



L'Interactif

Le journal des étudiants du département d'Informatique et Recherche Opérationnelle de l'Université de Montréal
Vol 7

No 7

aujéitam seb eido



HAPPY
EASTER



L'Interactif

Volume 7 Numéro 7 — Mars 1989

L'Interactif est le journal des étudiants du département d'informatique et de Recherche Opérationnelle de l'Université de Montréal.

L'Interactif est publié le deuxième mercredi de chaque mois, et est distribué gratuitement aux étudiants du département d'IRO.

Rédacteur en Chef :

Sylvain Boulé

Mise en Page :

Serge Meynard

Publicité :

Patrick Chénard (2202)

Correcteur :

Philippe Marazzani

Chroniques :

Martin Dubuc (Macintosh)

Yves Gaudreau (Sondage) (1027)

Jean-Pierre Le-Boeuf

(Technologie) (1091)

Sylvain Verreault (Nouveautés) (1212)

Autres contributeurs:

Marc Bégin

Carl Bilodeau

Guy Pilon

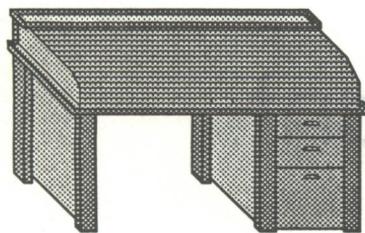
Jean-Pierre Léger

Table des matières

Éditorial	3
Message de l'AEIROUM	4
Technologie	5
Quelle vie!	8
Le Macintosh	9
Le mot du chialeux	15
Lettre ouverte	16
Guide Mi-Chemin	17
Nouveautés	18
Les langages objets	20
Les mouvements Madeau	23
Sondage	24

Toute correspondance générale ou soumission de textes ou illustrations devrait être envoyée par mail sur Apollo au u1027; la date de tombée est le premier mercredi de chaque mois.





Editorial

Le printemps qui s'en vient active les neurones de nos lecteurs qui, subitement, se sont transformés en écrivains; nous sommes envahis de textes! Par pitié cessez de nous envoyer du matériel à publier, nous ne fournissons plus!

Nous avons reçu trois textes pour le même numéro! Un du désormais célèbre Jean-Pierre Léger, un de Guy Pilon (excellent d'ailleurs) et, enfin, un de Carl Bilodeau (que je ne peux commenter à cause de sa teneur politique).

En plus, Yves Gaudreau, notre valeureux chroniqueur des sondages mensuels, s'étant dernièrement procuré un Mac à la grande joie de Serge Meynard notre biaisé responsable de la mise en page, s'est proposé pour la succession de notre Serge national pour ce poste. BRAVO!

Ensuite, deux valeureux de deuxième année se sont portés volontaires pour la noble, valorisante, et enrichissante tâche de rédacteur en chef du volume 8 de l'Interactif en 1989-90. Il s'agit de messieurs Yves Calloc'h et Ivan Cat-alin. FÉLICITATIONS, hip-hip, clap, clap...

Il ne reste plus qu'à trouver quelques chroniqueurs et le tour sera joué. Dépêchez-vous, les places sont limitées... L'Interactif est plus en forme que jamais et son avenir immédiat est assuré. Tous ces événements viennent couronner mon mandat dans la joie et le bonheur.

Nos chroniqueurs réguliers ont bien répondu ce mois-ci, en particulier Marc Bégin (Info-Mag) qui nous a offert une performance digne de Jean-Pierre Le-Boeuf ou de Sylvain Verreault (il a changé un peu la teneur de sa chronique).

Une ombre reste malheureusement au tableau: les finissants ne s'étant pas inscrits en maîtrise ne pourront pas lire régulièrement leur mensuel préféré (l'Interactif, voyons) l'an prochain!

Après des semaines de réflexion (à temps partiel), il m'est venu à l'idée de créer ce que l'on désigne tout honnêtement par le terme: abonnement. Ces abonnements seront inscrits une année à l'avance (mois d'avril 89 pour les 8 numéros de l'an prochain), le coût

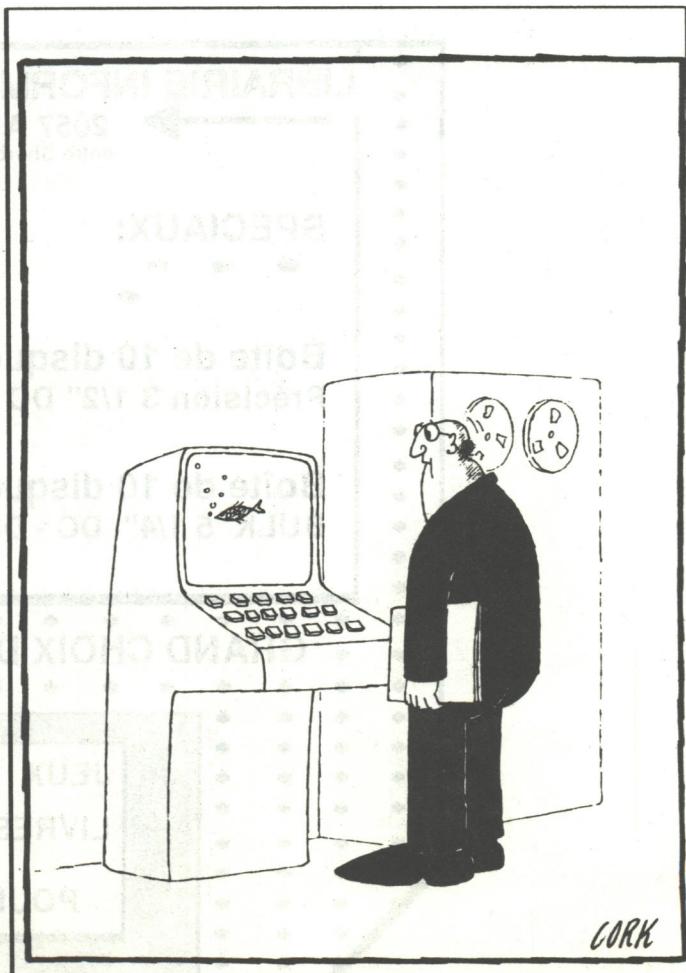
en est de \$15.00 (frais de port et de manutention) et la livraison se fera par le courrier.

Les futurs rédacteurs n'auront qu'à se trouver un brave qui s'occupera de poster les numéros à mesure qu'ils sortiront des presses (j'espère qu'ils seront d'accord...). Il y aura officiellement au moins un abonnement (le mien...). [Et rajoute le mien! -Serge]

En finissant, joyeuse pâque et pensez à votre ligne lorsque vous serez en admiration devant les nombreux étalages de chocolat, achetez plutôt du chocolat à vos confrères/consoeurs de troisième!

MMMMMMmmmmmmmm...

Sylvain Boulé
R.E.C.



Message important de l'AEIROUM

Je voudrais vous entretenir d'un problème qui est plus important qu'il n'en paraît. Voilà, cette session, le CAFIRO n'est ouvert que trois jours par semaine, soit le lundi, le mardi et le jeudi. Je sais ce que vous pensez déjà: une autre annonce achalante pour trouver des bénévoles! Mais vous devriez continuer à lire.

Au cas où vous ne seriez pas au courant, les locaux de l'aile U-500 n'appartiennent aux étudiants d'informatique que sous condition d'y tenir un café étudiant CINQ (5) jours par semaine. Donc, présentement, l'Université serait en droit de réclamer les locaux pour y aménager ce qu'elle veut. (Allez hop! Une peinture neuve à la poubelle.) Il ne resterait aucun endroit aux étudiants pour pouvoir étudier.

La solution, vous la connaissez: un peu de bonne volonté.

En passant, la responsable du CAFIRO termine cette année; il n'y aura donc plus personne pour s'en occuper l'an prochain.

Vous pouvez vous impliquer, ou faire semblant d'oublier comme avec toutes les autres choses qui se passent sur ce caillou minable. C'est tellement plus facile!

Normand Lamothe
V.P.E. AEIROUM

LIBRAIRIE INFORMATIQUE MICRO - INFO

2057 A St - Denis
entre Sherbrooke et Ontario

SPECIAUX:

Boîte de 10 disquettes= \$16.95
Précision 3 1/2" DC - DD

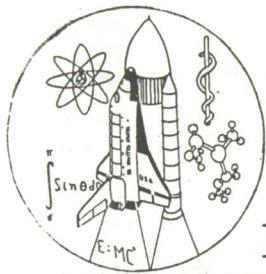
Boîte de 10 disquettes= \$ 4.45
BULK 5 1/4" DC - DD

GRAND CHOIX DE LOGICIELS

JEUX
LIVRES
POUR
LOGICIELS
ACCESSOIRES
MAC & IBM

Technologie

Jean-Pierre Le-Boeuf



Le Canada, puissance aérospatiale

Dès le début de l'aviation, au Canada, on s'intéressait déjà au développement de ce moyen de transport; notamment, le premier vol en territoire canadien eut lieu 6 ou 7 ans après le vol des frères Wright, en décembre 1903.

Pour la construction aéronautique, la Seconde Guerre Mondiale fut le point de départ d'une industrie aéronautique qui devint parmi les plus importantes au monde. Le Canada devint ce que le président Franklin D. Roosevelt avait dit, c'est à dire l'aérodrome des démocraties dans la lutte contre l'Allemagne. En effet, des firmes se mirent à fabriquer des avions sous licence par centaines et même par milliers pour les forces armées américaines, du Commonwealth, les forces françaises libres et l'Union Soviétique. Une de ces firmes, la Canadian Car & Foudries, fabriqua plusieurs types d'appareils de conception américaine et britannique.

Après la guerre, le Canada était devenu la quatrième puissance industrielle en aéronautique. C'est alors que trois firmes ont vu le jour; Canadair à Montréal, DeHavilland de Toronto et la AVRO Canada à Bradford en Ontario.

Au début des années cinquante, Canadair était déjà la firme la plus importante du pays; elle produisait plusieurs types d'appareils en sous-traitance, dont le meilleur chasseur de l'époque, le F-86 SABRE conçu aux Etats-Unis en 1949. Canadair en construisit plus d'un millier, qui ont servi pour les forces aériennes du Canada, de l'Allemagne Fédérale et du Pakistan. Elle fabriqua aussi l'avion d'entraînement T-33, lui aussi d'origine américaine, à plus de 300 exemplaires, et l'hydravion de patrouille et de lutte anti-incendie CANSO, extrapolé de l'hydravion de patrouille PBY CATALINA qui a acquis ses lettres de noblesse lors de la Deuxième Guerre. Canadair construisit déjà des avions qu'elle avait elle-même conçus, notamment l'avion de transport quadrimoteur CL-44 pour

le fret dont certains furent vendus aux U.S.A.

Il y avait aussi l'avion de patrouille anti-sous-marin ARGUS (plus de 15 exemplaires) et un petit appareil d'entraînement à réaction, le TUTOR (plus de 300 avions construits) qui servit au sein des forces canadiennes et malaises. Cet avion est encore en service, c'est sur cet appareil que les Snowbirds exécutent leurs acrobaties aériennes.

À la même époque DeHavilland connut un succès foudroyant avec son petit avion de brousse, le BEAVER. Construit à plusieurs centaines d'exemplaires, il servit à travers le monde entier. Il fut très apprécié des forces armées canadiennes, britanniques, des forces du Sultanat d'Oman, de l'armée américaine et des entreprises de transport aérien régional. DeHavilland avait alors joué un vilain tour au géant Cessna dans le domaine de l'aviation légère et de touriste. Quelques Beaver servent encore, dont un qui est exploité par les Ailes de l'espérance au Pérou. DeHavilland produisit aussi le successeur du Beaver, le Otter, qui dériva plus tard (début des années 70) au Turbo-Otter et au Twin-Otter muni de 2 turbopropulseurs. Ce dernier fut vendu à plusieurs compagnies aériennes du Canada, des États-Unis, d'Europe et d'Australie. Le Twin-Otter servit aussi dans les forces aériennes du Canada, de l'Ethiopie et du Chili. Dans les années soixante DeHavilland devint le chef de file mondial dans le domaine des avions à décollages et atterrissages courts (A.D.A.C.). Le premier de cette catégorie fut le DHC-5 Caribou, qui servit autant dans le domaine civil que celui du transport militaire (forces aériennes américaines, australiennes et de certains pays africains). Il fut suivi du DHC-6 Buffalo et du fameux DHC-7 DASH 7 qui connut un grand succès commercial au Canada et aux États-Unis. Aujourd'hui DeHavilland, société du gouvernement fédéral jusqu'en 1985 alors qu'elle devint filiale de Boeing, ne produit que le DASH-8 qui connaît des difficultés commerciales.

Mais la plus spectaculaire des firmes canadiennes d'aéronautique a été AVRO du Canada. Avant 1950, AVRO se lança dans un pro-

jet très ambitieux, celui de mettre au point le premier avion commercial à réaction en Amérique du Nord. Cet appareil appelé JETLINER possédait 4 moteurs à jet. Il fut le premier avion à jet commercial à avoir transporté du colis en Amérique du Nord. Cela a eu l'effet d'une bombe aux Etats-Unis; un journal new-yorkais écrivait à ce sujet en première page que les U.S.A. ne possédaient pas le monopole de l'industrie aéronautique. Le JETLINER aurait connu un très grand succès car son seul concurrent, le COMET britannique fut retiré de la circulation à la suite de nombreux accidents. Le JETLINER aurait été alors le seul avion de sa catégorie sur le marché jusqu'à l'apparition du Boeing 707 en 1954. Mais en 1950 survint la guerre de Corée, le gouvernement somma AVRO de se consacrer uniquement à la fabrication de l'intercepteur CF-100 et le JETLINER fut relégué aux oubliettes. Le CF-100 (appelé Canuck) fut le premier et le dernier chasseur de conception canadienne. C'était un monoplace propulsé par 2 turboréacteurs aussi de conception canadienne. Il fut le fer de lance de la chasse jusqu'au milieu des années 1960. Certains furent vendus à la Belgique et entre 200 et 400 exemplaires furent produits.

Au milieu des années 50, on songea à trouver un successeur au Canuck. AVRO fut chargée de cette tâche. Elle conçut alors un appareil tout à fait révolutionnaire, le CF-105 ARROW. C'était un véritable bolide. Il était propulsé par 2 turboréacteurs qui lui permettait des vitesses de l'ordre de 2 fois la vitesse du son. Ces moteurs (ORENDA de conception canadienne) étaient si puissants que l'on disait qu'ils ne pouvaient être poussés à plus de 75% du régime maximum car l'avion se serait désintégré en vol. Le ARROW avait une aile delta en position haute. Il fut aussi muni d'un système de tir entièrement automatisé (le système ASTRA de conception canadienne) pour le guidage des missiles Sparrow. Le premier vol fut un brillant succès. Ce vol fut en lui même un exploit car aucun prototype n'avait été construit pour tester les composantes; on avait testé cet avion sans que tout le fonctionnement des systèmes ait été

vérifié au préalable.

On avait tenté de le vendre aux États-Unis mais en vain, car ce pays développait déjà plus de 5 types de chasseurs, dont le F-4 PHANTOM II. N'ayant pas essayé de passer par le gouvernement américain pour promouvoir la vente de cet appareil auprès des pays de l'OTAN (ce qui aurait fonctionné), le gouvernement canadien se voyait alors seul à payer pour le développement de ce superbe chasseur. De surcroît, les missiles nucléaires avaient remplacé les bombardiers et l'idée d'un intercepteur était devenu caduque; le ARROW fut abandonné. Du moins c'était la version officielle du gouvernement, puisque par la suite le même gouvernement à fait l'acquisition de 66 chasseurs F-101 usagés provenant des surplus des forces américaines (de moindre qualité que le CF-105) en plus de missiles anti-aériens BOMARC guère plus efficaces et cela presque qu'au même prix que l'acquisition de quelques dizaines de ARROW. En plus, la Grande-Bretagne avait demandé quelques CF-105 pour évaluation dans le cadre d'acquisition d'un intercepteur. La demande a été ignorée par le gouvernement. Mais plus grave, ce gouvernement de Diefenbaker signa avec le gouvernement américain que dorénavant le Canada demanderait ni plus ni moins la permission au gouvernement américain pour la mise au point d'avions de chasse au Canada.

Par la suite, non seulement les six exemplaires du ARROW furent démolis comme de la vulgaire ferraille, mais la firme AVRO ferma aussi ses portes. Les ingénieurs ayant conçu le ARROW émigrèrent aux États-Unis pour travailler à la NASA. Le chef ingénieur fut un des importants concepteurs du module lunaire qui amena les astronautes sur la Lune.

Plus réjouissant cependant est le fait que le Canada fut le troisième pays à posséder un satellite en orbite; il s'agissait du satellite ALOUETTE. Aujourd'hui le Canada est le leader mondial dans le domaine des télécommunications devançant les États-Unis et le Japon. Le développement des télécommunications passe par la technologie des satellites.

leiv elleuQ

éduca novyc

En effet, nous retrouvons la firme SPAR Aérospatiale qui s'est fait surtout connaître pour sa conception du bras télémanipulateur équipant les navettes spatiales américaines.

Le Canada vient de se doter d'une agence spatiale lui permettant d'unifier sous une même administration tout le développement des technologies spatiales qui jusqu'à maintenant était dispersée parmi une dizaine de ministères fédéraux. Cette agence aura, entre autres, à s'occuper du projet de station orbitale mise en oeuvre par la NASA, avec la collaboration de l'agence spatiale européenne (E.S.A), du Japon et du Canada. Ce projet sera entrepris sur une période de 30 ans. L'agence aura aussi à s'occuper de projets de moindre envergure en coopération avec d'autres nations, comme le projet d'étude de la haute atmosphère en région arctique avec l'Union Soviétique, et de projets purement nationaux comme le programme des astronautes canadiens.

L'industrie aérospatiale du Canada est aujourd'hui au sixième rang (après les U.S.A., l'U.R.S.S., la France, la Grande-Bretagne et l'Allemagne Fédérale). Cette industrie est très active. Canadair connaît maintenant du succès avec son avion d'affaires CHALLENGER; l'avion citerne CL-215T pourrait être exporté pour remplacer les CL-215 plus anciens. Cette même firme a reçu un important contrat de sous-traitance du consortium européen Airbus pour la fabrication de pièces importantes pour les nouveaux Airbus A-330 et A-340, en plus du contrat d'entretien des CF-18 canadiens et de la fabrication d'environ 400 systèmes CL-284 pour la France et l'Allemagne Fédérale, un missile récupérable de reconnaissance. Il faut ajouter aussi la fabrication de pièces pour tous les F-18 produits par McDonnell Douglas.

Il y a Pratt & Whitney de Longueuil qui produit des moteurs d'hélicoptères et d'avions qui connaissent un succès mondial, Marconi du Canada qui est un important fournisseur de systèmes avioniques et de guidage, CAE Electronique qui est de loin le leader mondial pour les simulateurs de vol, en plus de systèmes avioniques très avangardistes

comme un casque muni d'une visière sur laquelle est affiché tout le tableau de bord. Sans parler de Bell Hélicoptère, de Rolls-Royce qui fabrique des pièces pour moteur d'avions, d'Oerlikon et de Virtual Prototypes, firme qui développe des logiciels de CAO pour l'industrie aéronautique, qui se fait connaître au Canada et en Europe. Toutes les firmes énumérées sont situées dans la région de Montréal sauf SPAR qui possède une division robotique à Toronto; cependant, sa division satellite se trouve à Ste-Anne de Bellevue sur l'île de Montréal. En Colombie-Britanique, on retrouve CONAIR qui convertit des anciens avions de patrouille anti-sous-marin en appareils de lutte anti-incendie. À Winnipeg on retrouve Litton qui fabrique des pièces sous licence.

La recherche commence à se développer. Canadair et l'École Polytechnique viennent de former un groupe pour conduire des projets de recherche conjoints. À Ottawa, un groupe de recherche est à mettre au point un avion mû par micro-ondes. Le principe est qu'une antenne terrestre émettrice transmet des micro-ondes à un petit avion qui décrit un trajet circulaire au-dessus de l'antenne sur un grand rayon. L'avion capte les ondes par des récepteurs disposés sous les ailes et ces ondes sont transformées en énergie électrique alimentant un petit moteur actionnant une hélice. La principale application serait de munir l'avion d'une antenne émettrice-réceptrice. Dès lors l'avion volant à haute altitude servirait de relais pour les communications radio dans les régions éloignées. Cet appareil ne nécessiterait aucun ravitaillement sauf une source micro-onde lui permettant de demeurer indéfiniment en vol. Ce système serait beaucoup moins dispendieux qu'un satellite.

Avec tout ce potentiel, le Canada demeurera très fort dans le domaine aérospatial en autant qu'on ne fasse d'autres gaffes comme ce fut le cas avec le ARROW et le JETLINER.

Jean-Pierre Le-Boeuf



Quelle vie!

Sylvain Boulé

C'est sous des conditions climatiques intolérables que je sortis de chez moi par la grande porte transparente ce matin là; il pleuvait. Un temps de chien.

Juste le temps de faire cette constatation et l'humain qui sent bizarre me ferma la porte au nez, me laissant poireauter sous l'averse comme un imbécile.

Voulant me protéger de la pluie au maximum, je me précipitai sous le perron. J'y rencontraï Tomasina la belle chatte du voisin qui, cherchant un abri, était venue s'y réfugier.

Il y avait plusieurs bestioles sous le perron ce matin-là; sans doute se protégeaient-elles de la pluie elles aussi. Quel régâl nous avons fait. Bien meilleur que la fade nourriture en boîte sur laquelle il y a la photo d'un chat mongolien...

L'univers du dessous de perron ayant été exploré dans tous les sens, Tomasina n'étant pas en période propice (elle a repoussé à plusieurs reprises mes avances), je décidai de rentrer à la maison.

Quel idiot cet humain à l'intérieur de la maison, me regardant d'un air nul au travers de cette porte transparente, tout en transformant son visage ou en émettant des sons. J'avais beau crier: "Viens donc m'ouvrir la porte pauvre connard!". Il faisait mine de ne rien comprendre.

Après maints appels et gestes démonstratifs de mon intention d'entrer, il vint finalement m'ouvrir la porte transparente. Je pris bien mon temps pour entrer, juste pour l'emmerder un peu car il attendait que je sois à l'intérieur pour refermer la porte.

Je me frottai quelque peu sur ses jambes pour lui faire croire que j'avais de l'affection pour lui et le féliciter d'avoir finalement obéi à ma demande.

L'intérieur de la maison recelait des odeurs intolérables. Souvent j'avais tenté d'asperger les pires endroits de mon urine mais le sale humain (jamais je ne l'avais vu se lècher le corps) m'avait frappé! Si on ne peut plus rendre service...

Je décidai de sortir par la porte opaque à l'autre extrémité de la maison lorsque je vis l'humain s'y diriger. Ne m'ayant pas vu, cet imbécile me ferma la porte sur le corps avec violence. Quelle douleur!

L'humain comme pour s'excuser, appliqua sa bouche sur ma tête; sans lècher, comme toujours... Il me caressa le dos de ses grandes pattes moites dépourvues de poils, ce qui n'atténua en rien la vive douleur. Je fis semblant de rien, boitant à peine.

Quelle ne fut pas ma surprise lorsque je constatai qu'il pleuvait ici aussi! En plus, de ce côté il n'y avait pratiquement pas d'abris. L'humain était parti dans son gigantesque véhicule meurtrier, la porte était fermée. Je fus donc contraint, malgré la douleur qui persistait, de sauter sur le bord de la fenêtre et d'attendre le retour de l'humain.

Ce que je regrettai le perron de la porte vitrée! Impossible d'y aller, deux portes m'en empêchaient. Heureusement j'aperçus quelques oiseaux, reniflai plusieurs odeurs dont quelques bonnes comme celle d'une chatte en bonne période; mais me faire mouiller pour une chatte... Qu'elle vienne, elle.

Le paysage s'était assombri lorsque l'humain revint enfin; ce qu'ils peuvent être lents ces humains. J'étais trempé jusqu'aux os malgré l'abri provisoire sur lequel j'avais pris place. Quelle sensation désagréable de se sentir humide sur tout le corps.

L'humain m'emmêla les poils avec un grand chiffon qui dégageait une odeur insoutenable qui se retrouva sur mon corps. J'entrepris aussitôt de me laver entièrement.

Après la besogne accomplie, j'entrepris de manger. Je me frottai sur la jambe de l'humain tout en demandant mon dû. Après une longue période, pendant laquelle il faisait semblant de ne pas me voir ni m'entendre, il me servit ma bouffe. Encore cette photo du chat idiot! Simili poisson cette fois.

Par instinct de survie, je dévorai à belles dents cette purée pour chaton mâchée d'avance, ensuite je m'étendis sur mon fauteuil préféré pour ma toilette du soir tout en observant, mine de rien, les déplacements de l'humain.

S'étant enfin arrêté de bouger, l'humain me fit signe de venir sur ses pattes du bas. Quel plaisir je ressentais lors de ces rares soirées où couché sur lui je me détenais totalement sous les caresses de ses grandes pattes dépourvues de poils. Mais qu'il vienne me chercher, je ne me déplacerai pas de mon fauteuil pour lui...

En effet il vint me chercher et entreprit de me caresser jusqu'au moment d'aller s'enfermer dans une pièce et de me laisser seul dans le noir. N'ayant pas pu dormir de la journée, je décidai de me reposer après toutes ces émotions...

Le Continuum en délire!

ordinateur XT : 1 lecteur disquettes, 1 disque dur; 30 MB, 640 K.

Ecran monochrome, clavier carte ATI, imprimante LX 800 + logiciels neufs.

Appelez 345-9158. (23)

Oradent : poudre dentaire naturelle pour l'entretien des gencives et contre la mauvaise haleine. Satisfaction garantie ou argent remis. 40g. : 6.00\$, 6x40g. : 30.00\$. Bioscan Inc., St-Eustache. Tél : 491-5807 ou 491-5037. (26)

À sous-lo-

Contre la bélugivite?



Le Macintosh

Martin Dubuc

Dans la présente chronique, nous verrons une partie de l'implantation de l'interface. Nous avons vu la fois dernière le squelette d'un programme Pascal qui faisait appel à l'interface. Je continuerai donc à batir autour de cette ébauche en expliquant certaines parties du programme du mois, un mini-paint.

Les "managers" de l'interface

Macintosh

L'ensemble des routines du Toolbox a été divisé en plusieurs sections, chacune ayant sa propre fonction logique. Il existe diverses relations entre celles-ci. Par exemple, la partie graphique de l'interface, appelée Quickdraw, est à la base de plusieurs autres: Menu, Window, Dialog et Control "Manager".

Au début de chaque programme, il faut initialiser chacune de ces sections. La suite d'instructions qu'on retrouve dans la procédure Initialisations doit être exécutée.

L'initialisation des diverses sections du Toolbox permet l'allocation des tampons et espaces-mémoire nécessaires pour l'utilisation des ressources désirées. Pour le QuickDraw, entre autre chose, certaines variables globales sont mises à la disposition du programmeur: **thePort** (descripteur du port actif de dessin), **arrow** (le curseur en forme de flèche), **screenBits** (une structure de type BitMap pointant sur l'écran), **randSeed** (le germe utilisé dans la génération de nombres aléatoires) et quelques "patterns" communs. Fait intéressant: comme paramètre d'**InitDialogs**, on passe l'adresse de la procédure à exécuter lors d'une erreur de système. Il est donc possible de définir une procédure Crash et faire **InitDialogs(@Crash)** afin de permettre de revenir au Finder dans le cas d'une erreur relevée par le système d'exploitation.

Le traitement des événements

Comme expliqué dans le précédent article, le cœur d'un programme suivant le standard

est constitué d'une boucle d'attente d'événements. À chaque fois que le système d'exploitation (SE) reçoit un événement, il le met dans une file d'attente. Lors d'un appel à **GetNextEvent** (ou **WaitNextEvent** pour compatibilité avec MultiFinder), le SE nous renvoie le premier événement non-traité, suivant certaines priorités (les activateEvt ont plus de priorités).

Ici, pour plus de clarté, on conçoit une routine de traitement pour chacun des types d'événement rencontrés. On procédera donc selon le champ "what" de l'événement retourné par **GetNextEvent**:

mouseDown:

Le bouton de la souris a été actionné. On regarde où l'action s'est produite. Il peut s'agir d'une sélection de menu, du mouvement d'une fenêtre (drag), d'un clic dans le "close box" d'une fenêtre, d'un clic dans une fenêtre système (un "desk accessory" par exemple) ou d'un clic dans la fenêtre du programme. Pour toute ces actions, exceptée la dernière, il existe des routines de traitement. Par exemple, **MenuSelect(Event.where)** retourne un LongInt qui contient dans sa partie haute la colonne de menu sélectionnée et dans sa partie basse, le numéro de l'item concerné. Lorsqu'il s'agit d'une fenêtre d'application (**inContent**), il faut définir les étapes de traitement. Ici, j'appelle la routine **DrawObject** avec comme paramètre le type d'objet à dessiner (défini selon le modificateur sélectionné).

keyDown:

Un caractère a été tapé. Si la touche commande (%) est active, on sélectionne l'item de menu correspondant à ce caractère, sinon, on traite le caractère selon l'application. S'il s'agit d'un traitement de texte par exemple, on le dessine à l'écran à l'endroit du curseur.

updateEvt:

Un rafraîchissement d'écran est demandé. Le plus souvent, il s'agit d'une fenêtre qui devient active et à ce moment son contenu doit être mis à jour. Ici, la variable **curBitMap**

Le Macintosh

Windows Duplex



contient l'image de la fenêtre et tout ce qui reste à faire est de copier son contenu dans la fenêtre de dessin.

activateEvt:

Une fenêtre devient active ou inactive. Lorsqu'une fenêtre devient inactive, on doit effacer les "scroll bars" et le "size box", désactiver les sélections de texte et certains menus nécessaires uniquement dans la fenêtre concernée. Lorsqu'une fenêtre redevient active, on met à jour ces divers objets. Comme notre application ne contient aucun de ceux-ci, nous n'en tenons pas compte.

Les routines de dessins

La majorité des routines de QuickDraw sont exécutées dans le port actif de dessin sur lequel pointe la variable globale thePort. Ces routines utilisent en général des coordonnées locales au port courant (qui dans notre programme se trouve à être theWindow). Une des routines utilisée ici, CopyBits, travaille sur l'ensemble de l'écran et donc utilise le système de coordonnées globales. CopyBits est utilisée pour faire le transfert d'informations entre deux variables de type BitMap. Cette routine est très rapide. Normalement, lorsque l'on fait fréquem-

ment du dessin à l'écran, on dessine d'abord dans un BitMap en mémoire, puis on copie la région transformée sur l'écran. Ceci permet d'éviter le "flicker" qui se produit lorsque l'on dessine plusieurs objets régulièrement à l'écran.

Ceci termine donc notre survol de l'implantation de l'interface. On peut donc remarquer que cette interface, qui paraît au départ compliquée, est facilement programmable grâce aux outils fournis par le biais des ROM du Macintosh. De plus, une fois qu'un programme est terminé, il est facile d'intégrer cette partie d'interface dans les futurs projets, la seule chose restant à changer étant les routines de traitement propres à l'application.

Une dernière chose reste à remarquer: bien que les standards soient établis, il existe des possibilités d'expansion de l'interface et c'est ce qui fait la beauté de celle-ci. Ainsi, Apple et ses développeurs ont pu introduire des nouveaux concepts suivant la philosophie de l'interface-usager: menus hiérarchique, pop-up menus, tear-off menus, etc... qui facilitent l'utilisation de la machine (lorsque bien utilisés).

PROGRAMMEUR(EUSE) D'APPLICATION en langage BASIC recherché(e)

Doit posséder un peu d'expérience en programmation et des connaissances en comptabilité.

Travail sur la rive-sud, à temps plein dans le domaine de l'alimentation.

Envoyer votre curriculum vitae à:

SCI MD Ltée

1550 Ampère #411

Boucherville, Qué.

J4B 7L4

Interactif — Vol 7 No 7 — Page 11

```

PROGRAM Draw;
{$I-}
CONST
  MainLevee = 0;
  LigneDroite = 1;
  Rectangle = 2;
  Oval = 3;

VAR
  theWindow : WindowPtr;
  dragRect : Rect;
  AppleMenu, FileMenu, EditMenu : MenuHandle;
  done : Boolean;
  oldBitMap, curBitMap : BitMap;

PROCEDURE RestoreImage (theBitMap : BitMap);
{ Copie l'image contenue dans theBitMap à l'écran }
BEGIN
  CopyBits(theBitMap, thePort^.portBits,
            thePort^.portBits.bounds,
            thePort^.portBits.bounds, srcCopy, NIL)
END; { RestoreImage }

PROCEDURE SaveImage (theBitMap : BitMap);
{ Copie l'image à l'écran dans theBitMap }
BEGIN
  CopyBits(thePort^.portBits, theBitMap,
            thePort^.portBits.bounds,
            thePort^.portBits.bounds, srcCopy, NIL)
END; { SaveImage }

PROCEDURE DrawObject (where : Point;
                      typeOfObject : Integer);
{ Dessine un objet à l'écran de type typeOfObject }
VAR
  r : Rect;
  t : LongInt;
  oldPos, dest : Point;

BEGIN
  SaveImage(oldBitMap);
  r.topLeft := where;
  GetMouse(oldPos);
  MoveTo(oldPos.h, oldPos.v);
  r.botRight := oldPos;
  WHILE StillDown DO
    BEGIN
      GetMouse(dest);
{ Mettre dans dest la position courante du curseur }
      IF (oldPos.h <> dest.h) OR
          (oldPos.v <> dest.v) THEN
        BEGIN
          IF typeOfObject <> MainLevee THEN
            BEGIN
{ Redessine l'ancienne image sur l'écran }
              RestoreImage(oldBitMap);
              oldPos := dest;
{ Définition des coordonnées du rectangle }
              IF (dest.h < where.h) THEN
                BEGIN
                  r.left := dest.h;
                  r.right := where.h
                END
              ELSE
                BEGIN
                  r.left := where.h;
                  r.right := dest.h
                END;
{ Synchronisation pour dessin à l'écran }
{ (attente d'un tick) }
                WHILE t = TickCount DO
                  ;
{ Dessine l'objet en nouvelle position }
                CASE typeOfObject OF
                  MainLevee :
                    LineTo(dest.h, dest.v);
                  LigneDroite :
                    BEGIN
                      MoveTo(where.h, where.v);
                      LineTo(dest.h, dest.v);
                    END;
                  Rectangle :
                    FrameRect(r);
                  Oval :
                    FrameOval(r);
                END;
                t := TickCount
              END;
            END;
          SaveImage(curBitMap)
        END; { DrawObject }

PROCEDURE Crash;
BEGIN
  ExitToShell;
END; { Crash }

PROCEDURE ProcessMenu (mResult : LongInt);
{ Traitement des événements de la barre de menu }
VAR
  item, menu, dumb : Integer;
  name : Str255;

BEGIN
  item := LoWord(mResult);
{ L'item de menu sélectionné se retrouve }
{ dans la partie basse du code mResult, }
  menu := HiWord(mResult);
{ la colonne de menu dans la partie haute. }
CASE menu OF
  128 : { Pomme: sélectionner l'accessoire de }
         { bureau correspondant }
    BEGIN
      GetItem(AppleMenu, item, name);
      dumb := OpenDeskAcc(name);
      SetPort(theWindow);
    END;
  129 :
{ La colonne de menu Fenêtre a été choisie }

```

```

CASE item OF
  1 : ShowWindow(theWindow);
  2 : HideWindow(theWindow);
  3 : ; { Cet item n'est qu'un délimiteur }
  4 :
    done := True;
    { L'item Quit a été sélectionné }
END;

130 : { Menu Edit: appel à SystemEdit pour le }
      { traitement d'un DA }
IF NOT SystemEdit(item - 1) THEN
  CASE item OF
    1 : { Undo }
      RestoreImage(oldBitmap);
    OTHERWISE
      END; { CASE }
    OTHERWISE
      END; { CASE }
  HilitMenuItem(0)
  { On fait inverser le menu sélectionné }
END; { ProcessMenu }

PROCEDURE DealWithMouseDown(Event : EventRecord);
{ Traitement des événements de "clics" de souris }
VAR
  whichWindow : WindowPtr;
BEGIN
  CASE Event.modifiers OF
    0 : ; { Pas de touche }
    1 : ; { Tous les autres }
    2 : ; { Option }
    4 : ; { Shift }
    8 : ; { Option + Shift }
    16 : ; { Option + Ctrl }
    32 : ; { Shift + Ctrl }
    64 : ; { Option + Shift + Ctrl }
    128 : ; { Tous les autres }
    OTHERWISE
      END;
  END;
  CASE Event.where OF
    inSysWindow :
      SystemClick(Event, whichWindow);
    inMenuBar :
      ProcessMenu(MenuSelect(Event.where));
    inDrag :
      DragWindow(whichWindow, Event.where,
                 dragRect);
    inContent :
      { L'usager a cliqué à l'intérieur de la fenêtre }
      IF whichWindow <> FrontWindow THEN
        SelectWindow(whichWindow)
      ELSE
        BEGIN { Faire le traitement approprié }
        { Transférer le champ Event.where en coordonnées }
        { locales à la fenêtre courante }
        GlobalToLocal(Event.where);
        IF BitAnd(Event.modifiers, cmdKey) <> 0
        THEN
          { La touche commande est sélectionnée }
          DrawObject(Event.where, Rectangle)
        ELSE IF BitAnd(Event.modifiers, shiftKey)
               <> 0 THEN
          { La touche shift a été sélectionnée }
          DrawObject(Event.where, Oval)
        ELSE IF BitAnd(Event.modifiers, optionKey)
               <> 0 THEN
          { La touche option a été sélectionnée }
          DrawObject(Event.where, LigneDroite)
        ELSE
          { Aucune clé "spéciale" n'a été touchée }
        END;
  END;
END;

```



```

        DrawObject(Event.where, MainLevee);
    END;
inGoAway :
{ On a cliqué dans le "close box" de la fenêtre }
    HideWindow(whichWindow);
    OTHERWISE
END;
END; { DealWithMouseDown }

PROCEDURE DealWithKeyDown (Event : EventRecord);
{ Traitement des événements de touches de clavier }
VAR
    car : Char; { Le caractère tapé }

BEGIN
    car := Chr(BitAnd(Event.message, charCodeMask));
    IF BitAnd(Event.modifiers, cmdKey) <> 0 THEN
{ La touche commande a été sélectionnée : on doit }
{ traiter l'item de menu correspondant }
        ProcessMenu(MenuKey(car))
    ELSE { Traiter le cas d'un caractère tapé dans }
        { une fenêtre }
    END; { DealWithKeyDown }

PROCEDURE DealWithUpdates (Event : EventRecord);
{ Traitement des événements de rafraîchissement }
{ d'écran }
VAR
    oldPort : GrafPtr;

BEGIN
{ WindowPtr(Event.message) pointe sur la fenêtre }
{ à modifier }
    BeginUpdate(WindowPtr(Event.message));
    EraseRect(thePort^.portRect);
{ Redessiner ici ce qui a été effacé dans la }
{ fenêtre correspondante }
    GetPort(oldPort);
    SetPort(theWindow);
    RestoreImage(curBitMap);
    SetPort(oldPort);
    EndUpdate(WindowPtr(Event.message));
END; { DealWithUpdates }

PROCEDURE MainEventLoop;
{ Boucle d'attente et traitement d'événements }
VAR
    Event : EventRecord;

BEGIN
    done := False;
    REPEAT
        SystemTask;
        IF GetNextEvent(everyEvent, Event) THEN
            CASE Event.what OF
                mouseDown :
                    DealWithMouseDown(Event);
                keyDown, autoKey :
                    DealWithKeyDown(Event);
                updateEvt :
                    DealWithUpdates(Event);
                activateEvt :
                    BEGIN
                    END;
                OTHERWISE
                    END { CASE }
                    UNTIL done;
    END; { MainEventLoop }

PROCEDURE SetUpWindow;
{ Définition de la fenêtre du programme }
VAR
    refCon : LongInt;
    r : Rect; { Rectangle délimitant la fenêtre }
    visible, goAwayFlag : Boolean;
    behind : WindowPtr;
    Title : Str255;

BEGIN
    SetRect(r, 40, 42, 472, 302);
    title := 'Draw Demo'; { Titre de la fenêtre }
    visible := True; { La fenêtre est visible }
    goAwayFlag := True;
{ La fenêtre possède un "close box" }
    behind := Pointer(-1);
{ La fenêtre se retrouve en avant des autres }
    theWindow := NewWindow(NIL, r, title, visible,
                           rDocProc, behind,
                           goAwayFlag, refCon);
    SetPort(theWindow);
{ Assigner theWindow comme grafPort courant. }
{ Toutes les routines de dessin du QuickDraw }
{ apparaîtront dans cette fenêtre }
    WITH ScreenBits.bounds DO
{ Préparer la zone de "dragging" permissible }
        SetRect(dragRect, 4, 24, right-4, bottom-4);
END; { SetUpWindow }

PROCEDURE SetUpMenus;
{ Définition de la barre de menu du programme }
VAR
    str : Str255;

BEGIN
{ Préparer la barre de menus. Note: AppleMenu, }
{ EditMenu,... sont des variables globales au }
{ programme, de type MenuHandle }
    str := ' ';
    str[1] := Chr(appleMark);
    AppleMenu := NewMenu(128, str);
{ Rendre les accessoires de bureau disponibles }
    AddResMenu(AppleMenu, 'DRVR');
    InsertMenu(AppleMenu, 0);
    FileMenu := NewMenu(129, 'Fenêtre');
    AppendMenu(FileMenu,
               'Ouvrir;Fermer;(-;Quitter/Q');
    InsertMenu(FileMenu, 0);
    EditMenu := NewMenu(130, 'Edition');
    AppendMenu(EditMenu,
               'Annuler/Z;(-;Couper/X;Copier/C;Coller/V');
    InsertMenu(EditMenu, 0);
    DrawMenuBar; { Dessiner la barre de menus }
END; { SetUpMenus }

PROCEDURE Initialisations;
BEGIN
{ NB: Pour les utilisateurs de LightSpeed Pascal, }
{ les six premières initialisations sont générées }
{ automatiquement par le compilateur. }

```

```

InitGraf(@thePort);
{ Initialisation du QuickDraw }
  InitFonts;
{ Initialisation du Font Manager }
  InitWindows;
{ Initialisation du Window Manager }
  InitMenus;
{ Initialisation du Menu Manager }
  TEInit;
{ Initialisation de Text Edit }
  InitDialogs(@Crash);
{ Initialisation du Dialog Manager }
  InitCursor;
{ Initialisation du curseur: mis en flèche }
  FlushEvents(everyEvent, 0);
{ Appel au SE pour initialiser les queues }
{ d'événements }
{ Allocation de mémoire pour le programme (128K) }
  SetApplLimit(Ptr(Ord4(ApplicZone) + 128000));
  MaxApplZone;
  SetUpMenus;
{ Procédure usager de création des menus }
  SetUpWindow;
{ Procédure usager de création des fenêtres }
{ Définition d'un BitMap qui contiendra l'image }
{ avant l'ajout d'un nouvel objet }
  oldBitMap.rowBytes := thePort^.portBits.rowBytes;
  oldBitMap.baseAddr := NewPtr(Size(21888));
{ Test de l'allocation du pointeur: si égal à NIL, }
{ alors pas assez de mémoire: on quitte. }
  IF oldBitMap.baseAddr = NIL THEN
    ExitToShell;
  oldBitMap.bounds := thePort^.portBits.bounds;
  SaveImage(oldBitMap);
{ Définition d'un BitMap qui contiendra l'image }
{ après l'ajout d'un nouvel objet }
  curBitMap.rowBytes := thePort^.portBits.rowBytes;
  curBitMap.baseAddr := NewPtr(Size(21888));
{ Test de l'allocation du pointeur: si égal à NIL, }
{ alors pas assez de mémoire: on quitte. }
  IF curBitMap.baseAddr = NIL THEN
    ExitToShell;
  curBitMap.bounds := thePort^.portBits.bounds;
  SaveImage(curBitMap)
END; { Initialisations }

PROCEDURE Fermetures;
BEGIN
{ Abandon de la mémoire occupée par oldBitMap }
  DisposPtr(oldBitMap.baseAddr);
{ Abandon de la mémoire occupée par curBitMap }
  DisposPtr(curBitMap.baseAddr)
END; { Fermetures }

BEGIN { Main }
  Initialisations;
  MainEventLoop;
  Fermetures
END. { Main }

```



CAMELOT-INFO

La librairie informatique à l'avant-garde au Canada

- POUR SON CHOIX DE PUBLICATIONS DES PLUS COMPLETS
 - POUR SES LOGICIELS À DES PRIX COMPÉTITIFS
 - POUR SON PERSONNEL QUALIFIÉ
 - POUR SON SERVICE POSTAL EFFICACE

Camelot-Info: livres, périodiques et logiciels pour ordinateurs

**1191, PLACE PHILLIPS, MONTRÉAL, QUÉBEC H3B 3C9
(514) 861-5019**

LE MOT DU CHIALEUX

Jean-Pierre Léger

Au lieu de chialer contre le journal, j'ai décidé d'y aller de ma modeste contribution. Je m'excuse de faire un peu anachronique et de ne pas me situer dans le courant d'humour "pipi caca" qui semble très en demande. Ca y est, je viens de sentir la page du journal qui vient de tourner. Bon, alors pour ceux qui restent, je vous fais part de quelques unes de mes minis réflexions dans ma micro tête de linotte. Et puis si vous trouvez que c'est de la merde, dites vous que ça n'occupe que 3% du journal. Tiens, il me semble que j'ai entendu ça dernièrement....

Le fait de se retrouver plusieurs heures consécutives devant son écran cathodique ne donnerait pas seulement des problèmes aux yeux. Du moins, c'est ce que pensent un certain nombre de psychologues qui se sont penchés sur la question. On parle d'un phénomène relativement récent, une nouvelle "maladie" dont la proie serait les férus d'informatique. Vous savez ceux qui passent leurs soirées, leurs fins de semaine au complet devant leur écran. Ca ne vous dit rien? Ah bon. Disons qu'on n'est pas obligés de remettre en question sa vocation d'informaticien pour autant, mais on peut quand même se dire que s'il y a des gens qui s'en inquiètent, c'est qu'il y a peut-être de quoi là.

On peut facilement concevoir que la personne qui se retrouve isolée à pitonner dans son coin des années durant, oubliant qu'existent autour d'elle amis, famille, etc. risque de développer des petits problèmes. Because, aucun micro, même branché trèsss zénormément longtemps, ne dégagera jamais, mais alors là jamais, de chaleur humaine. Bon, peut-être

que t'en as rien à foutre, mais papa ici a envie d'y aller d'un certain appel à la prudence: quand tu te retrouveras au travail à passer tes journées à pitonner et que tu ne rêveras que de retrouver ton PC le soir venu, à ce moment là commence à t'inquiéter. La rupture avec le monde extérieur, on ne sait pas où ça peut mener....

Dernièrement dans *La Presse*, Francine Pelletier nous faisait part de ses inquiétudes face aux "logiciels pour adultes" qui circulent en Amérique du Nord, et de "Playmac" en particulier. Lors de salons informatiques, on retrouve les concepteurs qui, disques sous le bras, écoulent leur stock en des temps records. Faut croire que c'est "ben" populaire. Qu'on soit d'accord ou non avec ces logiciels, c'est une question qui déborde largement de notre petit monde de l'informatique. Scusez-moi, mais je me demande quand même quelle idéologie est derrière "Playmac" lorsqu'on y retrouve une fille écartelée, enchaînée par les bras et les jambes et qui crie des ah! et des oh! Ca promet avec les écrans de demain à haute résolution. C'est ti ça la technologie au service de l'"homme"?...

Bref, si on combine les inquiétudes des psychologues et celles de Francine Pelletier, on obtient que la probabilité qu'une personne qui passe des années isolée dans son sous-sol (qu'est-ce qui es isolé, la personne ou le sous-sol? Tant qu'à moi, ils sont finis tous les deux) à jouer à "Playmac" soit déséquilibrée est assez grande.

Rappel : Baiser pour vrai c'est franchement meilleur, surtout sans chaînes.

Jean-Pierre Léger



Lettre ouverte à Jean-Pierre Léger

Monsieur,

Suite à votre lettre qualifiant notre journal étudiant de torchon, je n'ai pu m'empêcher de constater le manque de vécu d'une personne qui a atteint l'âge adulte.

Vous pensiez sans doute que les étudiants d'informatique auraient tous été dessinés sur le même modèle que le vôtre, mais quelle fut votre déception de constater que vous vous retrouviez encore dans un milieu constitué de simples humains comme vous en avez tant rencontrés dans vos communautés scolaires antérieures.

Dans votre lettre vous n'insultez pas seulement ces bénévoles qui sont avant tout des informaticiens, mais aussi l'ensemble des membres de notre communauté étudiante.

Sachez cher Monsieur, que pour votre culture personnelle, vous avez plusieurs cours mis à votre disposition, pour l'actualité vous avez les journaux qui sont quotidiennement disponibles au premier étage, pour votre culture politique vous pouvez toujours lire le continuum et laisser donc l'Interactif aux plus simples d'esprit qui désirent se divertir un peu.

Pour ce qui est de votre complexe d'être un étudiant d'informatique, je vous affirme que vous seriez plus heureux en mettant de côté ces supposés préjugés et en prouvant à ceux qui vous fuient que vous êtes autre chose qu'une machine à programmer. Ainsi vous pourriez enfin marcher la tête haute.

Pour finir, j'espère que vous cesserez de faire le spectateur capricieux et que vous vous impliquerez dans un des services dispensés aux étudiants, ou, si vous préférez garder votre rôle de critique, que vous vous attaquerez à des GOUVERNEMENTS qui sont constitués de gens pas plus "réfléchis" que ceux d'informatique et qui s'acharnent à vouloir détruire un peuple et sa langue.

Sans rancune,

CARL BILODEAU

2e année





Critique gastronomique

tirée du guide Mi-chemin

Établissement : Cantine Canada

Adresse : 3200 Jean Brillant

Propriétaire : M. C. Canada

Chef : ne gagne pas à être connu

Lors de sa périgrination, notre correspondant cana..québécois (le Chef Groslot) a visité le site hautement touristique de l'Université de Montréal, soit la Cafétéria du Jean-Brillant. Voici les termes inexacts utilisés par ce flambeau de la gastronomie québécoise, qui n'a eu qu'une objection sur le vaste menu : il n'y avait pas de *crastillon*.

Revenu de cette première déception, il fait état de ces constatations. Tout d'abord il assiste à une marche de protestation organisée par des cuisses de poulets tandis que les poitrines, d'une même voix, s'élevaient contre la triste fin qu'on leur réservait. Ce triste incident avait pour motif, le peu de cas que le chef de cette institution ("Cantine Canada") faisait de ces petits êtres courageux ("Les Poulets").

Notre reporter était sur les lieux pour recueillir les commentaires des têtes dirigeantes, qui de leur propre aveu n'étaient pas dans le cou (tête, cou, la comprenez-vous??).

Interactif :

Le but de cette protestation, expliquez le-mel.

Président de l'association des Poulets aux Stéroïdes Anabolisants du Québec (A.P.S.A.Q.) : Cette manifestation a pour but de montrer notre indignation devant ce que nous qualifions d'abomination aberrante et outrancière, soit la façon inacceptable de présenter nos membres (la comprenez-vous?) au public. Notre mot d'ordre est :

"Mourir oui, mais pas pour devenir ça!"

Interactif :

Qui a eu l'idée de cette manifestation?

A.P.S.A.Q. :

Certainement pas nos membres, ils n'ont plus toute leur tête.

Interactif :

Prévoyez-vous d'autres coups d'éclats de cette

envergure?

A.P.S.A.Q. :

Je ne le poulet pas le dire, car nous demeurons dans la crainte que notre employeur, "Cantine Canada", fasse appel aux poulets de l'anti-émeute. Une chose est certaine, nous ne tolérerons pas que l'on nous prenne pour les "dindons de la farce".

Interactif :

Un poulet de mes amis m'a dit que votre regroupement n'était pas représentatif de la pensée "aviaire".

A.P.S.A.Q. :

Dans un premier temps je vous répondrai que dans l'état dans lequel "Cantine Canada" nous présente, bien malin celui qui pourrait affirmer sans l'ombre d'un micro-doute que nous sommes bel et bien des poulets.

Dans un deuxième temps, je peux vous dire que nos détracteurs ne sont que d'ignobles poules mouillées.

Notre interlocuteur étant monté sur ses ergots, toute communication est devenue impossible; nous n'avons pas voulu retenir des propos qui poulet dépasser sa pensée.

Pour ce qui est du reste du menu, le chef Groslot l'a trouvé vraiment infecte; un point positif cependant, les rôties (toasts) sont vraiment bien apprêtées, sur ce dernier item rien à redire, puisque ces rôties sont cuisinées avec le summum de la haute technologie, un grille-pain huit tranches programmable.

Vient le moment fatidique de l'évaluation; notre correspondant doit assigner ses étoiles :

* médiocre (voir cambuse)

** acceptable (voir Fast-Food de 3e ordre)

*** bien (voir No. 1 for two)

**** très bien (Catégorie du Ritz et compagnie)

Cantine Canada :

moins 5 étoiles (voir médecin après repas; Santé et Bien-être social Canada considère que le danger croît avec l'usage - éviter d'avaler)

Guy Pilon

Nouveautés

Sylvain Verreault

LA NOUVELLE STATION DE TRAVAIL SUN 386i

L'union de deux mondes

La compagnie Sun Microsystems fut fondée en 1982 pour développer des stations de travail puissantes, axées sur de hautes capacités graphiques et de réseaux. Aujourd'hui, Sun est une force dominante dans le marché des stations de travail basées sur le système UNIX. Les systèmes Sun sont basés sur le Motorola 68000, SPARC (l'architecture RISC développée par Sun) et, tout récemment, sur le INTEL 80386. C'est de cette toute nouvelle station dont je vais vous parler ce mois-ci.

Depuis longtemps, les compagnies de développement de logiciel ont envié l'environnement UNIX pour le développement de programmes d'application DOS. Comme la plupart d'entre vous l'ont déjà constaté en travaillant sur les stations APOLLO, le système UNIX est beaucoup plus flexible, et offre une librairie de fonctions beaucoup plus vaste que DOS. C'est en visant l'unification de ces deux mondes que Sun a créé la station 386i, qui offre une plate-forme de développement pour DOS, tout en ayant les avantages d'un environnement UNIX. Cependant, ne vous dé trompez pas: la station 386i est une station Sun, et donc est une implémentation UNIX dans toutes sa splendeur. Cependant, grâce au mode virtuel 8086 du processeur 386, la station peut simuler un environnement DOS complet avec 640k de mémoire. D'autres différences fondamentales viennent souligner le fait que le 386i est une station de travail Sun et non un PC. En effet, les constituants internes sont ceux de Sun (ROM, connecteur Ethernet, tampon vidéo de 32 bits, entre autres...).

Détails techniques

La station 386i est basée sur le microprocesseur 80386. Ce dernier peut fonctionner soit à une vitesse de 20 MHz pour le modèle 150, ou soit à une vitesse de 25 MHz pour le modèle 250. Les deux systèmes sont offerts de façon standard avec un co-processeur mathématique 80387 fonctionnant à la même vi-

tesse d'horloge que le CPU. Comme la plupart de vous le savent déjà, la résolution graphique des écrans Sun est spectaculaire. Sun offre une variété d'écrans vidéo pouvant être utilisés avec la station:

- un écran couleur de 16 ou 19 pouces, tous deux d'une résolution de 1156x900 pixels et offrant 256 couleurs choisies d'une palette de 24 bits (8 bits de rouge, 8 de vert et 8 de bleu): les images extraordinaires; cependant, ces deux moniteurs ne sont pas donnés (voir le tableau des prix)
- un écran couleur de 14 pouces offrant une résolution de 1024x768 (ce qui n'est pas si mal!)
- un écran monochrome de 19 pouces offrant une résolution de 1152x900 pixels.

Le support pour le standard SCSI est également standard sur le système. La station peut recevoir des unités de disques durs de 91 MB ou 327 MB. De plus, un connecteur SCSI additionnel à l'arrière de la station offre la possibilité d'intégrer d'autres unités à l'environnement de travail.

On retrouve également sur les stations la souris optique Sun, qui vient de façon standard avec la station de travail. Cette souris peut émuler celles qu'on retrouve dans le monde des PC, notamment la souris de Microsoft. Cependant, cette émulation n'est utilisable que lorsqu'on travaille dans une fenêtre en mode DOS.

Enfin, le 386i peut simuler la mémoire "étendue" (expanded memory) que l'on retrouve aujourd'hui sur la plupart des systèmes DOS à haute performance. Un "driver" EMS 4.0 est inclus dans chaque fenêtre DOS créée, et celui-ci supporte jusqu'à 32 MB de mémoire étendue. La mémoire est simulée grâce à la mémoire virtuelle sur un fichier UNIX.

Système d'exploitation

Le système d'exploitation est SunOS, qui est basé sur les deux versions UNIX standard, soit le système V version 3 et la version BKD 4.3 (celle des stations Apollo). Le système comprend l'interface graphique (menus et fenêtres) SunView, qui est semblable à l'interface

La station de travail Sun 386i

retrouvée sur les stations Apollo et Sun du département IRO. Je ne m'étendrai pas sur le sujet, mais je vais plutôt parler de la façon dont sont implémentées les fenêtres DOS.

Une fenêtre DOS, sur le 386i, est implémentée grâce au mode virtuel 8086 du microprocesseur 386. Dans ce mode, chaque fenêtre se comporte comme si elle possédait un PC avec 640 Kb de mémoire vive. Le processeur à un espace d'adresse équivalent à celui du 8086, mais peut cependant exécuter la plupart des instructions ajoutées au 186, 286 et 386 (dans le mode non-protégé).

Statistiques vitales...

Je vous dresse ici un tableau des prix (tirés de PC TECH journal, mars 89), pour vous donner une idée de ce qu'il vous faudra débourser pour acquérir une station Sun 386i...

Modèle 150 (\$7490)

- Micro-processeur INTEL80386 fonctionnant à 20 MHz
- Mémoire vive de 4 MB
- Coprocesseur INTEL 80387
- Horloge en temps réel
- Port série et parallèle
- Souris optique à trois touches
- Support SCSI
- Connecteur Ethernet de 10 Mbps
- Unité de disquette 3.5" (1.44 MB)
- Clavier à 107 touches
- SunOS avec MS-DOS 3.3 intégré

Modèle 250 (\$11045)

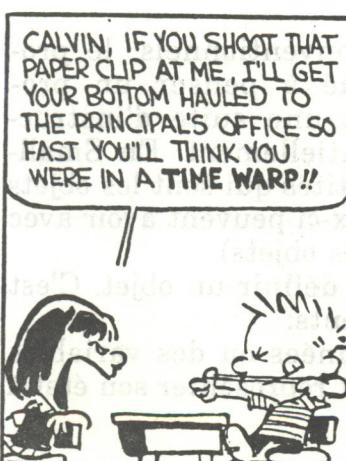
- Même que 150, sauf:
- Fonctionne à 25 MHz
 - Mémoire cache de 32 KB
 - Contrôleur de cache INTEL 82385
 - Mémoire vive de 8 MB

Options disponibles

- 4MB de mémoire (80 ns, SIMM) \$3400
- Moniteur couleur 19" (1156x900) \$8055
- Moniteur couleur 16" (1156x900) \$5055
- Moniteur couleur 14" (1024x768) \$3300
- Moniteur monochrome 19" (1152x900) \$2780
- Moniteur monochrome 15" (1152x900) \$1650
- Carte VGA \$895
- Disque dur SCSI 91 MB \$2200
- Disque dur SCSI 155 MB \$4125
- Disque dur SCSI 327 MB \$6600

Conclusion

La station de travail Sun 386i est conçue pour le développement professionnel d'applications soit UNIX, soit DOS. Elle peut également servir, comme les autres stations Sun, au design assisté par ordinateur ou à l'ingénierie. Il est évident que de tels outils de développement sont hors de la portée du simple utilisateur. Cependant, il est fascinant de comparer les capacités d'une telle station de travail à un simple micro-ordinateur, si puissant soit-il...



Les langages objets

Marc Bégin

Dans cet article je vais parler d'un sujet qui est de plus en plus à la mode: les langages objets. Ceux-ci commencent à prendre une place de plus en plus importante parmi la communauté scientifique. Beaucoup de scientifiques non programmeurs les emploient parce qu'il est facile pour eux de créer des relations entre les objets programmés et ceux de la réalité. Par exemple, un architecte va découper une maison en plusieurs objets tel que le toit, les murs, les fondations, le circuit électrique; et chacun des ces objets pourront à leurs tour être décomposés en plusieurs sous objets.

Parmi les langages les plus populaires, on retrouve:

- C++
Dérivé du langage C.
- Objective-C
Utilisé dans l'ordinateur NeXT.
- ADA
Implante certaines notions d'objet avec les "Packages".
- Simula
Utilisé pour faire de la simulation.
- Smalltalk
Langage tout objet avec une interface graphique et interactive.

Je vais parler plus spécialement de Smalltalk, pour deux raisons:

- 1- C'est un langage qui est totalement objet. Tout est implanté sous forme d'objets, même l'arithmétique et la compilation.
- 2- C'est le langage objet que je connais le mieux. (!!?)

Dans les langages conventionnels, le programmeur est habitué à penser un programme comme étant une suite d'instructions exécutées séquentiellement. En Smalltalk, on doit voir les entités qui sont les objets et les relations que ceux-ci peuvent avoir avec leurs congénères (autres objets).

Premièrement il faut définir un objet. C'est composé de deux éléments:

- Une structure de données ou des variables locales à l'objet qui vont représenter son état à tout moment.

- Des méthodes (équivalent à des procédures) qui sont utilisées pour communiquer avec les autres objets du système. Seules ces méthodes auront accès aux variables locales de l'objet.

Chaque objet fait partie d'une classe; on dit qu'il est une instance de cette classe. Lorsqu'on programme en Smalltalk, on ajoute des classes à partir desquelles les objets vont être créés (voir l'exemple à la fin de l'article).

Chaque classe est une sous classe d'une autre, et la classe qui se trouve à la racine se nomme "Object". Toutes les classes sont donc des descendantes de "Object". Ce qu'il y a d'intéressant dans la structure hiérarchique, c'est qu'une classe a accès à toutes les variables et méthodes de ses ancêtres. Donc tout le monde a accès aux méthodes définies dans la classe "Object".

Voici un exemple de déclaration de classe:

```
Noeud subclass: #NoeudCle
instanceVariableNames: 'cle droit gauche'
classVariableNames: ''
poolDictionaries: ''
```

NoeudCle est une descendante de la classe Noeud. Ici on déclare trois variables locales à l'objet: 'cle droit gauche'. Il faut remarquer que les variables n'ont pas de type. Dans Smalltalk les variables peuvent contenir n'importe quel objet; par exemple "cle" peut contenir la clé de l'arbre binaire et "droit" et "gauche" ses fils.

ClassVariableNames est utilisé pour déclarer des variables qui sont partagées par tous les objets d'une classe.

PoolDictionaries est très rarement utilisé. Il sert à déclarer des dictionnaires d'objets qui sont partagés entre plusieurs classes.

Il faut ensuite ajouter des méthodes à notre classe. Une méthode est envoyée à un objet. Cet objet est appelé le receveur. Une méthode c'est comme une procédure. Elle peut avoir des paramètres et des variables locales.

Voici un exemple de méthode:

```

auCarre: nombre | carre |
" Déclaration d'une variable locale à la
méthode"
carre := nombre * nombre ^carre.
" ^ = retourner la valeur "

```

OU

```

auCarre | nombre |
nombre := self. " Obtenir le receveur "
^ (nombre * nombre)

```

La première méthode a un paramètre, "nombre", tandis que la seconde n'en n'a pas, mais elle utilise le receveur "self" comme variable.

Le mot réservé "self" représente toujours le receveur. Dans ce cas ci, il faut que le receveur soit un nombre, parce qu'il doit répondre au message "*".

Lorsqu'on envoie un message, le receveur se trouve toujours à gauche du message.

Voici comment on pourrait utiliser l'exemple

```

| carre |
carre := 100 auCarre.
" 100 est le receveur "
carre printOn: Transcript.
" Affiche le receveur dans une fenêtre du
nom de Transcript"

```

En regardant l'exemple à la fin de l'article, il y a les "class methods" et des "methods", car il faut différencier les messages de création d'objet (par ex. le message "new") des messages adressés à l'objet lui-même.

Dans les messages de création, le receveur est le nom de la classe duquel on veut créer un objet. Exemple:

```
variableAleatoire := Random new.
```

où Random est une classe. Dans les messages de manipulation, c'est l'objet qui est le receveur. Exemple:

```
" Retourne le prochain nombre aléatoire "
NombreAleatoire := variableAleatoire next.
```

On a vu à date que les méthodes sont composées de messages envoyés et d'assignments. Quelqu'un pourrait se demander quand est-ce qu'il se passe réellement quelque chose dans le système parce que envoyer des messages ne fera pas afficher mon texte à l'écran ou calculer une valeur au carré. Un certain nombre

de méthodes (environ 100) sont des primitives. Au lieu d'envoyer d'autres messages, elles exécutent du code assembleur. Tout le reste du système est écrit en smalltalk et sa source est fournie.

Smalltalk est un environnement interprété et interactif. Tout est contenu dans un bloc monolithique. Il n'y a pas de notion de programme. Programmer consiste à ajouter des classes au système déjà existant. Smalltalk contient des classes pour créer et manipuler des fenêtres, la souris, du graphiques, des ensembles d'élément, des fichiers...

Il existe un implantation de Smalltalk sur PC: Smalltalk/V par Digitalk inc.

Sources:

Smalltalk/V par Digitalk inc.

Smalltalk-80. The language and its implementation
Adele Goldberg and David Robson.

Ce qui suit est un exemples complet. Il comprend les classes utiles pour créer un arbre binaire. On pourrait facilement l'étendre.

**** Arbre ****

Cette classe est utilisé pour faire la gestion de l'arbre. Elle contient deux variables: racine et nbElem.

racine: Contient un Noeud qui est la racine de l'arbre.

nbElem: Contient un entier qui compte le nombre d'elements dans l'arbre.

```
Object subclass: #Arbre
```

```
instanceVariableNames: 'racine nbElem'
```

```
classVariableNames: ''
```

```
poolDictionaries: ''
```

Arbre class methods

```
new " Creer un nouvel arbre et initialiser
les variables " ^super new initialize
```

Arbre methods

```
ajoute: unElement
```

```
" Ajoute un element dans l'arbre "
```

```
" On insere a partir de la racine "
```

```
racine := racine insert: unElement.
```

```
nbElem := nbElem + 1
```

```
initialize
```

```
" Creer un nouvel arbre "
```

```
nbElem := 0.
```

```
racine := NoeudNull new
```

```

recherche: unElement
" Recherche un element dans l'arbre "
^ racine recherche: unElement
***** Classe Noeud *****
C'est une classe générique qui a deux sous classes: NoeudCle, NoeudNull. Cette classe contient qu'une variable globale (NoeudNullVar) à toutes les instances de la classe et des ses sous classes. Cette variable est une instance de la classe NoeudNull est l'équivalent du pointeur nil de Pascal
Object subclass: #Noeud
instanceVariableNames: ''
classVariableNames: 'NoeudNullVar'
poolDictionaries: ''

***** NoeudCle *****
Les instances de cette classe sont les noeuds de l'arbre.
La variable "cle" contient l'objet qui représente la cle. Cet objet peut être n'importe quoi du moment qu'il répond aux messages ">=" et "= ". Ce message est utilisé pour comparer les clés dans les méthodes "insert:" et "recherche:".
Les variables "droit" et "gauche" contiennent le fils droit et le fils gauche de l'arbre. Ceux-ci sont soit une instance de "NoeudCle" si c'est un noeud normal, soient une instance de "NoeudNull" si c'est l'équivalent d'un pointeur nil.

Noeud subclass: #NoeudCle
instanceVariableNames: 'cle droit gauche'
classVariableNames: ''
poolDictionaries: ''

NoeudCle class methods
newCle: unElement
" Creer un nouveau noeud + initialiser la cle "
^super new initialize: unElement

NoeudCle methods
initialize: uneCle
" Initialiser un feuille avec une cle "
cle := uneCle.
droit := gauche := NoeudNullVar
insert: unElement
" Ajouter un element dans l'arbre "
cle >= unElement
ifFalse: [
droit := droit insert: unElement]
ifTrue: [
gauche := gauche insert: unElement].
^self yourself "Retourne le receveur"

```

```

recherche: unElement
cle = unElement
ifTrue: [ ^cle ]
ifFalse: [
cle >= unElement
ifFalse: [
^droit recherche: unElement]
ifTrue: [
^gauche recherche: unElement].
]

***** NoeudNull *****
Cette classe représente un pointeur nil dans l'arbre. Il doit n'y avoir qu'une seule instance: NoeudNullVar.

Noeud subclass: #NoeudNull
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''

NoeudNull class methods
new
^NoeudNullVar := super new
" super est une pseudo-variable comme self"
" qui est utilisé pour passer un message "
" au parent de la classe "

NoeudNull methods
insert: unElement
" Ajouter un element a l'arbre "
" Creer un nouvelle feuille "
^ (NoeudCle newCle: unElement).
recherche: unElement
" Rechercher un element dans l'arbre "
" Si on est rendu ici, c'est sur que "
" l'element n'existe pas "
"nil.
" Retourner une valeur qui indique "
" aucun element "

***** Exemples d'utilisation *****
| rand arbre |
" Deux variables locales "
rand := Random new.
" Creer un objet qui va nous donner "
" des nombres aleatoires"
arbre := Arbre new.
" Creer notre arbre binaire "
100 timesRepeat: [
" Ajouter 100 nombres aleatoires "
" dans l'arbre "
arbre ajoute: (rand next)].
```

Les mouvements Madeau

Sylvain Boulé

M. Euphrene Madeau a développé en revenant de la guerre 1914-18 des mouvements du corps qui, sans le moindre effort, permettent d'acquérir ou de maintenir une forme physique à toute épreuve et même de perdre du poids! Certaines personnes pratiquant les mouvements Madeau régulièrement depuis quelques années ont constaté une amélioration sans précédent de leur condition physique et mentale. Voici quelques témoignages véridiques:

Mme Thérèse Mondoux de Ste-Pie-de-Bagot:
Depuis que j'ai découvert les mouvements Madeau, ma vie est complètement transformée; finies les difficultés causées par l'arthrite, la mauvaise digestion, la tuberculose, les varices et autre malaises de toutes sortes.

M. George St-Jean de Pointe-aux-Pics:
M. Madeau m'a sauvé la vie! J'étais atteint d'un cancer en phase terminale; les médecins me donnaient un an à vivre. Ça fait dix ans de cela! J'ai perdu 50 livres en trois semaines et je respire mieux depuis que je pratique régulièrement les mouvements Madeau, et ce, sans effort! Je peux enfin vivre pleinement ma vie sans aucune contrainte.

La méthode Madeau, c'est **facile et sans effort**. Tout le monde peut pratiquer les mouvements Madeau tous les jours **sans aucun risque** pour un bien-être total **garanti** ou argent remis. Voici, en introduction, quelques mouvements primaires de la méthode Madeau que vous pourrez essayer dès maintenant, à la maison (fermez les rideaux).

1) Mouvement du poisson: Permet de dégager les voies respiratoires et favorise la circulation sanguine au niveau des cuisses, et donc aide à réduire les varices. Couchez-vous sur le ventre, les bras en croix, les jambes à la verticale, les genoux pliés vers le bas du corps, les pieds carpés de façon à pointer vers le plafond tout en maintenant les talons plus hauts que la pointe des pieds. Levez et abaissez sporadiquement vos poignets et ce, avec modération; il ne s'agit pas de se fatiguer inutilement, il faut bouger en souplesse tout en respirant entre chaque centaine de mouvements.

2) Mouvement de l'oiseau: Permet au système cardio-vasculaire un relâchement bénéfique, tout en favorisant le mouvement des articulations au niveau du dos et donc aidant à réduire l'arthrite chronique. Couchez-vous sur la tête, les bras repliés en forme de cœur au dessus des orteils, les jambes maintenues sous les genoux et le bassin bien droit. Bougez vos jambes de façon alternative l'une après l'autre de façon à amener vos talons derrière la nuque puis vos pieds bien à plat dans le dos. Il est important de bien respirer et de bien détendre tous les muscles du corps pendant le mouvement.

3) Mouvement de la pelle mécanique: Permet d'accentuer l'élasticité du cou au niveau des cinquième et sixième vertèbres cervicales. Accentue également la région du cerveau réservée au bonheur en oxygénant la région concernée de façon adéquate. Debout sur le pied gauche, la jambe droite sur l'épaule (par derrière), penchez votre buste en arrière de façon à former un angle de 90 degrés avec la jambe de soutien (la gauche). Entrecroisez vos mains sur votre entrejambe de façon à avoir un bras au dessus du corps (sur le ventre) et l'autre au dessous (dans le dos). Faites des cercles avec votre tête en accentuant le mouvement de plus en plus telle une spirale jusqu'à ce que vous voyiez vos mains des deux côtés (avant et arrière). N'oubliez pas de respirer!

4) Mouvement de la pieuvre: Permet de générer un soulagement sans précédent au niveau des cors aux pieds et des problèmes de vessie, du pancréas ainsi que des cheveux gras et laids. Position debout, sautez sur place tout en bougeant vos bras dans tous les sens de façon aléatoire. Pliez votre corps d'avant en arrière tout en maintenant vos oreilles au niveau du nombril. Vos pieds devront successivement toucher vos omoplates. Faites les mouvements à votre rythme sans essayer de tenir plusieurs heures la première fois; une heure suffit amplement.

Pour plus de détails sur les mouvements ci-haut mentionnés et plus de 1042 autres tout aussi bénéfiques et faciles, envoyez-nous dès aujourd'hui une demande d'adhésion accompagnée de la modique somme de \$149.99 à l'adresse suivante:

123 Rang des Culs-de-Poules
Ste-Émilie de l'Énergie
A1B 2C3
Tél: 123-4567

Sondage

Yves Gaudreau

Un étudiant m'a demandé de vous poser la question suivante :

Seriez-vous intéressé à participer à un rallye automobile qui serait possiblement organisé le dimanche 2 avril. Le coût en serait de 10\$ par automobile. Il y aurait des prix de participation et des prix aux gagnants. Ensuite, il y aurait un party "après-rallye" au clandestin pour la remise des prix.

Etes-vous intéressé ? OUI NON

Maintenant, voici les résultats des deux derniers sondages.

Le premier sondage portait, je vous le rappelle, sur les revues informatiques et les langages de programmation. Sur un total de 15 répondants, 15 ont répondu lire des revues.

Voici les résultats:

Byte : 7, PC Magazine : 4, Amiga World : 4, Transactor : 3, Mac World : 2, Mac User : 2, Informatique et Bureautique : 2, Amiga Time : 1, PC World : 1, Turbo Technics : 1, PC Tech Journal : 1, les autres revues : 0.

Une personne a répondu lire ACM Communication et une autre le Unix World.

Pour les langages de programmation, 6 aiment le PASCAL, 5 le ADA, 3 le C et 2 l'Assembleur.

Pour les langages d'intelligence artificielle, 4 aiment le Prolog, 2 le Lisp et une a répondu le Sheme. Une personne a répondu aimer le Logo (!).

Pour les langages de bases de données, 6 aiment le D-Base, 2 le Oracle, une le MainSail. Deux ajouts à la liste, Paradox pour une personne et DML pour une autre.

Le deuxième sondage portait sur le genre d'activité parascolaire que vous aimeriez avoir. Dix personnes ont courageusement répondu. Voici les résultats partiels et suggestions.

Pour ce qui est du genre de musique qu'on aimerait entendre dans un party, le Commercial et le Danse Music reviennent le plus souvent. On a fait mention de Heavy Metal, Rock, Techno-Rock/Pop, Alternatif, Jazz, Québécois, Nouvel Age, House Music et Musique Latino-américaine.

Bref, on aime de tout.

On a fait une suggestion intéressante : des midis-party auquel un genre de musique est attribué, par exemple un midi Jazz, un midi Heavy-Metal.

Je dois m'arrêter ici; les résultats complets seront dans le prochain Interactif.

Qu'y aura-t-il de neuf au mois d'avril? Notre super numéro de fin d'année!