



# FONTE

SEMAINE 8

18 AVRIL 2012

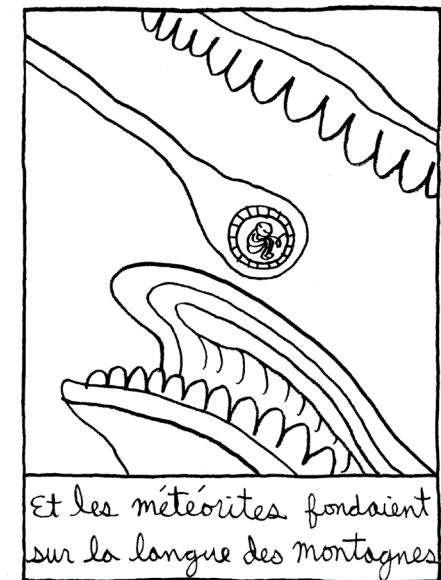
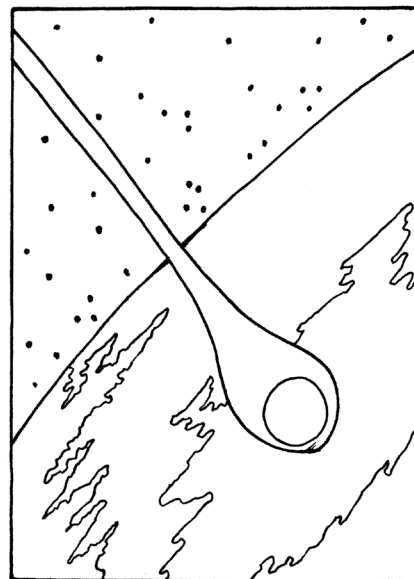
LA MAISON  
DE LA BANDE DESSINÉE  
DE MONTRÉAL

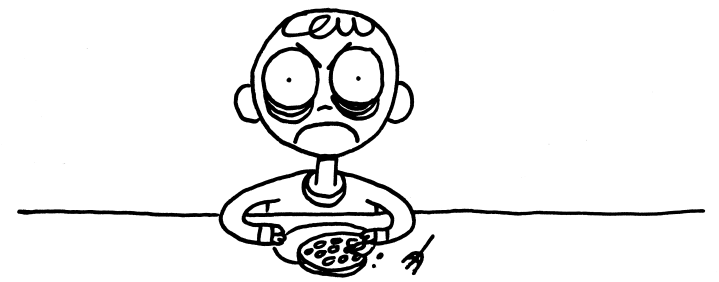
VINCENT GIARD ÉDITEUR  
FONTE@ENCRE.ORG  
NÉ #ÉDITEUR: 84

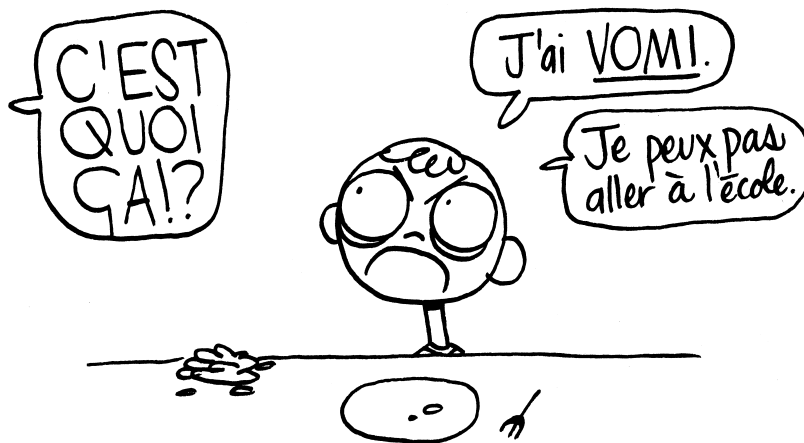
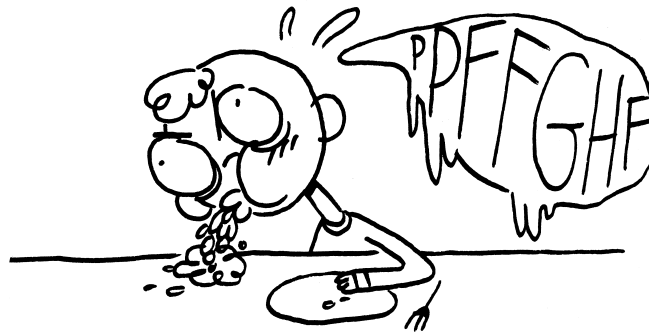
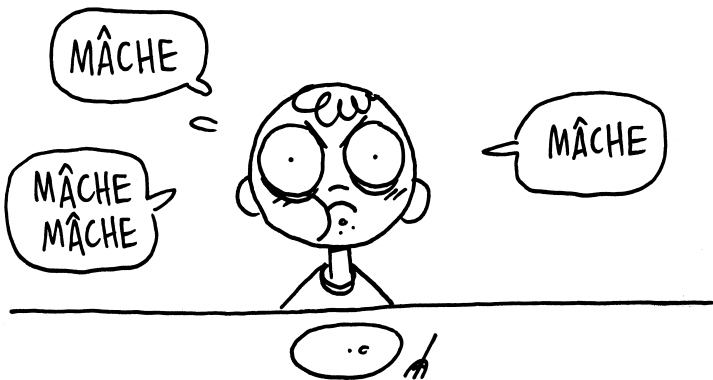
UN POUR RELIRE  
UN POUR DONNER  
UN POUR VENDRE

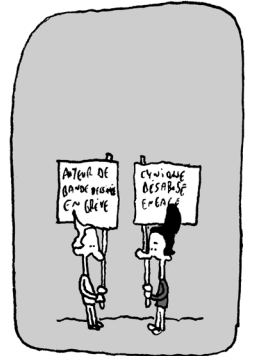
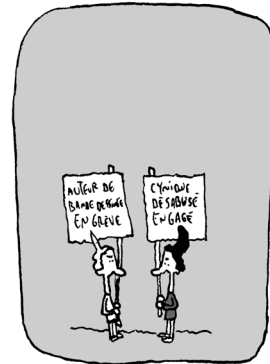
TIRAGE: **3 EXEMPLAIRES  
PAR PARTICIPANT,  
TANT QUE LA GRÈVE  
ÉTUDIANTE PERDURERA.**

AVEC: **J — B — R**  
**FRANÇOIS** SAMSON DUNLOP  
**DAVID TURGEON**  
**JULIEN CASTANIÉ**  
**CATHON** FONTAINE ROUSSEAU  
**ALEXANDRE**  
**EVLYN M.** **LACHAPELLE**  
**NICOLAS**  
**DARNZIAK**  
**GÈBÉ** (REMIX NICOLAS LACHAPELLE)  
**VINCENT** **GIARD.**









# C

On peut définir un langage de programmation de la manière suivante : c'est une manière de dire à une machine (notamment un ordinateur) les tâches qu'on souhaite qu'elle accomplisse ; une manière de traduire un projet quelconque en une série d'*instructions* dépourvues de toute ambiguïté. Ces instructions doivent toujours être mises en contexte : le programme détermine dans quel ordre, à combien de reprises et sous quelles conditions elles doivent s'exécuter.

La programmation, vue de cet angle, est simplement une technique consistant à *faire faire* : la machine est un objet extérieur à soi, sur lequel on a tous les pouvoirs, et que l'on contrôle (donc) à la manière d'un général ses troupes. Une assertion informatique (une ligne de code) est, d'un point de vue linguistique, un énoncé performatif, qui « fait ce qu'il dit ». Il s'agit d'un ordre de surcroît toujours obéi, sauf — et c'est quand même là tout le problème — s'il est syntaxiquement incorrect, ce qui arrive somme toute assez fréquemment.

Si je dis à la machine :

```
printf('L'obligatoire est ce qu'il est nécessaire que fasse l'homme bon.');
```

alors la machine agira conséquemment en imprimant (sur l'écran, ou sur tout autre support de sortie prédéterminé)

de somme prédéterminé) la phrase indiquée. Si cependant j'ometts la moindre parenthèse, ou si j'oublie de protéger la moindre apostrophe du texte par une barre oblique inversée (ce qui eût fait croire au compilateur que la chaîne de texte est terminée, alors qu'elle ne l'est pas), alors il y a erreur de syntaxe et, irrémédiablement, le programme flanche, la machine abandonne, avec dans le meilleur des cas une indication au programmeur de ce qu'il convient de faire pour réparer l'erreur.

Le compilateur (c'est le nom du processus qui traduit tout programme dans un dialecte que l'ordinateur peut immédiatement saisir, le *langage machine*, j'y reviendrai), le compilateur donc n'admet aucune faute de syntaxe, c'est-à-dire qu'il refuse absolument toute ambiguïté. Le compilateur ne veut pas avoir à choisir entre une interprétation et une autre : il demande au programmeur une clarté absolue.

Paradoxalement donc pour un outil dont on s'attend à ce qu'il obéisse à chacun de nos commandements, *on ne discute pas avec la machine*. Il n'y a rien de plus idiot qu'un ordinateur. Si on lui demandait d'interpréter nos hypothétiques sentences ambiguës, il lui faudrait dépenser une partie de ses capacités à cette seule fin, ce qui lui en laisserait moins pour accomplir la tâche pour laquelle on le destinait vraiment. Mieux vaut *optimiser* sa requête, donc, et donner tout de suite à la machine ce qu'elle requiert. Paradoxe donc, en ce que l'humain, face à la machine, *reste fondamental*

machine, reste fondamentalement démuni : s'il n'apprend pas à s'adresser à elle, celle-ci restera bornée, idiote, et ne lui sera strictement d'aucune utilité.

Il convient donc de corriger notre définition initiale, et de considérer le langage de programmation, non comme une manière brute d'accumuler les instructions, mais plutôt comme un *moyen d'expression* visant à décrire et à résoudre un problème donné. L'avantage de cette définition est qu'elle met entre parenthèses l'existence même de la machine censée opérer ladite résolution du problème. En mathématiques, l'algèbre et la notation fonctionnelle sont des langages de programmation en miniature, et leur invention prédate de loin celle de l'informatique. De même, un langage informatique suffisamment avancé peut exprimer des concepts qui transcendent l'utilisation d'une machine particulière (parce que, bien sûr, chacune a ses particularités) : on dit qu'il gagne en *abstraction*.

Et l'histoire des langages informatiques est précisément le récit de cette conquête d'une abstraction toujours plus grande, qui pourtant conserve quelque chose comme le souvenir de ses origines concrètes : la forme binaire du courant électrique, l'accumulation de la mémoire, la mécanique des branchements.

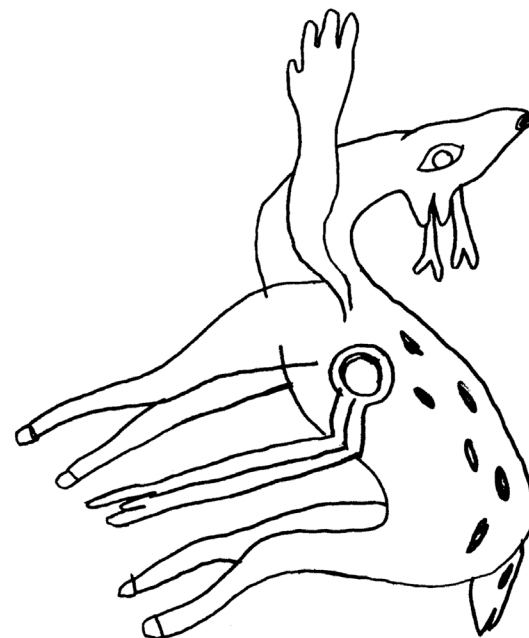
Surtout il s'agit d'une histoire d'hommes.

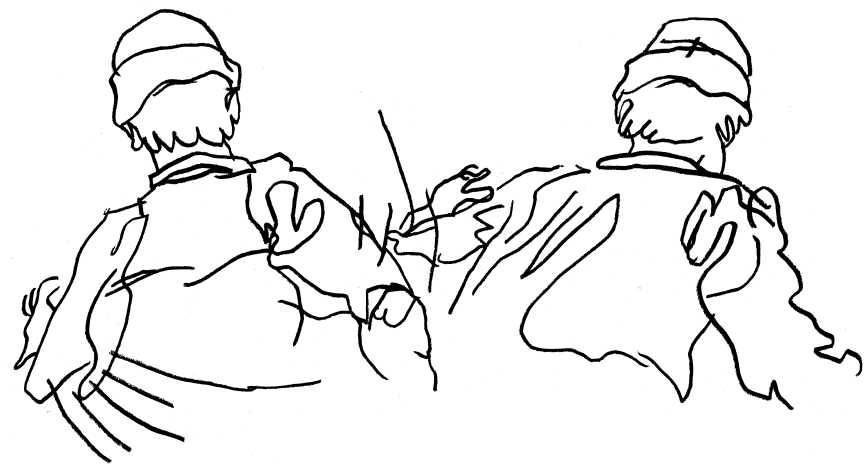
—dt.

ma chérie,



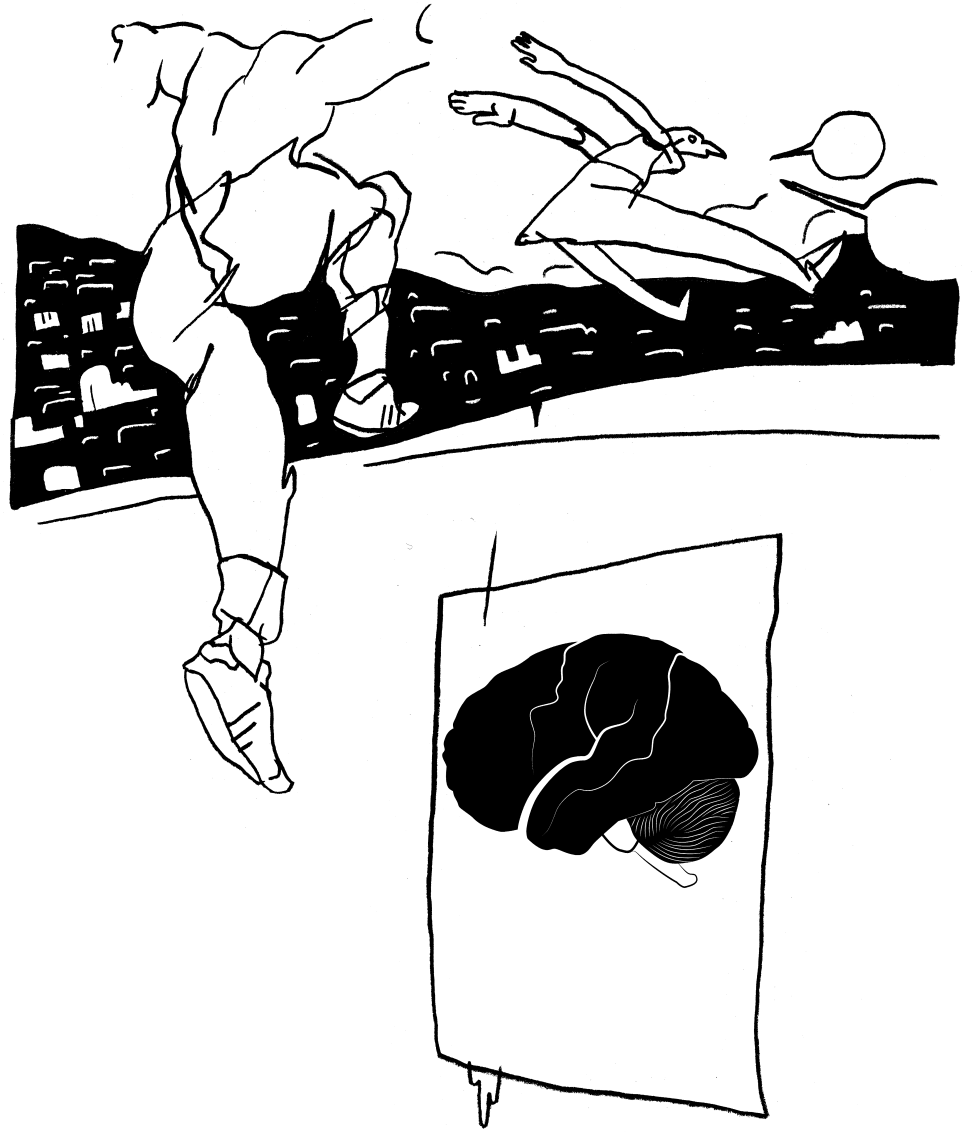
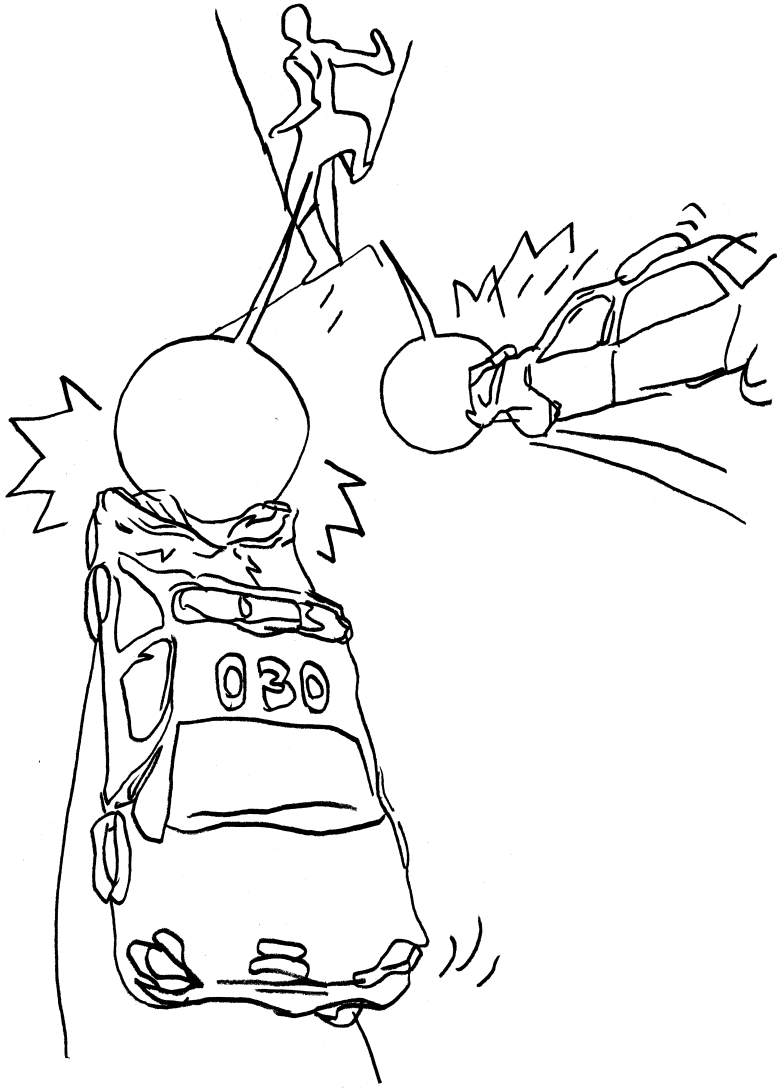
papa ne reviendra  
plus à la maison.











Je bois un café à  
Val-David



Il coûte 4,50\$ et c'est  
un latte dans lequel on a  
rajouté un



horrible  
shit  
à la  
vanille  
(sans le  
mentionner)

on arrive à un beau lac  
et Manel met sa main  
dans l'eau.



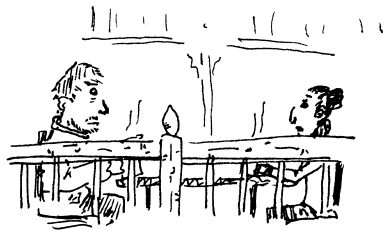
Ensuite on trouve un avertissement  
que le lac est plein de coliformes  
fécaux.



Moi et Manel faisons un  
retour en nature.



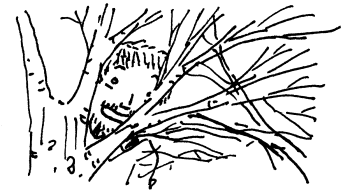
Qui consiste à boire du  
mauvais café à Val-David



Manel veut faire pipi dans  
le bois.



Moi j'essaie de voir ses  
fesses à travers les branches.



LES GENS PORTENT DES  
VÊTEMENTS FAITS POUR  
ALLER EN NATURE.



Ensuite on marche dans le  
parc linéaire. En fait c'est  
une longue trail de petites roches.

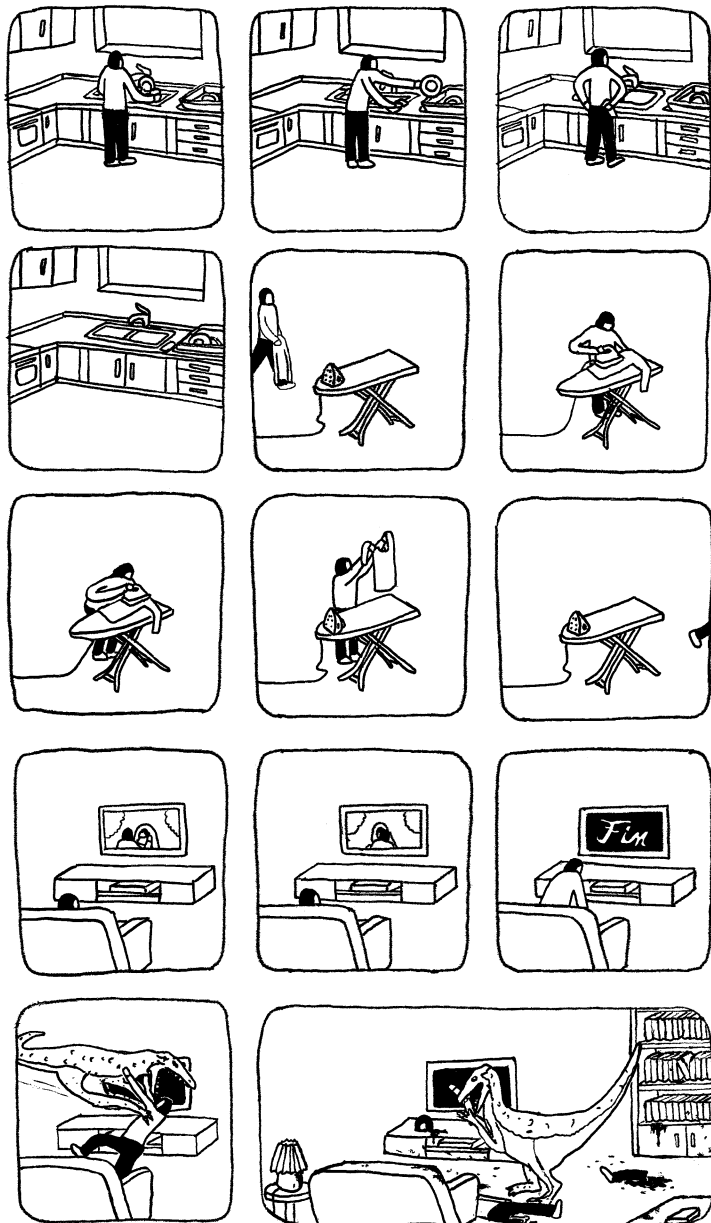


Manel m'a perçait soudain.



Elle me sourit. Ça y est,  
notre retour en nature est  
réussi.





ANTONIN BUISSON

LECTURES  
DE LA  
SEMAINE :

« PINKERTON »  
« APARTEMENT #3 »  
« GRIZZLY COMIX »  
« IN SITU 2 »  
« À PROPOS DE  
L'ÉDUCATION »

FRANK ♥  
ALEX  
PASCAL  
GIRARD  
PASCAL  
GIRARD  
SOPHIE  
YANOW

COLOSSE / EXPORT

COLOSSE / EXPORT

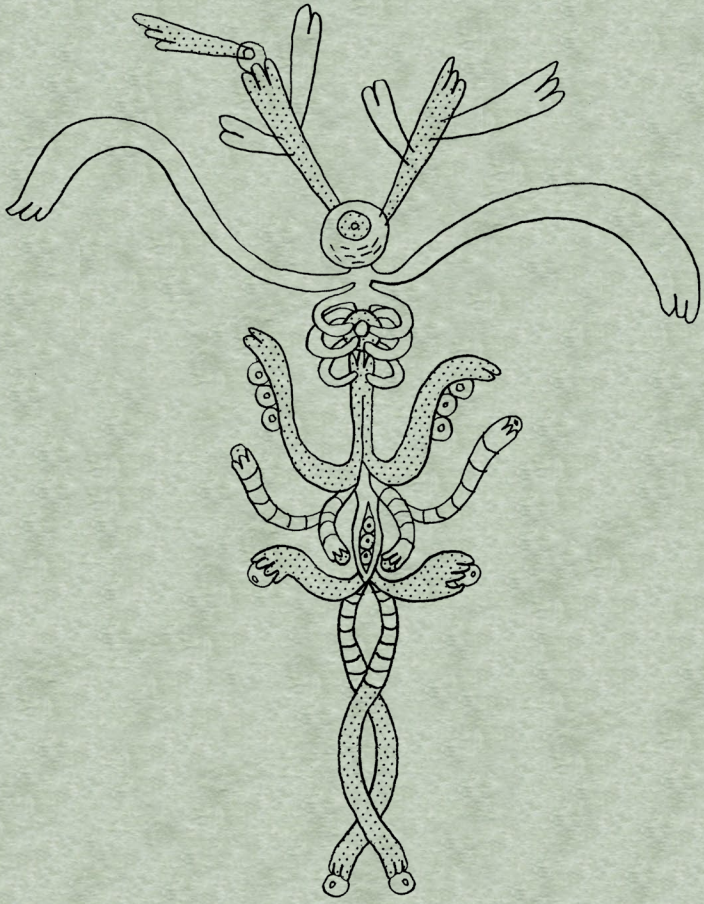
COLOSSE / EXPORT

ZVIANE



— J'ÉTAIS VRAIMENT ENDORMIE

— ÇA DEVRAIT ÊTRE ÇA QU'ILS MARQUENT  
SUR LES PLAQUES D'IMMATRICULATION.



deer Spirit