

**.forme dynamique,
geste,
comportement,
connectivité**

forme dynamique, geste, comportement, connectivité

Erg (École de Recherche Graphique) - Bruxelles.
Arts numériques 1e, 2e, 3e & 4e année.
Professeur: Marc Wathieu.

Mise à jour: 12 février 2007.

*Ce livret PDF a été conçu comme un diaporama destiné à être projeté et commenté.
Pour un affichage optimisé, je vous recommande une résolution 1024 X 768,
une visualisation avec Acrobat Reader
et le raccourci ctrl+I (Windows) ou pomme+I (Mac OSX).*

Télécharger ici Acrobat Reader.

.préambule

Dans le cadre d'une introduction à la programmation destinée aux étudiants en arts numériques*, l'objectif de cette documentation est de susciter votre curiosité en posant une série de repères en images, commentés en classe.

Les exemples qui suivent expriment quelques notions propres à la conception de programmes (forme dynamique, geste, comportement, connectivité, etc) expliquées et développées ensuite à l'aide d'exemples dans PROCESSING.

* Voir : <http://www.erg.be/multimedialab/cours/programmation/index.htm>



Le comportement apparemment chaotique
d'un vol d'oiseau ou d'un banc de poissons.



Craig Reynolds : Boids (1986).
Boids est un programme de vie artificielle
simulant le comportement d'une nuée d'oiseaux en vol.

<http://www.red3d.com/cwr/boids/index.html>



Le «Jeu de la vie» imaginé par John Horton Conway en 1970 est un automate cellulaire doté de règles simples qui engendrent une complexité semblable à celle du développement de la vie.

http://en.wikipedia.org/wiki/Conway's_Game_of_Life

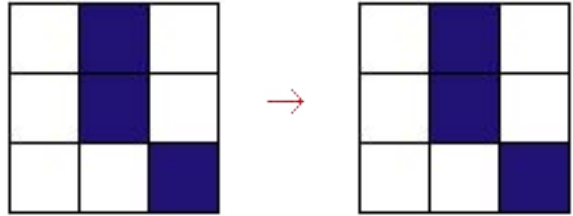
<http://www.metaphorical.net/code/processing/index2.html?page=life>
<http://fr.wikipedia.org/wiki/Complexité>

Le jeu de la vie ne nécessite aucun joueur : il s'agit d'un automate cellulaire, c'est-à-dire un modèle où chaque état conduit mécaniquement à l'état suivant à partir de règles pré-établies.

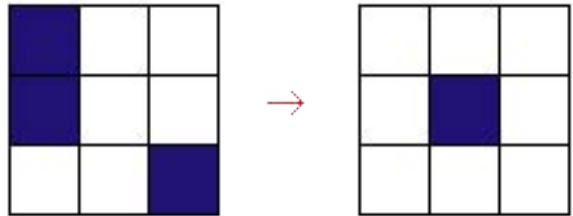
Le jeu se déroule sur une grille à deux dimensions, dont les cases (les «cellules») peuvent prendre deux états distincts : «vivantes» ou «mortes».

À chaque étape, l'évolution d'une cellule est entièrement déterminée par l'état de ses huit voisines de la façon suivante :

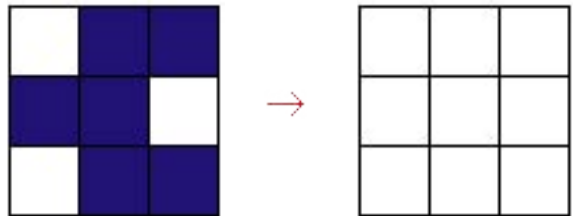
1- Si une cellule vivante (ici en bleu) est entourée de deux ou trois voisines vivantes, alors elle reste vivante. le reste, sinon elle meurt.

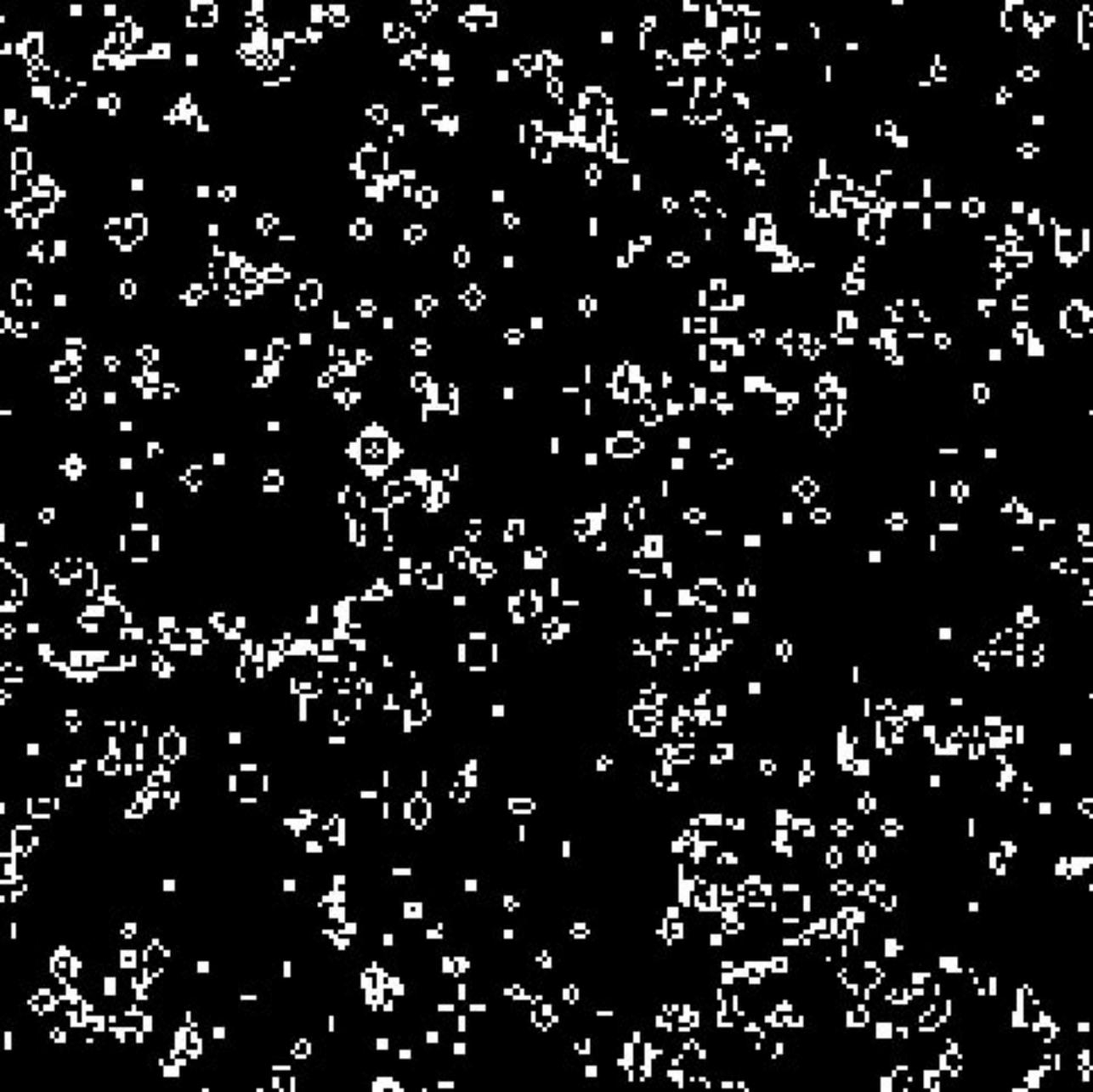


2- Si une cellule morte est entourée d'exactly trois voisines vivantes devient vivante (elle naît).



3- Dans tous les autres cas, la cellule meurt.





Michael Davis a programmé un «Jeu de la vie» avec PROCESSING :
<http://processing.org/learning/examples/cellularautomata1.html>

Autre exemple programmé avec PROCESSING par William Ngan :
<http://www.metaphorical.net/code/processing/index2.html?page=life>

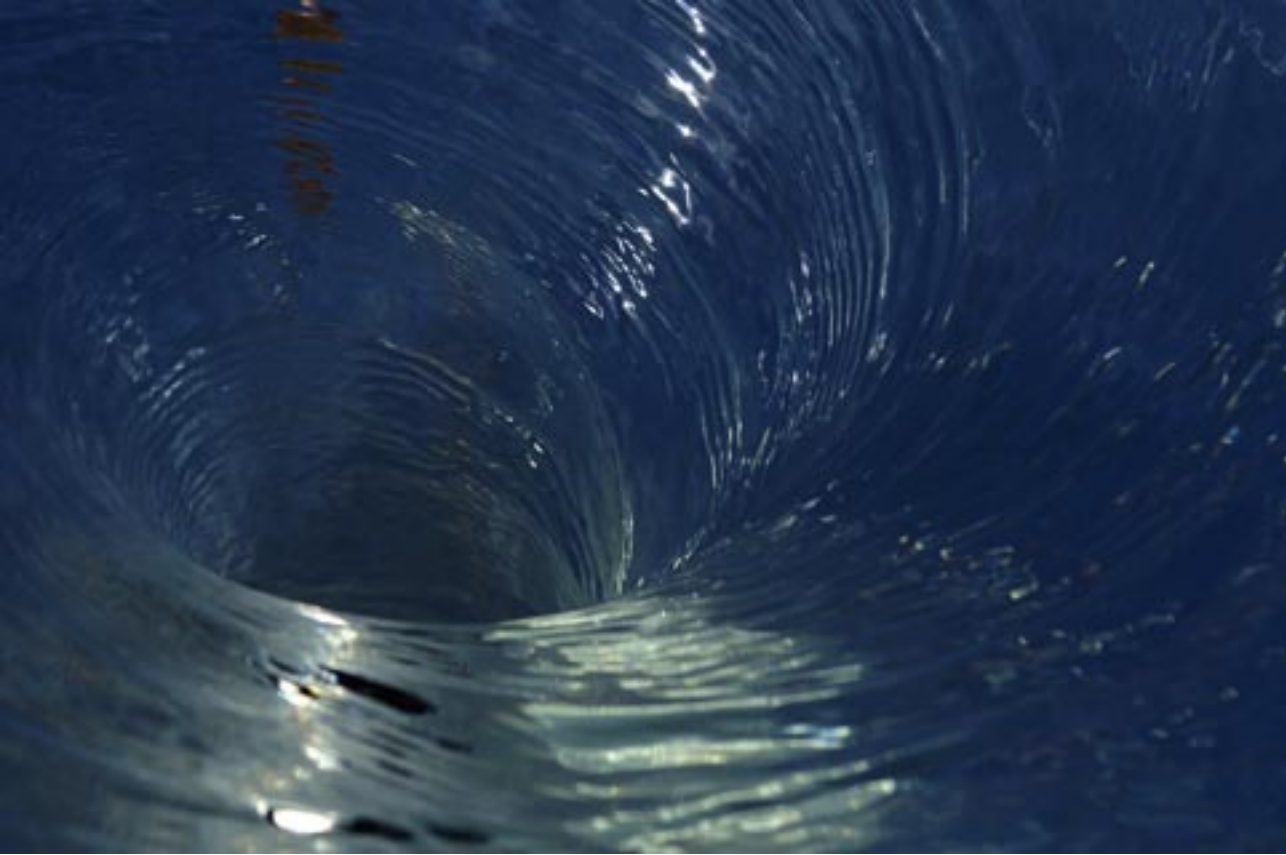


Le mouvement d'une chenille.



Alvar Aalto : Paravent (1935-1936).

<http://www.dmk.dk/details/13405/>



Mécanique des fluides : un vortex.



L'ouragan Katrina.

http://fr.wikipedia.org/wiki/Ouragan_Katrina



Attraction/magnétisme.



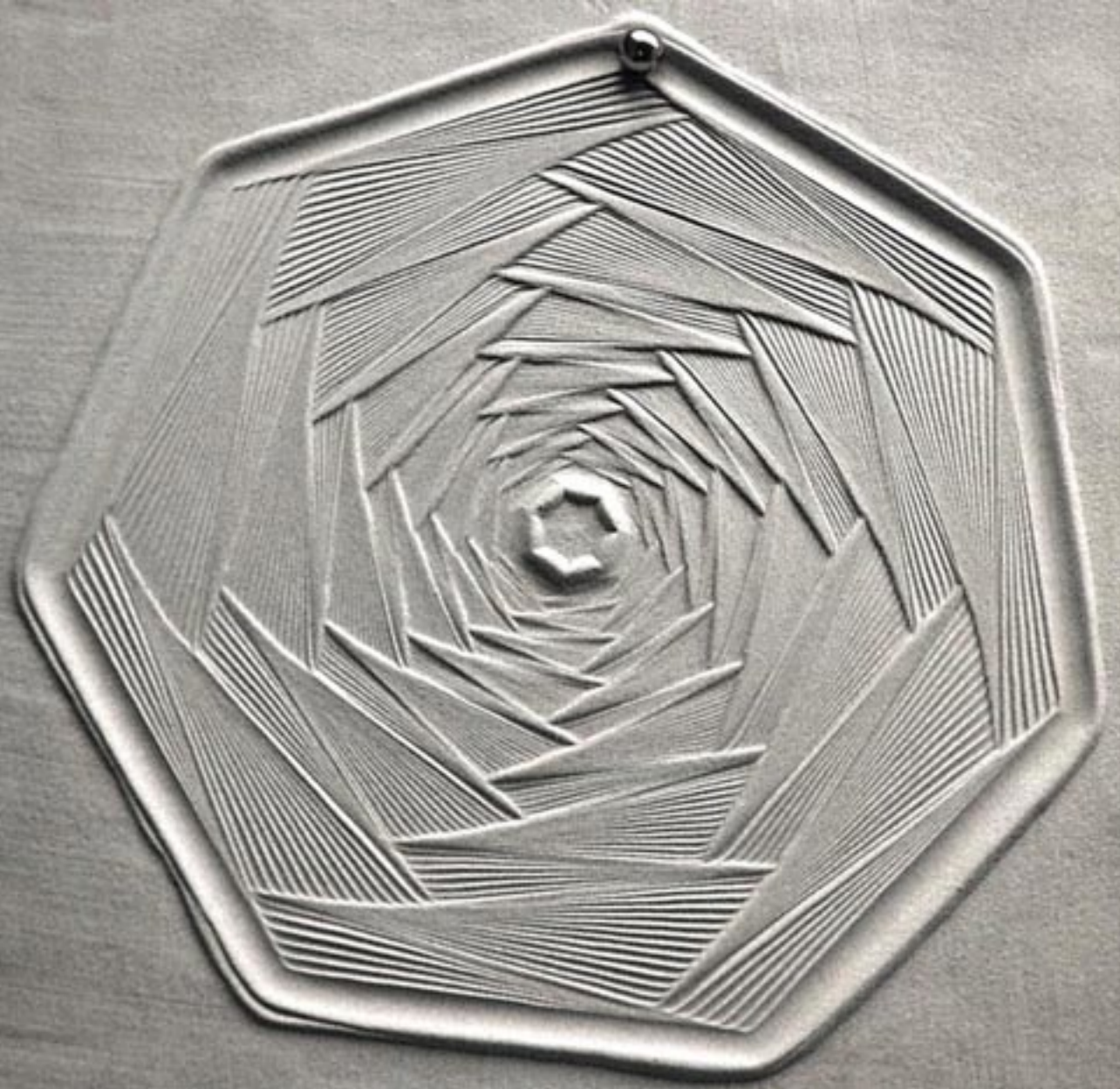
Josef Beuys : I like America and America likes me (1974).
Emmené directement en ambulance de l'aéroport JFK à la galerie René Block, Beuys s'y enferme pendant 3 jours pour cohabiter avec un coyote.

http://en.wikipedia.org/wiki/Joseph_Beuys



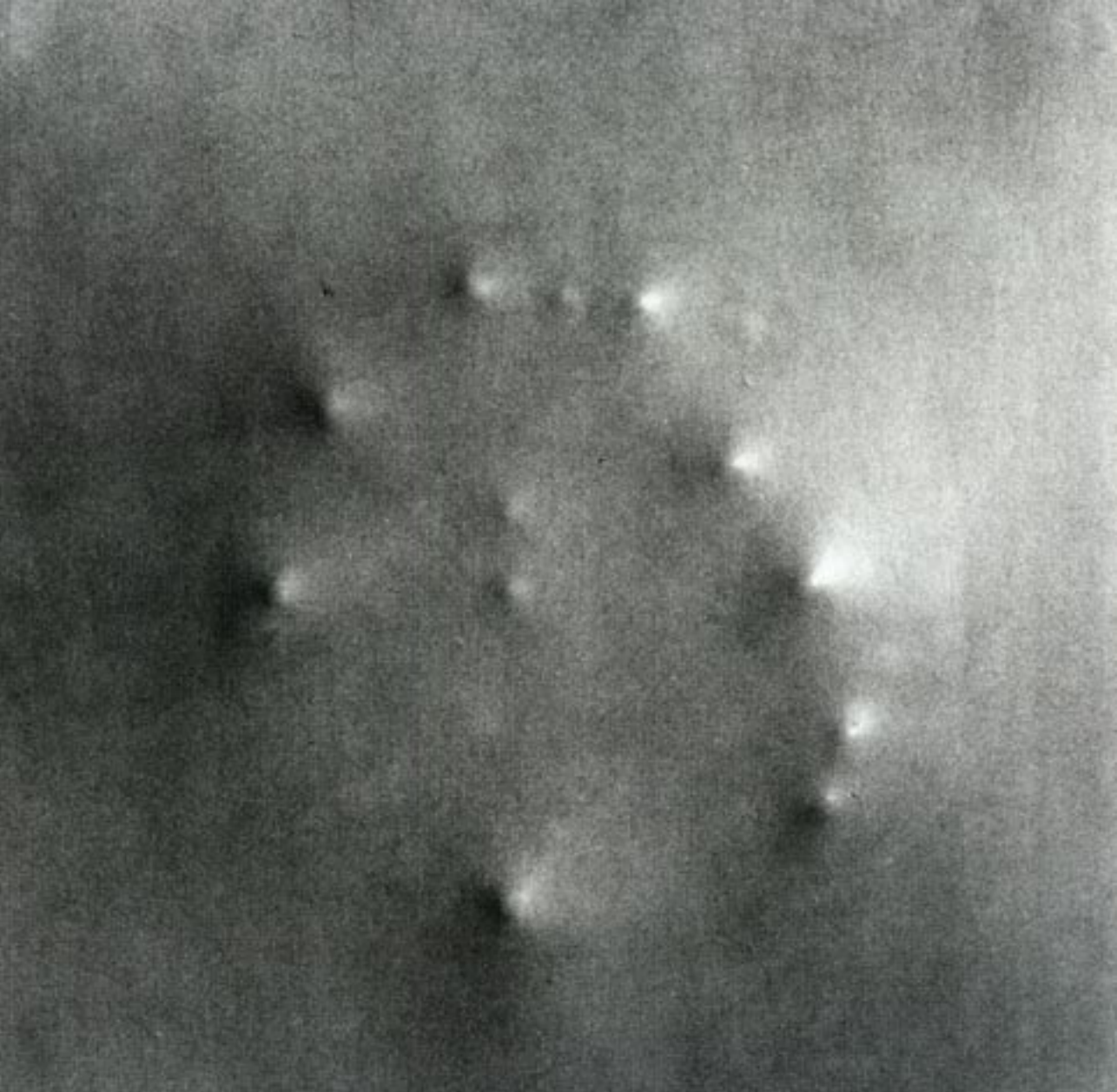
Jardin Zen : le temple de Ryoan-ji à Kyoto.
Chaque jour, un moine exécute un rituel de méditation
qui consiste à ratisser harmonieusement le sable du jardin.

<http://en.wikipedia.org/wiki/Ryoan-ji>

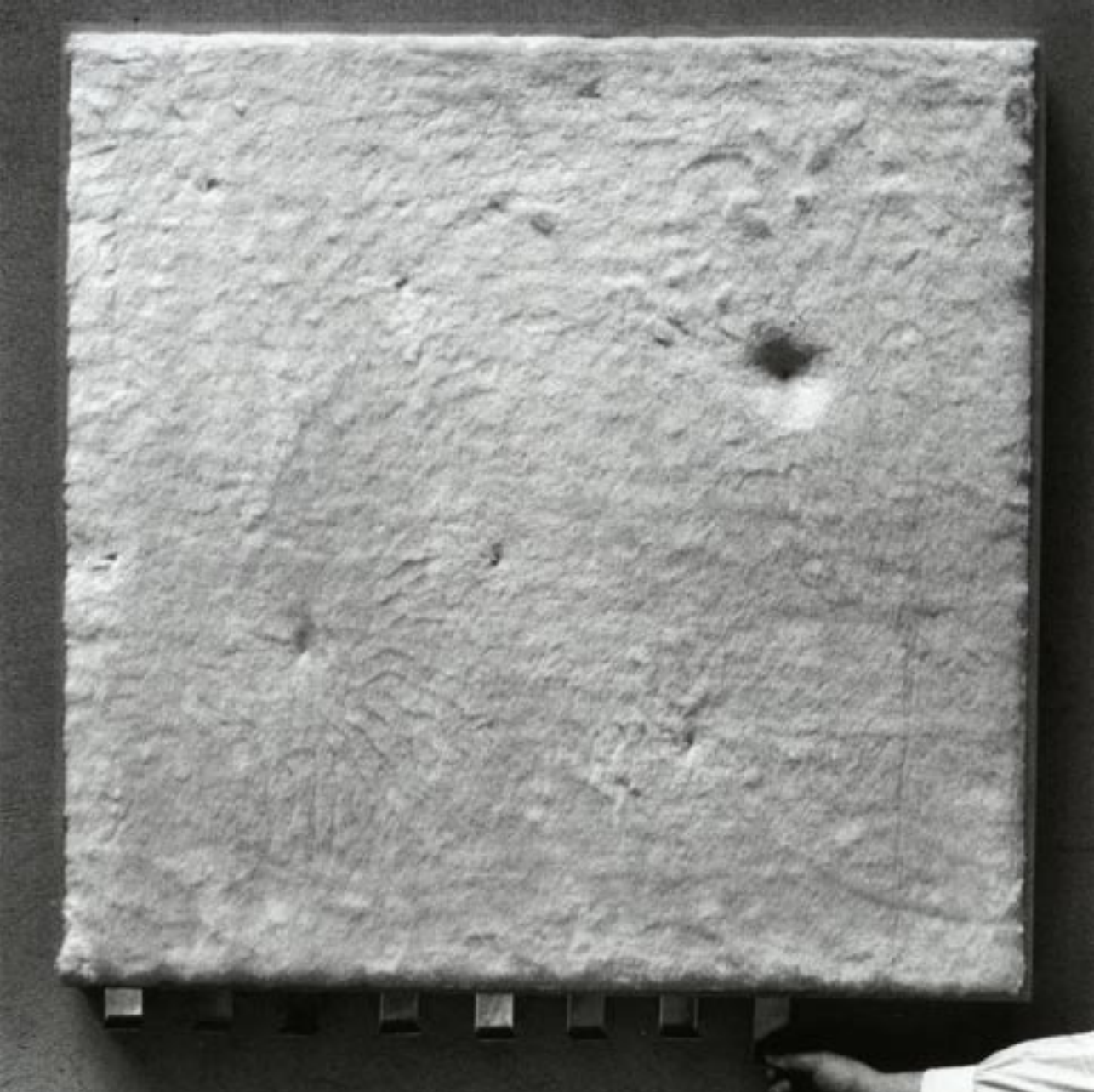


Jean-Pierre Hébert : Sand As Medium - Sisyphus (1998).
Aimant motorisé, bille, logiciel, sable.

<http://hebert.kitp.ucsb.edu/sand/news.html>



Gianni Colombo : Rilievo Intermutabile (1959).
Évocation du toucher.



Gianni Colombo : Superficie in variazione (1959).
Sollicitation du spectateur, sculpture programmée.



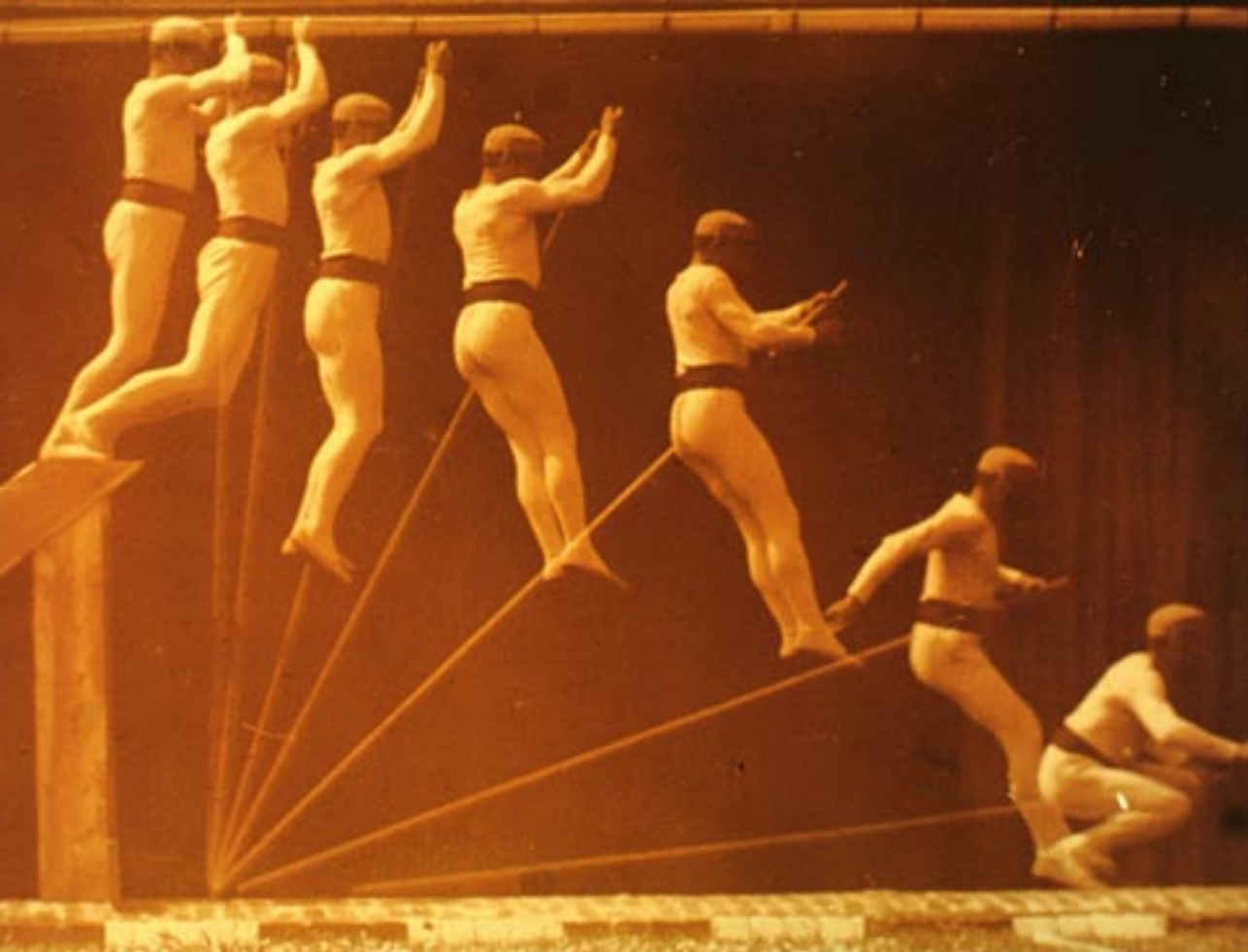
MIT Tangible Media Group : Sandscape (2002-2003).
Interface tactile, manipulation de sable et modélisation en temps réel.

<http://tangible.media.mit.edu/projects/sandscape/>



Christa Sommerer & Laurent Mignonneau : Nanoscape (2002).
Perception de formes invisibles définies
par un champ magnétique (attraction/répulsion).

<http://www.interface.ufg.ac.at/christa-laurent/>



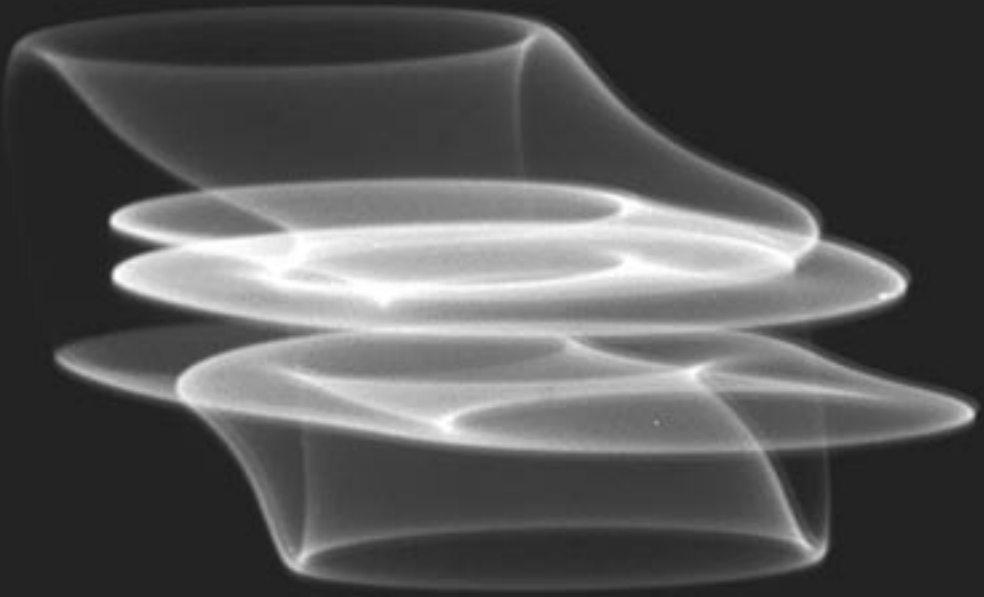
Etienne-Jules Marey : Le saut à la perche (1887-1890).
Analyse du mouvement.

<http://www.expo-marey.com/>



David Rokeby : Watch (1995).
Détection de mouvement, logiciel, projection.

<http://homepage.mac.com/davidrokeby/home.html>

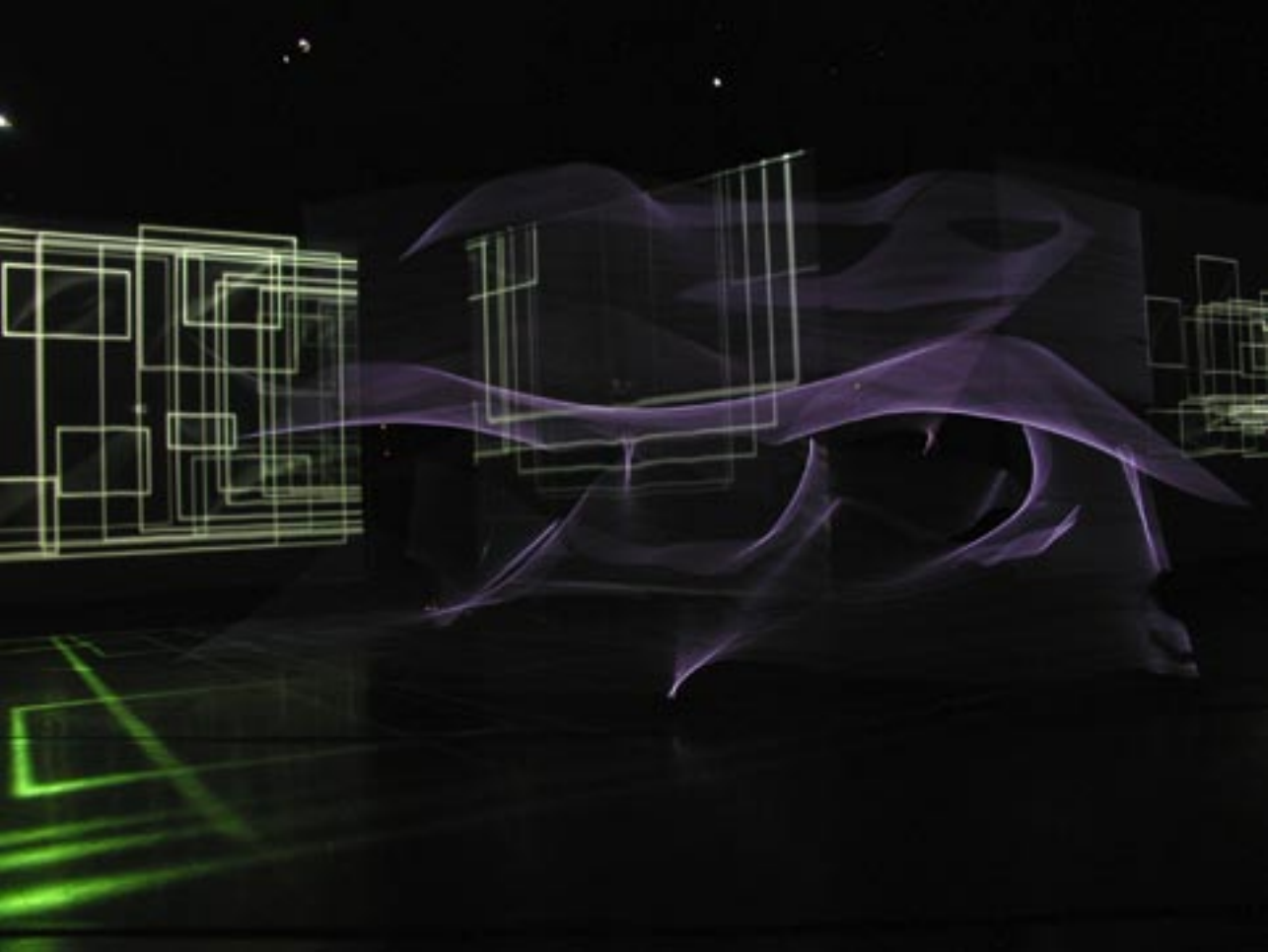


Herbert W. Franke : Elektronische Grafiken (1961/62).

<http://www.zi.biologie.uni-muenchen.de/~franke/>

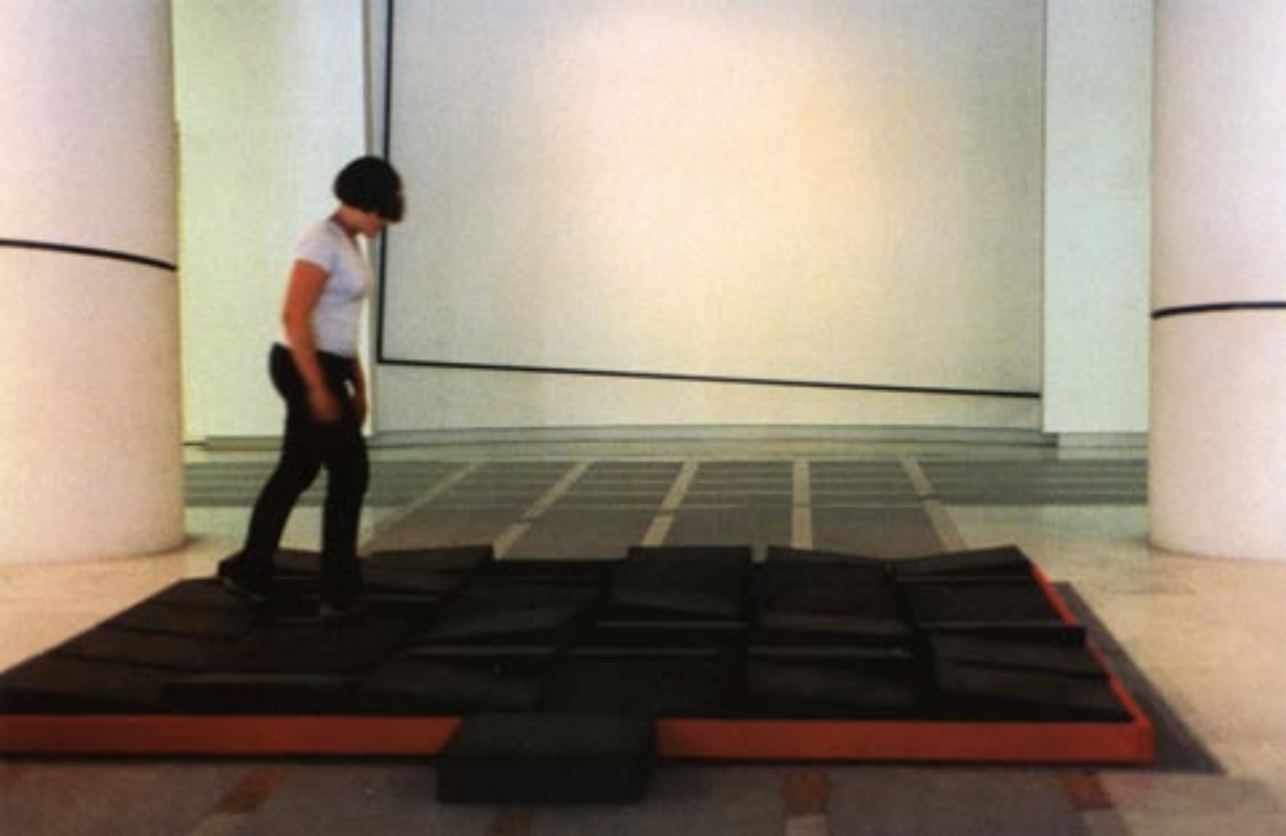


Etienne-Bertrand Weill : Allegro Vivace (1982).
Métaformes, visualisation du mouvement.



Lab[au] (Laboratory for architecture and urbanism) :
Man in eSPACE.mov (2005-2006).
Captation de mouvement, logiciel, projection.

<http://www.lab-au.com/>



Julio Le Parc : Dalles mouvantes (1964).
Sculpture interactive.



Seiko Mikami & Sota Ichikawa : Gravicells - Gravity and resistance (2004).
Sol réactif, logiciel, projection.

<http://www.g--r.com/>



Hanna Haaslahti & Yves Bernard : White Square (2002-2003).
Captation de mouvements, logiciel, projection.

<http://www.imal.org/WhiteSquare/>



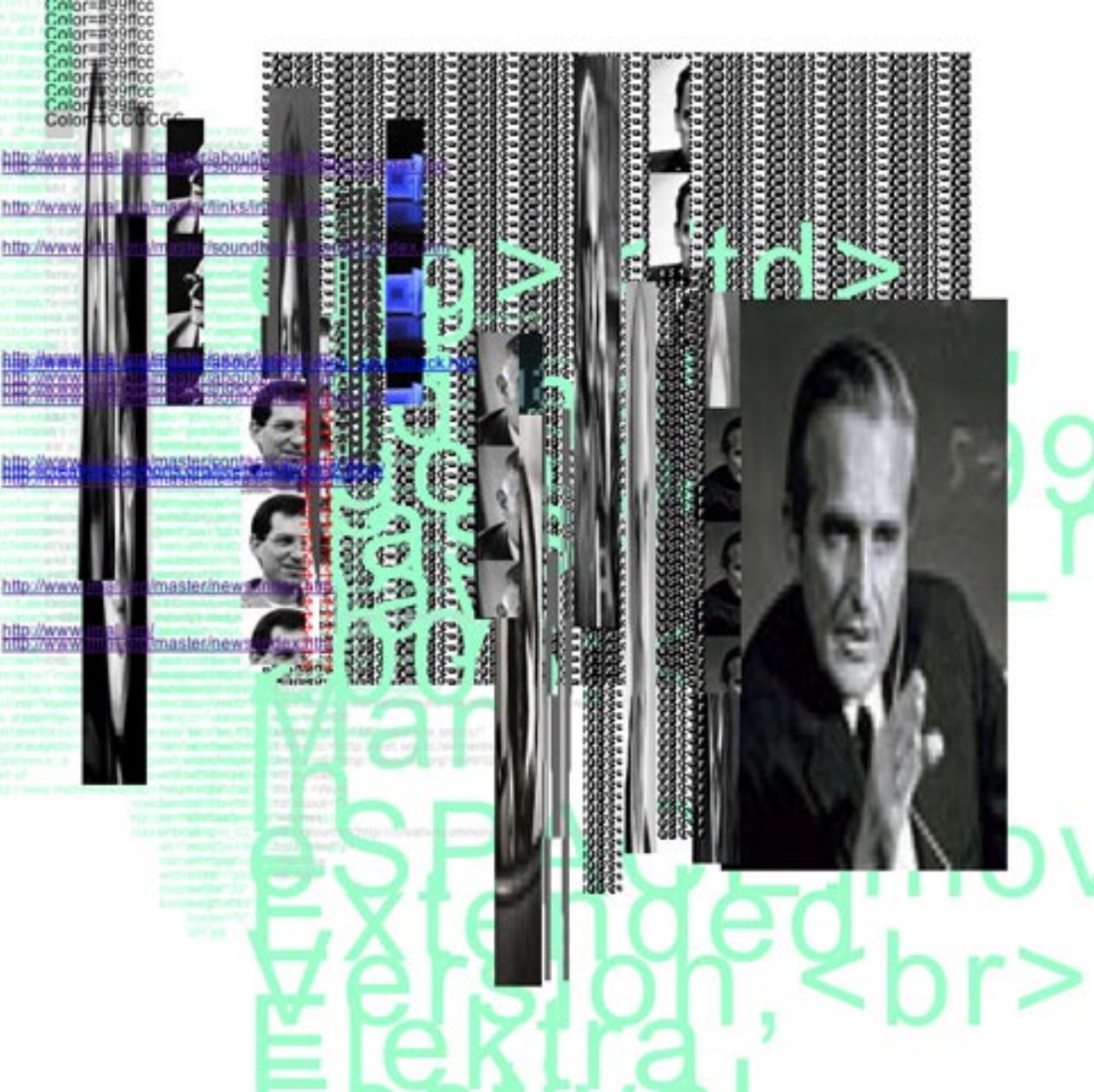
Joseph Beuys : Capri Battery (1985).
Connectivité.

http://en.wikipedia.org/wiki/Joseph_Beuys
http://www.tate.org.uk/modern/exhibitions/beuys/room_beuys.shtm



David Rokeby : n-Cha(n)t (2001).
Machines en réseau, reconnaissance vocale, diffusion synchronisée.

<http://homepage.mac.com/davidrokeby/home.html>



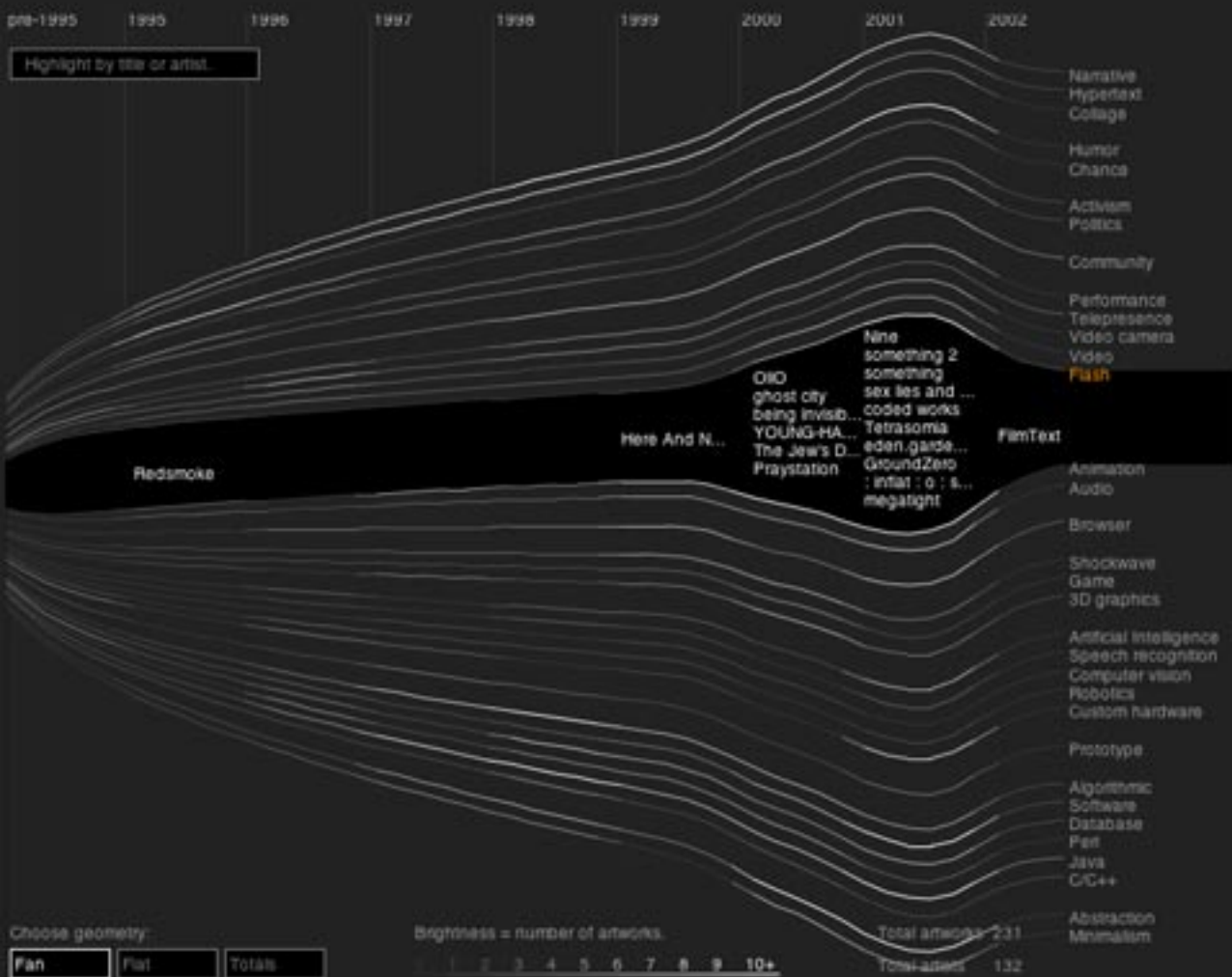
Mark Napier : Shredder (1998).
Altération du code HTML avant sa lecture par le navigateur.

<http://potatoland.com/shredder/>



Mongrel : Nine(9) - a social software project (2003-2004).
Outil dynamique et collaboratif en ligne.

<http://www.mongrelx.org/>
<http://www.linker.org.uk/>
http://kop.fact.co.uk/KOP/artists_2003/html/a2.html



Martin Wattenberg : A Net Art Idea Line (2001).
Interface, chronologie interactive du net-art.

<http://artport.whitney.org/commissions/idealine.shtml>

.réactivité du logiciel

Un logiciel est un objet réactif: il possède la capacité d'interagir dynamiquement avec son environnement.

Cette spécificité soulève certainement des questions de sens, de conception et de réalisation, mais aussi de perception. Elle renouvelle certains aspects de notre rapport à l'art et modifie en tout cas les relations entre art et science. En effet, la notion d'«interaction dynamique» suppose une modélisation de propriétés physiques, une procédure, un traitement de données, l'expression de valeurs, etc... Tout un vocabulaire issu des technologies informatiques au sens large (logique, intelligence artificielle, mathématiques) qui, brutalement, fait irruption dans le domaine de l'art.

Le code est un matériau. Le langage est un matériau. On prend la plume ou la parole pour exprimer des idées ou des points de vue. La grammaire et la rhétorique sont des outils propres au langage. Ainsi, l'expression par le code possède ses propres outils : un ordinateur, un environnement de programmation, un langage et une syntaxe. Mais il faut surtout des outils conceptuels appropriés qui permettront de manipuler ce code-matériau de manière pertinente.

C'est ici qu'apparaît une possible et surprenante confusion entre le fond (le sens des choses) et la forme (l'apparence des choses). En effet, l'acte de programmation peut être associé à un acte de modélisation : modéliser une forme, modéliser un comportement, modéliser une action, etc.

Edmond Couchot* l'explique clairement :

«[...] techniquement, l'image numérique est étroitement dépendante des processus programmatiques qui la produisent. Or ces modèles de simulation numérique utilisés dans les programmes sont, comme tout modèle scientifique, déjà des interprétations formalisées du réel. [...] Il en résulte que sur un écran d'ordinateur on ne peut figurer, donner une forme visible, sensible, qu'à ce qui est déjà intelligible, déjà interprétation rationnelle du monde. Les artistes se trouvent alors dans la délicate nécessité de créer du sensible (des formes artistiques) avec de l'intelligible (des programmes informatiques), en quelque sorte des résidus applicatifs de la science [...]».

Cette réalité fondamentale de la programmation va nécessiter beaucoup de vigilance pour garder du recul sur le sens et les motivations d'un travail artistique.

* Lire l'extrait «La science comme présence efficiente» en annexe.

Casey Reas, lors de sa conférence* au festival Ars Electronica 2003 (Linz - Autriche), citait six aspects spécifiques («noyaux d'expressions numériques») à la réactivité et aux processus d'un logiciel, c'est-à-dire six propriétés exploitables lors de la conception d'un programme :

Forme dynamique :

une forme réactive, qui réagit à des stimuli externes en se reconfigurant.

Geste :

la capacité de transcrire et d'interpréter un geste.

Comportement :

un mouvement possédant l'apparence d'une intention.

Simulation :

la simulation de phénomènes physiques (gravité, vitesse, etc).

Auto-Organisation :

la capacité des éléments à s'auto-structurer, à se régénérer.

Adaptation :

une capacité à changer, à s'adapter, induisant que le logiciel possède une représentation de lui-même et une compréhension de son contexte.

* Voir «Code - The Language of our Time» :

http://www.erg.be/multimedialab/doc/casey_reas.htm

Je retiendrai ici les trois premières notions, qui me semblent être les plus génériques. Sans pour autant les minimiser, *simulation*, *auto-organisation* et *adaptation* traduisent des singularités de *comportement* (souvent cumulées) qui nécessitent une solide expertise en programmation et qui pourront ensuite faire l'objet d'une recherche plus avancée.

J'ajouterai cependant la notion de «connectivité» qui illustre la capacité d'un logiciel à se connecter à un réseau ou à d'autres logiciels, incluant la notion de communauté.

Récapitulons :

Forme dynamique :

une forme réactive, qui réagit à des stimuli externes en se reconfigurant.

Geste :

la capacité de transcrire et d'interpréter un geste.

Comportement :

un mouvement possédant l'apparence d'une intention.

Connectivité :

la capacité de se connecter et d'échanger de l'information

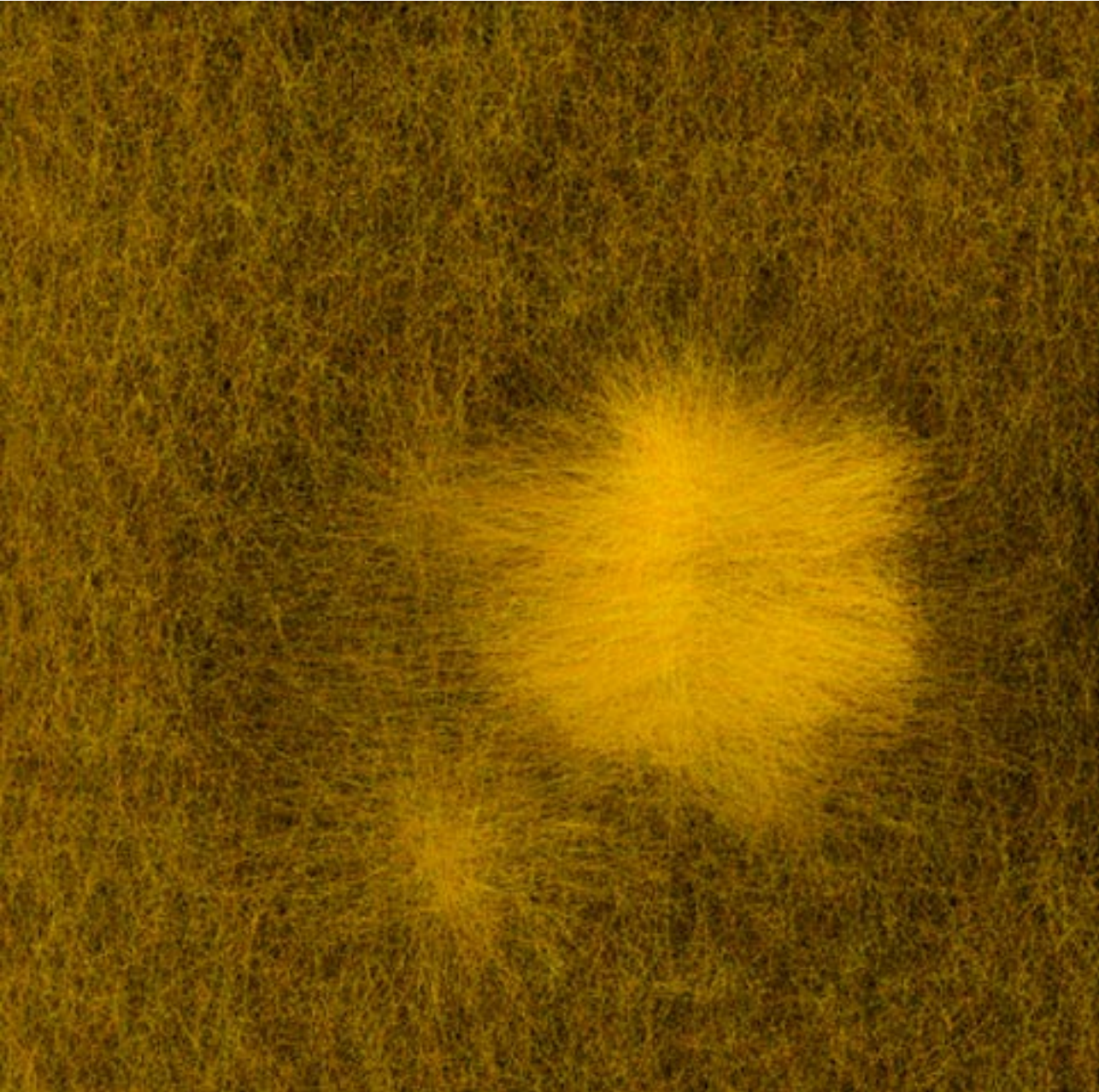
Pour illustrer ces propriétés, j'ai choisi des exemples *complexes* mais bien documentés , suivis d'exemples *simples* que nous examinerons ensemble.

Pour ce qui est de l'installation/performance de LAB[au], il s'agit d'un projet développé avec le logiciel «SPACE» développé par LAB[au], peu documenté en ligne, mais que je connais particulièrement bien.

Forme dynamique :

Keith Peters : Ring of fire, un nuage de particules qui se reconfigure et évolue selon les mouvements et les clics.

<http://www.bit-101.com/p5/particles/ringoffire/applet/>



Exemple simple de forme dynamique:

```
// Mouse 2D by REAS
// http://reas.com
// Moving the mouse changes the position and size of each box.
// Updated 21 August 2002

void setup()
{
  size(200, 200);
  noStroke();
  colorMode(RGB, 255, 255, 255, 100);
  rectMode(CENTER);
}

void draw()
{
  background(51);
  fill(255, 80);
  rect(mouseX, height/2, mouseY/2+10, mouseY/2+10);
  fill(255, 80);
  rect(width-mouseX, height/2, ((height-mouseY)/2)+10, ((height-mouseY)/2)+10);
}
```

Voir en ligne:

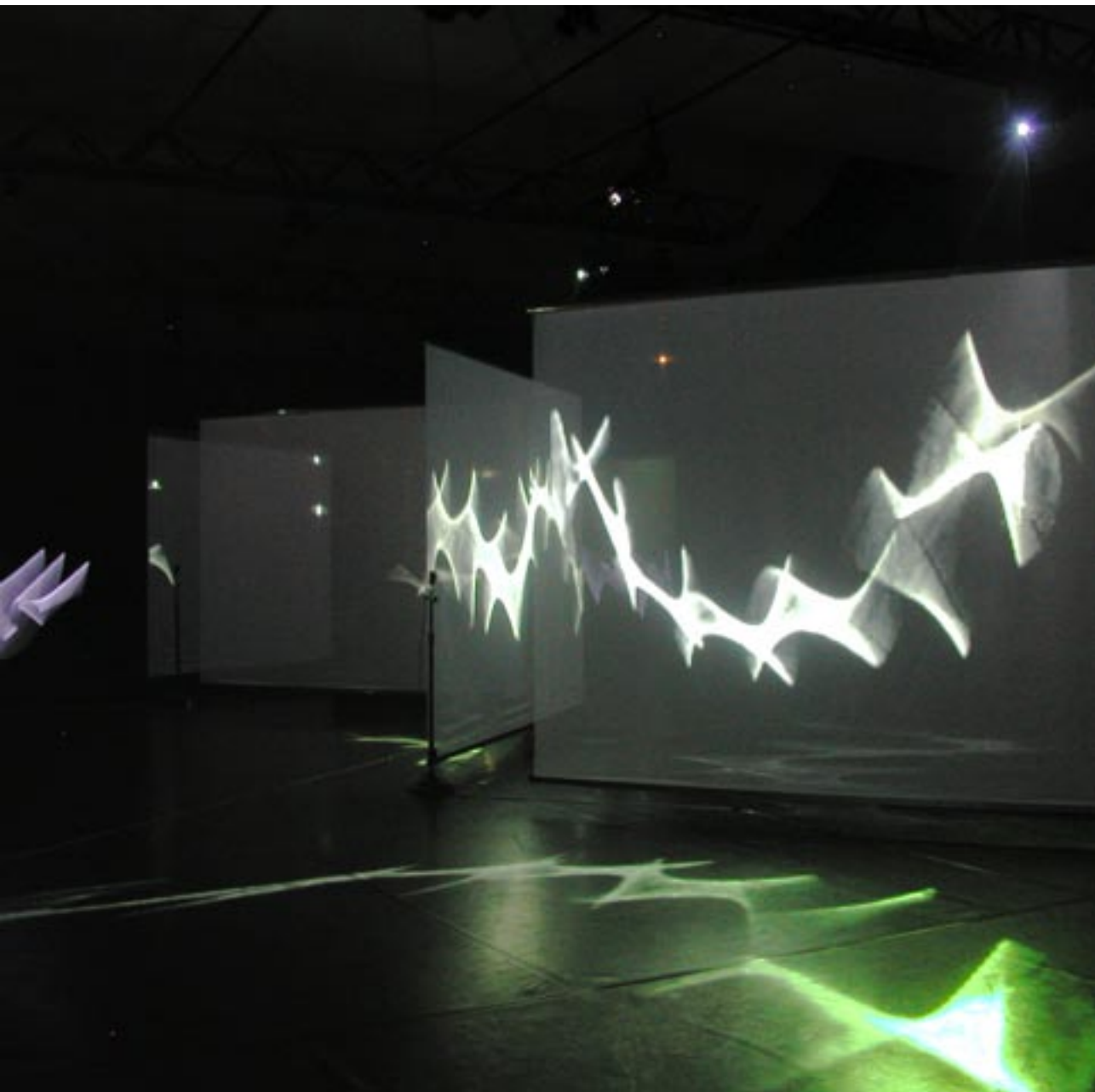
<http://processing.org/learning/examples/mouse2d.html>

Geste :

LAB[au] : «Man in eSPACE.mov», installation-performance (2004/2006),
captation video, software, projection multi-écrans.

<http://www.lab-au.com/>

<http://www.mast-r.org/>



Exemple d'une traduction de geste:

```
// Click by REAS  
// http://reas.com  
// Move the mouse to position the shape.  
// Press the mouse button to invert the color.  
// Updated 21 August 2002
```

```
int size = 30;
```

```
void setup() {  
  size(200, 200);  
  ellipseMode(CENTER);  
  fill(126);  
  noStroke();  
  rect(0, 0, width, height);  
}
```

```
void draw() {  
  if(mousePressed) {  
    stroke(255);  
  } else {  
    stroke(51);  
  }  
  line(mouseX-30, mouseY, mouseX+30, mouseY);  
  line(mouseX, mouseY-30, mouseX, mouseY+30);  
}
```

Voir en ligne:

<http://processing.org/learning/examples/click.html>

Comportement :

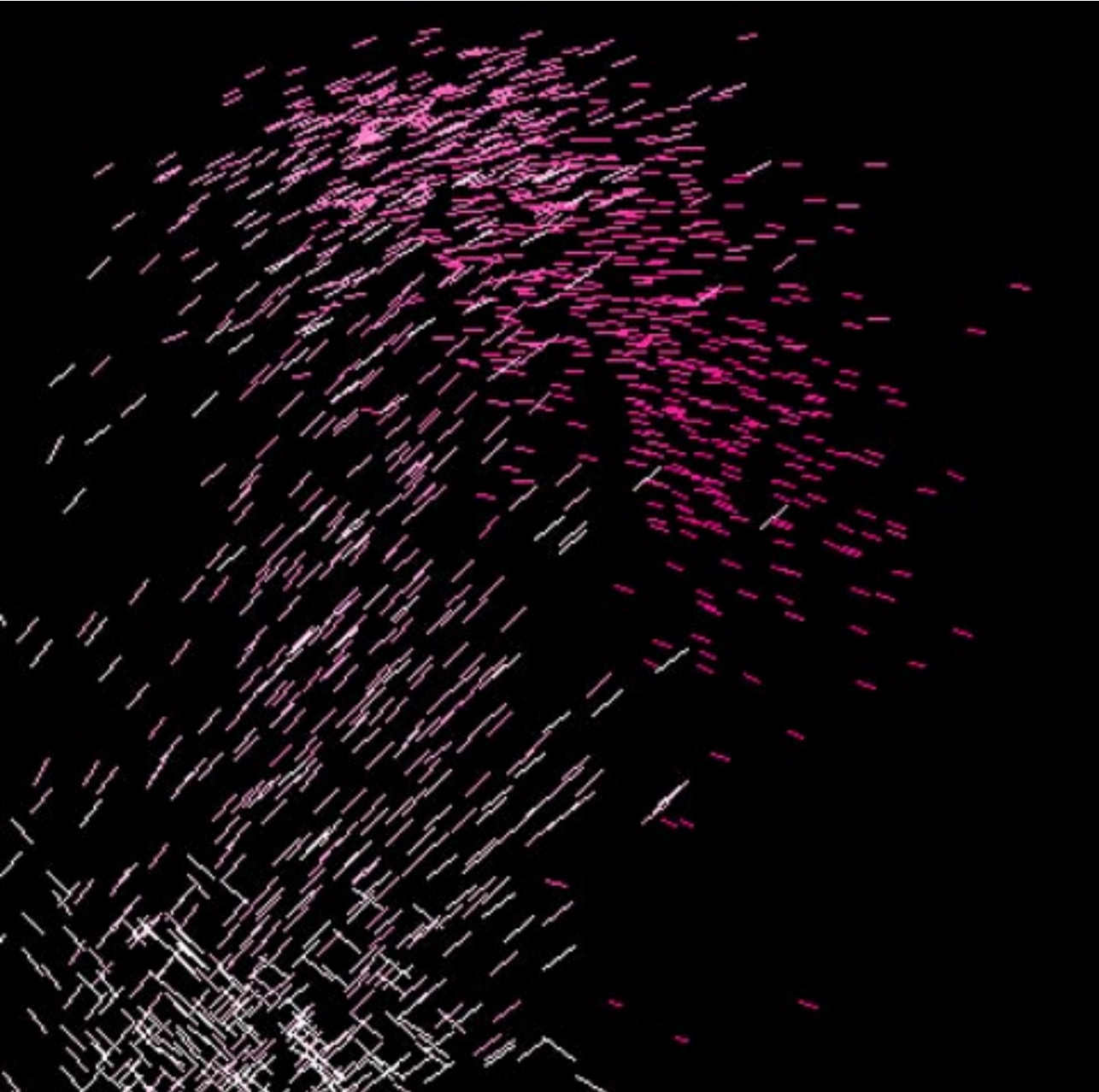
Florito (Markus Graf) : Fly Swarm, créé avec PROCESSING, semble être un essaim d'insectes possédant une inertie (un poids) et une vitesse propres.

<http://www.florito.net/FlySwarm/applet/index.html>

Voir aussi :

William Ngan : «Pond», créé avec PROCESSING, décrit le comportement d'un groupe de poissons dans un étang face à certains types d'évènements.

http://metaphorical.net/nature/pond_p5.html



Exemple : contrariété.

```
// déclaration d'une variable globale  
int largeur;
```

```
// définir les conditions d'affichage  
void setup() {  
  largeur = 400; // donner une valeur à notre variable  
  size(largeur, 400); // taille de la fenêtre  
  noStroke(); // pas de trait de contour  
  smooth(); // pas de crénelage  
}
```

```
// mode «dessin»  
void draw() {
```

```
  int x; // déclaration d'une variable  
  
  background(255, 100, 20); // le fond annule la trace (supprimez-le pour voir)  
  
  x = largeur - mouseX; // utilisation de la variable  
  fill (10, 150, 20); // couleur du cercle  
  ellipse(x,mouseY,55,55); // définition d'un cercle  
}
```

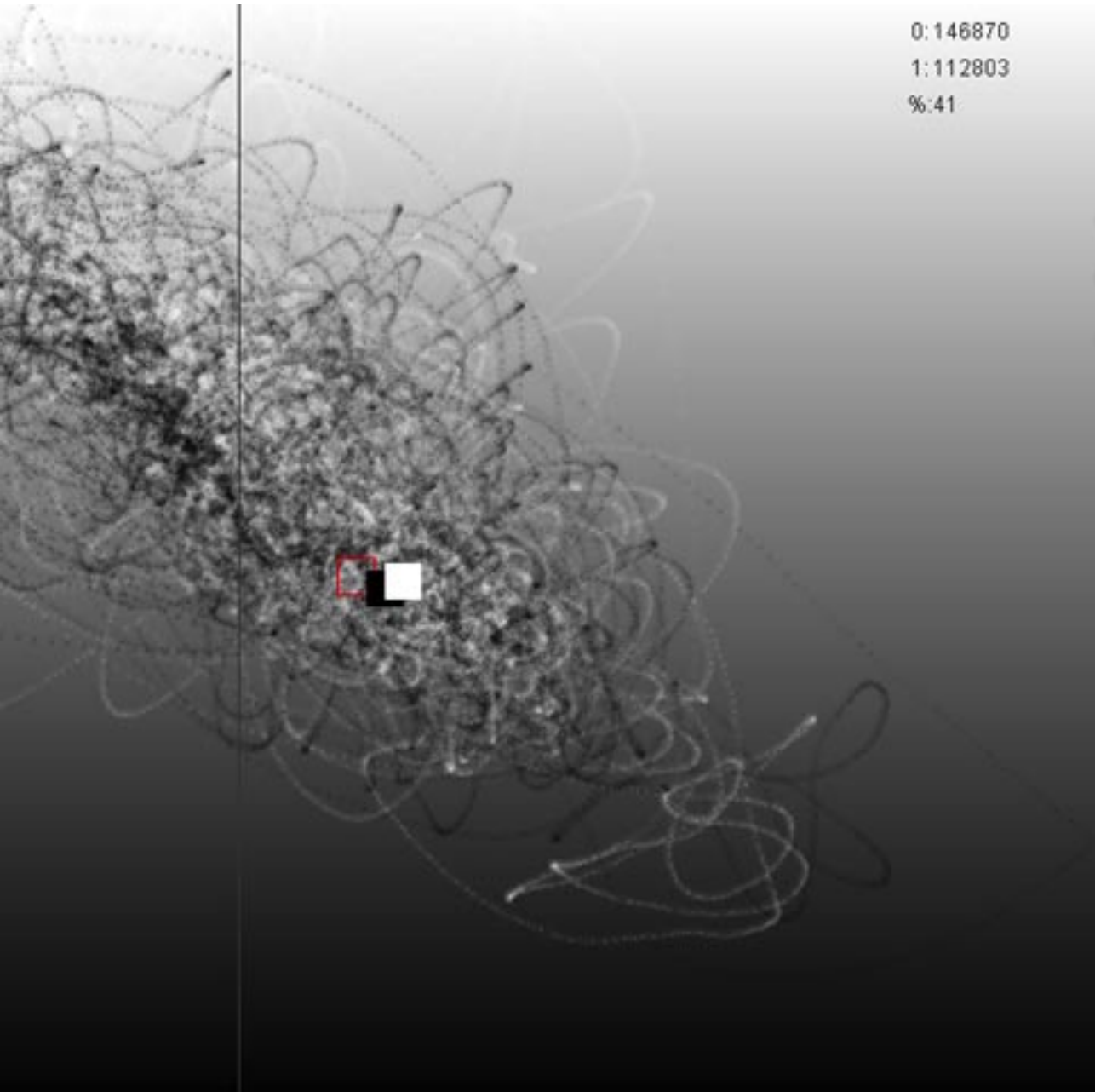
Connectivité :

Black and white crée un dessin d'après le flux de données binaires de CNN.COM.
Les 0 apparaissent en noir et déplacent le traceur horizontalement.
Les 1 apparaissent en blanc et déplacent le traceur verticalement.
Les deux traceurs s'attirent mutuellement.

<http://www.potatoland.org/blackwhite>

<http://itserve.cc.ed.nyu.edu/RSG/carnivore/>

<http://itserve.cc.ed.nyu.edu/RSG/carnivore/processing.php>



Exemple : Accès à Internet (affichage d'images provenant d'une webcam).

// Laurent Doucet, Erg 2005.

// définir la taille de la fenêtre

```
void setup() {  
  size(352,240);  
}
```

// mode «dessin»

```
void draw() {
```

// chargement de l'image

```
PImage a;
```

```
a = loadImage(«http://images.earthcam.com/ec_metros/ourcams/fridays.jpg»);
```

// masque alpha

```
tint(255, 255, 255, 50);
```

// position de l'image

```
image(a,0,0);
```

```
}
```