

# La méthode CAR : le codage pour toutes et tous

Rupert Meurice de Dormale

fadagogo.com  
tsd62525@scarlet.be

## Résumé

*Le codage permet de développer des capacités intellectuelles intéressantes qui auront des retombées directes sur les facultés d'expression, de lecture et d'apprentissage de nos élèves. Le choix de la méthode est essentiel ; derrière le strass doivent se trouver de réelles qualités pédagogiques car nous avons l'obligation de réussir pour la toute grande majorité de nos élèves. Dans ce cadre, la méthode CAR (Computers Are Robots) est présentée, ses caractéristiques et avantages essentiels sont décrits. Cette méthode a été utilisée durant près de vingt ans pour enseigner avec succès le codage à des élèves de 16 à 18 ans.*

**Mots clés :** apprendre à coder, robotique virtuelle, choix des méthodes de codage, abstraction, algorithmique, CAR (Computers Are Robots)

## 1 Introduction

L'objectif de l'apprentissage du codage à nos jeunes élèves n'est pas de former des cohortes de programmeurs. La formation à ce métier est l'apanage de l'enseignement supérieur, qui laissera les moins aptes sur le bord du chemin. Notre but, à travers le codage et pour la très grande majorité de nos élèves, est de développer leurs facultés d'analyse, d'abstraction, d'anticipation et de logique. La pratique de l'algorithmique aura des retombées directes sur la maîtrise du langage naturel. Communiquer, c'est construire son raisonnement dans le cerveau de l'autre : l'émetteur doit coder sa pensée pour que le récepteur puisse la décoder. Un des intérêts majeurs du codage est d'améliorer les capacités de l'élève confus, mauvais lecteur, ayant souvent des difficultés en grammaire, mathématiques et sciences, toutes les matières faisant appel à la construction de règles. Notre objectif, bien au-delà de la sensibilisation au numérique, est d'apporter des solutions à nos élèves en « souffrance logique ».

## 2 Attentes

Ceci implique que la méthode utilisée donne leur chance aux plus démunis. Parmi l'explosion des offres, la question du choix est complexe. L'aspect ludique, les belles couleurs et la Reine des Neiges en lieu et place de la tortue sont séduisants, mais les caps difficiles à franchir sont la séquence, les branchements (dans les structures conditionnelles et répétitives), les opérateurs logiques et la notion de variable. Il serait également utile qu'une seule et même méthode englobe, de manière cohérente et progressive, l'ensemble de ces difficultés dans un cursus continu partant de zéro jusqu'à la maîtrise des variables indicées.

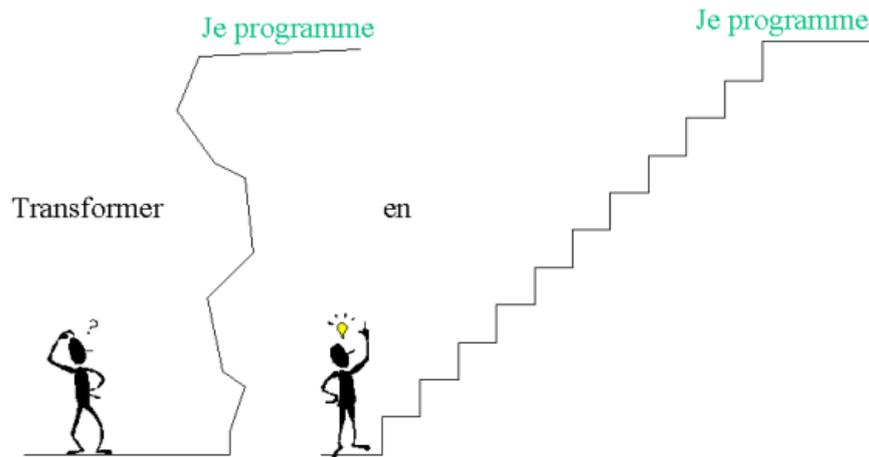


FIGURE 1 : la qualité d'un système d'apprentissage se juge à la progressivité des concepts transmis

### 3 La méthode CAR

Pour répondre à ce cahier des charges, la méthode CAR (Computers Are Robots) va s'appuyer sur un certain nombre de piliers dont voici les essentiels.

a. L'ordinateur est considéré comme un « robot à traiter de l'information », ce qui va permettre de mettre en place les bases de l'algorithmique *avant* d'aborder la gestion des variables.

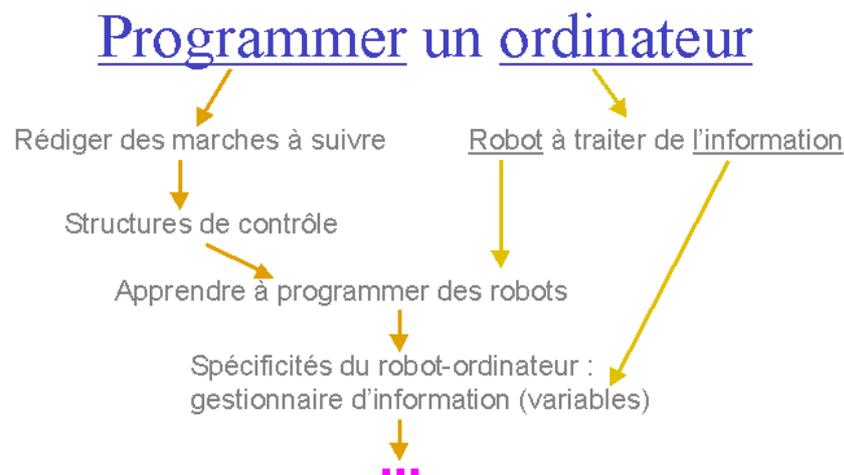


FIGURE 2 : scinder les difficultés pour faciliter le cursus

b. « Donnez-moi un point d'appui, et je soulèverai le monde. » (Archimède). Mais quel point d'appui fournir à un élève ayant de faibles capacités logiques et d'abstraction ? Cet élève connaît déjà des « tas de choses » qui sont essentiellement concrètes. La mise en jeu de la programmation de petits robots virtuels, dont les capacités et les tâches sont clairement définies, va lui permettre, lors d'essais et erreurs, de naviguer simultanément dans le concret de l'évolution du robot et l'abstrait de l'exécution de sa marche à suivre (son code).

c. L'élève est seul (ou accompagné s'il le désire) devant la tâche à accomplir et peut, autant de fois qu'il le veut et sans être jugé, explorer ses hypothèses jusqu'à trouver la ou les bonnes solutions. Ses erreurs, en tant que contre-exemples, sont formatives car l'univers de référence dans lequel il se trouve est bien défini et stable.

d. En cas d'erreur de codage, l'élève doit visualiser que, lorsque l'exécution de son programme arrive précisément à ce niveau, la tâche réalisée par le robot dérape et mène à une action indésirable. Il pourra ainsi faire un parallèle entre l'action erronée du robot et la ligne correspondante de son code. Ceci est particulièrement important dans

l'exécution des structures de contrôle qui engendrent des branchements vers d'autres lignes du code. La maîtrise de ces branchements est sans doute la difficulté la plus importante en algorithmique.

Jusqu'ici, on pourrait objecter que la virtualité des robots n'est pas pertinente. Comment être plus dans le concret qu'en utilisant les vrais engins de la robotique pédagogique ? Il y a à cela deux réponses, l'une pratique et l'autre stratégique. Au niveau pratique, les robots virtuels ont un coût nul une fois conçus, sont moins complexes à mettre en œuvre, sont facilement (trans)portables et permettent de développer des situations ou univers quasiment infinis. Stratégiquement, ces univers variés vont permettre de rester cohérent dans l'abord des variables. Chaque variable sera une ardoise qu'il faut effacer avant d'y écrire le nouveau contenu, ce qu'un robot virtuel pourra faire dans de multiples situations. L'élève peut aussi être seul et non face à un groupe et son erreur est clairement située dans les situations plus complexes des structures de contrôle. Ce serait par ailleurs une erreur d'assimiler la virtualité à l'abstraction. Le concret ne signifie pas qu'il faille toucher, je peux très bien observer une situation concrète de ma fenêtre tout en comprenant parfaitement ce qui se passe.

e. La maîtrise de l'algorithmique étant acquise, la notion de variable est abordée. Comme toujours, certains élèves auront compris le principe en quelques secondes alors que d'autres risquent de ne pas comprendre ou, plus grave, de développer des concepts corrompus du fonctionnement des variables : le fait que la variable s'efface quand on la lit, le fait de pouvoir faire appel à un ancien contenu, penser qu'une variable s'initialise automatiquement, qu'elle peut contenir plusieurs valeurs... L'analogie retenue pour éviter toutes ces mauvaises croyances et la bonne vieille ardoise (dont il faut maintenant expliquer le fonctionnement...). Cette analogie canalise l'abstraction de l'élève en l'empêchant de développer des idées erronées concernant le fonctionnement des variables.

L'essentiel du cursus est maintenant terminé, il peut être arrêté ou poursuivi, mais les objectifs de base sont atteints, à savoir développer les facultés d'analyse, d'abstraction, d'anticipation et de logique. Si, à ma connaissance, aucune étude rigoureuse n'a encore été menée sur l'impact du codage sur l'esprit logique des élèves, vingt ans d'expérience, avant et après la méthode CAR, me confortent dans l'idée que, d'une part, cette méthode est réellement efficace et permet au plus grand nombre de réussir et que, d'autre part, le codage apporte un réel développement des capacités logiques et, en conséquence, a un effet positif sur la structuration du langage et l'acquisition de toutes les matières faisant appel à l'application de règles.

## 4 Conclusions

Notre tâche est d'amener le plus grand nombre d'élèves d'âge jeune à une maîtrise du codage, non pour la capacité en elle-même mais parce qu'à travers cette pratique on vise le développement de facultés intellectuelles essentielles. Parmi l'explosion d'offres, le responsable doit être extrêmement attentif aux qualités pédagogiques des produits choisis, et notamment à leur cohérence et leur progressivité dans le cursus. La méthode CAR proposée a été pensée en ce sens et répond parfaitement aux attentes qu'on peut avoir dans cet apprentissage délicat et complexe qu'est le codage. Cette méthode, utilisée pendant vingt ans avec des élèves de 16 à 18 ans, a montré toute son efficacité.

Vous trouverez cette méthode sur le site <<http://fadagogo.com>> Boutons « Introductions au codage » → « Algorithmique et programmation en pseudocode »

## Références

Barth B. M., *L'apprentissage de l'abstraction*, Paris, Retz

Cunniff N., Taylor R. P., Black J. B., *Does programming language affect the type of conceptual bugs in beginners's programs ? A comparison of FPL and Pascal in Studying the Novice Programmer*, University of Michigan 1989 PP 419 à 429

du Boulay B., *Some difficulties of learning to program in Studying the Novice Programmer*, University of Michigan 1989 PP 283 à 299

du Boulay B., O'Shea T., Monk J. B., *The black box inside the glass box in Studying the Novice Programmer*, University of Michigan 1989 PP 431 à 446

- Duchâteau Ch., *Faut-il enseigner l'informatique à ses utilisateurs ? Communication au 4<sup>o</sup> colloque francophone sur la didactique de l'informatique*, Québec, Formation-recherche en éducation n° 5.38, Publications du CeFIS, Namur (B), Facultés Notre-Dame de la Paix
- Duchâteau Ch., *Images pour programmer*, CeFIS, DET, FUNDP, Namur. Consultable sur Internet à l'adresse <<http://www.det.fundp.ac.be/cefis/publications/doconline.html>>
- Duchâteau Ch., *Robotique-Informatique mêmes ébats, mêmes débats, mêmes combats ?* in Denis B. et Baron G.-L., *Regards sur la robotique pédagogique, Actes du quatrième colloque international sur la robotique pédagogique*, Liège (B), Université de Liège, pp 10 à 33
- Duchâteau Ch., *Socrate au pays des ordinateurs*, revue de l'EPI n° 74, juin 1994 pp 159 à 177
- Linn M. C., Dalbey J., *Cognitive consequences of programming instruction in Studying the Novice Programmer*, University of Michigan 1989 PP 57 à 81
- Meurice de Dormale R., *Robotique virtuelle in Les actes de la 5ème rencontre francophone sur la didactique de l'informatique*, Monastir 1996
- Meurice de Dormale R., *Robotique virtuelle : un environnement pour un apprentissage dynamique de l'algorithmique in Les actes du 5ème colloque international sur la robotique pédagogique*, Montréal 1997
- Putnam R. T., Sleeman D., Baxter J. a. and Kuspa L. K., *A Summary of Misconceptions of High-School BASIC programmers in Studying the Novice Programmer*, University of Michigan 1989 pp 301 à 314
- Spohrer J. C., Soloway E., *Novice mistake : are the folk wisdoms correct ? in Studying the Novice Programmer*, University of Michigan 1989 pp 401 à 416