

INFORMATIQUE ET ÉDUCATION

APL UN LANGAGE D'AVENIR

UNE EXPÉRIENCE À L'ÉCOLE SECONDAIRE
JOSEPH-FRANÇOIS PERRAULT

par NICOLE LEMIEUX

professeur
École secondaire
Joseph-François Perrault

Depuis septembre 1982, il se vit à l'école secondaire Joseph-François-Perrault une expérience en pédagogie informatique : la programmation en langage APL. Conduite par la DSIR (Direction des Services Informatiques aux Réseaux), cette expérience vise les 4 objectifs suivants : 1) prouver qu'APL est un langage très bien adapté aux besoins des applications pédagogiques de l'ordinateur (APO); 2) défaire le mythe : APL langage d'universitaire ; 3) explorer les possibilités de développement de didacticiels ; 4) amorcer la formation du personnel enseignant de l'école en APL. Après un an d'expérience dans ce milieu, on possède de bons éléments pour en rendre compte de façon concrète.

1. APL : Un langage très bien adapté aux besoins des APO

Les APO (Applications pédagogiques de l'ordinateur) sont très variées et bien souvent encore à un niveau expérimental. On peut classer ces applications en quelques grandes catégories :

- enseignement délivré par ordinateur ;
- exercices répétés, avec ou sans diagnostic ;
- simulation, calculs de laboratoire ;
- commandes de processus ;
- *résolution de problèmes.*

Cette dernière catégorie d'applications nous semble être la plus enrichissante pour l'élève. On ne peut en effet trouver de meilleure façon de démystifier l'ordinateur car l'élève apprend à commander (programmer) lui-même la machine en résolvant ses propres problèmes selon un algorithme qu'il choisit.

Naturellement, cette APO exige de l'élève l'apprentissage d'un langage d'ordinateur. L'élève doit donc consacrer une certaine partie de son temps à cet apprentissage, mais il faut noter que cette application n'exclut pas les autres catégories d'utilisation. Au contraire, elle ne peut que les faciliter, car l'élève qui a appris les rudiments des commandes et du langage de la machine n'aura aucune difficulté à faire exécuter des programmes déjà disponibles à d'autres fins. Nous croyons que pour répandre l'usage de l'ordinateur dans plusieurs disciplines, il est nécessaire d'employer un langage de haut niveau, c'est-à-dire un langage qui n'est pas lié aux concepts de programmation de l'informatique professionnelle. Dans cette optique, nous croyons que le langage BASIC, qui est le plus utilisé, est un langage d'informaticien. Bien qu'il soit largement répandu sur les micro-ordinateurs actuels, le langage BASIC répond mal aux exigences d'un langage pour les APO, et cela pour les raisons suivantes :

- il ne permet pas une *programmation modulaire* ;
- il manque de *flexibilité* au niveau des facilités de programmation : l'utilisateur est contraint à des artifices laborieux pour résoudre des problèmes conceptuellement simples ;
- il n'est pas standardisé : les commandes varient sensiblement d'une machine à l'autre.

De plus, s'il est vrai que l'apprentissage du BASIC est facile (au niveau des opérations ou des commandes élémentaires), il n'est pas aussi simple de traduire en BASIC un algorithme quelconque. En effet, le niveau du langage impose une décomposition qui déforme même la pensée algorithmique naturelle. En fait, l'élève sacrifie plus de temps à la machine qu'à son propre problème. On peut dire que, dans une certaine mesure, le langage BASIC freine la créativité de l'élève à cause de la lourdeur de sa manipulation.

On sait que les progrès fulgurants de la technologie permettent maintenant d'aller vers des langages de plus haut niveau pour se libérer de la machine. Il y a cependant certaines habitudes et résistances au changement à vaincre particulièrement chez les informaticiens qui, en général, font les choix des machines et des langages.

Si on ne peut pas échapper à toutes les contraintes d'une machine et d'un langage, il est sou-

haitable de s'en affranchir le plus possible pour rendre l'ordinateur plus accessible. Le langage APL a été créé en fonction de cet objectif.

APL offre actuellement la meilleure *flexibilité*, par rapport aux autres langages, au niveau : de l'interaction de l'ordinateur ; de l'écriture modulaire des programmes ; de la facilité de stockage des données ; de l'allocation dynamique de mémoire ; de la généralisation des opérations primitives à des ensembles des données ; de la variété et de la puissance des opérations primitives.

Avec tous ces outils puissants, on peut donc développer très rapidement des logiciels. Et grâce à la facilité de stockage des données, le professeur intéressé peut enrichir lui-même les programmes avec ses propres données.

Le langage APL a permis également de développer facilement, pour une table traçante, un « package » graphique analogue à celui du langage LOGO, avec des primitives AVANCE, TOURNE, etc. Vous trouverez en annexe un exemple classique de graphisme réalisé avec APL.

2. Le mythe : APL, langage d'universitaire

Grâce aux mémoires plus grandes et moins chères qui sont maintenant disponibles sur les micro-ordinateurs, la voie est maintenant ouverte à l'utilisation d'un langage de programmation à la fois très simple et très souple d'utilisation comme APL.

Programmer en APL, c'est agréable ! Les primitives (procédures ou fonctions du langage) extrêmement puissantes sont directement exécutables. La puissance des primitives permet de faire en peu de code des opérations importantes comme des tris, des vérifications d'appartenance, des inversions, etc. La structure des données est très fonctionnelle. On travaille sur des tableaux et les opérations s'effectuent globalement sur ces tableaux. Le langage est modulaire. Un programme peut en appeler un autre ou se servir des résultats d'un autre programme.

Bien sûr, APL est aussi un langage d'universitaire pour celui qui veut exploiter toutes les possibilités et les subtilités du langage, mais il peut aussi s'adresser à des gens plus jeunes. À l'École Joseph-François-Perrault, des élèves de secondaire I à V ont participé dans le cadre d'une activité scolaire à la réalisation de programmes en APL.

Le symbolisme du langage, loin de les rebuter, les amuse. De fait, quand l'élève connaît les principales primitives (ça vient assez rapidement), il

APL

commence à rédiger ses propres programmes, il enrichit le langage de nouveaux mots. L'âge de ces élèves varie de 12 à 17 ans.

Un sondage auprès des élèves, réalisé à la fin de l'année, révèle qu'ils ont tous aimé l'expérience de programmation avec APL et qu'ils ont même, en grande partie, trouvé que l'apprentissage d'APL est facile.

3. Explorer les possibilités de développement de didacticiels

Parmi les didacticiels réalisés, nous avons choisi de présenter rapidement deux prototypes, l'un qui peut être utilisé dans plusieurs matières d'enseignement, l'autre qui peut jouer plusieurs rôles dans l'apprentissage d'une langue.

Premier prototype : DEVINPAI

Ce logiciel consiste à faire deviner un pays à l'utilisateur en donnant quelques indices. Le nombre d'indices augmente après chaque erreur de l'utilisateur. De plus, si la réponse de l'utilisateur comporte une erreur (ex. : répondre La France alors qu'un indice précise que le pays cherché est en Afrique) l'ordinateur décèle cette erreur et la signifie à l'utilisateur.

Ce petit programme (40 lignes environ) a été réalisé en une dizaine d'heures. Comme nous l'écrivions plus haut, le même programme peut être appliqué à plusieurs matières :

- histoire (découvrir un homme historique)
 - géographie (les villes du Québec)
 - biologie (les organes du corps humain)
 - chimie (les éléments chimiques)
- etc.

Le professeur qui voudrait utiliser ce programme dans sa matière peut entrer lui-même les données, un programme étant conçu à cet effet.

Un autre logiciel qui s'appelle PAYS se sert des mêmes données que le programme DEVINPAI pour faire le processus inverse. C'est l'utilisateur qui a en tête un nom de pays et l'ordinateur essaie de trouver le pays en posant des questions. Ce programme décèle aussi les indications fausses de l'utilisateur.

Deuxième prototype : DICMAGIC

Ce logiciel joue plusieurs rôles dans l'apprentissage du langage ou de l'orthographe.

- On fait une dictée au sens habituel du mot. Il faut corriger les fautes d'un texte donné par l'ordinateur. Ce texte est entré par le professeur grâce à un programme spécial. Le professeur écrit d'abord le texte avec les fautes qu'il veut présenter à l'élève, puis il corrige ensuite ces fautes.
- On fait accorder des verbes. Les verbes sont présentés entre() à l'infinitif. L'élève doit les accorder.
- On doit écrire correctement des homonymes. Ex. : Écrire correctement les sons « CES SES C'EST SAIS... » dans un texte qui apparaît sur l'écran.
- Dictée en anglais
- Accord des verbes en anglais
- Etc. selon l'imagination des professeurs !!!

Le manque de mémoire, l'absence de couleur, de son et de graphisme représentent des handicaps sérieux pour le développement de logiciels plus sophistiqués, mais ce matériel étant promis pour bientôt, nous pourrions développer, grâce à APL et à ces micros plus puissants des logiciels qui pourront répondre à certains besoins pédagogiques d'une école.

4. Amorcer la formation des professeurs de l'école

Six professeurs de l'école qui avaient choisi l'activité APL dans leur horaire venaient une fois par semaine suivre un cours d'APL. Ils ont ainsi eu la possibilité de s'initier à la programmation, d'analyser les possibilités de l'ordinateur comme outil pédagogique spécialement dans leur matière d'enseignement. Avec le professeur de physique, nous avons développé un logiciel sur le lancement d'un projectile. Tous les étudiants du cours de physique sont venus expérimenter la simulation du lancement d'un projectile à une vitesse et à un degré de lancement de leur choix.

Ces professeurs ont eu une formation légère, c'est-à-dire qu'ils ont pu s'initier mais ils ne sont pas encore autonomes face à un ordinateur. La formation en APL demande plus d'heures de travail car le langage est très riche. Suite à un récent sondage, ils constatent :

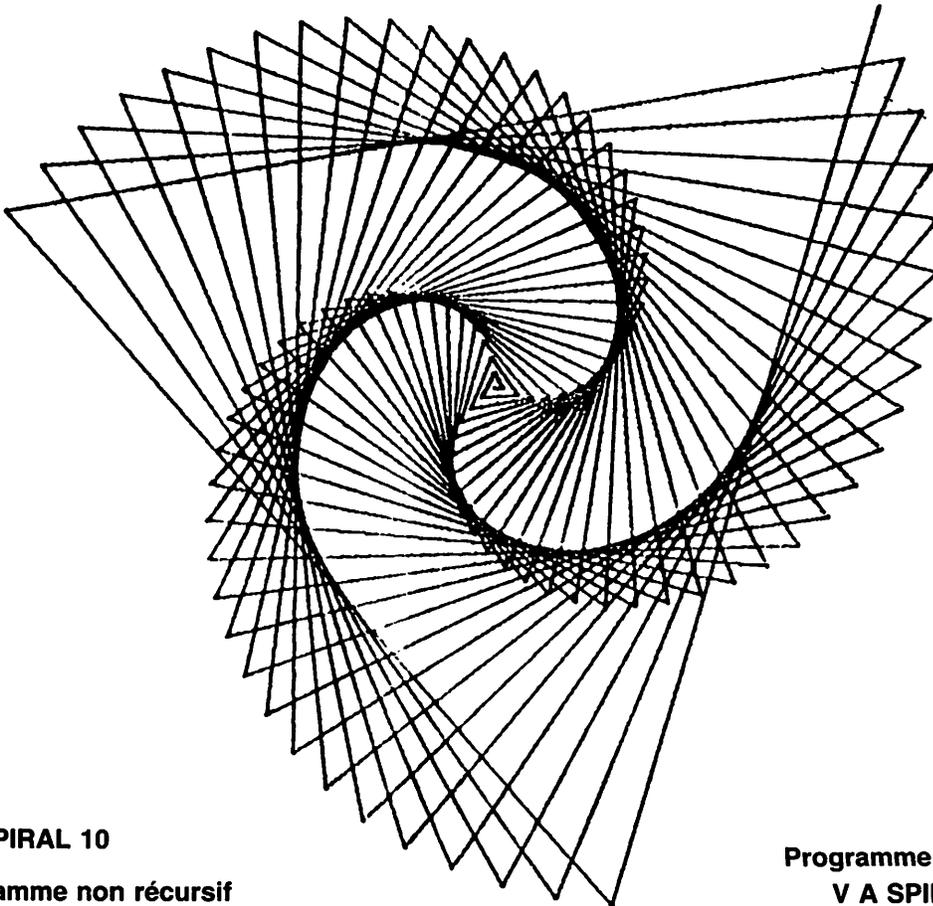
LANGAGE D'AVENIR

- une satisfaction face à l'expérience de programmation
- le manque de temps pour pratiquer
- le sérieux de l'apprentissage d'un langage de programmation
- le goût de poursuivre l'apprentissage de l'APL

Des parents d'élèves de l'école ont également eu la possibilité de s'initier à APL grâce à des cours offerts bénévolement. Ces parents poursuivent maintenant par eux-mêmes leur formation en venant régulièrement tous les jeudis soir dans le local d'informatique que la direction de l'école a la générosité de mettre à leur disposition.

Conclusion

L'expérience a déjà permis de prouver qu'APL était parfaitement accessible aux élèves du secondaire. Au cours de la 2^{ème} année de l'expérience, APL sera utilisé pour le cours ISI et, grâce à des ordinateurs plus puissants, nous comptons réaliser des logiciels bien adaptés aux besoins pédagogiques de l'école.



122 SPIRAL 10

Programme non récursif

```
V A SPIR D
[1] AVANCE D
[2] TOURNE A
[3] 1 SI 1000 ≥ D D + 10
V
```

Programme récursif

```
V A SPIRAL D
[1] O SI D > 1000
[2] AVANCE D
[3] TOURNE A
[4] A SPIRALE D + 10
V
```