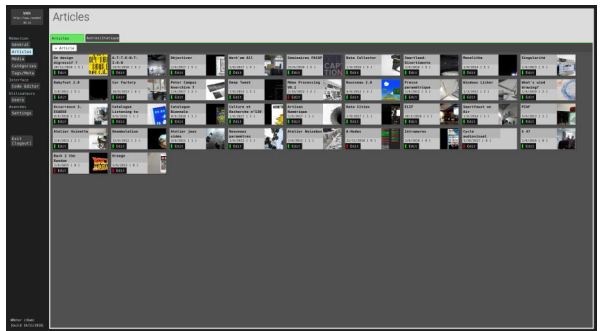
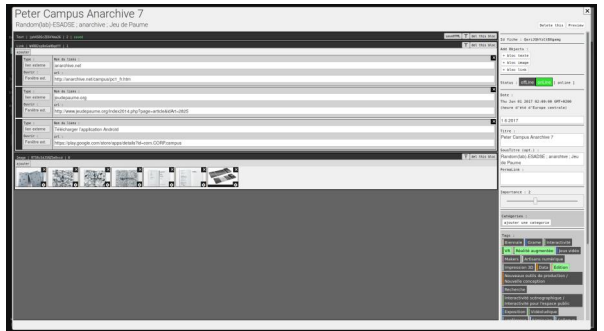
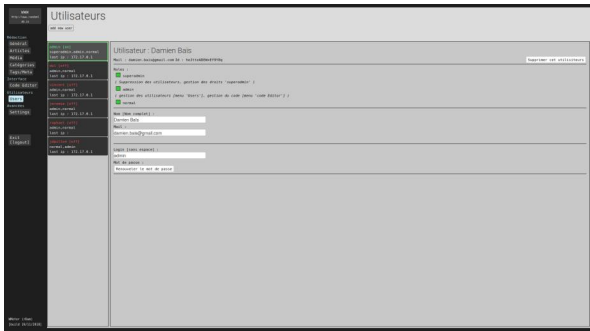
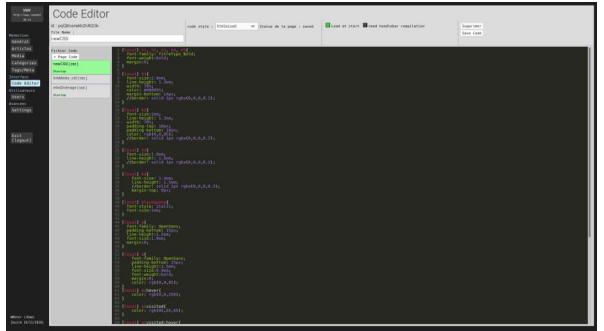
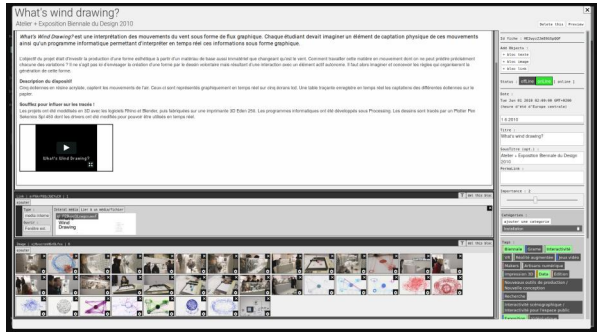
Le site du Random(lab) est construit à partir d'un backoffice maison et "from scratch". Il apporte donc le maximum de souplesse et offre les avantages de se libérer de wordpress ou d'autres frameworks.

Le projet WMotor comme "Web Motor" est un projet libre permettant aux étudiants de gérer et construire des sites web complexes sans se soucier de la partie base de données, puisque celle-ci est entièrement autonome du front-office autrement dire de la partie visible sur internet. Plusieurs modules sont développés par an, ajoutant continuellement de nouvelles fonctionnalités.

↗↘ Captures
d'écran de
l'application

↓ Capture d'écran
du module de
mise en page





Ultimate Cavefighter in outer space

Installation interactive / jeux vidéo multijoueurs
2019

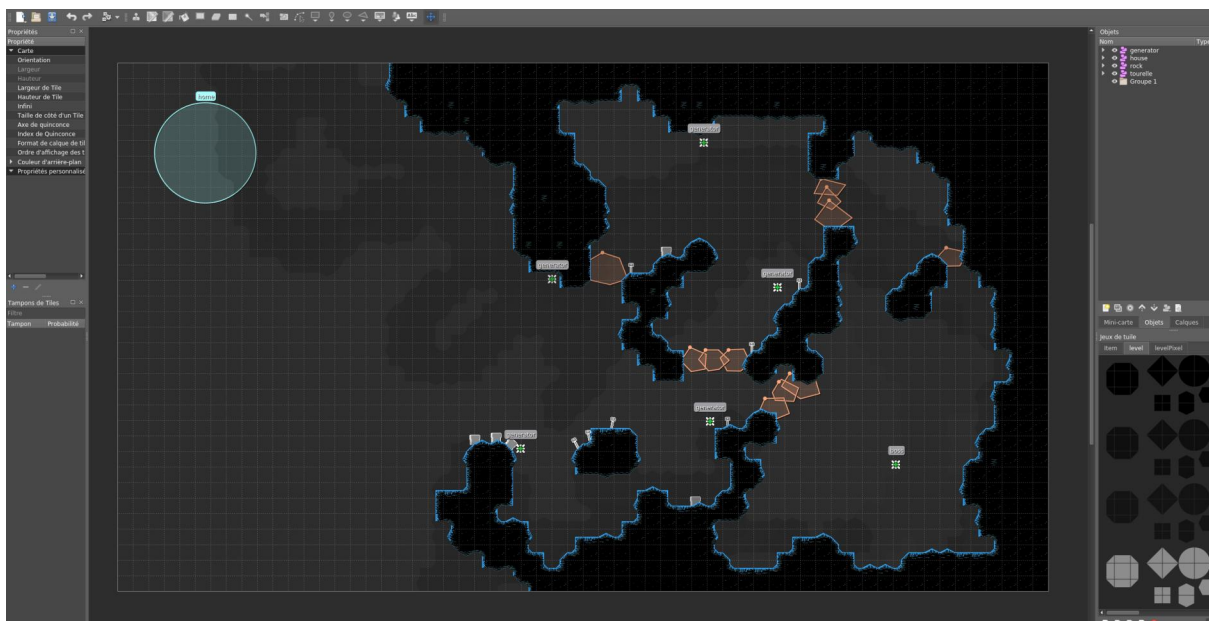
L'assaut final à commencer, l'ultime bataille. Dans les retranchements d'une planète obscure, vous, joueurs, allez devoir atomiser des drones, arpenter des passages, pour enfin détruire le mal ultime. Afin de réussir votre quête vous allez développer des stratégies collectives pour vaincre les adversaires.

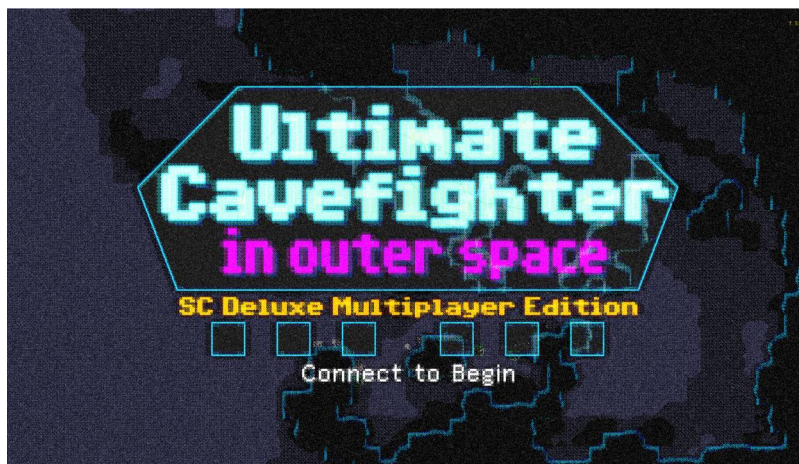
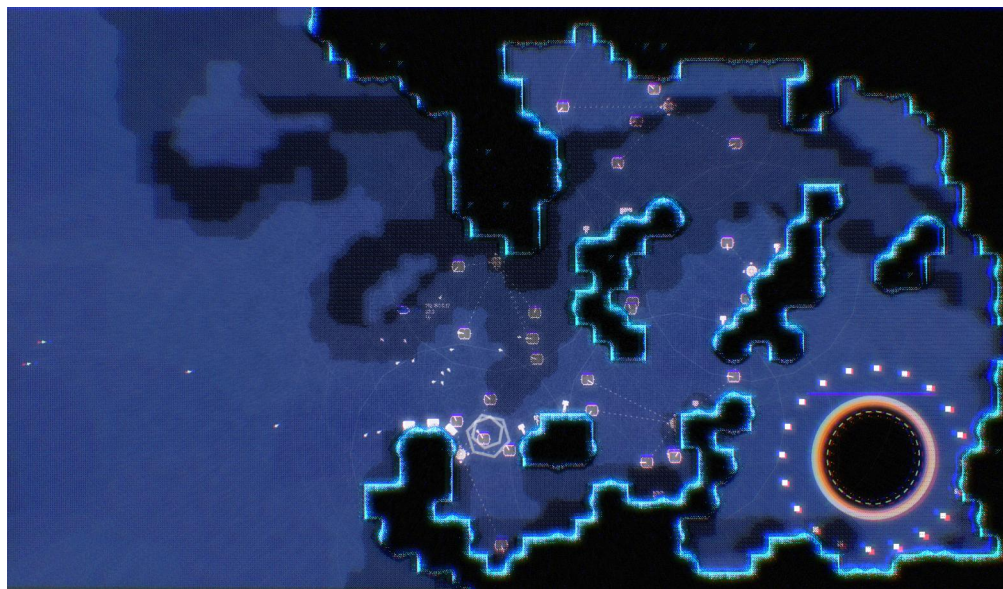
Cave fighter est inspiré des jeux vidéo des années 80, des "fixed shooter", jeux de tirs à position fixe, dont les plus célèbres sont, "Spacewar!"¹, "Space Invaders"², "Communists Mutants from Space"³ ou encore "Centipede's"⁴.

Pour entrer dans la partie vous devez vous connecter au réseau "Cavefighter" et télécharger l'application cavefighter (Android seulement) qui sert de manette de jeu.

1. "Spacewar!", Steve Russell, 1962
2. "Space Invaders", Taito, 1978, Gen 2-3
3. "Communists Mutants from Space", Starpath, 1982, Gen 2
4. "Centipede's", Atari, Inc., 1981, Gen 2-3

↓ Éditeur de
niveau du jeu





↑ — Captures
d'écran de
l'application

↓ Exposition du
jeu vidéo à Ars
Electronica

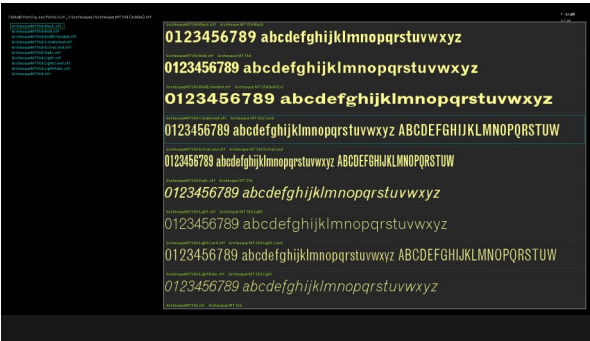
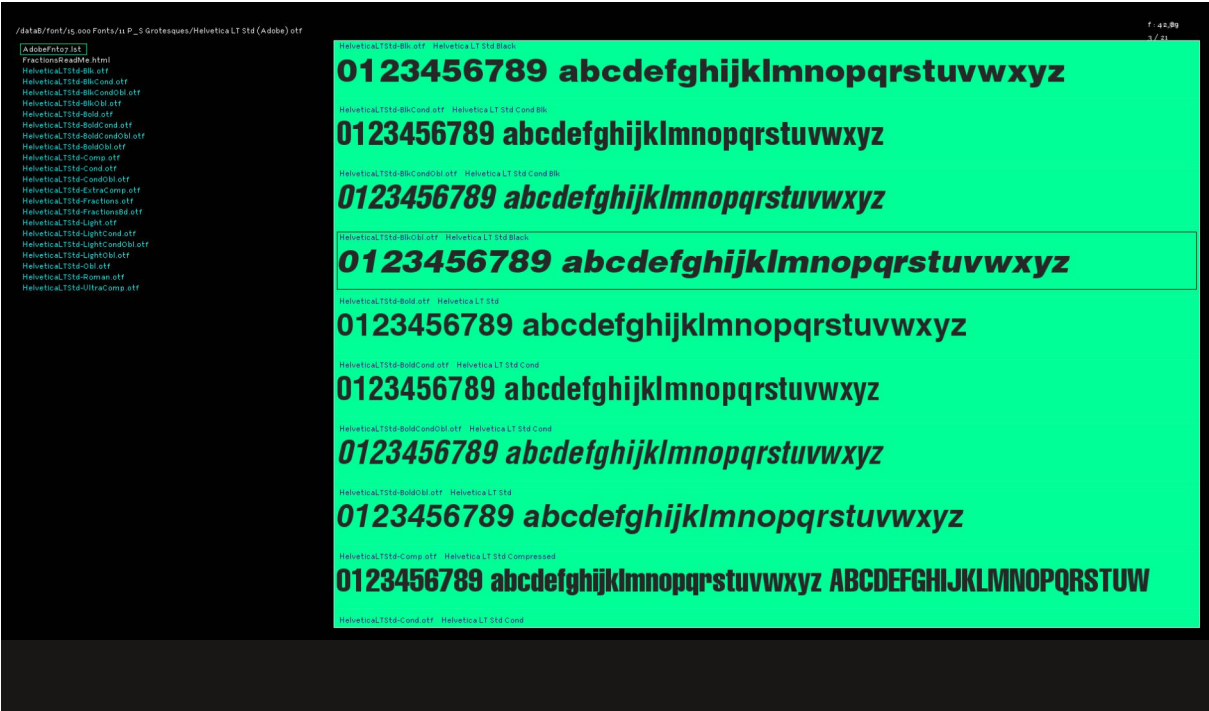
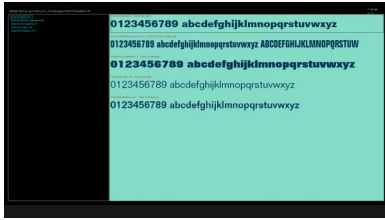
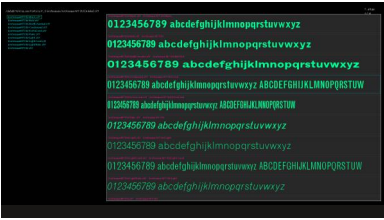
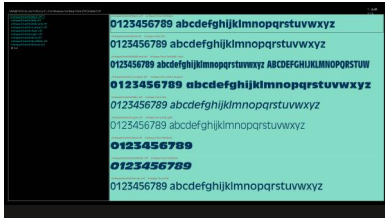
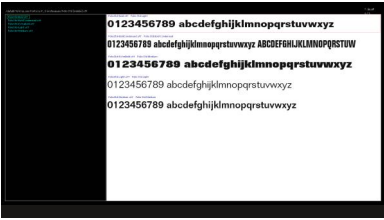
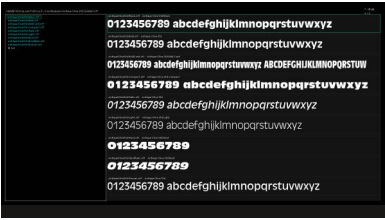
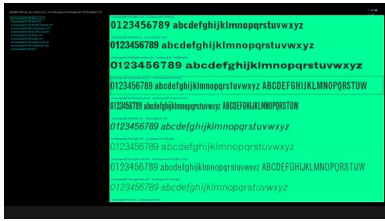
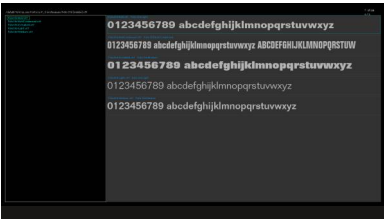
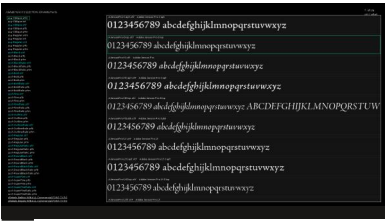


TypeProcess

Logiciel openSource / Design Graphique
2020

Type Process est un gestionnaire de typographie "from scratch", il fait partie des outils simples pouvant être codée / modifiée par des étudiants. De plus, il est rapide, très léger et supporte pleinement l'accélération matérielle. À terme, son code source doit être libre et facilement accessible.

↗→↘ Capture
d'écran de
l'application

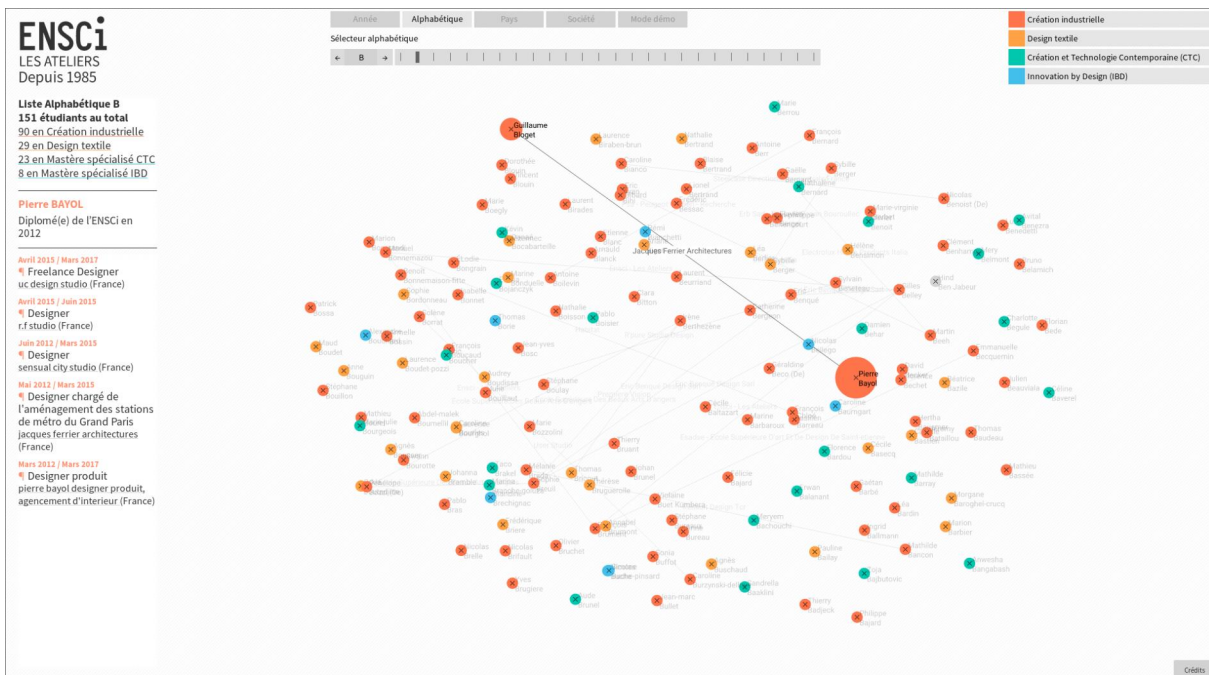
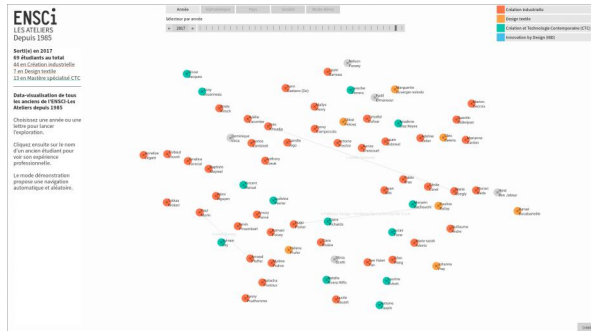
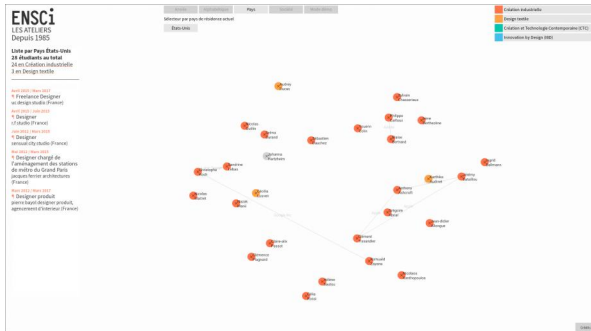
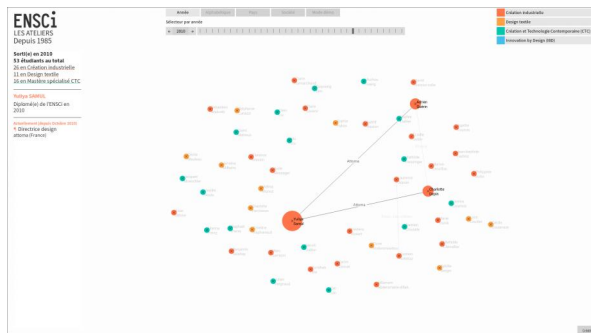
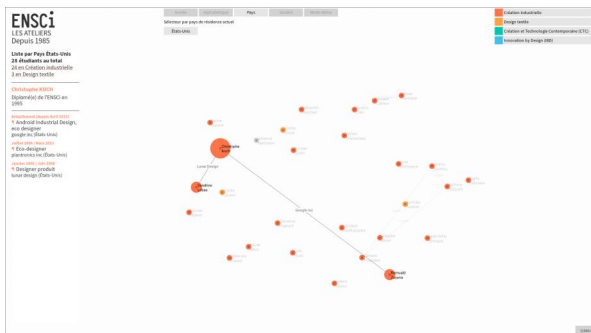


Ensci / Réseau des anciens

Design Graphique / Site internet / Data visualisation
2018

Data visualisation interactive des anciens étudiants de l'ENSCi – Les Ateliers. Un second réseau en arrière-plan introduit les expériences professionnelles en commun. Cette visualisation est une exploitation de la base de données collectée par l'école.

<https://anciens.ensci.com/>



↑ Capture d'écran de l'application

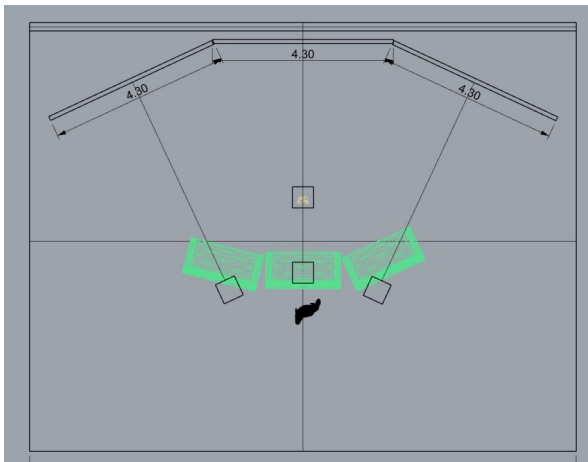
Indie Game Lounge

Design Graphique / Site internet / Data visualisation
2020

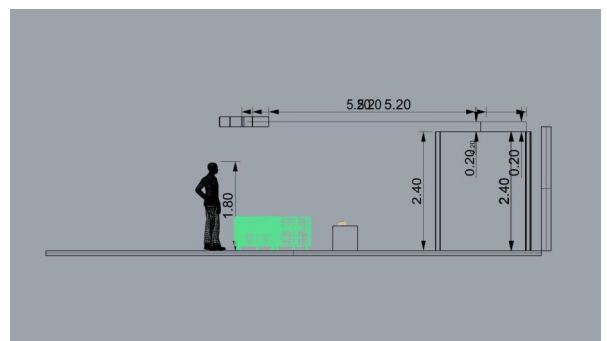
Apparu au début des années 2000 du fait de la simplification des techniques de programmation informatique, le jeu vidéo "indépendant" adonné naissance à une scène de création extrêmement vive où des jeux produits par une minuscule équipe côtoient des créations de studio en marge de l'industrie. Plus libres dans leur approche que les jeux "grand public", ces productions abordent des thèmes inhabituels (ArtSqool, Papers, Please...), osent des parti-pris esthétiques radicaux (140, Limbo, Protheus...) ou des mécaniques de jeu surprenantes (Defcon...). Véritables œuvres d'art pour certains (Everything, Inside...), brillants exercices de « level design » (Fez) pour d'autre, ces jeux touchent un public moins vaste que ceux de l'industrie mais gagnent parfois le statut de jeu culte (Braid...) ou de jeu viral (Hotline Miami...), formes de reconnaissance souvent plus attendues que le succès financier.

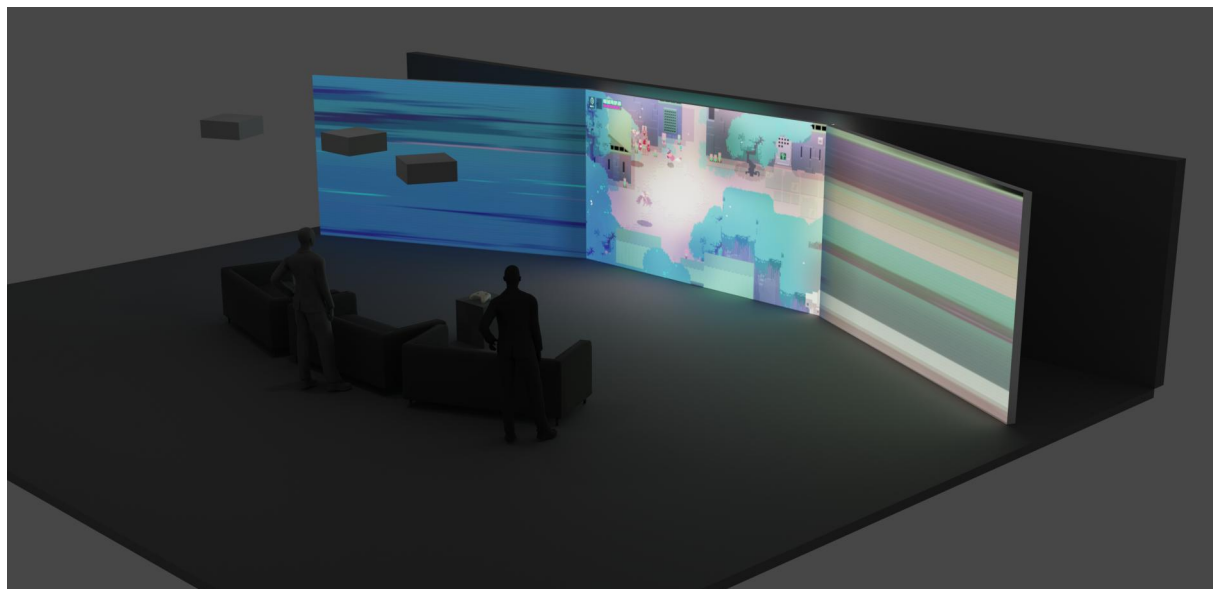
Le Indie Game Corner/lounge/room proposé ici par le Random(lab) souhaite rendre hommage à ces créations en proposant une expérience de jeu immersive, en rupture avec le cadre domestique, où l'image est augmentée d'effets divers afin de plonger les joueurs dans le monde éclectique et fascinant des jeux indépendants.

→ Rendu 3D de
l'espace "Indie
Game Lounge"



✓ ↓ Plans de
l'espace "Indie
Game Lounge"





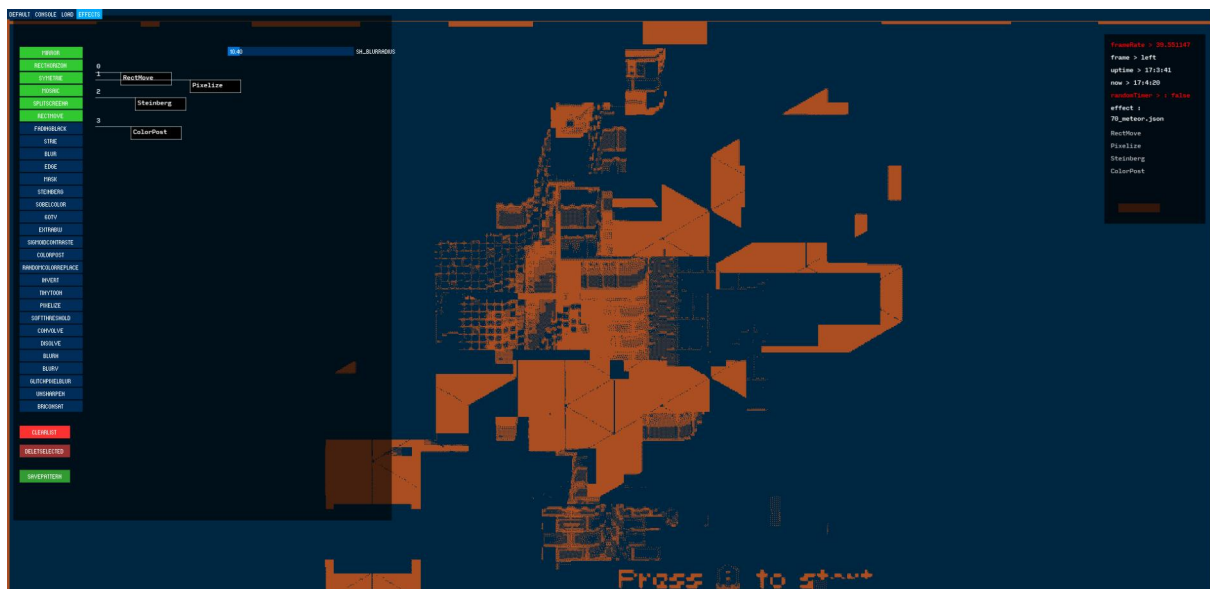
↑ Montage de
l'espace "Indie
Game Lounge"

Indie Game Lounge (Post effect)

Installation interactive / Dispositif temps réel
2020

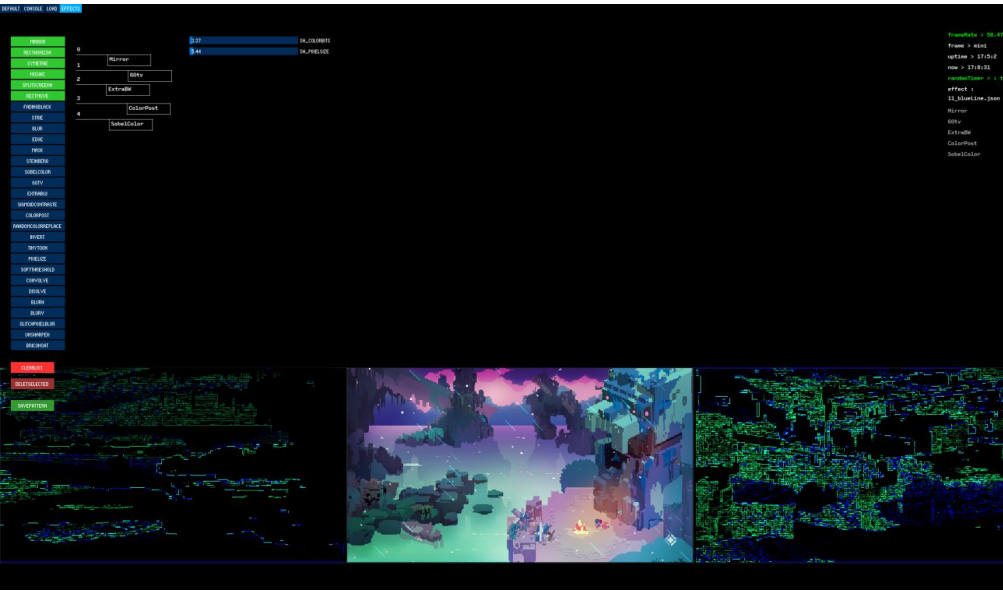
"Post effect" est une partie de l'exposition Indies Game Lounge, ce programme informatique traite en temps réel l'image principale du jeu vidéo affiché pour produire deux autres images sur des écrans parallèles. Les effets sont générés et combinés pseudos aléatoirement au sein d'une game d'une quarantaine de patterns. 180 combinaisons ont été pré-combinées grâce à un système d'enregistrements. Ces "effets graphiques" font référence souvent à d'autres rendus de jeux vidéo ou de consoles particulières (Amiga, Game boy, Amstrad, Vectrex, ...) mais aussi à la scène "dém".

↓ Détail de
l'interface de
composition du
logiciel d'effet





↑ Divers résultats d'effets



→ Détail de l'interface de composition du logiciel d'effet

↓ Divers résultats d'effets



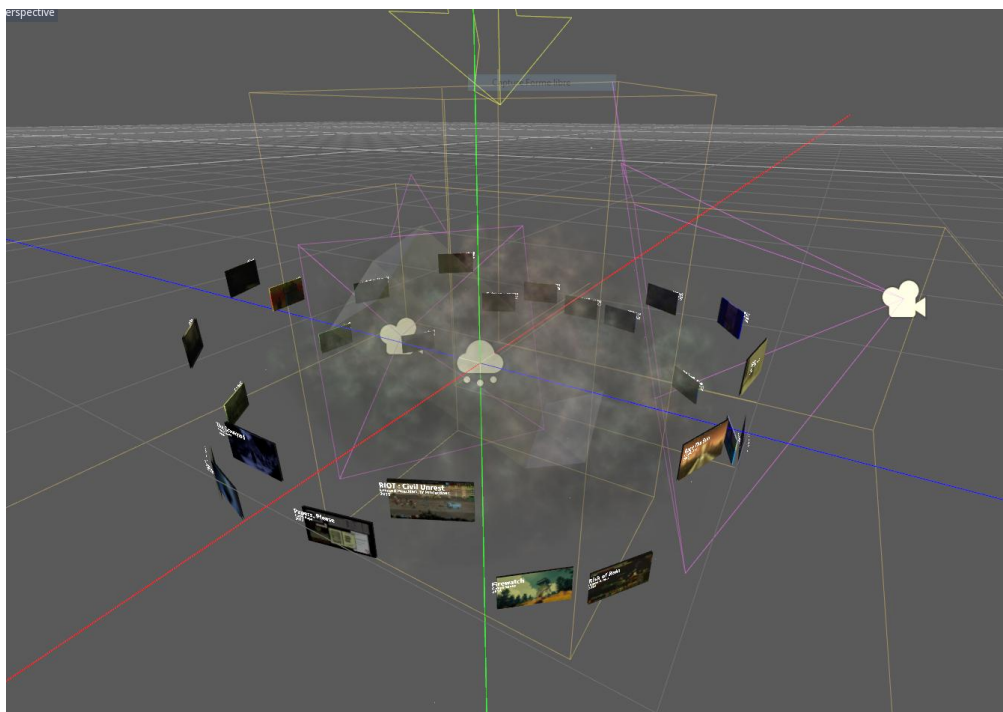
Indie

Game Lounge (Game select)

—
Installation interactive / Dispositif temps réel
2020

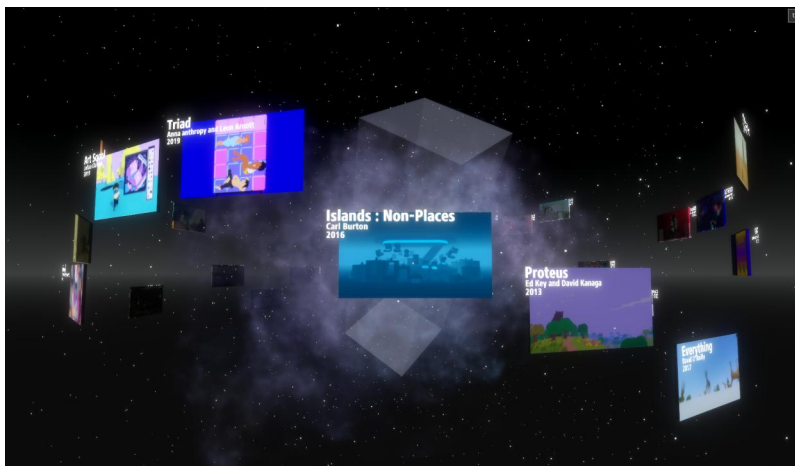
—
"Game select" est une interface modulaire de lancement et de contrôle d'activité de jeux vidéo. Elle s'appuie sur un code JSON, qui permet d'ajouter ou de supprimer des entrées de programme. "Game select" est lui-même codé avec un moteur de jeu vidéo (Godot Engine).

→ Capture d'écran de la scène sous "Godot"

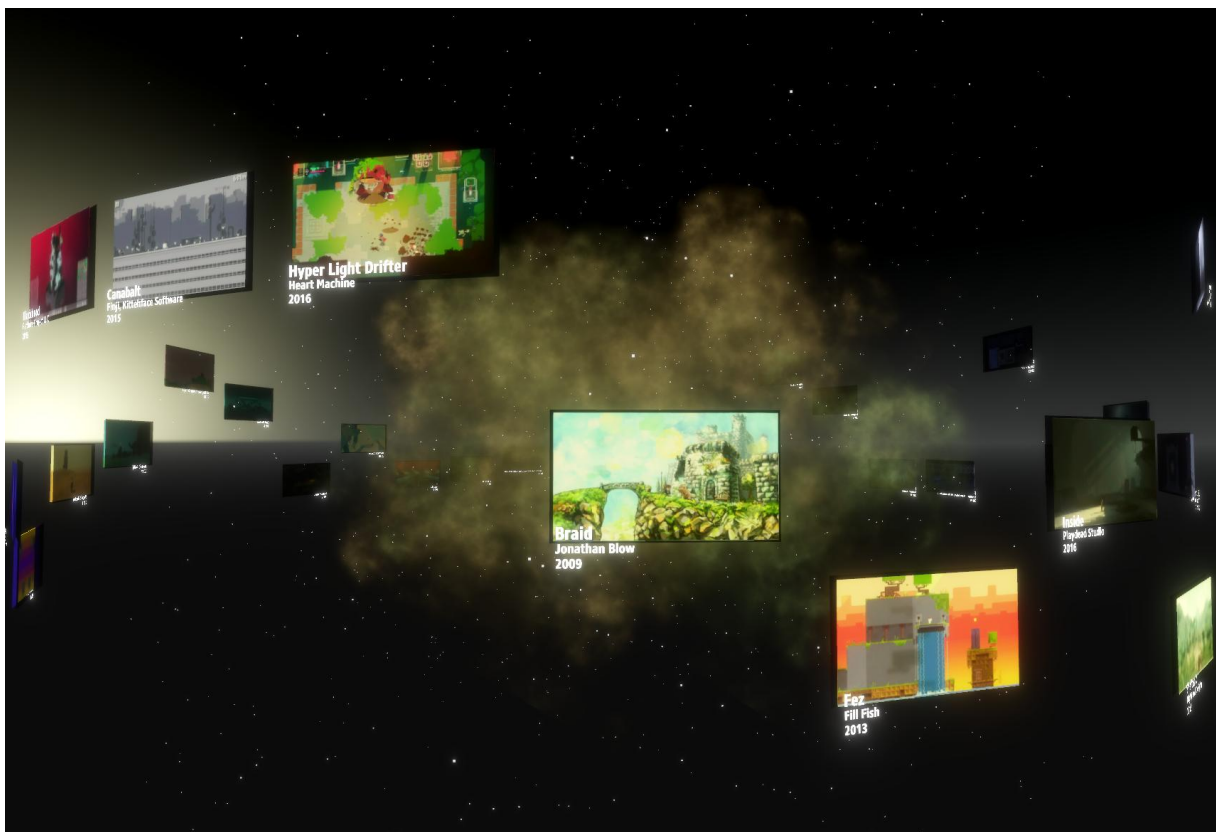




←/ Différentes
scène finales de
l'interface de
selection de jeux



↓ Scène finale de
l'interface de
selection de jeux



Objects of thinks

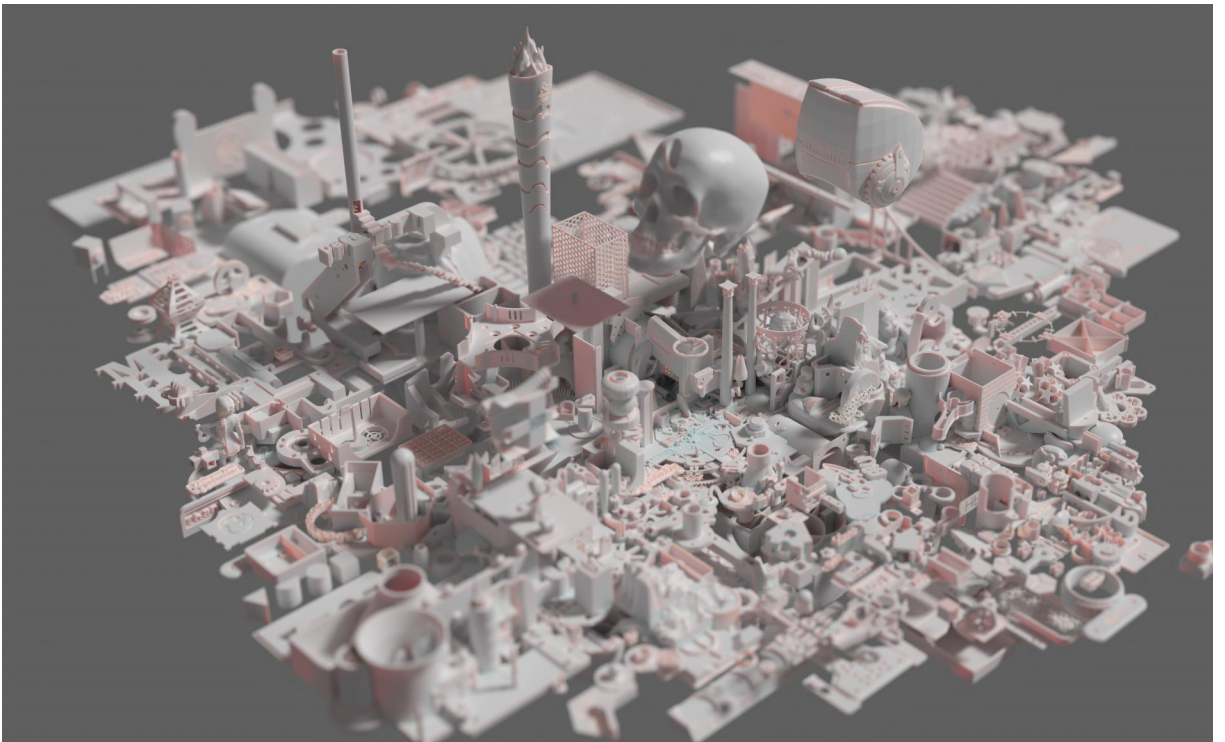
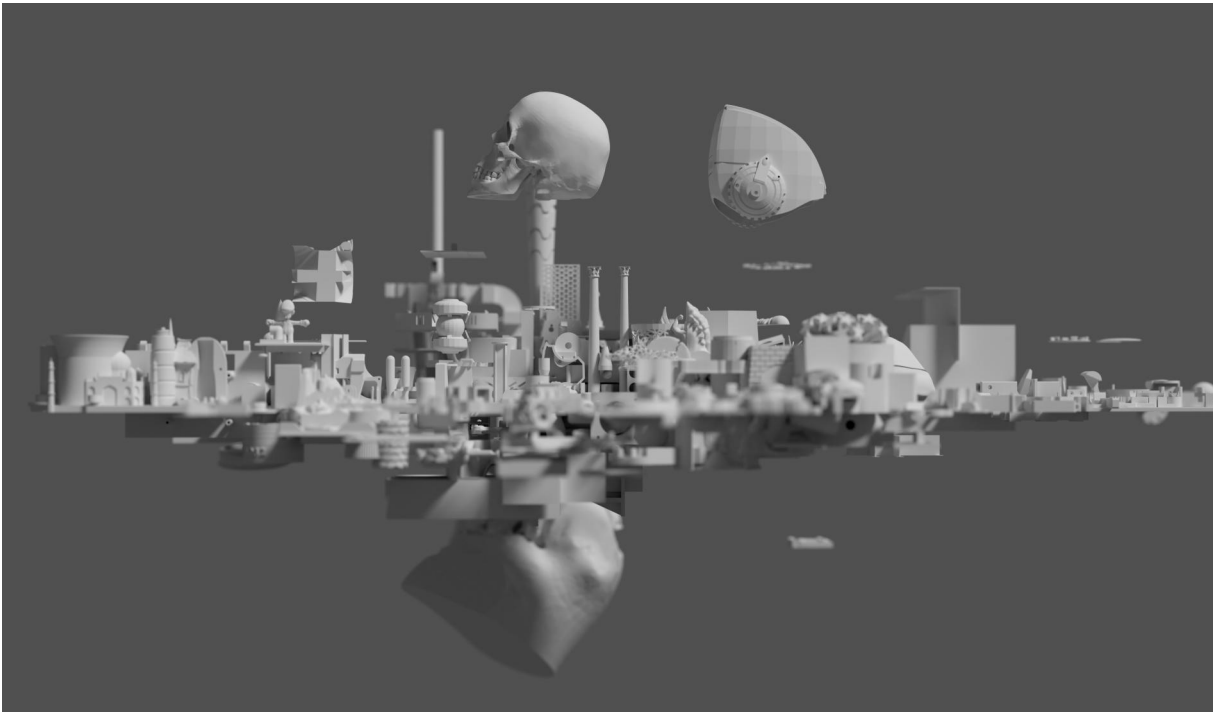
Rendu 3D / automate de téléchargement / Recherche formelle
2021

"Object of thinks" collecte, télécharge de manière automatique des objets sur diverses plates-formes de contribution de fichier 3D. Cette recherche se souhaite témoin d'une activité de prolifération des modèles / plans stockés sur des serveurs informatiques. Les résultats sont ensuite dispersés de manière cohérente sur des "scènes 3D" afin de pouvoir effectuer des rendus. Les images produites sont les restes d'activités diverses de construction d'objet.

— \ Test de
collection d'objet
avec différents
points de vue et
matériaux sous
Blender

↓ Détail d'un
rendu





Cartographie des outils numériques en Art et en Design

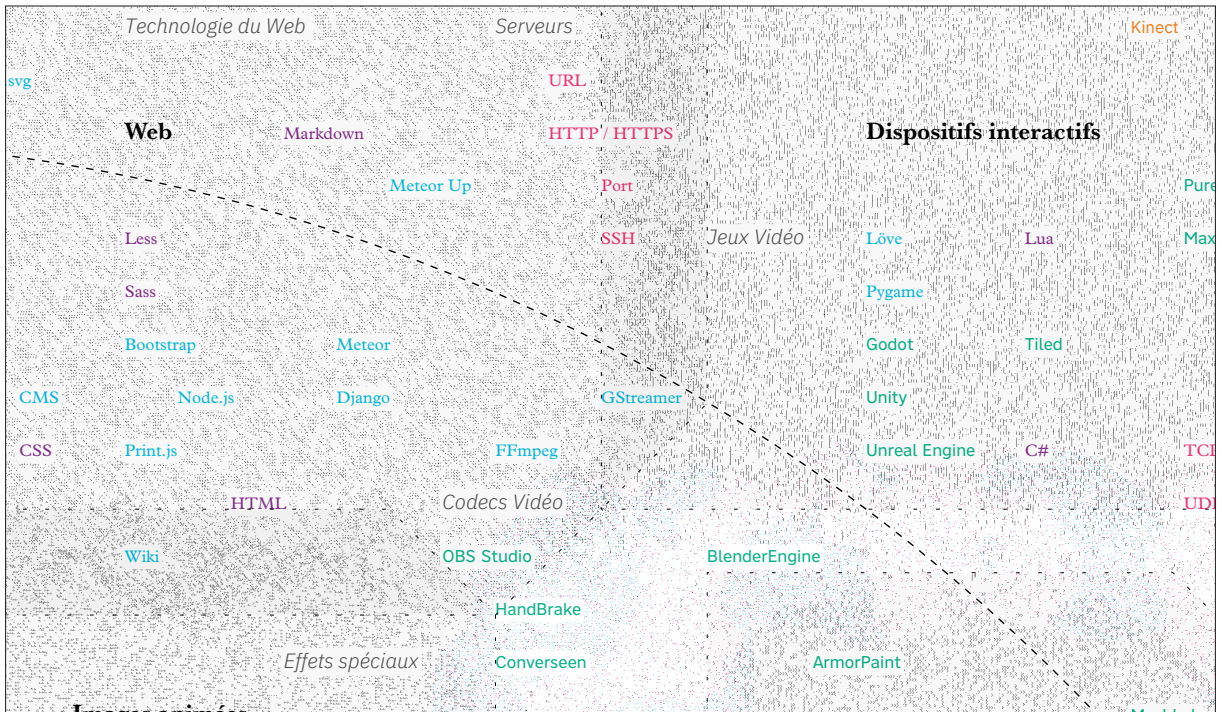
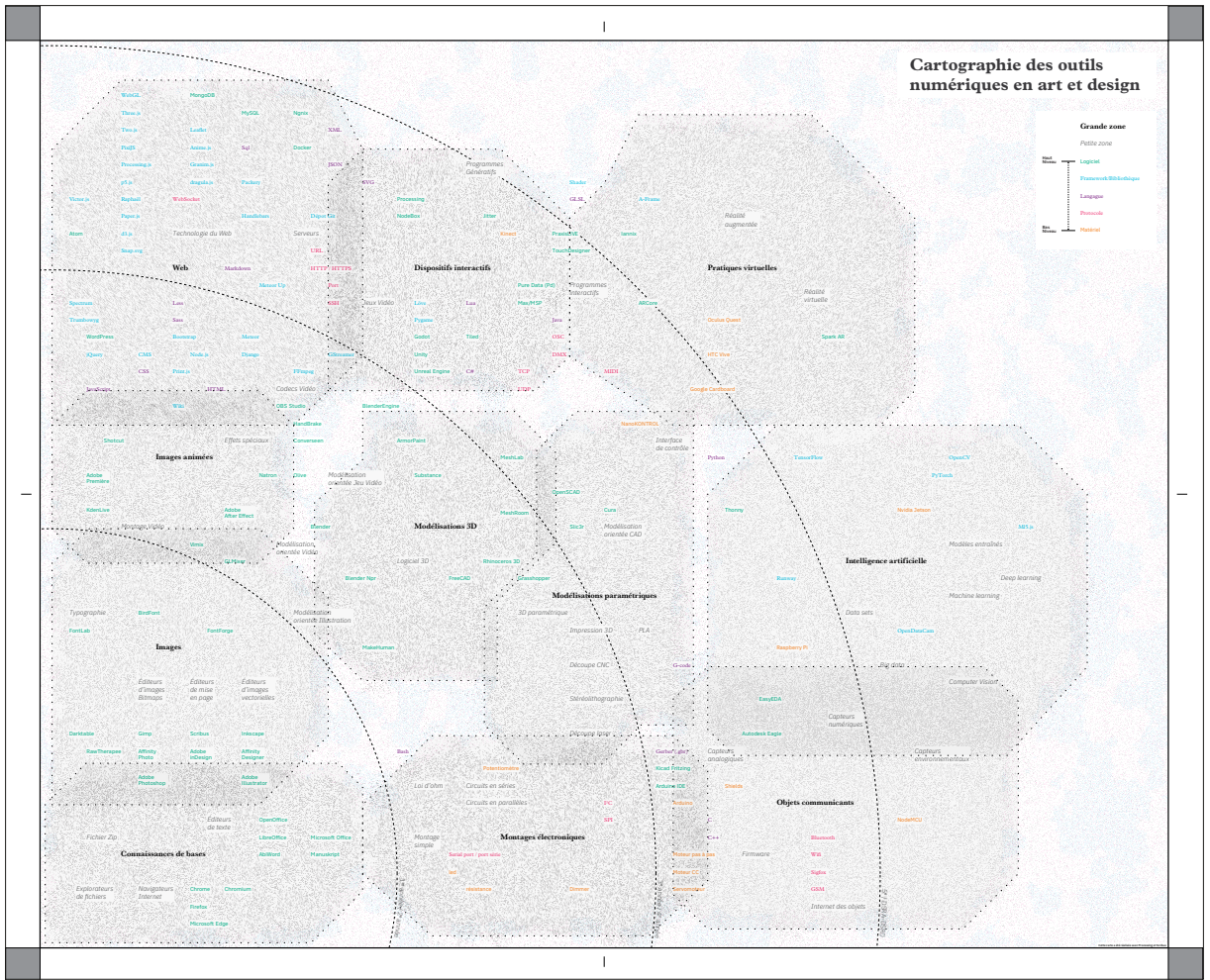
Outils numériques / Enseignement
2021

Cette liste de logiciels, applications et langages est représentative d'outils numériques utilisés en art et design. Elle n'est évidemment pas exhaustive. C'est une ressource pour comprendre globalement l'écosystème numérique mobilisable pour la création actuelle.

Certaines des définitions qui suivent sont inspirées ou résumées du contenu issu de l'encyclopédie en ligne Wikipédia. L'essentiel des informations a néanmoins été collecté prioritairement sur les sites des auteur·trice·s et éditeur·trice·s, ou rédigé à partir de l'expérience d'utilisation des outils cités.

→ Carte global
des champs des
pratiques
numériques en
école d'art et
design

→ Détail de la
carte



Cartographie des outils numériques en Art et en Design (Glossaire)

Outils numériques / Enseignement 2021

AbiWord

AbiWord, un éditeur de texte simple, efficace et extrêmement léger.
www.abisource.com

A-Frame

A-Frame est une infrastructure de développement pour le web (web framework) facilitant la création d'environnements 3D pour la réalité virtuelle et augmentée. Sa syntaxe proche du HTML permet un développement rapide.
www.aframe.io

Anime.js

Anime.js est une bibliothèque d'animation JavaScript.
www.animejs.com

ARCore

Google ARCore est un outil de « vision par ordinateur » destiné à la réalité augmentée qui permet à une machine d'analyser une image ou une vidéo. Google ARCore détecte les visages, les personnes, les surfaces plane afin d'y superposer un modèle 3D.
www.arvr.google.com/arcore

Arduino

Arduino est une plateforme de prototypage open source qui permet aux utilisateurs de créer des objets interactifs à partir de cartes électroniques libres sur lesquelles se trouve un microcontrôleur (Atmega328p, Cortex-M3). Les schémas des cartes électroniques sont publiés en licence libre. Arduino est très utilisée en art, design, ingénierie, etc.
www.arduino.cc

Arduino IDE

Arduino IDE (integrated development environment, « environnement de développement ») est le logiciel permettant de compiler aisément les programmes Arduino sur la carte de prototypage. Il intègre aussi un gestionnaire de cartes électroniques et de bibliothèques d'extension.
www.arduino.cc

ArmorPaint

ArmorPaint est un utilitaire pour peindre des objets en 3D.
www.armorpaint.org

Atom

Atom est un éditeur de texte et de code libre et open source, créé par l'équipe de GitHub. Il est hackable (possibilité de le modifier et de lui ajouter des fonctionnalités personnalisées) et supporte un grand nombre de langages.
www.atom.io

Autodesk Eagle

Autodesk Eagle est un logiciel de conception assistée par ordinateur de circuits imprimés. Il comprend un éditeur de schémas, un logiciel de routage de circuit imprimé avec une fonction d'autoroutage et un éditeur de bibliothèques.
www.autodesk.fr/products/eagle

Bash

Bash (acronyme de Bourne-again shell) est l'interpréteur de commande principal sous Linux. Extrêmement pratique, il permet d'automatiser de nombreuses tâches fastidieuses.
www.gnu.org/software/bash

BirdFont

Birdfont est un éditeur typographique libre et open source pour des projets typographiques simples.
<http://birdfont.org>

Blender

Blender est un logiciel libre de modélisation 3D, d'animation et de rendu. Véritable « couteau suisse », il prend également en charge le montage vidéo non linéaire, la composition, la création nodale de matériaux, ainsi que diverses simulations physiques telles que les particules, les corps rigides, les corps souples et les fluides. Ses capacités sont par ailleurs très extensibles, grâce à un système de greffons continuellement développés par la communauté.
www.blender.org

BNPR/BEER

Blender NPR (Blender non-photorealistic render) et BEER (Blender extended expressive render) rassemblent de multiples outils de rendu non photoréaliste pour Blender.
www.blendernpr.org

Bluetooth

Bluetooth est un protocole de communication permettant l'échange bidirectionnel de données à courte distance en utilisant des ondes radio. Il simplifie les connexions entre les appareils électroniques situés à proximité, sans utiliser de liaison filaire.

Bootstrap

Développée par Twitter, Bootstrap est une collection d'outils destinée au design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs.
www.getbootstrap.com

C

Le C est un langage de programmation impératif généraliste. C'est un langage ancien (1972) de bas niveau, à l'origine de langages de plus hauts niveaux.

C# (C sharp)

C# est un langage de programmation orientée objet dérivé du C++, développé par Microsoft depuis 2000 et destiné à la plateforme Microsoft .NET. Très proche du langage Java, il est notamment intégré dans Unity.

C++

C++ est un langage de programmation compilé permettant la programmation sous de multiples paradigmes. Il est notamment très utilisé pour programmer les plateformes Arduino.

Chrome

Chrome est un navigateur web propriétaire développé par Google, basé sur le projet libre Chromium fonctionnant sous Windows, Mac, Linux, Android et iOS. Chrome est quasi similaire à Chromium mais capte les données de l'utilisateur.
www.google.com/chrome

Chromium

Chromium est un navigateur web libre qui sert de base à plusieurs autres navigateurs, dont certains open source (Iridium) et d'autres, propriétaires (Google Chrome, Opera, Microsoft Edge). Le moteur de rendu (moteur qui permet la transformation du HTML et des autres ressources en affichage) de Chromium s'appelle Blink.
www.chromium.org

CMS

Un CMS (content management system ou « système de gestion de contenu ») est une famille de logiciels destinés à la conception et à la mise à jour dynamique de sites web ou d'applications multimédias. Les CMS permettent notamment de séparer les opérations de gestion de la forme d'un site et de son contenu.

Converseen

Converseen est un logiciel gratuit de conversion d'images multiformat par lots, permettant de lire et convertir plus de 100 formats différents.
<http://converseen.fasterland.net>

CSS

Les CSS (cascading style sheets, « feuilles de style en cascade ») constituent un langage informatique qui décrit la présentation des documents HTML et XML. Ce langage est à la base de la mise en forme de la plupart des pages web actuelles.
www.w3schools.com/css

Cura

Cura est le logiciel libre de découpe de formes 3D en tranches (slicer) permettant l'impression en couches sur les imprimantes 3D. Développé par Ultimaker, il est devenu d'un usage très répandu.
www.ultimaker.com/software/ultimaker-cura

D3.js

D3.js est une bibliothèque JavaScript de visualisation de données. Utilisable à la fois pour l'affichage des données et pour les calculs complexes, elle contient notamment quelques algorithmes célèbres, tel le ForceAtlas, un algorithme de répartition des données dans l'espace afin de rendre les visualisations lisibles et cohérentes.
www.d3js.org

Darktable

Darktable est une application open source de travail photographique et de développement d'image RAW. Il intègre une table lumineuse et une chambre noire virtuelles pour les photographes, et gère les négatifs numériques dans une base de données : une alternative sûre à Adobe Lightroom.
www.darktable.org

Dimmer

Un dimmer (« gradateur » ou « variateur ») est probablement le plus simple montage électronique. Il permet de faire varier un courant de son plus fort potentiel au plus faible.

Django

Django est un framework web écrit en Python, encourageant un développement rapide et une conception pragmatique. Son code est open source et libre.
www.djangoproject.com

DMX

Le DMX (digital multiplexing) est une norme de transmission de données essentiellement utilisée pour le contrôle de l'éclairage et des effets de scène dans le spectacle vivant.

Docker

Docker est un logiciel libre permettant de lancer des applications dans des conteneurs. Cette approche permet d'accroître la flexibilité et la portabilité d'exécution d'une application, laquelle va pouvoir tourner de façon fiable et prévisible sur une grande variété de machines hôtes, que ce soit sur la machine locale, un cloud privé ou public, une machine nue, etc. Il est très utilisé pour la mise en route de systèmes d'exécution complexe, d'environnements web ou d'intelligences artificielles.
www.docker.com

Ddragula.js

Dragula.js est une bibliothèque de glisser/déplacer pour JavaScript.
www.bevacqua.github.io/dragula

Découpe laser

La découpe laser est un procédé de fabrication qui consiste à découper la matière grâce à un laser concentré sur une très faible surface. Cette technologie, majoritairement destinée aux chaînes de production industrielles, est aussi très utilisée dans les fab labs et les écoles de création.

EasyEDA

EasyEDA est un outil gratuit et facile d'utilisation pour la simulation et la conception de circuits imprimés. Il contient une importante base de données et permet d'archiver chaque projet sous Git.
www.easveda.com

FFmpeg

FFmpeg est une collection de logiciels libres destinés au traitement de flux audio ou vidéo (enregistrement, lecture, conversion d'un format à un autre).
www.ffmpeg.org

FontForge

FontForge est un éditeur de polices de caractères numériques gérant la majorité des formats. C'est un logiciel libre sous licence GNU GPL. Il possède une interface de script permettant de manipuler les fontes avec un langage propre, ou bien en utilisant Python.
www.fontforge.org/en-US

FontLab

Probablement le plus populaire des éditeurs typographiques, FontLab est une boîte à outils d'édition de polices.
www.fontlab.com

FreeCAD

FreeCAD est un modèleur 3D paramétrique open source conçu principalement pour concevoir des objets réels de toutes tailles. La modélisation paramétrique permet de modifier le design d'un modèle en remontant dans son historique et en modifiant ses paramètres.
www.freecadweb.org

G-code

Le G-code (norme RS-274) est le langage de programmation le plus répandu pour communiquer avec les machines-outils à commande numérique (CN et CNC).
Gerber (.gbr)
Gerber est un format de fichier permettant de contenir toutes les informations afin de produire des cartes électroniques, comme les pistes, les pastilles, les plages CMS, les vias, etc.

Git et dépôt Git

Dépôt Git est un logiciel de gestion de versions décentralisé. Il permet, par exemple, de garder l'historique des versions de son propre code et de travailler sur un programme à plusieurs. Les dépôts Git les plus connus sont GitHub et GitLab.
www.git-scm.com

GIMP

GIMP (acronyme de GNU image manipulation program) est un outil d'édition et de retouche d'image, libre et gratuit. Il se présente comme l'alternative libre la plus sérieuse à Adobe Photoshop.
www.gimp.org

GLMixer

GLMixer est un mixeur vidéo multi-sources utilisé pour des performances live. Il permet de facilement préconfigurer des scénarios d'enchaînements.
www.sourceforge.net/projects/glmixer

GLSL

Le GLSL ou OpenGL Shading Language est un langage de programmation de shaders.

Godot

Godot est un moteur de jeu libre, open source et exportable sur de nombreuses plateformes (Windows, Linux, web, Android, entre autres). Il comporte toutes les briques de base pour le développement de jeux vidéo, de l'interface au moteur physique en passant par les shaders graphiques. Il se positionne comme un concurrent direct des moteurs de jeu propriétaires.
www.godotengine.org

Google Cardboard

Deux lentilles, un carton bien ajusté et votre téléphone portable suffisent à créer un dispositif de réalité virtuelle. Google Cardboard est un projet idéal pour tester rapidement des projets en VR.
www.arvr.google.com/intl/fr_fr/cardboard

Grasshopper

Grasshopper est un plug-in pour Rhinoceros. Il permet de construire des formes 3D à partir d'un environnement nodal. Celui-ci peut recevoir des données en entrée afin de créer des processus de construction modulaire. Ce procédé présente l'avantage d'être non destructeur sur les formes créées.
www.grasshopper3d.com

GSM

GSM (global system for mobile communications) est l'appellation employée pour désigner un standard numérique de seconde génération (2G) dans le domaine de la téléphonie mobile.

GStreamer

GStreamer est une bibliothèque permettant de construire des chaînes de traitement pour les fichiers vidéo et sonores numériques. Les applications qu'elle prend en charge vont de la simple lecture, du streaming audio/vidéo au traitement audio et vidéo complexe.
www.gstreamer.freedesktop.org

HandBrake

HandBrake est un transcodeur vidéo multiplateforme, open source, libre et gratuit. Un outil très polyvalent pour le travail des fichiers vidéo.
www.handbrake.fr

Handlebars

Handlebars est un « décorateur » HTML qui permet d'insérer des éléments JavaScript dans un code HTML. Ce principe est à l'origine de nombreux frameworks HTML actuels, parmi lesquels BlazeJS ou VueJS.
www.handlebarsjs.com

HTC Vive

HTC Vive est un casque de réalité virtuelle fonctionnant sur ordinateur. Il est très utilisé pour les phases de prototypage ou les réalisations d'expériences interactives.
www.vive.com/fr

HTML

HTML (hypertext markup language) est le langage pour créer des pages internet. HTML fonctionne grâce à des « balises » qui structurent les pages. Il est probablement le langage le plus populaire, le plus simple et souvent synonyme des premiers pas vers la compréhension d'un langage interprété.

HTTP / HTTPS

HTTP (hypertext transfer protocol) et HTTPS (pour la version sécurisée) est le principal protocole de communication utilisé pour le transfert et l'affichage des pages web. Il peut aussi être utilisé pour du transfert de données et de la communication entre les objets.

I2C

L'I2C est un bus informatique (dispositif de transmission de données) standard qui permet de connecter un microcontrôleur à divers composants au sein d'un système électronique.

IanniX

IanniX est un séquenceur graphique open source et gratuit, inspiré des travaux de Iannis Xenakis, destiné à la création numérique. IanniX synchronise via Open Sound Control (OSC) des événements et courbes vers un environnement temps-réel.
www.iannix.org/fr/download-iannix

Inkscape

Inkscape est un logiciel libre de dessin vectoriel sous licence GNU GPL. Il gère des fichiers conformes avec les standards XML, SVG et CSS du W3C. Equivalent d'Illustrator, il se distingue néanmoins par sa gestion fine du XML qui permet d'ajouter des propriétés à des objets pour une utilisation dans des environnements de programmation. De nombreux plug-ins existent, notamment pour les tables traçantes, les brodeuses numériques ou les découpeuses lasers.

www.inkscape.org/fr

Java

Java est un langage de programmation orienté objet parmi les plus utilisés au monde. Il est simple et peut se compiler/s'exécuter sur plusieurs plateformes via la machine virtuelle Java (JVM). Processing est basé sur le langage Java.

www.java.com/fr

www.openjdk.java.net

JavaScript

JavaScript (souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. Mais il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs web tels que Node.js. Le code JavaScript est souvent interprété ou compilé à la volée (compilation JIT, « Just in Time »).

jQuery

jQuery est une bibliothèque JavaScript qui servait à l'origine à unifier les variations de code pouvant exister dans les différents navigateurs. Extrêmement populaire, elle simplifie l'écriture de scripts et de fonctionnalités (animations, manipulation CSS) en HTML.

www.jquery.com

JSON

Le JSON (JavaScript object notation) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de présenter de l'information structurée dans un format compatible avec un grand nombre de langages de programmation et de logiciels, ce qui permet une large interopérabilité.

www.json.org

KdenLive

KdenLive est un logiciel open source de montage vidéo remarquable notamment par son développement rapide et prometteur.

www.kdenlive.org/fr

Kinect

La Kinect est un périphérique originalement développé par Microsoft pour la console de jeux Xbox. C'est une webcam couplée avec un émetteur/récepteur infrarouge qui permet de reconstituer une image 3D de l'espace visionné. Cette caméra, très utilisée dans les installations interactives, n'est plus produite aujourd'hui, mais d'autres constructeurs proposent des alternatives.

Leaflet

Leaflet est une bibliothèque de cartographie open source et « agnostique », permettant d'utiliser n'importe quel service de cartographie – Google Maps, OpenStreetMap, Mapbox... Elle dispose de nombreuses fonctionnalités utiles telles que l'annotation, l'affichage de tracé ou encore la gestion des zooms et des déplacements.

www.leafletjs.com

LED

Une LED (light-emitting diode) – le plus visuel des composants électroniques – permet notamment d'opérer un débogage simple en électronique pour comprendre des signaux électriques.

Less

Less (leaner style sheets) est un préprocesseur open source de CSS, un métalangage pour produire des feuilles de style de manière plus complexe avec moins de répétition. Il supporte par exemple l'héritage, les variables et les calculs mathématiques.

<http://lesscss.org>

LibreOffice

LibreOffice est une suite bureautique connue pour être complète, libre, open source et multiplateforme.

www.libreoffice.org

Löve

Löve est un moteur de jeu vidéo minimaliste mais puissant par sa conception supportant l'accélération graphique. Il n'a pas besoin d'être compilé, il suffit de l'interpréter avec le langage Lua.

www.love2d.org

Lua

Lua est un langage de script s'adossant généralement à un moteur, notamment de jeu vidéo ou à un framework. Il est simple d'utilisation et facile d'accès.

www.lua.org

MakeHuman

MakeHuman est un logiciel de conception de personnages humanoïdes en 3D. Le modèle est construit à partir de variations de caractéristiques physiques.

<http://www.makehumancommunity.org>

Manuskript

Manuskript est un logiciel d'aide à l'écriture open source qui permet de structurer un texte avec différents outils « sans distraction ».

www.theologeek.ch/manuskript

Markdown

Markdown est un langage de balisage léger qui a pour but d'offrir une syntaxe facile à lire et à écrire. Un document balisé en Markdown peut être consulté sans être perturbé par le balisage. Il est possible ensuite de le transformer en HTML, PDF ou d'autres formats.

www.daringfireball.net/projects/markdown

Max/MSP

Max/MSP est un logiciel musical permettant de faire de la synthèse sonore, de l'analyse, de l'enregistrement, ainsi que du contrôle d'instrument MIDI. Il a été développé par l'Ircam dans les années 1980 et est l'un des logiciels musicaux parmi les plus utilisés tant par les musiciens professionnels que par les amateurs. La branche de développement libre et open source de ce programme est Pure Data.

www.cycling74.com/products/max

MeshLab

MeshLab est un logiciel libre et open source spécialisé dans le traitement de fichier tridimensionnel. Il permet, par exemple, de corriger des modèles ou de mettre en place des processus de reconstruction.

www.meshlab.net

MeshRoom

Meshroom est un logiciel libre et open source de photogrammétrie.

www.alicevision.org/#meshroom

Meteor

Meteor est un framework open source de développement web en JavaScript basé sur Node.js. Le projet vise à créer un framework induisant un changement de paradigme, notamment sur la manière de considérer les architectures client-serveur.

www.meteor.com

Meteor Up

Meteor Up est un outil permettant la publication de sites internet basés sur Meteor. Il permet de configurer un serveur et d'y déployer facilement une application Meteor.

<http://meteor-up.com>

Microsoft Edge

Microsoft Edge est le navigateur internet officiel de Microsoft. Basé sur le moteur de rendu de Chromium, il est multiplateforme.

www.microsoft.com/fr-fr/edge

Microsoft Office

Microsoft Office est une suite bureautique propriétaire permettant l'édition de documents, de feuilles de calcul, de présentations, etc.

www.microsoft.com/fr-fr/microsoft-365

MIDI

Le MIDI (Musical Instrument Digital Interface) est un protocole de communication et un format de fichier dédiés à la musique. Il est utilisé pour la communication entre instruments électroniques, contrôleurs, séquenceurs, logiciels de musique ainsi que pour de nombreuses installations interactives en temps réel.

www.midi.org

MI5.js

MI5.js est une bibliothèque JavaScript permettant d'utiliser des outils de machine-learning dans un navigateur web.

www.mi5js.org

MongoDB

MongoDB est un système de gestion de base de données orienté document, distribué et universel. Il est fréquemment couplé avec Node.js.

www.mongodb.com

Moteur CC

Les moteurs CC (moteurs à courant continu) sont des moteurs peu précis mais véloces et puissants. Ils sont très utilisés pour mouvoir des machines telles que des ventilateurs, des hélices, des roues, etc.

Moteur pas à pas

Les moteurs pas à pas sont précis car chaque impulsion électrique correspond à un déplacement d'angle donné. Ils sont très utilisés pour coordonner des mouvements mécaniques comme dans les CNC, les imprimantes 3D ou la robotique.

Mozilla Firefox

Mozilla Firefox est un navigateur web de référence développé et distribué par la Mozilla Foundation. Il est libre, gratuit et open source.

www.mozilla.org/fr/firefox

MySQL

MySQL est un système de gestion de bases de données relationnelles. Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données parmi les plus utilisés au monde, autant par le grand public (applications web, principalement) que par des professionnels.

www.mysql.com

NanoKONTROL de Korg

Les NanoKONTROL de la marque Korg sont une série de contrôleurs MIDI portables et relativement abordables en termes de prix. Ils peuvent être connectés avec un grand nombre de logiciels dans le cadre de la création d'interfaces.

www.korg.com/fr/products/computergear/nanokontrol2/

Natron

Natron est un puissant logiciel pour les effets spéciaux vidéo. Il dispose d'outils d'incrustation, de rotoscopie, de suivi de caméra, etc. Il est libre, open source et gratuit.

www.natrongithub.github.io

Nginx

Nginx [engine X] est un serveur web open source qui, du fait de son succès, est désormais utilisé comme reverse proxy (le proxy inverse permet à un utilisateur d'internet d'accéder à des serveurs internes), cache HTTP et load balancer (répartition d'un ensemble de tâches sur un ensemble de ressources). Asynchrone, il constitue souvent la première porte d'entrée vers un espace web.

www.nginx.org

NodeBox

NodeBox est un logiciel multiplateforme de programmation nodale et en langage Python destiné à la visualisation graphique. Il peut se présenter comme une sorte de Processing nodal.

www.nodebox.net

Node.js

Node.js est une plateforme logicielle libre en JavaScript orientée vers les applications nécessitant un développement en équipe avec de fortes montées en charge. C'est un environnement bas niveau permettant l'exécution de JavaScript côté serveur. Il est une des multiples alternatives à PHP.

www.nodejs.org

NodeMCU

NodeMCU est une plateforme électronique programmable à volonté. Basée sur les processeurs ESP8266, elle est puissante et contient des options de connectivité étendue. Cette plateforme de prototypage est pleinement ancrée dans le champ des objets connectés.

<http://www.nodemcu.com>

Nvidia Jetson

Nvidia Jetson sont des cartes électroniques dédiées à l'intelligence artificielle combinant performances de calcul et un coût énergétique réduit.

www.nvidia.com/fr-fr/autonomous-machines/embedded-systems

OBS Studio

OBS Studio, abrégé en OBS (open broadcaster software), est un logiciel libre et open source de capture d'écran et de streaming multiplateforme. Il permet aussi de faire du montage en temps réel.

www.obsproject.com

Oculus Quest

Oculus Quest est un casque de réalité virtuelle lancé en mai 2019 et créé par Oculus VR, une filiale de la société Facebook. C'est un des premiers casques de réalité virtuelle complètement autonome.

www.oculus.com

Olive

Olive est un éditeur vidéo libre, open source et gratuit.

www.olivevideoeditor.org

OpenCV

OpenCV est une bibliothèque de « vision par ordinateur » qui permet de discriminer et quantifier les objets dans le champ de vision d'une webcam. OpenCV est disponible dans de nombreux environnements de programmation.

www.opencv.org

OpenDataCam

OpenDataCam est un dispositif comprenant une caméra et un mini-ordinateur. Il exécute des algorithmes de détection d'objets qui comptent et suivent les objets en mouvement. C'est un outil performant pour l'analyse d'images et le traitement vidéo.

www.move-lab.com/proiect/opendatacam

OpenOffice

OpenOffice est une suite bureautique complète, multiplateforme, maintenue par Oracle.

www.openoffice.org/fr

OpenSCAD

OpenSCAD est un logiciel 3D qui ne se concentre pas sur les aspects artistiques de la modélisation 3D, contrairement à la plupart des logiciels libres de création 3D (tels que Blender) mais sur les aspects CAO. Il est entièrement scriptable et permet de modéliser des objets via du code. Il est gratuit et disponible pour Linux/UNIX, Windows et Mac OS X.

www.openscad.org

OSC

L'Open Sound Control est un format de transmission de données entre ordinateurs, synthétiseurs, robots ou tout autre matériel ou logiciel compatible. Conçu pour le contrôle en temps réel, il utilise le réseau à travers des protocoles UDP ou TCP et apporte des améliorations en termes de rapidité et flexibilité par rapport à l'ancienne norme MIDI. Il est supporté par beaucoup de logiciels et donc très intéressant pour transférer des données d'un logiciel/matériel à l'autre.

<http://opensoundcontrol.org>

P5.js

P5.js est une adaptation JavaScript de Processing pour le web.

www.p5js.org

Packery

Packery est une bibliothèque JavaScript effectuant des calculs de placement et de déplacement de tuiles dans une page web afin d'obtenir des mises en pages dynamiques. La licence est libre pour des projets open source.

www.packery.mtazfzzy.co

Paper.js

Paper.js est une bibliothèque d'aide à la composition vectorielle. Initiée par Jürg Lehni et Jonathan Puckey, elle permet de composer des calques, animer des formes, mettre en place des compositions graphiques automatisées, etc. L'avantage de cette bibliothèque est de pouvoir éditer ces formes directement, à la manière d'un logiciel vectoriel (type Illustrator ou Inkscape).

<http://paperis.org>

PixiJS

PixiJS est un moteur de rendu 2D, basée sur WebGL. Il permet de considérablement accélérer l'affiche graphique bidimensionnelle en HTML5. Il est distribué en licence MIT.

www.pixijs.com

PLA

PLA est l'abréviation de l'acide polylactique, un polymère biodégradable qui peut être obtenu à base d'amidon de maïs. Il est le matériau d'origine naturelle le plus utilisé dans les imprimantes 3D additives.

Port

Le port logiciel permet, dans la suite des protocoles internet (TCP/IP), de distinguer différents interlocuteurs sur un ordinateur donné. Ces interlocuteurs sont des programmes informatiques qui, selon les cas, écoutent ou émettent des informations sur ces ports. Un port est distingué par son numéro.

PraxisLIVE

PraxisLIVE est une version nodale de Processing. Chaque module se comporte comme un sketch (programme) de Processing que l'on peut modifier, agencer, coupler et/ou activer. PraxisLive peut être utilisé notamment pour des performances en temps réel.

www.praxislive.org

Print.js

Print.js est une bibliothèque JavaScript qui permet de simplifier les fonctions d'impression en HTML afin de préparer par exemple une publication à partir d'un navigateur internet.

www.printis.crabbly.com

Processing

Processing est un environnement de développement simple et un langage pour apprendre à coder dans le contexte des arts visuels. Ce projet, initié par Ben Fry et Casey Reas, est devenu très populaire dans le milieu de l'art, du design, du graphisme et de la recherche. Il est notamment utilisé pour le prototypage, la mise en œuvre d'installations interactives ou la visualisation de donnée.

www.processing.org

Processing.js

Processing.js est la version JavaScript de Processing. Contrairement à p5.js, Processing.js garde exactement la même syntaxe que Processing. Il peut être utilisé pour porter du code Processing vers du web sans traduction. Le code source de ce projet n'est toutefois plus maintenu.

www.github.com/processing-js/processing-js

Pure Data (Pd)

Pure Data (souvent abrégé « Pd ») est un logiciel de programmation graphique pour la création musicale et multimédia en temps réel. Il permet notamment de gérer des signaux entrant et sortant de l'ordinateur.

<http://puredata.info>

<http://msp.ucsd.edu/software.html>

Pygame

Pygame est un moteur de jeu vidéo pour Python.

www.pygame.org

Python

Python est un langage de programmation très utilisé, notamment dans le champ de l'intelligence artificielle. Interprété, multiparadigme et multiplateforme, Python fonctionne sur smartphones, Windows, Unix, Linux, Android, iOS, entre autres. Il est en licence libre.

www.python.org

PyTorch

PyTorch est une bibliothèque logicielle Python open source d'apprentissage machine qui s'appuie sur la bibliothèque Torch développée par Facebook. PyTorch permet notamment d'effectuer les calculs nécessaires pour l'apprentissage profond (deep learning).

www.pytorch.org

Raphaël

Raphaël est une des multiples bibliothèques JavaScript permettant de manipuler de manière interactive des graphismes vectoriels (type SVG) dans une page web. Elle est utilisée dans la visualisation de données.

www.dmitrybaranovskiy.github.io/raphael

Raspberry Pi

Le Raspberry Pi est un nano-ordinateur monocarte à processeur ARM. Son faible coût en fait une machine idéale pour du prototypage, de l'installation interactive, etc.

www.raspberrypi.org

RawTherapee

RawTherapee est un logiciel de traitement d'image Raw open source de qualité professionnelle. Il constitue une alternative à Adobe Camera Raw.

www.rawtherapee.com

Rhinoceros 3D

Rhinoceros 3D est un logiciel de conception assistée par ordinateur utilisé notamment dans le milieu du design industriel ou de l'architecture. Il sert à concevoir des formes complexes (véhicules, mobilier, produits électroniques, objets, etc).

www.rhino3d.com

Runway

Runway est un logiciel à destination des créateurs visant à faciliter l'utilisation d'algorithmes d'intelligences artificielles.

www.runwayml.com

Sass

Sass est un préprocesseur CSS qui permet la génération de dynamique de CSS. SassScript est un langage de script pouvant être utilisé à l'intérieur du code Sass.

www.sass-lang.com

Scribus

Scribus est une alternative libre à Adobe InDesign. Son architecture robuste et scriptable fait de lui un logiciel permettant la publication de documents imprimés de qualité professionnelle.

www.scribus.net

Serial port / port série

Un port série (généralement compatible avec la norme RS-232) est une interface de communication série par laquelle l'information est transférée en entrée ou en sortie de manière séquentielle, un bit à la fois, contrairement à un port parallèle qui communique plusieurs bits simultanément en parallèle. Du fait de sa simplicité, il est encore très utilisé pour coder les cartes électroniques du type Arduino, Teensy, NodeMCU, etc.

Servomoteur

Les servomoteurs sont des moteurs capables de maintenir une position à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure. Le servomoteur intègre, dans un même boîtier, la mécanique (moteur et engrenage) et l'électronique pour la commande et l'asservissement du moteur.

Shader

Un shader est un programme informatique pour paramétrer une partie du processus de rendu d'une image de synthèse. L'étape de shading est exécutée par le moteur de rendu logiciel ou, préférentiellement, par la carte graphique.

Shields

Les shields (boucliers), principalement adaptés au Arduino, sont une sorte de plugin électronique que l'on ajoute à la carte principale afin d'étendre ses fonctionnalités – par exemple, pour contrôler des moteurs, des courants forts, un affichage LED.

Shotcut

Shotcut est un logiciel libre de montage vidéo multiplateforme léger et complet.

www.shotcut.org

Sigfox

Sigfox est un opérateur de télécommunications français créé en 2009, dédié à l'internet des objets. Les communications sont plus efficaces en termes de qualité et de distance mais avec des temps de rafraîchissement longs. Depuis peu, des plateformes telles qu'Arduino proposent des bibliothèques associées aux émetteurs-récepteurs Sigfox.

www.sigfox.com/

Slic3r

Slic3r est un programme open source de découpage de formes 3D permettant de produire des fichiers G-code lisibles par les imprimantes 3D – un fichier G-code consiste en la description d'un parcours mécanique pour les buses d'impression.

www.slic3r.org

Snap.svg

Snap.svg ou JavaScript Snap.svg est une bibliothèque facilitant l'intégration des graphismes au format SVG dans les pages web.

<http://snapsvg.io>

Spark AR

Spark AR est un ensemble de logiciels et bibliothèques servant à créer des filtres photographiques en réalité augmentée, destinés à Facebook ou Instagram.

www.sparkar.facebook.com

Spectrum

Spectrum est une bibliothèque JavaScript permettant d'afficher et d'utiliser des sélecteurs de couleurs sur les pages web.

www.bgrins.github.io/spectrum

SPI

Une liaison SPI (serial peripheral interface) est un bus de données série synchrone utilisé dans les circuits électroniques. Les composants communiquent selon un schéma maître-esclave où le maître contrôle la communication.

SQL

SQL (structured query language) est un langage informatique normalisé qui permet d'exploiter les bases de données relationnelles.

SSH

SSH (pour Secure Shell) est un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Il est très utilisé pour sécuriser les communications entre ordinateurs et serveurs.

Stéréolithographie

La stéréolithographie est une technique de prototypage rapide qui permet de fabriquer des objets solides à partir d'un modèle numérique. L'objet est obtenu par superposition de couches fines de matière (méthode additive). Elle se distingue de la méthode du dépôt de fil.

Substance

Substance d'Adobe est un ensemble d'outils logiciels propriétaires pour créer des textures destinées à la 3D.

Substance Alchemist permet la création de textures à partir de photographies ; Substance Designer, à partir de générateurs mathématiques (shaders) ; quant à Substance Painter, il permet de peindre directement les textures sur des objets 3D.

www.substance3d.com

SVG

Le format SVG (scalable vector graphics) est un format de données ASCII (norme informatique de codage de caractères) conçu pour décrire des ensembles de graphiques vectoriels. Il est très utilisé, notamment en cartographie et pour la conception d'interface.

TCP

Le TCP (transmission control protocol, « protocole de contrôle de transmissions ») est un protocole de transport de données. Il est notamment utilisé pour transmettre le contenu des pages internet.

TensorFlow

TensorFlow est un outil open source d'apprentissage automatique développé par Google. Il est l'un des outils parmi les plus utilisés dans le domaine de l'intelligence artificielle.

www.tensorflow.org

Thonny

Thonny est un environnement de développement pour Python conçu pour les débutants. Il supporte différentes façons de parcourir le code, l'évaluation des expressions étape par étape et un mode d'explication des concepts de référence. Il peut s'apparenter à un éditeur Processing mais pour Python.

www.thonny.org/

Three.js

Three.js est une puissante bibliothèque JavaScript qui permet de créer des scènes 3D dans un navigateur web, sans faire appel à un plug-in extérieur. Ses fonctionnalités reposent sur WebGL.

www.threejs.org

Tiled

Tiled est un éditeur de niveaux de jeu vidéo 2D. Sa fonction principale est d'éditer des « tuiles » de différentes formes afin de constituer des niveaux de jeu. Le résultat peut être utilisé dans n'importe quel moteur de jeu.

www.mapeditor.org

TouchDesigner

Développé par la société Derivative, TouchDesigner est un logiciel propriétaire basé sur de la programmation nodale. Il est notamment utilisé pour produire des installations interactives en temps réel et diverses applications créatives.

www.derivative.ca

Trumbowyg

Trumbowyg est une bibliothèque JavaScript pour l'édition de contenu en mode WYSIWYG (what you see is what you get, ce qui est affiché sur l'écran est identique à ce qui sera imprimé sur papier). Reposant sur jQuery, il est très utile aux développeurs souhaitant gérer la publication en ligne.

www.alex-d.github.io/Trumbowyg

Two.js

Two.js est une bibliothèque de dessin en 2D destinée aux navigateurs web actuels. Elle permet de dessiner dans de multiples contextes tel SVG, Canvas ou WebGL.

www.two.is.org

UDP

Le UDP (user datagram protocol ou protocole de datagramme utilisateur), abrégé en UDP, est un des principaux protocoles de télécommunication utilisés par internet. Le rôle de ce protocole est de permettre la transmission de données (sous forme de paquets de données ou datagrammes) de manière très simple entre deux entités, chacune étant définie par une adresse IP et un numéro de port. Aucune communication préalable n'est requise pour établir la connexion, au contraire du TCP. L'UDP est très utilisé pour transmettre des informations de manière rapide, entre plusieurs entités informatiques.

Unity

Unity est un moteur de jeu multiplateforme (smartphone, ordinateur, consoles de jeux vidéo et web) développé par Unity Technologies. Il est l'un des plus répandus dans l'industrie du jeu vidéo, aussi bien auprès des grands studios que des indépendants, car il permet un prototypage rapide des applications. Sa licence dite « Personal » est gratuite.

www.unity.com

Unreal Engine

Unreal Engine est un des moteurs de jeu vidéo parmi les plus avancés en termes de photoréalisme, de physique et d'effets spéciaux. Sa licence « permissive » donne la possibilité aux développeurs indépendants d'utiliser un outil robuste et gratuit, dans la limite d'un certain chiffre d'affaires.

www.unrealengine.com

URL

Une URL (uniform resource locator), couramment appelée « adresse web », est une chaîne de caractères uniforme qui permet d'identifier une ressource web par son emplacement et de préciser le protocole internet pour la récupérer (par exemple, HTTP ou HTTPS). Elle peut localiser divers formats de données : document HTML, image, son...

Victor.js

Victor.js est une bibliothèque JavaScript permettant la manipulation de vecteurs mathématiques afin de calculer des distances, des produits scalaires, des magnitudes, des rotations, etc.

<http://victoris.org>

Vimix

Vimix est un mélangeur vidéo basé sur une approche graphique. Il peut être utilisé en live afin de mixer plusieurs sources vidéos audio et/ou sources extérieures. Son développement prend la suite du logiciel GLMixer.

www.brunoherbelin.github.io/vimix

WebGL

WebGL permet d'utiliser le standard OpenGL ES au sein d'une page web basée sur le standard HTML5, en s'aidant du langage JavaScript, de données au format JSON et de l'accélération matérielle pour les calculs et le rendu 3D. La plupart des jeux vidéo en HTML convoquent cette interface de programmation JavaScript.

WebSocket

Le WebSocket est un protocole de communication web qui permet notamment de recevoir des données depuis le serveur, sans pour autant avoir effectué de requête vers celui-ci.

Wifi

Le ou la wifi est un ensemble de protocoles de communication sans fil. Un réseau wifi permet de relier par ondes radio plusieurs appareils informatiques (ordinateur, routeur, smartphone, modem internet, etc.) au sein d'un réseau informatique afin de permettre la transmission de données entre eux.

Wiki

Un wiki est une application web qui permet la création, la modification et l'illustration collaborative de pages à l'intérieur d'un site web. Il utilise un langage de balisage et son contenu est modifiable au moyen d'un navigateur web.

WordPress

WordPress est un système de gestion de contenu (CMS) gratuit, libre et open source. Ce logiciel écrit en PHP repose sur une base de données MySQL et est distribué par la fondation WordPress.org.

wordpress.org

XML

Le XML (extensible markup language) est un métalangage (langage de programmation conçu pour décrire d'autres langages de programmation informatique) de balisage. Sa syntaxe est dite « extensible » car elle permet de définir différents langages avec, pour chacun, un vocabulaire et une grammaire propres – le SVG, par exemple.

www.w3.org/XML

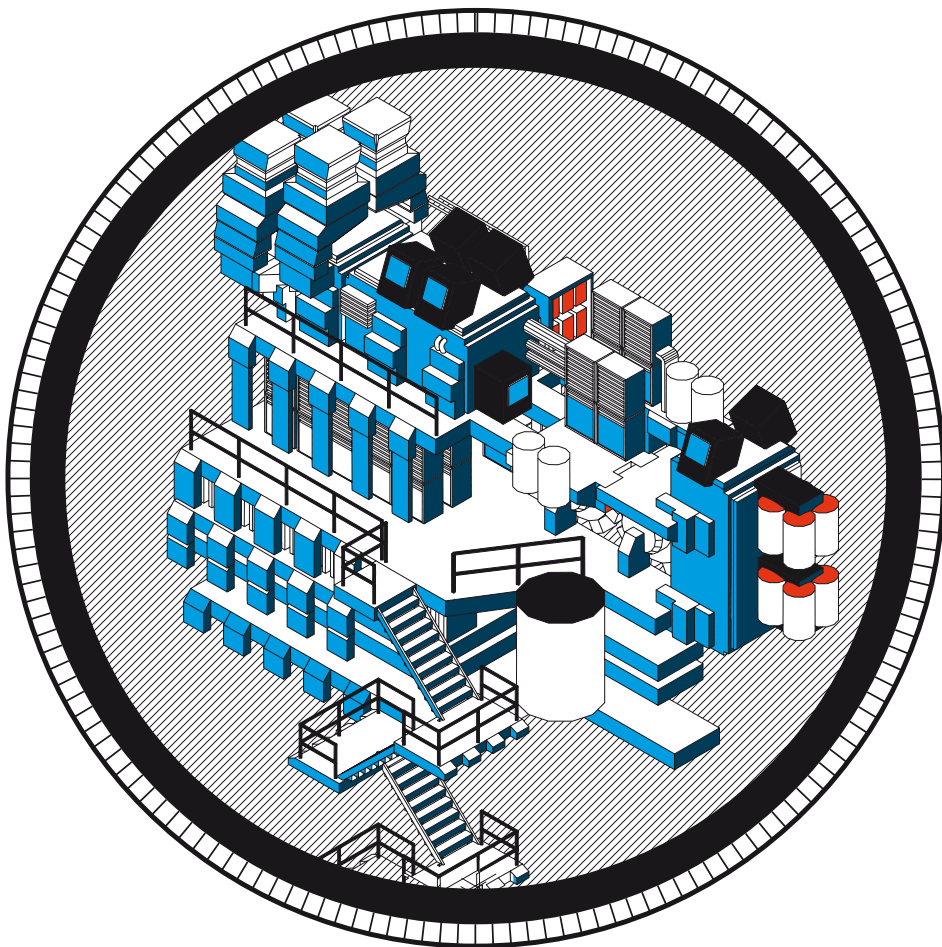
.CORP /

Textes et images, fictions

La Plateforme

Serais-ce une usine d'assemblage de matières premières ? Pas de bois d'ébène, de pétrole ou de gaz naturel mais des citernes de lettres de tous les alphabets, des caisses d'idéogrammes, des torrents de pixels, des kilomètres de lignes de code, des étendues infinies de coups de crayon, des boccas de neurones. Et une mécanique pas prête de s'enrayer. Les turbines chauffent à fond, les pistons coulisent, les tambours rythment la danse et des palettes glissent jusque dans les camions. Sous le cellophane, des piles de livres, des rayons de revues, des pyramides de catalogues; et sous le papier bulle, des cartons de flèches signalétiques, des boîtes de logos, des fagots de planches gravées d'identités visuelles, des rouleaux d'affiches de communication et des rouleaux de cartographies de territoires indéterminés; sous les plaques de polystyrène, des ribambelles de typographies, des piles de photographies, des amas de pièces d'auteur; et sous les couvertures à carreaux, des stocks d'écrans, et sur les écrans, des interactions de personnages de jeu vidéo avec des sites internet. C'est un lieu étrange fait de ponts et de tunnels, d'escaliers qui montent et d'escaliers qui descendent, d'ascenseurs horizontaux, de colonnes de béton montées sur roulettes, de turbines silencieuses fixées au plancher qui ronronne, de tuyaux métalliques en liberté, de machines qui ronflent, d'éclats de pixel suspendus, d'un terrain vague intérieur. Un monde à part relié au monde de toute part. Une machine en mouvement, ambulante. On distingue tout de même trois niveaux. Dans les deux premiers niveaux, des pôles en open-space, des galeries ouvertes, des cimaises molles, des cascades d'eau sèche comme séparations des entités. Tout fermé et tout ouvert. Du dehors dedans.

NIVEAU 1 _ Dans un coffre en fer forgé, suspendu à une guirlande en papier, un portrait en défonce, une silhouette défenestrée dans un paysage de lettres ou de pixels, de vecteurs ou de coups de crayons selon la situation du coffre dans l'usine. Une spécialité, une identité se distingue. Là, un monte-charge traverse horizontalement l'espace, en faisant de petits sauts singuliers, et déverse en mikado, une tonne de lignes encodées. Un oiseau mécanique bringuebalant s'affaire à retirer une à une, les lignes sans faire vibrer les autres. Il balance ensuite énergiquement le code comme des flèches qui s'enfoncent dans un miroir reflétant l'envers du décor. Malgré des cernes en couvercle de boîtes de conserves recyclées, l'oiseau fait preuve d'adresse. Chaque projection de ligne remue la plate-forme entière, gémissante et fait valdinguer ses organes pour un mouvement perpétuel. Pour se déplacer dans cette machine-territoire, il existe plusieurs types d'escaliers. Les plus commodes sont les escaliers qui montent. Bien qu'à sens unique, ceux-ci ne ressemblent en aucun cas à des escalators de galeries (qu'elles soient marchandes ou underground). Ils ont d'ailleurs l'aspect d'escaliers en béton tout à fait classiques, seulement, les marches n'existent que dans le sens de la montée ; un effet d'optique engendrant, malgré tout, son lot de dégringolade, de fracture du tibia et d'arrêt de travail. De la même manière, les escaliers qui descendent sont infranchissables pour quiconque veut accéder au plateau supérieur. Ils ont, néanmoins, le désavantage de présenter un appui pour les pieds



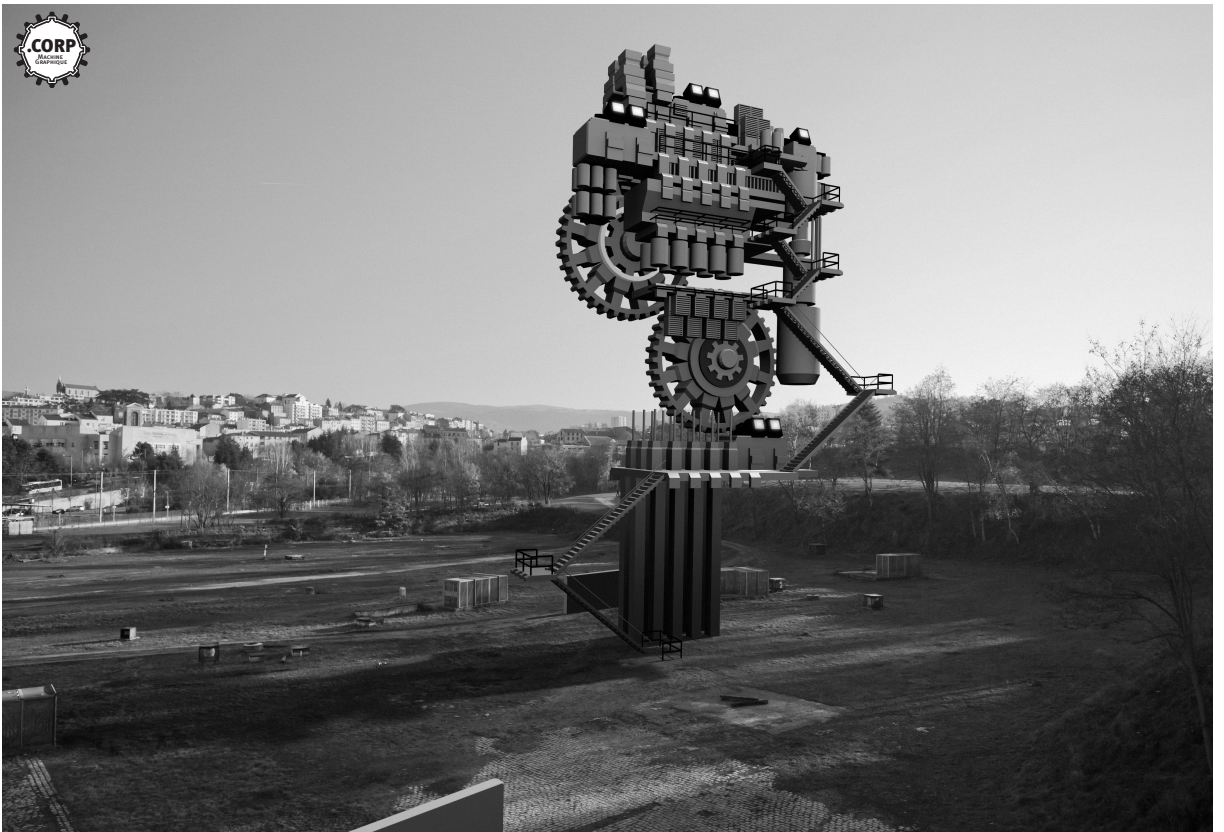
† Logo fictif
de .CORP

incliné de 10% en haut à 45% en bas. Ainsi, les visiteurs non habitués sont régulièrement repêchés dans la fontaine de leds lumineuses installées à l'arrivée de chacun des escaliers de ce type. Enfin, les escaliers transversaux s'étendent de salles des machines en salles des machines, sur un même niveau et sont les uniques accès praticables, à l'exception des échelles renversées que l'on parcourt à quatre pattes. Le coin des marches se trouve en sommet d'où l'utilisation recommandée de chaussures de sécurité à semelles creuses. Dans un coin de pelouse goudronnée, éclairée par un soleil branché sur secteur, un marteau piqueur muni d'une pointe de rotring s'agite. Le bras mécanique qui le guide s'articule en déversant des syllabes jusqu'à la mine du piqueur. A l'origine du bras, une caisse en cuivre bat un rythme sourd et couvre à peine un son d'orgue accompagné d'un rire machiavélique. Le piqueur est résigné à laisser s'échapper ces flux de part et d'autres de ses membres. Il compose. Alimenté par des tuyaux sortant des murs et des pôles supérieurs, de lettres minuscules au dessin maîtrisé, le piqueur est sous perfusion. Des rails quadrillent ce niveau. Plus que ça, ils le maintiennent. Ce sont les rails fondateurs dont la disposition est calquée sur le nombre d'or. Selon les formes que prend la structure, en constante réorganisation, les rails se déplacent pour conserver l'harmonie de la grille. Durant ces mouvements : le chariot qui s'agrippe, les cimaises qui basculent dans les chaudrons de peinture fraîche, les escaliers descendant qui remontent la pente, les filets d'eau des cascades qui se nouent aux fils électriques suspendus lors d'un charmant accouplement et un long vrombissement. Le chariot reprend vite ses esprits et parcourt à toute allure la nouvelle grille. Il balise le nouveau territoire. Monté sur rails, il n'en garde pas moins son authentique déséquilibre provoqué par des roulements carrés

encastrés dans trois coins sur quatre. Sa cargaison regorge de collections diverses, classées, répertoriées, casées. Collection de pixels bruts datant de la première moitié d'un siècle à déterminer, collection de « a », avec ou sans accent, laissant bouche bée, collection d'une pièce abstraite en valant deux, collection de points de vue, collection de pages, collections de lignes droites destinées aux diagrammes et de lignes distordues pour les représentations territoriales. Collection de collections. D'énormes réseaux de tuyauterie alimentent ce niveau depuis l'extérieur. L'activité de la machine est dépendante de cet apport en énergie. Sa gueule béante avalent les données extérieures et la machine répond, après digestion et passage dans les différents organes, par la production de bobines potentiellement explosives. Le niveau est poreux, il fait corps avec son environnement. Les rails s'allongent, s'étendent et s'enracinent dans le territoire voisin. La plateforme se soulève, gonfle ses poumons et s'étire plus largement. Les rails sortent de terre et s'enracinent un peu plus loin. Un plateau intermédiaire brinquebalant s'élève au dessus du premier niveau. Il flotte parfois à ras du sol. A d'autre moment, il faut s'allonger pour ne pas s'assommer aux conduits du plafond. Un tube, s'enroulant autour du plateau formant un ventre en son milieu et s'achevant par une sorte de pavillon de trompette dirigé vers le sol, permet l'ascension du semi-étage. Malgré les apparences, il ne s'agit pas de souffler dans l'embouchure du tube. Une lentille géante montée sur un tricycle à moteur, se déplace sur le plateau, et filtre les rayons lumineux projetés de toute part dans l'immense plateforme. Ceux-ci sont alors redirigés vers des capteurs photosensibles et en devenant image éphémère s'abîment dans un bruissement harmonieux (ou pas). Quoiqu'il en soit, ces infimes notes émises par les rayons agonisant jusqu'à l'extinction sont perçues par le tube qui se contente d'en faire une énergie mécanique pour élever le niveau. NIVEAU 2 _ A cet étage, la machine est auto-productrice. Les impulsions extérieures ne lui sont pas nécessaires. Nuages bas, arbustes et étendues sauvages s'accordent à sortir ensemble sur papier photo. Les lignes de codes tricotent un personnage avec des lettres qu'un stylo étourdi à laisser traîner. Le voilà d'ailleurs qui commence et recommence à tracer les mêmes lettres. Un stroboscope racole en projetant des slogans aux couleurs affriolantes. Des cimaises attendent leur tour pendant que des cellules cérébrales s'agitent et s'occupent à trouver des mots pour les idées et réciproquement. NIVEAU 3 _ A la fois complètement dans la plateforme et totalement en dehors, ce niveau à ciel ouvert est une station de production d'air comprimé, redistribué dans certains caissons des niveaux 1 et 2. Cette filiale comprend une chaîne de réflexion, maturation, communication ainsi qu'une chaîne de réception/création, tampon, expédition constituant le circuit édition.

→ Image
fictionnel
de .CORP rendu
sous blender

→ Image
fictionnel
de .CORP



—

—

—
—

**Sujets de cours/
Workshop/
Documents pédagogiques**

Sujet de cours

—

Pratiques numériques / Cultures électroniques / Collection de sujet de cours

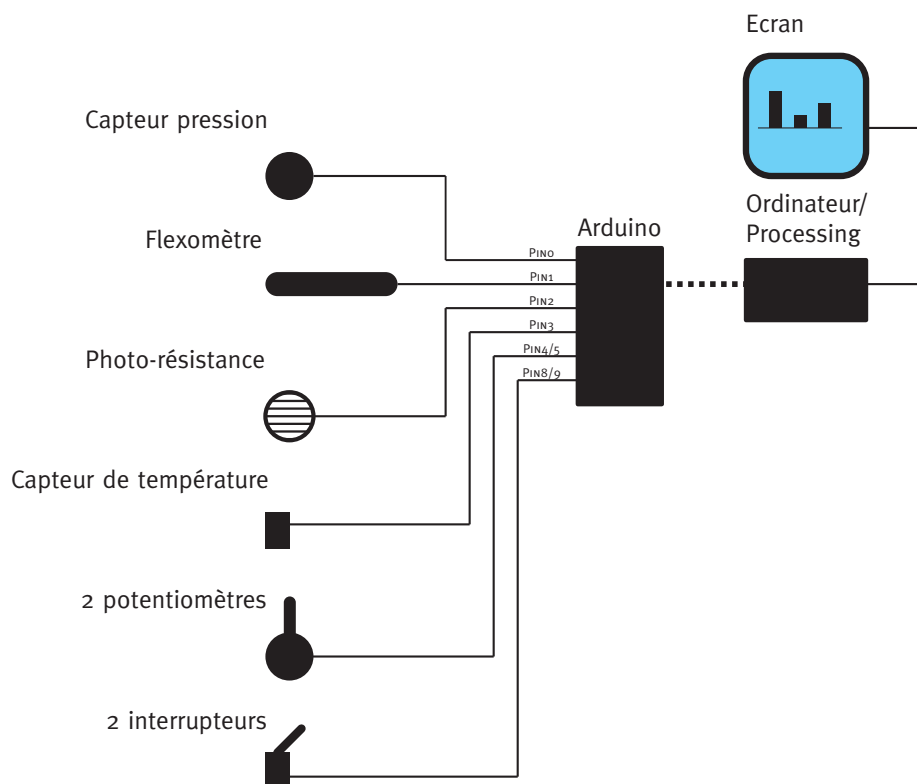
—

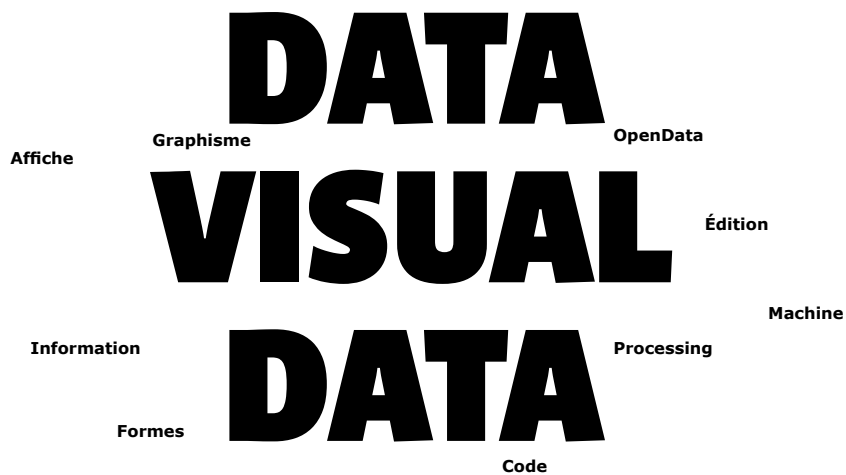
Visualiser des Interfaces

L'exercice est de visualiser graphiquement des capteurs sensoriels afin d'en comprendre le fonctionnement, par groupe de 3 vous devez récupérer les données depuis des capteurs afin de les mettre en forme sur l'écran.

Les UV sont répartie en trois points :

- La présence au cours.
- L'aspect technique de la réalisation.
- La forme graphique du programme.





Texte original de DotWalk

```
// Classic Walk  
Repeat  
{  
  1 st street left  
  2 st street right  
  3 st street left  
}
```

Traduction française en Processing

```
//Marche classique  
void marche(){  
  1ère_rue_a_gauche();  
  2ème_rue_a_droite();  
  3ème_rue_a_gauche();  
  marche(); //recommencer  
}
```

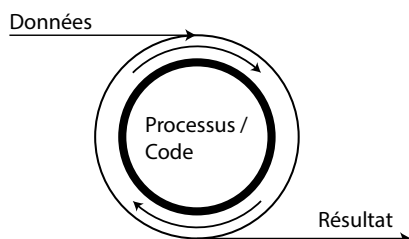
Wilfred Hou Je Bek, *Walk for dummies*, consultable sur SocialFiction,
<http://www.socialfiction.org/dotwalk/dummies.html>

Exercice 1 sous Processing.

```
void setup(){  
  size(displayWidth,displayHeight);  
}  
void draw(){  
  background(int(random(255)),int(random(255)),int(random(255)));  
}
```

WorkShop DEC. TOULOUSE 2013,

Contraintes :



Calendrier :

Semaine de Décembre :

Mardi 3 Décembre au Vendredi 6 Décembre

Ce workshop est un prétexte à mettre au point une mécanique, via la programmation, devant générer des formes graphiques à partir de données numériques (dit «data»). Ces données, impérativement numériques, peuvent être des listes de nombres, des tableaux, des textes, des images, du son ou même de la vidéo.

A partir de cette matière vous élaborerez un procédé permettant de produire du graphisme. Votre production sera amenée à être diffusée sous une de ces formes :

-Affiche, livre, cartographie, site web, ...

Pour cela une formation non exhaustive sur «Processing» vous sera enseignée.

Site Internet :

<http://processing.org/>
<http://www.creativeapplications.net/>
<http://www.visualcomplexity.com/>

Livres :

Paloque-Bergès Camille,
Poétique des codes sur le réseau informatique,
Paris, Éditions des archives contemporaines 2009
193 pages, Broché
9782914610704

David Olivier-Lartigaud,
Art++,
Paris, HXX, 2011,
456 pages, Broché
978-2910385514

Manuel Lima,
Visual Complexity Mapping Patterns of Information
New York, Princeton Architectural Press, 2009
272 pages, Broché
978-1-56898-936-5

John Maeda,
De la simplicité
Dijon, Petit Bilibliothèque Payot, 2009
192 pages, broché
978-2-228-90432-2

Jean-Michel Gériidan, Jean-Noël Lafarge,
Processing, le code informatique comme outil de création
Paris, Pearson, 2011
296 pages, broché
978-2-7440-2453-5

David McCandless,
Information is beautiful,
Londre, Collin, 2009
260 pages, broché
978-0-00-729466-4

David McCandless,
Datavision,
Paris, Robert Laffond, 1011
132 pages, brochés
978-2-221-12-675-2

C.N.A.P. (Centre National des Arts Plastiques),
Graphisme en France / Code <> outils <> Design
Paris, C.N.A.P., 2012
48 pages, piqûre-à-cheval
ISSN 1286-2584

Peter Wildbur et Michael Burke,
Le Graphisme d'information,
Carte, Diagrammes, Interfaces et Signalétique
Paris, Thames & Hudson, 2001

NORM,
The things,
Zurich, NORM, 2002
240 pages, broché double
3-931126-75-7

Hartmut Bohnacker, Benedikt Gross, Julia Laub, and Claudius Lazzaroni,
Generative Design,
, Princeton Architectural, 2012
472 pages, broché
978-1616890773

Edwin A.Abbott,
FlatLand,
Paris, Zones Sensibles, 2012
164 pages, broché
978-2-93-060-105-2

Daniel Shiffman
The Nature of Code: Simulating Natural Systems with Processing
Berlin, The Nature of Code, 2012
520 pages, broché et numérique
978-0985930806

Ben Fry,
Visualizing Data: Exploring and Explaining Data with the Processing Environment
Sebastopol, O'Reilly, 2008
384 pages, broché
978-0596514556

SOUND IS VISUAL

Graphisme
Affiche
mp3
Information
Formes
OpenData
Édition
Son
Machine
Processing
Code

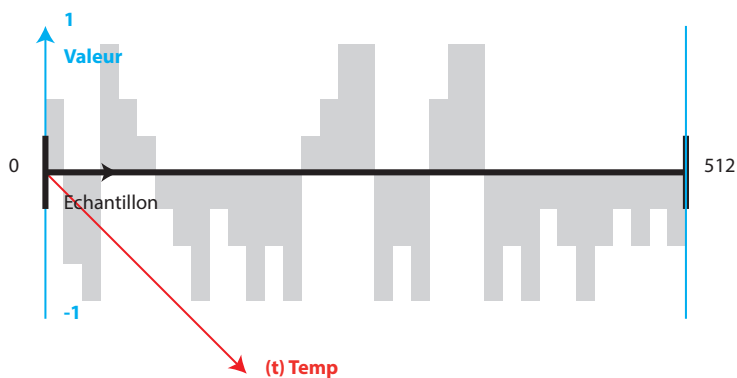
Texte original de DotWalk

Exercice 1 sous Processing.

```
// Classic Walk
Repeat
{
  1 st street left
  2 st street right
  3 st street left
}
```

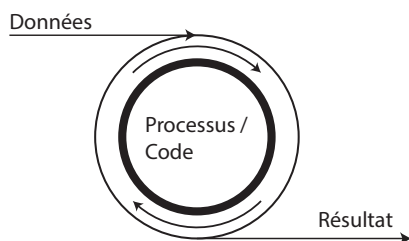
```
void setup(){
  size(displayWidth,displayHeight);
}
void draw(){
  background(int(random(255)),
    int(random(255)),int(random(255))
  );
}
```

Wilfred Hou Je Bek, *Walk for dummies*, consultable sur SocialFiction,
<http://www.socialfiction.org/dotwalk/dummies.html>



WorkShop Renne,

Contraintes :



Calendrier :

Semaine 1 - Lancement du projet :

Mardi 8 octobre : 14h > 17h (lancement sujet) (3h)

Semaine 2 - Apprentissage de Processing :

Mardi 15 octobre : 14h > 18h (4h)

Mercredi 16 Octobre : 14h > 18h (4h)

Jeudi 17 Octobre : 10h15 > 12h15 ; 14h > 18h (6h)

Semaine 3 - Aide individuelle / Rendu de Projet :

Mercredi 6 Novembre : 14h > 18h (4h)

Jeudi 7 Novembre : 10h15 > 12h15 ; 14h > 18h (6h)

Vendredi 8 Novembre 8h15 > 12h15 - 14h > 17h (7h)

Ce workshop est un prétexte à mettre au point une mécanique, via la programmation, devant générer des formes graphiques à partir de données numériques (dit «data»). Ces données, impérativement numériques proviendront de fichiers sonores. Le flux sonore est un flux continu générant un grand nombre de données.

A partir de cette matière vous élaborerez un procédé permettant de produire du graphisme. Votre production sera amenée à être diffusée sous une de ces formes :

-Affiche, livre, cartographie, ...

Pour cela une formation non exhaustive sur «Processing» vous sera enseignée.

Site Internet :

<http://processing.org/>
<http://www.creativeapplications.net/>
<http://www.visualcomplexity.com/>

Livres :

Paloque-Bergès Camille,
Poétique des codes sur le réseau informatique,
Paris, Éditions des archives contemporaines 2009
193 pages, Broché
9782914610704

David Olivier-Lartigaud,
Art++,
Paris, HXX, 2011,
456 pages, Broché
978-2910385514

Manuel Lima,
Visual Complexity Mapping Patterns of Information
New York, Princeton Architectural Press, 2009
272 pages, Broché
978-1-56898-936-5

John Maeda,
De la simplicité
Dijon, Petit Bliothèque Payot, 2009
192 pages, broché
978-2-228-90432-2

Jean-Michel Gériidan, Jean-Noël Lafarge,
Processing, le code informatique comme outil de création
Paris, Pearson, 2011
296 pages, broché
978-2-7440-2453-5

David McCandless,
Information is beautiful,
Londre, Collin, 2009
260 pages, broché
978-0-00-729466-4

David McCandless,
Datavision,
Paris, Robert Laffond, 2011
132 pages, brochés
978-2-221-12-675-2

C.N.A.P. (Centre National des Arts Plastiques),
Graphisme en France / Code <> outils <> Design
Paris, C.N.A.P., 2012
48 pages, piqûre-à-cheval
ISSN 1286-2584

Peter Wildbur et Michael Burke,
Le Graphisme d'information,
Carte, Diagrammes, Interfaces et Signalétique
Paris, Thames & Hudson, 2001

NORM,
The things,
Zurich, NORM, 2002
240 pages, broché double
3-931126-75-7

Hartmut Bohnacker, Benedikt Gross, Julia Laub, and Claudius Lazzeroni,
Generative Design,
Princeton Architectural, 2012
472 pages, broché
978-1616890773

Edwin A. Abbott,
FlatLand,
Paris, Zones Sensibles, 2012
164 pages, broché
978-2-93-060-105-2

Daniel Shiffman
The Nature of Code: Simulating Natural Systems with Processing
Berlin, The Nature of Code, 2012
520 pages, broché et numérique
978-0985930806

Ben Fry,
Visualizing Data: Exploring and Explaining Data with the Processing Environment
Sebastopol, O'Reilly, 2008
384 pages, broché
978-0596514556

Affiche

Modules

Formes

Matériaux

Ville

IsoXotico

Dessin

Image

Paysage

Rendu

3d

Phase 1 : Collecter une collection d'image/de forme

Phase 2 : Dessiner la composition de son illustration d'une manière globale

Phase 3 : Modéliser / rendre / exécuter des tests de texture

Phase 4 : Faire une ou plusieurs affiches (format A2 minimum) présentant sa vision de l'exotisme

IsoXotico est un sujet d'illustration où vous allez devoir utiliser un logiciel 3d. (Blender / Rhino). Il s'agit de concevoir un paysage en trois dimensions isométrique. La perspective cavalière permet une vision utopique sans point de fuite. Le registre visuel doit appartenir à l'exotisme, à l'imaginaire d'un lieu qu'on ne connaît pas ou qui n'existe pas.



Jeux :

2005-2014 OpenTTD Team,
Open Transport Tycoon,
www.openttd.org

Tokyo 42,
<http://www.tokyo42.com/>

Sony Game,
echochrome II
<https://www.playstation.com/fr-lu/games/echochrome-ii-ps3/>

plethora project,
Block'Hood
<http://www.plethora-project.com/blockhood/>

ustwo
monument Valley
www.monumentvalleygame.com

Software Creation Ltd,
Solstice

Fill Fish
FEZ
POLYTRON

Livres :

Peter Wildbur et Michael Burke,
Le Graphisme d'information, Carte, Diagrammes, Interfaces et Signalétique
Paris, Thames & Hudson, 2001

Hartmut Bohnacker,
Benedikt Gross, Julia Laub, and Claudius Lazzaroni,
Generative Design,
Princeton Architectural, 2012
472 pages, broché

Edwin A. Abbott,
FlatLand,
Paris, Zones Sensibles, 2012
164 pages, broché

NORM,
The things,
Zurich, NORM, 2002
240 pages, broché double

Daniel Shiffman
The Nature of Code:

Simulating Natural Systems with Processing
Berlin, The Nature of Code, 2012
520 pages, broché et numérique

Ben Fry,
Visualizing Data: Exploring and Explaining Data with the Processing Environment
Sebastopol, O'Reilly, 2008
384 pages, broché

David McCandless,
Datavision,
Paris, Robert Laffond, 1011
132 pages, brochés

Collections

Modules

Modèles

Data

Livre

dataBook

Dessin

Paysage

Textes

Film

Grille

Phase 1 : Sélectionner des datas : film(s)/image(s)/texte(s)/flux twitter-tumblr-autres...
Phase 2 : Dessiner la composition probable de livre automatique à partir de ces sources
Phase 3 : Faire une proposition de livre automatique. (A5 max)
Phase 4 : Présenter son programme / Imprimer

Data Book est un programme de livre automatique ; il récupère automatiquement des sources ; il transforme, remixe, donne un sens à un flow. Il peut prendre automatiquement des séquences d'un film, crée des collages aléatoire d'une collection d'image, mélanger des textes, écrire l'oscillographe d'une musique, etc ...



Livres :

Peter Wildbur et Michael Burke,
Le Graphisme
d'information, Carte,
Diagrammes, Interfaces
et Signalétique
Paris, Thames & Hudson, 2001

Hartmut Bohnacker,
Benedikt Gross, Julia
Laub, and Claudius
Lazzeroni,
Generative Design,
Princeton Architectural, 2012
472 pages, broché

Oulipo,
Atlas de littérature
potentielle,
Paris, Gallimart, 1988
432 pages, broché

Raymond Queneau
Cent mille milliards de
poème
Paris, Gallimart, 1961
38 pages, Relié

Jean-Michel Géri-dan,
Jean-Noël Lafarge,
Processing, le code
informatique comme

outil de création
Paris, Pearson, 2011
296 pages, broché

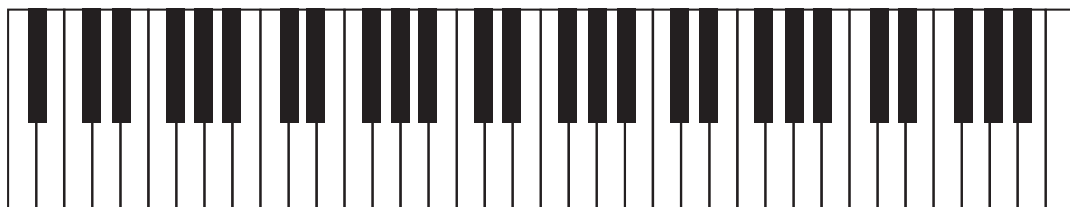
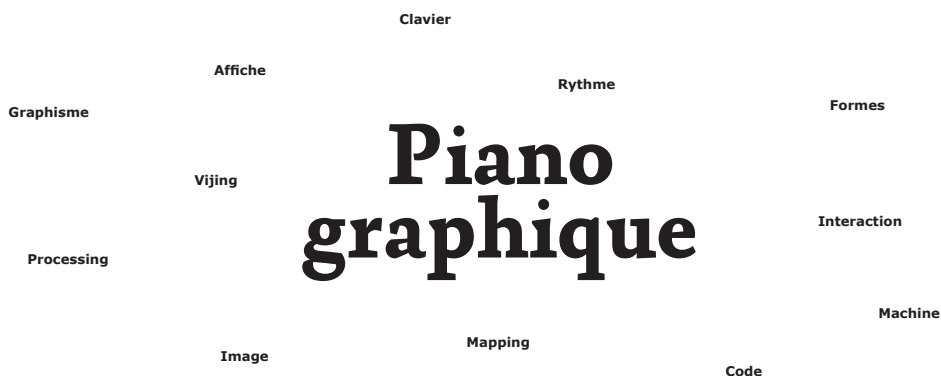
David McCandless,
Information is beautiful,
Londre, Collin, 2009, 260 pages,
broché

C.N.A.P. (Centre National
des Arts Plastiques),
Graphisme en France /
Code<>outils<>Design
Paris, C.N.A.P., 2012
48 pages, piqure-à-cheval

NORM,
The things,
Zurich, NORM, 2002
240 pages, broché double

Paloque-Bergès Camille,
Poétique des codes sur
le réseau informatique,
Paris, Éditions des archives
contemporaines 2009, 193
pages, Broché

David Olivier-Lartigaud,
Art++,
Paris, HYX, 2011, 456 pages,
Broché



Exercice 1 sous Processing

[www.processing.org/download]

```
void setup(){  
    size(displayWidth,displayHeight);  
}  
void draw(){  
    background(int(random(255)),int(random(255)),int(random(255)));  
}
```

Workshop Rennes nov.déc. 2016 - Damien Baïs et Fabrice Sabatier - .Corp

Le piano graphique est un instrument qui génère du graphisme par interaction avec les touches d'un clavier (d'ordinateur ou de piano). Vous devrez penser, programmer et accorder un piano graphique qui génèrera un dispositif graphique (affiches, des livres, des cartes, etc.). Pour la justesse de l'instrument, un univers graphique cohérent sera défini au préalable. Certaines séances seront réservées à l'enseignement de la programmation informatique.

SITOGRAPHIE

Processing, *en ligne*. *consulté le 08 octobre 2016*. Disponible sur : < <http://www.processing.org/> >

Creative applications, *en ligne*. *consulté le 08 octobre 2016*. Disponible sur : < <http://www.creativeapplications.net/> >

Visual Complexity, *en ligne*. *consulté le 08 octobre 2016*. Disponible sur : < <http://www.visualcomplexity.com/> >

Yugop, *en ligne*. *consulté le 08 octobre 2016*. Disponible sur : < <http://www.yugop.com/> >

Mathieu Badimon, *en ligne*. *consulté le 08 octobre 2016*. Disponible sur : < <http://lab.mathieu-badimon.com/> >

Levitated, *en ligne*. *consulté le 08 octobre 2016*. Disponible sur : < <http://www.levitated.net/daily/index.html> >

False, *en ligne*. *consulté le 08 octobre 2016*. Disponible sur : < <http://www.false.jp/> >

Andre Michelle, *en ligne*. *consulté le 08 octobre 2016*. Disponible sur : < <http://lab.andre-michelle.com/moonpatrol/> >

Incomplet, *en ligne*. *consulté le 08 octobre 2016*. Disponible sur : < <http://incomplet.gskinner.com/> >

OUVRAGES :

Paloque-Bergès Camille,
Poétique des codes sur le
réseau informatique,
Paris, Editions des archives contem-
poraines 2009
193 pages, Broché

David Olivier-Lartigaud,
Art++,
Paris, HX, 2011,
456 pages, Broché

Manuel Lima,
Visual Complexity
Mapping Patterns of Infor-
mation
New York, Princeton Architectural
Press, 2009
272 pages, Broché

John Maeda,
De la simplicité
Dijon, Petit Bibliothèque Payot,
2009
192 pages, broché

**Jean-Michel Gérardan, Jean-
Noël Lafarge,**
Processing, le code infor-
matique comme outil de
création
Paris, Pearson, 2011
296 pages, broché

David McCandless,
Information is beautiful,
Londre, Collin, 2009
260 pages, broché

David McCandless,
Datavision,
Paris, Robert Laffond, 2011
132 pages, brochés

**C.N.A.P. (Centre National
des Arts Plastiques),**
Graphisme en France /
Code<>outils<>Design
Paris, C.N.A.P., 2012
48 pages, piqûre-à-cheval

**Peter Wildbur et Michael
Burke,**
Le Graphisme
d'information
Carte, Diagrammes, Interfaces et
Signalétique
Paris, Thames & Hudson, 2001

NORM,
The things,
Zurich, NORM, 2002
240 pages, broché double

**Hartmut Bohnacker,
Benedikt Gross, Julia Laub,
and Claudius
Lazzeroni,**
Generative Design,
Princeton Architectural, 2012
472 pages, broché

Edwin A.Abbott,
FlatLand,
Paris, Zones Sensibles, 2012
164 pages, broché

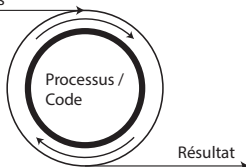
Daniel Shiffman

The Nature of Code: Simu-
lating Natural
Systems with Processing
Berlin, The Nature of Code, 2012
520 pages, broché et numérique

Ben Fry,
Visualizing Data: Exploring
and Explaining Data with
the Processing Environ-
ment
Sebastopol, O'Reilly, 2008
384 pages, broché

Contraintes :

Données



Textes

Téléphones

Filtre

Application

Modèles

Data

Portrait

Collections

Grille

Film

Le(s) filtre(s) photographique(s)

Phase 1 : Collectionner des filtres existants.

Phase 2 : Dessiner la composition probable de l'interface graphique de votre application.

Phase 3 : Programmer l'appareil.

Phase 4 : Prendre des photos.

A la manière des filtres "Snapchat" ou "Instagram", ou d'autres applications de prise de vue sur un téléphone portable, vous devez concevoir un programme qui capture la réalité et ajoute un ou des artefacts. Les champs d'exploration et les thématiques de ces filtres peuvent être très diverses.

L'application peut intervenir sur l'image en général ou des parties précises de celle-ci.



Site Internet :

processing.org
www.creativeapplications.net
www.visualcomplexity.com

Livres :

David Olivier-Lartigaud,
Art++,
Paris, HYX, 2011, 456 pages,
Broché

Manuel Lima,
Visual Complexity
Mapping Patterns of
Information

New York, Princeton Architectural
Press, 2009, 272 pages, Broché

John Maeda,
De la simplicité

Dijon, Petit Bilibliothèque Payot,
2009, 192 pages, broché

Jean-Michel Géridan,
Jean-Noël Lafarge,
Processing, le code
informatique comme
outil de création

Paris, Pearson, 2011
296 pages, broché

David McCandless,
Information is beautiful,

Londre, Collin, 2009, 260 pages,
broché

Peter Wildbur et Michael Burke,
Le Graphisme
d'information, Carte,
Diagrammes, Interfaces
et Signalétique

Paris, Thames & Hudson, 2001

Hartmut Bohnacker,
Benedikt Gross, Julia
Laub, and Claudius
Lazzeroni,
Generative Design,
Princeton Architectural, 2012
472 pages, broché

NORM,
The things,

Zurich, NORM, 2002
240 pages, broché double

Daniel Shiffman
The Nature of Code:
Simulating Natural
Systems with Processing

Berlin, The Nature of Code, 2012
520 pages, broché et numérique

Frédéric Kaplan, Nicolas Nova,
La culture internet des
mêmes

Lausanne : Presses
polytechniques et universitaires
romandes
94 pages, broché

Collections

Modules

Livre

Modèles

Dessin

Data

Poésie automatique

Paysage

Textes

Film

Grille

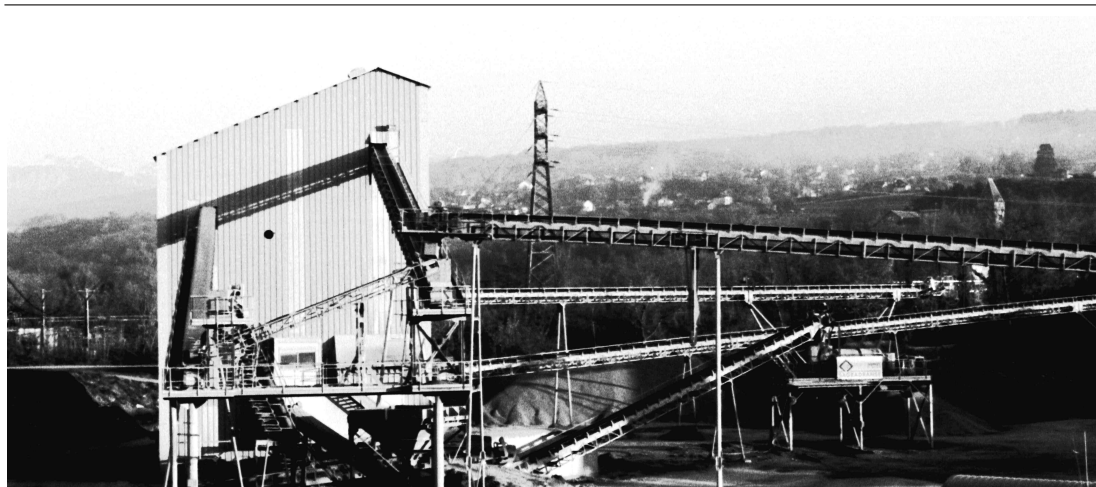
Phase 1 : Collectionner images/textes sur un sujet.

Phase 2 : Dessiner la composition probable de livre automatique à partir de texte.

Phase 3 : ~~Proposer~~, programmer de la poésie automatique. (A4 max)

Phase 4 : Présenter son programme / Imprimer

Poésie automatique est un programme qui génère de la poésie ; il récupère automatiquement des sources ; il transforme, remixe. Il peut prendre automatiquement des morceaux de texte, créer des collages, des assemblages de mot. A chaque fois que le programme s'exécute, il produit un livret de 12 pages minimum.



Livres :

Peter Wildbur et Michael Burke,
Le Graphisme
d'information, Carte,
Diagrammes, Interfaces
et Signalétique
Paris, Thames & Hudson, 2001

Hartmut Bohnacker,
Benedikt Gross, Julia
Laub, and Claudius
Lazzeroni,
Generative Design,
Princeton Architectural, 2012
472 pages, broché

Oulipo,
Atlas de littérature
potentielle,
Paris, Gallimart, 1988
432 pages, broché

Raymond Queneau
Cent mille milliards de
poème
Paris, Gallimart, 1961
38 pages, Relié

Jean-Michel G ridan,
Jean-No l Lafarge,
Processing, le code
informatique comme

outil de cr ation
Paris, Pearson, 2011
296 pages, bro  

David McCandless,
Information is beautiful,
Londre, Collin, 2009, 260 pages,
bro  

C.N.A.P. (Centre Nation-
nal des Arts Plastiques),
Graphisme en France /
Code<>outils<>Design
Paris, C.N.A.P., 2012
48 pages, piqu re- -cheval

NORM,
The things,
Zurich, NORM, 2002
240 pages, bro   double

Paloque-Berg s Camille,
Po tique des codes sur le
r seau informatique,
Paris,  ditions des archives
contemporaines 2009, 193 pages,
Bro  

David Olivier-Lartigaud,
Art++,
Paris, HYG, 2011, 456 pages,
Bro  

Textes

Filtre

Science-fiction

Rendu

Modèles

3D

Texture

Portrait

Uchronie

Film

Uchrono-vision / Futuro-vision

Phase 1 : Voir un ou plusieurs films dans la filmographie

Phase 2 : Dessiner/imaginer une partie de monde probable du film

Phase 3 : Modéliser / rendre / exécuter sous blender

Phase 4 : Faire une ou plusieurs affiches (format A2 minimum) présentant d'un futur probable

Uchrono-vision / Futuro-vision est un sujet d'illustration où vous allez devoir utiliser un logiciel 3d. (Blender / Rhino). Il s'agit de concevoir un paysage/un lieu en trois dimensions reprennant l'esthétique, la vision d'un des films dans la filmographie ci-dessous.



Film :

Roger Vadim
Barbarella

1978, Franco-Italien
Production Dino De Laurentiis
Cinematografica et Marianne Productions

David Lynch
Dune

1982, Etats-Unis
Production 20th Century Fox,
Universal Pictures

Stanley Kubrick
2001, l'odyssée de l'espace

1968, Royaume-Uni, Etats-Unis
Metro-Goldwyn-Mayer

Ridley Scott
Blade Runner

1982, Etats-Unis
Production The Ladd Company

Richard Fleischer
Soleil vert (Soylent Green)

1973, Etats-Unis
Production Metro-Goldwyn-Mayer

George Lucas
THX 1138

1971, Etats-Unis
American Zoetrope
Production

Michael Crichton
Mondwest

1973, Etats-Unis
Production Metro-Goldwyn-Mayer

Chris Marker
La Jetée

1962, France
Production Argos Films

John Boorman
Zardoz

1974, Royaume-Uni
Production John Boorman pour
John Boorman Productions

George Miller
Mad Max II

1981, Australie
Production Kennedy Miller
Productions

Andrei Tarkovski
Solaris

1972, Union soviétique
Production Mosfilm

Sujet de 3ème Année

Option : Design numérique / Cultures électroniques

Ref : Damien BAÏS / Random(lab)

Météo sensible

À partir de donnée météo, vous devrez imaginer une interface pouvant représenter un bulletin de prévision météorologique.

Cette interface doit posséder un vocabulaire de forme suffisant pour permettre de visualiser au moins huit états, soit au minimum :

Ensoleillé, nuageux, pluie, brouillard, neige, vent, gèle, tempête, ...

Elle peut aussi donner une représentation de la température, de la direction et la vitesse du vent, de l'humidité, du taux d'U.V., etc.

Le dispositif peut-être un programme informatique, une application, un objet, une affiche, un dispositif artistique, une musique, ...

Le dispositif, même s'il n'est pas fonctionnel, doit pouvoir indiquer une forme de variation.

Certaines séances seront réservées à l'enseignement de la programmation informatique.

Site Internet :

<http://processing.org/>
<http://www.creativeapplications.net/>
<http://www.visualcomplexity.com/>

Livres :

Paloque-Bergès Camille,
Poétique des codes sur le réseau informatique,
Paris, Éditions des archives contemporaines 2009
193 pages, Broché
9782914610704

David Olivier-Lartigaud,
Art++,
Paris, HYX, 2011,
456 pages, Broché
978-2910385514

Manuel Lima,
Visual Complexity Mapping Patterns of Information
New York, Princeton Architectural Press, 2009
272 pages, Broché
978-1-56898-936-5

John Maeda,
De la simplicité
Dijon, Petit Bibliothèque Payot, 2009
192 pages, broché
978-2-228-90432-2

Jean-Michel Gérihan, Jean-Noël Lafarge,
Processing, le code informatique comme outil de création
Paris, Pearson, 2011
296 pages, broché
978-2-7440-2453-5

David McCandless,
Information is beautiful,
Londre, Collin, 2009
260 pages, broché
978-0-00-729466-4

David McCandless,
Datavision,
Paris, Robert Laffont, 2011
132 pages, brochés
978-2-221-12-675-2

C.N.A.P. (Centre National des Arts Plastiques),
Graphisme en France / Code > outils < Design
Paris, C.N.A.P., 2012
48 pages, piqure-à-cheval
ISSN 1286-2584

Peter Wildbur et Michael Burke,
Le Graphisme d'information,
Carte, Diagrammes, Interfaces et Signalétique
Paris, Thames & Hudson, 2001

NORM,
The things,
Zurich, NORM, 2002
240 pages, broché double
3-931126-75-7

Hartmut Bohnacker, Benedikt Gross, Julia Laub, and Claudio Lazzaroni,
Generative Design,
Princeton Architectural, 2012
472 pages, broché
978-1616890773

Edwin A. Abbott,
FlatLand,
Paris, Zones Sensibles, 2012
164 pages, broché
978-2-93-060-105-2

Daniel Shiffman
The Nature of Code: Simulating Natural Systems with Processing
Berlin, The Nature of Code, 2012
520 pages, broché et numérique
978-0985930806

Ben Fry,
Visualizing Data: Exploring and Explaining Data with the Processing Environment
Sebastopol, O'Reilly, 2008
384 pages, broché
978-0596514556

Écran

Data

Dispositif

Balise

Programme

3D

Météo

Climat

Futur

Graphisme

Global Warming/ Local Warning

Phase 1 : Dessiner collectivement un dispositif de collecte météorologique (Station météo)

Phase 2 : Réaliser collectivement le dispositif

Phase 3 : Imaginer / dessiner un programme de visualisation

Phase 4 : Programmer individuellement un visualisation de collecte de données.

Dans un contexte de réchauffement climatique, cette pièce doit mettre en valeur les datas météorologiques comme indicateur d'un environnement local et/ou global. Elle peut être un objet fictionnel émettant des données/signaux et/ou virtuels/réels.



David Olivier-Lartigaud,
Art++,
Paris, HXX, 2011, 456 pages,
Broché

Manuel Lima,
Visual Complexity
Mapping Patterns of
Information
New York, Princeton
Architectural Press, 2009, 272
pages, Broché

Peter Wildbur et Michael Burke,
Le Graphisme
d'information, Carte,
Diagrammes, Interfaces et
Signalétique
Paris, Thames & Hudson, 2001

**Hartmut Bohnacker,
Benedikt Gross, Julia
Laub, and Claudius
Lazzeroni,**
Generative Design,
Princeton Architectural, 2012
472 pages, broché

NORM.
The things,
Zurich, NORM, 2002
240 pages, broché double

Daniel Shiffman
The Nature of Code:
Simulating Natural
Systems with Processing
Berlin, The Nature of Code, 2012
520 pages, broché et numérique

**Jean-Michel Géri dan,
Jean-Noël Lafarge,**
Processing, le code
informatique comme outil
de création
Paris, Pearson, 2011
296 pages, broché

David McCandless,
Information is beautiful,
Londre, Collin, 2009, 260 pages,
broché

**C.N.A.P. (Centre National
des Arts Plastiques),**
Graphisme en France /
Code<outils>Design
Paris, C.N.A.P., 2012
48 pages, piqure-à-cheval

Paloque-Bergès Camille,
Poétique des codes sur le
réseau informatique,
Paris, Éditions des archives
contemporaines 2009, 193 pages,
Broché

**Sandra Rendgen, Julius
Wiedemann**
Sur la piste des
informations
Paris, Tachen, avec 6 pages
dépliantes, 462 pages, Relié

Andy Kirk
Data Visualisation: A
Handbook for Data Driven
Design
SAGE Publications Ltd 2016, 368
pages, Relié

Bruit

Data

Dispositif

Musique

Programme

3D

Vidéo

Temps-réel

Interface

Graphisme

L'image sonore/ Vjing

Phase 1 : Collecter des sons / bruits / musiques

Phase 2 : Relier son projet à des références artistiques / sonores / scientifique

Phase 3 : Imaginer / dessiner un programme de visualisation

Phase 4 : Programmer un prototype

La traduction sonore en image générée en temps réel est souvent effectuée par des logiciels dédiés. Bien souvent, les traductions graphiques sont génériques et rarement bien adaptées à la musique. Pour ce sujet, je vous invite à réfléchir à une traduction contextualisée suivant le son/musique/bruit que vous avez préalablement choisi.



David Olivier-Lartigaud,
Art++,
Paris, HYX, 2011, 456 pages,
Broché

Manuel Lima,
Visual Complexity
Mapping Patterns of
Information
New York, Princeton
Architectural Press, 2009, 272
pages, Broché

Peter Wildbur et Michael Burke,
Le Graphisme
d'information, Carte,
Diagrammes, Interfaces et
Signalétique
Paris, Thames & Hudson, 2001

**Hartmut Bohnacker,
Benedikt Gross, Julia
Laub, and Claudius
Lazzeroni,**
Generative Design,
Princeton Architectural, 2012
472 pages, broché

NORM.
The things,
Zurich, NORM, 2002
240 pages, broché double

Daniel Shiffman
The Nature of Code:
Simulating Natural
Systems with Processing
Berlin, The Nature of Code, 2012
520 pages, broché et numérique

**Jean-Michel Gériidan,
Jean-Noël Lafarge,**
Processing, le code
informatique comme outil
de création
Paris, Pearson, 2011
296 pages, broché

David McCandless,
Information is beautiful,
Londre, Collin, 2009, 260 pages,
broché

**C.N.A.P. (Centre National
des Arts Plastiques),
Graphisme en France /
Code<outils>Design**
Paris, C.N.A.P., 2012
48 pages, piqure-à-cheval

Paloque-Bergès Camille,
Poétique des codes sur le
réseau informatique,
Paris, Éditions des archives
contemporaines 2009, 193 pages,
Broché

**Sandra Rendgen, Julius
Wiedemann**
Sur la piste des
informations
Paris, Tachen, avec 6 pages
dépliantes, 462 pages, Relié

Andy Kirk
Data Visualisation: A
Handbook for Data Driven
Design
SAGE Publications Ltd 2016, 368
pages, Relié

Mémo de programmation [processing]

Pratiques numériques / Cultures électroniques

Mémo présentant les fonctions courantes sous processing. Vous pouvez télécharger la version imposé ou non-imposé. Il est hébergé sur github, libre de droit, téléchargeable et imprimable gratuitement.

https://github.com/randomDam/memo_processing

MEMO PROCESSING

Version Conseillé
Processing V2.1

RANDOM(LAB) | ESADSE | VERSION 0.2 | SEPTEMBRE 2013 | WWW.PROCESSING.ORG

Petit mémo (non exhaustif) des fonctions à l'usage des Graphistes, Artistes et Designers

COMMENTAIRES

// Une ligne de commentaire

/*
Un paragraphe de commentaire
*/

INITIALISATION D'UNE SCÈNE (FONCTION DE BASE)

```
void setup(){ //Une seule fois au début }  
void draw(){ //Tout le temps en boucle }
```

FONCTIONS DE CONSTRUCTION

// Taille de la fenêtre
size(l,h); //où l et h est un int

//Nombre d'images/sec
frameRate(c); //où c est un int

FONCTIONS : ÉVÈNEMENT SOURIS & CLAVIER

//Quand une touche est appuyée (type KeyEvent)
void keyPressed(){
}
//Quand une touche est relâchée (type KeyEvent)
void keyReleased(){
}
//Quand la souris est cliquée (type MouseEvent)
void mouseClicked(){
}
//Quand la souris est appuyée (type MouseEvent)
void mousePressed(){
}
//Quand la souris est relâchée (type MouseEvent)
void mouseReleased(){
}
//Quand la souris est en mouvement (type MouseEvent)
void mouseMoved(){
}
//Quand la souris est en mouvement+appuyée (type MouseEvent)
void mouseDragged(){
}

VARIABLES

//où bob est le nom que vous souhaitez

```
int bob1=5;  
float bob2=5.545;  
char bob3='A';  
boolean bob4= true;  
color bob5= color(250,231,32);  
String bob6="Hello World!";
```

VARIABLES GLOBALES

```
width // largeur de la fenêtre  
height // hauteur de la fenêtre  
mouseX // position en X de la souris  
mouseY // position en Y de la souris  
pmouseX // position en X de la souris à l'image précédente  
pmouseY // position en Y de la souris à l'image précédente  
frameCount // nombre d'images depuis le début  
frameRate // cadence d'images/seconde  
displayWidth // largeur de l'écran  
displayHeight // hauteur de l'écran
```

DANS LES "MOUSEEVENT" :

mouseButton //Correspond au bouton de la souris
//Peut être égal à LEFT ou RIGHT

DANS LES "KEYBOARDEVENT" :

keyCode //Correspond au int unique d'une touche clavier
// par exemple 32 pour la touche ESPACE.

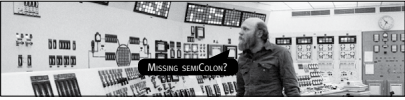
key //Correspond à la touche clavier char

CONSOLE

```
println("Hello World");  
println("Hello World");
```

ASTUCE :

//println peut recevoir tout type de variable.
//Quelques exemples :
println("frame : "+frameCount);
println("valeur : "+bob);
println("Hello "+ "my name is "+" Pat!");
println("Hello "+ random(100);



FONCTION DE DESSIN - GÉNÉRAL

FORMES :

/* Ici,
x est un float - position en x,
y est un float - position en y,
z est un float - position en z,
l est un float - largeur de la forme,
h est un float - hauteur de la forme,
*/

```
point(x,y); //un point  
point(x,y,z); //un point  
rect(x,y,l,h); //un rectangle  
rect(x,y,l,h,r); //un rectangle arrondis de r  
rect(x,y,l,h,r1,r2,r3,r4);  
ellipse(x,y,l,h); //une ellipse  
ellipse(x,y,l,h); //une ellipse 2D  
line(x1,y1,x2,y2); //une ligne 2D  
line(x1,y1,x2,y2,z1,z2); //une ligne 3D  
triangle(x1,y1,x2,y2,x3,y3); //un triangle  
quad(x1,y1,x2,y2,x3,y3,x4,y4); //un quadrilatère
```

COULEURS :

/* Ici,
g est un int : valeur de gris,
(255=blanc, 0=noir)
r est un int : quantité de rouge de 0 à 255
g est un int : quantité de vert de 0 à 255
b est un int : quantité de bleu de 0 à 255
a est un int : quantité de transparence de 0 à 255 */

```
fill(g); //remplissage en gris  
fill(g,alpha); //remplissage en gris+alpha  
fill(r,v,b); //remplissage en couleur  
fill(r,v,b,alpha); //remplissage en couleur+alpha
```

ATTRIBUTS :

```
stroke(g); //contour gris  
stroke(g,alpha); //contour gris+alpha  
stroke(r,v,b); //contour couleur  
stroke(r,v,b,alpha); //contour couleur+alpha
```

noFill(); //pas de remplissage
noStroke(); //pas de contour

//ici;
//a peut être égal à ROUND
//a peut être égal à SQUARE
//a peut être égal à PROJECT
strokeCaps(a); //Style des fin de ligne

//ici;
//b peut être égal à MITER
//b peut être égal à ROUND
//b peut être égal à BEVEL
strokeJoin(b); //Style des angles de ligne

OPTION DE MÉLANGE :

```
blendMode(0);  
/* Ajouter un effet de superposition au forme suivante.  
( de type calque de fusion ).  
*/
```

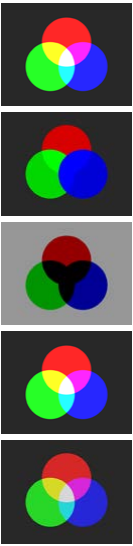
ADD - Mode additif :
C = min(A*factor + B, 255)

SUBTRACT - Mode soustractif : C = max(B - A*factor, 0)

DARKEST - Obscurcir : C = min(A*factor, B)

LIGHTEST - Eclaircir : C = max(A*factor, B)

DIFFERENCE - soustrait les couleurs de l'image sous-jacente.



EXCLUSION - Similaire à différence, mais plus important.

MULTIPLY - Multiplie la couleur.

SCREEN - opposé à MULTIPLIER, utilise la valeur inverse des couleurs.

BLEND - Mode de superposition classique.

EXEMPLE :

blendMode(ADD);

```
fill(255,0,0,150);
ellipse(width/2,height/2,300,2,300,300);
```

```
fill(0,255,0,150);
ellipse(width/2-300/3,height/2,300,300);
```

```
fill(0,0,255,150);
ellipse(width/2+300/3,height/2,300,300);
```

FONCTION DE DESSIN - AVANCÉ

LES VERTEX :

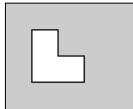
```
/* Ici,
x est la position en x en float
y est la position en y en float
*/
```

```
vertex(x,y); //Un point de polygone
```

```
beginShape(); //le debut du polygone
endShape(); //la fin du polygone
```

Exemple:

```
beginShape();
vertex(20, 20);
vertex(40, 20);
vertex(40, 40);
vertex(60, 40);
vertex(60, 60);
vertex(20, 60);
endShape();
```



Résultat :

LES TRANSFORMATIONS :

```
/* Ici,
x est le décalage en x en float
y est le décalage en y en float
z est le décalage en z en float */
translate(x,y); //décalage en 2D
translate(x,y,z); //décalage en 3D (OPENGL)

// Ici,
// r est le décalage d'angle en RADIANT (PI/1) en float
rotate(0); //décalage d'angle en 2D
rotateX(0); //décalage d'angle X en 3D (OPENGL)
rotateY(0); //décalage d'angle Y en 3D (OPENGL)
rotateZ(0); //décalage d'angle Z en 3D (OPENGL)
```

ASTUCE :

Faire sa rotation **en degrés** et pas en radians!

```
rotate(radians(45)); //une rotation à 45 degrés!
```

```
/* Ici,
a,x,y et z est un float = indice d'agrandissement
si a,x,y,z est plus petit que 1 la forme devient plus petite
si a,x,y,z est plus grand que 1 la forme devient plus grande
si a,x,y,z est égale à 1 la forme à sa taille d'origine */
scale(a); //changement de taille sur tout les axes X-Y-Z

scale(x,y); //chgmt de taille sur X et Y
scale(x,y,z); //chgmt de taille sur X, Y et Z
pushMatrix(); //Initialiser une transformation
popMatrix(); //Finir une transformation
resetMatrix(); //Redéfinir une transformation
```

SAUVEGARDE D'IMAGE

Save & SaveFrame:

```
save();
```

3

PGRAPHICS

DÉCLARATION :

/* PGraphics permet de créer un clip graphique dans lequel on peut dessiner, intégrer des images, faire des manipulations de pixels ou même implémenter de la 3D. Il comprend quasiment toutes les fonctions de PApplet c'est à dire de la scène principale. On l'affiche grâce à la fonction image(); */

PGraphics pg;

```
void setup() {
  size(300, 300);
  pg = createGraphics(40, 40);
  //Crée un objet PGraphic d'une taille de 40 par 40
}
```

```
void draw() {
  pg.beginDraw(); //On commence à dessiner
```

```
  pg.background(100);
  pg.stroke(255);
  pg.line(20, 20, mouseX, mouseY);

  pg.endDraw(); //On termine de dessiner
```

```
  image(pg, 9, 30); //On affiche
  image(pg, 51, 30);
```

```
}
```

HINT - GESTION DE L'AFFICHAGE

DÉCLARATION :

hint(DISABLE_OPENGL_2X_SMOOTH);
//Permet de désactiver le lissage des formes en PSD, voir aussi la fonction smooth();

hint(DISABLE_DEPTH_TEST);
//Désactive le «z-buffer», c'est à dire la profondeur 3D, cette option est utilisée pour par exemple dessiner au dessus de scènes 3D, par exemple mettre des boutons de control au-dessus d'une scène en 3D.

hint(ENABLE_DEPTH_TEST);
//Permet de réactiver l'option 3D.

hint(ENABLE_DEPTH_SORT);
/*Activer primitive «z-sorting» des triangles et des lignes dans PSD et OpenGL. Cela peut ralentir considérablement les performances, et l'algorithme n'est pas encore parfait. Restaurer la valeur par défaut avec un : hint(DISABLE_DEPTH_SORT); */

4

FONCTIONS PERSONNELLES (SANS RETOUR > VOID)

FONCTION SIMPLE :

```
void draw(){
  maFunction(); //utilisation de ma fonction
}
```

```
void maFunction(){
  //mes actions à effectuer.
}
```

FONCTION AVEC PASSAGE D'ARGUMENTS :

```
void draw(){
  maFunction(55); //utilisation de ma fonction
  maFunction(99); //utilisation de ma fonction
}
```

```
void maFunction(int bob){
  //mes actions à effectuer.
  //bob, un int n'existe que dans cette fonction
  rect(20, 20, bob, bob);
  //utilisation de la variable bob
  //Dans un premier cas bob vaut 55
  //Dans un deuxième cas 99
}
```

AUTRE EXEMPLE :

```
void draw(){
  maFunction("Salut",55); //utilisation de ma fonction
  maFunction("Yop!",99); //utilisation de ma fonction
}
```

```
void maFunction(String pat, int bob){
  /* mes actions à effectuer.
  bob, un int et pat un String n'existe que dans
  cette fonction
  */
  text(pat, bob, bob);
  //utilisation de la variable bob et pat
  //Dans un premier cas bob vaut 55 et pat vaut "Salut"
  //Dans un deuxième cas bob vaut 99 et pat vaut "Yop!"
}
```

FONCTIONS PERSONNELLES (AVEC RETOUR)

FONCTION SIMPLE :

```
void draw(){
  int bob = maFunction();
  //Ici bob prend la valeur 5.
}
```

```
int maFunction(){
  //retourne le nombre 5.
  return 5;
}
```

/* Ici on utilise un retour d'entier, mais tout les types sont envisageable comme float, String, boolean, PImage, PGraphics, PShape, Object, ... ou même une classe personnalisée. */

FONCTION AVEC PASSAGE D'ARGUMENTS :

```
/*Par exemple une fonction d'addition
void draw(){
  int bob = maFunction(3,2); //bob vaut 5
  int pat = maFunction(100,5); //pat vaut 105
}
```

```
int maFunction(int a, int b){
  return a + b;
}
```

AUTRE EXEMPLE :

```
void draw(){
  println(maFunction("Salut",55));
  //utilisation de ma fonction avec un retour de chaîne
  de caractère, affiche «Salut55»
  */
  println(maFunction("Yop!",99));
  //utilisation de ma fonction avec un retour de chaîne
  de caractère, affiche «Yop99»
  */
}
```

```
String maFunction(String pat, int bob){
  String couple = at+str(bob);
  return couple;
}
```

5

LES IMAGES (OBJETS BITMAPS) : PIMAGE

/*Emplacement mémoire qui contient mon image pour l'instant vide.

PImage bob;

```
void setup(){
  //Chargement dans mon image dans l'emplacement mémoire
  bob = loadImage("bob_l_eponge.png");
}
void draw(){
  //affichage de mon image
  image(bob,40,40);
}
```

VARIABLES ACCESSIBLES DEPUIS UN OBJET PIMAGE

```
pixels[] // Tableau de pixels contenant la totalité des pixels d'une image.
width // Largeur de l'image.
height // Hauteur de l'image.
//où x est un int et designe la position en x du pixel.
//où y est un int et designe la position en y du pixel.
//où c est un int et designe la nouvelle couleur du pixel.
```

MÉTHODES ACCESSIBLES DEPUIS UN OBJET PIMAGE

```
.loadPixels(); //Charge les pixels de l'image au tableau de pixels accessibles : array[]

.updatePixels(); //Charge les pixels du tableau de pixels[] à l'image.

.mask(image); //Applique un masque alpha à l' image à l'aide d'une autre image.
//où x est un int et designe la position en x du pixel.
//où y est un int et designe la position en y du pixel.
//où ma est un PImage et designe le morceau d'image à coller.

.resize(w,h); //change la taille de l'image en largeur et hauteur
//où w est un int pour la nouvelle largeur
//où h est un int pour la nouvelle hauteur

.filter(type); //Applique un filtre à l' image
//où type est le type de filtre à appliquer.
```

les types possibles :

• Pour une image bob.

```
PImage bob;
void setup(){
  bob = loadImage("monImage.png");
}
```

THRESHOLD

• Converti tout les pixels soit en noir soit en blanc. Le paramètre par défaut est 0.5 c'est à dire que tout les pixels ayant une valeur de gris au dessus de 255*0.5 = 127.5

6

OPÉRATIONS MATHÉMATIQUES	LES TABLEAUX
float bob = 3; float pat = -4.456; float gary = 5.11; VALEURS ABSOLUE : abs() float r = abs(pat); <i>//Retourne la valeur absolue de -4.456 soit 4.456.</i> PUISSANCE : sq() float r = sq(bob); <i>//Retourne la puissance de 2 de bob 3² soit 9.</i> PUISSANCES DE N : pow() float r = pow(bob,4); <i>// ou 4 peut être n'importe quel float</i> <i>// Retourne la puissance de 3 de bob 3x3x3x3 soit 81.</i> RACINE CARRÉE : sqrt() float r = sqrt(bob); <i>// Retourne la racine carré de bob : √3 soit 1.732...</i> TRONCAGE HAUT : ceil() float r = ceil(gary); <i>// Retourne la valeur arrondie haute de 5.11 soit 6.</i> float r = ceil(pat); <i>// Retourne la valeur arrondie haute de -4.456 soit -4.</i> TRONCAGE BAS : floor() float r = floor(gary); <i>// Retourne la valeur arrondie basse de 5.11 soit 5.</i> float r = floor(pat); <i>// Retourne la valeur arrondie basse de -4.456 soit -5.</i> ARRONDIS : round() float r = round(gary); <i>// Retourne l'arrondi basse de 5.11 soit 5.</i> float r = round(pat); <i>/* Retourne l'arrondi basse de -4.456 soit -4.</i> <i>pour un nombre entre 0 et 1,</i> <i>si le nombre est supérieur ou égal à 0.5 donne 1.</i> <i>si le nombre est inférieur à 0.5 donne 0.</i> <i>*/</i> ALÉATOIRE : random() float r = random(59); <i>//tir un nombre aléatoire (un float) de 0 à 59</i> float r = random(22,59); <i>//tir un nombre aléatoire (un float) de 22 à 59</i>	REEMPLISSAGE VIA LE SETUP <i>//Déclare un tableau de int à 6 cases vides de type int</i> int [] bob = new int[6]; <i>//Remplissage du tableau via le setup;</i> void setup(){ bob[0]=9; <i>//a case 0 prend 9</i> bob[1]=81; <i>//a case 1 prend 81</i> bob[2]=111; <i>//a case 2 prend 111</i> } <i>// Et donc les cases 4 et 5 sont égales à 0.</i> REEMPLISSAGE VIA LA DÉCLARATION float [] bob = {65,5,7,67}; <i>/* Ici la case 0 prend 65</i> <i>// Ici la case 1 prend 5.7</i> <i>// Ici la case 2 prend 67</i> <i>// et donc le tableau a une longueur de 3.</i> AUTRES EXEMPLES, AVEC D'AUTRES TYPES String [] tabPays = { "France", "Italie", "Es- pagne", "Allemagne" } <i>/* Ici la case 0 prend «France»</i> <i>ici la case 1 prend «Italie»</i> <i>ici la case 2 prend «Espagne»</i> <i>ici la case 3 prend «Allemagne»</i> <i>et donc le tableau a une longueur de 4.</i> <i>*/</i> boolean [] tabBool = {true,false,true,true,false} <i>// Initialisation d'un tableau de boolean contenant 5 valeurs booléennes.</i> color [] tabCol= (color(255),color(251,15,25), color(255,255,0)); <i>// Initialisation d'un tableau de couleur contenant 3 couleurs.</i> Pimage [] tabImage = new Pimage[4]; <i>// Initialisation d'un tableau pouvant contenir 4 images.</i> void setup(){ tabImage[0]=loadImage («bob.png»); tabImage[1]=loadImage («gary.png»); tabImage[2]=loadImage («pat.png»); } <i>// et tabImage[3] est égal à null puisque c'est des objets.</i>
	11

<i>// [0] "NY"</i> <i>// [1] "CA"</i> <i>// [2] "VA"</i> <i>// [3] "CO"</i> <i>// [4] "IL"</i> String[] sa3 = subset(sa1, 2, 3); println(sa3); <i>// Sort l'actualisation du tableau dans la console</i> <i>// [0] "CA"</i> <i>// [1] "VA"</i> <i>// [2] "CO"</i>	NOTE SUR LES TABLEAUX <i>- N'importe quel objet peut être stocké dans un arrayList.</i> <i>- Il existe dans processing d'autres types de Tableau</i> <i>comme les StringList, IntList, StringDir, IntDir ou encore</i> <i>les Table.</i> <i>- Pour plus d'informations, consultez la référence.</i>
AUTRES TABLEAUX (ArrayList)	VARIABLES DE TEMPS
<i>/* Dans cette exemple on suppose une classe Ball contenant</i> <i>des méthodes .move(), .display(), .finished()</i> <i>*/</i> ArrayList balls; void setup() { size(200, 200); balls = new ArrayList(); <i>// Crée un arrayList vide</i> balls.add(new Ball(width/2, 0, 48)); <i>// Commencons à ajouter un éléments.</i> } void draw() { background(255); <i>// On fait une boucle pour parcourir tout notre tableau.</i> for (int i = 0 ; i < balls.size()-1; i++) { <i>// Un arrayList ne sait pas quel type d'objet il contient</i> <i>// Donc nous devons typé l'objet de sortie.</i> Ball ball = (Ball) balls.get(i); ball.move(); <i>// Ceci est une fonction de Ball</i> ball.display(); if (ball.finished()) { <i>// les objets peuvent être supprimée avec .remove();</i> balls.remove(i); } } } <i>// Quand on clic on ajoute un objet à notre tableau</i> void mousePressed() { <i>// Un nouvel objet est ajouté dans la list, par défaut, à</i> <i>// la dernière position</i> balls.add(new Ball(mouseX, mouseY, ballWidth)); } <i>/* ArrayList contient donc les méthodes</i> .add(); Ajoute un objet .get(); Prendre un objet .remove(); Supprimer un objet. <i>*/</i>	TEMPS PAR RAPPORT A L'ORDINATEUR
	DAY(); <i>//Retourne le jour en int</i> bob = day(); HOURL(); <i>//Retourne l'heure en int</i> bob = hour(); MINUTE(); <i>//Retourne la minute en int</i> bob = minute(); MONTH(); <i>//Retourne le mois en int</i> bob = month(); SECOND(); <i>//Retourne la seconde en int</i> bob = second(); YEAR(); <i>//Retourne l'année en int</i> bob = year(); EXEMPLE : <i>//Retourne la date du jour.</i> String today = "Le" + day() + "/" + month() + "/" + year() + " à " + hour() + ":" + minute() + ":" + second(); println(today); TEMPS PAR RAPPORT AU PROGRAMME MILLIS(); <i>//Retourne le nombre de millisecondes exécutées</i> <i>depuis le debut du programme.</i>
	13

LES TABLEAUX (OPÉRATIONS)	REVERSE()
APPEND() <i>//Ajoute un éléments à la fin d'un tableau</i> String[] pays1 = { "France", "Suisse", "Italie"}; String[] pays2 = append(pays1, "Allemagne"); println(pays2); <i>// Sort l'actualisation du tableau dans la console</i> <i>// [0] "France"</i> <i>// [1] "Suisse"</i> <i>// [2] "Italie"</i> <i>// [3] "Allemagne"</i> ARRAYCOPY() <i>//Copie l'intégralité d'un tableau dans un autre</i> String[] europe = { "France", "Suisse", "Italie"}; String[] asie = { "Chine", "Japon", "Laos"}; arrayCopy(europe, asie); println(asie); <i>//Contenu du tableau asie</i> <i>// [0] "France"</i> <i>// [1] "Suisse"</i> <i>// [2] "Italie"</i> CONCAT() <i>//Unifie deux tableaux du même type dans un troisième</i> String[] europe = { "France", "Suisse", "Italie"}; String[] asie = { "Chine", "Japon", "Laos"}; String[] monde = concat(europe, asie); println(monde); <i>//Contenu du tableau monde</i> <i>// [0] "France"</i> <i>// [1] "Suisse"</i> <i>// [2] "Italie"</i> <i>// [3] "Chine"</i> <i>// [4] "Japon"</i> <i>// [5] "Laos"</i> EXPAND() <i>//Par défaut double la longueur d'un tableau</i> int[] data = {0, 1, 3, 4}; println(data.length); <i>// Prints "4"</i> data = expand(data); println(data.length); <i>// Prints "8"</i> data = expand(data, 512); println(data.length); <i>// Prints "512"</i>	<i>//Inverse toutes les valeurs d'un tableau</i> String sa[] = { "OH", "NY", "MA", "CA"}; sa = reverse(sa); <i>// Sort l'actualisation du tableau dans la console</i> <i>// [0] "CA"</i> <i>// [1] "MA"</i> <i>// [2] "NY"</i> <i>// [3] "OH"</i> SHORTEN() <i>//Supprime le dernier élément d'un tableau</i> String[] sa1 = { "OH", "NY", "CA"}; String[] sa2 = shorten(sa1); println(sa1); <i>// sa1 contient toujours OH, NY, CA</i> println(sa2); <i>// sa2 maintenant contient OH, NY</i> SORT() <i>//Pour une String : Trie les éléments par ordre alphabétique.</i> <i>//Pour un Int, Float, Char, ... : Trie les éléments du plus grand au plus petit.</i> String[] monde = { "France", "Italie", "Allemagne", "Kenya", "Belgique"}; monde = sort(monde); println(monde); <i>//Contenu du tableau monde</i> <i>// [0] "Allemagne"</i> <i>// [1] "Belgique"</i> <i>// [2] "France"</i> <i>// [3] "Italie"</i> <i>// [4] "Kenya"</i> SPLICE() String[] a = { "OH", "NY", "CA"}; a = splice(a, "KY", 3); <i>// Ajoute une valeur dans un tableau à la position désirée.</i> <i>//Splice : tableau, élément, position dans le tableau);</i> println(a); <i>// Trace le nouveau tableau</i> <i>// [0] "OH"</i> <i>// [1] "KY"</i> <i>// [2] "NY"</i> <i>// [3] "CA"</i> SUBSET() String[] sa1 = { "OH", "NY", "CA", "VA", "CO", "IL"}; String[] sa2 = subset(sa1, 1); println(sa2); <i>// Sort l'actualisation du tableau dans la console</i>
	12

PVECTOR (LES VECTEURS)	
PVector v1, v2; void setup() { v1 = new PVector(40, 20); v2 = new PVector(25, 50); } void draw() { ellipse(v1.x, v1.y, 12, 12); ellipse(v2.x, v2.y, 12, 12); v2.add(v1); ellipse(v2.x, v2.y, 24, 24); } VARIABLES ACCESSIBLES x <i>//Variable x du vecteur</i> y <i>//Variable y du vecteur</i> mag <i>//Variable z du vecteur</i> EXEMPLE : rect(v1.x,v1.y,10,10); <i>//Affiche un carré en position x et y de mon vecteur.</i> METHODES ACCESSIBLES .set() <i>//Attribut les composants x,y ou x,y et z du Vecteur.</i> Exemple : v1.set(9,3); <i>//Attribut la valeur 9 en x et 3 en y</i> .get() <i>//Récupère un copy du Vecteur</i> Exemple : v2 = v1.get(); <i>//Le Vecteur v2 à maintenant toutes les valeurs de v1</i> <i>// 44.72136</i> .mag() <i>//Récupère la magnitude d'un vecteur (sa longueur)</i> float r = v1.mag(); println(r); <i>// r équivaut à</i> <i>// 44.72136</i> .angleBetween() <i>//Calcule et retourne un angle entre deux vecteurs.</i> .array() <i>//Retourne une représentation d'un vecteur par un tableau de float</i>	.add() <i>//Ajoute un vecteur avec un autre vecteur.</i> PVector v1, v2; void setup() { v1 = new PVector(40, 20); v2 = new PVector(25, 50); v2.add(v1); } QUELQUES AUTRES METHODES ACCESSIBLES... <i>// Pour plus d'informations, consultez la référence.</i> .random2d() <i>//Donne deux valeurs : x et y aléatoire au vecteur compris</i> <i>entre -1 et 1.</i> .random3d() <i>//Donne trois valeurs : x, y et z aléatoire au vecteur compris</i> <i>entre -1 et 1.</i> .sub() <i>//Soustrait les valeurs x, y et z du vecteur par les valeurs</i> <i>d'un autre, ou de deux vecteurs indépendants.</i> .mult() <i>//Multiplie un vecteur par un scalaire, ou par un autre vecteur.</i> .div() <i>//Divise un vecteur par un scalaire, ou par un autre vecteur.</i> .dist() <i>//Calcul la distance entre deux points.</i> .normalize() <i>//Normalise la longueur d'un vecteur à 1.</i> .limit() <i>//Limite la magnitude/longueur d'un vecteur</i> .setMag() <i>//Définit la magnitude/longueur d'un vecteur.</i> .angleBetween() <i>//Calcule et retourne un angle entre deux vecteurs.</i> .array() <i>//Retourne une représentation d'un vecteur par un tableau de float</i>
	14

CHARGEMENT DE DONNÉES CSV (SIMPLE)

Table mData;

```
void setup(){
  size(800,600);
  mData = loadTable("data.csv","header");
}

void draw(){
  background(200);
  for(TableRow row : mData.rows()){
    fill(row.getInt("r"),row.getInt("v"),row.getInt("b"));
    noStroke();

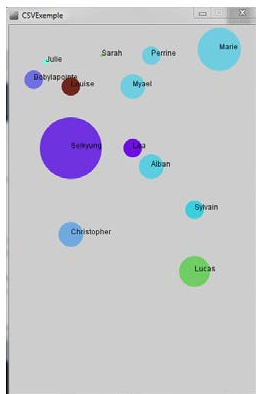
    ellipse(row.getFloat("x"),row.getFloat("y"),row.getFloat("diametre"),row.getFloat("diametre"));

    fill(o);
    text(row.getString("Nom"),row.getFloat("x"),row.getFloat("y"));
  }
}
```

DÉTAIL DU FICHIER CSV

La première ligne correspond à l'en-tête de colonne

```
Nom,x,y,diametre,r,v,b
Boby,100,100,40,89,30,100,100,220
Julie,60,60,10,100,255,220
Sarah,150,50,4,100,200,30
Marie,340,40,70,100,200,220
Louise,100,100,30,100,25,20
Myael,200,100,40,100,200,220
Seikyung,100,200,100,100,40,220
Perrine,230,50,30,100,200,220
Lucas,300,400,50,100,200,90
Christopher,100,340,40,100,160,220
Alban,230,230,40,80,200,220
Sylvain,300,300,30,50,200,220
Lea,200,200,30,100,10,220
```



15

CONVERSIONS (TRANSTYPAGE)

BINARY();

//Retourne une représentation binaire d'un char, byte, ou d'un int sous forme de String
color c = color(255, 204, 0);
println(c); // Prints ~13312~
println(binary(c));
 // Prints ~111111111111111111001100000000~
println(binary(c, 16)); // Prints ~1100110000000000~

BOOLEAN();

//Converti un int, String ou un array en boolean, pour un int 0 revient à false et 1 à true.
String s = "true";
println(boolean(s));
 //Prints true

BYTE();

//Converti un char en byte, de -128 à 127
char c = 'E';
byte b = byte(c);
println(c + " : " + b); // Prints ~E : 69~

int i = 130;
byte b = byte(i);
println(i + " : " + b); // Prints ~130 : -126~

CHAR();

int i = 65;
char c = char(i);
println(i + " : " + c); // Prints ~65 : A~

byte b = 65;

c = char(b);
println(b + " : " + c); // Prints ~65 : A~

FLOAT();

// Converti un int, string, ou un tableau en float.
String bob = "65.9";
float pat = float(bob);
println(pat); // Trace 65.9 dans la console

HEX();

//Converti un byte, char, int, ou un objet couleur dans la représentation hexadecimal String (donc la chaîne de caractère correspondante)
color c = color(255, 204, 0);
println(c); // Prints ~13312~
println(hex(c)); // Prints ~FFFFCC00~

INT();

//Converti un datatype (float, boolean, char, byte, color), string ou tableau dans sa représentation entière.
float f = 65.0;
int i = int(f);
println(f + " : " + i); // Prints ~65.0 : 65~

char c = 'E';

int i = int(c);
println(c + " : " + i); // Prints ~E : 69~

STR(); EN STRING

//Retourne la représentation d'un primitif en chaîne de caractères "

boolean b = false;

byte y = -28;

char c = 'R';

float f = -32.6;

int i = 1024;

String sb = str(b);

String sy = str(y);

String sc = str(c);

String sf = str(f);

String si = str(i);

sb = sb + sy + sc + sf + si;

println(sb); // Prints ~false-28R-32.61024~

UNBINARY();

// Converti une représentation binaire String en un binaire entier.

String s1 = "00010000";

String s2 = "00001000";

String s3 = "00000100";

println(unbinary(s1)); // Prints ~16~

println(unbinary(s2)); // Prints ~8~

println(unbinary(s3)); // Prints ~4~

UNHEX();

// Converti une chaîne de caractère en objet hexadécimal.

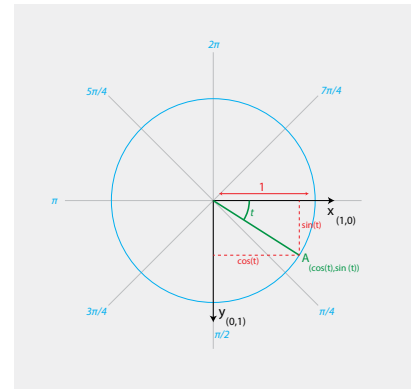
String hs = "FF006699";

int hi = unhex(hs);

fill(hi);

rect(30, 20, 55, 55);

CERCLE TRIGONOMETRIQUE (ORIENTATION X,Y DE PROCESSING)



PI
 //Renvoie la valeur de PI
 //C'est à dire :
 //3.1415927

radians();

//Transforme des degrés en radians
float r = PI/2;
println(degrees(r));
 //renvoie 90
 //donc 90 degrés.

degrees();

//Transforme des radians en degrés
float r = 45;
println(radians(r));
 //renvoie 0.7853982
 //rét qui équivaut à PI / 4;
 //donc 45 degrés.

dis();

//Calcul de la distance entre 2 points
 //où x1 est un **float**
 //où x2 est un **float**
 //où y1 est un **float**
 //où y2 est un **float**
float r=dist(x1,y1,x2,y2);
println(r);

FONCTIONS SINUS ET COSINUS

float bob = cos(t);
 //la valeur périodique cosinus d'un angle ou d'une valeur
float bob = sin(t);
 //la valeur périodique sinus d'un angle ou d'une valeur

Astuce :

//Utilisation régulière de fonction sinus et cosinus
 // par exemple pour tracer un cercle de centre px,py et de rayon r ;

float t=0;
float px,py=400;
float r = 100;

```
void setup(){
  size(800,800);
}
void draw(){
  point(px+cos(t)*r , py+sin(t)*r );
  t+=0.01;
}
```

18

VIDEO / WEBCAM : INITIALISATION

// Importation de la librairie video

import processing.video.*;

// Initialisation de l'objet vidéo

Capture cam;

```
void setup() {
  size(200, 200);
  // Construction de l'objet vidéo
  cam = new Capture(this);
  // Lancement de l'objet vidéo
  cam.start();
}
```

```
void draw() {
  // Affichage de l'objet vidéo
  image(cam, 0, 0);
}
```

// Événement de réception d'image de l'objet vidéo

```
void captureEvent(Capture c) {
  c.read();
}
```

17

LES STRINGS OU "CHAÎNES DE CARACTÈRE"

```
String bob = "Super bob, super, bob !";
```

EQUALS();

```
//Comparaison de deux chaînes de caractères.  
//Ne pas utiliser == pour comparer deux chaînes.  
//Utiliser .equals()  
if(bob.equals("bob")){  
    //si la variable bob est égale à bob.  
}
```

CHARAT();

```
//Récupération d'un caractère à un index donné.  
char r = bob.charAt(3);  
println(r);  
// r va retourner "e", car dans  
// "Super bob,superbob !" le 3ème caractère  
// est le "e".
```

INDEXOF();

```
//Recherche d'un caractère ou d'une chaîne donnée et  
retour de son index  
int r = bob.indexOf("bob");  
println(r);  
// r retourne 6  
//en effet "bob" se trouve au niveau du 6ème caractère
```

LENGTH();

```
// Récupération du nombre de caractères dans la chaîne  
int r = bob.length();  
println(r);  
// r retourne 23
```

SUBSTRING();

```
// Avec 1 paramètre ou 11 est un int  
// récupère depuis l'index donné le reste  
// de la chaîne de caractères.  
String r = bob.substring(11);  
println(r);  
// Affiche dans la console :  
// super, bob !  
  
// Avec 2 paramètres où 6 et 3 sont des int  
// récupère depuis l'index et le nombre de caractères  
// à récupérer  
String r = bob.substring(6,3);  
println(r);  
// Affiche dans la console :  
// bob
```

TOLOWERCASE();

```
//transforme tout les caractères en caractères bas de  
casse  
String r = bob.toLowerCase();  
println(r);  
// affiche dans la console :  
// super bob, super, bob !
```

TOUPPERCASE();

```
//transforme tous les caractères en caractères hauts de  
casse  
String r = bob.toUpperCase();  
println(r);  
// affiche dans la console :  
// SUPER BOB, SUPER, BOB !
```

QUELQUES AUTRES FONCTIONS POUR "STRING"

JOIN();

```
//Permet de grouper un tableau de String  
//1er paramètre : tableau à joindre  
//2ème paramètre : Chaîne de jointure  
String[] animaux = new String[3];  
animaux[0] = "crabe";  
animaux[1] = "chat";  
animaux[2] = "singe";  
String groupeAnimaux = join(animaux, " : ");  
println(groupeAnimaux);  
// Ecrit dans la console "crabe : chat : singe"
```

SPLIT();

```
/* Permet de découper une chaîne en "tableau de chaîne"  
1er paramètre : chaîne à découper  
2ème paramètre : chaîne de découpe */  
String animaux = "Chat,Chien,Singe";  
String[] tabAnimaux = split(animaux, ",");  
/* tabAnimaux[0] contient "Chat"  
tabAnimaux[1] contient "Chien"  
tabAnimaux[2] contient "Singe"  
*/
```

SPLITTOKEN();

```
/* Permet de nettoyer des données mal formatées  
Le " ", " " est le délimiteur qui permet d'enlever les  
caractères superflus. */  
String bob = "a, b c ,d ";  
String[] tab = splitTokens(bob, " , ");  
println(tab[0]); // retourne "a"  
println(tab[1]); // retourne "b"  
println(tab[2]); // retourne "c"  
println(tab[3]); // retourne "d"
```

LES CLASSES

```
//-----  
//SCENE PRINCIPALE  
//-----  
  
//Initialisation de 2 objets de type Monstre  
Monstre bob;  
Monstre pat;  
  
void setup(){  
    size(500,500);  
  
    //Création des objets Monstres  
    bob = new Monstre();  
    pat = new Monstre(100,20);  
}  
  
void draw(){  
    //Utilisation des méthodes internes à Monstre  
    //Ici void affich(); et run();  
    bob.run();  
    bob.affich();  
  
    //Utilisation des méthodes internes à Monstre  
    //Ici void affich();  
    pat.affich();  
}
```

MA CLASS MONSTRE

```
//-----  
//Onglet pour une classe Monstre (de préférence).  
//-----  
  
//Déclaration de ma classe Monstre  
class Monstre{//Début de classe  
  
    //Variable interne à une classe  
    int px=0;  
    int py=0;  
  
    //Constructeur sans paramètres  
    Monstre(){  
    }  
  
    //Constructeur avec 2 paramètres  
    Monstre(int px,int py){  
        //Récupération des paramètres  
        this.px=px;  
        this.py=py;  
    }  
  
    //Fonction personnelle  
    void affich(){  
        //Affichage par exemple  
        rect(px,py,10,10);  
    }  
  
    //Autre fonction personnelle sans paramètre  
    void run(){  
        px++;  
    }  
}  
//Fin de classe
```

Mémo informatique

[Terminal Cmd + App Linux]

Pratiques numériques / Cultures électroniques

Ce mémo est dédié aux artistes souhaitant travailler sous linux. Il présente à la fois les fonctions du terminal de commande mais aussi contient une liste des principaux logiciels libres et d'outils exploitables et représentant un intérêt pour les pratiques artistiques. Vous pouvez télécharger la version imposé ou non-imposé. Il est hébergé sur github, libre de droit, téléchargeable et imprimable gratuitement.

https://github.com/randomDam/memo_cmd

MÉMO

TERMINAL CMD + APP GNU/LINUX

RANDOM(lab) | ESADSE | version 1.0 | Janvier 2020 | <http://randomlab.io>

Mémo (guide de résistance) des usages du terminal à l'usage des Graphistes/Artistes/Designers/Ingénieurs

BASES DE COMPREHENSION DU MÉMO

rep, source et cible (en vert) designent un répertoire/dossier du type :

- Bureau/
- Bureau/monDossier
- ou
- /home/bob/Bureau/monDossier

file, \$source et \$cible (en violet) designent un fichier du type :

- MonTexte.txt
- ou
- Bureau/monImage.png
- ou
- /home/bob/Bureau/maVideo.mp4

Pour toutes les commandes, on peut afficher l'aide avec :

- NonDeMaCommande -h ou
- NonDeMaCommande --help ou
- man NonDeMaCommande (touche Q pour quitter man)

Cli : Command Line Interface
Interface en ligne de commande

Tui : Text User Interface
Interface en mode text dans une ligne de commande

Gui : Graphic User Interface
Interface graphique utilisateur

NAVIGUER DANS LES RÉPERTOIRES

Afficher le répertoire courant

```
pwd
```

Se placer dans le répertoire rep

```
cd rep
```

Se placer dans le répertoire de l'utilisateur ~/

```
cd ~
```

Se placer dans le répertoire parent

```
cd ..
```

Lister les fichiers du répertoire rep

```
ls rep
```

ls avec les fichiers cachés

```
ls -a
```

ls avec les droits d'accès et la taille

```
ls -l
```

ACTIONS SUR LES FICHIERS/DOSSIERS

Déplacer le fichier source vers cible

```
mv source cible
```

Copier le fichier source vers cible

```
cp source cible
```

Copier le répertoire source vers cible

```
cp -R source cible
```

Créer un lien fort de source vers lien

```
ln source lien
```

Créer un lien symbolique de source vers lien

```
ln -s source lien
```

Créer le fichier file ou met à jour sa date de modification

```
touch file
```

Créer un répertoire rep

```
mkdir rep
```

mkdir avec création du rep parent si nécessaire

```
mkdir -p rep/rep2
```

Supprimer le fichier file

```
rm file
```

Supprimer le fichier file protégé en écriture

```
rm -f file
```

Supprimer un répertoire vide

```
rmdir rep
```

Supprimer un répertoire (vide ou plein)

```
rm -R rep
```

Afficher la taille de file ou du répertoire rep

```
du -h file ou rep
```

Afficher la taille d'un répertoire en se limitant aux répertoires et trier la sortie par taille décroissante

```
du -hd 1 rep | sort -hr
```

Afficher le contenu de l'arborescence des fichiers dans un répertoire

```
tree rep
```

(si il n'est pas encore installé :)

```
sudo apt install tree
```

Afficher le type d'un fichier

```
file fichier
```

Obtenir les informations d'un fichier

```
stat fichier
```

Créer un lien symbolique

```
ln -s target_file linked_file
```

Supprimer un lien symbolique

```
unlink linked_file
```

Voir le contenu (la cible) d'un lien symbolique

```
readlink linked_file
```

MC (Tui)

GNU Midnight Commander (mc) est un gestionnaire de fichiers multiplate-forme en mode texte. L'interface principale se compose de deux « panneaux » qui affichent les fichiers présents par rapport à leur emplacement sur le disque.

mc

si il n'est pas encore installé :

```
sudo apt install mc
```

LFM(Tui)

Last File Manager est un puissant gestionnaire de fichiers pour la console. Il possède une interface Cli.

lfm

si il n'est pas encore installé :

```
sudo apt install lfm
```

AFFICHER/COMPARER LES FICHIERS

Compte le nombre de lignes, de mots, d'octets de fichier

```
wc fichier
```

Concatène/affiche les fichiers

```
cat fichiers
```

Affiche fichier page après page 'Espace'=page suivante, 'Entrée'=ligne suivante, 'u'=remonter

```
more fichier
```

Affiche fichier avec une navigation au clavier

```
less fichier
```

Affiche les x premières lignes de fichier

```
head -n x fichier
```

Affiche les x dernières lignes de fichier

```
tail -n x fichier
```

Affiche la dernière ligne de fichier en temps réel

```
tail -f fichier
```

Affiche les différences entre deux fichiers texte

```
diff fichier1 fichier2
```

Affiche les différences au format patch

```
diff -u fichier1 fichier2
```

Compare deux fichiers binaires

```
cmp fichier1 fichier2
```

Compare deux fichiers, file1 à partir du nième octet, et file2 à partir du nième

```
cmp fichier1 fichier2 nN
```

UTILISATEURS

Affiche le login de l'utilisateur

```
whoami
```

Affiche les utilisateurs connectés

```
who
```

Afficher les uid, gid et groupes de l'utilisateur

```
id
```

Afficher les uid, gid et groupes de user (root only)

```
id $user
```

Affiche les informations de user

```
finger $user
```

ex: finger bob

si il n'est pas encore installé :

```
sudo apt install finger
```

Afficher un message sur le terminal de user

```
write $user
```

Afficher le nom de son terminal

```
tty
```

Passer en mode administrateur, super-utilisateur

```
su (ou) sudo
```

Changer le mot de passe de l'utilisateur courant

```
passwd
```

Ajouter un utilisateur

```
adduser
```

Supprimer un utilisateur

```
deluser
```

Ajouter un groupe

```
addgroup
```

Supprimer un groupe

```
delgroup
```

PROCESSUS

Afficher les processus de l'utilisateur

```
ps
```

Afficher tous les processus

```
ps ax
```

Afficher les processus et leur utilisateur

```
ps aux
```

Afficher les processus dans une arborescence

```
pstree
```

Affiche un tableau des processus gourmands

```
top
```

Tuer un processus en utilisant son pid

```
kill signal pid
```

Tuer un processus en utilisant le nom du programme

```
pkill signal nom
```

SIGNAUX UTILISÉS POUR KILL/TKILL

- 1 (HUP) recharger le fichier de configuration du processus
- 2 (INT) interrompre le processus
- 3 (QUIT) quitter le processus
- 9 (KILL) tuer le processus (à éviter, tenter -15 avant)
- 15 (TERM) terminer le processus proprement
- 18 (STOP) geler le processus
- 20 (CONT) reprendre l'exécution d'un processus gelé

Trouver la liste des processus qui utilise un répertoire (en root)

```
lssof -x +0 /repertoire
```

Trouver la liste des fichiers ouverts par un processus (en root) lsuf -p pid	NMCLI (CUI) SE CONNECTER EN MODE CUI Affiche le status des connections nmcli devices status
HTOP (TUI) VISUALISER ET GERER L'ENSEMBLE DES PROCESSUS htop si il n'est pas encore installé : sudo apt install htop	Affiche le réseau wifi dans le terminal nmcli device wifi list Se connecte à un réseau wifi nmcli device wifi connect \$ssid password \$pass
MATERIEL Liste les périphériques de type USB connectés lsusb Liste les périphériques de type PCI connectés lspci Affiche les informations processeur cat /proc/cpuinfo Affiche les partitions montées cat /proc/partitions Afficher le modèle de sa carte graphique : lspci egrep "3D Display VGA" Afficher le modèle de sa carte Wi-Fi : lspci grep -i "net" cut -d: -f3 Afficher le modèle de sa carte son : lspci grep -i audio cut -d: -f3 Affiche les message du noyau linux. Très pratique pour voir les périphériques connecté ou déconnecté. dmesg dmesg avec option de suivie continue dmesg -w	NMTUI (TUI) SE CONNECTER EN MODE TUI Gestion du réseau en mode TUI (Text User Interface) nmui WAVEMON (TUI) VISUALISER L'ENSEMBLE DES RESEAUX WIFI Analyse du réseau wifi sudo wavemon si il n'est pas encore installé : sudo apt install wavemon
	TELNET(CUI) Telnet est un protocole utilisé sur tout réseau TCP/IP, permettant de communiquer avec un serveur distant en échangeant des lignes de texte et en recevant des réponses également sous forme de texte. [1] Il n'est pas sécurisé... (voir ssh) sudo apt install telnet
	SOCAT(CUI) Socat est un utilitaire en ligne de commande qui établit deux flux d'octets bidirectionnels et transfère des données entre eux. Par exemple : rediriger un fichier texte vers un port distant. sudo apt install socat
	NETCAT(CUI) En informatique, netcat, également abrégé nc, est un utilitaire permettant d'ouvrir des connexions réseau, que ce soit UDP ou TCP. Il est conçu pour être incorporé aisément dans un large éventail d'applications. sudo apt install netcat
	NMAP(CUI) Pour scanner les ports d'un ordinateur distant. Nmap utilise diverses techniques d'analyse qui s'appuient sur des protocoles tels que TCP, IP, UDP ou ICMP. De même, il se fonde sur les réponses qu'il obtient à des requêtes particulières pour obtenir une empreinte de la pile IP, souvent propre au système qui l'utilise. sudo apt install nmap
	ZENMAP Version GUI de nMap sudo apt install zenmap
	WIRESHARK(GUI) OUTIL DE DIAGNOSTIQUE RESEAU Version de la distribution sudo apt install wireshark Dernière version disponible : sudo add-apt-repository ppa:wreshark-dev/stable sudo apt-get update
	3

CALCULATRICE lancer bc en ligne de commande (calcule entier simple) bc lancer bc en ligne de commande (avec le module scientifique) bc -l s(x) sinus c(x) cosinus a(x) arctangente l(x) logarithme naturel e(x) exponentielle sqrt(x) racine carrée x^y puissance x%y reste de la division euclidienne Calculer une opération avec bc bc -l <<< "i+sqrt(5)" Calculer le nombre Pi à 10 décimal pi=\$(bc -l <<< "scale=10; 4*a(1)") && echo "Pi est égal à \$pi"	ARCHIVES .tar.bz2, .tbz2 Compresser : tar -cvzf archive.tar.bz2 repertoire Décompresser : tar xvzf .tar.gz, .tgz Compresser : tar -cvzf archive.tar.gz repertoire Décompresser : tar xvzf .bz2 Compresser : bzip2 fichiers Décompresser : bunzip2 .RAR Compresser : rar a archive.rar repertoire Décompresser : unrar x .GZ Compresser : gzip fichiers Décompresser : gunzip .JAR Compresser : tar -cvf archive.tar fichiers Décompresser : tar xvf .ZIP Compresser : zip -r archive.zip fichiers Décompresser : unzip .Z Compresser : compress fichiers Décompresser : uncompress .Tz Compresser : Tz a fichiers Décompresser : Tz x .XZ Compresser : xz -z repertoire Décompresser : unxz
MONITORING Afficher la liste des processus en arrière plan jobs Place au premier plan le processus pid fg pid Place en arrière plan le processus pid bg pid Redéfinir le niveau de priorité du processus pid renice +1pid Tuer le job [n] kill %n	RECHERCHE Recherche sur un nom non correspond au motif locate motif Mettre à jour la base de données de locate updatedb Recherche les fichiers dans chemin avec option find chemin options Recherche sur le nom du fichier find -name motif Recherche par type où f=fichier,d=répertoire,l=lien find -type [fdl] Exécute la commande cmd à tous les fichiers trouvés find -exec cmd Rechercher un fichier d'après son nom dans le répertoire dir find dir -type f -name filename EXEMPLE : Trouver toutes les images avec l'extension png dans le dossier 'Images' de l'utilisateur et les copier dans le dossier tmp ('/') find \$HOME/Images -name "*.png" -exec cp {} \$HOME/tmp/ \;
4	

KERNEL Version du noyau Linux utilisé, son nom, la version du compilateur utilisé : cat /proc/version Version du kernel : uname -r Liste les noyaux installés sur votre machine dpkg -l egrep "linux-(header image)"	Composer un PDF à partir de plusieurs pdftk A=fil1.pdf B=fil2.pdf cat A1-3 B2 A4 output outfile.pdf Supprimer des pages d'un PDF (ici 7,9,14) pdftk infile.pdf cat <'7-9-14' output outfile.pdf Inverser les pages d'un PDF pdftk infile.pdf cat end-1 output outfile.pdf Composer un PDF livret en 2 pages par feuilles pdftbook --short-edge --paper a4paper --outfile outfile.pdf infile.pdf Composer un PDF avec 4 pages par feuilles pdfnup --nup 2x2 --paper a4paper --landscape --outfile outfile.pdf infile.pdf CRON Crontab est un outil qui permet de lancer des applications de façon régulière, pratique pour un serveur pour y lancer des scripts de sauvegardes, etc. Pour afficher le contenu du fichier crontab : crontab -l Pour supprimer toutes les actions du fichier crontab : crontab -r Pour éditer les actions du fichier crontab : crontab -e
SON, MUSIQUE ET ENREGISTREMENT Donne des informations sur les cartes sons amixer Permet de gérer lesniveaux des différentes sorties de la carte son alsamixer Permet de gérer/selectionner/dispatcher les niveaux et sorties des périphériques/son/sorties audios (Tui) pulseixer (apt install pulseixer, s'il n'est pas présent) Permet de gérer lesniveaux des différentes sorties audios (Gui) pavucontrol (apt install pavucontrol, s'il n'est pas présent) Lire un fichier audio simple (wav,...) aplay file Lire un fichier audio file.mp3 mpg321 file.mp3 (apt install mpg321, s'il n'est pas présent) Lire un fichier audio file.mp3 avec plus d'information mpg321 -v file.mp3	DANS LA COMMANDE crontab -e, FORME CANONIQUE : @minute heure jourDansLeMois Mois JourDansLaSemaine user Command Exécution tous les jours à 22h00 d'une commande action.sh : 00 22 * * * /home/bob/bin/action.sh Exécution d'une commande action.sh toutes les 6 heures : 00 */6 * * * /home/bob/bin/action.sh Exécution d'une commande toutes les heures : 00 */1 * * * /home/bob/bin/action.sh Exécution d'une commande toutes les minutes que les lundis : * * * * 1 /home/bob/bin/action.sh Exécute chaque jour, de chaque mois à 2:15 : 15 02 * * * /home/bob/bin/action.sh Exécution d'une commande toutes les minutes : * * * * * /home/bob/bin/action.sh Exécution d'une commande au démarrage : @reboot /home/bob/bin/action.sh
MOC (TUI) PUISSANT LECTEUR MUSICAL DE DOSSIER/FICHER Lecteur et explorer de bibliothèque musical mocp (apt install moc, s'il n'est pas présent)	ÉTAT DE CRON Voir l'état de cron : sudo service cron status Démarrer le service cron : sudo service cron start Arrêter le service cron : sudo service cron stop
CORRECTIONS ORTHOGRAPHIQUE ET GRAMMATICALE Vérifier l'orthographe d'un fichier écrit en français aspell check --lang=fr infile	
EXPORTER LE CONTENU, CHANGER DE FORMAT Exporter un fichier texte en PDF (HTML, DOCX,...) pandoc infile -o outfile.pdf Voir un fichier Markdown en rich text dans le terminal pandoc file.md w3m -T text/html	
MANIPULATION DE PDF Afficher les informations fichier d'un PDF pdinfo file.pdf Convertir un PDF en fichier texte pdftotext infile.pdf outfile.txt Renommer des fichiers (Debian) rename 's/exp1/exp2/' xxxexp1xx Renommer en fichier en *.old mv filename{,.old}	5

HISTORIQUE DES COMMANDES Afficher l'historique des commandes history Exécuter la dernière commande !! Exécuter la commande numérotée n !n	Donner une chaîne de caractères en argument d'une commande command <<< string ou (équivalent mais moins bien) echo string command Écrire dans un fichier en sortie standard stoud, par exemple pour continuer un long pipe command tee outfile ... Éditer le résultat d'une commande avec Vim command vim -
EXÉCUTIONS SÉQUENTIELLES OU CONDITIONNELLES Exécuter de façon séquentielle deux commandes command1 ; command2 Exécuter de façon parallèle deux commandes command1 && command2 Exécuter la command2 que si la command1 a réussie command1 && command2 Exécuter la command2 que si la command1 a échoué command1 command2	MANIPULATION DE FICHER RECHERCHER ET REMPLACER Trouver une expression dans un fichier grep 'expression' infile Trouver une expression dans un fichier avec les numéros de lignes (-n) grep -n 'expression' infile DECOUPER, ASSEMBLER Découper un fichier texte par blocs de 100 lignes split -l 100 -d infile Découper un fichier texte sur un motif split infile '/[section]/' {f} -f 'prefix_' Assembler une collection de fichiers texte de noms x00, x01 cat x* > outfile Concaténer des fichiers (verticalement) cat infile1 infile2 ... Concaténer les colonnes de fichiers (horizontalement) paste infile1 infile2
REDIRECTION DE RESULTAT Rediriger la sortie standard de command vers le fichier outfile command > outfile Rediriger la sortie standard de la command vers le fichier outfile en fin de fichier command >> outfile Rediriger l'entrée standard de la command vers le fichier infile command < infile Rediriger l'entrée standard de command2 vers la sortie standard de la command1 command2 < (command1) Rediriger la sortie d'erreur standard de command vers le fichier outfile command 2> outfile Rediriger l'erreur standard de command au même endroit que la sortie standard command 2&1 Rediriger la sortie d'erreur standard de command au même endroit que la l'erreur standard command &1&2 TUBES (PIPELINE) Exécuter la command2 avec le résultat de la command1 command1 command2 Scinder les éléments provenant du tube en différentes lignes ... xargs -n 1 echo Concaténer les lignes provenant du tube en une ligne ... xarg echo xtraire du tube les lignes contenant expression ... grep expression xtraire du tube les lignes ne contenant pas l'expression ... grep -v expression Lancer une commande de manière répétitive avec chaque élément provenant du pipe comme paramètre ... xargs -n 1 command	REMETTRE EN FORME Limiter la longueur des lignes à 72 caractères fold -sw 72 infile > outfile Remettre en forme un fichier texte en joignant les lignes successives trif infile > outfile FAIR DES LIGNES: sort sort -n Tri au format numérique sort -h Tri au format "human readable" sort -M Tri par mois sort -k2 Tri effectué sur le second champ de la ligne Exemple: sort monFichier.txt
6	

RECHERCHER DES DOUBLONS : <code>uniq</code> <code>uniq</code> s'opère que sur les lignes qui se suivent. Compare la ligne suivante à la ligne précédente. Supprimer les lignes en doublons de <code>fic.txt</code> <code>uniq fic.txt</code> <code>idem</code> <code>uniq -u fic.txt</code> N'afficher que les lignes en doublons <code>uniq -d fic.txt</code> Indiquer le nombre d'occurrences de chaque ligne <code>uniq -c fic.txt</code> N ne pas comparer les <code>N</code> premiers caractères <code>uniq -s N fic.txt</code> Ne pas comparer les <code>N</code> premiers champs <code>uniq -f N fic.txt</code> MODIFICATION DE CARACTÈRES : <code>tr</code> Remplacer les caractères <code>abc</code> par <code>xyz</code> <code>tr abc xyz</code> Supprimer les doublons de caractère <code>tr -s [caractère]</code> Supprimer les espaces doublons <code>tr -s " "</code> Supprimer 123 <code>tr -d 123</code> DÉCOUPEUR DE FICHIER PAR COLONNES : <code>cut</code> Découper de la première colonnes du fichier <code>txt</code> <code>cut -f fic.txt</code> Choix d'un champ à afficher <code>cut -f n fic.txt</code> Choix d'une plage à afficher <code>cut -f n,x,y fic.txt</code> Choix des champs à afficher <code>cut -d "char" fic.txt</code> Affichage des champs <code>n</code> puis de tous les champs <code>cut -f n- fic.txt</code> Choix du caractère délimiteur de champs <code>cut -d "char" fic.txt</code> Choix de plage de caractères au lieu de champs <code>cut -c n-m fic.txt</code> Afficher la 3ème colonne d'un <code>fic.txt</code> , celles-ci étant délimitées par ; <code>cut -d ; -f3 fic.txt</code> Exemple : Affichage des logins, des commentaires et des informations qui suivent depuis le fichier <code>/etc/passwd</code> <code>cut -d : -f 3,5 /etc/passwd</code> Affichages des points de montage actifs <code>cut -d " " -f 2 /etc/mtab</code> Affichage de syslog à partir du caractère 27 <code>cut -c 27- /var/log/syslog</code>	MODIFICATION DE TEXTE LIGNE PAR LIGNE : <code>sed</code> Modifier la sortie standard le contenu de <code>fic.txt</code> selon l'action indiquée. On peut invoquer plusieurs fois l'option <code>-e</code> et passer plusieurs actions à la commande <code>sed</code> . <code>sed -e 'action' fic.txt</code> Écrit directement les modifications dans <code>fic.txt</code> <code>sed -i 'action' fic.txt</code> Récupère les actions dans le fichier <code>act.sed</code> <code>sed -f act.sed fic.txt</code> Utilisation de <code>sed</code> en mode silencieux <code>sed -n 'action' fic.txt</code> Exemple de syntaxe d'action : (pour <code>'action'</code> précédemment utilisé) : Substitution de <code>rech</code> par <code>modif</code> (première occurrence) <code>s/rech/modif</code> Substitution sur toutes les occurrences <code>s/rech/modif/g</code> Suppression des lignes contenant le motif <code>recherche</code> <code>/recherche/d</code> Suppression des lignes 3 à 7 <code>3,7d</code> Ajout de nouveau avant la ligne contenant <code>rech</code> <code>/rech/i "nouveau"</code> Ajout de nouveau après la ligne contenant <code>rech</code> <code>/rech/a "nouveau"</code> Remplace la ligne contenant <code>rech</code> par nouveau <code>/rech/c "nouveau"</code> Exemples complets : Remplacer une expression par une autre dans un fichier <code>sed -i 's/expression1/expression2/g' infile</code> Supprimer les lignes vides <code>sed -i '/^\$/d' infile</code> Modifier le nom de <code>srvtmp</code> par <code>srv001</code> dans le fichier <code>/etc/hostname</code> <code>sed -i 's/srvtmp/srv001' /etc/hostname</code> Supprime les lignes vides des fichiers <code>fic.txt</code> <code>sed -i '/^\$/d' fic.txt</code> Affiche le contenu de <code>/etc/passwd</code> en utilisant un espace comme séparateur des divers champs <code>sed -e 's/;/\t/g' -e '/\t/_/g' -e 's/^\t/_/g' /etc/passwd</code> 7
---	---

TELECHARGEMENT Télécharger un fichier <code>wget http://path.to.the/file</code> ou <code>curl http://path.to.the/file</code> Télécharger un fichier et le renommer <code>wget http://path.to.the/file -O nouveau</code> Reprendre un téléchargement avorté <code>wget -c http://path.to.the/file</code> Télécharger un fichier après authentification HTTP <code>curl -u username:password URL</code> Télécharger tous les PDF d'un site <code>wget -e robots=off -r -l1 --no-parent -A.pdf URL</code> Lister les médias disponibles sur un site de streaming <code>youtube-dl -f URL</code> Télécharger une vidéo sur un site de streaming en limitant la bande passante à 150 ko <code>youtube-dl -r 150k -f <format code> URL</code> PARTAGE DE FICHIERS (LOCAL ET TEMPORAIRE) <code>python3 -m http.server 8080</code> Un serveur web qui sert le dossier courant sur le port 8000 FTP Lister le contenu d'un répertoire FTP <code>curl ftp://username:password@example.com</code> Envoyer un fichier par FTP <code>lftp -u \$FTPuser,\$FTPwd ftp://\$FTPserver -e "put outfile; quit"</code> Créer un dossier mineur par FTP <code>lftp -u \$FTPuser,\$FTPwd ftp://\$FTPserver -e "mkdir -e -R localdir FTPdir; quit"</code> NAVIGATEURS Afficher la météo à Paris dans son terminal <code>curl wttr.in/paris,france</code> ou à Lyon en français <code>curl fr.wttr.in/Lyon,france</code> W3M (Cr.) Faire une recherche sur le web en tapant directement dans le terminal Recherche sur duckduckgo : "google" <code>w3m https://www.google.com/search?q="§"</code> Recherche sur duckduckgo : "ma recherche" <code>w3m https://duckduckgo.com/?q="§"</code> IMPRESSION Lister les imprimantes <code>lpstat -p</code> ou en se limitant aux noms <code>lpstat -a cut -f1 -d ' '</code>	Soumettre un travail d'impression à l'imprimante printer <code>lpr -P printer filename</code> Limiter l'impression à certaines pages <code>lpr -o page-ranges=1-4,7,9-12 filename</code> Imprimer un document en format paysage ajusté à un A4 <code>lpr -o landscape -o fit-to-page -o media=A4 filename</code> Imprimer 4 pages par feuille (2 pages/page et recto-verso) <code>lpr -o number-up=2 -o sides=two-sided-long-edge filename</code> Préciser le nombre de copies <code>n</code> à imprimer <code>lpr -#n filename</code> Voir la pile d'impression de l'imprimante printer <code>lpq -P printer</code> Supprimer le travail d'impression job-id <code>lprm job-id</code> Changer l'imprimante destinataire d'un travail job-id <code>/usr/sbin/lpmove job-id new-printer</code> ADMINISTRATION SYSTÈME Lancer une commande en administrateur <code>sudo command</code> ou <code>su -c command</code> Autoriser un utilisateur à utiliser le système de fenêtres X <code>sudo xhost local:user</code> Arrêter le système (en root) <code>shutdown -h now</code> ou <code>su -c poweroff</code> Afficher depuis combien de temps vous êtes connecté et la charge de la machine <code>uptime</code> Afficher l'historique des redémarrages du système <code>last reboot</code> ÉTAT DU SYSTÈME Nom de machine <code>hostname</code> Distribution <code>lsb_release -ldrc</code> Afficher les informations de la distribution <code>lsb_release -a</code> Informations du système <code>uname -a</code> Noyau <code>uname -r</code>
---	--

MANIPULATION D'IMAGES La manipulation d'image pass en général par imageMagick, en voici quelques fonctions : Pour l'installer : <code>sudo apt install imagemagick</code> animate Animer une séquence d'images sur n'importe quel serveur X. compare Annoter mathématiquement et visuellement la différence entre une image et sa reconstruction. composite Superposer une image sur une autre. conjure Interpréter et exécuter des scripts écrits dans le Magic Scripting Language (MSL). convert Convertir entre les formats d'image ainsi que redimensionner une image, flouter, recadrer, découper, tergiverser, dessiner, retourner, joindre, ré-échantillonner, et bien plus encore. display Afficher une image ou une séquence d'images sur n'importe quel serveur X. identify Décrire le format et les caractéristiques d'un ou plusieurs fichiers d'images. import Enregistrer toute fenêtre visible sur un serveur X et la sortir sous forme de fichier image. Vous pouvez capturer une seule fenêtre, l'écran entier ou toute partie rectangulaire de l'écran. mogrify Redimensionner une image, flouter, recadrer, découper, tergiverser, dessiner, retourner, joindre, ré-échantillonner, et bien plus encore. Mogrify efface le fichier image original, alors que, convert écrit dans un fichier image différent. montage Créer une image composite en combinant plusieurs images séparées. Les images sont superposées sur l'image composite, éventuellement agrémentées d'une bordure, d'un cadre, d'un nom d'image, etc. stream Un outil léger pour diffuser en continu un ou plusieurs composants de pixels de l'image ou d'une partie de l'image vers les formats de stockage de votre choix. Il écrit les composants de pixels tels qu'ils sont lus à partir de l'image d'entrée, une rampe à la fois, ce qui rend le flux souhaitable lorsque vous travaillez avec des images de grande taille ou lorsque vous avez besoin de composants de pixels bruts.	Extraire les images d'un GIF <code>convert -coalesce infille.gif img_md.png</code> Faire un GIF avec les pages d'un PDF <code>convert -verbose -delay 50 -alpha off -background #FFFFFF -loop 0 -density 300 infille.pdf outfile.gif</code> VIGNETTES Convertir une page de PDF en PNG <code>convert -r1a infille.pdf[0] -quality 100 -flatten -resize x300 outfile.png</code> AUTRES MÉDIAS : VIDEO/Son AUDIO Convertir un mp3 en wav <code>mpg123 -w outfile.wav infille.mp3</code> VIDÉOS Convertir un avi en mov <code>ffmpeg -i input.avi -acodec libmp3lame -ab 192 output.mov</code> Découper une vidéo à 24'53 <code>ffmpeg -i video.mp4 -acodec copy -vcodec copy -ss 0 -t 00:24:53 video1.mp4 (n)</code> <code>ffmpeg -i video.mp4 -acodec copy -vcodec copy -ss 00:24:53 video2.mp4</code> MEDIAINFO Afficher toutes les infos d'un fichier <code>mediainfo file.png</code> (Si il n'est pas installé :) <code>sudo apt install mediainfo</code> SSH Se connecter en SSH <code>ssh user@host</code> Envoyer un fichier par SSH <code>scp infile user@host:directory/</code> ou en sens inverse <code>scp user@host:directory/infile .</code> Envoyer un dossier par ssh <code>scp -r dir user@host:directory/</code> RSYNC Synchroniser 2 répertoires <code>rsync -a --stats --progress --delete \$SRC \$DST/</code> Synchroniser 2 répertoires par SSH <code>rsync -azv --delete -e ssh \$SRC user@host: \$DST/</code> Sauvegarde incrémentale <code>rsync -abz --stats --progress --delete --suffix="d" -b --backup-dir=\$ (base name \$SRC) "\$ (date +%Y-%m-%d) \$SRC \$DST/</code> GIF & Co. Faire un gif avec une collection d'images <code>convert -delay 20 -loop 0 img_*.gif output.gif</code> 8	VARIABLES RÉSERVÉES <code>\$?</code> Code retour de la fonction précédente <code>\$0</code> Nom la commande, du script lancé <code>\$n</code> Appel de l'argument numéroté <code>n</code> passé avec la commande, la fonction ou le script. (Ex argument 1, \$5 argument 5, etc) <code>\$*</code> Affiche le nombre d'arguments passés avec la commande, la fonction ou le script. <code>\$@</code> Appel de l'ensemble des arguments sans séparateurs <code>\$*</code> Appel de l'ensemble des arguments sans séparateurs <code>\$\$</code> Numéro du processus en cours <code>\$!</code> Numéro du dernier processus lancé en arrière-plan GESTION AVANCÉE DES VARIABLES Mettre une variable en lecture seul <code>declare -r var</code> Redonner l'attribut écriture à la variable <code>declare -r var</code> Créer <code>var</code> avec son contenu en minuscules <code>declare -l var = "Hello"</code> Créer <code>var</code> avec son contenu en majuscules <code>declare -u var = "Hello"</code> Créer <code>var</code> avec l'attribut entier 123 <code>declare -i var = 123</code> VARIABLES ET TABLEAUX Affecte le contenu "World" à l'index <code>n</code> de la variable <code>var[n] = "Hello"</code> Déclare un tableau avec les valeurs 1, 2, 3 et 4 <code>declare -a var=(val1 val2 val3 val4)</code> Affecte le contenu "World" à l'index <code>n</code> de la variable <code>echo \${var[n]}</code> TABLEAUX ASSOCIATIFS Déclaration d'un tableau où les variables sont désignés par un nom <code>declare -A colours</code> <code>colours[black]=noir</code> <code>colours[red]=rouge</code> <code>colours[green]=vert</code> SUBSTITUTION DE VARIABLE Si <code>var</code> n'est pas défini, affiche la valeur <code>\${var:-valeur}</code> Si l'argument correspondant à la variable réservée <code>\$n</code> n'est pas défini, utiliser <code>valeur</code> <code>\${n:-valeur}</code> Si <code>var</code> n'est pas défini, affecter puis afficher valeur <code>\${var:=valeur}</code> STRUCTURES DE TEST Test standard, (attention au espace entre les crochets) [condition] Test standard inversé [!condition]
---	---	--

COMPARAISON	
Entier :	
[int1 - eq int2]	Entier1 égal à entier2
[int1 - ne int2]	Entier1 différent de entier2
[int1 - gt int2]	Entier1 supérieure de entier2
[int1 - ge int2]	Entier1 supérieure ou égal de entier2
[int1 - lt int2]	Entier1 inférieure a entier2
[int1 - le int2]	Entier1 inférieure ou égal de entier2
Chaine de caractère :	
[int1 = int2]	Chaine est égal de chaine2
[int1 != int2]	Chaine est différente de chaine2
[int1 - ef int2]	Fichier1 est identique à fichier2
CONDITION	
[- f f1c]	Vrai si le fichier existe
[- f f1c]	Vrai si f1c est un fichier standard
[- d f1c]	Vrai si f1c est un repertoire
[- r f1c]	Vrai si f1c est accessible en lecture
[- w f1c]	Vrai si f1c est accessible en écriture
[- x f1c]	Vrai si f1c est executable
[- s var]	Vrai si la variable existe
[- z var]	Vrai si la variable n'existe ou a une valeur null
LA STRUCTURE IF	
Version 1	
if [condition] ; then	
commande	
commande	
fi	
Version 2	
if [condition] ;	
then	
commande	
commande	
fi	
Imbrication avec elif (si, sinon si)	
if [condition] ; then	
commande	
elif	
commande	
fi	
LA STRUCTURE CASE	
case svar in	
expression1)	
commande	
expression2)	
commande	
expression3)	
commande	
expression4)	
commande	
expression5)	
commande	
expression6)	
commande	
expression7)	
commande	
expression8)	
commande	
expression9)	
commande	
expression10)	
commande	
expression11)	
commande	
expression12)	
commande	
expression13)	
commande	
expression14)	
commande	
expression15)	
commande	
expression16)	
commande	
expression17)	
commande	
expression18)	
commande	
expression19)	
commande	
expression20)	
commande	
expression21)	
commande	
expression22)	
commande	
expression23)	
commande	
expression24)	
commande	
expression25)	
commande	
expression26)	
commande	
expression27)	
commande	
expression28)	
commande	
expression29)	
commande	
expression30)	
commande	
expression31)	
commande	
expression32)	
commande	
expression33)	
commande	
expression34)	
commande	
expression35)	
commande	
expression36)	
commande	
expression37)	
commande	
expression38)	
commande	
expression39)	
commande	
expression40)	
commande	
expression41)	
commande	
expression42)	
commande	
expression43)	
commande	
expression44)	
commande	
expression45)	
commande	
expression46)	
commande	
expression47)	
commande	
expression48)	
commande	
expression49)	
commande	
expression50)	
commande	
expression51)	
commande	
expression52)	
commande	
expression53)	
commande	
expression54)	
commande	
expression55)	
commande	
expression56)	
commande	
expression57)	
commande	
expression58)	
commande	
expression59)	
commande	
expression60)	
commande	
expression61)	
commande	
expression62)	
commande	
expression63)	
commande	
expression64)	
commande	
expression65)	
commande	
expression66)	
commande	
expression67)	
commande	
expression68)	
commande	
expression69)	
commande	
expression70)	
commande	
expression71)	
commande	
expression72)	
commande	
expression73)	
commande	
expression74)	
commande	
expression75)	
commande	
expression76)	
commande	
expression77)	
commande	
expression78)	
commande	
expression79)	
commande	
expression80)	
commande	
expression81)	
commande	
expression82)	
commande	
expression83)	
commande	
expression84)	
commande	
expression85)	
commande	
expression86)	
commande	
expression87)	
commande	
expression88)	
commande	
expression89)	
commande	
expression90)	
commande	
expression91)	
commande	
expression92)	
commande	
expression93)	
commande	
expression94)	
commande	
expression95)	
commande	
expression96)	
commande	
expression97)	
commande	
expression98)	
commande	
expression99)	
commande	
expression100)	
commande	
expression101)	
commande	
expression102)	
commande	
expression103)	
commande	
expression104)	
commande	
expression105)	
commande	
expression106)	
commande	
expression107)	
commande	
expression108)	
commande	
expression109)	
commande	
expression110)	
commande	
expression111)	
commande	
expression112)	
commande	
expression113)	
commande	
expression114)	
commande	
expression115)	
commande	
expression116)	
commande	
expression117)	
commande	
expression118)	
commande	
expression119)	
commande	
expression120)	
commande	
expression121)	
commande	
expression122)	
commande	
expression123)	
commande	
expression124)	
commande	
expression125)	
commande	
expression126)	
commande	
expression127)	
commande	
expression128)	
commande	
expression129)	
commande	
expression130)	
commande	
expression131)	
commande	
expression132)	
commande	
expression133)	
commande	
expression134)	
commande	
expression135)	
commande	
expression136)	
commande	
expression137)	
commande	
expression138)	
commande	
expression139)	
commande	
expression140)	
commande	
expression141)	
commande	
expression142)	
commande	
expression143)	
commande	
expression144)	
commande	
expression145)	
commande	
expression146)	
commande	
expression147)	
commande	
expression148)	
commande	
expression149)	
commande	
expression150)	
commande	
expression151)	
commande	
expression152)	
commande	
expression153)	
commande	
expression154)	
commande	
expression155)	
commande	
expression156)	
commande	
expression157)	
commande	
expression158)	
commande	
expression159)	
commande	
expression160)	
commande	
expression161)	
commande	
expression162)	
commande	
expression163)	
commande	
expression164)	
commande	
expression165)	
commande	
expression166)	
commande	
expression167)	
commande	
expression168)	
commande	
expression169)	
commande	
expression170)	
commande	
expression171)	
commande	
expression172)	
commande	
expression173)	
commande	
expression174)	
commande	
expression175)	
commande	
expression176)	
commande	
expression177)	
commande	
expression178)	
commande	
expression179)	
commande	
expression180)	
commande	
expression181)	
commande	
expression182)	
commande	
expression183)	
commande	
expression184)	
commande	
expression185)	
commande	
expression186)	
commande	
expression187)	
commande	
expression188)	
commande	
expression189)	
commande	
expression190)	
commande	
expression191)	
commande	
expression192)	
commande	
expression193)	
commande	
expression194)	
commande	
expression195)	
commande	
expression196)	
commande	
expression197)	
commande	
expression198)	
commande	
expression199)	
commande	
expression200)	
commande	
expression201)	
commande	
expression202)	
commande	
expression203)	
commande	
expression204)	
commande	
expression205)	
commande	
expression206)	
commande	
expression207)	
commande	
expression208)	
commande	
expression209)	
commande	
expression210)	
commande	
expression211)	
commande	
expression212)	
commande	
expression213)	
commande	
expression214)	
commande	
expression215)	
commande	
expression216)	
commande	
expression217)	
commande	
expression218)	
commande	
expression219)	
commande	
expression220)	
commande	
expression221)	
commande	
expression222)	
commande	
expression223)	
commande	
expression224)	
commande	
expression225)	
commande	
expression226)	
commande	
expression227)	
commande	
expression228)	
commande	

FONTMANAGER [TYPO] Gestionnaire de typographie, permet devaliser, trier, classer les typographies de son système. Version de la distribution : sudo apt install font-manager Dernière version : sudo add-apt-repository ppa:font-manager/staging sudo apt-get update sudo apt-get install font-manager	HANDBRAKE [VIDEO] Logiciel de conversion video simple et puissant https://handbrake.fr/ Version de la distribution : sudo apt install handbrake Dernière version disponible : sudo add-apt-repository ppa:stebbins/handbrake-releases sudo apt-get update sudo apt install handbrake-gtk handbrake-cli
FONTFORGE [TYPO] Editeur typographique historique et puissant sous linux apt install fontforge	CONVERSEEN [VIDEO] Logiciel de traitement d'image part lot. apt install converseen
BIRDFONT [TYPO] Editeur typographique simple Version de la distribution : apt install birdfont Dernière version : https://birdfont.org/	MYSTIQT VIDEO CONVERTER [CONVERSION VIDEO] Conversion video bien pratique https://mystiqapp.com/ Section download et choisir l'Appimage (le plus simple) PROCESSING [GRAPHIC AP] Ide de programmation simple dédié aux applications graphiques https://www.processing.org/
KRITA [IMAGE] Logiciels libres pour les Concept Art, la peinture numérique et les illustrateurs. Version de la distribution : sudo apt install krita	ARDUINO [ELECTRONIQUE AP] Ide de programmation simple pour carte électronique https://www.arduino.org/ Version de la distribution : sudo apt install arduino
QSTPMOTION [ANIMATION] Logiciel d'animation stop motion https://www.qstpmotion.org Version de la distribution : sudo apt install qstpmotion	LOVE2D [MOTEUR DE JEU] Moteur de jeu vidéo 2D avec compilé avec amour, scriptable en LUA. https://love2d.org/
SYNFIG [ANIMATION] Logiciel d'animation vectoriel https://www.synfig.org/ Version de la distribution : sudo apt install synfig	FRITZING [DESSIN DE CARTE ELECTRONIQUE] Logiciel simple de conception de carte électronique https://fritzing.org/home/ Version de la distribution : sudo apt install fritzing
PENCIL2D [ANIMATION] Logiciel pour dessin animé https://www.pencil2d.org/ Version de la distribution : sudo apt install pencil2d	IANNIX [CONTROLEUR OSC] Editeur / Contrôleur OSC basé sur la philosophie de Iannis Xenakis, fonctionne avec processing, etc... Crée et maintient par l'IRCAM https://www.iannix.org/fr/download-iannix/
OPENTOONZ [ANIMATION] Le logiciel d'animation utilisé par le studio Ghibli. https://opentoonz.github.io/ Version de la flatpack : flatpak install flathub io.github.Opentoonz	GODOT [MOTEUR DE JEU] Moteur de jeu vidéo complet et très prometteur https://godotengine.org/download/linux Dernière version disponible : https://godotengine.org/download/linux
INSTALLER UNE IMPRIMANTE PDF [GRAPHIQUE] sudo apt-get install printer-driver-cups-pdf	UNITY [MOTEUR DE JEU] Moteur de jeu le plus utilisé actuellement, se code en C#. (C sharp) https://unity3d.com/ Choisir la version Unity Hub ici : https://unity3d.com/fr/get-unity/download

15

THONNY [IDE PYTHON] Ide de programmation simple dédié à Python https://thonny.org/ Version de la distribution : sudo apt install thonny BLUEJ [IDE JAVA] Ide de programmation simple dédié à Java Allez sur le site et choisir l'option .deb https://www.bluej.org/	BALENAETCHER [UTILITAIRE] Utilitaire permettant de faire des clés de démarrage simplement, parfait pour faire des images RaspberryPi. Télécharger l'Appimage sur : https://www.balena.io/etcher/
CODE VISUAL STUDIO [IDE MULTILANGAGE] Ide de programmation polyvalent et supportant le java, javascript, C, C++, C#, python, html,css,sass.... bref presque tout. Allez sur le site et choisir l'option .deb https://code.visualstudio.com/	MUSIC FOR PROGRAMMING [MUSIQUES] Pour rester zen quand on est concentré devant son écran http://musicforprogramming.net/
GEDIT [TEXT] Editeur de texte simple et efficace, supporte la colorisation syntaxique de code sudo apt install gedit	KMPLOT [MATHEMATIQUE] Visualisateur de courbe mathématique sudo apt install kplot
GHOSTWRITER [TEXT] Editeur markdown http://wereturtle.github.io/ghostwriter/	ARDOUR [COMPOSITEUR MUSICALE] Arduor est une station audio-numérique libre souvent présentée comme une alternative libre au logiciel Pro tools. Version de la distribution : apt install arduor Dernière version : https://community.ardour.org/download
REMARKABLE [TEXT] Editeur markdown https://remarkableapp.github.io/linux.html	PUREDATA [SYNTHESE MUSICALE] Pure Data (en abrégé pd) est un logiciel de programmation graphique pour la création musicale et multimédia en temps réel. Version de la distribution : sudo apt install pd
RETEXT [TEXT] Editeur markdown https://github.com/retext-project/retext	AUDACITY [TRAITEMENT DE SIGNAL AUDIO] Audacity est un logiciel d'enregistrement de son numérique et d'édition de sources audio numériques sous différents formats (mp3, Wave, AIFF, Flac, Ogg...).
VLC [LECTEUR VIDEO] Célèbre lecteur vidéo de l'école Central Paris https://www.videolan.org/ Version de la distribution : sudo apt install vlc	LMMS [COMPOSITEUR MUSICALE] Linux Multi Media Studio, ou LMMS, est une station audio numérique, séquenceur, synthétiseur et compositeur pour GNU/Linux, Windows et MacOS. Télécharger l'Appimage sur : https://lmms.io/download#linux
GIMGAREADER [LECTEUR OCR] Gimgareader est un lecteur OCR, il permet de reconnaître du texte sur des documents. Très pratique pour les document scannés. https://github.com/manisandro/gimgareader Version de la distribution : sudo apt install gimgareader	PARLATYPE [LECTEUR AUDIO] Parlatype est un lecteur audio minimal pour la transcription manuelle de la parole. Il lit les sources audio pour les transcrire dans votre application texte préférée. https://gkarsay.github.io/parlatype/
LIBREOFFICE [BUREAUTIQUE] Suite bureautique complète et opensource https://fr.libreoffice.org/ Version de la distribution : sudo apt install libreoffice	SAYONARA-PLAYER https://sayonara-player.com/
PSENSOR [UTILITAIRE] Utilitaire permettant de surveiller son ordinateur, température, activité. apt install psensor	Version de la distribution : sudo add-apt-repository ppa:gaber-karsay/parlatype sudo apt-get update sudo apt install parlatype

16

SOUNDCONVERTER [CONVERTISSEUR AUDIO] Logiciel pour convertir des fichiers sons simplement. apt install soundconverter
SOUNDKONVERTER [CONVERTISSEUR AUDIO] Logiciel pour convertir des fichiers sons simplement. apt install soundkonverter
SOUND JUICER [UTILITAIRE AUDIO] Logiciel pour riper un CD audio apt install sound-juicer
VcV Rack [SYNTHESEUR MODULAIRE] Synthétiseur modulaire virtuel https://vcvrack.com/
JITSI [COMMUNICATION] Skype en libre pour linux https://jitsi.org/downloads/
SIMPLE SCREEN CAPTURE [CAPTURE D'ECRAN EN VIDEO] SimpleScreenRecorder peut capturer un enregistrement audio-vidéo de tout ou partie de l'écran de l'ordinateur ou enregistrer directement des applications OpenGL telles que des jeux vidéo. https://www.maartenbaert.be/simplescreenrecorder/ Version de la distribution : sudo apt-get install simplescreenrecorder Dernière version : sudo add-apt-repository ppa:maarten-baert/simplescreenrecorder sudo apt-get update sudo apt-get install simplescreenrecorder
MAPMAP [MAPPING] MapMap est un logiciel de mapping vidéo gratuit. https://mapmapteam.github.io/
GLMIXER [MIXAGE VIDEO, VJING] GLMixer réalise le mixage graphique en temps réel de plusieurs clips vidéo et de graphiques générés par ordinateur.Déposez les fichiers vidéo dans l'espace de travail de mixage et placez-les dans une zone circulaire pour changer leur opacité; si vous sélectionnez deux vidéos, les déplacer ensemble effectue une transition de fondu. Ce principe se généralise à un grand nombre de vidéos. L'interaction directe avec la vidéo permet d'être rapide et réactif, et de les déplacer et déformer à l'écran. https://sourceforge.net/projects/glmixer/files/Linux/
ALTERNATIVE [UTILITAIRE SYSTEME] Outil graphique pour configurer les alternatives, pour aider l'administrateur système à choisir quel programme devrait offrir un service donné. sudo apt install galternatives

17

QUELQUES "HELLO WORLD" PROCESSING-JAVA HELLO WORLD Processing-Java est un utilitaire permettant de compiler un dossier processing en programme exécutable. Il se trouve dans le dossier processing. Dans le dossier HelloWorld, pour le fichier HelloWorld.pde: <pre>void setup(){ size(800,800); println("Hello World"); } void draw(){ }</pre> Compilation depuis la console: processing-java --sketch="dossier" --run Dans le dossier en cours on peut écrire : processing-java --sketch="SPWD" --run	NODE HELLO WORLD Dans un fichier helloNode.js <pre>const { createServer } = require('http'); //Création du serveur const server = createServer((request, response) => { response.writeHead(200, {'Content-Type': 'text/plain'}); response.end('Hello World\n'); }); server.listen(3000, () => console.log("Adresse du serveur : http://localhost:3000"));</pre> Exécuter le code avec : node helloNode.js et ouvrir un page web à http://localhost:8080/
JAVA HELLO WORLD Dans un fichier HelloWorld.java : <pre>public class HelloWorld{ public static void main(String [] args){ System.out.println("Hello World"); } }</pre> Compilation du code : javac *.java Exécution du code : java HelloWorld	
PYTHON HELLO WORLD Dans un fichier HelloWorld.py : <pre>print('Hello world')</pre> Exécution du code : python3 HelloWorld.py	
C HELLO WORLD Dans un fichier hello.c <pre>#include <stdio.h> int main(int arc,char* arg) { printf("hello, world\n"); return 0; }</pre> Compilation depuis la console : gcc hello.c qui produit un a.out, et pour exécuter : ./a.out	
BASH HELLO WORLD Dans un fichier hello.sh <pre>#!/bin/bash echo 'Hello, world!'</pre> Et pour exécuter : ./hello.sh	

18

SHELL

MODIFIER L'INTERPRÉTEUR :

Interpréteur shell actuellement utilisé : (en général bash)

ps -p \$\$_

Lister les interpréteur sur la machine :

cat /etc/shells

Changer d'interpréteur par default : (et relancer le terminal)

chsh -s /bin/bash

NOTES :

L'ARBORESCENCE DES FICHIERS EXT ORGANISÉE DE LA FAÇON SUIVANTE :

/ : La racine du disque dur
/bin : les commandes de base (mv, rm, ls...)
/dev : les 'devices', accès aux périphériques du système
/dev/stdin : désigne l'entrée standard
/dev/stdout : désigne la sortie standard
/dev/stderr : désigne la sortie d'erreur standard
(L'entrée (stdin), la sortie (stdout) et l'erreur standard (stderr) ont respectivement les descripteurs de fichiers 0, 1 et 2.)
/dev/tty : désigne le terminal auquel vous êtes connecté
/dev/null : Ce fichier se comporte comme un « trou noir » en absorbant toute redirection sans être modifié.
/dev/zero : Ce fichier renvoie des octets nuls à l'infini.
/dev/urandom : Ce fichier renvoie des octets aléatoires à l'infini.
/etc : les fichiers de configuration
/etc/crontab : configuration de cron
/etc/fstab : configuration des partitions et des points de montage
/etc/group : la liste des groupes
/etc/hosts : correspondance adresse IP - Nom de machine codé en dur
/etc/passwd : La liste des utilisateurs
/home : les répertoires des utilisateurs
/home/user : Répertoire d'un utilisateur (Vous êtes ici (~*))
/home/user/bin : Répertoire des exécutable de l'utilisateur (ce qui les rends accessible depuis le terminal)
/home/user/Bureau : Bureau de utilisateur
/lib : les librairies
/root : le répertoire de l'utilisateur root
/sbin : les commandes système pour root (fdisk, fsck...)
/tmp : le répertoire des fichiers temporaires (effacés régulièrement)
/usr : les programmes pour les utilisateurs
/usr/bin : les programmes de base des utilisateurs
/usr/include : les fichiers d'en-tête pour la programmation
/usr/lib : les fichiers des librairies pour l'utilisation des programmes
/usr/local : les programmes compilés ou écrits en local
/usr/sbin : les programmes d'accès au système pour les utilisateurs
/usr/src : les sources du kernel (si installées)
/usr/X11R6 : les programmes concernant l'interface graphique
/var : les données variables (les logs, les spools, les mails...)

DESCRIPTION DE LA HIÉRARCHIE DU SYSTÈME DE FICHIERS

man hier

RACOURCIS CLAVIERS SUR UN TERMINAL

GÉNÉRAL

Tab : Autocomplétion de mot/programme et de dossier/fichier

Tab Tab : Voir la proposition d'autocomplétion de mot/programme et de dossier/fichier

Ctrl + b : Permetts de reculer le curseur d'un caractère à la fois. (on obtient le même résultat avec la flèche de gauche)

Ctrl + f : Permetts d'avancer le curseur d'un caractère à la fois. (on obtient le même résultat avec la flèche de droite)

Ctrl + d : Efface le caractère sur lequel est positionné le curseur

Ctrl + a : Aller au début de la ligne de commande.

Ctrl + e : Aller à la fin de la ligne

Ctrl + t : Nettoyer l'écran (équivalent de cls ou clear)

COPIER/COLLER

Ctrl + k : Pour couper à partir de là où est positionné votre curseur jusqu'à la fin de la ligne.

Meta + d : Pour couper à partir de là où est positionné votre curseur jusqu'à la fin du mot

Meta + Retour arrière (backspace) : Pour couper à partir de là où est positionné votre curseur jusqu'au début du mot

Ctrl + w : Pour couper à partir de là où est positionné le curseur jusqu'à l'espace précédent

Ctrl + y : Pour coller ce que vous avez coupé

Meta + . : Permet de parcourir les derniers arguments des précédentes commandes pour les replacer dans la commande en cours.

Ctrl + r : Permet de rechercher à partir d'une lettre ou d'un mot une commande précédente. Et si vous appuyez plusieurs fois de suite, vous pouvez parcourir ces résultats.

Ctrl + g : Pour annuler la recherche et revenir à votre ligne en cours.

—

—

**Ateliers réalisés
en école d'art et de design**

—

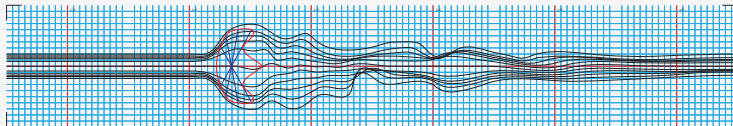
—

What's ^{3d+}
wind
drawing?

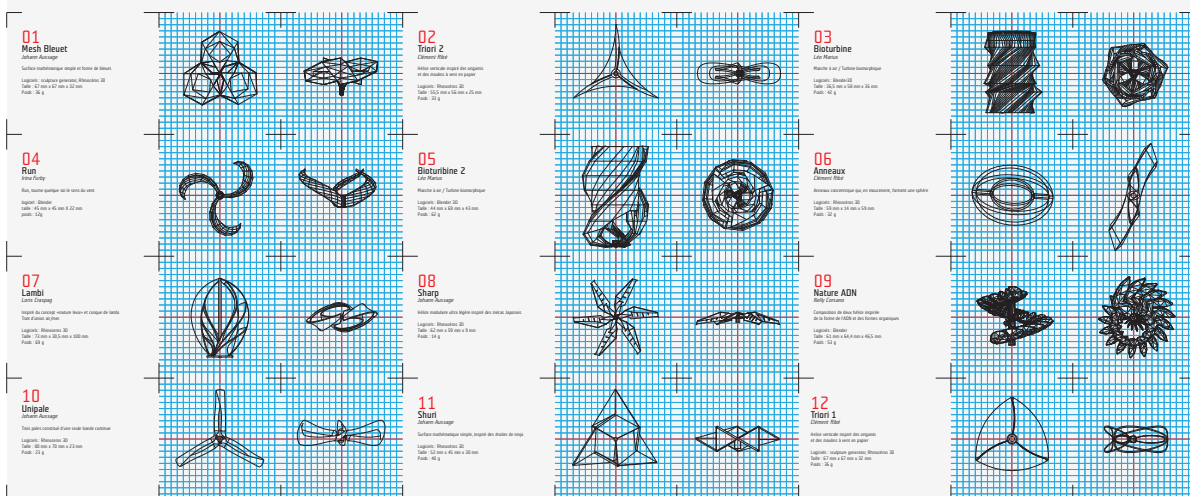


Le vent souffle, la machine dessine...

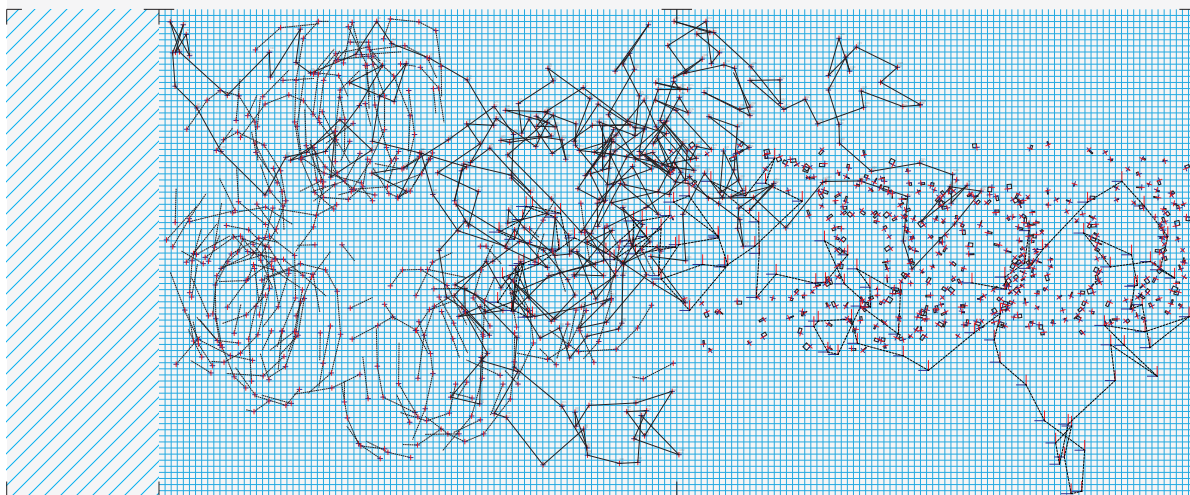
Des éoliennes conçues et prototypées en 3D enregistrent
les mouvements du vent et les transmettent à un ordinateur.
Un programme informatique interprète alors ces informations
pour dessiner simultanément une forme graphique du vent.



PÔLE NUMÉRIQUE DE L'ESADSE, FRANÇOIS BRUMENT - GÉRARD VEROT - DAMIEN BAIS



PROTOTYPING FAN, ARDUINO BOARD, COMMUN COMPUTER, PLOTTER PEN SEKONIC SPL 450, COMMUN PENS AND PENCILS



AFFICHE : DAMIEN BAIS (CONTACT@CORP-LAB.COM)

↑ Affiche du
workshop

WorkShop / Car Factory

Enseignement / Atelier de recherche
2013

Selon le collectif de designers HEHE, initiateurs du projet, la Car Factory peut se décrire de la manière suivante : «[...] l'usine à voiture, est un projet de conception/fabrication de voiture où le véhicule est considéré comme un objet. Plus largement, c'est une recherche sur le design automobile et le rôle de cette industrie en crise. Le projet suggère deux notions: imaginer un catalogue de véhicules et concevoir une ligne de production automobile locale.»

Le projet Car Factory, réinterprété par le Random(lab) se fonde sur la contrainte suivante : créer un véhicule piloté numériquement et pouvant tracter jusqu'à 250 kg à partir de matériaux standards recyclés ou achetés dans la grande distribution pour un budget minime (350 euros). Le véhicule est composé de deux parties : une partie porteuse fabriquée avec des éléments standards (moteur d'essuie-glaces, diables...); une partie benne/cabine en conception libre selon les besoins. Dans la logique de ce projet, la description des éléments nécessaires à

l'élaboration du véhicule et du mode de construction constitue une part du projet aussi importante que le véhicule en lui-même. La mise en ligne du processus de fabrication avec, à la clef, nous l'espérons, des retours d'internautes ayant eux-mêmes construit le véhicule devrait permettre d'estimer l'impact usuel de ce moyen de déplacement et d'en améliorer la conception tout en en diminuant le coût.

Ce futur véhicule ne correspond pas à une «commande» ou à une attente mais est fondé sur l'hypothèse de la transformation d'habitudes de transport par un dispositif constructible par un particulier.

↓ Prototype de
véhicule
autonome



WORKSHOP

NOUVELLE FABRIQUE & ESADSE

Inscription au random() du 11 au 15 février

RDV lundi 18 février au random()
pour le début du workshop

Inscription par mail :

Johann.Aussage@esadse.fr

Damien.Bais@esadse.fr

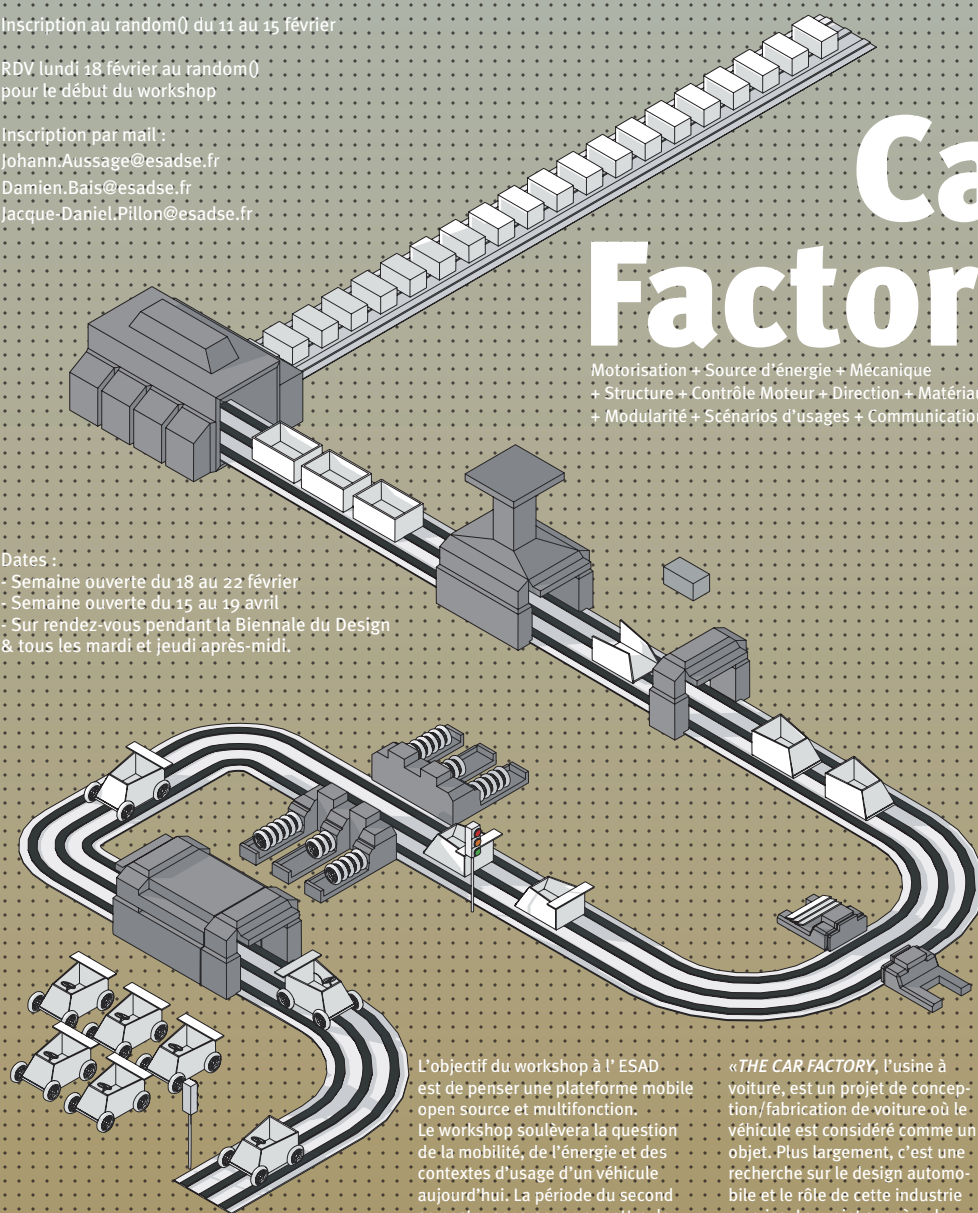
Jacque-Daniel.Pillon@esadse.fr

Car Factory

Motorisation + Source d'énergie + Mécanique
+ Structure + Contrôle Moteur + Direction + Matériaux
+ Modularité + Scénarios d'usages + Communication du projet

Dates :

- Semaine ouverte du 18 au 22 février
- Semaine ouverte du 15 au 19 avril
- Sur rendez-vous pendant la Biennale du Design
& tous les mardi et jeudi après-midi.



L'objectif du workshop à l' ESAD est de penser une plateforme mobile open source et multifonction. Le workshop soulèvera la question de la mobilité, de l'énergie et des contextes d'usage d'un véhicule aujourd'hui. La période du second semestre 2012-2013 permettra de mettre en place la conception d'un système de châssis multi-usages, de présenter les différents scénarios d'utilisation.

«THE CAR FACTORY, l'usine à voiture, est un projet de conception/fabrication de voiture où le véhicule est considéré comme un objet. Plus largement, c'est une recherche sur le design automobile et le rôle de cette industrie en crise. Le projet suggère deux notions : Imaginer un catalogue de véhicules et concevoir une ligne de production automobile locale.»
HEHE

Ecole
supérieure
d'art
et design



Random()

ars longa

NOUVELLE FABRIQUE

↑ Affiche du
workshop

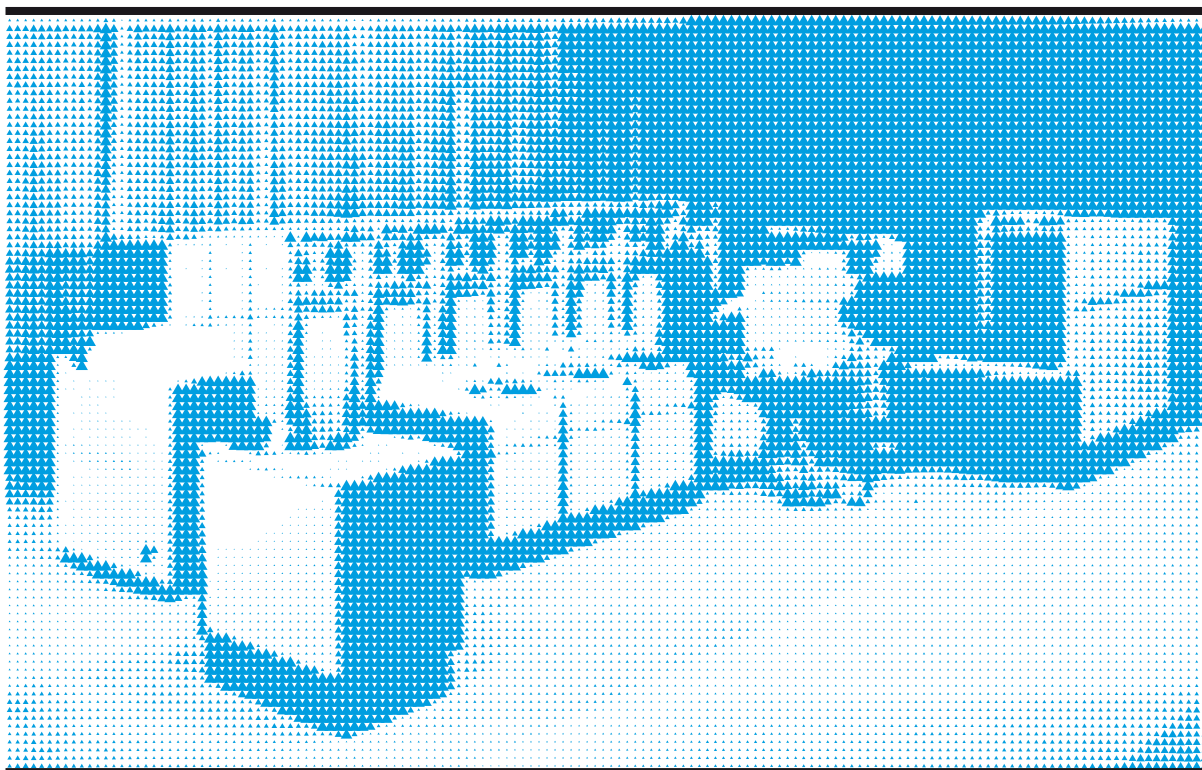
PROCESSING

MASTER CLASS

SEMESTRE 2

Micro WorkShop.scr ScreenSaver

```
//Exercice 0 :  
//1- Allez sous http://processing.org/download/  
//2- Télécharger Processing avec JAVA-Inclus.  
//3- Ouvrir Processing et taper :  
void setup(){  
  frameRate(10);  
  size(screen.width,screen.height);  
  smooth();  
}  
void draw(){  
  background(random(255),random(255),random(255));  
  fill(0);  
  ellipse(width/2,height/2,300,300);  
}
```



```
String Name = new string ("Processing 1.0.9");  
Name += "JoGL";  
Name += "FullScreenLib";  
Name += "Minim";
```

```
String dates= new string ("Session_1 : Intro - 8 Novembre");  
dates += "Session_2 : 12 Novembre";  
dates += "Session_3 : 29 Novembre";  
dates += "Session_4 : 3 Décembre";
```

```
String Date= new string ("Session1 : Intro - 8 Novembre");  
Name += "Session2 : 12 Novembre";  
Name += "Session1 : 29 Novembre";  
Name += "Session3 : 3 Décembre";
```

↘ Affiche d'atelier
pour des cours de
code via
"Processing"

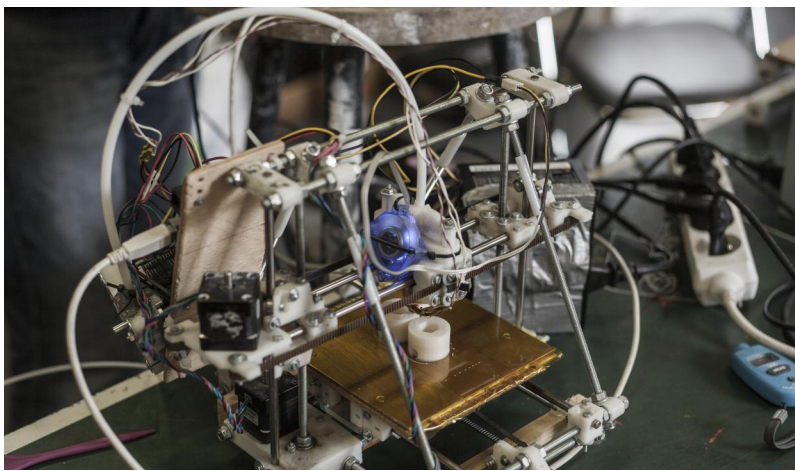
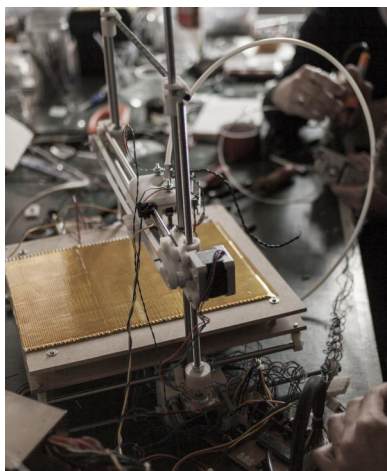
↑ Affiche de
workshop sur les
écran de veille

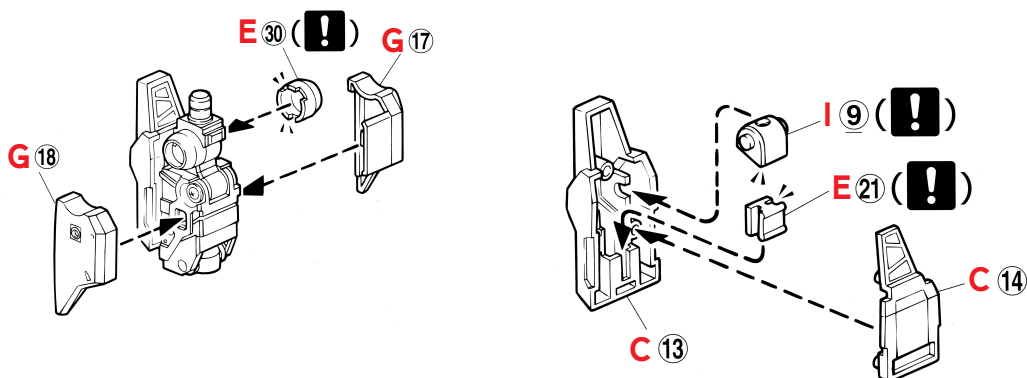
WorkShop / Usinette

Enseignement / Atelier de recherche
2011

Cet atelier a été l'occasion d'inviter des membres du collectif Usinette, connu pour son implication dans la création de nouvelles machines de prototypage. Durant la semaine de présence du collectif, les étudiants ont donc pu fabriquer de toutes pièces une prototypeuse de type "regrap".

↙↘↙ Atelier
prototypage



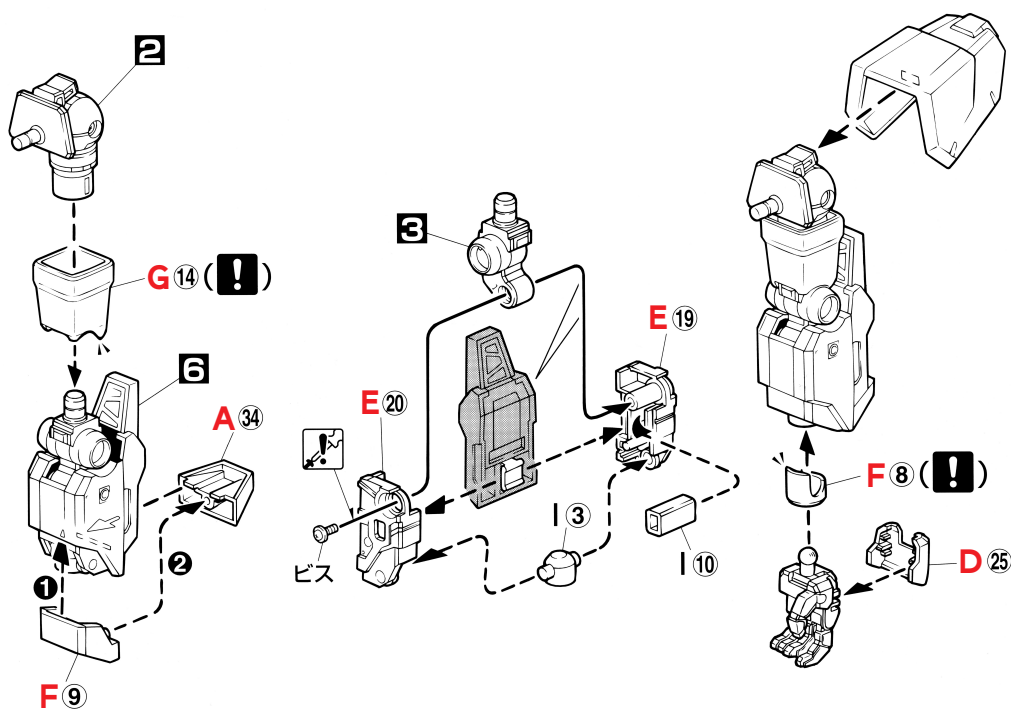


FRANÇOIS BRUMENT - GÉRARD VÉROT - DAMIEN BAÏS - DAVID-OLIVIER LARTIGAUD

PROTOTYPAGE 3D MÉCANIQUE

SEMAINE OUVERTE 12>15.12.2011

Réunion lundi 12 décembre à 10h



PÔLE NUMÉRIQUE / RANDOM (LAB)

† Affiche de
workshop sur le
prototype rapide

WorkShop / GameSession

Enseignement / Atelier de recherche
2012

S'inscrivant dans la lignée d'ateliers sur les jeux vidéo qui eurent lieu à l'École supérieure d'art Annecy Alpes (ESAAA), le Random(lab) a proposé d'organiser une session de travail sur les jeux vidéo à l'ESADSE. Cet atelier s'est penché plus particulièrement sur le phénomène du retrogaming et donna lieu à des productions graphiques en rapport avec ce thème.

→ Vernissage au
9bis



→ Préparation
des jeux vidéo



1\ Vernissage au
9bis et
installation
sonore in-situ





Game Session

9 Bis

Mardi 24, Mercredi 25, Jeudi 26 Janvier 2012

À partir de 20h : Retro Gaming Session

NES, SuperNES, MEGADRIVE, N64, SATURN, PS1/2/3,

DREAMCAST, GAMECUBE, etc. + jeux

Vous pouvez aussi apporter vos jeux préférés !



Jeudi 26 Janvier 2012

À partir de 22h : Concert sidéral « LE VIN DU SOLITAIRE »

Un évènement galactic RANDOM(off)

Dommage, Clément ne sera pas là !

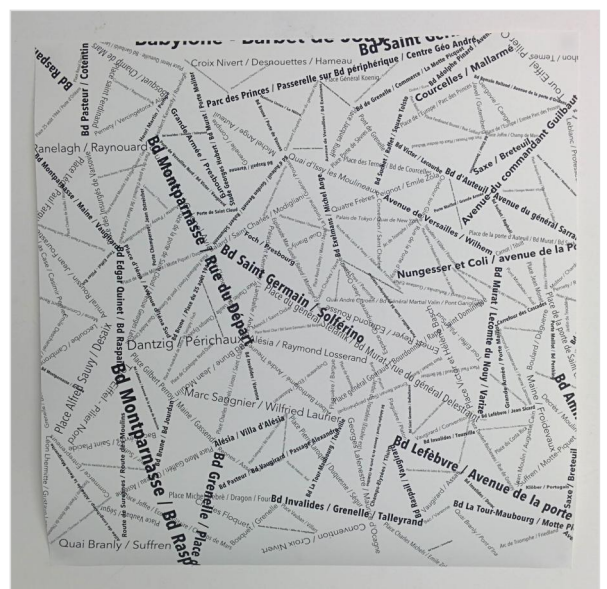
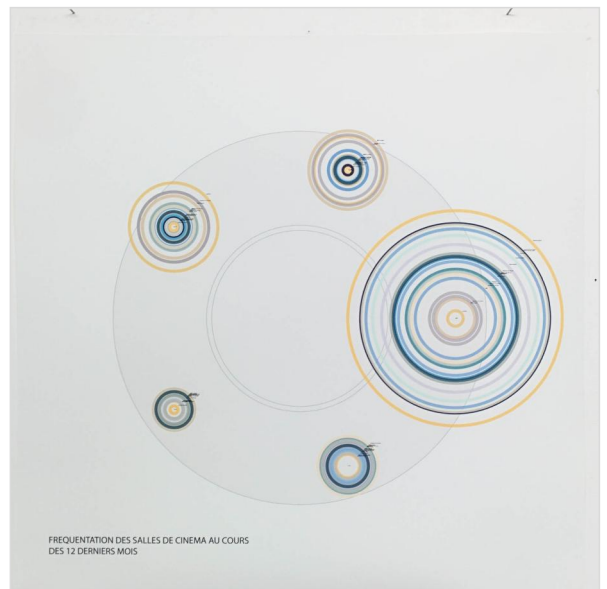
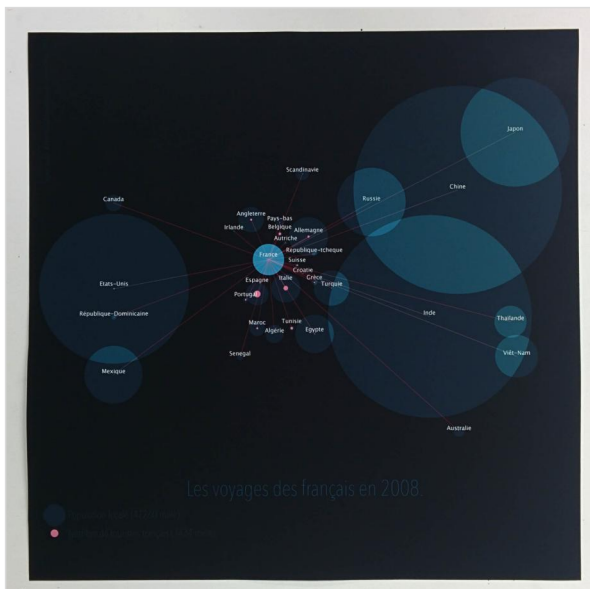
9 Bis, rue François Gillet – Saint-Étienne

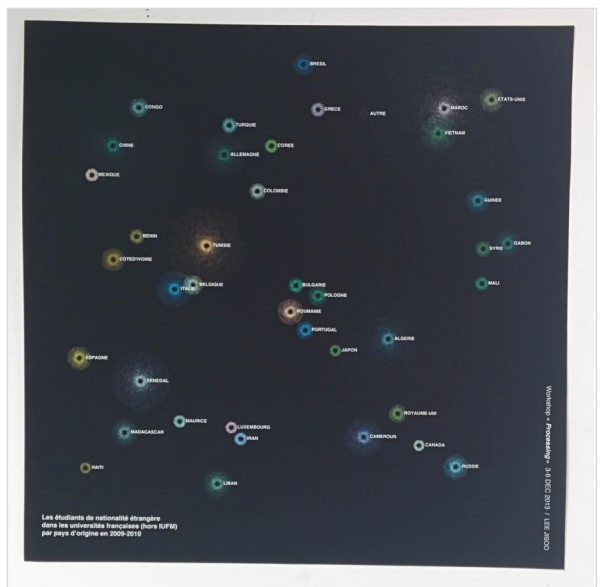
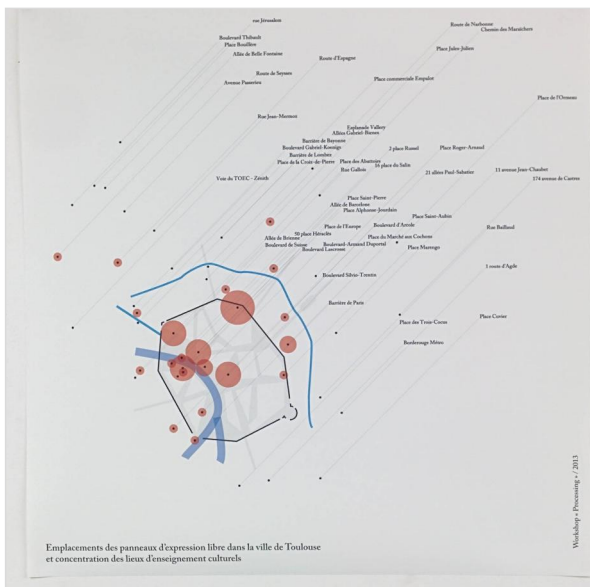
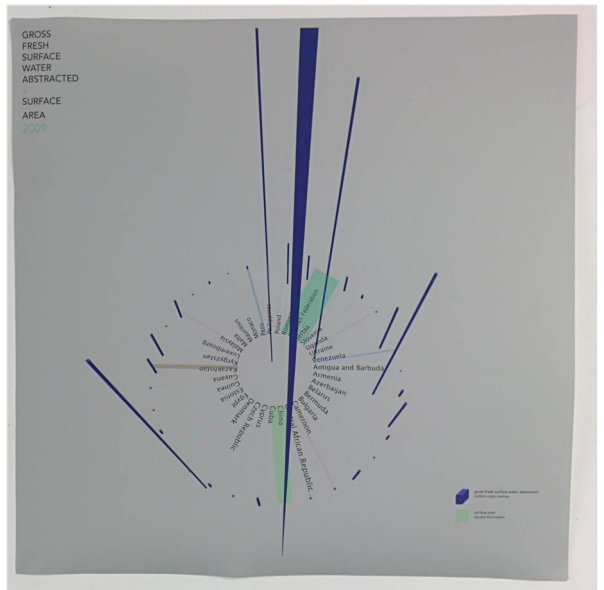
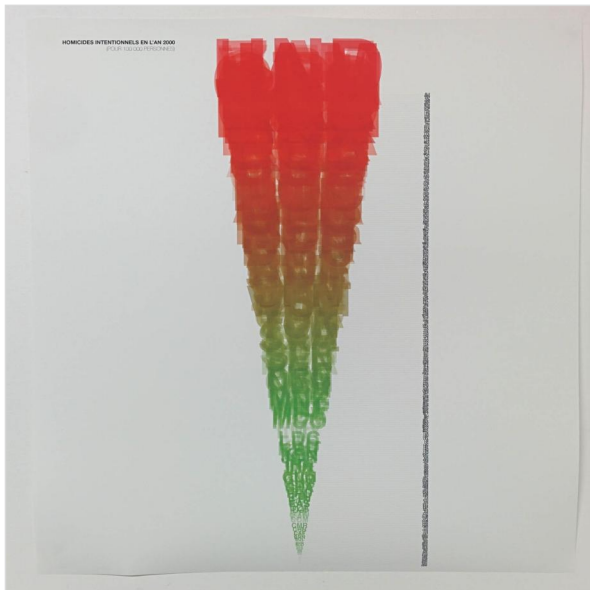
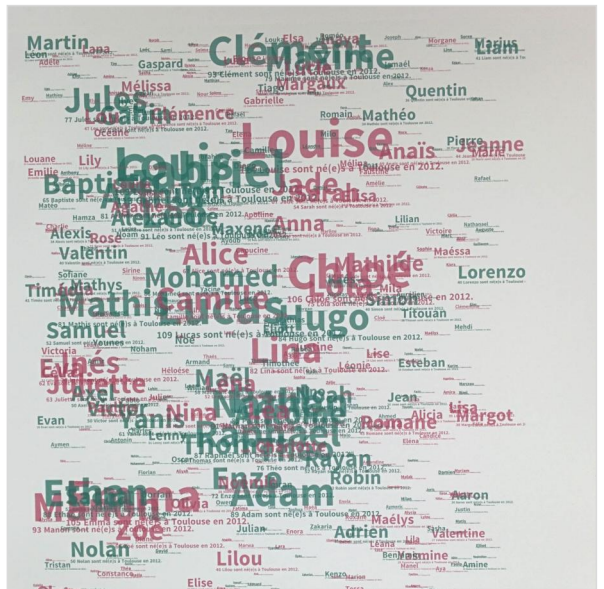
Datavisualisation à l'ISDAT

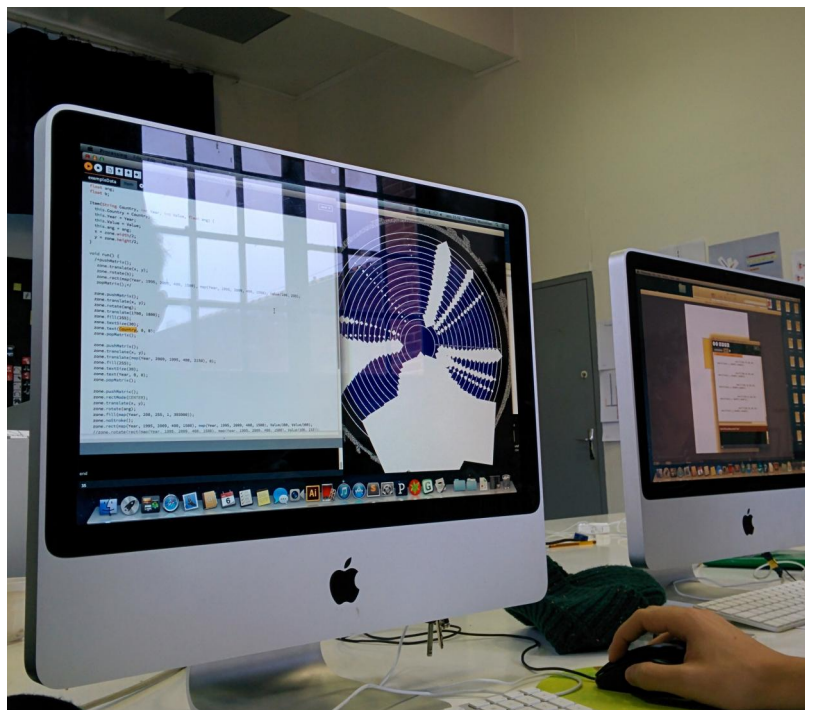
Enseignement / Atelier de recherche
2013

Ce workshop intitulé «data visualisation », était l'occasion de faire découvrir aux étudiants les enjeux des données publiques et leurs potentiels graphiques. À l'issue d'une formation sous processing express en 2 jours, les étudiants ont eu trois jours pour créer des affiches en rapport avec les données publiques. Suivant les données choisies, les étudiants ont mis en forme des programmes permettant de présenter des grandes quantités de d'information. Les résultats sont visibles sur les pages suivantes.









—

—

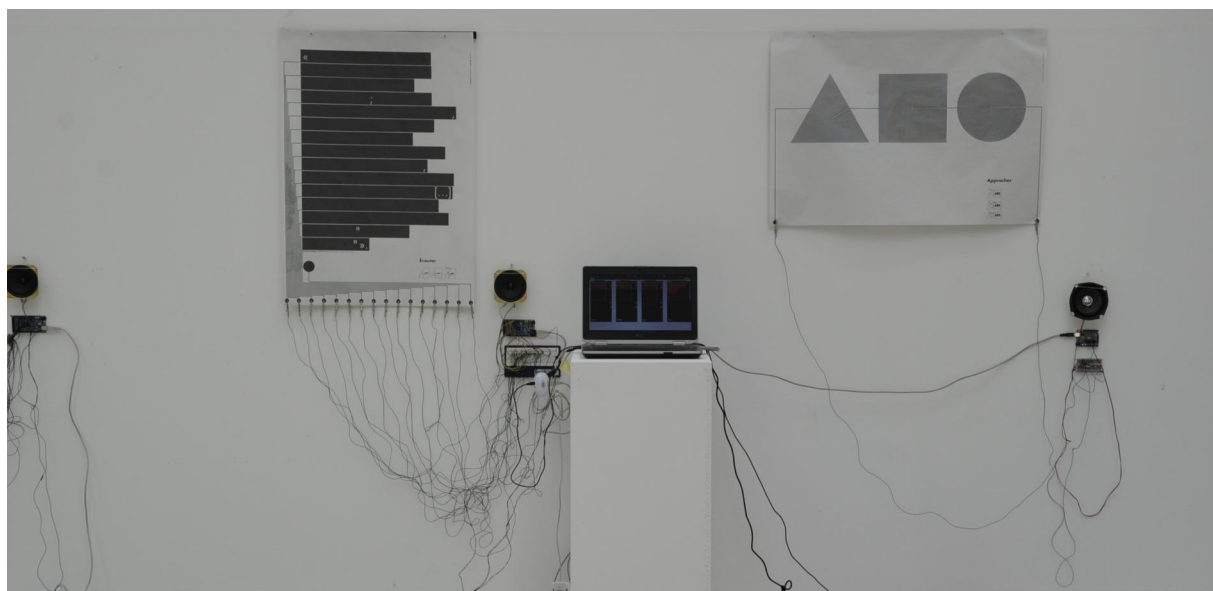
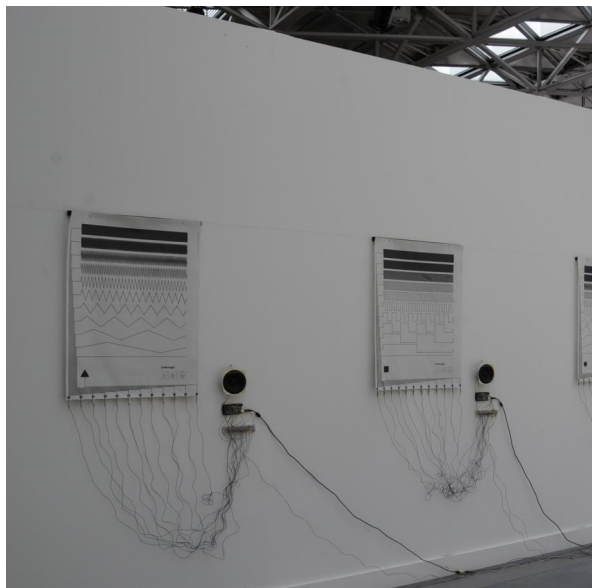
**DNSEP, (Master)
Suivis de diplôme,
étudiants en Design Média**

Suivie de DNSEP

Léo Virieu, Affone

—
DNSEP
2011
—

Affone est un projet d'affiches sonores interactives. Elles sont des objets tangibles (objets physiques, permettant de manipuler du numérique), réalisées à l'aide d'encre conductrice sur du papier cyclus 80g. Le principe est de sortir le numérique de l'écran ; de garder l'aspect premier d'une affiche sérigraphiée avec ses qualités de grammage, de relief, de « toucher », mais en utilisant, pour certaines, des interactions numériques programmées. Affone est une série de huit affiche expérimentant le son : haut parleur, microphone, texte lu, lignes jouées... Elles mettent en scène des actions/réactions avec le spectateur et interrogent le support du numérique, la proximité...



Suivie de DNSEP

Lucile Schrenzel, Jewel

DNSEP
2011

Evo Jewels est une réflexion autour de l'auto conception par les nouvelles technologies notamment le numérique et le prototypage rapide, cela appliqué au domaine de la bijouterie contemporaine. Il s'agit d'un dispositif permettant au futur acquéreur du bijou d'agir directement de sa main sur la forme de son bijou et donc d'obtenir une pièce unique et personnelle. Ce dispositif est composé de différents éléments ; une application numérique, quatre interfaces sensibles en interaction avec l'application et une interface graphique.

Définitions d'une méthode de modélisation du bijou en 4 phases : Nombres d'éléments et positionnement dans l'espace, échelles, modelage et motif. Ces quatre paramètres sont appliqués sur une forme de base type anneau, cube ou autre.

En interaction avec le programme, quatre objets sensibles et intuitifs mettent en valeur la gestuelle et l'action de la main sur le modèle. Dans une volonté de sortir du simple tableau de contrôle classique, pauvre dans sa manipulation, j'ai souhaité réaliser de réels objets à prendre en mains. Dans des formes douces, ils invitent au toucher, à une manipulation sensuelle précise et délicate.

L'interface graphique permet à l'utilisateur, au début du programme, de choisir le style de bijoux qu'il souhaite réaliser. Quatre familles apparaissent à l'écran Ruban, Dentelle, Perle et Cube. Dans chacune d'elles, défilent différents exemples de bijoux réalisés. Une fois que l'utilisateur à cliquer sur la famille de son choix il doit alors décider entre, la création d'un bracelet, de boucles d'oreille, d'une bague, ou d'un pendentif. Il n'a plus qu'à cliquer pour commencer la modélisation de son bijou.



Suivie de DNSEP

Léo Marius, Affone

—
DNSEP
2012
—

Open reflex

L'Open Reflex est un appareil photo argentique entièrement conçu et fabriqué à l'aide d'une imprimante 3D de bureau. Ce projet se situe dans la continuité d'un travail de recherches autour de l'open design (un fichier numérique publié sur Internet de manière à pouvoir être téléchargé, copié, modifié et fabriqué sur une machine à commande numérique, sans outil spécifique) et des usages des imprimantes 3D. L'Open Reflex cherche à mettre en avant la possibilité de produire des objets finis et fonctionnels directement avec ce type de machines.

Il possède un viseur permettant une visée directe, un obturateur mécanique fonctionnant à 1/60° de secondes et il est compatible avec de nombreux objectifs issus d'autres appareils photos reflex, quelle que soit leur marque.

Le choix de fabriquer cet appareil photo argentique en open source a été fait en opposition aux nombreux objets technologiques qui nous entourent et dont le fonctionnement interne nous est très souvent inaccessible, que je qualifie de «boîte noire» (camera obscura). L'objectif était d'ouvrir la boîte noire et de rendre accessible aux utilisateurs l'intégralité de son fonctionnement. L'ensemble des plans nécessaires à sa reproduction sont disponibles librement en ligne. N'importe qui disposant d'une imprimante 3D dans le monde est en mesure d'en fabriquer une copie pour moins de 25€ de matières premières.



Suivie de DNSEP

Martin Guillaumi, Et dans la ...

DNSEP
2012

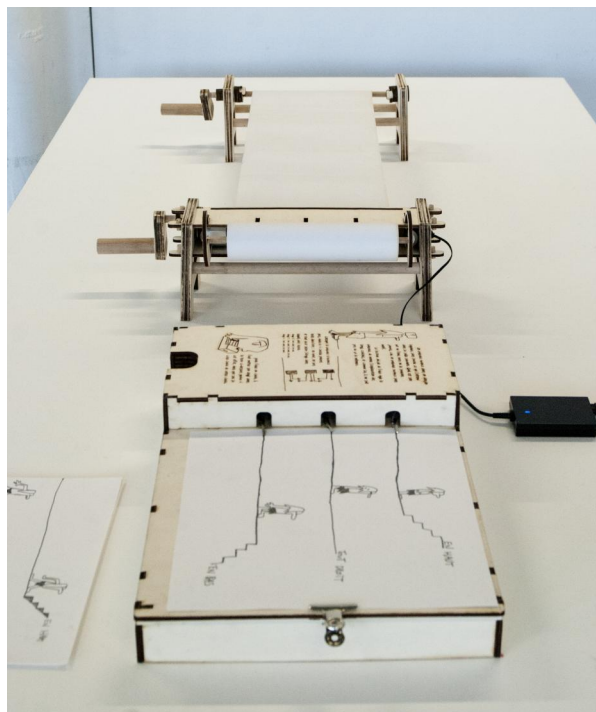
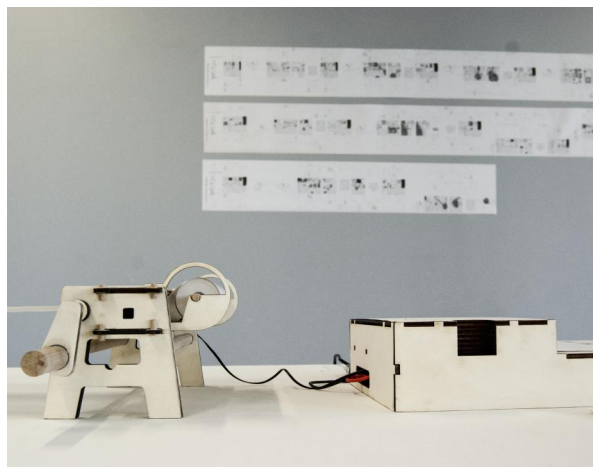
Et... Dans la salle suivante

La Bande Dessinée numérique tend à s'éloigner du papier, qui a pu contraindre le neuvième art. Elle lui préfère l'espace numérique où elle peut se déployer dans des échelles inédites et enrichir la narration d'algorithmes complexes et d'interactions inattendues.

Et dans la salle suivante... est une BD créée selon un processus aléatoire. Elle s'imprime à mesure qu'on la lit. Ce dispositif de « Bande Dessinante », nouvelle forme de cet art narratif, joue à la fois des possibilités sensibles du papier et des possibilités narratives du numérique.

Pour créer une histoire selon un schéma narratif flottant, l'auteur doit laisser intervenir un autre acteur : l'ordinateur. Ainsi redéfinit-il sa pratique et doit créer ses propres outils. La machine s'inspire des presses archaïques pour nier sa propre technicité, rassurer le lecteur. Cette simplicité

apparente l'incite à tourner les rouages grippés de ce vieil instrument. Le lecteur devient acteur, puisque sans lui l'histoire ne peut prendre vie. L'auteur se laisse oublier, au profit de la machine qui devient un inlassable raconteur d'histoires. Chaque lecture nouvelle créera une histoire différente.



Suivie de DNSEP

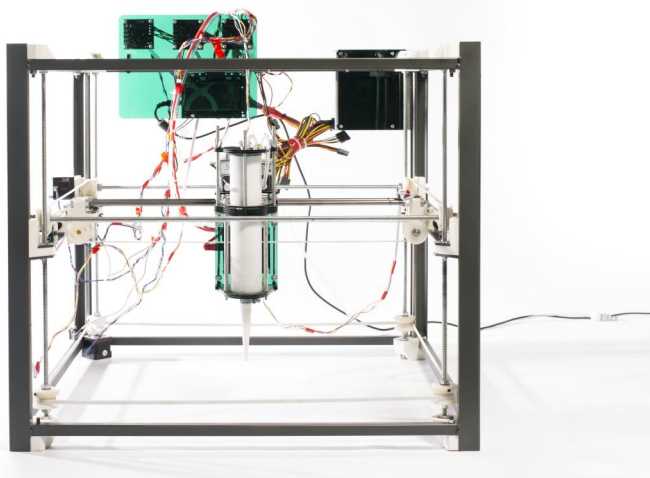
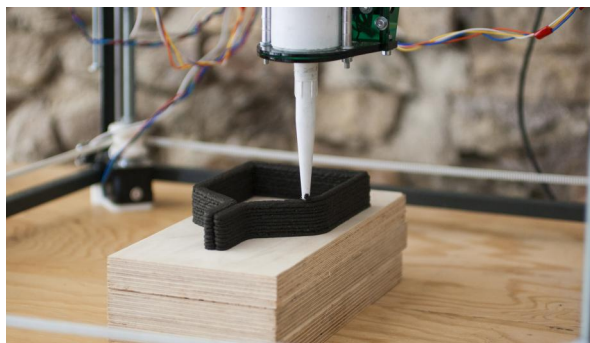
Pierrick Faure, Impression 3D bois

DNSEP
2014

Procédé d'impression 3D de bois

Imaginé comme une expérience d'autoproduction au sein d'un atelier de menuiserie, mon projet, à mi-chemin entre artisanat et fabrication numérique, tend à redéfinir le statut de l'impression 3D en la combinant avec les procédés d'usinage traditionnels du bois. Si chaque découpe, chaque usinage produit des copeaux et engendre de la sciure s'accumulant dans l'atelier, leur recyclage est pourtant incertain. Or, cette poussière est toujours du bois et constitue une ressource locale à profusion dont on ne sait que faire. En mélangeant ces résidus avec certains liants, ils redeviennent matière première au même titre que leur matériau d'origine, et par le biais de l'impression 3D par dépôt de fil, ils permettent d'en tirer des volumes. En jouant sur ses propriétés d'étreinte, la matière extrudée vient s'agglomérer directement sur les pièces usinées afin de créer différentes typologies d'assemblages définitifs. En partant du principe que l'impression 3D ne se justifie pas pour toutes les phases de la production ou que certains usinages sont trop coûteux en matière, il s'agit d'exploiter au mieux les capacités techniques des différents modes de fabrication et leur langage plastique.

C'est cette idée de juste équilibre entre économie de matière et des ressources techniques qui est au coeur de mon projet.



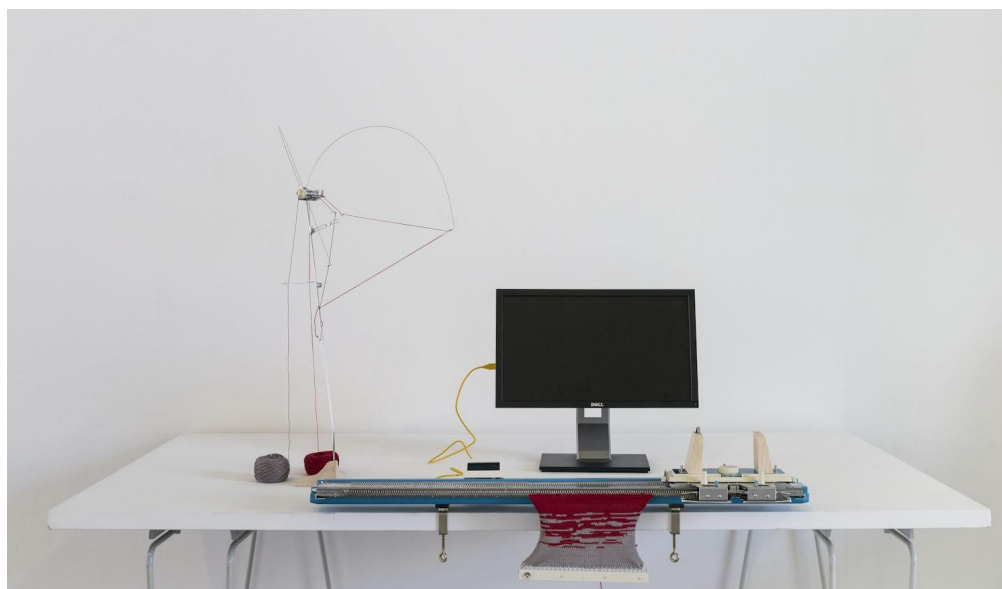
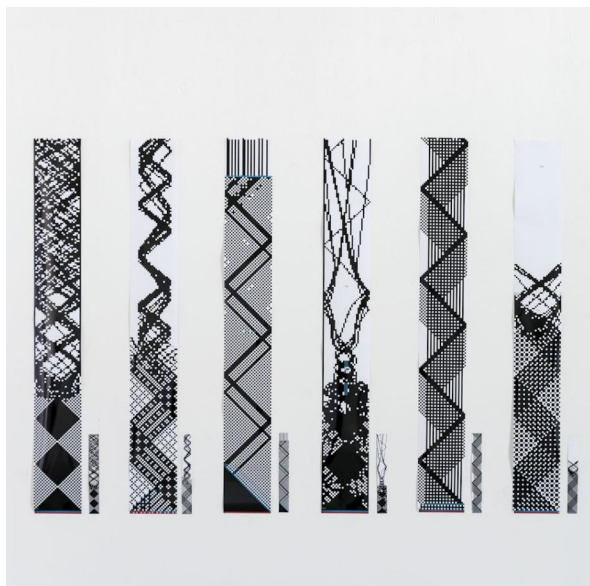
Suivie de DNSEP

Malo Germond, Codons la maille

DNSEP
2015

Codons la maille

Codons la Maille est un périphérique nouveau accompagné d'une application permettant de porter un regard différent sur l'informatique en montrant ce que le code peut nous apporter dans notre utilisation quotidienne de l'ordinateur. L'ordinateur est une machine automatisée amenant à des usages prédéfinis qui ne nécessitent pas de connaître son fonctionnement. Voir le langage informatique, c'est-à-dire le code et sa logique, permet de révéler les répétitions, de les détourner, pour ouvrir l'usage sur d'autres possibilités. L'enjeu de ce périphérique est de permettre à l'utilisateur de comprendre le code et sa logique. Il se lit et s'écrit à travers le regard et la manipulation d'un outil ayant des ressemblances avec l'informatique, la machine à tricoter. En transformant cette machine à tricoter en un périphérique se branchant sur l'ordinateur, on peut voir le code s'exécuter et interagir dessus. On compose un motif issu du code qui sera tricoté par la machine et qui amène à comprendre son fonctionnement de manière concrète et palpable.



Suivie de DNSEP

Camille d'Arondel de Hayes, Écriture lumineuse

DNSEP
2017

Écriture lumineuse

Écriture lumineuse est une réflexion sur le mouvement de la lumière nocturne et sur sa dimension chorégraphique, au travers de notre mobilité et celle de notre environnement.

Dans une époque qui se gorge de vitesses et de techniques toujours plus grandes, j'ai envisagé de nouveaux moyens d'écrire et de composer la lumière urbaine par le biais de dispositifs numériques interactifs. Comme une mémoire de moments vécu : ces dispositifs sont des jeux de sensations sur les images perçus de nos environnements nocturnes, par la lumière. Par

l'installation de systèmes numériques utilisant, l'électronique, le langage de programmation et le prototypage rapide, je porte un autre regard sur l'espace urbain nocturne, par une narrativité de la lumière programmée.

Ce projet évolue dans la perspective de penser l'éclairage non pas seulement comme un objet signalétique mais comme une nouvelle écriture faisant sens avec nos environnements connectés et interactifs. Il réécrit la ville par une poétique de la lumière artificielle.

1. Fluctuations

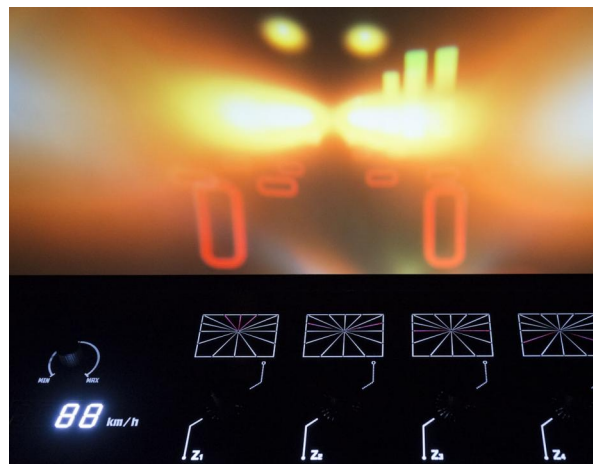
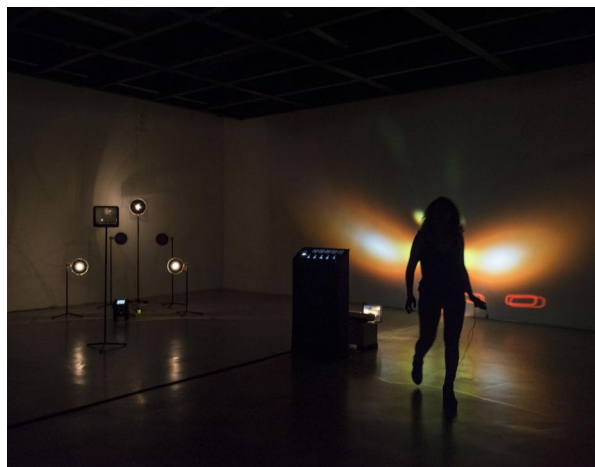
Variations de la densité de flux lumineux causés par l'intervention d'un capteur de présence directionnel.

2. Asphalté

Déambulation aux travers d'espaces virtuels, par le biais d'une manette relié à un programme. Cette manette, permet la variation : de la vitesse de déplacement et de l'architecture lumineuse.

3. Intervalles

Ensemble de cinq objets motorisés, illustrant l'orchestration de rythmiques lumineuses, par la lecture de partition via processing.



Suivie de DNSEP

Mathilde Reynaud, Odysée virtuelle

DNSEP
2019

Odysée virtuelle

Odysée virtuelle résulte de la sélection de trois supports de prédilection de la 3D? (jeu vidéo, réalité augmentée, réalité virtuelle) et permet ici de constituer trois îles de nature tautologique. Elles forment un archipel, qui invite à la traversée et à la pratique du détour, en constituant un corpus dont les expériences s'entrelacent. Les notions abordées par les différents médiums font référence aux trois espaces de perception cités par Maurice Merleau Ponty dans *Phénoménologie de la perception*. Les niveaux de spatialité, les possibles interactions et les degrés de représentation gravitent autour de ces territoires immersifs et l'eau, qui baigne entre les îlots, impose ainsi un devenir aux images.

L'île où gisent des fragments de mémoire est celle du jeu vidéo. C'est un espace mythique, sans cesse mouillé, avec des reflets fabuleux où le paysage se désorganise et s'altère laissant, peu à peu, place à une poétique de la ruine. Les lieux s'effacent alors comme autant de signes distinctifs pour devenir des traces flottantes. Dans la réalité augmentée, la progression de l'eau nous achemine vers l'espace-rêve. Les images des mythes des fonds dormants de l'île flottent au-dessus d'objets-ruines qui s'insèrent dans un devenir archéologique. Puis l'espace du casque de réalité virtuelle se

remplit progressivement, jusqu'à submerger la dernière île dans une eau noire et abyssale qui emporte vers un au-delà de la rêverie, vers un au-delà des formes, vers un monde étrangement informe.

La finalité de cette traversée est moins d'atteindre un but que d'éprouver sa temporalité : c'est un temps d'attente qui permet à la mémoire de se constituer. *Odysée virtuelle* participe ainsi à la construction d'un imaginaire poétique et anthropocénique empreinte d'une mémoire à la fois passée et future. Il s'agit d'une matrice des mondes dans lesquels nous évoluons aujourd'hui.



—

—

—
—

**Recherches/
Textes/
Publications**

Le contrat du jeu vidéo

—
Texte
2008
—

Comment parler d'expansion physique et mentale, dans le cas où le joueur et tous les joueurs, je suppose, accomplissent une sorte de transposition dans le jeu ; il est facile de glisser dans un monde virtuel finalement. Alors que c'est un personnage, pourquoi ne dit-on pas : qu'est-ce que le personnage doit faire ? Finalement le personnage est lié à nos réflexes, à notre dextérité à manier des manettes, un peu comme un instrument de musique. Et aussi, à concevoir un espace qui n'existe pas, qui pourtant paraît être une réalité. Accepter de devenir un autre que soi, juste soi-même comme un autre. À la fois différent et identique. Le jeu permet de déterminer un écart dans ma personne, ma personnalité. C'est dans cet écart probable que je risque ce que je suis. Dans les épreuves du jeu, ce petit moi nouveau doit se consolider dans le temps, mais il se risque à chaque instant dans des épreuves qui se transforment en performance de soi. Pas question simplement de se dépasser, question aussi de durer, dans les péripéties du jeu, ce qui surgit dans le jeu de manière soudaine, brusque et qui étoffe mon devenir de personnage, durer aussi dans le contentement de ce que le jeu apporte. Le contentement, ce n'est pas simplement le pur plaisir d'être ce que je suis en train de devenir dans le jeu, mais aussi se remplir assez : en avoir son content – commencer à se remplir suffisamment pour avoir de la consistance. Du coup, une

résistance aux épreuves qui arrivent à ce "corps-autre-moi" que j'ai accepté de prendre au commencement de la partie. Ce contentement, c'est avoir une certaine masse qui éprouve sa résistance à tenir dans la partie, mais au-delà, dans la totalité du jeu. Ensuite, il faut que cela tienne, que cela continue à durer dans l'épreuve d'un intérêt. Mais là, c'est ce qui se passe dans le jeu comme événement nouveau qui fait que mon attention ne peine pas, ne reste pas en arrière et que du coup, ce corps qui ne se fatigue jamais dans sa dynamique d'action ne me lasse plus vis-à-vis de mon corps réel.

J'entre, j'allume, je suis sous contrat.

Le fait d'appuyer sur ce bouton, c'est signer.

Accepter de croire que je ne suis plus vraiment moi, je suis un autre personnage, fait de pixels, au mieux de vecteurs, dans un monde avec ces propres lois.

Ce contrat ne peut être rompu qu'à une seule condition :

Celle de la lassitude. Tant que je serais performant, je pourrais jouer.

Sinon, il ne m'acceptera plus, je ne pourrais plus avancer.

Je devrais donc rompre le pacte, appuyer sur le bouton – off...



Quand je "joue"

(==play)

Texte
2008

(PLAY) -> Quand je joue, je fais abstraction de mon corps, mes mouvements se limitent à la commande. Dans l'absolu, j'envoie des ordres à mon personnage, dans mon personnage : il effectue les actions instantanément, elles sont simples : avancée, recul, saut, tir. Ces actions sont très proches de celles que je peux demander à propre corps. C'est peut-être pour ça que le personnage devient, pas mon corps, mais un corps dirigeable. Moi, mes mouvements se limitent au clavier, à de micropressions sur des touches, des mouvements rapides, de petites combinaisons. Lui, il effectue des mouvements amples, il court sans fatigue, il saute plus de quatre fois sa propre taille, il peut interagir sur des objets lourds, il traverse des mondes entiers, il voit en permanence de nouveaux paysages. Il voit ? Ou est-ce moi qui voit ? Est-ce bien moi qui le commande d'ailleurs ? Et si Lui, il a une manette et que c'est Lui qui commande mes mouvements, qui me regarde jouer à travers l'écran. Jaloux de ma position confortable, assis sur ma chaise de bureau, comme moi j'envie son extrême vélocité. M'observe-t-il, ce petit bonhomme quadrangulaire ? Être plus malin que lui, brouiller les pistes. Réfléchissons, si je le commande parce qu'il me commande, si je l'observe parce qu'il m'observe... En fait la cible, c'est moi, si je veux l'égarer, je dois brouiller les pistes, même celles qui me guident, hop au code. Il faut épurer les décors, ça sera plus simple. Sinon, il pourra s'y dissimuler. Il faut des ennemis. Mais qu'ils ne manifestent pas vraiment leur hostilité ! Mon personnage ne doit pas savoir que ce sont des ennemis ! Bon, il faut qu'il avance un peu, mais pas trop. Si je l'aide un

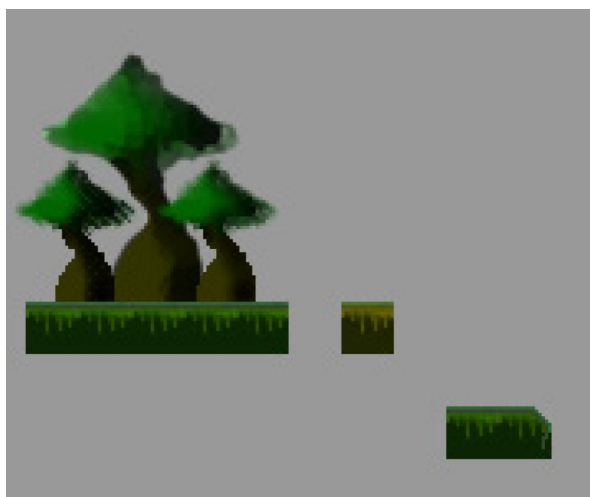
peu, m'aidera-t-il ? Peut-être. Peut-être que si je suis sympa avec lui, il sera sympa avec moi. Si je lui fais sentir que je le commande peu, il me commandera moins... possible? ... Peut-être que si je le commande bien, il me commandera bien. Quand je le commande, il me commande. J'actionne, j'observe, j'anticipe, j'actionne. Cette petite boucle me permet de jouer en instantané. C'est mon programme?



Devant l'écran

Texte
2008

Je suis là, debout, en forme. Devant ce paysage couchant. Le soleil traverse l'épaisse couche de nuages gris. La forêt en contre bas baigne dans une lumière orange. Je vois les derniers rayons lumineux soulever une poussière, celle qui précède ces jours remplis de chaleur. Je commence maintenant à sentir le vent frais, celui apporté par la pluie. Les gouttes commencent à tomber. Une à une, elles ruissellent sur mon visage. Devant ce spectacle naturel, je reste immobile à contempler les éclairs. Je pense à la première fois que j'ai vu un éclair. La lumière aveuglante, je compte, 2, 3, 4, le bruissement sourd. Il est encore loin. Encore un peu de temps, encore quelques minutes pour mes yeux. Ma prochaine mission doit se situer derrière cette montagne. L'écran se fige. Je fatigue, ça sera pour demain. 8 secondes sur le bouton du clavier, on coupe tout, il est 4 heures. Je jette un coup d'oeil à la fenêtre, le même paysage statique depuis des mois. La façade de l'immeuble d'en face. Blanc immaculé. Je soupire.





Game over

Texte
2008

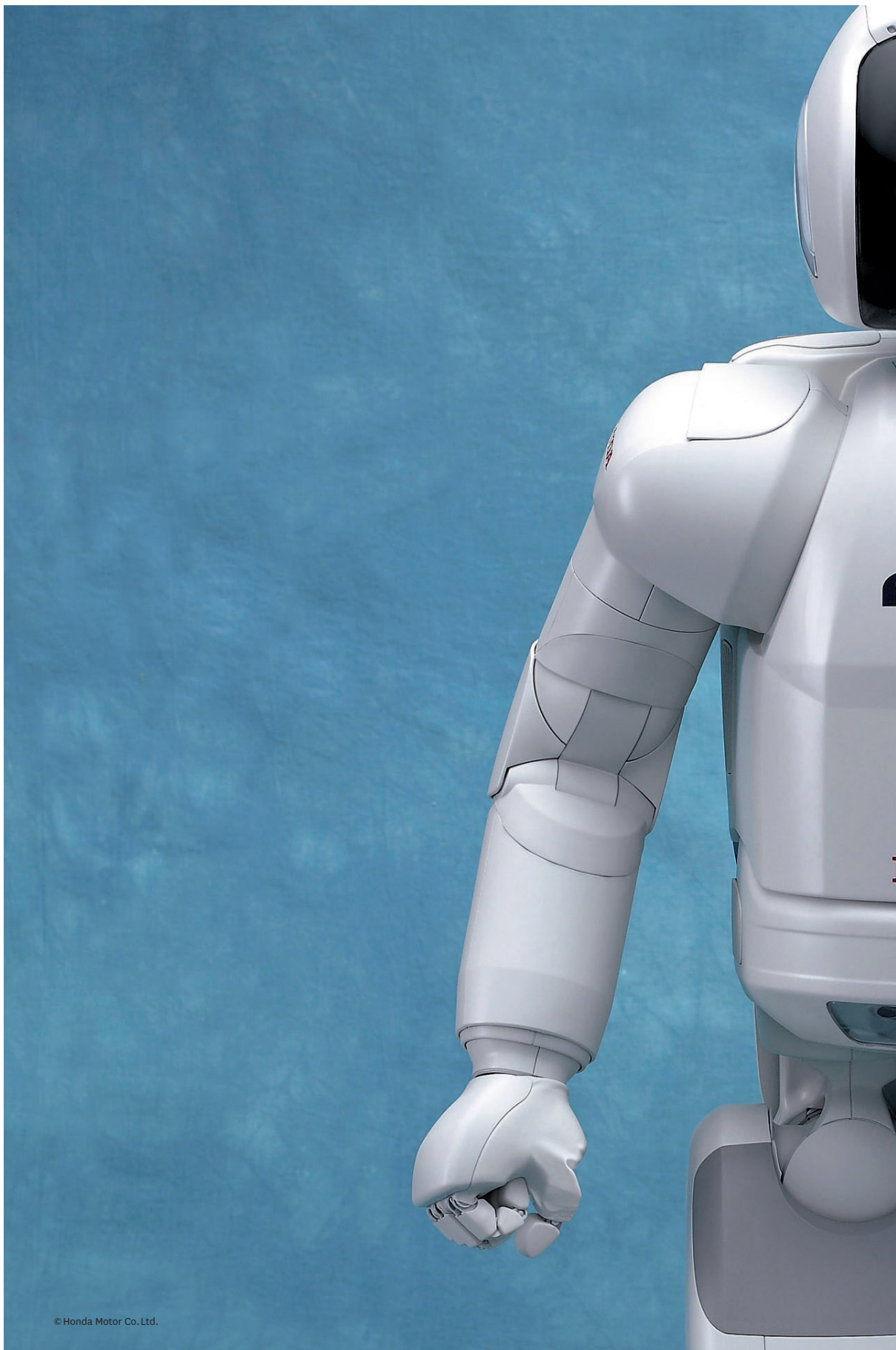
Le joueur suit cette temporalité, il entre dans son système de perception, le moment le plus dur est la sortie. Il existe trois sorties possibles de jeu : La première est la sauvegarde, qui lui permet de figer le monde virtuel dans sa temporalité. De bloquer à la fois l'espace et le temps dans son parcours. Dans un sens, cette sauvegarde permet d'intercaler la vie du jeu avec la vie réel, en figeant le temps de jeu. La deuxième est le petit "Game over", celui qui nous informe que l'on vient d'échouer, que le jeu a été plus fort que nous. Que la dextérité du joueur n'a pas été assez précise.

La troisième, le grand "Game over", la fin du jeu, est plus délicat à appréhender, en effet à force de passer du temps, à vaincre des épreuves, à se divertir et aussi à pousser la virtuosité de jouer, il arrive un moment où le but est accompli, où le générique de fin du jeu apparaît, suivis de sont "Game over". Et bien c'est à cet instant où le joueur meurt. Il vient de passé tout son temps à finir, et c'est cette fin précisément qu'il le bascule dans la réalité. C'est au moment où il est le plus fort qu'il est expulsé du système du jeu, sans même pouvoir en profiter. C'est dans cette petite mort, que le joueur revient du jeu. Bannis parce qu'il a réussi.
GAME OVER



—

—



INTERFACES ENTRE RÉALITÉ ET (SCIENCE-) FICTION

.CORP (INDIVIDU 2) & STUDIO LO

1. Les smartphones font partie des générations évoluées des téléphones portables, notamment grâce à l'ajout d'applications logicielles comme un agenda, un navigateur Web, un GPS, des jeux, un appareil photo, etc.

2. *Radio frequency identification*, ces puces permettent d'envoyer à distance des données informatiques pouvant être lues par des bornes. Les tailles de ces puces sont de l'ordre de quelques centimètres et peuvent être captées à plusieurs mètres.

3. La conception *wearable computer* (Informatique ubiquitaire) désigne l'interconnexion et la communication permanente de plusieurs systèmes informatiques permettant à un environnement de réagir en permanence avec des humains.

Ordinateurs, smartphones¹, bornes interactives, étiquetage RFID², etc. : notre quotidien se peuple à un rythme effréné de nouvelles interfaces numériques. Colonisant peu à peu les objets qui nous entourent, elles nous amènent à une nouvelle réflexion sur le rôle du design, à l'heure où objets du quotidien et informatique pourraient être sur le point de fusionner, rendant intelligent ou communicant ce qui était autrefois inerte.

Les interfaces qui sont appelées à nous assister demain, existent déjà aujourd'hui. Confidentielles ou à l'état de prototypes, sortant des laboratoires universitaires, d'unités de recherche et développement ou de l'imagination d'un designer d'interaction, elles ont en commun l'exigence d'une intuitivité toujours plus poussée. Faisant corps avec nos gestes et nos comportements, elles décuplent notre capacité d'action et de perception et nous ouvrent un monde considérablement enrichi, dans lequel il est possible d'opérer depuis New York une patiente installée dans un bloc opératoire à Strasbourg, d'afficher des informations directement dans notre environnement physique ou de se repérer inconsciemment en n'importe quel endroit du monde.

Cette recherche de fluidité et de simplicité conduit à un élargissement corporel, sensible dans la notion de *wearable computer*³, évident dans le cas des interfaces neuronales du professeur Kevin Warwick, nous faisant entrevoir la possibilité d'interfaces embarquées dans des objets usuels, mais aussi dans les corps.

Mais qu'elles soient ou non invasives, ces nouvelles interfaces sont toujours plus que d'innocentes innovations technologiques. L'*ambient intelligence*⁴, l'Internet des objets, la réalité augmentée et la réalité virtuelle immersive transformeront peut-être profondément la société, voire l'humanité. Des hypothèses développées notamment par des auteurs de science-fiction, reprises aujourd'hui par des designers.

4. *Ambient intelligence* est un nouveau modèle de développement de l'informatique, non plus centré sur des terminaux (ordinateurs), mais déconcentré au travers des objets les plus divers et suivant des logiques d'interactions intuitives et multimodales (tactile, reconnaissance vocale, etc.)

5. Edmond Couchot, *La technologie dans l'art*, Nîmes, Jacqueline Chambon, 1998

Nous sommes environnés par ces nouvelles technologies, mais savons-nous ce qu'est une interface numérique? Si l'on se reporte à Edmond Couchot⁵, ce qui différencie d'abord un marteau d'un ordinateur – qui sont tout deux des interfaces, au sens où ils sont des médiums entre une tâche et un utilisateur – c'est la médiation numérique et non plus analogique de l'information. Au choc violent du marteau s'abattant sur le clou, se substituent des séries discrètes de 0 et de 1.

Les interfaces numériques, par cette transposition computationnelle, dépassent l'interaction et sa causalité physique déterminée, pour atteindre l'interactivité. La reconfiguration continue de l'interface numérique en fonction de notre comportement compense alors le répertoire d'interactions limitées (pousser, tracter, tourner, etc.) des interfaces physiques inertes, capables au mieux d'exécuter des tâches planifiées à l'avance.

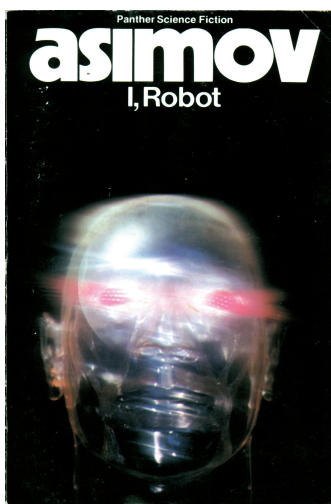
Si l'on peut effectivement identifier des amorces d'interactivité dans des systèmes mécaniques tel que la Pascaline, ceux-ci prennent cependant appui sur un système de transmission de l'information (les engrenages) aussi volumineux que rigide, qui en font des interfaces dédiées, capables seulement d'une interactivité faible et spécifique, là où les interfaces numériques sont capables d'une interactivité forte qu'autorise un support souple indéfiniment reprogrammable.

Concevoir ou travailler avec les objets intelligents, à commencer par l'ordinateur, pose à nouveau au designer la question de la fonction. Sous les traits de l'interactivité, elle est généralement comprise comme une activité d'échanges entre plusieurs entités s'influant mutuellement dans une situation ouverte, l'une de ces entités pouvant être un humain, mais pas nécessairement. On peut d'ailleurs voir dans l'interactivité une simulation d'intersubjectivité, comme Jean-Louis Weissberg, enseignant à Paris 8 et spécialiste des arts numériques, c'est-à-dire prêter un comportement autonome à ces entités, qu'elles soient ou non vivantes. Ce concept a été introduit dès les années quatre vingt pour décrire le nouveau rapport homme-machine marqué par l'apparition d'environnements de dialogue utilisant la puissance des ordinateurs à d'autres fins que le calcul. Ces environnements, dont Windows est l'archétype, permettent à la machine de s'adresser à nous de la manière la plus intuitive possible, l'échange verbal constituant sans doute en la matière l'horizon ultime.

6. La machine de Turing est un modèle abstrait des fonctionnements d'appareils mécaniques servant à suivre des processus de calcul. Créé par Alan Turing, ce modèle sera une inspiration pour fonder l'architecture des ordinateurs modernes.

7. L'Inhumain, causerie sur le temps, Jean-François Lyotard, Paris, Galilée, «débats», 1993, p127.

8. Surface est une table/écran tactile de Microsoft, permettant, en disposant des objets sur sa surface, d'interagir avec des données numériques. Une des grandes nouveautés technologiques de cet objet est la manipulation par plusieurs utilisateurs en même temps.



Couverture du livre *I, robot*. Isaac Asimov présente dans ce livre les lois de la robotique permettant de préserver l'intégrité humaine aux yeux des robots.

Cette activité d'échanges prend appui sur une interface numérique parfois difficile à délimiter, car elle implique corollairement une modification de notre psychologie et de nos postures. Quoi qu'il en soit, ce rapport d'influences mutuelles ne peut s'établir que si la machine est dotée de périphériques d'entrées et de sorties, comme le clavier ou l'écran, lui permettant d'être un milieu ouvert et réactif, un peu à la manière d'un être vivant doué d'organes d'action et de perception. De notre côté, l'effort d'interfaçage est suffisamment important pour qu'Edmond Couchot considère que la relation qui nous lie à la machine aujourd'hui ne soit plus seulement un régime de communication, mais de commutation. Cette «deuxième interactivité», comme il la nomme en référence à la deuxième cybernétique, diffère de la première en ce que la situation interactive n'est plus fondée sur le modèle «émetteur-récepteur» à la Jacobson ou sur celui de la machine de Turing⁶, tous deux pensés sur un mode dialogique, mais sur le modèle d'une corporalité élargie. Il ne faudrait plus alors prêter à la machine le statut de sujet, mais celui d'organe. Nous ferions corps avec la machine. Nul besoin, donc, de mêler la chair et l'électronique comme Kevin Warwick, nous sommes déjà en quelque sorte des cyborgs. Nos gestes, nos comportements, nos manières de réfléchir sont largement dépendants des interfaces numériques... Un rapport d'intimité qui s'intensifie avec la multiplication des systèmes haptiques, fondé sur une prise en compte de toute la richesse sensorielle de l'utilisateur.

Ce statut équivoque de l'interactivité, dans lequel on ne sait si l'on s'adresse à un sujet, à un objet, ou à soi-même, amène le philosophe Jean-François Lyotard à la considérer comme une illusion de réciprocité, mais aussi à y voir un refus de perte de contrôle: «On ne demandait pas des "interventions" au regardeur quand on faisait de la peinture, on alléguait une communauté. Ce qui est visé aujourd'hui... c'est, au contraire, que celui qui reçoit ne reçoive pas, c'est qu'il ne se laisse pas décontenancer, c'est son auto-constitution comme sujet actif par rapport à ce qu'on lui adresse»⁷.

Au-delà de cette critique du refus d'altérité, on peut se demander s'il n'y a pas différents niveaux d'interactivité. En effet, doit-on considérer l'interactivité avec une borne d'informations équivalente à celle entretenue dans un iPhone? Dans un cas, nous sommes confrontés à un système fermé, mono-fonctionnel et, dans l'autre, à un environnement riche, capable de supporter des applications diverses. On pourrait alors distinguer une interactivité faible et une interactivité forte, avec d'un côté des systèmes limités et de l'autre une activité d'échanges variés.

L'interactivité forte n'est pas pour autant la panacée. En effet, l'opposition entre technologies propriétaires et *Open Source*, de même que celle entre utilisateurs experts et simples usagers, pose des questions en termes de citoyenneté numérique et peut faire craindre les implications d'une interactivité forte opaque, donc hors de contrôle. Le risque est d'autant plus grand que les interfaces numériques à interactivité forte, sous couvert de personnalisation des services, détiennent de plus en plus d'informations personnelles à notre sujet et sont de plus en plus invisibles.

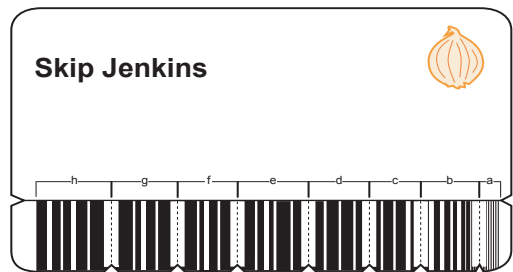
À ce propos, *l'ambient intelligence*, défendant l'intégration des interfaces numériques dans les objets usuels et l'environnement quotidien, est en passe de devenir le paradigme technique dominant et oblige donc le designer à se positionner sur ces questions. James Auger, enseignant au département *design interactions* au Royal College of Art, défendait justement à la dernière conférence Lift, la pertinence d'un design critique, susceptible de mettre ces questions en débat. Ses scénarios d'objets, de la *Dent-téléphone* aux *Augmented animals*, par leur dimension provocatrice, nous rendent présentes ces perspectives futures, renouvelant en quelque sorte le rôle critique du récit d'anticipation.

Dans *Philosophie et science-fiction*, Jean-Noël Missa définit d'ailleurs l'apport du genre SF en termes d'«expériences de pensée spéculative» et Isabelle Stengers y voit un terrain prospectif pour les «sciences humaines et sociales», sur les effets des inventions techno-scientifiques.

La littérature et le cinéma d'anticipation sont, en effet, un matériau de choix pour élaborer une réflexion sur les techno-sciences et on peut considérer que les auteurs de SF classique (Huxley, Wells, Orwell) pensaient déjà en philosophes. Leurs successeurs ont d'ailleurs construit de véritables concepts qui s'imposent aujourd'hui dans les débats sur les nouvelles technologies, à l'exemple du *cyberspace* de William Gibson.

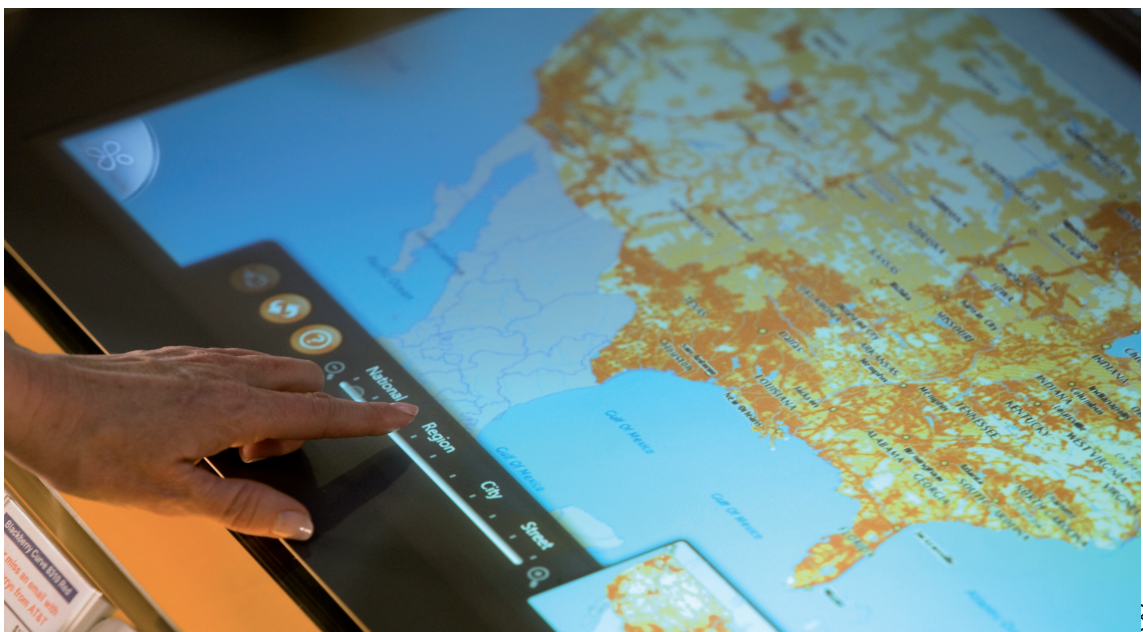
De *Minority Report* à *Surface*⁸ de Microsoft, il semble d'ailleurs que la latence séparant la chose imaginée de la chose réalisée tende à se réduire. Cette accélération technologique nous poussant brutalement dans le futur, nous sommes amenés à nous tourner vers la SF pour en éviter les écueils.

Onion card est un scénario de carte de visite modulable. L'utilisateur peut choisir le niveau d'informations auxquelles il donne accès en retirant des parties de la carte : téléphone, adresse, profession, niveau d'étude, finance, casier judiciaire, préférences sexuelles, etc. © Jimmy Loizeau & James Auger



Capture vidéo du film *Minority Report*. Le modèle d'interface imaginé dans ce film a été une source d'inspiration pour beaucoup de constructeurs et de designers. © Twentieth Century Fox Entertainment

Surface de Microsoft, projet de table tactile multimodal © Microsoft



La société de contrôle

Un des dangers premiers pourrait être l'absolue surveillance, développée magistralement dans le roman *1984* de Georges Orwell, ou plus récemment dans *Minority Report* de Philip K. Dick. Un tel scénario pourrait aisément se produire dans notre société de la communication. À force de multiplier les dispositifs technologiques visant à simplifier notre vie et à assurer notre sécurité (services de paiement, traçabilité des produits, passeport biométrique, etc.), nous jetons les bases, à notre insu, d'une société policière.

Notre vie réelle se duplique déjà dans un archivage d'informations de tout ordre, dans des multiples bases de données. Certes, aujourd'hui, pour consulter et recouper ces bases de données, il faut des autorisations bien particulières qui sont du seul pouvoir des services de renseignements (DST, NSA, etc.) ou des autorités judiciaires. Mais imaginons, à l'extrême, que l'interface technique nous permettant de payer, voyager, se repérer dans l'espace et le temps, de recevoir messages et appels – l'Internet – soit plus facilement consultable. Notre vie serait tracée en permanence, permettant de prévoir tout types de comportements psychotiques, violents ou dissidents. Ces informations accessibles pourraient jouer en notre défaveur durant un entretien d'embauche, ou même pour l'attribution d'une allocation, d'un logement.

Toutes ces métas-données pourraient être couplées avec celles des réseaux sociaux afin de surveiller nos amis et nos activités pour contrôler les mœurs (par exemple le cas des licenciements à cause de Facebook). Et puis pourquoi ne pas vendre ces informations à des sociétés, de façon à ce que les publicitaires restreignent leurs cibles et vendent leurs produits avec de faibles marges d'erreurs? Des événements clairement envisageables, et qui ont déjà cours, notamment à travers les technologies Google. 23andMe, une société sponsorisée par le numéro un des moteurs de recherche, offre le décryptage de son génome pour trois cent quatre-vingt-dix dollars, via un kit de prélèvement. Une porte ouverte à un autre système de discrimination, exploré notamment dans le film *Bienvenue à Gattaca* d'Andrew Niccol. James Auger exprime sa méfiance vis-à-vis de l'idéal de transparence souvent mis en avant dans les interfaces innovantes, à travers son projet *Onion card*. Redirigeant vers des bases de données personnelles, les codes-barres figurant sur cette carte de visite sont détachables, en autorisant ou refusant l'accès : un objet qui permet de ritualiser et de graduer concrètement l'échange de données personnelles en fonction de la confiance entretenue avec son interlocuteur.

Le soulèvement des machines

Un autre scénario pourrait être une pure et simple perte de contrôle de notre environnement technique, conséquence de la «singularité», concept attribué au mathématicien John Von Neumann et popularisé par l'écrivain de science-fiction Vernor Vinge. La singularité désigne un point de bascule hypothétique, à partir duquel la civilisation humaine serait dépassée par les machines.

La loi de Moore prévoit, en effet, que la puissance de calcul des ordinateurs double toutes les 1,92 années. Cette loi se vérifiant depuis l'invention du premier microprocesseur en 1970 jusqu'à aujourd'hui, certains chercheurs pensent qu'en suivant cette évolution, l'ordinateur serait capable de surpasser l'intelligence humaine d'ici quelques décennies.

Depuis plusieurs années, on observe des projets ambitieux de robots capables de se mouvoir de la même manière que des humains. On peut citer le projet *Asimo* de Honda, pour l'instant projet de recherche non commercialisé, ou encore *Aibo* développé par Sony, premier robot-chien de compagnie. Encore plus impressionnant, *BigDog* est un prototype de robot quadrupède de soixante-dix centimètres de haut pouvant transporter un barda de cinquante kilogrammes sur terrain accidenté, impraticable pour des véhicules militaires. L'ère des robots soldats a commencée.

La guerre interfacée – drones, missiles intelligents, vision augmentée – se dessine progressivement sous l'impulsion de grands États militaires soucieux de ne plus porter les hommes sur les zones de combats. Visant le risque-zéro tout autant que l'efficacité, cette guerre de machines pourrait mener à des armes intelligentes capables de prendre des décisions rapides sur le théâtre des opérations, sans intervention humaine. Isaac Asimov, un des grands auteurs de romans d'anticipation, a pressenti ces hypothèses dans une œuvre qui n'est pas seulement critique, puisqu'elle propose un ensemble de règles susceptibles de nous prémunir contre un éventuel soulèvement des machines. Ces lois fondamentales de la robotique constituent souvent une base de débats sur les implications sociétales de l'intelligence artificielle.





MQ-1 Predator fait partie de la première génération de drones de combat mis au point par les États-Unis. Copyleft Wikipedia

Social Tele-presence, projet de James Auger et Jimmy Loizeau, consiste à louer le corps d'une autre personne. La première image présente l'emprunteur, qui perçoit à l'aide d'un casque l'environnement du corps loué, lui permettant ainsi de le diriger.

La deuxième image présente celui qui loue son corps. Il est coupé du monde par un dispositif captant ses stimuli et ses sens, qui sont redirigés vers la première personne dont il dépend. © Jimmy Loizeau & James Auger

L'humain augmenté

Il pourrait être le fruit d'un couplage de notre corps et d'interfaces de visualisation permanentes de métas-informations nous renseignant sur notre environnement. Par exemple, un piéton, grâce à un dispositif de lunettes ou même un écran minuscule sur une lentille, verrait en permanence sa position dans la ville, l'état de ses réseaux sociaux, l'arrivée d'un nouveau mail et pourrait voir s'afficher directement sur la vitrine d'un magasin, de façon interactive, des informations correspondant à son profil d'acheteur.

Les chaussures *CabBoots*, du designer d'interaction Martin Frey, permettant de se repérer intuitivement en n'importe quel endroit du monde (cf. Chronologie des interfaces p115), illustrent bien cette conception de l'humain augmenté (médicaments, prothèses, produit dopants, interventions chirurgicales, etc). Partant du principe que l'humain n'est plus une espèce gouvernée par la nature, depuis l'invention de la médecine, certains auteurs, entrevoient un autre scénario d'humanité augmentée, notamment par l'incorporation de « bio-interfaces ». Jean-Pierre Andrevon, par exemple, imagine des dispositifs de régulation du corps par des interfaces mi-humaines, mi-électroniques. Ces interfaces, sur le modèle du *pace-maker*, permettent de réguler les flux hormonaux, le système immunitaire, de déterminer des activités corporelles défaillantes, afin de prévoir n'importe quel type de maladie, ou de stabiliser le stress. Chez George Alec Effinger, un auteur de SF américain, le scénario va plus loin, puisque les dispositifs d'interface permettent de modifier directement dans les zones cérébrales les systèmes de pensée, modifiant à volonté les capacités intellectuelles ou les traits de caractères.

L'analyse notionnelle des interfaces numériques, l'examen de la littérature SF et du design critique, tout comme la vulgarisation des dernières innovations en matière d'interactivité, permettent d'explorer les implications possibles de ces interfaces. Elles présagent aussi bien des futurs sombres dans lesquels notre société de l'information se muerait en société de surveillance ou deviendrait hors de contrôle, que des futurs radieux, dans lesquels les rapports de l'homme et de son environnement seraient placés sous le signe de l'intuitivité et de la transparence ; des anticipations sans doute un peu radicales, forcément en décalage avec une histoire humaine toujours imprévisible et complexe, mais que les designers ne peuvent ignorer s'ils veulent imaginer les futurs souhaitables de ces technologies.



PETITE CHRONOLOGIE DES INTERFACES

1725 - Basile Bouchon invente le papier perforé, support de mémoire pour des tâches répétitives mécaniques.

1801 - Joseph Marie Jacquard met au point une machine semi-automatique pour les tâches répétitives des métiers à tisser.

1844 - Samuel Morse invente le télégraphe.

1854 - George Boole développe une logique essentielle : l'algèbre de Boole.

1924 - Invention de la télévision cathodique.

1934 - Charles Babbage pose théoriquement les principaux concepts sur lesquels reposent les machines informatiques.

1936 - Alan Turing ébauche la théorie des Machine de Turing, prémisse des calculs informatiques.

1946 - ENIAC est fini d'être assemblé, il sera le premier ordinateur électronique de l'histoire, son interface est faite avec des commutateurs.

1945 - Vannevar Bush publie un article intitulé «*As we may think*» et imagine une machine conceptuelle graphique baptisée *Memex*.

1969 - Alan Kay, du Carnegie Mellon University, propose dans sa thèse l'idée de «fenêtres se superposant les unes sur les autres».

1652

LA PASCALINE

Blaise Pascal



Inventée en 1652 par Blaise Pascal pour alléger le travail fastidieux de son père, collecteur des impôts, la Pascaline peut être considérée comme l'ancêtre de l'ordinateur. Cette calculatrice mécanique, permettant d'automatiser les additions et les soustractions, était d'un usage aussi simple que possible pour l'époque. Pour «rendre le mouvement de l'opération plus simple», Pascal comprit qu'il fallait «que la machine ait été construite d'un mouvement plus composé», une préfiguration de l'approche *user friendly*. La saisie des nombres se faisait à l'aide d'un stylet actionnant les engrennages et le résultat s'affichait sur un totaliseur à tambours. L'objet se présentait dans un coffret compact et solide qui en faisait un calculateur «nomade» susceptible de voyager sans «souffrir de la moindre altération», ce que Pascal expérimenta avec succès sur un trajet aller-retour entre Rouen et Clermont-Ferrand. La Pascaline reçut par la suite de nombreuses améliorations qui conduisirent jusqu'aux premiers supercalculateurs. IBM réalisa dans les années soixante une petite machine mécanique de poche, en plastique, basée sur le principe de la Pascaline pour calculer rapidement en hexadécimales suivant les standard de programmation de l'époque. (Image © Musée des arts et métiers-CNAM, Paris/photo JC Wetzel)

1963

SKETCHPAD

Ivan Edward Sutherland, MIT



SketchPad est la première interface graphique de l'histoire. Développé au MIT sur un ordinateur, ce logiciel de dessin est également une des premières applications interactives. Les supercalculateurs de l'époque fonctionnant alors par traitement automatisé de l'information, la mise au point de *SketchPad* a nécessité des modifications de l'architecture du TX-2 pour qu'il réponde en temps réel aux sollicitations successives de l'utilisateur. Très en avance sur son temps, *SketchPad* permettait de dessiner directement sur un écran cathodique à l'aide d'un stylet optique, de modifier un tracé, de zoomer ou même de copier/coller. Les interfaces graphiques modernes sont redevables à *SketchPad* de nombreux autres concepts fondamentaux, comme la désignation directe des objets à l'écran, le retour d'information immédiat ou les contraintes de dessin (parallèle, angle droit, etc.). Le système de pointage et de manipulation directe de *SketchPad* préfigure le modèle des écrans tactiles du futur, à l'image de *Surface* de Microsoft. (Image © MIT)

1968

LA SOURIS

Douglas Engelbart,
Stanford Research Institute



Dispositif de pointage informatique toujours le plus répandu actuellement, la souris a été inventée en 1968 par Douglas Engelbart au sein du Stanford Research Institute. Faisant partie du NLS (*On-line system*), un système conçu pour la navigation en ligne sur Arpanet, la souris d'Engelbart est pensée pour évoluer intuitivement dans des interfaces graphiques, riches et complexes, disposant de plusieurs fenêtres telles que nous les connaissons actuellement. Constituée alors d'une boîte en bois équipée d'un bouton de validation ainsi que de deux roues perpendiculaires qui captent sa position en abscisse et en ordonnée, la souris d'Engelbart sera améliorée par Jean-Daniel Nicoud à l'EPFL dès 1979 grâce à l'adjonction d'une boule et de capteurs, une révolution qui mènera à la création de l'actuel leader mondial, Logitech. La souris contribuera au succès du Macintosh dès 1984, et deviendra par la suite le standard de pointage des ordinateurs personnels que l'on connaît. (Image © SRI International, Menlo Park, Calif)

1973 - Xerox PARC lance le projet *Alto* Xerox : premier prototype d'ordinateur avec une interface utilisateur graphique.

1974 - Le TCP/IP (*Transmission control protocol and Internet protocol*) est créé pour uniformiser le réseau. Ce système est toujours utilisé de nos jours.

1975 - Bill Gates et Paul Allen fondent Microsoft afin de commercialiser l'interpréteur du langage informatique : le BASIC.

1975 - Robert E. Kahn développe l'idée d'un réseau d'ordinateurs pouvant communiquer entre eux à grande distance.

1977 - Apple commercialise un des premiers micro-ordinateurs : l'*Apple II*.

1977 - Commodore International commercialise un des premiers micro-ordinateurs : le Commodore PET.

1981 - Xerox commercialise le premier ordinateur à interface graphique piloté par souris : le *Star 8010*.

1980 - Microsoft publie le MS-DOS sur un IBM PC.

1983 - Microsoft publie la première version de Word.

1985 - Microsoft commercialise sa première version de Windows (Windows 1.0).

1986 - Première version commercialisée de X-Window (Unix).

1989 - Début de l'Internet grand public TCP/IP.

1990 - Microsoft lance Windows 3.0 et remporte un succès mondial : dix millions d'exemplaires vendus en 2 ans.

1973

ALTO XEROX

Butler Lampson, Xerox PARC



Alto Xerox, développé au Xerox PARC, un centre de recherche dédié à l'innovation dans l'édition et notamment à l'origine de l'impression laser, l'*Alto Xerox* est le premier ordinateur personnel à interface graphique. S'il se distingue par sa compacité et son disque dur, c'est avant tout son bureau virtuel, parfaite illustration de l'acronyme *wysiwyg* ou *What you see is what you get*, popularisé par Apple, qui révolutionne l'informatique à une époque où les ordinateurs sont conçus pour des utilisateurs experts, fonctionnent avec des cartes perforées et affichent les données dans une police de caractère unique. Donnant enfin à la souris d'Engelbart un environnement de travail à sa mesure, cet ancêtre de windows et de mac os permet d'afficher simultanément plusieurs fenêtres, de les déplacer ou de cliquer à l'aide d'une souris. Capable de faire tourner des applications de dessin et de traitement de texte très puissantes, tout en étant d'une simplicité d'utilisation telle que des enfants pouvaient s'en servir, l'*Alto Xerox* réalise ce que Vannevar Bush avait imaginé en 1945 dans *As we may think*, le tout premier essai d'ergonomie pour l'accès à l'information. De conception trop coûteuse, l'*Alto Xerox* ne sera jamais commercialisé. (Image © PARC, Inc., a Xerox Company)

1983

APPLE LISA

Steve Jobs, Apple Computer, Inc.



L'Apple Lisa est le tout premier ordinateur commercial à posséder une souris et une interface graphique développée par Apple. Le concept du Lisa visait à faciliter l'utilisation de l'outil informatique à tous types d'utilisateurs, qu'ils soient néophytes ou professionnels. L'interface était inspirée du projet *Alto Xerox*. Le succès commercial aurait dû être au rendez-vous mais son coût prohibitif de 9995\$ en fit un échec. Néanmoins cet ordinateur jeta les bases d'un système permettant de comprendre et d'utiliser un ordinateur sans être un professionnel. (Image © Musée de l'informatique, Paris)

1983

LE MOBILE

DynaTAC 3200 de Motorola



Le téléphone portable est la révolution des télécommunications du début des années quatre-vingt-dix. Une des premières interfaces réellement portable permettant d'utiliser la fonctionnalité du téléphone presque partout à n'importe quel moment. L'utilisation du portable, à l'époque, dépendait de l'étendue et la densité du réseau téléphonique. Le téléphone portable, en tant que moyen de télécommunication vocal, a complètement changé les mœurs, transformant complètement notre rapport au temps et à l'espace et dotant chaque utilisateur des facultés d'ubiquité et de téléportation. (Image © .CORP)

1990 - Photoshop 1.0 voit le jour sur Macintosh, développé par John et Thomas Knoll.

1993 - Windows est utilisé par plus de 25 millions de personnes à travers le monde.

1993 - Lancement de Mosaic, premier navigateur Web.

1993 - Honda développe Asimo, robot humanoïde avec des bras et des jambes articulés. Il se déplace en reconnaissant l'environnement dans lequel il évolue.

1994 - Netscape commercialise à grande échelle son navigateur Web. Il dominera le marché jusqu'à la fin des années quatre-vingt-dix.

1997 - Le super-ordinateur Deep Blue bat le champion du monde d'échecs Gary Kasparov (2 victoires, 3 nuls et 1 défaite).

1998 - Kevin Warwick se fait poser une puce RFID dans le bras.

1995 - Lancement de Windows 95. Quatre jours après sa sortie, un million de copies sont déjà vendues.

1998 - Lancement de l'Imac, ordinateur dessiné par le designer Jonathan Ive.

1999 - Lancement du robot de compagnie Aibo par la firme Sony.

2001 - Fondation de Wikipedia.

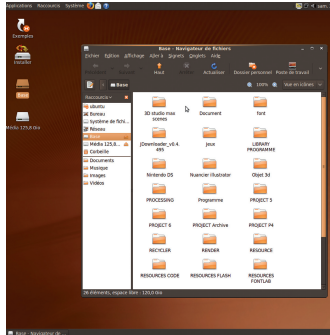
2001 - Processing est un langage de programmation et un environnement de développement basé sur le langage Java. Créé par Benjamin Fry et Casey Reas, il se positionne comme le prolongement du projet Design by number de John Maeda. Cette environnement rend accessible la programmation aux artistes et aux graphistes.

2005 -Création du robot quadrupède militaire Big Dog.

1991

LINUX

Linus Torvalds



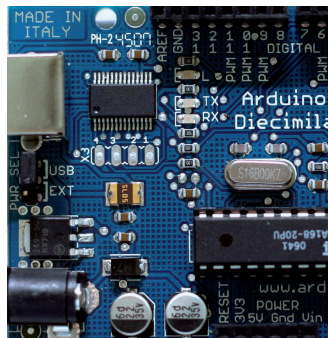
Parmi les systèmes d'exploitations, on peut en relever un qui aboutit à une interactivité plus importante. Linux est, en effet, un système dit ouvert, qui induit que tout les utilisateurs peuvent contribuer à son évolution. Ce système met en avant une modification permanente, non seulement de l'interface, mais aussi de la programmation même du système d'exploitation, ce qui permet d'ajuster très finement les besoins de chacun vis-à-vis des diverses utilisations informatiques. Le problème de ce genre de système très évolué est qu'il gagne rapidement en complexité, il faut donc un certain niveau de connaissance pour l'exploiter pleinement. Voici quelques exemples de distributions Linux: Debian, Mandrake, Red Hat, Ubuntu, etc.

(Image : capture d'écran Ubuntu)

2005

ARDUINO

Massimo Banzi, David Cuatrecasas, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti.



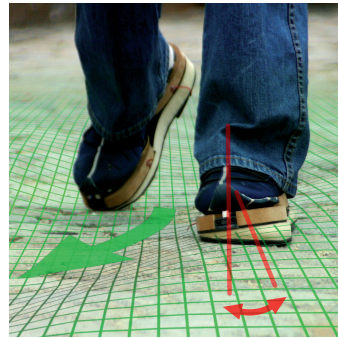
Arduino est une carte électronique, sous licence copyleft, permettant aux artistes et graphistes de construire des objets interactifs indépendants ou connectés à un ordinateur. Arduino simplifie l'accès à des composants électroniques (accéléromètre, webcam, bouton, capteur infrarouge ou de température, moteur pas à pas, etc.), permettant de les contrôler et de les coupler en dispositifs d'interfaces complexes, dites de haut niveau. Arduino possède l'avantage de fournir une plateforme didactique simple et économique.

(image copyleft Wikipédia)

2006

CABBOOTS

Martin Frey



CabBoots est un système de navigation pédestre haptique. Alors que le modèle actuel de développement des technologies de navigation se fonde sur des interfaces de sorties visuelles ou sonores, Martin Frey a imaginé d'intégrer l'interface sous le pied. Le piéton ainsi équipé trouve son chemin par un processus d'accompagnement semi-conscientisé. Si l'on s'éloigne de l'itinéraire de navigation, la semelle se penche par l'entremise d'éléments électromécaniques, restituant l'impression produite en s'écartant du sillon d'un chemin de campagne. Cette simulation topographique virtuelle laisse l'utilisateur libre de vaquer aux tâches normalement assumées par un piéton maîtrisant son environnement.

(Image © Martin Frey)

2007 - Microsoft présente son projet d'ordinateur tactile *Surface*.

2007 - Intel lance la première génération de puce à 45 nm.

2007 - Wacom lance la première génération de tablette graphique directement couplée avec un écran.

2007 - Lancement du iPhone, le smartphone tactile d'Apple.

2008 - Lancement du prototype d'XO, le PC portable dessiné par Yves Behar dans le cadre du projet *One Laptop per Child*.

2008 - Lancement de la première génération de *NetBook*, ordinateur de taille très réduite facilement transportable. Son prix est généralement inférieur à quatre cents euros.

2008 - *Gordon*, conçu par la Reading University, devient le premier robot piloté par des neurones de rat.

2008 - Adobe sort la version 1.0 de Flash, logiciel phare des designers d'interfaces Web.

2008 - Mise en service à Los Alamos de *Roadrunner*, le premier super-ordinateur dépassant les 1 pétaFLOPS (soit 10¹⁵ opérations flottantes par seconde).

2009 - Sortie officielle de Windows 7, système d'exploitation de la société Microsoft.

2009 - Sortie officielle de Mac OS X *Snow Leopard*, système d'exploitation de la société Apple.

2009 - Sortie officielle de Ubuntu *Jaunty Jackalope*, système d'exploitation libre développé par la communauté Ubuntu et sponsorisé par Canonical.

2008 SIXTH SENSE

Pranav Mistry, MIT Media Lab



Sixth Sense est une interface de réalité augmentée portable. Elle propose une solution simple et originale au problème de la multiplication des supports. Constituée d'un collier comportant une caméra, un projecteur, un miroir et un téléphone interconnectés, *Sixth Sense* permet d'afficher dans notre environnement toutes sortes d'informations, mais aussi de manipuler directement ces informations par gestes, à l'aide de bagues de couleurs placées sur les phalanges terminales du pouce et de l'index. *Sixth Sense* est, très littéralement, une extension de notre champ de perception et d'interaction. Elle permet d'imaginer toutes sortes de nouveaux scénarios de consommation, mais aussi de socialisation, inspirés des réseaux sociaux en ligne. (Image © Pranav Mistry, Fluid Interfaces Group, MIT Media Lab)

2008 CARNIVOROUS DOMESTIC ENTERTAINMENT ROBOTS

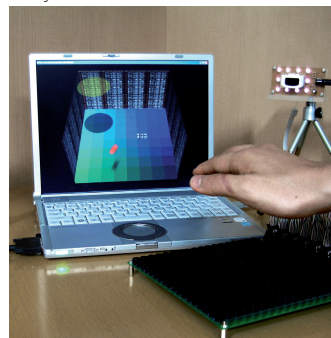
James Auger, Jimmy Loizeau, Département design d'interaction du Royal College



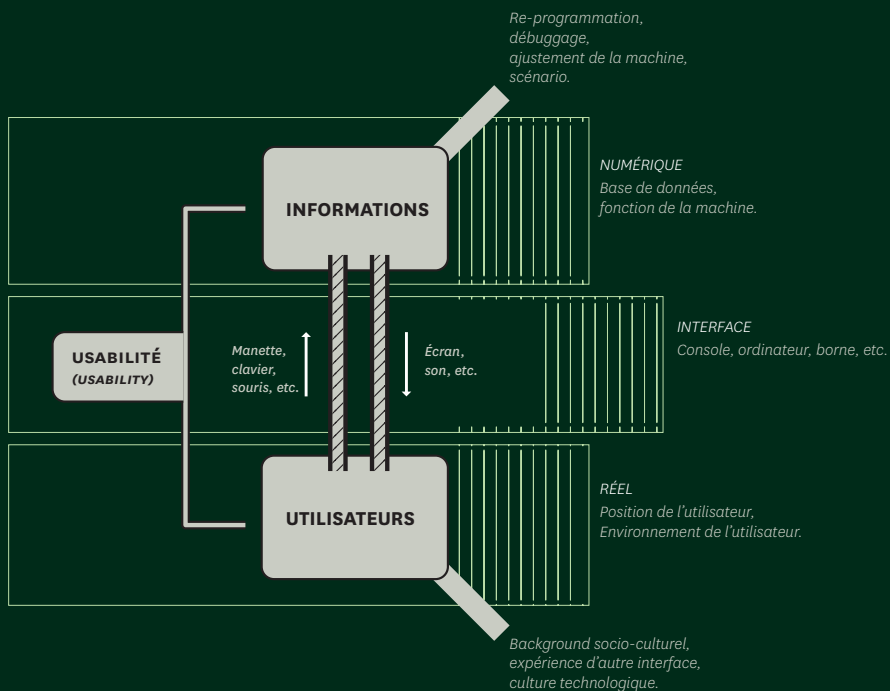
«Le design sert-il seulement à rendre les choses jolies?», demandait à son auditoire le designer James Auger à la dernière conférence Lift. Auger mène, avec Jimmy Loizeau, une recherche en design, explorant notamment la capacité des objets à interagir avec leur environnement pour recueillir l'énergie nécessaire à leur fonctionnement. Ces systèmes cybernétiques mimant les organismes vivants, traquent la biomasse disponible dans nos foyers, en particulier les animaux nuisibles, pour éclairer ou alimenter l'électro-ménager. Le robot ci-dessus coopère avec une araignée pour récupérer les mouches piégées dans sa toile et convertir en électricité l'énergie libérée pendant leur décomposition. (Image © Jimmy Loizeau & James Auger)

2009 TOUCHABLE HOLOGRAPHY

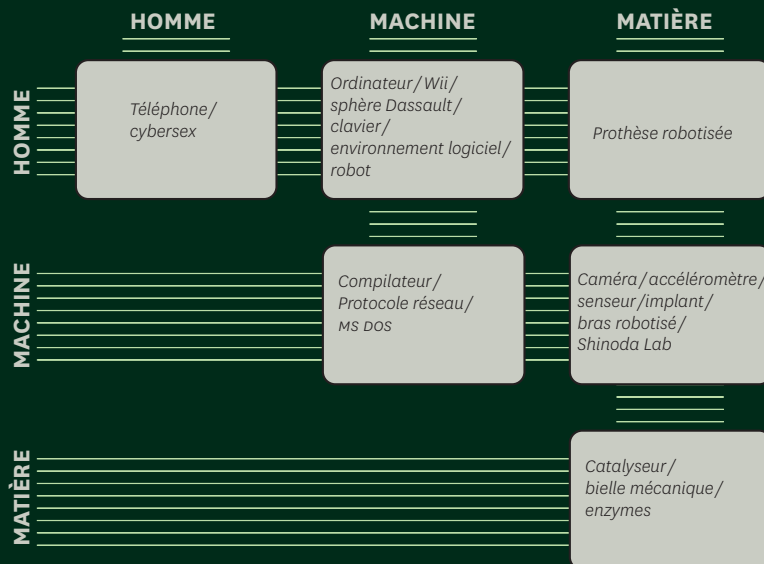
Takayi Iwamoto, Shinoda Lab



Les interfaces de restitution 3D sont souvent encombrantes et centrées sur le sujet immergé. La voie ouverte par le Shinoda Lab de l'université de Tokyo autorise à faire participer simultanément plusieurs sujets à la simulation, sans porter de matériel lourd. Elle permet de ressentir, au toucher, un objet virtuel, seulement visible sur l'écran. Fonctionnant sur un principe d'interférences d'ondes de hautes fréquences, qui génèrent un champ de forces localisé, ce système permet un *feed-back* tactile d'une résolution d'un 1cm. Une interface qui tend à rendre caduque l'opposition entre matériel et virtuel. Il s'agit pour l'instant d'un périphérique de sortie, mais il n'est pas interdit d'imaginer, à partir de cette technologie, toutes sortes de fonctions, pour l'instant réservées aux objets physiques. (Image © Koichi Shinoda, Department of Information Physics and Computing, The University of Tokyo)

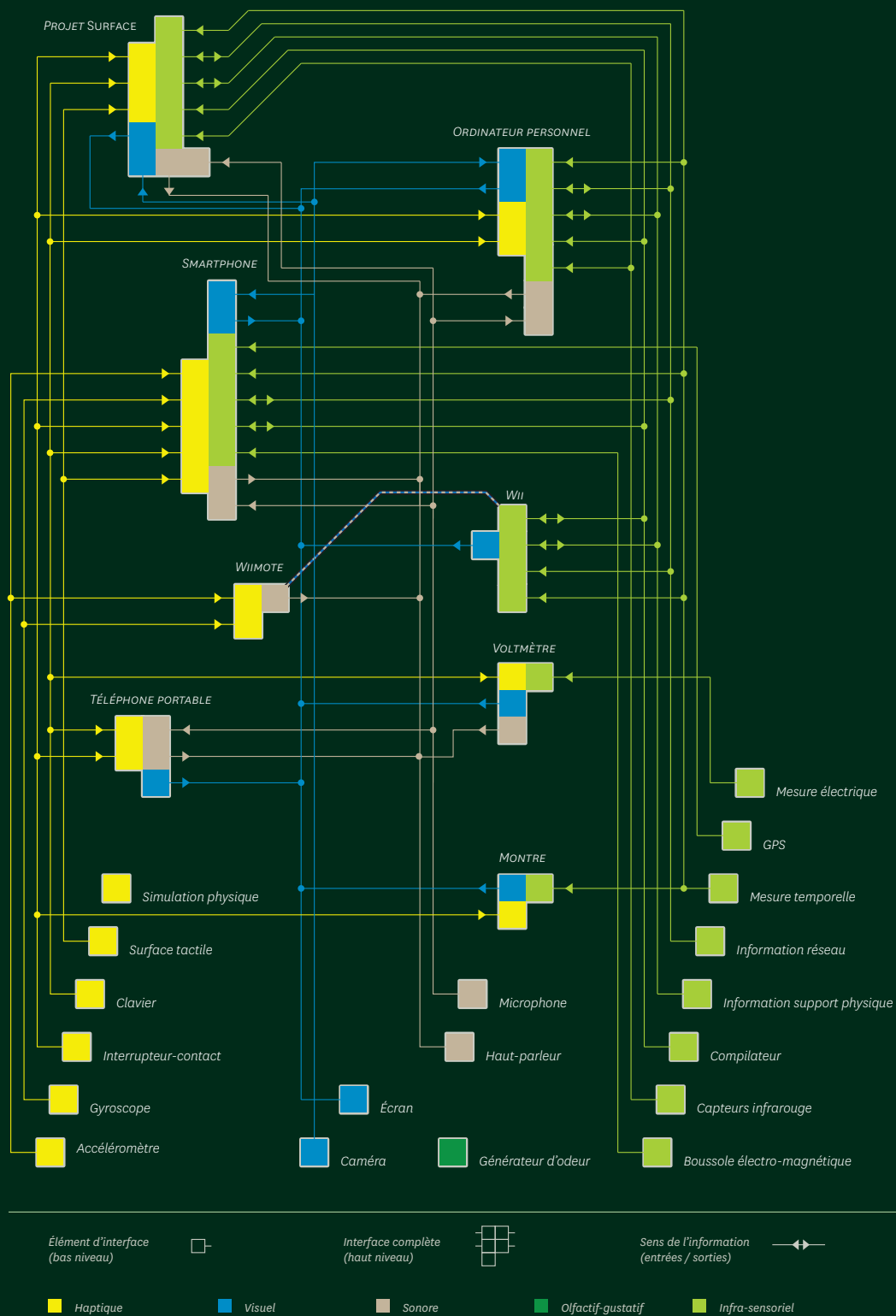


ORGANISATION D'UNE INTERFACE HOMME/MACHINE



CLASSIFICATION D'INTERFACES NUMÉRIQUES PAR ASSOCIATION D'ENTITÉS

SCHÉMA D'ORGANISATION DE PROCESSUS HOMME/MACHINE



CLASSIFICATION PAR NIVEAU D'INTÉGRATION D'INTERFACES

ENTRETIEN AVEC JEAN-LOUIS FRÉCHIN

.CORP (INDIVIDU 2) & STUDIO LO

1. *User experience* est l'ensemble des événements et des sensations que sera amené à vivre l'utilisateur face à une interface.

2. Linux, ou GNU/Linux, est un système d'exploitation *open source* basé sur le noyau UNIX. Cf. « Petite chronologie des interfaces » p114.

C. SL. Pouvez-vous vous présenter brièvement et nous expliquer en quelques mots quel a été votre parcours.

JLF. J'ai une formation d'architecte. Après mon DPLG, j'ai décidé de faire du design, parce que je trouvais qu'il manquait de machines, d'économie, de réel dans les études d'architecture. J'ai donc intégré l'ENSCI. Après mon diplôme, j'ai rejoint un institut de recherche en informatique au Portugal. Il y faisait beau, il y avait des plages, mais surtout il y avait des gens qui avaient été formés aux États-Unis : au Massachusetts Institute of Technology, au Carnegie Mellon University. Ces gens-là avaient conscience du potentiel du design dans les nouvelles technologies, notamment celles où on ne voit rien.

En rentrant à Paris, j'ai créé une agence de design global avec Stéphane Bureaux, mon associé pendant 5 ans. Nous avons été pionniers dans l'utilisation des outils numériques, sur de tout petits systèmes 3D, des projets d'architecture commerciale, de produit, de mobilier, de graphisme : un rêve de design global. Puis, un jour, j'ai rencontré un éditeur de CD-ROM éducatifs, Montparnasse Multimédia. Embauché comme designer (à l'époque, on ne savait pas comment appeler ce boulot : designer d'interface ? designer d'interaction ? designer d'interactivité ?), j'y ai réalisé une quinzaine de CD-ROM avec des interfaces non standards riches, complexes et complètes. On a créé les premiers produits scientifiques avec des simulateurs de phénomènes complexes, une sorte de Cité des Sciences virtuelle. Les CD-ROM du Louvre, les Petits Débrouillards se sont bien vendus et ont été reconnus dans le monde entier. C'était vraiment une époque géniale, l'invention du CD-ROM culturel à la française.

Au bout de 7 ans, j'ai fini *publishing manager*. C'était très intéressant et valorisant pour un créatif « analphabète » de se retrouver avec des responsabilités, mais je voulais faire mes propres projets, essayer de marier l'interaction et les objets, travailler directement avec des entreprises pour expliquer ce qu'elles sont. Le design stratégique ne fait pas très « école d'arts », mais l'entreprise m'intéressait. Pour moi, c'est une matière que l'on peut traiter en designer, d'une manière très différente d'un consultant standard. Alors je suis parti et j'ai créé NoDesign.

Qu'est ce qu'un designer d'interface ou d'interface objet dans votre cas ?

Je ne parle pas de design d'interface, ni de design d'objet, je parle de design numérique. C'est un nom que l'on a créé parce que design d'interaction, comme l'a proposé Bill Moggridge, ce n'est qu'une partie du problème. En tant que français, avec une approche assez universaliste, assez humaniste et qui veut comprendre l'interaction, devenir un expert tactique de l'interaction ne m'intéresse pas. Le design numérique propose un repas complet. Interagir pour quoi faire ? Qu'est ce qu'on voit et dans quel contexte ? C'est ce que j'appelle la situation. Comment va-t-on créer des relations, c'est-à-dire des interactions, qu'est ce que l'on va voir ou ne pas voir ? Ça ce sont les représentations.

Les anglo-saxons ont tendance à transformer le design en fonction tactique, avec une division du travail très poussée. On a le spécialiste du design d'interaction, le spécialiste du design d'interface visible, le spécialiste du design d'interface non visible, le spécialiste de la *user experience*¹, etc. Avec toutes ces expertises, on se retrouve rapidement à 20 pour s'occuper d'un écran de téléphone qui fait seulement 320 pixels.

Je préfère avoir une approche globale et mériter les produits traditionnels et les produits numériques. Aujourd'hui tout est numérique, d'une manière ou d'une autre : Vélib est numérique, les tickets d'avion sont numériques, souvent on n'a même plus de ticket, c'est une chaîne complète de services numériques. On est aussi en train de réinventer la chaîne conception/fabrication/distribution dans l'industrie. Le monde change, le numérique est partout. Dans une « société que l'on pourrait appeler une société des échanges et de la transformation », la notion de valeur évolue de la propriété vers le service. La possession est moins importante, et ses usages sont parfois limités dans le temps. L'exemple, c'est Linux², son accès est gratuit mais son coût dépend de ce qu'on veut lui faire faire. C'est comme si dans une voiture le moteur était gratuit et qu'on ne vous faisait payer que le voyage. Cela change beaucoup de choses d'un point de vue économique. Cette société des échanges est un bouleversement aussi important que le communisme ou le marxisme, qui remet en cause beaucoup de choses : une nouvelle chaîne de valeurs, une nouvelle chaîne de droits d'auteurs, une nouvelle notion du bien commun, etc. Dans tout cela, le design numérique est cette approche, généraliste, de tout ce que l'on peut faire dans les possibles du numérique.

3. Le multi-touch est une technologie d'écran, popularisée par l'iPhone, permettant de gérer simultanément plusieurs pressions tactiles.

4. Widget est une contraction de Window et gadget, un widget est un petit outil qui permet d'obtenir des informations (météo, actualité, dictionnaire, carte routière).

Mais il y a bien un moment où il y a un objet qui émerge d'un projet ?

Ah, je vois bien le designer qui a du mal à renoncer à son bout de plastoc ! C'est vrai que certains pensent qu'on est dans une société de l'immatériel, que tout est virtuel, mais pour faire tout ça, il faut beaucoup de matériel, il faut des ordinateurs, des réseaux, des trottoirs, des câbles dans les trottoirs, etc. donc les notions d'immatériel et de virtuel sont toutes relatives. Et les objets sont finalement les plus formidables interfaces que l'on puisse imaginer, parce qu'ils ont une unicité et une existence temporelle, alors que l'interface de votre ordinateur est une chaîne d'événements, très dure à saisir. Le grand gourou de l'interaction de Microsoft, Bill Buxton, a reconnu à la dernière conférence HCI-09 (*Human computer interaction* 2009) qu'ils avaient « sous-estimé l'importance du style ». Ces gens-là ne se sont jamais posé la question d'une relation forme/fonction/usage, ce qui, pour des designers, est le fondement de leur pratique depuis le Bauhaus. Cette unité entre, ce que je vois, ce que ça fait et comment je l'utilise n'a jamais été documentée, inventoriée. Pourtant, il n'y a pas de meilleure représentation d'un objet. L'exemple le plus connu, c'est le Nokia 5100, un petit téléphone-bijou rouge, qui était une représentation des interactions possibles avec le téléphone. On ne voyait rien, juste un joli design traditionnel qui enveloppait un écran microscopique et une interface gérée avec le pouce. Si Nokia est devenu par la suite le leader de la téléphonie mobile, c'est parce qu'il a été le premier à faire du design de très haut niveau en termes d'expérience et en termes de design traditionnel tout simplement, pendant que nos ingénieurs français de Sagem ou Alcatel faisaient des téléphones quasi militaires sous les ordres d'un marketing extinctif.

J'ai créé l'Atelier Design Numérique de l'ENSCI, justement parce que je me suis rendu compte que les représentations jouaient un rôle très important dans ces dispositifs numériques compliqués. En fait, ce que l'on voit est presque aussi important que ce que ça fait. Toutes les boîtes qui ont compris ça et qui l'ont mis à l'équilibre – Nokia, Apple – sont aujourd'hui en première ligne. Dans un pays intelligent comme le nôtre, avec une culture de l'abstraction, l'utilitaire, le rationnel et l'intelligence s'opposent toujours au beau et aux sens, c'est pour ça que le design a eu du mal à s'imposer chez nous. Marius Vachon, le fondateur du Musée d'art et d'industrie de Saint-Étienne, disait : « Il faudrait que les artisans deviennent artistes et que les artistes deviennent commerçants pour que l'on vende du beau ». C'était en 1875 et déjà il pointait le retard français sur les allemands et les anglo-saxons. Avec des machines comme le iPhone, on change de paradigme : l'objet est l'interface et le design de l'objet c'est l'écran. On passe d'une culture du pouce à une culture des yeux. Les gens de la téléphonie mobile, les ingénieurs, les gens de Samsung, Alcatel, etc. ne savent pas être ambitieux en termes de soft. Ils se disent toujours : « C'est facile de changer, on peut tout faire à la dernière minute ». Contrairement à Apple, ils ne se posent pas les questions comme on se les pose avec des objets physiques qu'on est obligé de terminer, parce qu'il faut injecter, fraiser, etc.

C'est plutôt l'interface qui donne l'objet, c'est quelque chose de nouveau...

Oui et c'est une vraie révolution. Les gens du soft prennent le pouvoir sur les gens du design traditionnel.

Mais quand vous dites « voir », ce n'est pas voir un bel objet, le iPhone c'est aussi quelque chose que l'on touche...

Non, voir, c'est regarder, regarder l'interface. L'image a un rôle extrêmement important. Toucher, ce n'est pas la question, l'iPhone aurait été de cette qualité-là avec un clavier et un joystick.

Donc c'est vraiment une question d'aspect, de cosmétique de l'objet...

Non, ce n'est pas une question de cosmétique, c'est une question d'esthétique au sens le plus philosophique du terme. C'est vraiment le côté puissant de la représentation que l'on propose qui fait la différence. Bien sûr le toucher du iPhone est magique, ce n'est pas neutre.

Le multi-touch³, les widgets⁴, c'est tout ça qui fait le iPhone, non ?

Non, le multi-touch, ça existe dans des laboratoires depuis 20 ans. Le vrai génie de l'iPhone, c'est que c'est un objet « conséquence de services » qui permet, à vous, à moi, de casser le monopole de services des grands opérateurs de téléphonie mobile. Apple a toute latitude parce qu'il a fait un produit séduisant, avec des visuels de qualité offrant des services désirables. Toutefois, il n'y a rien de nouveau dans tout cela, c'est juste une transformation géniale, c'est Mac os 10 dans un téléphone.

Cette machine a permis de rentrer un coin dans le monopole des opérateurs téléphoniques. Pour vous, designer, cela vous paraît intéressant mais pas du tout innovant quand je vous dis « on voit ». Pourtant, c'est révolutionnaire : avant l'iPhone, on ne savait pas ce que c'était la puissance des images dans la téléphonie mobile. Les Nokia, Samsung, LG, HTC, Sagem, Alcatel n'ont aucune culture esthétique, artistique, ils sont dans le pur utilitaire. Apple a amené d'autres dimensions : symbolique, esthétique. Contempler un tableau, une image, c'est un langage d'une puissance infinie. Ce sont ces choses-là qui sont importantes, ce n'est pas justes les gadgets, les clics, les double-clics, le multi-touch. La technologie n'a jamais fait des usages, elle doit être dominée par une idée : il faut aller vers le meilleur pour vraiment servir les gens, sans sociologue, sans marketing, sans ergonome, sans spécialiste de la user expérience, c'est-à-dire avec un entrepreneur et des designers. Je dis cela par provocation car j'ai le sentiment que nous coulons et que nous perdons du temps... Prenons des risques, qu'avons-nous à perdre ? Nous restons qui nous sommes, n'ayons pas peur.

5. Linden Lab est l'éditeur du monde persistant *Second Life*.

6. *Fabless* désigne les entreprises de biens et services assurant une activité de conception mais ne disposant pas d'unité de production.

7. L'Internet des objets est un néologisme qui se rapporte à l'extension d'Internet dans le monde matériel au travers d'objets informatiques.

8. Titre d'un ouvrage collectif sur l'invisibilité croissante des interfaces. *Disappearing computer*, Achilles Kameas, Irene Mavrommati, Norbert Streitz, Berlin, Springer-verlag, 2007.

9. Twitter est une plate-forme Internet de micro-blogging et réseaux sociaux. Il permet de suivre des flux de messages courts de 140 caractères maximum. La popularité de Twitter tient au fait que l'on peut actualiser et aussi être averti de messages reçus via SMS.

10. *Hacker* est à l'origine un mot anglais signifiant bricoleur, il désigne le possesseur d'une connaissance technique suffisante pour détourner ou modifier un objet ou un mécanisme afin de lui faire faire autre chose que ce qui était initialement prévu.

C'est comme cela que fonctionne Apple, avec une faible division du travail ?

Il y a un groupe de designers, il y a le *software group* et un patron qui sait ce qu'il veut. Il n'y a pas cinquantes avis de spécialistes qui s'annulent les uns les autres. Il faut voir que dans le monde qui est le mien, les designers sont en minorité. Ceux qui font la loi, ce sont les ergonomes, des gens qui revendiquent un savoir scientifique de ce qu'est un usage, de ce qui est bon ou pas bon. Est-ce qu'un designer qui dessine une chaise consulte encore des ergonomes ? Dans le design traditionnel, il y a longtemps qu'il n'y a plus d'ergonome. L'ergonomie du travail, qui a existé dans les années soixante, a disparu et les ergonomes se sont tous recyclés dans le monde du design numérique. C'est une espèce de chappe de plomb utilitaire, rationnel et de spécification. Toutes les grandes entreprises *high tech* françaises de produits grand public ont ainsi disparu... Le combat que l'on mène ici à NoDesign, à l'Encsi et dans les endroits où l'on intervient, c'est de dire : « Ce n'est pas parce que c'est nouveau que toute la culture passée est à oublier ». Au contraire, la bonne intelligence, c'est de marier ces cultures pour créer du nouveau. Pourquoi *Second Life* ne marche plus ? Et bien, allez expliquer à Linden Lab⁵ que, plus la peinture a cherché la représentation fidèle, plus cela a été un échec, et que ce qui a explosé ce schéma-là au 19^e siècle, ce sont les impressionnistes. Pourquoi cela serait différent dans les systèmes de représentation synthétiques et dans les mondes virtuels ? L'impressionnisme, Renoir, Monet, Manet, c'est plus puissant que Gerricault et tous ces pompiers que l'on voit au musée d'Orsay. C'est vrai que, dans la téléphonie et le monde des nouvelles technologies, cette vision large n'est pas courante. C'est quand même un monde d'ingénieurs, de marketing, et là c'est plus compliqué parce que je pense qu'une grande partie de ces problèmes vient de cette course systématique au renouveau artificiel basé sur des fondations qui ne relèvent pas du sens mais d'une recherche de performance à court terme. Il faut donc créer des harmonies nouvelles et peu de marques y parviennent, en tout cas dans les produits *high tech*.

Vous parlez justement d'objets et d'interfaces, comment la rencontre se fait-elle selon vous, comment peut-elle être intelligente et comment marche t-elle ?

Elle suit un chemin compliquée. L'objet manufacturé, produit, vendu, est mort depuis que l'on a quitté les espaces de fabrication et de maîtrise de la fabrication pour dire que ça ne vaut plus rien. On fait cela en Chine et nos entreprises sont innovantes parce qu'elles sont *Fabless*⁶. On a scindé la branche sur laquelle on était assis. Aujourd'hui, il y a des systèmes basés sur des usages et des services qui recréent une nouvelle nécessité de l'objet. Ces objets ne sont plus conséquences de domination énergétique, comme le moulin-à-café électrique et tous les objets de la grande période de la consommation de masse. Depuis les années soixante-dix, depuis la séparation de l'or et du dollar, on est entré dans une société des échanges et de l'immatériel. Avec la fin de la convertibilité de l'argent en l'or, l'argent n'est plus l'argent, sa valeur est fixée par convention et plus par connexion à des métaux, au matériel. C'est très symbolique, et pour les objets c'est pareil. Aujourd'hui, les nouveaux objets, que j'appelle néo-objets, sont conséquences non pas d'une énergie, d'une valeur, mais de services. Le modèle présidant à l'existence des objets pendant tout le 20^e siècle, celui de l'industrialisation en série (d'objets), se trouve progressivement remplacé par un nouveau modèle d'objets rendus nécessaires par les besoins de l'époque, les envies de l'époque et les nouvelles technologies. C'est une réinvention des objets qui sont maintenant connectés entre eux. C'est pour cela que l'on parle d'Internet des objets⁷, d'*ambient intelligence* (Cf. note 4, p107) ou de *disappearing computer*⁸. Les ordinateurs sont partout, les objets eux-mêmes deviennent des ordinateurs. Tout cela ne forme pas un univers virtuel que l'on regarde à travers un écran, on est dans une machine géante dans laquelle tous les objets se parlent entre eux et peuvent nous aider à mieux vivre, à partager des informations sur notre environnement. On peut imaginer de nouvelles façons de vivre avec le numérique : des objets que l'on porte sur soi, des objets dans la maison et des objets sur le territoire, c'est vraiment multi-échelle. Nous avons besoin de ces objets pour créer des symboles, parce que, pour faire société, pour vivre ensemble, pour comprendre les mêmes choses, on a besoin de symboles. Et l'objet, c'est un symbole, c'est la plus extraordinaire des interfaces.

Comment distinguer l'objet et l'interface ?

On assiste à l'éclosion d'un nouveau monde industriel, dans lequel les objets sont conséquences de services et d'usages. Dans la société de l'information qui est la nôtre, ces objets doivent être dessinés et inventés. Par exemple, ils peuvent être enfouis dans un téléphone portable comme l'iPhone, mais aussi être embarqués dans un miroir, comme celui sur lequel nous travaillons, connecté à Twitter⁹. Ce miroir reflète notre vie numérique, il est le miroir des gens que nous aimons et qui sont connectés sur le Web, mais il est aussi notre reflet, tel Narcisse, qui nous rappelle que les autres sont là. C'est un peu l'archétype de ce que font les hommes depuis toujours.

11. Le *low tech* est l'action d'utiliser une vieille technologie en général pour ses caractéristiques simples, économiques et populaires. Mais aussi dans l'optique de recycler de vieilles machines. *Low tech* s'oppose à *high tech*.

12. Le Web 2.0 est une expression pour qualifier le renouveau de l'Internet avec les caractéristiques suivantes : la participation collaborative des internautes, la naissance des réseaux sociaux, les applications Internet riches (notamment grâce à l'évolution de la technologie flash, Shockwave, PHP)

13. Blogspot est une plate-forme blog appartenant à Google Inc. Ce service est le moyen le plus simple pour publier du contenu sur Internet. De plus, il est complètement gratuit et sans publicité.

14. WordPress est un moteur de gestion de contenu sur le Web, il nécessite un hébergement adapté pour fonctionner. Contrairement à un Blogspot, WordPress est complexe à installer.

15. PHP est un langage de programmation pour Internet utilisé principalement pour générer des systèmes dynamiques permettant de modifier des bases de données.

16. contraction de *Fabrication Laboratory*, un *Fab Lab* est un atelier équipé de machine-outils assistées par ordinateur (Cf. « Design en auto-production » pp 92-103).

Quels problèmes rencontre-t-on dans la conception d'interface ? En quoi diffèrent-elles de celles du design d'objet ?

Vous parlez beaucoup d'objet... Le français est une langue merveilleuse, l'objet, en français, c'est autant ce que l'on peut toucher que ce qui se présente à l'esprit. Le mot *objet* est intéressant dans ces deux sens. L'interface est l'objet des designers du 21^e siècle. Elle peut aussi devenir un objet (matériel), mais cette chose n'est que la conséquence d'un processus, elle n'est pas le plus important. Le plus important c'est l'intention qui lui a donné naissance.

L'innovation dans les interfaces dépend-elle uniquement de l'avancée des technologies ou peut-elle être « rétro-technologique » ?

La technologie est toujours présente avant son utilisation. Après, au sein du large panel technologique, certaines sont plus intéressantes que d'autres, notamment celles qui sont ou deviennent partageables, donc accessibles à tous, ce qui n'est pas le cas des technologies issues des laboratoires ou d'entreprises classiques.

Les technologies auxquelles on peut « tordre le cou », que l'on peut utiliser en tant que designers et créateurs, sont importantes, car les objets sont humains quand on peut les modifier, les transformer, les *hacker*¹⁰.

Les technologies sont donc une condition préalable. À notre niveau, on ne peut utiliser que des technologies qui sont accessibles pour proposer et promouvoir de nouveaux usages. Chez NoDesign, nous ne faisons pas de différence entre *high tech* et *low tech*¹¹, ce qui nous intéresse, c'est l'harmonie. Parfois, nous faisons de la déconstruction, c'est-à-dire que nous enlevons tout ce qui est inutile, et parfois, nous en rajoutons, mais ce qui est important, c'est de choisir la bonne technologie, pour la bonne utilisation. Connecter un miroir à Twitter via *Processing* (Cf. note : 2001 - *Processing*, p 114.), c'est assez *low tech* finalement.

Ne va-t-on pas, avec la complexification des interfaces et des objets, vers une logique de la boîte noire, susceptible de dérives ?

Certainement, la logique de la boîte noire n'est pas une bonne chose, il faut toujours donner les clés, casser le sceau de garantie. Sur la stratégie de la boîte noire, on a un exemple merveilleux : la semaine dernière, General Motors, la première capitalisation américaine, a fait faillite. L'automobile était un objet que les gens aimaient, parce que, sous le capot d'une *2 chevaux*, ou d'une *DS*, on voyait les systèmes mécaniques, on comprenait comment ça marchait. Aujourd'hui, on ne comprend plus rien à ce qui se passe : sous les capots, tout est sécurisé, enfermé. Vous n'êtes bon qu'à une chose quand ça tombe en panne, c'est payer.

Finalement, on nous a exclu de l'objet, en le verrouillant.

On ne peut plus le modifier, le *hacker*. La boîte noire, c'est exclure les gens. Et tous les gens qui promeuvent ces systèmes de boîte noire vont mourir et c'est l'histoire qui nous le montre. Si vous excluez les gens des dispositifs qu'ils achètent, vous les condamnez à n'utiliser que les usages définis par vous-même et ils ne peuvent plus s'approprier l'objet. Si, par contre vous créez des objets avec des fonctions fabricatrices, comme le relève Gilbert Simondon, vous obtenez des systèmes plus intéressants, plus adaptés aux diverses utilisations possibles. Apple est un exemple d'intelligence, il n'est ni fermé, ni ouvert, il navigue entre ces deux mondes.

Cela dit, tout le monde ne pouvant pas devenir *hacker*, il faut moduler les objets suivant les compétences de l'utilisateur pour en faire des objets cognitifs, qui expriment ce qu'ils sont, et considérer l'objet technique comme un instrument et non comme une chose dont les usages sont pensés par les gens du marketing. Parce que, pour eux, on n'est bon qu'à payer, à consommer. Il est évident que cela ne doit pas se passer comme cela, et d'ailleurs tout les grands corps industriels fermés, avec leur stratégie de sécurisation, sont en train de se remettre en question.

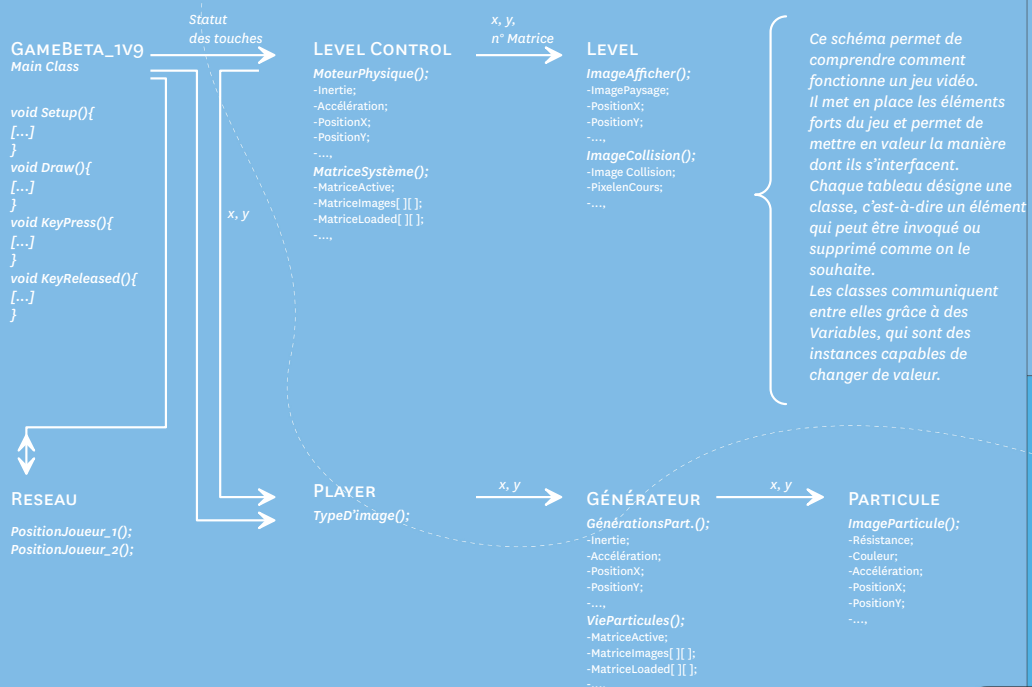
Il faut inventer la notion d'objet « post-produit » qui coïncident avec la complexité de la vie, car, après tout, il y a des gens qui s'intéressent aux choses et d'autres pas. À l'époque, quand les *2 chevaux* ne démarraient pas, il y a ceux qui démontaient les bougies, les soufflaient, les brossaient et les autres qui appelaient le garagiste. Pareil pour les blogs du Web 2.0¹² : quand on veut faire un blog, il y a des niveaux de complexité, depuis le Blogspot¹³ clé en main, en passant par le WordPress¹⁴ à installer sur son hébergement en PHP¹⁵, jusqu'au codage complet du site. Les designers doivent prendre en compte cette dimension.

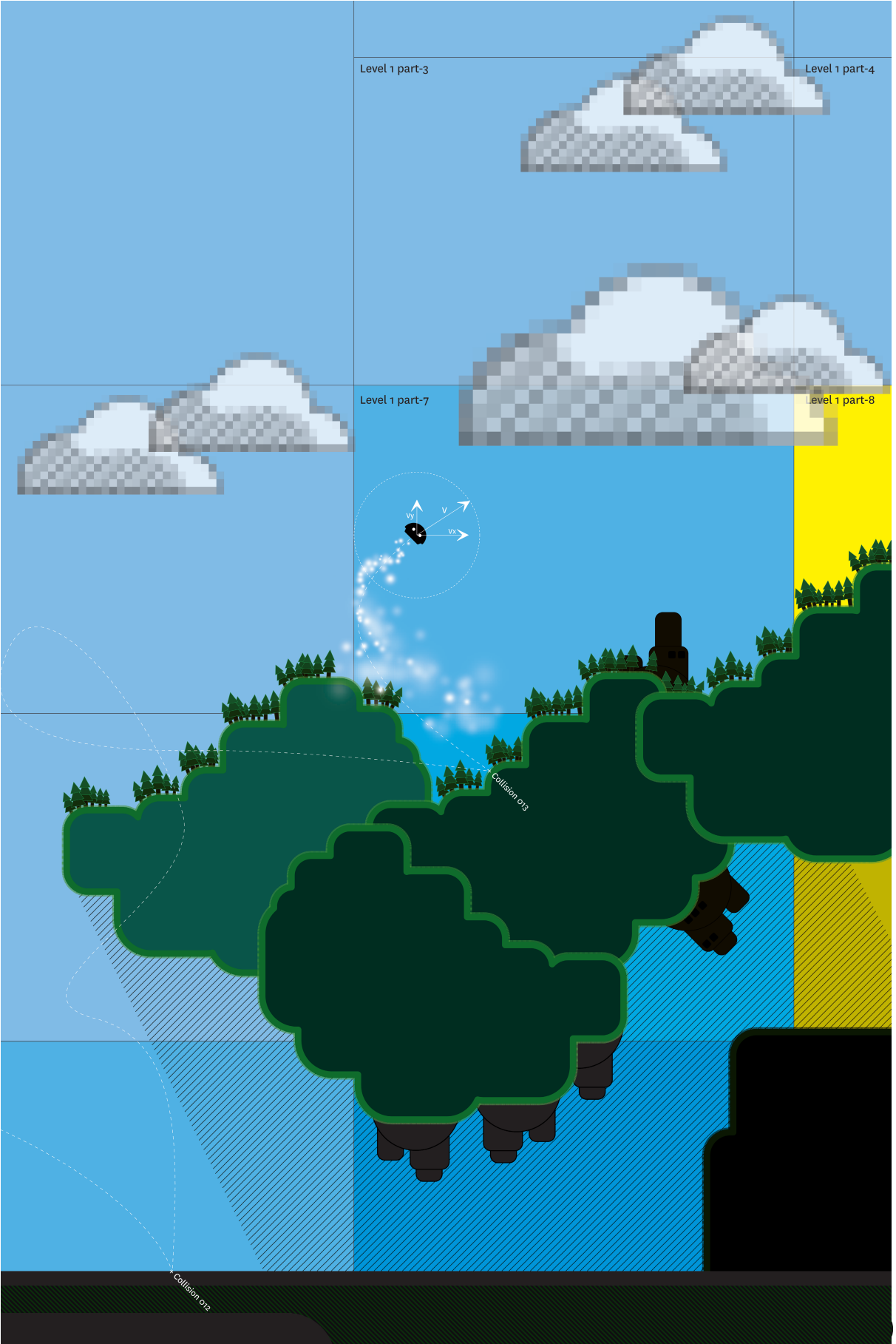
Créer des objets ouverts, *hackables*, modifiables, beaucoup pensent que c'est une sorte de renoncement au design, mais en fait ce serait plutôt l'émergence d'un méta-design. L'industrie n'est pas finie, elle prend une nouvelle forme, elle se rapproche de l'artisanat, mais avec des moyens technologiques énormes. Parmi ceux-ci, le vaste réseau du Web permet de créer et de fabriquer des objets uniques. Dans ce cas précis, nous ne sommes plus dans un système concepteur avec bureaux d'études, ouvriers, administration, etc. Nous sommes dans un système de production optimisé appelé : *Fab lab*¹⁶. Le *Fab Lab* permet de raccourcir les processus de conception/fabrication/distribution et de maîtriser toute la chaîne de production. Cela revient à produire de manière industrielle de l'unique et permet, de plus, par la technologie numérique, de produire la pièce en n'importe quel point de l'espace. Avant, on avait des objets qui produisaient des services, maintenant, on a des services qui produisent des objets. Le paradigme s'est inversé.

Hide & seek gameplay for two players

Damien Baïs & .CORP

Hide & Seek est un jeu vidéo basé sur le principe du cache-cache, présenté sous forme d'installation mimant de anciennes bornes d'arcades. Les joueurs se voient attribuer un rôle : celui qui doit se cacher et celui qui doit retrouver. Le gameplay du jeu est augmenté par des items attribués aux personnages permettant d'aider les joueurs ou de les déstabiliser, par exemple, être invisible pendant quelques secondes, devenir plus rapide ou se fondre dans le décor. Hide & Seek cherche à mettre le jeu vidéo dans dans un autre contexte que celui des pratiques de hard gaming.





—

—

