

## POUR QUELQUES APPROCHES DE PLUS...

Sylvie PAINVIN, Jean-Michel CAGNAT, Jean-Pierre PEYRIN,  
Inggriani LIEM

En revenant de son cours, un professeur de programmation est perdu dans ses pensées... Des difficultés ont surgi ce matin-là. Il a essayé plusieurs approches, mais aucune ne convenait à tous ; il a consommé beaucoup de craie, les étudiants étaient passifs et ne savaient pas comment poser des questions. Il aurait aimé disposer d'autres moyens, rendre les étudiants plus actifs, leur permettre un travail personnel et des expérimentations de l'objet enseigné. Et il aurait été plus disponible pour les problèmes propres de chacun.

### 1. DES OUTILS POUR APPRENDRE

Enseignant la programmation, nous sommes chaque jour amenés à réfléchir au bien-fondé de notre enseignement : « Ai-je raison d'introduire la notion de récursivité avant celle de liste chinée ? », « Et mes collègues, comment font-ils ? », et « Pourquoi F., G. et P. ont-ils compris tout de suite et pas les autres ? ». Autant d'individus, autant de façons d'apprendre et d'enseigner, semble-t-il... L'ordinateur, qui, dans le domaine de la programmation, est l'outil de travail principal, peut être aussi un outil d'apprentissage pour l'étudiant, notamment pour certaines activités difficiles (voire impossibles) à faire en cours. Disposant d'une grande variété de « logiciels-outils » adéquats, en complément des autres formes d'enseignement, l'étudiant pourrait apprendre la programmation de manière plus personnelle et plus vivante.

Comme ce sont la plupart du temps les bases mal assimilées qui empêchent les apprentissages ultérieurs et plus complexes, nous avons commencé à travailler sur les concepts et méthodes de base. Ainsi, à la formulation près et les outils se voulant simples d'emploi, la plupart des activités que nous proposons peuvent être pratiquées par des étudiants de toutes disciplines et même par des lycéens.

Un outil permet d'effectuer certaines activités. Pour nous, une activité, ce n'est pas taper sur une touche ou lire un message, actions qui sont trop élémentaires, ou réfléchir, mémoriser, actions qui sont trop vagues. Ce serait plutôt un ensemble d'opérations élémentaires et concrètes, permettant d'atteindre une certaine maîtrise du sujet traité. Exemple : lire un programme avec un certain regard ; exécuter un programme en mode démonstration ; manipuler ou transformer son code...

Un outil permet d'effectuer certaines activités et pas d'autres. Cette idée est fondamentale : nous avons choisi de réaliser de petites entités, faciles d'accès et d'emploi, très ciblées, sachant qu'il faut trouver un bon équilibre entre la richesse d'un environnement et le confort qu'il procure (sécurité, maîtrise, efficacité).

À la base de la conception d'un outil, il y a nos expériences d'enseignement en algorithmique et en programmation, dans d'autres thèmes de l'informatique, et nos propres expériences d'apprentissage. Il y a aussi la connaissance de l'existant. Mais il y a aussi nos intuitions (qui viennent de nos expériences de vie) et notre créativité. Nous voulons prendre le risque de nous tromper, proposer des activités « inédites », voire inhabituelles, si elles nous paraissent utiles. Autant de sensibilités différentes, autant de façons de traiter un sujet.

Les « logiciels-outils » que nous avons réalisés permettent à l'étudiant d'apprendre, plus qu'ils n'enseignent. En effet, l'accent est mis sur l'activité de l'étudiant et non sur celle de l'enseignant.

## **2. QUELQUES EXEMPLES D'OUTILS POUR UN THÈME : LE TRI**

Pourquoi ce thème, déjà très étudié et ne présentant pas de grosses difficultés ? Précisément parce qu'il existe beaucoup d'exemples de tris dans la vie courante et qu'il permet de ce fait d'aborder facilement l'algorithmique. Ce thème est une mine de nombreux algorithmes basés sur des approches différentes.

Nous proposons à l'étudiant quatre outils (réalisés sur Apple Macintosh) mettant en œuvre diverses activités : manipulation d'objets et de valeurs, visualisation de processus, lecture d'algorithmes et de programmes, comparaison d'algorithmes, exécution de programmes. Ces logiciels concrétisent différentes approches, parmi lesquelles l'étudiant

peut choisir selon ses goûts, ses incompréhensions, ses doutes, sa façon d'apprendre...

## 2.1. Meccano de tri

Ce logiciel met à la disposition de l'étudiant un ensemble d'outils permettant de réaliser des tris d'objets représentés à l'écran par des cubes. L'objectif essentiel est de mettre en évidence le rôle de manipulations concrètes dans le processus d'apprentissage. Le tri, de par son existence dans la vie quotidienne, convient bien à cet objectif. Dans une salle de cours, on demande parfois aux étudiants de trier effectivement des objets concrets : cartes à jouer, devoirs notés (ordre de mérite ou ordre alphabétique), petits cartons portant des numéros. Cette approche est très fructueuse et permet de « découvrir » bon nombre d'algorithmes classiques.

Toutefois, en plus des difficultés matérielles de distribution des objets et des sources de distraction qu'ils procurent, on constate que les techniques mises en œuvre sont parfois assez peu algorithmiques. Il est en particulier très gênant que tous les objets aient des valeurs visibles ; un coup d'œil suffit en effet à repérer le minimum, voire même à mémoriser la position de plusieurs valeurs. Il est également difficile de « structurer » une idée d'algorithme.

Le logiciel proposé permet en partie de répondre à ces problèmes :

- Le logiciel tient lieu de matériel.
- La pratique individuelle se fait dans le calme et peut avoir lieu en dehors des cours.
- Les valeurs des objets sont invisibles.
- La manipulation se faisant par l'intermédiaire exclusif des outils fournis, il devient possible de forcer l'emploi d'opérations ayant un sens plus algorithmique.

Les options de fonctionnement sont peu nombreuses :

- un menu permet le choix du nombre de cubes : 9 (suffisant pour la plupart des algorithmes) ou 18 (pour confirmer une très bonne pratique d'un algorithme) - un deuxième menu permet de sélectionner les outils avec lesquels on désire travailler :

+ socle de travail

+ deuxième ligne de socles

+ marquage : donne la possibilité de marquer, ou peindre, les cubes de deux façons différentes et aussi de les démarquer.

+ échange : permet d'échanger deux cubes préalablement sélectionnés.

+ plus petit de 2 : permet de trouver quel est celui des deux cubes sélectionnés qui contient la plus petite valeur.

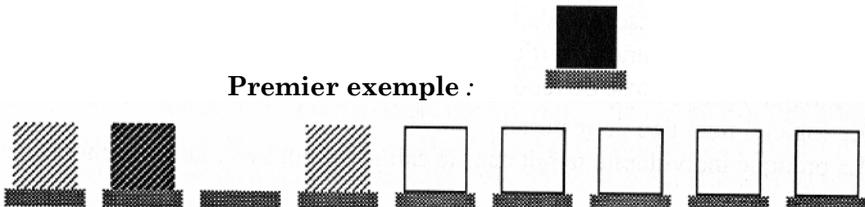
+ plus petit de  $n$  : comme le précédent, mais fonctionne pour deux cubes ou plus.

L'étudiant peut choisir librement des outils parmi ceux qui lui sont proposés. Il doit ensuite travailler avec les outils qu'il a choisis. Selon les choix faits, il devient possible (ou non) de faire, plus ou moins rapidement, des tris suivant tel ou tel algorithme : tri par minimums (ou par maximums) successifs ; tri par insertion ; tri par bulles ; tri par fusion ; tri de Shell ; etc.

Une approche par niveaux descendants est possible en jouant sur le choix d'outils (par exemple, « plus petit d'un nombre quelconque de cubes » dans un premier temps, puis dans un deuxième temps « plus petit de 2 cubes »).

L'étudiant peut également revenir à l'état initial sans changer les valeurs, afin de recommencer ou d'essayer un autre algorithme, ou en changeant les valeurs contenues dans les cubes. Enfin, il existe une possibilité de faire apparaître les valeurs des cubes (mais dans ce cas aucune manipulation n'est autorisée tant que les valeurs sont visibles).

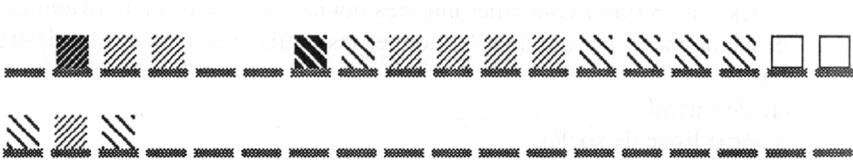
Notons que l'étudiant reste libre dans ses manipulations. On ne cherche ni à le guider vers une solution, ni à vérifier si ses manipulations sont conformes à un algorithme particulier.



*Cette situation peut apparaître au cours d'un tri par insertion. Les 3 cubes hachurés ont été déjà triés. On cherche la position du cube qui se trouve sur le socle de travail par rapport  $G$  ces 3 cubes. Une première*

comparaison a permis de décider que ce nouveau cube devait être avant le cube hachuré le plus à droite ; ce dernier a donc été déplacé d'une position vers la droite. On est prêt maintenant à comparer les valeurs des deux cubes sélectionnés.

Deuxième exemple :



Ce deuxième exemple, avec 18 cubes, illustre une phase d'un tri (itératif) par fusion. Les phases précédentes ont déjà permis de trier des sous-groupes de 4 cubes adjacents ; ces sous-groupes sont repérés par des marquages différents. On est ici en train d'interclasser les deux premiers groupes de 4 cubes. Le premier cube du premier groupe et les deux premiers cubes du deuxième groupe ont déjà été placés dans le bon ordre sur la ligne du bas. Il s'agit maintenant de décider lequel des deux cubes sélectionnés (en sombre) doit être transféré sur la ligne du bas.

Il est intéressant de remarquer que ce petit nombre d'opérateurs suffit pour mettre en œuvre un grand nombre d'algorithmes de tris classiques. Il faut écarter cependant les algorithmes récursifs et ceux qui travaillent sur une structure arborescente.

Différents usages pédagogiques sont envisageables, tels que l'expérimentation libre ; l'essai de reproduction d'un algorithme (défini par son nom ou par une description de son principe), avec ou sans contraintes ou suggestions sur le choix des opérateurs.

## 2.2. Tri Guidé

Ce logiciel propose plusieurs modes de travail, dont la combinaison permet plusieurs démarches d'apprentissage d'un algorithme simple de tri. Nous voulons guider un processus d'apprentissage relativement autonome en proposant différents degrés de liberté et différents niveaux de détails.

Le logiciel fournit des objets et des opérations sur ces objets choisies de façon à permettre la mise en œuvre d'un algorithme précis. L'étudiant peut alors se placer dans différents modes de travail :

- démonstration automatique : il voit fonctionner l'algorithme, sans intervention de sa part.

- mode suggéré : seule l'opération nécessaire est activable ; elle est mise en évidence. L'idée est de faire découvrir l'enchaînement des opérations tel qu'il est dans l'algorithme.

- mode vérifié : toutes les opérations sont disponibles, mais seule la bonne opération sera acceptée. Il s'agit ici de s'assurer que l'ordre des opérations a été bien compris.

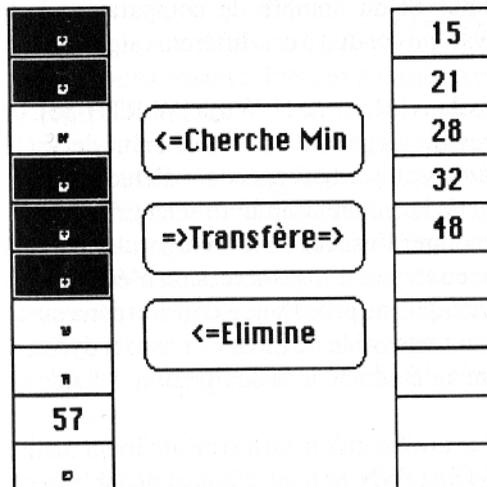
- mode libre : tout ce qui n'est pas absurde est permis, que cela corresponde ou non à l'algorithme. L'étudiant teste ainsi s'il est capable de répéter correctement tout l'algorithme, sans aucune aide.

L'algorithme choisi est un tri par minimums successifs avec un tableau de Données et un tableau de Résultats. Des opérateurs placés entre les deux tableaux permettent d'effectuer les opérations suivantes :

- rechercher le minimum dans le tableau Données ;

- transférer un élément du tableau Données dans le tableau Résultats (avec remplissage de haut en bas du tableau Résultat) ;

- éliminer la valeur sélectionnée dans le tableau Données (la case passe visuellement en noir ; son contenu est remplacé par une « grande » valeur).



*Les 5 plus petites valeurs ont déjà été transférées dans le tableau de droite. L'opérateur "Cherche Min" vient de faire apparaître la valeur 57 qui est le minimum des valeurs restant dans le tableau de gauche.*

*Il suffit d'appliquer "Transfère" pour copier la valeur 57 juste après la valeur 48 dans le tableau de droite. L'opérateur "Elimine" fera alors disparaître la valeur 57 du tableau de gauche et mettra sa case en noir.*

Un deuxième niveau est également proposé, dans l'optique d'une approche descendante. Il force à détailler la réalisation de l'opération « Recherche du minimum ». Les quatre modes de travail décrits ci-dessus sont disponibles (pour chacun des deux niveaux). Il est de plus toujours possible de revenir à l'état initial (avec ou sans changement des valeurs contenues dans le tableau de départ.)

Ce logiciel est moins riche dans ses usages pédagogiques que le Meccano de Tri, dans la mesure où il ne porte que sur un seul algorithme, élémentaire de surcroît. Il est toutefois intéressant de remarquer que l'étudiant reste libre de ses manipulations et qu'il peut suivre différentes démarches. En voici quelques-unes, à titre d'exemples :

- Essayer en mode libre. En cas d'échec, essayer en mode vérifié ; ou même en mode suggéré ; et en mode démonstration, en dernier recours.

- Observer la démonstration automatique ; reprendre en mode suggéré, puis en mode vérifié, et, pour finir, en mode libre.

- Observer la démonstration automatique. En mode libre, essayer de reproduire l'algorithme. En cas de difficulté, essayer en mode vérifié. Si les difficultés persistent, essayer en mode suggéré.

Dans tous les cas, il convient de reprendre ensuite avec de nouvelles valeurs, puis de recommencer avec le deuxième niveau.

### **2.3. Tris internes**

Ce logiciel permet d'une part l'observation de l'exécution des algorithmes de tris les plus courants et d'autre part leur comparaison (en termes du temps d'exécution, du nombre d'affectations et du nombre de comparaisons). Il n'aborde pas les problèmes de l'analyse qui conduit à ces différents algorithmes. Les tris choisis sont ceux présentés dans le

livre de N. Wirth [WIRTH-86] : tri par insertion, tri par minimums successifs, tri par bulles, tri shaker, tri de Shell, tri par tas, tri rapide. Ce choix est arbitraire ; il permet cependant d'étudier le rôle d'un tel logiciel comme complément de la lecture d'un livre scientifique.

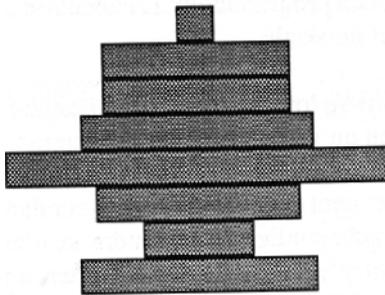
Il est difficile, dans un livre, de montrer l'exécution d'un algorithme. Il est éventuellement possible de la décrire en termes d'une succession d'états sur un petit nombre de jeux d'essais caractéristiques au prix d'une extension démesurée du texte. Il est en tout cas parfaitement impossible de montrer l'aspect dynamique de cette exécution et de permettre à l'étudiant de la comprendre à l'aide de situations non prévues par l'auteur.

La représentation visuelle d'un tableau a été tirée d'un travail antérieur réalisé à l'Université Brown aux États-Unis [BROWN-85]. Un élément du tableau est symbolisé par un trait d'une longueur proportionnelle à la valeur de cet élément.

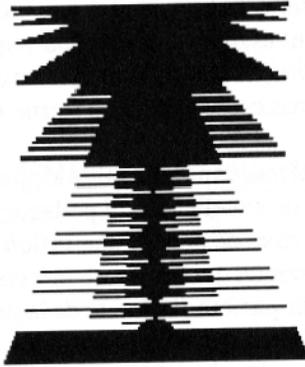
Un tableau est donc symbolisé par une séquence de traits. Ainsi l'exécution du tri est visualisée par les déplacements des traits.

Le logiciel réunit deux applications :

a) La première application permet de commander l'exécution d'un algorithme de tri : l'étudiant choisit un algorithme ; il choisit également les caractéristiques des données (aléatoires, ordonnées, inversées, égales ou bien saisies graphiquement par lui-même) ainsi que le nombre d'éléments. Il décide de la vitesse de l'exécution. Il peut recommencer autant de fois qu'il le désire un tri sur un jeu de données. Il peut à tout moment changer l'algorithme et le jeu de données.



*Un exemple d'image d'un tri de 8 éléments.*



*Un exemple d'image du tri par tas sur 150 éléments classés au départ par ordre croissant. On observe la destruction de l'ordre initial par la formation des tas et la reconstruction du tableau.*

Nous avons observé, lors des premières expérimentations, deux formes principales d'utilisation.

La première forme consiste à travailler sur un petit nombre d'éléments (de l'ordre d'une dizaine) et à faible vitesse (de l'ordre d'une opération élémentaire par seconde). Cette forme convient à l'étudiant qui désire comprendre le processus séquentiel de l'exécution dans ses moindres détails.

La deuxième forme consiste au contraire à travailler sur un grand nombre d'éléments et à grande vitesse (le tri rapide d'un tableau rempli aléatoirement avec 256 valeurs est effectué en une trentaine de secondes). Cette forme convient à l'étudiant qui désire comprendre le principe de transformation du tableau dans sa globalité.

b) La seconde application permet de comparer les algorithmes de tri à l'aide d'une évaluation du temps d'exécution et des nombres de comparaisons et d'affectations. L'évaluation est présentée par des histogrammes : on observe soit différents algorithmes pour un seul type de données, soit un seul algorithme pour plusieurs types de données. Pour éviter un long temps d'attente (l'ensemble des tris pour un tableau de 2 048 éléments sur l'ensemble des types de données nécessiterait plus de quatre heures), les évaluations sont pré-enregistrées pour des valeurs prédéfinies ; les tris sont effectivement exécutés pour des données saisies par l'étudiant.

## 2.4. La gare de triage

Il s'agit d'un dessin animé (réalisé à l'aide de VideoWorks), sérieux et humoristique à la fois, centré sur l'algorithme de tri par minimums successifs. L'étudiant assiste au triage des wagons d'un train par deux cheminots, et à la formulation de cette activité par un analyste et un programmeur. L'exécution du programme est rendue sous forme visuelle et musicale.

Le cinéma, dans ses multiples formes (court ou long métrage, dessin animé), peut être un excellent outil pédagogique. Bien qu'il ne permette pas l'interactivité, il permet de donner une vision globale d'un phénomène, dans une présentation continue qui lui assure une cohérence et peut entraîner le spectateur dans une pensée parfois différente de la sienne. Ce qui est utile, car apprendre, ce n'est pas seulement contrôler, savoir consciemment ; c'est aussi se laisser porter, immerger, sans toujours comprendre [REBOUL-80]. Lequel d'entre nous n'a pas eu le sentiment d'avoir « appris » quelque chose d'un film, d'en être sorti changé, différent, enrichi ?

Les psychologues ont d'ailleurs depuis longtemps mis en évidence l'efficacité de l'identification à autrui, de l'assistance à des actions, ou encore de la catharsis dans le développement des personnes.

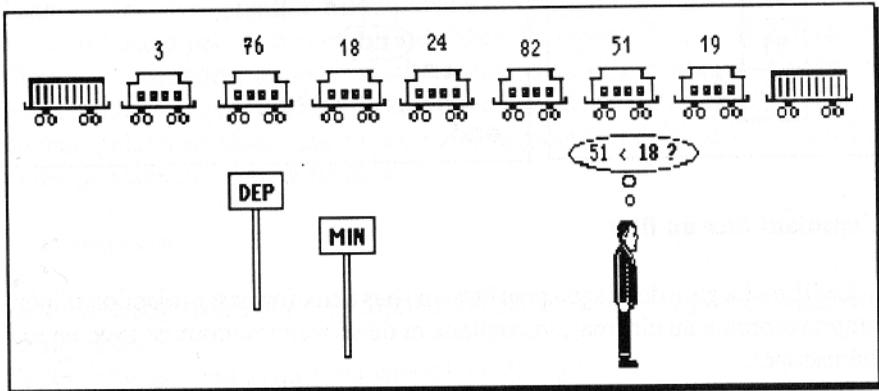
Dans les disciplines scientifiques, l'étudiant apprend surtout en lisant, en observant, en réfléchissant, en prouvant. Pourtant, apprendre met en jeu plusieurs sens simultanément et repose aussi sur des processus inconscients. Si la formulation verbale extériorisée ou mentale est essentielle à l'intégration d'une idée, l'image (schématique, réaliste ou mentale) et le son (musique, parole ou bruit) favorisent la motivation, la compréhension et la mémorisation [DENIS80], [MALONE-80]. Et le rôle de l'art, de la musique, de l'humour, de l'imaginaire, de l'analogie dans les apprentissages et acquisitions de connaissances est sans doute plus important qu'il n'est actuellement reconnu.

### Le film

Le thème du tri par minimums ayant été retenu, le choix a été fait de faire ressortir l'essentiel du processus de recherche de la valeur minimum, et de mettre l'accent sur cette action en la distinguant nettement de celle d'échange des valeurs. Les cas particuliers (cas de valeurs égales, cas où le premier élément de la liste des valeurs est le

minimum) n'ont pas été visualisés et le processus d'échange des valeurs n'a pas été développé.

La liste des valeurs est représentée par une suite de wagons de voyageurs, les fourgons de tête et de queue en marquant les bornes. La recherche des valeurs minimums successives est représentée par un cheminot qui parcourt le train de la gauche vers la droite. Il plante et déplace, suivant l'algorithme, les piquets de marquage DEP (le début de la suite à trier) et MIN (le minimum rencontré).

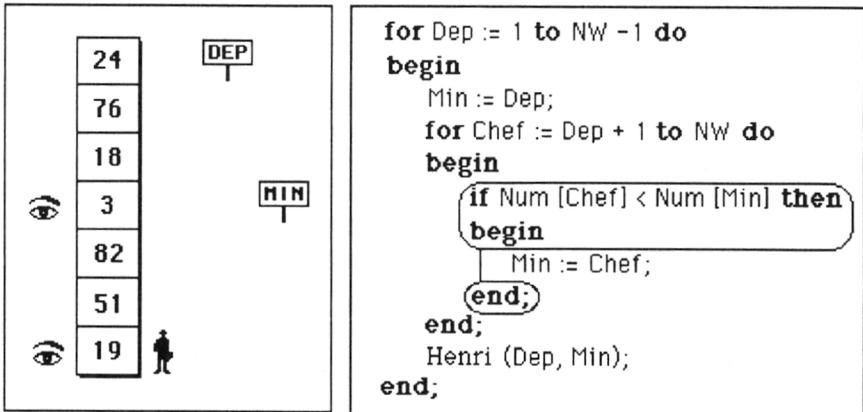


L'échange des valeurs, à la fin de chaque parcours, est représenté par un autre cheminot qui déplace les wagons. Celui-ci n'exécute pas la procédure d'échange telle qu'elle peut être programmée en Pascal. Seul le résultat de l'échange compte ici.

L'analyse de cette scène de tris de wagons est faite par un étudiant en programmation. Il la commente en isolant 6 actions différentes du chef cheminot planter le piquet DEP, appeler Henri pour l'échange des wagons, planter le piquet MIN devant le nouveau minimum, planter le piquet MIN devant le piquet DEP, s'arrêter devant chaque wagon, comparer les numéros des wagons. Un personnage intervient pour lui en proposer une traduction en langage Pascal. L'assemblage de ces actions produit le corps de la procédure du tri par minimums successifs. Chaque action du cheminot est représentée sous une forme visuelle (le chef marche, plante, réfléchit), verbale écrite (en français courant, en Pascal, en notation algorithmique) et sonore (des sons et des notes de musique). L'exécution finale de la procédure rassemble certains de ces aspects sous forme d'un concert : chaque instruction exécutée est accompagnée d'effets visuels et sonores ; elle est encadrée ; l'état des valeurs, actions et variables est affiché ; des notes de musique particulières lui sont

LE BULLETIN DE L'EPI POUR QUELQUES APPROCHES DE PLUS...

associées. La succession des instructions fait entendre une mélodie et apparaît une animation.



L'étudiant face au film

Le film « La gare de triage » peut être vu sous deux formes : projection ininterrompue (comme au cinéma) ; visionnement de séquences (comme avec un magnétoscope).

Dans le premier cas, l'étudiant assiste à la projection, l'écran de l'ordinateur étant transformé en écran de cinéma. Ses degrés d'initiative sont très limités : il peut arrêter le film à tout instant, ou le revoir à volonté.

Dans le second cas, le film ayant été fragmenté en 4 petits films, l'étudiant choisit les séquences du film qu'il veut visionner. Il peut aussi, entre deux séquences, utiliser l'un des autres logiciels sur le tri. Pour chaque film, il dispose d'un tableau de commande (analogue à celui d'un magnétoscope) et il peut ainsi revenir en arrière, sauter des étapes, changer la vitesse de projection, couper le son, etc. Il a le temps de s'attarder sur un détail de l'algorithme ou sur une explication, et il peut prendre des notes.

L'étudiant est essentiellement placé en situation d'observateur. Il assiste à six « activités » liées à la programmation : le tri en situation concrète, l'analyse d'un problème par décomposition, le recours à une notation algorithmique, le codage d'actions repérées, la rédaction d'un programme, son exécution. En les observant, il pratique mentalement ces activités : il peut anticiper les actions montrées, les contester, imaginer une autre visualisation et d'autres algorithmes de tri.

## CONCLUSION

La didactique se préoccupe à la fois de la théorie et de la méthode de l'enseignement. Dans le domaine de la programmation, nous avons suivi avec intérêt les travaux de Dijkstra, Wirth, Arzac... qui, en étudiant l'objet programmation, ont su en tirer des conclusions pour son enseignement, sur un plan théorique et pratique. Nous nous sommes aussi référés aux travaux locaux [SCHOLL-87], et les « logiciels-outils » proposés en constituent certainement un complément utile.

Notre travail sur le tri (et sur d'autres thèmes tels que : la récursivité ou la spécification d'un problème) nous a permis de tester cette complémentarité, mais nous y ajoutons l'idée que l'ordre « a priori » de présentation d'un concept ou d'une méthode n'est pas foncièrement déterminant, les étudiants (et l'enseignant) ayant leurs propres schémas mentaux de compréhension.

L'ordinateur peut offrir cette part de liberté, au même titre que les livres, les films, les rencontres, les expériences de vie, etc. qui appartiennent à chacun. Nous souhaitons que l'étudiant puisse effectuer les activités de programmation en manipulant, en observant, en écoutant, en jouant, et qu'il structure ainsi de façon personnelle ses acquisitions.

Sylvie PAINVIN,  
Jean-Michel CAGNAT,  
Jean-Pierre PEYRIN,  
Inggriani LIEM

## RÉFÉRENCES

[BROWN-85] BROWN M., SEDGEWICK R. « Techniques for Algorithm Animation »,

IEEE Software, vol 2, no 1, janvier 85, pp 28 - 39.

[CAGNAT-88] CAGNAT JM., GUERAUD V., LIEM I., PAINVIN S., PEYRIN J. « Un laboratoire pour l'enseignement de la programmation » Colloque sur l'évolution de l'outil informatique à l'Université. Poitiers, Septembre 88.

[DENIS-80] DENIS M. « Les images mentales » P.U.F. (1980)

[GUERAUD-88] GUERAUD V., PEYRIN J.-P. « Un jeu de rôles pour (enseignement de la programmation », Colloque francophone sur la didactique de l'informatique. Paris, Septembre 1988

[MALONE-80] MALONE T.W. « What Makes Things Fun To Learn ? A Study Of Intrinsically Motivating Computer Games » XEROX Palo Alto Research Center (1980)

[REBOUL-80] REBOUL O. « Qu'est-ce qu'apprendre ? » P.U.F. (1980)

[SCHOLL-87] SCHOLL PC., PEYRIN JP. « Schémas algorithmiques fondamentaux », Publications IGEL, Université J. Fourier (Grenoble I) (1987)

[WIRTH-86] WIRTH N. « Algorithmme and Data Structures », Prentice Hall (1986)

Sylvie PAINVIN, Jean-Michel CAGNAT, Jean-Pierre PEYRIN, Inggriani LIEM. Projet ARCADE, Laboratoire de Génie Informatique, IMAG-Campus, BP 53 X - 38041 GRENOBLE cedex