

Esquisse d'un programme d'informatique pour le Collège

Comme le rappelle le rapport *L'enseignement de l'informatique en France – Il est urgent de ne plus attendre*, publié par l'Académie des Sciences en mai 2013, l'informatique, en ce début de XXI^e siècle, bouleverse de nombreux domaines de l'activité humaine : la plupart des secteurs de l'industrie, le commerce, les services, les transports, la médecine, la communication entre les personnes, les loisirs, etc. Dans les sciences, l'informatique a totalement renouvelé l'instrumentation et elle conduit à une nouvelle forme de pensée, la « pensée informatique », où les algorithmes sont placés au même niveau que les équations. Les philologues, les historiens, les juristes, etc. utilisent des ordinateurs pour archiver et indexer des corpus de grande taille, mais aussi pour assembler des fragments de statues ou de manuscrits afin de les reconstituer. Dans les arts, de nouveaux modes de création apparaissent, ainsi que de nouveaux modes de diffusion des œuvres. Les sportifs utilisent une instrumentation informatique complexe pour augmenter leurs performances, etc.

Cette place que l'informatique a prise dans ces domaines de l'activité humaine explique l'urgence d'un enseignement de l'informatique, comme l'ont compris de nombreux pays développés et émergents. Cet enseignement doit permettre à tous les élèves – y compris, et surtout, ceux qui ne deviendront pas informaticiens – de comprendre le monde qui les entoure et d'accéder aux nouvelles formes de pensée qui accompagnent le développement de l'informatique. Il doit être doublement au service des élèves, citoyens et futurs professionnels. Au collège, tout particulièrement, il doit éclairer l'orientation des élèves. L'actuel déficit de vocations pour les métiers de l'informatique, qui impacte sérieusement notre compétitivité, peut en effet en partie s'expliquer par l'absence d'enseignement de l'informatique au moment où les élèves choisissent leur orientation.

L'Académie des Science distingue trois modes d'apprentissage de l'informatique, qui correspondent à trois phases successives de cet apprentissage, même si elles se recoupent partiellement : la découverte, l'acquisition de l'autonomie et la maîtrise des concepts. Si l'École est le lieu privilégié de la première phase, la découverte, et le Lycée et l'Université ceux de la troisième, la maîtrise des concepts, le Collège, quant à lui, est celui de la deuxième : l'acquisition de l'autonomie.

L'acquisition de l'autonomie

La constitution de soi-même comme sujet, c'est-à-dire comme acteur, et non simplement spectateur, du monde, demande d'être capable d'agir sur lui et de le transformer. Quand ce monde est peuplé d'objets informatiques, ordinateurs, programmes, téléphones, etc., la capacité d'agir repose en grande partie sur la capacité à construire soi-même de tels objets. Les adolescents quittent ainsi le statut de simples utilisateurs du monde technologique contemporain et prennent peu à peu conscience du rôle créatif et innovant qu'ils peuvent jouer dans ce monde. Parmi ces objets, certains demandent des moyens matériels et conceptuels importants pour être construits. Ainsi, il est impossible, au Collège, de faire construire un microprocesseur ou un ordinateur de toutes pièces aux élèves. En revanche, parce qu'ils sont immatériels, écrire des programmes, demande beaucoup moins de moyens – c'est d'ailleurs une des clés qui expliquent la rapidité de la troisième révolution

industrielle – et il est possible de faire écrire des programmes aux élèves dès l'École ou le Collège.

Il n'est pas difficile de distinguer les personnes qui ont suivi un apprentissage de la programmation ou non dans leur rapport aux objets informatiques : pour ces dernières ce sont des objets opaques, souvent inquiétants, parfois personnifiés ou fétichisés, pour les premières, ce sont des objets transparents et utiles, que ces personnes auraient pu elles-mêmes concevoir. L'apprentissage de la programmation aide ainsi les jeunes à affronter sans crainte des systèmes dont l'apparente complexité pourrait les rebuter.

L'apprentissage de la programmation est aussi un moyen de découvrir les rudiments de la pensée informatique, c'est-à-dire des modes de pensée propres aux informaticiennes et informaticiens. Dès que l'on apprend à programmer, on comprend en effet que certains problèmes, comme vérifier la présence d'un visage dans une image, qui semblent simples à résoudre, sont très difficiles à programmer, contrairement à d'autres, comme vérifier la présence de la lettre « a » dans un mot. Une nouvelle signification de l'adverbe interrogatif « comment » émerge, où la réponse à une question formée avec cet adverbe est un algorithme.

Des compétences fondamentales

L'enseignement de l'informatique, et notamment de la programmation, permet de développer des compétences que l'entrée dans le monde informatisé a rendu centrales.

- *Modéliser* : un élève qui écrit, par exemple, un programme qui permet de mettre un nom commun au pluriel, exprime sous la forme d'un algorithme formulé dans un langage formel un ensemble de règles de grammaire, qui indiquent quels mots prennent un « s » au pluriel, quels mots prennent un « x », quels mots sont invariables, etc. Il apprend ainsi à utiliser un langage formel pour décrire un phénomène : il apprend à modéliser.

- *Collecter des données en vue d'un traitement* : un élève qui écrit, par exemple, un programme qui permet d'effectuer des transformations d'images, doit collecter des exemples d'images sur lesquelles exécuter son programme. Il doit pour cela trouver une source d'images : le site web d'un musée, des photos qu'il a prises, etc. Il doit s'interroger sur le format dans lequel ces images sont exprimées, sur la taille des données à traiter, sur la licence sous laquelle ces images sont distribuées, sur les éventuels problèmes de protection de la vie privée que pose la collecte de ces images.

- *Abstraire* : un élève qui écrit, par exemple, un programme qui permet de trouver la sortie d'un labyrinthe doit s'interroger sur la manière de représenter son labyrinthe. Rapidement il s'apercevra que beaucoup de détails sont sans importance : que son labyrinthe soit un labyrinthe végétal dans les jardins d'un château ou un labyrinthe dessiné dans la rubrique jeux d'un journal, les algorithmes permettant de trouver la sortie sont les mêmes, et la mise en œuvre de ces algorithmes demande de gommer ces détails pour se focaliser sur les éléments pertinents : la liste des carrefours et la liste des couloirs reliant ces carrefours. Il apprend ainsi à abstraire une situation en gommant les détails non pertinents, à retrouver des schèmes et des invariants dans des situations *a priori* différentes.

- *Projeter* : un élève qui écrit un programme doit imaginer ce que ce programme fera avant que cet objet n'existe, et évaluer la difficulté de sa réalisation afin de se fixer des objectifs ambitieux, mais réalistes. Au cours de cette réalisation, il devra réviser ses objectifs, souvent à la baisse, car il rencontrera des obstacles imprévus, parfois à la hausse, car la réalisation elle-même suscitera de nouvelles idées.

- *Réaliser* : pour atteindre ces objectifs, il devra décomposer son projet en parties – par exemple un programme en fonctions – ayant chacune ses objectifs propres, réaliser chaque composant, le valider, par exemple par une batterie de tests, identifier ses erreurs et les corriger, avant de valider ce composant à nouveau. Valider l'ensemble du projet enfin. Parce que les objets construits sont immatériels, l'informatique est la technique où la boucle de test et de correction d'erreurs est la plus courte et la plus facile à mettre en œuvre. L'erreur en informatique n'est, en outre, pas une faute qui dévalorise l'élève, mais une composante essentielle du processus de création.

- *Travailler en groupe* : si les premiers exercices d'un élève sont souvent des activités individuelles, dès les premiers mois de l'apprentissage de l'informatique, les élèves réalisent des projets en groupe, souvent à deux. Cela leur demande de négocier les objectifs et la répartition des tâches, d'expliquer leur travail et de comprendre le travail de leurs camarades.

- *Interagir avec un objet matériel* : un élève n'écrit jamais un programme pour lui seul, ou pour son professeur, mais avant tout pour que ce programme soit exécuté par une machine, c'est-à-dire un objet matériel. La faculté d'interagir avec un objet matériel est une clé essentielle d'entrée dans le monde informatisé dans lequel nous vivons. L'apprentissage de l'informatique permet de comprendre les différences entre l'interaction avec une personne et l'interaction avec une machine, que l'on peut résumer par le fait que l'interaction avec une machine comporte une part moindre d'implicite. Cela implique, par exemple, que l'on doive utiliser un langage et non une langue pour interagir avec une machine.

Beaucoup de ces compétences sont transverses mais l'informatique les décline de manière spécifique, en particulier parce qu'elle réconcilie le langage et l'action, catégories naguère – et parfois encore – perçues comme antithétiques.

La réalisation de projets est l'occasion de travail en coopération avec les enseignants de toutes les autres disciplines – aussi bien les sciences de la vie et de la Terre que le français ou les langues. De manière plus générale, l'informatique permet de renouveler l'enseignement de plusieurs autres disciplines, grâce aux outils matériels, mais surtout conceptuels – notamment de modélisation – qu'elle propose.

Les deux phases de l'apprentissage de la programmation

L'apprentissage de la programmation est à la fois une acquisition de connaissances et une acquisition de savoir-faire. De ce fait, l'apprentissage de la programmation demande d'accorder une place importante aux activités pratiques, reflétant ainsi un élément essentiel de la nature de l'informatique, qui est simultanément une science et une technique.

On commence à apprendre à programmer en apprenant les rudiments d'un langage de programmation, limités par exemple aux notions de

- affectation
- séquence
- test
- boucle
- tableau

bien qu'il soit aussi possible d'introduire des notions plus avancées, telles les fonctions, les structures de données dynamiques ou les objets. Cet apprentissage doit être illustré par des

exercices d'application courts, visant typiquement à l'écriture de programmes d'une dizaine de lignes. À des fins d'illustration, quelques exemples sont donnés en annexe.

Dans une seconde phase on apprend à utiliser ces notions dans la construction de programmes plus conséquents, typiquement, au Collège, de plusieurs dizaines de lignes. Cet apprentissage s'appuie sur des exercices qui se déroulent sur des temps plus longs, plusieurs semaines, et en petits groupes, typiquement de deux élèves, travaillant sur des exercices différents. Ces exercices sont souvent appelés des « projets ». Une partie importante de la tâche demandée aux élèves est de définir en partie eux-mêmes le sujet de leur projet, souvent à partir du développement d'un exercice court. Cela leur demande d'apprendre de nouveaux savoir-faire : s'organiser à plusieurs, sur le temps long, en redéfinissant périodiquement ses objectifs. Cette méthode d'enseignement par projet, qui est déjà en place en sciences de la vie et de la Terre, en sciences physiques et en technologie au Collège, peut être vue comme une initiation aux méthodes de la recherche. Dans un tel enseignement, l'enseignant a le rôle central de guider les élèves, de répondre à leurs questions et de les accompagner dans leur recherche, sans brider leur créativité.

Les capacités que les élèves doivent acquérir au cours de ces deux phases sont différentes. Dans la première phase, le but est essentiellement qu'ils sachent mobiliser leurs connaissances sur les langages de programmation pour résoudre des problèmes simples : qu'ils sachent écrire un test, une boucle, etc. ainsi que l'identification des fonctionnalités de ces langages utiles pour résoudre un problème donné : par exemple utiliser deux boucles imbriquées pour réaliser un calendrier. Dans la seconde phase, le but est qu'ils apprennent à organiser des programmes de plus grande taille, à les tester, à prendre en compte les besoins de leurs utilisateurs, à définir des objectifs, à réviser ces objectifs au cours du déroulement du projet, à travailler en groupe et à présenter leur travail par écrit et par oral.

Une évaluation spécifique de ce nouvel enseignement, bien entendu distincte du B2i, doit être mise en place et faire l'objet d'une mention explicite sur le Diplôme National du Brevet. Cette évaluation peut prendre soit la forme d'une épreuve écrite ou orale individuelle, soit celle d'un travail collectif sur un projet conduit en cours de formation.

D'autres aspects de l'informatique

Si l'apprentissage de la programmation nous semble devoir constituer le cœur de l'enseignement de l'informatique au Collège, il ne doit pas en constituer la totalité et il nous semble important d'aborder aussi, au Collège, d'autres domaines.

1. Tout d'abord, si certains systèmes informatiques sont autonomes, par exemple un ordinateur de bureau, d'autres sont utilisés au sein de systèmes plus complexes, avions, trains, voitures, machines-outils, etc., qui articulent des composants mécaniques avec des composants informatiques. Dès le Collège, il est possible de faire prendre conscience aux élèves de cette ouverture de l'informatique vers d'autres domaines techniques en leur faisant programmer un petit robot. Les imprimantes d'objets tridimensionnels et les machines-outils à commande numérique, qui permettent de programmer, non des mouvements, mais des objets, illustrent un autre aspect de ces liens entre l'informatique et les autres domaines techniques.

2. Le Collège est aussi le lieu où peuvent être abordés les réseaux, et en particulier les notions

d'adressage et de routage. Alors qu'à l'école primaire, il était simplement possible de s'interroger sur la manière dont un paquet trouve son chemin sur un réseau et de donner des réponses schématiques, il devient possible au Collège d'introduire les notions de routeur, de table de routage et des algorithmes de routage et de mise à jour des tables de routage. Cette initiation à la notion de réseau est bien entendu un moment privilégié pour introduire des notions de sécurité, de vie privée ou de propriété que leur développement induit.

3. Le Collège est aussi le lieu où les élèves peuvent commencer à prendre conscience de l'importance des données dans notre monde, à apprendre à mesurer des ordres de grandeur, à comprendre la persistance de ces données, etc. Cela peut se faire par la réalisation d'une base de données ou d'un site web, structurant une partie des connaissances acquises par les élèves au Collège.

4. C'est au Collège que les élèves peuvent commencer à réfléchir aux modifications profondes que l'informatique apporte à notre société. Par exemple, les adolescents savent très tôt télécharger des pièces de musique. Au-delà de la question de la légalité de ces pratiques, il est possible de les mener à s'interroger sur la raison pour laquelle cette question de la légalité se pose pour un fichier mp3, mais non pour un disque vinyle ou même un disque compact. La réponse à cette question peut les mener à comprendre que la non-rivalité est un aspect essentiel des biens immatériels. L'évolution des pratiques, mais aussi des normes et même du concept de propriété privée est déterminée par l'évolution des propriétés techniques des objets échangés. L'étude des différents types de licences logicielles, des formats de données et de leurs propriétés peut également être menée dans ce cadre.

- Serge Abiteboul, Professeur au Collège de France (2012), membre de l'Académie des sciences, membre du Conseil National du Numérique.
- Jean-Pierre Archambault, Président de l'association Enseignement Public et Informatique (EPI).
- Gérard Berry, Professeur au Collège de France, membre de l'Académie des sciences et de l'Académie des technologies.
- Colin de la Higuera, Président de la Société Informatique de France (SIF).
- Gilles Dowek, Directeur de recherche à l'INRIA, Grand Prix de philosophie de l'Académie Française.
- Maurice Nivat, membre de l'Académie des sciences.

Le 23 janvier 2014

Annexe

Cette liste, qui n'a aucune vocation à l'exhaustivité, donne quelques exemples d'exercices d'initiation à la programmation

- calculs de calendrier (calcul du calendrier d'une année donnée, jour d'une date, etc.),
- transformations d'images (changement de couleurs, de contraste, etc.),
- calculs de pièces de musique (note à note, par transposition, ...)
- calculs de tables de conjugaison (par groupes de verbes),
- calculs de pluriels (en tenant compte des mots « souris », « cheval », « chacal », etc.),
- calculs de traitement du texte (rechercher, rechercher et remplacer, etc.),

- calculs formels de fractions (multiplication, addition, etc.),
- réalisation de sondages suivis de calculs statistiques (moyenne, médiane, etc.),
- jeux (bataille navale, nim, tic tac toe, mastermind, etc.),
- calculs de répertoire d'adresses (recherche, recherche inversée, etc),
- calculs de chiffrement (Morse, César, ROT13, leet speak, etc.),
- des recherches dans une liste triée ou non (par dichotomie, en séquence, aléatoirement),
- recherche de la sortie d'un labyrinthe,
- calculs de graphes (par exemple de confiance dans le graphe valué d'un réseau social).